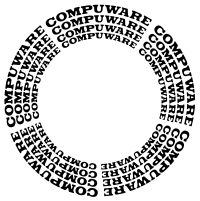
File-AID *for DB2*

Reference Manual

Release 4.8.1



COMPUWARE®

Please direct questions about File-AID *for DB2*
or comments on this document to:

File-AID *for DB2* Technical Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099
1-800-538-7822

Outside the USA and Canada, please contact
your local Compuware office or agent.

This document and the product referenced in it are subject to the following legends:

Copyright 1989-2005 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

DBA-XPERT, File-AID, and XPEDITER are registered trademarks of Compuware Corporation.

IBM, BookManager, DATABASE 2 (DB2), VMS/ESA, and MVS/XA are trademarks or registered trademarks of International Business Machines Corporation.

Adobe ® Acrobat ® Reader copyright © 1987-2005 Adobe Systems Incorporated. All rights reserved. Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

All other company and product names are trademarks or registered trademarks of their respective owners.

Doc. CWFDRX4G
October 3, 2005

Contents

Figures . . . . . **xv**

Tables . . . . . **xxiii**

Introduction . . . . . **xxv**
Operating Environment Support xxvi
Notation Rules xxvi
Reading the Syntax Diagrams xxvi
Related Publications xxviii
FrontLine Support Website xxviii
Online Documentation xxviii
World Wide Web xxviii
Getting Help xxviii

Chapter 1. Product Overview **1-1**
Product Capabilities 1-1
Creating Objects 1-1
Creating Test Tables 1-1
Customizing Data 1-2
Authorizing Objects 1-2
Reducing the DB2 Learning Curve 1-2
Reducing Test Time 1-2
Restoring Programmer Self-Sufficiency 1-2
Improving Testing 1-3
Prototyping SQL Code 1-3
Enhancing SQL Efficiency 1-3
Producing an Audit Trail 1-3
DBCS (Double-Byte Character Set) and Uppercase-Only Support 1-3
Product Functions 1-3
User Parameters 1-4
Browse 1-4
Edit 1-4
Utilities 1-5
Print Table Data or Audit Trail 1-6
Relationship Facilitator 1-6
XPEDITER ***for DB2*** Extension 1-6
File-AID/Related Data XPERT 1-6

Chapter 2. Product Conventions **2-1**
Accessing and Exiting File-AID *for DB2* 2-1
Subsystem ID Selection Window 2-2
DB2 Connection Type 2-2
Alternate Catalog Plan 2-2
DB2 Call Attach Alert 2-3
Invoking File-AID *for DB2* from Source Code Edit Session 2-4
Viewing DB2 Version 8 Long Names 2-4
Using the LEFT and RIGHT Commands 2-4
Using the EXPAND Command 2-5
Long Names Printed on Reports 2-5
Using ISPF Keylists 2-5
Common Screen Fields 2-5

Action Bar. 2-6
Pull-Downs 2-6
Windows. 2-7
Screen Title 2-7
COMMAND Field. 2-7
SCROLL Field 2-8
Subsystem Identification 2-8
CURRENT Date/Time 2-8
Wildcard Characters 2-9
Wildcard Character for PDS Member Fields 2-9
Wildcard Characters for Relationship Option. 2-9
Partial Column Names 2-9
Tutorials. 2-10

Chapter 3. Primary Commands 3-1
Description of Primary Commands 3-2
ADD 3-3
ALTER (ALT) 3-3
ATOMIC 3-3
BOTTOM (BOT) 3-3
BROWSE (BRO) 3-3
CANCEL (CAN) 3-4
CAPS. 3-4
CHANGE (C) 3-5
CHANGE Command Window 3-9
Timestamp Pop-up Window 3-11
Date Arithmetic and Increment Example 3-14
Time Pop-up Window 3-14
Date Pop-up Window. 3-15
COLCHECK 3-16
CREATE (CRE) 3-16
CUT (CU) 3-16
DEFAULT 3-17
DELETE (DEL). 3-17
DOWN 3-17
EDIT (ED) 3-18
END 3-18
EXCLUDE (X). 3-18
EXECUTE (EXEC). 3-19
FADB2. 3-19
FIND (F) 3-19
Using FIND During a Browse or Edit Session. 3-19
FIND Command Window 3-22
Using FIND in Relationship Object Lists 3-23
FLIP. 3-24
FOREIGN (FOR) 3-24
GRANT 3-24
GRAPHIC (GR) 3-25
HELP. 3-25
HEX 3-25
HIDE. 3-25
HIDE Command Window 3-26
HOLD 3-27
HOLD Command Window 3-27
IDENTITY 3-28
INFO. 3-29
Information Display Window 3-29

INSERT (INS) 3-30
JUMP (J) 3-30
KEYLIST 3-31
KEYS 3-31
LEFT 3-31
LOCATE (L) 3-32
    LOCATE in Browse or Edit 3-32
    LOCATE Command Window 3-32
    LOCATE in an Object List 3-33
MAX 3-33
MIN 3-33
MODIFY (MOD) 3-33
NAMCHECK 3-33
NEXT (N) 3-34
PARMS 3-34
PARTITN (PART) 3-34
PASTE (PAS) 3-34
    Paste Dataset 3-35
PATH 3-37
PREV (P) 3-37
PRIMARY (PRIM) 3-37
PRINTDS (PRTDS) 3-38
PRINTP (PRTP) 3-38
PROFILE (PRO) 3-39
    Profile Parameters Window 3-39
RBROWSE (RB) 3-41
RCHANGE 3-42
REDIT (RE) 3-42
REMARKS 3-43
REPEAT (REP) 3-43
    REPEAT Window 3-43
    Timestamp Pop-up Window 3-46
    Time Pop-up Window 3-47
    Date Pop-up Window 3-48
RESET (RES) 3-48
    RESET Command Window 3-49
RETPARM 3-50
RETRIEVE 3-50
RETURN 3-50
REVOKE (REV) 3-50
RFIND 3-50
RI 3-50
RIGHT 3-51
ROW 3-51
ROWH 3-52
RUNOPTS 3-52
SAVE 3-52
SORT 3-52
    SORT in Browse or Edit 3-52
    SORT Command Window 3-53
    SORT in an Object List 3-54
SPLIT 3-55
SQL 3-55
SQLID 3-55
SRCPARM 3-56
SWAP 3-56

TABLE (TAB) 3-56
TOP. 3-57
UDFOPTS 3-57
UDT 3-57
UP. 3-59
VALUES. 3-59
WHENCOND 3-59
WRAP 3-60
ZOOM 3-60
ZOOMH 3-60

Chapter 4. Line Commands 4-1
Description of Line Commands 4-2
A (After) 4-2
B (Before) 4-2
BR (Browse) 4-2
C (Copy) 4-3
C or CC as Line Markers for CUT and CHANGE Primary Commands . . 4-3
D (Delete) 4-3
ED (Edit) 4-3
EX (Execute) 4-4
I (Insert) 4-4
M (Move) 4-4
R (Repeat) 4-4
RI (Referential Integrity) 4-5
S (Select) 4-5
S (Show) 4-5
X (Exclude) 4-6
XP (Explain) 4-6

Chapter 5. Defaults and User Parameters 5-1
Option 1, Browse/Edit/SQL Analysis Display Options 5-2
Option 2, SQL Dataset Allocation 5-5
Allocate SHOW SQL Dataset 5-5
Option 3, Batch Options 5-6
Option 5, Index, Tablespace, and Database Default Values 5-7
Option 6, Logging Options 5-9
Option 7, Print Options 5-10
Option 8, Current Path Options 5-11
Option 9, Trigger Terminator 5-11

Chapter 6. Browse and Edit 6-1
Editing Considerations 6-1
Obscured Columns 6-2
Column Snap when Scrolling 6-2
Pending State 6-2
Maximum Rows to Select 6-3
Locking a Table During Edit 6-3
Method of Updating DB2 6-3
Editing Primary Column in a View 6-4
Columns with Defined Default Values 6-4
Starting a Browse or Edit Session 6-4
Location Selection Window 6-7
Selection Criteria Member List 6-8
Table Selection Window 6-9
Edit Options 6-9

Edit Selection Template 6-9
Selection Template Not Prefilled 6-11
SQL Entry Panel 6-12
Edit SPUFI Selection 6-13
Display Modes 6-14
Table Mode 6-14
Row Mode 6-17
Edit End Confirmation 6-19
Concurrent Update Intercept 6-20
Check Constraint Error 6-21
Editing Multiple Unrelated Tables 6-22
Viewing Multiple Tables 6-24
Manipulating Multiple Tables 6-25
Editing Multiple Tables in Row Mode 6-26
Referential Integrity Errors 6-27
RI Browse Window 6-29
Resolving RI Errors in Table Mode 6-29
Resolving RI Errors in Row Mode 6-30
Resolving Multiple Errors 6-31
Creating an Audit Trail Dataset 6-31
SMF Audit Trail Option 6-32
Audit Trail Print Facility 6-32
Audit Trail Dataset Usage and Maintenance 6-33

Chapter 7. Related Table Browse and Edit 7-1
UDT Columns, Triggers and LOB Columns 7-1
Referential Integrity 7-1
Application Relationship 7-2
Relationship File 7-3
Relationship File Compatibility Notes 7-3
Relationship File Characteristics 7-4
Creating, Loading, and Maintaining the Relationship File 7-4
Determining Relationships 7-4
Using an RI Relationship 7-5
Using an Application Relationship 7-5
Accessing Related Table Edit 7-7
Parent to Dependent 7-8
Dependent to Parent 7-9
Editing Requirements - Related Dependent Tables 7-9
Window Display 7-9
Table Requirements 7-10
Use of Views 7-11
Using Related Table Edit 7-11
One-to-One or One-to-Many Table Relationship 7-12
Selecting a Parent Relationship 7-14
Model Relationship 7-16
No Relationships 7-18
Creating a New Relationship 7-19
Building Column Relationships 7-20
Extended Column Relationship Definition 7-23
Extended Column Relationships Examples 7-24
Using Additional Windows 7-26
Specifying a New Creator 7-27
Relationship Detail Window 7-28
Relationship Information Window 7-29
Application Relationship (AR) Information 7-29
Referential Integrity (RI) Information 7-30

Manipulating Windows 7-30
Navigating Through Multiple Windows 7-31
Scrolling Windows 7-32
Sizing Windows 7-33
Using Row Mode 7-34
Using Zoom Mode 7-35
Opening Additional Related Table Paths 7-36
Switching To Other Paths 7-38
Closing Individual Windows and Paths 7-39
Updating Data 7-39
Saving Changes 7-39
Inserting Data 7-40
Deleting Data 7-41
Changing Parent Record 7-42
Application Relationships 7-42
DB2 RI Relationships 7-42
Graphic Relationship Display 7-42
Receiving an Option List 7-43

Chapter 8. Objects Utility 8-1
Objects - Create, Drop, Alter 8-1
Option 1, Tables 8-2
Table Create 8-3
Model Table Selection 8-7
Model Table Area 8-8
Distinct Type Selection 8-9
Primary Key Definition 8-9
Foreign Key Definition 8-10
Default Values 8-12
Column Check Constraints 8-13
Named Check Constraints 8-14
Identity Column 8-15
Implicit Tablespace Confirmation 8-17
Index Create 8-18
Table Alter 8-20
Creating or Modifying Keys 8-22
Defining or Modifying Default Values or Column Constraints 8-22
Confirm Drop 8-22
Rename Table 8-23
Option 2, Tablespaces 8-24
Tablespace Create 8-25
Tablespace Create 8-25
Partitioned Tablespace Create 8-28
Tablespace Alter 8-29
Partitioned Tablespace Alter 8-31
Confirm Drop 8-33
Option 3, Indexes 8-33
Index Create 8-34
Index Create Values 8-36
Index Create Partition 8-38
Index Alter 8-39
Index Alter for Nonpartitioned Indexes 8-39
Index Alter for Partitioned Indexes 8-41
Confirm Drop 8-43
Option 4, Views 8-43
View Create 8-44
Confirm Drop 8-45

Option 5, Synonyms 8-45
Modify Synonyms 8-46
Synonym Create 8-46
Confirm Drop 8-47
Option 6, Aliases 8-47
Modify Aliases 8-48
Alias Create 8-49
Confirm Drop 8-50
Option 7, Distinct Types 8-50
Distinct Type Create 8-51
Confirm Drop 8-52
Option 8, User Defined Functions 8-52
Create External Scalar 8-53
Input Parameters 8-53
Return Parameter 8-56
Create External Table 8-57
Input Parameters 8-57
Return Parameters 8-58
Create Sourced Function 8-59
Input Parameters 8-59
Return Parameter 8-61
Source Parameters 8-62
Parameters 8-63
Function Options 8-63
Run Time Options 8-67
Comment On 8-67
View/Edit DDL 8-68
Execute Create 8-68
Alter Function 8-68
Parameter Information 8-70
Confirm Drop 8-74
Option 9, Stored Procedures 8-74
Stored Procedure Create 8-75
Parameters 8-78
Run Time Options 8-81
Comment On 8-82
View/Edit DDL 8-82
Execute Create 8-83
Alter 8-83
Confirm Drop 8-84
Option 10, Triggers 8-85
Trigger Create 8-86
Update Column 8-88
WHEN Search Condition 8-89
Begin Atomic...End 8-90
Comment On 8-90
View/Edit DDL 8-91
Execute Create 8-92
Confirm Drop 8-92

Chapter 9. Copy Utility 9-1
Table Copy Selection 9-1
Copy From List of Tables 9-3
Copy To List of Tables 9-4
Auto Create Window 9-4
Trigger Warning 9-5
Copy Column Selection 9-6

Copy Row Selection 9-7
SQL Entry Panel. 9-8
Edit SPUFI Selection 9-9

Chapter 10. Object List Utility 10-1
Access Object List Utility. 10-1
Browse Object 10-3
Edit Object. 10-3
Drop Object 10-3
Alter Object 10-3
Grant Privileges on Object 10-4
Revoke Privileges on Object 10-4
Generate DDL Statements. 10-4
Load Table 10-5
Extract from Table. 10-5
Display Privileges 10-5
Privilege Modify 10-6
Object Information 10-6
Table 10-6
Tablespace 10-10
Index 10-11
View 10-14
Plan 10-14
DBRM 10-16
Synonym 10-17
Alias 10-18
Package 10-18
Auxiliary Table 10-22
Trigger 10-22
Trigger Text. 10-24
Comment On 10-24
Distinct Type Information 10-25
Stored Procedure 10-25
Parameter Information. 10-29
User Defined Function. 10-31
UDF Detail 10-31
Performance 10-35
Java 10-36
Parameter Information. 10-36

Chapter 11. SQL Command Manager Utility 11-1
SPUFI 11-1

Chapter 12. Privileges Utility 12-1
Privileges - General 12-1
Collection Privileges 12-3
Display Privileges for Collection 12-3
Grant Privileges for Collection 12-4
Revoke Privileges for Collection 12-5
Modify Privileges for Collection 12-5
Package Privileges 12-6
Display Privileges for Package 12-6
Grant Privileges for Package 12-6
Revoke Privileges for Package 12-7
Modify Privileges for Package 12-7
Plan Privileges 12-8
Display Privileges for Plan 12-8

Grant Privileges for Plan 12-8
Revoke Privileges for Plan 12-9
Modify Privileges for Plan 12-10
Table/View Privileges 12-10
Display Privileges for Table/View 12-10
Grant Privileges for Table/View 12-12
Revoke Privileges for Table/View 12-14
Modify Privileges for Table/View 12-16
Tablespace Privileges 12-16
Display Privileges for Tablespace 12-16
Grant Privileges for Tablespace 12-17
Revoke Privileges for Tablespace 12-18
Modify Privileges for Tablespace 12-18
Schema Privileges 12-19
Display Privileges for Schema 12-19
Grant Privileges for Schema 12-20
Revoke Privileges for Schema 12-21
Modify Privileges for Schema 12-21
Stored Procedure Privileges 12-21
Display Privileges for Stored Procedures 12-21
Grant Privileges for Stored Procedures 12-22
Revoke Privileges for Stored Procedures 12-23
Modify Privileges for Stored Procedures 12-23
Distinct Type Privileges 12-23
Display Privileges for Distinct Type 12-23
Grant Privileges for Distinct Type 12-24
Revoke Privileges for Distinct Type 12-25
Modify Privileges for Distinct Type 12-25
User Defined Function (UDF) Privileges 12-25
Display Privileges for UDF 12-25
Grant Privileges for UDF 12-26
Revoke Privileges for UDF 12-27
Modify Privileges for UDF 12-27

Chapter 13. Impact Analysis Utility 13-1
Column Impact Analysis 13-1
Impact Analysis Tables 13-3
Create Impact Analysis Tables 13-3
Impact Selection Template 13-4
Batch Output Specification 13-5
Column Impact Analysis Report Example 13-5

Chapter 14. Extract and Load Utility 14-1
UDT Columns, Triggers and LOB Columns 14-1
Extract/Load Menu 14-2
Option 1 - Extract from a Table to a Dataset 14-2
Online Processing 14-2
Selection Criteria Member List 14-5
Extract Table Selection 14-6
EXTRACT Selection 14-6
Selection Template Not Prefilled 14-8
SQL Entry 14-9
Edit SPUFI Select 14-10
Extract Table 14-10
Delimited Format Options 14-12
Extract Dataset Allocation 14-13
Control Card Dataset Allocation 14-15

Record Layout Allocation 14-16
Extract Confirmation 14-17
Batch Processing 14-18
Batch Table Extract to a Dataset 14-18
Batch Table Extract to Tape 14-19
Option 2 - Load from a Dataset to a Table 14-19
Load Options 14-19
Load Dataset Specifications 14-20
Online Processing 14-21
SQL LOAD Table 14-21
Discard Dataset Allocation 14-24
Load Confirmation 14-25
Online Load Summary 14-26
Detailed Load Report 14-27
Batch Processing 14-29
Load Option 14-29
DB2 Batch Load Table 14-29
Generating the DB2 Batch Load 14-30
Option 3 - Extract and Load 14-31

Chapter 15. Batch Facility 15-1
Create Trigger Statement 15-1
Accessing the Batch Facility 15-2
Batch Execute SQL 15-2
Specify SQL Dataset 15-3
Specify Processing Options 15-3
Specify Print Options for SELECT Statements 15-4
Editing SQL 15-4
Batch Facility Output Specification 15-4

Chapter 16. SQL Development and Analysis 16-1
Toleration of DB2 Version 8 SQL Clause or Syntax 16-1
Accessing the SQL Analysis Function 16-2
Option 0, Create or Maintain Explain Table(s) 16-2
1, Create PLAN_TABLE, DSN_STATEMNT_TABLE, DSN_FUNCTION_TABLE . 16-3
2, Alter or Upgrade PLAN_TABLE 16-3
Upgrade Plan Table 16-4
Drop and Recreate Plan Table and F2 Plan Table 16-4
3, Delete Rows from Explain Table 16-4
Row Deletion 16-6
Maintenance Detail 16-6
Option 1, SQL Plan Analysis 16-7
SQL Plan Analysis Specification 16-8
Use Filters Window 16-11
SQL Plan Selection 16-11
Sorting Plan Entries 16-12
DBRM Selection 16-13
Sorting DBRM Entries 16-13
Collection ID Selection 16-14
Package Selection 16-15
Sorting Package Entries 16-15
DBRM/Package Selection 16-16
Sorting DBRMs or Packages 16-17
SQL Statement Selection 16-17
SQL Analysis Summary 16-17
Sorting DBRM/Package Entries 16-19

Plan Table Analysis 16-19
Access Path Analysis 16-20
Explain Detail 16-23
Predicate Analysis 16-25
Catalog Information 16-28
Cost Detail 16-35
Function Detail 16-35
Explaining Objects in Batch 16-36
SQL Analysis Batch Output Specification 16-37
Explaining Objects in Batch via a Source Listing File 16-37
DD Statements 16-38
JCL Input 16-39
Parameters 16-39
Sample Explain JCL 16-41
Option 2, Review or Print DB2 Explain Reports 16-41
Changing Report Format Skeletons 16-42
F2PTROW 16-42
Option 3, ISPF/PDF SQL Source Development and Analysis 16-44
Using the SQL Analysis Line Commands 16-46
Entering Host Variable Values 16-47
Prototyping New SQL Statements 16-49
Prototyping Cursor Processing 16-50
Browsing the Results Table 16-53
Editing the Results Table 16-54
Dynamically Changing and Testing SQL Statements 16-54
Explaining SQL Statements 16-56
Resolving SQL Coding Errors 16-57
Unqualified Objects Window 16-58
Modifying Column Boundaries 16-58
COMMIT/ROLLBACK Prompt 16-58
Option L, LIBRARIAN Source Development and Analysis 16-59
Option P, PANVALET Source Development and Analysis 16-59

Chapter 17. Print Table Data or Audit Trail 17-1
Print Menu 17-1
Printing Table Data 17-1
Print Options Screen 17-1
Selection Criteria Member List 17-3
Table Selection 17-4
PRINTP Selection Template 17-5
Selection Template Not Prefilled 17-7
SQL Entry Panel 17-7
Edit SPUFI Selection Screen 17-8
Table Print Output Specification Screen 17-9
Sample Row Mode Report 17-11
Sample Table Mode Report 17-12
Printing an Audit Trail Report 17-13
Print Audit Trail Specification Screen 17-13
Print Audit Trail Selection Criteria Screen 17-14
Sample Audit Trail Report 17-15

Chapter 18. Relationship Facilitator 18-1
Overview 18-1
Relationship File Compatibility Notes 18-1
Relationship File Contents 18-1
DB2 Referential Integrity (RI) relationships 18-1
Application Relationships (AR) 18-2

Considerations for MVS Objects 18-2
Use of DB2 Views 18-2
DB2 Relationship Facilitator 18-3
DB2 Object List 18-5
Table List in Relationship File 18-8
Table List - SYSIBM.SYSTABLES 18-9
MVS Relationship Facilitator. 18-10
MVS Object List 18-12
Object List in Relationship File 18-14
Relationship Detail 18-15
Graphical Relationship Display 18-18
Understanding the Graphical Relationship Display 18-18
Option List 18-19
Relationship Information Screen. 18-20
RI Relationship Detail Screen 18-25
RI Information Screen 18-28
Application Relationship Detail 18-31
Relationship Information 18-33
Add Relationships — Specify Object Types 18-34
Add Application Relationships — DB2 to DB2 18-35
Add Relationships — DB2 to MVS. 18-37
Add Relationships — DB2 to IMS 18-40
Add Relationships — MVS to DB2. 18-42
Add Relationships — MVS to MVS 18-44
Add Relationships — MVS to IMS 18-47
Relationships 18-50
Building Column/Field Relationships 18-56
Extended Column Relationship Definition 18-58
Rules for DECIMAL and CHARACTER Columns/Fields 18-59
Extended Column Relationships Examples. 18-60
Example 1 18-60
Example 2 18-61
Specify MVS Field Subscripts 18-65
Print Relationship File 18-68
Formatted Relationship Report Screen 18-68
Print Options Screen 18-69
Edit Print JCL 18-70
Formatted Relationship Report 18-71

Appendix A. Predicate Analysis Rules Abbreviations A-1

Appendix B. Action Bar and Pull-Downs Summary B-1
Edit Action Bar Items and Pull-Downs B-1
File Pull-Down B-1
Edit Pull-Down. B-2
View Pull-Down B-3
Options Pull-Down B-4
Open Pull-Down B-5
Menu Pull-Down B-5
Utilities Pull-Down. B-6
Help Pull-Down B-8
Objects Utility Action Bar Items and Pull-Downs B-8
Define Pull-Down B-9
Actions Pull-Down B-10

Index I-1

Figures

1-1. File-AID *for DB2* Primary Option Menu 1-4
2-1. File-AID *for DB2* Primary Option Menu 2-1
2-2. Subsystem ID Selection Window 2-2
2-3. Alternate Catalog Plan Window 2-3
2-4. DB2 Call Attach Alert 2-3
2-5. Table Print Report With Long Name 2-5
2-6. Views Pull-down Menu Example 2-6
2-7. Sort Command Window 2-7
2-8. File-AID *for DB2* Screen Header with Action Bar 2-7
2-9. File-AID *for DB2* Screen Header with SSID — Updates Allowed 2-8
2-10. Column Selection window 2-10
2-11. Tutorial Table of Contents 2-10
3-1. Column Selection window 3-8
3-2. CHANGE Command window 3-9
3-3. CHANGE Command Window with Timestamp Increment 3-12
3-4. CHANGE Command WIndow with Time Increment 3-15
3-5. CHANGE Command window with Date Increment 3-16
3-6. FIND Command Window 3-23
3-7. Enter Find Parameter window 3-24
3-8. HIDE Command window 3-26
3-9. HOLD Command Window 3-28
3-10. Information Display Window 3-29
3-11. LOCATE Command window 3-32
3-12. PASTE Dataset screen 3-35
3-13. PASTE Column Selection window 3-36
3-14. Profile Parameters Window 3-39
3-15. REPEAT Column Range Specification Window 3-44
3-16. REPEAT Generate - Timestamp Specification Window 3-46
3-17. REPEAT Generate - Time Specification Window 3-47
3-18. REPEAT Generate - Date Specification Window 3-48
3-19. RESET window in Edit 3-49
3-20. SORT Command Window 3-54
3-21. SQLID Pop-Up 3-56
3-22. Distinct Data Type Definition window 3-57
3-23. Distinct Type Information pop-up 3-58
3-24. Autocast Pop-up 3-58
5-1. User Parms Menu 5-1
5-2. Browse/Edit/SQL Analysis Display Options screen (Part 1) 5-2
5-3. Browse/Edit/SQL Analysis Display Options screen (Part 2) 5-3
5-4. Dataset Allocation screen 5-5
5-5. Allocate Show SQL Dataset screen 5-5
5-6. Batch Options screen 5-6
5-7. Object Creation Default Values screen 5-7
5-8. Logging Options screen 5-9
5-9. Print Options screen 5-10
5-10. Current Path screen 5-11
5-11. Trigger Terminator screen 5-12
6-1. Edit Options screen 6-5
6-2. Location Selection Window 6-8
6-3. Member Selection Criteria List screen 6-8
6-4. Edit Table Selection Window 6-9
6-5. Edit Selection Template 6-10
6-6. Selection Template Not Prefilled 6-12
6-7. SQL Entry Panel 6-13
6-8. Edit SPUFI Selection screen 6-13
6-9. SQL Entry Panel 6-14

6-10. Edit screen in Table Mode 6-15
6-11. Edit screen in Row Mode 6-18
6-12. Edit End Confirmation screen 6-19
6-13. Concurrent Update Intercept Window 6-20
6-14. Check Constraint Error Window 6-21
6-15. Check Constraint Error Rows Marked 6-22
6-16. Using the EDIT Primary Command (from Open Pull-down) 6-23
6-17. Edit Options Window — After Entering EDIT 6-23
6-18. Multiple Unrelated Edit Display 6-25
6-19. Issuing the JUMP Primary Command 6-26
6-20. New Display — After issuing the JUMP Primary Command 6-26
6-21. Issuing the ROW Primary Command 6-27
6-22. Window 2 in Row Mode 6-27
6-23. Inserting a Row into the Employee Table 6-28
6-24. RI Relationship Error Window 6-28
6-25. RI Browse Window — Using the RI Line Command 6-29
6-26. RI Error Resolved — Table Mode 6-30
6-27. RI Browse Window — Entering the RI Primary Command 6-30
6-28. RI Error Resolved — Row Mode 6-31
6-29. Create Audit Trail Field on Edit Options Screen 6-32
6-30. Audit Trail Print Facility screen 6-32
6-31. Sample Audit Trail Report 6-33
7-1. Referential Integrity 7-2
7-2. Application Relationship 7-3
7-3. Diagram of Determining Relationships 7-6
7-4. Edit Options screen 7-7
7-5. REDIT Primary Command 7-8
7-6. Related Table Edit Session - Dependent Table 7-8
7-7. Related Table Edit Session - Parent Table 7-9
7-8. Related Table Edit Session with Dependent Tables 7-10
7-9. Enter REDIT Primary Command 7-13
7-10. Related Table List WIndow 7-13
7-11. Related Table Edit Display — EMP to PROJ Table 7-14
7-12. Related Table List Window 7-14
7-13. REDIT Parent Selection Template 7-15
7-14. Related Table Edit Display — Dependent to Parent 7-16
7-15. Related Table Edit Display — DEPT to EMP Tables 7-17
7-16. Relationship Listing Window 7-17
7-17. Related Table Edit Display — after Selecting a Model Relationship 7-18
7-18. No Relationships Exist Window 7-19
7-19. Add Application Relationships screen Invoked by ADD Command 7-20
7-20. Column Relationships screen 7-20
7-21. Column Relationships screen—Creating a new column relationship 7-21
7-22. Extended Column Relationship Definition Window 7-22
7-23. Column Relationships screen — New Column Relationship Established 7-22
7-24. Related Table Edit Display — DEPT to EMP Tables 7-23
7-25. Extended Column Relationship Definition Window 7-23
7-26. Column Relationships screen with column-number command 7-25
7-27. Extended Column Relationship Definition Window 7-25
7-28. Column Relationships screen with column-number command 7-26
7-29. Extended Column Relationship Definition Window 7-26
7-30. Table Not Found Window 7-27
7-31. Creator Selection Window 7-28
7-32. Related Table Edit Display After Selecting New Creator 7-28
7-33. Relationship Detail Window 7-29
7-34. AR Relationship Information 7-30
7-35. RI Relationship Information Window 7-30
7-36. JUMP Command used to Edit Table Number 2 7-31
7-37. JUMP Command — Window 2 Now Active 7-32
7-38. DOWN Primary Command 7-32
7-39. New Display after Scrolling Down 7-33

7-40. MAX Primary Command....7-33
7-41. Related Table Edit DIsplay after the MAX Primary Command....7-34
7-42. ROW Primary Command....7-34
7-43. Window 2 in Row Mode....7-35
7-44. ZOOM Primary Command....7-35
7-45. Maximized Zoom Mode....7-36
7-46. Related Table Edit Display — EMP to PROJ Table....7-36
7-47. Related Table List WIndow....7-37
7-48. Related Table Edit — New Path EMP to DEPT....7-37
7-49. Referential Integrity....7-38
7-50. PATH Command used to switch to Path Number 2....7-39
7-51. Related Table Edit....7-40
7-52. Inserting a Row — Foreign Key Prefilled....7-41
7-53. Deleting a Row in Related Table Edit....7-41
7-54. Display Adjusted After a Deleted Row....7-42
7-55. Graphical Relationship Display....7-43
7-56. Option List for Table EMP....7-44
8-1. Utility Selection Menu....8-1
8-2. Object Selection Menu....8-2
8-3. Tables Options Screen....8-2
8-4. Table Create Screen....8-4
8-5. ROWID Datatype pop-up....8-7
8-6. Model Table Selection Window....8-8
8-7. Distinct Data Type Selection Window....8-9
8-8. Primary Key Definition Window....8-10
8-9. Foreign Key Definition Screen....8-11
8-10. Default Values Definition Window....8-13
8-11. Column Check Constraints Window....8-14
8-12. Column Check Constraints Window....8-15
8-13. Identity Column Definition Window....8-16
8-14. Identity Column Attributes Window....8-16
8-15. Implicit Tablespace Confirmation Screen....8-17
8-16. Index Create screen....8-18
8-17. Table Alter Screen....8-20
8-18. Confirm Drop Screen....8-23
8-19. Rename Table Screen....8-23
8-20. Tablespaces Option Screen....8-24
8-21. Tablespace Create Screen....8-25
8-22. Partitioned Tablespace Create Screen....8-28
8-23. Non-Partitioned Tablespace Alter Screen....8-29
8-24. Partitioned Tablespace Alter Screen....8-32
8-25. Indexes Option Screen....8-34
8-26. Index Create screen....8-35
8-27. Index Create Values screen....8-37
8-28. Index Create Partition Screen....8-38
8-29. Index Alter screen for Nonpartitioned Indexes....8-39
8-30. Index Alter screen for Partitioned Indexes....8-41
8-31. Views Option Screen....8-43
8-32. View Create Screen....8-44
8-33. Synonyms Option Screen....8-45
8-34. Modify Synonyms Screen....8-46
8-35. Synonym Create Screen....8-47
8-36. Aliases Option Screen....8-48
8-37. Modify Aliases Screen....8-49
8-38. Alias Create Screen....8-49
8-39. Distinct Type Option Screen....8-50
8-40. Distinct Type Create Screen....8-51
8-41. User Defined Function Screen....8-53
8-42. Function Create (external scalar) Screen....8-54
8-43. Function Create (external scalar) Popup....8-56
8-44. Function Create (external table) Screen - Input Parameters....8-57

8-45. Function Create (external table) Return Parameters 8-58
8-46. Function Create (sourced) 8-59
8-47. Function Create (sourced) Return Parameters Popup 8-61
8-48. Function Create (sourced) Source Parameters Popup 8-62
8-49. Create Function Options Screen 8-64
8-50. Run Time Options Window 8-67
8-51. Comment On Screen 8-68
8-52. Function Alter Selection List 8-69
8-53. Alter Functions Options 8-69
8-54. User Defined Function Information Screen 8-70
8-55. UDF Parameter Information Screen 8-71
8-56. Alter Function Options Screen 8-71
8-57. Stored Procedures Screen 8-74
8-58. Stored Procedure Create Screen 8-75
8-59. Parameter Definition window 8-79
8-60. Built-In Datatype Pop-up Sample 8-81
8-61. Run Time Options Window 8-82
8-62. Comment On Screen 8-82
8-63. DDL to Create Stored Procedure 8-83
8-64. Confirm Alter Screen 8-84
8-65. Stored Procedure Selection Screen 8-85
8-66. Triggers Screen 8-85
8-67. Trigger Create Screen 8-86
8-68. Update Column Selection List 8-88
8-69. WHEN Search Condition Screen 8-89
8-70. Begin Atomic...End Screen 8-90
8-71. Comment On Screen 8-91
8-72. DDL to Create Trigger 8-91
9-1. Table Copy Selection Screen 9-1
9-2. Copy From List of Tables 9-3
9-3. Copy To List of Tables 9-4
9-4. Auto Create Window 9-5
9-5. Trigger Warning Window 9-5
9-6. Copy Column Selection screen 9-6
9-7. Copy Row Selection screen 9-7
9-8. SQL Entry Panel 9-8
9-9. Edit SPUFI Selection screen 9-9
10-1. Object List Utility Screen 10-1
10-2. Object List Selection Screen 10-2
10-3. Table Information Display Screen 10-7
10-4. Index Information Display screen 10-12
10-5. PathSchemas Information for Package Screen 10-18
10-6. Auxiliary Table Information Display Screen 10-22
10-7. Trigger Information Display Screen 10-23
10-8. Trigger Text Screen 10-24
10-9. Comment On Screen 10-25
10-10. Distinct Type Information window 10-25
10-11. Stored Procedure Information Screen 10-26
10-12. Parameter Information Screen 10-29
10-13. UDF Information Screen 10-31
10-14. Parameter Information Screen 10-36
11-1. SPUFI screen 11-1
12-1. Privileges Screen 12-1
12-2. Privileges pop-up window sample 12-2
12-3. Privilege Object Selection Screen 12-3
12-4. Privileges for Collection Screen 12-4
12-5. Grant for Collection Screen 12-5
12-6. Privileges for Package Screen 12-6
12-7. Grant for Package Screen 12-7
12-8. Privileges for Plan Screen 12-8
12-9. Grant for Plan Screen 12-9

12-10. Revoke for Plan Screen 12-10
12-11. Privileges for Table Screen 12-11
12-12. Privileges for View Screen 12-11
12-13. Grant for Table Screen 12-12
12-14. Grant for View Screen 12-13
12-15. Revoke for Table Screen 12-14
12-16. Revoke for View Screen 12-15
12-17. Privileges for Tablespace Screen 12-16
12-18. Grant for Tablespace Screen 12-17
12-19. Revoke for Tablespace Screen 12-18
12-20. Privileges for Schema Screen 12-19
12-21. Grant for Schema Screen 12-20
12-22. Privileges for Stored Procedure Screen 12-22
12-23. Grant for Stored Procedure Screen 12-22
12-24. Privileges for Distinct Type Screen 12-24
12-25. Grant for Distinct Type Screen 12-24
12-26. Privileges for UDF Screen 12-26
12-27. Grant for User Defined Function Screen 12-26
13-1. Column Impact Analysis Screen 13-2
13-2. Create Impact Analysis Table Screen 13-4
13-3. Impact Analysis Selection Template 13-4
13-4. Impact Analysis Output Specification Screen 13-5
13-5. Column Impact Analysis Report 13-5
14-1. Extract/Load Menu 14-2
14-2. Extract Options Screen 14-3
14-3. Selection Criteria - EXTRCRIT 14-5
14-4. Extract Table Selection Window 14-6
14-5. Extract Selection Template 14-7
14-6. Selection Template Not Prefilled 14-9
14-7. SQL Entry Panel (SPUFI) 14-10
14-8. Edit SPUFI Screen with Sample Select Statement 14-10
14-9. Extract Table Screen 14-11
14-10. Delimited Format Options Screen 14-13
14-11. Extract Dataset Allocation Screen 14-14
14-12. Control Card Dataset Allocation Screen 14-15
14-13. Record Layout Dataset Allocation Screen 14-16
14-14. Extract Confirmation Screen 14-18
14-15. Load Options Screen 14-20
14-16. Load Dataset Specifications Screen 14-21
14-17. Load Table Screen 14-22
14-18. ROWID pop-up for GENERATED ALWAYS 14-23
14-19. ROWID pop-up for GENERATED BY DEFAULT 14-24
14-20. DISCARD DATASET ALLOCATION Screen 14-24
14-21. Load Confirmation Screen 14-26
14-22. Load Confirmation Screen with Report Pop-Up 14-27
14-23. Detailed Load Report: Control Card Input 14-28
14-24. Detailed Load Report: Input Error Information 14-28
14-25. Detailed Load Report: Summary Window Information 14-29
14-26. Batch Load Table Screen 14-29
14-27. Extract and Load Options Screen 14-31
15-1. File-AID *for DB2* Primary Option Menu 15-2
15-2. Batch Execute SQL 15-3
15-3. Batch Facility Output Specification 15-5
15-4. Sample Batch Facility JCL (1 of 2) 15-6
15-5. Sample Batch Facility JCL (2 of 2) 15-7
16-1. SQL Clause or Syntax Not Supported 16-1
16-2. SQL Analysis and Development Menu 16-2
16-3. Create or Maintain Explain Table(s) Screen 16-3
16-4. Create Explain Table(s) screen 16-3
16-5. Alter Plan Table Screen 16-4
16-6. Upgrade Plan Table Screen 16-4

16-7. Explain Table(s) Maintenance Specification Screen....16-5
16-8. SQL Explain Tables Row Deletion Screen....16-6
16-9. Maintenance Detail Screen....16-7
16-10. SQL Plan Analysis Flow Chart — Online Processing Mode....16-8
16-11. SQL Plan Analysis Specification Screen....16-9
16-12. Filters Used to Select Object Window....16-11
16-13. SQL Plan Selection....16-12
16-14. DBRM Selection Screen....16-13
16-15. Collection ID Selection Screen....16-14
16-16. Package Selection Screen....16-15
16-17. DBRM/Package Selection Screen....16-16
16-18. SQL Statement Selection Screen....16-17
16-19. SQL Analysis Summary....16-18
16-20. Plan Table Analysis Screen....16-19
16-21. Access Path Analysis Screen....16-20
16-22. SQL Analysis Output Destination Specification Screen....16-22
16-23. SQL Source — After Executing the SQL Command....16-22
16-24. SQL Analysis Summary Screen — After a Re-explain....16-23
16-25. Explain Detail Screen....16-24
16-26. FIeld Explanation....16-25
16-27. SQL Predicate Analysis....16-26
16-28. SQL Predicate Analysis — Detail Recommendation Information....16-26
16-29. Edit session initiated after executing the SQL command....16-27
16-30. Predicate Analysis Reports Window....16-27
16-31. Object Selection Screen....16-28
16-32. Table Information screen....16-29
16-33. Column Information Screen....16-30
16-34. Column Information Screen — After Scrolling RIght....16-30
16-35. Column Information Screen — After Scrolling Right Twice....16-31
16-36. Column Partition Statistics Screen....16-31
16-37. Table Index Selection Screen....16-32
16-38. Index Information screen....16-32
16-39. Index Key Information....16-33
16-40. Index Partition Information Screen....16-33
16-41. Related Table Information Screen....16-34
16-42. Table Partition Statistics Screen....16-34
16-43. Check Constraints Screen....16-35
16-44. Cost Detail Information Screen....16-35
16-45. Function Detail Information Screen....16-36
16-46. Batch Plan Analysis Reports Window....16-36
16-47. SQL Analysis Batch Output Specification Screen....16-37
16-48. JCL to Explain via SLS....16-38
16-49. Sample JCL to Explain with SLS....16-41
16-50. SQL Explain Report Specification Screen....16-41
16-51. SQL Analysis Edit Entry Screen....16-45
16-52. Edit Screen....16-46
16-53. Executing a DECLARE CURSOR SQL Statement Dynamically....16-47
16-54. Entering Host Variable Values on the Edit Screen....16-48
16-55. Host Variable Resolution Screen Prompting for Values....16-48
16-56. Entering Values on the Host Variable Resolution Screen....16-49
16-57. Entering Host Variable Values on the Edit Screen....16-50
16-58. Message Indicating Successful Execution of the SQL Statement....16-51
16-59. Executing an OPEN CURSOR SQL....16-51
16-60. Executing an SQL FETCH....16-52
16-61. Display Showing SQL FETCH Statement and Table Data....16-52
16-62. Second Row Fetched Into Storage....16-53
16-63. Browsing the Results Table....16-54
16-64. Dynamically Changing an SQL Statement....16-55
16-65. Entering Host Variable Values on the Edit Screen....16-55
16-66. Explaining SQL Statements with the XP Line Command....16-56
16-67. Plan Table Analysis Screen Resulting from an XP Line Command....16-57

16-68. Error Was Encountered During Execution of the SQL Statement 16-57
16-69. Help Screen for an SQL Return Code of -206 16-58
16-70. Unqualified Objects WIndow 16-58
16-71. COMMIT/ROLLBACK Prompt 16-59
17-1. Print Menu Screen 17-1
17-2. Print Options Screen 17-2
17-3. Print Selection Criteria Member List 17-4
17-4. Print Table Selection Window 17-4
17-5. Print Selection Template 17-5
17-6. Print Selection Template Not Prefilled 17-7
17-7. SQL Entry Panel 17-8
17-8. Edit SPUFI Selection Screen 17-9
17-9. SQL Entry Panel 17-9
17-10. Table Print Output Specification 17-10
17-11. Example Row Mode Table Data Print to Dataset 17-11
17-12. Example Table Mode Table Data Print to Database 17-12
17-13. Print - Audit Trail Facility Screen 17-13
17-14. Print — Audit Trail Selection Criteria 17-14
17-15. Example Audit Trail Report 17-15
18-1. DB2 Relationship Facilitator 18-3
18-2. DB2 Object List 18-6
18-3. Table List in Relationship File 18-8
18-4. Table List - SYSIBM.SYSTABLES 18-9
18-5. MVS Relationship Facilitator 18-10
18-6. MVS Object List 18-13
18-7. Object List in Relationship File 18-14
18-8. Relationship Detail 18-15
18-9. Graphical Relationship Display Screen 18-18
18-10. Graphical Relationship Display with Option List 18-19
18-11. Relationship Information Screen—RI relationship information 18-21
18-12. Relationship Information Screen—MVS to DB2 Application Relationship Information 18-21
18-13. Relationship Information Screen—MVS to MVS AR Relationship Information 18-22
18-14. RI Relationship Detail Screen 18-26
18-15. RI Information Screen 18-28
18-16. Application Relationship Detail Screen 18-31
18-17. Object Type Selection Screen for Adding Application Relationship 18-34
18-18. Add Relationships Screen — DB2 to DB2 18-35
18-19. Add Relationships Screen — DB2 to MVS 18-37
18-20. Add Relationships Screen — DB2 to IMS 18-40
18-21. Add Relationships Screen — MVS to DB2 18-42
18-22. Add Relationships Screen — MVS to MVS 18-44
18-23. Add Relationship Screen — MVS to IMS 18-47
18-24. Relationships Screen - Column Relationships not yet defined 18-50
18-25. Relationships Screen — FIeld Relationships already defined 18-51
18-26. Relationships Screen — Creating a DB2 to MVS Relationship 18-56
18-27. Extended Column Relationship Definition Window 18-57
18-28. Relationships Screen — New DB2 Column to MVS Field Relationship Established. 18-57
18-29. Extended Column Relationship Definition Window 18-58
18-30. Relationship Screen with column-number command 18-61
18-31. Extended Column Relationship Definition Window 18-61
18-32. Column/Segment Relationships Screen with Segment-Number command 18-62
18-33. Extended Column Relationship Definition Window 18-62
18-34. Changed Dependent Start Position and Length 18-63
18-35. Partial Column/Segment Relation Established 18-63
18-36. Column/Field Relationships Screen Mapping DECIMAL to CHAR 18-64
18-37. Extended Column Relationship Definition Window with DECIMAL to CHAR 18-64
18-38. Extended Column Relationship Definition Window after Modification 18-65
18-39. Specify MVS Field Subscripts for Parent Field 18-65
18-40. Specify MVS FIeld Subscripts for Dependent Field 18-66

18-41. Extended Field Relationship Definition with Field Subscripts Defined....18-67
18-42. Extended Field Relationship with Defined Subscripted Relationship 18-68
18-43. Formatted Relationship Report Screen....18-68
18-44. Print Options Screen....18-69
18-45. ISPF/PDF EDIT Screen — File-AID *for DB2*-generated JCL 18-71
18-46. Formatted Relationship Report 18-72
B-1. Action Bar Example - Row Mode....B-1
B-2. Action Bar Example - Table Mode....B-1
B-3. File Pull-down....B-2
B-4. Edit Pull-down in Table Mode....B-2
B-5. View Pull-down in Row Mode....B-3
B-6. Options Pull-Down....B-4
B-7. Open Pull-Down....B-5
B-8. Menu Pull-Down B-6
B-9. Utilities Pull-Down B-7
B-10. Help Pull-Down Menu....B-8
B-11. Action Bar Example - Tables Option B-8
B-12. Action Bar Example - Tables Create Panel....B-9
B-13. Define Pull-down Menu Example B-9
B-14. Actions Pull-down Menu Example....B-10

Tables

3-1. Primary Command Availability by Function 3-1
3-2. Columns Searched by Data Type 3-5
3-3. Date Arithmetic Example 3-14
3-4. Date Increment Example 3-14
4-1. Line Command Availability by Function 4-1
6-1. Valid Values Based on Data Type 6-16
6-2. Sample Relationship 6-28
7-1. Sample Relationship File 7-3
7-2. Base Table Relationship 7-11
7-3. View Relationship 7-11
7-4. Relationship File — One-to-Many Relationship 7-12
7-5. Relationship File — Model Relationship 7-16
7-6. Relationship File — No Relationship 7-18
13-1. Impact Analysis Report Explanation 13-6
13-2. Impact Analysis Codes Explanation 13-6
18-1. Relationship Titles 18-50
18-2. Data Type Values 18-52
18-3. Column/Field Types Mapping 18-53
A-1. Abbreviations Used in Rule Recommendations A-1
B-2. File Pull-Down B-2
B-3. Edit Pull-Down B-2
B-4. View Pull-Down B-3
B-5. Options Pull-Down B-5
B-6. Open Pull-Down B-5
B-7. Menu Pull-Down B-6
B-8. Utilities Pull-Down B-7
B-9. Help Pull-Down B-8
B-10. Define Pull-Down B-9
B-11. Actions Pull-Down B-10

Introduction

This manual documents File-AID *for DB2*, a Compuware database management tool that lets you interactively edit, browse, and manipulate DB2 tables, views, and other objects. The manual consists of the following chapters:

- Chapter 1, "Product Overview" introduces File-AID *for DB2* and briefly describes its features and functions.
- Chapter 2, "Product Conventions" presents information that is common throughout File-AID *for DB2* such as PF key settings.
- Chapter 3, "Primary Commands" describes the File-AID *for DB2* primary commands.
- Chapter 4, "Line Commands" describes the File-AID *for DB2* line commands.
- Chapter 5, "Defaults and User Parameters" describes the File-AID *for DB2* user-definable parameters.
- Chapter 6, "Browse and Edit" describes the File-AID *for DB2* Browse and Edit functions.
- Chapter 7, "Related Table Browse and Edit" describes the File-AID *for DB2* Related Table Browse and Edit function.
- Chapter 8, "Objects Utility" describes the utility used to create, drop, or alter DB2 objects.
- Chapter 9, "Copy Utility" describes the copy utility used to copy rows between tables.
- Chapter 10, "Object List Utility" describes the utility that allows you to display object lists and manipulate them simply by placing a mnemonic beside them.
- Chapter 11, "SQL Command Manager Utility" describes File-AID *for DB2*'s SPUFI connection.
- Chapter 12, "Privileges Utility" describes the utility used to display, grant, revoke, or modify object privileges.
- Chapter 14, "Extract and Load Utility" describes the utility used to extract table data to a file or load data into a table.
- Chapter 13, "Impact Analysis Utility" describes the utility used to produce a report that helps you determine the impact of dropping or modifying a column. It identifies views, aliases, synonyms, indexes, plans/packages, and referentially related columns.
- Chapter 15, "Batch Facility" describes the utility used to execute SQL in batch mode under the control of File-AID *for DB2*.
- Chapter 16, "SQL Development and Analysis" describes four facilities of the File-AID *for DB2* SQL Development and Analysis function: Plan table maintenance, SQL plan analysis, Review or Print DB2 Explain reports, and SQL Source Development and Analysis.
- Chapter 17, "Print Table Data or Audit Trail" describes the table data and audit trail print facilities.
- Chapter 18, "Relationship Facilitator" describes the utility used to view and manipulate the relationship file.
- Appendix A, "Predicate Analysis Rules Abbreviations" describes the short and long messages you may receive when using Predicate Analysis.

- Appendix B, "Action Bar and Pull-Downs Summary" provides a summary of the available action bar items and pull-down menus.

Operating Environment Support

File-AID *for DB2* requires the installation of the following software:

- DB2 Universal Database for OS/390 and z/OS Release 6.1, 7.1, or 8.1
- One of the following ISPF releases: ISPF for z/OS Release 1.4 with APAR OW57368, 1.5 or 1.6
- IBM z/OS Release 1.4 or higher

Compuware recommends a TSO region size of 8M for File-AID *for DB2*. If you want to use a smaller region size, use caution when entering wildcards or working with large amounts of data.

If you want to install the optional Column Impact Analysis utility, one of the following libraries is required:

- PL/I Base and Common Libraries, Release 2 or higher
- LE370 Libraries, Release 1.2 or higher

File-AID *for DB2* utilizes SAS/C run-time library release 7.50F, included in the File-AID *for DB2* load library. If you should experience incompatibility problems with other SAS/C release levels, see the *File-AID for DB2 Installation Guide*, *Appendix J*.

Notation Rules

A vertical change bar (|) in the outside margins indicates a change to this manual for this release.

This manual uses the following notation rules when describing the keys and/or the information displayed on the screen:

- The names of PF keys are printed with capital letters, for example, UP, DOWN, PF3. Named keys are printed with initial capital letters, for example, Enter, Shift.
- Information to be typed in fields on the screen is printed in **bold** type and should be entered exactly as shown.
- Information that appears on the screen (field names, commands) is printed in capital letters.
- Ellipses (...) are used in screen displays to indicate that the illustration spans more than one physical screen image.
- Screen fields and values are discussed immediately following the screen example.

Reading the Syntax Diagrams

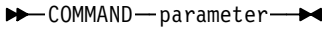
Syntax diagrams define primary command syntax.

A **parameter** is either a keyword or a variable.

All KEYWORDs are shown in uppercase characters and must be spelled exactly as shown. You cannot substitute another value. If any part of a KEYWORD is shown in lowercase characters, that part is optional.

Variables are user-specified values and are printed in lowercase italics. For example, *dataset-name* indicates you are to substitute a value.

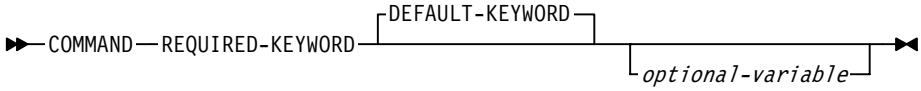
The syntax for commands is described in diagrams that help you visualize parameter use. The following example shows a command and a parameter:



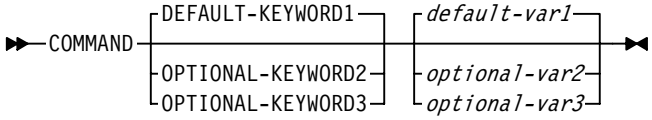
Read the diagrams from left to right and from top to bottom. These symbols help you follow the path of the syntax:

- ▶▶ indicates the beginning of a statement.
- → indicates the statement is continued on the next line.
- ► indicates the statement is continued from the previous line.
- →◀ indicates the end of a statement.

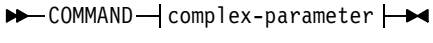
Required parameters appear on the horizontal line (the main path). Optional parameters appear below the main path. Default parameters appear above the main path and are optional. The command executes the same regardless whether the default parameter is included.



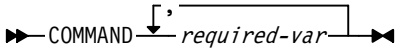
Vertically stacked parameters are mutually exclusive. If you must choose a parameter, one item of the stack appears on the main path. If the parameters are optional, the entire stack appears below the main path. If a parameter in a stack is the default, it appears above the main path.



If the same parameters are used with several commands their syntax may be documented in a separate diagram. In the command syntax, these common parameters are indicated with separators before and after the parameter name.



An arrow returning to the left indicates a repeatable item. If the arrow contains a comma, separate the repeated items with a comma.



Related Publications

- The *File-AID for DB2 Installation Guide* details the installation and customization of File-AID *for DB2*.
- The *File-AID for DB2 Quick Reference* provides easy access to File-AID *for DB2*'s command descriptions.
- The *XPEDITER/TSO Assembler Reference Manual* and the *XPEDITER/TSO COBOL Reference Manual* provide information related to XPEDITER/TSO and the XPEDITER for DB2 Extension.

For DB2 information related to File-AID *for DB2*, refer to the following IBM publications:

- *IBM Database 2 Messages and Codes*
- *IBM Database 2 SQL Reference*

FrontLine Support Website

Access online technical support for Compuware products through our FrontLine support website. View or download documentation, frequently asked questions, and product fixes, or directly e-mail Compuware with questions or comments. To access FrontLine, you must first register and obtain a password at **http://frontline.compuware.com**.

Online Documentation

Documentation for this product is provided on CD-ROM in the following electronic formats:

- View PDF files with the free Adobe Acrobat Reader, available at **http://www.adobe.com**.
- View HTML files with any standard Web browser.
- View BookManager softcopy files with any version of IBM BookManager READ or the IBM Softcopy Reader. To learn more about BookManager or to download the free Softcopy Reader, go to **http://www.ibm.com**.

World Wide Web

Compuware's site on the World Wide Web provides information about Compuware and its products. The address is **http://www.compuware.com**.

Getting Help

At Compuware we strive to make our products and documentation the best in the industry. Feedback from our customers helps us maintain our quality standards.

If problems arise, consult your manual or the File-AID *for DB2* technical support representative at your site. If problems persist, please obtain the following information before calling Compuware's 24-hour Product Support Hotline. This information helps us determine the cause of the problem efficiently.

All Problems

- File-AID *for DB2* release (select option 0 from the Primary Option Menu)
- DB2 release (select option 0 from the Primary Option Menu)
- Screen prints of sequence of events

- File-AID *for DB2* logging information (select option 0.6 from the Primary Option Menu and specify LOG)

Problems Relating to Batch Processing

- Input JCL
- Job Output

ISPF Dialog Manager Errors

- Screen print of TSO LISTA STAT messages
- Screen print of Dialog Manager Error panel

Negative SQL Code

Screen print of Extended Help

Problems Relating to File-AID for DB2 Functions

- Type of table (Alias, View, Synonym, or Table)
- Key information
- DDL for the table and related objects
- Sample data

Abends

- Screen print of LSCX041 message
- Screen print of Calling Trace
- Summary dump
- SYSUDUMP

Problems with the SQL Analysis function

- Source of SQL statement
- Current SQLID

Problems with the Audit Trail

Copy of the Audit Trail dataset

File-AID *for DB2* Technical Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099
1-800-538-7822

Chapter 1.
Product Overview

File-AID *for DB2* is a self-contained DB2 database management and testing tool that lets you create, populate, customize and refresh data in your DB2 database with ease. The File-AID *for DB2* interface is very similar to that of ISPF. The product's familiar look and feel helps you learn and use it quickly.

Product Capabilities

File-AID *for DB2* is designed for the application programmer. It lets you attain self-sufficiency by removing barriers and helping you obtain the data needed to get your job done. Everything you need to create an effective DB2 test environment — *without coding SQL* — is included in one self-contained product. You can follow a logical work flow from start to finish without relying on IBM utilities.

You can create a table modeled from a DBA-designed table, copy data into it from an existing table, and then customize the data to represent exactly the conditions needed to test your program. Valuable test time is not spent working with awkward utilities or coding SQL.

During the application test cycle, you can review a concisely formatted list of privileges for an object. This helps avoid authorization failures and lets authorized users grant and revoke privileges.

You can interactively execute SQL statements and receive analysis of Explain information indicating how the statements perform and suggestions for improvement. This eliminates the need to recompile and rebind in order to verify the SQL. This feature also allows you to prototype SQL code before incorporating it into your application.

Creating Objects

Simple ISPF-like panels guide you through DB2 object definition, modification, and deletion, without ever requiring that you code SQL. A comprehensive set of DB2 objects is supported, including tables, tablespaces, indexes, views, synonyms, and aliases. When creating tables, you can use the modeling option to help you build duplicates of existing tables, right down to primary and foreign key definitions, without knowing what the SQL DDL statements look like. File-AID *for DB2* automatically assists in creating a primary key index.

Alter and drop capabilities, equivalent to those found in SQL, are provided so you can manage your own test tablespace environments.

Creating Test Tables

The easiest way to create test data is to start with data that already exists and then customize it. File-AID *for DB2* gives you the capability to populate test tables with data copied from existing test or production tables without coding SQL. Using the Copy Rows Between Tables utility, just specify the `From` and `To` tables. If the `To` table does not exist, File-AID *for DB2* provides the option of automatically creating it. Templates based on DB2 table definitions are displayed with like-named columns already matched. Instead of cumbersome SQL subselects, you indicate the columns where data is to be copied, even columns with different table definitions. You can further customize the copy request

with selection criteria based on data content and row counts. Extended WHERE clauses and SQL Boolean operators are both supported.

Customizing Data

File-AID *for DB2* gives programmers the power to quickly customize table data to test all layers of program logic. You do not have to know DB2, just ISPF, to perform full-screen browse and edit of DB2 tables. The DB2 table definition is presented to you, so there is no need to remember specific column name spellings and partial column names may be entered when desired.

Both column and row selections are available so that you can work with a subset of the table that is tailored just for you. Row selection supports all SQL Boolean operators and extended WHERE clauses. Selection criteria, once developed, may be saved and reused. Using SPUFI-like input, File-AID *for DB2* also offers table data selection that leverages your existing SQL.

Once in Browse/Edit, table data is displayed using the existing DB2 table definition. Column names, length, and format are always displayed and a column ruler is shown to indicate the start of column data.

Table data can be scrolled and sorted, and rows can be excluded, copied, inserted, deleted, and repeated individually or in blocks — all using standard ISPF commands. Any number of columns can be held, hidden and redisplayed. The FIND and CHANGE commands can be applied globally or limited to a specific column.

Rapid Data Generation techniques with the REPEAT and CHANGE commands let you specify a key range to quickly create large amounts of unique records.

Authorizing Objects

File-AID *for DB2* provides a clear, concise listing of authorizations on a given object, without coding SQL. Programmers can use this information to avoid frustrating reruns due to authorization failures. As objects are created or staff reassigned, DBAs, project leaders, or other authorized users can grant and revoke privileges using a familiar ISPF-like format. Familiar line commands like Insert, Repeat, and Delete make it easy. A Privilege edit capability lets you enter changes directly from an object authorization list.

Reducing the DB2 Learning Curve

There is much to learn in DB2, including SQL. File-AID *for DB2* provides a familiar ISPF-like interface that lets you easily see, interpret, and change DB2 data and delete, define, authorize, and alter objects without coding SQL. You can choose to see the SQL that File-AID *for DB2* uses to complete a request. This generated SQL can be edited or saved, then copied into a program or reused by File-AID *for DB2* or even IBM's SPUFI. The ability to review SQL and to see and work with the results of calls to DB2 can rapidly increase understanding of the DB2 environment.

Reducing Test Time

Easy ISPF-like screens walk you through the process of creating, populating, and customizing a table. Having your own test table eliminates the frustration of being locked out of a table by other testers or having other testers overwrite your carefully planned test conditions. The combination of no contention and effective test data helps reduce test time without sacrificing quality in production.

Restoring Programmer Self-Sufficiency

Constant requests to manage the DB2 test data environment can delay high priority projects. DBAs define the initial rules and environment. Subsequently, with File-AID *for*

DB2, programmers can independently proceed to develop test data because File-AID *for DB2* combines all the basic requirements into one self-contained product.

Programmers can work through a logical flow from start to finish without ever leaving File-AID *for DB2* or relying on IBM utilities. As an added benefit, this frees DBAs to focus on their own priority assignments.

Improving Testing

You do not need to test in production. File-AID *for DB2*'s full-screen editor lets you customize table data easily so that you can test the full range of conditions that can occur in the production environment.

Prototyping SQL Code

The SQL Source Development and Analysis facility lets you insert and analyze new SQL statements in your source code, as well as modify existing SQL statements. This is made easy with the full power of ISPF edit. The cursor logic validation functions allow you to execute the statements, and browse or edit the results. This feature can be accessed through File-AID *for DB2* or through an ISPF/PDF Edit session.

Enhancing SQL Efficiency

You can analyze your SQL statements, Plan, DBRM, Collection ID, or Package and maintain your Plan_Table with File-AID *for DB2*. You can also obtain an Explain report. This gives you vital information about the execution of your SQL statements to save you valuable execution time. Rules-based analysis of the Explain is provided in plain language combined with object attributes and catalog statistics. This text provides conditions, consequences, and recommendations relating to the performance of the SQL statement. You can generate an Explain report for SQL statements containing the DECLARE CURSOR, DELETE, INSERT, SELECT, and UPDATE verbs.

Producing an Audit Trail

Producing an audit trail in File-AID *for DB2* is an installation option. An audit trail can be produced automatically for every edit session, or it can be requested by an individual user. An audit trail dataset can be printed by selecting an option from the Primary Option Menu. When the SMF audit option has been installed, audit records are written to the SMF log file whenever an object is browsed or edited.

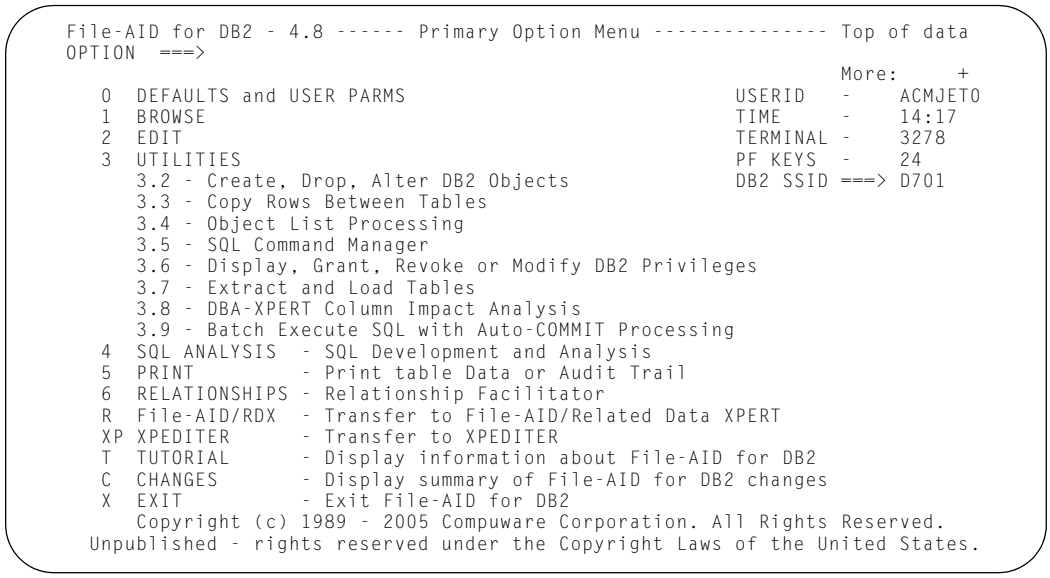
DBCS (Double-Byte Character Set) and Uppercase-Only Support

You can install double-byte character set (DBCS) and/or uppercase-only support. DBCS allows editing and browsing of international languages that require two bytes of storage for each character. Uppercase-only support is provided for terminal types that do not support the display of lowercase characters.

Product Functions

File-AID *for DB2* provides you with functions for creating, updating, and maintaining objects in your DB2 environment, as well as developing and analyzing SQL statements. File-AID *for DB2*'s Primary Option Menu provides immediate access to all product functions. Press PF7 or PF8 to scroll up and down in this menu.

Figure 1-1. File-AID *for DB2* Primary Option Menu



User Parameters

The User Parameters option allows you to set the following parameters:

- Browse/Edit/Analysis display options
- Dataset allocation for SQL
- Batch options
- Index, tablespace, and database default values
- Logging options
- Print options
- System Path options

Browse

The Browse option lets you display a table, alias, or view, either in its entirety or by selected rows or columns (LOB do not appear). You can specify which information you want to see, or you can indicate that you want certain data excluded. You cannot, however, make any changes to the table or view. You can browse a single table at one time, or you can browse multiple related or unrelated tables.

Edit

The Edit option also lets you display a table, alias, or view. Again, you can display the table in its entirety or by selected rows or columns (LOBs do not appear). Optionally, you can save your edit template selection criteria in a dataset for convenient recall in a future edit or browse session. With Edit, however, you can make changes to a table and most views. There are some views that cannot be edited. If you attempt to modify one of these, DB2 locks out the change and File-AID *for DB2* notifies you that you are not allowed to make this change. You can edit a single table at one time, or you can edit multiple related or unrelated tables.

The Related Table Edit function enables you to edit tables related through DB2 referential integrity or through an application relationship. Application relationships can be created using File-AID *for DB2* or File-AID/Related Data XPERT. An unlimited number of related tables may be edited.

Utilities

The Utilities option provides access to a number of different utilities. These utilities allow you to easily manage your DB2 work environment. The utilities are:

- **Create, Drop, Alter DB2 Objects**: Allows you to create and drop ten different DB2 object types: tables, tablespaces, indexes, views, synonyms, aliases, distinct types (UDT), user defined functions (UDF), stored procedures, and triggers. Tables, tablespaces, indexes, UDFs, and stored procedures can be altered, and synonyms and aliases can be modified.
- **Copy Rows Between Tables**: Allows you to copy data between tables. All rows and columns can be copied, or selected rows and columns can be copied.
- **Object List:** Lets you display object lists and perform functions such as Browse and Edit.
- **SQL Command Manager:** Lets you connect with IBM's SPUFI without leaving File-AID *for DB2*.
- **Display, Grant, Revoke, or Modify DB2 Privileges:** You can easily display, grant, revoke, or modify privileges for tables, views, plans, tablespaces, collections, packages, schemas, stored procedures, UDTs, and UDFs by entering simple mnemonics on a screen.
- **Extract, Load Tables:** You can extract a table, alias, or view to a dataset, or load extracted data to a table, alias, or view. Optionally, you can also extract data in delimited file format. Both online and batch processing are available.
- **DBA-XPERT Column Impact Analysis:** You can view or print a batch report that helps you determine the impact of dropping or modifying a column by identifying views, aliases, synonyms, indexes, plans/packages, and referentially related columns.
- **Batch Execute SQL with Auto-COMMIT Processing:** You can execute SQL in batch mode under control of File-AID *for DB2*. Any SQL that can be executed dynamically can execute in File-AID *for DB2* batch. When the SQL contains INSERT, UPDATE or DELETE processing, File-AID *for DB2* will add automatic commit processing at intervals which you specify.
- **SQL Development and Analysis:** The SQL Development and Analysis option is an interactive tool for SQL source development and analysis, SQL plan analysis, Plan table maintenance, and review or printing of a DB2 Explain report.

  SQL Source Analysis and Development is accessed through File-AID *for DB2*, or it can be invoked directly from an ISPF Edit session. With this feature, you can:

  – Dynamically execute SQL statements
  – Browse and edit the DB2 results table
  – Obtain Explain information about how SQL statements perform
  – Obtain help information for any SQL coding errors

  With SQL Plan Analysis, you can explain a plan, DBRM, collection ID, or package. A rules-based analysis of the explain is available in plain language. Predicates can be analyzed to determine if they are indexable, stage I, or both. Based upon the analysis, recommendations for improving the performance of the SQL are provided.

  The print option allows you to review or print a DB2 Explain report from a specified plan table. The report gives you vital information about the execution of selected SQL statements in your application program. The DB2 Explain report is also available when compiling with XPEDITER and Abend-AID.

  Plan table maintenance allows you to delete rows from a plan table.

  In addition to ISPF, File-AID *for DB2* provides an interface to both CA-PANVALET and CA-LIBRARIAN. The SQL Source Analysis and Development facility can be used to analyze members stored in either CA-PANVALET or CA-LIBRARIAN without the need to copy the member into a sequential or partitioned dataset.

Print Table Data or Audit Trail

The Print option on the Primary Option Menu allows you to print selected data from a DB2 table or an audit trail dataset. If the audit trail facility is enabled, an audit trail dataset is created each time an object is edited with File-AID *for DB2*.

Relationship Facilitator

The Relationship Facilitator function allows you to view and manipulate the relationship file that is used in the Related Table Browse and Edit functions. You can view all relationships defined to File-AID *for DB2*, load, refresh, and delete DB2 referential integrity relationships, and add and delete application relationships. File-AID *for DB2* now also provides column subfield support for application relationships.

XPEDITER *for DB2* Extension

The XPEDITER *for DB2* Extension option lets you focus on interactive testing without interruption. You can dynamically enter and experiment with SQL statements during an XPEDITER interactive test session without stopping to compile or bind. You can immediately proceed with the test and see the impact of a new statement in your program's logic flow. Most File-AID *for DB2* data management tools can be accessed from the XPEDITER test session, including the SQL analysis capability.

File-AID/Related Data XPERT

File-AID/Related Data XPERT (RDX) lets you easily extract and load a complete subset of related DB2 and MVS data. You can copy data related by referential integrity or application-defined relationships within the DB2 and MVS environments — with full support for all relational constraints.

File-AID/RDX gives programmers ISPF-like facilities to create and populate test tables and datasets that accurately reflect production relationships, yet contain only a portion of the data, without coding SQL or writing programs. DBAs can migrate related sets of data from one DB2 subsystem and/or MVS environment to another — maintaining all parent/child data relationships without extensive manual procedures.

Relationships defined to File-AID/RDX can be shared with File-AID *for DB2*'s Related Table Edit function.

Chapter 2.
Product Conventions

This chapter presents information that is common throughout File-AID *for DB2*. The information applies to all screens in the system unless otherwise specified. Topics in the chapter include the following:

- Accessing and Exiting File-AID *for DB2*
- Viewing DB2 Version 8 Long Names
- Using ISPF Keylists
- Common Screen Fields
- Current Date/Time
- Wildcard Characters
- Partial Column Names
- Tutorials

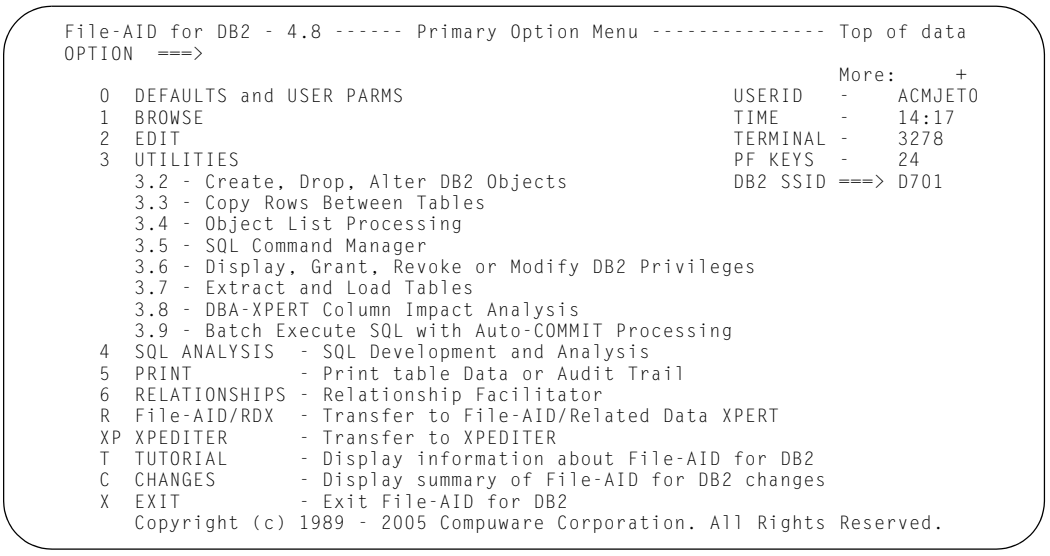
Accessing and Exiting File-AID *for DB2*

Accessing and exiting File-AID *for DB2* is familiar to any user of ISPF. To start, log onto TSO and follow your local procedures for invoking ISPF-based products. Select the option on the ISPF/PDF Primary Option Menu that corresponds to File-AID *for DB2*.

The Primary Option Menu (Figure 2-1) is the first screen displayed when you access File-AID *for DB2*. It lists the major functions of the product for selection.

To exit File-AID *for DB2*, type **=X** on the command line of any File-AID *for DB2* screen and press Enter.

Figure 2-1. File-AID *for DB2* Primary Option Menu



OPTION

Enter the desired File-AID *for DB2* option.

USERID

Current User Identification. This field is display only.

TIME

Current time. This field is display only.

TERMINAL

Terminal type. This field is display only.

PF KEYS

Number of PF keys (12 or 24) defined. This field is display only.

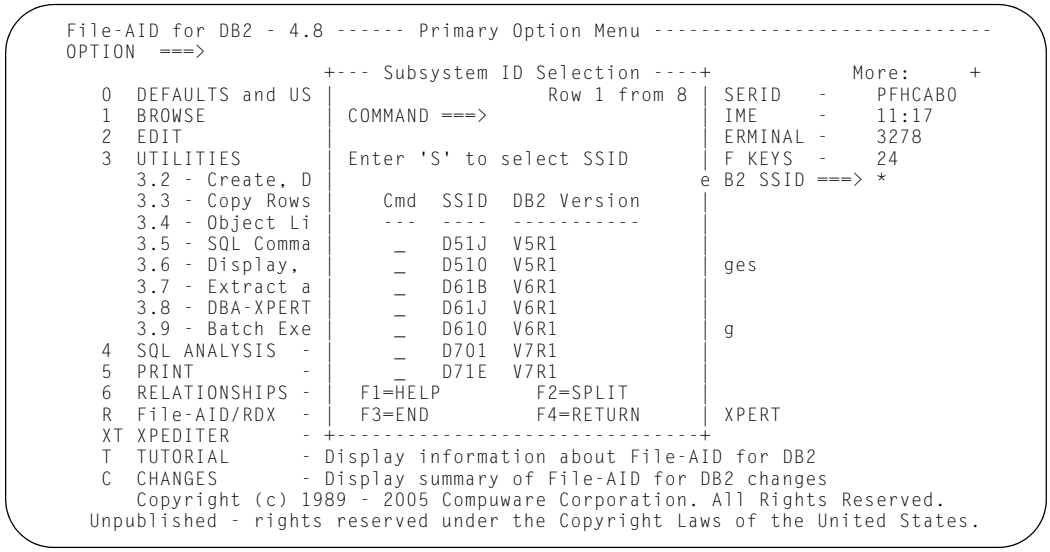
DB2 SSID

Current subsystem identification of your DB2 system. This field can be updated. Entering a new value in this field allows you to immediately connect to a new subsystem. A wildcard character (see "Wildcard Characters" on page 2-9) can be used to display the Subsystem ID Selection window (Figure 2-2).

Subsystem ID Selection Window

The Subsystem ID Selection window (Figure 2-2) displays when you enter a wildcard character in the DB2 SSID field on the Primary Option menu. Select an SSID to connect to a new subsystem.

Figure 2-2. Subsystem ID Selection Window



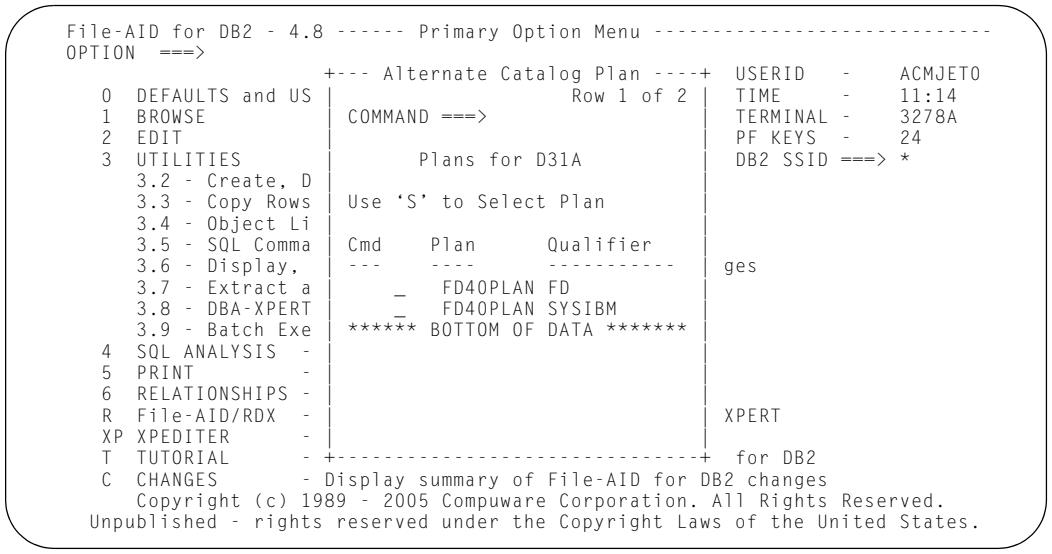
DB2 Connection Type

When the File-AID *for DB2* bridge is used from Xpediter to debug a stored procedure, File-AID *for DB2* uses Resource Recovery Services Attachment Facility (RRSAF). In all other cases, File-AID *for DB2* first attempts a DB2 connection through Call Attach Facility (CAF). If this attempt fails, a second attempt is made using RRSAF. If that also fails, an error message is displayed. This process addresses the issue of another application running in the address space that already has established a DB2 connection.

Alternate Catalog Plan

The Alternate Catalog Plan window (Figure 2-3) displays when an alternate catalog has been defined for the selected subsystem.

Figure 2-3. Alternate Catalog Plan Window



This window includes a selection list of all alternate plans that have been defined for the selected subsystem during File-AID *for DB2*'s installation.

Move the cursor to the desired plan and enter the **S** line command.

When you receive the message

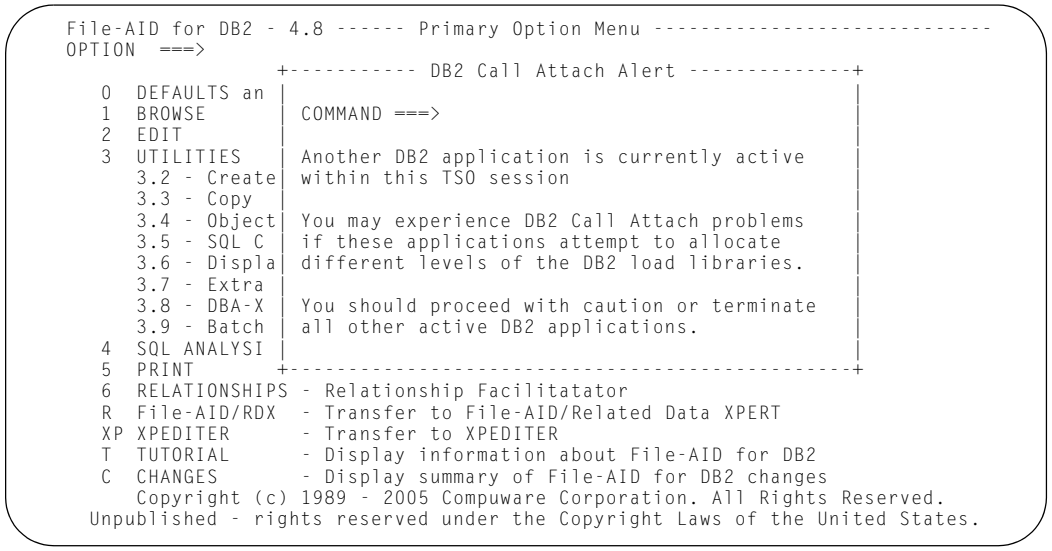
```
Successful Connect
```

you have successfully switched to another subsystem.

DB2 Call Attach Alert

The DB2 Call Attach Alert window (Figure 2-4) displays when another application other than File-AID/Related Data XPERT or File-AID *for DB2* is accessing DB2 in your current TSO session.

Figure 2-4. DB2 Call Attach Alert



Disregard this alert only when the other application shares the same DB2 load libraries with File-AID *for DB2*. To prevent DB2 call attach problems with unpredictable results, Compuware recommends terminating the other applications before you continue with the File-AID *for DB2* session.

Press ENTER to continue with your File-AID *for DB2* session.

Invoking File-AID *for DB2* from Source Code Edit Session

If you are editing SQL in ISPF, or a Library Management Tool, such as CA-PANVALET or CA-LIBRARIAN, you can now directly invoke File-AID *for DB2* Browse, Edit, and SQL Source Development and Analysis with the FADB2 primary command. (See "FADB2" on page 3-19 and "Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44.)

Note: This access option must be identified during system installation. (See "Task 5, Generate ISPF/PDF Primary Panel" on page 2-40 in the *File-AID for DB2 Installation Guide*.) Consult your system administrator on how to use this access option.

Viewing DB2 Version 8 Long Names

Note: DB2 Version 8 changed the maximum length of many fields in the DB2 catalog. For example: The previous maximum length for tables names was 18; with DB2 Version 8, tables names can be up to 128 characters long.

File-AID *for DB2* supports DB2 Version 8 long names. You can display long names in two ways:

- Using the LEFT and RIGHT commands to scroll through the long name within an existing field
- Using the EXPAND command to display the long name in a separate window

Using the LEFT and RIGHT Commands

In order to view a long name in File-AID *for DB2*, use the LEFT and RIGHT commands to scroll in the field on the panel. A "Start of field" or "End of field" message displays in the upper right-hand corner of the screen, when the beginning or end of the field is reached.

Scrollable fields are not supported within dynamic display panels. The following is a list of the most commonly used dynamic display panels:

- Browse or edit session in table mode
- Browse or edit session in row mode
- ZOOM or ZOOMH command
- Issuing an EDIT command when already in a browse or edit session
- Issuing a BROWSE command when already in a browse or edit session

Note: Scrollable fields are a standard ISPF feature. Refer to the "ISPF Dialog Developer's Guide" for more information about scrollable fields.

Use the KEYS command to set LEFT and RIGHT commands to a PF key.

Using the EXPAND Command

The EXPAND command enables you to view a long name in a separate window from the panel of the field. In order to view a long name, type EXPAND in the command line of the panel and press ENTER.

Use the KEYS command to set the EXPAND command to a PF key.

Long Names Printed on Reports

The following reports print long names on two lines, instead of one line: Print Table Mode, Print Row Mode, Audit Trail, Print DS (dataset), Print P (current table). Figure 2-5 is an example of the Table Print Report with a long name.

Figure 2-5. Table Print Report With Long Name

```
File-Aid for DB2 4.8                                    TABLE PRINT REPORT
                                                        DETAIL TABLE DATA



SUBSYSTEM:  D809
LOCATION:   D809CW09
CREATOR:    FDDBA
TABLE/VIEW: ITEM_LONG_TABLE_NAME_TEST


ROW SEQ NO:  1


---COLUMN NAME---- -----FORMAT----- -KEY-- ---------------------------------
PREFIX             CHAR(5)          PRIM   KEVIN
NUMBER             CHAR(32)         PRIM   12345678901234567890123456789012
CATEGORY_NAME      CHAR(32)                dashboard
SUB_CATEGORY_NAME  CHAR(32)                sda
NAME               CHAR(50)                sdfg
LENGTH_LONG_COLUMN DEC(10,2)               10.00
DIAMETER           DEC(10,2)               10.00
*** END OF ROW ***
```

Using ISPF Keylists

File-AID *for DB2* utilizes ISPF keylists in its product screens. Keylists are activated in the ISPF Settings Option (Option 0 from the ISPF Primary Option Menu), using the Function keys menu and selecting Option 10, Enable keylists. Keylists are also be activated by issuing the ISPF command KEYLIST ON.

Common Screen Fields

In general, the first five lines, with Action Bar active, or three lines of each display are formatted as follows:

Action Bar	List of available Pull-down menus
Action Bar Divider Line	

Screen Title	Short Message or DB2 Subsystem ID
Command or Option prompt	Scroll amount or DB2 Subsystem prompt
Long Message	

Action Bar

An action bar at the top of a panel shows the action items you can select on that panel. Appendix B, "Action Bar and Pull-Downs Summary" lists all action items and pull-down choices.

Use the TAB key to move the cursor to an action item. Typical action items are: Menu, Action, View, Options, or Help. Press ENTER to select an action item.

Pull-Downs

When you select an action item on the action bar, File-AID *for DB2* shows you a pull-down with choices related to the item you selected (Figure 2-6). To select a choice, do one of the following:

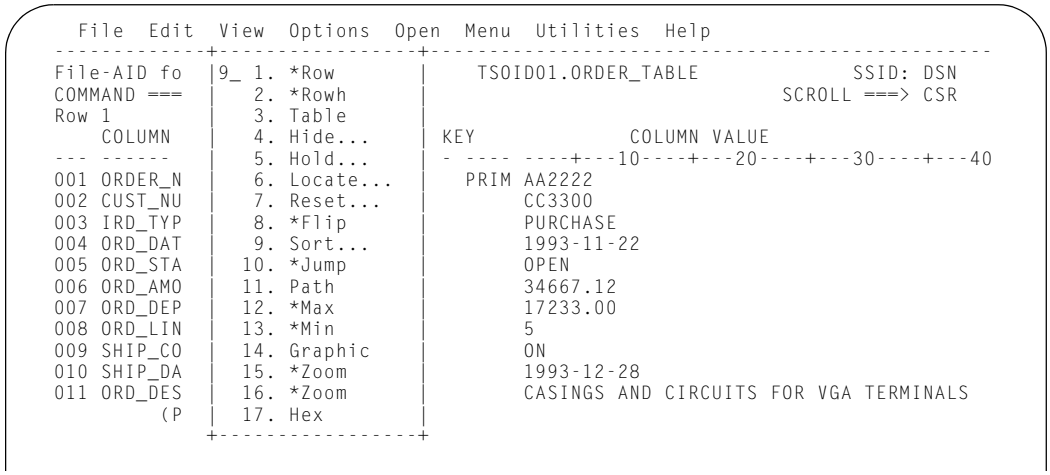
- Type in the Option number, for example 1 for User Parms and press ENTER.
- Move the cursor with the UP or DOWN arrow keys to the desired Option row and press ENTER.

An asterisk (*) in front of an item tells you that you cannot select it because it is unavailable for the current screen. If you try to select an item marked with an asterisk, File-AID *for DB2* ignores it.

An ellipsis (...) after an item tells you that File-AID *for DB2* will request additional information before completing the item's action.

While a pull-down displays, only the items in the pull-down or on the action bar are available. For example, if you select a pull-down and the item you want is not listed in it, you can select another pull-down on the action bar. However, you cannot mark or enter anything on the background panel until the pull-down is removed. Press the END or RETURN PF keys to remove the pull-down when you do not want to select an item.

Figure 2-6. Views Pull-down Menu Example



Windows

If you select a pull-down action that is followed by an ellipsis, File-AID *for DB2* removes the pull-down and displays a window with more choices related to the action you selected, or with an area in which you must type the information it needs to complete the action (Figure 2-7).

If you select a window action that is followed by an ellipsis, File-AID *for DB2* displays another window on top of the first one.

Figure 2-7. Sort Command Window

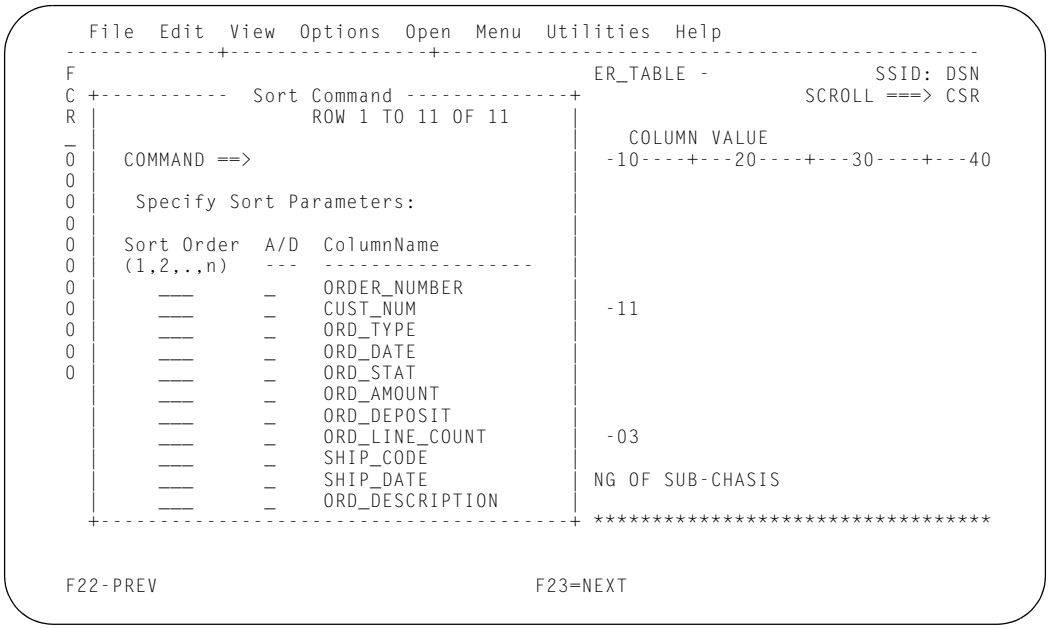
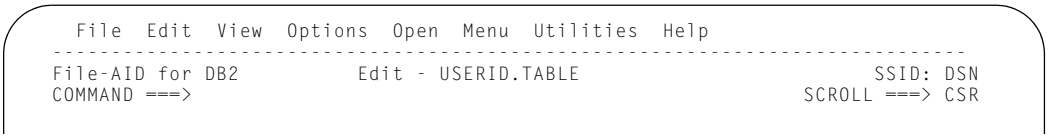


Figure 2-8 illustrates the header area, with Action Bar.

Figure 2-8. File-AID *for DB2* Screen Header with Action Bar



Screen Title

The screen title is usually displayed at the top center of each File-AID *for DB2* screen. It consists of a short description of the screen's function.

COMMAND Field

The COMMAND field (command line) is shown on most screens. On some screens, it is shown as the OPTION field. Use this field to enter primary commands or option selections for File-AID *for DB2*. Some primary commands are valid for all screens, while others are active only on selected screens.

For more detailed information about the File-AID *for DB2* primary commands, see Chapter 3, "Primary Commands".

SCROLL Field

The SCROLL field lets you set the scroll value for screens that permit scrolling. The following positioning commands are used with this field: DOWN, UP, LEFT, and RIGHT.

If you modify the SCROLL field, the value you enter is retained for the duration of the File-AID *for DB2* session. If you wish, you can set the SCROLL field to one value for Edit and Browse, and another for the utilities. The following values are valid for the SCROLL field:

CSR

Scrolling based on the cursor's location.

HALF

Half-screen scrolling.

MAX

Scrolling to the beginning or end of a data area.

PAGE

Full-screen scrolling.

nnnn

Scrolling *nnnn* lines. Valid entry is from 1 to 9999.

Subsystem Identification

Most File-AID *for DB2* screens display your current Subsystem identification (SSID) and some allow you to update it. If the SSID is display only, it displays on the title line and is not highlighted (Figure 2-8). If the SSID can be updated, it displays on the command line and is highlighted (Figure 2-9). For information on setting the default SSID value, see "User Parms Menu" on page 5-1.

Figure 2-9. File-AID *for DB2* Screen Header with SSID — Updates Allowed

```
File-AID for DB2 ----------------- Edit Options --------------------------------
COMMAND ===>                                                  SSID ===> DSN
```

CURRENT Date/Time

All Date, Time, and Timestamp columns let you enter the keyword CURRENT, or its abbreviation CUR. The current date, time, or timestamp is then filled in as appropriate. A user default option is provided so that current time is local or Greenwich Mean Time (GMT). This option is set with the PROFILE primary command (see "PROFILE (PRO)" on page 3-39). To put values in these columns, use the following formats:

DATE=yyyy-mm-dd

TIME=hh.mm.ss

TIMESTAMP=yyyy-mm-dd-hh.mm.ss.nnnnnn

where *yyyy* is year, *mm* is month, *dd* is day, *hh* is hour, *mm* is minute, *ss* is second, and *nnnnnn* is nanosecond.

Date and time fields are displayed in the default format defined at your site. File-AID *for DB2* accepts any valid format, such as European or USA. File-AID *for DB2* validates all

date, time and timestamp fields and issues appropriate error messages before sending the fields to DB2.

Wildcard Characters

Many screens allow you to use wildcard characters. If you enter an asterisk (*****) in fields that allow this wildcard character, File-AID *for DB2* displays a selection list of objects. You can then select the appropriate name from the list. The optional fields treat a blank as an asterisk. The valid wildcard characters are:

- **Asterisk (*)** – All values match. Must be entered by itself, cannot be used with other characters.
- **Underscore (_)** – Occurrence of any single character matches. For example, xyz, xaz, and xbz all match **x_z.**
- **Percent sign (%)** – Occurrence of no characters or an unlimited number of characters match. For example, xyz, xz, and xmnoz all match **x%z.**

Note: DBCS only: The wildcard character (*****, %, or _) must be inside the SHIFT-OUT/SHIFT-IN key sequence to apply to the double-byte characters.

Wildcard Character for PDS Member Fields

When specifying a member name for a partitioned dataset, the asterisk (*****) is the only valid wildcard character. File-AID *for DB2* displays a selection list of member names. You can narrow the appropriate name from the list. The optional fields treat a blank as an asterisk. The valid wildcard characters are:

- **Asterisk (*)** by itself – All values match.
- **Beginning name string with an asterisk (*)** – For example, **SH*** lists only member names beginning with SH, such as SHOWSQL, SHOWSQ2, and SHUTDWN.

Wildcard Characters for Relationship Option

In the Relationship Option, Option 6, the use of wildcard characters is as follows:

- **Asterisk (*)** – Matches all values. Valid for specifying objects, creators, tables, and members.
- **Underscore (_)** – Matches an occurrence of any single character. If an underscore is embedded in a string, the string must end with a percent sign (%). For example, **xyz**, **xaz**, and **xbz** all match **x_z%**. Valid for specifying DB2 objects only.
- **Percent sign (%)** – Matches the occurrence of no characters or an unlimited number of characters. If a percent sign is embedded in a string, the string must end with a percent sign (%). For example, **xyz**, **xz**, and **xmnoz** all match **x%z%**. Valid for specifying DB2 objects only.

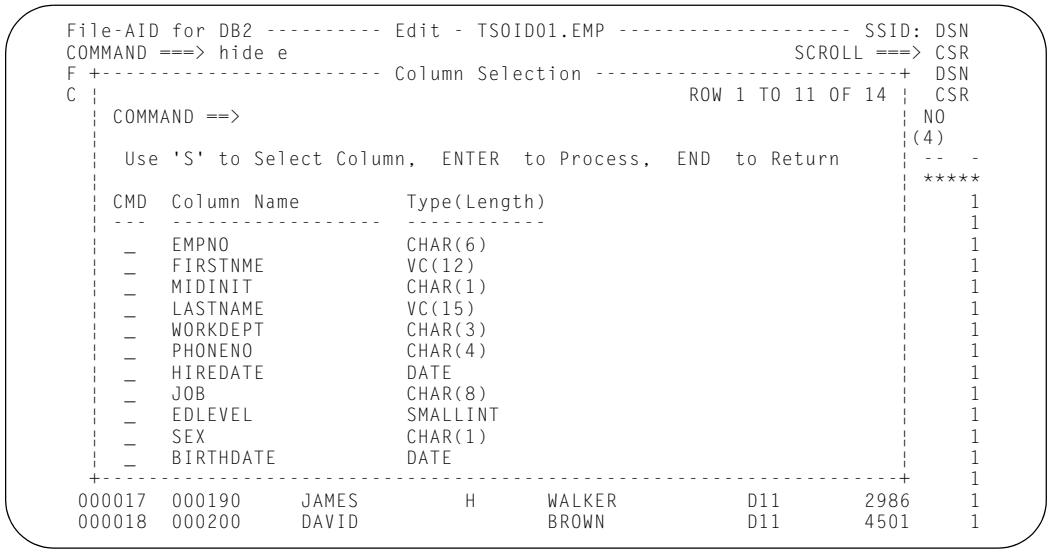
After you enter a pattern, File-AID *for DB2* displays a selection list. You can then select the appropriate name from the list.

Partial Column Names

Primary commands that accept a *column-name* parameter allow you to specify a partial column name instead of the entire name. Simply enter at least the first character of the column's name. Do not use a wildcard character along with the column name. For example, to locate the column FIRSTNME, enter **LOCATE F** on the command line. If the partial name qualifies more than one column, you can choose a column from a selection

list. Multiple windows display if multiple partial column names are specified for a single command. Figure 2-10 shows a sample column selection window.

Figure 2-10. Column Selection window

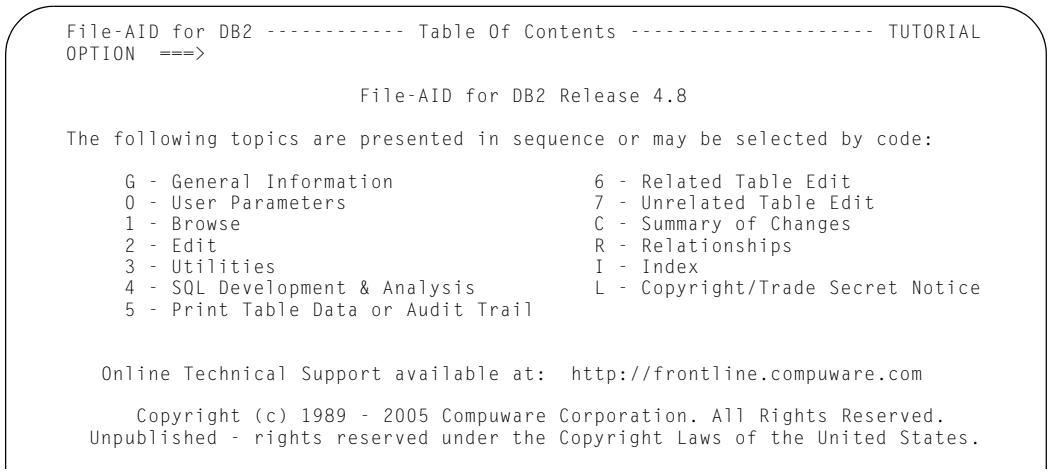


Tutorials

Tutorials are available for each function and option in File-AID *for DB2* to provide detailed information on the function, and to explain error conditions. You can access tutorials from any File-AID *for DB2* screen by pressing HELP or by selecting option T from the File-AID *for DB2* Primary Option Menu.

File-AID *for DB2* also has a Tutorial Table of Contents screen (Figure 2-11). Select option T from the File-AID *for DB2* Primary Option Menu (Figure 2-1 on page 2-1) to access it. You can select a specific topic from the Tutorial Table of Contents screen, or read the tutorials sequentially. The tutorials contain an alphabetic index of topics. To access the index, type the letter I on any tutorial and press Enter.

Figure 2-11. Tutorial Table of Contents



Chapter 3.
Primary Commands

The File-AID *for DB2* primary commands are entered in the COMMAND field. Whenever the action bar is active for a screen, you can also select the primary commands from the action bar pull-down menus. Table 3-1 shows primary command availability by function.

Table 3-1. Primary Command Availability by Function

Primary Command	Browse	Edit	Table Mode	Row Mode	SQL Source Analysis	Utilities	Relationships
ADD	x	x	x	x			x
ALTER						x	
ATOMIC						x	
BOTTOM	x	x	x	x	x	x	x
BROWSE	x	x	x	x			
CANCEL	x	x	x	x	x	x	x
CAPS	x	x	x	x			
CHANGE		x	x	x			
COLCHECK						x	
CREATE						x	
CUT		x	x				
DEFAULT						x	
DELETE		x		x			
DOWN	x	x	x	x	x	x	x
EDIT	x	x	x	x			
END	x	x	x	x	x	x	x
EXCLUDE ALL	x	x	x				
EXECUTE						x	
FADB2					x		
FIND	x	x	x	x			x
FLIP	x	x	x				
FOREIGN						x	
GRANT						x	
GRAPHIC	x	x	x	x			x
HELP	x	x	x	x	x	x	x
HEX	x	x		x			
HIDE	x	x	x	x			
HOLD	x	x	x	x			
IDENTITY						x	
INFO	x	x	x	x	x	x	x
INSERT		x		x			
JUMP	x	x	x	x			
KEYLIST							x
KEYS	x	x	x	x	x	x	x
LEFT	x	x	x		x		
LOCATE	x	x	x	x		x	x
MAX	x	x	x	x			
MIN	x	x	x	x			
MODIFY						x	
NAMCHECK						x	

Table 3-1. Primary Command Availability by Function

Primary Command	Browse	Edit	Table Mode	Row Mode	SQL Source Analysis	Utilities	Relationships
NEXT	x	x		x			
PARMS						x	
PARTITN						x	
PASTE		x	x				
PATH	x	x	x	x			
PREV	x	x		x			
PRIMARY						x	
PRINTDS	x	x	x	x			
PRINTP	x	x	x	x	x		
PROFILE	x	x	x	x			
RBROWSE	x	x	x	x			
RCHANGE		x	x	x			
REDIT	x	x	x	x			
REMARKS						x	
REPEAT		x		x			
RESET	x	x	x	x			
RETPARM						x	
RETRIEVE	x	x	x	x	x	x	x
RETURN	x	x	x	x	x	x	x
REVOKE						x	
RFIND	x	x	x	x			
RI [1]		x		x			
RIGHT	x	x	x		x		
ROW	x	x	x				
ROWH	x	x	x				
RUNOPTS						x	
SAVE		x	x	x	x		
SORT	x	x	x	x	x	x	x
SPLIT	x	x	x	x	x	x	x
SQL	x	x			x	x	
SQLID	x	x	x	x	x	x	x
SRCPARM						x	
SWAP	x	x	x	x	x	x	x
TABLE	x	x		x			
TOP	x	x	x	x	x	x	x
UDFOPTS						x	
UDT	x	x	x	x		x	x
UP	x	x	x	x	x	x	x
VALUES						x	
WHENCOND						x	
WRAP	x	x	x				
ZOOM	x	x	x				
ZOOMH	x	x	x				

[1] Valid only from the RI Browse window.

Description of Primary Commands

The primary commands are presented in alphabetical order with syntax diagrams. Command abbreviations are in the command heading enclosed in parentheses.

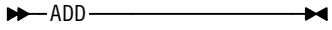
ADD

The ADD command allows you to add a new application relationship to the relationship file. ADD is valid in Related Table Browse or Edit from the following windows:

- No Relationships Exist (Figure 7-18 on page 7-19)
- Relationship Listing (Figure 7-16 on page 7-17)
- Related Table List (Figure 7-10 on page 7-13)

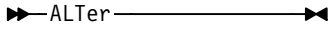
ADD is also valid in Option 6, Relationships from the Relationship Detail screen (Figure 7-33 on page 7-29).

See "Creating a New Relationship" on page 7-19 for more information on the ADD primary command.



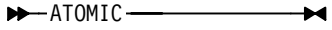
ALTER (ALT)

The ALTER command allows you to modify a DB2 object when in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Execute ALTER on the appropriate screen when you complete your changes. See Chapter 8, "Objects Utility" for more information.



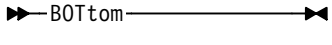
ATOMIC

Used in Trigger Create to define the triggered SQL statement. The triggered SQL statement displays after the BEGIN ATOMIC clause in the CREATE TRIGGER DDL. There may be multiple triggered SQL statements, but each statement should be followed by a semicolon.



BOTTOM (BOT)

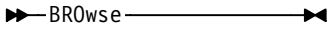
The BOTTOM command is used to scroll the data on the screen to reveal the last line of data if it is not already displayed. BOTTOM is equivalent to the DOWN MAX primary command and may be used on any scrollable screen. In Browse or Edit row mode, the BOTTOM command scrolls the data on the screen to reveal the last column of data in the row being viewed. A **BOTTOM OF DATA** message displays to notify you that the last line of data has been reached.



BROWSE (BRO)

The BROWSE command initiates a multiple browse session. The table does not have to be related to the one currently being browsed or edited. The Edit Options window displays allowing you to enter a table name. See "Editing Multiple Unrelated Tables" on page 6-22

for more information on the BROWSE command. BROWSE is not valid in Related Table Browse or Edit. See Chapter 7, “Related Table Browse and Edit”.



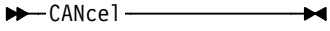
CANCEL (CAN)

The CANCEL command cancels all uncommitted editing changes made since the last time a SAVE command was issued and ends the edit session. Using CANCEL can cause a problem if you decide to cancel a change but do not want to lose other edits made since the last SAVE command. To avoid this problem, save your editing changes periodically.

Note: In a multiple edit session (related or unrelated), executing the CANCEL command cancels all uncommitted changes made to the active table and its dependent tables.

In functions other than Edit, CANCEL cancels the current activity and takes you back to the previous screen.

Note: Setting Auto Commit to ON nullifies this command because changes are saved each time you press Enter.



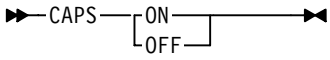
CAPS

The CAPS ON or OFF command controls whether characters entered in lowercase format are automatically set to uppercase when in an edit session.

You can execute the CAPS command during a browse or edit session. You can also set this option through Option 0, Defaults and User Parms (see “Option 1, Browse/Edit/SQL Analysis Display Options” on page 5-2) or through the PROFILE primary command (see “PROFILE (PRO)” on page 3-39).

Note: The CAPS option has no effect on the OBJECT NAMES UPPERCASE option set through the PROFILE parameters window. See “Profile Parameters Window” on page 3-39.

Since CAPS is a profile option, its status remains in effect until you change it.



ON

All text of the column field that you enter, insert, or change, is converted to uppercase format. The conversion occurs when you press Enter, issue an END command, or issue a SAVE command. The conversion affects all text in a modified field up to the end of the current line of the field. For example, you change the letter

u to i in the following 20 character column with WRAP 10 in effect, and all 10 letters in the changed line of the column are capitalized.

Before

```
ROW ID:       9
          ----+---10
 000005   Thus is an
    392   ----+---20
          example.
```

After

```
ROW ID:       9
          ----+---10
 000005   THIS IS AN
     18   ----+---20
          example.
```

OFF

All characters are accepted in the same case in which they were entered. No lowercase to uppercase conversion occurs.

CHANGE (C)

The CHANGE command finds a data value to be changed and replaces it with a new data value. It is valid only in Edit. The CHANGE command cannot be used in ZOOM or ZOOMH mode.

Note: Entering CHANGE with no parameters brings up the Change Command window (Figure 3-2 on page 3-9). This gives you access to expanded functionality such as specifying incremental replacement values or performing date and time arithmetic (data aging). You may also select a block of rows with the CC line markers to limit the effective search range for the CHANGE command.

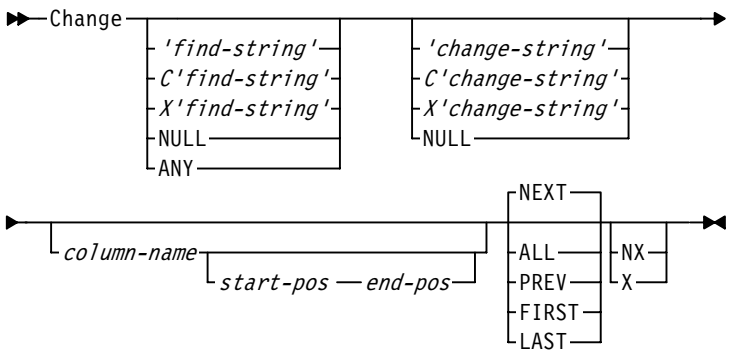
When you specify a column name, you limit the search only to data in the specified column. When you specify positions within the column, you further restrict the search. If no column name is entered, all columns in the table are searched for the data. However, the CHANGE command searches only certain column types based on the data type of the find-string. Table 3-2 describes which columns are searched.

Table 3-2. Columns Searched by Data Type

Find-string Data Type	Column Type Searched		
	Character	**Numeric**	**Graphic**
Character	x		
Numeric	x (substring match)	x (exact numeric match)	
Hexadecimal	x		x
DBCS	x		x
Mixed	x		

By default, the CHANGE command searches both excluded and non-excluded rows as well as both hidden and non-hidden columns. You can limit the search to only excluded rows and hidden columns or only non-excluded rows and non-hidden columns.

Note: If the CHANGE command is issued in row mode, all qualifying data, rather than qualifying data in the displayed row only, is searched.



Find and Change Strings

The first parameter for the CHANGE command is always the find-string, the second parameter the change-string. Enclose the strings in either single or double quotes when:

- the *find-string* or *change-string* contains blanks
- the *find-string* or *change-string* matches the characters of reserved words, for example NULL, but you only want to change words with these characters in them. Valid reserved words for the CHANGE command are:

ANY (only in find string)	CURRENT_TIME
CUR	CURRENT TIMESTAMP
CURR	CURRENT_TIMESTAMP
CURRENT	USER
CURRENT DATE	NULL
CURRENT_DATE	NULLS
CURRENT TIME	

- the *find-string* or *change-string* contains a single quote (’), you must enclose it in double quotes
- the *find-string* or *change-string* contains double quotes ("), you must enclose it in single quotes

find-string

Value to be changed. The find-string value is not case sensitive, unless enclosed in quotes and prefaced by C.

ANY

Change *any* value. Allows you to change data in a column without any regard to its current value. Specify ANY as the find-string, but do not enclose it in single or double quotes. You must also specify a column name when using the reserved word ANY as a find-string.

If ANY is too general as a wildcard, you may want to substitute picture strings for ANY as the find-string.

P

Specifies a picture string. P'picture' may contain one or a combination of the following:

=	any character
¬	any non-blank character
.	any undisplayable character
#	any numeric character
-	any non-numeric character
@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character

Note: Picture strings are accepted only as find-strings. They may not always achieve the expected results. For example, P'-' locates the blank filler character before an integer. Use ANY instead wherever possible.

change-string

New value. The change-string replaces the find-string. Lowercase characters only change to uppercase when CAPS ON is in effect. See "CAPS" on page 3-4.

C

Specifies a case sensitive string. Enclose the string in quotes and preface it with a C. For example, if you want to search for a value of SMITH in all capital letters, specify **c'SMITH'**. To change a value to Smith in mixed case, specify **c'Smith'**.

X

Specifies that the value is in hexadecimal format. When specifying a hexadecimal value, you must enclose the find and change strings in single or double quotes.

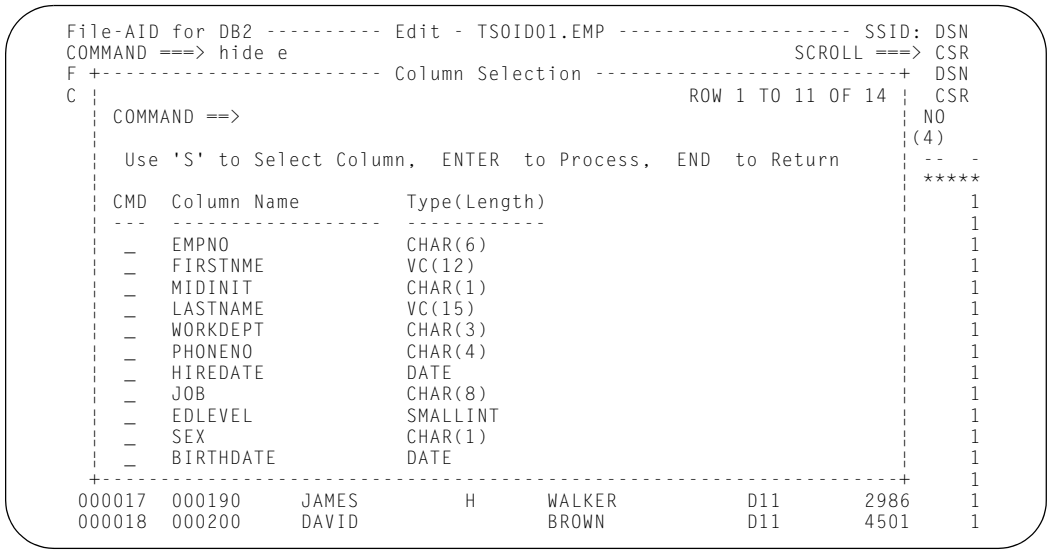
NULL

Specifies a search for null values or a change to null values.

column-name

Name of the DB2 column in which you want to restrict the change. You can enter a partial column name. If more than one column name matches the partial name, a window displays all matching column names, allowing you to select one. Figure 3-1 on page 3-8 shows a sample column selection window. See "Partial Column Names" on page 2-9 for more information.

Figure 3-1. Column Selection window



Start position

Starting position within the specified column name in which you want to restrict the change. If only one position is specified, it is processed as a starting position.

End position

Ending position within the specified column name in which you want to restrict the change. If only one position is specified, it is processed as a starting position.

Note: The Start/End positions are valid only for string and date/time fields.

Modifiers

Specify one of the following parameters to modify the search order.

NEXT

Default. Starts a search for the next occurrence of the *find-string* at the beginning of the first line being displayed (if the cursor is still in the COMMAND field), or at the cursor location (if the cursor is in the data part of the display). If you specify both a column name and the NEXT parameter, the next occurrence in the specified column of the *find-string* is changed.

ALL

Changes all occurrences of the *find-string* starting at the beginning of the first line of the data and continues to the bottom of data. If you specify both a column name and the ALL parameter, all occurrences in the specified column of the *find-string* are changed.

PREV

Starts a search for the previous occurrence of the *find-string* starting at the end of the line preceding the first line being displayed (if the cursor is still in the COMMAND field), or at the cursor location (if the cursor is in the data display). If you specify both a column name and the PREV parameter, the previous occurrence in the specified column of the *find-string* is changed.

FIRST

Starts a search for the first occurrence of the *find-string* starting at the beginning of the first line of the data and continues until the string is found or the bottom of data

is reached. If you specify both a column name and the FIRST parameter, the first occurrence in the specified column of the *find-string* is changed.

LAST

Starts a search for the last occurrence of the *find-string* starting at the end of the last line of the data and continues backward until the string is found, or the top of data is reached. If you specify both a column name and the LAST parameter, the last occurrence in the specified column of the *find-string* is changed.

Rows/Columns to Search:

Specify one of the following parameters to determine which rows or columns to search.

Blank

Default. Searches for the string in all rows and columns.

NX

Searches for the string only in those rows that are not excluded and columns that are not hidden.

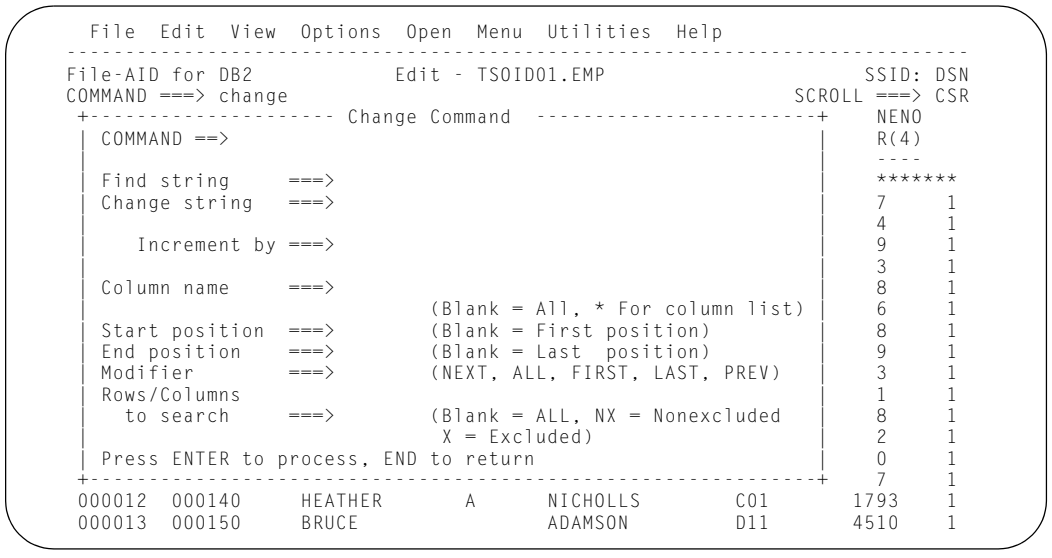
X

Searches for the string only in those rows that are excluded and columns that are hidden.

CHANGE Command Window

The CHANGE command window (Figure 3-2) prompts you for the information required to complete your change request. It is only displayed when you enter CHANGE with no parameters. Use the CHANGE command window to specify special change requests, such as replace ANY information in a column or increment the change string to create unique values. After you provide the required change information, press Enter on this screen to process the change and exit the window. Press END to cancel the CHANGE command and exit the window.

Figure 3-2. CHANGE Command window



Find and Change Strings

For general description see “Find and Change Strings” on page 3-6.

Find-string

Value to be changed.

To specify any value, enter the reserved word **ANY**. ANY replaces any value in the specified column positions regardless of its current content.

Note: When entering ANY you must also identify the column for which values are to be replaced.

Change string

New value (maximum 40 characters, including quotes).

When you also fill in a value in the Increment by field, the change string represents the initial value that is to be incremented. For example: 100 is the change string for the Employee Number. This value is used as the first replacement string. Then the Increment by value, for example 10, modifies the initial value. Thus, the following Employee Number values of a multiple change are 100 (initial), 110, 120, 130, etc. The new value must fit the required format and length for the selected column, or you receive an error message.

Increment by

Increment to modify alphabetic (a through z) and numeric characters (0 through 9) of the change string. Other characters (e.g %, #, =, etc.) are not changed by the increment value. The default increment is blank (no increment). The increment is always a numeric value. A negative increment, for example (-1), is valid and decreases from the initial value (100 initial, 99, 98, 97, etc.).

Positive increment is assumed and must be entered without a sign.

The column data type determines how increments are interpreted. Date and time actually have additional pop-up displays (see "Timestamp Pop-up Window" on page 3-11, "Time Pop-up Window" on page 3-14, and "Date Pop-up Window" on page 3-15).

Specify an increment when you want to change a range of records with one CHANGE command and avoid duplicate change strings.

When the change string is an alpha character, for example A, and the increment is 2, the resulting string is A, C, E, G, I, K, etc.

When the change string is alphanumeric, for example AB3, and the increment is 4, the resulting string is AB3 (initial), AB7, AC1, AC5, AC9, AD3, etc.

Alphabetic and numeric data values wrap when they reach the maximum allowable value for the column, starting over at 0 (numeric) or A (alphabetic). Alphabetic data includes all letters of the alphabet in upper and lower case. Integers wrap at their maximum values, for example, increment -1 wraps from -2147483648 to 2147483647, increment 1 wraps from 2147483647 to 0.

Note: Unless the column data type is CHAR or VARCHAR the Increment operand and Start/End Position operands are mutually exclusive.

Column name

Name of the DB2 column in which you want to restrict the change. Entering an asterisk (*) presents a list of columns from which you select a column name. You can also enter partial column names. If more than one column name matches the criteria, a window displays all matching columns, allowing you to select one. See Figure 3-1 on page 3-8 for a sample window. If you do not specify a column name, the change is applied to all columns.

Note: You must enter a column name whenever you execute a CHANGE ANY or CHANGE with an increment value.

The column data types DATE, TIMESTAMP, and TIME actually have additional pop-up displays to perform date and time arithmetic or incremental changes (see "Timestamp Pop-up Window" on page 3-11, "Time Pop-up Window" on page 3-14, and "Date Pop-up Window" on page 3-15).

Start position

Starting position within the specified column name in which you want to restrict the change. If no end position is specified, the search continues until the end of the column.

End position

Ending position within the specified column name in which you want to restrict the change. If no start position is specified, the search begins at the beginning of the column.

Note: The Start/End positions are valid only for string and date/time fields. Unless the column data type is CHAR or VARCHAR the increment operand and Start/End position operands are mutually exclusive.

Modifier

NEXT is the default, unless you have specified a search range with the CC line markers. In that case, ALL is the default within the range. See "Modifiers" on page 3-8 for an explanation of valid values.

Rows/Columns to search

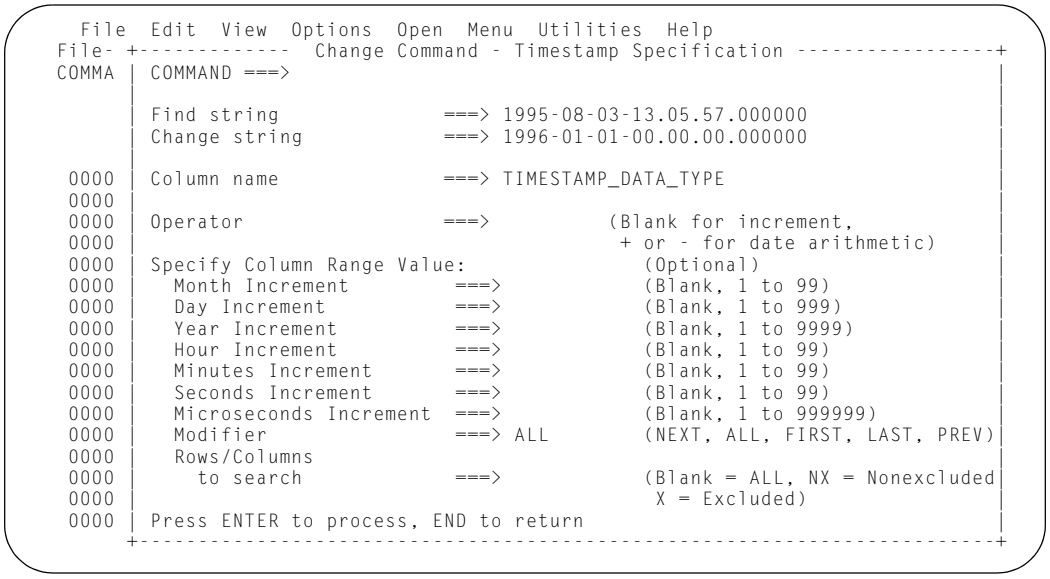
See "Rows/Columns to Search:" on page 3-9 for an explanation of valid values.

Timestamp Pop-up Window

When the column data type in the CHANGE command window is TIMESTAMP, the Timestamp Specification window (Figure 3-3 on page 3-12) displays. You can either apply increments to the *change-string* for *rapid data generation* or perform date and time arithmetic (add or subtract) on the *find-string* for *data aging* purposes. After you provide the operator and increment information, press Enter on this screen to process the change and exit the window. Press END to cancel the CHANGE command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates are 1998/01/31, 1998/02/28, 1998/03/28, etc.

Figure 3-3. CHANGE Command Window with Timestamp Increment



Find string

Prefilled with the value you specified in Change Command window. You are allowed to change the prefilled Find string.

Change string

Blank or prefilled with the value you specified in Change Command window.

The *change-string* must be blank, if you want to perform *data aging* you entered a + or - character in the Operator field.

The *change-string* must be a complete timestamp value if you want to generate incremental timestamp value(s) (Operator field is blank).

Column name

Prefilled with the value you specified in Change Command window. To change the column press END to return to the Change Command window.

Operator

Determine how the date and time should be changed.

Blank

The *change-string* is incremented as specified by the increment values, after the initial replacement when multiple records are replaced with the same CHANGE command.

\+ The increment values are added to the *find-string*, for example, 1998 plus year increment 13 results in 2011. When adding dates, they are added in the order YEARS + MONTHS + DAYS.

\- The increment values are subtracted from the *find-string*, for example, 1998 minus year increment 13 results in 1985. When subtracting dates, they are subtracted in the order DAYS - MONTHS - YEARS.

Notes:

1. Date Arithmetic
   When rolling over dates (resulting year is greater than 9999, or less than 1), File-AID *for DB2* issues a warning and quits the CHANGE command.

2. Incremental
   When rolling over dates (resulting year is greater than 9999), File-AID *for DB2* continues and starts again at 1, for example 9985, 9995, 0001, 0011, etc.

Month Increment

Increment to modify the month portion of the timestamp. The default increment is blank (no increment). Enter the desired month increment (1 to 99).

Day Increment

Increment to modify the day portion of the timestamp. The default increment is blank (no increment). Enter the desired day increment (1 to 999).

Year Increment

Increment to modify the year portion of the timestamp. The default increment is blank (no increment). Enter the desired year increment (1 to 9999).

Hour Increment

Increment to modify the hour portion of the timestamp. The default increment is blank (no increment). Enter the desired hour increment (1 to 99).

Minutes Increment

Increment to modify the minutes portion of the timestamp. The default increment is blank (no increment). Enter the desired minutes increment (1 to 99).

Seconds Increment

Increment to modify the seconds portion of the timestamp. The default increment is blank (no increment). Enter the desired seconds increment (1 to 99).

Microseconds Increment

Increment to modify the microseconds portion of the timestamp. The default increment is blank (no increment). Enter the desired microseconds increment (1 to 999999).

Modifier

Prefilled with the value you specified in Change Command window. You are allowed to change the prefilled Modifier value.

See “Modifiers” on page 3-8 for an explanation of valid values.

Rows/Columns to search

Prefilled with the value you specified in Change Command window. You are allowed to change the prefilled search restriction.

See “Rows/Columns to Search:” on page 3-9 for an explanation of valid values.

Date Arithmetic and Increment Example

The following table is an example that illustrates the difference between date arithmetic and incremental.

Table 3-3. Date Arithmetic Example

Date Records	To Age by 13 Years	Results
1998-12-31	Enter **ANY** in the Find string	2011-12-31
1999-12-31	**Blank out** the Change string	2012-12-31
1985-12-31	Enter + in the Operator field	1998-12-31
1978-02-26	Enter **13** in the Year Increment field	1991-02-26
1994-08-01	Enter **ALL** in the Modifier field	2007-08-01
1987-03-31	Press Enter	2000-03-31

Table 3-4. Date Increment Example

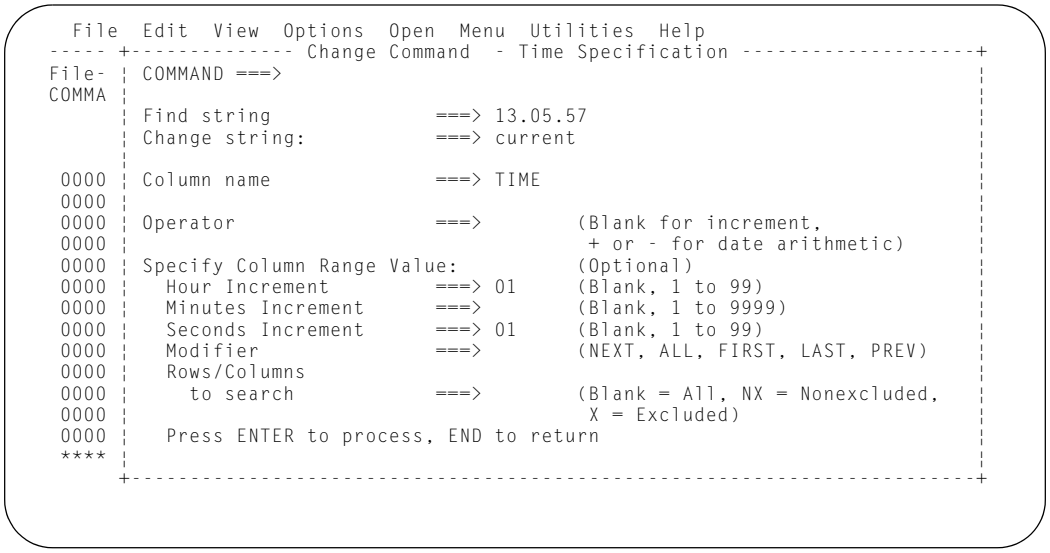
Date Records	To increment by 13 Years	Results
1998-12-31	Enter **ANY** in the Find string	2010-12-31
1999-12-31	Enter **2010-12-31** in the Change string	2023-12-31
1985-12-31	**Blank out** the Operator field	2036-12-31
1978-02-26	Enter **13** in the Year Increment field	2049-12-31
1994-08-01	Enter **ALL** in the Modifier field	2062-12-31
1987-03-31	Press Enter	2075-12-31

Time Pop-up Window

When the column data type in the CHANGE command window is TIME, the Time Specification window (Figure 3-4 on page 3-15) displays. You can either apply increments to the *change-string* for *rapid data generation* or perform date and time arithmetic (add or subtract) on the *find-string* for *data aging* purposes. After you provide the operator and increment information, press Enter on this screen to process the change and exit the window. Press END to cancel the CHANGE command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates would be 1998/01/31, 1998/02/28, 1998/03/28, etc.

Figure 3-4. CHANGE Command WIndow with Time Increment



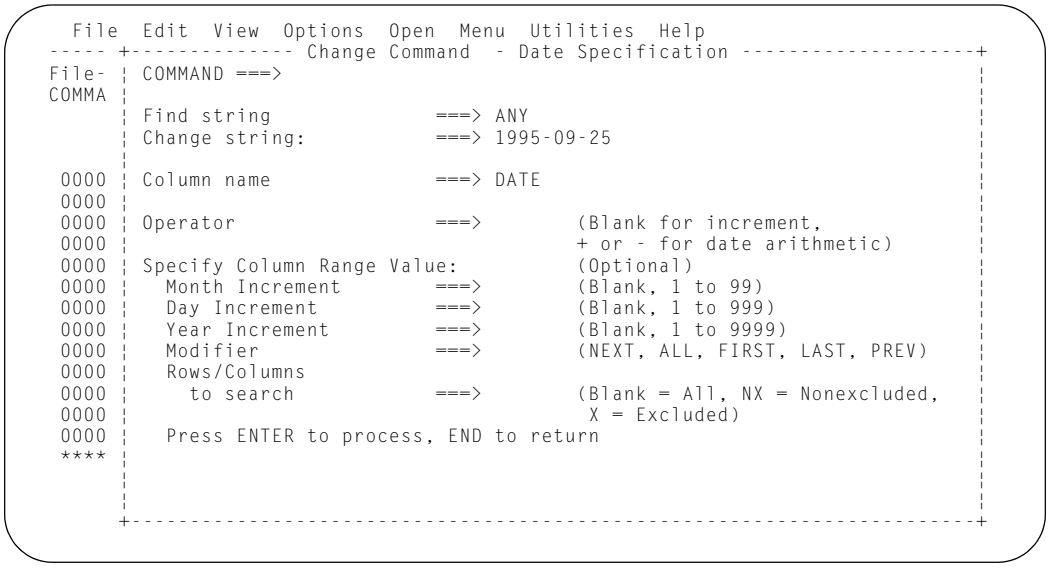
The Time Specification window is similar to the Timestamp Specification window, however, it only includes the increment fields for hour, minutes, and seconds. See "Timestamp Pop-up Window" on page 3-11 for a description of the fields.

Date Pop-up Window

When the column data type in the CHANGE command window is DATE, the Time Specification window (Figure 3-5 on page 3-16) displays. You can either apply increments to the *change-string* for *rapid data generation* or perform date arithmetic (add or subtract) on the *find-string* for *data aging* purposes. After you provide the operator and increment information, press Enter on this screen to process the change and exit the window. Press END to cancel the CHANGE command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates would be 1998/01/31, 1998/02/28, 1998/03/28, etc.

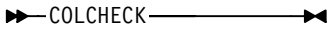
Figure 3-5. CHANGE Command window with Date Increment



The Date Specification window is similar to the Timestamp Specification window, however, it only includes the increment fields for month, day, and year. See "Timestamp Pop-up Window" on page 3-11 for a description of the fields.

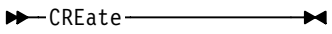
COLCHECK

The COLCHECK command is used to define column check constraints for a DB2 object in the Objects utility. See "Column Check Constraints" on page 8-13 for more information.



CREATE (CRE)

The CREATE command is used to define a new DB2 object in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Execute CREATE after you have entered the definition information on a DB2 Object Create screen. See Chapter 8, "Objects Utility" for more information.



CUT (CU)

CUT is only valid in Edit table mode. Use CUT to copy (append) one or more rows of data into the current paste buffer, or clipboard. Mark the rows to be copied with the C (for a single row) or CC (for multiple rows) line command (see also "C (Copy)" on page 4-3). The paste buffer stores the data until the next CUT REPLACE command overwrites the current content. Use the PASTE command to insert the paste buffer contents into the current table or use PASTE DS to copy its content to a dataset (see "PASTE (PAS)" on page

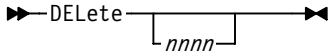
3-34). File-AID *for DB2* supports one paste buffer per session. The message **n Row(s) cut** indicates that the selected row(s) have been copied to the clipboard.

DEFAULT

The DEFAULT command is used to define default values for a DB2 object in the Objects utility. See "Default Values" on page 8-12 for more information.

DELETE (DEL)

The DELETE command deletes one or more rows of data, starting with the row you are currently viewing. DELETE is only valid in Edit row mode. It is not valid in table mode.

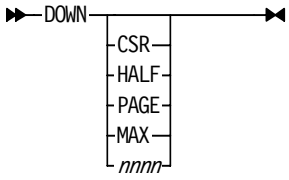


nnnn

Deletes *nnnn* number of rows. Defaults to one row.

DOWN

The DOWN command scrolls the data display by the specified scroll amount to reveal the data that follows the last line currently displayed. Unless you specify a temporary scrolling value with the DOWN command, the value in the SCROLL field is used. In row mode, DOWN scrolls to reveal the columns that follow the last column currently displayed. To view the next row in the table, issue the NEXT primary command.



CSR

Scrolls through the data, positioning the line where the cursor currently rests at the top of the data display. If the cursor is positioned outside of the data area, DOWN CSR has the same effect as DOWN PAGE.

HALF

Scrolls through the data half a page at a time. A page equals the number of lines of data displayed on the screen.

PAGE

Scrolls through the data one page at a time. A page equals the number of lines of data displayed on the screen.

MAX

Scrolls to the last line of data in the selected object.

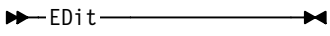
nnnn

Scrolls through the data *nnnn* lines from the top of the data display. Valid values are from 1 to 9999.

EDIT (ED)

The EDIT command initiates a multiple edit session. The table does not have to be related to the one currently being browsed or edited. The Edit Options window displays allowing you to enter a table name. See "Editing Multiple Unrelated Tables" on page 6-22 for more information on the EDIT command.

EDIT is not valid in Related Table Browse or Edit. For more information, see Chapter 7, "Related Table Browse and Edit".



END

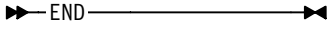
The END command ends the current File-AID *for DB2* function and takes you back to the previous screen.

When in an edit session, all changes are saved if AUTOSAVE is set to ON. If AUTOSAVE is set to OFF, the Edit End Confirmation window displays (Figure 6-12 on page 6-19).

When you end a multiple edit session (related or unrelated), all changes to all parent and dependent tables are saved if AUTOSAVE is ON. The active window and its dependent windows are closed. If AUTOSAVE is OFF, the Edit End Confirmation window (Figure 6-12 on page 6-19) displays for each table in which there are pending changes.

Note: If the Auto Commit option is ON, all changes to all tables are committed each time you press Enter during an edit session. Therefore, the Edit End Confirmation window is not displayed because there are no outstanding changes when you execute the END command.

For more information on Related Table Edit, see Chapter 7, "Related Table Browse and Edit". For information on unrelated edit, see "Editing Multiple Unrelated Tables" on page 6-22.



EXCLUDE (X)

The EXCLUDE command excludes all rows from an Edit display. It is only valid in table mode. The ALL keyword is required. Excluded rows are replaced in the display by information lines.

Once all rows are excluded, you can use the FIND command to display specific rows. Then, you can use the FLIP command to redisplay only the excluded rows and to exclude the visible rows. See "FLIP" on page 3-24 for more information.

To exclude one or more rows, use the X line command as described in Chapter 4, "Line Commands".

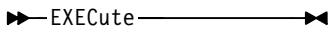
Use the S line command to redisplay one line. Use the RESET ALL primary command to redisplay all lines.

ALL

Excludes all rows. This is a required keyword.

EXECUTE (EXEC)

The EXECUTE command processes your request for data selection. It is active on all selection template screens, as well as some others. File-AID *for DB2* dynamically assigns this command to a PF key and displays it at the bottom of the screen on which it is supported.



FADB2

The FADB2 command is entered from the SQL Development and Analysis Edit screen (Option 4.3 from the Primary Option Menu). Most commonly, the command is issued without parameters to process one of the following line commands: BR (Browse), ED (Edit), EX (Execute), or XP (Explain). After typing the line command, type FADB2 in the COMMAND field and press Enter, or press the associated PF key. See Chapter 4, "Line Commands" for descriptions of the BR, ED, EX, and XP line commands.

Se "Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44 for more information on the FADB2 command.

Blank

Processes a SQL Development and Analysis line command (BR, ED, EX, or XP).

BOUNDS

Displays the current column boundaries. These boundaries delimit the code processed by the FADB2 command. Press any key to remove the FADB2 BOUNDS display.

position1 position2

Sets new boundaries to delimit the source. When the command executes, it returns a long message giving you the new boundaries. Use this form of the FADB2 command to eliminate line numbers, comments, or any other extraneous material that could interfere with processing. Any data outside the column boundaries is ignored by the SQL Development and Analysis line commands: BR, ED, EX, and XP.

FIND (F)

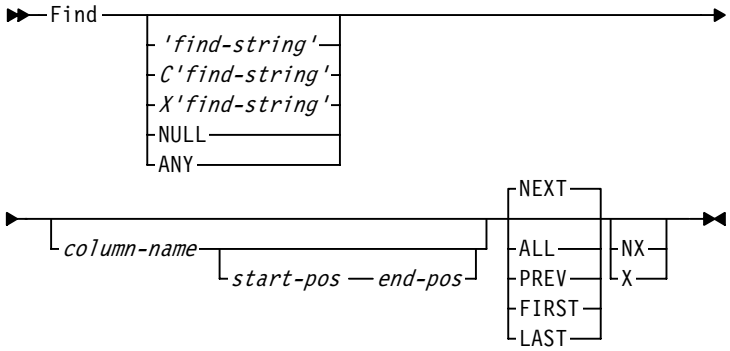
Using FIND During a Browse or Edit Session

The FIND command positions the cursor at the first occurrence of the specified value. It can be issued during a browse or edit session.

Note: Entering FIND with no parameters displays the FIND Command window (Figure 3-6 on page 3-23).

Use the column-name parameter with the FIND command to request that only data in the specified column is found. You may further restrict the data search by specifying positions within the column. If no column name is entered, all columns in the table are searched. However, the FIND command searches only certain column types based on the type of the *find-string* parameter. The chart in Table 3-2 on page 3-5 describes which columns are searched.

By default, the FIND command searches both excluded and non-excluded lines as well as both hidden and non-hidden columns. You can limit the search to only excluded rows and hidden columns or only non-excluded rows and non-hidden columns.



Find String

Value to be found. The first parameter for the FIND command is always the find-string. The find-string value is not case sensitive, unless enclosed in quotes and prefaced by C. Enclose the string in either single or double quotes when

- the *find-string* contains blanks
- the *find-string* matches the characters of reserved words, for example NULL, but you only want to change words with these characters in them. Valid reserved words for the FIND command are:

ANY (only in find string)	CURRENT_TIME
CUR	CURRENT TIMESTAMP
CURR	CURRENT_TIMESTAMP
CURRENT	USER
CURRENT DATE	NULL
CURRENT_DATE	NULLS
CURRENT TIME	

- the *find-string* contains a single quote (’), you must enclose it in double quotes.
- the *find-string* contains double quotes ("), you must enclose it in single quotes.

ANY

Finds *any* value in a column without any regard to its current value. Specify ANY as the 'find-string' but don’t enclose it in single or double quotes. You must also specify a column name when using the reserved word ANY.

If ANY is too general as a wildcard you may want to substitute picture strings for ANY as the find-string.

P

Specifies a picture string. P'picture' may contain one or a combination of the following:

=	any character
¬	any non-blank character
.	any undisplayable character
#	any numeric character
-	any non-numeric character
@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character

Note: Picture strings may not always achieve the expected results. For example, P'-' locates the blank filler character before an integer. Use ANY instead wherever possible.

X

Specifies that the value is in hexadecimal format. When specifying a hexadecimal value, you must enclose the find string in single or double quotes.

NULL

Specify to search for null values.

column-name

Name of the DB2 column to be searched for the specified value. If you do not specify a column name, the find is applied to all columns. You can specify a partial column name. If more than one column name matches the criteria, a window displays all matching columns, allowing you to select one. See Figure 3-1 on page 3-8 for a sample column selection window.

Start position

Starting position within a specified column name in which you want to restrict the search. If only one position is specified, it is processed as a starting position.

End position

Ending position within a specified column name in which you want to restrict the search. If only one position is specified, it is processed as a starting position.

Note: The Start/End positions are valid only for string and date/time fields.

Modifiers

Specify one of the following parameters to modify the search order.

NEXT

Default. Starts a search for the next occurrence of the *find-string* at the beginning of the first line being displayed (if the cursor is still on the COMMAND line), or at the cursor location (if the cursor is in the data part of the display). If you specify both a column name and the NEXT parameter, the next occurrence in the specified column of the *find-string* is found.

ALL

Finds all occurrences of the *find-string* starting at the beginning of the first line of the data and continues to the bottom of data. If you specify both a column name and the ALL parameter, all occurrences in the specified column of the *find-string* are found.

PREV

Starts a search for the previous occurrence of the *find-string* starting at the end of the line preceding the first line being displayed (if the cursor is still on the COMMAND line), or at the cursor location (if the cursor is in the data display). If you specify both a column name and the PREV parameter, the previous occurrence in the specified column of the *find-string* is found.

FIRST

Starts a search for the first occurrence of the *find-string* starting at the beginning of the first line of the data and continues until the string is found or the bottom of data is reached. If you specify both a column name and the FIRST parameter, the first occurrence in the specified column of the *find-string* is found.

LAST

Starts a search for the last occurrence of the *find-string* starting at the end of the last line of the data and continues backward until the string is found, or the top of data is reached. If you specify both a column name and the LAST parameter, the last occurrence in the specified column of the *find-string* is found.

Rows/Columns to Search:

Specify one of the following parameters to determine which rows or columns to search.

Blank

Default. Searches for the string in all rows and columns.

NX

Searches for the string only in those rows that are not excluded and columns that are not hidden.

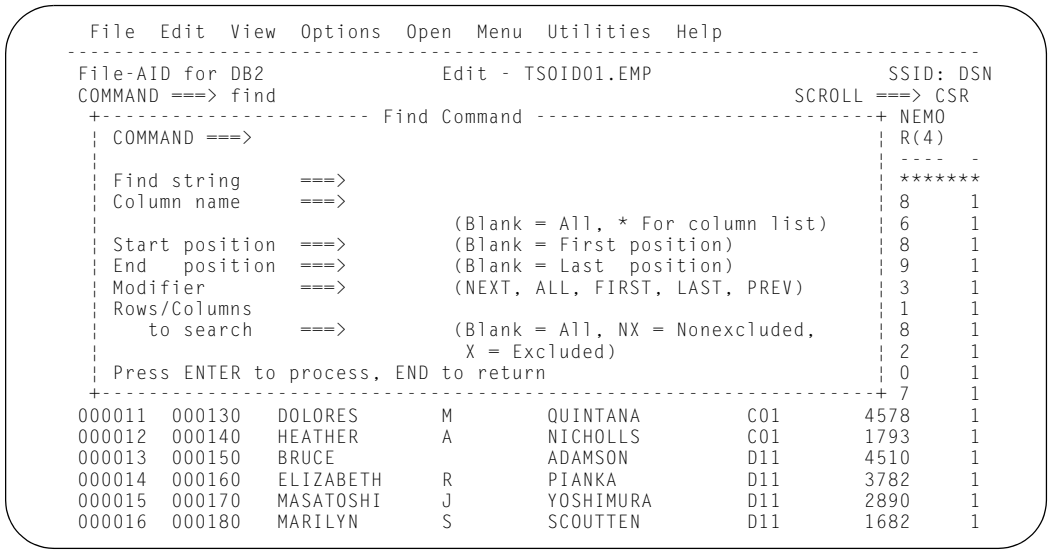
X

Searches for the string only in those rows that are excluded and columns that are hidden.

FIND Command Window

The FIND command window (Figure 3-6) prompts you for the information required to complete the find. It displays when you enter FIND with no parameters. Press Enter on this screen to process the find and exit the window. Press END to cancel the FIND command and exit the window.

Figure 3-6. FIND Command Window



Find string

Value to be found. To specify a hexadecimal value, enclose the string in single or double quotes and preface it with an X. To specify a null value, enter NULL. To specify a case sensitive string, enclose the string in quotes and preface it with a C.

Column-name

Name of the DB2 column in which you want to restrict the search. Entering an asterisk (*) presents a list of columns from which you can select a column name. You can also enter partial column names. If more than one column name matches the criteria, a window displays all matching columns, allowing you to select one. See Figure 3-1 on page 3-8 for a sample window. If you do not specify a column name, the search is applied to all columns.

Start position

Starting position within the specified column name in which you want to restrict the search. If no end position is specified, the search continues until the end of the column.

End position

Ending position within the specified column name in which you want to restrict the search. If no start position is specified, the search begins at the beginning of the column.

Note: The Start/End positions are valid only for string and date/time fields.

Modifier

See “Modifiers” on page 3-8 for an explanation of valid values.

Rows/Columns to search

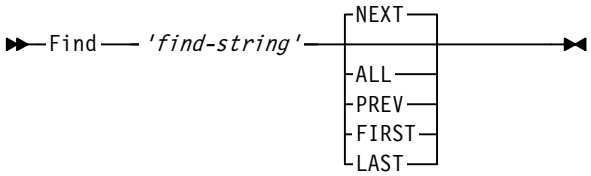
See “Rows/Columns to Search:” on page 3-9 for an explanation of valid values.

Using FIND in Relationship Object Lists

FIND command in relationship object lists:

The FIND command searches through data in the display and positions the line with the next occurrence of the specified find-string to the first line of the displayed data. It can be issued whenever File-AID *for DB2* displays a list in the relationships option.

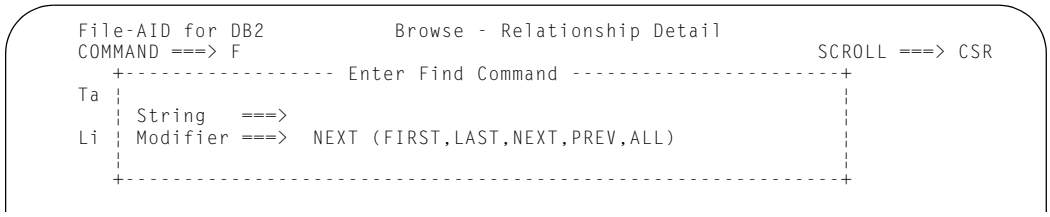
Note: Entering FIND with no parameters displays the window (Figure 3-7 on page 3-24).



FIND Parameter Window

When you enter the FIND command without any find-string, the Find Parameter window (Figure 3-7) displays. Enter the find-string and modifier.

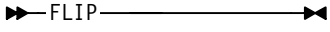
Figure 3-7. Enter Find Parameter window



FLIP

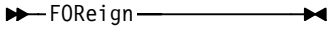
The FLIP command changes the display status of all rows with respect to exclusion. All excluded rows become visible and all visible rows become excluded. It is only valid in table mode. You must have excluded rows in the Edit display for this command to be effective.

Use the EXCLUDE ALL primary command (see "EXCLUDE (X)" on page 3-18) or the X line command (see "X (Exclude)" on page 4-6) to exclude rows.



FOREIGN (FOR)

The FOREIGN command defines foreign keys when you create or alter a table in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Execute FOREIGN on the appropriate screen when you have completed your changes. See Chapter 8, "Objects Utility" for more information.



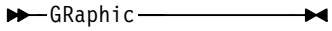
GRANT

The GRANT command grants privileges to a DB2 table, view, database, tablespace, plan, collection, package, schema, stored procedure, distinct type, or user defined function in the Privileges utility. File-AID *for DB2* dynamically assigns this command to a PF key and

displays it at the bottom of each screen on which it is supported. Execute GRANT on the appropriate screen when you have completed your changes. See Chapter 12, “Privileges Utility” for more information on using this command.

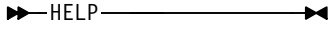
GRAPHIC (GR)

The GRAPHIC command displays a graph of the tables related to the table you are browsing or editing. For more information on using this command, see “Graphic Relationship Display” on page 7-42.



HELP

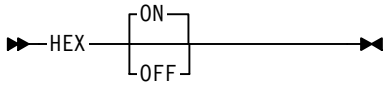
The HELP command invokes the help facility of File-AID *for DB2*. It causes a help screen to be displayed for the function you are performing. It also provides expanded information about an error message if one is currently displayed. When an SQL error is displayed, the HELP command provides the DB2 SQL error text returned from DB2.



HEX

The HEX command causes all data on the screen to be displayed in hexadecimal format. The commands HEX and HEX ON are equivalent. To restore data displayed in hexadecimal format to character format, issue the HEX OFF command.

The HEX command can be issued during a browse or edit session in row mode. It is not valid in table mode.



ON

Displays data in hexadecimal format.

OFF

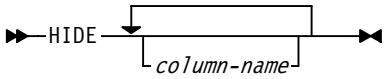
Displays data in character format.

HIDE

The HIDE command causes a column to be hidden from a browse or edit session. You can specify a column name in three ways.

- Enter one or more column-name parameters.
- Enter no parameters to display the window shown in Figure 3-8 on page 3-26.
- Type HIDE in the COMMAND field, place the cursor on the desired column data, and press Enter.

Columns remain hidden until you execute the RESET command (see "RESET (RES)" on page 3-48).



column-name

Name of the column to be hidden. You can enter partial column names. If more than one column name matches the criteria, the window in Figure 3-1 on page 3-8 displays the matching columns. See "Partial Column Names" on page 2-9 for more information. Specify multiple column names separated with a blank.

HIDE Command Window

The HIDE command window (Figure 3-8) lists each column in its initial display order and the column's status. It displays when you enter HIDE with no parameters. One or more columns can be hidden, held, or reset from the display.

Press Enter on this screen to process the commands and exit the window. Press END to cancel the HIDE command and exit the window.

Figure 3-8. HIDE Command window

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
F +------ Hide/Hold/Reset Command ------+                        SSID: DSN
C |                  ROW 1 TO 10 OF 14 |                  SCROLL ===> CSR
  | COMMAND ==>                        |    FIRSTNME      WORKDEPT  PHONENO
  |                                    | )  VC(12)        CHAR(3)   CHAR(4)
  |  Line Commands:                    | Y  ------------  --------  -------
  |    H - Hold   X - Hide   R - Reset | OF DATA ****************************
  |                                    |    CHRIS         A00       3971
  |         Column Name          Status|    MICHAEL       B01       3472
  |-------  ------------------  ------ |    SALLY         C01       4733
  | 000001  EMPNO                      |    JOHN          E01       6784
  | 000002  FIRSTNME                   |    IRVING        D11       6425
  | 000003  MIDINIT             HIDE   |    EVA           D21       7836
  | 000004  LASTNAME            HOLD   |    EILEEN        E11       5497
  | 000005  WORKDEPT                   |    THEODORE      E21       0978
  | 000006  PHONENO                    |    VINCENZO      A00       3499
  | 000007  HIREDATE                   |    SEAN          A00       2169
  | 000008  JOB                 HOLD   |    DOLORES       C01       4570
  | 000009  EDLEVEL                    |    ELIZABETH     D11       3786
  | 000010  SEX                        |    MASATOSHI     D11       2897
  +------------------------------------+    MARILYN       D11       1688
 000017  WALKER           DESIGNER  000190   JAMES         D11       2983
 000018  BROWN            DESIGNER  000200   DAVID         D11       4504
```

Line Command Area

Specify the line command you want to invoke for a column.

X, XX, Xn

Hide a column or columns. Hides a column from the display.

R, RR, Rn

Reset a column or columns. Restores a hidden or held column. See "RESET (RES)" on page 3-48 for more information.

H, HH, Hn

Hold a column or columns. Holds a column to the left side of the screen in table mode or to the top in row mode. See "HOLD" below for more information.

Column Name

Name of the table column.

Status

Indicates if the column is hidden or held by indicating HIDE or HOLD, respectively.

HOLD

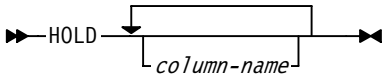
The HOLD command causes a column to be fixed, or "anchored", to the left side of the screen in table mode or to the top in row mode during a browse or edit session. You can specify a column name in three ways.

- Enter a column-name parameter.
- Enter no parameters to display the window shown in Figure 3-9 on page 3-28.
- Enter HOLD, place the cursor on the desired column data, and press Enter.

The column name is highlighted when a column is held. Columns are held in the order in which they appear in the table, not in the order in which they were held.

When you scroll the screen right or left, the held column(s) remain in view. Note that HOLD affects the displayed data only. It does not affect the actual contents of the table or the search order for FIND and CHANGE commands.

Columns remain held until you execute the RESET command. (See "RESET (RES)" on page 3-48.)



column-name

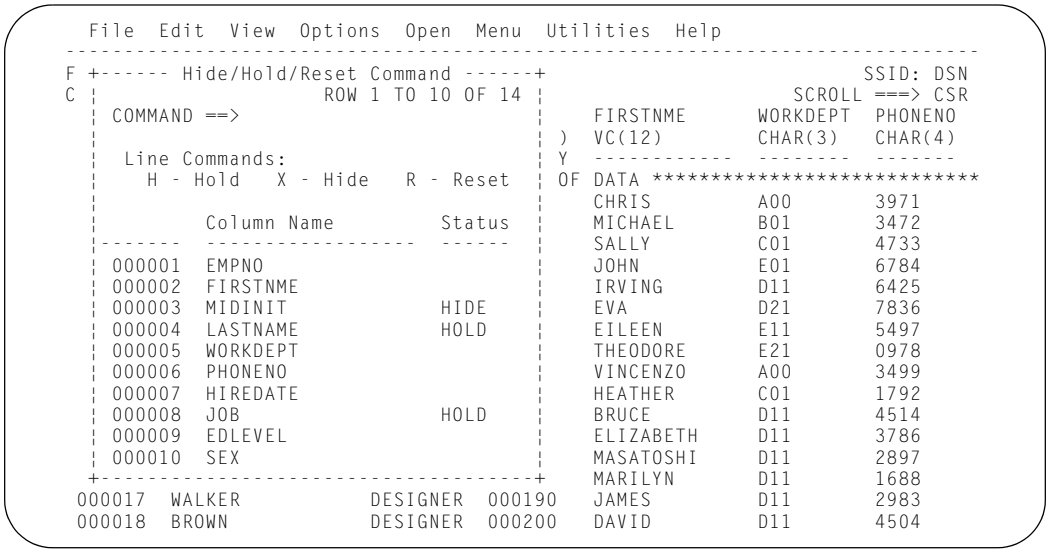
Name of the column to be held. You can enter partial column names. If more than one column name matches the criteria, the window in Figure 3-1 on page 3-8 displays the matching columns. See "Partial Column Names" on page 2-9 for more information. Specify multiple column names separated with a blank.

HOLD Command Window

The HOLD command window (Figure 3-9 on page 3-28) lists each column in its initial display order and the column's status. It displays when you enter HOLD with no parameters. You can select one or more columns to be held, hidden or reset from the display.

Press Enter on this screen to process the commands and exit the window. Press END to cancel the HIDE command and exit the window.

Figure 3-9. HOLD Command Window



Line Command Area

Specify the line command you want to invoke for a column.

H, HH, Hn

Hold a column or columns. Holds a column to the left side of the screen in table mode or to the top in row mode.

X, XX, Xn

Hide a column or columns. Hides a column from the display. See "HIDE" on page 3-25 for more information.

R, RR, Rn

Reset a column or columns. Restores a hidden or held column. See "RESET (RES)" on page 3-48 for more information.

Column Name

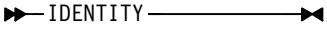
Name of the table column.

Status

Indicates if the column is hidden or held by indicating HIDE or HOLD respectively.

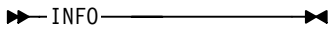
IDENTITY

The IDENTITY command displays the Identity Column Definition window (Figure 8-13 on page 8-16), where you can identify Identity column values for columns with a type of either INTEGER, SMALLINT, or DECIMAL with no scale.



INFO

The INFO command displays a window (Figure 3-10 on page 3-29) showing File-AID *for DB2* system information. The information in this window is for display only; the values may not be updated. INFO is valid on any File-AID *for DB2* screen.



Information Display Window

The INFO display window (Figure 3-10 on page 3-29) lists File-AID *for DB2* system information.

Figure 3-10. Information Display Window

```
File-AID for DB2 ---------------------- Edit Options ------------------------
COMMAND ===> INFO                                              SSID ===> DSN

S +---------------- Information Display -------------------+
  ¦                                                        ¦
  ¦ COMMAND ===>                                           ¦
  ¦                                                        ¦
  ¦ Location ID : DSNDETRT                                 ¦
S ¦ Subsystem ID: DSN                                      ¦  list)
  ¦ Plan Name   : FDPLAN01         SQL ID : TSOID01        ¦
  ¦ ISPF APPLID : F2DE             User ID: TSOID01        ¦
  ¦                                                        ¦
  ¦ File-AID for DB2 Version 4.8.1.xx                      ¦
S ¦ DB2 Version 6.1.1                                      ¦
  ¦                                    More:     +       ¦ ple rows per scree
n,
  ¦   Copyright (c) 1989-2005 Compuware Corporation.       ¦  per screen)
  ¦ All rights reserved. Unpublished - rights reserved     ¦
  ¦ under the Copyright Laws of the United States.         ¦
  ¦  ---------------------------------------------         ¦
  +----------------------------------------------------------+
   Relationship File   ===> 'FADB2.V4R8Mx.RELATE'
```

Location ID

Location identification of your DB2 system.

Subsystem ID

Current subsystem identification of your DB2 system.

SQL ID

Current SQL identification.

User ID

TSO user identification under which the user has attached to the DB2 subsystem.

Plan Name

Currently active File-AID *for DB2* plan.

ISPF Applid

Current File-AID *for DB2* ISPF APPLID.

File-AID *for DB2* **Version**

Current version of File-AID *for DB2*

DB2 Version

Current version of DB2.

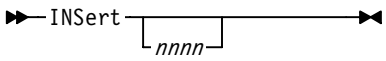
INSERT (INS)

The INSERT command inserts one or more new rows into a table. It is valid only in Edit row mode. (Use the INSERT line command when in Edit table mode.)

When you execute the INSERT command, the current row mode screen displays with no column values, allowing you to enter column data. If more than one row is inserted, you must issue the NEXT primary command to view the remaining inserted rows. The added row is indicated by the message **Insert Row Pending** under the COMMAND field.

New rows are inserted immediately following the last existing row viewed. Inserted rows are held in pending status until saved.

Note: For character columns with defined default values the Profile option "SPACES IMPLY NULL" must be ON so File-AID *for DB2* fills in the default values for blank columns.



nnnn

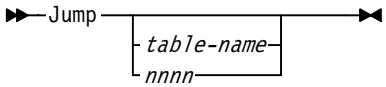
Inserts *nnnn* number of rows. Defaults to one row.

JUMP (J)

The JUMP command moves the cursor from window to window in a multiple table browse or edit session. Once you jump to a new window, it becomes active. You can only edit data and issue commands in the active window.

To facilitate moving from table to table, assign the JUMP command to a PF key.

See "Manipulating Windows" on page 7-30 and "Navigating Through Multiple Windows" on page 7-31 for more information on using the JUMP primary command.



Blank

Makes the next sequential window active. Wraps to the first window when the last window is reached.

Table name

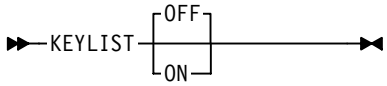
Makes the window displaying the specified table active. You only need to prefix the table name with the creator if the creator uniquely identifies the table. For example, you are editing both TSOID01.EMP and TSOID99.EMP.

nnnn

Makes the *nnnnth* window active.

KEYLIST

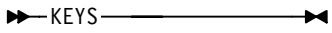
Calls the ISPF key list utility so you can browse and modify the File-AID *for DB2* keylists as defined for all Option 6, Relationships, panels.



Note: ON and OFF are only valid for ISPF version 4.1 or later.

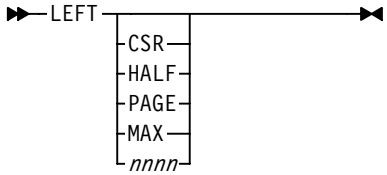
KEYS

The KEYS command displays the File-AID *for DB2* PF Key Definitions and Labels screen. The screen shows your default PF key settings and allows you to change them. Press END to exit the screen.



LEFT

The LEFT command scrolls to reveal data to the left of the current display. LEFT is only valid during a browse or edit session in table mode. It is not valid in row mode. Unless you specify a temporary scrolling value with the LEFT command, the value in the SCROLL field is used.



CSR

Scrolls through the data, making the current position of the cursor the right edge of the data display and showing the data to the left of that point. If the cursor is positioned outside the data area, LEFT CSR has the same effect as LEFT PAGE.

HALF

Scrolls through the data half a page at a time. A page equals the number of character spaces displayed on the screen. If scrolling a half page of data causes a column to be obscured, File-AID *for DB2* adjusts the scroll amount so that the column is not obscured.

PAGE

Scrolls through the data a page at a time. A page equals the number of character spaces displayed on the screen. If scrolling a page of data causes a column to be obscured, File-AID *for DB2* adjusts the scroll amount so that the column is not obscured.

MAX

Scrolls to the left-most data position of the selected object.

nnnn

Scrolls through the data *nnnn* characters. A valid entry is from 1 to 9999.

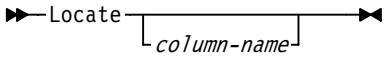
LOCATE (L)

The LOCATE command is used to locate a column in a browse or edit session or to find a value in an object list.

LOCATE in Browse or Edit

In Browse or Edit table mode, the LOCATE command scrolls the screen left or right to position the display at a specified column. In row mode, the screen scrolls up and down. You may specify a whole or partial column name.

Note: Entering LOCATE with no parameters displays the "LOCATE Command Window" on page 3-32.



column name

Column you want to locate. You can enter a partial column name. If more than one column name matches the criteria, the window displays only those matching columns.

LOCATE Command Window

The LOCATE command window (Figure 3-11 on page 3-32) lists all the columns in the table. It displays when you enter LOCATE with no parameters. Press Enter on this screen to process your selection and exit the window. Press END to cancel the LOCATE command and exit the window.

Figure 3-11. LOCATE Command window

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
F +------------------------ Column Selection --------------------------+  DSN
C |                                                 ROW 1 TO 11 OF 14 |  CSR
  | COMMAND ==>                                                       | NO
  |                                                                   |(4)
  |  Use 'S' to Select Column,  ENTER  to Process,  END  to Return    | --  -
  |                                                                   | *****
  | CMD  Column Name         Type(Length)                             |     1
  | ---  ------------------  ------------                             |     1
  |  _   EMPNO               CHAR(6)                                  |     1
  |  _   FIRSTNME            VC(12)                                   |     1
  |  _   MIDINIT             CHAR(1)                                  |     1
  |  _   LASTNAME            VC(15)                                   |     1
  |  _   WORKDEPT            CHAR(3)                                  |     1
  |  _   PHONENO             CHAR(4)                                  |     1
  |  _   HIREDATE            DATE                                     |     1
  |  _   JOB                 CHAR(8)                                  |     1
  |  _   EDLEVEL             SMALLINT                                 |     1
  |  _   SEX                 CHAR(1)                                  |     1
  |  _   BIRTHDATE           DATE                                     |     1
  +-------------------------------------------------------------------+     1
 000017  000190   JAMES         H        WALKER           D11       2986
 000018  000200   DAVID                  BROWN            D11       4501     1
```

Line command area

Type **S** to select the column to locate.

Column Name

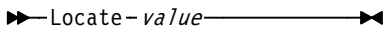
Name of the table column.

Type(Length)

Type and length of the table column.

LOCATE in an Object List

If positioned at an object list, the LOCATE command locates a value in an object list.

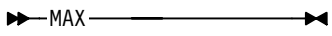


value

Value to be searched for.

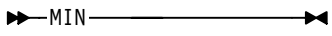
MAX

The MAX command enlarges a window to the maximum screen size. It is only valid in a multiple table browse or edit session. Once you execute the MAX command, all windows are maximized when using the JUMP command. See "Manipulating Multiple Tables" on page 6-25 and "Sizing Windows" on page 7-33 for more information on using the MAX primary command.



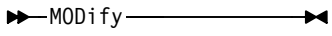
MIN

The MIN command reduces the size of a window to allow multiple table viewing. It is only valid in a multiple table browse or edit session. When in a related table display, one row from any parent table displays and the last dependent table opened covers the remaining window display. See "Manipulating Multiple Tables" on page 6-25 and "Sizing Windows" on page 7-33 for more information on using the MIN primary command.



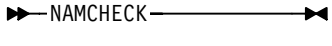
MODIFY (MOD)

The MODIFY command modifies synonyms and aliases in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Enter MODIFY in the COMMAND field of the appropriate screen when you have completed your changes. See Chapter 8, "Objects Utility" for more information.



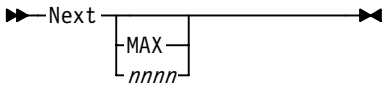
NAMCHECK

The NAMCHECK command is used to define named column check constraints for a DB2 object in the Objects utility. See "Named Check Constraints" on page 8-14 for more information.



NEXT (N)

The NEXT command displays the next row in a table in Browse or Edit row mode. It is not valid in table mode. If NEXT is issued and the last row in the table is already displayed, the message **BOTTOM OF DATA REACHED** displays. To view a previous row, issue the PREV command.



MAX

Scrolls to the last row in the table.

nnnn

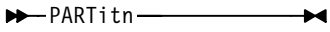
Scrolls forward *nnnn* number of rows. A valid entry is from 1 to 9999.

PARMS

Brings up the Parameter Definition window (Figure 8-59 on page 8-79).

PARTITN (PART)

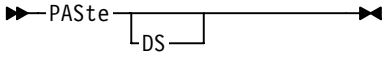
The PARTITN command creates a partitioned tablespace or index in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Enter PARTITN in the COMMAND field of the appropriate screen when you have completed your changes. See Chapter 8, “Objects Utility” for more information.



PASTE (PAS)

PASTE is only valid in Edit table mode. Use the PASTE command to copy the contents of the current paste buffer, or clipboard, into the current table or into a dataset (PASTE DS). When pasting into the current table mark the destination row with the A (after current row), or B (before current row) line command (see also “A (After)” on page 4-2 and “B (Before)” on page 4-2). When you press Enter, the “Paste Column Selection Window” on page 3-35 displays so you can verify column matching.

The paste buffer keeps its content until the next CUT REPLACE command overwrites the current content (see “CUT (CU)” on page 3-16). File-AID *for DB2* keeps the paste buffer active for the entire File-AID *for DB2* session.



DS

Saves the contents of the paste buffer to a dataset, in delimited format.

Paste Dataset

The Paste Dataset screen displays when you enter the PASTE DS command. Use it to specify the output dataset.

File-AID *for DB2* prefills the dataset name with a fully-qualified dataset name constructed from the user ID and the current object name. The dataset name may be modified, if necessary. Figure 3-12 shows a sample screen.

Figure 3-12. PASTE Dataset screen

```
File-AID for DB2 ---------- Paste  table  USERID.EMP --------------- SSID: DSN
COMMAND  ===>

Paste dataset format is delimited:
   Change Delimited Options  ===> YES            (Yes or No)

Specify Output Datasets:
  PASTE Dataset        ===> 'USERID.*.UNLOAD'
     Member            ===>                    (* for member list)


An asterisk (*) in the above Dataset names will be replaced with:

   Table Name            ===> EMP2
```

Paste Dataset Format is Delimited

Change Delimited Options

Specify whether you want to change the current delimited file format options or not. Enter **Yes** to view and change the current options (see "Delimited Format Options" on page 14-12). Enter **No** if you do not want to view or change the current delimited format options.

Specify Output Datasets Area

PASTE Dataset

The name of the dataset which contains the pasted data. The second level qualifier is an asterisk (*****) that is replaced with the table name specified below. The dataset may be either sequential, partitioned, new or existing. If the dataset exists, its LRECL **must** be greater than or equal to the calculated record length of the output data record. If it is not, you receive an allocation error.

If the dataset is new, Figure 14-11 on page 14-14 displays. It may be allocated as either sequential or partitioned. The default is to create the dataset as sequential

Member

If the file is a partitioned dataset, enter an existing or new member name. Enter an asterisk (*****) to select a member from a member list.

Table Name

The table name that replaces the asterisk (*****) in the above dataset name.

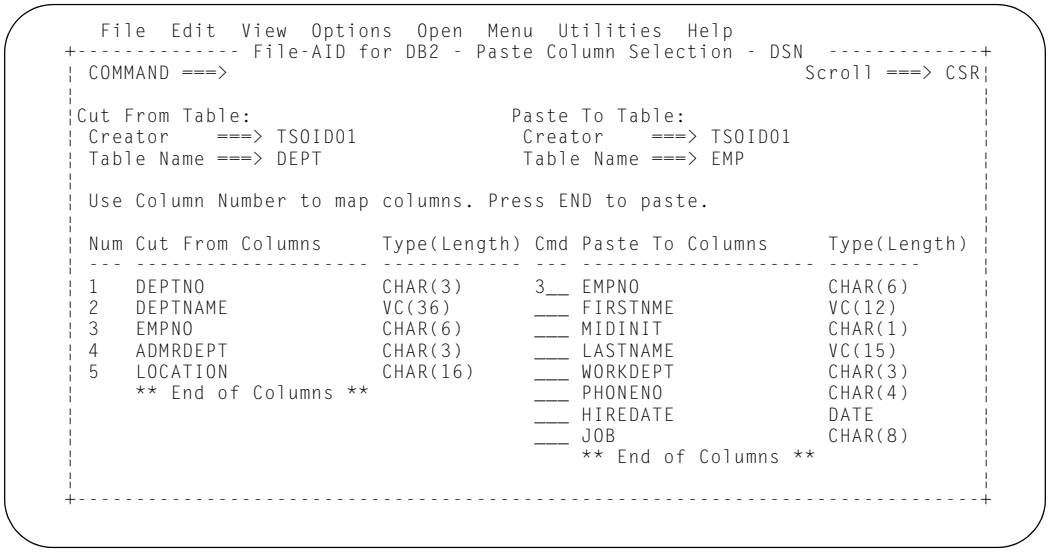
Once you specify the dataset name File-AID *for DB2* continues either with "Delimited Format Options" on page 14-12 , or "Extract Dataset Allocation" on page 14-13 to save the contents of the paste buffer in the specified dataset.

Paste Column Selection Window

Before File-AID *for DB2* executes the PASTE command it allows you to select the columns you want to paste into your current table. When column names of the paste buffer and destination table do not match, you can manually select and match them with the Paste

Column Selection Window. The Paste Column Selection Window (Figure 3-13) prompts you for the information required to complete the paste operation.

Figure 3-13. PASTE Column Selection window



Selecting Columns

Similar to Building Column Relationships (compare "Building Column Relationships" on page 7-20), you match column names in the Paste Column Selection window.

The fields in the lower portion of the Column Selection screen comprise two lists. These lists contain names, data types, and lengths of columns from the Cut From and Paste To tables specified in the upper portion of the screen. Use these lists to select and match column names.

Follow this procedure to select and match column names:

1. Review the prefilled information. In this example, 3 is prefilled in the CMD field for Paste To column EMPNO, same column name, type, and length. If it matches what you want, go to step 5.
2. Decide which Cut From and Paste To columns you want to select.
3. Enter the number from the column NUMber field that corresponds to the Cut From column into the CMD column next to the name of the Paste To column.

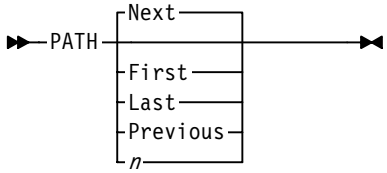
   Corresponding Cut From and Paste To columns must have the same data type, length, and nullability. Otherwise, File-AID *for DB2* displays an error message and does not perform the paste.

   For example: You want to paste the data of the DEPTNO column from the DEPT table to the WORKDEPT column of the paste to table EMP (see Figure 3-13). Since the DEPTNO column number is 1, enter a 1 in the CMD field next to the WORKDEPT column of the Paste To table. Both of these columns have the same column type and length: CHAR(3).
4. Enter the matching numbers for all remaining columns you want to select for the paste operation.
5. Enter END to submit the column selections.

The paste operation concludes by filling all matched columns.

PATH

PATH is only valid in Related Table Edit mode. Use the PATH command to switch to another related table path for the current driving table of the Related Table Edit session. (see also "REDIT (RE)" on page 3-42 and "Opening Additional Related Table Paths" on page 7-36).



Next

Switches to the next path.

First

Switches the active window to the first window of the first path **(PATH 1 OF 9)**.

Last

Switches to the last path **(PATH 9 OF 9)**.

Previous

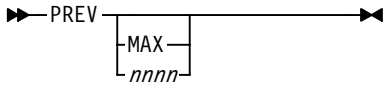
Switches to the previous path.

n

Number - Displays path number *n*, for example, PATH 2 displays the second path.

PREV (P)

The PREV command displays the previous row in a table in Browse or Edit row mode. It is not valid in table mode. If PREV is issued and the first row in the table is already displayed the message **TOP OF DATA REACHED** displays. To view the next row, issue the NEXT command.



MAX

Scrolls to the first row in the table.

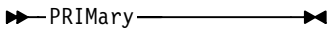
nnnn

Scrolls backward *nnnn* number of rows. A valid entry is from 1 to 9999.

PRIMARY (PRIM)

The PRIMARY command defines primary keys when creating or altering a table in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Enter

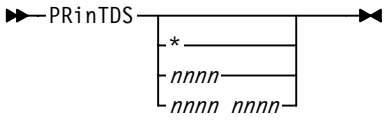
PRIMARY in the COMMAND field of the appropriate screen when you have completed your changes. See Chapter 8, “Objects Utility” for more information.



PRINTDS (PRTDS)

The PRINTDS command prints all rows displayed on the screen, a specified number of rows displayed on the screen, all rows of a displayed table or view, or a specified number or range of rows in a displayed table or view. Output is routed to the dataset that you specify. The default attributes of the output dataset are set in Option 0, Defaults and User Parms. See “Option 7, Print Options” on page 5-10.

You can execute the PRINTDS command in Browse or Edit mode.



Blank

Prints the screen currently being displayed.

*** (asterisk)**

Prints from the first row in the current display window to the end of the table.

nnnn

Number of rows to be printed starting with the first row in the current display window. A valid entry is from 1 to 9999.

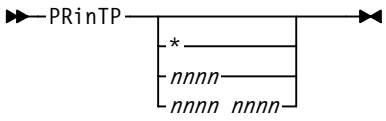
nnnn nnnn

Range of rows to be printed. A valid range is from 1 to 9999.

PRINTP (PRTP)

The PRINTP command prints all rows displayed on the screen, a specified number of rows displayed on the screen, all the rows of a displayed table or view, or a specified number or range of rows in a displayed table or view. Output is routed either to a local printer or a JES queue. The default printer attributes are set through Option 0, Defaults User Parms. See “Option 7, Print Options” on page 5-10.

You can execute the PRINTP command in Browse or Edit mode. PRINTP is also valid in the SQL Development and Analysis function.



Blank

Prints the screen currently being displayed.

*** (asterisk)**

Prints from the first row in the current display window to the end of the table.

nnnn

Number of rows to be printed starting with the first row in the current display window. A valid entry is from 1 to 9999.

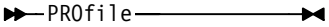
nnnn nnnn

Range of rows to be printed. A valid range is from 1 to 9999.

PROFILE (PRO)

The PROFILE command displays a window (Figure 3-14 on page 3-39) showing your current user profile defaults and allows you to update them. It is valid during a browse or edit session.

When you make changes to a user profile, they are applied immediately and remain in effect until you alter them. Each user's profile is independent. Modifications to one user profile have no effect on other user profiles. Changes to some options may not be realized until the next browse or edit session. These options have an asterisk next to them in the Profile Parameters window.



Profile Parameters Window

The Profile Parameters window (Figure 3-14) displays your current user profile defaults. Type over the parameter values to make a change. Press END to exit the window and save your changes. Type CANCEL to exit the window and cancel your changes.

Figure 3-14. Profile Parameters Window

```
File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
F +-------------------- PROFILE Parameters --------------------+ SSID: D701
C | COMMMAND ==>                                               | OLL ===> PAGE
  |                                                            |
  |   AUTOCOMMIT              : ON       (ON or OFF)           |
  |   AUTOSAVE                : ON       (ON or OFF)           |
  |   CAPS                    : OFF      (ON or OFF)           | ************
  | * OBJECT NAMES UPPERCASE  : ON       (ON or OFF)           |
  | * MAXROWS                 : 2000     (* = all rows)        |
  |   NON-DISPLAY             : ~                              |
  |   NULL ENTRY              : @                              |
  |   NULL DISPLAY            : -                              |
  |   SPACES IMPLY NULL       : OFF      (ON or OFF)           |
  |   VARIABLE LENGTH MARK    : #                              |
  |   UPDATE PRIMARY KEY      : OFF      (ON or OFF)           |
  | * SHOW FOREIGN KEYS       : OFF      (ON or OFF)           |
  |   VARIABLE LENGTH TRUNCATE: ON       (ON or OFF)           |
  |   INSERT DT/TM/TS DEFAULT : 1        (1 = LOCAL, 2 = GMT)  |
  | * RETRIEVE UNIQUE KEY INFO: OFF      (ON or OFF)           |
  |   REPEAT COMMAND PROMPT   : ON       (ON or OFF)           |
  | * UPDATE IDENTITY COLUMN  :          (ON or OFF)           |
  | * Changes will take effect in the next Edit/Browse session |
  +------------------------------------------------------------+
```

AUTOCOMMIT

Specify ON to commit changes each time you press Enter during an edit session. In Related Table Edit, all changes to all tables are saved each time you press Enter. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

AUTOSAVE

Specify ON to save changes each time you press END during an edit session. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

When you issue the END command in a multiple edit session (related or unrelated), all outstanding changes in all open tables are saved if AUTOSAVE is ON. The active window and its dependent windows are closed. If AUTOSAVE is OFF, the Edit End Confirmation window (Figure 6-12 on page 6-19) displays for each table in which there are pending changes. For more information on Related Table Edit, see Chapter 7, "Related Table Browse and Edit". For more information on unrelated edit, see "Editing Multiple Unrelated Tables" on page 6-22.

Note: If the Auto Commit option is ON, all changes are committed each time you press Enter during an edit session. Therefore, the Edit End Confirmation window is not displayed because there are no outstanding changes when you execute the END command. See "AUTOCOMMIT" above for information on setting the Auto Commit option.

Since AUTOSAVE is a profile option, its status remains in effect until you change it.

CAPS

Specify ON to automatically change all data entered in lowercase characters to uppercase characters. You can also set this option with the CAPS primary command or through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

OBJECT NAMES UPPERCASE

Specify ON to automatically change all object names entered in lowercase characters to uppercase characters. Specify OFF if you need to edit a table with a DBCS table name or DBCS column name. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

MAXROWS

Specify a numeric value or an asterisk (*) to denote the number of rows to be selected. You can also change this value for individual browse or edit sessions. See "Maximum Rows to Select" on page 6-3 for more information.

NON-DISPLAY

Specify the character that File-AID *for DB2* uses to signify nondisplayable data when in browse mode. Does not apply for edit mode. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

NULL ENTRY

Specify the character that, when entered, sets a column to null. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

NULL DISPLAY

Specify the character that displays for null columns. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

SPACES IMPLY NULL

Specify ON to have spaces imply nulls for character columns. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

VARIABLE LENGTH MARK

Specify the character used to indicate the end of a column. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

UPDATE PRIMARY KEY

Specify ON to update primary keys during an edit session. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

Note: In order to be able to update primary key columns in a ***view***, set RETRIEVE UNIQUE KEY INFO to ON before your view editing session. This setting also determines if the primary key, foreign key, and identity column labels appear for views that are being edited or browsed.

SHOW FOREIGN KEYS

Specify ON to show foreign keys during an edit session. Foreign Key information is only shown for local tables. Because DB2 does not maintain an index on the table containing foreign key information, setting this option to ON may have a negative impact on performance. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

VARIABLE LENGTH TRUNCATE

Specify ON to truncate trailing spaces on an insert or update during an edit session. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

INSERT DT/TM/TS DEFAULT

Specify 1 for LOCAL or 2 for GMT (Greenwich Mean Time) to determine the date/time format used as the default for fields defined as not null with default and when CURRENT is used to enter a date, time, or timestamp value. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2. See "CURRENT Date/Time" on page 2-8 for more information on entering data and time fields.

RETRIEVE UNIQUE KEY INFO

Specify ON to retrieve unique key information before the edit session begins. This value applies only if there is no primary key. Unique key information is only retrieved for local tables. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2.

REPEAT COMMAND PROMPT

Specify ON to display the REPEAT Column Range Specification window (see "REPEAT Window" on page 3-43) when you enter the REPEAT primary or line command. Specify OFF to suppress the pop-up window.

UPDATE IDENTITY COLUMN

Specify ON to allow update of identity columns in edit. You can also set this option through Option 0, Defaults and User Parms. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2. This applies only to columns set to GENERATED BY DEFAULT. Columns set to GENERATED ALWAYS cannot be edited. If the column is both an identity column and a primary key, then UPDATE PRIMARY KEY must also be ON to edit the column.

RBROWSE (RB)

The RBROWSE command initiates a Related Table Browse session for a parent or dependent table specified in the relationship definition.

Relationship definitions are defined through DB2 referential integrity or through a user-defined application relationship. If no relationship definition exists for the current table, you can add an application relationship or select a model definition if available.

RBROWSE is valid in Edit or Browse mode. When editing or browsing a related *dependent* table it can be specified in any window.

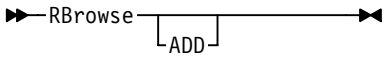
You can only open a parent table with RBROWSE when its dependent table is the only open table.

RBROWSE is disallowed when the parent table was opened with a previous REDIT or RBROWSE. RBROWSE is also disallowed in a multiple, *unrelated* table edit or browse session.

The RBROWSE command is cursor sensitive. To select a specific row, place the cursor on the desired line before pressing Enter. If a row is not selected, the top row is the default selected row.

See Chapter 7, "Related Table Browse and Edit" for more information on the RBROWSE command.

See "Editing Multiple Unrelated Tables" on page 6-22 for more information on editing unrelated tables.



ADD

Displays the Column Relationships screen, allowing you to add a new application relationship. See "Creating a New Relationship" on page 7-19 for more information on adding relationships.

RCHANGE

The RCHANGE command repeats the execution of the last CHANGE command executed during the current edit session. If the last CHANGE command included an increment, RCHANGE will continue the increment.

REDIT (RE)

The REDIT command initiates a Related Table Edit session for a parent or dependent table specified in the relationship definition.

Relationship definitions are defined through DB2 referential integrity or through a user-defined application relationship. If no relationship definition exists for the current table, you can add an application relationship or select a model definition if available.

REDIT is valid in Edit or Browse mode. When editing or browsing a related *dependent* table it can be specified in any window.

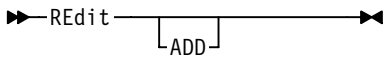
You can only open a parent table with REDIT when its dependent table is the only open table.

REDIT is disallowed when the parent table was opened with a previous REDIT or RBROWSE. REDIT is also disallowed in a multiple, *unrelated* table edit or browse session.

The REDIT command is cursor sensitive. To select a specific row, place the cursor on the desired line before pressing Enter. If a row is not selected, the top row is the default selected row.

See Chapter 7, "Related Table Browse and Edit" for more information on the REDIT command.

See "Editing Multiple Unrelated Tables" on page 6-22 for more information on editing unrelated tables.



ADD

Displays the Column Relationships screen, allowing you to add a new application relationship. See "Creating a New Relationship" on page 7-19 for more information on adding relationships.

REMARKS

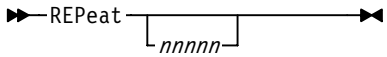
Used in trigger, stored procedure, and UDF create and alter to define the text for a DB2 COMMENT ON statement. The comment on text is a maximum of 254 characters for DB2 Version 6 and 7, and a maximum of 762 characters for DB2 Version 8.

REPEAT (REP)

The REPEAT command adds one or more duplicate rows to a table. It is valid only in Edit row mode. (Use the REPEAT line command when in Edit table mode.)

When your user profile has the Repeat Command Prompt set to ON (see "PROFILE (PRO)" on page 3-39) the REPEAT Column Range Specification window displays (see Figure 3-15 on page 3-44). If you repeat the row more than once, you must issue the NEXT command to view the additional rows.

The added row is indicated by the message under the COMMAND field, **Row Repeat Pending**. The row is placed in a pending state, allowing you to make any further changes until the data is saved.



nnnnn

Repeats the row *nnnnn* number of times. Defaults to one row.

Note: While you can repeat a row up to 99,999 times, consider possible consequences, for example not sufficient memory allocation, before issuing a high repeat value.

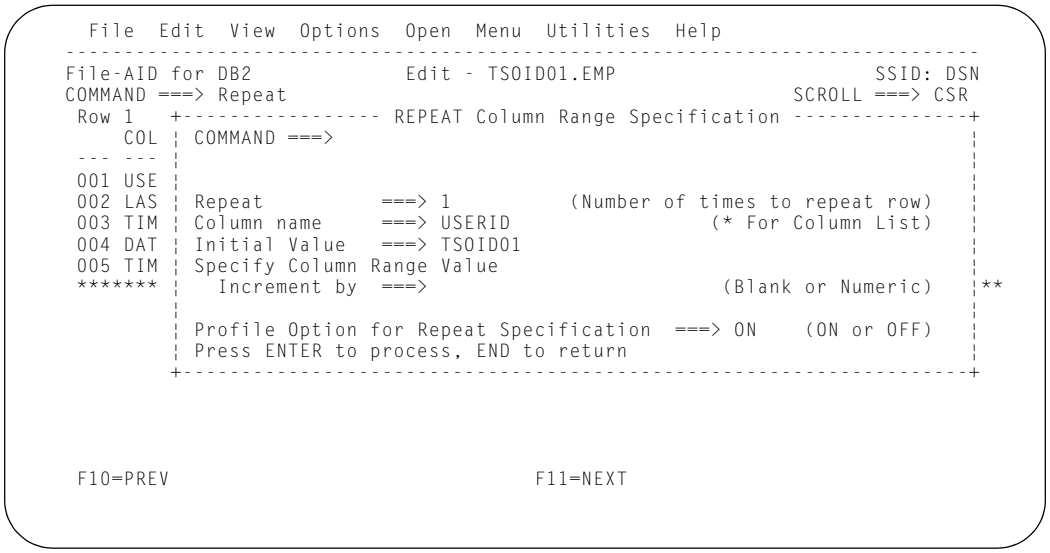
REPEAT Window

The REPEAT command window (Figure 3-15 on page 3-44) provides additional rapid data generation functionality. It works similar to the CHANGE command window (see "CHANGE Command Window" on page 3-9). Just specify an increment for a column value. File-AID *for DB2* will repeat the row the specified number of times with the

incremented values filled in the column. It is only displayed when your user profile has the Repeat Command Prompt set to ON (see "PROFILE (PRO)" on page 3-39). The screen prefills the Repeat field (1), the Column name field (last selected column, or first column in primary key, or first column) and the Initial value field (selected column's data).

After you make your changes to the fields, press Enter on this screen to process the repeat request and exit the window. If you want the rows repeated without changed column data, do not change any fields and press Enter. Press END to cancel the REPEAT command and exit the window.

Figure 3-15. REPEAT Column Range Specification Window



Repeat

Enter the number of times the row should be repeated. Default is 1. If you entered the REPEAT n command the value for n is prefilled.

Column name

Name of the DB2 column in which you want to apply the column range increment. The first column in primary key or first column is prefilled by default. Entering an asterisk (*) presents a list of columns from which you select a column name. You can also enter partial column names. If more than one column name matches the criteria, a window displays all matching columns, allowing you to select one. See Figure 3-1 on page 3-8 for a sample window. You must specify a column name, or you receive an error message.

Initial Value

Value in the selected column for the first repeated row. Prefilled with the selected column's data (maximum 40 characters, including quotes).

Note: The initial value will replace the entire column's value even when the column length is greater than 40. Blank out this field when you do not want to change the column's value.

Enclose the initial value in either single or double quotes when:

– the *initial value* contains blanks

– the *initial value* matches the characters of reserved words, but you only want to use these words as constants, not as reserved words. Valid reserved words for the REPEAT command are:

CUR	CURRENT_TIME
CURR	CURRENT TIMESTAMP
CURRENT	CURRENT_TIMESTAMP
CURRENT DATE	USER
CURRENT_DATE	NULL
CURRENT TIME	NULLS

– the *initial value* contains a single quote (’), you must enclose it in double quotes
– the *initial value* contains double quotes ("), you must enclose it in single quotes
– the *initial value* must not be changed to uppercase characters

To specify a hexadecimal value, enclose the string in single or double quotes and preface with an X. To specify a null value, enter NULL. To specify a case sensitive string, enclose the string in quotes and preface it with a C.

When you also fill in a value in the Increment by field the initial value will be modified and incremented for all repeated rows. For example, 100 is the initial value for the Employee Number. This value is used as the base for the modification. Then, the Increment by value, for example 10, modifies the initial value. Thus, the following Employee Number values of a four (4) time repeat will be 110 (first new row), 120, 130, and 140. The new value must fit the required format and length for the selected column, or you receive an error message.

Increment by

Increment to modify alphabetic (a through z) and numeric characters (0 through 9) of the initial value. Other characters (e.g %, #, =, etc.) are not changed by the increment value. The default increment is blank (no increment). The increment is always a numeric value. A negative value, for example (-1) is valid and will decrease from the initial value (100 initial value: 99, 98, 97, etc.). Positive increment is assumed and must be entered without a sign.

The column data type determines how increments are interpreted. Date and time actually have additional pop-up displays (see “Timestamp Pop-up Window” on page 3-11, “Time Pop-up Window” on page 3-14, and “Date Pop-up Window” on page 3-15).

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a one month increment for a starting date of 1998/01/31, the resulting dates are 1998/01/31, 1998/02/28, 1998/03/28, etc.

Specify an increment when you want to repeat a row with one REPEAT command and avoid duplicate data.

When the initial value is an alpha character, for example A, and the increment is 2, the resulting string will be C, E, G, I, K, etc.

When the initial value is alphanumeric, for example AB3 and the increment is 4, the resulting string will be AB7, AC1, AC5, AC9, AD3, etc.

Alphabetic and numeric data values wrap when they reach the maximum allowable value for the column, starting over at 0 (numeric) or A (alphabetic). Alphabetic data includes all letters of the alphabet in upper and lower case. Integers wrap at their maximum values, for example, increment -1 wraps from -2147483648 to 2147483647, increment 1 wraps from 2147483647 to 0.

Profile Option for Repeat Specification:

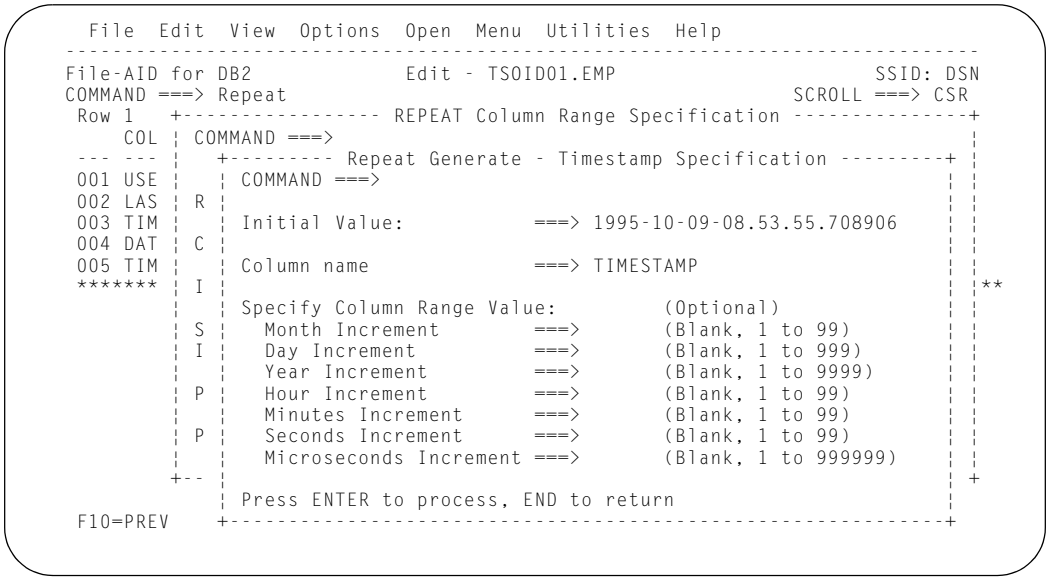
ON or OFF. ON = Repeat always brings up the Column Range Specification window. OFF = Column Range Specification window is not enabled for the REPEAT command. Use the PROFILE command's REPEAT COMMAND PROMPT option (see "PROFILE (PRO)" on page 3-39) to enable the Column Range Specification window again after it is turned OFF.

Timestamp Pop-up Window

When the column data type for an incremental repeat is TIMESTAMP, you can specify the increment(s) in more detail in the following Timestamp Specification window (Figure 3-16 on page 3-46). After you provide the increment information, press Enter on this screen to process the repeat and exit the window. Press END to cancel the REPEAT command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates would be 1998/02/28, 1998/03/28, etc.

Figure 3-16. REPEAT Generate - Timestamp Specification Window



Initial value

Prefilled with the value you specified in Repeat Column Range Specification window. You are allowed to change the prefilled Initial value.

Column name

Prefilled with the value you specified in Repeat Column Range Specification window.

Month Increment

Increment to modify the month portion of the timestamp. The default increment is blank (no increment). Enter the desired month increment (1 to 99).

Day Increment

Increment to modify the day portion of the timestamp. The default increment is blank (no increment). Enter the desired day increment (1 to 999).

Year Increment

Increment to modify the year portion of the timestamp. The default increment is blank (no increment). Enter the desired year increment (1 to 9999).

Hour Increment

Increment to modify the hour portion of the timestamp. The default increment is blank (no increment). Enter the desired hour increment (1 to 99).

Minutes Increment

Increment to modify the minutes portion of the timestamp. The default increment is blank (no increment). Enter the desired minutes increment (1 to 99).

Seconds Increment

Increment to modify the seconds portion of the timestamp. The default increment is blank (no increment). Enter the desired seconds increment (1 to 99).

Microseconds Increment

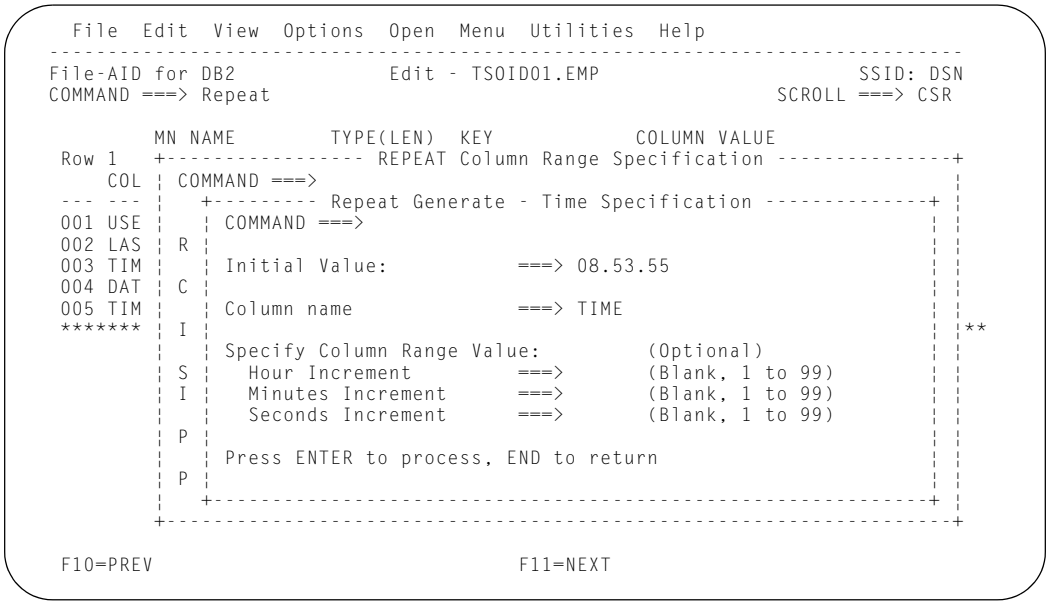
Increment to modify the microseconds portion of the timestamp. The default increment is blank (no increment). Enter the desired microseconds increment (1 to 999999).

Time Pop-up Window

When the column data type for an incremental repeat is TIME, you can specify the increment(s) in more detail in the following Time Specification window (Figure 3-17 on page 3-47). After you provide the increment information, press Enter on this screen to process the change and exit the window. Press END to cancel the REPEAT command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates would be 1998/02/28, 1998/03/28, etc.

Figure 3-17. REPEAT Generate - Time Specification Window



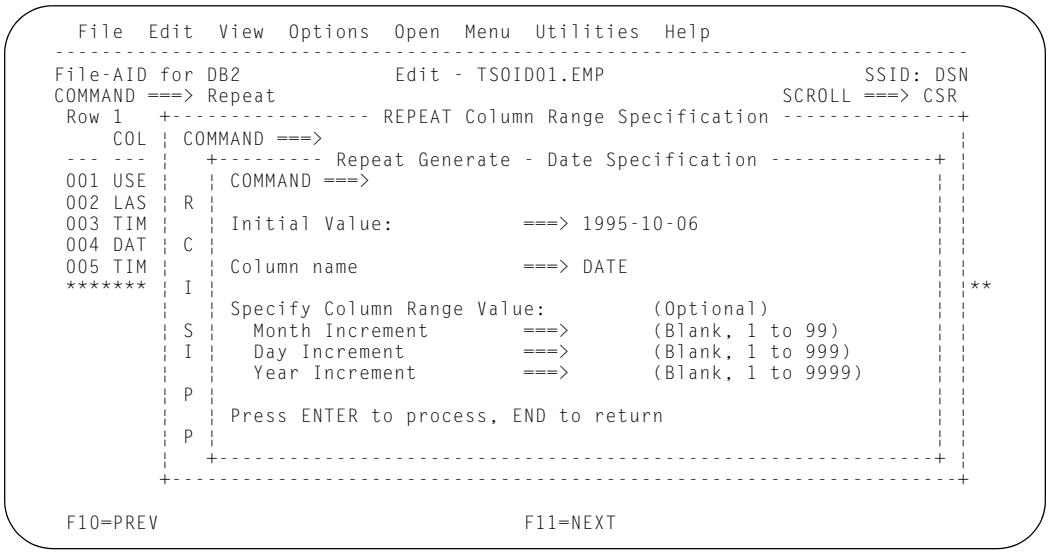
The Time Specification window is similar to the Timestamp Specification window, however, it only includes the increment fields for hour, minutes, and seconds. See "Timestamp Pop-up Window" on page 3-46 for a description of the fields.

Date Pop-up Window

When the column data type for an incremental repeat is DATE, you can specify the increment(s) in more detail in the following Date Specification window (Figure 3-18 on page 3-48). After you provide the increment information, press Enter on this screen to process the change and exit the window. Press END to cancel the REPEAT command and exit the window.

Note: Date and time increments always result in valid DB2 date and time values. For example, if you use a 1 month increment for a starting date of 1998/01/31, the resulting dates would be 1998/01/31, 1998/02/28, 1998/03/28, etc.

Figure 3-18. REPEAT Generate - Date Specification Window

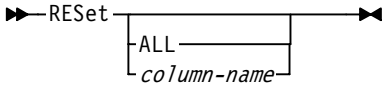


The Date Specification window is similar to the Timestamp Specification window, however, it only includes the increment fields for month, day, and year. See "Timestamp Pop-up Window" on page 3-46 for a description of the fields.

RESET (RES)

The RESET command restores hidden and held columns, and excluded and blank inserted lines during a browse or edit session. You may specify a whole or partial column name.

Note: Entering the command with no parameters displays the window discussed in "RESET Command Window" on page 3-49.



ALL

Resets all hidden and held columns, excluded and blank inserted rows, and pending row markers.

column-name

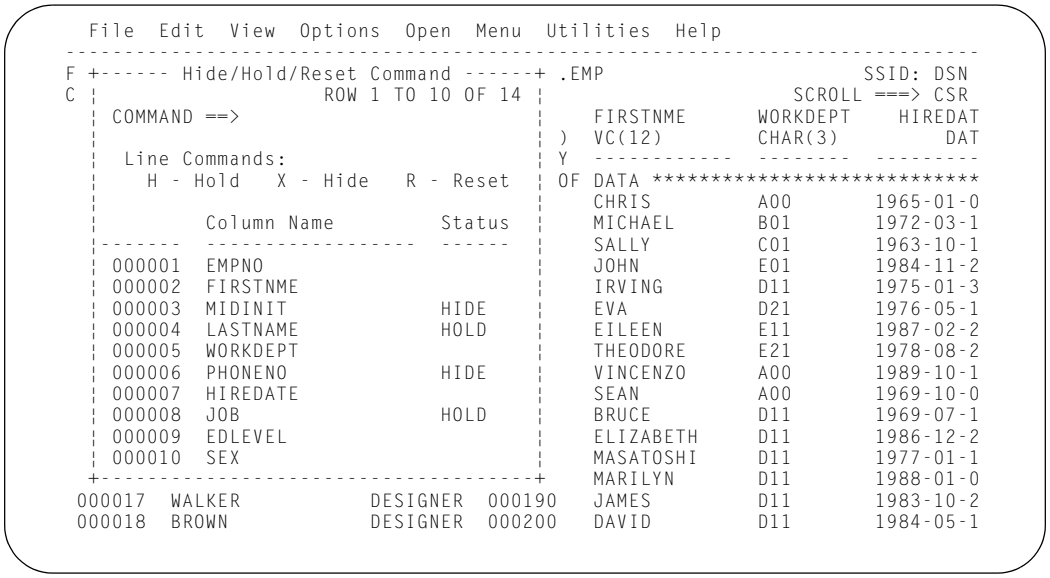
Name of the column to be released. Resets the specified hidden or held column.

RESET Command Window

The RESET command window (Figure 3-19) lists each column and its current status. It displays when you enter RESET with no parameters. One or more column names may be selected.

Press Enter on this screen to process the commands and exit the window. Press END to cancel the RESET command and exit the window.

Figure 3-19. RESET window in Edit



Line Command Area

Specify the line command you want to invoke for a column.

H, HH, Hn

Hold a column or columns. Holds a column to the left side of the screen in table mode or to the top in row mode. See “HOLD” on page 3-27 for more information.

X, XX, Xn

Hide a column or columns. Hides a column from the display. See “HIDE” on page 3-25 for more information.

R, RR, Rn

Reset a column or columns. Restores a hidden or held column.

Column Name

Name of the table column.

Status

Indicates if the column is hidden or held by indicating HIDE or HOLD, respectively.

RETPARM

The RETPARM command can be entered on any of the Function Create screens in the Objects Utility to display the Return Parameter popup where you can enter return parameters for a function. For more information, see "Option 8, User Defined Functions" on page 8-52.

RETRIEVE

The RETRIEVE command redisplays the last command you entered. This is most helpful when you have executed a long command and want to enter the same or a similar command without retyping it. Multiple iterations of RETRIEVE continues to display the command issued previous to the retrieved command.

RETURN

The RETURN command displays the File-AID *for DB2* Primary Option Menu or redisplays the screen that was showing when you pressed HELP.

REVOKE (REV)

The REVOKE command revokes the privileges to a DB2 table, view, database, tablespace, or plan in the Privileges utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it at the bottom of each screen on which it is supported. Enter REVOKE in the COMMAND field of the appropriate screen when you have completed your changes. See Chapter 12, "Privileges Utility" for more information.

RFIND

The RFIND command repeats the execution of the previous FIND command.

RI

The RI command allows you to resolve referential integrity errors. It is only valid in Edit row mode from the RI Browse window.

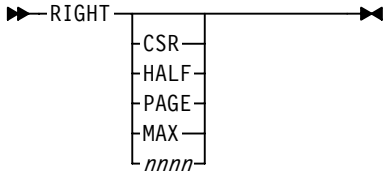
When you receive an RI error in an edit session, you can display the parent table in the RI Browse window. When you issue the RI command, you select the row currently being viewed. The data in the selected row's primary key column replaces the data in error in the dependent table's foreign key column.

See "Referential Integrity Errors" on page 6-27 for more information on using the RI command.

Note: Referential integrity errors are resolved in table mode with the RI line command. See "RI (Referential Integrity)" on page 4-5 for more information.

RIGHT

The RIGHT command scrolls to reveal data to the right of the current display. RIGHT can be issued during a browse or edit session. It is not valid in row mode. Unless you specify a temporary scrolling value with the RIGHT command, the value in the SCROLL field is used.



CSR

Scrolls through the data, making the current position of the cursor the left edge of the data display and showing the data to the right of that point. If the cursor is positioned outside the data area, RIGHT CSR has the same effect as RIGHT PAGE.

HALF

Scrolls through the data half a page at a time. If scrolling a half page of data causes a column to be obscured, File-AID *for DB2* adjusts the scroll amount so that the column is not obscured.

PAGE

Scrolls through the data a page at a time. A page equals the number of character spaces displayed on the screen. If scrolling a page of data causes a column to be obscured, File-AID *for DB2* adjusts the scroll amount so that the column is not obscured.

MAX

Scrolls to the right-most data position of the selected object.

nnnn

Scrolls through the data *nnnn* characters at a time. A valid entry is from 1 to 9999.

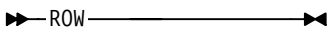
ROW

The ROW command switches you from table to row mode. It is only valid during a browse or edit session in table mode. Executing the command displays the row that was at the top of the screen in table mode. You can display a specific row from the table mode screen by typing the ROW command, positioning the cursor on the selected row, and pressing Enter.

Note: To facilitate switching from table to row mode, assign the ROW command to a PF key.

The Initial Display Mode field on the Edit and Browse options screens controls whether data is initially displayed in row or table mode. See "Display Modes" on page 6-14 for

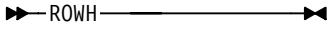
more information. After issuing the ROW command, your Initial Display Mode field defaults to ROW.



ROWH

The ROWH command switches you from table to hexadecimal row mode when in a browse or edit session. Executing the command displays the row that was at the top of the screen in table mode. You can display a specific row from the table mode screen by typing the ROWH command, positioning the cursor on the selected row, and pressing Enter. This command is only valid during a browse or edit session.

The Initial Display Mode field on the Edit and Browse options screens controls whether data is initially displayed in row or table mode. See "Display Modes" on page 6-14 for more information. After issuing the ROWH command, your Initial Display Mode field defaults to ROW.



RUNOPTS

The RUNOPTS command brings up the Run-Time Options window (Figure 8-61 on page 8-82). Used in triggers, stored procedures, and UDFs.

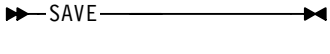
SAVE

The SAVE command saves the changes you have made during the current edit session. You should save your changes periodically while editing.

In Related Table Edit, the SAVE command saves all changes to all tables. For more information on Related Table Browse or Edit, see Chapter 7, "Related Table Browse and Edit".

Note: If the Auto Commit profile option is set to ON, your changes are committed (saved) every time you press Enter.

From the SQL Development and Analysis Edit screen, the SAVE command saves any changes made to the source. It does not, however, commit any DB2 changes.



SORT

The SORT command sorts table rows during a browse or edit session and sorts objects in a list. The SORT command can also be issued on SQL Development and Analysis selection screens. The valid sort parameters for these screens are listed with each screen display. See Chapter 16, "SQL Development and Analysis" for more information.

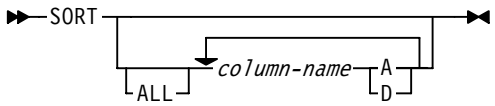
SORT in Browse or Edit

In a browse or edit session, the SORT command sorts the rows of a table by the columns you specify. You must specify a sort order (ascending or descending) for all columns

except the last one entered. The display is repositioned to the beginning row of the sorted results.

Note: Entering SORT with no column name, or entering SORT ALL, displays the window discussed in "SORT Command Window" on page 3-53.

The SORT command defaults to sorting only non-excluded rows. Excluded rows are placed at the end of the sorted table. If the ALL parameter is used, excluded rows become visible and are sorted.



ALL

Specifies the sort is done on all rows (excluded and non-excluded). If specified, ALL must be the first parameter.

column-name

Name of the table column by which you want to sort. Partial column names may be entered. If more than one column name matches the criteria, the window in Figure 3-1 on page 3-8 displays only those matching the criteria. See "Partial Column Names" on page 2-9 for more information.

A

Specifies the sort is to be in ascending order. Default only for the last column name specified.

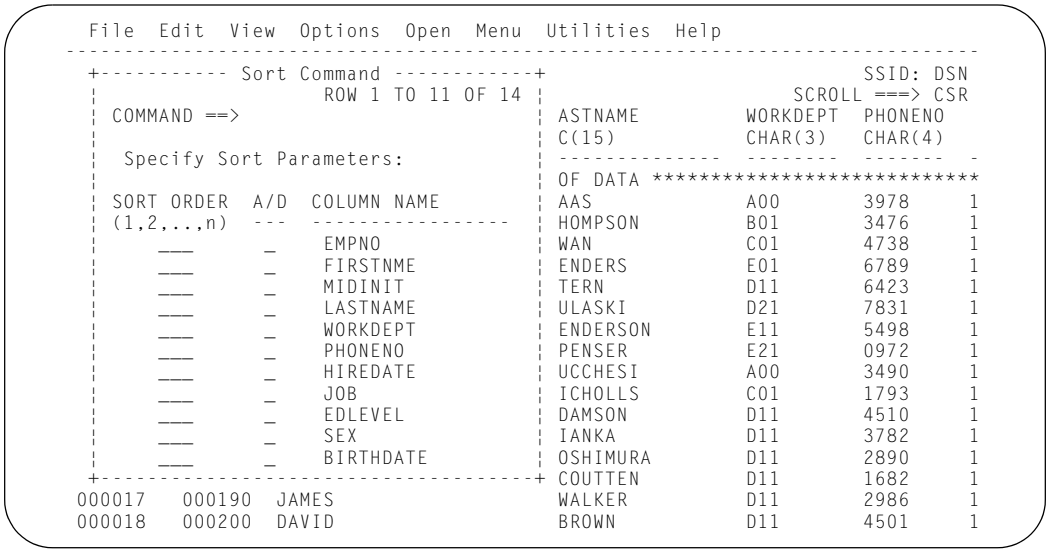
D

Specifies the sort is to be in descending order.

SORT Command Window

The SORT command window (Figure 3-20 on page 3-54) lists all the columns in the table, allowing you to select your sort order. It displays when you enter SORT with no parameters. Press Enter on this screen to process the sort and exit the window. Press END to cancel the SORT command and exit the window.

Figure 3-20. SORT Command Window



Sort order

Specify the sort order of the columns. For example, **1** is the primary sort order, **2** is the secondary, etc.

A/D

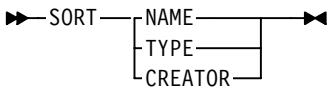
Specify A to sort in ascending order and D to sort in descending order. Ascending is the default.

Column name

Name of the table column.

SORT in an Object List

From a displayed object list, the SORT command sorts by name, type, or creator.



NAME

Sorts by object name.

TYPE

Sorts by object type.

CREATOR

Sorts by creator's user ID.

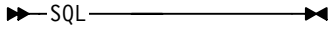
SPLIT

The SPLIT command splits the screen in two at the position at which the cursor is located. This lets you run two sessions at the same time.

SQL

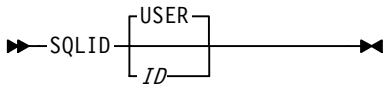
The SQL command stores the SQL generated for the current function in your SHOW SQL dataset as specified in your User Parameters (see "Option 2, SQL Dataset Allocation" on page 5-5). The resulting SQL displays on your screen. You may edit, save, and/or execute the SQL. This command is active only for the following screens:

- Browse or Edit Selection Template
- Table, Tablespace, Index, Synonym, Alias, and View Create
- Table, Tablespace, and Index Alter
- Table, Tablespace, Index, Synonym, Alias, and View Drop
- Modify Alias, Synonym
- Copy Row Selection
- Grant Privileges for Table, Tablespace/Database, View, and Plan
- Revoke Privileges for Table, Tablespace/Database, View, and Plan
- Modify Privileges for Table, Tablespace/Database, View, and Plan
- Online Extract Selection Template



SQLID

The SQLID command is used to view the current SQLID or set it to a new value. See also the Current SQLID field in the "User Parms Menu" on page 5-1.



USER

Default. Set the current SQLID to your user ID.

ID

Any user ID that is an associated secondary AUTHID.

You can also use the main SQLID command by itself on any command line. Press Enter and the SQLID pop-up displays.

Figure 3-21. SQLID Pop-Up

```
 MENU  UTILITIES  HELP
-------------------------------------------------------------------------------
FILE-AID FOR DB2                STORED PROCEDURES
OPTION  ===> SQLID +---------- SQLID ----------+           SSID  ===> D610
                   | COMMAND ===>               |           SQLID ===> EJPYXTO
   C - CREATE  PRO |                            |
   A - ALTER   PRO | SQLID ===> EJPYXTO         |
   D - DROP    PRO |                            |
                   |                            |
ENTER:       - USE +----------------------------+ T FOR D (ROP) AND A (LTER) -

   PROCEDURE NAME ===>
   SCHEMA NAME    ===> EJPYXTO

   USE 'S' TO SELECT ONE APPLICABLE LANGUAGE:  - FOR C (REATE) AND A (LTER) -

   _ ASSEMBLER         _ PLI

   _ C                 _ REXX

   _ COBOL             _ SQL
```

On this pop-up you can view the current SQLID or change it as necessary. If you blank out the value shown, File-AID *for DB2* enters the value USER.

SRCPARM

The SRCPARM command can be entered on any of the Function Create screens in the Objects Utility to display the Source Parameter popup where you can enter source parameters for a function. For more information, see “Option 8, User Defined Functions” on page 8-52.

SWAP

The SWAP command is used in split-screen mode to move the cursor to the last position it occupied in the opposite session. This lets you move back and forth between two screens easily without exiting from either.

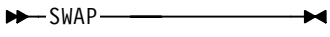
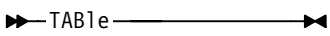


TABLE (TAB)

The TABLE command switches you from row mode to table mode. It is valid only during a browse or edit session in row mode. The row displayed at the top of the screen is the row which was displayed in row mode.

Note: To facilitate switching from row to table mode, assign the TABLE command to a PF key.

The INITIAL DISPLAY MODE field on the Edit and Browse Options screens controls whether data is initially displayed in row or table mode. See “Display Modes” on page 6-14 for more information. After issuing the TABLE command, the INITIAL DISPLAY MODE field defaults to TABLE.

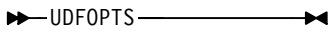


TOP

In table mode, the TOP command scrolls the data on the screen to reveal the first line of data. In row mode, the TOP command scrolls the data on the screen to reveal the first column of data in the row being viewed. A marker **TOP OF DATA** displays to notify you that you have reached the top of a table. This command is equivalent to the UP MAX command. You can issue it on any scrollable screen.

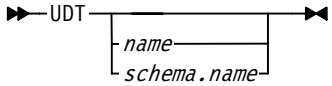
UDFOPTS

The UDFOPTS command can be entered from within the Objects Utility to display the User Defined Function Options screen where you can enter function options. For more information, see "Option 8, User Defined Functions" on page 8-52.



UDT

The UDT command displays information about the source data type for a DISTINCT type. The UDT command can be used to display information about one or more distinct types on any panel that contains a distinct type.



Blank

Only valid for browse or edit. Enter the UDT command with the cursor placed on a distinct type column header or distinct type column. The UDT Definition window displays the built-in data type information for the selected distinct type (see Figure 3-22).

Figure 3-22. Distinct Data Type Definition window

```
File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit   TSOID01.EMP                      SSID: DSN
COMMAND ===> UDT                                                SCROLL ===> CSR
         EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT    PHONENO
         CHAR(6)  VC(12)        CHAR(1)  VC(15)           DEPARTMENT  CHAR(4)
         PRFN---  ------------  -------  ---------------  FOREIGN---  -------
 000001  000010   CHRISTINE     I        HAAS             A00         3978
 000002  000020   MICHAEL       L        THOMPSON         B01         3476
 000003  000030   SALLY         A        KWAN             C01         4738
 000004  000050   JOHN          B        GEYER            E01         6789
 000005  000060   IRVING        F        STERN            D11         6423
 000006  000070   EVA           D        PULASKI          D21         7831
 000007  000090   EILEEN        W        HENDERSON        E11         5498
 000008  000100   THEODOR  +-----Distinct Data Type Definition------+ 0972
 000009  000110   VINCENZ  ¦                                       ¦ 3490
 000010  000120   SEAN     ¦ Schema       :  TSOID02               ¦ 2167
 000011  000130   DOLORES  ¦ Type         :  DEPARTMENT            ¦ 4578
 000012  000140   HEATHER  ¦                                       ¦ 1793
 000013  000150   BRUCE    ¦ Source Type  :  CHAR(3)               ¦ 4510
 000014  000160   ELIZABE  ¦                                       ¦ 3782
 000015  000170   MASATOS  +---------------------------------------+ 2890
 000016  000180   MARILYN       S        SCOUTTEN         D11         1682
 000017  000190   JAMES         H        WALKER           D11         2986
```

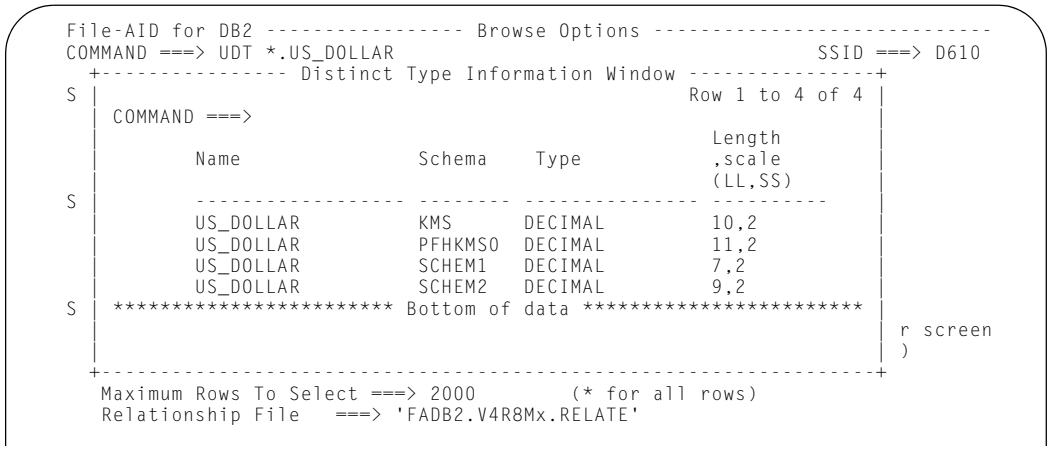
Name

Name of the DISTINCT type to be displayed. Wildcards (*) are valid.

Schema

Name of the schema associated with the DISTINCT type to be displayed. Information for one or more DISTINCT types displays (Figure 3-23).

Figure 3-23. Distinct Type Information pop-up

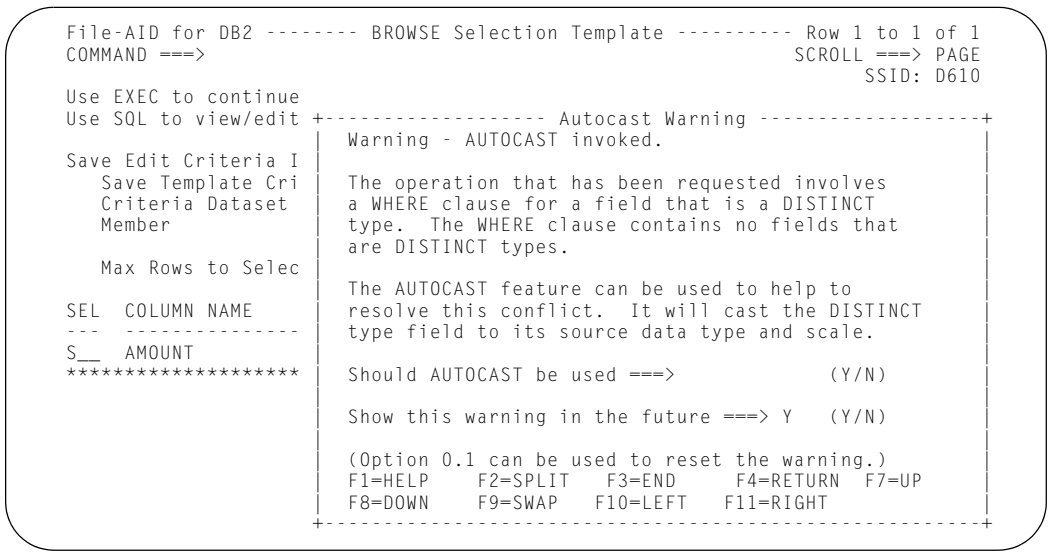


Autocast Feature

Autocast automatically casts a DISTINCT type field to its source data type and scale, and is set on the Browse/Edit Display Options screen (Figure 5-3 on page 5-3). Automatic casting is applied to the field on the left-hand side of a WHERE clause if and only if the WHERE clause contains no fields that are DISTINCT types. This applies in edit, browse, copy, and extract/load.

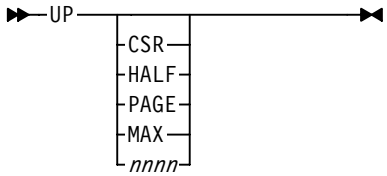
When you select a distinct type row, the Autocast Warning window displays, where you can select whether to apply the cast or not, and whether you want to see the pop-up in the future (save the setting you choose).

Figure 3-24. Autocast Pop-up



UP

The UP command scrolls the data display by the specified scroll amount to reveal the data that precedes the top line currently displayed. Unless you enter a temporary scrolling value with the UP command, the value in the SCROLL field is used. In row mode, UP scrolls to reveal the columns preceding the column currently displayed. To view the previous row, issue the PREV primary command.



CSR

Scrolls through the data, positioning the line on which the cursor currently rests at the bottom of the data display. If the cursor is positioned outside of the data area, UP CSR has the same effect as UP PAGE.

HALF

Scrolls through the data half a page at a time. A page equals the number of lines of data displayed on the screen.

PAGE

Scrolls through the data one page at a time. A page equals the number of lines of data displayed on the screen.

MAX

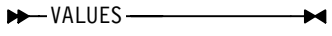
Scrolls to the first line of data in the selected object.

nnnn

Scrolls through the data *nnnn* lines at a time. A valid entry is from 1 to 9999.

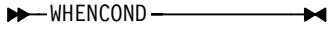
VALUES

The VALUES command lets you tailor values to be used for an index space creation when in the Objects utility. File-AID *for DB2* dynamically assigns this command to a PF key and displays it in the pull-down menu of each screen on which it is supported. Enter VALUES in the COMMAND field of the appropriate screen when you have completed your changes. See Chapter 8, “Objects Utility”, for more information.



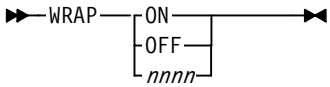
WHENCOND

Used in Trigger Create to define a WHEN condition clause for the CREATE TRIGGER statement. If the WHEN condition clause is true, then the trigger activates.



WRAP

The WRAP command controls the display of a column of data when in a browse or edit session after issuing the ZOOM or ZOOMH command.



ON

Data displays across the screen and wrapped to the next line.

OFF

Data displays on a single line. Scroll RIGHT to view additional characters.

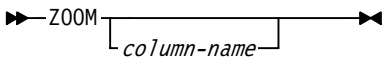
nnnn

Data displays across the screen *nnnn* number of characters for each line of wrapped data.

ZOOM

The ZOOM command examines data from a specific column in character format. You may issue ZOOM with a whole or partial column name, or you may enter ZOOM in the COMMAND field, position the cursor to the desired column and press Enter.

The ZOOM command can be issued during a browse or edit session in table mode. It is not valid in row mode.



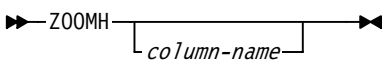
column-name

Name of the column to be examined. You can enter partial column names. If more than one column name matches the criteria, a window displays the matching columns, allowing you to select one. See Figure 3-1 on page 3-8 for a sample partial column window. See “Partial Column Names” on page 2-9 for more information.

ZOOMH

The ZOOMH command examines data from a specific column in hexadecimal format. You may issue ZOOMH with a whole or partial column name, or you may enter ZOOMH in the COMMAND field, reposition the cursor to the desired column, and press Enter.

The ZOOMH command can be issued during a browse or edit session in table mode. It is not valid in row mode.



column-name

Name of the column to be examined. You can enter partial column names. If more than one column name matches the criteria, a window displays the matching columns, allowing you to select one. Figure 3-1 on page 3-8 shows a sample partial column window. See “Partial Column Names” on page 2-9 for more information.

Chapter 4.
Line Commands

This chapter describes the line commands available for use in File-AID *for DB2*. Line commands are commands you use to work with a line or block of data.

Not all line commands are valid in all areas of File-AID *for DB2*. Table 4-1 shows where they are valid.

Note: Line commands are not available in row mode.

Line commands are entered by typing over the characters in the sequence number field.

Table 4-1. Line Command Availability by Function

Line Command	Edit table mode	Browse table mode	ZOOM and ZOOMH mode	Create and Alter Table	SQL Source Analysis	DB2 Privileges
A	x			x	x	x
B	x			x	x	x
BR					x	
C, CC	x			x	x	x
D, Dn, DD	x			x	x	x
ED					x	
EX					x	
I, In	x			x	x	x
M, MM				x	x	x
R, Rn, RR	x			x	x	x
RI [1]	x					
S, Sn	x	x	x		x	
S (select)				x		
X, Xn, XX	x	x	x		x	
XP					x	
[1] **Valid only from the RI Browse window.**						

The execution of a line command can encompass:

- Single lines.
- Range of lines. Enter the line command immediately followed by a number. The number specifies either the number of lines to be affected, or the number of times the command is to be performed.
- Block of lines. Enter the block syntax of the line command on the first and on the last lines of a block of lines.

Line commands can be revised by typing over the command with the desired entry. They can be erased by:

- Entering the RESET ALL primary command.

- Ending the browse or edit session.
- Typing over the sequence number field with blanks.

A line command can be entered in any character position (1-6) in the sequence field. It is not necessary to begin in column 1. However, if one or more numbers entered as part of the line command match one or more of the system-generated numeric sequence number characters, the characters which match are not recognized by the system. This situation can be corrected in one of three ways:

- Position the cursor on the sequence line immediately after the number, and press Enter.
- Enter the line command and its number on a line where the command number and the sequence number characters do not match.
- Entering a blank after the command.

Line commands can be used on all data lines. The I (insert) and A (after) line commands can be entered on the TOP OF DATA line by typing over the asterisks in the first six columns. The B (before) line command can be used on the BOTTOM OF DATA line.

I (Insert), C (Copy), and R (Repeat) create new rows. Copied and repeated rows are placed in a pending state (see “Pending State” on page 6-2 for more details) so that you may change primary keys if necessary. Before saving a pending row, all columns defined as NOT NULL must be entered.

Key-based updating is not supported for remote objects. If you attempt it, a negative SQL code occurs.

Description of Line Commands

This section describes each line command in more detail.

A (After)

Inserts data on the line *after* the line on which the command is entered. It specifies a destination for copied or moved lines, or rows copied with the PASTE command.

B (Before)

Inserts data on the line *before* the line on which the command is entered. It specifies a destination for copied or moved lines, or rows copied with the PASTE command.

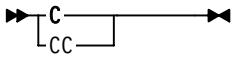
BR (Browse)

Browses the DB2 results table based on the SQL statement for which the line command was entered. This command can be used only within the SQL Source Development and Analysis function.

After entering the BR line command, enter the FADB2 primary command in the COMMAND field and press Enter, or use the associated PF key to execute the command.

C (Copy)

Copies one or more lines of data. Use the A (After) or B (Before) line command to specify a destination.



C

Copies the line on which the command is entered.

CC

Copies the block of lines delimited by the two lines on which the command is entered, inclusively. Copies the block of lines delimited by the two lines on which the command is entered, inclusively.

C or CC as Line Markers for CUT and CHANGE Primary Commands

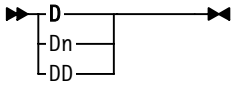
C marks the single line or CC the block of lines (CC) to which the primary command CUT or CHANGE will be applied.

With the CUT command, the marked lines are copied (appended) into the paste buffer (see "CUT (CU)" on page 3-16).

With the CHANGE command, the search of the CHANGE is limited to the marked lines and CHANGE ALL within the range of lines is the default. CHANGE must be entered as a single command without any parameters to invoke the change command window (see "CHANGE Command Window" on page 3-9).

D (Delete)

Deletes one or more lines of data.



D

Deletes the line on which the command is entered.

Dn

Deletes the block of lines delimited by the line on which the command is entered, inclusively. The *n* represents the number of lines to be deleted. For example, **D10** deletes ten lines. A valid entry is from 1 to 99999.

DD

Deletes the block of lines delimited by the two lines on which the command is entered, inclusively.

ED (Edit)

Edits the DB2 results table based on the SQL statement for which the command was entered. The command can be used only within the SQL Source Development and Analysis function.

After entering the ED line command, enter the FADB2 primary command in the COMMAND field and press Enter, or use the associated PF key to issue the line command.

EX (Execute)

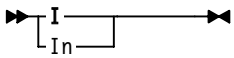
Dynamically executes the SQL statement for which it was entered. The command can be used only within the SQL Source Development and Analysis function.

After entering the EX line command, enter the FADB2 primary command in the COMMAND field to issue the line command.

I (Insert)

Inserts one or more blank lines of data. On the Edit table mode screen, the I line command creates a new row with null values initialized.

Note: For character columns with defined default values the Profile option SPACES IMPLY NULL must be ON so File-AID *for DB2* fills in the default values for blank columns.



I

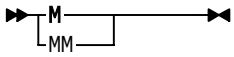
Inserts a blank line after the line on which the command is entered.

In

Inserts a block of blank lines after the line on which the command is entered. The *n* represents the number of lines to be inserted. For example, **I10** inserts ten blank lines. A valid entry is from 1 to 99999.

M (Move)

Moves one or more lines of data. Use the A (After) or B (Before) line command to specify a destination. Not available in Edit and Browse table or Zoom and Zoomh modes.



M

Moves the line on which the command is entered.

MM

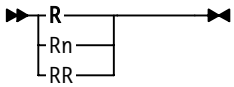
Moves the block of lines delimited by the two lines on which the command is entered, inclusively.

R (Repeat)

Repeats one or more lines of data. The new line immediately follows the original.

When your user profile has the Repeat Command Prompt set to ON (see "PROFILE (PRO)" on page 3-39) the REPEAT Column Range Specification window appears. Use the REPEAT

window to generate unique values in one of the columns. See "REPEAT (REP)" on page 3-43 for more information.



R

Repeats the line on which the command is entered.

Rn

Repeats a line n times. The *n* represents the number of times a line is to be repeated. For example, **R10** repeats a line ten times. A valid entry is from 1 to 99999. The new block of lines immediately follows the original.

RR

Repeats the block of lines delimited by the two on which the command is entered, inclusively. The new block immediately follows the original.

RI (Referential Integrity)

Resolves referential integrity errors. The RI line command can only be issued in the RI Browse window. When you receive an RI error in an edit session, you can display the parent table in the RI Browse window. Enter the RI command in the line command area of the row containing the desired data. The data in the selected row's primary key column replaces the data in error in the dependent table's foreign key column.

See "Referential Integrity Errors" on page 6-27 for more information.

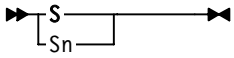
Note: Referential integrity errors are resolved in row mode with the RI primary command. See "RI" on page 3-50 for more information.

S (Select)

When used on a parameter line that contains either a built-in datatype, distinct type, or table, brings up the Built-In Datatype pop-up (Figure 8-60 on page 8-81), the Distinct Type Information window (Figure 10-10 on page 10-25, or the Table Information Display (Figure 10-3 on page 10-7), respectively.

S (Show)

Redisplays one or more excluded lines of data.



S

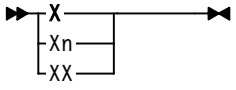
Redisplays the first line in a set of excluded lines.

Sn

Redisplays a block of excluded lines starting with the first line in the set of excluded lines. The *n* represents the number of lines to be redisplayed. For example, **S10** redisplays up to ten excluded lines. A valid entry is from 1 to 99999.

X (Exclude)

Excludes one or more lines of data from the display. The excluded lines are replaced on the display by information lines. Use the S line command to redisplay the line.



X

Excludes from the display the line on which the command is entered.

Xn

Excludes a block of lines from the display starting with line the command is entered on. The *n* represents the number of lines to be excluded. For example, **X10** excludes the next ten lines. A valid entry is from 1 to 99999.

XX

Excludes from the display the block of lines delimited by the two lines on which the command is entered, inclusively. The excluded lines are replaced on the display by an information line.

XP (Explain)

Displays an Explain report. The XP line command executes a DB2 EXPLAIN without prompting for host variable values. The command can be used only within the SQL Source Development and Analysis function. The command may be entered on SQL statements that contain the DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE verbs.

Note: In the case of the DECLARE statement, the XP command is performed on the SELECT portion of the statement. A WHERE CURRENT OF clause in an UPDATE statement is not explained. Therefore, it is more meaningful to issue the XP command on the corresponding DECLARE statement.

After entering the XP line command, enter the FADB2 primary command in the COMMAND field and press Enter, or use the associated PF key to issue the line command.

Chapter 5.
Defaults and User Parameters

This chapter describes the File-AID *for DB2* user-definable parameters. These parameters include such things as display options, default values, and function key definitions. The following screens are described:

- User Parameters
- Browse/Edit/SQL Analysis Display Options
- Dataset Allocation
- Allocate Show SQL Dataset
- Batch Options
- Index, Tablespace and Database Default Values
- Logging Options
- Print Options
- Current Path Options
- Trigger Terminator

Select option **0**, Defaults and User Parms from the Primary Option Menu. The User Parms menu displays (Figure 5-1). From this menu, you can select the set of parameters you want to modify.

Figure 5-1. User Parms Menu

```
FILE-AID FOR DB2 ---------- USER PARMS -----------------------------
OPTION  ===>

DB2 SUBSYSTEM: D610         ===> D610           DB2 VERSION: 6.1.2
CURRENT SQLID: USERID1      ===> USERID1      FADB2 VERSION: 4.8.1.00
                                               SQLP VERSION: 04.07.00
                                                 FM VERSION: R2.1
                 1  BROWSE/EDIT/SQL ANALYSIS DISPLAY OPTIONS

                 2  SQL DATASET ALLOCATION

                 3  BATCH OPTIONS

                 4  NOT USED

                 5  INDEX, TABLESPACE, AND DATABASE DEFAULT VALUES

                 6  LOGGING OPTIONS

                 7  PRINT OPTIONS

                 8  CURRENT PATH OPTIONS

                 9  TRIGGER TERMINATOR
```

DB2 Subsystem

The first value next to the field name is the current DB2 subsystem and is display only. The subsystem is modified by typing over the value to the right of the arrow and pressing Enter. If the DB2 subsystem is modified, the File-AID *for DB2* plan name is also changed to the subsystem's default plan name defined when the product is installed.

DB2 Version

Version of DB2 you are using. Display only.

FADB2 Version

Version of File-AID *for DB2* you are using. Display only.

Current SQLID

The first value next to the field name is the current SQLID. You can change the SQLID by typing over the value to the right of the arrow, and pressing Enter.

SQLP Version

Version of the SQL processor you are using. Display only.

FM Version

Version of the Function Manager you are using. Display only.

Option 1, Browse/Edit/SQL Analysis Display Options

The Browse/Edit/SQL Analysis Display Options screens are displayed when you select option **1** from the User Parms menu. They let you specify the Browse, Edit, and SQL Analysis display options.

Figure 5-2. Browse/Edit/SQL Analysis Display Options screen (Part 1)

```
File-AID for DB2 ---- Browse/Edit/SQL Analysis Displa           Required field
COMMAND ===>

  Primary, Foreign and Unique Key Options
    Allow update of primary key in edit            ===> OFF     (ON or OFF)
    Indicate foreign key columns while in edit     ===> OFF     (ON or OFF)
    Retrieve UNIQUE KEY information prior to edit  ===> OFF     (ON or OFF)
  Commit Options in EDIT
    AUTOCOMMIT commits changes upon pressing enter ===> ON      (ON or OFF)
    AUTOSAVE commits changes upon exiting session  ===> ON      (ON or OFF)
  Lowercase Support for Object Names Browse/Edit
    Object names are converted to uppercase        ===> ON      (ON or OFF)
  Lowercase Support for Data in Browse/Edit/SQL Analysis
    Data is converted to uppercase                 ===> OFF     (ON or OFF)
  Row Selection Option
    Maximum Rows to Select                         ===> 2000    (* = all rows)
  Identity column Option
    Allow update of Identity Columns in edit       ===>        (ON or  OFF)
---------------------- press DOWN for more display options ----------------------
```

Primary, Foreign and Unique Key Options Area

Allow update of primary key in edit

Specify ON to update primary keys during an edit session. This option can also be set with the PROFILE primary command.

Indicate foreign key columns while in edit

Specify ON to highlight foreign keys during an edit session. This option only applies to local tables.Because DB2 does not maintain an index on the table containing foreign key information, setting this option to ON may have a negative impact on performance. This option can also be set with the PROFILE primary command.

Retrieve UNIQUE KEY information prior to edit

Specify ON to retrieve unique key information before the edit session begins. This applies only if there is no primary key. This option only applies to local tables.

Note: In order to update primary key columns in a ***view***, set RETRIEVE UNIQUE KEY INFO to ON before your view editing session. This setting also determines if the primary key, foreign key, and identity column labels appear for views that are being edited or browsed.

Commit Options in Edit Area

AUTOCOMMIT commits changes upon pressing enter

Specify ON to commit changes each time you press Enter during an edit session. This option can also be set with the PROFILE primary command.

AUTOSAVE commits changes upon exiting session

Specify ON to save changes each time you press END during an edit session. This option can also be set with the PROFILE primary command during a browse or edit session.

Lowercase Support for Object Names Browse/Edit

Object names are converted to uppercase

Specify ON to automatically convert all object names entered in lowercase characters to uppercase characters. Specify OFF if you need to edit a table with a DBCS table name or DBCS column name. This option can also be set with the PROFILE primary command.

Lowercase Support for Data in Browse/Edit/SQL Analysis Area

Data is converted to uppercase

Specify ON to automatically convert all data entered in lowercase characters to uppercase characters.

Row Selection Option Area

Maximum Rows to Select

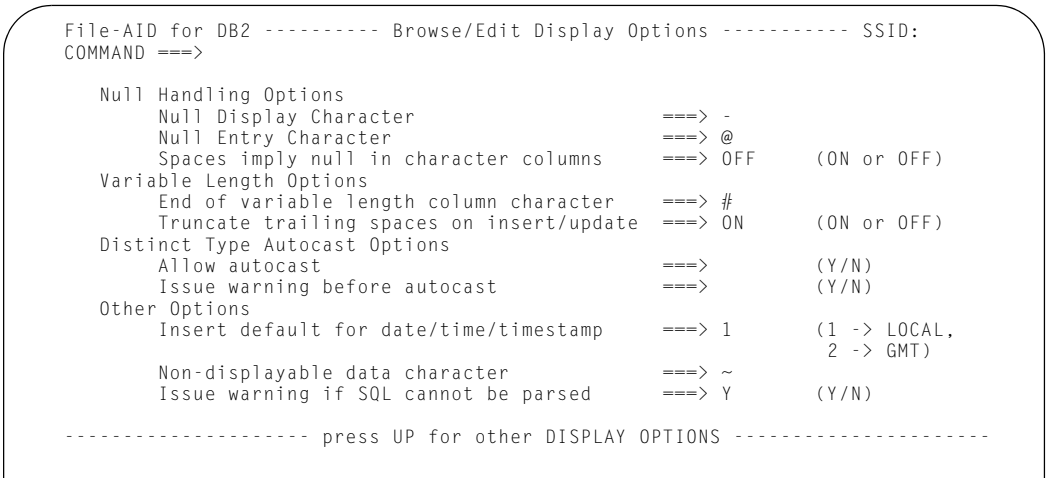
Specify a numeric value or an asterisk (*****) to denote the maximum number of rows to be selected for an edit/browse session or a table extract. The value may be changed for individual edit or browse sessions, (“Maximum Rows to Select” on page 6-3) or table extracts (“Maximum Rows to Select” on page 14-4).

Identity Column Options Area

Allow update of Identity Columns in edit

Specify ON to allow update of identity columns in edit. This option can also be set with the PROFILE primary command. This applies only to columns set to GENERATED BY DEFAULT. Columns set to GENERATED ALWAYS cannot be edited. If the column is both an identity column and a primary key, then ALLOW UPDATE OF PRIMARY KEY IN EDIT must also be ON to edit the column.

Figure 5-3. Browse/Edit/SQL Analysis Display Options screen (Part 2)



Null Handling Options Area

Null Display Character

Specify the character that displays for null columns. This option can also be set with the PROFILE primary command.

Null Entry Character

Specify the character that, when entered, sets a column to null. You must enter the null entry character in the first position of the column for it be converted to nulls. If the character is not entered in the first position, it is accepted as the actual character and not converted to nulls. If the column is defined as NOT NULL, the null entry character is also accepted as the actual character and not converted. This option can also be set with the PROFILE primary command.

Spaces imply null in character columns

Specify **ON** to have spaces imply nulls for character columns. This can also be set with the PROFILE primary command.

Note: For character columns with defined default values the Profile option SPACES IMPLY NULL must be ON so File-AID *for DB2* fills in the default values for blank columns.

Variable Length Options Area

End of variable length column character

Specify the character used to signal the end of a column. This option can also be set with the PROFILE primary command.

Truncate trailing spaces on insert/update

Specify **ON** to truncate trailing spaces on an insert or update. This option can also be set with the PROFILE primary command.

Distinct Type Autocast Options Area

Allow Autocast

Indicates whether File-AID *for DB2* should automatically cast a distinct type field to its source data type and scale. Specify Y to apply automatic casting to the field on the left-hand side of a WHERE clause if and only if the WHERE clause contains no fields that are DISTINCT types. Specify N if you do not want automatic casting to occur. This applies to edit, browse, copy, and extract/load.

Issue warning before Autocast

Specify **Y** to display a warning in a window before applying the cast. The window asks whether the cast should be applied and whether the warning should appear in the future.

Other Options Area

Insert default for date/time/timestamp

Specify 1 for local or 2 for GMT (Greenwich Mean Time) to determine the date/time format used as the default during inserts and when CURRENT is used to enter a date/time field. This option can also be set with the PROFILE primary command.

Note: To support GMT, you must provide this information at IPL.

Non-displayable data character

Specify the character used to signify nondisplayable data when in browse mode. Does not apply for edit mode (“NON-DISPLAY” on page 3-40). This option can also be set with the PROFILE primary command.

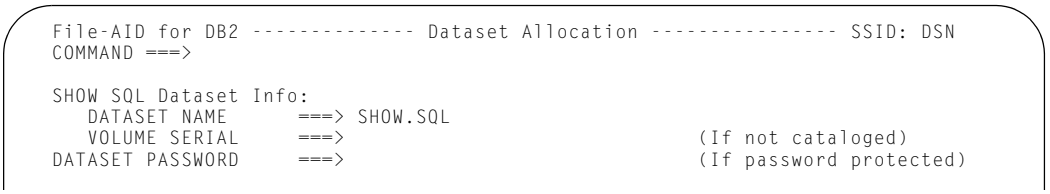
Issue warning if SQL cannot be parsed

Specify **Y** to display a warning in a window if SQL for existing criteria cannot be parsed into the Template Selection screen (Figure 6-5 on page 6-10). If **N** is specified, then File-AID *for DB2* will proceed with the action last specified in the warning window, either **Execute SQL** or **Edit SQL**.

Option 2, SQL Dataset Allocation

The Dataset Allocation screen displays when you select option **2** from the User Parms menu. It enables you to enter name and volume information for the SHOW SQL dataset.

Figure 5-4. Dataset Allocation screen



Dataset Name

Name of the dataset that is to contain the SHOW SQL information. If the dataset is a PDS and you omit the member name, File-AID *for DB2* uses member SHOWSQL.

Volume Serial

Optional. Disk pack on which the dataset is to reside if it is not cataloged.

Dataset Password

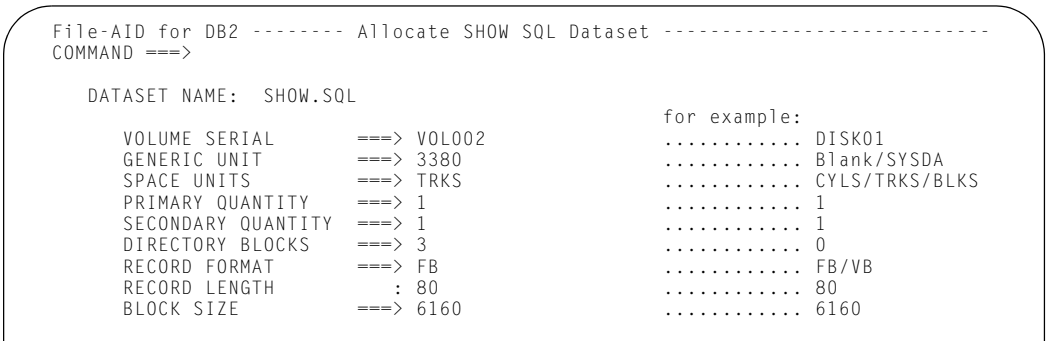
Optional. Password for the dataset if an existing password-protected dataset is identified.

Note: This password must be defined through ISPF.

Allocate SHOW SQL Dataset

The Allocate SHOW SQL Dataset screen displays when you type a new dataset name on the Dataset Allocation screen and press Enter. The screen provides a means of specifying allocation attributes for the SHOW SQL dataset.

Figure 5-5. Allocate Show SQL Dataset screen



Dataset Name

Name of the SHOW SQL dataset you entered on the Dataset Allocation screen.

Volume Serial

Optional. Disk pack on which the dataset is to reside.

Generic Unit

Optional. Installation-defined device type (SYSDA, for example).

Space Units

Type of space units defined for the dataset (blocks, for example).

Primary Quantity

Number of primary space units to be allocated for the dataset.

Secondary Quantity

Incremented amount of space to be allocated for the dataset if it exceeds the primary quantity.

Directory Blocks

Number of directory blocks to be allocated for the dataset (if it is partitioned). Zero (0) implies a sequential dataset.

Record Format

Record format of the dataset.

Record Length

The maximum length for each record must be 80 and cannot be changed.

Block Size

Size of each block of data in the dataset.

Option 3, Batch Options

The Batch Options screen displays when you select option **3** from the User Parms menu. It lets you specify job card information for a batch job.

Figure 5-6. Batch Options screen

```
File-AID for DB2 ---------------- Batch Options ------------------- SSID: DSN
COMMAND ===>

Job Statement Information:
  ===> //*
  ===> //*
  ===> //*
  ===> //*
```

Job Statement Information

Area in which you enter your job card information.

Option 5, Index, Tablespace, and Database Default Values

The Object Creation Default Values screen displays when you select option **5** from the User Parms menu. It lets you specify the default tablespace and database and is also used to set default values for tablespace and index creation.

Figure 5-7. Object Creation Default Values screen

```
File-AID for DB2 ---------- Object Creation Default Values -------- SSID: DSN
COMMAND ===>

DATABASE Name   ===> DBNAME
TABLESPACE Name ===> TSNAME

------------------------- TABLESPACE Creation Default Values ------------------
  STOGROUP ===> STOGP          FREEPAGE ===>            BUFFERPOOL ===> BP0
 PRIQTY(K) ===>                PCTFREE  ===> 25         LOCKSIZE   ===> ANY
 SECQTY(K) ===>                                        CLOSE      ===> YES
    ERASE  ===> NO                                     DSETPASS   ===>
  TRACKMOD ===>                                        SEGSIZE    ===>
    -- OR --
 VCAT      ===>

--------------------------- INDEX Creation Default Values ----------------------
  STOGROUP ===> STOGP          FREEPAGE ===>            SUBPAGES   ===> 4
 PRIQTY(K) ===>                PCTFREE  ===> 25         BUFFERPOOL ===> BP0
 SECQTY(K) ===>                COPY     ===>            CLOSE      ===> YES
    ERASE  ===> NO                                     DSETPASS   ===>
    -- OR --
 VCAT      ===>
```

DATABASE Name

Name of the database used as a default.

TABLESPACE Name

Name of the tablespace used as a default.

Tablespace Creation Default Values Area

STOGROUP

Name of the set of volumes where the DB2 data for the tablespace is stored. This field and the VCAT field are mutually exclusive.

PRIQTY

Amount of space (in kilobytes) for the dataset where your tablespace or partition resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

ERASE

Specify whether the dataset is erased if it is dropped.

TRACKMOD

Indicate whether DB2 tracks modified pages in the space map pages of the tablespace or partition.

VCAT

VSAM catalog that contains the datasets for the tablespace. This field and the STOGROUP field are mutually exclusive.

FREEPAGE

Rate at which a page is left free when the tablespace or partition is loaded or reorganized.

PCTFREE

Percentage of each page left free when the tablespace or partition is loaded or reorganized.

BUFFERPOOL

Name of the buffer pool associated with the tablespace.

LOCKSIZE

Required locking level for the tablespace. Valid entries are ANY, PAGE, TABLESPACE, and TABLE.

CLOSE

Indicate whether the datasets supporting the tablespace are closed when no users are using the tablespace.

DSETPASS

Password passed to VSAM (virtual storage access method) when DB2 uses the datasets that support the tablespace.

SEGSIZE

Specify whether the tablespace is segmented, and if so, how many pages are assigned to each segment.

Index Creation Default Values Area

STOGROUP

Name of the set of volumes where the DB2 data for an index is stored. This field and the VCAT field are mutually exclusive.

PRIQTY

Amount of space (in kilobytes) allocated for the dataset where your index resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicate whether the dataset is erased if it is dropped.

VCAT

VSAM catalog containing the datasets for the index. This field and the STOGROUP field are mutually exclusive.

FREEPAGE

Rate at which a page is left free when the index is created or reorganized.

PCTFREE

Percentage of each page left free when the index is created or reorganized.

COPY

Indicate whether Image copy of the index is allowed.

SUBPAGES

Number of subpages each page has.

BUFFERPOOL

Name of the buffer pool associated with the index.

CLOSE

Indicate whether the datasets supporting the index are closed when no users are using it.

DSETPASS

Password passed to VSAM (virtual storage access method) when DB2 uses the datasets that support the index.

Option 6, Logging Options

The Logging Options screen displays when you select option **6** from the User Parms menu. It lets you set options for trace information which can be used for diagnostic purposes.

Figure 5-8. Logging Options screen

```
File-AID for DB2 ----------- Logging Options ---------------------  SSID: D701
COMMAND ===>

Logging options for Trace information and generated SQL

    Write data to            ===> 4       1 = Write to ISPF Log,
                                          2 = Write to screen,
                                          3 = Write to Log and Screen,
                                          4 = No logging

    Log SQL information      ===> NO      YES/NO

    Note:  Additional logging may impact performance

    Log ISPF calls           ===> NO      YES/NO

    Log transfer of control  ===> NO      YES/NO
```

Write data to

Specify one of four logging options listed on the screen:

1 Writes to the ISPF log.

2 Writes to the screen.

3 Writes to the ISPF log and the screen.

4 No logging.

Log SQL information

Default is YES. Enter NO to turn off logging of SQL information.

Log ISPF calls

Default is NO. Enter YES to turn on logging of ISPF calls.

Log transfer of control

Default is NO. Enter YES to turn on logging of transfer of control.

Option 7, Print Options

The Print Options screen displays when you select option 7 from the User Parms menu. It lets you set the printer and output dataset characteristics used when you issue the PRINTP or PRINTDS command from Browse/Edit, or when generating SQL Analysis, Extract Formatted Print (printer only), and Audit Trail reports. Depending on the choice and form of the command, your output is printed on a printer or to a dataset with some or all of the currently displayed rows.

Figure 5-9. Print Options screen

```
File-AID for DB2 -------------- Print Options --------------------  SSID: DSN
COMMAND ===>

Printer Characteristics:
   Printer Destination  ===> PRT1
   Lines per Page       ===> 55              ......... 55
   Sysout Class         ===> A               ......... A

Output Dataset Characteristics:
   Dataset Name         ===> FD.PRINT
   Volume Serial        ===> DISK01          ......... DISK01
   Generic Unit         ===> SYSDA           ......... SYSDA
   Directory Block      ===> 0               ......... 0
   Record Format        ===> FB              ......... FB or FBA
   Record Length        ===> 133             ......... 4092
   Block Size           ===> 1330            ......... 4096
   Dataset Password     ===>                     (If password protected)
```

Printer Characteristics Area

Printer Destination

Local printer or JES queue.

Lines per Page

Number of lines printed on each page.

Sysout Class

Assigned class for SYSOUT.

Output Dataset Characteristics Area

Dataset Name

Destination dataset for output.

Volume Serial

Disk pack where the dataset will reside if it is not catalogued. Required if GENERIC UNIT is not specified.

Generic Unit

Installation-defined device type (SYSDA, for example). Required if VOLUME SERIAL is not specified.

Directory Block

Number of directory blocks to be allocated for the dataset. Must be zero (0).

Record Format

Record format for the output dataset.

Record Length

Maximum length for each record.

Block Size

Size of each block of data in the dataset.

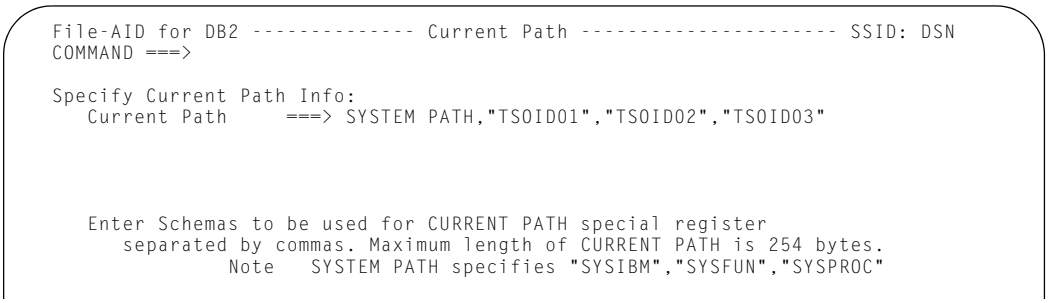
Dataset Password

Password for the dataset if it is password protected.

Option 8, Current Path Options

The Current Path screen displays when you select option **8** from the User Parms menu. It lets you specify a list of schema (creator) names to be added to the CURRENT PATH Special Register. CURRENT PATH is used in the resolution of unqualified references to user-defined functions, distinct types, and stored procedures.

Figure 5-10. Current Path screen



Current Path

Specify a list of schema (creator) names to be added to the CURRENT PATH Special Register.

The initial value of the CURRENT PATH special register is one of the following:

- The last saved CURRENT PATH values in your user profile.
- The value of the PATH bind option.
- "SYSIBM", "SYSFUN", "SYSPROC", "CURRENT SQLID" if the PATH bind option was not specified or if you blank out this field.

You may also enter **SYSTEM PATH** which translates into "SYSIBM", "SYSFUN", "SYSPROC".

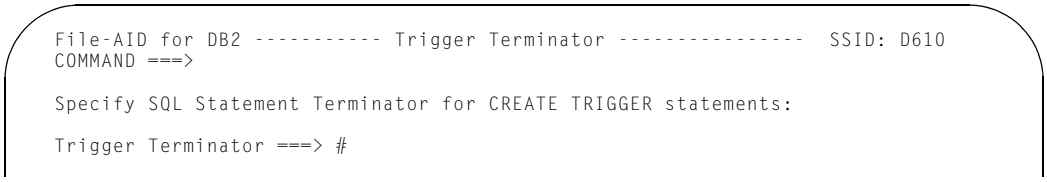
The total length of the string of all schema names, including delimiters, in the CURRENT PATH register cannot exceed 254 characters. Remember also that SYSTEM PATH expands into "SYSIBM", "SYSFUN", "SYSPROC" for the SET CURRENT PATH statement.

Option 9, Trigger Terminator

The Trigger Terminator screen displays when you select option **9** from the User Parms menu. It lets you specify an SQL statement terminator different than the default value (a semicolon). The ability to change the SQL statement terminator is necessary because the CREATE TRIGGER statement may contain multiple SQL statements within the trigger body, each of which must end with a semicolon.

File-AID *for DB2* will override the default DB2 SQL statement terminator for the CREATE TRIGGER statement only. It is not used outside the context of the trigger.

Figure 5-11. Trigger Terminator screen



The default trigger terminator for File-AID *for DB2* is the pound sign (#).

To specify no trigger terminator, enter **NONE**. This is not usually advised, as it can cause problems with a DDL that has more than one BEGIN ATOMIC SQL statement or more than one CREATE TRIGGER statement.

Certain characters are invalid for use as a trigger terminator. These include:

	Blank
,	Comma
''	Double quote
(	Left parenthesis
)	Right parenthesis
'	Single quote
_	Underscore
%	Percent
&	Ampersand
+	Plus sign
-	Minus sign
*	Asterisk
/	Back slash
=	Equal sign
<	Less than sign
>	Greater than sign
|	Or bar
a...z A...Z	Alphabet, lower case or capitals

Restricting these characters minimizes the possibility of using a character that may exist within the SQL statement as a trigger terminator.

Chapter 6.
Browse and Edit

This chapter describes the File-AID *for DB2* Browse and Edit functions. The **Browse** function lets you display a table or view, either in its entirety or by selected rows and fields. You can specify the rows you want to see or indicate that you want certain areas excluded. You cannot, however, make any changes to the table or view.

The **Edit** function provides all of the capabilities of the Browse function, and also allows you to update the data. Because the Browse and Edit screens are so similar, this chapter shows only the Edit version of the screens.

These functions also enable you to browse and edit multiple tables in a single session. The **Related Table Browse and Edit** function enables you to browse and edit multiple tables related through DB2 referential integrity or through a user-defined application relationship. This function is described in detail in Chapter 7, "Related Table Browse and Edit". You can also browse and edit multiple tables that are unrelated. Information on working with unrelated tables is presented in this chapter.

The following topics and screens are described in this chapter:

- Editing Considerations
- Starting a Browse or Edit Session
- Edit Options
- Display Modes
- Edit End Confirmation
- Concurrent Update Intercept
- Check Constraint Error
- Editing Multiple Unrelated Tables
- Referential Integrity Errors
- Creating an Audit Trail Dataset

Complete descriptions of the primary commands available for browse and edit are described in Chapter 3, "Primary Commands". Table 3-1 on page 3-1 summarizes where the primary commands are valid. Line commands are described in Chapter 4, "Line Commands". Table 4-1 on page 4-1 summarizes where the line commands are valid.

Editing Considerations

There are several things to consider when you use Edit. The following topics include situations specific to DB2 and some general rules for keeping your data in order.

- Obscured Columns
- Column Snap when Scrolling
- Pending State
- Max Rows to Select
- Locking a Table During Edit
- Method of Updating DB2
- Editing Primary Column in a View
- Columns with Defined Default Values

Obscured Columns

Obscured columns are DB2 columns that are only partly visible on the screen. They are displayed highlighted and cannot be edited. This feature prevents you from changing a column for which you see only a portion of the available data. For example, you may see a 1 that is actually part of the number 122. File-AID *for DB2* displays the following formatted data types as obscured columns:

- Smallint (small integer)
- Integer
- Decimal
- Float
- Time
- Date
- Timestamp

To edit an obscured column, the entire column must be visible on the screen. To view an entire column, scroll left or right, or issue the ROW or ZOOM command. A column snap feature when scrolling reduces the frequency of obscured columns. See "Column Snap when Scrolling" on page 6-2 for details.

File-AID *for DB2* does not display the following data types as obscured columns. The data in these columns can be edited when the columns are only partly visible.

- Char (character)
- Varchar (variable character)
- Long Varchar
- Graphic
- Vargraphic (variable graphic)
- Long Vargraphic

Column Snap when Scrolling

If scrolling left or right causes a column on the far left side of the screen to become obscured (see "Obscured Columns"), File-AID *for DB2* will shift the display to unobscure the column. This feature is called a column snap.

File-AID *for DB2* performs a column snap under all circumstances *except* the following:

- The scroll amount is set to an integer.
- The scroll amount is set to MAX.
- The scroll amount is set to CSR and the cursor is in the scrollable area of the screen.
- Scrolling right results in displaying the far right data in the table.

Note: If scrolling right to the end of the data leaves an obscured column on the left, subsequent scrolling to the right toggles between displaying the left column as unobscured and displaying the far right data.

Pending State

When you repeat or copy a row using the Edit function, the new row is placed into a pending state and a PEND flag displays on the table mode screen in place of the row's line number. When you insert or repeat a row in row mode, a row pending message displays under the command line for the new row. The rows are not actually added to the table until you save them. You can do one of the following to make pending rows a part of your table:

- Make a change to the new row and press Enter with Auto Commit ON.
- Issue the SAVE primary command.
- END the Edit session with AUTOSAVE ON.

Setting new rows into the pending state prevents you from adding exact copies of columns that are supposed to be unique. You can revise the columns before they are included in the table.

Although they are not part of the table, pending rows are treated just like all other rows. You can perform any operation on them, such as delete, copy, or exclude. The FIND and CHANGE primary commands affect them in the usual way, but the RESET command does not clear them from the table.

The only way to change a row from pending to permanent is to save it by one of the methods listed above.

Maximum Rows to Select

The MAXIMUM ROWS TO SELECT field displays on the Edit Selection Template screen. Entering a value in it limits the number of rows to be displayed. Because File-AID *for DB2* uses a DB2 cursor to fetch all DB2 information shown on the screen, limiting the number of rows eliminates reading more rows than necessary.

For DB2 Version 7 and higher, File-AID *for DB2* adds a FETCH FIRST n ROWS clause into the internaly generated SQL SELECT statement, using the specified number in MAXIMUM ROWS TO SELECT field. This improves the performance of the query and uses less memory than if the SQL SELECT statement were issued without the limiting clause. If an existing criteria is used and it includes FETCH FIRST n ROWS clause in it, File-AID *for DB2* replaces this field value by the number in the FETCH FIRST n ROWS clause.

When you have set a MAXIMUM ROWS TO SELECT value and the limit is satisfied, File-AID *for DB2* replaces the BOTTOM OF DATA marker with a MAXIMUM SELECTED ROWS marker and a row of plus signs. This message tells you that you are at the end of the selected data area but more of the object exists. The default value for this field is set through Option 0, Defaults and User Parms (see "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2) or through the PROFILE primary command (see "PROFILE (PRO)" on page 3-39).

Locking a Table During Edit

The LOCK TABLE option on the Edit Options screen lets you lock a table while you edit it. This prevents other users from accessing and updating the table until you are finished. File-AID *for DB2* accomplishes this by issuing the following SQL:

```
LOCK TABLE xxxx IN EXCLUSIVE MODE
```

Since the lock is released and immediately reacquired whenever you issue a SAVE command, it is possible for another user to access the table during the release.

The lock is also released and reacquired whenever you press Enter with Auto Commit ON. For this reason, you may want to set Auto Commit OFF so you have exclusive access to your table between SAVE commands. The Auto Commit option can be set through Option 0, Defaults and User Parms (see "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2) or through the PROFILE primary command (see "PROFILE (PRO)" on page 3-39).

Method of Updating DB2

When you update data, File-AID *for DB2* builds the appropriate SQL statement to reflect the changed row in the table. The only columns included in the statement are the ones in which a value has changed. However, the WHERE clause identifies the old values for all columns to ensure that the correct row gets updated.

If the table has a primary key or a unique index with the RETRIEVE UNIQUE KEY INFO option set to ON (see "PROFILE (PRO)" on page 3-39), File-AID *for DB2* produces the following statement:

```
UPDATE your-table
   SET col2 = col2-newvalue, col7 = col7-newvalue
   WHERE col1 = col1-oldvalue, col2 = col2-oldvalue, ....
```

If there is no unique index for the table or the RETRIEVE UNIQUE KEY INFO option is OFF and no primary key exists, File-AID *for DB2* produces the following statement:

```
DECLARE cursorname CURSOR FOR
   SELECT col2, col7
   WHERE col1 = col1-oldvalue, col2 = col2-oldvalue, ....
   FOR UPDATE OF col2, col7

UPDATE tablename
   SET col2 = col2-newvalue, col7 = col7-newvalue
   WHERE CURRENT OF cursorname
```

Thus, if multiple rows match the WHERE clause, File-AID *for DB2* updates only the first one it finds.

If DB2 returns a +100 code **Row not found for UPDATE** File-AID *for DB2* displays the Concurrent Update Intercept window (Figure 6-13 on page 6-20) for you to decide how to continue.

Editing Primary Column in a View

When browsing or editing a view with primary key columns, the primary key columns are not identified as primary unless the Profile option RETRIEVE UNIQUE KEY INFO had been set to ON. In order to update a primary key column in a view, the Profile option RETRIEVE UNIQUE KEY INFO must be ON before you start the edit session. Edit the view in a single table/view edit session. Unless the view is the only edit object, or the first in a multiple table/view edit session, updates to the primary key columns fail (DB2 returns -535). See also "Method of Updating DB2" on page 6-3 and "Retrieve UNIQUE KEY information prior to edit" on page 5-2 and "RETRIEVE UNIQUE KEY INFO" on page 3-41.

Columns with Defined Default Values

For character columns with defined default values the Profile option "SPACES IMPLY NULL" must be ON so File-AID *for DB2* can fill in the default values for blank columns when inserting or updating the column.

Starting a Browse or Edit Session

To access the **Browse** function, select option **1**, Browse, from the Primary Option Menu. To access the **Edit** function, select option **2**, Edit, from the Primary Option Menu. The Browse or Edit Option screen displays (Figure 6-1).

The Browse Options screen and the Edit Options screen are identical except for different screen titles. Also, the Lock Table and Audit Trail options only appear on the Edit Options screen.

Figure 6-1. Edit Options screen

```
File-AID for DB2 ----------------- Edit Options -------------------------------
COMMAND ===>                                                    SSID ===> DSN

Specify DB2 Table (* or pattern for list)
   Location ID ===>                        Optional:
   Creator     ===> USERID                    Database    ===>
   Table Name  ===> *                         Tablespace  ===>

Specify Edit Template Information:
   Use Existing Criteria ===> NO       (Yes or No)
   Edit Criteria Dataset ===> EDITCRIT
   Member                ===>          (Blank or Pattern for list)

Specify Edit Options:

   Initial Display Mode   ===> 1           (1 - Table, Multiple rows per screen
                                            2 - Row, One row per screen)

   Lock Table during Edit ===> NO          (Yes or No)
   Use Uncommitted Read   ===> NO          (Yes or No)
   Maximum Rows to Select ===> 2000        (* for all rows)

   Create Audit Trail     ===> NO          (Yes or No)
   Relationship File   ===> 'USERID1.KSDS.RELATION'
```

Specify DB2 Table Area

Location ID

Optional. Location of the table or view you plan to edit. Specify a value if you want to browse or edit a table not at your current location. Wildcard characters (*****, _, %) are valid. See "Wildcard Characters" on page 2-9 for more information.

If you enter a wildcard character in this field, the Location Selection pop-up (Figure 6-2 on page 6-8) displays.

Note: File-AID *for DB2* must be installed on the remote DB2 subsystem with packages. DDF must have been installed for your current subsystem or you receive the error message **Location name not in use**.

Creator

ID of the person who created the table or view you plan to edit. Wildcard characters (*****, _, %) are valid. See "Wildcard Characters" on page 2-9 for more information.

If you enter a wildcard character in this field, the Edit Table Selection list (Figure 6-4 on page 6-9) displays.

Table Name

Name of the object you want to edit. Tables, aliases, synonyms and views are valid. Wildcard characters (*****, _, %) are also valid. See "Wildcard Characters" on page 2-9 for details.

If you enter a wildcard character in this field, the Table Selection window (Figure 6-4 on page 6-9) displays.

Optional Area

Note: Specifying an optional database and/or tablespace is only in effect when the Creator field or the Table Name field contain an asterisk.

Database

Name of the database to which you want to confine your selection list.

Tablespace

Name of the tablespace to which you want to confine your selection list.

Specify Edit Template Information Area

Use Existing Criteria

Enter whether you want to use a previously saved Edit Selection Template dataset. Edit Selection Criteria consists of conditions to limit and order column selection and/or row selection based on column values. This field is prefilled with the last saved entry (YES or NO).

NO Do not use a previously saved Edit Selection Criteria dataset.

YES Apply the criteria stored in the dataset specified in the **Edit Criteria Dataset** field. Then you can verify and/or modify the selection criteria in the Edit Selection Template screen (Figure 6-5 on page 6-10).

Edit Criteria Dataset

If you specified **Yes** in the **Use Existing Criteria** field, specify the name of the dataset that contains the edit template criteria you want to use. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Note: The criteria dataset must contain a valid SELECT statement. File-AID *for DB2* only uses the first SELECT statement of the dataset. If a valid SELECT is missing, or the table name of the SELECT statement does not match the entered name in the above Table Name field, File-AID *for DB2* gives an error (see "Selection Template Not Prefilled" on page 6-11). When File-AID *for DB2* processes the SELECT statement to prefill the selection template, it ignores extraneous clauses such as WITH, OPTIMIZE, FOR FETCH, and CURRENT OF.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to use. The field is prefilled with the last specified member name. If you leave this field blank or specify an asterisk (*****), File-AID *for DB2* displays the Member List for the dataset. Specify a pattern, for example PART*****, to narrow down the member list to the matching member names. The asterisk (*****) is the only valid pattern character.

Specify Edit Options Area

Initial Display Mode

Enter the desired initial display mode. This field is updated with the last mode in use before exiting your edit session.

1 Displays multiple rows per screen. Column names are listed across the top of the screen. This is called *table mode*.

2 Displays one row per screen. Column names are listed down the left side of the screen and column data is listed down the right side of the screen. This is called *row mode*.

See "Display Modes" on page 6-14 for more information.

Lock Table During Edit

Indicate whether you want exclusive access to the table during Edit. (See "Locking a Table During Edit" on page 6-3 for more information.)

Use Uncommitted Read

Specify whether to enable retrieval of uncommitted data for the browse or edit session. The field is prefilled with the last specified value (YES or NO).

NO Only retrieve committed data from tables without write locks. Even if your selection criteria SELECT statement includes a WITH UR clause, it is not honored.

YES Retrieve data, including data not yet committed by another user or application. The SELECT statement which File-AID *for DB2* uses to get table data into edit/browse has the suffix WITH UR appended. This allows reading through any locks which may be held on that table.

Maximum Rows to Select

Specify a numeric value indicating the maximum number of rows to be selected, or specify an asterisk (*) to select all rows. The default value is retrieved from User Parms (Option 0). See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2 for information on changing the default. See "Maximum Rows to Select" on page 6-3 for more information on this field.

Create Audit Trail

Indicate whether you want to create an audit trail for your Edit session. This field may not be present on your screen. See "Creating an Audit Trail Dataset" on page 6-31 for a detailed explanation of this feature.

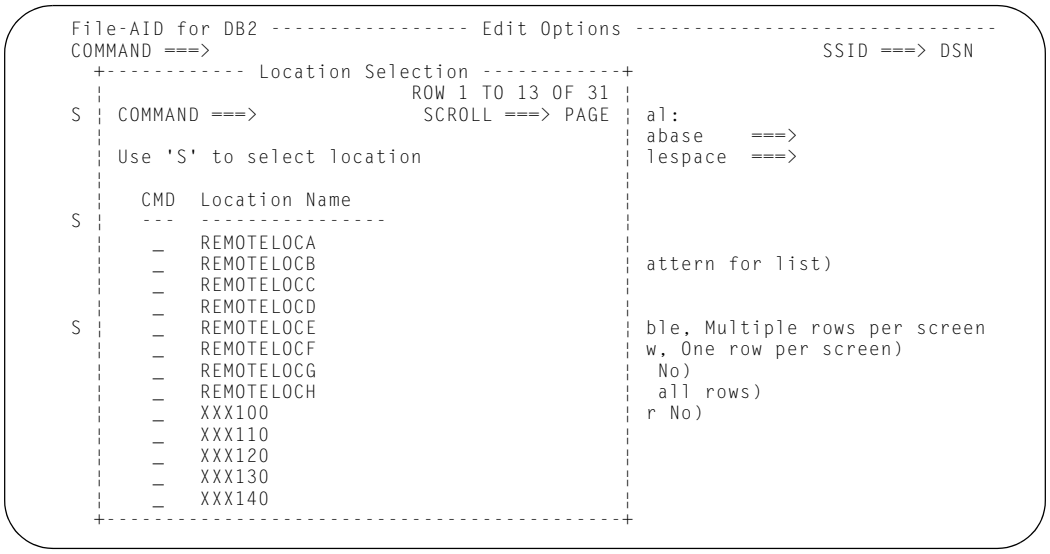
Relationship File

Optional. Enter the name of the file that stores your relationship definitions. You must specify this file to browse or edit related tables. See Chapter 7, "Related Table Browse and Edit" for more information.

Location Selection Window

The Location Selection window displays if you type an asterisk or wildcard character in the Location field on the Edit Options screen. The window provides you with a list of locations to be selected. If you press END from this window without selecting a location, File-AID *for DB2* uses the current location.

Figure 6-2. Location Selection Window



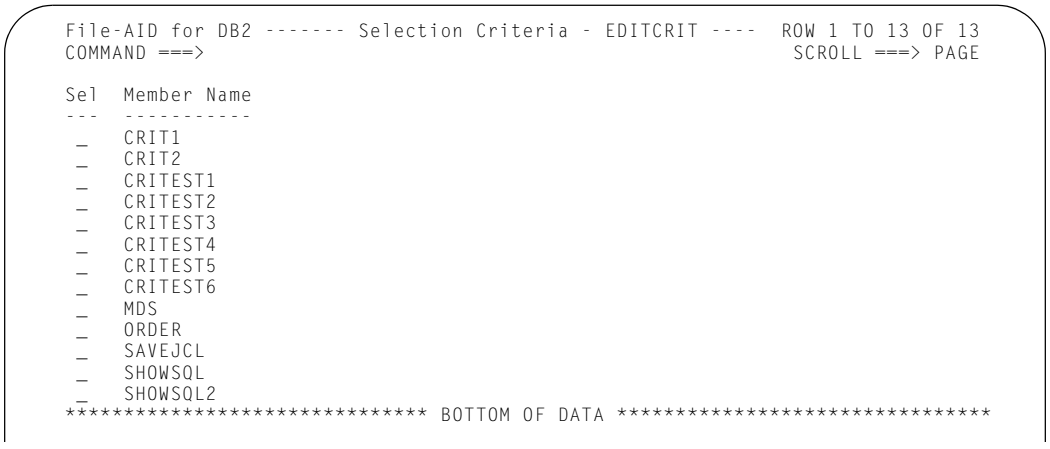
Sel Entry field for selection.

Location DB2 remote location.

Selection Criteria Member List

The Selection Criteria Member List displays if you left the Selection Criteria Member field on the Edit Options screen blank or entered a ***** pattern. It provides a list of members of the specified Selection Criteria dataset.

Figure 6-3. Member Selection Criteria List screen



Sel Enter the **S** line command to select one of the listed members as your selection criteria.

Member Name Lists all member names of the selection criteria dataset.

Table Selection Window

The Table Selection window displays when you enter a wildcard character in the Creator and/or Table Name field on the Edit Options screen. It provides a list of tables or views to be selected for browsing or editing.

Figure 6-4. Edit Table Selection Window

```
File-AID for DB2 ----------------- Edit Options -------------------------------
COMMAND ===>                                                    SSID ===> DSN
            +--------  Edit Table Selection List - SSID:DSN  ---------+
Specify DB2 |                                    ROW 1 TO 10 OF 10  |
   Location | COMMAND ===>                       SCROLL ===> PAGE   |
   Creator  |                                                       |
   Table Na | Use 'S' to select                                     |
            |                                                       |
Specify Edi |   Cmd  Creator  Table              Type               |
   Use Exis |   ---  -------- ------------------ -----              |
   Edit Cri |    _   USERID   EMP                TABLE              |
   Member   |    _   USERID   EMP_NEW            TABLE              |
            |    _   USERID   EMPA               TABLE              |
Specify Edi |    _   USERID   EMPALS             ALIAS              |
   Initial  |    _   USERID   EMPNEW             TABLE              | screen
            |    _   USERID   EMPPROJACT         TABLE              |
   Lock Tab |    _   USERID   EMPTAB1            TABLE              |
   Maximum  |    _   USERID   EMP2               TABLE              |
   Create A |    _   USERID   EMP3               TABLE              |
            |    _   USERID   EMP4               TABLE              |
   Relation | ****************** BOTTOM OF DATA ********************  |
            |                                                       |
            +-------------------------------------------------------+
```

Cmd Enter **S** to select the table or view.

Creator Lists creator IDs of the tables to be selected.

Table Lists the tables to be selected.

Type Object type for each object listed.

Edit Options

This section discusses the edit options available in File-AID *for DB2* including:

- Selection Template
- SQL Selection (SPUFI)

Edit Selection Template

The Selection Template screen displays after you completed the Edit Options screen. It lets you specify the rows or columns of the object you want to edit. You can specify the order in which the columns are displayed. You can also specify the rows you want to use and the order in which they are displayed.

The message `Trigger(s) exist` informs you that one or more triggers are defined against the selected table. Trigger failures during the edit session will generate visible SQL codes.

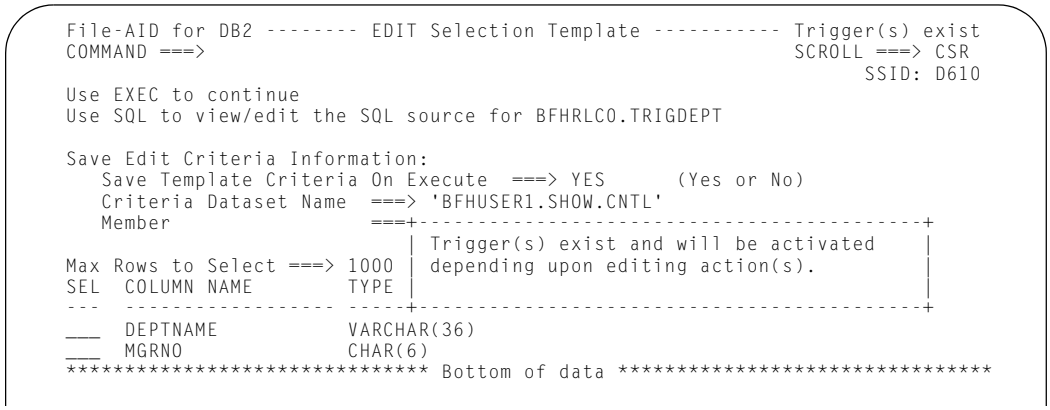
The short message **LOB(s) exist** informs you that one or more LOB columns are part of the selected table. The LOB column is not listed in the column list area as it cannot be selected for edit or browse.

If you are using an existing Edit Criteria Dataset, the SEL column is already prefilled with the selections from the Edit Criteria Dataset. You can then either accept or modify them as desired.

Note: You achieve "Full Table Edit" by NOT selecting any columns in the SEL column when entering the EXECUTE command.

A warning message displays on the Edit Selection template if triggers exist and is activated by editing.

Figure 6-5. Edit Selection Template



Save Edit Criteria Information Area

Save Template Criteria On Execute

Enter **YES** to save the current Edit Selection Criteria as shown on the current EDIT Selection Template screen to a dataset.

Note: The selection criteria is saved when you enter the EXECUTE command on this screen. It is, however, NOT saved if you enter the SQL command and execute the SQL directly. The executed SQL is saved in the dataset as specified with "Allocate SHOW SQL Dataset" on page 5-5.

Enter **NO** (default) when you do not want to save the current Edit Selection Criteria to a dataset.

Edit Criteria Dataset

Enter the name of the dataset you want the Edit Selection Criteria to be saved to. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to save the selection criteria to.

Max Rows to Select

Maximum number of rows to be returned based on your selection criteria. See "Maximum Rows to Select" on page 6-3 for details.

SEL

Type **S** (select) beside the column or columns you want to edit. If you want the columns displayed in a particular order, type the appropriate integer beside each column. Mixing integers and S selections displays in the integer-selected columns first, with the S-selected columns following. File-AID *for DB2* displays the column(s)

in edit or browse when you use the EXECUTE command. If you do not mark any columns for selection, ***all*** columns are selected and presented in the order listed.

Column Name

Names of all columns within the table or view.

Type(Length)

Type and length of the data in the column.

Distinct types are identified by their names.

Note: Columns with type LOB or distinct based on LOB will not be displayed on the template selection screen as they cannot be edited or browsed.

Order

Specify an integer to define how the returned rows should be sorted on the screen, based on the data content of the columns. Entering **1** indicates the primary sort column.

A/D

Specify whether the rows on this column data are sorted in ascending (A) or descending (D) order. If neither is specified, the columns are sorted in ascending order.

Where

Specify values limiting the rows displayed. The WHERE delimiter is analogous to the SQL WHERE clause in a SELECT statement. Up to 256 characters are available by scrolling right. You may use the following delimiter in your WHERE clauses:

>	OR	< >
<	> =	BETWEEN
=	< =	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

Note: Special considerations for distinct (UDT) data type columns: File-AID for DB2 generates the appropriate CAST function for simple WHERE clauses that contain just one of the following operators:

>	> =	< >
<	< =	¬ <
=	¬ =	¬ >

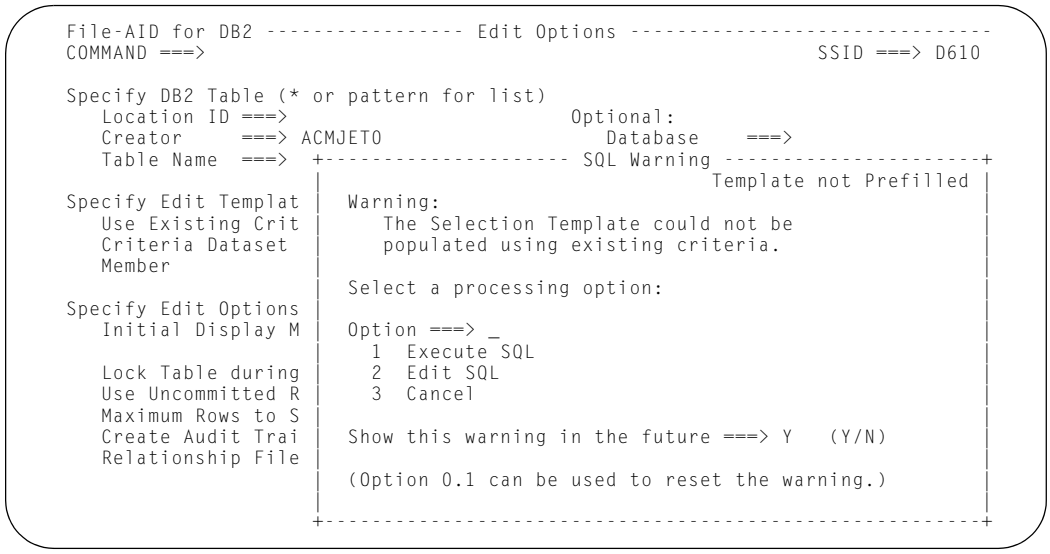
For all other WHERE clauses, you must enter the CAST function manually.

Selection Template Not Prefilled

The Selection Template screen cannot be prefilled when there is an error processing the selection criteria. These errors could include:

- a mismatch between the table name in the SELECT statement of the Edit Criteria dataset and the one specified in the Table field of the Edit Options screen
- the specified criteria does not contain a valid SELECT statement
- the dataset is empty

Figure 6-6. Selection Template Not Prefilled



Now you have several choices:

- Select option 1 to execute the SQL of the existing Selection Criteria Dataset. If it contains a valid SELECT statement you can then edit or browse the table specified in the selection criteria.
- Select option 2 to view and edit the SQL of the existing Selection Criteria Dataset before executing it. If the dataset or dataset member is empty or non-existing you can insert a SELECT statement in the Edit SPUFI Selection screen. See also “SQL Entry Panel” on page 6-12 and “Edit SPUFI Selection” on page 6-13. You then edit or browse the table specified in the SELECT statement.
- Specify option 3 to cancel and return to the Edit Options screen where you can:
  - Select a different creator and table combination that match the table name of the Edit Criteria.
  - Specify **NO** in the Use Existing Criteria field.
  - Specify a different Edit Criteria dataset member combination that matches the creator and table combination.

SQL Entry Panel

The SQL Entry Panel displays if you enter the **SQL** primary command on the Edit Selection Template screen whenever the screen could not be prefilled. It prompts you to name and describe a dataset. The dataset can have a previously saved SELECT statement or it can be empty, allowing you to enter a new SELECT statement. If the dataset contains more than one SELECT statement, only the first statement is executed.

Figure 6-7. SQL Entry Panel

```
File-AID for DB2 ------------------ SQL ENTRY PANEL ----------------- SSID: DSN
COMMAND ===>

Specify the input dataset name:         (Can be sequential or partitioned)
 1  DATASET NAME  ... ===> EDITCRIT(SHOWSQL)
 2  VOLUME SERIAL ... ===>              (Enter if not cataloged)
 3  DATASET PASSWORD  ===>              (Enter if password protected)
 4  MAX SELECT ROWS . ===> 2000         (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT ..... ===> YES           (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

Dataset Name

Name of the dataset (either partitioned or sequential) in which the SQL query resides or will reside. Prefilled with the Edit Criteria dataset name.

Volume Serial

Disk pack on which the dataset resides, if it is not cataloged.

Dataset Password

Password for the dataset, if it is password protected.

Max Select Row

Maximum number of rows to be returned for the SELECT. (An asterisk in this field indicates that all rows should be returned. See "Maximum Rows to Select" on page 6-3 for more information.) Only the first SELECT command in a dataset is executed.

Edit Input

Specify whether you want to edit the SQL before it is executed.

If the value of the Edit Input field is YES, pressing Enter displays the Edit SPUFI Selection screen. If the value of the Edit Input field is NO, pressing Enter executes the SQL statement and displays the Edit mode screen.

Edit SPUFI Selection

You access the Edit SPUFI selection screen from either of the following:

- SQL Entry Panel when you enter **YES** in the Edit Input field on the SQL Entry Panel.
- Prefilled EDIT Selection Template screen when you enter the **SQL** command. In that case the SQL is saved in the default SQL dataset as specified in "Allocate SHOW SQL Dataset" on page 5-5.

On this screen, you can enter a new SELECT statement or alter the existing one. The statement should be in standard SPUFI format.

Figure 6-8. Edit SPUFI Selection screen

```
EDIT --------------- USERID.EDITCRIT(SHOWSQL) - COLUMNS  001 072
COMMAND ===>                                          SCROLL ===> PAGE
****** ***************************** TOP OF DATA ******************************
000001 SELECT LASTNAME, FIRSTNAME, MIDINIT, EMPNO, JOB, BIRTHDATE, PHONENO,
000002   HIREDATE
000003   FROM USERID.EMP
000004   WHERE EMPNO > '000110'
000005   ORDER BY USERID.EMP.EMPNO, USERID.EMP.JOB;
****** **************************** BOTTOM OF DATA ********************************
```

Text Entry Area

Change or add to the SELECT statement using ISPF/PDF Edit. For more information concerning SPUFI, see the *IBM SQL Reference Guide*.

When you are satisfied with your SQL, press END to end the edit session.

- If you entered from the SQL Entry screen you return to the SQL Entry screen and are advised to press Enter to execute the SQL statement (Figure 6-9).
- If you entered from the EDIT Selection Template screen and modified the displayed SQL, you are asked **Execute modified SQL?**. YES executes the SQL statement and you enter Edit or Browse; NO returns to the Selection Template screen.

Figure 6-9. SQL Entry Panel

```
File-AID for DB2 ------------------ SQL ENTRY PANEL ----------------- SSID: DSN
COMMAND ===>

Edit session has completed. Press enter to execute SQL statement
Enter the input dataset name:           (Can be sequential or partitioned)
 1  DATASET NAME  ... ===> EDITCRIT(SHOWSQL)
 2  VOLUME SERIAL ... ===>              (Enter if not cataloged)
 3  DATASET PASSWORD  ===>              (Enter if password protected)
 4  MAX SELECT ROWS . ===> 2000         (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT ...... ===> *            (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

Display Modes

File-AID *for DB2* offers two different methods of browsing and editing object data: table mode and row mode. You can switch between these modes by entering the primary commands: TABLE and ROW. If you are in table mode and would like to see as many columns as possible, enter the ROW command to toggle into row mode. The row that was at the top of the screen displays. You can display a specific row from the table mode screen by typing the ROW command, positioning the cursor on the desired row, and pressing Enter. Conversely, if you are in row mode and would like to switch back to table mode to work with multiple rows of data, enter the TABLE command. The row that was displayed when the TABLE command was entered displays at the top of the screen.

Note: You can also select these commands from the Views pull-down menu. However, the cursor sensitive feature, as described for the ROW command above, for example, is not available then.

Table Mode

In table mode, each row of data displays in columns listed across the screen. This mode allows you to work with as many rows of data as can be displayed on the screen.

The Edit screen displays in table mode when you either:

- Enter **1**, for Table, in the Initial Display Mode field on the Edit Options screen.
- Enter **TABLE** on the command line on the Edit row mode screen.
- Select **TABLE** from the VIEWS pull-down menu on the Edit row mode screen.

The table mode screen displays the rows of an object in column format. The TABLE command, when entered on the command line, is cursor sensitive. When going from row mode to table mode, you can position the table mode display at a particular column. Type TABLE, place the cursor on the desired column, and press Enter.

When in Browse, you can only review the data. When in Edit, you can change the data by typing over the body of the screen. Primary commands for table display mode can be entered from the command line or selected from the pull-down menus. Line commands are also available in the line-number area.

Figure 6-10. Edit screen in Table Mode

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit - USERID.EMP                    SSID: DSN
COMMAND ===>                                                  SCROLL ===> PAGE
        LASTNAME         FIRSTNAME     MIDINIT   EMPNO    JOB        BIRTHDATE
        VC(15)           VC(12)        CHAR(1)   CHAR(6)  CHAR(8)         DATE
        ---------------  ------------  -------   PRIMARY  --------  ----------
******  *************************** TOP OF DATA ****************************
000001  O'CONNELL        SEAN                    000120   CLERK     1942-10-18
000002  QUINTANA         DOLORES       M         000130   ANALYST   1925-09-15
000003  NICHOLLS         HEATHER       A         000140   ANALYST   1946-01-19
000004  ADAMSON          BRUCE                   000150   DESIGNER  1947-05-17
000005  PIANKA           ELIZABETH     R         000160   DESIGNER  1955-04-12
000006  YOSHIMURA        MASATOSHI     J         000170   DESIGNER  1951-01-05
000007  SCOUTTEN         MARILYN       S         000180   DESIGNER  1949-02-21
000008  WALKER           JAMES         H         000190   DESIGNER  1952-06-25
000009  BROWN            DAVID                   000200   DESIGNER  1941-05-29
000010  JONES            WILLIAM       T         000210   DESIGNER  1953-02-23
000011  LUTZ             JENNIFER      K         000220   DESIGNER  1948-03-19
000012  JEFFERSON        JAMES         J         000230   CLERK     1935-05-30
000013  MARINO           SALVATORE     M         000240   CLERK     1954-03-31
000014  SMITH            DANIEL        S         000250   CLERK     1939-11-12
000015  JOHNSON          SYBIL         V         000260   CLERK     1936-10-05
000016  PEREZ            MARIA         L         000270   CLERK     1953-05-26
```

Line Number Area

Entry area for line commands. All line commands except M and the SQL Analysis commands EX, XP, BR, and ED are valid for the Edit table mode screen. See Chapter 4, “Line Commands” for details.

Column Headings

- First Line - the column name.
- Second Line - column data type:

```
CHAR            INTEGER          ROWID
VC              DEC
LVC             FLOAT
GRAPH           DATE
VG              TIME
LVG             TIMESTAMP
SMALLINT        DISTINCT NAME
```

For the data types CHAR, VC, LVC, GRAPH, VG, LVG, and DEC, the length is in parentheses.

- Third Line - If the column length is less than 19 and the column is not part of an index, dashes are displayed to indicate where the data begins. If the column length is greater than 18 and the column is not part of an index, a ruler displays.

  ```
  For example:  ----+---10----+---2
  ```

If the column is part of an **index**, the type of index is shown by the indicators listed below. The index type indicators are available only for local tables.

PRIMARY Column is part of a primary key.

UNIQUE Column is part of a unique index. The RETRIEVE UNIQUE KEY INFO option must be set to ON for this heading to display. See "PROFILE (PRO)" on page 3-39 for information on setting this option.

FOREIGN Column is part of a foreign key relationship. The SHOW FOREIGN KEYS option must be set to ON for this heading to display. See "PROFILE (PRO)" on page 3-39, for more information on setting this option.

PRIM+FRGN Column is part of both a primary key and foreign key relationship.

If the column is an identity column, the type of identity column is shown by the indicators listed below. The identity column type indicators are available only for local tables.

IDENTITY Column is an identity column.

PR+FR+ID Column is part of a primary key, a foreign key, and is an identity column.

PRIM+ID Column is part of a primary key and is an identity column.

UN+IDNT Column is part of a unique key and is an identity column.

Column Values

This area displays the data values. Character and graphic fields are left justified. Numeric fields are right justified. Valid values based on data type are listed in Table 6-1.

Table 6-1. Valid Values Based on Data Type

Data Type	Valid Value
CHAR	Fixed-length character strings less than or equal to the field length specified when the column was created. Maximum field length is 255 characters.
VCHAR	Variable-length character strings less than or equal to the field length specified when the column is created. Maximum allowable field length is 4056 characters. If the column is defined as FOR BIT DATA, the data cannot be successfully edited.
LVCHAR	Variable-length character strings less than or equal to the field length specified when the column is created. Maximum allowable field length is 4056 characters. If the column is defined as FOR BIT DATA, the data cannot be successfully edited.
GRAPH	Fixed-length graphic strings less than or equal to the field length specified when the column was created. Maximum field length is 127 DBCS characters.
VARGRAPHIC	Variable--length graphic strings less than or equal to the field length specified when the column was created. Maximum allowable field length is 4046 bytes (2023 DBCS characters) for 4-KB pages; 8128 bytes (4064 DBCS characters) for 8-KB pages; 16320 bytes (8160 DBCS characters for 16-KB pages; 32704 bytes (16352 DBCS characters) for 32-KB pages.
LVARG	Variable--length graphic strings less than or equal to the field length specified when the column was created. Maximum allowable field length is 4046 bytes (2023 DBCS characters) for 4-KB pages; 8128 bytes (4064 DBCS characters) for 8-KB pages; 16320 bytes (8160 DBCS characters for 16-KB pages; 32704 bytes (16352 DBCS characters) for 32-KB pages.
SMALLINT	Fixed-length, binary integers in the range of -32768 to +32767, requiring two bytes of storage.
INTEGER	Fixed-length, binary integers in the range of -2,147,483,648 to +2,147,483,647, requiring four bytes of storage.
DEC	Fixed-length, decimal numbers in the range of -10E15 to +10E15.

Table 6-1. Valid Values Based on Data Type

Data Type	Valid Value
FLOAT	Fixed-length, floating-point numbers in the approximate range of -7.2E75 to +7.2E75.
DATE	A three-part value (year, month, day), whose format is determined by DB2.
TIME	A three-part value (hour, minute, second), whose format is determined by DB2.
TIMESTAMP	A seven-part value (year, month, day, hour, minute, second, microsecond) whose format is YYYY-MM-DD-HH.MM.SS.TTTTTT.
DISTINCT	Distinct is identified by its name. A Distinct type is based on one of the built-in data types. Use the UDT primary command (see "UDT" on page 3-57) to view the Distinct type, schema, and the built-in data type.
ROWID	Uniquely identifies a row in a table. Each value must be unique. Generated by DB2.Length as described in the LENGTH column of the catalog table SYSCOLUMNS is the internal length, 17 bytes. Length as described in the LENGTH2 column of catalog table SYSCOLUMNS is the external length, 40 bytes.

Row Mode

In row mode, a single row of data displays on the screen. If there is more data than can fit on a screen, you must scroll down to view the data. Column names are listed down the left side of the screen, and column attributes are listed in the center of the screen. Data displays on the right side of the screen.

The Edit screen displays in row mode when you either:

- Enter **2**, for row, in the Initial Display Mode field on the Edit Options screen.
- Enter the **ROW** command in the command line on the Edit table mode screen.
- Select **Row** from the VIEWS pull-down menu on the Edit table mode screen.

The row mode screen displays the first row of the object you selected or the first row displayed on the table mode screen. The ROW command is cursor sensitive when entered on the command line. When going from table mode to row mode, you can display a particular row from the table mode screen. Type ROW, place the cursor on the desired row, and press Enter.

If you are in Browse, you can only review the data. In Edit, you can change the data by typing over the column values. You can also add or delete a row using the INSERT, REPEAT, and DELETE primary commands. See Chapter 3, "Primary Commands" for more information. Figure 6-11 shows a sample Edit screen in row mode.

Figure 6-11. Edit screen in Row Mode

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2          Edit - USERID.EMP                        SSID: DSN
COMMAND ===>                                                 SCROLL ===> PAGE
Row 1
    COLUMN NAME        TYPE(LEN)  KEY            COLUMN VALUE
*** ------------------ --------- ----- ----+---10----+---20----+---30----+---40
001 EMPNO              CHAR(15)  PRIM  000290
002 FIRSTNME           VC(12)          JOHN
003 MIDINIT            CHAR(1)         R
004 LASTNAME           VC(15)          PARKER
005 WORKDEPT           CHAR(3)   FRGN  E11
006 PHONENO            CHAR(4)         4502
007 HIREDATE           DATE            1980-05-30
008 JOB                CHAR(8)         PROGRAMMER
009 EDLEVEL            SMALLINT        12
000 SEX                CHAR(1)         M
011 BIRTHDTE           DATE            1946-07-09
012 SALARY             DEC(9,2)        1534.00
013 BONUS              DEC(9,2)        300.00
014 COMM               DEC(9,2)        1227.00
015 DESC               CHAR(120)       Responsible for developing and
        (POS 41-80)                    maintaining payroll programs.
F10=PREV                                F11=NEXT
```

On the Edit row mode screen, the row number is indicated by the row counter on the third line. An added row (INSERT or REPEAT) is kept in a pending state until you change the data appropriately and save it (SAVE). The message line on the right-hand side of the title line indicates the status of an added or deleted row with messages such as **Insert Pending**, **2 Rows Saved**, **or 1 Row Deleted**.

Column Name

Displays either the name of the column or the positions of the COLUMN VALUE column when more than one line is required to display the contents of the column.

Type(Length)

Displays the column's data type and length (if applicable): CHAR, VC, LVCHAR, GRAPH, VG, LVG, SMALLINT, INTEGER, DEC, FLOAT, DATE, TIME, TIMESTAMP, DISTINCT, or ROWID. See Table 6-1 on page 6-16 for more information.

Key

If the column is part of an index, the index type indicators listed below are shown. The index type indicators are available only for local tables.

PRIM

Column is part of a primary key.

UNIQ

Column is part of a unique index. The INDICATE UNIQUE KEY INFORMATION PRIOR TO EDIT option must be set to ON for this heading to display. See User Parms, "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2, for more information.

FRGN

Column is part of a foreign key relationship. The INDICATE FOREIGN KEY COLUMNS WHILE IN EDIT option must be set to ON for this heading to display. See User Parms, "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2, for more information.

IDNT

Column is an identity column. The ALLOW UPDATE OF IDENTITY COLUMNS IN EDIT option must be set to **ON** for this heading to display. If the column is both an identity column and a primary key, then UPDATE PRIMARY KEY must also be **ON** to

edit the column. See User Parms, "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2, for more information.

PRID

Column is part of a primary key and is an identity column.

PRFI

Column is part of a primary key, a foreign key, and is an identity column.

IDFN

Column is part of a foreign key, and is an identity column.

UNID

Column is part of a unique key, and is an identity column.

Column Values

This area displays the data type values in either character or hexadecimal mode. A character column longer than the width of the area is split across as many consecutive lines as necessary. To change the column value, type over the existing value.

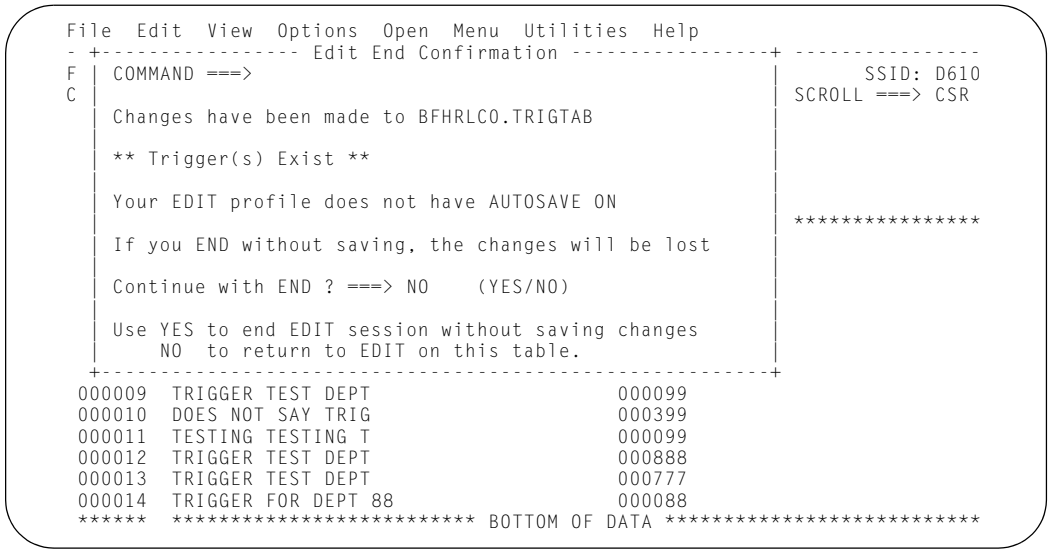
Edit End Confirmation

The Edit End Confirmation window displays if you have outstanding changes to your data, AUTOSAVE is OFF in your profile, and you enter the END command. It reminds you that if changes have been made, they will be lost unless saved. This window also notifies you if triggers exist (Figure 6-12).

If Auto Commit is ON, all changes are saved each time you press Enter. Therefore, this window is not displayed because there are no outstanding changes when you end your Edit session.

After your changes are saved, the message 'Trigger(s) Exist' displays if triggers exist.

Figure 6-12. Edit End Confirmation screen



Continue with END?

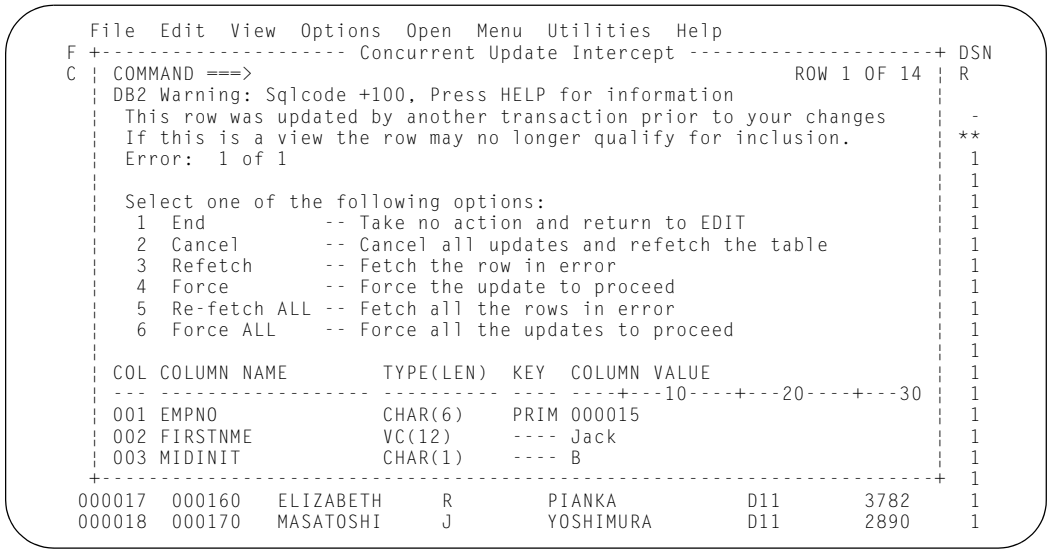
Specify YES to proceed and cancel your changes. Specify the default of NO to return to your edit session. You can also press END to return to the edit session. When in the edit session, issue the SAVE command to save changes.

Note: If you are in a multiple edit session (related or unrelated), this window displays for each table in which there are pending changes.

Concurrent Update Intercept

For each row that you change, an update is submitted to DB2 using all columns to qualify the update (see also "Method of Updating DB2" on page 6-3). If the row has not changed since being retrieved, the update succeeds and processing continues normally. However, if the row has changed, or when updating a view and the row no longer matches the view criteria, DB2 returns a +100 SQL code. Detecting the error, File-AID *for DB2* flags the row with **ERR==>** and displays the Concurrent Update Intercept window (Figure 6-13). The window displays once for each row in error, providing you with the ability to specify the disposition for each error individually. Before the window displays, the edit panel is scrolled to position the row in error at the top of the display.

Figure 6-13. Concurrent Update Intercept Window



End

Take no action and return to the Edit panel. This option provides you with full access to the data in the table before taking further action.

Cancel

Cancel all updates and refetch the table. A rollback is issued to cancel all changes since the last commit. The table is re-fetched and the display is positioned at row 1.

Refetch

Fetch the row in error. The row in error is re-fetched using all columns that are part of unique indexes, except columns that have changed. The row is flagged with **FETCH>** and the table is redisplayed. If the table is non-indexed, or the row no longer exists (for example, another user changed the index columns), the row cannot be re-fetched. In this case, an error message displays and the row is removed from the display.

Force

Force the update to proceed. The row in error is updated, using all columns in unique indexes, except those that have changed. The row is flagged with **UPDATE**. Your change overlays the update made by another user. If the update fails because the row no longer exists, the row is inserted and flagged with **INSERT**.

Re-fetch ALL

Fetch all the rows in error. Processing is the same as for re-fetch except that all rows in error are re-fetched. If you select this option, you avoid repeated window displays because File-AID *for DB2* processes all rows in error identically.

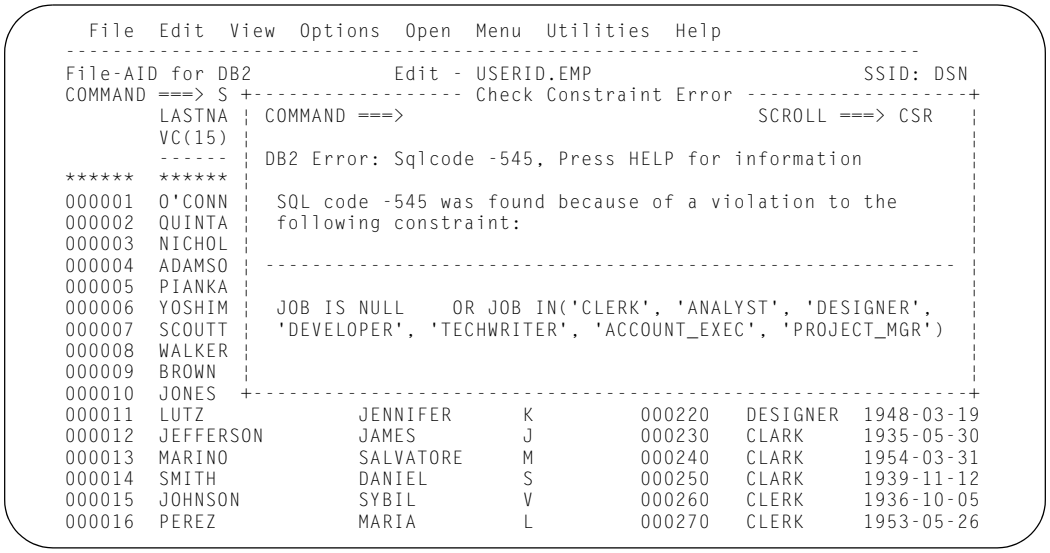
Force ALL

Force all the updates to proceed. Processing is the same as for force except that all rows in error are updated or inserted. If you select this option, you avoid repeated window displays because File-AID *for DB2* processes all rows in error identically.

Check Constraint Error

For each row that you update, File-AID *for DB2* verifies that it meets all table check constraints. If it detects a table check constraint violation, it displays the Check Constraint Error window (Figure 6-14). Press Help to view more detailed error descriptions. When you have viewed all available help, press END to exit the window. File-AID *for DB2* marks the rows in error with the **==ERR>** indicator so you can quickly correct the entered value for the column with the check constraint violation.

Figure 6-14. Check Constraint Error Window



The top part of the window shows the error message, while the bottom contains the check constraint statement for the column.

Note: When editing a view the constraint statement is not displayed.

Help Press HELP to view all extended help panels associated with this error window.

End Closes the error window. All rows in error are marked with **==ERR>** so you can quickly find and correct them (Figure 6-15).

Figure 6-15. Check Constraint Error Rows Marked

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit - USERID.EMP                    SSID: DSN
COMMAND ===>                                                  SCROLL ===> PAGE
        LASTNAME         FIRSTNAME     MIDINIT   EMPNO    JOB        BIRTHDATE
        VC(15)           VC(12)        CHAR(1)   CHAR(6)  CHAR(8)         DATE
        ---------------  ------------  -------   PRIMARY  --------  ----------
******  **************************** TOP OF DATA ****************************
000001  O'CONNELL        SEAN                    000120   CLERK     1942-10-18
000002  QUINTANA         DOLORES       M         000130   ANALYST   1925-09-15
000003  NICHOLLS         HEATHER       A         000140   ANALYST   1946-01-19
000004  ADAMSON          BRUCE                   000150   DESIGNER  1947-05-17
000005  PIANKA           ELIZABETH     R         000160   DESIGNER  1955-04-12
000006  YOSHIMURA        MASATOSHI     J         000170   DESIGNER  1951-01-05
000007  SCOUTTEN         MARILYN       S         000180   DESIGNER  1949-02-21
000008  WALKER           JAMES         H         000190   DESIGNER  1952-06-25
000009  BROWN            DAVID                   000200   DESIGNER  1941-05-29
000010  JONES            WILLIAM       T         000210   DESIGNER  1953-02-23
000011  LUTZ             JENNIFER      K         000220   DESIGNER  1948-03-19
==ERR>  JEFFERSON        JAMES         J         000230   CLARK     1935-05-30
==ERR>  MARINO           SALVATORE     M         000240   CLARK     1954-03-31
==ERR>  SMITH            DANIEL        S         000250   CLARK     1939-11-12
000015  JOHNSON          SYBIL         V         000260   CLERK     1936-10-05
000016  PEREZ            MARIA         L         000270   CLERK     1953-05-26
```

Editing Multiple Unrelated Tables

You can browse or edit multiple tables in a single session. The tables do not need to be related. For information on browsing and editing related tables, see Chapter 7, “Related Table Browse and Edit”.

An unlimited number of tables may be opened by repeatedly issuing the BROWSE or EDIT primary commands. These commands are only available in a standard browse or edit session; they are not available in Related Table browse and edit.

See “Referential Integrity Errors” on page 6-27 for information on using this feature to correct referential integrity errors. You can use the EDIT command to add the correct data to the parent table without leaving your edit session.

Figure 6-16. Using the EDIT Primary Command (from Open Pull-down)

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
-------------------------- +------------------+ ------------------------------
File-AID for DB2           ¦ 2  1. Browse...  ¦ MER_TABLE -         SSID: DSN
COMMAND ===>               ¦    2. Edit...    ¦               SCROLL ===>CSR
         SSNR          LAST ¦    3. Rbrowse... ¦         STREETADR
         CHAR(11)      VC(1 ¦    4. Redit...   ¦         VC(20)
         -----------   ---- +------------------+ ------  -----+---10----+---20
******   *************************** TOP OF DATA ***************************
 000001  217-33-4378  JOHNSON          LARRY             160 EMERSON AVENUE
 000002  216-56-4537  ADAMS            MIKE              23543 WEST STREET
 000003  215-87-5843  OWENS            PAT               156 SHERRIDAN AVE
 000004  214-48-2695  ROGERS           ROBERT            400 RACE STREET
 000005  213-25-4365  MCCARTHY         KEN               107 SAN CARLOS
 000006  211-24-6373  KAUFMAN          ED                160 LYNDALE AVENUE
 000007  210-36-2557  MALLONE          MIKE              475 49TH STREET
 000008  121-23-5837  REAGAN           BOB               7879 MAIN STR.
 000009  171-29-3842  BUSH             DONNA             780 LAKE SHORE
 000010  181-53-2737  KENNEDY          STEVE             180 5TH AVENUE
 000011  191-78-3623  FORD             MARK              340 WALL STREET
 000012  161-76-4728  BAKER            VINCE             460 POINT ROAD
 000013  151-78-3636  LINCOLN          FRED              120 FREEWAY DR
 000014  141-42-1626  STEVENSON        PEGGY             685 ROUTE 66
 000015  131-25-2826  ANDERSON         MIKE              5    SUNSET STRIP
 000016  121-23-4598  HARRIS           TOM               45665 EAST STREET
```

When you execute the EDIT primary command as (Figure 6-16), the Edit Options window displays (Figure 6-17 on page 6-23). In this window you specify the creator and table name and whether to use an existing Edit Criteria dataset. It is similar to the Edit Options screen (Figure 6-1 on page 6-5), however, it does not include the *Specify Edit Options area*.

Figure 6-17. Edit Options Window — After Entering EDIT

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit - USERID1.EMP                    SSID: DSN
COMMAND ===> EDIT                                              SCROLL ===> CSR
         EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONENO
+----------------- File-AID for DB2 - Edit Options - DSN --------------------+
¦ COMMAND ===>                                                               ¦
¦                                                                            ¦
¦ Specify DB2 Table (* or pattern for list)                                  ¦
¦    Creator     ===> USERID                   Database    ===>              ¦
¦    Table Name  ===> *                        Tablespace  ===>              ¦
¦                                                                            ¦
¦ Specify Edit Template Information:                                         ¦
¦    Use Existing Criteria ===> NO       (Yes or No)                         ¦
¦    Edit Criteria Dataset ===> EDITCRIT                                     ¦
¦    Member                ===>          (Blank or Pattern for list)         ¦
¦                                                                            ¦
+------------------------------------------------------------------------------+

 000012  000180   MARILYN       S        SCOUTTEN          D11       1682     1
 000013  000170   MASATOSHI     J        YOSHIMURA         D11       2890     1
 000014  000160   ELIZAAETH     R        PIANKA            D11       ----     1
 000015  000150   BRUCE                  ADAMSON           A00       4510     1
 000016  000120   SEAN                   O'CONNELL         A00       2167     1
```

Specify DB2 Table Area

Creator

ID of the person who created the table or view you plan to edit. Wildcard characters (*, _, %) are valid. See "Wildcard Characters" on page 2-9 for more information.

Table Name

Name of the table you want to edit.

Optional Area

Note: Specifying an optional database and/or tablespace is only in effect when the Creator field or the Table Name field contain an asterisk.

Database

Name of the database to which you want to confine your selection list.

Tablespace

Name of the tablespace to which you want to confine your selection list.

Specify Edit Template Information Area

Use Existing Criteria

Enter whether you want to use a previously saved Edit Selection Template dataset. Edit Selection Criteria consists of conditions to limit and order column selection and/or row selection based on column values. This field is prefilled with the last saved entry (YES or NO).

NO Don't use a previously saved Edit Selection Criteria dataset.

YES Apply the criteria stored in the dataset specified in the **Edit Criteria Dataset** field. Then you can verify and/or modify the selection criteria in the "Edit Selection Template" on page 6-9.

Edit Criteria Dataset

If you specified **Yes** in the **Use Existing Criteria** field, you must specify the name of the dataset that contains the edit template criteria you want to use. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

The criteria dataset must contain a valid SELECT statement. File-AID *for DB2* only uses the first SELECT statement of the dataset. If a valid SELECT is missing, or the table name of the SELECT statement does not match the entered name in the above Table Name field, File-AID *for DB2* gives an error (see "Selection Template Not Prefilled" on page 6-11). When File-AID *for DB2* processes the SELECT statement to prefill the selection template, it ignores extraneous clauses such as WITH, OPTIMIZE, FOR FETCH, and CURRENT OF.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to use. If you leave this field blank or specify an asterisk (*****), File-AID *for DB2* displays the Member List for the dataset. The asterisk (*****) is the only valid pattern character.

Viewing Multiple Tables

When viewing multiple tables, the second table displays in a window in the lower half of the screen (Figure 6-18). A maximum of three windows can be displayed at one time. If you open a fourth table, the first table scrolls off of the display. Each additional table moves up to make room for the next one.

Figure 6-18. Multiple Unrelated Edit Display

```
+--------- File-AID for DB2 - Edit - USERID1.EMPTAB1 - 1 of 2 - DSN ----------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===>                                          SCROLL ===> CSR     |
|         SSNR         LASTNAME         FIRSTNAME        STREETADR          |
|  ******  *************************** TOP OF DATA ***********************  |
|  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE |
|  000002  216-56-4537  ADAMS            MIKE             23543 WEST STREET  |
|  000003  215-87-5843  OWENS            PAT              156 SHERRIDAN AVE  |
|  000004  214-48-2695  ROGERS           ROBERT           400 RACE STREET    |
|  000005  213-25-4365  MCCARTHY         KEN              107 SAN CARLOS     |
+-------------------------------------------------------------------------------+
+--------- File-AID for DB2 - Edit - USERID1.EMPTAB2 - 2 of 2 - DSN ----------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===>                                          SCROLL ===> CSR     |
|         SSNR         LASTNAME         FIRSTNAME        STREETADR          |
|  ******  ************************* TOP OF DATA *************************  |
|  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE |
|  000002  216-56-4537  ADAMS            MIKE             23543 WEST STREET  |
|  000003  215-87-5843  OWENS            PAT              156 SHERRIDAN AVE  |
|  000004  214-48-2695  ROGERS           ROBERT           400 RACE STREET    |
|  000005  213-25-4365  MCCARTHY         KEN              107 SAN CARLOS     |
+-------------------------------------------------------------------------------+
```

You can only edit one table at a time. The table that can be edited is the *active table* in the *active window*. Active windows are highlighted. Non-active windows are protected and appear in blue. You can move between tables using the JUMP primary command, but you cannot move the cursor manually to another table. See "Manipulating Multiple Tables" on page 6-25 for more information on using the JUMP command.

Each window has a title bar consisting of the following information:

- Product name
- Product mode: EDIT or BROWSE
- Fully qualified table name: Location.creator.table
- Window number and total number of windows open. If a window displays: **2 of 6**, the 2 indicates that it is the second window opened, and the 6 indicates that six windows are currently open.
- DB2 subsystem identification

Only column headings display in the windows. The second and third heading lines (column type, column length, key descriptions, and underscores) displayed in standard Edit no longer display.

When viewing an active window:

- Column headings for primary keys are highlighted white.
- Column headings for foreign keys are highlighted yellow.

Manipulating Multiple Tables

Use the JUMP primary command to move from window to window. Once you jump to a new window, it becomes active. You can only edit data in an active window. See "JUMP (J)" on page 3-30 for command syntax. Figure 6-19 shows the command **JUMP 1** being issued in the last window. After this command is executed, window 1 displays at the top of the screen and becomes active. Window 4 scrolls off of the display (Figure 6-20).

To facilitate jumping to a new table, assign JUMP to a PF key.

Figure 6-19. Issuing the JUMP Primary Command

```
+--------- File-AID for DB2 - Edit - USERID1.EMPTAB2 - 2 of 4 - DSN ----------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR      ¦
¦          SSNR         LASTNAME         FIRSTNAME        STREETADR         ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE ¦
+------------------------------------------------------------------------------+
+----------- File-AID for DB2 - Edit - USERID1.EMP - 3 of 4 - DSN ------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR      ¦
¦          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONEN ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  000330   BOB                     LEE              E21       2103   ¦
+------------------------------------------------------------------------------+
+---------- File-AID for DB2 - Edit - USERID1.ORDER - 4 of 4 - DSN -----------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===> JUMP 1                                   SCROLL ===> CSR      ¦
¦          ORD_NUM  CUST_NUM  PART_NUM  ORDER_DATE   PLN_QTY    PLN_AMT      ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  FF6666   CN0001    JK980S    ----------       234    6432.68      ¦
+------------------------------------------------------------------------------+
```

Figure 6-20. New Display — After issuing the JUMP Primary Command

```
+--------- File-AID for DB2 - Edit - USERID1.EMPTAB1 - 1 of 4 - DSN ----------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR      ¦
¦          SSNR         LASTNAME         FIRSTNAME        STREETADR         ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE ¦
+------------------------------------------------------------------------------+
+--------- File-AID for DB2 - Edit - USERID1.EMPTAB2 - 2 of 4 - DSN ----------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR      ¦
¦          SSNR         LASTNAME         FIRSTNAME        STREETADR         ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE ¦
+------------------------------------------------------------------------------+
+----------- File-AID for DB2 - Edit - USERID1.EMP - 3 of 4 - DSN ------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                    ¦
¦ ------------------------------------------------------------------------   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR      ¦
¦          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONEN ¦
¦  ******  ************************ TOP OF DATA ************************      ¦
¦  000001  000330   BOB                     LEE              E21       2103   ¦
+------------------------------------------------------------------------------+
```

Use the MAX primary command to enlarge a window to full screen size.Use the MIN primary command to reduce the window size and resume multiple table viewing. Once a table is maximized, all windows are maximized when using the JUMP command. Tables remain maximized until you issue the MIN command.

To facilitate sizing tables, assign MIN and MAX to PF keys, or select the commands from the View pull-down menu.

Editing Multiple Tables in Row Mode

Enter the ROW primary command to display the active window in row mode. For further information on editing in row mode, see “Row Mode” on page 6-17. Figure 6-21 shows the ROW primary command being issued. Figure 6-22 shows window 2 in row mode.

Figure 6-21. Issuing the ROW Primary Command

```
+----------- File-AID for DB2 - Edit - USERID1.EMP - 1 of 2 - DSN ------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===>                                          SCROLL ===> CSR   |
|          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONEN |
|  ******  ************************** TOP OF DATA ************************** |
|  000001  000330   BOB                     LEE              E21       2103  |
|  000002  000320   SAM           V         MEHTA            E21       9990  |
|  000003  000310   MAUDE         F         SETRIGHT         E11       3332  |
|  000004  000300   PHILIP        X         SMITH            E11       2095  |
|  000005  000290   JOHN          R         PARKER           E11       4502  |
+-------------------------------------------------------------------------------+
+--------- File-AID for DB2 - Edit - USERID0.EMPTAB1 - 2 of 2 - DSN ----------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===> ROW                                      SCROLL ===> CSR   |
|          SSNR         LASTNAME         FIRSTNAME        STREETADR         |
|  ******  ************************ TOP OF DATA ************************** |
|  000001  217-33-4378  JOHNSON          LARRY            160 EMERSON AVENUE |
|  000002  216-56-4537  ADAMS            MIKE             23543 WEST STREET  |
|  000003  215-87-5843  OWENS            PAT              156 SHERRIDAN AVE  |
|  000004  214-48-2695  ROGERS           ROBERT           400 RACE STREET    |
|  000005  213-25-4365  MCCARTHY         KEN              107 SAN CARLOS     |
+-------------------------------------------------------------------------------+
```

Figure 6-22. Window 2 in Row Mode

```
+----------- File-AID for DB2 - Edit - USERID0.EMP - 1 of 2 - DSN ------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===>                                          SCROLL ===> CSR   |
|          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONEN |
|  ******  ************************** TOP OF DATA ************************** |
|  000001  000330   BOB                     LEE              E21       2103  |
|  000002  000320   SAM           V         MEHTA            E21       9990  |
|  000003  000310   MAUDE         F         SETRIGHT         E11       3332  |
|  000004  000300   PHILIP        X         SMITH            E11       2095  |
|  000005  000290   JOHN          R         PARKER           E11       4502  |
+-------------------------------------------------------------------------------+
+--------- File-AID for DB2 - Edit - USERID0.EMPTAB1 - 2 of 2 - DSN ----------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
| ------------------------------------------------------------------------    |
| COMMAND ===>                                          SCROLL ===> CSR   |
|     COLUMN NAME        TYPE(LEN)  KEY  COLUMN VALUE          ROW 1        |
| 001 SSNR               CHAR(11)        217-33-4378                        |
| 002 LASTNAME           VC(15)          JOHNSON                            |
| 003 FIRSTNAME          VC(15)          LARRY                              |
| 004 STREETADR          VC(20)          160 EMERSON AVENUE                 |
| 005 CITY               VC(15)          BOSTON                             |
| 006 STATE              CHAR(2)         MA                                 |
+-------------------------------------------------------------------------------+
```

Referential Integrity Errors

If you encounter a Referential Integrity (RI) error while updating or inserting a row, File-AID *for DB2* assists you in resolving the error without leaving your edit session.

Note: The following section describes a Referential Integrity error based on the following relationship.

The department table (DEPT) is the parent table and the employee table (EMP) is the dependent table. The DEPTNO column in the DEPT table is the primary key. The WORKDEPT column in the EMP table is a foreign key; it refers to the DEPTNO column.

Every WORKDEPT number in the employee table must exist in the department table. Table 6-2 shows the relationship.

Table 6-2. Sample Relationship

Parent Table	Primary Key	Dependent Table	Foreign Key
DEPT	DEPTNO	EMP	WORKDEPT

Figure 6-23. Inserting a Row into the Employee Table

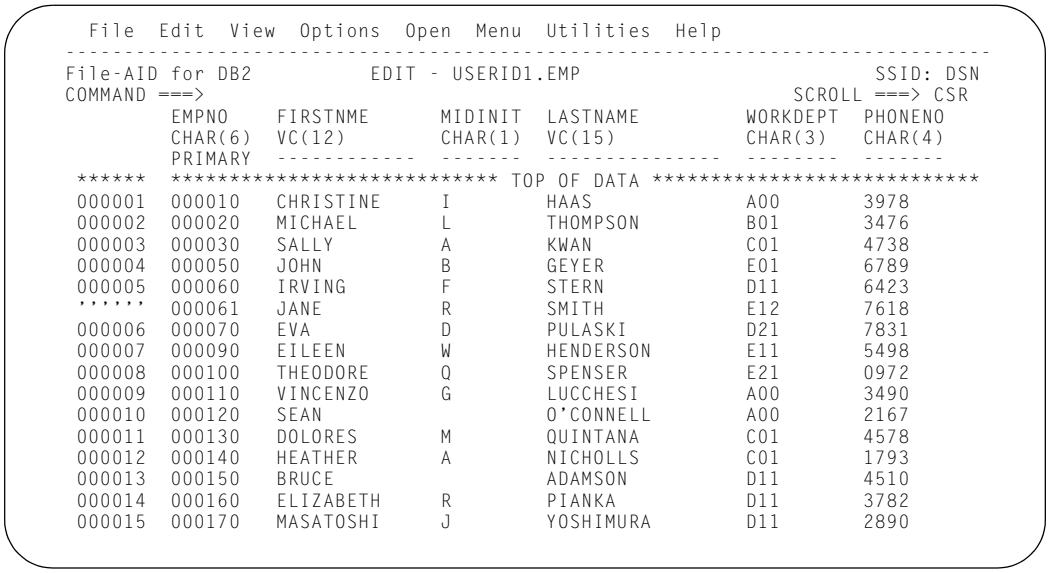


Figure 6-23 shows a new row being inserted into the employee table. The RI Relationship Error window (Figure 6-24) displays when File-AID *for DB2* encounters a -530 referential integrity error.

Figure 6-24. RI Relationship Error Window

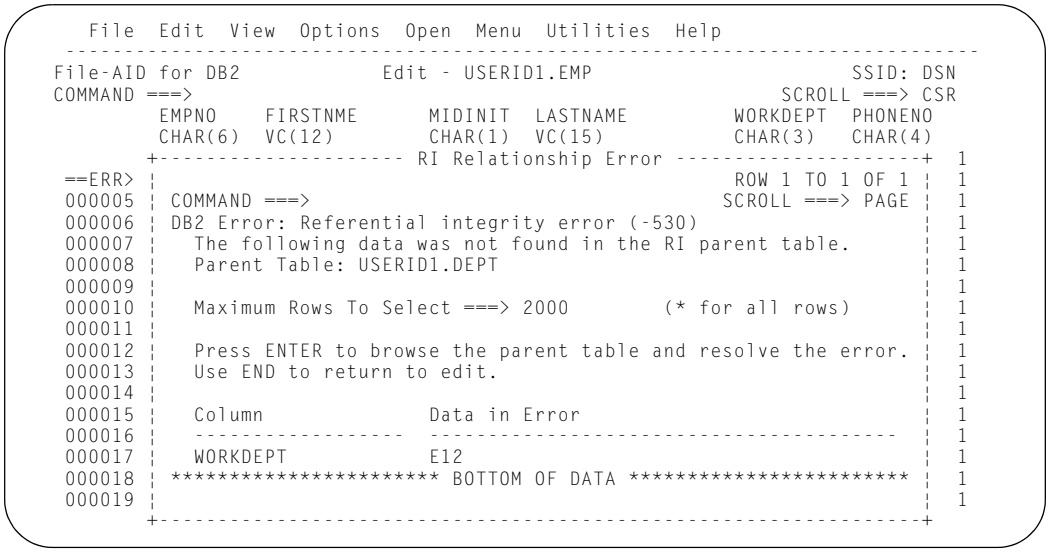


Figure 6-24 shows an SQL error because the WORKDEPT entered is not a department in the parent table, DEPT. The row in error moves to the top of the display and is flagged

with **==ERR>** in the line command area. The window displays the DB2 error message and code. You can also enter the maximum number of rows to select. The column and data in error are identified. You can issue the UP and DOWN commands to view up to 64 foreign key columns. The first 40 characters of the Data in Error field are displayed to identify the problem. Data beyond this limit is not shown.

To resolve the RI error, you can do one of the following:

- Press Enter on this screen to browse the parent table and select the correct data. See "RI Browse Window" for more information.
- Enter END to exit the window and return to the edit session. Then, you can edit the parent table by issuing the EDIT primary command and add the correct data. See "Editing Multiple Unrelated Tables" on page 6-22 for more information.

RI Browse Window

Once the RI Browse window displays, you can resolve the RI error by selecting a row from the parent table. When you select a row, the data in the primary key column (DEPTNO) replaces the data in the dependent table's foreign key column (WORKDEPT). See "Resolving RI Errors in Table Mode" on page 6-29 to resolve the error in table mode. See "Resolving RI Errors in Row Mode" on page 6-30 to resolve the error in row mode.

You can only issue standard browse mode commands from the RI Browse window. You cannot issue Related Table Edit commands (REDIT, RBROWSE, JUMP, MAX, and MIN).

Primary key columns are left justified and highlighted.

Resolving RI Errors in Table Mode

Use the **RI** line command to select the row containing the desired column data.

Figure 6-25. RI Browse Window — Using the RI Line Command

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
 --------------------------------------------------------------------------------
 File-AID for DB2          EDIT - USERID1.EMP                      SSID: DSN
 COMMAND ===>                                                   SCROLL ===> CSR
          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONENO
          CHAR(6)  VC(12)        CHAR(1)  VC(15)           CHAR(3)   CHAR(4)
          PRIMARY  ------------  -------  ---------------  --------  -------
+- File-AID for DB2 - RI Browse  USERID1.DEPT ------------------- SSID: DSN --+
| COMMAND ===>                                                  SCROLL ===> CSR |
| Use the 'RI' line command to select the row that resolves the RI error       |
|          DEPTNO   DEPTNAME                             MGRNO    ADMRDEPT     |
| 000001      A00  JIFFY COMPUTER SERVICE DIVISION.      000010       A00     |
| 000002      B01  PLANNING                              000020       A00     |
| 000003      C01  DATA PROCESSING                       000030       A00     |
| 000004      D01  DEVELOPMENT CENTER                    ------       A00     |
| 000005      D11  MANUFACTURING SYSTEMS                 000060       D01     |
| 000006      D21  ADMINISTRATION SYSTEMS                000070       D01     |
| 000007      E01  SUPPORT SERVICES                                   A00     |
| 000008      E11  OPERATIONS                                         E01     |
| RI  09      E21  SOFTWARE SUPPORT                                   E01     |
| 000010      F22  BRANCH OFFICE F2                                   E01     |
| 000011      G22  BRANCH OFFICE G2                                   E01     |
+------------------------------------------------------------------------------+
```

Figure 6-25 shows the user selecting department number E21 by typing RI in the line number area. Press Enter to automatically replace the data in error, E12, with the new work department, E21. Figure 6-26 shows the RI error resolved.

Figure 6-26. RI Error Resolved — Table Mode

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2          EDIT - USERID1.EMP                     SSID: DSN
COMMAND ===>                                                  SCROLL ===> CSR
         EMPNO    FIRSTNME      MIDINIT  LASTNAME        WORKDEPT  PHONENO
         CHAR(6)  VC(12)        CHAR(1)  VC(15)          CHAR(3)   CHAR(4)
         PRIMARY  ------------  -------  ---------------  --------  -------
 000006  000061   JANE          R        SMITH           E21       7618
 000007  000070   EVA           D        PULASKI         A00       7831
 000008  000090   EILEEN        W        HENDERSON       E11       5498
 000009  000100   THEODORE      Q        SPENSER         E21       0972
 000010  000110   VINCENZO      G        LUCCHESI        A00       3490
 000010  000120   SEAN                   O'CONNELL       A00       2167
 000011  000130   DOLORES       M        QUINTANA        C01       4578
 000012  000140   HEATHER       A        NICHOLLS        C01       1793
 000013  000150   BRUCE                  ADAMSON         D11       4510
 000014  000160   ELIZABETH     R        PIANKA          D11       3782
 000016  000170   MASATOSHI     J        QUINTANA        C01       4578
 000017  000180   MARILYN       S        SCOUTTEN        D11       1682
 000018  000190   PETER         N        ROSE            D11       3578
 000019  000200   ELIZABETH     R        PIANKA          D11       3782
 000020  000210   MASATOSHI     J        YOSHIMURA       D11       2890
 000021  000220   MARILYN       S        SCOUTTEN        D11       1682
```

Once you have resolved the referential integrity error, you can continue your edit session.

Resolving RI Errors in Row Mode

Use the **RI** primary command to select the row containing the desired column data.

Figure 6-27. RI Browse Window — Entering the RI Primary Command

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2          Edit - USERID1.EMP                     SSID: DSN
COMMAND ===>                                                  SCROLL ===> CSR
Row 1
    COLUMN NAME        TYPE(LEN)  KEY            COLUMN VALUE
--- ------------------ ---------- ---- ----+---10----+---20----+---30----+---40
+- File-AID for DB2 - RI Browse  USERID1.DEPT -------------------- SSID: DSN --+
| COMMAND ===> RI                                               SCROLL ===> CSR |
| Use the 'RI' primary command to select the row that resolves the RI error    |
|     COLUMN NAME        TYPE(LEN)  KEY  COLUMN VALUE          ROW 9           |
| --- ------------------ ---------- ---- ----+---10----+---20----+---30----+---|
| 001 DEPTNO             CHAR(3)    PRIM E21                                   |
| 002 DEPTNAME           VC(36)          SOFTWARE SUPPORT                     |
| 003 MGRNO              CHAR(6)         000010                               |
| 004 ADMRDEPT           CHAR(3)         E21                                  |
| 005 LOCATION           CHAR(16)                                             |
| ****************************** BOTTOM OF DATA ******************************|
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
| F22=PREV  F23=NEXT                                                           |
+------------------------------------------------------------------------------+
```

Figure 6-27 shows the user selecting department number E21 by typing RI in the COMMAND field. Press Enter to automatically replace the data in error, E12, with the new work department, E21. Figure 6-28 shows the RI error resolved.

Figure 6-28. RI Error Resolved — Row Mode

```
   File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2          Edit - USERID1.EMP                      SSID: DSN
COMMAND ===>                                                   SCROLL ===> CSR
Row 1
    COLUMN NAME        TYPE(LEN)  KEY            COLUMN VALUE
--- ------------------ ---------- ---- ----+---10----+---20----+---30----+---40
005 WORKDEPT           CHAR(3)         E21
006 PHONENO            CHAR(4)         3978
007 HIREDATE           DATE            1965-01-01
008 JOB                CHAR(8)         PRES
009 EDLEVEL            SMALLINT        18
010 SEX                CHAR(1)         F
011 BIRTHDATE          DATE            1933-08-14
012 SALARY             DEC(9,2)        9999999.00
013 BONUS              DEC(9,2)        1000.00
014 COMM               DEC(9,2)        4220.00
******************************* BOTTOM OF DATA ********************************



F10=PREV  F11=NEXT
```

Once you have resolved the referential integrity error, you can continue your edit session.

Resolving Multiple Errors

If multiple RI errors are flagged, File-AID *for DB2* displays the RI Relationship Error window (Figure 6-24 on page 6-28) for every error encountered. That is, after resolving the first error, it reappears whenever you attempt to scroll, enter a command, or press Enter. To resolve the error, you can:

- Press Enter to display the RI Browse window and issue the RI line or primary command.
- Enter END to return to the edit session and type over the data.
- Enter END to return to the edit session and issue a CHANGE or CANCEL command.

Creating an Audit Trail Dataset

An Audit Trail dataset contains a record of all successful modifications to DB2 table data, including inserts, deletes, updates, commits, and rollbacks. The report displays the before and after image of the data. A sample Audit Trail report displays in Figure 6-31 on page 6-33.

The user's ability to generate an audit trail is largely determined by decisions which were made when File-AID *for DB2* was installed. If your site chose to never produce an audit trail, the Create Audit Trail field (Figure 6-29) is not present on the Edit Options screen. If your site chose to always produce an audit trail, the field is present with a prefilled value of **YES**. You cannot change this value. The last option is to let the user, along with an audit trail exit routine, choose to produce an audit trail. In this case, the Create Audit Trail field is present on the Edit Options screen and the user can enter either **YES** or **NO**. If you enter **YES**, an audit trail is generated. If you enter **NO**, an audit trail is not generated unless an audit trail exit routine determines that one is mandatory for the selected object.

Figure 6-29. Create Audit Trail Field on Edit Options Screen

```
Specify Edit Options:
   Initial Display Mode   ===> 1           (1 - Multiple rows per screen,
                                            2 - One row per screen)
   Lock Table during Edit ===> NO          (Yes or No)
   Use Uncommitted Read  ~===> NO          (Yes or No)
   Maximum Rows to Select ===> 2000        (* for all rows)
   Create Audit Trail     ===> NO          (Yes or No)
   Relationship File   ===> 'FD.RDX.RELATE'
```

SMF Audit Trail Option

If your site chooses to direct File-AID *for DB2* audit trail records to the SMF log, File-AID *for DB2* creates an audit record each time you browse or edit an object and modify table data, then writes it to the SMF log. In addition, the Create Audit Trail field does not appear on the Edit Options screen, however, the message "Audit Trail Activated" displays at the start of the browse/edit session. Furthermore, "Audit Trail Print Facility" on page 6-32, and Option 5, Print Audit Trail will not be available. See the *File-AID for DB2 Installation Guide* for information on how to print File-AID *for DB2*'s audit trail records from the SMF log or how to modify the audit trail exit routine.

Audit Trail Print Facility

If you requested an audit trail report and made modifications to your data, the Audit Trail Print Facility window (Figure 6-30 on page 6-32) displays when you exit from your edit session. If you want to print your audit trail, enter the required information and either **SUB** to submit the job, or **JCL** to edit the JCL, on the command line. If you enter **JCL**, you must still enter **SUB** on the command line to submit the job. Press END to save the audit trail dataset and exit the print facility without printing the dataset. You can print an audit trail dataset at a later time through Option 5, Print. See Chapter 17, "Print Table Data or Audit Trail" for more information.

Figure 6-30. Audit Trail Print Facility screen

```
+--------------- File-AID for DB2 - Audit Trail Print Facility ---------------+
| COMMAND ===>                                                                |
|                                                                             |
| Specify Printer Characteristics:                                            |
|  Printer Destination ===> PRINT1                 Lines Per Page ===> 55     |
|  Sysout Class        ===> G                                                 |
|                                                                             |
| Specify Audit Trail Options:                                                |
|  Dataset Name: 'USERID1.FADB2.D950509.T141645'                              |
|                                                                             |
|  Description:  USERID2.EMP has been edited_____________________________     |
| ____________________________________________________________________________|
| ____________________________________________________________________________|
| ____________________________________________________________________________|
|                                                                             |
| JOB Card Information:                                                       |
| ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----  |
| //USERID1A                                                                  |
| //*                                                                         |
| //*                                                                         |
| //*                                                                         |
| Use SUB to submit the batch job; JCL to edit the generated JCL              |
| Use END to save audit trail file and exit                                   |
+-----------------------------------------------------------------------------+
```

Specify Printer Characteristics Area

Printer Destination

Enter the printer destination for the audit trail report.

Lines Per Page

Enter the number of lines to print per page.

Sysout Class

Enter the sysout class for the audit trail report.

Specify Audit Trail Options Area

Dataset Name

Specifies the audit trail dataset name. This name defaults to your most current audit trail dataset. You may not override the name.

Description

Specifies comments that are printed on the audit trail report. Up to four lines of comments may be entered.

Job Card Information

Enter a valid job card. See User Parms, "Option 3, Batch Options" on page 5-6 for entering a default job card.

Note: Whenever a character field contains a non-displayable character the hexadecimal representation for that field is always printed on the Audit Trail Report.

Figure 6-31. Sample Audit Trail Report

```
OBJECT NAME : USERID.EMP                                  DATE - 02/08/95
OBJECT TYPE : TABLE                                       Time - 09:21:23
LOCATION :
CREATOR  :                                  PRIMARY AUTHID   : USERID
NAME     :                                  SECONDARY AUTHID :
***
ACTION : UPDATE

COLUMN NAME          KY  TYPE         OLD DATA          NEW DATA
-------------------  --  -----------  ----------------  ------------------
EMPNO                PK  CH(6)        000020
FIRSTNME                 VC(12)       MICHAEL
LASTNAME                 VC(15)       THOMPSON
PHONENO                  CH(4)        3476
HIREDATE                 DATE         1973-10-10
JOB                      CH(8)        MANAGER
                                      DCDCCCD           01234545
                                      4151759           49953369
SALARY                   DEC(9,2)     41250.00          45100.00
BONUS                    DEC(9,2)       800.00           1000.00
COMM                     DEC(9,2)      3300.00              0.00
***
ACTION : COMMIT                       DATE - 02/08/1995, TIME - 09:23:10
***
EDIT SESSION TERMINATED               DATE - 02/08/1995, TIME - 09:23:12

*** COMMENTS :
USERID.EMP has been edited
```

Audit Trail Dataset Usage and Maintenance

Each time an audit trail is generated, a new, variable blocked sequential dataset is created. Unless modified by the audit trail exit routine, the default audit trail dataset name is:

```
Creator ID.FADB2.Date Stamp.Time Stamp
```

An example dataset name is: USERID.FADB2.D95025.T1200000.

File-AID *for DB2* does not delete an existing audit trail dataset. It is therefore the user's responsibility to perform regular DASD management.

While you are editing a table, a message may display advising you that the audit trail dataset is filled. If this happens, exit the edit session and begin a new session. File-AID *for DB2* closes the filled audit trail dataset and allocates a new, uniquely named dataset for the second edit session.

Once all editing for the specified object is completed, the audit trail datasets are merged if necessary. If this condition occurs frequently, ask the installer to increase the amount of space allocated to each audit trail dataset.

Chapter 7.
Related Table Browse and Edit

This chapter describes the File-AID *for DB2* Related Table Browse and Edit function. This function enables you to browse and edit tables related through DB2 referential integrity (RI) or through a user-defined application relationship (AR). Related Table Browse lets you display related tables, but you may not make any changes to the tables. Related Table Edit lets you display and edit related tables. Since the Browse and Edit functions are so similar, this chapter only shows the Related Table Edit version of the screens and windows. Unless otherwise stated, you can substitute the word browse for edit.

The following topics are presented in this chapter:

- UDT Columns, Triggers and LOB Columns
- Referential Integrity
- Application Relationship
- Relationship File
- Determining Relationships
- Accessing Related Table Edit
- Using Related Table Edit
- Creating a New Relationship
- Using Additional Windows
- Manipulating Windows
- Opening Additional Related Table Paths
- Updating Data
- Graphic Relationship Display

UDT Columns, Triggers and LOB Columns

File-AID *for DB2* Related Table Browse and Edit function handle UDT, Trigger and LOB as follows:

Tables with UDT columns:

Related edit or browse is supported for tables containing UDT columns.

Triggers:

When opening a table for related browse or edit, File-AID *for DB2* informs you that trigger actions are defined against this table.

Tables with LOB columns:

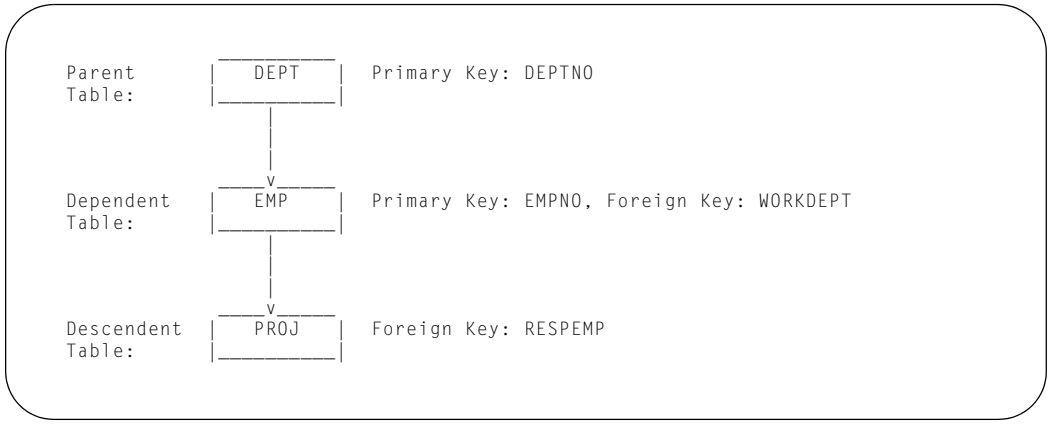
Related edit or browse is supported for tables containing LOB columns. LOB columns are not selected for edit or browse.

Referential Integrity

An RI relationship is one in which the dependent table rows contain a foreign key, the value of which is the primary key of the parent table. Figure 7-1 on page 7-2 shows an RI relationship. The value of every WORKDEPT in the EMP table must appear as a DEPTNO in the DEPT table. The value of every RESPEMP in the PROJ table must appear as a EMPNO in the EMP table. The rules that define RI relationships are enforced by DB2. The following definitions help to explain it in more detail:

- **Primary Key:** The column or set of columns that provide the unique identifiers of the rows.
- **Foreign Key:** A column or set of columns that refer to a primary key. Establishing a foreign key defines a relationship between two tables.
- **Referential Constraint:** The rule that a value of a foreign key must appear as a value of the primary key of some specific table.
- **Parent Table:** The table referenced in a referential constraint that contains the primary key.
- **Dependent Table:** The table referenced in a referential constraint that containing the foreign key.
- **Descendent Table:** A dependent of a dependent table.
- **Referential Integrity:** The enforcement of all referential constraints.

Figure 7-1. Referential Integrity



Application Relationship

An AR is similar to referential integrity in that references from one table to another are valid. However, the relationships are defined internally within the logic of application programs, rather than externally through DB2. Since application programs are not involved in an Edit session under File-AID *for DB2*, an external definition is required. You can meet this requirement by defining application relationships through File-AID *for DB2*'s Option 6, Relationships, or through Option 2, Edit. The relationship you define lets File-AID *for DB2* simultaneously access tables with application relationships.

Figure 7-2 shows an application relationship. Every CUSTNO in the ORDER table must be a CUSTNO in the CUSTOMER table. Every ORDERNO in the PARTS table must be a ORDERNO in the ORDER table.

You define the relationship to relate columns from one table to another: The CUSTOMER table is a parent to the ORDER table. The ORDER table is a dependent to the CUSTOMER table, and a parent to the PARTS table. The PARTS table is a dependent to the ORDER table and a descendent to the CUSTOMER.

Figure 7-2. Application Relationship

Unlike referential integrity, an application relationship has no external control to enforce the specified relationships. Only the application program can prevent a user from entering a foreign key in a dependent table that does not exist in the parent table. However, once an AR is defined to File-AID *for DB2* (see "Creating a New Relationship" on page 7-19), Related Table Edit can provide this control.

Relationship File

The relationship file is integral to the Related Table Edit function of File-AID *for DB2*. It acts as a repository of information about both DB2 RI and application relationships (AR).

Each relationship stored in the file is referred to as a *relationship definition*. File-AID *for DB2* uses the RI and AR definitions in the relationship file to determine table dependencies when initiating a Related Table Edit session.

Table 7-1 shows two sample relationship definitions. Only the fundamental record information is shown. Relationship **1** is a referential integrity relationship between the DEPT and EMP tables. Relationship **2** is an application relationship between the CUSTOMER and ORDER tables.

Table 7-1. Sample Relationship File

	Parent Table	Dependent Table	Type	Primary Key	Foreign Key
1	TEST.DEPT	TEST.EMP	RI	DEPTNO	WORKDEPT
2	PROD.CUSTOMER	PROD.ORDER	AR	CUSTNO	CUSTNO

Relationship File Compatibility Notes

1. **Relationship files can be shared** between File-AID *for DB2* and File-AID/RDX. Relationship files created or converted with File-AID/RDX Release 2.0, or higher, or File-AID *for DB2* Release 3.9.1, or higher, are Version 2 relationship files and are not downward compatible with earlier product releases.
2. **Version 1 Relationship files**, those created with earlier releases of either product, must be converted to a Version 2 relationship file. See Appendix C of the File-AID *for DB2* Installation Guide for more information on the creation and conversion procedures.
3. **DB2 Version 8 long name support** requires File-AID/RDX Release 4.3, or higher, or File-AID *for DB2* Release 4.8.1, or higher. Once a DB2 long name has been added to a

Version 2 relationship file, File-AID/RDX updates it to a Version 3 relationship file. Version 3 relationship files are not downward compatible with earlier product releases.

Relationship File Characteristics

A relationship file created with File-AID *for DB2* Release 3.9.0, or earlier, or File-AID/RDX Release 1.6.1, or earlier, has the following characteristics:

File Organization	VSAM-KSDS
Record Format	Variable
Logical Record Length	Maximum record size is 3900
Key Length	156

A relationship file created with File-AID for DB2 Release 3.9.1, or later, or File-AID/RDX Release 2.0, or later, can also include relationships with MVS datasets as parent and/or dependent objects. However, File-AID *for DB2*'s REDIT and RBROWSE functions only recognize relationships with DB2 parent and dependent objects. That relationship file has the following characteristics:

File Organization	VSAM-KSDS
Record Format	Variable
Logical Record Length	Maximum record size is 4765
Maximum columns	38
Key Length	74

Creating, Loading, and Maintaining the Relationship File

When File-AID *for DB2* was installed, the relationship file was created and loaded with referential integrity relationships from the DB2 catalog. A batch job was submitted to perform the load. The installer had the option of loading the relationship file with RI relationships from a single subsystem or from multiple subsystems. The job is also used to refresh the relationship file. For more information, see the *File-AID for DB2 Installation Guide* or the person who installed File-AID *for DB2*.

Once the relationship file is created, you can load DB2 RI and application relationships using the Relationship function, Option 6 from the Primary Option Menu, or using the Related Table Edit ADD function described in this chapter.

The relationship file should be refreshed regularly to maintain integrity between it and the DB2 catalog. Additionally, it should be backed-up and reorganized periodically.

Determining Relationships

To initiate a Related Table Edit session, File-AID *for DB2* must determine the relationships that exist for the table you are editing. It searches table SYSIBM.SYSRELS in the DB2 catalog to determine if referential integrity relationships exist. The relationship file is searched to determine what application relationships exist. If one or more relationships are found, the relationships appear in a selection list in a window. You select the desired relationship from the window or add a new one. If no relationships are found, you have the option of creating an AR.

See Figure 7-3 on page 7-6 for a diagram of how relationships are determined.

Using an RI Relationship

If you select an RI relationship from the window or if the only relationship defined is an RI, File-AID *for DB2* must retrieve the foreign key information before it can begin the session. File-AID *for DB2* uses the foreign key information from the relationship file if it is the most current information. If the DB2 catalog information is more current, File-AID *for DB2* retrieves the foreign key information from table SYSIBM.SYSFOREIGNKEYS.

If foreign key information must be retrieved from SYSIBM.SYSFOREIGNKEYS, the following message displays:

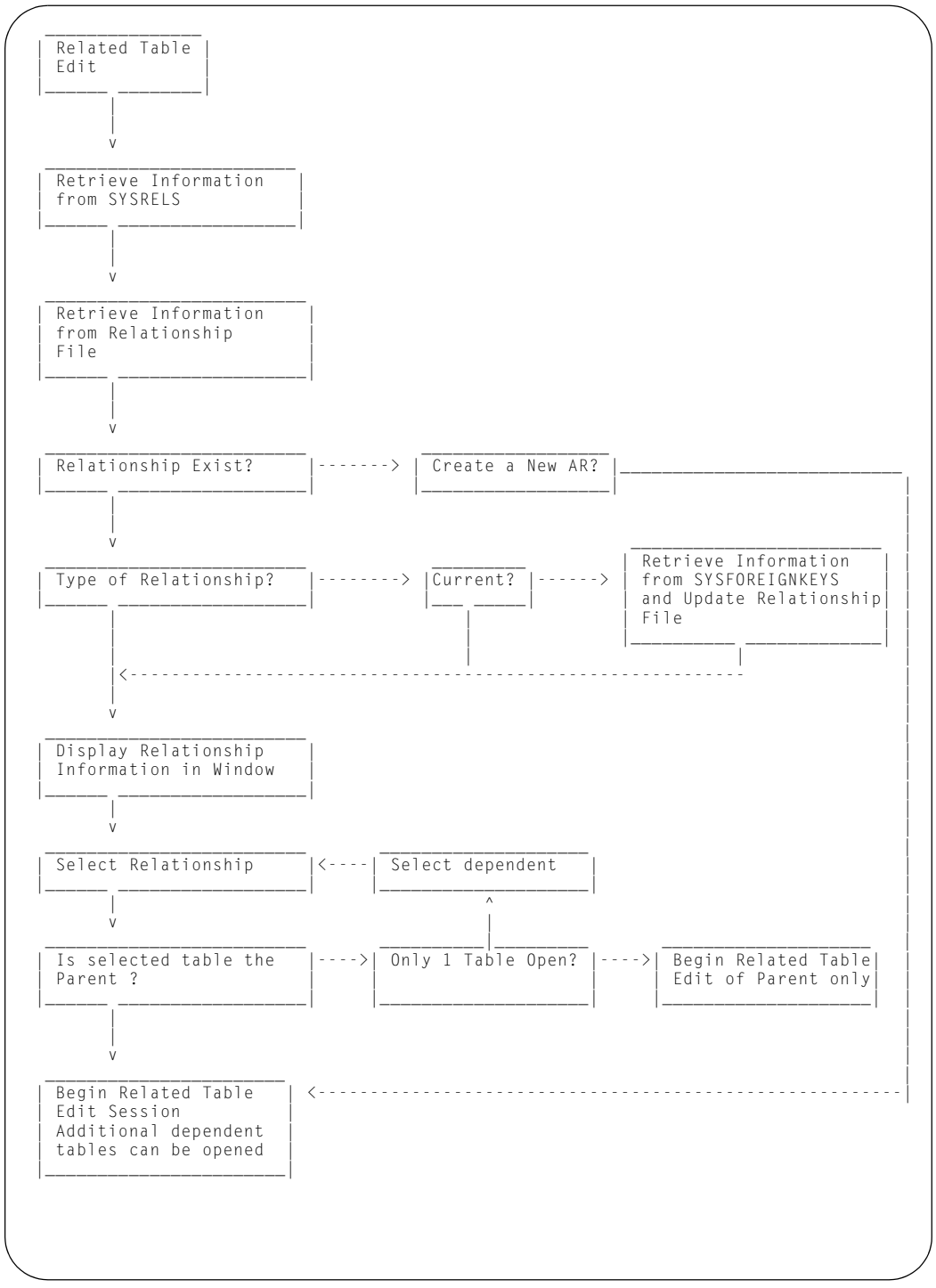
```
Referential Integrity information is being retrieved, please wait.

To avoid getting this message, a batch job may be run to update the File-AID for DB2
relationshipfile.SeetheFile-AIDforDB2InstallationGuideappendixonRefreshingthe
Relationship File.
```

Once the RI information is retrieved, File-AID *for DB2* automatically updates the relationship file with the new information. Updating the file enables you to keep the relationship file as current as possible and avoids future queries to SYSIBM.SYSFOREIGNKEYS. Your Related Table Edit session can now be initiated.

Using an Application Relationship

If you select an application relationship or if the only relationship defined is an AR, File-AID *for DB2* has all the required relationship information from the relationship file. A Related Table Edit session can begin without any further queries.

Figure 7-3. Diagram of Determining Relationships



Accessing Related Table Edit

To access Related Table Edit, you must first access the Edit function by selecting option **2** from the Primary Option Menu. To edit a related table, you must specify a relationship file at the bottom of the Edit Options screen (Figure 7-4).

Figure 7-4. Edit Options screen

```
File-AID for DB2 ----------------- Edit Options -------------------------------
COMMAND ===>                                                    SSID ===> DSN

Specify DB2 Table (* or pattern for list)
   Location ID ===>                        Optional:
   Creator     ===> TSOID01                   Database    ===>
   Table Name  ===> DEPT                      Tablespace  ===>

Specify Edit Template Information:
   Use Existing Criteria ===> NO       (Yes or No)
   Edit Criteria Dataset ===> SHOW.SQL
   Member                ===>          (Blank or Pattern for list)

Specify Edit Options:
   Initial Display Mode   ===> 1           (1 - Table, Multiple rows per screen
                                           2 - Row, One row per screen)
   Lock Table during Edit ===> NO          (Yes or No)
   Maximum Rows to Select ===> 2000        (* for all rows)
   Create Audit Trail     ===> NO          (Yes or No)
   Use Uncommitted Read   ===> NO          (Yes or No)
   Relationship File   ===> 'USERID1.KSDS.RELATION'
```

The **REDIT** primary command initiates a Related Table Edit session (Figure 7-5 on page 7-8). The **RBROWSE** primary command initiates a Related Table Browse session. They can be issued from either a Browse or Edit session.

Open Related Parent

If only one table is currently open you can either open a parent or a dependent of the current table with the REDIT and RBROWSE commands. After you open the parent table from a dependent table editing behavior is the same as multiple unrelated table editing (see "Editing Multiple Unrelated Tables" on page 6-22). However, you cannot open any additional tables until the parent table has been closed.

Open Related Dependent

After you open a dependent table you can open any descendent table (dependent of a dependent table) with the REDIT/RBROWSE commands. Issue REDIT or RBROWSE in the **last** opened window (window 5 of 5, for example).

Open Another Related Dependent Path

Issue REDIT or RBROWSE in **any but the last** window display (for example 2 of 5) to open another dependent path of the first opened (original parent) table. Use the PATH command to switch between paths (see also "PATH" on page 3-37).

The commands are cursor sensitive. To select a specific row, place the cursor on the desired line before pressing Enter. If a row is not selected or the REDIT or RBROWSE command is selected from the Open pull-down, the top row is the default selected row.

All primary commands that are valid in Edit are also valid in Related Table Edit. All primary commands valid in Browse are also valid in Related Table Browse.

Figure 7-5. REDIT Primary Command

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2          EDIT - TSOID01.DEPT                      SSID: DSN
COMMAND ===> REDIT                                              SCROLL ===> CSR
         DEPTNO   DEPTNAME                                MGRNO    ADMRDEPT
         CHAR(3)  VC(36)                                  CHAR(6)  CHAR(3)
         PRIMARY  ---------------------------------------  -------  --------
******   ***************************** TOP OF DATA ****************************
000001   A00      SPIFFY COMPUTER SERVICE DIVISION.      000010   A00
000002   B01      PLANNING                               000020   A00
000003   C01      DATA PROCESSING                        000030   A00
000004   D01      DEVELOPMENT CENTER                     ------   A00
000005   D11      MANUFACTURING SYSTEMS                  000060   D01
000006   D21      ADMINISTRATION SYSTEMS                 000070   D01
000007   E01      SUPPORT SERVICES                       000050   A00
000008   E11      OPERATIONS                             000090   E01
000009   E21      SOFTWARE SUPPORT                       000100   E01
000010   F22      BRANCH OFFICE F2                       ------   E01
000011   G22      BRANCH OFFICE G2                       ------   E01
000012   H22      BRANCH OFFICE H2                       ------   E01
000013   I22      BRANCH OFFICE I2                       ------   E01
000014   J22      BRANCH OFFICE J2                       ------   E01
000015   P01      CICS PLAYBACK                          ------   C01
******   ************************** BOTTOM OF DATA ****************************
```

Related Table Edit sessions are between *parent* and *dependent* tables.

Parent to Dependent

When you initiate a *parent* and *dependent* session, File-AID *for DB2* shows those rows in the *dependent* table that is related to the selected row in the *parent* table (Figure 7-6 on page 7-8).

Related table windows appear after the selected row from the parent table. Only one row from the *parent* table displays; it is protected. The last *dependent* table opened covers the remaining window display.

Figure 7-6. Related Table Edit Session - Dependent Table

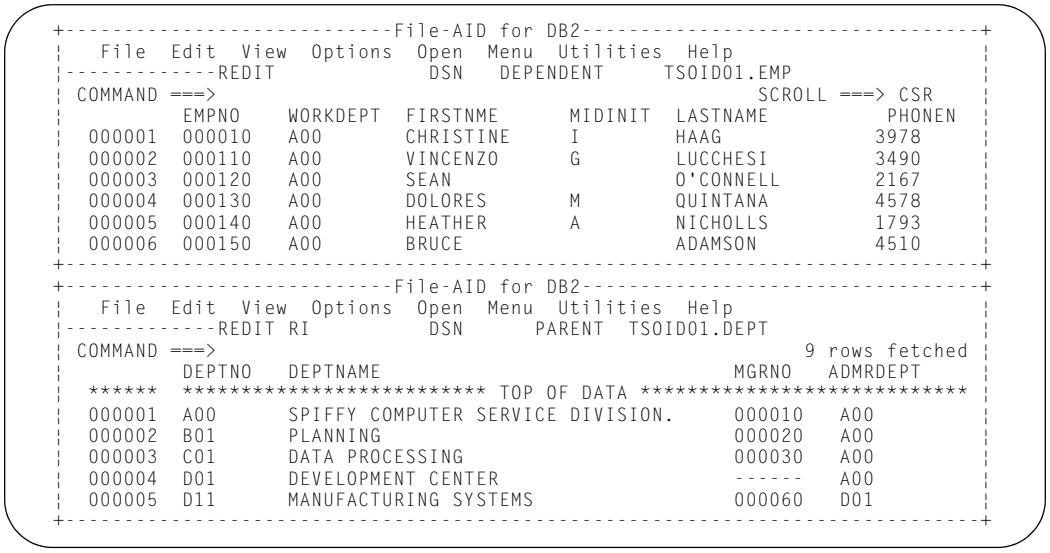
```
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|------------REDIT            DSN   1 OF 2     TSOID01.DEPT                  |
| COMMAND ===>                                          SCROLL ===> CSR       |
|          DEPTNO   DEPTNAME                          MGRNO   ADMRDEPT         |
|  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.  000010   A00           |
+------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|------------REDIT RI          DSN      2 OF 2  TSOID01.EMP                   |
| COMMAND ===>                                                 9 rows fetched  |
|          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME          PHONEN  |
|  000001  000010   A00       CHRISTINE     I        HAAG              3978    |
|  000002  000110   A00       VINCENZO      G        LUCCHESI          3490    |
|  000003  000120   A00       SEAN                   O'CONNELL         2167    |
|  000004  000130   A00       DOLORES       M        QUINTANA          4578    |
|  000005  000140   A00       HEATHER       A        NICHOLLS          1793    |
|  000006  000150   A00       BRUCE                  ADAMSON           4510    |
|  000007  000160   A00       ELIZABETH     R        PIANKA            3782    |
|  000008  000170   A00       MASATOSHI     J        YOSHIMURA         2890    |
|  000009  000180   A00       MARILYN       S        SCOUTTEN          1682    |
|  ******  ************************ BOTTOM OF DATA ************************    |
|                                                                              |
+------------------------------------------------------------------------------+
```

Dependent to Parent

When you initiate a *dependent to parent* session, File-AID *for DB2* displays the row in the *parent* table that is the parent to the selected row in the *dependent* table (Figure 7-7) unless you modified the selection template to select a different row.

Remember: Editing behavior is the same as for multiple unrelated table editing, so, for example, synchronized scrolling is not active.

Figure 7-7. Related Table Edit Session - Parent Table



Editing Requirements - Related Dependent Tables

You can only edit the *related rows* of a dependent table. For example, the second window in Figure 7-6 shows only those rows in the employee table that are related to the selected row in the department table; that is, employees that work in department A00. To edit any other rows in the employee table, you must scroll the parent table to select another row. See “Scrolling Windows” on page 7-32 for more information.

You can only edit one table at a time. The table that can be edited is the *active table* in the *active window*. Active windows are highlighted. Non-active windows are protected and not highlighted. You can move between tables using the JUMP primary command, but you cannot move the cursor manually to another table. See “Navigating Through Multiple Windows” on page 7-31 for more information on using the JUMP command. You can move between paths using the PATH primary command. See “Switching To Other Paths” on page 7-38 and “PATH” on page 3-37 for more information on using the PATH command.

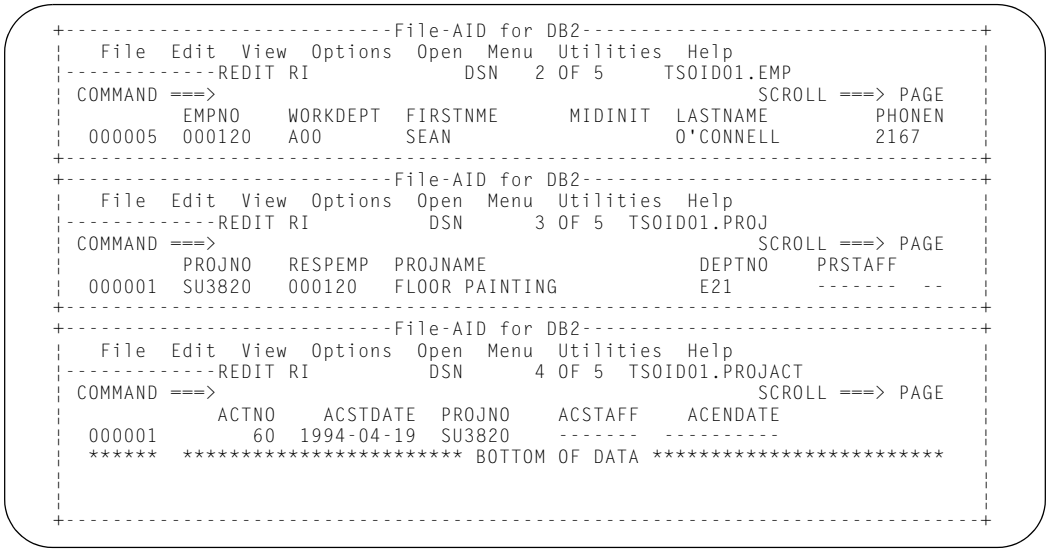
Window Display

An unlimited number of tables within up to 16 paths may be opened during an edit session by repeatedly issuing the REDIT command. A maximum of three windows can appear at one time. When the screen is full and you open another table, the table at the top of the display rolls off of the display. Each table moves up to make room for the next table.

Figure 7-8 on page 7-10 shows a sample Related Table Edit session with five open tables; three of them are shown.

Note: After opening the related parent table with REDIT, no other windows can be opened until the parent table has been closed.

Figure 7-8. Related Table Edit Session with Dependent Tables



The window title bar displays the following information:

- Product name.
- Product mode: REDIT or RBROWSE.
- Type of relationship: AR (Application Relationship) or RI (Referential Integrity).
- Fully qualified table name: Location.creator.table.
- Path number and total number of opened paths. Displays only when more than one path is open and only in the first window of each path. For example: **PATH 2 OF 6**, the 2 indicates that it is the second path opened and the 6 indicates that six paths are currently open.
- Window number and total number of windows open, within the current path. For example: If a window displays **2 OF 6**, the 2 indicates that it is the second window opened and the 6 indicates that six windows are open in the current path.
- DB2 subsystem identification.

Only column headings display in Related table windows. The second and third heading lines (column type, column length, key descriptions, and underscores) shown in standard Edit no longer appear.

When viewing an active window:

- Primary and foreign keys appear first and are protected.
- Column headings for non-keyed columns are blue.
- Column headings for primary keys are highlighted.
- Column headings for foreign keys are yellow.

Table Requirements

The initial table from which you initiate a Related Table Edit session is referred to as the driving table. The driving table must be a table, view, local alias or synonym. It cannot be a remote alias.

Dependent tables can be a table or view. You cannot edit related synonyms, or aliases.

Use of Views

File-AID *for DB2* supports views of tables in the Related Edit/Browse function. If you specify a view as the *driving table*, File-AID *for DB2* lists relationships defined for the view, not the underlying base table. As in previous releases, you can also change dependent table names to view names on the Related Table List enabling you to edit/browse dependent views instead of the listed tables.

Note: In releases prior to Release 3.9, when you specified a view as the driving table, File-AID *for DB2* used the relationships defined for the view's base table as File-AID *for DB2* did not support defining relationships for views. Starting with Release 3.9 you receive the message "No relationships exist for table: *Viewname*" unless you already defined an AR relationship for the view. Now define application relationships (see "Application Relationships (AR)" on page 18-2) for the View based on the direct dependencies of the view's base table. For example, in releases prior to 3.9 the driving table/view DSN8410.VDEPT used the relationships defined for base table DSN8410.DEPT. To achieve similar related edit/browse results in Release 3.9 and later, you just need to define AR relationships (Option 6.3) from view VDEPT to the direct dependents of the view's base table DEPT.

Base table DSN8410.DEPT has these relationships:

Table 7-2. Base Table Relationship

Parent			Dependent	
Table	**Column**		**Table**	**Column**
DEPT	DEPTNO	==>	EMP	WORKDEPT
DEPT	DEPTNO	==>	DEPT	ADMRDEPT
DEPT	DEPTNO	==>	PROJ	DEPTNO
EMP	EMPNO	==>	DEPT	MGRNO

Now define these relationships for the view VDEPT:

Table 7-3. View Relationship

Parent			Dependent	
Table	**Column**		**Table**	**Column**
VDEPT	DEPTNO	==>	EMP	WORKDEPT
VDEPT	DEPTNO	==>	VDEPT	ADMRDEPT
VDEPT	DEPTNO	==>	PROJ	DEPTNO
EMP	EMPNO	==>	VDEPT	MGRNO

Then, when you enter VDEPT as the driving table for related edit/browse, File-AID *for DB2* will list all relations for view VDEPT and other associated relationships treating the view as a separate object from the base table.

Using Related Table Edit

When you issue the REDIT command, File-AID *for DB2* searches the DB2 catalog and the relationship file to determine what tables are related to the driving table. "Determining

Relationships" on page 7-4 describes this process in detail. The screen or window that File-AID *for DB2* then displays is based on what relationships are determined.

This section explains the following possible scenarios:

- A one-to-one or one-to-many table relationship exists. At least one relationship definition exists for the table you are editing. You are prompted with a list of tables from which you select one of the dependent tables. See "One-to-One or One-to-Many Table Relationship" on page 7-12 for more information.
- You select a parent table of the driving table. The parent row of the driving table's current row and additional rows of the parent table appear. You cannot open any other tables as long as the parent table is open (REDIT/RBROWSE are disallowed). See "Selecting a Parent Relationship" on page 7-14 for more information.
- Model relationships exist. No relationship definitions exist for the table you are editing; however, relationships exist for the same table name with different creators. You can select one of these relationships as a model and use the relationship to initiate a related edit session. When you use a model relationship, the relationship is not saved for your creator ID; it is only temporary. See "Creating a New Relationship" on page 7-19 for information on adding a new relationship. See "Model Relationship" on page 7-16 for more information on using a model relationship.
- No relationships exist. No relationship definitions exist for the table you are editing. You can dynamically create an application relationship from the edit session with the driving table as the parent. Once created, you can edit the related dependent table. See "No Relationships" on page 7-18 for more information.

  If a relationship is found but there are no rows in the dependent table that relate to the selected row in the parent, an insert line displays in the dependent table. The foreign key field is filled with the parent table's primary key value.

In the next four sections, a sample relationship file that supports each scenario is shown in tabular format.

One-to-One or One-to-Many Table Relationship

A one-to-one relationship exists when there is only one relationship defined for the table you are editing. A one-to-many table relationship is one where multiple relationships exist for the table you are editing. Table 7-4 shows a relationship file that supports a one-to-many relationship.

Table 7-4. Relationship File — One-to-Many Relationship

	Parent Table	Dependent Table	Type	Primary Key	Foreign Key
1	TSOID01.DEPT	TSOID99.EMP	RI	DEPTNO	WORKDEPT
2	TSOID01.EMP	TSOID99.DEPT	RI	EMPNO	MGRNO
3	TSOID01.EMP	TSOID01.EMPPROJACT	RI	EMPNO	EMPNO
4	TSOID01.EMP	TSOID01.PROJ	RI	EMPNO	RESPEMP

Figure 7-9. Enter REDIT Primary Command

```
 File  Edit  View  Options  Open  Menu  Utilities  Help
-------------------------------------------------------------------------------
File-AID for DB2            Edit - TSOID01.EMP                      SSID: DSN
COMMAND ===> REDIT                                          SCROLL ===> CSR
         EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONENO
         CHAR(6)  VC(12)        CHAR(1)  VC(15)           CHAR(3)   CHAR(4)
         PRIMARY  ------------  -------  ---------------  -----------------  -
 ******  **************************** TOP OF DATA ****************************
 000001  000010   CHRISTINA     M        HAAG             A00       3978     1
 000002  000012   CHRIS         I        STRETTER         A00       9978     1
 000003  000015   JOE           B        COOL             A00       7725     1
 000004  000020   MICHAEL       S        THOMPSON         B01       3476     1
 000005  000025   SUSAN         S        THOMPSON         B01       3476     1
 000006  000030   SALLY         b        KWAN             C01       4738     1
 000007  000050   PENDERS       B        JOHN             E01       6789     1
 000008  000060   IRVING        F        STERN            D11       6423     1
 000009  000070   EVA           D        PULASKI          D21       7831     1
 000010  000090   EILEEN        W        HENDERSON        E11       5498     1
 000011  000100   THEODORE      Q        SPENSER          E21       0972     1
 000012  000110   VINCENZO      G        LUCCHESI         A00       3490     1
 000013  000120   SEAN                   O'CONNELL        A00       2167     1
 000014  000130   DOLORES       M        QUINTANA         C01       4578     1
 000015  000140   HEATHER       A        NICHOLLS         C01       1793     1
 000016  000150   BRUCE                  ADAMSON          D11       4510     1
```

When you enter REDIT (Figure 7-9), File-AID *for DB2* determines what tables are related to TSOID01.EMP (see "Determining Relationships" on page 7-4.). The sample relationship file in Table 7-4 on page 7-12 shows that there are four relationships defined for TSOID01.EMP. The Related Table List window (Figure 7-10) displays.

Figure 7-10. Related Table List WIndow

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
- +-------------------------- Related Table List ---------------------------+
F |                                                        ROW 1 TO 4 OF 4 |
C | COMMAND ===>                                                           |
  |                                                                        |
  | A one to one or one to many relationship exists.                       |
  | Use ADD to create a new application relationship or select an existing  |
  | relationship definition.                                               |
  | To edit a view: overtype displayed table name with a valid single table |
  |  view name.                                                            |
  |                                                                        |
  | Line Commands:                                                         |
  |   S - Select table                                                     |
  |   I - Relationship Information                                         |
  |                                                                        |
  |   CMD  Creator  Table/View           Type                              |
  |   ---  -------- -------------------  --------------------------------  |
  |    _   TSOID01  DEPT                 RI - PARENT                       |  1
  |    _   TSOID01  DEPT                 RI - CHILD                        |  2
  |    _   TSOID01  EMPPROJACT           RI - CHILD                        |  3
  |    S   TSOID01  PROJ                 RI - CHILD                        |  4
  | *************************** BOTTOM OF DATA ****************************|
  |                                                                        |
  +------------------------------------------------------------------------+
```

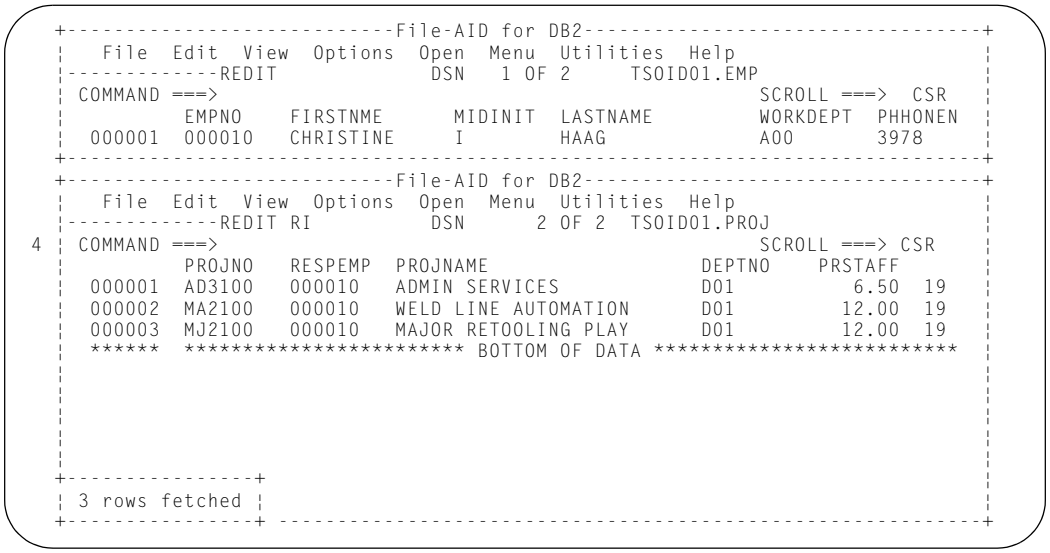
The Related Table List window displays the relationships in the bottom half of the window. You have the following options:

- Enter the **S** line command to select a table to edit.
- Enter the **I** line command to display information about the relationship. See "Relationship Information Window" on page 7-29.

- Enter the **ADD** primary command to create a new application relationship. The Column Relationships screen (Figure 7-19 on page 7-20) displays. See "Creating a New Relationship" on page 7-19.
- Enter **END** to exit the window and cancel the REDIT command.

Figure 7-10 on page 7-13 shows relationship **4** being selected. The relationship is between TSOID01.EMP and TSOID01.PROJ. Since the EMPNO of the selected row is 000010, File-AID *for DB2* displays a window of PROJ table rows where the RESPEMP is also equal to 000010. That is, you can view all the projects for employee number 000010 (Figure 7-11).

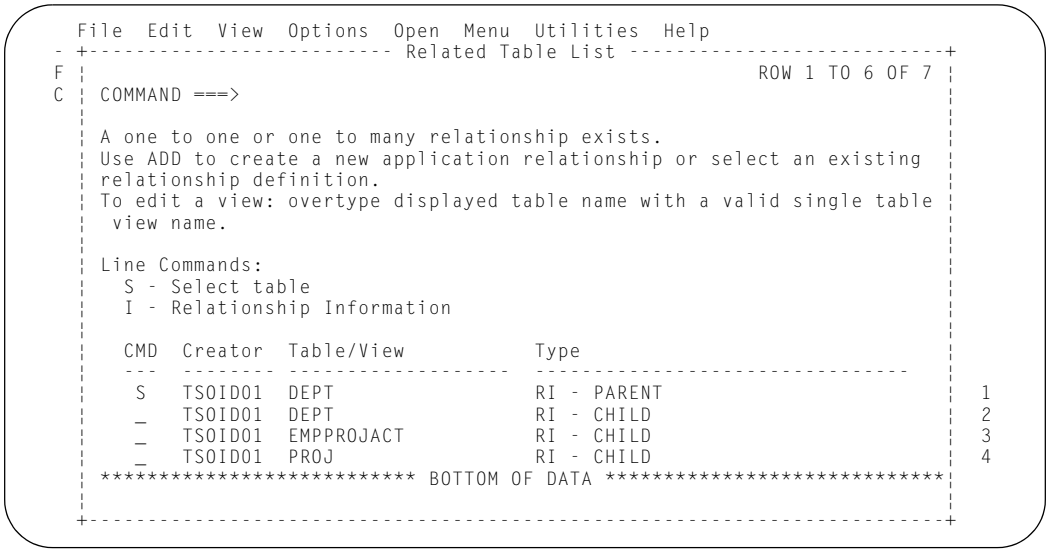
Figure 7-11. Related Table Edit Display — EMP to PROJ Table



Selecting a Parent Relationship

If you selected a PARENT relationship (Figure 7-12 on page 7-14), you enter a related edit/browse session of the parent.

Figure 7-12. Related Table List Window



After you select the RI - PARENT relationship for table DEPT and pressed ENTER, the Selection Template screen allows you to specify which rows and columns of the parent table should be displayed for editing. The key value of the first or current row of the driving table is prefilled in the template (Figure 7-13).

Figure 7-13. REDIT Parent Selection Template

```
+------------------ File-AID for DB2 - EDIT TEMPLATE - DSN --------------------+
|                                                                 Row 1 to 5 of 5 |
| COMMAND ===>                                                  SCROLL ===> PAGE |
|                                                                      SSID: DSN  |
| Table Name: TSOID01.DEPT                                                        |
|                                                                                 |
| Save Edit Criteria Information:                                                 |
|    Save Template Criteria On Execute  ===> NO       (Yes or No)                |
|    Edit Criteria Dataset  ===> EDIT.CRIT2                                       |
|    Member                 ===>                                                  |
|                                                                                 |
| SEL  COLUMN NAME        TYPE(LEN)       ORDER (A/D)  WHERE          More: >    |
| ---  ------------------ ----------------- --    -    ----+---10----+---20--   |
| ___  DEPTNO             CHAR(3)                       = 'A00'                   |
| ___  DEPTNAME           VARCHAR(36)                                             |
| ___  MGRNO              CHAR(6)                                                 |
| ___  ADMRDEPT           CHAR(3)                                                 |
| ___  LOCATION           CHAR(16)                                                |
| ***************************** Bottom of data ******************************** |
|                                                                                 |
|                                                                                 |
|                                                                                 |
|  F6=EXECUTE                                                                     |
+---------------------------------------------------------------------------------+
```

Modify the selection criteria as desired and then enter the EXECUTE command or press the associated PF key. See “Edit Selection Template” on page 6-9 for a detailed description of this screen.

In the shown example, the relationship is between TSOID01.EMP and TSOID01.DEPT. Since the current value for the dependent column WORKDEPTNO is A00 (Figure 7-9 on page 7-13), File-AID *for DB2* prefills the WHERE field in the DEPTNO column of the Selection Template for the DEPT table with the key value A00. You can now change the selection or accept it as is. Remove all selection criteria and you can edit the full table.

Figure 7-14 shows the Related Table Edit session for a parent table.

Figure 7-14. Related Table Edit Display — Dependent to Parent

```
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------REDIT             DSN   DEPENDENT     TSOID01.EMP               |
| COMMAND ===>                                        SCROLL ===>  CSR         |
|          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHONENO  |
|  000001  000010   CHRISTINA     M        HAAG             A00       3978     |
|  000002  000012   CHRIS         I        STRETTER         A00       9978     |
|  000003  000015   JOE           B        COOL             A00       7725     |
|  000004  000020   MICHAEL       S        THOMPSON         B01       3476     |
|  000005  000025   SUSAN         S        THOMPSON         B01       3476     |
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------REDIT RI          DSN      PARENT  TSOID01.DEPT                  |
1| COMMAND ===>                                        SCROLL ===> CSR          |
|          DEPTNO   DEPTNAME                          MGRNO   ADMRDEPT         |
|  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.  000010   A00           |
|  000002  B01      PLANNING                          000020   A00           |
|  000003  C01      DATA PROCESSING                   000030   A00           |
|  000004  D01      DEVELOPMENT CENTER                ------   A00           |
|  000005  D11      MANUFACTURING SYSTEMS             000060   D01           |
+-------------------------------------------------------------------------------+
```

Model Relationship

If no relationships exist for the table you are editing, but there are relationships defined for the same table name with a different creator, you can use one of these relationships as a model. Selecting a model enables you to initiate a Related Table Edit session using a relationship defined under a different creator. Table 7-5 shows a relationship file that supports this scenario.

Note: When you use a model relationship, the relationship is not saved for your creator ID. It is only temporary. See "Creating a New Relationship" on page 7-19 for information on adding a new relationship.

Table 7-5. Relationship File — Model Relationship

	Parent Table	Dependent Table	Type	Primary Key	Foreign Key
5	PROD.EMP	PROD.PROJ	RI	EMPNO	RESPEMP
6	TSOID23.EMP	TSOID23.EMPPROJACT	RI	EMPNO	EMPNO
7	TSOID57.EMP	TSOID57.PROJ	RI	EMPNO	RESPEMP

Figure 7-15. Related Table Edit Display — DEPT to EMP Tables

```
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|------------REDIT             DSN   1 OF 2     TSOID01.DEPT                  |
| COMMAND ===>                                              SCROLL ===> CSR    |
|          DEPTNO   DEPTNAME                              MGRNO   ADMRDEPT      |
|  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.     000010  A00           |
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|------------REDIT RI          DSN   2 OF 2     TSOID01.EMP                   |
| COMMAND ===> REDIT                                        SCROLL ===> CSR     |
|          EMPNO   WORKDEPT  FIRSTNME      MIDINIT  LASTNAME          PHONEN   |
|  000001  000010  A00       CHRISTINE     I        HAAG              3978     |
|  000002  000110  A00       VINCENZO      G        LUCCHESI          3490     |
|  000003  000120  A00       SEAN                   O'CONNELL         2167     |
|  000004  000130  A00       DOLORES       M        QUINTANA          4578     |
|  000005  000140  A00       HEATHER       A        NICHOLLS          1793     |
|  000006  000150  A00       BRUCE                  ADAMSON           4510     |
|  000007  000160  A00       ELIZABETH     R        PIANKA            3782     |
|  000008  000170  A00       MASATOSHI     J        YOSHIMURA         2890     |
|  000009  000180  A00       MARILYN       S        SCOUTTEN          1682     |
|  ******  ************************* BOTTOM OF DATA *************************   |
|                                                                               |
+-------------------------------------------------------------------------------+
```

When you enter REDIT (Figure 7-15), File-AID *for DB2* determines what tables are related to TSOID01.EMP. (See "Determining Relationships" on page 7-4.) The sample Relationship File (Table 7-5 on page 7-16) shows that no RI or AR definition exists for TSOID01.EMP. However, there are existing definitions under different creators, PROD **5**, TSOID23 **6**, and TSOID57 **7** that can be used as models.

The Relationship Listing window (Figure 7-16) displays.

Figure 7-16. Relationship Listing Window

```
+---------- File-AID for DB2 - REDIT - TSOID01.DEPT - 1 OF 2 - DSN -----------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                     |
| ------------------------------------------------------------------------    |
| C +------------------- Relationship Listing --------------------+ ==> CSR    |
|   |                                          ROW 1 TO 6 OF 23 | DMRDEPT    |
|   | COMMAND ===>                            SCROLL ===> PAGE  | 01         |
+-- | Multiple relationships exist for table: EMP               | ---------- +
+-- |                                                           | ---------- +
|   | Use the ADD primary command to create a new application   | ==> PAGE   |
| - | relationship or select an existing relationship definition. |   PHONEN   |
| C |                                                           |   0942     |
|   | Line Commands:                                            |   1682     |
|   |   S - Select relationship definition                      |   2890     |
|   |   I - Relationship Information                            |   ----     |
|   |                                                           |   6423     |
|   | Cmd  SSID Location         Creator  Table                 | ********   |
|   | ---  ---- ---------------- -------- ----------------      |            |
|   |  S   DSN  LOCA             PROD     EMP      5           |            |
|   |  _   DSN  LOCA             TSOID23  EMP      6           |            |
|   |  _   DSN  LOCB             TSOID57  EMP      7           |            |
|   |                                                           |            |
|   +-----------------------------------------------+           |            |
|   | Relationship not found for table: TSOID01.EMP |           |            |
+-- +-----------------------------------------------+ ----------+ ---------- +
```

The Relationship Listing window displays the model relationships, PROD.EMP, TSOID23.EMP, and TSOID57.EMP in the bottom half of the window. You have the following options:

- Enter the **S** line command to select a relationship as a model for your Related Table Edit session.

- Enter the **I** line command to display information about the relationship. See "Relationship Detail Window" on page 7-28.
- Enter the **ADD** primary command to create a new application relationship. The Column Relationships screen (Figure 7-19 on page 7-20) displays. See "Creating a New Relationship" on page 7-19.
- Enter **END** to exit the window.

Figure 7-16 on page 7-17 shows relationship **5** being selected. After in the Related Table List screen this relationship between PROD.EMP and PROD.PROJ had been selected, a Related Table Edit session is initiated for the PROJ table (Figure 7-17).

Note: When you use a model relationship definition, File-AID *for DB2* uses the creator of the last opened table, not the creator of the model table, when opening tables for the Related Table Edit session. In this example, the last opened table is *TSOID01*.EMP. Therefore, the user is placed into edit for *TSOID01*.PROJ, not *PROD*.PROJ. If the table does not exist, the Table Not Found window displays. See "Specifying a New Creator" on page 7-27 for more information.

Figure 7-17. Related Table Edit Display — after Selecting a Model Relationship

```
    +----------------------------File-AID for DB2---------------------------------+
    |   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
    |-------------REDIT             DSN   1 OF 3     TSOID01.DEPT                 |
    | ---------------------------------------------------------------------------  |
    | COMMAND ===>                                              SCROLL ===> PAGE  |
    |          DEPTNO   DEPTNAME                              MGRNO    ADMRDEPT   |
    |  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.     000010   A00        |
    +-----------------------------------------------------------------------------+
    +----------------------------File-AID for DB2---------------------------------+
    |   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
    |-------------REDIT RI          DSN      2 OF 3  TSOID01.EMP                  |
    | COMMAND ===>                                              SCROLL ===> PAGE  |
    |          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME         PHONEN  |
    |  000001  000010   A00       CHRISTINE     I        HAAG             3978    |
    +-----------------------------------------------------------------------------+
    +----------------------------File-AID for DB2---------------------------------+
    |   File  Edit  View  Options  Open  Menu  Utilities  Help                    |
    |-------------REDIT RI          DSN      3 OF 3  TSOID01.PROJ                 |
  5 | COMMAND ===>                                              SCROLL ===> CSR   |
    |          PROJNO   RESPEMP  PROJNAME                  DEPTNO    PRSTAFF      |
    |  000001  AD3100   000010   ADMIN SERVICES            D01          6.50  19  |
    |  000002  MA2100   000010   WELD LINE AUTOMATION      D01         12.00  19  |
    +----------------+                                                            |
    | 3 rows fetched |                                                            |
```

Note: If you select an RI model relationship, File-AID *for DB2* updates the relationship file if the DB2 catalog information is more current than the information in the relationship file. See "Determining Relationships" on page 7-4 for more information.

No Relationships

If no relationships are defined for the table you are editing, and no relationships exist for the same table name with a different creator, you can dynamically create a new application relationship. Table 7-6 shows a relationship file that supports this scenario.

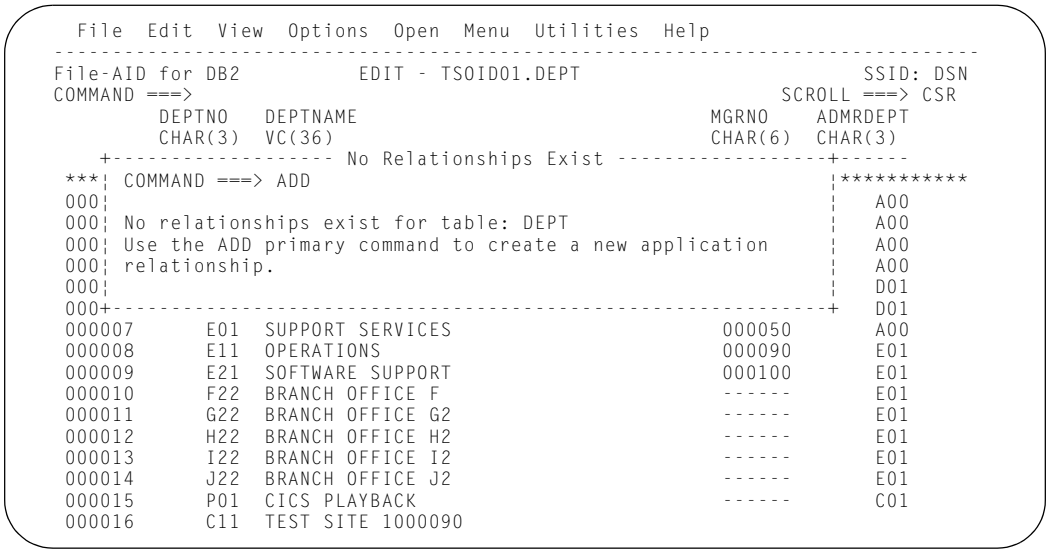
Table 7-6. Relationship File — No Relationship

Parent Table	Dependent Table	Type	Primary Key	Foreign Key
PROD.ORDER	PROD.PARTS	AR	ORDERNO	ORDERNO

When you enter REDIT (Figure 7-5 on page 7-8), File-AID *for DB2* determines what tables are related to TSOID01.DEPT. (See "Determining Relationships" on page 7-4.) The sample relationship file shown in Table 7-6 shows that no relationships exist for TSOID01.DEPT and there are no relationships for the same table name with a different creator.

The No Relationships Exist window (Figure 7-18) displays.

Figure 7-18. No Relationships Exist Window



You have the following options:

- Enter the **ADD** primary command to create a new application relationship. The Column Relationships screen (Figure 7-19 on page 7-20) displays. See "Creating a New Relationship" on page 7-19.
- Enter **END** to exit the window.

Creating a New Relationship

You can dynamically create a new application relationship in the following ways:

- Enter **REDIT ADD** in the Command field of the last opened window.
- Enter **ADD** in the Command field of the following windows:
  - No Relationships Exist window (Figure 7-18)
  - Relationship Listing window (Figure 7-16 on page 7-17)
  - Related Table List window (Figure 7-10 on page 7-13).

Note: To delete an application relationship, you must use the Maintain Relationship Definitions function, Option 6 from the Primary Option Menu. You can also add an application relationship with this function. See Chapter 18, "Relationship Facilitator" for more information.

After specifying that you want to add a new relationship, the Add Application Relationships screen screen displays (Figure 7-19 on page 7-20).

Figure 7-19. Add Application Relationships screen Invoked by ADD Command

```
File-AID for DB2 -------------  Add Application Relationships  ----------------
Command ===>

Specify Parent DB2 Table for Relationship Definition:
 Location     ===> LOCATION
 Creator      ===> TSOID01              (* or pattern for selection list)
 Table Name   ===> DEPT                 (* or pattern for selection list)

Specify Dependent DB2 Table for Relationship Definition:
 Location     ===>
 Creator      ===>                   (* or pattern for selection list)
 Table Name   ===>                   (* or pattern for selection list)



SSID          ===> DSN               (* or pattern for selection list)
Description   ===>
```

The creator and table name of the parent table is prefilled with values from the last opened table. Enter the dependent creator and table name for which you want to define a relationship. You can use a wildcard character in either field to display a selection list of tables. See "Wildcard Characters" on page 2-9 for more information. Enter a description of the relationship in the description field. A maximum of 128 characters is allowed.

Press Enter on this screen to bring up the Column Relationships screen (Figure 7-20).

Figure 7-20. Column Relationships screen

```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.DEPT - DB2
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
____ DEPTNO                          CHAR(3)
____ DEPTNAME                        VC(36)
____ MGRNO                           CHAR(6)
____ ADMRDEPT                        CHAR(3)
____ LOCATION                        CHAR(16)
Dependent: LOCATION.TSOID01.EMP - DB2
CMD  Dependent Columns               Type(Length)
---- ------------------------------- ------------
   1 EMPNO                           CHAR(6)
   2 FIRSTNME                        VC(12)
   3 MIDINIT                         CHAR(1)
   4 LASTNAME                        VC(15)
   5 WORKDEPT                        CHAR(3)
   6 PHONENO                         CHAR(4)
```

Building Column Relationships

To create application relationships, you create column relationships on the Column Relationships screen. An application relationship can consist of more than one column relationship. The fields in the lower portion of the Column Relationships screen comprise two lists. These lists contain names, data types, and lengths of columns from the parent and dependent tables specified in the upper portion of the screen. Use these lists to build column relationships.

The following procedure gives the steps to build column relationships.

1. Verify the parent and dependent table creator and table name fields. These fields are prefilled with information from the previous screen.
2. Verify or enter a description for the relationship.
3. Decide which parent and dependent columns you want to be related. This relationship should reflect the logic in the application program that uses these tables.
4. Enter the number from the COLUMN NUMBER field that corresponds to the dependent column in the CMD column next to the name of the parent column. One of the following must be true of any pair of corresponding parent and dependent columns:
   - Parent and dependent columns contain the same data type and are the same length.
   - One of the pair is INTEGER or SMALLINT data and the other is CHAR. In this case, the column lengths need not match.
   - Both parent and dependent columns contain character (CHAR or VCHAR) data or graphic (GRPH or, VGRPH) data. In this case, the columns need not be the same length. When you map a column containing character or graphic data to another column of the same data type, the Extended Column Relationship Definition pop-up window displays. The Extended Column Relationship Definition window enables you to create column relationships using partial columns. See "Extended Column Relationship Definition" on page 7-23 for more information about partial column relationships.

     Otherwise, File-AID *for DB2* displays an error message and does not establish the relationship.

     For example: Figure 7-21 shows how to relate the DEPT table to the EMP table. To relate the WORKDEPT column from the EMP table to the DEPTNO column from the DEPT enter a 5 in the CMD field next to the DEPTNO column. Both of these columns have the same column type and length: CHAR(3).

Figure 7-21. Column Relationships screen—Creating a new column relationship

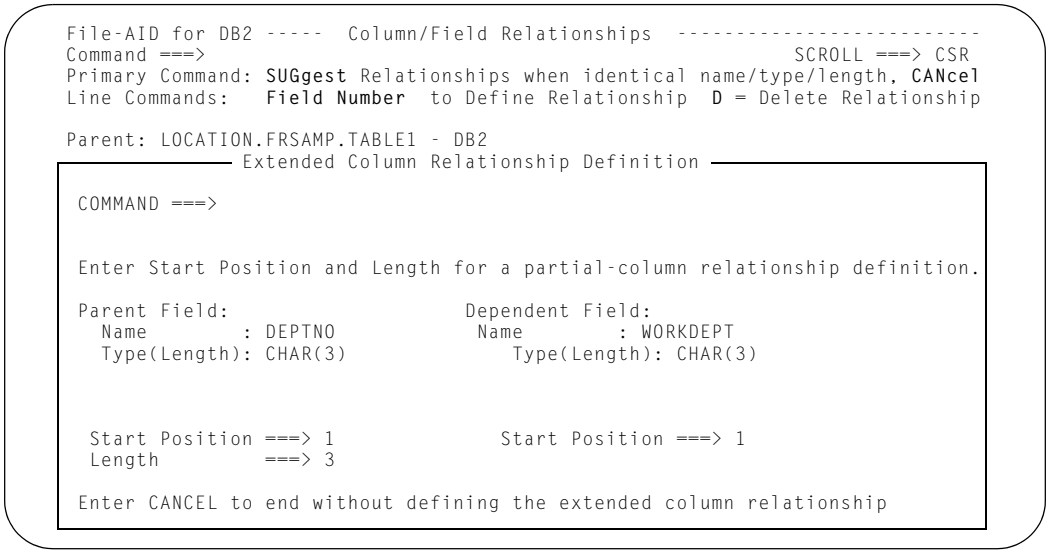
```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.DEPT - DB2
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
5___ DEPTNO                         CHAR(3)
____ DEPTNAME                        VC(36)
____ MGRNO                           CHAR(6)
____ ADMRDEPT                        CHAR(3)
____ LOCATION                        CHAR(16)
Dependent: LOCATION.TSOID01.EMP - DB2
CMD  Dependent Columns               Type(Length)
---- ------------------------------- ------------
   1 EMPNO                           CHAR(6)
   2 FIRSTNME                        VC(12)
   3 MIDINIT                         CHAR(1)
   4 LASTNAME                        VC(15)
   5 WORKDEPT                        CHAR(3)
   6 PHONENO                         CHAR(4)
```

5. Press Enter.

   As the parent and dependent columns both contain CHAR data, the Extended Column Relationship Definition pop-up window displays (Figure 7-22 on page 7-22).

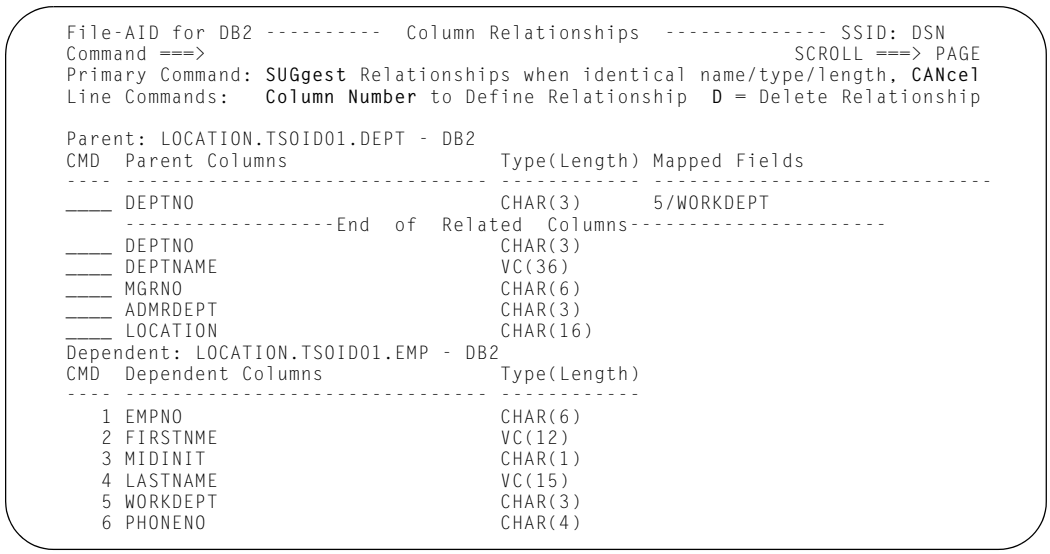
Press Enter to accept the full-column relationship or change the prefilled start positions and length values for a partial-column relationship.

Figure 7-22. Extended Column Relationship Definition Window



The Column Relationships screen now displays the defined related columns in the upper part of the columns listing. If one column contains INTEGER (or SMALLINT) data and the other contains CHAR data or both columns contain the same data type and are the same length, the column relationship is established. File-AID *for DB2* positions the names of the parent and dependent columns in the column relationship you just created to the top of their respective lists and above the dashed line (Figure 7-23 on page 7-22). In addition, the number that appeared in the COLUMN NUMBER column next to the dependent column is replaced by an arrow that points from the parent column to the dependent column and the remaining column numbers are renumbered. The message **Column Added** displays in the upper left corner.

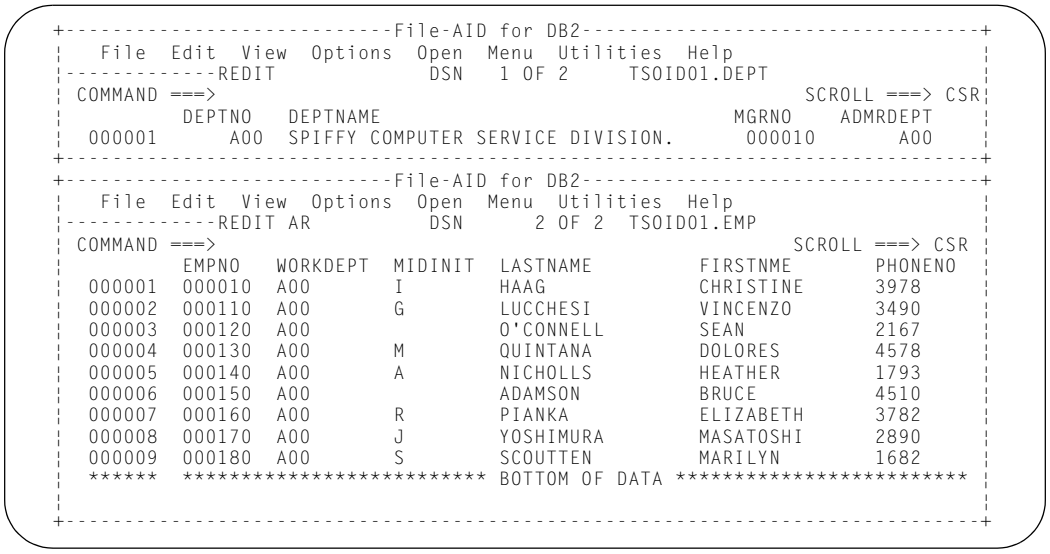
Figure 7-23. Column Relationships screen — New Column Relationship Established



6. Repeat steps 3-5 for all columns that you want to include in the application relationship.

Once the relationship is defined, enter **END** on this screen to begin the Related Table Edit session (Figure 7-24). The relationship definition is saved into the relationship file when you exit the Column Relationships screen.

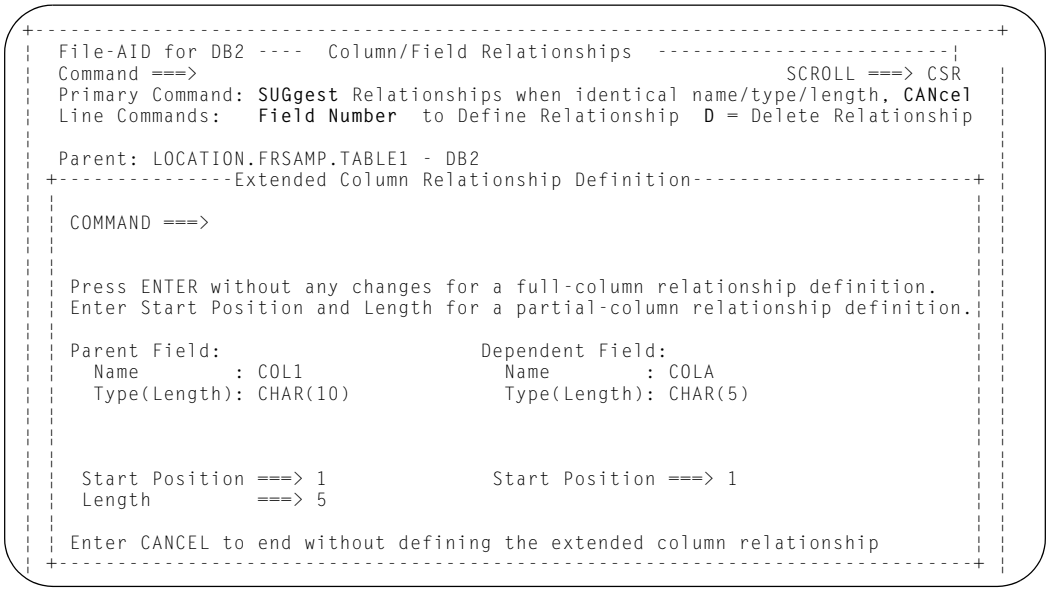
Figure 7-24. Related Table Edit Display — DEPT to EMP Tables



Extended Column Relationship Definition

The Extended Column Relationship Definition pop-up window (Figure 7-25) displays when you attempt to map two columns containing character (CHAR or VCHAR) or graphic (GRPH or VGRPH) data.

Figure 7-25. Extended Column Relationship Definition Window



Use this window to create partial-column relationships.

If the columns you want to map are the same length, the fields in this window are prefilled and the following message displays:

```
Press ENTER without any changes for a full-
column relationship definition
```

Complete this window and press Enter to proceed.

Parent Column

Name Displays the name of the parent column.

Type(Length) Displays the type of data contained in the parent column. The length of the parent column in parentheses follows the data type.

Start Position Specify the starting position of the string in the parent column that will comprise the column relationship.

Length Specify the length of the string contained in the parent column on which the column relationship is based.

Dependent Column

Name Displays the name of the dependent column.

Type(Length) Displays the type of data contained in the dependent column. The length of the dependent column in parentheses follows the data type.

Start Position Specify the starting position of the string in the dependent column that will comprise the column relationship. Length of the dependent column is determined by the length of the string in the corresponding parent column.

Extended Column Relationships Examples

The following describes several situations in which the Extended Column Relationship Definition window displays.

Example 1

Assume you have entered a column-number line command (as illustrated in Figure 7-26) to create a column relationship between two column with data types of CHAR and lengths of 6.

Figure 7-26. Column Relationships screen with column-number command

```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: Suggest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.CUSTOMER_TABLE - DB2
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
2___ CUSTOMER_NUMBER                 CHAR(6)
____ CUSTOMER_NAME                   CHAR(40)
____ ADDR_LINE_1                     CHAR(30)
____ ADDR_LINE_2                     CHAR(30)
____ CITY                            CHAR(25)
____ STATE                           CHAR(2)
Dependent: LOCATION.FRSAMP.ORDER_TABLE - DB2
CMD  Dependent Columns               Type(Length)
---- ------------------------------- ------------
   1 ORDER_NUMBER                    CHAR(6)
   2 CUST_NUM                        CHAR(6)
   3 ORD_TYPE                        CHAR(2)
   4 ORD_STAT                        CHAR(2)
   5 FIRST_ACT_DATE                  CHAR(10)
   6 LAST_ACT_DATE                   CHAR(10)
```

The Extended Column Relationship Definition pop-up window displays (Figure 7-27).

Figure 7-27. Extended Column Relationship Definition Window

```
File-AID for DB2 ------------- Column Relationships ----------------- SSID: DSN
COMMAND ===>                                                  SCROLL ===> CSR

Parent Table:                          Dependent Table:
  Creator    ===> FRSAMP                 Creator    ===> FRSAMP
  Table Name ===> TABLE1                 Table Name ===> TABLE2
+---------------Extended Column Relationship Definition-------------------------+
|                                                                               |
| COMMAND ===>                                                                  |
|                                                                               |
| Press ENTER without any changes for a full-column relationship definition.    |
| Enter Start Position and Length for a partial-column relationship definition. |
|                                                                               |
| Parent Field:                        Dependent Field:                         |
|   Name        : CUSTOMER_NUMBER       Name        : CUST_NUM                  |
|   Type(Length): CHAR(6)               Type(Length): CHAR(6)                   |
|                                                                               |
|   Start Position ===> 1               Start Position ===> 1                   |
|   Length         ===> 6                                                       |
|                                                                               |
| Use CANCEL to end without defining the extended column relationship           |
+-------------------------------------------------------------------------------+
```

Note: When you select parent and dependent columns with the same data type and length, the Extended Column Relationship Definition window is prefilled and the following message displays:

```
Press ENTER without any changes for a full-
column relationship definition
```

Example 2

Assume you have entered a column-number line command (Figure 7-28) to create a column relationship between two column with data types of CHAR and different lengths.

Figure 7-28. Column Relationships screen with column-number command

```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: Suggest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.CUSTOMER_TABLE - DB2
CMD  Parent Columns                 Type(Length) Mapped Fields
---- ------------------------------ ------------ ------------------------------
____ CUSTOMER_NUMBER                CHAR(6)
1___ CUSTOMER_NAME                  CHAR(40)
____ ADDR_LINE_1                    CHAR(30)
____ ADDR_LINE_2                    CHAR(30)
____ CITY                           CHAR(25)
____ STATE                          CHAR(2)
Dependent: LOCATION.FRSAMP.ORDER_TABLE - DB2
CMD  Dependent Columns              Type(Length)
---- ------------------------------ ------------
   1 ORDER_NUMBER                   CHAR(6)
   2 CUST_NUM                       CHAR(6)
   3 ORD_TYPE                       CHAR(2)
   4 ORD_STAT                       CHAR(2)
   5 FIRST_ACT_DATE                 CHAR(10)
   6 LAST_ACT_DATE                  CHAR(10)
```

The Extended Column Relationship Definition pop-up window displays (Figure 7-29).

Figure 7-29. Extended Column Relationship Definition Window

```
File-AID for DB2 ------------- Column Relationships ----------------- SSID: DSN
COMMAND ===>                                                  SCROLL ===> CSR

Parent Table:                          Dependent Table:
  Creator    ===> FRSAMP                 Creator    ===> FRSAMP
  Table Name ===> TABLE1                 Table Name ===> TABLE2
+---------------Extended Column Relationship Definition-------------------------+
|                                                                               |
| COMMAND ===>                                                                  |
|                                                                               |
| Press ENTER without any changes for a full-column relationship definition.    |
| Enter Start Position and Length for a partial-column relationship definition. |
|                                                                               |
| Parent Field:                    Dependent Field:                             |
|   Name         : CUSTOMER_NAME       Name        : ORDER_NUMBER               |
|   Type(Length): CHAR(40)             Type(Length): CHAR(6)                    |
|                                                                               |
|   Start Position ===> 1              Start Position ===> 5                    |
|   Length         ===> 2                                                       |
|                                                                               |
| Use CANCEL to end without defining the extended column relationship           |
+-------------------------------------------------------------------------------+
```

The values entered in the window (Figure 7-29) indicate that the first two characters in the parent column and the last two characters in the dependent column comprise the column relationship.

Using Additional Windows

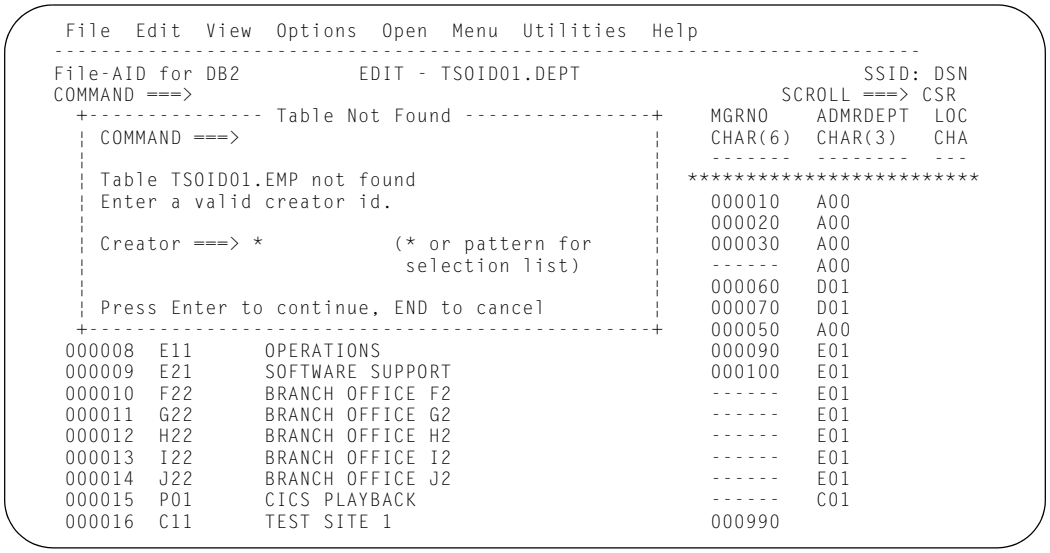
Additional windows in File-AID *for DB2* Related Table Edit provide further options and flexibility.

- **Table Not Found window:** enables you to enter a valid creator or wildcard if a table is not found during the Related Table Edit process.
- **Creator Selection window:** provides a list of creator identifications that have tables matching the selected dependent table.
- **Relationship Detail window:** provides parent and dependent table information.
- **Relationship Information window:** provides parent and dependent column information.

Specifying a New Creator

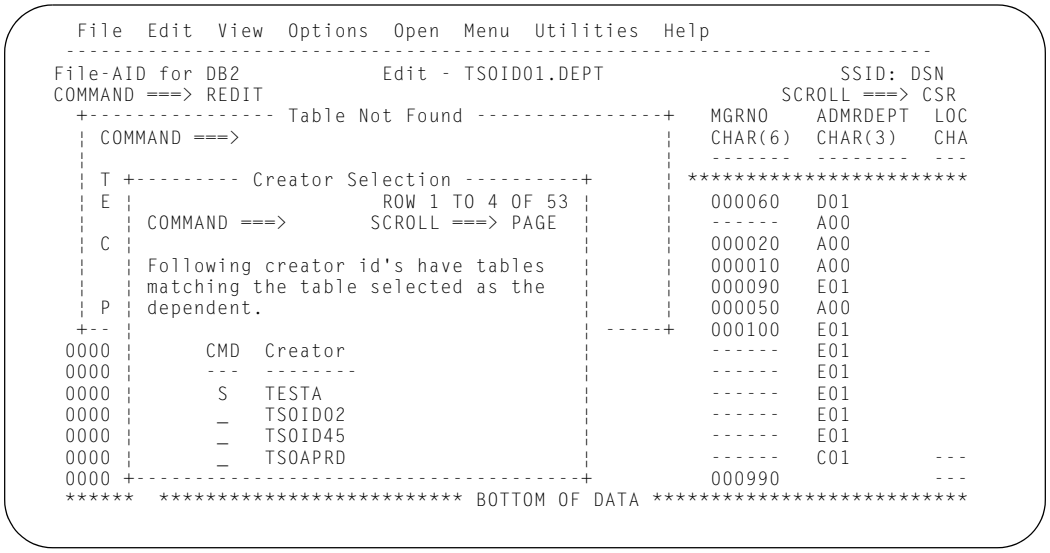
If the relationship you want to use references a table that is not found in the DB2 catalog, the Table Not Found window displays. It enables you to enter a valid creator for the selected table.

Figure 7-30. Table Not Found Window



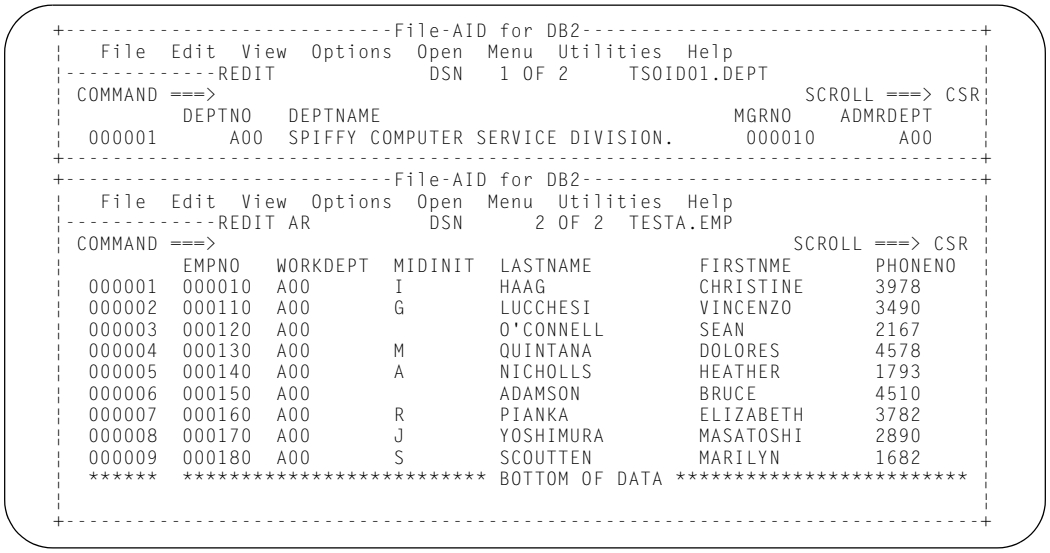
If you enter an asterisk (*****) in the CREATOR field, the Creator Selection window displays. Use the **S** line command to select the desired creator.

Figure 7-31. Creator Selection Window



Once you select a valid table, the Related Table Edit session begins (Figure 7-32 on page 7-28).

Figure 7-32. Related Table Edit Display After Selecting New Creator

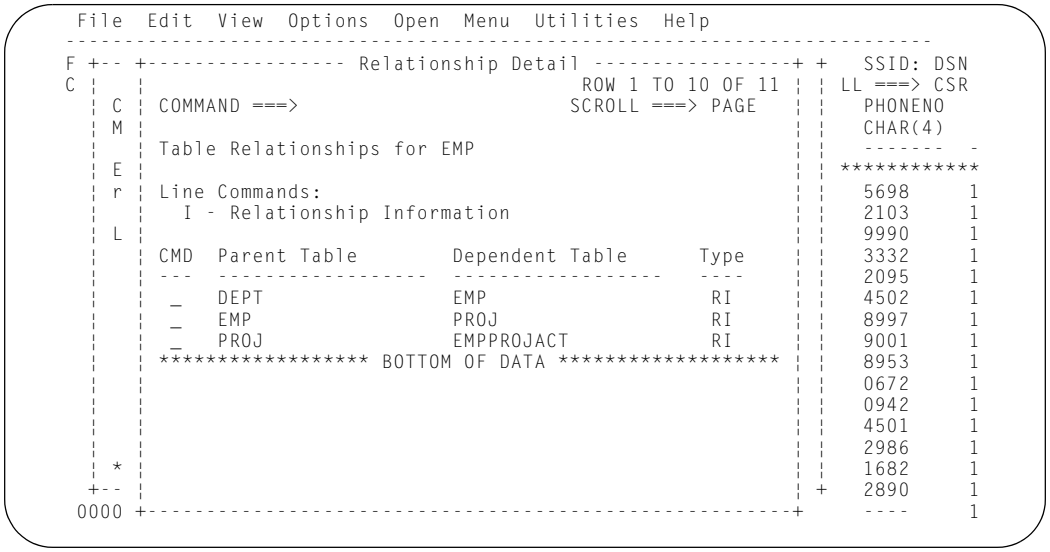


Relationship Detail Window

The Relationship Detail window (Figure 7-33) displays when you enter the **I** line command on the Relationship Listing window (Figure 7-16 on page 7-17).

This window lists all relationships associated with the selected table, including relationships involving descendent tables. The type of relationship, RI or AR, is also listed. Enter the **I** line command to bring up the Relationship Information window. See "Relationship Information Window" on page 7-29.

Figure 7-33. Relationship Detail Window



Relationship Information Window

The Relationship Information window (Figure 7-34 and Figure 7-35) displays when you enter the **I** line command in the Related Table List window (Figure 7-10 on page 7-13) or the Relationship Detail window (Figure 7-33).

The Relationship Information window shows parent and dependent column information. The related columns appear under their respective parent or dependent table. Multiple columns may appear. Use standard scrolling commands to view additional columns.

The window is slightly different depending on whether the relationship is application relationship(Figure 7-34) or an referential integrity (Figure 7-35).

Application Relationship (AR) Information

If the relationship information is an application relationship, AR displays first in the Description field. The bottom of the screen displays the date it was created, the user identification of the person who created the relationship, the date the relationship was last modified, and the user identification of the person who last modified the relationship.

Figure 7-34. AR Relationship Information

```
+-------------------------------------------------------------------------------+
| File-AID for DB2 -------  Relationship Information  ------- Row 1 to 1 of 2 |
| Command ===>                                              Scroll ===> PAGE |
|                                                                               |
|                                                                               |
|  Parent Name:    PROD.CUSTOMER                                 Type: DB2    |
|                                                                               |
|                                                                               |
|                                                                               |
|  Dependent Name: PROD.ORDER                                    Type: DB2    |
|                                                                               |
|                                                                               |
|                                                                               |
| Relationship Details:                                                         |
|  Description:    AR - Production Customer to Order Relationship               |
|  Status:                                                                      |
|  Created:        2005-04-26 16.04.26.450002                 User: TSOID99     |
|  Last Modified                                              User:            |
| Parent Columns  /Type(Length)             Dependent Columns  /Type(Length)    |
| --------------------------------------- ---------------------------------     |
| CUSTNO                                    CUSTNO                              |
|    CH(6)                                   CH(3)                              |
|  F1=HELP     F5=RFIND    F6=NEXT    F12=CANCEL                                |
+-------------------------------------------------------------------------------+
```

Referential Integrity (RI) Information

If the relationship information is based on referential integrity, RI displays first in the Description field. The bottom of the screen displays the date the relationship was loaded, and the user identification of the person who loaded the relationship.

Figure 7-35. RI Relationship Information Window

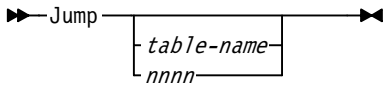
```
+-------------------------------------------------------------------------------+
| File-AID for DB2 -------  Relationship Information  ------- Row 1 to 1 of 2 |
| Command ===>                                              Scroll ===> PAGE |
|                                                                               |
|                                                                               |
|  Parent Name:    TSOID15.DEPT                                  Type: DB2    |
|                                                                               |
|                                                                               |
|                                                                               |
|  Dependent Name: TSOID15.EMP                                   Type: DB2    |
|                                                                               |
|                                                                               |
|                                                                               |
| Relationship Details:                                                         |
|  Description:    RI -  DEPTWRK                                                |
|  Status:                                                                      |
|  Created:        2004-07-20 12.48.38.018771                 User: TSOID15     |
|  Last Modifi|d   2004-07-20 12.48.38.018771                 User: TSOID15     |
| Parent Columns  /Type(Length)             Dependent Columns  /Type(Length)    |
| --------------------------------------- ---------------------------------     |
| DEPTNO                                    WORKDEPT                            |
|    CH(3)                                   CH(3)                              |
|  F1=HELP     F5=RFIND    F6=NEXT    F12=CANCEL                                |
+-------------------------------------------------------------------------------+
```

Manipulating Windows

Once a Related Table Edit session is initiated, manipulating the windows is simple using primary commands. Compuware recommends that you assign commonly used primary commands (JUMP, REDIT, ROW and TABLE) to a PF key to facilitate working with multiple windows, or select the commands from the pull-down menus.

Navigating Through Multiple Windows

File-AID *for DB2* allows an unlimited number of tables to be open during an edit session. However, only one table can be edited at a time. Use the **JUMP** command to move from window to window. Once you jump to a new window, it becomes active. You can only edit data in an active window.



Blank

Makes the next sequential table active. Wraps to the first table when the last table is reached.

table name

Makes the specified table active. You only need to prefix the table name with the creator if the creator uniquely identifies the table; for example, you are editing both TSOID01.EMP and TSOID99.EMP.

table number

Moves to the *nnnth* table.

Figure 7-37 shows a Related Table Edit session. The active window is window number 4. Enter **JUMP 2** to make window number 2 active and to allow editing of the EMP table.

Figure 7-36. JUMP Command used to Edit Table Number 2

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT            DSN   2 OF 4     TSOID01.EMP                   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME         PHONENO  ¦
¦  000001  000270   D21       MARIA         L        PEREZ            9001     ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT             DSN      3 OF 4  TSOID01.PROJ                  ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          PROJNO   RESPEMP  PROJNAME                  DEPTNO    PRSTAFF       ¦
¦  000001  AD3113   000270   ACCOUNT.PROGRAMMING       D21          2.00       ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT           DSN      4 OF 4  TSOID01.PROJACT                 ¦
¦ COMMAND ===> JUMP 2                                   SCROLL ===> CSR        ¦
¦             ACTNO    ACSTDATE  PROJNO    ACSTAFF    ACENDATE               ¦
¦  000001        60  1982-03-01  AD3113       0.75  1982-10-15               ¦
¦  000002        60  1982-04-01  AD3113       1.00  1982-09-01               ¦
¦  000003        60  1982-09-01  AD3113       0.25  1982-10-15               ¦
¦  000004        70  1982-06-01  AD3113       1.25  1982-12-15               ¦
+-------------------------------------------------------------------------------+
```

When you jump to a new window, it is highlighted and the previous window becomes protected and not highlighted.

Figure 7-37. JUMP Command — Window 2 Now Active

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT             DSN   2 OF 4     TSOID01.EMP                  ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME         PHONEN   ¦
¦  000001  000270   D21       MARIA         L        PEREZ            9001     ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT           DSN      3 OF 4  TSOID01.PROJ                    ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          PROJNO   RESPEMP  PROJNAME                  DEPTNO    PRSTAFF       ¦
¦  000001  AD3113   000270   ACCOUNT.PROGRAMMING       D21          2.00       ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT           DSN      4 OF 4  TSOID01.PROJACT                 ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦             ACTNO    ACSTDATE  PROJNO    ACSTAFF    ACENDATE               ¦
¦  000001        60  1982-03-01  AD3113       0.75  1982-10-15               ¦
¦  000002        60  1982-04-01  AD3113       1.00  1982-09-01               ¦
¦  000003        60  1982-09-01  AD3113       0.25  1982-10-15               ¦
¦  000004        70  1982-06-01  AD3113       1.25  1982-12-15               ¦
+-------------------------------------------------------------------------------+
```

Scrolling Windows

When you issue the UP or DOWN primary commands, or a FIND command that results in scrolling to a new row, all dependent tables are refetched so that the rows in the dependent tables relate to the new row in the parent table. If scrolling results in no rows in the dependent table relating to the parent table, an insert line displays. The foreign key field is filled with the primary key value from the parent table. You cannot insert data into a dependent table if its parent window is empty.

Figure 7-38. DOWN Primary Command

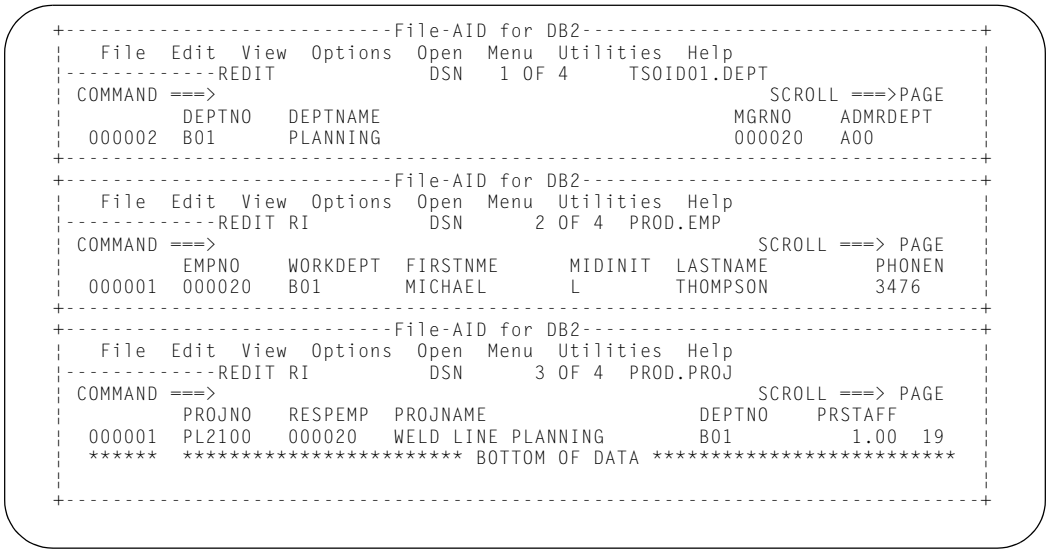
```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT             DSN   1 OF 4     PROD.DEPT                    ¦
¦ COMMAND ===> DOWN                                   SCROLL ===> PAGE         ¦
¦          DEPTNO   DEPTNAME                           MGRNO    ADMRDEPT      ¦
¦  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.   000010   A00           ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT RI          DSN      2 OF 4  PROD.EMP                      ¦
¦ COMMAND ===>                                          SCROLL ===> PAGE       ¦
¦          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME         PHONEN   ¦
¦  000001  000010   A00       CHRISTINE     I        HAAG             3978     ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦-------------REDIT RI          DSN      3 OF 4  PROD.PROJ                     ¦
¦ COMMAND ===>                                          SCROLL ===> PAGE       ¦
¦          PROJNO   RESPEMP  PROJNAME                  DEPTNO    PRSTAFF       ¦
¦  000001  AD3100   000010   ADMIN SERVICES            D01          6.50  19    ¦
¦  ******  ********************** BOTTOM OF DATA ************************       ¦
¦                                                                               ¦
+-------------------------------------------------------------------------------+
```

When you issue the DOWN command in window 1, the windows redisplay as follows:

- Window 1: The DEPT table display is scrolled one row. The new DEPTNO is B01.
- Window 2: The EMP rows are refetched where the WORKDEPT = B01. The new EMPNO is 000020.

- Window 3: The PROJ rows are refetched where the RESPEMP = 000020. The new PROJNO is PL2100.
- Window 4 (not currently displayed): The PROJACT rows are refetched where the PROJNO = PL2100.

Figure 7-39. New Display after Scrolling Down



Sizing Windows

Use the MAX primary command to enlarge a window to full screen size. Use the MIN primary command to reduce the window size and resume multiple table viewing. Once a table is maximized, all windows are maximized when using the JUMP command. Tables remain maximized until you issue the MIN command.

Figure 7-40. MAX Primary Command

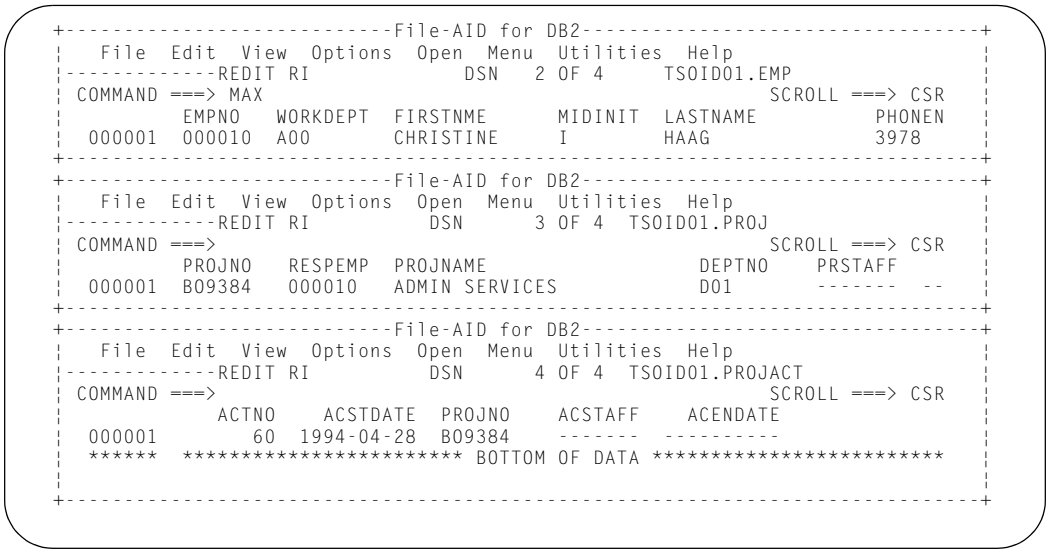
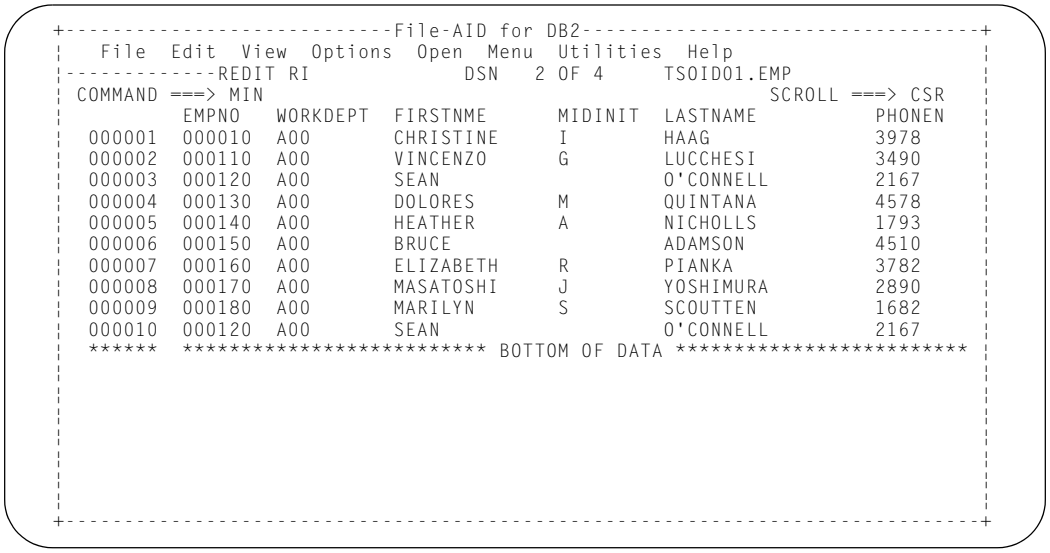


Figure 7-41. Related Table Edit DIsplay after the MAX Primary Command



Using Row Mode

Enter the ROW primary command to display the active window in row mode. For further information on editing in row mode, see “Row Mode” on page 6-17.

Figure 7-42. ROW Primary Command

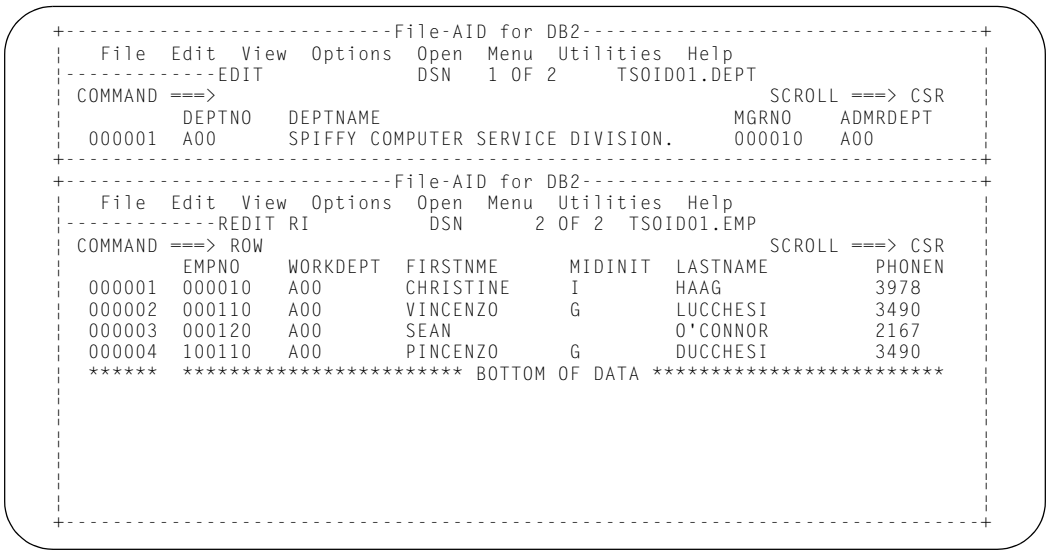


Figure 7-43. Window 2 in Row Mode

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------EDIT            DSN   1 OF 2     TSOID01.DEPT                   ¦
¦ COMMAND ===>                                              SCROLL ===> CSR    ¦
¦         DEPTNO   DEPTNAME                                 MGRNO    ADMRDEPT   ¦
¦  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.      000010   A00        ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT RI          DSN      2 OF 2  TSOID01.EMP                   ¦
¦ COMMAND ===>                                              SCROLL ===> CSR    ¦
¦     COLUMN NAME        TYPE(LEN)  KEY  COLUMN VALUE                          ¦
¦ 001 EMPNO              CHAR(6)    PRIM 000010                               ¦
¦ 002 WORKDEPT           CHAR(3)         A00                                  ¦
¦ 003 FIRSTNME           VC(12)          CHRISTINE                            ¦
¦ 004 MIDINIT            CHAR(1)         I                                    ¦
¦ 005 LASTNAME           VC(15)          HAAG                                 ¦
¦ 006 PHONENO            CHAR(4)         3978                                 ¦
¦ 007 HIREDATE           DATE            1965-01-01                           ¦
¦ 008 JOB                CHAR(8)         PRES                                 ¦
¦ 009 EDLEVEL            SMALLINT        18                                   ¦
¦ 010 SEX                CHAR(1)         F                                    ¦
¦ 011 BIRTHDATE          DATE            1933-08-14                           ¦
+-------------------------------------------------------------------------------+
```

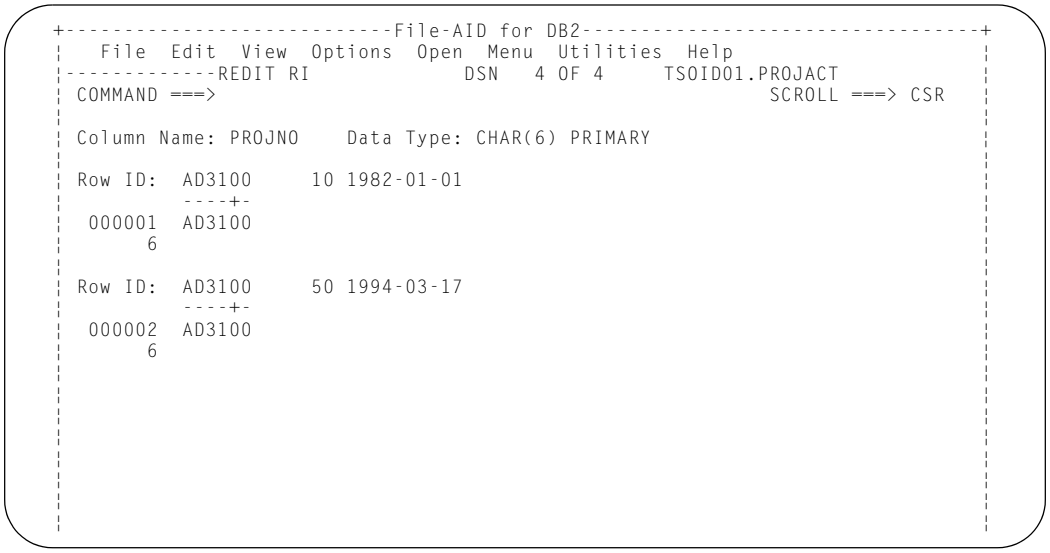
Using Zoom Mode

Enter the ZOOM primary command to display the active window in zoom mode. Zoom mode is always displayed in a maximized window. The MAX and MIN commands are not valid in zoom mode. To return to the original display, press END. For further information on editing in zoom mode, see "ZOOM" on page 3-60.

Figure 7-44. ZOOM Primary Command

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT RI             DSN   2 OF 4     TSOID01.EMP                ¦
¦ COMMAND ===>                                              SCROLL ===> CSR    ¦
¦          EMPNO   WORKDEPT  FIRSTNME      MIDINIT  LASTNAME          PHONENO  ¦
¦  000001  000010  A00       CHRISTINE     I        HAAG              3978     ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT RI          DSN      3 OF 4  TSOID01.PROJ                  ¦
¦ COMMAND ===>                                              SCROLL ===> CSR    ¦
¦          PROJNO   RESPEMP   PROJNAME                  DEPTNO   PRSTAFF      ¦
¦  000001  AD3100   000010    ADMIN SERVICES            D01         6.50  19    ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT RI          DSN      4 OF 4  TSOID01.PROJACT               ¦
¦ COMMAND ===> ZOOM PROJNO                                  SCROLL ===> CSR    ¦
¦            ACTNO    ACSTDATE  PROJNO    ACSTAFF    ACENDATE                ¦
¦  000001        10  1982-01-01  AD3100       0.50  1982-07-01               ¦
¦  000002        50  1994-03-17  AD3100    -------  ----------               ¦
¦  ******  ************************ BOTTOM OF DATA ************************    ¦
¦                                                                             ¦
+-------------------------------------------------------------------------------+
```

Figure 7-45. Maximized Zoom Mode



Opening Additional Related Table Paths

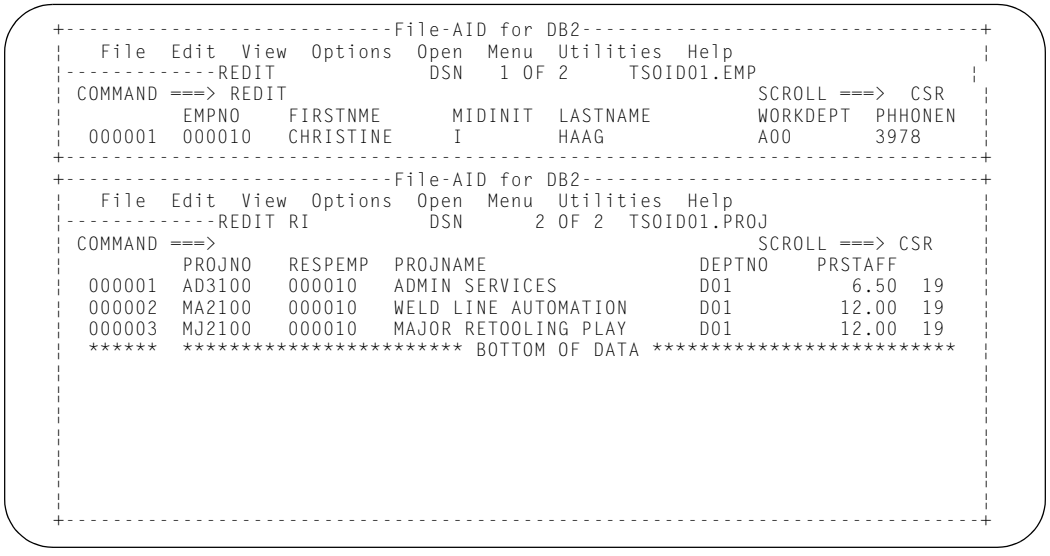
Once a Related Table Edit session of dependent tables is initiated, you can open additional related paths, all originating from the same original driving table. You can open up to 16 paths or until your memory limit is reached.

Note: Contact Compuware Technical Support if you need to increase the 16 paths limit.

Enter the REDIT command in any open window ***other than the last window***, e.g. in window **2 OF 3**, not **3 OF 3**.

For example, you have opened a related dependent table PROJ, jumped back to the driving table EMP and entered the REDIT command (Figure 7-46).

Figure 7-46. Related Table Edit Display — EMP to PROJ Table



The Related Table List window (Figure 7-47) displays.

Figure 7-47. Related Table List WIndow

```
+---------- File-AID for DB2 - REDIT - TSOID01.EMP - 1 OF 2 - DSN ------------+
|   +-------------------------- Related Table List ----------------------------+
| - |                                                              ROW 1 TO 4 OF 4 |
| C | COMMAND ===>                                                                 |
|   |                                                                              |
|   | A one to one or one to many relationship exists.                             |
+-- | Use ADD to create a new application relationship or select an existing       |
+-- | relationship definition.                                                     |
|   | To edit a view: overtype displayed table name with a valid single table      |
| - |  view name.                                                                  |
| C |                                                                              |
|   | Line Commands:                                                               |
|   |   S - Select table                                                           |
|   |   I - Relationship Information                                               |
|   |                                                                              |
|   |   CMD  Creator  Table/View           Type                                    |
|   |   ---  -------- -------------------  --------------------------------        |
|   |    _   TSOID01  DEPT                 RI - PARENT                          | 1
|   |    S   TSOID01  DEPT                 RI - CHILD                           | 2
|   |    _   TSOID01  EMPPROJACT           RI - CHILD                           | 3
|   |    _   TSOID01  PROJ                 RI - CHILD                           | 4
|   | ************************** BOTTOM OF DATA ******************************** |
|   |                                                                              |
+-- +------------------------------------------------------------------------------+
```

In the Related Table List window select one of the listed CHILD relationships or ADD a new relationship to open a new dependent path for the driving table EMP. Remember, you can't select a PARENT as you already have more than one table open.

The above example selects the dependent table DEPT 2 and opens it for a related table edit session (Figure 7-48 on page 7-37).

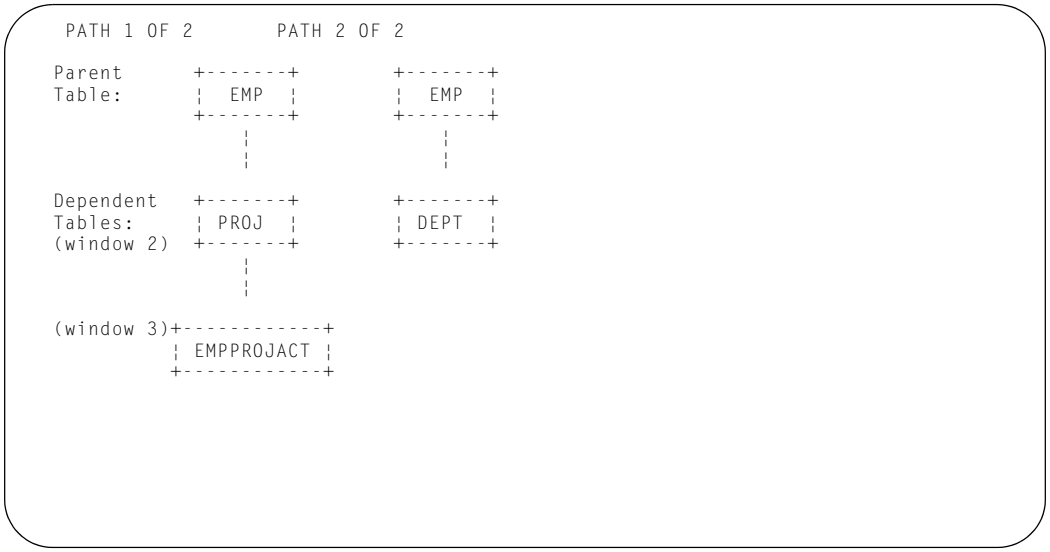
Figure 7-48. Related Table Edit — New Path EMP to DEPT

```
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------REDIT             DSN   PATH 2 OF 2     TSOID01.EMP             |
| COMMAND ===>                                         SCROLL ===>  CSR        |
|          EMPNO    FIRSTNME      MIDINIT  LASTNAME         WORKDEPT  PHHONEN  |
|  000001  000010   CHRISTINE     I        HAAG             A00       3978     |
+------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------REDIT RI          DSN      2 OF 2  TSOID01.DEPT                  |
2| COMMAND ===>                                          1 rows fetched       |
|          DEPTNO   MGRNO    DEPTNAME                        ADMRDEPT          |
|  ******  ************************ TOP OF DATA ***************************    |
|  000001  A00      000010   SPIFFY COMPUTER SERVICE DIV.     A00               |
|  ******  ********************** BOTTOM OF DATA ************************      |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
+------------------------------------------------------------------------------+
```

The display now shows the new dependent table DEPT (2) in the active window (**2 OF 2**) presenting only the dependent row(s) of the row displayed in the driving (parent) table. Notice that only the driving table includes which path is currently open (**PATH 2 OF 2**).

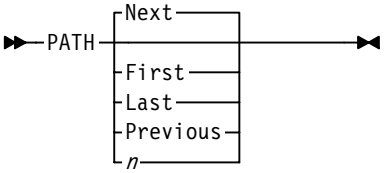
Now you have opened two parallel paths. Figure 7-49 on page 7-38 shows the paths' hierarchy.

Figure 7-49. Referential Integrity



Switching To Other Paths

File-AID *for DB2* allows up to 16 paths to be open during a related edit session. However, only one table can be edited at a time. Use the **PATH** command to move to another path. When switching to another path, File-AID *for DB2* makes the driving table window the active window (PATH 1 of 2). You can only edit data in an active window.



Use the **JUMP** command to move to other windows within the current path (see also "Navigating Through Multiple Windows" on page 7-31).

Blank or Next

Makes the next sequential path active. Wraps to the first path when the last path is reached.

n

Number - switches to the specified path number.

First

Switches to the first path (i.e. **PATH 1 OF 9**).

Last

Switches to the last path (i.e. **PATH 9 OF 9**).

Previous

Makes the previous sequential path active. Wraps to the last path when the first path is reached.

Figure 7-50 on page 7-39 shows a Related Table Edit session. The active path is PATH 4 and the active window is window number 3. Enter **PATH 2** to make path number 2 active.

Figure 7-50. PATH Command used to switch to Path Number 2

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT            DSN   PATH 4 OF 4     TSOID01.EMP               ¦
¦ COMMAND ===>                                        SCROLL ===> CSR           ¦
¦          EMPNO    WORKDEPT  FIRSTNME      MIDINIT  LASTNAME          PHONENO  ¦
¦  000001  000270   D21       MARIA         L        PEREZ             9001     ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT           DSN      2 OF 4  TSOID01.PROJ                     ¦
¦ COMMAND ===>                                          SCROLL ===> CSR       ¦
¦          PROJNO   RESPEMP  PROJNAME                  DEPTNO    PRSTAFF       ¦
¦  000001  AD3113   000270   ACCOUNT.PROGRAMMING       D21          2.00       ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT           DSN      3 OF 4  TSOID01.PROJACT                  ¦
¦ COMMAND ===> PATH 2                                   SCROLL ===> CSR       ¦
¦            ACTNO    ACSTDATE  PROJNO    ACSTAFF    ACENDATE                  ¦
¦  000001       60  1982-03-01  AD3113       0.75  1982-10-15                 ¦
¦  000002       60  1982-04-01  AD3113       1.00  1982-09-01                 ¦
¦  000003       60  1982-09-01  AD3113       0.25  1982-10-15                 ¦
¦  000004       70  1982-06-01  AD3113       1.25  1982-12-15                 ¦
+-------------------------------------------------------------------------------+
```

When you switch to a new path, you jump to window 1 of the path. It becomes highlighted and the other windows are protected.

Closing Individual Windows and Paths

Use the END command to close related editing windows and/or paths:

- END command issued in Windows greater than 3 (3 of 5) closes only this window and all its dependent windows.
- END command issued in Window 2 (2 of 5 in path 3, for example) closes the entire path.
- END command issued in Window 1 (1 of 5, for example) closes the entire related edit session.

Refer also to “Saving Changes” on page 7-39.

Updating Data

The following section pertains to Related Table Edit only. You cannot update data in Related Table Browse.

Saving Changes

When you issue the SAVE primary command, or the END primary command with AUTOSAVE set to ON, all changes to *all* tables are saved.

If you have the Auto Commit option set to ON, all changes to *all* tables are saved each time you press Enter.

When you issue the END primary command with AUTOSAVE set to OFF and Auto Commit set to OFF, the Edit End Confirmation window displays (Figure 6-12 on page 6-19). The window displays for each table in which there are pending changes, starting with the last opened table. The window asks if you want to continue processing the END command. If you enter YES, all changes made to the specified table are cancelled. If you enter NO, you are placed back into edit for the specified table. To save your changes, use the SAVE primary command.

When you issue the CANCEL primary command, all uncommitted changes made to the active table and its dependent tables are cancelled.

If you enter **=x** in any related table window, the entire edit session ends and you return to the Edit Options screen.

Inserting Data

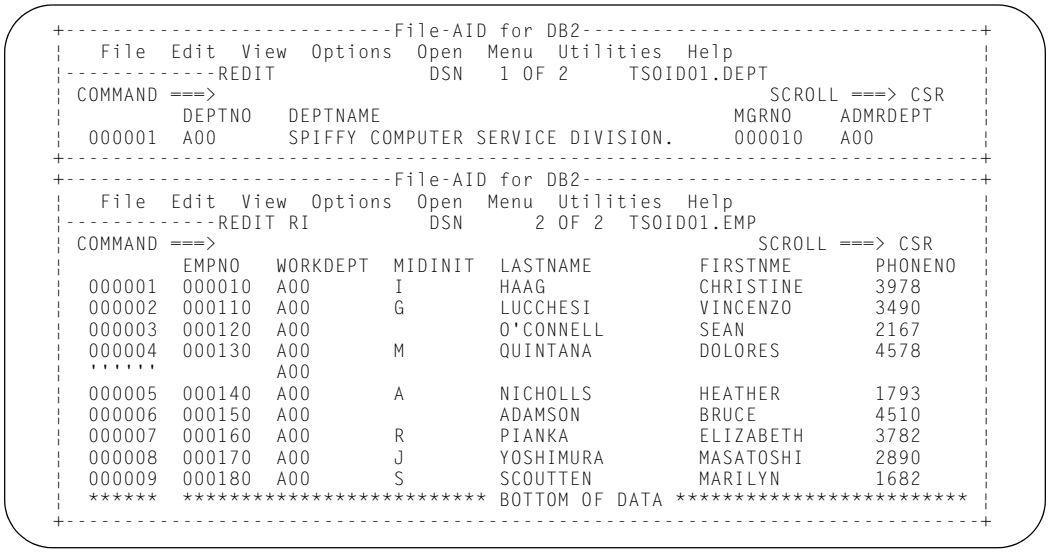
When you insert data in Related Table Edit, the foreign key field is prefilled and protected.

Figure 7-51. Related Table Edit

```
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------EDIT            DSN   1 OF 2     TSOID01.DEPT                   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          DEPTNO   DEPTNAME                          MGRNO    ADMRDEPT       ¦
¦  000001  A00      SPIFFY COMPUTER SERVICE DIVISION.  000010   A00            ¦
+-------------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
¦   File  Edit  View  Options  Open  Menu  Utilities  Help                      ¦
¦------------REDIT RI          DSN      2 OF 2  TSOID01.EMP                   ¦
¦ COMMAND ===>                                          SCROLL ===> CSR        ¦
¦          EMPNO   WORKDEPT  MIDINIT  LASTNAME         FIRSTNME       PHONENO  ¦
¦  000001  000010  A00       I        HAAG             CHRISTINE      3978     ¦
¦  000002  000110  A00       G        LUCCHESI         VINCENZO       3490     ¦
¦  000003  000120  A00                O'CONNELL        SEAN           2167     ¦
¦  I   04  000130  A00       M        QUINTANA         DOLORES        4578     ¦
¦  000005  000140  A00       A        NICHOLLS         HEATHER        1793     ¦
¦  000006  000150  A00                ADAMSON          BRUCE          4510     ¦
¦  000007  000160  A00       R        PIANKA           ELIZABETH      3782     ¦
¦  000008  000170  A00       J        YOSHIMURA        MASATOSHI      2890     ¦
¦  000009  000180  A00       S        SCOUTTEN         MARILYN        1682     ¦
¦  ******  ************************ BOTTOM OF DATA ************************    ¦
+-------------------------------------------------------------------------------+
```

The foreign key in the employee table is the WORKDEPT column. When a new row is inserted, the WORKDEPT column is prefilled with the value A00 (Figure 7-52).

Figure 7-52. Inserting a Row — Foreign Key Prefilled



Deleting Data

If you delete a row from a parent table while in a Related Table Edit session, File-AID *for DB2* adjusts the display for the dependent tables accordingly. If more than two tables are open, the deleted row in the parent table affects the display for all its dependent tables.

If your tables are related through an application relationship, the dependent table rows are only refetched; they are not deleted. If your tables are related through referential integrity, the delete rule defined through DB2 (cascade, restrict, or set null) is honored and DB2 may delete the dependent table rows.

If scrolling results in no rows in the dependent table relating to the parent table, an insert line displays. The foreign key field is filled with the primary key value from the parent table.

Figure 7-53. Deleting a Row in Related Table Edit

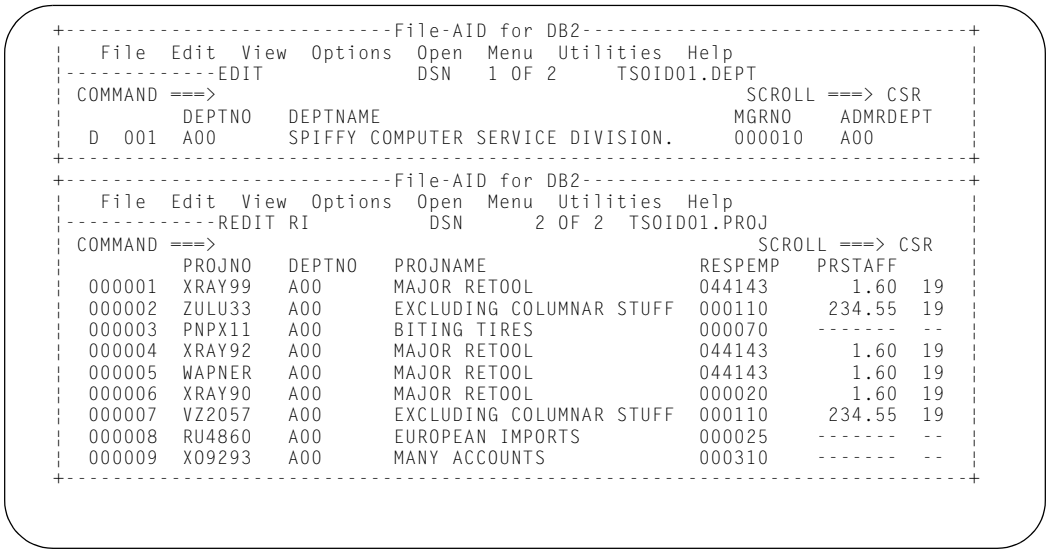


Figure 7-53 shows a Related Table Edit session between the DEPT and PROJ tables. The relationship is based on the DEPTNO column in both tables. The D line command is issued to delete department A00 from the department table.

Figure 7-54. Display Adjusted After a Deleted Row

```
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------EDIT             DSN   1 OF 2     TSOID01.DEPT                  |
| COMMAND ===>                                          SCROLL ===> CSR       |
|          DEPTNO   DEPTNAME                           MGRNO    ADMRDEPT       |
|  000001  B01      PLANNING                           000020   B01            |
+-----------------------------------------------------------------------------+
+----------------------------File-AID for DB2-----------------------------------+
|   File  Edit  View  Options  Open  Menu  Utilities  Help                      |
|-------------REDIT RI          DSN      2 OF 2  TSOID01.PROJ                  |
| COMMAND ===>                                            SCROLL ===> CSR     |
|          PROJNO   DEPTNO   PROJNAME                 RESPEMP   PRSTAFF       |
|  000001  PL2100   B01      WELD LINE PLANNING       000020       1.00  19   |
|  000002  MM3116   B01      ROBOTICS                 000020     999.99  19   |
|  000003  N36352   B01      VOORHEES ANNIHILATOR     000020    -------  --   |
|  ******  ********************** BOTTOM OF DATA ************************       |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
+-----------------------------------------------------------------------------+
```

Figure 7-54 shows the department table's display reset to the next row, department B01. All the dependent rows in the project table's display are fetched for the current parent row and refreshed. The project table's display now shows all projects in department B01.

Changing Parent Record

When you change the related key column in a parent record while in Related Table Edit, the results differ for application relationships (AR) and DB2 RI relationships (RI).

Application Relationships

If a parent column of a relationship is updated, only that one row is updated. The corresponding key columns of the dependent table are not changed. After a successful update of the parent column in a Parent to Dependent REDIT session, the dependent table display is refreshed to show any existing dependents of the newly updated parent row. (Remember, in a Dependent to Parent REDIT session, the dependent table display is not refreshed as it behaves like an unrelated table edit session.) Whenever this type of change is made, the dependent rows become orphaned.

DB2 RI Relationships

A parent column of a relationship may only be updated if no dependent rows exist. Otherwise, DB2 returns a -531 SQL code when the update is attempted. This prevents orphaned rows in the dependent table. If an update of the parent column is successful, the dependent table display is refreshed to show that no dependent rows exist.

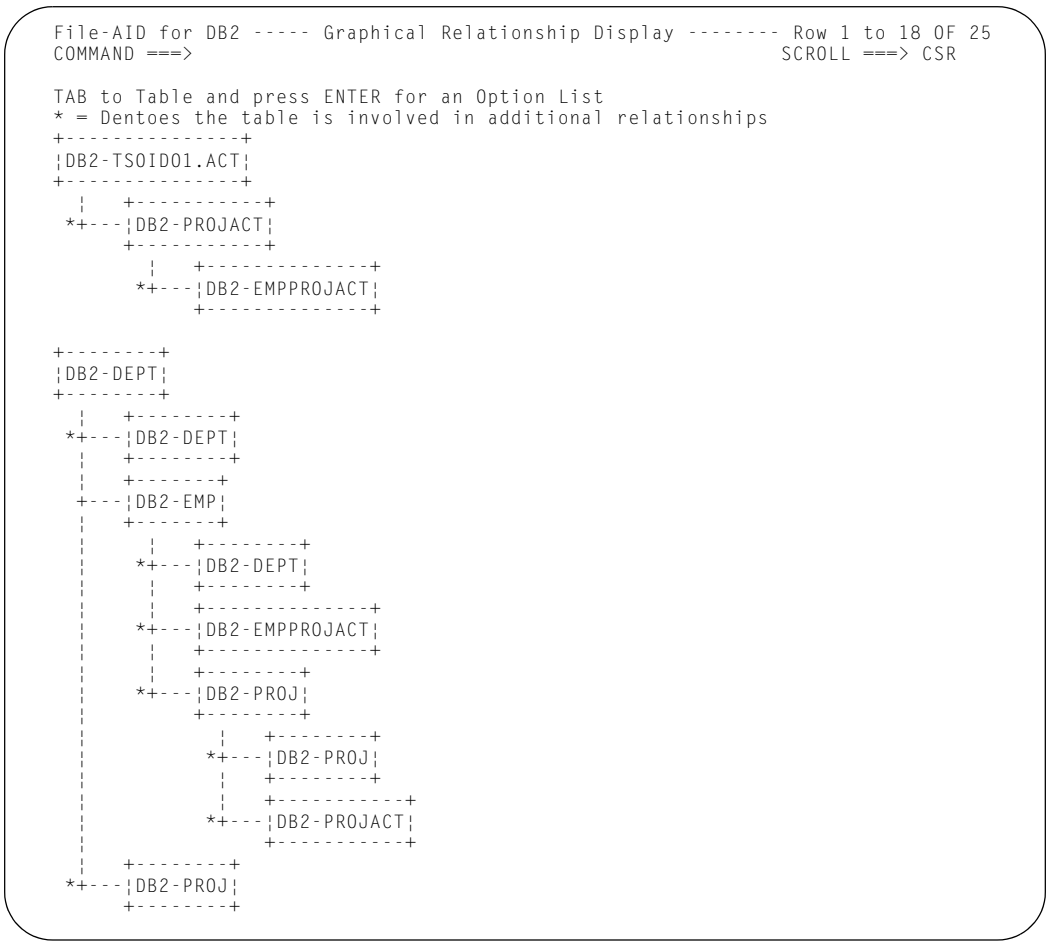
Graphic Relationship Display

You can use the GRAPHIC primary command to receive a graphic display of related tables. The graph includes both AR and RI relationships and both parent and dependent tables for any table involved in the relationship.

Figure 7-55 shows a graphical relationship display for the DEPT table. The current table is highlighted. You can scroll up/down and left/right to display information that is not visible.

If a table has an asterisk (*****) next to it, it displays somewhere else in the display; that is, it is related to more than one table. If a table displays in the graph more than once, its dependent tables only appear the first time.

Figure 7-55. Graphical Relationship Display

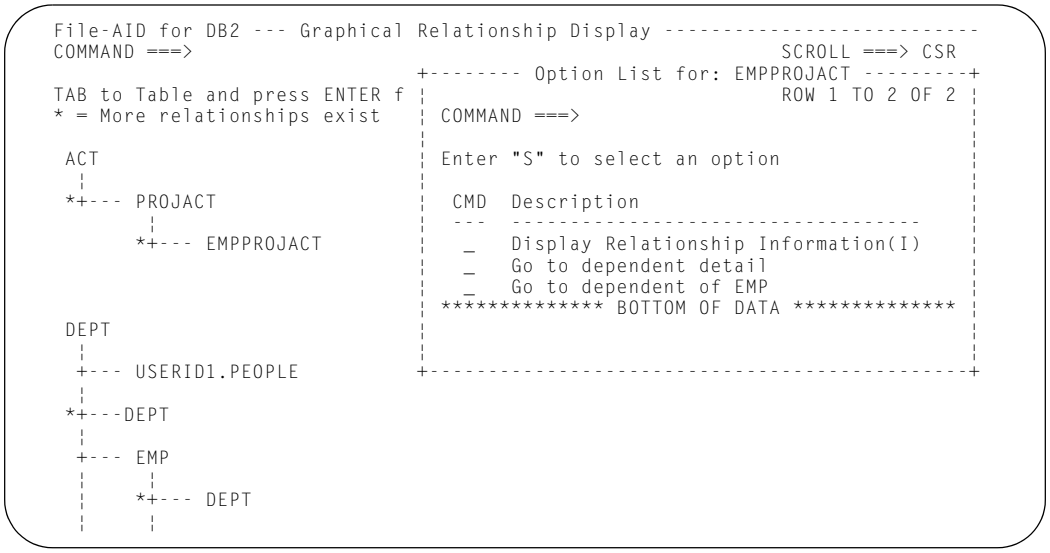


Receiving an Option List

From a Graphical Relationship Display, you can tab to any table name and press Enter to receive an Option List (Figure 7-56). It provides a list of options that you can perform for the selected table.

The items that display in the list are dependent on the selected table.

Figure 7-56. Option List for Table EMP



- Display Relationship Information: Displays the Relationship Information window. See "Relationship Information Window" on page 7-29.
- Go to Dependent Detail. Positions the cursor at the first occurrence of the table in the graph. This option will not appear if at the first occurrence of a table.
- Go to Dependent of *table name*. Positions the cursor at the first occurrence of the dependent of the specified table name. This option will not appear if there are no dependents for the selected table.

Chapter 8.
Objects Utility

This chapter describes the File-AID *for DB2* utility that lets you create, drop, or alter DB2 objects including tables, tablespaces, indexes, views, synonyms, aliases, distinct types, User Defined Functions (UDF), stored procedures, and triggers. The screens are discussed in the order displayed on the Object Selection menu.

Enter option **3** (Utilities) on the Primary Option Menu to access the Utility Selection Menu (Figure 8-1). From this menu, all of the utilities are available for selection.

Figure 8-1. Utility Selection Menu

```
File-AID for DB2 ------------------ Utility Selection Menu ---------------------
OPTION  ===>                                                    SSID ===> DSN

   1  NOT AVAILABLE   - Reserved for future use

   2  OBJECTS         - Create, Drop, Alter
                          TABLE, TABLESPACE, INDEX, VIEW, SYNONYM, ALIAS,
                          DISTINCT TYPE, DEFINED FUNCTION, TRIGGER

   3  COPY            - COPY Rows Between TABLES
                          Copy Entire TABLES or Rows/Columns Selectively

   4  OBJECT LIST     - Process DB2 Object Lists

   5  SQL MGR         - SQL Command Manager
                          SPUFI Connection

   6  PRIVILEGES      - Display, Grant, Revoke, or Modify DB2 Privileges

   7  EXTRACT, LOAD   - Extract and Load Tables

   8  IMPACT ANALYSIS - Identify DB2 objects affected by a column change

   9  BATCH FACILITY  - Batch Execute SQL with Auto-COMMIT Processing
```

Objects - Create, Drop, Alter

Enter option **2** on the Utility Selection Menu for the Create, Drop, Alter DB2 Objects utility. The Object Selection menu displays, as in Figure 8-2.

Figure 8-2. Object Selection Menu

```
File-AID for DB2 -------------- Object Selection ------------------------------
OPTION  ===>                                                  SSID ===> D610


          1  Tables                  - Create, Drop, Alter or Rename

          2  TableSpaces             - Create, Drop, Alter

          3  Indexes                 - Create, Drop, Alter

          4  Views                   - Create, Drop

          5  Synonyms                - Create, Drop, Modify

          6  Aliases                 - Create, Drop, Modify

          7  Distinct Types          - Create, Drop

          8  User Defined Functions  - Create, Drop, Alter

          9  Stored Procedures       - Create, Drop, Alter

         10  Triggers                - Create, Drop
```

Option 1, Tables

The Tables option screen displays when you select option **1** from the Object Selection menu. It lets you specify whether you want to create, alter, or drop a table.

Figure 8-3. Tables Options Screen

```
  Menu  Utilities  Help
-------------------------------------------------------------------------------
File-AID for DB2                        Tables
OPTION  ===>                                                SSID ===>  DSN

   C - Create TABLE
   A - Alter  TABLE
   D - Drop   TABLE
   R - Rename TABLE



Enter:
   Creator        ===> USERID                  Database       ===> DBNAME
   Table Name     ===> EMP                     Tablespace     ===> TSNAME
```

OPTION

Specify the letter that represents the activity you want to perform.

C Create a table. The Create Table screen displays in Figure 8-4 on page 8-4.

A Alter a table. The Alter Table screen displays in Figure 8-17 on page 8-20.

D Drop a table. The Confirm Drop screen displays in Figure 8-18 on page 8-23.

R Rename a table. The Rename Table screen displays in Figure 8-19 on page 8-23.

Creator

Specify your creator ID. Required if you are altering, renaming, or dropping a table; optional if you are creating one.

Table Name

Name of the table you want to create, alter, or drop. Required if you are altering, renaming, or dropping a table; optional if you are creating one.

Database

Name of the database that contains the table you want to alter or drop or will contain the table you are creating. If this information is available from your user profile (see "Option 5, Index, Tablespace, and Database Default Values" on page 5-7 for details), this field is prefilled.

Tablespace

Name of the tablespace that contains the table you want to alter or drop or will contain the table you are creating. If this information is available from your user profile (see "Option 5, Index, Tablespace, and Database Default Values" on page 5-7 for details), this field is prefilled.

Table Create

The Table Create screen displays when you Enter **C** on the Tables option screen. It also displays when you press Enter on the Table Copy Selection screen (Figure 9-1 on page 9-1) if the TO TABLE does not exist.

The Table Create screen lets you define the table you want to create. Also, you need to define the primary and foreign keys by issuing the PRIMARY and FOREIGN commands, or selecting Primary Keys and Foreign Keys from the Define pull-down menu. (See "Primary Key Definition" on page 8-9 and "Foreign Key Definition" on page 8-10 for details.)

You can also define column default values, column check constraints, named constraints, and identity columns. See "Default Values" on page 8-12, "Column Check Constraints" on page 8-13, "Named Check Constraints" on page 8-14, "Implicit Tablespace Confirmation" on page 8-17, and "Identity Column" on page 8-15 for details.

After you have defined your table and, optionally, its primary and foreign keys, default values and constraints, return to the Table Create screen and either enter the CREATE command, or select Execute Create from the Actions pull-down.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the table. Enter SQL on the command line, or select View/Edit DDL from the Actions pull-down, after you have defined your table and before you execute the CREATE primary command.

After the CREATE command is processed, the Index Create screen (Figure 8-26 on page 8-35) displays automatically if an index is required.

Figure 8-4. Table Create Screen

```
Menu  Define  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                  Table Create               Row 1 to 5 of 8
COMMAND ===>                                               SCROLL ===> PAGE
                                                             SSID: D701
Creator    => USERID1            Database   =>           Editproc  =>
Table Name =>                    Tablespace =>           Validproc =>
Restrict Drop => NO  (Yes or No)      CCSID =>          (EBCDIC or ASCII)
Data Capture  => NO  (Yes or No)
Temporary     => NO  (Yes or No)

MODEL TABLE (Optional, * or pattern for list)
Creator    => USERID1            Model Identity Attributes => NO   (Yes or No)
Table Name =>                    Model Check Constraints => NO   (Yes or No)

                                          Length   Unit   Allow
       Column Name        Type             ,scale   (for   Nulls   Fieldproc
                                           (LL,SS)   LOB)  (Y/N)
       ------------------ ---------------- -------  ----- -------  ----------
''''''
''''''
''''''
''''''
''''''
```

Table Information Area

Creator

Specify your user ID as the person creating the table. If this information was provided on the Tables option screen, this field is prefilled.

Table Name

Name of the table you want to create. If this information was provided on the Tables option screen, this field is prefilled.

Restrict Drop

Specify whether to place the drop restriction on the table or not. You may change the drop restriction later with the Alter Table function ("Table Alter" on page 8-20).

NO Do not place drop restriction on the table.

YES Place drop restriction on the table so table cannot be dropped until the drop restriction is specifically removed.

Data Capture

Specify whether the logging of SQL INSERT, UPDATE, and DELETE operations on the table is to be augmented by additional information. You may change the data capture specification later with the Alter Table function ("Table Alter" on page 8-20).

NO Do not enable data capture for the table.

YES Enable data capture. DATA CAPTURE YES can greatly increase the size of your log.

Temporary

Specify whether the table will be a global temporary table or not.

NO The new table is not a temporary table.

YES The new table is a global temporary table. File-AID *for DB2* creates a table using the SQL CREATE GLOBAL TEMPORARY TABLE statement that is used to hold temporary data.

Note: File-AID *for DB2* only supports creation of temporary tables. You cannot use File-AID *for DB2* to browse, edit, copy, extract, load, or print a temporary table.

Database

Name of the database to contain the table you are creating. If this information was provided on the Tables option screen, this field is prefilled.

Tablespace

Name of the tablespace (optional) in which you want the table you are creating to reside. If you enter a tablespace name here, it must already exist and allow adding a new table. If the information is contained in your user profile (see Chapter 5, "Defaults and User Parameters" for details), this field is prefilled. Leave this field blank, and Tablespace defaults to the table name (see also "Implicit Tablespace Confirmation" on page 8-17).

CCSID

Specify the encoding scheme for data stored in the table.

EBCDIC (Default) Specifies that the data must be encoded by using the EBCDIC CCSIDs that are specified during installation.

ASCII Specifies that the data must be encoded by using the ASCII CCSIDs that are specified during installation.

UNICODE Specifies that the data must be encoded by using the UNICODE CCSIDs that are specified during installation. (DB2 Version 7 or later only.)

Note: File-AID *for DB2* does not support exchanging data between different encoding schemes. For example, you cannot copy or paste rows from an ASCII table into an EBCDIC table.

Editproc

(Optional) Name of the program that translates the data (at the row level) into the format you want displayed on the screen.

Validproc

(Optional) Name of the program that validates the information added to the table.

Creator

ID of the model table's creator. This field defaults to the creator ID as entered in User Parms, Option 0. An asterisk (*) can be used as a wildcard character to display a selection list of model tables (Figure 8-6 on page 8-8).

Table Name

Name of the table you want to use as a model. An asterisk (*) can be used as a wildcard character to display a selection list of model tables (Figure 8-6 on page 8-8).

Model Identity Attributes

Specify whether existing identity column attributes of the model table shall be copied to the new table.

NO Do not copy identity column attributes.

YES Copy all existing identity column attributes from the model table selected.

Model Check Constraints

Specify whether existing check constraints (column and/or named constraints) of the model table shall be copied to the new table.

NO Do not copy constraints.

YES Copy all existing constraints as they apply to the modelled columns. You can then modify the constraints using the COLCHECK ("Column Check Constraints" on page 8-13) and NAMCHECK ("Named Check Constraints" on page 8-14) commands.

Column Information Area

Line commands

Enter line commands to modify the table columns. (See Chapter 4, "Line Commands" for list and descriptions of line commands.

Column Name

Enter your table column names.

Type

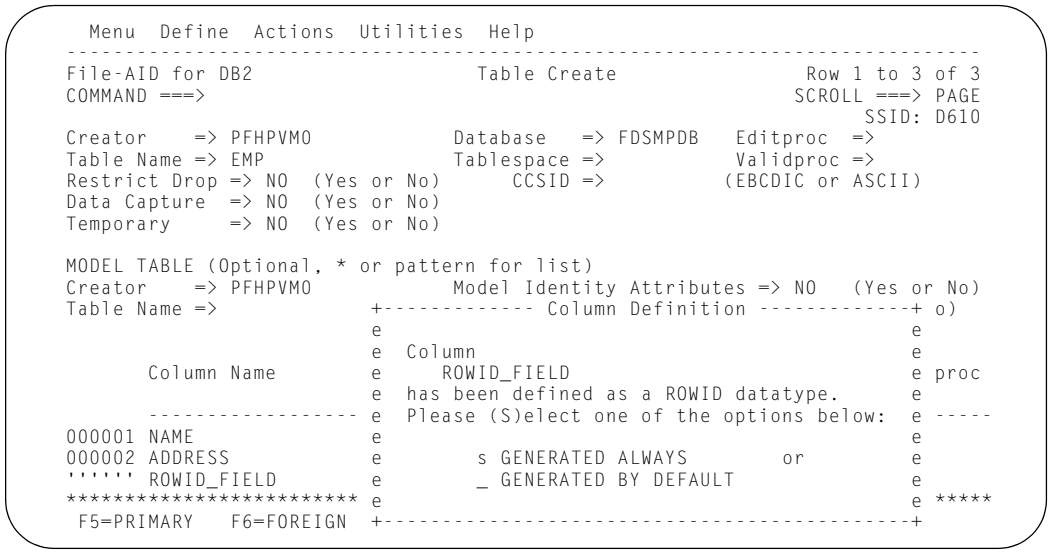
Enter the datatypes for your table columns. All valid DB2 types are supported including: CHAR, VARCHAR, INTEGER, SMALLINT DATE, TIME, TIMESTAMP, GRAPHIC, DECIMAL, LONGVAR, FLOAT, VARG, Distinct Types, Large Object Types (LOBs), and ROWIDs.

For **Distinct Types** enter the name of a previously defined distinct type ("Distinct Type Selection" on page 8-9). An asterisk (*) or a pattern in the Type field also displays the selection list. A list of distinct types matching the name entered displays for selection.

For **LOBs:** When you enter a large object type (BLOB, CLOB, or DBCLOB) you must also define a ROWID column for the table. A SET CURRENT RULES = 'STD' is issued when creating a base table with LOB data. LOB columns are set to DEFAULT NULL.

For **ROWIDs:** Columns can be either GENERATED ALWAYS or GENERATED BY DEFAULT. The popup in Figure 8-5 allows you to select one of these options.

Figure 8-5. ROWID Datatype pop-up



Length

Enter the lengths of your table column values.

If the column type is a LOB specify the maximum length of the variable string in the unit specified in the Unit (for LOB) field. For example, a length of 2 and a unit of M specifies a maximum length of 2 megabytes (2097152 bytes).

For specific length information, see "Column Values" on page 6-16.

Unit (for LOBs)

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K kilobytes

M megabytes

G gigabytes

Allow Nulls

Indicate if you want columns in your table to allow null entries.

Y Nulls permitted.

N Nulls not permitted.

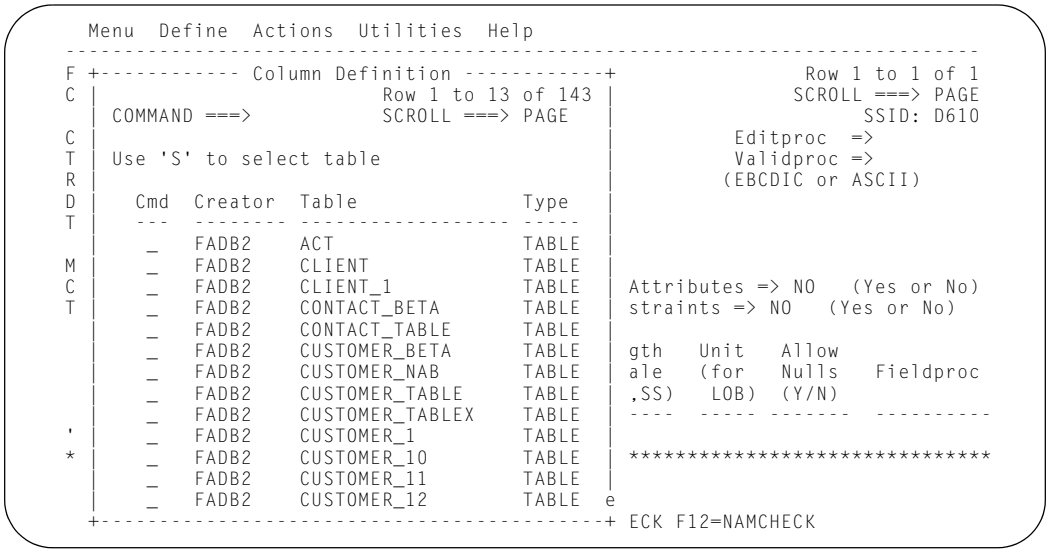
FIELDPROC

(Optional) Names of the programs that translate the data (at the column level) into the format you want displayed on the screen.

Model Table Selection

If you use a wildcard character ("Wildcard Characters" on page 2-9) in either the CREATOR or TABLE NAME fields in the Model Table Area on the Table Create screen, the Model Table Selection window displays all matching tables. After selecting a table, press Enter to process the selection.

Figure 8-6. Model Table Selection Window



Cmd

Enter **S** to select the model table.

Creator

Lists the creator IDs to be selected.

Table

Lists the table names to be selected.

Type

Type of object. Valid values are TABLE or VIEW.

Model Table Area

When you use the Table Create screen to create a new table, File-AID *for DB2* lets you copy the column attributes of a previously defined table into the working area. The copy is controlled by the MODEL TABLE CREATOR and TABLE NAME fields on the screen. If you do not enter a value for the CREATOR, it defaults to the current user ID.

The column attributes of the specified table are added to the current work area. It is possible to add multiple tables to the work area. You may also enter a model table after you have entered columns manually. After the copy, you can use standard editing techniques to rename or redefine the columns before the table is created.

File-AID *for DB2* copies only the following column attributes:

- Name
- Type
- Length
- Scale
- Not null
- Not null with default
- Fieldproc
- Primary key
- Default values
- Column constraints and MODEL CHECK CONSTRAINTS=YES)
- Named constraints and MODEL CHECK CONSTRAINTS=YES)
- Identity

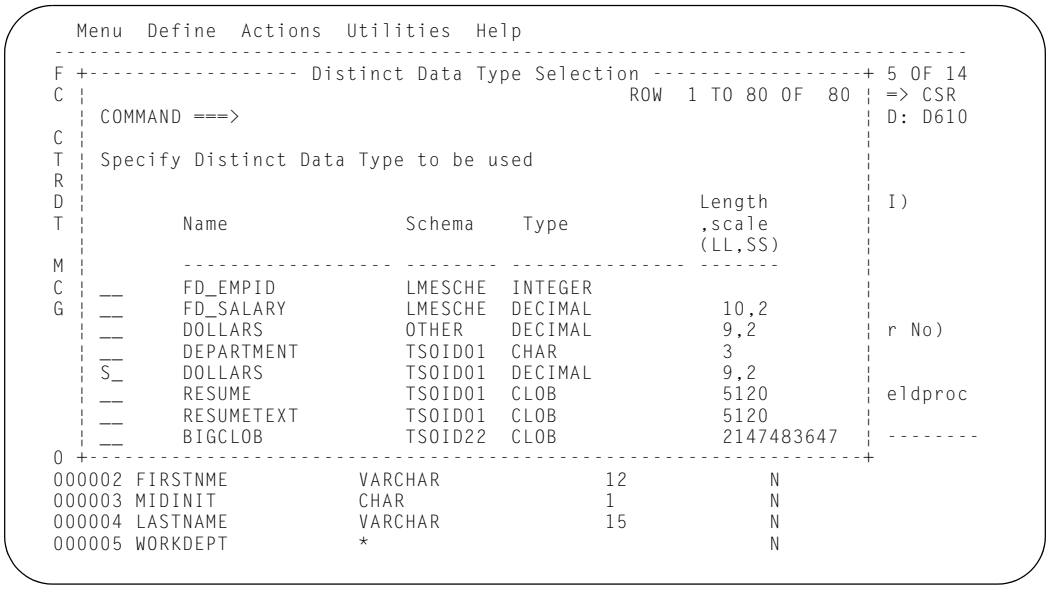
File-AID *for DB2* does not copy the following column attributes from the model table:

- Foreign key
- Editproc
- Validproc

Distinct Type Selection

If you use a wildcard character ("Wildcard Characters" on page 2-9) in the Type field of the Column Area on the Table Create or Alter screen, the Distinct Type Selection window displays all matching distinct types. After selecting a distinct type, press Enter to process the selection.

Figure 8-7. Distinct Data Type Selection Window



Name

Displays the name of the distinct type.

Schema

Displays the schema (ID) of the distinct type.

Type

Built-in column datatype of the distinct type.

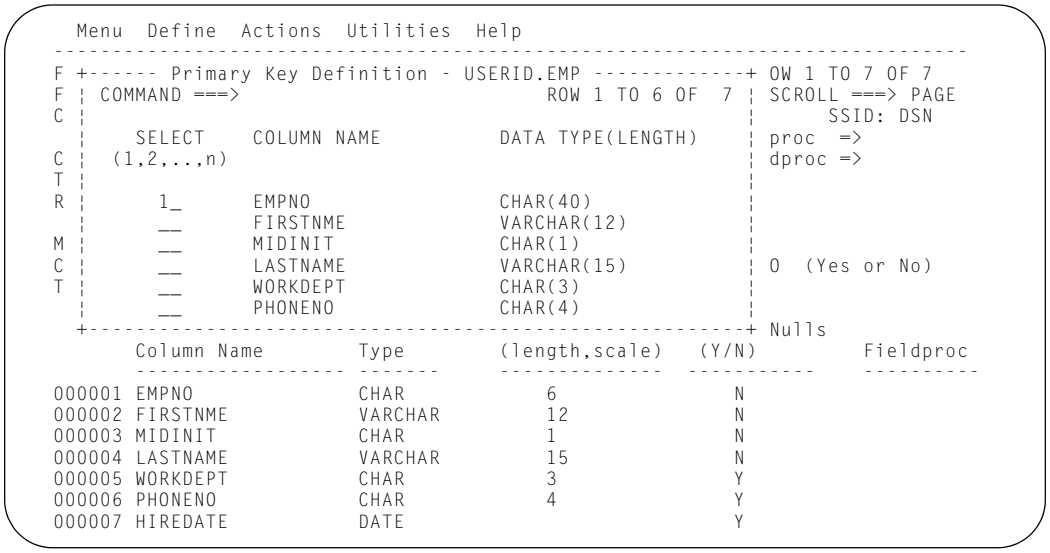
Length, Scale

Length and scale of the built-in column datatype for the distinct type.

Primary Key Definition

The Primary Key Definition window displays by selecting Primary Key from the Define pull-down menu, or entering the PRIMARY command after you finish typing in information on the Table Create screen. It lets you set up your primary key for the table. When you have defined your primary and foreign keys, you must return to the Table Create screen and enter the CREATE primary command to create the table.

Figure 8-8. Primary Key Definition Window



Select

Specify which column or columns you want defined as the primary key. If you want multiple columns, the order of the columns should also be indicated in the entry (1, 2, 3, and so on).

Column Name

Names of the columns in which **Y** was specified in the ALLOW NULLS field of the Table Create screen.

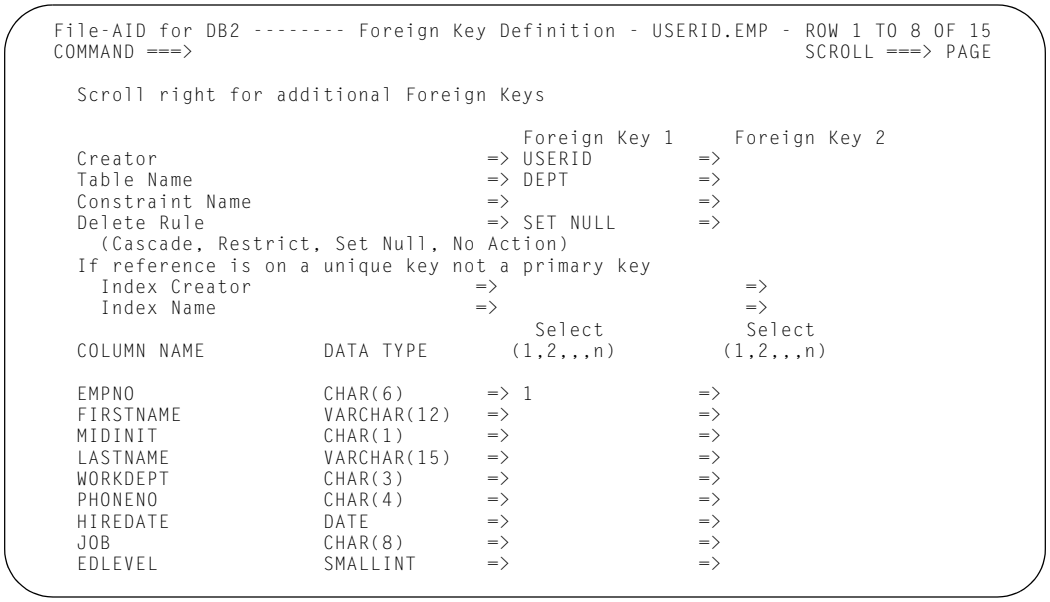
Data Type (Length)

Types and lengths of the values in the COLUMN NAME field.

Foreign Key Definition

The Foreign Key Definition screen displays by entering the FOREIGN command, or selecting Foreign Keys from the Define pull-down menu. It lets you set up one or more foreign keys for the table. When you have defined your primary and foreign keys, return to the Table Create screen and either select Execute Create from the Actions pull-down menu or enter the CREATE command to create the table.

Figure 8-9. Foreign Key Definition Screen



Creator

ID of parent table creator.

Table Name

Name of the parent table. Must have same CCSID as the table to be created.

Constraint Name

(Optional) Name of the constraint.

Delete Rule

Specify the course of action when data is deleted from the key:

C (Cascade)

If a row is deleted from the parent table, any dependent rows are automatically deleted.

R (Restrict)

Disallows any delete request that would violate referential integrity.

S (Set Null)

If a row is deleted from the parent table, any dependent data is set to nulls.

N (No Action)

Disallows any delete request that would violate referential integrity.

Note: The only difference between NO ACTION and RESTRICT is when the referential constraint is enforced. RESTRICT (IBM SQL rules) enforces the rule immediately, and NO ACTION (SQL standard rules) enforces the rule at the end of the statement. This difference matters only in the case of a searched DELETE involving a self-referencing constraint that deletes more than one row. NO ACTION might allow the DELETE to be successful where RESTRICT (if it were allowed) would prevent it.

Index Specification Area

Index information is only needed if reference is on a unique key, not a primary key.

Index Creator

Creator ID of parent table's index.

Index Name

Name of the parent table's index.

Column Name

Names of the columns. This information is carried forward from the Table Create screen.

Data Type (Length)

Types and lengths of the values in the columns.

Select

Specify which column or columns you want defined as the foreign key. If you want multiple columns, the order of the columns should also be indicated in the entry (1, 2, 3, and so on).

Default Values

The Default Values window displays by selecting Default Values from the Define pull-down menu, or entering the DEFAULT command after you finish typing in information on the Table Create screen. It lets you define the default values for all columns that you have marked Y or N in the Allow Nulls field. When you have defined your default values, you must return to the Table Create screen and enter the CREATE primary command to create the table.

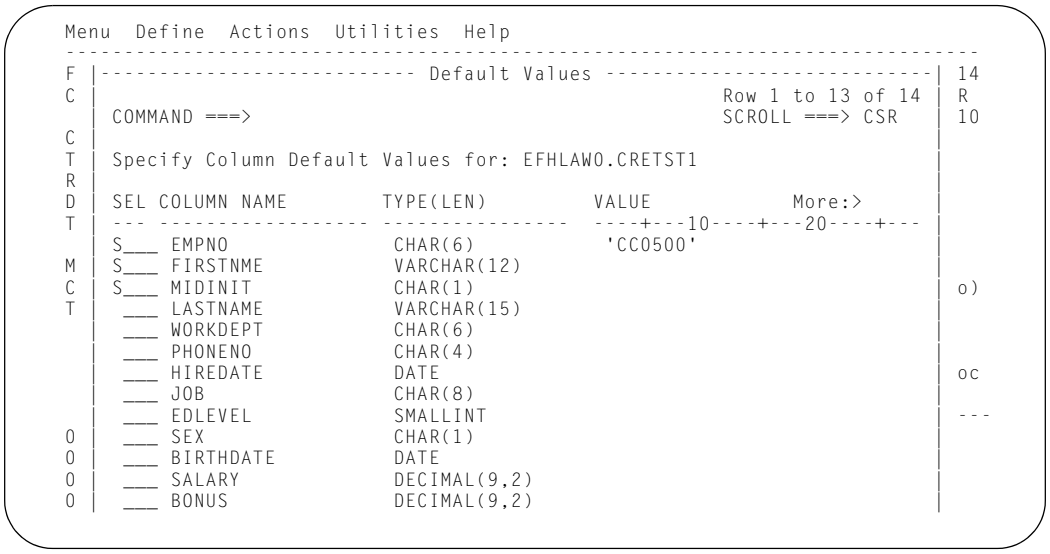
Note: Once the table has been created you cannot change or add the default values for existing columns.

Figure 8-10 shows a sample Default Values screen. From this screen, you select a specific column and press Enter. The value field is either populated from a model, or you can enter a specific value.

If you enter a reserved word, File-AID *for DB2* validates the type and length of the column to determine if the result of the reserved-word substitution is compatible (for example, CURRENT SQLID requires a CHAR or VARCHAR column of at least 8 characters).

If the column type is DISTINCT, File-AID *for DB2* validates the underlying built-in datatype, the cast syntax, and any reserved words. If the column selected is DISTINCT type, and the cast function is not specified, it is added for you.

Figure 8-10. Default Values Definition Window



Column Name

Names of the columns in which Y or N was specified in the ALLOW NULLS field of the Table Create screen.

Data Type (Length)

Types and lengths of the values in the COLUMN NAME field.

Value

Type in the default values for each column as desired. Enclose text strings in single or double quotes. The specified default value must match the column datatype and cannot exceed the length. The only reserved words allowed are USER, CURRENT_SQLID, and NULL. To use reserved words as text strings, enclose them with single or double quotes. If the model table has user defined default values they are prefilled. Modify them as desired.

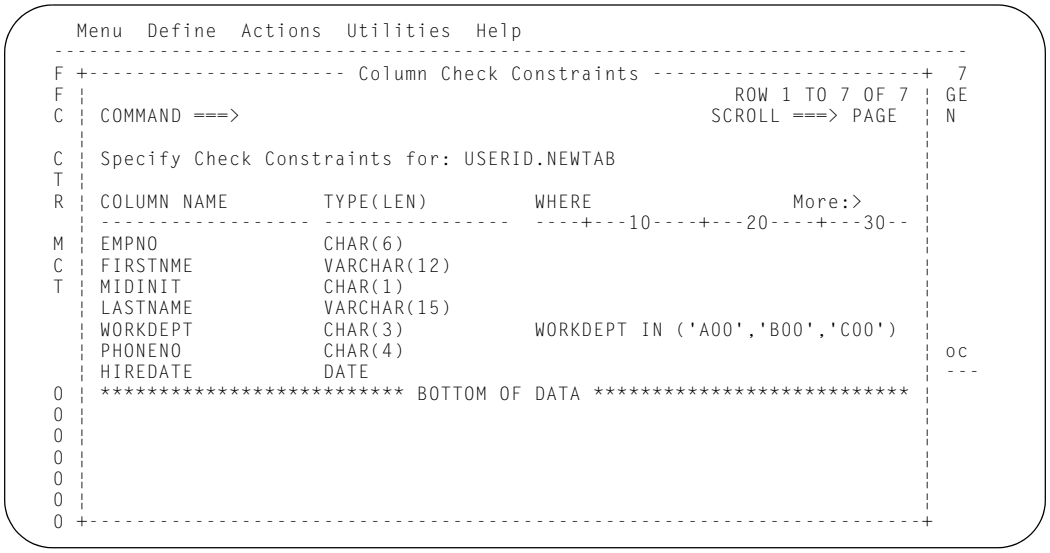
Use the RIGHT and LEFT commands to scroll the Value field.

Column Check Constraints

The Column Check Constraints window displays when you select Column Constraints from the Define pull-down menu, or enter the COLCHECK command after you finish typing in information on the Table Create screen. It lets you define Column Check Constraints for all columns. When you have finished specifying your column constraints you must return to the Table Create screen and enter the CREATE primary command to create the table.

Note: Once you have created the table you can modify the defined column check constraints later using the Alter Table and Define Named Constraint functions, as column constraints are listed there under their column name (also “Defining or Modifying Default Values or Column Constraints” on page 8-22).

Figure 8-11. Column Check Constraints Window



Column Name

Names of all columns in the new table.

Data Type (Length)

Types and lengths of the values in the COLUMN NAME field.

Where

For each column that you want to apply a column check constraint, type in the condition as desired. The WHERE delimiter is analogous to the SQL WHERE clause in a SELECT statement. Up to 3800 characters are available by scrolling right. You may use the following delimiters in your WHERE clauses:

>	OR	< >
<	> =	BETWEEN
=	< =	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

Note: If the column name is a distinct type, and there are no distinct types in the WHERE clause, then the column name is automatically cast if AUTOCAST is turned on. If there are distinct types in the WHERE clause, then this does not occur. If you want to cast it, you must do it manually.

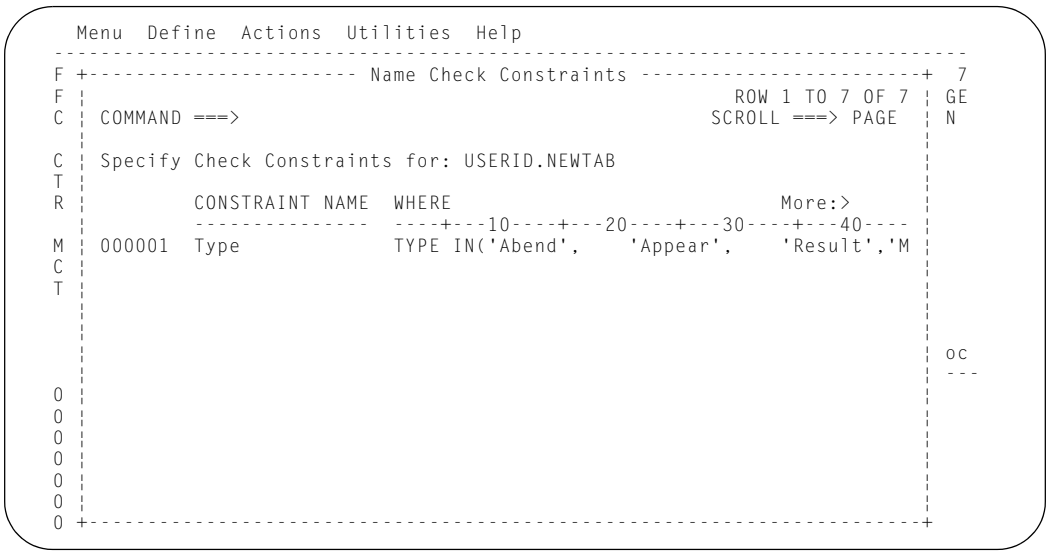
Named Check Constraints

The Named Check Constraints window displays when you select Named Constraints from the Define pull-down menu, or entering the NAMCHECK command after you finish typing in information on the Table Create screen. It lets you define Named Check Constraints for the new table that reference one or more columns. When you have finished specifying your named check constraints use the END command to return to the Table Create screen and enter the CREATE primary command to create the table.

Note: Once you have created the table you can modify the defined named check constraints later using the Alter Table and Define Named Constraint functions

(see also "Defining or Modifying Default Values or Column Constraints" on page 8-22).

Figure 8-12. Column Check Constraints Window



Constraint Name

Names of all defined named check constraints for the new table.

Where

For each named check constraint, type in the condition as desired. The WHERE delimiter is analogous to the SQL WHERE clause in a SELECT statement. Up to 3800 characters are available by scrolling right. You may use the following delimiters in your WHERE clauses:

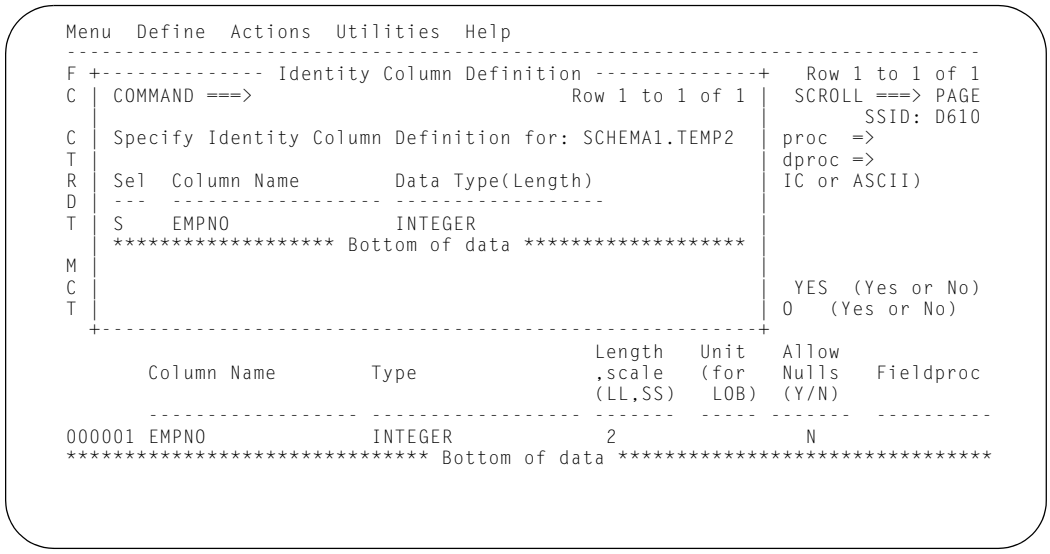
>	OR	< >
<	> =	BETWEEN
=	< =	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

Note: If the column datatype is DISTINCT you must enter the CAST function manually.

Identity Column

The Identity Column definition window displays when you select Identity Column from the Define pull-down menu, or enter the IDENTITY command, after you have identified at least one column of either INTEGER, SMALLINT, or DECIMAL with no scale type.

Figure 8-13. Identity Column Definition Window



Column Name

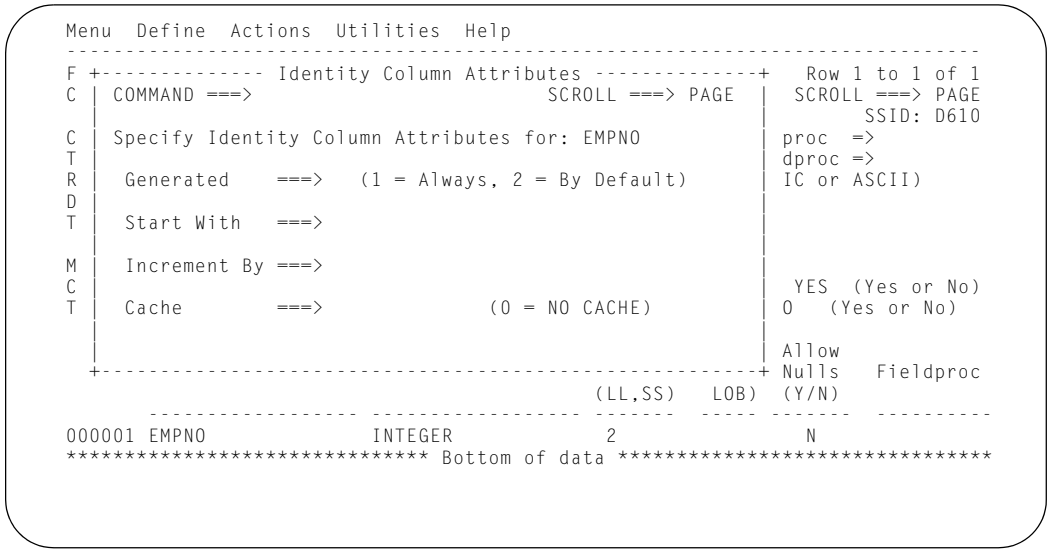
Name of the columns of type INTEGER, SMALLINT, or DECIMAL.

Data Type

Type and length of the column.

Select the specific column name you want to modify and press Enter. The Identity Column Attributes window displays.

Figure 8-14. Identity Column Attributes Window



Generated

Valid values are ALWAYS and DEFAULT.

1 ALWAYS - DB2 always generates a value for this column when a row is inserted into the table.

2 DEFAULT - DB2 generates a value for the column when a row is inserted into the table unless a value is specified.

Start with

The first value for the identity column.The value can be any positive or negative value that could be assigned to the column without non-zero digits existing to the right of the decimal point.

Increment by

The interval between consecutive values of the identity column. The value can be any positive or negative value that is not zero, does not exceed the value of a large integer constant, and could be assigned to the column without any non-zero digits existing to the right of the decimal point. Default is 1.

Cache

The number of values to keep in memory. Valid values are 2 through the largest integer value; the SQL shows CACHE *nn* where nn is the value. If cache is 0, then the SQL shows NOCACHE.

Implicit Tablespace Confirmation

The Implicit Tablespace Confirmation screen displays if you do not name a tablespace before you enter the CREATE command. It lets you confirm that the omission was intentional. The tablespace that is implicitly created follows DB2 rules for implicit tablespace creation.

Figure 8-15. Implicit Tablespace Confirmation Screen

```
File-AID for DB2 ------------ Implicit Tablespace Confirmation -----------------
COMMAND ===>

Table Name: USERID.EMP
Database  : DBNAME

Tablespace Name ===>

No Tablespace was specified; a tablespace will be created in the Database.

Continue with table creation?  YES  (YES/NO)

   Enter YES to continue
         NO  to return to the specification panel
```

Table Name

Name of the table you are creating.

Database

Name of the database in which the table will reside.

Tablespace Name

Name of an explicit tablespace that already exists within the specified database. If this field is left blank, File-AID *for DB2* follows DB2 rules for implicit tablespace creation.

Continue with table creation?

Indicate whether you want to create a tablespace in the database, since none was named.

Index Create

The Index Create screen displays automatically if an index is required. When you have defined your index, either select Execute Create from the Actions pull-down menu or enter the CREATE primary command to create the index and return to the Table Create screen. For more information on the Index Create screen, see "Index Create" on page 8-34.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the index. Enter SQL on the command line after you have defined your index and before you execute the CREATE primary command.

Figure 8-16. Index Create screen

```
 Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                  Index Create              ROW 1 TO 9 OF 14
COMMAND ===>                                                 SCROLL ===> CSR
Table created, Index required on primary key

Creator    ===> USERID            Creator   : USERID
Index Name ===>                   Table Name: EMP1

Index Type ===> 1 (1/2)           Partitioned ===> N (y/n)
Cluster    ===> Y (y/n)           Unique      ===> 1 (1. Yes, 2. No,
Piecesize  ===>          (K)                        3. Unique when not null)

   SELECT    A/D  COLUMN NAME          DATA TYPE(LENGTH)
 (1,2,..,n)

     1            EMPNO                CHAR(6)
                  FIRSTNME             VARCHAR(12)
                  MIDINIT              CHAR(1)
                  LASTNAME             VARCHAR(15)
                  WORKDEPT             CHAR(3)
                  PHONENO              CHAR(4)
                  HIREDATE             DATE(10)
                  JOB                  CHAR(8)
```

Creator (Index)

Your user ID. This field is prefilled with the value from the Table Create screen.

Index Name

Name of the index you want to create.

Creator (Table)

ID of person who created the table on which the index is based. This field is prefilled with the value from the Table Create screen and protected from update.

Table Name

Name of the table on which the index is based. This field is prefilled with the value from the Table Create screen and protected from update.

Index Type

DB2 can create only type 2 indexes.

Cluster

Indicate whether the index is designated as clustering. This field defaults to **N**.

Piecesize

Indicate maximum size of dataset storage piece (in kilobytes (K) that is used by DB2 for nonpartitioned indexes. You must enter the value in kilobytes. Megabytes and gigabytes are only referenced for your convenience. Allowable values are:

- 256 K
- 512 K
- 1024 K (or 1 M)
- 2024 K (or 2 M)
- 4096 K (or 4 M)
- 8192 K (or 8 M)
- 16384 K (or 16 M)
- 32768 K (or 32 M)
- 65536 K (or 64 M)
- 131072 K (or 128 M)
- 262144 K (or 256 M)
- 524288 K (or 512 M)
- 1048576 K (or 1024 M or 1 G)
- 2097152 K (or 2048 M or 2 G)
- 4194304 K (or 4096 M or 4 G) with LARGE, DSSIZE 4 G (or greater)
- 8388608 K (or 8192 M or 8 G) with DSSIZE 8 G (or greater)
- 16777216 K (or 16384 M or 16 G) with DSSIZE 16 G (or greater)
- 33554432 K (or 32768 M or 32 G) with DSSIZE 32 G (or greater)
- 67108864 K (or 65536 M or 64 G) with DSSIZE 64 G.

The default for piecesize is 2 G (2 GB) for indexes backed by non-LARGE tablespaces and 4 G (4 GB) for indexes backed by LARGE or DSSIZE tablespaces.

Partitioned

Indicate whether the indexspace for the index is partitioned. This field defaults to **N**.

Unique

Indicate whether more than one row in the indexed base table can have identical values. This field defaults to **2** (NO).

1 Yes	All rows must be unique.
2 No	Rows can have identical values.
3 Unique when not null	Null rows can have identical values, all others must be unique. Option 3, Unique where not null is only valid for Index Type 2.

Column List Area

Select

Specify which column(s) in the table is the key, indicating the order of the keys (values 1, 2, 3, and so forth) if there is more than one.

A/D

Specify whether the index is in ascending or descending order within the column. The default is ascending.

Column Name

Names of the table columns on which the index is based.

Data Type(Length)

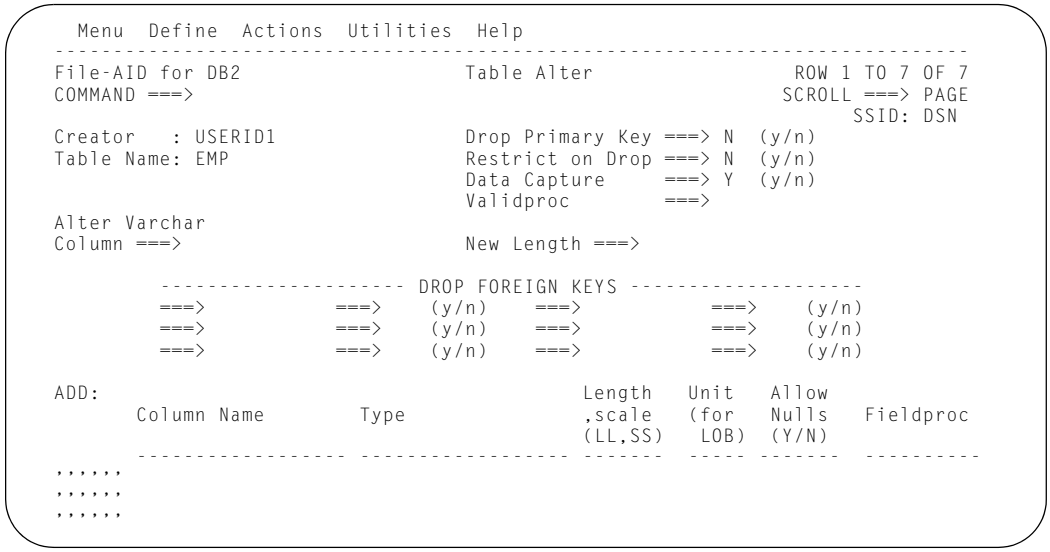
Datatypes and lengths of the table columns on which the index is based.

Table Alter

The Table Alter screen displays when you choose **A** from the Tables Option screen. It lets you add columns to an existing table. After you specify your changes, you can create or modify your primary and foreign keys. See "Primary Key Definition" on page 8-9 and "Foreign Key Definition" on page 8-10 for descriptions of those screens. When all changes have been made, enter the ALTER primary command, or select "Execute Alter" from the Actions pull-down, on the Table Alter screen to effect those changes.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to alter the table. Enter SQL on the command line after you have altered your table and before you execute the ALTER primary command.

Figure 8-17. Table Alter Screen



Creator

Creator ID. If this information was provided on the Tables option screen, this field is prefilled.

Table Name

Name of the table you want to alter. If this information was provided on the Tables option screen, this field is prefilled.

Drop Primary Key

Indicate whether you want to drop the current primary key.

Restrict On Drop

Specify whether to place the drop restriction on the table or not.

NO Do not place drop restriction on the table.

YES Place drop restriction on the table so table cannot be dropped until the drop restriction is specifically removed.

Data Capture

Specify whether the logging of SQL INSERT, UPDATE, and DELETE operations on the table is to be augmented by additional information. DATA CAPTURE YES can greatly

increase the size of your log. You may change the data capture status that was established with the Create Table function (see "Table Create" on page 8-3).

NO Do not enable Data Capture for the table.

YES Enable Data Capture.

VALIDPROC

(Optional) Name of the program that validates the information added to the table.

Alter Varchar Column

Enter the column name whose length you want to alter. You can only alter one column at a time, and that column must exist and be a varchar column. Wildcards are not valid. Distinct types with a built-in type of varchar are not valid.

New Length

Enter the new length for the varchar column listed in the Alter Varchar Column field. You must enter a length greater than the existing length.

Drop Foreign Keys

(Optional) When the Table Alter screen displays, the first six foreign key constraint names appear in the DROP FOREIGN KEYS area. Typing a **Y** after one or more arrows causes the referential integrity constraint to be dropped. If there are more than six constraint names and the constraint to be dropped is not displayed, you can enter the desired name by typing over one of the displayed names.

ADD Area (Optional)

Column Name

Type in the names of the columns you want to add.

Type

Enter the datatypes for your new table columns. All valid DB2 types are supported including: CHAR, VARCHAR, INTEGER, SMALLINT DATE, TIME, TIMESTAMP, GRAPHIC, DECIMAL, LONGVAR, FLOAT, VARG, Distinct Types, Large Object Typs (LOBs).

For **Distinct Types**, enter the name of a previously defined distinct type. If the distinct type name is not unique, select it from the distinct type selection list ("Distinct Type Selection" on page 8-9). An asterisk (*) or a pattern in the Type field will also display the selection list.

For **LOBs**: When you enter a large object type (BLOB, CLOB, or DBCLOB) you must also define a ROWID column for the table, and whether that ROWID is GENERATED ALWAYS or GENERATED BY DEFAULT.

Note: When adding a LOB column to an existing table, you must create its auxiliary table and index manually.

Length

Enter the lengths of your table column values.

If the column type is a LOB specify the maximum length of the variable string in the unit specified in the Unit (for LOB) field. For example, a length of 2 and a unit of M specifies a maximum length of 2 megabytes (2097152 bytes).

Unit (for LOBs)

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

Allow Nulls

Indicate whether the new columns in your table have default values when they are not explicitly set.

Y	Allow nulls.
N	Nulls are not allowed.

Fieldproc

(Optional) Names of the programs that translate data (at the column level) into the format you want displayed on the screen.

Creating or Modifying Keys

If you wish to change primary or foreign keys, see "Primary Key Definition" on page 8-9 and "Foreign Key Definition" on page 8-10 for examples and descriptions of the screens. To evoke the Primary Key Definition screen, select Primary Keys from the Define pull-down menu or enter the PRIMARY command. To evoke the Foreign Key Definition screen, select Foreign Keys from the Define pull-down menu or enter the FOREIGN command.

Defining or Modifying Default Values or Column Constraints

If you wish to add or change the definitions for Default Values, Column or Named Constraints, see "Default Values" on page 8-12, "Column Check Constraints" on page 8-13, and "Named Check Constraints" on page 8-14 for examples and descriptions of the screens.

- To evoke the Default Values screen, select Default Values from the Define pull-down menu or enter the DEFAULT command. Default values can only be defined for newly added columns.
- To evoke the Column Check Constraints Definition screen, select Column Constraints from the Define pull-down menu or enter the COLCHECK command. Column Constraints can only be defined for newly added columns. If you need to change a previously defined column constraint use the Named Check Constraints function.
- To evoke the Named Check Constraints Definition screen, select Named Constraints from the Define pull-down menu or enter the NAMCHECK command. The Named Check Constraints window lists all defined check constraints by name. Previously defined column check constraints are listed using their column's name. You may add, modify, or delete check constraint definitions. Your changes take effect when you execute the table alter command.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Tables option screen. It gives you the opportunity to confirm whether you want to drop the table you have

specified. This is a safety feature intended to prevent you from dropping a table unintentionally. Figure 8-18 shows a sample screen.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to drop your table. Enter SQL on the command line before pressing Enter or executing the END command.

Figure 8-18. Confirm Drop Screen

```
File-AID for DB2 -------------- Confirm Drop ---------------------- SSID: DSN
COMMAND ===>

TABLE NAME: USERID.TESTEMP


INSTRUCTIONS:

   Press ENTER key to confirm DROP request.
      (The TABLE will be dropped)

   Enter SQL to generate the SQL source without executing the drop.

   Enter END command to cancel DROP request.
```

Table Name

Name of the table to be dropped.

The same screen is presented for the D option of each menu selection in this utility: tables, tablespaces, views, indexes, synonyms, aliases, distinct types, triggers, stored procedures, and user defined functions. The screen is identical for tablespaces, indexes and views. For synonyms and aliases, the screen includes the TABLE field to identify the source of the synonym or alias. When dropping distinct types, the screen displays the distinct type's schema and name.

Rename Table

The Rename Table screen displays when you choose **R** from the Tables option screen. It gives you the opportunity to enter a new name for an existing table. Figure 8-19 shows a sample screen.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to drop your table. Enter SQL on the command line before pressing Enter or executing the END command.

Note: DB2 does not allow renaming of tables that have existing triggers.

Figure 8-19. Rename Table Screen

```
File-AID for DB2 -------------- Rename Table ---------------------- SSID: DSN
COMMAND ===>

          CREATOR: TSOID01
   OLD TABLE NAME: EMPTEST
   NEW TABLE NAME: EMP

INSTRUCTIONS:

   Enter new table name and press ENTER key to process RENAME request.
      (The TABLE will be renamed)

Use SQL to generate the SQL source without executing the RENAME

Use END command to cancel RENAME request
```

Creator

Creator ID of the table to be renamed.

Old Table Name

Name of the table to be renamed.

New Table Name

Enter the new name for the above table. You must have the required DB2 privileges to rename the table.

An error is issued if the source table is currently referenced in any view definitions. There is no support for changing the name of an alias. An alias on the source table continues to refer to the source table after the rename.

The specified table is renamed to the new name. All privileges and indexes on the table are preserved.

Option 2, Tablespaces

The Tablespaces option screen displays when you select option **2** from the Object Selection menu. It lets you specify whether you want to create, alter, or drop a tablespace.

Figure 8-20. Tablespaces Option Screen

```
  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2                    Tablespaces
OPTION  ===>                                                 SSID ===> DSN

   C - Create TABLESPACE
   A - Alter  TABLESPACE
   D - Drop   TABLESPACE

Enter:
   Database Name   ===> DBNAME
   Tablespace Name ===> TSNAME
```

OPTION

Specify the letter representing the activity you want to perform:

C

Create a tablespace. (Figure 8-21 on page 8-25 or Figure 8-22 on page 8-28.)

A

Alter a tablespace. (Figure 8-23 on page 8-29 or Figure 8-24 on page 8-32.)

D

Drop a tablespace. A screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23) displays.

Database Name

Name of the database in which the tablespace resides.

Tablespace Name

Name of the tablespace you want to create, alter, or drop.

Tablespace Create

The Tablespace Create screen displays when you choose **C** from the Tablespaces option screen. This screen lets you define the tablespace you want to create. When you have defined the tablespace you are creating, enter the CREATE primary command or select Execute Create from the Actions pull-down menu to create the tablespace.

When you create a tablespace that is partitioned, request the Partitioned Tablespace Create screen by completing the NUMPARTS field and entering the PARTITN command (or selecting Define Partitions from the Actions pull-down menu).

Tablespace Create

The Tablespace Create screen displays when you choose to create a tablespace. When you have defined the tablespace you are creating, enter either the CREATE primary command for a nonpartitioned tablespace, or the PARTITN primary command for a partitioned tablespace. Figure 8-21 on page 8-25 shows a sample Tablespace Create screen.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the tablespace. Enter SQL on the command line after you have defined your tablespace and before you execute the CREATE primary command.

Figure 8-21. Tablespace Create Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2              Tablespace Create                  SSID: DSN
COMMAND ===>

Database Name   ===> DBNAME
Tablespace Name ===> TSNAME

  STOGROUP  ===> STOG1        FREEPAGE ===> 5       BUFFERPOOL ===> BP0
  PRIQTY(K) ===> 1            PCTFREE  ===> 15      LOCKSIZE   ===> ANY
  SECQTY(K) ===> 1            NUMPARTS ===>         CLOSE      ===> YES
  ERASE     ===> NO           GBPCACHE ===>         DSETPASS   ===> DSN
  TRACKMOD  ===> YES          LOCKMAX  ===>         SEGSIZE    ===>
  DSSIZE    ===>      (G)     COMPRESS ===>         LOCKPART   ===>
  MEMBER CLUSTER  ===>        LARGE    ===>         MAXROWS    ===>

    -- OR --

  VCAT     ===>
```

Database Name

Name of the database in which the tablespace will reside. If a database was provided on the Tablespaces Option screen, this field is prefilled.

Tablespace Name

Name of the tablespace you want to create. If a tablespace was provided on the Tablespaces Option screen, this field is prefilled.

STOGROUP

Specify the name of the set of volumes where the DB2 data for the tablespace is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

The amount of space in kilobytes allocated for the dataset where your tablespace will reside.

SECQTY

The amount of space in kilobytes allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicate whether the dataset is erased if it is dropped.

TRACKMOD

Indicate whether DB2 tracks modified pages in the space map pages of the tablespace or partition. Valid entries are: YES or NO. Do not specify this clause for LOB tablespace.

YES DB2 tracks changed pages in space map pages of tablespaces or partition.

NO DB2 does not track changed pages in the space map pages. It uses the LRSN value in each page to determine whether a page has been changed.

DSSIZE

A value in gigabytes that indicates the maximum size for each partition or, for LOB tablespaces, each dataset. If you specify DSSIZE, you must also specify NUMPARTS or LOB.

You can use the DSSIZE option to create a tablespace that holds up 16 terabytes of data. Leave blank and DB2 uses a default of 4 G. Valid entries are: blank and 1 through 64 (Gigabytes).

MEMBER CLUSTER

Specify how the data inserted by the INSERT statement is to be clustered.

NO (default). The data inserted by the INSERT statement is clustered by the implicit clustering index (the first index) or the explicit clustering index.

YES DB2 chooses where to locate the data in the tablespace based on available space when inserting data with the INSERT statement.

Note: Do not use MEMBER CLUSTER for a LOB tablespace or a tablespace in a work file database.

VCAT

VSAM catalog that contains the datasets for the tablespace. VCAT and STOGROUP are mutually exclusive.

FREEPAGE

Specify the rate at which a page is left free when the tablespace is loaded or reorganized. The default is zero.

PCTFREE

Enter the percentage of each page left free when the tablespace is loaded or reorganized. The default is 5.

NUMPARTS

Enter a value for a partitioned tablespace only. Specify the number of partitions allocated for the tablespace within the database. Valid values are 1-254. If you leave this field blank, the tablespace is nonpartitioned. NUMPARTS and SEGSIZE are mutually exclusive.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are ALL, CHANGED, NONE or SYSTEM.

LOCKMAX

Specify the maximum number of locks per user to acquire for the table or tablespace before escalating to the next locking level. 0 = lock escalation does not occur. n, where n > 0, = the number of locks (row or page locks for the table or tablespace) an application process can acquire before lock escalation occurs. Or enter SYSTEM to specify LOCKMAX SYSTEM.

COMPRESS

Indicates whether data compression applies to the rows of the tablespace or partition.

LARGE

YES or NO. YES identifies a tablespace as large. A large tablespace is a partitioned tablespace that can hold more than 64GB of data, either compressed or uncompressed. A large tablespace can have a maximum of 254 partitions with a maximum size of 4GB per partition. If LARGE is YES, NUMPARTS must be specified. If you enter NO for LARGE and the value for NUMPARTS is greater than 64, File-AID *for DB2* changes LARGE to YES.

Note: Do not specify LARGE if LOB or DSSIZE is specified.

BUFFERPOOL

Enter the name of the buffer pool associated with the tablespace.

LOCKSIZE

Required locking level for the tablespace. Valid entries are ANY, PAGE, TABLESPACE, and TABLE.

CLOSE

Indicate whether the datasets supporting the tablespace are closed when no one is using the tablespace.

DSETPASS

Password that the system passes to VSAM (virtual storage access method) when DB2 uses the datasets that support the tablespace.

SEGSIZE

Enter a value for a segmented tablespace only. Specify how many pages are assigned to each segment. This number must be a multiple of 4, with valid entries ranging from 4 to 64 (inclusive). Leave the field blank if you want a 0 SEGSIZE. SEGSIZE and NUMPARTS are mutually exclusive.

LOCKPART

YES or NO. Indicate whether selective partition locking (SPL) is to be used when locking a partitioned tablespace. If LOCKPART YES is specified and all conditions required for SPL are met, only the partitions accessed are locked. If LOCKPART YES is specified and all conditions required for SPL are not met, every partition of the tablespace is locked.

If you specify LOCKPART NO, selective partition locking is not used. The tablespace is locked with a single lock on the last partition. This has the effect of locking all partitions in the tablespace.

In order to alter the LOCKPART option, you must stop the entire tablespace with the STOP DATABASE command.

MAXROWS

Specify the maximum number of rows that DB2 considers placing on each data page. The value can range from 1 through 255. If you leave this field blank, 255 is the default.

You cannot specify the MAXROWS value for a tablespace in a work file database.

Partitioned Tablespace Create

The Partitioned Tablespace Create screen displays when you select Define Partitions from the Actions pull-down menu or enter PARTITN on the command line of the Tablespace Create screen. Figure 8-22 shows a sample screen. When you have defined the partitions, enter End to return to the Tablespace Create screen and enter the CREATE command or select Execute Create from the Actions pull-down menu to create the tablespace.

Figure 8-22. Partitioned Tablespace Create Screen

```
File-AID for DB2 ------- Partitioned Tablespace Create ------  ROW 1 TO 4 OF 4
COMMAND ===>                                                 SCROLL ====> PAGE

Database Name  : DBNAME
Tablespace Name: TSNAME

              VCAT     STOGROUP              ERASE FREE PCT  GBP      COM- TRK
         PART NAME     NAME     PRIQTY SECQTY RULE PAGE FREE CACHE   PRESS MOD
000001   1             STG001    3     3      NO   0    5    ALL      YES  YES
000002   2             STG001    3     3      NO   0    5    ALL      YES  YES
000003   3             STG001    3     3      NO   0    5    ALL      YES  YES
000004   4             STG001    3     3      NO   0    5    ALL      YES  YES
****************************** BOTTOM OF DATA ********************************
```

Database Name

Name of the database from the Tablespace Create screen.

Tablespace Name

Name of the tablespace from the Tablespace Create screen.

PART

Number of the partition for which you are specifying values.

VCAT

VSAM catalog that contains the datasets for the tablespace. VCAT and STOGROUP are mutually exclusive.

STOGROUP

Name of the set of volumes where the DB2 data for the tablespace is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the partition.

SECQTY

Amount of space (in kilobytes) allocated for the partition if it exceeds the primary quantity.

ERASE

Indicate whether the partition is erased if it is dropped.

FREEPAGE

Rate at which a page is left free when the partition or tablespace is loaded or reorganized.

PCTFREE

Percentage of each page left free when the partition or tablespace is loaded or reorganized.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are ALL, CHANGED, NONE, or SYSTEM.

COMPRESS

Indicates whether data compression applies to the rows of the tablespace or partition.

TRACKMOD

Indicate whether DB2 tracks modified pages in the space map pages of the tablespace or partition. Valid entries are: YES or NO. Do not specify this clause for LOB tablespace.

YES DB2 tracks changed pages in space map pages of tablespaces or partition.

NO DB2 does not track changed pages in the space map pages. It uses the LRSN value in each page to determine whether a page has been changed.

Tablespace Alter

The Tablespace Alter screen allows you to define the tablespace you want to change. File-AID *for DB2* automatically returns the appropriate Tablespace Alter screen (nonpartitioned or partitioned), depending on the tablespace you choose to alter.

The Tablespace Alter screen (Figure 8-23) displays when you choose **A** from the Tablespaces option screen and specify a nonpartitioned tablespace. When you have changed the tablespace, enter the ALTER primary command or select Execute Alter from the Actions pull-down menu to process your changes.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to alter the tablespace. Enter SQL on the command line after you have defined your tablespace and before you execute the ALTER primary command.

Figure 8-23. Non-Partitioned Tablespace Alter Screen

```
File-AID for DB2 ------------- Tablespace Alter ------------------- SSID: DSN
COMMAND ===>

Enter SQL to view/save the SQL source

Database Name  : DBNAME
Tablespace Name: TSNAME        Pagesize : 4

  STOGROUP  ===> STG003     FREEPAGE ===> 0          BUFFERPOOL ===> BP0
  PRIQTY(K) ===> 12         PCTFREE  ===> 5          CLOSE      ===> YES
  SECQTY(K) ===> 12         LOCKSIZE ===> ANY        DSETPASS   ===>
  ERASE     ===> NO         COMPRESS ===> NO         GBPCACHE   ===> CHANGED
  TRACKMOD  ===> YES        LOCKMAX  ===> 0            MAXROWS  ===> 0
  DSSIZE    ===>     (G)    LOCKPART ===> 0
   -- OR --
   VCAT     ===>
```

Database Name

Name of the database in which the tablespace resides. If a database was provided on the Tablespaces Option screen, this field is prefilled.

Tablespace Name

Name of the tablespace you want to alter. If a tablespace was provided on the Tablespaces Option screen, this field is prefilled.

Pagesize

Size in kilobytes of a page within the tablespace. This field is display only.

STOGROUP

Name of the set of volumes where the DB2 data for the tablespace is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

The amount of space in kilobytes allocated for the dataset where your tablespace resides.

SECQTY

The amount of space in kilobytes allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicate whether the dataset is erased if it is dropped.

TRACKMOD

Indicate whether DB2 tracks modified pages in the space map pages of the tablespace or partition. Valid entries are: YES or NO. Do not specify this clause for LOB tablespace.

YES DB2 tracks changed pages in space map pages of tablespaces or partition.

NO DB2 does not track changed pages in the space map pages. It uses the LRSN value in each page to determine whether a page has been changed.

DSSIZE

A value in gigabytes that indicates the maximum size for each partition or, for LOB tablespaces, each dataset. If you specify DSSIZE, you must also specify NUMPARTS or LOB.

You can use the DSSIZE option to create a tablespace that holds up 16 terabytes of data. Leave blank and DB2 will use a default of 4 G. Valid entries are: blank, 1 through 64 (Gigabytes).

VCAT

VSAM catalog that contains the datasets for the tablespace. VCAT and STOGROUP are mutually exclusive.

FREEPAGE

Rate at which a page is left free when the tablespace is loaded or reorganized.

PCTFREE

Percentage of each page left free when the tablespace is loaded or reorganized.

LOCKSIZE

Locking level for the tablespace. Valid entries are ANY, PAGE, TABLESPACE, and TABLE.

COMPRESS

Indicates whether data compression applies to the rows of the tablespace or partition.

LOCKMAX

Specify the maximum number of locks per user to acquire for the table or tablespace before escalating to the next locking level. 0 = lock escalation does not occur. n,

where n > 0, = the number of locks (row or page locks for the table or tablespace) an application process can acquire before lock escalation occurs. Or enter SYSTEM to specify LOCKMAX SYSTEM.

LOCKPART

YES or NO. Indicate whether selective partition locking (SPL) is to be used when locking a partitioned tablespace. If LOCKPART YES is specified and all conditions required for SPL are met, only the partitions accessed will be locked. If LOCKPART YES is specified and all conditions required for SPL are not met, every partition of the tablespace is locked.

If you specify LOCKPART NO, selective partition locking is not used. The tablespace is locked with a single lock on the last partition. This has the effect of locking all partitions in the tablespace.

In order to alter the LOCKPART option, you must stop the entire tablespace with the STOP DATABASE command.

BUFFERPOOL

Name of the bufferpool associated with the tablespace.

CLOSE

Indicates supporting the tablespace are closed when no one is using the tablespace.

DSETPASS

Password that the system passes to VSAM (virtual storage access method) when DB2 uses the datasets that support the tablespace.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are ALL, CHANGED, NONE or SYSTEM.

MAXROWS

Specify the maximum number of rows that DB2 will consider placing on each data page. The value can range from 1 through 255. 255 is the default if you leave this field blank.

The change takes effect immediately for new rows added. However, the space class settings for some pages may be incorrect and could cause unproductive page visits. It is highly recommended to reorganize the tablespace after altering MAXROWS.

You cannot alter the MAXROWS value of a tablespace in a work file database or for the following tablespaces:

- DSNDB06.SYSDBASE
- DSNDB06.SYSDBAUT
- DSNDB06.SYSGROUP
- DSNDB06.SYSPLAN
- DSNDB06.SYSVIEWS

Partitioned Tablespace Alter

The Tablespace Alter screen (Figure 8-24) displays when you choose **A** from the Tablespaces option screen and you specify a partitioned tablespace.

Figure 8-24. Partitioned Tablespace Alter Screen

```
File-AID for DB2 ------- Partitioned Tablespace Alter -------  ROW 1 TO 4 OF 4
COMMAND ===>                                                  SCROLL ===> PAGE

Database Name  : DBNAME
Tablespace Name: TSNAME        Pagesize : 4

LOCKSIZE ===> PAGE             BUFFERPOOL ===> BP0      DSETPASS ===>
CLOSE    ===> NO               LOCKMAX    ===> SYSTEM
                                                    PCT   GBP     COM-  TRK
PART  VCAT     STOGROUP PRIQTY SECQTY ERASE FREEPAGE FREE  CACHE   PRESS MOD
 1             DSN8G610 96     20            0        5     CHANGED  YES  YES
 2             DSN8G610 96     20            0        5     CHANGED  YES  YES
 3             DSN8G610 48     12            0        5     CHANGED  NO   YES
 4             DSN8G610 96     20            0        5     CHANGED  YES  YES
******************************** BOTTOM OF DATA ********************************
```

Database Name

Name of the database you entered on the Tablespaces option screen.

Tablespace Name

Name of the tablespace you entered on the Tablespaces option screen.

BUFFERPOOL

Name of the bufferpool associated with the tablespace.

PAGESIZE

Size in kilobytes of a page within the tablespace. This field may not be updated.

LOCKSIZE

Locking level for the tablespace. Valid entries are ANY, PAGE, TABLESPACE, and TABLE.

DSETPASS

Password that the system passes to VSAM (virtual storage access method) when DB2 uses the datasets that support the tablespace.

CLOSE

Indicates whether the datasets supporting the tablespace are closed when no one is using the tablespace.

LOCKMAX

Specify the maximum number of locks per user to acquire for the table or tablespace before escalating to the next locking level. 0 = lock escalation does not occur. n, where n > 0, = the number of locks (row or page locks for the table or tablespace) an application process can acquire before lock escalation occurs. Or enter SYSTEM to specify LOCKMAX SYSTEM.

PART

Partition number of the partitioned tablespace. This field may not be updated.

VCAT

VSAM catalog that contains the datasets for the tablespace. VCAT and STOGROUP are mutually exclusive.

STOGROUP

Name of the volume set where DB2 data for the tablespace is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the partition.

SECQTY

Amount of space (in kilobytes) allocated for the partition if it exceeds the primary quantity.

ERASE

Indicate whether the partition is erased if it is dropped.

FREEPAGE

Rate a page is left free when the partition is loaded or reorganized.

PCTFREE

Percentage of each page left free when the tablespace is loaded or reorganized.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are ALL, CHANGED, NONE, or SYSTEM.

COMPRESS

Indicates whether data compression applies to the rows of the tablespace or partition.

TRACKMOD

Indicate whether DB2 tracks modified pages in the space map pages of the tablespace or partition. Valid entries are: YES or NO. Do not specify this clause for LOB tablespace.

YES DB2 tracks changed pages in space map pages of tablespaces or partition.

NO DB2 does not track changed pages in the space map pages. It uses the LRSN value in each page to determine whether a page has been changed.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Tablespaces option screen. It gives you the opportunity to confirm whether you want to drop the tablespace you have specified. This is a safety feature intended to prevent you from dropping a tablespace unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 3, Indexes

The Indexes option screen displays when you select option **3** from the Object Selection menu. It lets you specify whether you want to create, alter, or drop an index. Figure 8-25 shows a sample screen.

Figure 8-25. Indexes Option Screen

```
  Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                          Indexes
OPTION  ===>                                                 SSID ===> DSN

   C - Create INDEX
   A - Alter  INDEX
   D - Drop   INDEX

Enter:
   Creator        ===> USERID
   Index Name     ===> EMP

Table:
   Creator        ===> USERID
   Table Name     ===> EMPNO
```

OPTION

Specify the letter that represents the activity you want to pursue.

C

Create an index. (Displayed in Figure 8-26 on page 8-35.)

A

Alter an index. (Displayed in Figure 8-29 on page 8-39 or Figure 8-30 on page 8-41.)

D

Drop an index. A screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23) displays.

Creator

Specify your creator ID. (Required if you are altering or dropping an index; optional if you are creating one.)

Index Name

Name of the index you want to create, alter, or drop.

Table Area

Specify only when you are creating an index.

Creator

ID of the user who created the table on which the index is based. If this information is in your user profile (see Chapter 5, "Defaults and User Parameters" for details), the field is prefilled.

Table Name

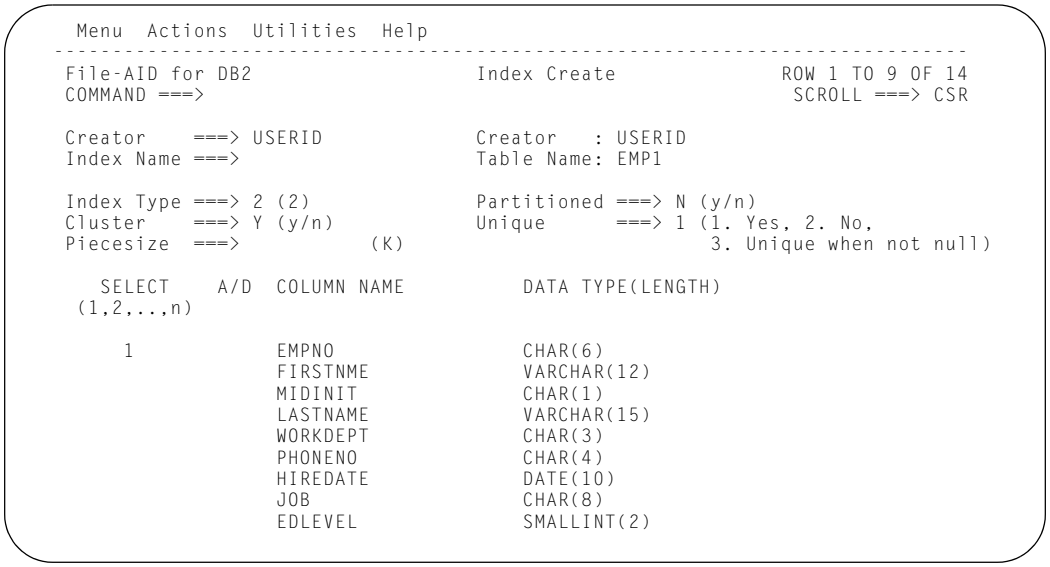
Name of the table on which the index is based. If you have just finished creating the table, the field is prefilled.

Index Create

The Index Create screen displays when you choose **C** from the Indexes option screen. This screen is also displayed when you press Enter on the Table Create screen (Figure 8-4 on page 8-4) if an index is required. It lets you define the index you want to create. When you have defined your index, either select Execute Create from the Actions pull-down menu or enter the CREATE primary command to create the index.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the index. Enter SQL on the command line after you have defined your index and before you execute the CREATE primary command.

Figure 8-26. Index Create screen



Creator (Index)

Your user ID. If this information was provided on the Indexes option screen, this field is prefilled with the value.

Index Name

Name of the index you want to create. If this information was provided on the Indexes option screen, this field is prefilled with the value. You can change the index name by typing over it on this screen.

Creator (Table)

ID of person who created the table on which the index is based. This field is prefilled with the information provided on the Indexes option screen and protected from update.

Table Name

Name of the table on which the index is based. This field is prefilled with the information provided on the Indexes option screen protected from update.

Index Type

DB2 can create only type 2 indexes.

Cluster

Indicate whether the index is designated as clustering.

Piecesize

Indicate maximum size of dataset storage piece (in kilobytes (K) that is used by DB2 for nonpartitioned indexes. You must enter the value in kilobytes. Megabytes and gigabytes are only referenced for your convenience. Allowable values are:

- 256 K
- 512 K
- 1024 K (or 1 M)
- 2024 K (or 2 M)
- 4096 K (or 4 M)
- 8192 K (or 8 M)
- 16384 K (or 16 M)
- 32768 K (or 32 M)

- 65536 K (or 64 M)
- 131072 K (or 128 M)
- 262144 K (or 256 M)
- 524288 K (or 512 M)
- 1048576 K (or 1024 M or 1 G)
- 2097152 K (or 2048 M or 2 G)
- 4194304 K (or 4096 M or 4 G) with LARGE, DSSIZE 4 G (or greater)
- 8388608 K (or 8192 M or 8 G) with DSSIZE 8 G (or greater)
- 16777216 K (or 16384 M or 16 G) with DSSIZE 16 G (or greater)
- 33554432 K (or 32768 M or 32 G) with DSSIZE 32 G (or greater)
- 67108864 K (or 65536 M or 64 G) with DSSIZE 64 G.

The default for piecesize is 2 G (2 GB) for indexes backed by non-LARGE tablespaces and 4 G (4 GB) for indexes backed by LARGE or DSSIZE tablespaces.

Partitioned

Indicate whether the indexspace for the index is partitioned.

Unique

Indicate whether more than one row in the indexed base table can have identical values. This field defaults to **2** (NO).

1	**Yes**	All rows must be unique.
2	**No**	Rows can have identical values.
3	**Unique when not null**	Null rows can have identical values, all others must be unique. Option 3, Unique where not null is only valid for Index Type 2.

Select

Specify which column(s) in the table is the key, indicating the order of the keys (values 1, 2, 3, and so forth) if there is more than one.

A/D

Specify whether the index is in ascending or descending order within the column. The default is ascending.

Column Name

Names of the table columns on which the index is based. Column names cannot be modified.

Data Type(Length)

Datatypes and lengths of the table columns the index is based on. Datatypes cannot be modified.

Index Create Values

The Index Create Values screen displays if you select **N** for the PARTITIONED field on the Index Create screen and either select Define Index Values from the Actions pull-down menu or enter VALUES on the command line. It lets you create a nonpartitioned index. You can specify STOGROUP or VCAT information, but not both.

Note: This screen displays regardless of whether you are creating a nonpartitioned or partitioned index. If you are creating a nonpartitioned index, enter the values for your nonpartitioned index on this screen. If you are creating a partitioned index, select Define Partitions from the Actions pull-down menu or enter PARTITN on the command line. The Index Create Partition screen (Figure 8-28 on page 8-38) displays.

Figure 8-27. Index Create Values screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                  Index Create Values
COMMAND ===>

Creator   : USERID
Index Name: IDXEMP

   STOGROUP ===> STGP          FREEPAGE ===>            BUFFERPOOL ===> BPO
  PRIQTY(K) ===>               PCTFREE  ===> 25         DSETPASS   ===>
  SECQTY(K) ===>               GBPCACHE ===>            CLOSE      ===> YES
     ERASE  ===> NO            SUBPAGES ===> 4          COPY       ===>

   -- OR --

   VCAT     ===>
```

Creator

ID of the person creating the index.

Index Name

Name of the index being created.

STOGROUP

Name of the set of volumes where the DB2 data for the index is stored. STOGROUP and VCAT are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the dataset where your index resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicate whether the dataset is erased if it is dropped.

VCAT

VSAM catalog that contains the datasets for the index. STOGROUP and VCAT are mutually exclusive.

FREEPAGE

Rate at which a page is left free when the index is loaded or reorganized.

PCTFREE

Percentage of each page left free when the index is loaded or reorganized.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are **NONE.**

SUBPAGES

Number of subpages for each page.

BUFFERPOOL

Name of the buffer pool associated with the index.

DSETPASS

Password the system passes to VSAM (virtual storage access method) when DB2 uses the datasets that support the index.

CLOSE

Specify whether the datasets supporting the index are closed when no users are using the index.

COPY

Indicate whether the COPY utility is allowed for the index.

NO Does not allow full image or concurrent copies or the use of the RECOVER utility on the index. NO is the default.

YES Allows full image or concurrent copies and the use of the RECOVER utility on the index.

Index Create Partition

The Index Create Partition screen displays if you select **Y** at the PARTITIONED field of the Index Create screen and either select Define Partitions from the Actions pull-down menu or enter the PARTITN command at the Index Create Values screen. It lets you create a partitioned index by specifying partitioned index values. You can also override any value(s) specified on an individual partition level, independently of any others.

Figure 8-28. Index Create Partition Screen

```
  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2                 Index Create Partition         ROW 1 TO 3 OF 3
COMMAND ===>                                                   SCROLL ===> PAGE


Creator   : USERID
Index Name: EMPIDX

PARTITION  VCAT      STOGROUP  PRIQTY(K) SECQTY(K) ERASE  FREEPAGE  PCTFREE
   1
GBPCACHE  ===>         Values  ===>
   2
GBPCACHE  ===>         Values  ===>
   3
GBPCACHE  ===>         Values  ===>
******************************** BOTTOM OF DATA ********************************
```

Creator

ID of the person creating the index.

Index Name

Name of the index.

PARTITION

Number of the partition you are defining values for. There are 2 input lines for each partition

VCAT

VSAM catalog to contain the datasets for the index. VCAT and STOGROUP are mutually exclusive.

STOGROUP

Name of the set of volumes where the DB2 data for the index is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the dataset where your index resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicate whether the dataset is erased if it is dropped.

FREEPAGE

Rate at which a page is left free when the index is created.

PCTFREE

Percentage of each page left free when the index is created.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are NONE.

VALUES

Highest allowable value of the index columns for each individual partition.

Index Alter

An Index Alter screen displays when you enter **A** on the Indexes option screen. It lets you change an existing index. When you complete your changes, either select Execute Alter from the Actions pull-down menu or enter the ALTER primary command to effect those changes.

There are two different Index Alter screens. One is for nonpartitioned indexes, the other is for partitioned indexes. File-AID *for DB2* automatically returns the appropriate one, based on the index type you are altering.

Index Alter for Nonpartitioned Indexes

The Index Alter screen for nonpartitioned indexes displays if you have named a nonpartitioned index.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to alter the index. Enter SQL on the command line after you have defined your index and before you execute the ALTER primary command.

Figure 8-29. Index Alter screen for Nonpartitioned Indexes

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                        Index Alter                 SSID: DSN
COMMAND ===>

Creator   : TSOID01           Index Type ===> 2  (2)
Index Name: PLAN_TABLE_IX       Subpages ===>    (1, 2, 4, 8, or 16)
    STOGROUP ===> SGO            FREEPAGE ===> 0          BUFFERPOOL ===> BP0
   PRIQTY(K) ===> 3              PCTFREE  ===> 0          CLOSE      ===> NO
   SECQTY(K) ===> 3              ERASE    ===> NO         DSETPASS   ===>
   GBPCACHE  ===> CHANGED                                 COPY       ===> YES
   PIECESIZE ===> 2097152      (K)

   -- OR --

   VCAT     ===>
```

Creator

ID of the person who created the index.

Index Type

DB2 can create only type 2 indexes.

Index Name

Name of the index.

SUBPAGES

Number of subpages for each page.

STOGROUP

Name of the set of volumes where the DB2 data for the index is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the dataset where your index resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entry is NONE.

Piecesize

Indicate maximum size of dataset storage piece (in kilobytes (K) that is used by DB2 for nonpartitioned indexes. You must enter the value in kilobytes; Megabytes and Gigabytes are only referenced for your convenience. Allowable values are:

- 256 K
- 512 K
- 1024 K (or 1 M)
- 2024 K (or 2 M)
- 4096 K (or 4 M)
- 8192 K (or 8 M)
- 16384 K (or 16 M)
- 32768 K (or 32 M)
- 65536 K (or 64 M)
- 131072 K (or 128 M)
- 262144 K (or 256 M)
- 524288 K (or 512 M)
- 1048576 K (or 1024 M or 1 G)
- 2097152 K (or 2048 M or 2 G)
- 4194304 K (or 4096 M or 4 G) with LARGE, DSSIZE 4 G (or greater)
- 8388608 K (or 8192 M or 8 G) with DSSIZE 8 G (or greater)
- 16777216 K (or 16384 M or 16 G) with DSSIZE 16 G (or greater)
- 33554432 K (or 32768 M or 32 G) with DSSIZE 32 G (or greater)
- 67108864 K (or 65536 M or 64 G) with DSSIZE 64 G.

The default for piecesize is 2 G (2 GB) for indexes backed by non-LARGE tablespaces and 4 G (4 GB) for indexes backed by LARGE or DSSIZE tablespaces.

Note: The value of zero (0) indicates that this index is a partitioning index.

VCAT

VSAM catalog containing the datasets for the index. VCAT and STOGROUP are mutually exclusive.

FREEPAGE

Rate at which a page is left free when the index is loaded or reorganized.

PCTFREE

Percentage of each page left free when the index is loaded or reorganized.

ERASE

Indicate whether the dataset is erased if it is dropped.

BUFFERPOOL

Name of the bufferpool associated with the index.

CLOSE

Indicate whether the datasets supporting the index are closed when no one is using the index.

DSETPASS

Password that the system passes to VSAM (virtual storage access method) when DB2 uses the datasets supporting the index.

COPY

Indicate whether the COPY utility is allowed for the index.

NO Does not allow full image or concurrent copies or the use of the RECOVER utility on the index. NO is the default.

YES Allows full image or concurrent copies and the use of the RECOVER utility on the index.

Index Alter for Partitioned Indexes

The Index Alter screen for partitioned indexes displays automatically if you name a partitioned index for altering.

Figure 8-30. Index Alter screen for Partitioned Indexes

```
Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                 Index Alter                   Row 1 to 4 of 4
COMMAND ===>                                                 SCROLL ===> PAGE

Creator   : DSN8610
Index Name: XEMP1

BufferPool ===> BP0        Index Type ===> 2   (1 or 2)
Close      ===> NO         Subpages   ===>     (1, 2, 4, 8, or 16)
DsetPass   ===>            Copy       ===> NO

PARTITION  VCAT      STOGROUP  PRIQTY(K) SECQTY(K) ERASE  FREEPAGE  PCTFREE
   1               DSN8G610  12        12                          10
Values: '099999'                                    GBPCACHE ===> CHANGED
   2               DSN8G610  12        12                          10
Values: '199999'                                    GBPCACHE ===> CHANGED
   3               DSN8G610  12        12                          10
Values: '299999'                                    GBPCACHE ===> CHANGED
   4               DSN8G610  12        12                          10
Values: '999999'                                    GBPCACHE ===> CHANGED
****************************** Bottom of data ********************************
```

Creator

ID of the person who created the index.

Index Name

Name of the index.

BUFFERPOOL

Name of the bufferpool associated with the index.

CLOSE

Indicates whether the datasets supporting the index are closed when no one is using the index.

DSETPASS

Password the system passes to VSAM (virtual storage access method) when DB2 uses the datasets supporting the index.

Index Type

DB2 can create only type 2 indexes.

SUBPAGES

Number of subpages for each page.

COPY

Indicate whether the COPY utility is allowed for the index.

NO Does not allow full image or concurrent copies or the use of the RECOVER utility on the index. NO is the default.

YES Allows full image or concurrent copies and the use of the RECOVER utility on the index.

PARTITION

Number of the partition for which the values are defined.

VCAT

VSAM catalog that contains the datasets for the index. VCAT and STOGROUP are mutually exclusive.

STOGROUP

Name of the set of volumes where the DB2 data for the index is stored. VCAT and STOGROUP are mutually exclusive.

PRIQTY

Number of kilobytes allocated for the dataset where your index resides.

SECQTY

Amount of space (in kilobytes) allocated for the dataset if it exceeds the primary quantity.

ERASE

Indicates whether the dataset is erased if it is dropped.

FREEPAGE

Rate at which a page is left free when the index is loaded or reorganized.

PCTFREE

Percentage of each page left free when the index is loaded or reorganized.

VALUES

Highest value in the corresponding partition of the tablespace. This field cannot be altered.

GBPCACHE

Specify how the group buffer pool should be used for the index to support inter-DB2 sharing in a data sharing environment. Valid entries are NONE.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Indexes option screen. It gives you the opportunity to confirm whether you want to drop the index you have specified. This is a safety feature intended to prevent you from dropping an index unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 4, Views

The Views option screen displays when you select option **4** from the Object Selection menu. It lets you specify whether you want to create or drop a view. Figure 8-31 shows a sample screen.

Figure 8-31. Views Option Screen

```
  Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                                Views
OPTION  ===>                                                  SSID ===> DSN

   C - Create VIEW
   D - Drop   VIEW

Enter:
   Creator       ===> USERID
   View Name     ===> EMP_VIEW

For Table:
   Creator       ===> USERID
   Table Name    ===> EMP
```

OPTION

Specify the letter representing the activity you want to pursue:

- **C** Create a view (Figure 8-32 on page 8-44).
- **D** Drop a view. A screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23) displays.

Creator

ID of the person creating the index.

View Name

Name of the view you want to create or drop.

For Table Area

Creator

ID of person who created the table for which you are creating or dropping a view.

Table Name

Name of the table for which you are creating or dropping a view.

View Create

The View Create screen displays when you enter **C** on the Views Option screen. It lets you define the view you want to create. Once you define your view, either press select Execute Create from the Actions pull-down menu or enter the CREATE primary command to create the view.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the view. Enter SQL on the command line after you have defined your view and before you execute the CREATE primary command.

Figure 8-32. View Create Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                        View Create           ROW 1 TO 12 OF 14
COMMAND ===>                                                   SCROLL ===> PAGE
                                                                     SSID: DSN
View  Name: USERID.EMP_VIEW
Table Name: USERID.EMP

With check option ===> N      (Y/N/C - Cascade/L - Local)

SEL   COLUMN NAME        TYPE(LEN)           WHERE
---   ------------------ ------------------- ----+----10---+----20---+----30----
+
4     EMPNO              CHAR(6)             > '000100' AND
2     FIRSTNAME          VARCHAR(12)
3     MIDINIT            CHAR(1)
1     LASTNAME           VARCHAR(15)         > 'H'
S     WORKDEPT           CHAR(3)
S     PHONENO            CHAR(4)
      HIREDATE           DATE
      JOB                CHAR(8)
      EDLEVEL            SMALLINT
      SEX                CHAR(1)
      BIRTHDATE          DATE
      SALARY             DECIMAL(9,2)
```

View Name

Name of the view you want to create. If this information was provided on the Views Option screen, this field is prefilled when the screen is invoked.

Table Name

Name of the table for which you are creating a view. If this information was provided on the Views Option screen, this field is prefilled when the screen is invoked.

WITH CHECK OPTION

Indicates whether updates or inserts that create rows outside of the view's WHERE clauses are disallowed.

Y

Enforce check option.

N

No check option

L

Enforce Local check option. Search conditions of underlying views are checked conditionally, that is, only if the underlying view is defined with a check option or if it is an underlying view of a view that is defined with a check option.

C

Enforce Cascaded check option. All search conditions of underlying views are checked unconditionally, regardless of how those underlying views are defined.

SEL

Specify which column or columns you want included in the view. If you want multiple columns, the order of the columns should also be indicated in the entry (1, 2, 3, and so on). Single columns can also be selected by entering **S** beside each one. Mixing multiple and single selections results in the multiple selections being displayed first, followed by the single selections. If you do not indicate any specific columns for selection, all columns are included in the view.

COLUMN NAME

Names of the columns in the table for which you are creating a view. This field cannot be modified.

TYPE (LENGTH)

Types and lengths of the values in the columns. This field cannot be modified.

WHERE

WHERE information for the view. WHERE clauses may be scrolled left and right.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Views option screen. It gives you the opportunity to confirm whether you want to drop the view you have specified. This is a safety feature intended to prevent you from dropping a view unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 5, Synonyms

The Synonyms option screen displays when you select option **5** from the Object Selection menu. It lets you specify whether you want to modify, create, or drop a synonym.

Figure 8-33. Synonyms Option Screen

```
  Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                  Synonyms
OPTION  ===>                                                 SSID ===> D610

   Blank - Modify Synonyms (Use  *  or  pattern  for list)
   C - Create Synonym
   D - Drop Synonym

Specify DB2 synonym (for options C and D)
   Synonym Name     ===> *
   Effective SQLID  ===> USERID
```

OPTION

Specify the letter representing the activity you want to perform.

Blank

Change existing synonyms. If you choose this option, a list of synonyms displays (Figure 8-34 on page 8-46).

C

Create a synonym. (Figure 8-35 on page 8-47).

D

Drop a synonym. A screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23) displays.

Specify DB2 Synonym Area

Synonym Name

Name of the synonym you want to create, drop, or modify. An asterisk (*****) or a pattern such as **BFH*,P124467_** are allowed when modifying synonyms. This field is required.

Effective SQLID

SQLID for the synonym name. This may be the same as the user ID. An asterisk (*****) or a pattern such as **BFH*,P124467_** are allowed when modifying synonyms. This field is required.

Modify Synonyms

The Modify Synonyms screen displays if you leave the OPTION field blank on the Synonyms option screen. It provides a list of synonyms for modification. If you entered a pattern such as **BFHRW*.P124467*** on the Synonyms option screen, a synonym with the creator BFHRW* and name P124467* is displayed. Use the line commands I (Insert), D (Delete), or R (Repeat) to make any changes, and then select Execute Modify from the Actions pull-down menu or enter the MODIFY primary command.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to modify the synonym. Select View/Edit DDL from the Actions pull-down menu or enter SQL on the command line after you have modified your synonym and before you execute the MODIFY primary command.

Figure 8-34. Modify Synonyms Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                    Modify Synonyms            ROW 1 TO 4 OF 4
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: DSN
         SYNONYM Name          For    Creator   TABLE Name
         ------------------           --------  ------------------
000001   EMPLOY                      USERID    EMP
000002   SQLS                        USERID    SQL
000003   SQLSXX                      USERID    SQL
000004   STUFF                       USERID    AREA
********************************** BOTTOM OF DATA ****************************
```

SYNONYM Name

Name of the synonym to be edited.

Creator

ID of person who created the table related to the synonym. This field defaults to the creator ID as entered in User Parms, Option 0.

TABLE Name

Name of the table or view related to the synonym being editing.

Synonym Create

To create a synonym, enter **C** in the OPTION field of the Synonyms option screen. Complete the required fields—SYNONYM NAME, EFFECTIVE SQLID—and press Enter.

The Create Synonym screen, (Figure 8-35), displays. Type entries for the CREATOR and TABLE NAME fields, and press Enter or select Execute Create from the Action pull-down menu. If the synonym is created, you are returned to the Synonyms option screen with the message “Synonym Created” on the title line.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the synonym. Enter SQL on the command line or select View/Edit DDL from the Action pull-down menu after you have entered the FOR table name and before you press Enter.

Figure 8-35. Synonym Create Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2               Create Synonym
OPTION  ===>                                                 SSID ===> DSN


Synonym Name   : SDEPT
Effective SQLID: USERID

Specify "FOR" Table Name:
   Creator     ===> USERID
   Table Name  ===> DEPT
```

Synonym Name

Displays the name of the synonym to be created. This field cannot be changed.

Effective SQLID

Displays the SQLID for the synonym name. May be the same as the user ID. This field cannot be changed.

Specify “FOR” Table Name Area

Creator

ID of the table’s creator.

Table Name

Name of the table or view related to the synonym.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Synonyms option screen. It gives you the opportunity to confirm whether you want to drop the synonym you have specified. If the synonym is to be dropped, press Enter. To cancel the drop process, enter the END primary command. This is a safety feature intended to prevent you from dropping a synonym unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 6, Aliases

The Aliases option screen (Figure 8-36) displays when option **6** is selected on the Object Selection menu. It allows you to create, modify, or drop an alias.

Figure 8-36. Aliases Option Screen

```
  Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                Aliases
OPTION  ===>                                                 SSID ===> D610

   Blank - Modify Aliases (Use  *  or  pattern  for list)
   C - Create Alias
   D - Drop Alias

Specify DB2 alias (for options C and D)
   Creator      ===> USERID
   Alias Name   ===> *
```

OPTION

Specify the letter of the required option.

Blank

Permits modification of an existing alias. The screen shown in Figure 8-37 on page 8-49 displays.

C

Create an alias. The Alias Create screen (Figure 8-38 on page 8-49) displays.

D

Drop an alias. Displays a screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23).

Specify DB2 Alias Area

Creator

Creator of the table to which the alias is related. An asterisk (*****) or a pattern such as **BFH*,P124467_** are allowed when modifying aliases. This field is required.

Alias Name

Name of the alias to be created or dropped. The alias name can be up to eighteen characters in length. This may be the same as the user ID. An asterisk (*****) or a pattern such as **BFH*,P124467_** are allowed when modifying aliases. This field is required.

Modify Aliases

The Modify Aliases screen (Figure 8-37 on page 8-49) displays when the OPTION field is left blank on the Aliases Option screen. It displays a list of aliases that can be modified. If you entered a pattern such as '**BFHRW*.P124467***' on the Aliases option screen, an alias with the creator BFHRW* and name P124467* is displayed. Use the insert (I), delete (D), or repeat (R) line commands to make the desired changes. Once all changes have been made, select Execute Modify from the Actions pull-down menu or enter the MODIFY command on the primary command line to modify the aliases.

Once the aliases have been modified, the following message displays in the upper right corner of the screen:

```
ALIASES Modified
```

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to modify the alias. Select View/Edit DDL from the Actions pull-down menu or enter SQL on the command line after you have entered your changes and before you execute the MODIFY command.

Figure 8-37. Modify Aliases Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                  Modify Aliases              ROW 1 TO 5 OF 5
COMMAND ===>                                                  SCROLL ===> CSR
                                                                    SSID: DSN
Use SQL to view/save the SQL source

       ALIAS Name         For  Location            Creator   TABLE Name
       ------------------      ------------------  --------  ------------------
000001 D                                          USERID    DEPT
000002 SQLIZER                 SAN FRANCISCO      USERID    TSQLSAMPLE
000003 TEST                    COMPUWARE          USERID    EMP
000004 VSQLSAMP                                   USERID    TSQLSAMPLE
000005 VXXXSAMP                ALASKA             USERID    TESTSAMPLE
********************************** BOTTOM OF DATA ****************************
```

ALIAS Name

Name of the alias to be modified, created, or dropped.

Location

Location for which the alias is being defined. The location must be defined in the SYSIBM.SYSLOCATIONS table or an error occurs.

Creator

ID of the individual that created the table.

TABLE Name

Name of the table or view related to the alias being edited.

Alias Create

The Alias Create screen (Figure 8-38) displays when option **C** is selected on the Aliases Option screen and after the alias name to be created has been entered.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the alias. Select View/Edit DDL from the Actions pull-down menu or enter SQL on the command line after you have entered the FOR table name and before you press Enter.

Figure 8-38. Alias Create Screen

```
  Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2                Alias Create                      SSID: DSN
OPTION  ===>

Alias Name: USERID.TEST_ALIAS

Specify "FOR" Table Name:
   Location    ===> ENGLAND                  (blank for LOCAL Subsystem)
   Creator     ===> USERID
   TABLE Name  ===> DEPT
```

Alias Name

Alias name to be created. This field contains the name of the alias that was entered on the Aliases Option screen. The data is highlighted and protected. In addition, the data cannot be changed on this screen.

Specify "FOR" Table Name Area

Location

Location for which the alias is being defined. If the subsystem is local, this field can be left blank.

Creator

Creator ID of the table on which the alias is to be created. This value is extracted from the CREATOR field on the Aliases Option screen and can be changed. This field cannot be left blank.

Table Name

Name of the table or view for which the alias is being created. This field cannot be left blank.

Confirm Drop

The Confirm Drop screen displays when the drop option is selected on the Aliases Option screen and after the name of the existing alias is entered. If the alias is to be dropped, press Enter. To cancel the drop process, enter the END primary command. This is a safety feature intended to prevent you from dropping alias unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 7, Distinct Types

The Distinct Types option screen (Figure 8-39) displays when option 7 is selected on the Object Selection menu. It allows you to create or drop a DISTINCT (or user defined) column datatype.

Figure 8-39. Distinct Type Option Screen

```
Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2             User Defined Type
OPTION  ===>                                                 SSID ===> DSN

   C - Create User Defined Type
   D - Drop   User Defined Type

Enter:
   Schema       ===> TSOID01
   UDT Name     ===> USDOLLARS
```

OPTION

Specify the letter of the required option:

C Create a distinct type. The screen shown in Figure 8-40 on page 8-51 displays.

D Drop a distinct type. Displays a screen similar to the Confirm Drop screen (Figure 8-18 on page 8-23).

Enter Area

Schema

Enter the name of the schema or owner of the distinct type to be created or dropped. This field defaults to the Current User ID.

Distinct Type Name

Name of the distinct type to be created or dropped. This field is required when the create (C) or drop (D) options are selected. The distinct type name can be up to eighteen characters in length.

Distinct Type Create

The Distinct Type Create screen (Figure 8-40), displays when option C is selected on the Distinct Type Option screen and after the distinct type name to be created has been entered.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to create the distinct type. Select View/Edit DDL from the Actions pull-down menu or enter SQL on the command line after you have entered the distinct type name and before you press Enter.

Figure 8-40. Distinct Type Create Screen

```
Menu  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2             Distinct Type Create                  SSID: DSN
OPTION  ===>

Schema        ===> TSOID01
Distinct Type ===> USDOLLARS

 Source:

  Schema SYSIBM                    Length     Scale    Unit (for LOB datatypes)
  Type          ===> DEC          => 9      => 2    =>
  For           ===>          Data (Bit, SBCS, Mixed)
  CCSID         ===> EBCDIC   (EBCDIC OR ASCII)
```

Schema

Enter the name of the schema or owner of the distinct type to be created. This field defaults to the current userID.

Distinct Type

Enter the name of the distinct type to be created. The distinct type name must be a valid object name (18 characters or fewer) and NOT be the name of a built-in datatype.

Source Area

Schema

Always prefilled with SYSIBM. Identifies the schema for the built-in datatype.

Type

Enter one of the built-in (standard) datatypes -- Integer, Char, Smallint, Varchar, Date, etc. that is the underlying column datatype for your distinct type.

Length

If the source datatype is Char, Varchar, Decimal, Float, Double, Graphic, Vargraphic, or one of the LOB datatypes, enter the length value of the source datatype.

Scale

If the source datatype is Decimal, enter the desired scale value.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

For

If the distinct type is sourced on a character string datatype, the FOR clause indicates the subtype (Bit, SBCS, Mixed).

CCSID

If the distinct type is sourced on a string datatype, the CCSID clause indicates whether the encoding scheme of the data is ASCII or EBCDIC.

Confirm Drop

The Confirm Drop screen displays when the drop option is selected on the Distinct Types Option screen and after the name of the existing distinct type is entered. If the distinct type is to be dropped, press Enter. To cancel the drop process, enter the END primary command. This is a safety feature intended to prevent you from dropping a distinct type unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 8, User Defined Functions

There are many rules for handling User Defined Functions (UDFs). You can see them in the SQL Reference Guide, or on the web at http://www-3.ibm.com/software/data/db2/os390/v7books.html.

The User Defined Function screen (Figure 8-41) displays when option 8 is selected on the Object Selection Menu. From the User Defined Function screen, you can create an external scalar, external table, or sourced function.

External Scalar

A function which returns a single value.

External Table

A function which returns rows of results.

Sourced

A function based on a previously defined function, either built-in or user-defined.

Figure 8-41. User Defined Function Screen

```
Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2            User Defined Function
OPTION  ===>                                                SSID  ===>
                                                            SQLID ===>
   C - Create  Function    (Select type)
           _ External Scalar    _ External Table    _ Sourced

   A - Alter   Function
   D - Drop    Function

Enter:                Use * or pattern for selection list for DROP or ALTER

   Schema Name   ===>
   Function Name ===>
   Specific Name ===>                      (Required for DROP or ALTER)
```

Option

Enter either C(reate), A(lter), or D(rop). Enter only the first letter of each command (C, A, or D).

Schema Name

The name of the schema in which the new UDF should be created. This field defaults to the SQL authorization ID if no schema is entered. Must not begin with 'SYS' unless it is 'SYSADMIN' or 'SYSPROC'.

Function Name

Name of the function as it is called by SQL statements.

Specific Name

Specific name of the UDF. If you do not provide a name, DB2 generates a unique name in the form SQLxxxxxxxxxxxx.

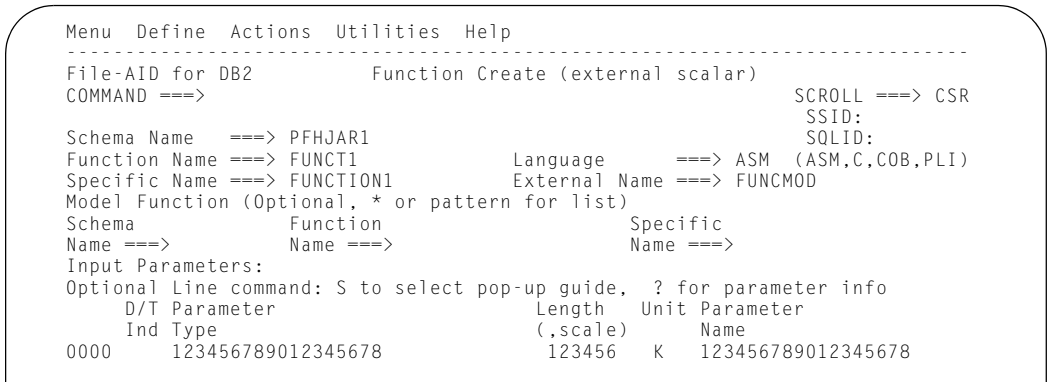
Create External Scalar

The Function Create (external scalar) screen (Figure 8-42) displays when option C for function type **External Scalar** is selected on the User Defined Function screen (Figure 8-41 on page 8-53).

Input Parameters

Enter the information as described below. Enter an S to request a pop-up to guide you in defining the specific parameter type, or a question mark (?) for more information about the specific parameter type (this links to the applicable option 3.4 information panel).

Figure 8-42. Function Create (external scalar) Screen



Schema Name

Taken from the User Defined Function screen. The name of the schema in which the new UDF should be created.This field defaults to the SQL authorization ID if no schema is entered. Must not begin with 'SYS' unless it is 'SYSADMIN' or 'SYSPROC'.

Function Name

Name of the function as it is called by SQL statements.

Specific Name

Taken from the User Defined Function screen. Specific name of the UDF. If you do not provide a name, DB2 generates a unique name in the form SQLxxxxxxxxxxxx.

Language

Required field. Choose one of the four available languages (Assembler, PLI, C, or COBOL). Only available with either the Create or Alter options. You can only select one language.

External Name

Eight character module name. You can change the external name on this screen. If you blank out the external name, File-AID *for DB2* passes the EXTERNAL parameter to DB2.

Model Function

If you want to model this function on another, enter any or all of the schema, name, and specific name of the function you want to model the new function on. If the information entered does not uniquely identify a function, a selection window displays.

If you enter a model name, any model you previously applied is replaced.

You cannot use the following functions as a model:

- Non-scalar functions (if creating a scalar function)
- Non-table functions (if creating a table function)
- System-generated CAST functions
- Functions created with TABLE LIKE input parameter
- Functions created with CAST FROM return parameter
- Functions created with a language other than Assembler, C, Cobol, or PL1

D/T Ind

Indicates whether the parameter is a distinct type or table like parameter. Leave blank for built-in datatypes.

Parameter Type

Required. Select either built-in datatype, table like, or distinct type. In addition, you can use the S line command on one or more lines to request built-in datatype, distinct type, or table-like pop-up windows to guide you in defining each parameter type. You can also use a question mark (?) on one or more lines to see more information about each parameter type. Using the question mark (?) brings up either a built-in datatype pop-up, distinct type information screen, or table information screen. The distinct type and table information screens are useful for getting source type information for the distinct type and table columns. Note that an arrow (==>) on the Parameter Definition screen points to the parameter to which the pop-up applies.

Wildcards are valid and cause a selection list to appear.

Length (,scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

Parameter Name

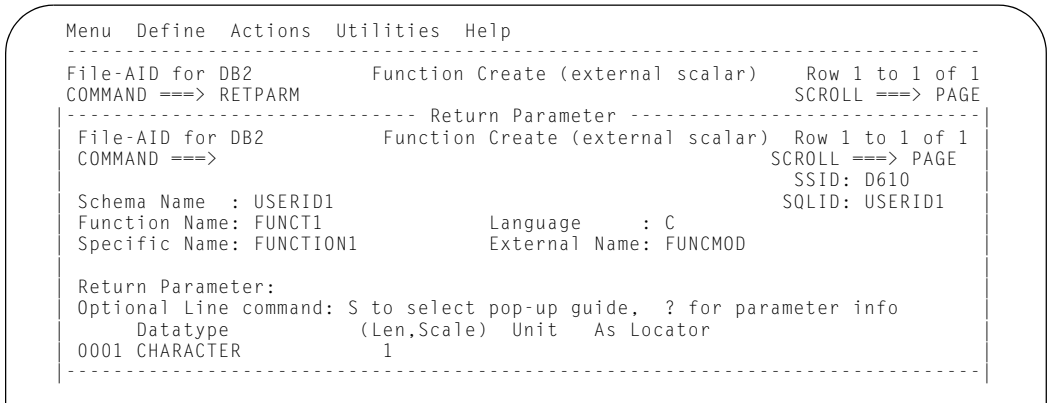
Name of the parameter. Optional.

Type RETPARM on the command line and press Enter to enter return parameter. The Function Create (external) popup displays (Figure 8-43 on page 8-56).

Note: If you try to view/edit DDL or execute create for the function at this point (before entering return parameters), the Return Parameter pop-up displays. You cannot do any other actions until you have entered return parameters and reviewed the resulting options.

Return Parameter

Figure 8-43. Function Create (external scalar) Popup



Note: This screen is a popup with no action bars. The Schema, Function, Specific Name, Language, and External Name fields are display only.

Datatype

Specific built-in or distinct datatype.

You can use the S line command on one or more lines to request either a built-in datatype or distinct type pop-up window to guide you in defining each parameter type. You can also use a question mark (?) on one or more lines to see more information about each parameter type. Using the question mark (?) brings up either a built-in datatype pop-up or distinct type information screen. The distinct type information screens are useful for getting source type information for the distinct type. Note that an arrow (==>) on the Return Parameter screen points to the parameter to which the pop-up applies.

Len , Scale

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

As Locator

Specifies that a location pointing to a value is passed rather than the actual value. Valid only on LOBs or distinct types based on LOB data.

Type END on the command line and press Enter. The Function Options screen (Figure 8-49 on page 8-64) displays. Review the options, change as necessary, and press Enter to return to the Function Create (external) screen. On this screen, select Execute Create from the Actions Pull down menu to create the function.

Create External Table

The Function Create (external table) screen (Figure 8-44) displays when option C for function type **External Table** is selected on the User Defined Function screen (Figure 8-41 on page 8-53).

Input Parameters

Enter the information as described below. When complete, type RETPARM on the command line and press Enter to go to the next screen.

Figure 8-44. Function Create (external table) Screen - Input Parameters

```
Menu  Define  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2          Function Create (external table)
COMMAND ===>                                                 SCROLL ===> CSR
                                                              SSID:
Schema Name   ===>                                            SQLID:
Function Name ===>                    Language      ===>      (ASM,C,COB,PLI)
Specific Name ===>                    External Name ===>
Model Function (Optional, * or pattern for list)
Schema             Function                    Specific
Name ===>          Name ===>                   Name ===>
Input Parameters:
Optional Line command: S to select pop-up guide,  ? for parameter info
     D/T Parameter                        Length   Unit Parameter
     Ind Type                             (,scale)      Name
```

Schema Name

Taken from the User Defined Function screen. The name of the schema in which the new UDF should be created. This field defaults to the SQL authorization ID if no schema is entered. Must not begin with 'SYS' unless it is 'SYSADMIN' or 'SYSPROC'.

Function Name

Name of the function as it is called by SQL statements.

Specific Name

Taken from the User Defined Function screen. Specific name of the UDF. If you do not provide a name, DB2 generates a unique name in the form SQLxxxxxxxxxxxx.

Language

Required field. Choose one of the four available languages (Assembler, PLI, C, or COBOL). Only available with either the Create or Alter options. You can only select one language.

External Name

Eight character module name. You can change the external name on this screen. If you blank out the external name, File-AID *for DB2* passes the EXTERNAL parameter to DB2.

Model Function

If you want to model this function on another, enter any or all of the schema, name, and specific name of the function you want to model the new function on. If the information entered does not uniquely identify a function, a selection window displays.

If you enter a model name, any model you previously applied is replaced.

You cannot use the following functions as a model:

– Non-scalar functions (if creating a scalar function)

- Non-table functions (if creating a table function)
- System-generated CAST functions
- Functions created with TABLE LIKE input parameter
- Functions created with CAST FROM return parameter
- Functions created with a language other than Assembler, C, Cobol, or PL1

Input Parameters Area

D/T Ind

Indicates whether the parameter is a distinct type or table like parameter. Leave blank for built-in datatypes.

Parameter Type

Required. Select either built-in datatype, table like, or distinct type. You can enter an S to request a pop-up to guide you in defining the specific parameter type, or you can enter a question mark (?) for more information about the specific parameter type (this links to the applicable option 3.4 information panel). Wildcards are valid and cause a selection list to appear.

Length (,scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

Parameter Name

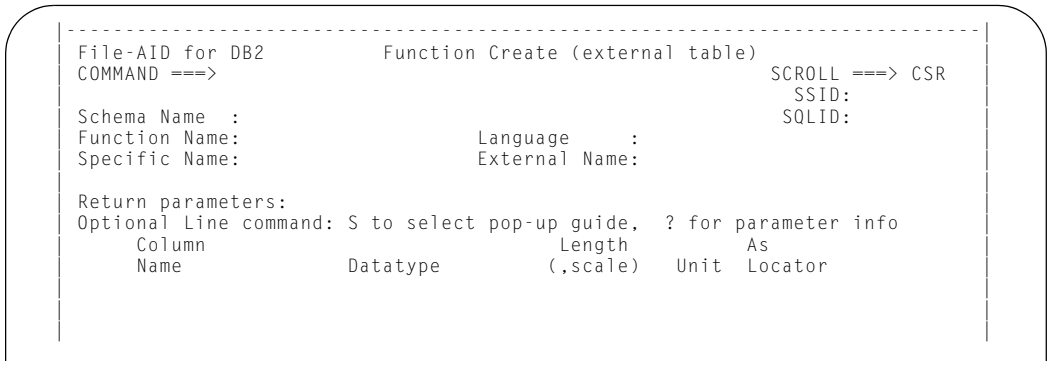
Name of the parameter. Optional.

Type the command RETPARM on the command line and press enter. The Function Create (external table) Return Parameters screen displays (Figure 8-45 on page 8-58).

Return Parameters

Enter the information as described below.

Figure 8-45. Function Create (external table) Return Parameters



Note: This screen is a popup with no action bars. The Schema, Function, Specific Name, Language, and External Name fields are display only.

Return Parameters Area

Column Name

Names of the columns.

Datatype

Specific built-in or distinct datatype.

Length (,scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

As Locator

Specifies that a location pointing to a value is passed rather than the actual value. Valid only on LOBs or distinct types based on LOB data.

Type END on the command line and press Enter. The Function Options screen (Figure 8-49 on page 8-64) displays. Review the options, change as necessary, and press Enter to return to the Function Create (external table) screen showing input parameters (Figure 8-44 on page 8-57). On this screen, select Execute Create from the Actions pull-down menu to create the External Table function.

Create Sourced Function

The Function Create (sourced) screen (Figure 8-46) displays when option C for function type **Sourced** is selected on the User Defined Function screen (Figure 8-41 on page 8-53).

Input Parameters

Figure 8-46. Function Create (sourced)

```
Menu  Define  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2          Function Create (sourced)
COMMAND ===>                                                 SCROLL ===> CSR
                                                              SSID:
Schema Name   ===> USERID1                                    SQLID:
Function Name ===> FUNCT1
Specific Name ===> FUNCTION1

Input Parameters:
Optional Line command: S to select pop-up guide,  ? for parameter info
     D/T Parameter           Length   Unit Parameter
     Ind Type                (,scale)      Name
```

Schema Name

Taken from the User Defined Function screen. The name of the schema in which the new UDF should be created.This field defaults to the SQL authorization ID if no schema is entered. Must not begin with 'SYS' unless it is 'SYSADMIN' or 'SYSPROC'.

Function Name

Name of the function as it is called by SQL statements.

Specific Name

Taken from the User Defined Function screen. Specific name of the UDF. If you do not provide a name, DB2 generates a unique name in the form SQLxxxxxxxxxxxx.

D/T Ind

Indicates whether the parameter is a distinct type or table like parameter. Leave blank for built-in datatypes.

Parameter Type

Required. Select either built-in datatype, table like, or distinct type. You can enter an S to request a pop-up to guide you in defining the specific parameter type, or you can enter a question mark (?) for more information about the specific parameter type (this links to the applicable option 3.4 information panel). Wildcards are valid and cause a selection list to appear.

Length (,scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

Parameter Name

Name of the parameter. Optional.

From here you can:

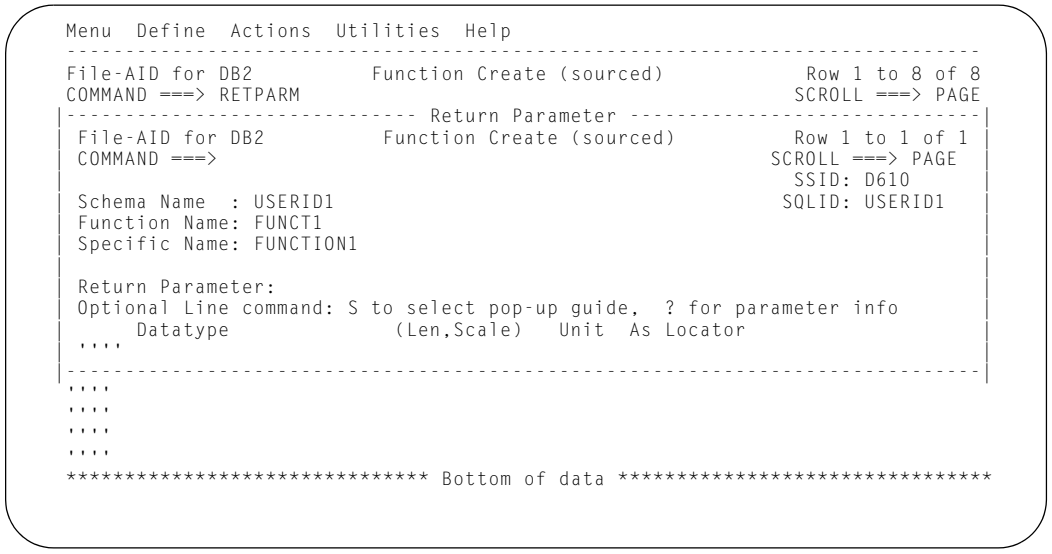
1. Type RETPARM on the command line and press Enter to enter a return parameter. The Function Create (external) return parameter popup displays (Figure 8-47).

   Note: If you try to view/edit DDL or execute create for the function at this point (before entering a return parameter), the Return Parameter pop-up displays. You cannot do any other actions until you have entered a return parameter.

2. Type SRCPARM on the command line and press Enter to enter source parameters. The Function Create (external) source parameters popup displays (Figure 8-48 on page 8-62).
3. Select Execute Create from the Actions pull-down menu to create the Sourced function.

Return Parameter

Figure 8-47. Function Create (sourced) Return Parameters Popup



Note: This screen is a popup with no action bars. The Schema, Function, and Specific Name fields are display only.

Datatype

Specific built-in or distinct datatype.

You can use the S line command on one or more lines to request either a built-in datatype or distinct type pop-up window to guide you in defining each parameter type. You can also use a question mark (?) on one or more lines to see more information about each parameter type. Using the question mark (?) brings up either a built-in datatype pop-up or distinct type information screen. The distinct type information screens are useful for getting source type information for the distinct type. Note that an arrow (==>) on the Return Parameter screen points to the parameter to which the pop-up applies.

(Len, Scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

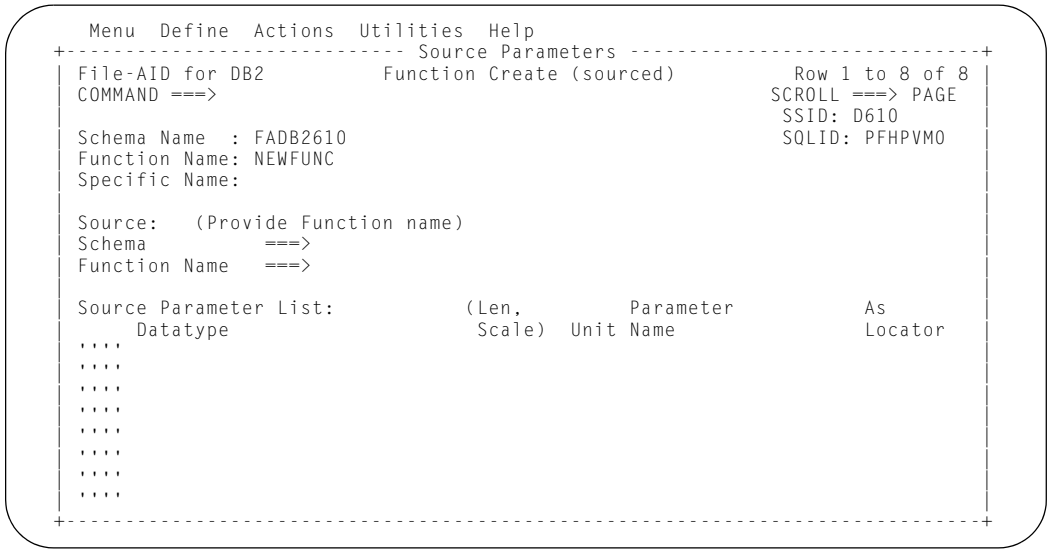
As Locator

Specifies that a location pointing to a value is passed rather than the actual value. Valid only on LOBs or distinct types based on LOB data.

Press END to return to the Function Create (sourced) screen (Figure 8-46).

Source Parameters

Figure 8-48. Function Create (sourced) Source Parameters Popup



Note: This screen is a popup with no action bars. The Schema, Function, and Specific name fields are display only.

Source Area

Schema

The name of the schema where the source function is located.

Function Name

Name of the function as it is called by SQL statements.

Specific Name

Specific name of the UDF. If you do not provide a name, DB2 generates a unique name in the form SQLxxxxxxxxxxx.

Source Parameter List Area

Datatype

Specific built-in or distinct datatype.

(Len,Scale)

Length and scale of the built-in column datatype.

Unit

If the column type is a LOB, specify the size unit for the value entered under the Column Length field to specify the maximum variable string length of the LOB.

K	kilobytes
M	megabytes
G	gigabytes

Parameter Name

Name of the parameter. Optional.

As Locator

Specifies that a location pointing to a value is passed rather than the actual value. Valid only on LOBs or distinct types based on LOB data.

Press END to return to the Function Create (sourced) screen (Figure 8-46).

Parameters

Many parameters are available to only a subset of datatypes. When you select a line on any of the function create screens (either external scalar, external table, or sourced), a definition pop-up displays.

The fields and their values on this pop-up vary depending on the datatype.

Any data on the function create screens that applies is propagated to the pop-up, such as length and unit. Any data entered into the pop-up is propagated back to the main function create screen.

For more information, refer to “Built-In Datatype, Distinct Type, and Table-Like Pop-up Windows” on page 8-80.

Function Options

From either the Function Create (external scalar) or Function Create (external table) screen, select Option 1, Function Options, from the Define pull-down menu. The Function Create Function Options screen displays. The screen is scrollable; Figure 8-49 shows the entire screen.

Figure 8-49. Create Function Options Screen

```
Menu  Define  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2            Create Function Options
COMMAND ===>                                                  SSID: D701

    Function:                              Specific Name:
                                                             More:     +
 Language          ===>      (ASM, C, COB, PLI)

 External Name     ===>

 Deterministic     ===>    (Y/N)

 Final Call        ===>    (Y/N)

 External Action   ===>    (Y/N)

 Stay Resident     ===>    (Y/N)

 DBInfo            ===>    (Y/N)

 Security          ===>    (1 DB2, 2 User, 3 Definer)

 Parallel          ===>    (1 Allow, 2 Disallow)

 Program Type      ===>    (1 Main, 2 Sub)

 On Null Input     ===>    (1 Returns Null on Null Input,
                            2 Called on NUll Input)

 WLM Environment   ===>

 Collid            ===>

 ASUTime Limit     ===>       (1-32767 or blank for NO Limit)

 Scratchpad        ===>       (1-32767 or blank for NO Scratchpad)

 SQL Data          ===>    (1 Reads SQL Data, 2 No SQL,
                            3 Modifies SQL Data, 4 Contains SQL)

 Special Registers ===>    (1 Inherit, 2 Default)

 Parameter CCSID   ===>    (1 Ascii, 2 Ebcdic, 3 Unicode)
```

Language

Required field. Choose one of the four available languages (Assembler, PLI, C, or COBOL). Only available with either the Create or Alter options. You can only select one language.

External Name

Eight character module name.

Deterministic

Default is N.

If you select Y, then the UDF returns the same result from successive calls that have identical input arguments.

Final Call

The final call option of an external function.

N Final call is not made to the function.

Y Final call is made to the function.

External Action

The external action option of an external function.

N	No side effects.
E	External side effects (number of invocations is important).

Stay Resident

Default is N.

If you select Y, then the UDF load module remains resident in memory after the UDF ends.

DBInfo

Default is N.

If you select Y, then an additional argument is passed when the UDF is invoked. The argument may contain information such as the application run-time authorization ID or schema name.

Security

Default is DB2. Specifies how the UDF works with external security products.

DB2	No special external security environment needed. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID associated with the UDF address space.
USER	An external security environment should be established for the UDF. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID of the user who invoked the UDF.
DEFINER	An external security environment should be established for the UDF. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID of the owner of the UDF.

Parallel

The PARALLEL option of an external function.

1	Allow invocation by parallel tasks.
2	Disallow invocation by parallel tasks.

Program Type

Select either MAIN or SUB. If you do not enter either MAIN or SUB, an error message displays.

MAIN	Run the UDF as a main routine.
SUB	Run the UDF as a subroutine.

On Null Input

1 Returns null on null input - Default. The function is not called if any of the input arguments is null. The result is the null value.

2 Called on null input - the function is called regardless of whether any of the input arguments is null, making the function responsible for testing null argument values. The function can return a null or non-null value.

WLM Environment

Identifies the MVS workload manager environment in which to run the UDF. To use a WLM environment, you must have authority for that environment.

The name can be up to 20 characters, with up to 18 characters for the WLM environment name, and 2 characters to support the nested WLM environment option.

To use the nested WLM environment option, you must type the characters ",*" (without quotes) after the WLM environment name (see example below).

```
wlmname,*
```

If you do not enter a WLM environment name, File-AID *for DB2* enters NO WLM ENVIRONMENT.

Collid

18 character collection ID name. If you do not enter a collection ID, then File-AID *for DB2* enters NO COLLID.

ASUTime Limit

Default is NO LIMIT. Specifies the total amount of processor time (in CPU service units) that a single run of the UDF can use. Values must be between 1 to 2 GB.

Scratchpad

Default is NO SCRATCHPAD. Specifies the scratchpad length for this UDF. Values must be between 1 and 32767.

SQL Data

Default is MODIFIES. Indicates whether the UDF may execute any SQL statements, and if so, what type.

READS UDF cannot run SQL statements that modify data.

NO SQL UDF cannot run any SQL statements.

MODIFIES UDF can run any SQL statement except statements not supported in any UDF.

CONTAINS UDF cannot run any SQL statements that read or modify data.

Special Register

Describes how special register values are inherited. Displays only if you select DB2 version 7 or later.

INHERIT The values of special registers are inherited based on rules listed in the table for characteristics of special registers (see DB2 SQL Reference Manual)

DEFAULT The values of special registers are initialized to the default values.

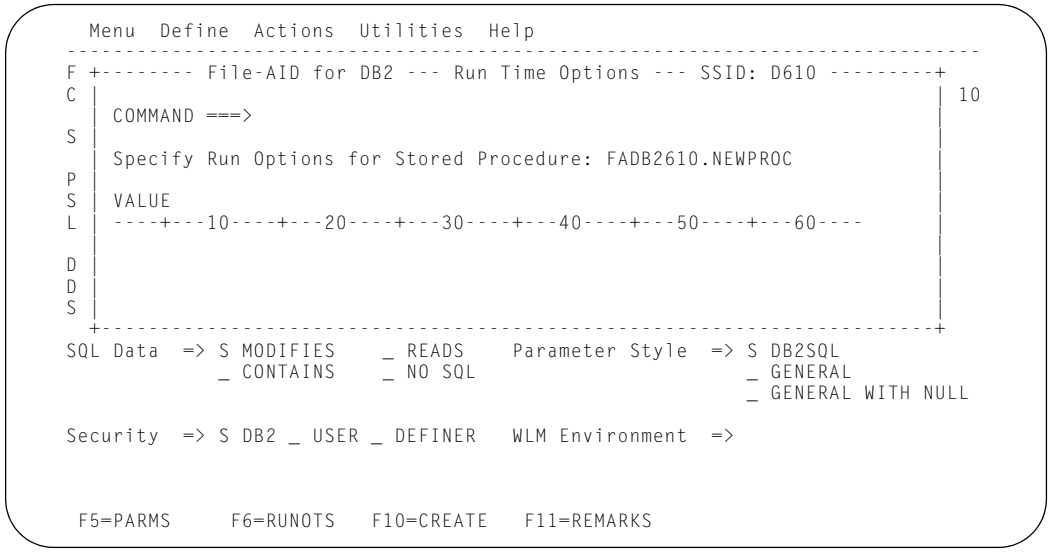
Parameter CCSID

This parameter indicates whether the encoding scheme for string parameters is ASCII, EBCDIC, or UNICODE. Displays only if you select DB2 version 7 or later.

Run Time Options

The Run Time Options window displays when you select Run Time Options from the Define pull-down menu, or enter the RUNOPTS command after you finish typing in information on the Function Options screen. Press End to return to the Function Options screen.

Figure 8-50. Run Time Options Window

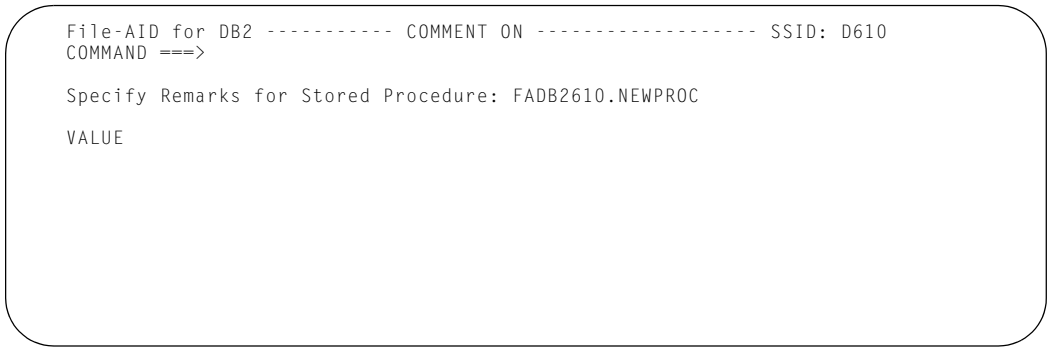


Run Time Options can be a maximum of 254 characters long. The value may wrap across four lines. Schema and procedure name are populated from the Create Functions screen.

Comment On

The Comment On screen displays when you select Remarks/Comment On from the Define pull-down menu, or enter the REMARKS command after you finish typing in information on the Function Options screen. Press End to return to the Function Options screen. The message 'Remarks Defined' displays.

Figure 8-51. Comment On Screen



The comment on text is a maximum of 254 characters for DB2 Version 6 and 7, and a maximum of 762 characters for DB2 Version 8.

View/Edit DDL

From any of the Function Create screens, select the View/Edit DDL option from the Actions pull-down menu or enter the SQL primary command. The DDL you have created through the previous screens displays. You can view or edit the DDL on this screen. If you later encounter errors in the executing the function create, you can return to this screen to correct them.

Execute Create

To create the function, select the Execute Create option from the Actions pull-down menu, or enter CREATE command and press Enter.

If you encounter errors, press HELP for a DB2 error information. Return to the View/Edit DDL screen to correct the DDL. After correction, File-AID *for DB2* asks if you want to execute the modified DDL.

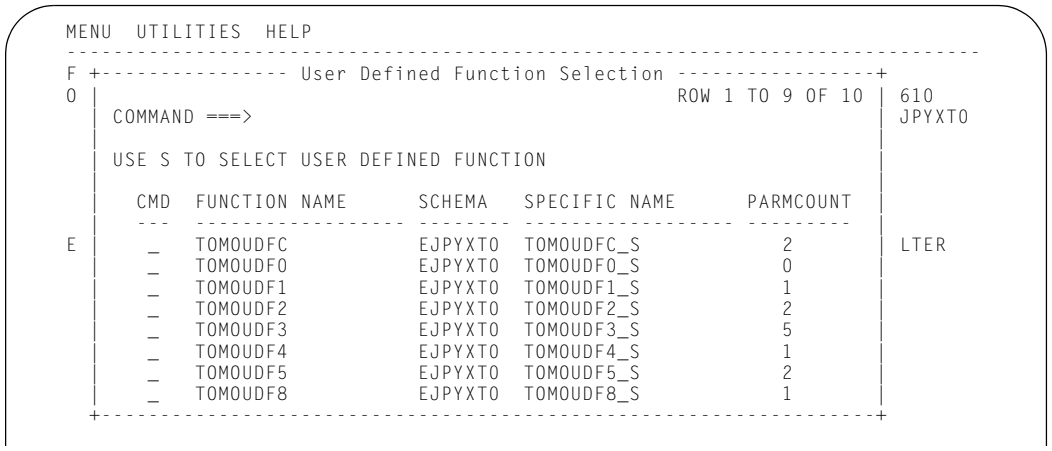
If the execute create runs successfully, the message “UDF Created” displays in the upper right-hand corner of the Function Create screens.

Alter Function

Select A from the User Defined Function screen (Figure 8-41 on page 8-53) and enter Function, Schema, and Specific Name information as appropriate.

If you enter a non-specific function, a pop-up displays with all qualifying functions.

Figure 8-52. Function Alter Selection List



From the pop-up, enter an S to select a function to alter, or enter the I command to see more information on the function. Figure 8-53 displays when you enter S. Figure 8-54 displays when you enter the I command.

Figure 8-53. Alter Functions Options

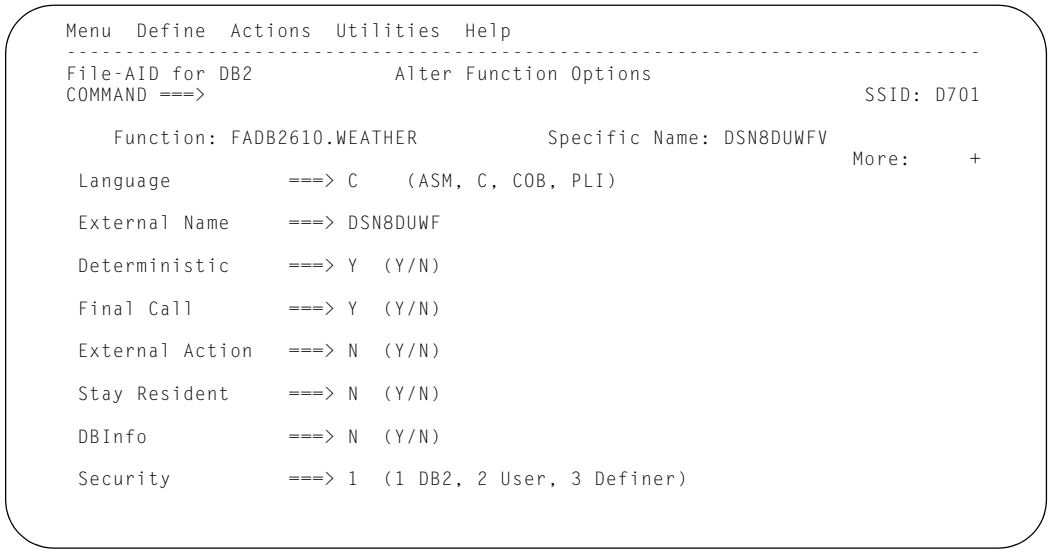


Figure 8-54. User Defined Function Information Screen

```
File-AID for DB2 ------ User Defined Function Information --------- SSID: D7O1
 COMMAND ===>                                                 SCROLL ===> CSR

Specify 'S' to select more information:
  _ Parameter Information
  _ Run Time Information
  _ Remarks/Comment On

 Function Name: FADB261O.WEATHER                              Language: C
 Specific Name: DSN8DUWFV
 External Name: DSN7DUWF

 UDF Detail ------------------------------------------  Parameter---------
  Owner           : PFHJGDO      Routineid       : 771     Parm Count  : 1
  CreatedBy       : PFHJGDO      Routine type    :         Parm Style  : D
  Function Type   : T            Origin          : E       Return type :
  Cast Function   : N            Program Type    : S       Result Sets :
  Final Call      : Y            Deterministic   : Y       Result Cols : 7
  SQL Data        : N            External Action : N       LOBcolumns  : 0
  IBMREQD         : N            Null Call       : Y
  Security        : D            DBinfo          : N
  Scratchpad Size : 100          Special Register: I
  SourceSchema    :              COMMIT on return:
  Sourcespecific  :
  Createdts       : 2002-05-16-09.15.04.718925
  Alteredts       : 2002-05-16-09.15.04.718925

 Performance ------------------------------------------------------------------
  Fenced          : Y            Cardinality     :
  Parallel        : D            IOS per Invoc   :
  Stay Resident   : N            Insts per Invoc  :
  ASUtime         : 5            Initial IOS     :
  WLM Environment : WLMENV       Initial Insts    :
  WLM Env Nested  : N

 JAVA -------------------------------------------------------------------------
  JAR ID      :
  JAR Schema  :
  CLASS       :
```

Parameter Information

Selecting this option brings up the UDF Parameter Information screen (Figure 8-55 on page 8-71).

Run Time Information

Selecting this option brings up the Run Time Options window (Figure 8-50 on page 8-67).

Remarks/Comment On

Selecting this option brings up the Comment On screen (Figure 8-51 on page 8-68).

Parameter Information

The Parameter Information window displays when you select Parameter Information from the User Defined Function Information screen (Figure 8-54 on page 8-70). Press End to return to the Function Create screen.

Figure 8-55. UDF Parameter Information Screen

```
File-AID for DB2 -------- UDF Parameter Information ---------- Row 1 to 6 of 6
COMMAND ===>
Function Name: EJPYXT0.TOMOUDF3                            Language: C
Specific Name: TOMOUDF3

Parameters are listed in Ordinal number Order
RT = Row type  SS = Scale  ST = Subtype  NEC = Encoding Scheme
                                                                  E
PARAMETER          TYPE     TYPE               R                 S N AS  TABLE
NAME               SCHEMA   NAME               T  LENGTH     SS T C LOC COLNO
------------------ -------- ------------------ -- ---------- -- - - --- -----
                   SYSIBM   INTEGER            C           4  0      N
Table Parameter    SYSIBM   CHAR               P         254  0 S E  Y      1
                   SYSIBM   DECIMAL            P           5  0      Y      2
                   SYSIBM   TIMESTAMP          P          10  0 S E  Y      3
                   SYSIBM   ROWID              P          17  0      Y      4
                   EJPYXT1  LONGUDTNAMECLOB2GG P           4  0 S E  Y      5
******************************* Bottom of data ********************************
```

When you have selected a function (S) from the Function Alter Selection List (Figure 8-52), the Alter Function Options screen displays. File-AID *for DB2* only supports altering options. The screen is scrollable; Figure 8-56 on page 8-71 shows the entire screen. When you have changed the options as necessary, select Execute Alter from the Actions pull-down menu.

Figure 8-56. Alter Function Options Screen

```
Menu  Define  Actions  Utilities  Help
-------------------------------------------------------------------------------
File-AID for DB2            Alter Function Options
COMMAND ===>                                                      SSID: D701

    Function:                           Specific Name:
                                                                  More:     +
 Language          ===>      (ASM, C, COB, PLI)

 External Name     ===>

 Deterministic     ===>    (Y/N)

 Final Call        ===>    (Y/N)

 External Action   ===>    (Y/N)

 Stay Resident     ===>    (Y/N)

 DBInfo            ===>    (Y/N)

 Security          ===>    (1 DB2, 2 User, 3 Definer)

 Parallel          ===>    (1 Allow, 2 Disallow)

 Program Type      ===>    (1 Main, 2 Sub)

 On Null Input     ===>    (1 Returns Null on Null Input,
                            2 Called on NUll Input)

 WLM Environment   ===>

 Collid            ===>

 ASUTime Limit     ===>       (1-32767 or blank for NO Limit)

 Scratchpad        ===>       (1-32767 or blank for NO Scratchpad)

 SQL Data          ===>    (1 Reads SQL Data, 2 No SQL,
                            3 Modifies SQL Data, 4 Contains SQL)

 Special Registers ===>    (1 Inherit, 2 Default)
```

Language

Required field. Choose one of the four available languages (Assembler, PLI, C, or COBOL). Only available with either the Create or Alter options. You can only select one language.

External Name

Eight character module name.

Deterministic

Default is N.

If you select Y, then the UDF returns the same result from successive calls that have identical input arguments.

Final Call

The final call option of an external function.

N	Final call is not made to the function.
Y	Final call is made to the function.

External Action

The external action option of an external function.

N	No side effects.
E	External side effects (number of invocations is important)

Stay Resident

Default is N.

If you select Y, then the UDF load module remains resident in memory after the UDF ends.

DBInfo

Default is N.

If you select Y, then an additional argument is passed when the UDF is invoked. The argument may contain information such as the application run-time authorization ID or schema name.

Security

Default is DB2. Specifies how the UDF works with external security products.

DB2	No special external security environment needed. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID associated with the UDF address space.
USER	An external security environment should be established for the UDF. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID of the user who invoked the UDF.

DEFINER An external security environment should be established for the UDF. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID of the owner of the UDF.

Parallel

The PARALLEL option of an external function.

1 Allow invocation by parallel tasks.

2 Disallow invocation by parallel tasks.

Program Type

Select either MAIN or SUB. If you do not enter either MAIN or SUB, an error message displays.

MAIN Run the UDF as a main routine.

SUB Run the UDF as a subroutine.

On Null Input

1 Returns null on null input.

2 Called on null input.

WLM Environment

Identifies the MVS workload manager environment in which to run the UDF. To use a WLM environment, you must have authority for that environment.

The name can be up to 20 characters, with up to 18 characters for the WLM environment name, and 2 characters to support the nested WLM environment option.

To use the nested WLM environment option, you must type the characters "**,***" (without quotes) after the WLM environment name (see example below).

```
wlmname,*
```

If you do not enter a WLM environment name, File-AID *for DB2* enters NO WLM ENVIRONMENT.

Collid

18 character collection ID name. If you do not enter a collection ID, then File-AID *for DB2* enters NO COLLID.

ASUTime Limit

Default is NO LIMIT. Specifies the total amount of processor time (in CPU service units) that a single run of the UDF can use. Values must be between 1 to 2 GB.

Scratchpad

Default is NO SCRATCHPAD. Specifies the scratchpad length for this UDF. Values must be between 1 and 32767.

SQL Data

Default is MODIFIES. Indicates whether the UDF may execute any SQL statements, and if so, what type.

NO SQL UDF cannot run any SQL statements.

MODIFIES UDF can run any SQL statement except statements not supported in any UDF.

READS UDF cannot run SQL statements that modify data.

CONTAINS UDF cannot run any SQL statements that read or modify data.

Special Register

Describes how special register values are inherited. Displays only if you select DB2 version 7 or later.

INHERIT The values of special registers are inherited based on rules listed in the table for characteristics of special registers (see *DB2 SQL Reference Manual*).

DEFAULT The values of special registers are initialized to the default values.

Confirm Drop

The Confirm Drop screen displays when the drop option is selected on the User Defined Function Option screen and after the name of the existing UDF is entered. If the UDF is to be dropped, press Enter. To cancel the drop process, enter the END primary command. This is a safety feature intended to prevent you from dropping a UDF unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

Option 9, Stored Procedures

The Stored Procedures screen (Figure 8-57) displays when option 9 is selected on the Object Selection Menu.

Figure 8-57. Stored Procedures Screen

```
 Menu  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2               Stored Procedures
OPTION  ===>                                              SSID  ===> D610
                                                          SQLID ===> FDSQLID
     C - Create  PROCEDURE
     A - Alter   PROCEDURE
     D - Drop    PROCEDURE

Enter:       - Use * or pattern for selection list for D(rop) and A(lter) -

     Procedure Name ===> FDSTPROC
     Schema Name    ===>

     Use 'S' to select one applicable Language:  - for C(reate) and A(lter) -

     _ Assembler         _ PLI

     _ C                 _ REXX

     _ COBOL             _ SQL
```

Procedure Name

The name you choose for the stored procedure. Can be up to 18 characters long. Wildcards are valid when using D(rop) and A(lter). The name, including the implicit or explicit schema qualifier, must be unique on the current server. If you use a wildcard, a selection list displays.

Schema Name

The name of the schema in which the new stored procedure should be created.This field defaults to the SQL authorization ID if no schema is entered. Must not begin with 'SYS' unless it is 'SYSADMIN' or 'SYSPROC'.

Applicable Language

Required field. Choose one of the six available languages (Assembler, PLI, C, REXX, COBOL, or SQL). Only available with either the Create or Alter options. You can only select one language. Subsequent screens vary slightly depending on what language you choose and what version of DB2 you use.

Stored Procedure Create

Each Stored Procedure Create screen varies slightly based on the language you chose and the DB2 version you are using. The screen shown in Figure 8-58 displays when you select COBOL, Assembler C, and PLI languages, and DB2 version 7 or later.

Defaults are prefilled on this screen. If you blank out a default, but do not replace it with another value, File-AID *for DB2* does not pass the value to DB2.

Figure 8-58. Stored Procedure Create Screen

```
  Menu  Define  Actions  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Stored Procedure Create
COMMAND ===>                                                    SSID:  D610

Specify:  'S' to Select   'Y' for Yes  'N' for No      or   enter value

Procedure Name => NEWPROC             External Name    =>
Schema Name    => FADB2610            Collection Id    =>
Language:         ASSEMBLER           Program Type     => _ MAIN   _ SUB

DBINFO         => N  (Y/N)            Result Sets      => 0
Deterministic  => N  (Y/N)            Commit on Return => N  (Y/N)
Stay Resident  => N  (Y/N)            ASUTIME Limit    =>

SQL Data  => S MODIFIES    _ READS    Parameter Style  => S DB2SQL
             _ CONTAINS    _ NO SQL                      _ GENERAL
                                                         _ GENERAL WITH NULL

Security  => S DB2 _ USER _ DEFINER   WLM Environment  =>


 F5=PARMS     F6=RUNOTS   F10=CREATE   F11=REMARKS
```

Procedure Name

The procedure name from the Stored Procedures screen (Figure 8-57 on page 8-74) (if entered).

Schema Name

The schema name from the Stored Procedures screen (Figure 8-57 on page 8-74) (if entered). If you did not specify a schema name on the Stored Procedures screen, File-AID *for DB2* enters the Current SQLID value. You can change the schema name on

this screen. If you do not want the schema name explicitly coded into the CREATE PROCEDURE statement, then blank it out on this screen.

Language

The language(s) selected on the Stored Procedures screen (Figure 8-57 on page 8-74). The language is shown in white; it cannot be changed on this screen. The display of this panel is dependent on the language selection.

External Name

Eight character module name. You can specify the external name on this screen. Default is blank. If you leave this field blank, File-AID *for DB2* passes the EXTERNAL parameter, without any name, to DB2.

Collection ID

18 character collection ID name. Default is blank. If you leave this field blank, File-AID *for DB2* passes NO COLLID to DB2. If you have chosen REXX as the language, valid values are either DSNREXRR, DSNREXRS, DSNREXCR, or DSNREXCS.

Program Type

Select either MAIN or SUB. If you do not enter either MAIN or SUB, an error message displays. If you have chosen REXX as the language, only SUB is applicable.

MAIN Run the stored procedure as a main routine.

SUB Run the stored procedure as a subroutine.

DBINFO

Default is N.

If you select Y, then an additional argument is passed when the stored procedure is invoked. The argument may contain information such as the application run-time authorization ID or schema name. If you select Y, then Parameter Styles (below) must be DB2SQL If you have chose REXX or SQL as the language, this field cannot be Y.

Deterministic

Default is N.

If you select Y, then the stored procedure returns the same result from successive calls that have identical input arguments.

Stay Resident

Default is N.

If you select Y, then the stored procedure load module remains resident in memory after the stored procedure ends.

Result Sets

Default is 0. A synonym for Dynamic Result Sets. Must be a number between 0 and 32767.

Commit on Return

Default is N.

If you select Y, then DB2 commits the transaction immediately on return from the stored procedure.

ASUTIME Limit

Default is blank. If you leave this field blank, File-AID *for DB2* passes NO LIMIT to DB2. Specifies the total amount of processor time (in CPU service units) that a single run of the stored procedure can use. Values must be between 1 to 2147483647.

SQL DATA

Default is MODIFIES. Indicates whether the stored procedure may execute any SQL statements, and if so, what type. If you choose SQL as language, NO SQL is not applicable and does not appear on the screen.

MODIFIES Stored procedure can run any SQL statement except statements not supported in any stored procedure.

READS Stored procedure cannot run SQL statements that modify data.

CONTAINS Stored procedure cannot run any SQL statements that read or modify data.

NO SQL Stored procedure cannot run any SQL statements.

Parameter Style

Default is DB2SQL. Identifies the linkage convention used to send parameters to the stored procedure. If you choose SQL as language, Parameter Style is not applicable and does not appear on the screen.

DB2SQL Sends the parameters on the CALL statement to the stored procedure, along with a null indicator for each parameter on the CALL statement, the SQLSTATE to return to DB2, the qualified and specific name of the stored procedure, and the SQL diagnostic string to return to DB2. If you choose REXX as language, DB2SQL is not applicable and does not appear on the screen.

GENERAL Sends only the parameters on the CALL statement to the stored procedure.

GENERAL WITH NULLS Sends the parameters on the CALL statement to the stored procedure, along with an additional argument containing a vector of null indicators for each of the parameters on the CALL statement related to null parameter values.

Security

Default is DB2. Specifies how the stored procedure works with external security products.

DB2 No special external security environment needed. If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID associated with the stored procedure address space.

If NO WLM ENVIRONMENT is set, then SECURITY DB2 is the only valid value.

USER An external security environment should be established for the stored procedure.

If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID of the user who invoked the stored procedure.

DEFINER An external security environment should be established for the stored procedure. If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID of the owner of the stored procedure.

WLM Environment

Identifies the MVS workload manager environment in which to run the stored procedure. To use a WLM environment, you must have authority for that environment.

The name can be up to 20 characters, with up to 18 characters for the WLM environment name, and 2 characters to support the nested WLM environment option.

To use the nested WLM environment option, you must type the characters ",*" (without quotes) after the WLM environment name (see example below).

```
wlmname,*
```

If you do not enter a WLM environment name, File-AID *for DB2* passes NO WLM ENVIRONMENT. This value is only valid if the program type is MAIN, security is DB2, and there are no parms with LOB datatype or distinct type based on an LOB datatype.

If you chose REXX language, you must enter a WLM environment name, or an error message displays.

Special Register

Describes how special register values are inherited. Displays only if you select DB2 version 7 or later.

INHERIT The values of special registers are inherited based on rules listed in the table for characteristics of special registers (see DB2 SQL Reference Manual).

DEFAULT The values of special registers are initialized to the default values.

Parameter CCSID

This parameter indicates whether the encoding scheme for string parameters is ASCII, EBCDIC, or UNICODE. Displays only if you select DB2 version 7 or later.

Parameters

The Parameters window displays when you select Parameters from the Define pull-down menu, or enter the PARMS command after you finish typing in information on the Stored Procedures Create screen. Press End to return to the Stored Procedures create screen.

You can define parameters to specify built-in data types, distinct types, or tables (table-like).

There are some considerations depending on the language chosen for the stored procedure:

- If the language is SQL, then some built-in datatypes (CLOB, BLOB, DBCLOB and ROWIDs), and distinct types are not valid.
- If the language is REXX, only one parameter can be specified as O(ut) or IO(In/Out).
- If the language is SQL, then CCSID cannot be UNICODE.

In Figure 8-59 on page 8-79, the user has selected lines 2, 6, and 7 to show a popup guide. Because line 6 contains a distinct type, the Distinct Type Selection pop-up also displays (Figure 8-7 on page 8-9).

Figure 8-59. Parameter Definition window

```
 Menu  Define  Actions  Utilities  Help
--------------------------------------------------------------------------------
+------ File-AID for DB2 ---- PARAMETER DEFINITION ---- SSID: D701 ------------+
|                                                          Row 1 to 8 of 8     |
| COMMAND ===>                                          SCROLL ===> CSR        |
|                                                                              |
| Stored Procedure: FDSQLID.FDSTPROC                    Language: COBOL        |
|                                                                              |
| Enter optional Line Commands:  S to Select pop-up Guide   ? to list Info     |
|                                                                              |
|     In                                    Length                             |
|     Out   Ind Parameter                   ,Scale          Parameter          |
|     InOut D/T Type                        (LL,SS)    Unit Name               |
|     ----- --- --------------------------- ---------- ---- ------------------ |
| 0001 I        CHARACTER     25                                               |
| S002 O        CLOB                        2147483647      CLOBPARM1          |
| 0003 IO       BLOB                        2048         M                     |
| 0004 I     D  BFHUSER1.DOLLARS                           DISTINCTPARM1      |
| 0005 IO    T  FADB2710.EMP_PHOTO_RESUME                                      |
| S006 O     D  DO*                                                            |
| S007          DECIMAL                                                        |
| ****************************** Bottom of data ****************************** |
+------------------------------------------------------------------------------+
```

The following message displays in the upper right corner:

```
Parameters Defined
```

Line Commands

Enter line commands to modify the parameter entries. You can use the following regular line commands:

- A(fter)
- B(efore)
- C(opy)
- D(elete)
- M(ove)
- R(epeat)
- I(nsert)

In addition, you can use the S line command on one or more lines to request built-in datatype, distinct type, or table-like pop-up windows to guide you in defining each parameter type. You can also use a question mark (?) on one or more lines to see more information about each parameter type. Using the question mark (?) brings up either a built-in datatype pop-up, distinct type information screen, or table information screen. The distinct type information screens are useful for getting source type information for the distinct type.

Note that an arrow (==>) on the Parameter Definition screen points to the parameter to which the pop-up applies.

In/Out/InOut

Default is I.

I(n) The parameter is an input parameter to the stored procedure.

O(ut) The parameter is an output parameter that is returned by the stored procedure.

IO The parameter is both an input and output parameter for the stored procedure.

Ind D/T

Indicates whether the parameter is a distinct type or table like parameter. Leave blank for built-in datatypes.

Parameter Type

Required. Specify built-in datatype, table name, or distinct type name. Valid built-in datatypes include:

BLOB (BL)	CHARACTER (CH)	CLOB (CL)	DBCLOB (DB)
GRAPHIC (GR)	VARCHAR (VC)	VARGRAPHIC (VG)	DATE (DT)
INTEGER (IN)	DOUBLE PRECISION (DP)	DOUBLE (DO)	REAL (RL)
ROWID (RI)	SMALLINT (SI)	TIME (TM)	TIMESTAMP (TS)
FLOAT (FL)	DECIMAL (DC)	NUMERIC (NU)	

For table name and distinct name, wildcards are valid and cause a selection list to appear. If only one object is found using the wildcards, File-AID *for DB2* treats the object as selected and a selection list does not appear.

Length, Scale

Length and scale of the built-in column datatype.

Unit

If the source datatype is a large object (LOB), specify the storage unit (K, M, or G).

Parameter Name

Optional. Name of the parameter. If you define multiple parameters and name them, each parameter name needs to be unique in the stored procedure.

Built-In Datatype, Distinct Type, and Table-Like Pop-up Windows

When you specify S as a line command on a parameter line containing a built-in datatype and with either D or T in the IND column, then either the Built-in Datatype, Distinct Type or Table Like pop-up displays (see sample pop-up in Figure 8-60 on page 8-81).

The fields and their values on these pop-up windows vary depending on the data type. Refer to the table below for applicable fields by datatypes.

Any data on the Parameter Definition screen (Figure 8-59 on page 8-79) that applies displays on the pop-up (such as datatype and In/Out/InOut). Any data entered on these pop-up windows is propagated back to the Parameter Definition screen.

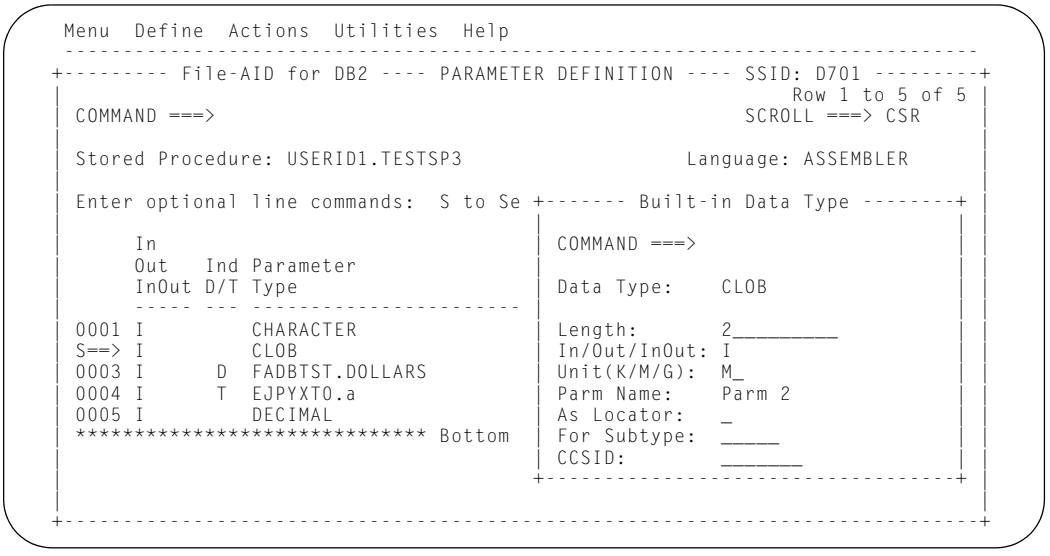
If you use the question mark (?) line command on a parameter line containing a built-in datatype, the Built-In Data Type pop-up displays. This shows more parameter information about the specific built-in datatype, but all fields are protected (you cannot update them).

You can use wildcards in the Name and Schema fields for distinct type, and in the name and creator fields for table-like. Selection lists appear.

Note that an arrow (==>) on the Parameter Definition screen points to the parameter to which the pop-up applies.

Datatypes/ Applicable Fields	**CHAR VARCHAR**	**DBCLOB**	**CLOB BLOB**	**GRAPHIC VARG**	**DATE/TIME T_STAMP INTEGER DOUBLE REAL/ROWID SMALLINT**	**DECIMAL NUMERIC**	**FLOAT**	**DISTINCT**	**TABLE**
Data Type (NAME)	X	X	X	X	X	X	X	X	X
I/O/IO	X	X	X	X	X	X	X	X	X
Length	X	X	X	X					
Unit		X	X						
Parm Name	X	X	X	X	X	X	X	X	X
As Locator		X	X					X (for LOB)	X
For Subtype	X		X						
CCSID	X	X	X	X					
Precision						X	X		
Scale						X			
Schema								X	
Creator									X

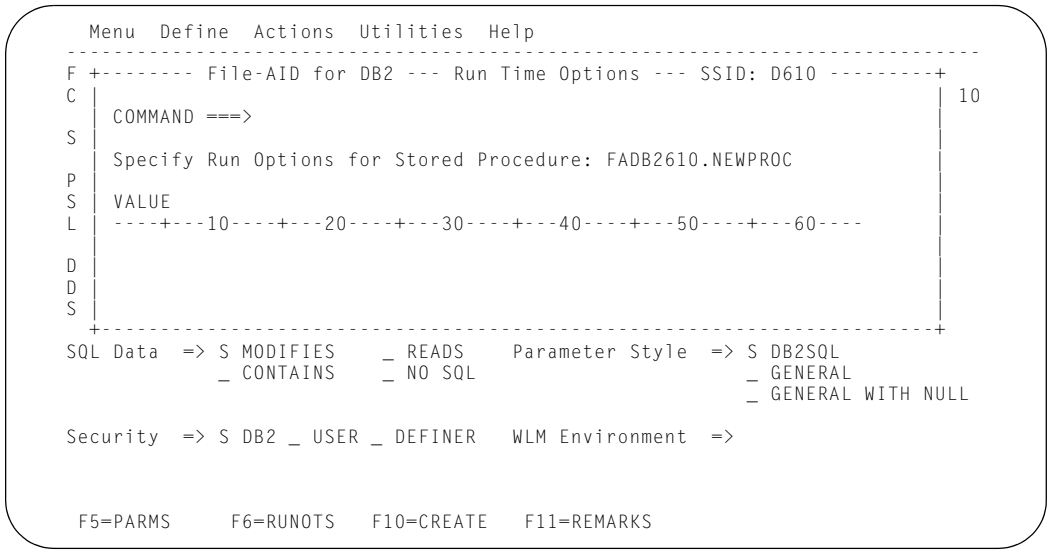
Figure 8-60. Built-In Datatype Pop-up Sample



Run Time Options

The Run Time Options window displays when you select Run Time Options from the Define pull-down menu, or enter the RUNOPTS command after you finish typing in information on the Stored Procedures Create screen. Press End to return to the Stored Procedures create screen.

Figure 8-61. Run Time Options Window



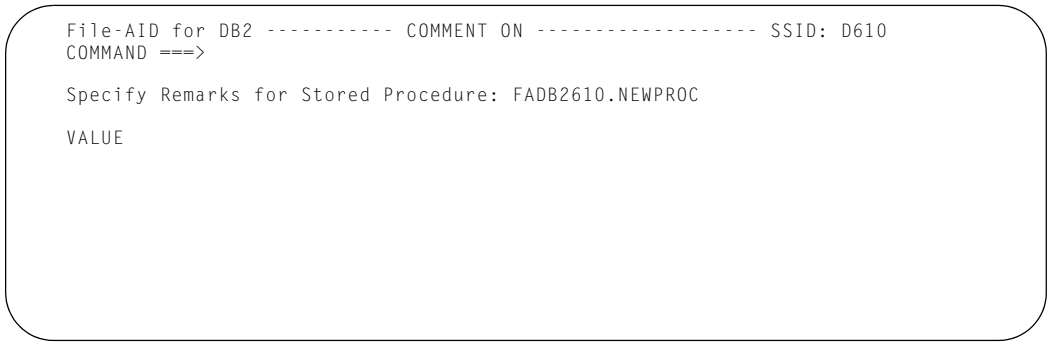
Run Time Options can be a maximum of 254 characters long. The value may wrap across four lines. Schema and procedure name are populated from the Stored Procedure Create panel.

Comment On

The Comment On screen displays when you select Remarks/Comment On from the Define pull-down menu, or enter the REMARKS command after you finish typing in information on the Stored Procedures Create screen. Press End to return to the Stored Procedures create screen. The following message displays:

Remarks Defined

Figure 8-62. Comment On Screen



The comment on text is a maximum of 254 characters for DB2 Version 6 and 7, and a maximum of 762 characters for DB2 Version 8.

View/Edit DDL

From the Stored Procedure Create screen, select the View/Edit DDL option from the Actions pull-down menu or enter the SQL primary command. The DDL you have created through the previous screens displays. You can view or edit the DDL on this screen. If

you later encounter errors in the executing the stored procedure create, you can return to this screen to correct them.

Figure 8-63. DDL to Create Stored Procedure

```
EDIT ---- PFHPVM0.SHOW.SQL(SHOWSQL) - 01.00 ------------------ Columns 001 072
COMMAND ===>                                                  SCROLL ===> HALF
****** ***************************** Top of Data ******************************
000001  CREATE PROCEDURE FADB2610.NEWPROC
000002    DYNAMIC RESULT SETS 0
000003    EXTERNAL
000004    LANGUAGE ASSEMBLE
000005    PARAMETER STYLE DB2SQL
000006    NOT DETERMINISTIC
000007    FENCED
000008    CALLED ON NULL INPUT
000009    MODIFIES SQL DATA
000010    NO DBINFO
000011    NO COLLID
000012    NO WLM ENVIRONMENT
000013    ASUTIME NO LIMIT
000014    STAY RESIDENT NO
000015    PROGRAM TYPE MAIN
000016    SECURITY DB2
000017    COMMIT ON RETURN NO;
****** **************************** Bottom of Data ****************************
```

FENCED and CALLED ON NULL INPUT are coded in the Create Procedure statement, but do not appear on any screens. The values shown are defaults and are currently the only allowable OS/390 values for DB2 version or later.

File-AID *for DB2* does not support SQL/PSM procedure statements (e.g. PROCEDURE BODY) of SQL stored procedures. If you selected SQL as language, File-AID *for DB2* generates a dummy SQL body:

```
SELECT * FROM SYSIBM.SYSDUMMY1
```

Execute Create

To create the stored procedure, select the Execute Create option from the Actions pull-down menu, or enter CREATE command and press Enter. Executing the create also executes the COMMENT ON statement, if used.

If you encounter errors, press HELP for a DB2 error information, including COMMENT ON statement errors. Return to the View/Edit DDL screen (Figure 8-63 on page 8-83) to correct the DDL. After correction, File-AID *for DB2* asks if you want to execute the modified DDL.

If the execute create runs successfully, the following message displays in the upper right-hand corner of the Stored Procedure Create screen:

```
Stored Procedure Created
```

Alter

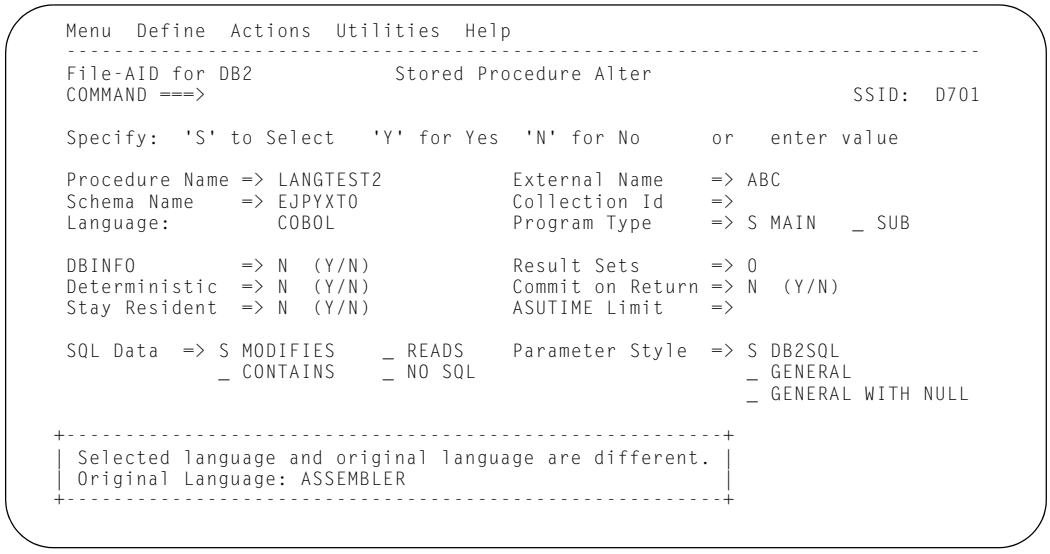
To alter a procedure, enter the procedure name and schema of the procedure you want to alter on the Stored Procedures screen (Figure 8-57 on page 8-74), along with a language and the primary command A. Press Enter.

Wildcards are valid on the Stored Procedures screen. Using a wildcard for procedure name or schema name or both brings up the Stored Procedure Selection screen, where you can choose a stored procedure to alter (Figure 8-65 on page 8-85).

Language is a required field. Choose one of the six available languages (Assembler, PLI, C, REXX, COBOL, or SQL). You can only select one language. Subsequent screens will vary slightly depending on what language you choose and what version of DB2 you use.

If the language you choose does not match the language of the stored procedure you entered, a message window displays on the Stored Procedure Alter screen.

Figure 8-64. Confirm Alter Screen



The Stored Procedure Alter screen works in the same way as the Stored Procedure Create screen (Figure 8-58 on page 8-75), with the exception of stored procedure parameters, which cannot be altered.

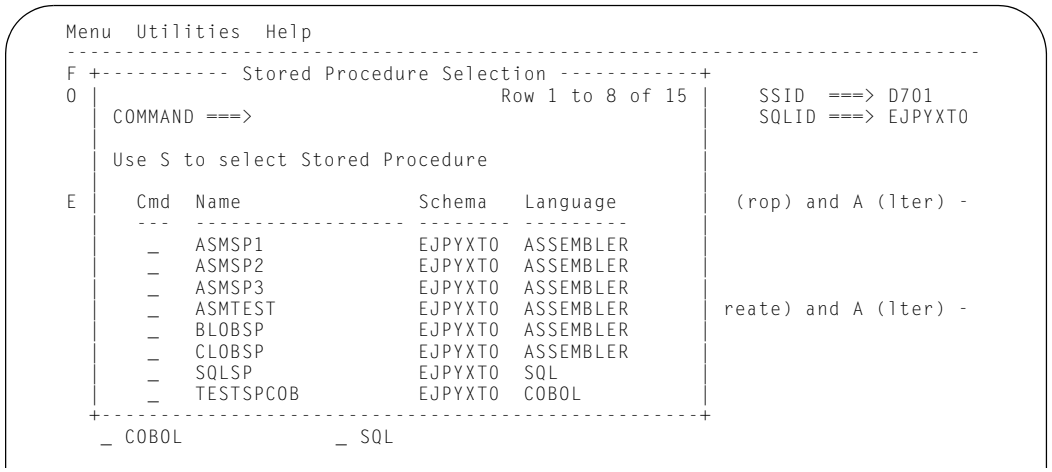
Confirm Drop

The Confirm Drop screen displays when you choose D from the Stored Procedures screen. It gives you the opportunity to confirm whether you want to drop the stored procedure you have specified. This is a safety feature intended to prevent you from dropping a stored procedure unintentionally. See Figure 8-18 on page 8-23 for an example of a similar screen.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to drop the stored procedure. Enter SQL on the command line before pressing Enter or executing the END command.

Wildcards are valid on this screen. Using a wildcard for procedure name or schema name or both brings up the Stored Procedure Selection screen, where you can choose a stored procedure to drop.

Figure 8-65. Stored Procedure Selection Screen



Option 10, Triggers

The Triggers screen (Figure 8-66), displays when option 10 is selected on the Object Selection menu. It allows you to create or drop a trigger.

Triggers on the same object are executed in the order they were created. A trigger cannot be altered, so a drop and recreate of a trigger can potentially change the order in which triggers are executed.

Figure 8-66. Triggers Screen

OPTION

Specify the letter of the required option:

C Create a trigger. Displays the screen shown in Figure 8-67 on page 8-86.

D Drop a trigger.

Enter Area

Trigger Name

Name of the trigger to be created or dropped. This field is required when the create (C) or drop (D) options are selected. The trigger name cannot be more than 8 characters. When creating a trigger, the name and schema combination must be unique on the current server.

Schema

Enter the name of the schema or owner of the trigger to be created or dropped. This field defaults to the Current User ID if no schema is entered. Must not begin with SYS unless it is SYSADMIN. The schema name cannot be more than 8 characters.

Enter a unique name for the trigger, a schema name (if desired) and the command C (Create). Press Enter. The Trigger Create screen displays.

Trigger Create

If any errors occur during trigger create, you can correct them by stepping back through the screens. You do not have to use the VIEW DDL screen to correct SQL errors.

Figure 8-67. Trigger Create Screen

```
 Menu  Define  Actions  Utilities  Help
------------------------------------------------------------------------------
File-AID for DB2               Trigger Create
COMMAND ===>                                                  SSID: D610

Specify:  'S' to Select   or   enter value          Trigger Terminator: #

Trigger Name     => TRIGDEPT   Triggering Table => FD.TRIGDEPTTAB
Schema Name      => FDSQLID    Activation Time  => _ BEFORE S AFTER
                               Granularity      => S ROW    _ STATEMENT
                               Triggering Event => S UPDATE _ INSERT _ DELETE

 Parameters below are Optional:

(* or pattern for Column Selection List or ** to display chosen Update Columns)
                             0 Update Column(s) => *

                               Referencing
                                 OLD AS         => OLDDEPT
                                 NEW AS         => NEWDEPT
                                 OLD TABLE AS   =>
                                 NEW TABLE AS   =>
```

Trigger Terminator

This is based on the trigger terminator set on the User Parms screen. See "Option 9, Trigger Terminator" on page 5-11 for more information.

Trigger Name

Trigger name from Triggers screen (Figure 8-66 on page 8-85) if entered. When creating a trigger, the name and schema combination must be unique on the current server.

Schema Name

Schema name from Triggers screen (Figure 8-66 on page 8-85) if entered. If no schema name was specified on the Triggers screen, this field shows the value from the CURRENT SQLID special register, which is used by DB2 if no schema name is provided.

Triggering Table

Enter an explicitly qualified table in the form CREATOR.TABLENAME. You can also enter an asterisk (*) for either creator or tablename.

If you do not provide a Creator name, and there is more than one table with the name you entered for triggering table, a popup displays showing all tables on the current server with that table name.

Activation Time

Specifies when the trigger activity takes place.

A Trigger activity occurs after the triggering event.

B Trigger activity occurs before the triggering event.

Use **S** to select. If you select BEFORE, then the Granularity setting cannot be STATEMENT, and the optional referencing selections OLD TABLE AS and NEW TABLE AS are not valid.

Granularity

Trigger is executed once per:

S Statement

R Row

Use **S** to select. If you select STATEMENT, then the optional referencing selections OLD AS and NEW AS are not valid.

Triggering Event

Operation that activates the trigger:

I Insert

D Delete

U Update

Use **S** to select. If you select DELETE, then the optional referencing selections NEW AS and NEW TABLE AS are not valid.

If you select INSERT, then the optional referencing selections OLD AS and OLD TABLE AS are not valid.

Optional Parameter Section

Update Column(s)

Can be used only if the Triggering Table name is entered and UPDATE is selected as the triggering event. You can enter one specific update column on this screen or use an asterisk (*****) to list all columns valid for update.

The number to the left of the Update Column field shows the number of columns selected for update. This is protected and cannot be edited by the user.

OLD AS

The name that identifies the state of the row prior to the triggering operation.

Maximum of 18 characters. Not valid if you select Triggering Event INSERT.

Not valid if you select Granularity STATEMENT.

NEW AS

The name that identifies the state of the row after the triggering operation.

Maximum of 18 characters. Not valid if you select Triggering Event DELETE.

Not valid if you select Granularity STATEMENT.

OLD TABLE AS

The name of a temporary table that identifies the state of the complete set of rows modified by a triggering operation before that operation occurs.

Maximum of 18 characters. Not valid if you select Activation Time BEFORE.

Not valid if you select Triggering Event INSERT.

NEW TABLE AS

The name of a temporary table that identifies the state of the complete set of rows modified by a triggering operation after that operation occurs.

Maximum of 18 characters. Not valid if you select Activation Time BEFORE.

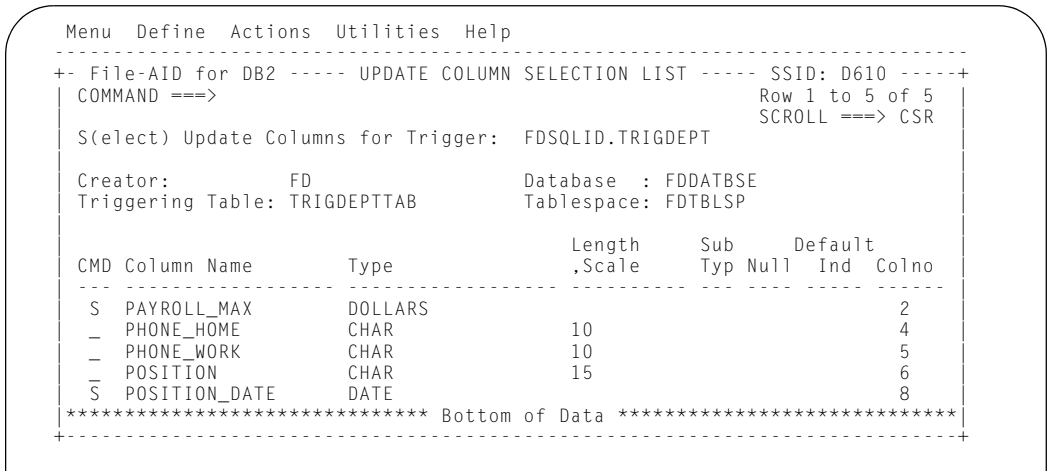
Not valid if you select Triggering Event DELETE.

Update Column

The Update Column Selection List (Figure 8-68 on page 8-88) displays when you have entered an asterisk in the Update Column field of the Trigger Create screen (Figure 8-67 on page 8-86).

After you press Enter, the Trigger Create screen redisplays with the message “Update Column(s) selected” in the upper right-hand corner, and the Update Columns field shows the number of columns that were chosen on the Update Column Selection List.

Figure 8-68. Update Column Selection List



Creator

Filled in by File-AID *for DB2* based on the information entered on the Trigger Create screen.

Triggering Table

Filled in by File-AID *for DB2* based on the information entered on the Trigger Create screen.

Database

Filled in by File-AID *for DB2* based on the information entered on the Trigger Create screen.

Tablespace

Filled in by File-AID *for DB2* based on the information entered on the Trigger Create screen.

CMD

Enter **S** to select a specific column name to update.

Column Name

Column that matched the criteria you entered in the Trigger Create screen. Only columns that can be updated are displayed.

Type

For columns that are distinct types, the name of the distinct type displays in this column.

Length,Scale

Length and scale of the column datatype.

Sub Typ

Indicates the subtype of the data.

Null

Indicates whether or not the column may contain NULL values. Y means the column may contain NULL values, **N** means that the column may not contain NULL values.

Default Ind

The type of default value defined for the indicated column.

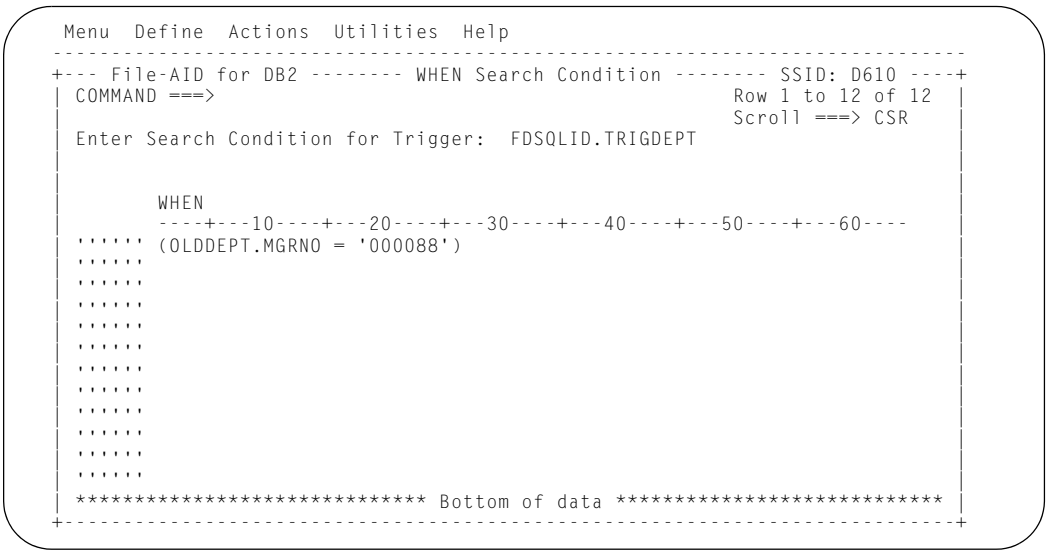
Colno

The column number of the column listed.

WHEN Search Condition

The WHEN Search Condition window displays when you select When Search Condition from the Define pull-down menu, or enter the WHENCOND command after you finish typing in information on the Trigger Create screen. Press End to return to the Trigger Create screen.

Figure 8-69. WHEN Search Condition Screen

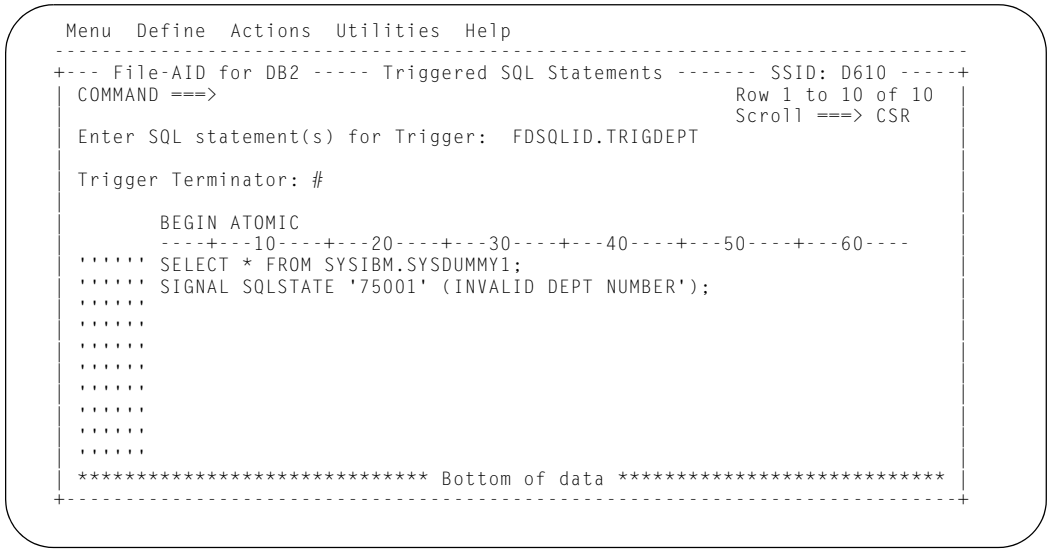


Do not type in WHEN - when you select this action, WHEN is generated automatically at trigger create time. You can enter other search conditions as shown in the example using standard ISPF commands.

Begin Atomic...End

The Begin Atomic...End screen displays when you select Begin Atomic...End from the Define pull-down menu, or enter the ATOMIC command after you finish typing in information on the Trigger Create screen. Press End to return to the Trigger Create screen.

Figure 8-70. Begin Atomic...End Screen



Do not type in the Begin Atomic or End statement. The statement is generated automatically at trigger create time. Enter any required triggered SQL statements, ending each with a semicolon (;), using standard ISPF commands. All statements are enclosed by File-AID *for DB2* within the Begin Atomic...End clause.

File-AID *for DB2* includes the trigger terminator specified by the user in the User Parms. The trigger terminator also displays on the Begin Atomic...End screen, as shown in the example.

Comment On

The Comment On screen displays when you select Remarks/Comment On from the Define pull-down menu, or enter the REMARKS command after you finish typing in information on the Trigger Create screen. Press End to return to the Trigger Create screen.

To select this screen, you must have a trigger terminator other than NONE, as the terminator is necessary for other SQL to coexist with the CREATE TRIGGER DDL statement.

Figure 8-71. Comment On Screenf

```
File-AID for DB2 ---------- COMMENT ON ------------------------- SSID: D610
COMMAND ===>

Specify Remarks for Trigger: FDSQLID.TRIGDEPT

VALUE

USED BY THE QUALITY ASSURANCE GROUP TO CREATE AUDIT INFORMATION
ON DEPARTMENT
```

The comment on text is a maximum of 254 characters for DB2 Version 6 and 7, and a maximum of 762 characters for DB2 Version 8.

View/Edit DDL

From the Trigger Create screen, select the View/Edit DDL option from the Actions pull-down menu or enter the SQL primary command. The DDL you have created through the previous screens displays. You can view or edit the DDL on this screen. If you later encounter errors in the executing the trigger create, you can return to this screen to correct them.

Figure 8-72. DDL to Create Trigger

```
 EDIT ---- BFHUSER1.FILEAID.DB2.SHOW.SQL ----------------------- Columns 001 072
COMMAND ===>                                                  SCROLL ===> CSR
****** ***************************** Top of Data ******************************
010000   --#SET TERMINATOR #
010000   CREATE TRIGGER FDSQLID.TRIGDEPT
020000     AFTER
030000     UPDATE
040000     OF PAYROLL_MAX, POSITION_DATE
050000     ON FD.TRIGDEPTTAB
060000     REFERENCING NEW AS NEWDEPT
070000                 OLD AS OLDDEPT
080000     FOR EACH ROW MODE DB2SQL
090000     WHEN (OLDDEPT.MGRNO = '000088')
100000     BEGIN ATOMIC
110000     SELECT # FROM SYSIBM.SYSDUMMY1;
120000     SIGNAL SQLSTATE '75001' ('INVALID DEPT NUMBER') ;
130000     END #
140000   --#SET TERMINATOR ;
150000 COMMENT ON TRIGGER FDSQLID.TRIGDEPT
160000  IS
170000 'USED BY THE QUALITY ASSURANCE GROUP TO CREATE AUDIT INFORMATION ON INFO
180000 RMATION ON DEPARTMENT';
****** *************************** Bottom of Data ****************************
```

File-AID *for DB2* inserts a --#SET TERMINATOR statement before and after the CREATE TRIGGER statement. The setting in the before statement is the value in the new trigger terminator parameter. The setting in the after statement is the semicolon (the default SQL statement terminator). The END keyword is followed by the currently set trigger terminator. This allows you to run this DDL through DSNTEP2 at a later time.

If the trigger terminator is set to NONE, the --#SET TERMINATOR statements are not included, and the END keyword is not followed by a terminator.

Execute Create

To create the trigger, select the Execute Create option from the Actions pull-down menu, or enter CREATE command and press Enter. Executing the create also executes the COMMENT ON statement, if used.

If you encounter errors, press HELP for a DB2 error information, including COMMENT ON statement errors. Return to the View/Edit DDL screen (Figure 8-72 on page 8-91) to correct the DDL. After correction, File-AID *for DB2* asks if you want to execute the modified DDL.

If the execute create runs successfully, the message "Trigger Created" displays in the upper right-hand corner of the Trigger Create screen.

Confirm Drop

The Confirm Drop screen displays when you choose **D** from the Triggers screen. It gives you the opportunity to confirm whether you want to drop the trigger you have specified. This is a safety feature intended to prevent you from dropping a trigger unintentionally. Figure 8-18 on page 8-23 shows an example of a similar screen.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to drop the trigger. Enter SQL on the command line before pressing Enter or executing the END command.

Chapter 9.
Copy Utility

This chapter describes the Copy Rows Between Tables utility, which provides a means of copying data from one DB2 table or view to another table. Both tables must have the same CCSID (ASCII or EBCDIC). If the target table does not exist, File-AID *for DB2* transfers to the Create Table utility. Once you create the table, you are automatically returned to the Copy utility. The following topics are described in this chapter:

- Table Copy Selection
- Auto Create
- Trigger Warning
- Copy Column Selection
- Copy Row Selection
- SQL Entry Panel
- EDIT SPUFI Selection

Table Copy Selection

The Table Copy Selection screen (Figure 9-1) appears when you select option **3** from the Utility Selection Menu.

Figure 9-1. Table Copy Selection Screen

```
File-AID for DB2 -------------- Table Copy Selection --------------------------
 COMMAND ===>                                                  SSID ===> DSN

 Specify Table Copy Option:
    Specify Option ===>                 Blank - Full Table Copy
                                        T - Use a Selection Template
                                        S - Use SQL for Selection (SPUFI)

                  From Table                          To Table
                                                      ---------
                  ---------------       Location   ===> LOCATION NAME
 Creator     ===> USERID                Creator    ===> USERID
 Table Name  ===> NEW                   Table Name ===>


 Maximum Rows To Select  ===> *         (* for all rows)
 Column Selection        ===> NO        (Yes or No, ignored for S option)
 Delete Existing Rows    ===> NO        (Yes or No)
 Column Name Matching    ===> YES       (Yes or No, ignored for S option)
 Discard Limit           ===> 0
 Use Uncommitted Read    ===> YES       (Yes or No)
```

Specify Option

Specify the option representing the activity you want to pursue:

Blank	Copy an entire table or view.
T	Use a Selection Template to copy only selected rows.
S	Use SPUFI-type selection. (Displayed in Figure 9-8 on page 9-8.)

From Table Area

Creator

ID of person who created the table or view to be copied.

Table Name

Name of the table, alias, synonym, or view to be copied. Wildcard characters (see "Wildcard Characters" on page 2-9) can be used to display the Copy From Selection window (Figure 9-2 on page 9-3).

To Table Area

Location

DB2 Subsystem Location ID of the table to which the data in the From Table is being copied to. If you leave the Location field blank, then the To Table Area is assumed to be on same subsystem to which File-AID *for DB2* is connected.

Creator

ID of person who created the table or view to which you are copying.

Table Name

Name of the table, synonym, view, or alias to which you are copying. Wild card characters (see "Wildcard Characters" on page 2-9) can be used to display the Copy To Selection window (Figure 9-3 on page 9-4).

If you specify a table that does not exist, the Auto Create window appears (Figure 9-4 on page 9-5).

Note: If the source (copy from) table has at least one trigger defined, a message indicating that a trigger exists appears in the Auto Create confirmation screen.

Maximum Rows To Select

Enter the maximum number of rows to be copied.

Column Selection

Indicate whether you want to select columns to be copied. This option, for example, lets you map columns with like data types but unlike names. Enter **Y** when you want to copy only selected columns of similar or dissimilar tables/views. Column selection is ignored for the S option.

If you enter **Y** in this field, the Copy Column Selection screen (Figure 9-6 on page 9-6) appears. If you enter **N**, the Copy Row Selection screen (Figure 9-7 on page 9-7) appears.

Delete Existing Rows

Indicate whether you want to clear out all existing rows in the table you are copying to before the new rows are inserted.

Column Name Matching

Indicate whether the columns you are copying will be mapped to target columns with the same names. This option is convenient if you are populating a table or view that is defined the same as the one you are copying from, or if the FROM and TO tables or views have columns with matching names but a different order. Matching is ignored for the S option.

Discard Limit

Maximum number of From Table rows you want discarded before the copy process is halted. Enter zero, or any value up to 9999. A value of zero allows the entire input dataset to be discarded if errors occur. If a non-zero value is entered, load processing

is halted and updates rolled back when the number of errors exceeds the specified value. The default is zero.

Use Uncommitted Read

Allows an application to bypass acquiring the locks that ensure no other user is updating the data being read. Bypassing the locks causes the application to run faster.

CAUTION:
If you choose the Use Uncommitted Read option, and the application fails and backs out the updates, then the rows read by the application using uncommitted read could be invalid.

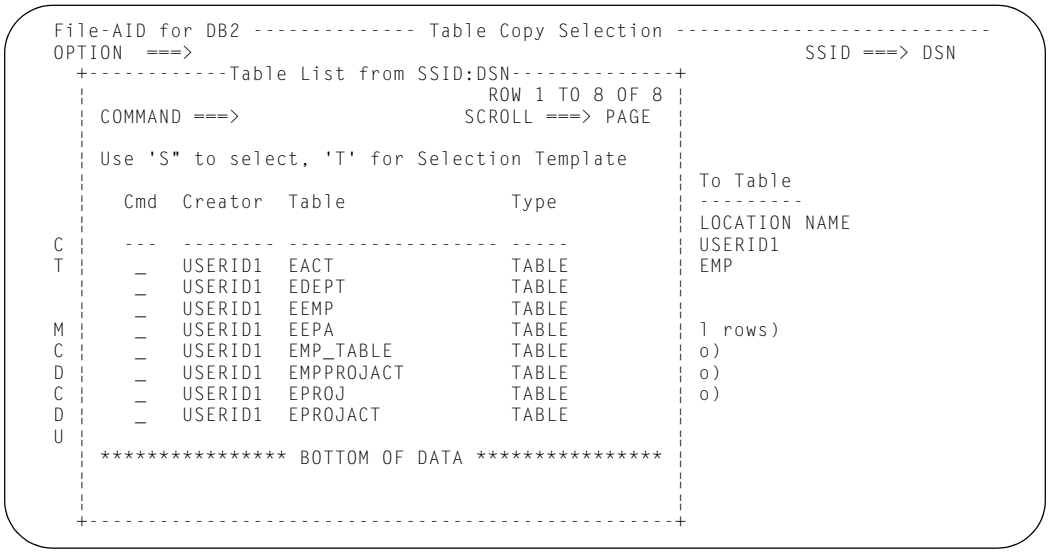
NO Default. Enter NO if you do not want the application to bypass acquiring the locks which ensure that no other user is updating the data being read.

YES Enter YES if you want the application to bypass acquiring the locks which ensure that no other user is updating the data being read. If you enter YES, the rows read by the application using uncommitted read could be invalid.

Copy From List of Tables

A list of tables appears when a wildcard pattern in the From Table Name field on the Table Copy Selection screen finds one or more matches.

Figure 9-2. Copy From List of Tables



Cmd

Enter the desired option.

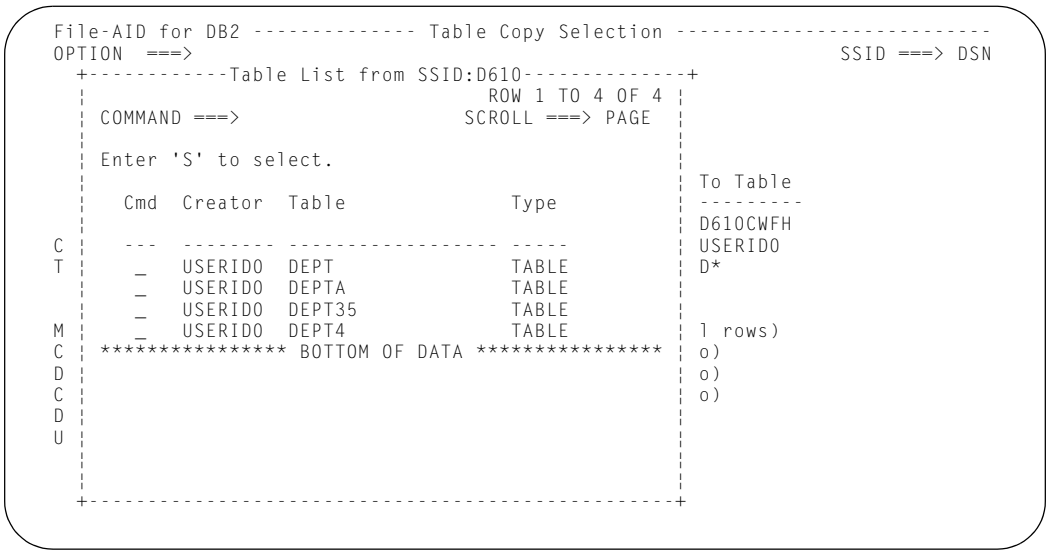
S Perform a full table copy.

T Use a Selection Template.

Copy To List of Tables

A table list appears when a wild card pattern in the To Table Name field on the Table Copy Selection screen finds one or more matches.

Figure 9-3. Copy To List of Tables



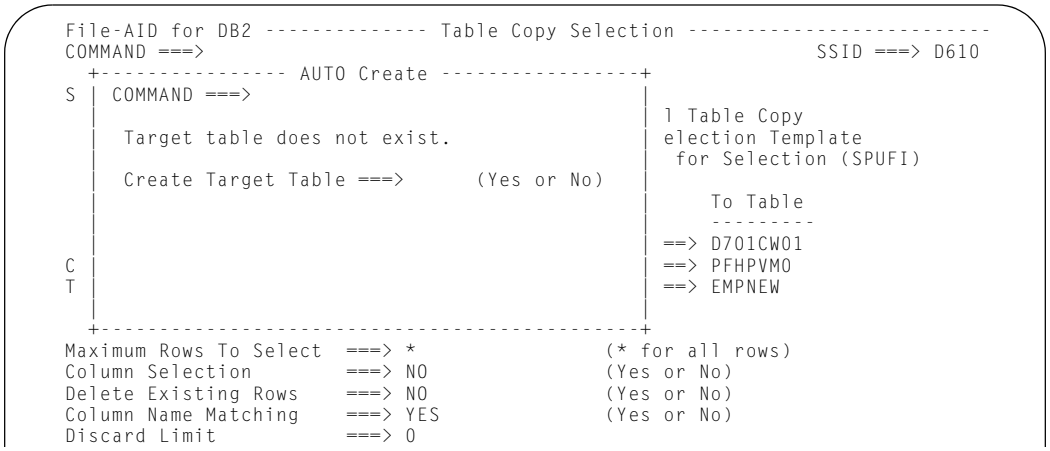
Cmd

Enter **S** to select a table.

Auto Create Window

The Auto Create window appears when the To Table specified on the Table Copy Selection screen (Figure 9-1 on page 9-1) does not exist. It allows you to create the target table automatically by taking you directly into the Table Create facility. See “Table Create” on page 8-3 for more information. When you exit from the table create screens, you return to the Table Copy Selection screen.

Figure 9-4. Auto Create Window



Create Target Table

Y Displays the Table Create screen (Figure 8-4 on page 8-4). See "Option 1, Tables" on page 8-2 for information on creating a table.

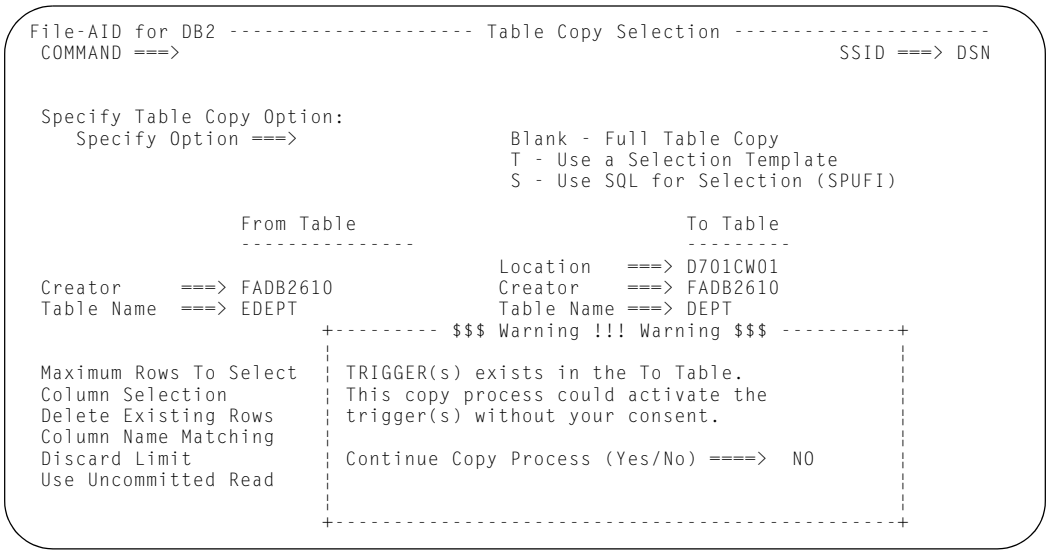
N Exits the window and displays the Table Copy Selection screen.

Trigger Warning

The Trigger Warning window appears when triggers exist in the To Table specified on the Table Copy Selection screen (Figure 9-1 on page 9-1). This warning window allows you to either continue with the copy process or to return to the Table Copy Selection screen.

Note: As File-AID *for DB2* has no knowledge of what the trigger actions are, it is your responsibility to decide whether or not the copy process should continue.

Figure 9-5. Trigger Warning Window



Continue Copy Process

Yes

The copy process continues. File-AID *for DB2* copies any qualifying data into the To Table but it has no control over the trigger actions that might apply.

No

Exits the window and displays the Table Copy Selection screen again, allowing you to specify another To Table.

Copy Column Selection

The Copy Column Selection screen appears for Full Table and Selection Template Copy options and when you enter **Y** in the Column Selection field (see "Table Copy Selection" on page 9-1). It lets you map columns from one table to another.

If you have specified **Y** in the Column Name Matching field and the column names and data types are identical, they are mapped to the screen in the original order (see the Map field). If column names and data types are not identical, or you specified **N** in the Column Name Matching field, you must indicate which columns are to be mapped by inserting the appropriate integers in the Map field.

Unmatched column names that are defined as NOT NULL are initialized with a map value of D, meaning that they will be set to the default value for the data type.

To copy from or into distinct type columns both columns must have the same UDT with identical schema, name and built-in data type.

You can execute the SQL primary command on this screen to view or save the SQL File-AID *for DB2* generated to complete the task.

Figure 9-6. Copy Column Selection screen

```
File-AID for DB2 -------- Copy Column Selection ----------- ROW 1 TO 13 OF 14
COMMAND ===>                                                SCROLL  ===> PAGE
                                                                   SSID: DSN
Enter SQL to view/save the SQL source

From Table: USERID.EMP
To Table  : USERID.EMP_NEW

    From Table Columns   Data Type       Map  To Table Columns    Data Type
    ------------------   ----------      ---  -----------------   ---------

 1  EMPNO                CH(6)            1   EMPNO               CH(6)
 2  FIRSTNAME            VC(12)           D   FNAME              VC(12)
 3  MIDINIT              CH(1)            D   MIDDLE              CH(1)
 4  LASTNAME             VC(15)           D   LNAME              VC(15)
 5  WORKDEPT             CH(3)            5   WORKDEPT            CH(3)
 6  PHONENO              CH(4)            6   PHONENO             CH(4)
 7  HIREDATE             DT               7   HIREDATE            DT
 8  JOB                  CH(8)            8   JOB                 CH(8)
 9  EDLEVEL              SI               9   EDLEVEL             CH(1)
 10 SEX                  CH(1)            10  SEX                 DT
 11 BIRTHDATE            DT               11  BIRTHDATE           DT
 12 SALARY               DC(9,2)          12  SALARY              DC(9,2)
 13 BONUS                DC(9,2)          13  BONUS               DC(9,2)

F18=EXECUTE
```

From Table

Name of the table or view to be copied.

To Table

Name of the table or view to which the From Table is being copied.

From Table Columns

Columns to be copied.

Data Type

Data types and lengths of the columns to be copied.

Map

Specify in what order the columns of the From Table are to be mapped. Type in the column numbers of the From Table, or enter **D** to fill in a default value.

To Table Columns

Names of the columns to which you are copying.

Date Type

Data types and lengths of the columns to which you are copying.

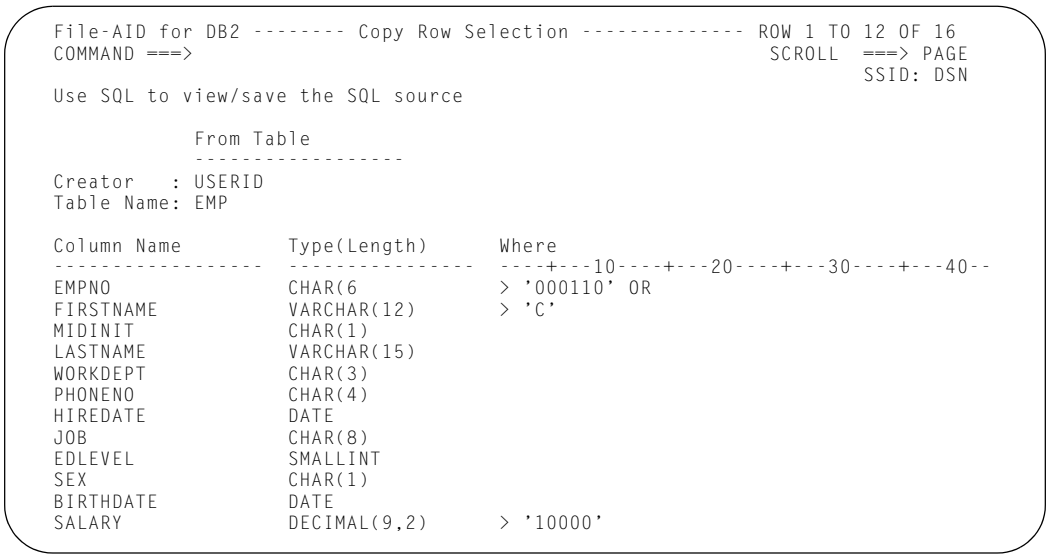
Copy Row Selection

The Copy Row Selection screen appears by entering **T** in the OPTION field on the Table Copy Selection screen (see "Table Copy Selection" on page 9-1). You bypass the Copy Column Selection screen only when you enter **N** in the Column Selection field. You can also reach the Copy Row Selection screen from the Copy Column Selection screen by typing TEMPLATE and pressing Enter, or pressing the associated PF key. The Copy Row Selection screen lets you specify WHERE statements that are used to define specific rows to be copied.

When you complete your entries, type EXECUTE and press Enter or press the associated PF key to complete the copy. A message stating how many rows have been copied appears on the Table Copy Selection screen.

You can execute the SQL primary command on this screen to view or save the SQL File-AID *for DB2* generated to complete the task.

Figure 9-7. Copy Row Selection screen



```
File-AID for DB2 -------- Copy Row Selection -------------- ROW 1 TO 12 OF 16
COMMAND ===>                                                SCROLL  ===> PAGE
                                                                    SSID: DSN
Use SQL to view/save the SQL source

            From Table
            ------------------
Creator   : USERID
Table Name: EMP

Column Name         Type(Length)      Where
------------------  ----------------  ----+---10----+---20----+---30----+---40--
EMPNO               CHAR(6            > '000110' OR
FIRSTNAME           VARCHAR(12)       > 'C'
MIDINIT             CHAR(1)
LASTNAME            VARCHAR(15)
WORKDEPT            CHAR(3)
PHONENO             CHAR(4)
HIREDATE            DATE
JOB                 CHAR(8)
EDLEVEL             SMALLINT
SEX                 CHAR(1)
BIRTHDATE           DATE
SALARY              DECIMAL(9,2)      > '10000'
```

From Table

Displays the Creator and Table Name of the table or view to be copied.

Column Name

Column names in the from table.

Type(Length)

Column type and length.

Where

Type the WHERE clauses used to select the rows to be copied. The field can be scrolled left and right. You may use the following delimiters in your clauses:

>	OR	< >
<	> =	BETWEEN
=	< =	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

SQL Entry Panel

The SQL Entry Panel appears when you specify **S** on the Table Copy Selection screen.

This screen prompts you for a dataset name in which you have previously saved a SELECT statement.

Figure 9-8. SQL Entry Panel

```
File-AID for DB2 ------------ SQL Entry Panel ---------------------- SSID: DSN
COMMAND ===>

Specify the input dataset name:         (Can be sequential or partitioned)
 1  DATASET NAME  ... ===> SHOW.SQL(MYSELECT)
 2  VOLUME SERIAL ... ===>              (Enter if not cataloged)
 3  DATASET PASSWORD  ===>              (Enter if password protected)
 4  MAX SELECT ROWS . ===> 2000         (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT ...... ===> YES          (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

DATASET NAME

Name of the dataset (either partitioned or sequential) in which the SQL SELECT statement resides.

VOLUME SERIAL

Disk pack on which the dataset resides if it is not cataloged.

DATASET PASSWORD

Password for the dataset if it is password protected.

MAX SELECT ROWS

Maximum number of rows to be returned for the SELECT. (An asterisk in this field indicates that all rows should be returned. See “Maximum Rows to Select” on page 6-3 for more information.) Only the first SELECT command in a dataset is executed.

EDIT INPUT

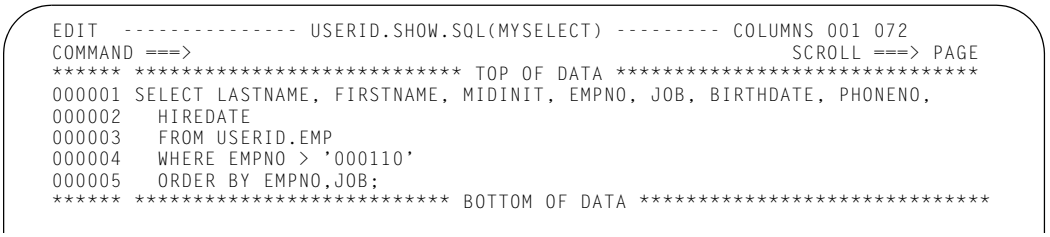
Specify whether you want to edit the SQL statement before it is executed.

Press Enter to process.

Edit SPUFI Selection

The Edit SPUFI selection screen appears if the EDIT INPUT field on the SQL Entry Panel is set to YES. This screen is also displayed if the **SQL** primary command is entered on the Copy Row Selection screen. It lets you enter a SELECT statement in standard SPUFI format. You can view, save or edit the statement using the ISPF/PDF Edit facility, adding WHERE or ORDER BY clauses or changing the selected columns.

Figure 9-9. Edit SPUFI Selection screen



Text Entry Area

Change or add to the SELECT statement using ISPF/PDF Edit.

Press END to exit SPUFI. A message is returned indicating any changes you have made and telling you to press Enter to execute the statement. Since all columns in your TO dataset are retained, the paired FROM/TO columns must be similar in type and length. Note that most similar data types are automatically converted, as follows:

SMALL	to	INTEGER
CHAR	to	VARCHAR
VARGR	to	GRAPHIC
DEC	to	SMALL
FLOAT	to	DEC
DATE	to	TIMESTAMP

Chapter 10.
Object List Utility

This chapter describes the Object List utility, which allows you to display object lists and manipulate them using simple line commands.

Access Object List Utility

When you select option **4** from the Utility Selection Menu, the Object List Utility screen appears (Figure 10-1). You can select multiple objects at once. After you press Enter, the requested objects appear; page through to see all of them.

Figure 10-1. Object List Utility Screen

```
File-AID for DB2 ------------- Object List Utility -----------------------------
 COMMAND  ===>                                                  SSID ===> D610

 Enter both of the parameters below:

   Object Name    ===> ____________________ (* for all)
   Object Type    ===> _                   (* = All or use S to select from list)

  _ Table             _ View              _ Synonym       _ Alias
  _ Index             _ Tablespace        _ Plan          _ Package
  _ Temporary table   _ Auxiliary table   _ Trigger       _ User Defined Function
  _ Distinct type     _ Stored Procedure


 Specify Selection Criteria:
   Object Creator ===> *                  (For all except Tbsp,Pkg,Trigger & SP)
   Database Name  ===> *                  (For Tablespace only)
   Collection ID  ===> *                  (For Package only)
   Schema Name    ===> *                  (For Trigger, SP, UDF & Distinct Type)

 Specify Drop Request Option:
   Confirm Drop   ===> YES                (Yes or No)
```

Object Name

Name of the objects you want listed. Wild card characters are allowed. See "Wildcard Characters" on page 2-9 for more information.

Object Type

Enter an asterisk (*****) to select all objects, or leave blank and use an **S** to select specific objects from those listed on the screen.

Object Creator

ID of person who created the objects you want to list. Does not apply to tablespaces, packages, triggers, stored procedures, distinct types, or user defined functions.

Database Name

Name of the database that contains the tablespaces you want to list. Required for an object type of Tablespace.

Collection ID

Name of the collection ID that contains the packages you want to list.

Schema Name

Schema or authid for triggers, stored procedures, UDF or distinct type you want to list.

Specify Drop Request Option

Specify whether you want a Confirm Drop screen to be displayed when you choose to drop an object. The default value is YES.

When you have entered the name and type of object you want to see, File-AID *for DB2* returns a list of those objects. You can change the order in which they are displayed by using the SORT command (see "SORT" on page 3-52 for details). Figure 10-2 shows a sample Object List selection screen for a list of tables.

Figure 10-2. Object List Selection Screen

```
File-AID for DB2 ------------ Object List Selection ---------- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: D610

E - Edit   B - Browse   A - Alter   D - Drop   G - Grant   R - Revoke  GD - DDL
L - Load   X - Extract  P - Privileges  PM - Privilege Modify  I - Information

Cmd  Name                          Creator   Type          Message
------------------------------------------------------------------------------
__   FADB2610.up ACT               PFHJMBO   PACKAGE
__   FADB2610.ADD_DEPT             PFHJMBO   PACKAGE
__   FADB2610.ADD_EMP              PFHJMBO   PACKAGE
__   FADB2610.LOWERCSE             PFHJMBO   PACKAGE
__   FADB2610.SQLDEM2              DFHCAP1   PACKAGE
****************************** Bottom of data ********************************
```

CMD

Specify the line command you want to invoke for an object. Valid commands are listed across the top of the screen.

NAME

Names of the objects in the list.

CREATOR

ID of person who created the listed objects.

TYPE

Type of the listed objects. For stored procedure, includes both language and SP (right justified). When sorting on stored procedures, the sort goes first by SP, then by the language.

Note: File-AID *for DB2* cannot display additional information for stored procedures with language COMPJAVA or JAVA (for DB2 version 7 and 8). If you enter a line command for a line with language COMPJAVA or JAVA, the error message 'No Commands Allowed' appears.

MESSAGE

Messages (if any) for the listed objects. Indicates last action. For UDFs, displays the specific name of the function.

Browse Object

The Browse option of the Object List utility operates just as the Browse function (option 1 on the Primary Option Menu) does. The Browse Options screen has fewer fields, however, and the object name and creator fields are not modifiable. The object types you can browse are the following:

- Tables
- Aliases
- Views
- Synonyms

If you need information on browsing, see Chapter 6, “Browse and Edit”.

Edit Object

The Edit option of the Object List utility operates just as the Edit function (option 2 on the Primary Option Menu) does. The Edit Options screen has fewer fields, however, and the table name and creator fields are not modifiable. The object types you can edit are the following:

- Tables
- Aliases
- Views
- Synonyms

If you need information on editing, see Chapter 6, “Browse and Edit”.

Drop Object

The Drop option of the Object List utility operates just as it does in the Objects utility (option 3.2 on the Primary Option Menu). The object types you can drop are the following:

- Tables
- Tablespaces
- Indexes
- Views
- Synonyms
- Aliases
- Temporary Tables
- Triggers
- Distinct Types
- User Defined Functions
- Stored Procedures

If you need information on dropping an object, see Chapter 8, “Objects Utility”.

Alter Object

The Alter option of the Object List utility operates just as it does in the Objects utility (option 3.2 on the Primary Option Menu). The object types you can alter are the following:

- Tables
- Tablespaces

- Indexes
- User Defined Functions
- Stored Procedures

If you need information on altering an object, see Chapter 8, “Objects Utility”.

Grant Privileges on Object

The Grant option of the Object List utility operates just as it does in the Display, Grant, Revoke, and Modify DB2 Privileges utility (option 3.6 on the Primary Option Menu). You can grant privileges for the following object types:

- Tables
- Tablespaces
- Views
- Plans
- Packages
- Distinct Types
- User Defined Functions
- Stored Procedures

If you need information on granting privileges, see Chapter 12, “Privileges Utility”.

Revoke Privileges on Object

The Revoke option of the Object List utility operates just as it does in the Display, Grant, Revoke, and Modify DB2 Privileges utility (option 3.6 on the Primary Option Menu). You can revoke privileges for the following object types:

- Tables
- Tablespaces
- Views
- Plans
- Packages
- Distinct Types
- User Defined Functions
- Stored Procedures

If you need information on revoking privileges, see Chapter 12, “Privileges Utility”.

Generate DDL Statements

The Generate DDL option of the Object List utility generates DDL statements for the selected object. You can generate DDL for the following object types:

- Tables
- Tablespaces
- Views
- Indexes
- Synonyms
- Aliases
- Triggers
- Distinct Types
- User Defined Functions
- Stored Procedures

You cannot use the GD command for the following stored procedures and functions:

- Stored procedures created with TABLE LIKE input parameter
- Functions created with the TABLE LIKE input parameter
- System generated CAST functions
- Sourced functions
- Functions created with the CAST FROM return parameter

Load Table

The Load function of the Object List utility operates just as it does in the Extract, Load Tables utility (option 3.7 on the Primary Option Menu). You can use the load function for the following objects types:

- Tables
- Aliases
- Views

If you need information on loading objects, see Chapter 14, “Extract and Load Utility”.

Extract from Table

The Extract function of the Object List utility operates just as it does in the Extract, Load Tables utility (option 3.7 on the Primary Option Menu). You can use the Extract function for the following object types:

- Tables
- Aliases
- Views

If you need information on extracting objects, see Chapter 14, “Extract and Load Utility”.

Display Privileges

The Privileges option of the Object List utility operates just as it does in the Display, Grant, Revoke, and Modify DB2 Privileges utility (option 3.6 on the Primary Option Menu). You can display privileges for the following object types:

- Tables
- Tablespaces
- Views
- Plans
- Packages
- Distinct Types
- User Defined Functions
- Stored Procedures

If you need information on this function, see Chapter 12, “Privileges Utility”.

Privilege Modify

The Privilege Modify option of the Object List utility operates just as it does in the Display, Grant, Revoke, and Modify DB2 Privileges utility (option 3.6 on the Primary Option Menu). You can display privileges for the following object types:

- Tables
- Tablespaces
- Views
- Packages
- Plans
- Distinct Types
- User Defined Functions
- Stored Procedures

If you need information on modifying privileges, see Chapter 12, “Privileges Utility”.

Object Information

The Information option of the Object list utility displays creator and parameter information for an object. You can display information for the following object types:

- Tables
- Temporary Tables
- Auxiliary Tables
- Tablespaces
- Indexes
- Views
- Plans
- DBRMs
- Synonyms
- Aliases
- Packages
- Triggers
- User Defined Function
- Distinct Types
- Stored Procedures

When you enter **I** on the Object List selection screen (Figure 10-2 on page 10-2), an Information display screen is shown. A unique screen displays for each object type.

Table

Figure 10-3 on page 10-7 shows a sample Information screen for a table or global temporary table. Additional column, index, and related table information can be selected from this screen by placing an **S** next to the desired field.

Note: You can use the tab key to go between selection list items by row between columns (Column Information, then Table Partition Statistics, then back to Index information). Alternatively, you can use the new line key (consult your keyboard mapping for the location of this key) to go between selection list items for the first column only (such as Column Information, then Index Information).

Figure 10-3. Table Information Display Screen

```
File-AID for DB2 ------------ Table Information ----------------- SSID: D610
COMMAND ===>

Use 'S' to select more information for Table: USERID1.TRIGTAB
 _  Column Information                  _ Table Partition Statistics
 _  Index Information                   _ Table Check Constraints
 _  Related Table Information           _ Remarks/Comment On
 S  Related Trigger Information

CREATEDBY: BFHUSER1   PCTPAGES: -1      AUDITING  : N
COLCOUNT : 2          KEYCOLS : 0       CHECKRID  : <NONE>
DBNAME   : BFHUSRDB   KEYOBID : 0       CHECKFLAG : <NONE>
TSNAME   : BFHUSRTS   STATUS  : <NONE>  RBA1      : 0003F58716AC
CARD     : -1         PARENTS : 0       RBA2      : 0003F58716AC
DBID     : 474        CHILDREN: 0       CREATEDTS : 2001-04-25-11.45.15.525438
OBID     : 170        DATACAPT: N       ALTEREDTS : 2001-04-25-11.45.15.525438
TYPE     : T          PCTRWCMP: -1      STATSTIME : 0001-01-01-00.00.00.000000
RECLENGTH: 53         EDPROC  : <NONE>     CHECKS : 0
NPAGES   : -1         VALPROC : <NONE>  LOCATION  : XXXXXXXXXXXXXXXX
ENCSCHEME: X          RELCRETD: X       CARDF     : -1.000000E+00
TABSTATUS: XXXXXXXXXX                   NPAGESF   :
AVGROWLEN: XXXXXXXXXX                   SPACEF    :
LABEL    : <NONE>
```

Additional Table Information Selection Area

Enter the **S** line command in front of the following fields:

Column Information

Displays the Column Information screen (Figure 16-33 on page 16-30).

Index Information

Displays the Index Information screen (Figure 16-38 on page 16-32).

Related Table Information

Displays the Related Table Information screen (Figure 16-41 on page 16-34).

Related Trigger Information

Displays a Trigger Selection screen from which you can select one or more Trigger Information screens (Figure 10-7 on page 10-23).

Table Partition Statistics

Displays the Table Partition Statistics screen (Figure 16-42 on page 16-34).

Table Check Constraints

Displays the Table Check Constraints Statistics screen (Figure 16-43 on page 16-35).

Remarks/Comment On

Displays the Remarks/Comment On for the table.

CREATED BY

Creator ID of person who created the table. (This is not necessarily the same as CREATOR.)

COLCOUNT

Number of columns within the table.

DBNAME

Name of the database in which the table resides.

TSNAME

Name of the tablespace in which the table resides.

CARD

Total number of rows in the table.

DBID

Identifier of the database in which the table resides.

OBID

Identifier of the table.

TYPE

Type of Object: A for alias, T for table, V for view.

RECLENGTH

Maximum length any record in the table may be.

NPAGES

Total number of pages that contain rows of the table.

ENCSCHEME

Encoding scheme for tables, views, and local aliases:

E	EBCDIC
A	ASCII
U	UNICODE
blank	For remote aliases

The value is **E** for tables in non-work file databases and **blank** for tables in work file databases created prior to version 5 or the default database, DSNDB04.

TABSTATUS

The reason for an incomplete table definition.

L	Incomplete because an auxiliary table or auxiliary index has not been defined for an LOB column.
P	Incomplete because the table lacks a primary index.
R	Incomplete because the table lacks a required index on a row ID.
U	Incomplete because the table lacks a required index on a unique key.
blank	Definition is complete.

AVGROWLEN

(DB2 Version 7 and later) Average length of rows for the tables in the table space. If the table space is compressed, the value is the compressed row length. If the table space is not compressed, the value is the uncompressed row length. The value is -1 if statistics have not been gathered.

LABEL

The label as given by a LABEL ON statement.

PCTPAGES

Percentage of the total number of pages in the tablespace that contain rows of the table.

KEYCOLS

Number of columns the table has in its primary key.

KEYOBID

DB2 identifier of the index or link that ensures that the table's primary key is unique.

STATUS

Table status.

PARENTS

Number of relationships in which the table is a dependent.

CHILDREN

Number of relationships in which the table is a parent.

DATACAPT

Records the value of the DATA CAPTURE option for a table. N indicates NO and Y indicates YES.

PCTRWCMP

Percentage of rows, multiplied by 100, compressed within the total number of active rows in the table.

EDPROC

Name of the program that encrypts, decrypts, and/or compresses the table data for storage and/or retrieval.

VALPROC

Name of the program that validates new data added to the table.

RELCRETD

(DB2 Version 7 and later) Release of DB2 used to create the object:

blank	Created prior to version 7
K	Created on version 7

AUDITING

Audit option.

CHECKRID

RID of the first row of any tablespace partition that violates referential constraints, if any.

CHECKFLAG

Indicates whether the table contains referential constraint violations.

RBA1

The log RBA when the table was created.

RBA2

The log RBA when the table was last altered.

CREATEDTS

Time when the CREATE statement was executed for the table, view, or alias.

ALTEREDTS

For a table, time when the latest ALTER TABLE statement was applied. If no ALTER TABLE statement has been applied, or if the row is for a view or alias, this field has the same value as CREATEDTS.

STATSTIME

If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics.

CHECKS

Identifies the number of check constraints defined on the table. If the row describes a view or an alias or if no constraints are defined on the table, it is 0.

LOCATION

Location name of the object of an alias. Blank for a table, a view, or for an alias not defined with a three-part object name.

CARDF

Total number of rows in the table. The value is -1 if statistics have not been gathered or the row describes a view, alias, or temporary table.

NPAGESF

(DB2 Version 7 and later) Number of pages used by the table. The value is -1 if statistics have not been gathered. This column is updatable.

SPACEF

(DB2 Version 7 and later) Kilobytes of DASD storage. The value is -1 if statistics have not been gathered. This column is updatable.

Tablespace

The following fields appear if you are displaying information for a tablespace:

NAME

Name of the tablespace.

CREATOR

Creator ID of the tablespace.

DBNAME

Name of the database in which the tablespace resides.

DBID

Identifier of the database in which the tablespace resides.

OBID

Identifier of the tablespace file descriptor.

PSID

Identifier of the tablespace.

BPOOL

Name of the buffer pool used for the tablespace.

PARTITIONS

Number of partitions in a partitioned tablespace. This value is 0 if it is a nonpartitioned tablespace.

LOCKRULE

Lock size of the tablespace.

PGSIZE

Size, in kilobytes, of a page within the tablespace.

ERASERULE

Indicates whether the data is to be erased when it is dropped.

STATUS

Availability of the tablespace.

IMPLICIT

Indicates whether the tablespace was created implicitly.

NTABLES

Number of tables contained in the tablespace.

NACTIVE

Number of active pages in the tablespace.

DSETPASS

Password for the datasets of the tablespace.

CLOSERULE

Indicates whether the datasets are to be closed when the tablespace is not in use.

SPACE

Amount, in kilobytes, of DASD storage allocated to the tablespace.

ROOTNAME

Unqualified name of the root table of the space. If the tablespace is not structured, the value is blank.

ROOTCREATOR

ID of person who created the root table. If the tablespace is not structured, the value is blank.

SEGSIZE

Number of pages in each segment of a segmented tablespace.

CREATED BY

Primary creator ID of person who created the tablespace.

LOCKMAX

The maximum number of locks per user to acquire for the table or tablespace before escalating to the next locking level. 0 = Lock escalation does not occur. n, where n > 0, = the number of locks (row or page locks for the table or table space) an application process can acquire before lock escalation occurs. -1 = represents LOCKMAX SYSTEM. The value of field LOCKS PER TABLE(SPACE) on installation panel DSNTIPJ determines lock escalation. If the value of the field is 0, lock escalation does not occur. If the value is n, where n > 0, lock escalation occurs as it does for LOCKMAX n.

Index

Figure 10-4 shows a sample information screen for an index. Index Key and Partition information can be selected by placing an **S** next to the desired field.

Figure 10-4. Index Information Display screen

```
File-AID for DB2 ------------- Index Information ------------------ SSID: DSN
COMMAND  ===>

Enter 'S' for additional Index Information:

 _ Index Key Information
 _ Index Partition Information

Creator   : TSOID01            Database  : DATABASE
Table Name: EMP                Tablespace: TABLESPC

Index Information for: TSOID01.XEMP1

CREATEDBY    : SYSIBM     INDEXSPACE : DSNIDX     NLEAF     : 173
UNIQUERULE   : TSOID01    DBNAME     : DSNDBNE    NLEVELS   : 3
COLCOUNT     : 4          BPOOL      : BP0        PAGESIZE  : 4096
CLUSTERING   : N          DBID       : 6          FRSTKYCRD : 141
CLUSTERED    : N          OBID       : 163        FULLKYCRD : 4962
CLUSTERRATIO : 81         ISOBID     : 164        ERASERULE : N
DASDSPACE    : 0          DSETPASSW  :            CLOSERULE : N
STATSTIME    : 1994-01-04-12.05.32.000100        INDEXTYPE :
FIRSTKEYCARDF:                                   PIECESIZE : 2097152
FULLKEYCARDF :                                   COPY:
```

INDEX KEY INFORMATION

Displays the Index Key Information screen (Figure 16-39 on page 16-33).

INDEX PARTITION INFORMATION

Displays the Index Partition Information screen (Figure 16-40 on page 16-33).

CREATEDBY

Primary creator ID of person who created the index.

UNIQUERULE

Indicates whether the index is unique.

COLCOUNT

Number of columns in the key.

CLUSTERING

Indicates whether CLUSTER was specified when the index was created.

CLUSTERED

Indicates whether the table is actually clustered by the index.

CLUSTERRATIO

Percentage of rows that are in clustering order.

DASDSPACE

Amount, in kilobytes, of DASD storage space allocated to the index.

STATSTIME

If RUNSTATS updated the statistics, the date and time when the last invocation of RUNSTATS updated the statistics.

INDEXSPACE

Name of the indexspace in which the index resides.

FIRSTKEYCARDF

Number of distinct values of the first key column. This number is an estimate if updated while collecting statistics on a single partition. The value is -1 before statistics are gathered.

FULLKEYCARDF

Number of distinct values of the key. The value is -1 before statistics are gathered.

DBNAME

Name of the database in which the index resides.

BPOOL

Name of the bufferpool used for the index.

DBID

Internal identifier of the database in which the index resides.

OBID

Internal identifier of the index.

ISOBID

Internal identifier of the indexspace.

DSETPASSW

Password for the datasets of the index.

NLEAF

Number of active leaf pages in the index.

NLEVELS

Number of levels in the index tree.

PAGESIZE

Size of subpages within the index.

FRSTKYCRD

Number of distinct values of the first 8 bytes of the first key column or of the entire column if it is narrower than 8 bytes.

FULLKYCRD

Number of distinct values of the key.

ERASERULE

Indicates whether the index is to be erased when it is dropped.

CLOSERULE

Indicates whether datasets are to be closed when the index is not in use.

INDEXTYPE

Indicates the index type. blank = Type 1 index. 2 = Type 2 index.

PIECESIZE

Indicates maximum size of dataset storage piece, in kilobytes (K), that is used by DB2 for nonpartitioned indexes.

COPY

Indicate whether the COPY utility is allowed for the index.

NO Does not allow full image or concurrent copies or the use of the RECOVER utility on the index. NO is the default.

YES Allows full image or concurrent copies and the use of the RECOVER utility on the index.

View

The following fields are displayed if you are displaying information for a view:

NAME

Name of the view.

CREATOR

Creator ID of the view.

DBNAME

Name of the database in which the view resides.

TSNAME

Name of the tablespace in which the view resides.

COLCOUNT

Number of columns in the view.

CHECK

If an underlying table violates referential integrity constraints.

CREATED BY

Primary creator ID of person who created the view.

REMARKS

Remarks (if any) that the creator included through the use of the COMMENT statement.

TEXT

Text (or portion of text) of the CREATE VIEW statement. The text is scrollable.

Plan

The following fields are displayed if you are displaying information for a plan:

NAME

Name of the plan.

CREATOR

Creator ID of the plan.

BINDDATE

Date, in *yymmdd* format, when the last BIND or REBIND was performed for the plan.

VALIDATE

Indicates whether validity checking is performed at run time or bind time.

ISOLATION

Isolation level.

VALID

Indicates whether the application plan is valid (that is, it can be run without rebinding).

OPERATIVE

Indicates whether the application plan can be allocated.

BINDTIME

Time, in *hhmmssth* format, of the last BIND.

GROUP MEMBER

The DB2 data sharing member name of the DB2 subsystem that performed the most recent bind. This column is blank if the DB2 subsystem was not in a DB2 data sharing environment when the bind was performed.

KEEPDYN

KEEPDYNAMIC, YES or NO. Indicates whether prepared dynamic SQL statements are kept past commit (YES) or destroyed at commit (NO).

BOUNDTS

Timestamp, in *yyyy-mm-dd-hh.mm.ss.tttttt* format, of the last BIND.

PLSIZE

Size of the base section of the plan. DB2 uses this figure to allocate storage for the control structure.

IBMREQD

Indicates whether the row came from the basic tape.

AVGSIZE

Average size of those sections that contain DML statements processed at BIND time.

ACQUIRE

Indicates when resources are acquired.

RELEASE

Indicates when resources are released.

EXPLAIN

Indicates whether the plan was bound with EXPLAIN YES.

BOUND BY

Primary user ID of the person who is the binder of the plan.

QUALIFIER

Default high-level qualifier to use for unqualified table name references.

DYNAMIC RULE

Dynamic Rules.

B

Dynamic SQL statements are handled like static SQL statements at run time.

R

Dynamic SQL statements are handled like dynamic SQL statements at run time.

blank

DYNAMICRULES is not specified for the package. The package uses the DYNAMICRULES value of the plan to which the package is appended at execution time.

REOPTVAR

Reoptimization Variable. Identifies whether the access path is determined again at execution time using input variable values.

N

Bind option NOREOPT(VARS) indicates that the access path is determined at bind time.

Y

Bind option REOPT(VARS) indicates that the access path is determined at execution time for SQL statements with variable values.

You have the option of viewing DBRM and Package Information. If you specify YES the Information for Package (see "Package" on page 10-18) or the Information for DBRM (see "DBRM" on page 10-16) screen appears. If more than one DBRM or package is in the plan, select one from the Selection List.

DBRM

DBRM NAME

Name of the DBRM.

TIMESTAMP

Time of precompilation in internal format (shown as hex characters).

PLNAME

Current PLAN name.

PLCREATOR

Authid of creator of this plan.

PRECOMPTIME

Time of precompilation in the form *hhmmssth*.

PRECOMPDATE

Date of precompilation in the form *yymmdd*.

QUOTE

SQL escape character:

N

Apostrophe

Y

Quotation mark

PDSNAME

Name of the partitioned dataset of which the DBRM is a member.

VERSION

Version ID assigned when plan was bound.

PRECOMPTS

Timestamp when the DBRM was precompiled.

COMMA

Indicates whether the decimal point is a period or a comma:

N

Period

Y

Comma

HOSTLANG

The host language used:

B

Assembler

C

COBOL

D

C

F

Fortran

P

PL/1

2

VS COBOL II

3

IBM COBOL (Release 2 or later)

4

C++

IBMREQD

Indicates whether the row came from the basic tape.

CHARSET

K indicates KATAKANA character set yes. A indicates KATAKANA character set no.

MIXED

Mixed data option used at precompile time.

DEC31

Decimal 31 option used at precompile time.

Synonym

The following fields are displayed if you are displaying information for a synonym:

NAME

Name of the synonym.

CREATOR

Creator ID of the synonym.

TBNAME

Name of the table or view to which the synonym applies.

TBCREATOR

Creator ID of person who created the table or view to which the synonym applies.

Alias

The following fields are displayed if you are displaying information for a alias:

NAME

Name of the alias.

CREATOR

Creator ID of the alias.

LABEL

Label assigned to alias.

LOCATION

LOCATION ID for this alias. Blank means LOCAL.

TBCREATOR

Contains the high-level qualifier of the table or view to which the alias applies.

TBNAME

Contains the name of the table or view to which the alias applies.

CREATED BY

Contains the primary ID of the person who created the alias.

CREATEDTS

Timestamp when alias was created (yy-mm-dd-hh.mm.ss.ttttt).

REMARKS

Remarks for the alias.

Package

The following fields are displayed if you are displaying information for a package:

PATHSCHEMAS

The SQL path specified on the bind or rebind command that bound the package. When you select this option using S, the PathSchemas Information for Package screen appears and shows this information.

Figure 10-5. PathSchemas Information for Package Screen

```
File-AID for DB2 ------ PathSchemas Information for Package -------------------
 COMMAND ===>

 Pathschemas:

 "SYSIBM","SYSFUN","SYSPROC","PFHCZK1"
```

PACKAGE NAME

Name of the package.

COLLID

Collection ID containing the package.

CONTOKEN

Hex value of consistency token.

OWNER

Authid of the owner of the package.

CREATOR

Authid of the creator of the package.

TIMESTAMP

Time package first bound (yy-mm-dd-hh.mm.ss.tttt).

BINDTIME

Time package last bound (yy-mm-dd-hh.mm.ss.tttt).

QUALIFIER

Default high-level qualifier to use for unqualified table name references.

PKSIZE

Indicates the size of the base section of the package. DB2 uses this figure to allocate storage for the package.

AVGSIZE

Indicates the average size of those sections that contain DML statements processed at BIND time.

DEGREE

The DEGREE option used when the package was last bound; ANY for DEGREE(ANY), and 1 or blank for DEGREE(1).

PCTIMESTAMP

Time of pre-compile (yy-mm-dd-hh.mm.ss.tttt).

GROUP MEMBER

The DB2 data sharing member name of the DB2 subsystem that performed the most recent bind. This column is blank if the DB2 subsystem was not in a DB2 data sharing environment when the bind was performed.

PDSNAME

Name of dataset containing DBRM (precompiler output).

VERSION

Version ID assigned when package was bound.

FUNCTIONTS

Timestamp when the function was resolved.

OPTHINT

Value of the OPTHINT bind option. Identifies rows in the authid.PLAN_TABLE to be used as input to the optimizer.

ENCODE CCSID

(DB2 version 7 and later) The CCSID corresponding to the encoding scheme or CCSID as specified for the bind option ENCODING.

ISOLATION

Indicates the isolation level. S=Cursor Stability.

RELEASE

Indicates when resources are released.

EXPLAIN

Indicates if package bound with Explain.

HOSTLANG

The host language used.

B	Assembler
C	COBOL
D	C
F	Fortran
P	PL/1
2	VS COBOL II
3	IBM COBOL (Release 2 or later)
4	C++

SYSENTRIES

Count of system entries.

DEFERPREP

Indicates the CURRENTDATA option when the package was bound or rebound.

A	Data currency is required for all cursors. Inhibit blocking for all cursors.
B	Data currency is not required for ambiguous cursors.
C	Data currency is required for ambiguous cursors.
blank	The package was created before CURRENTDATA option was available.

DEFERPREPARE

Identifies whether PREPARE processing is deferred until OPEN is executed.

N	Bind option NODEFER(PREPARE) indicates that PREPARE processing is not deferred until OPEN is executed.
Y	Bind option DEFER(PREPARE) indicates that PREPARE processing is deferred until OPEN is executed.
blank	Bind option not specified for the package. It is inherited from the plan.

SQLERROR

Bind option setting to permit errors during bind.

REOPTVAR

Reoptimization Variable. Identifies whether the access path is determined again at execution time using input variable values.

N	Bind option NOREOPT(VARS) indicates that the access path is determined at bind time.
Y	Bind option REOPT(VARS) indicates that the access path is determined at execution time for SQL statements with variable values.

KEEPDYNAMIC

KEEPDYNAMIC, YES or NO. Indicates whether prepared dynamic SQL statements are kept past commit (YES) or destroyed at commit (NO).

VALID

Indicates whether the package is valid.

OPERATIVE

Indicates whether the package can be allocated.

VALIDATE

Indicates validation point - B-bind, R-run.

DYNAMIC RULES

B	Dynamic SQL statements are handled like static SQL statements at run time.
R	Dynamic SQL statements are handled like dynamic SQL statements at run time.
blank	DYNAMICRULES is not specified for the package. The package uses the DYNAMICRULES value of the plan to which the package is appended at execution time.

DBPROTOCOL

Whether remote access for SQL with 3-part names is implemented with DRDA (D) or DB2 private protocol access (P).

IMMEDWRITE

(DB2 version 7 and later) When to do writes of updated group buffer pool dependent pages. Only applicable for data sharing environments.

N	Normal write activity
Y	Immediate writes for updated group buffer pool dependent pages.
1	Updated group buffer pool dependent pages are written at or before phase 1 commit.
blank	Migrated package.

RELBOUND

(DB2 version 7 and later) The release when the package was bound or rebound. Blank for bound prior to version 7, K for bound on version 7.

QUOTE

SQL string delimiter - NO=Default, YES=Single quote

COMMA

Comma means decimal point (YES). Period is decimal (NO).

MIXED

Mixed data option used at precompile time.

DEC31

Decimal 31 option used at precompile time.

REMOTE

Indicates package was bound on a remote system.

CHARSET

K indicates KATAKANA character set yes. A indicates KATAKANA character set no.

TYPE

Type of package; how the package was created. Blank for created with a BIND PACKAGE command, T for created using a CREATE TRIGGER statement.

Auxiliary Table

Figure 10-6 shows a sample Auxiliary Information screen for an auxiliary table.

Figure 10-6. Auxiliary Table Information Display Screen

```
File-AID for DB2 -------- Information for Auxiliary Table ----------------------
COMMAND ===>

INFORMATION FOR AUXILIARY TABLE:

Base TBOwner       : TSOID01

Base TBName        : EMP_PHOTO_RESUME

LOB Column Name    : BMP_PHOTO

Auxiliary TBOwner  : TSOID01

Auxiliary TBName   : AUX_BMP_PHOTO
```

The following fields are displayed if you are displaying information for an auxiliary table:

Base TBOwner

Authid of owner of the base table.

Base TBName

Name of the base table for this Auxiliary Table.

LOB Column Name

Name of the LOB column in the base table.

Auxiliary TBOwner

Authid of owner of this Auxiliary Table.

Auxiliary TBName

Name of this Auxiliary Table.

Trigger

Figure 10-7 shows a sample Information screen for a trigger.

Figure 10-7. Trigger Information Display Screen

```
File-AID for DB2 ------------- Trigger Information --------------- SSID: D610
COMMAND ===>
Specify 'S' to Select more information for Trigger: BFHUSER1.TRIGDEPT

S Trigger Text
_ Remarks/Comment On

Createdby : FDSQLID  TBName  : XXXXXXXXXXXXXXXXX
Owner     : FDSQLID  TBOwner : XXXXXXXX

Trigger Time : X     DBID : XXXXXX
Trigger Event: X     OBID : XXXXXX
Granularity  : X

Createdts : 2001-08-20-13.15.00.556173
```

The following fields appear if you are displaying information for a trigger:

Trigger Text

Brings up a separate screen showing any text associated with the trigger. Use S to select.

Remarks/Comment On

Brings up a separate screen showing any remarks associated with the trigger. If no remarks exist, the message 'No Remarks Exist' appears in the upper right-hand corner of the screen.

Created By

Authid of the creator of the trigger.

Owner

Authid of the owner of the trigger.

TBName

Name of the table to which this trigger applies.

TBOwner

Qualifier of the name of the table to which this trigger applies.

Trigger Time

Time when triggered actions are applied to the base table, relative to the event that activated the trigger:

B Trigger is applied before the event.

A Trigger is applied after the event.

Trigger Event

Operation that activates the trigger:

I Insert

D Delete

U Update

Granularity

Trigger is executed once per:

S Statement

R Row

DBID

Identifier of the database in which the table resides.

OBID

Identifier of the table.

Createdts

Time when the CREATE statement was executed for this trigger. Multiple triggers appear in ascending order. Triggers of the same type are executed in the order they were created. A trigger cannot be altered, so a drop and recreate of a trigger can potentially change the order in which triggers are executed.

Trigger Text

The Trigger Text screen appears when you enter an **S** in the Trigger Text field on the Trigger Information screen.

Figure 10-8. Trigger Text Screen

```
+-------------------------------------------------------------------------------+
| File-AID for DB2 ------------- Trigger Text --------------- Row 1 to 9 of 9 |
| COMMAND ===>                                                                  |
|                                                                               |
| There were  1    SEQNO's                                                      |
|                                                                               |
| Text for Trigger:  FADB2610.ADD_EMP                                           |
|                                                                               |
|                                                                               |
|  CREATE TRIGGER FADB2610.ADD_EMP                                              |
|     AFTER INSERT ON FADB2610.EMP                                              |
|     REFERENCING NEW AS NEWEMP                                                 |
|     FOR EACH ROW                                                              |
|     MODE DB2SQL                                                               |
|     INSERT INTO FADB2610.ACT                                                  |
|     VALUES (INTEGER(NEWEMP.EMPNO),                                           |
|     NEWEMP.EMPNO,                                                             |
|     NEWEMP.LASTNAME)                                                          |
| ***************************** Bottom of data ****************************** |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

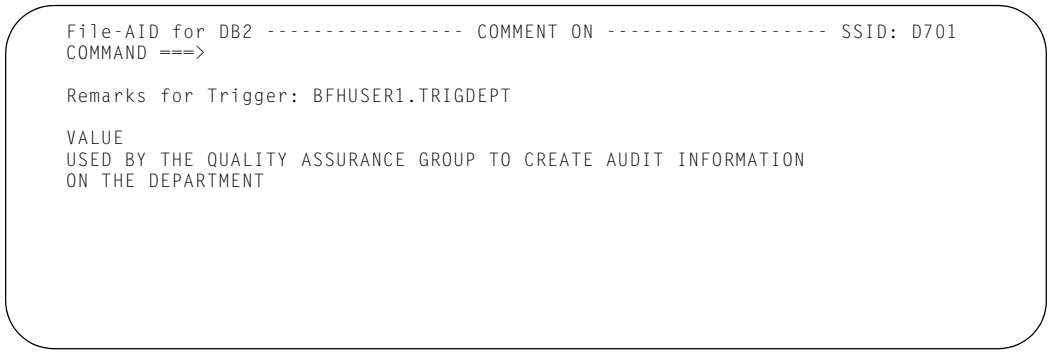
SEQNO

Sequence number of this row; the first portion of the trigger definition is in row 1, and successive rows have increasing SEQNO values. In this example, the text was across three rows in the SYSIBM.SYSTRIGGERS DB2 catalog table.

Comment On

The Comment On screen appears when you enter an **S** in the Remarks/Comment On field on the Trigger Information screen.

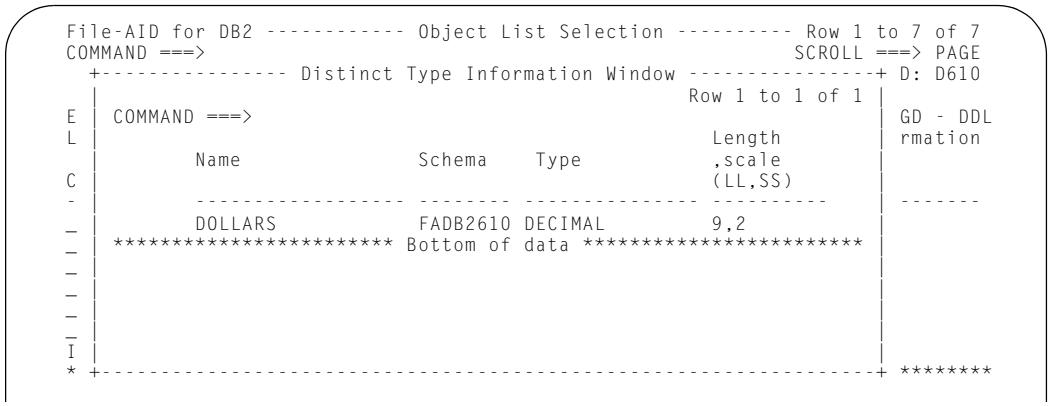
Figure 10-9. Comment On Screen



Distinct Type Information

Figure 10-10 shows a sample Distinct Type Information window for a distinct type. This window appears when you use the command UDT.SCHEMA.NAME.

Figure 10-10. Distinct Type Information window



Name

Column name that is a distinct type.

Schema

Schema or authid for the distinct type.

Type

Data type for this column.

Scale

Length and scale of the column data type.

Stored Procedure

Figure 10-11 shows a sample information screen for a stored procedure.

Figure 10-11. Stored Procedure Information Screen

```
File-AID for DB2 ----------- Stored Procedure Information --------- SSID: D610
COMMAND ===>
Stored Procedure: FADB2610.EURO                            Language: ASSEMBLER

Specify 'S' to select more information:
 _ Parameter Information
 _ Run Time Information
 _ Remarks/Comment On

Createdby   : PFHJGDO     Origin    : E   External Name   : EURO
Owner       : PFHJGDO     Null Call : Y   Collection ID   :
Program Type: M           Fenced    : Y   Specific Name   : EURO

Result Sets : 0           DBinfo    : N   Commit on Return: N
Parm Count  : 3           SQL Data  : M   Deterministic   : N
Asutime     : 0           Parm Style: D   Stay Resident   : N
LOBcolumns  : 0           Security  : D

Createdts   : 2002-12-19-13.59.30.922475  Routineid       : 1785
Alteredts   : 2002-12-19-13.59.30.922475  WLM Environment :
                                          WLM Env Nested  : N
```

The following fields appear if you are displaying information for a stored procedure:

Parameter Information

When you select this option, the Parameter Information screen appears (Figure 10-12 on page 10-29).

Run Time Information

When you select this option, the Run Time Options window appears (Figure 8-61 on page 8-82).

Remarks/Comment On

When you select this option, the Comment On screen appears (Figure 8-62 on page 8-82).

Createdby

Primary creator ID of person who created the stored procedure.

Origin

Origin of the routine; E for stored procedure.

External Name

Eight character module name.

Owner

Authid of the owner of the stored procedure.

Null Call

The CALLED ON NOT NULL INPUT option of a stored procedure.

N Stored procedure is not called if any parameter has a NULL value.

Y Stored procedure is called if any parameter has a NULL value.

blank Stored procedure is a function, but not an external function.

Collection ID

Name of the collection ID that contains the stored procedure.

Program Type

Either M(ain) or S(ub).

M Run the stored procedure as a main routine.

S Run the stored procedure as a subroutine.

Fenced

Indicates that this routine runs separately from the DB2 address space. The values that appear are defaults and are currently the only allowable OS/390 values.

Specific Name

Specific name of the stored procedure.

Result Sets

A synonym for Dynamic Result Sets. A number between 0 and 32767.

DBINFO

Default is N. If Y, then an additional argument is passed when the stored procedure is invoked. The argument may contain information such as the application run-time authorization ID or schema name.

Commit on Return

Default is N. If Y, then DB2 commits the transaction immediately on return from the stored procedure.

Parm Count

Number of parameters for the stored procedure.

SQL Data

Default is MODIFIES. Indicates whether the stored procedure can execute any SQL statements, and if so, what type.

N(o SQL) Stored procedure cannot run any SQL statements.

M(odifies) Stored procedure can run any SQL statement except statements not supported in any stored procedure.

R(eads) Stored procedure cannot run SQL statements that modify data.

C(ontains) Stored procedure cannot run any SQL statements that read or modify data.

Deterministic

Default is N. If Y, then the stored procedure returns the same result from successive calls that have identical input arguments.

ASUTIME

Default is NO LIMIT. Specifies the total amount of processor time (in CPU service units) that a single run of the stored procedure can use.

Parm Style

Default is DB2SQL. The linkage convention used to send parameters to the stored procedure.

D(B2SQL)	Sends the parameters on the CALL statement to the stored procedure, along with a null indicator for each parameter on the CALL statement, the SQLSTATE to return to DB2, the qualified and specific name of the stored procedure, and the SQL diagnostic string to return to DB2.
G(eneral)	Sends only the parameters on the CALL statement to the stored procedure.
N(general with nulls)	Sends the parameters on the CALL statement to the stored procedure, along with an additional argument containing a record of null indicators for each of the parameters on the CALL statement related to null parameter values.

Stay Resident

Default is N. If Y, then the stored procedure load module remains resident in member after the stored procedure ends.

Special Register

Describes how special register values are inherited.

I(nherit)	The values of special registers are inherited based on rules listed in the table for characteristics of special registers (see *DB2 SQL Reference Manual*)
D(efault)	The values of special registers are initialized to the default values.

LOBcolumns

The number of LOB columns found in the parameter list.

Security

Default is DB2. Specifies how the stored procedure works with external security products.

D(B2)	No special external security environment needed. If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID associated with the stored procedure address space. If NO WLM ENVIRONMENT is set, then SECURITY DB2 is the only valid value.
U(ser)	An external security environment should be established for the stored procedure. If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID of the user who invoked the stored procedure.

C(definer) An external security environment should be established for the stored procedure. If the stored procedure accesses resources protected by an external security environment, then access is done using the authorization ID of the owner of the stored procedure.

Createdts

Timestamp when stored procedure was created (yy-mm-dd-hh.mm.ss.ttttt).

Routineid

Internal identifier of the stored procedure.

Alteredts

For a stored procedure, time when the latest ALTER PROCEDURE statement was applied. If no ALTER PROCEDURE statement was applied, this field has the same value as CREATEDTS.

WLM Environment

Identifies the MVS workload manager environment in which to run the stored procedure. To use a WLM environment, you must have authority for that environment.

WLM Env Nested

For nested stored procedure calls, indicates whether the address space of the calling stored procedure is used to run the nested stored procedure:

N The nested stored procedure does not run in the specified WLM environment if the calling stored procedure is not running in the specified WLM environment. WLM ENVIRONMENT name was specified.

Y The nested stored procedure runs in the same WLM environment as the calling stored procedure. WLM ENVIRONMENT(name,*) was specified.

blank WLM_ENVIRONMENT is blank.

Parameter Information

To get the source type for a distinct type, use the UDT command (see "UDT" on page 3-57).

Figure 10-12. Parameter Information Screen

```
File-AID for DB2 ------------ Parameter Information ---------- Row 1 to 3 of 3
COMMAND ===>
Stored Procedure: FADB2610.EURO                            Language: ASSEMBLER

Parameters are listed in Ordinal number Order
RT = Row type  SS = Scale  ST = Subtype  ENC = Encoding Scheme
                                                              E
PARAMETER          TYPE     TYPE               R               S N AS  TABLE
NAME               SCHEMA   NAME               T  LENGTH     SS T C LOC COLNO
------------------ -------- ------------------ -- ---------- -- - - --- -----
                   SYSIBM   DECIMAL            I           5  1      N
                   SYSIBM   DECIMAL            I           5  0      N
                   SYSIBM   DECIMAL            O           9  2      N
****************************** Bottom of data ********************************
```

PARAMETER NAME

Name of the parameter.

TYPE SCHEMA

Schema of the data type of the parameter.

TYPE NAME

Name of the data type of the parameter.

RT (Row Type)

I(n)	The parameter is an input parameter to the stored procedure.
O(ut)	The parameter is an output parameter that is returned by the stored procedure.
IO	The parameter is both an input and output parameter for the stored procedure.

LENGTH

Length of the data type.

SS (Scale)

Scale of the data type.

ST (Subtype)

If the data type is a distinct type, this is the subtype of the distinct type, which is based on the subtype of its source type:

B	Subtype is FOR BIT DATA.
S	Subtype is FOR SBCS DATA.
M	Subtype is FOR MIXED DATA.
blank	The source type is not a character type.

ENC (Encoding Scheme)

Encoding scheme of the parameter:

A	ASCII
E	EBCDIC
U	UNICODE
blank	The source type is not a character, graphic, or datetime type.

AS LOC

Locator.

Y	A locator to a value is to be passed.
N	The actual value is to be passed.

TABLE COLNO

For table parameters, the column number of the table. For all other, the value is blank.

User Defined Function

Figure 10-13 shows a sample information screen for user defined functions (UDFs).

Figure 10-13. UDF Information Screen

```
File-AID for DB2 ------ User Defined Function Information --------- SSID: D701
 COMMAND ===>                                                 SCROLL ===> CSR

Specify 'S' to select more information:
  _ Parameter Information
  _ Run Time Information
  _ Remarks/Comment On

 Function Name: FADB2610.WEATHER                            Language: C
 Specific Name: DSN8DUWFV
 External Name: DSN7DUWF

 UDF Detail ---------------------------------------------  Parameter---------
  Owner           : PFHJGDO      Routineid       : 771     Parm Count  : 1
  CreatedBy       : PFHJGDO      Routine type    :         Parm Style  : D
  Function Type   : T            Origin          : E       Return type :
  Cast Function   : N            Program Type    : S       Result Sets :
  Final Call      : Y            Deterministic   : Y       Result Cols : 7
  SQL Data        : N            External Action : N       LOBcolumns  : 0
  IBMREQD         : N            Null Call       : Y
  Security        : D            DBinfo          : N
  Scratchpad Size : 100          Special Register: I
  SourceSchema    :              COMMIT on return:
  Sourcespecific  :
  Createdts       : 2002-05-16-09.15.04.718925
  Alteredts       : 2002-05-16-09.15.04.718925

 Performance --------------------------------------------------------------
  Fenced          : Y            Cardinality     :
  Parallel        : D            IOS per Invoc   :
  Stay Resident   : N            Insts per Invoc  :
  ASUtime         : 5            Initial IOS     :
  WLM Environment : WLMENV       Initial Insts    :
  WLM Env Nested  : N

 JAVA ---------------------------------------------------------------------
  JAR ID      :
  JAR Schema  :
  CLASS       :
```

The following fields appear if you are displaying information for a UDF:

Parameter Information

When you select this option, the Parameter Information screen appears (Figure 10-14 on page 10-36).

Run Time Information

When you select this option, the Run Time Options window appears (Figure 8-61 on page 8-82).

Remarks/Comment On

When you select this option, the Comment On screen appears (Figure 8-62 on page 8-82).

UDF Detail

Owner

Authid of the owner of the UDF.

Routineid

Internal identifier of the UDF.

Parm Count

Identifies the number of parameters for the routine.

Createdby

Primary creator ID of person who created the UDF.

Routine Type

Identifies the type of routine. Either F(unction) or P (stored procedure).

F UDF or cast function.

P Stored procedure.

Parm Style

Default is DB2SQL. Identifies the parameter style of an external function or stored procedure.

D(B2SQL) According to the DB2SQL standard, parameters passed to the external function or the stored procedure .

G(eneral) All parameters are passed to the stored procedure according to the GENERAL standard convention.

N GENERAL CALL WITH NULLS. All parameters are passsed to the stored procedure according to the GENERAL WITH NULLS convention.

J(AVA) All parameters are passed to the function or prcoedure according to the conventions for JAVA and SQLJ specifications.

blank The column is blank if the ORIGIN is not 'E'.

Function Type

Type of function:

C Column

S Scalar

T Table

Origin

Origin of the routine; E for UDF.

Return Type

Describes the internal identifier of the result data type of the function. This field contains a -2 if the function is a table function.

Cast Function

Whether the UDF is a cast function or not.

Program Type

Either M(ain) or S(ub).

M Run the UDF as a main routine.

S Run the UDF as a subroutine.

Result Sets

Identifies the maximum number of ad hoc result sets that the stored procedure can return, if the Routine Type is P. The value is zereo, if there is no ad hoc result or the Routine Type is F.

Final Call

Final call option of an external function.

Deterministic

Default is N. If Y, then the UDF returns the same result from successive calls that have identical input arguments.

Result Cols

For table function, the number of columns in the result table. For other functions, value is 1.

SQL Data

Default is MODIFIES. Indicates whether the UDF can execute any SQL statements, and if so, what type.

N(o SQL) UDF cannot run any SQL statements.

M(odifies) UDF can run any SQL statement except statements not supported in any function.

R(eads) UDF cannot run SQL statements that modify data.

C(ontains) UDF cannot run any SQL statements that read or modify data.

External Action

External action option of an external function.

LOBcolumns

Identifies the number of LOB columns that are found in the parameter list for this UDF, if the ORIGIN is E. For any other function, the value is 0.

IBMREQD

Identifies whether or not the row came from the basic Machine Readable Material (MRM) tape. Y indicates that the row came from the MRM tape; N indicates it did not come from the MRM tape.

Null Call

The CALLED ON NOT NULL INPUT option of a UDF.

N UDF is not called if any parameter has a NULL value.

Y UDF is called if any parameter has a NULL value.

blank UDF is not an external function.

Security

Default is DB2. Specifies how the UDF works with external security products.

D(B2) No special external security environment needed. If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID associated with the stored procedure address space.

If NO WLM ENVIRONMENT is set, then SECURITY DB2 is the only valid value.

U(ser) An external security environment should be established for the UDF.

If the UDF accesses resources protected by an external security environment, then access is done using the authorization ID of the user who invoked the UDF.

C(definer) An external security environment should be established for the UDF. If the UDF accesses resources protected by an external security environment, access is done using the authorization ID of the owner of the UDF.

DBINFO

Default is N. If Y, then an additional argument is passed when the UDF is invoked. The argument may contain information such as the application run-time authorization ID or schema name.

Scratchpad Size

Length of the scratchpad.

Special Register

Describes how special register values are inherited. Appears only if you select DB2 version 7 or 8.

INHERIT The values of special registers are inherited based on rules listed in the table for characteristics of special registers (see DB2 SQL Reference Manual).

DEFAULT The values of special registers are initialized to the default values.

Sourceschema

Schema of the source function.

COMMIT on return

Identifies whether or not the transaction is committed immediately on a successful return (non-negative SQLCODE) from this stored procedure, when the Routine Type is P.

N	The unit of work continues.
Y	The unit of work is committed immediately.

If the Routine Type is F, the value is blank.

Sourcespecific

Specific name of the source function or source built-in function name.

Createdts

Timestamp when UDF was created (yy-mm-dd-hh.mm.ss.ttttt).

Alteredts

For a UDF, time when the latest ALTER FUNCTION statement was applied. If no ALTER FUNCTION statement was applied, this field has the same value as CREATEDTS.

Performance

Fenced

Indicates that this routine runs separately from the DB2 address space. The values that appear are defaults and are currently the only allowable OS/390 values.

Cardinality

Identifies the predicted cardinality of the routine. If the cardinality option was not specified and DB2 is to assume a finite value, the value is -1.

Parallel

Parallel option of an external function.

IOS per Invoc

Identifies the estimated number of I/Os that are required to execute the routine. If the estimated number is not known, the value is -1.

Stay Resident

Default is N. If Y, then the UDF load module remains resident in memory after the UDF ends.

Insts per Invoc

Identifies the estimated number of machine instructions that are required to execute the routine. If the estimated number is not know, the value is -1.

ASUTIME

Default is NO LIMIT. Specifies the total amount of processor time (in CPU service units) that a single run of the UDF can use.

Initial IOS

Identifies the estimated number of I/O's that are performed the first time or the last time the routine is invoked. If the estimated number is not known, the value is -1 .

WLM Environment

Identifies the MVS workload manager environment in which to run the UDF. To use a WLM environment, you must have authority for that environment.

Initial Insts

Identifies the estimated number of machine instructions performed the first time or the last time the routine is invoked. If the estimated number is not known, the value is -1 .

WLM Env Nested

For nested UDF calls, indicates whether the address space of the calling UDF is used to run the nested UDF:

N The nested UDF does not run in the specified WLM environment if the calling UDF is not running in the specified WLM environment. WLM ENVIRONMENT *name* was specified.

Y The nested UDF runs in the same WLM environment as the calling UDF. WLM ENVIRONMENT(*name*,*) was specified.

blank WLM_ENVIRONMENT is blank.

Java

JAR ID

JAR file name. This field is **blank** when the PARAMETER STYLE is not JAVA.

JAR Schema

Schema of the JAR file. This field is **blank** when the PARAMETER STYLE is not JAVA.

CLASS

The name of the class contained in the JAR file. This field is **blank** when the PARAMETER STYLE is not JAVA.

Parameter Information

To get the source type for a distinct type, use the UDT command (see “UDT” on page 3-57).

Figure 10-14. Parameter Information Screen

```
FILE-AID FOR DB2 -------- UDF PARAMETER INFORMATION ---------- ROW 1 TO 4 OF 4
COMMAND ===>
FUNCTION NAME: USERID1.UDFC                                  LANGUAGE: C
SPECIFIC NAME: UDFC_S

PARAMETERS ARE LISTED IN ORDINAL NUMBER ORDER
RT = ROW TYPE  SS = SCALE  ST = SUBTYPE  ENC = ENCODING SCHEME
                                                                  E
PARAMETER          TYPE     TYPE               R                S N AS  TABLE
NAME               SCHEMA   NAME               T  LENGTH     SS T C LOC COLNO
------------------ -------- ------------------ -- ---------- -- - - --- -----
                   SYSIBM   CHAR               R           8  0      N
                   SYSIBM   DECIMAL            C           8  0      N
                   SYSIBM   INTEGER            P           4  0      N
                   SYSIBM   DOUBLE             P           8  0      N
******************************* Bottom of data ********************************
```

PARAMETER NAME

Name of the parameter.

TYPE SCHEMA

Schema of the data type of the parameter.

TYPE NAME

Name of the data type of the parameter.

RT (Row Type)

P	Input
R	Result before casting
S	Input for sourced function

LENGTH

Length of the data type.

SS (Scale)

Scale of the data type.

ST (Subtype)

If the data type is a distinct type, this is the subtype of the distinct type, which is based on the subtype of its source type:

B	Subtype is FOR BIT DATA.
S	Subtype is FOR SBCS DATA.
M	Subtype is FOR MIXED DATA.
blank	The source type is not a character type.

ENC (Encoding Scheme)

Encoding scheme of the parameter:

A	ASCII
E	EBCDIC
U	UNICODE
blank	The source type is not a character, graphic, or datetime type.

AS LOC

Locator.

Y	A locator to a value is to be passed.
N	The actual value is to be passed.

TABLE COLNO

For table parameters, the column number of the table. For all other, the value is blank.

Chapter 11.
SQL Command Manager Utility

This chapter describes the SQL (structured query language) Command Manager utility, which allows you to execute SPUFI or SQL without leaving File-AID *for DB2*.

SPUFI

When you select option **5** from the Utility Selection Menu, File-AID *for DB2* accesses IBM SPUFI in DB2, and the SPUFI screen appears (Figure 11-1).

Figure 11-1. SPUFI screen

```
                                SPUFI                          SSID: DSN ===>

Enter the input dataset name:           (Can be sequential or partitioned)
  1  DATASET NAME  ... ===> SHOW.SQL(SHOWSQL)
  2  VOLUME SERIAL ... ===>             (Enter if not cataloged)
  3  DATASET PASSWORD  ===>             (Enter if password protected)

Enter the output dataset name:          (Must be a sequential dataset)
  4  DATASET NAME  ... ===> SQLOUT

Specify processing options:
  5  CHANGE DEFAULTS   ===> YES         (Y/N - Display SPUFI defaults panel?)
  6  EDIT INPUT ...... ===> YES         (Y/N - Enter SQL statements?)
  7  EXECUTE ......... ===> YES         (Y/N - Execute SQL statements?)
  8  AUTOCOMMIT ...... ===> YES         (Y/N - Commit after successful run?)
  9  BROWSE OUTPUT ... ===> YES         (Y/N - Browse output dataset?)

 For remote SQL processing:
 10  CONNECT LOCATION  ===>

PRESS: ENTER to process    END to exit    HELP for more information
```

1 DATASET NAME

Name of the dataset to contain the SQL statements; it must have been previously allocated

2 VOLUME SERIAL

Serial number of the volume where the input dataset resides (if it is not cataloged).

3 DATASET PASSWORD

Password, if any, for the input dataset.

4 DATASET NAME

Name of a dataset to contain the results of processing, which includes a copy of each SQL statement and any output it generated.

5 CHANGE DEFAULTS

Indicates whether you want the CURRENT SPUFI DEFAULTS screen shown before the first processing step. This screen lets you change certain characteristics of the SPUFI processing run and its output.

6 EDIT INPUT

Indicates whether you want to edit the input dataset before processing.

7 EXECUTE

Indicates whether the SQL statements in the input dataset should be processed.

8 AUTOCOMMIT

Indicates whether changes made to the database by the SQL statements processed should be automatically committed.

9 BROWSE OUTPUT

Indicates whether you want to browse the output dataset after the SQL statements in the input dataset have been processed.

10 CONNECT LOCATION

Indicates application server to which SPUFI will issue a SQL connect. All SQL statements you submit will then be executed by this server.

Chapter 12.
Privileges Utility

This chapter describes the utility that lets you easily display, grant, revoke, and modify DB2 privileges for various DB2 objects. When you modify or revoke privileges, File-AID *for DB2* shows only the privileges you can revoke. If you are a system administrator, you see all privileges.

Privileges - General

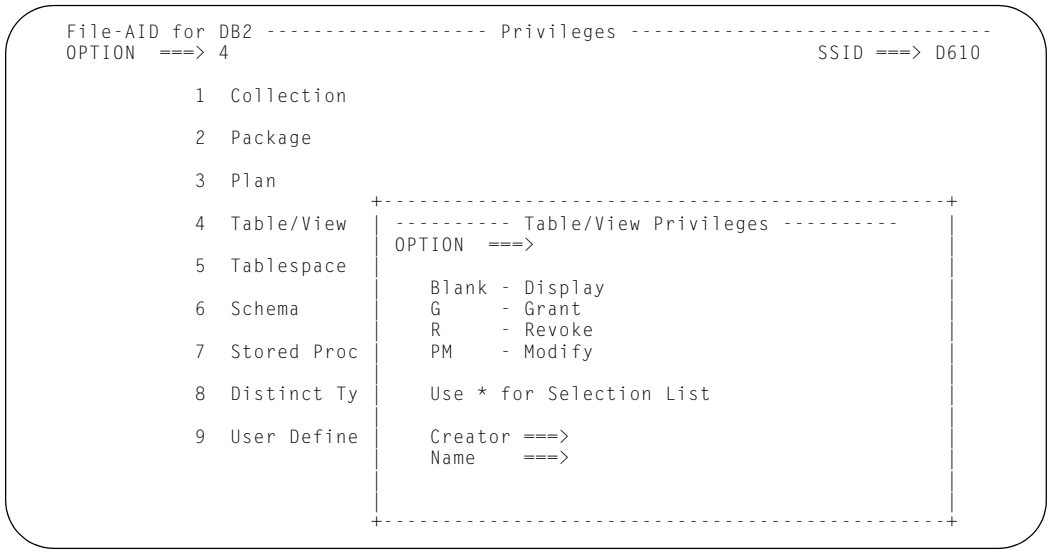
When you select option 6 from the Utility Selection Menu, the Privileges screen displays (Figure 12-1). Enter a number corresponding to the object type to which you will apply privileges and press Enter. You can only specify one object type at a time.

Figure 12-1. Privileges Screen

```
File-AID for DB2 ------------------- Privileges ---------------------------------
OPTION  ===>                                                  SSID ===> D610

            1  Collection

            2  Package

            3  Plan

            4  Table/View

            5  Tablespace

            6  Schema

            7  Stored Procedure

            8  Distinct Type

            9  User Defined Function
```

A pop-up window displays based on the object type you selected. All pop-up windows show the four available options, but other information varies depending on the object type.

Figure 12-2. Privileges pop-up window sample



OPTION

Blank Display the privileges assigned for an object.

G Grant privileges for an object.

R Revoke privileges for an object.

PM Modify privileges for an object.

Other fields on these pop-ups vary based on the object type selected:

Object Type	Fields
Collection	Collection Name
Package	Collection Name Package Name
Plan	Plan Name
Table/View	Creator Name
Tablespace	Database Tablespace
Schema	Schema Name
Stored Procedure	Schema Name Procedure Name
Distinct Type	Schema Name Distinct Type Name Process Implied Cast Function
User Defined Function	Schema Name Function Name

If you enter an asterisk (*****) in any of the object type specific fields, a selection list displays (Figure 12-3).

Figure 12-3. Privilege Object Selection Screen

```
File-AID for DB2 ----------------- Privilege Object Selection  Row 1 to 6 of 6
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: D610

SEL OBJECT                       TYPE
--- ---------------------------  ----------
_   USERID1.EMP                  TABLE
_   USERID1.EMPCOPY              TABLE
_   USERID1.EMPV1                VIEW
S   USERID1.EMPV2                VIEW
_   USERID1.EMP1                 TABLE
_   USERID1.EMP2                 TABLE
****************************** Bottom of data ********************************
```

SEL

Type an **S** (select) beside the object for which you want to display, grant, revoke, or modify privileges. The system displays that object when you press Enter.

OBJECT

List of objects for selection.

TYPE

Indicates the object type. If you access this screen from a Table/View Privileges pop-up, both tables and views show under TYPE.

For specific information on:

Object Type	See
Collection	“Collection Privileges” on page 12-3
Package	“Package Privileges” on page 12-6
Plan	“Plan Privileges” on page 12-8
Table/View	“Table/View Privileges” on page 12-10
Tablespace	“Tablespace Privileges” on page 12-16
Schema	“Schema Privileges” on page 12-19
Stored Procedure	“Stored Procedure Privileges” on page 12-21
Distinct Type	“Distinct Type Privileges” on page 12-23
User Defined Function	“User Defined Function (UDF) Privileges” on page 12-25

Collection Privileges

Display Privileges for Collection

The Privileges for Collection screen shows the privileges that are currently granted for the collection, but does not let you change the privileges.

Figure 12-4. Privileges for Collection Screen

Auth ID

User IDs of those granted privileges for the collection.

CREATE IN

G	The user is granted all privileges and can also grant them to others.
Y	Allows the user to name the collection in the BIND PACKAGE subcommand.
Blank	The user is not granted any privileges.

Note: CREATE IN and PACKADM ON are mutually exclusive. Choose only one.

PACKADM ON

For specific or all collections, allows the user all package privileges on all packages, plus the CREATE IN privilege on those collections.

Note: CREATE IN and PACKADM ON are mutually exclusive. Choose only one.

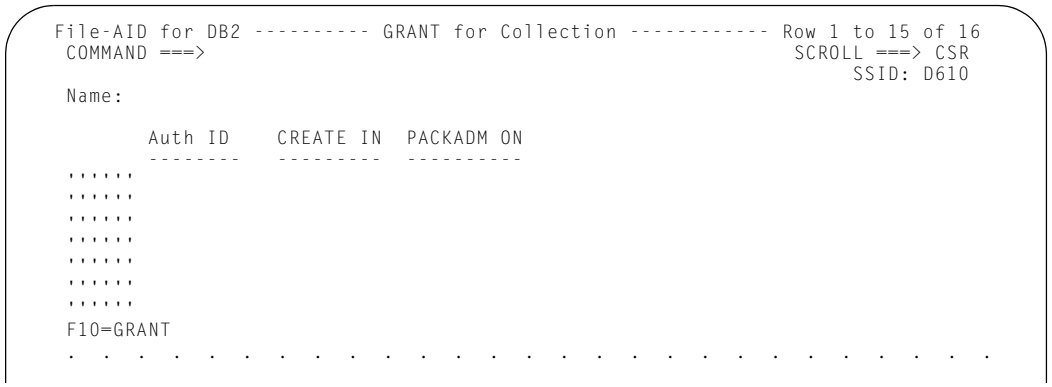
Grant Privileges for Collection

The Grant for Collection screen (Figure 12-5) lets you grant privileges for the specified Collection to others. When this screen first displays, it shows the valid privileges that are available for the Collection and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-5. Grant for Collection Screen



Auth ID

User IDs of person for whom you are granting collection privileges.

CREATE IN

G	The user is granted all privileges and can also grant them to others.
Y	Allows the user to name the collection in the BIND PACKAGE subcommand.
Blank	The user is not granted any privileges.

PACKADM ON

For specific or all collections, allows the user all package privileges on all packages, plus the CREATE IN privilege on those collections.

Revoke Privileges for Collection

The Revoke for Collection screen is identical to the Grant for Collection screen (Figure 12-5), except that it shows the privileges you have granted for a collection and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an R or a blank. To revoke all privileges, enter a D in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Collection screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Modify Privileges for Collection

The Modify for Collection screen shows the privileges you have granted for the collection and lets you grant new ones and/or revoke existing ones. The Modify for Collection screen is identical to the Grant for Collection screen (Figure 12-5) except for the screen title. To process a modification, type **MODIFY** on the command line and press enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Package Privileges

Display Privileges for Package

The Privileges for Package screen shows the privileges that are currently granted for the package, but does not let you change the privileges.

Figure 12-6. Privileges for Package Screen

```
File-AID for DB2 --------- PRIVILEGES for Package ----------- Row 1 to 15 of 16
 COMMAND ===>                                                 SCROLL ===> CSR
                                                                    SSID: D610
 Name:

        Auth ID    ALL  BIND  COPY  EXECUTE
        --------   ---  ----  ----  -------
 ''''''
 ''''''
```

Auth ID

User IDs of those granted privileges for the package.

ALL

Specifies whether a user is granted all privileges for the package:

G	The user is granted all privileges and can also grant them to others.
Y	The user is granted all privileges but does not have the authority to grant them to others.
Blank	The user is not granted all privileges.

BIND

Specifies the user's privileges for binding the package. Values are G, Y and blank, as defined for ALL.

COPY

Allows user to copy a package using the COPY option of BIND PACKAGE. Values are G, Y, and blank, as defined for ALL.

EXECUTE

Specifies the user's privileges for executing the package. Values are G, Y, and blank, as defined for ALL.

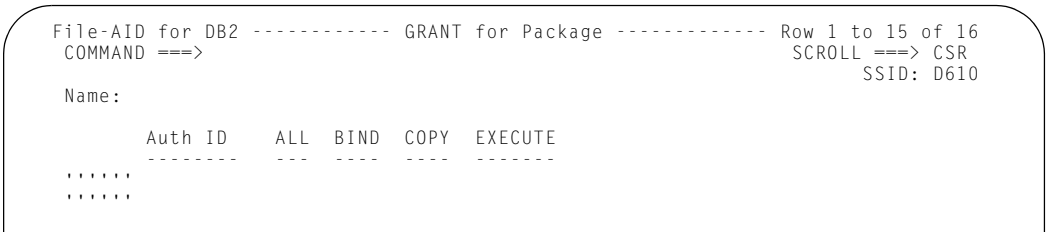
Grant Privileges for Package

The Grant for Package screen (Figure 12-7) lets you grant privileges for the specified package to others. When this screen first displays, it shows the valid privileges that are available for the package and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Figure 12-7. Grant for Package Screen



Auth ID

User IDs of those granted privileges for the package.

ALL

Specifies whether a user is granted all privileges for the package:

G	The user is granted all privileges and can also grant them to others.
Y	The user is granted all privileges but does not have the authority to grant them to others.
Blank	The user is not granted all privileges.

BIND

Specifies the user's privileges for binding the package. Values are G, Y, and blank, as defined for ALL.

COPY

Allows user to copy a package using the COPY option of BIND PACKAGE. Values are G, Y, and blank, as defined for ALL.

EXECUTE

Specifies the user's privileges for executing the package. Values are G, Y, and blank, as defined for ALL.

Revoke Privileges for Package

The Revoke for Package screen is identical to the Grant for Package screen, except that it shows the privileges you have granted for a package and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an **R** or a blank. To revoke all privileges, enter a D in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Package screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Modify Privileges for Package

The Modify for Package screen shows the privileges you have granted for the package and lets you grant new ones and/or revoke existing ones. The Modify for Package screen is identical to the Grant for Package screen (Figure 12-7) except for the screen title. To

process a modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Plan Privileges

Display Privileges for Plan

The Privileges for Plan screen shows the privileges that are currently granted for the plan, but does not let you change the privileges. Figure 12-8 shows a sample screen.

Figure 12-8. Privileges for Plan Screen

```
File-AID for DB2 -------- PRIVILEGES for plan PLAN94 --------  ROW 1 TO 3 OF 3
COMMAND ===>                                                 SCROLL ===> PAGE
                                                                   SSID: DSN

       Auth ID    EXECUTE  BIND  GRANTOR
       --------   -------  ----  --------
000001 USERID1    G        G     USERID1
000002 USERID2             G     USERID1
000003 USERID3             G     USERID1
000004 USERID4    Y              USERID1
****************************** BOTTOM OF DATA *******************************
```

Auth ID

User IDs of those granted privileges for the plan.

EXECUTE

Specifies the user’s privileges for executing the plan.

G The user may currently execute the plan and can also grant the privilege to others.

Y The user may currently execute the plan but does not have the authority to grant the privilege to others.

R The privilege is revoked.

BIND

Specifies the user’s privileges for binding the plan. The allowable values are G, Y, and R, as defined for EXECUTE.

GRANTOR

User ID of the person who gave the privilege to the Grantee.

Grant Privileges for Plan

The Grant for Plan screen lets you grant privileges for the specified plan to others. When this screen first displays, it shows the valid privileges that are available for the plan and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-9. Grant for Plan Screen

```
File-AID for DB2 -------- GRANT for plan PLAN94 ---------------  ROW 1 TO 1 OF 1
COMMAND ===>                                                   SCROLL ===> PAGE
                                                                     SSID: DSN
Enter SQL to view/save the SQL source

        Auth ID    EXECUTE    BIND
        -------    -------    ----
000001  NEWID      Y          Y
******************************** BOTTOM OF DATA ******************************









F22=GRANT
```

Auth ID

User IDs of persons granted plan privileges.

EXECUTE

Specify whether a user is granted the privilege of executing the plan.

G	The user may execute the plan and can also grant the privilege to others.
Y	The user may execute the plan but does not have the authority to grant the privilege to others.
Blank	The privilege is not granted.

BIND

Specify whether a user is granted the privilege of binding the plan. The allowable values are G, Y, and Blank, as defined for EXECUTE.

Revoke Privileges for Plan

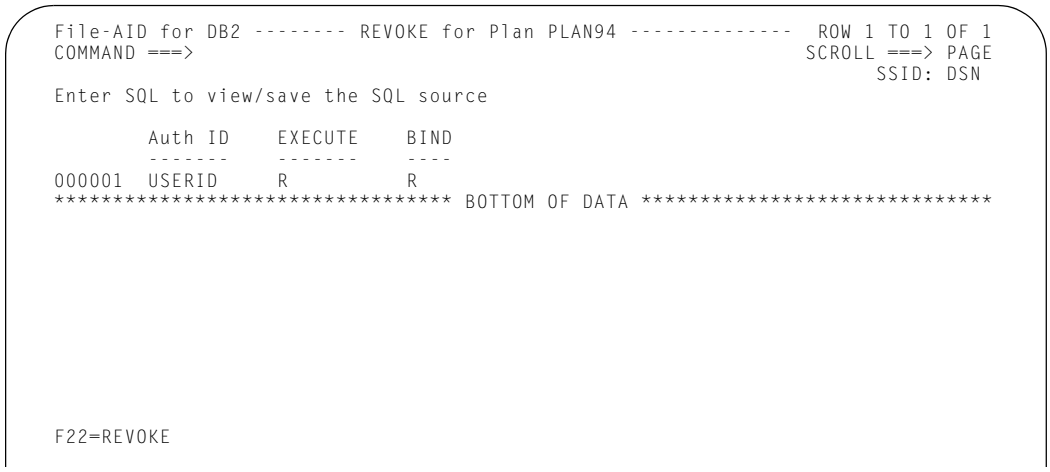
The Revoke for Plan screen shows the privileges you have granted for a plan and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an **R** or a blank. To revoke all privileges, enter a **D** in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Plan screen redisplays (Figure 12-10).

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-10. Revoke for Plan Screen



Auth ID

User IDs of those granted privileges for the plan.

EXECUTE

Specify whether to revoke a user's privilege of executing the plan.

G The user may currently execute the plan and can also grant the privilege to others.

Y The user may currently execute the plan but does not have the authority to grant the privilege to others.

R The privilege is revoked.

BIND

Specify whether to revoke a user's privilege of binding the plan. Values are G, Y, and R, as defined for EXECUTE.

Modify Privileges for Plan

The Modify for Plan screen shows the privileges you have granted for the plan and lets you grant new ones and/or revoke existing ones. The Modify for Plan screen is identical to the Revoke for Plan screen (Figure 12-10 on page 12-10) except for the screen title. To execute the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

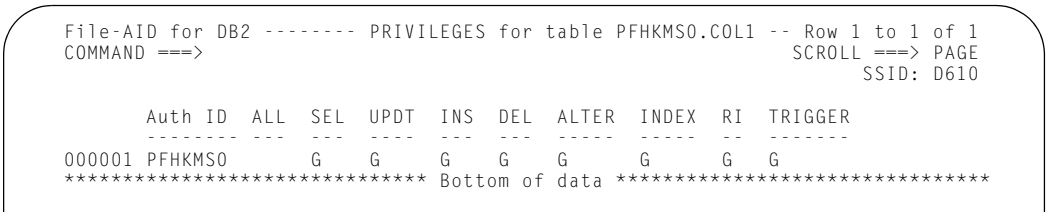
Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Table/View Privileges

Display Privileges for Table/View

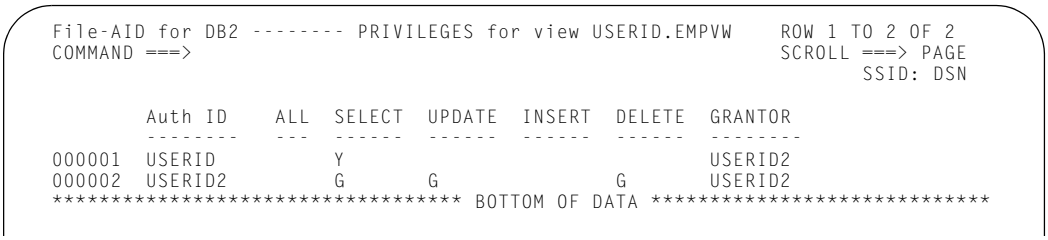
The Privileges for Table and Privileges for View screens show the privileges currently granted for the table or view, but does not allow you to change the privileges.

Figure 12-11. Privileges for Table Screen



```
File-AID for DB2 -------- PRIVILEGES for table PFHKMSO.COL1 -- Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: D610

       Auth ID  ALL  SEL  UPDT  INS  DEL  ALTER  INDEX  RI  TRIGGER
       -------- ---  ---  ----  ---  ---  -----  -----  --  -------
000001 PFHKMSO       G    G     G    G    G      G      G   G
******************************* Bottom of data ********************************
```

Figure 12-12. Privileges for View Screen



```
File-AID for DB2 -------- PRIVILEGES for view USERID.EMPVW    ROW 1 TO 2 OF 2
COMMAND ===>                                                 SCROLL ===> PAGE
                                                                    SSID: DSN

       Auth ID    ALL  SELECT  UPDATE  INSERT  DELETE  GRANTOR
       --------   ---  ------  ------  ------  ------  --------
000001 USERID          Y                               USERID2
000002 USERID2         G       G               G       USERID2
********************************** BOTTOM OF DATA ****************************
```

Auth ID

User IDs of those granted privileges for the table or view.

ALL

Specifies whether a user is granted all privileges for the table or view:

G	The user is granted all privileges and can also grant them to others.
Y	The user is granted all privileges but does not have the authority to grant them to others.
Blank	The user is not granted all privileges.

SELECT

Specifies whether the user is granted the Select privilege for the table or view. Values are G, Y and blank, as defined for ALL.

UPDATE

Specifies whether the user is granted the Update privilege for the table or view. Values are G, Y and blank, as defined for ALL.

INSERT

Specifies whether the user is granted the Insert privilege for the table or view. Values are G, Y and blank, as defined for ALL.

DELETE

Specifies whether the user is granted the Delete privilege for the table or view. Values are G, Y and blank, as defined for ALL.

ALTER

Specifies whether the user is granted the Alter privilege for the table or view. Values are G, Y and blank, as defined for ALL.

INDEX

Specifies whether the user is granted the Index privilege for the table or view. Values are G, Y and blank, as defined for ALL.

RI

Specifies whether the grantee can create or drop referential constraints in which the table or view is a parent.

TRIGGER

Specifies whether the grantee can create or drop triggers. Values are G, Y and blank, as defined for ALL.

GRANTOR

User ID of the person who gave the privilege to the Grantee.

Grant Privileges for Table/View

The Grant for Table and Grant for View screens let you grant privileges for the specified table or view to others. When this screen first displays, it shows the valid privileges that are available for the table or view and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

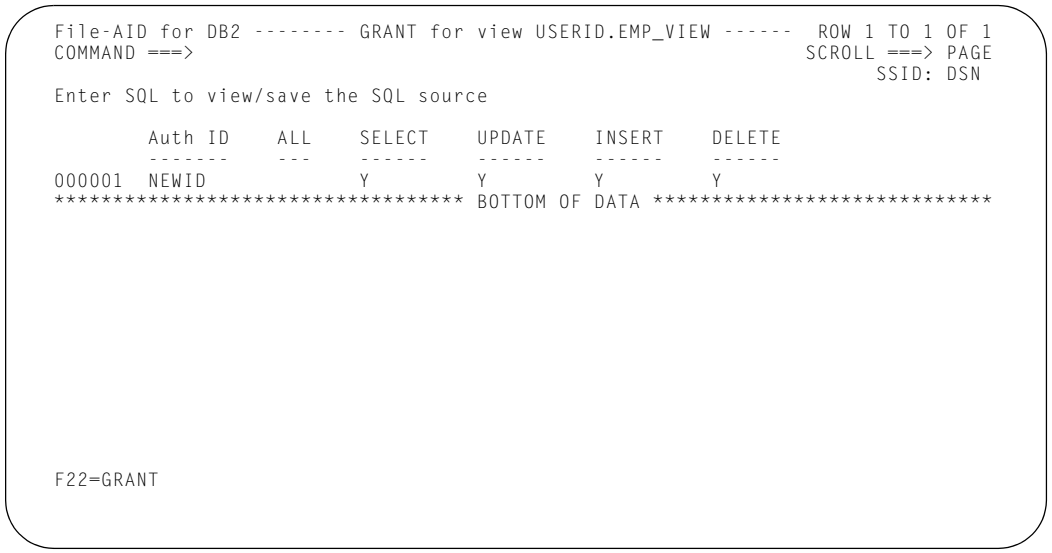
Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Note: Privileges for a given Auth ID are entered on a single line.

Figure 12-13. Grant for Table Screen

```
File-AID for DB2 -------- GRANT for table FDPROTO.EMP ------ Row 1 to 16 of 16
 COMMAND ===>                                                 SCROLL ===> CSR
                                                                    SSID: D610
 Use SQL to view/save the SQL source

        Auth ID    ALL  SEL  UPDT  INS  DEL  ALTER  INDEX  RI  TRIGGER
        --------   ---  ---  ----  ---  ---  -----  -----  --  -------
 '''''' BFHID1                                                  Y
 '''''' BFHID2                                                  G
 '''''' BFHID3     Y
 '''''' BFHID4          Y    G               Y                 G
 '''''' BFHID5     G
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 F10=GRANT
```

Figure 12-14. Grant for View Screen



Auth ID

User IDs of the people for whom you are granting table or view privileges.

ALL

Specify whether a user is granted privileges for all table or view actions:

G	The user is granted all privileges and can also grant them to others.
Y	The user is granted all privileges but does not have the authority to grant them to others.
Blank	The user is not granted all privileges.

Note: The above values can also be entered in the fields listed below.

SELECT

Specify whether the user is granted the Select privilege for the table or view. Values are G, Y and blank, as defined for ALL.

UPDATE

Specify whether the user is granted the Update privilege for the table or view. Values are G, Y and blank, as defined for ALL.

INSERT

Specify whether the user is granted the Insert privilege for the table or view. Values are G, Y and blank, as defined for ALL.

DELETE

Specify whether the user is granted the Delete privilege for the table or view. Values are G, Y and blank, as defined for ALL.

ALTER

Specify whether the user is granted the Alter privilege for the table or view. Values are G, Y and blank, as defined for ALL.

INDEX

Specify whether the user is granted the Index privilege for the table or view. Values are G, Y and blank, as defined for ALL.

RI

Specifies whether the grantee can create or drop referential constraints in which the table or view is a parent. Values are G, Y and blank, as defined for ALL.

TRIGGER

Specifies whether the grantee can create or drop triggers. Values are G, Y and blank, as defined for ALL.

Revoke Privileges for Table/View

The Revoke for Table and Revoke for View screens show the privileges you have granted for the table or view and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an **R** or a blank. To revoke all privileges, type an **R** in the ALL column, or enter a **D** (Delete) line command in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Table or Revoke for View screen redisplays.

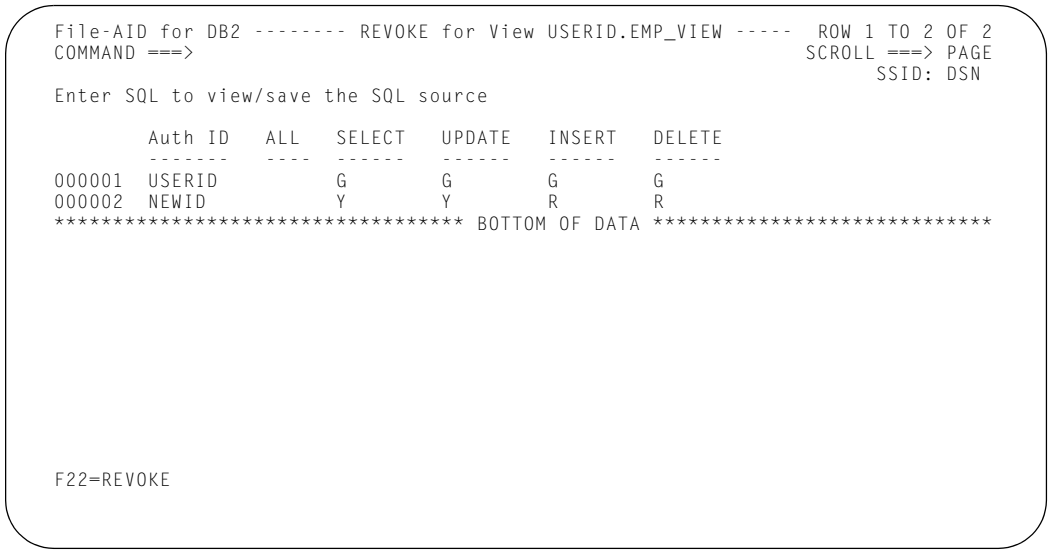
You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on these screens. See Chapter 4, “Line Commands” for more information.

Figure 12-15. Revoke for Table Screen

```
File-AID for DB2 -------- REVOKE for table FDPROTO.EMP ------- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> CSR
                                                                    SSID: D610
Use SQL to view/save the SQL source

         Auth ID   ALL   SEL  UPDT  INS  DEL  ALTER  INDEX  RI  TRIGGER
        --------   ---   ---  ----  ---  ---  -----  -----  --  --------
 D00001 BFHID1                                                   Y
 000002 BFHID3           Y     Y    Y    Y    Y      Y      Y   Y
 000003 BFHID4           Y                      Y
 000004 BFHID5     R     G     G    Y    G    G      G      G   Y
 000005 FDPROTO          G     G    G    G    G      G      G   G
******************************* Bottom of data ********************************









F10=REVOKE
```

Figure 12-16. Revoke for View Screen



Auth ID

User IDs of those who currently have privileges for the table or view.

ALL

Specify whether to revoke all privileges for a user ID.

G The user is granted all privileges and can also grant them to others.

Y The user is granted all privileges but does not have the authority to grant them to others.

R All privileges are to be revoked for that user.

SELECT

Specify whether to revoke the user's Select privilege for the table or view.

G The user is granted the privilege and can also grant the privilege to others.

Y The user is granted the privilege but does not have the authority to grant the privilege to others.

R The privilege is to be revoked for that user.

Blank The privilege is not currently granted.

Note: The above values can also be entered in the fields listed below.

UPDATE

Specify whether to revoke the user's Update privilege for the table or view. Values are G, Y, R and blank, as defined for SELECT.

INSERT

Specify whether to revoke the user's Insert privilege for the table or view. Values are G, Y, R and blank, as defined for SELECT.

DELETE

Specify whether to revoke the user's Delete privilege for the table or view. Values are G, Y, R and blank, as defined for SELECT.

ALTER

Specify whether to revoke the user's Alter privilege for the table or view.Values are G, Y, R and blank, as defined for SELECT.

INDEX

Specify whether to revoke the user's Index privilege for the table or view. Values are G, Y, R and blank, as defined for SELECT.

RI

Specifies whether the grantee can create or drop referential constraints in which the table or view is a parent. Values are G, Y, R and blank, as defined for SELECT.

TRIGGER

Specifies whether the grantee can create or drop triggers. Values are G, Y, R and blank, as defined for SELECT.

Modify Privileges for Table/View

The Modify for Table and Modify for View screens show the privileges you have granted for the table or view and lets you grant new ones and/or revoke existing ones. The Modify for Table and Modify for View screens are identical to the Revoke for Table and Revoke for View screens (Figure 12-15 on page 12-14 and Figure 12-16) except for the screen titles. To execute the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Tablespace Privileges

Display Privileges for Tablespace

The Privileges for Tablespace screen shows the privileges that are currently granted for the tablespace. It does not, however, let you change the privileges. The Privileges for Tablespace screen (Figure 12-17) displays.

Figure 12-17. Privileges for Tablespace Screen

```
File-AID for DB2 -------- PRIVILEGES for tablespace DBNAME.TS  ROW 1 TO 2 OF 2
COMMAND ===>                                                 SCROLL ===> PAGE
                                                                   SSID: DSN

       Auth ID    USE OF  GRANTOR
       --------   ------  --------
000001 USERID1    Y       USERID2
000002 USERID2    G       USERID3
******************************** BOTTOM OF DATA ********************************
```

Auth ID

User IDs of those granted privileges for the tablespace.

USE OF

Specifies whether a user is granted the privilege of using the tablespace.

G The user may use the tablespace and can also grant the privilege to others.

Y The user may use the tablespace but does not have the authority to grant the privilege to others.

GRANTOR

User ID of the person who gave the privilege to the Grantee.

Grant Privileges for Tablespace

The Grant for Tablespace screen (Figure 12-18 on page 12-17) lets you grant privileges for the specified tablespace to others. When this screen first displays, it shows the valid privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-18. Grant for Tablespace Screen

```
File-AID for DB2 -------- GRANT for tablespace DBNAME.TSNAME --  ROW 1 TO 1 OF 1
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                     SSID: DSN
Enter SQL to view/save the SQL source

        Auth ID    USE OF
        -------    ------
000001  NEWID      G
******************************** BOTTOM OF DATA ********************************









F22=GRANT
```

Auth ID

User IDs of person for whom you are granting tablespace privileges.

USE OF

Specify whether a user is granted the privilege of using the tablespace.

G The user may use the tablespace and can also grant the privilege to others.

Y The user may use the tablespace but does not have the authority to grant the privilege to others.

Revoke Privileges for Tablespace

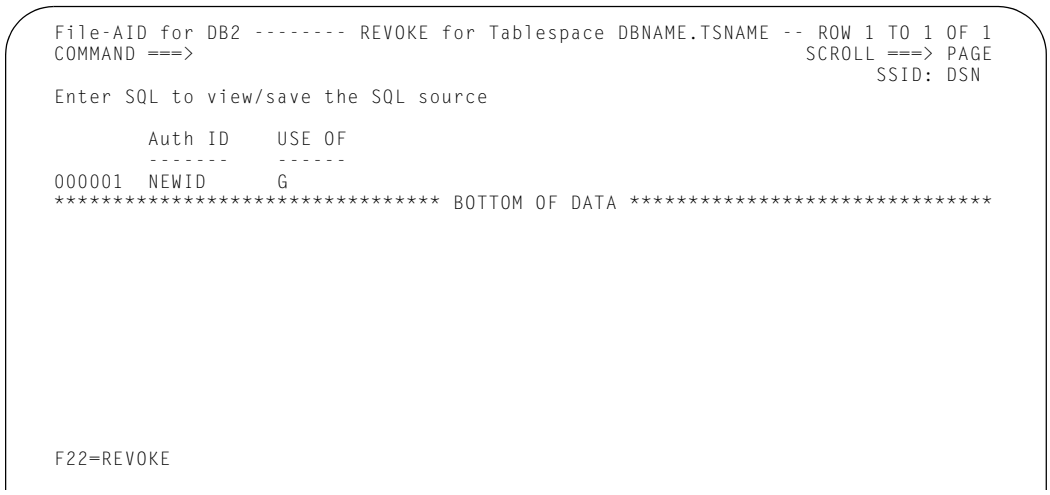
The Revoke for Tablespace screen shows the privileges that you have granted for the tablespace and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an R or a blank. To revoke all privileges, enter a **D** in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Tablespace screen (Figure 12-19 on page 12-18) redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-19. Revoke for Tablespace Screen



Auth ID

User IDs of those who currently have privileges for the tablespace.

USE OF

Specify whether to revoke a user’s privilege of using the tablespace.

G The user may currently use the tablespace and can also grant the privilege to others.

Y The user currently has use of the tablespace but does not have the authority to grant the privilege to others.

R The privilege is to be revoked.

Modify Privileges for Tablespace

The Modify for Tablespace screen shows the privileges you have granted for the tablespace and lets you grant new ones and/or revoke existing ones. The Modify for Tablespace screen is identical to the Revoke for Tablespace screen (Figure 12-19) except for the screen title. To execute the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Schema Privileges

Display Privileges for Schema

The Privileges for Schema screen shows the privileges that are currently granted for the schema, but does not let you change the privileges.

Figure 12-20. Privileges for Schema Screen

```
File-AID for DB2 ---------- PRIVILEGES for Schema ----------- Row 1 to 15 of 16
 COMMAND ===>                                                 SCROLL ===> CSR
                                                                    SSID: D610
 Name:

        Auth ID    ALTERIN  CREATEIN  DROPIN
        --------   -------  --------  ------
 ''''''
 ''''''
 ''''''
 ''''''
```

Auth ID

User IDs of those granted privileges for the schema.

ALTERIN

Allows user to alter user-defined functions (UDFs) or stored procedures. Also allows user to specify a comment for distinct types, UDFs, triggers and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

CREATEIN

Allows user to create distinct types, UDFs, triggers and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

DROPIN

Allows user to drop distinct types, UDFs, triggers, and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

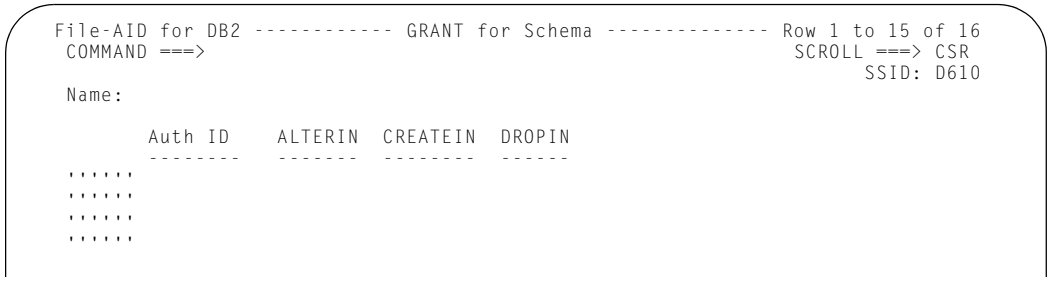
Grant Privileges for Schema

The Grant for Schema screen (Figure 12-21) lets you grant privileges for the specified schema to others. When this screen first displays, it shows the valid privileges that are available for the schema and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-21. Grant for Schema Screen



Auth ID

User IDs of those granted privileges for the schema.

ALTERIN

Allows user to alter user-defined functions (UDFs) or stored procedures. Also allows user to specify a comment for distinct types, UDFs, triggers and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

CREATEIN

Allows user to create distinct types, UDFs, triggers and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

DROPIN

Allows user to drop distinct types, UDFs, triggers, and stored procedures in specific schemas.

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

Revoke Privileges for Schema

The Revoke for Schema screen is identical to the Grant for Schema screen (Figure 12-21), except that it shows the privileges you have granted for a schema and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an R or a blank. To revoke all privileges, enter a **D** in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Schema screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Modify Privileges for Schema

The Modify for Schema screen shows the privileges you have granted for the schema and lets you grant new ones and/or revoke existing ones. The Modify for Schema screen is identical to the Grant for Schema screen (Figure 12-21) except for the screen title. To execute the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

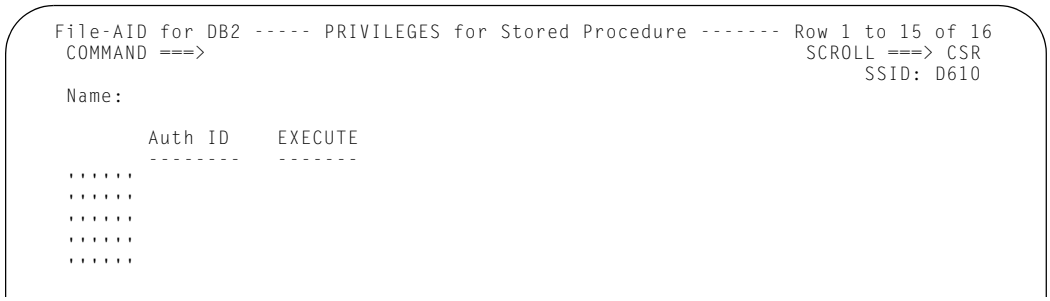
Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Stored Procedure Privileges

Display Privileges for Stored Procedures

The Privileges for Stored Procedure screen shows the privileges that are currently granted for the Stored Procedure, but does not let you change the privileges.

Figure 12-22. Privileges for Stored Procedure Screen



Auth ID

User IDs of those granted privileges for the Stored Procedure.

EXECUTE

Specifies the user's privileges for executing the Stored Procedure.

G The user may currently execute the Stored Procedure and can also grant the privilege to others.

Y The user may currently execute the Stored Procedure but does not have the authority to grant the privilege to others.

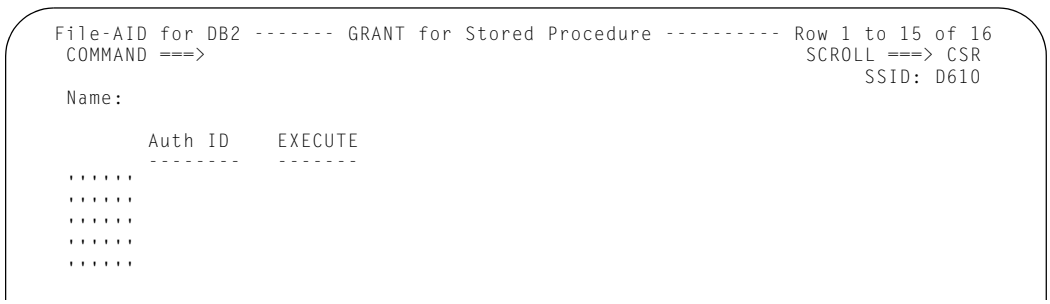
Grant Privileges for Stored Procedures

The Grant for Stored Procedure screen (Figure 12-23) lets you grant privileges for the specified stored procedures to others. When this screen first displays, it shows the valid privileges that are available for the stored procedures and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Figure 12-23. Grant for Stored Procedure Screen



Auth ID

User IDs of person for whom you are granting stored procedures privileges.

EXECUTE

Specifies the user's privileges for executing the stored procedure.

G The user may currently execute the stored procedure and can also grant the privilege to others.

Y The user may currently execute the stored procedure but does not have the authority to grant the privilege to others.

Revoke Privileges for Stored Procedures

The Revoke for Stored Procedure is identical to the Grant for Stored Procedure screen (Figure 12-23) except that it shows the privileges you have granted for stored procedures and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an R or a blank. To revoke all privileges, enter a **D** in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Stored Procedure screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Modify Privileges for Stored Procedures

The Modify for Stored Procedure screen shows the privileges you have granted for the stored procedure and lets you grant new ones and/or revoke existing ones. The Modify for Stored Procedure screen is identical to the Grant for Stored Procedure screen (Figure 12-23) except for the screen title. To execute the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Distinct Type Privileges

The popup displayed when Distinct Type is selected from the Privileges menu (Figure 12-1 on page 12-1) includes a Process Implied Cast Function field. If the option Process Implied Cast Function is set to Y, then after an action has been completed for privileges for a Distinct Type and PF3 is used to return, the same action will be taken for the implied cast function.

Display Privileges for Distinct Type

The Privileges for Distinct Type screen shows the privileges that are currently granted for the Distinct Type, but does not let you change the privileges.

Figure 12-24. Privileges for Distinct Type Screen

```
File-AID for DB2 ------ PRIVILEGES for Distinct Type --------- Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: D610
Name:  USERID1.US_DOLLAR

       Auth ID    USAGE ON  GRANTOR
       --------   --------  --------
000001 AAA        Y         USERID1
000002 PUBLIC     Y         USERID1
****************************** Bottom of data ********************************
```

Auth ID

User IDs of those granted privileges for the Distinct Type.

USAGE ON

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

GRANTOR

User ID of the person who gave the privilege to the Grantee.

Grant Privileges for Distinct Type

The Grant for Distinct Type screen (Figure 12-25 on page 12-24) lets you grant privileges for the specified distinct type to others. When this screen first displays, it shows the valid privileges that are available for the distinct type and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Figure 12-25. Grant for Distinct Type Screen

```
File-AID for DB2 --------- GRANT for Distinct Type --------- Row 1 to 16 of 16
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: D610
Name:  USERID1.US_DOLLAR

       Auth ID    USAGE ON
       --------   --------
'''''' USERID1    Y
''''''
''''''
''''''
''''''
''''''

****************************** Bottom of data ********************************
```

Auth ID

User IDs of person for whom you are granting distinct type privileges.

USAGE ON

Privilege status and action indicators are as follows:

Y The GRANTEE has (or will have, if granting) the privilege, but may not grant the privilege to others.

G The GRANTEE has (or will have, if granting) the privilege and may grant the privilege to others.

Revoke Privileges for Distinct Type

The Revoke for Distinct Type screen is identical to the Grant for Distinct Type screen (Figure 12-25) except that it shows the privileges you have granted for a distinct type and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an **R** or a blank. To revoke all privileges, enter a D in the line number area. To execute the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for Distinct Type screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

Modify Privileges for Distinct Type

The Modify for Distinct Type screen shows the privileges you have granted for the distinct type and lets you grant new ones and/or revoke existing ones. The Modify for Distinct Type screen is identical to the Grant for Distinct Type screen (Figure 12-25) except for the screen title. To process the modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

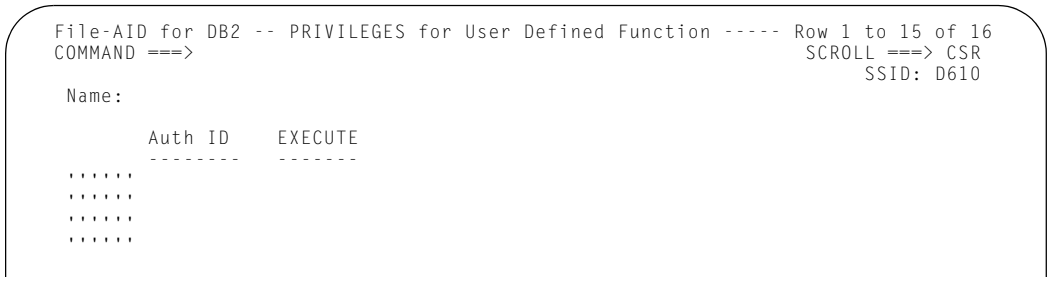
Line commands are valid on this screen. See Chapter 4, “Line Commands” for more information.

User Defined Function (UDF) Privileges

Display Privileges for UDF

The Privileges for UDF screen shows the privileges that are currently granted for the UDF, but does not let you change the privileges.

Figure 12-26. Privileges for UDF Screen



Auth ID

User IDs of those granted privileges for the UDF.

EXECUTE

Specifies the user's privileges for executing the UDF.

G The user may currently execute the UDF and can also grant the privilege to others.

Y The user may currently execute the UDF but does not have the authority to grant the privilege to others.

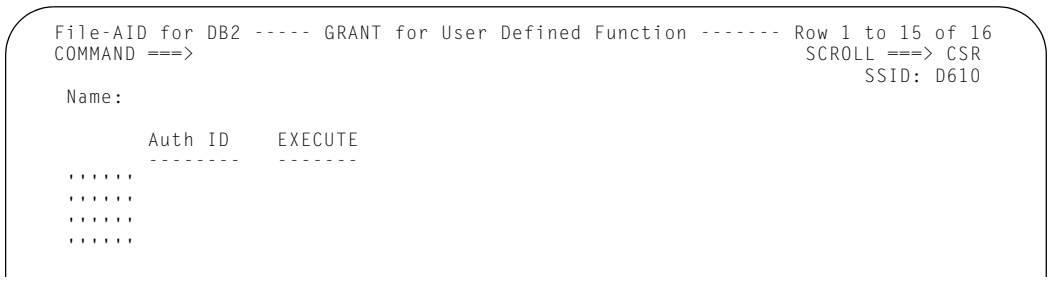
Grant Privileges for UDF

The Grant for User Defined Function screen (Figure 12-27 on page 12-26) lets you grant privileges for the specified UDF to others. When this screen first displays, it shows the valid privileges that are available for the UDF and allows you to insert new Auth IDs and privileges. To execute the grant, type **GRANT** on the command line and press Enter, or press the associated PF key.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to grant the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the GRANT primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Figure 12-27. Grant for User Defined Function Screen



Auth ID

User IDs of person for whom you are granting UDF privileges.

EXECUTE

Specifies the user's privileges for executing the Distinct Type.

G The user may currently execute the Distinct Type and can also grant the privilege to others.

Y The user may currently execute the Distinct Type but does not have the authority to grant the privilege to others.

Revoke Privileges for UDF

The Revoke for User Defined Function screen is identical to the Grant for User Defined Function screen (Figure 12-27) except that it shows the privileges you have granted for a UDF and lets you revoke the privileges for some or all users.

To revoke individual privileges, type over a displayed privilege with an **R** or a blank. To revoke all privileges, enter a **D** in the line number area. To process the revoke, type **REVOKE** on the command line and press Enter, or press the associated PF key. The Revoke for User Define Function screen redisplays.

You can execute the SQL primary command on this screen to view and/or save the SQL source generated to revoke the privileges. Enter **SQL** on the command line after you have entered your changes and before you execute the REVOKE primary command.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Modify Privileges for UDF

The Modify for User Defined Function screen shows the privileges you have granted for the UDF and lets you grant new ones and/or revoke existing ones. The Modify for User Defined Function screen is identical to the Grant for User Defined Function screen (Figure 12-27) except for the screen title. To process a modification, type **MODIFY** on the command line and press Enter, or press the associated PF key.

Line commands are valid on this screen. See Chapter 4, "Line Commands" for more information.

Chapter 13.
Impact Analysis Utility

This chapter describes the Column Impact Analysis utility, which greatly reduces the time required to analyze object structures.

The Column Impact Analysis utility details an object's dependencies and helps you visualize relationships. A batch report is produced that helps you determine the impact of dropping or modifying a column by identifying:

- Views
- Aliases
- Synonyms
- Indexes
- DBRMs/Packages
- Plans
- Referentially related columns
- Triggers

The report indicates if the column is used in any of the following clauses:

- GROUP BY
- HAVING
- ORDER BY

To create an accurate column impact report, you must have an up-to-date column data table.

A similar utility is available in DBA-Xpert. The File-AID *for DB2* utility generates JCL to produce the Column Impact Analysis report. You can print the report to a printer or to a dataset. A sample report appears in Figure 13-5 on page 13-5.

Column Impact Analysis

When you select option **8** from the Utility Selection Menu, the DBA-Xpert - Batch Column Impact Analysis screen appears.

Figure 13-1. Column Impact Analysis Screen

```
File-AID for DB2 ------DBA-Xpert Batch - Column Impact Analysis----------------
COMMAND ===>                                                    SSID ===> D610

Specify Option    ===> 1                 1 - Column Impact Analysis
                                         2 - Print Previous Report
Specify Report ID ===> P3860037          (Enter 1 to 8 characters)

Specify DB2 Table, View, Alias or Synonym:
   Creator        ===> BFHUSER1          (* or pattern for selection list)
   Name           ===>

Specify Report Options:  (Yes or No, include related objects in the Report)
   Views          ===> NO                DBRMs/Packages ===> NO
   Aliases        ===> YES               Plans          ===> NO
   Synonyms       ===> YES               RI Columns     ===> YES
   Indexes        ===> YES               Triggers       ===> YES
   Description    ===> UNIT TEST/AFT COMPILES-D610

 Indent character ===> |                 (Used for formatting report)
   Table prefix   ===>                   (Prefix for Control & Report Tables)
```

From the Column Impact Analysis screen, you must specify whether you want to create a new report or print a previously generated report. Each report has a unique report identification. You must also identify which related objects you want included in the report.

Specify Option

Enter the desired option.

1 Creates and prints a new impact analysis report.

2 Prints a previously created report.

Specify Report ID

Enter a name to identify the report. The identification name can contain up to eight characters. If you entered **1** in the option field, you must enter a new name.

If you entered **2**, you must enter a previously entered report identification. If you do not know the ID, browse the control table using File-AID *for DB2*. Review columns IMPID and DESCRIPTION to help determine the desired report ID. See "Impact Analysis Tables" on page 13-3 for information on the control table.

Specify DB2 Table, View, Alias or Synonym Area

Creator Creator of the table, view, alias, or synonym against which you are running the report.

Name Name of the table, view, alias, or synonym against which you are running the report.

Specify Report Options Area

Specify which objects you want included in the report.

Views Specify **Y** to include views in the report. Specify **N** to not include them.

Aliases Specify **Y** to include aliases in the report. Specify **N** to not include them.

Synonyms Specify **Y** to include synonyms in the report. Specify **N** to not include them.

Indexes	Specify **Y** to include indexes in the report. Specify **N** to not include them.
Description	Enter a description of the report. The description is saved in the control table
DBRMs/Packages	Specify **Y** to include DBRMs and packages in the report. Specify **N** to not include them.
Plans	Specify **Y** to include plans in the report. Specify **N** to not include them.
RI Columns (referentially related columns)	Specify **Y** to include referentially related columns in the report. Specify **N** to not include them.
Triggers	Specify **Y** to include triggers. Specify **N** to not include them.
Indent character	When formatting the report, indentation is used to display information. Enter the desired indent character. A vertical bar (|) is recommended.
Table prefix	Table prefix is used to create unique control and report tables for each report. Prefixes can contain up to eight characters. See "Impact Analysis Tables" on page 13-3 for further information on this field.

Impact Analysis Tables

When you create an impact report, File-AID *for DB2* creates two tables:

Table Name	Explanation
*Creator.table_prefix***IMPACT**	Report Table. Contains the information printed in the report.
*Creator.table_prefix***IMPACTID**	Control table. Contains the input to the report generation program.

At the time of installation, the installer specified if Impact Analysis would create system or user tables. The Table Prefix field does not appear on your screen if your site is using system tables. In this case, all users write to the same report tables.

If the Table Prefix field appears on your screen, you have the option of entering a value. Enter a unique value each time you run the report to have new tables created for each report. The Create Impact Analysis Tables screen (Figure 13-2) appears when new tables need to be created. If you always leave this field blank or enter the same prefix value, File-AID *for DB2* displays the Create Impact Analysis Tables screen only the first time the tables are created. Then, each successive time you run a report, File-AID *for DB2* appends the report information to the existing tables.

Create Impact Analysis Tables

The Create Impact Analysis Tables screen appears when File-AID *for DB2* needs to create new control and report tables.

Figure 13-2. Create Impact Analysis Table Screen

```
File-AID for DB2 ----------- Create Impact Analysis Tables -------- SSID: DSN
COMMAND ===>

There is no Impact Control Table and/or Report table for your
current SQLID>

Specify the database and tablespace in which these
are to be created:
   Database ===>
 Tablespace ===>               (blank for implicit Table Space)

Instructions:
   Press ENTER to continue.
     Use END to cancel.
```

Database

Enter the database name in which you want the tables created.

Tablespace

Enter the tablespace name in which you want the tables created. If you leave this field blank, File-AID *for DB2* creates an implicit tablespace.

Impact Selection Template

The Impact Selection Template screen enables you to select the columns you want included in the report. For each column you select, the report displays the related objects you selected on the Column Impact Analysis screen (Figure 13-1 on page 13-2).

Enter an **S** in the SEL field next to the desired column name. When you have selected all the columns, type EXECUTE on the command line and press Enter, or press the associated PF key.

Figure 13-3. Impact Analysis Selection Template

```
File-AID for DB2 -------- IMPACT Selection Template ------- ROW 1 TO 14 OF 14
COMMAND ===>                                                SCROLL ===> PAGE
                                                                  SSID: DSN
Table Name: DB2DFLT.EMP

SEL     COLUMN NAME
---     ------------------
_        EMPNO
_        FIRSTNME
_        MIDINIT
_        LASTNAME
S        WORKDEPT
_        PHONENO
_        HIREDATE
_        JOB
_        EDLEVEL
_        SEX
_        BIRTHDATE
_        SALARY
_        BONUS
_        COMM
****************************** BOTTOM OF DATA *************************
F18=EXECUTE
```

SEL

Enter an **S** to select a column.

Column Name

List the columns in the table.

Batch Output Specification

The Batch Output Specification screen (Figure 13-4 on page 13-5) appears when you use the EXECUTE command on the Impact Selection Template screen. It enables you to select whether you want the report to be written to a printer or to your output dataset. (See "Option 7, Print Options" on page 5-10 for information on setting the default values for this screen.)

Press Enter on this screen to view the generated JCL in ISPF/PDF Edit. The JCL is only temporary. You must use the CREATE or COPY primary command to save it. Execute the **SUB** command to submit the JCL and run the report. (See "Option 3, Batch Options" on page 5-6 for information on setting your default jobcard.)

Figure 13-4. Impact Analysis Output Specification Screen

```
File-AID for DB2 ------ Impact Analysis Output Specification --------- --------
COMMAND ===>

Select Printer Destination or Dataset then press ENTER

  _  TO PRINTER

  Printer Destination  ===> PRINTER1
  Sysout Class         ===> G



  _  TO DATASET

  Report Dataset Name  ===> ABC.OUTPUT
  Volume Serial        ===>
```

Column Impact Analysis Report Example

The following is a sample report. In the report, objects that are indented are dependent on the object in the preceding row. See Table 13-1 on page 13-6 for a line by line explanation of the report. Table 13-2 on page 13-6 provides an explanation of the codes used in the report.

Figure 13-5. Column Impact Analysis Report

```
                                        ROOT
CL: DB2DFLT.EMP.WORKDEPT                Type: CHAR     Length: 3 Scale: 0
 | Parent TB: DB2DFLT.DEPT
 | | Parent CL: DB2DFLT.DEPT.DEPTNO     Relname: RED      Col seq: 1
 | | | Dep TB: DB2DFLT.DEPT
 | | | | Dep CL: DB2DFLT.DEPT.ADMRDEPT  Relname: RDD      Col seq: 1
 | | | | | IX: DB2DFLT.XDEPT3           Col Seq: 1
 | | | Dep TB: DB2DFLT.PROJ
 | | | | Dep CL: DB2DFLT.PROJ.DEPTNO    Relname: RPD      Col seq: 1
 | | | IX: DB2DFLT.XDEPT1               Col Seq: 1
 | IX: DB2DFLT.XEMP2                    Col Seq: 1
```

Table 13-1. Impact Analysis Report Explanation

Name and Explanation	Additional Information
CL: DB2DFLT.EMP.WORKDEPT Starting column	Type: CHAR Length: 3 Scale: 0
| Parent TB: DB2DFLT.DEPT Parent table to column DB2DFLT.EMP.WORKDEPT	
| | Parent CL: DB2DFLT.DEPT.DEPTNO This primary key column references the foreign key column DB2DFLT.EMP.WORKDEPT	Relname: RED Col seq: 1
| | | Dep TB: DB2DFLT.DEPT Dependent table	
| | | | Dep CL: DB2DFLT.DEPT.ADMRDEPT This foreign key column references the primary key column DB2DFLT.DEPT.DEPTNO	Relname: RDD Col seq: 1
| | | | | IX: DB2DFLT.XDEPT3 Column DB2DFLT.DEPT.ADMRDEPT is in the first position of this index	Col Seq: 1
| | | Dep TB: DB2DFLT.PROJ Dependent table	
| | | | Dep CL: DB2DFLT.PROJ.DEPTNO This foreign key column references the primary key column DB2DFLT.DEPT.DEPTNO	Relname: RPD Col seq: 1
| | | IX: DB2DFLT.XDEPT1 Column DB2DFLT.DEPT.DEPTNO is in the first position of this index.	Col Seq: 1
| IX: DB2DFLT.XEMP2 Column DB2DFLT.EMP.WORKDEPT is in the first position of this index.	Col Seq: 1

Table 13-2. Impact Analysis Codes Explanation

Code	Explanation
AL COL	Alias column
CL:	Starting column
Dep CL	Dependent column
Dep TB	Dependent table
DBRM	DBRM
IX	Index
Parent CL	Parent column
Parent TB	Parent table
PK	Package
PL	Plan
Relname	Relationship name

Table 13-2. Impact Analysis Codes Explanation

Code	**Explanation**
Ref VW	Referenced view
Ref VW Col	Referenced view column
ROOT	Beginning of the analysis for a selected column.
SY COL	Synonym column
TR	Trigger

Chapter 14.
Extract and Load Utility

This chapter describes the Extract and Load Tables utility, which lets you extract data from a DB2 table to a dataset, or load data from a dataset to a DB2 table. A full table or a subset of rows can be extracted. Optionally, you can also extract data in delimited file format to load into other RDBMS or spreadsheets that accept delimited file format. The utility can be run both online and in batch. You can also extract a DB2 table to tape and load it from tape. This option is only available in batch mode. In addition, you can now combine extract and load into a single operation, enabling you to copy across subsystems or locations.

When you perform an extract operation in standard file format, two datasets are created: an unload dataset and a control dataset. The unload dataset contains all the row data extracted from the specified object. The control dataset contains information about the structure of the data. These two datasets are then used as the input to the load operation.

A third dataset, a record layout, can be created during the extract operation. This dataset can then be used in File-AID/MVS to view the extract dataset in formatted mode.

Note: You should not mix unload and control datasets from different releases of File-AID *for DB2*.

When you perform an extract operation in delimited file format (see "Specify Extract Dataset Format" on page 14-11), only the unload dataset is created. Unload datasets in delimited file format cannot be loaded with the File-AID *for DB2* load option.

When you perform a load operation, you may choose to perform either an SQL load or a DB2 load. If you select SQL load, data is loaded into the target table using a number of insert statements. If you select DB2 load, data is loaded into the target table using the IBM DB2 Batch load utility.

Unload and control card datasets are completely interchangeable between batch and online modes. Datasets created during an online extract operation can be used as source datasets in a batch load process. Datasets created during a batch extract operation can be used as the source datasets in an online load process.

UDT Columns, Triggers and LOB Columns

File-AID *for DB2*'s Extract and Load functions handle UDT, Trigger and LOB as follows:

Tables with UDT columns:

You can extract data from and load data to tables with distinct data type (UDT) columns as long as their schema is defined in the current path.

Triggers:

When opening a table for extract, File-AID *for DB2* informs you that trigger actions are defined against this table.

Tables with LOB columns:

When extracting from tables with LOB data types, LOB and ROWID columns will not be extracted. If rows are deleted, warnings are issued that they may be disconnected from the LOBs but that the LOB columns are still in the Auxiliary table. When a row is loaded into a table with LOB columns, the LOB column's value and ROWID will be

set to DEFAULT as long as the LOB column allows nulls. For NOT NULL LOB columns loading of rows is not allowed. The LOB types are binary large object (BLOB), character large object CLOB), and double-byte character large object (DBCLOB).

Extract/Load Menu

When you select option **7** from the Utility Selection Menu, the Extract/Load Menu is displayed (Figure 14-1 on page 14-2).

Figure 14-1. Extract/Load Menu

```
File-AID for DB2 -------------- Extract/Load Menu -----------------------------
OPTION  ===>                                                  SSID ===> D610

   1 - Extract from a Table to a dataset

   2 - Load from a dataset to a Table

   3 - Extract and Load
```

OPTION

Enter the number representing the activity you want to pursue:

1 Extract from a table to a dataset. The Extract Options screen (Figure 14-2) displays.

2 Load from a dataset to a table. The Extract Selection Template (Figure 14-5 on page 14-7) displays.

3 Extract from a table and load into another table. The Extract/Load Options screen (Figure 14-27 on page 14-31) displays.

Option 1 - Extract from a Table to a Dataset

Online Processing

The Extract Options screen displays when you select option **1** from the Extract/Load Menu. It presents the extract options available to you.

Figure 14-2. Extract Options Screen

```
File-AID for DB2 ------------ Extract Options --------------------------------

 COMMAND ===>                                                  SCROLL ===> PAGE
                                                                 SSID ===> D701
 Specify DB2 Table (* or pattern for list)
    Location ID ===>                          Optional:
    Creator     ===> USERID                     Database    ===>
    Table Name  ===> *                          Tablespace  ===>

 Specify Extract Template Information:
    Use Existing Criteria ===> NO       (Yes or No)
    Edit Criteria Dataset ===> EXTRCRIT
    Member                ===>          (Blank or Pattern for list)

 Specify Extract Options:
    Processing Mode          ===> 1             (1 = Online, 2 = Batch)
    Formatted Print Report   ===> NO            (Yes or No)
    Maximum Rows to Select   ===> 2000          (* for all rows)
    Extract Null Indicators  ===> YES           (Yes or No)
    Use Uncommitted Read     ===> NO            (Yes or No)
```

Specify DB2 Table Area

Entries are required.

Location ID

Location of the DB2 table.

Creator

ID of the person who created the DB2 table. Enter an asterisk (*****) for a selection list.

Table Name

Name of the DB2 table, synonym, view or alias from which data is being extracted. Enter an asterisk (*****) for a selection list.

Optional Area

Database

Name of the database. Use to limit selection list.

Tablespace

Name of the tablespace. Use to limit selection list.

Specify Extract Template Information Area

Use Existing Criteria

Enter whether you want to use a previously saved Edit/Extract Selection Template dataset. Edit/Extract Selection Criteria consists of conditions to limit and order column selection and/or row selection based on column values. This field is prefilled with the last specified value (YES or NO).

NO Do not use a previously saved Edit/Extract Selection Criteria dataset. You can still save the selection template for this session (see "Save Edit Criteria Information Area" on page 14-7).

YES Apply the criteria stored in the dataset specified in the **Edit Criteria Dataset** field. Then you can verify and/or modify the selection criteria in the "Extract Selection Template" on page 14-7 (See "Save Edit Criteria Information Area" on page 14-7).

Edit Criteria Dataset

If you specified **Yes** in the **Use Existing Criteria** field, specify the name of the dataset that contains the edit template criteria you want to use. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Note: The criteria dataset must contain a valid SELECT statement. File-AID *for DB2* only uses the first SELECT statement of the dataset. If a valid SELECT is missing, or the table name of the SELECT statement does not match the entered name in the above Table Name field, File-AID *for DB2* gives an error (see "Selection Template Not Prefilled" on page 14-8). When File-AID *for DB2* processes the SELECT statement to prefill the selection template, it ignores extraneous clauses such as WITH, OPTIMIZE, FOR FETCH, and CURRENT OF.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to use. If you leave this field blank or specify an asterisk (*****), File-AID *for DB2* displays the Member List for the dataset. The asterisk (*****) is the only valid pattern character.

Specify Extract Options Area

Processing Mode

Enter the desired processing mode.

1 Enter 1 for online processing.

2 Enter 2 for batch processing. See "Batch Processing" on page 14-18 for further information.

Formatted Print Report

YES Enter Yes to receive a printed report of the extract. The printer and output dataset characteristics are specified on the Print Options screen in User Parms, Option 0.7 from the Primary Option Menu. See "Option 7, Print Options" on page 5-10 for more information.

NO Enter No if you do not want to receive a report.

Maximum Rows to Select

Enter a numeric value for the maximum number of rows to be extracted. Enter an asterisk (*****) to select all rows.

Extract Null Indicators

YES Default. Enter Yes to include null indicators in the extract dataset. The null indicators appear as two bytes of low values for each null column in the table. They appear at the beginning of each record. If null indicators are included in the extract dataset, the record length of the output dataset is larger than if null indicators are not included.

NO Enter No if you do not want the extract dataset to include null indicators.

Use Uncommitted Read

Allows an application to bypass acquiring the locks that ensure no other user is updating the data being read. Bypassing the locks causes the application to run faster.

CAUTION:
If you choose the Use Uncommitted Read option, and the application fails and backs out the updates, then the rows read by the application using uncommitted read could be invalid.

NO Default. Enter NO if you do not want the application to bypass acquiring the locks which ensure that no other user is updating the data being read.

YES Enter YES if you want the application to bypass acquiring the locks which ensure that no other user is updating the data being read. If you enter YES, the rows read by the application using uncommitted read could be invalid.

Selection Criteria Member List

The Selection Criteria Member List displays if you left the Selection Criteria Member field blank (or entered a *****) on the Extract Options screen. It provides a list of all members of the specified Selection Criteria dataset.

Figure 14-3. Selection Criteria - EXTRCRIT

```
File-AID for DB2 ------- Selection Criteria - EXTRCRIT ----  ROW 1 TO 13 OF 13
COMMAND ===>                                                  SCROLL ===> PAGE

Sel  Member Name
---  -----------
 _   CRIT1
 _   CRIT2
 _   CRITEST1
 _   CRITEST2
 _   CRITEST3
 _   CRITEST4
 _   CRITEST5
 _   CRITEST6
 _   MDS
 _   ORDER
 _   SAVEJCL
 _   SHOWSQL
 _   SHOWSQL2
****************************** BOTTOM OF DATA ********************************
```

Sel

Enter the **S** line command to select one of the listed members as your selection criteria.

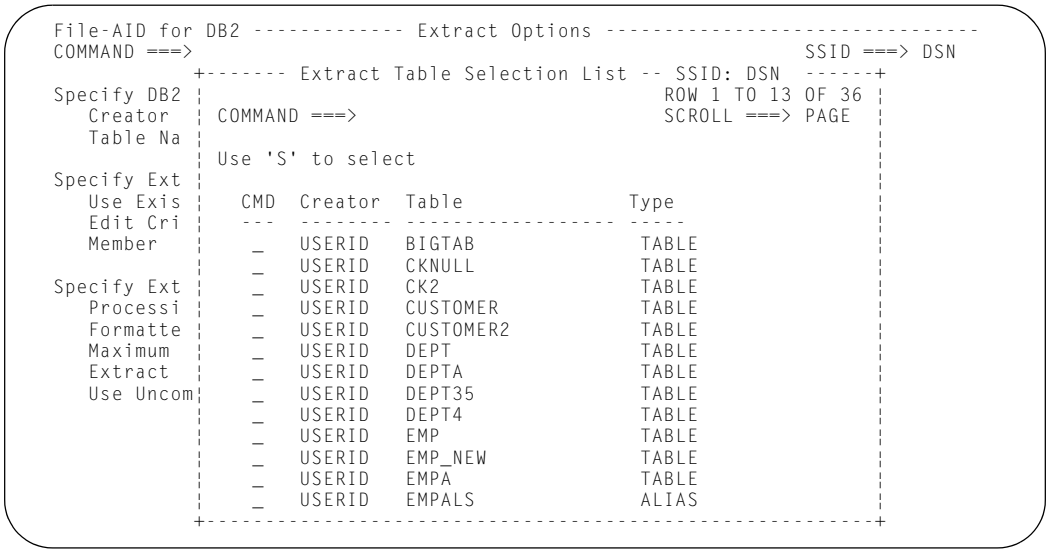
Member Name

Lists all member names of the selection criteria dataset.

Extract Table Selection

The Extract Table Selection window displays when an asterisk was entered in either the Creator field or the Table Name field of the Extract Options screen. It presents a list of tables, views and aliases (Figure 14-4).

Figure 14-4. Extract Table Selection Window



Cmd

Enter **S** to select a table to be extracted. The Extract Selection Template screen displays when you press Enter.

Creator

Displays the creators of the tables, views, or aliases.

Table

Displays the names of the tables, views, or aliases.

Type

Indicates either TABLE, VIEW, or ALIAS.

EXTRACT Selection

The EXTRACT Selection Template screen (Figure 14-5 on page 14-7) displays after you complete the Extract Options screen (Figure 14-2 on page 14-3). The screen is used to identify the columns to be extracted, indicate the order of extraction, and to provide selection criteria for extraction of specific rows of data.

If you are using an existing Extract Criteria Dataset the SEL column is already prefilled with the selections from the Extract Criteria Dataset. You can then either accept or modify them as desired.

Note: You achieve “Full Table Extract” by NOT selecting any columns in the SEL column when entering the EXECUTE command.

If triggers exist on the extract table, the message ‘Triggers not activated’ displays in the upper right-hand corner of the EXTRACT Selection Template screen.

After the appropriate selection criteria has been provided, type EXECUTE on the command line and press Enter, or press the associated PF key to process the request. The Extract Table screen displays. The SQL command may also be entered on this screen to review, save and/or modify the SQL generated by File-AID *for DB2*.

Figure 14-5. Extract Selection Template

```
File-AID for DB2 -------- EXTRACT Selection Template -------- Row 1 to 8 of 18
COMMAND ===>                                                 SCROLL ===> PAGE
                                                                   SSID: D610
Use EXEC to continue
Use SQL to view/edit the SQL source for FADB2610.CUSTOMER_TABLE

Save Edit Criteria Information:
   Save Template Criteria On Execute  ===> NO       (Yes or No)
   Criteria Dataset Name  ===> SHOW.SQL
   Member                 ===>

   Max Rows to Select     ===> 2000

SEL  COLUMN NAME        TYPE(LEN)        ORDER (A/D) WHERE             More: >
---  ------------------ ---------------- --    -   ----+---10----+---20----+
___  CUSTOMER_NUMBER    CHAR(6)
___  COMPANY_NAME       CHAR(30)
___  ADDRESS            CHAR(30)
___  CITY               CHAR(25)
___  STATE              CHAR(2)
___  ZIP_CODE           CHAR(9)
___  COUNTRY            CHAR(20)
___  AREA_CODE          CHAR(3)
 F6=EXECUTE
```

Save Edit Criteria Information Area

Save Template Criteria On Execute

Enter **YES** to save the current Edit/Extract Selection Criteria to a dataset.

Note: The selection criteria is saved when you enter the EXECUTE command on this screen. It is, however, NOT saved if you enter the SQL command and execute the SQL directly. The executed SQL is saved in the dataset as specified with "Allocate SHOW SQL Dataset" on page 5-5.

Enter **NO** when you do not want to save the current Edit/Extract Selection Criteria to a dataset.

Edit Criteria Dataset

Enter the name of the dataset to which you want the Extract Selection Criteria saved. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset to which you want the selection criteria saved.

Max Rows to Select

Enter the maximum number of rows to be extracted.

SEL

Enter **S** next to the columns from which you want data extracted.

COLUMN NAME

Displays the names of the columns in the table, view, or alias.

TYPE(LEN)

Displays the type and length of data in each column.

Distinct types are identified by their names.

Note: Columns with type LOB or distinct based on LOB do not appear on the template selection screen as they cannot be extracted.

ORDER

Specify an integer to define how the extracted rows should be sorted in the unload dataset, based on the data content of the columns. Entering **1** indicates the primary sort column.

A/D

Indicate whether ordered columns are to be sorted in ascending (A) or descending (D) order. If neither is specified, A is the default.

WHERE

Enter WHERE clauses to limit row selection. This field can be scrolled left and right. The following delimiters can be used in the clauses:

>	OR	< >
<	> =	BETWEEN
=	< =	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

Note: Special considerations for distinct (UDT) data type columns: File-AID *for DB2* generates the appropriate CAST function for simple WHERE clauses that contain just one of the following operators:

>	> =	< >
<	< =	¬ <
=	¬ =	¬ >

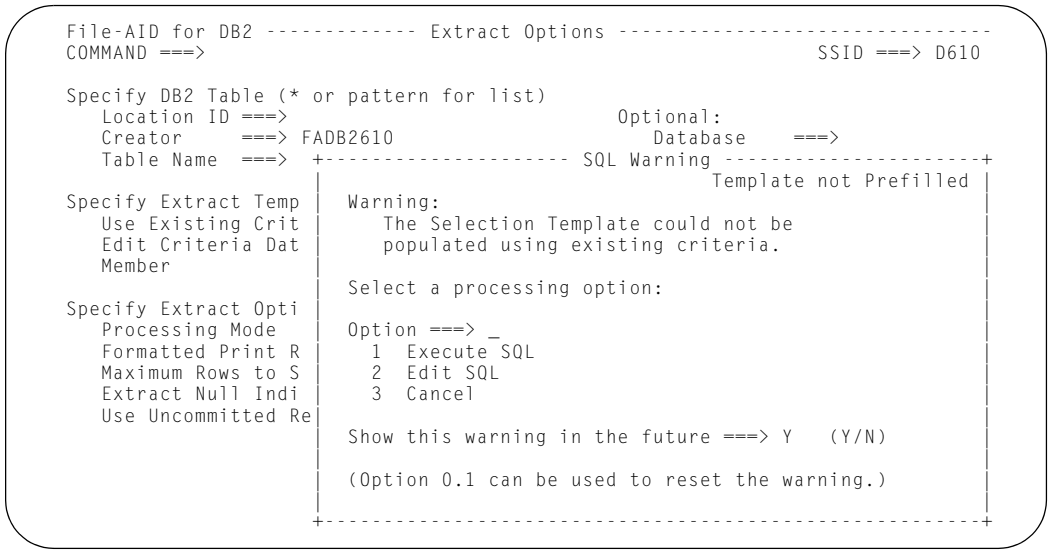
For all other WHERE clauses, you must enter the CAST function manually.

Selection Template Not Prefilled

The Selection Template screen cannot be prefilled when there is an error processing the selection criteria. These errors could include:

- A mismatch between the table name in the SELECT statement of the Edit Criteria dataset and the one specified in the Table field of the Edit Options screen.
- The specified criteria does not contain a valid SELECT statement.
- The dataset is empty.

Figure 14-6. Selection Template Not Prefilled



Now you have several choices:

- Select option 1 to execute the SQL of the existing Selection Criteria Dataset. If it contains a valid SELECT statement you can then extract the table specified in the selection criteria.
- Select option 2 to view and edit the SQL of the existing Selection Criteria Dataset before executing it. If the dataset or dataset member is empty or non-existing you can insert a SELECT statement in the Edit SPUFI Selection screen. See also "SQL Entry" on page 14-9 and "Edit SPUFI Select" on page 14-10. You then extract the table specified in the SELECT statement.
- Specify option 3 to cancel and return to the Extract Options screen where you can:
  - Select a different creator and table combination that match the table name of the Extract Criteria.
  - Specify **NO** in the Use Existing Criteria field.
  - Specify a different Extract Criteria dataset member combination that matches the creator and table combination.

SQL Entry

The SQL Entry Panel (Figure 14-7 on page 14-10) displays if you enter the **SQL** primary command on the EXTRACT Selection Template screen whenever the screen could not be prefilled. It prompts you to name and describe a dataset. The dataset can have a previously saved SELECT statement or it can be empty, allowing you to enter a new SELECT statement. If the dataset contains more than one SELECT statement, only the first statement is executed.

The panel prompts you for the input dataset name, maximum number of rows to be selected, and whether you want to edit the SELECT statement dataset before executing the SQL. When you press Enter from this screen, you proceed to the Edit SPUFI screen if you entered **YES** to the EDIT INPUT field. If you entered **NO**, File-AID *for DB2* executes your SQL immediately.

Figure 14-7. SQL Entry Panel (SPUFI)

```
File-AID for DB2 ------------- SQL Entry Panel --------------------------------
COMMAND  ===>

Enter the input dataset name:           (Can be sequential or partitioned)
 1  DATASET NAME     ===> SHOW2.SQL(SHOWSQL)
 2  VOLUME SERIAL    ===>               (Enter if not cataloged)
 3  DATASET PASSWORD ===>               (Enter if password protected)
 4  MAX SELECT ROWS  ===> 2000          (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT       ===> YES           (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

DATASET NAME

Name of the dataset (sequential or partitioned) in which the SQL query resides.

VOLUME SERIAL

Disk pack on which the dataset resides, if not cataloged.

DATASET PASSWORD

Password for the dataset, if password protected.

MAX SELECT ROWS

Maximum number of rows to be extracted from the table, view, or alias. An asterisk in the field indicates all rows are to be extracted. Only the first SELECT command in a dataset is executed.

EDIT INPUT

Indicate whether you want to edit the SQL before it is executed. If you enter **NO**, it is executed immediately.

Edit SPUFI Select

The Edit SPUFI Select screen displays when you enter the name of the dataset containing your SQL SELECT statement and **YES** in the EDIT INPUT field on the SQL Entry Panel. On this screen you can enter a new SELECT statement or alter the existing one. The statement should be in standard SPUFI format. Save the statement by pressing END and then press Enter from the SQL Entry Panel to proceed to the Extract Table screen (Figure 14-9 on page 14-11).

Figure 14-8. Edit SPUFI Screen with Sample Select Statement

```
EDIT ---- USERID.SHOW2.SQL(SHOWSQL) - 1.00 ------------------ COLUMNS 001 072
COMMAND  ===>                                                SCROLL ===> PAGE
****** ***************************** TOP OF DATA ****************************
000001 SELECT EMPNO,FIRSTNME,
000002    MIDINIT, LASTNAME, WORKDEPT,
000003    PHONENO, HIREDATE, JOB,
000004    EDLEVEL, SEX, BIRTHDTE,
000005    SALARY, BONUS, COMM
000006   FROM EMP2
000007   WHERE EMPNO > '000200'
```

Extract Table

The Extract Table screen is displayed once you identify the extract table and the extract criteria. Use it to specify the extract and control card output datasets. If you want to create a record layout of the extracted data, use this screen to specify the dataset name and the language in which you want it created (COBOL or PL/I).

File-AID *for DB2* prefills the dataset names with fully-qualified dataset names constructed from the user's ID and the object name. These dataset names may be modified, if necessary. Figure 14-9 shows a sample screen.

Figure 14-9. Extract Table Screen

```
File-AID for DB2 ---------- Extract Table USERID.EMP -------------- SSID: DSN
COMMAND  ===>

Specify Extract Dataset Format:
   EXTRACT Dataset format    ===>               (1 = Standard, 2 = Delimited)
   Change Delimited Options  ===>               (Yes or No)

Specify Output Datasets:
   EXTRACT DATASET      ===> 'USERID.*.UNLOAD'
     Member             ===>                    (* for member list)

   Control Card Dataset ===> 'USERID.*.CTROL'
     Member             ===>                    (* for member list)

   Create Layout File   ===> NO                 (Yes or No)
     LAYOUT Dataset     ===> 'USERID.*.LAYOUT'
     Member             ===>                    (* for member list)
     Language           ===>                    (1 = COBOL, 2 = PL/I)

An asterisk (*) in the above Dataset names will be replaced with:

   Qualifier            ===> EMP2
```

Specify Extract Dataset Format

Extract Dataset Format

Specify whether the data should be extracted in standard DB2 or delimited file format. You must enter either a 1 or 2, or you receive the error message "Invalid value".

Enter a **1** for standard file format.

Enter a **2** for delimited file format.

Note: Extracted data in delimited file format cannot be loaded using File-AID *for DB2*'s Load utility (see also "Load Dataset" on page 14-21).

Change Delimited Options

Specify whether you want to change the current delimited file format options or not. You must enter either Yes or No when you specify delimited format. Enter **Yes** to view and change the current options (see "Delimited Format Options" on page 14-12). Yes is only valid for delimited file format. Enter **No** if you do not want to change the current delimited options or if you want the standard extract dataset format.

Specify Output Datasets Area

Extract Dataset

The name of the dataset which will contain the extracted data. The second level qualifier is an asterisk (*****) that will be replaced with the table name specified below. The dataset may be either sequential, partitioned, new or existing. If the dataset exists, its LRECL **must** be greater than or equal to the calculated record length of the output data record. If it is not, you will receive an allocation error.

If the dataset is new, Figure 14-11 on page 14-14 is displayed. It may be allocated as either sequential or partitioned. The default is to create the dataset as sequential.

Member

If the file is a partitioned dataset, enter an existing or new member name. Enter an asterisk (*****) to select a member from a member list.

Control Card Dataset

The name of the dataset which will contain information about the structure of the data. The second level qualifier is an asterisk (*****) that will be replaced with the table name specified below. The dataset may be either sequential, partitioned, new or existing. The report displayed in Figure 14-23 on page 14-28 shows an example of control card data.

If the dataset is new, Figure 14-12 on page 14-15 is displayed. It may be allocated as either sequential or partitioned. The default is to create the dataset as sequential. with an LRECL of 80. The LRECL cannot be changed.

Member

If the file is a partitioned dataset, enter an existing or new member name. Enter an asterisk (*****) to select a member from a member list.

Note: Control Card Dataset will not be created for delimited extract file format.

Create Layout File

Enter YES to generate a COBOL or PL/I record layout when File-AID *for DB2* performs the extract. If you enter YES, you must supply values for the LAYOUT DATASET and LANGUAGE fields. Valid only with standard extract format.

Layout Dataset

The name of the dataset which will contain a COBOL or PL/I record layout based on the extract dataset. The second level qualifier is an asterisk (*****) that will be replaced with the table name specified below. The dataset may be either sequential, partitioned, new or existing.

If the dataset is new, Figure 14-13 on page 14-16 is displayed. It may be allocated as either sequential or partitioned. The default is to create the dataset as sequential with an LRECL of 80. The LRECL cannot be changed.

Note: When File-AID *for DB2* creates a COBOL record layout, an underscore (_) in any object name is converted to a dash (-).

Member

If the file is a partitioned dataset, enter an existing or new member name. Enter an asterisk (*****) to select a member from a member list.

Language

Enter a **1** for COBOL to generate a COBOL record layout. Enter a **2** for PL/I to generate a PL/I record layout.

Qualifier

The qualifier that replaces the asterisk (*****) in the above dataset names.

Delimited Format Options

The Delimited Format Options screen displays once you specified YES to the Change Delimited Options prompt on the “Extract Table Screen” on page 14-11. This screen shows the current options for the delimited extract file format. Use it to modify the options as needed to later download the extracted data. The modified options are saved until they are changed again. Figure 14-10 shows a sample screen with prefilled options.

Figure 14-10. Delimited Format Options Screen

```
File-AID for DB2 ---------- Extract Table USERID.EMP -------------- SSID: DSN
COMMAND  ===>

Specify Delimited Format Options:
   Quotation Marks               ===> 2             (1 = Single, 2 = Double)
   Column Delimiter              ===> ,
   Use Quotes on Character Data  ===> YES           (Yes or No)
   Use Quotes on Numeric Data    ===> NO            (Yes or No)
   Extract Column Names          ===> NO            (Yes or No)
   Date Format                   ===> YYYY-MM-DD    (Specify desired mask)
```

Specify Delimited Format Options

Quotation Marks

Specify which quotation marks, single (') or double (") should be used in the delimited file format. Double quotation marks are the initial default. Enter a **1** for single quotation mark. Enter a **2** for double quotation mark.

Column Delimiter

Specify which character should be used as the column delimiter. The column delimiter marks the end of the previous and the beginning of the next column. Any displayable character, including blank character, is acceptable. Comma (,) is the initial default.

Use Quotes on Character Data

Specify whether character data will be enclosed by quotation marks. When set to YES (initial default), the output looks like "JONES","MARY". Numbers containing special characters, for example date fields, are considered character data. For example, if set to YES the date field could look like this "1995-09-29". When set to NO output would look like this: JONES,MARY or 1995-09-29.

> **Note:** VARCHAR columns are expanded to their maximum length and padded with blanks.

Use Quotes on Numeric Data

Specify whether numeric data will be enclosed by quotation marks. When set to YES, the output looks like "12345","6789". There should not be any decimal points specified for fixed or floating point decimal columns. When set to NO, the output looks like 12345,6789. NO is the initial default.

Extract Column Names

Specify whether column names will be included as the first row in the extract file. Column names follow the same format as for Character data. A example is "EMPLOYEE_ID","LASTNAME","FIRSTNAME" etc. If YES, the extract file includes all column names regardless how many columns are extracted. The initial setting is NO.

Date Format

Specify the format to mask date data. Any valid date format is acceptable, for example YYYY/MM/DD, DD.MM.YYYY, or MM/DD/YY. The initial date format is the same as the installation default for date format.

Extract Dataset Allocation

The Extract Dataset Allocation screen (Figure 14-11 on page 14-14) displays if the extract dataset you specified on the Extract Table screen (Figure 14-9 on page 14-11) does not exist. When you have completed your entries, press Enter.

Figure 14-11. Extract Dataset Allocation Screen

```
File-AID for DB2 -------- EXTRACT Dataset Allocation --------------------------
COMMAND ===>

SPECIFY the EXTRACT Dataset attributes, then PRESS ENTER

EXTRACT Dataset      ===> 'USERID.EMP2.UNLOAD'
  Member             ===>

  VOLUME SERIAL      ===>            (Blank for authorized default volume) *
  GENERIC UNIT       ===>            (Generic group name or unit address) *
  SPACE UNITS        ===> TRKS       (BLKS, TRKS, or CYLS)
  PRIMARY QUANTITY   ===> 10         (In above units)
  SECONDARY QUANTITY ===> 5          (In above units)
  DIRECTORY BLOCKS   ===> 0          (Zero for sequential data set)
  RECORD FORMAT      ===> FB         (FB/VB)
  RECORD LENGTH      ===> 120
  BLOCK SIZE         ===> 23400



( * Only one of these fields may be specified)
```

EXTRACT DATASET

Name of the output extract dataset prefilled from the previous screen. May not be changed.

MEMBER

Member name of the output extract dataset. Prefilled if entered on the previous screen. May not be changed.

VOLUME SERIAL

Name of the disk pack where the dataset will reside.

GENERIC UNIT

Installation-defined device type (SYSDA, for example).

Notes:

- Space attributes are set to defaults defined during installation.
- If you are extracting to tape in batch mode specify the site specific valid tape unit in this field. The only valid tape unit is the one specified during the installation of File-AID *for DB2* with variable FITAPE. The initial default value is CART. See Table A-3 in the *File-AID for DB2 Installation Guide*.
- If you have only access to a different tape unit you **cannot** change the required FITAPE variable tape unit name to the one you can access **until** you edit the resulting JCL.

SPACE UNITS

The type of space unit used to define the dataset. Valid entries are BLKS (blocks), TRKS (tracks), or CYLS (cylinders).

PRIMARY QUANTITY

Number of space units allocated for the dataset.

SECONDARY QUANTITY

Amount of space allocated for the dataset if it exceeds the primary quantity.

DIRECTORY BLOCKS

Enter the number of directory blocks for the dataset. The default is zero (sequential). If a member name is specified on the Extract Table screen (Figure 14-9 on page 14-11), a value greater than zero should be entered. If no member is specified on the Extract Table screen and the number of directory blocks is changed to a value greater

than zero, File-AID *for DB2* creates a member named "TEMPNAME" in the unload dataset.

RECORD FORMAT

Indicates how the data in the dataset is structured: fixed blocked or variable blocked.

RECORD LENGTH

File-AID *for DB2* calculates the record length of the extract dataset based on the columns selected. You may increase this value, but if you decrease it, a file allocation error will result.

BLOCK SIZE

Size of each block of data. This value may be modified for more efficient use of disk space, or to comply with your site's standards. If you enter a block size that is less than the record length, File-AID *for DB2* sets a block size that is the same as the record length.

Control Card Dataset Allocation

The Control Card Dataset Allocation screen, displays if the control card dataset you specified on the Extract Table screen (Figure 14-9 on page 14-11) does not exist.

Figure 14-12. Control Card Dataset Allocation Screen

```
File-AID for DB2 -------- CONTROL CARD Dataset Allocation ----------------------
COMMAND ===>

SPECIFY the Control Card Dataset attributes, then PRESS ENTER

CONTROL CARD Dataset ===> 'USERID.EMP2.CTRL'
  Member             ===>

  VOLUME SERIAL      ===>            (Blank for authorized default volume) *
  GENERIC UNIT       ===>            (Generic group name or unit address) *
  SPACE UNITS        ===> TRKS       (BLKS, TRKS, or CYLS)
  PRIMARY QUANTITY   ===> 1          (In above units)
  SECONDARY QUANTITY ===> 1          (In above units)
  DIRECTORY BLOCKS   ===> 0          (Zero for sequential data set)
  RECORD FORMAT      ===> FB         (FB/VB)
  RECORD LENGTH         : 80
  BLOCK SIZE         ===> 6160



( * Only one of these fields may be specified)
```

CONTROL CARD DATASET

The name of the control card dataset. This name is prefilled from the Extract Table screen (Figure 14-9 on page 14-11). May not be changed.

MEMBER

Member name of the control card dataset, prefilled from the previous screen if entered. May not be changed.

VOLUME SERIAL

Name of the disk pack where the dataset will reside.

Note: Space attributes are set to defaults defined during installation.

GENERIC UNIT

Installation-defined device type (SYSDA, for example).

SPACE UNITS

The type of space unit used to define the dataset. Valid entries are BLKS (blocks), TRKS (tracks), or CYLS (cylinders).

PRIMARY QUANTITY

Number of space units allocated for the dataset.

SECONDARY QUANTITY

Amount of space allocated for the dataset if it exceeds the primary quantity.

DIRECTORY BLOCKS

Enter the number of directory blocks for the dataset. The default is zero (sequential). If a member name is specified on the Extract Table screen (Figure 14-9 on page 14-11), a value greater than zero should be entered. If no member is specified on the Extract Table screen and the number of directory blocks is changed to a value greater than zero, File-AID *for DB2* creates a member named "TEMPNAME" in the control card dataset.

RECORD FORMAT

Indicates how the data in the dataset is structured: fixed blocked or variable blocked.

RECORD LENGTH

Maximum record length. This value defaults to 80 and cannot be updated.

BLOCK SIZE

The size of each block of data in the dataset. This value may be changed for more efficient use of disk space, or to comply with site standards.

Record Layout Allocation

The Record Layout Dataset Allocation screen displays if the record layout dataset you specified on the Extract Table screen (Figure 14-9 on page 14-11) does not exist.

Figure 14-13. Record Layout Dataset Allocation Screen

```
File-AID for DB2 -------- RECORD LAYOUT Dataset Allocation ----------------------
-
COMMAND ===>

Specify the RECORD LAYOUT Dataset attributes, then PRESS ENTER

RECORD LAYOUT Dataset ===> 'USERID.EMP2.COBOL'
  Member              ===>

  VOLUME SERIAL       ===>           (Blank for authorized default volume) *
  GENERIC UNIT        ===>           (Generic group name or unit address) *
  SPACE UNITS         ===> TRKS      (BLKS, TRKS, or CYLS)
  PRIMARY QUANTITY    ===> 1         (In above units)
  SECONDARY QUANTITY  ===> 1         (In above units)
  DIRECTORY BLOCKS    ===> 0         (Zero for sequential data set)
  RECORD FORMAT       ===> FB        (FB/VB)
  RECORD LENGTH          : 80
  BLOCK SIZE          ===> 6160


( * Only one of these fields may be specified)
```

RECORD LAYOUT DATASET

The name of the record layout dataset. This name is prefilled from the Extract Table screen (Figure 14-9 on page 14-11) and may not be changed.

MEMBER

Member name of the record layout dataset, prefilled from the previous screen if entered. May not be changed.

VOLUME SERIAL

Name of the disk pack where the dataset will reside.

Note: Space attributes are set to defaults defined during installation.

GENERIC UNIT

Installation-defined device type (SYSDA, for example).

SPACE UNITS

The type of space unit used to define the dataset. Valid entries are BLKS (blocks), TRKS (tracks), or CYLS (cylinders).

PRIMARY QUANTITY

Number of space units allocated for the dataset.

SECONDARY QUANTITY

Amount of space allocated for the dataset if it exceeds the primary quantity.

DIRECTORY BLOCKS

Enter the number of directory blocks for the dataset. The default is zero (sequential). If a member name is specified on the Extract Table screen (Figure 14-9 on page 14-11), a value greater than zero should be entered. If no member is specified on the Extract Table screen and the number of directory blocks is changed to a value greater than zero, File-AID *for DB2* creates a member named "TEMPNAME" in the record layout dataset.

RECORD FORMAT

Indicates how the data in the dataset is structured: fixed blocked or variable blocked.

RECORD LENGTH

Maximum record length. This value defaults to 80 and cannot be updated.

BLOCK SIZE

The size of each block of data in the dataset. This value may be changed for more efficient use of disk space, or to comply with site standards.

Extract Confirmation

The Extract Confirmation screen (Figure 14-14 on page 14-18), displays after the unload, control card, and layout datasets are established. If all the datasets exist, the Extract Confirmation screen displays immediately after the dataset names have been entered on the Extract Table screen (Figure 14-9 on page 14-11).

The Extract Confirmation screen displays the name of the table from which the data will be extracted, as well as the output extract format, names and status of the unload, control, and layout datasets. All fields on this screen are protected and cannot be modified.

To begin the extract process, press Enter. The message **Extract in Progress** displays. To terminate processing, press END.

Figure 14-14. Extract Confirmation Screen

```
File-AID for DB2 -------------- Extract Confirmation -------------- SSID: DSN
COMMAND  ===>


Table Name: USERID.EMP2

Output Extract Dataset Format:  Standard

Output Extract Dataset                          Status
----------------------------------------------  -----------
'USERID.EMP2.UNLOAD'                            NEW


Output Control Card Dataset                     Status
----------------------------------------------  -----------
'USERID.EMP2.CTRL'                              NEW


Output Layout Dataset                           Status
----------------------------------------------  -----------
'USERID.EMP2.COBOL'                             NEW

Press ENTER to execute the Online Extract; End to cancel
```

Table Name

Name of the table being extracted.

Output Extract Dataset Format

Identifies the output extract dataset format, either Standard or Delimited.

Output Extract Dataset

Name of the output extract dataset.

Status

Status of the extract dataset: NEW or OLD.

Output Control Card Dataset

Name of the output control card dataset. Only for standard extract format.

Status

Status of the control card dataset: NEW or OLD.

Output Layout Dataset

Name of the output record layout dataset, if provided. Only for standard extract format.

Status

Status of the record layout dataset: NEW or OLD.

When you press Enter on the Extract Confirmation screen, the message **Extract in Progress** displays.

Batch Processing

Batch Table Extract to a Dataset

The batch processing mode is initiated for a table extract when you enter **2** for batch mode in the PROCESSING MODE field. The extract selection process is the same as for online processing. The following screens are presented:

- Extract Selection Template (Figure 14-5 on page 14-7) to select the columns to be extracted.
- Extract Table (Figure 14-9 on page 14-11) to specify extract, control card, and record layout datasets.
- Extract Dataset Allocation (Figure 14-11 on page 14-14) to specify space and DCB attributes for the extract dataset if it is a new dataset.
- Control Card Dataset Allocation (Figure 14-12 on page 14-15) to specify space and DCB attributes for the control card dataset if it is a new dataset.
- Record Layout Dataset Allocation (Figure 14-13 on page 14-16) to specify space and DCB attributes for the record layout dataset if it is a new dataset.

File-AID *for DB2* generates a temporary JCL dataset which is presented on an Edit screen. Modify it as necessary, then submit the job to process the extract.

Batch Table Extract to Tape

Table extract to tape uses the same procedure as "Batch Table Extract to a Dataset" on page 14-18 with one major exception:

On the Extract Dataset Allocation screen (Figure 14-11 on page 14-14) you specify the site specific valid tape unit in the GENERIC UNIT field. The only valid tape unit is the one specified during the installation of File-AID *for DB2* with variable FITAPE. The initial default value is CART. See Table A-3 in the *File-AID for DB2 Installation Guide*.

Note: If you have only access to a different tape unit you **cannot** change the required FITAPE variable tape unit name to the one you can access **until** you edit the resulting JCL.

Option 2 - Load from a Dataset to a Table

The sequence of screens described in this section illustrate the load option using SQL INSERT statements to process. The insert statements can be run either online or in batch.

Note: To load from tape you must use the SQL batch processing mode.

Load Options

The Load Options screen displays when you select option **2** from the Extract/Load Menu. It lets you specify the table or view into which you are loading data and your load processing option. Enter **SQL** in the LOAD OPTION field to use SQL insert statements for loading the data, either online or batch. The SQL mode is especially useful for loading a table residing in a tablespace that contains several tables or for a view that can be updated. If SQL is selected, you must have at least INSERT authority on the table and possibly DELETE and UPDATE, depending on the Load Processing Options defined later. If the DB2 option is selected, you need the authority to execute IBM's load utility.

Figure 14-15. Load Options Screen

```
File-AID for DB2 ------------------- Load Options -----------------------------
OPTION  ===>                                                  SSID ===> DSN

Specify DB2 Table:
   Location ID  ===>
   Creator      ===> USERID             (* for selection list)
   Table Name   ===> EMP2               (* for selection list)

Optional:
   Database     ===>
   Tablespace   ===>

Specify Load Option:
   Load Option  ===> 1                  (1 = SQL Insert Statements,
                                         2 = IBM DB2 Batch Load Utility)
```

Specify DB2 Table Area

Location ID

Location of the DB2 table.

Creator

ID of the person who created the DB2 table being used. Enter an asterisk (*****) for a selection list.

Table Name

Name of the DB2 table, synonym, view, or alias being used for selection. Enter an asterisk (*****) for a selection list.

Optional Area

Database

Name of the database to which you want your selection list confined.

Tablespace

Name of the tablespace to which you want your selection list confined.

Specify Load Option Area

Load Option

Specify the desired load option.

- **1** Uses SQL Insert statements. This option requires INSERT authority on the table.
- **2** Uses the IBM DB2 batch load utility. See also "DB2 Batch Load Table" on page 14-29.

Load Dataset Specifications

The Load Dataset Specifications screen (Figure 14-16 on page 14-21) displays after a DB2 table, view, or alias name has been entered and the **SQL** or **DB2** load option has been selected on the Load Options screen (Figure 14-15). Use this panel to specify the unload and control card datasets that are used as the input to the load process.

File-AID *for DB2* prefills both the unload and the control card dataset names with fully-qualified dataset names constructed from the user's ID and the object name. These dataset names may be modified, if necessary.

Figure 14-16. Load Dataset Specifications Screen

```
File-AID for DB2 - ---------- Load Dataset Specifications ----------- SSID: DSN
COMMAND  ===>

Specify Input Datasets:
   Load Dataset         ===> 'USERID.*.UNLOAD'
     Member             ===>                          (* for member list)

   Control Card Dataset ===> 'USERID.*.CTRL'
     Member             ===>                          (* for member list)









 An asterisk (*) in the above Dataset names will be replaced with:

    Qualifier           ===> EMP2
```

Specify Input Datasets Area

Load Dataset

The name of the dataset which contains the data that is loaded into the target object.

Note: Unload datasets in delimited file format (see "Specify Extract Dataset Format" on page 14-11) cannot be loaded with this load option. The second level qualifier is an asterisk (*****) that is replaced with the table name specified below. This dataset is an unload dataset created as a result of a previously executed standard extract operation (see also "Extract Table" on page 14-10). The dataset name may be fully-qualified or non-qualified.

Member

Specify a member name if the load dataset is partitioned. An asterisk may also be entered. If you enter an asterisk, you are presented with a list from which you may select a member.

Control Card Dataset

The name of the control card dataset which contains the control card information for the load process. The second level qualifier is an asterisk (*****) that is replaced with the table name specified below. This dataset is normally created when an extract is performed. It contains table information such as column names, lengths and types. The dataset name may be fully-qualified or non-qualified.

Member

Specify a member name when the control card dataset is partitioned. An asterisk may also be entered. If you enter an asterisk, you are presented with a list from which you may select a member.

Qualifier

The qualifier that replaces the asterisk (*) in the above dataset names.

Online Processing

SQL LOAD Table

The LOAD Table screen (Figure 14-17 on page 14-22) displays after the names of the unload and control card datasets have been entered on the Load Dataset Specifications

screen (Figure 14-16 on page 14-21). The parameters entered on this screen control the mode and manner in which the online load process functions.

If one or more triggers exist on the table being loaded, the message 'Applicable Trigger(s) will activate' displays in the upper right-hand corner.

Figure 14-17. Load Table Screen

```
File-AID for DB2 ----------- LOAD table - USERID.EMP2 ------------- SSID: DSN
COMMAND ===>

Specify Processing Option:
 Processing Mode             ===> 1         (1 = Online, 2 = Batch)

Specify Load Processing Option (select one) :
  _  Add (Discard Duplicate Keys)
  _  Add (Replace Duplicate Keys)
  _  Delete ALL Existing Data and Add New Data
  _  Add New Data to Empty Table

Optional Discard Processing:
 Save Discarded Rows         ===> No        (Yes or No)
 Maximum Rows to Discard     ===> 0         (* for no maximum)
 Dataset for Discarded Rows  ===>
   Member                    ===>           (* for member list)
```

Processing Mode

Two values may be entered: 1 for ONLINE or 2 for BATCH. The default is 1 (ONLINE). To perform an online load, leave this field unchanged.

Note: To load from tape you must specify **2** for SQL batch mode.

Specify Load Processing Option Area

Four options are listed in this area of the screen. These options describe the current status of the object into which the data will be loaded. Use an **S** line command to select an option. **Only one option may be selected**.

ADD (Discard Duplicate Keys)

Select this option when you are adding data to a table that is not empty. If this option is selected and the table has a unique key, a row from the load dataset is discarded if its key already exists in the target table. If the table does not have a unique key, duplicate rows are loaded. This option requires INSERT and DELETE authority on the table.

ADD (Replace Duplicate Keys)

Select this option when you are adding data to a table that is not empty. If this option is selected and the table has a unique key, a row from the load dataset will replace the existing row in the target table. If the table does not have a unique key, existing rows remain in the table and duplicate rows are loaded. This option requires INSERT and UPDATE authority on the table.

Delete ALL Existing Data and ADD New Data

Select this option if the contents of the target table are to be deleted before data from the load dataset is added to the table. This option requires INSERT and DELETE authority on the table.

Add New Data to Empty Table

Select this option if the table into which the data will be loaded is empty. If the target table is not empty and this option is selected, an error message displays. This option requires INSERT authority on the table.

Optional Discard Processing Area

Discarded rows are rows that cannot be loaded into the table due to a DB2 insert error.

Save Discarded Rows

YES Saves the discarded rows in the dataset specified in the Dataset for Discarded Rows field. If you enter YES in this field, you must also enter a discard dataset.

NO Discarded rows are not saved.

Maximum Rows to Discard

Enter a value from 0 to 99999999, or an asterisk (*****). The value entered in this field determines the number of times an insert error occurs before the load process is terminated. Once this value is reached, a rollback is performed and no rows are loaded into the table.

Dataset for Discarded Rows

Specify the name of the dataset in which the discarded data will be saved. The dataset name may be either fully-qualified or non-qualified. The dataset may be either a sequential file or a partitioned dataset. If the dataset does not exist, File-AID *for DB2* displays the Discard Dataset Allocation screen (Figure 14-20).

Care should be exercised when using an existing discard dataset. If the record length of the discard dataset is less than the record length of the dataset which contains the data that will be loaded into the target table, an allocation error will occur.

Member

If the discard dataset is a partitioned dataset (PDS), a member name may be specified in this field. An asterisk (*****) may also be entered. If an asterisk is entered, you are presented with a member list from which you may select a member.

When you press Enter, File-AID *for DB2* checks to see if there are any ROWIDs assigned to the target table. A pop-up displays allowing you to choose whether to continue copying or to cancel. The pop-up that displays depends on whether the ROWID column is set to GENERATED ALWAYS or GENERATED BY DEFAULT. Press Enter to continue with the copy or load, use END to terminate.

Figure 14-18. ROWID pop-up for GENERATED ALWAYS

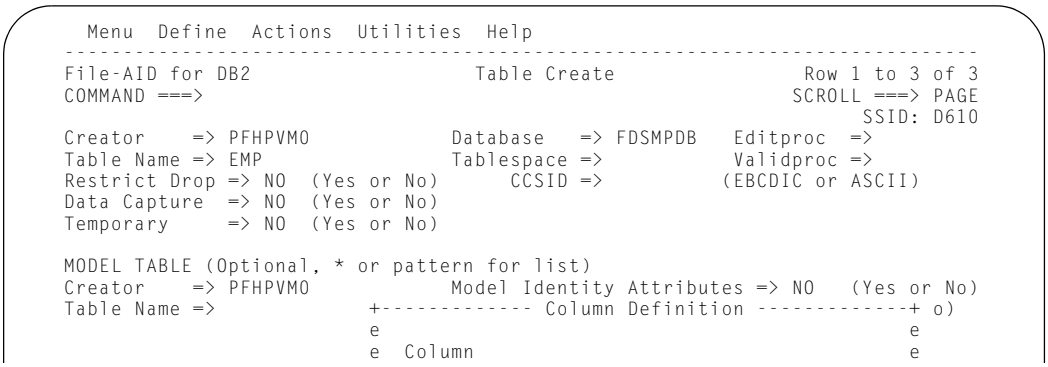


Figure 14-19. ROWID pop-up for GENERATED BY DEFAULT

```
+-----------------ROWID Column Processing---------------------+
|                                                             |
|   Warning - ROWID column defined with GENERATED BY DEFAULT  |
|                                                             |
|   The target table has the ROWID column defined as          |
|   GENERATED BY DEFAULT.  The values of the ROWID column     |
|   from the source table could introduce DUPLICATE           |
|   values in the target table.                               |
|                                                             |
|                                                             |
|   Press ENTER to continue with the Copy / Load              |
|     Use END to terminate the Copy / Load                    |
+-------------------------------------------------------------+
```

Discard Dataset Allocation

The Discard Dataset Allocation screen (Figure 14-20) displays if a discard dataset name has been specified in the DATASET FOR DISCARDED ROWS field of the Load Table screen (Figure 14-17 on page 14-22), and the dataset does not exist.

Figure 14-20. DISCARD DATASET ALLOCATION Screen

```
File-AID for DB2 -------- DISCARD Dataset Allocation --------------------------
COMMAND ===>

Specify the DISCARD Dataset attributes, then press ENTER

DISCARD Dataset Name  ===> 'USERID.DISCARD.FILE'
  Member              ===>

  VOLUME SERIAL       ===>            (Blank for authorized default volume) *
  GENERIC UNIT        ===> SYSDA      (Generic group name or unit address) *
  SPACE UNITS         ===> TRACKS     (BLKS, TRKS, or CYLS)
  PRIMARY QUANTITY    ===> 20         (In above units)
  SECONDARY QUANTITY  ===> 20         (In above units)
  DIRECTORY BLOCKS    ===> 0          (Zero for sequential dataset)
  RECORD FORMAT       ===> FB         (FB/VB)
  RECORD LENGTH       ===> 111
  BLOCK SIZE          ===> 23643


  ( * Only one of these fields may be specified )
```

DISCARD DATASET NAME

This field displays the name of the discard dataset that was entered on the Load Table screen (Figure 14-17 on page 14-22). This field is protected and may not be changed.

MEMBER

If a member name was provided on the LOAD TABLE screen, that member name is displayed. This field is protected and may not be changed.

VOLUME SERIAL

Name of the disk pack unit where the dataset is created.

GENERIC UNIT

A device type (SYSDA, for example), defined during the installation of File-AID *for DB2*.

SPACE UNITS

The type of space units that will be used to define the dataset. Permissible values are BLKS (blocks), TRKS (tracks), or CYLS (cylinders).

PRIMARY QUANTITY

The number of primary space units allocated for the dataset.

SECONDARY QUANTITY

The amount of space allocated for the dataset if the primary quantity is exceeded.

DIRECTORY BLOCKS

The number of directory blocks allocated for the dataset. The default is zero, indicating that the dataset is sequential. If a member name is specified on the LOAD TABLE screen, you should verify and, if necessary, change the number of directory blocks to a value greater than zero.

RECORD FORMAT

Indicates the structure of the discard dataset. Permissible values are FB (fixed block) and VB (variable blocked). The default is FB.

RECORD LENGTH

The length of each record in the discard dataset. The default value, which is displayed, is the LRECL of the load dataset that contains the data to be loaded into the target table. This value may be increased, but it should not be decreased. If the record length of the discard dataset is less than the record length of the load dataset, an allocation error will occur.

BLOCK SIZE

The size of each block of data in the discard dataset. This value may be changed for more efficient use of disk space, or to comply with site standards.

Load Confirmation

The Load Confirmation screen (Figure 14-21 on page 14-26) displays after the input load and control card datasets have been specified, and after the discard dataset is allocated, if necessary. This screen permits you to verify that the datasets specified for the load and the table that will be loaded are correct. All information on the screen is protected. No changes are allowed.

If the table and dataset names are correct, press Enter to begin the load process. If the table and/or dataset names are not correct, press END to return to the Load Table screen (Figure 14-17 on page 14-22).

Figure 14-21. Load Confirmation Screen

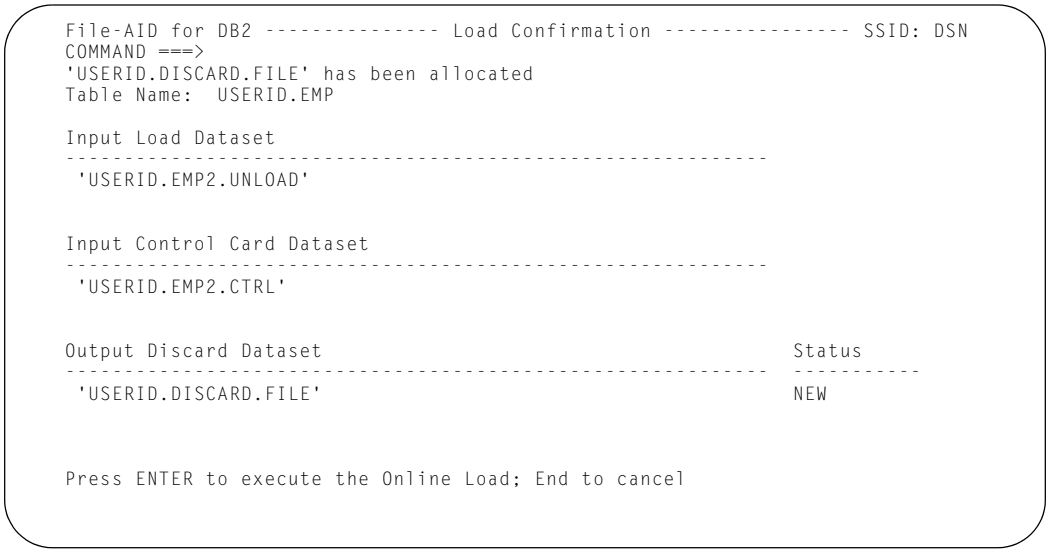


Table Name

The name of the table, view, or alias that is loaded.

Input Load Dataset

The name of the dataset which contains the data that is loaded into the object specified in the TABLE NAME field.

Input Control Card Dataset

The name of the dataset which contains the control card data that defines the data in the input load dataset.

Output Discard Dataset

The name of the discard dataset, if specified.

Status

The status of the discard dataset: NEW or OLD.

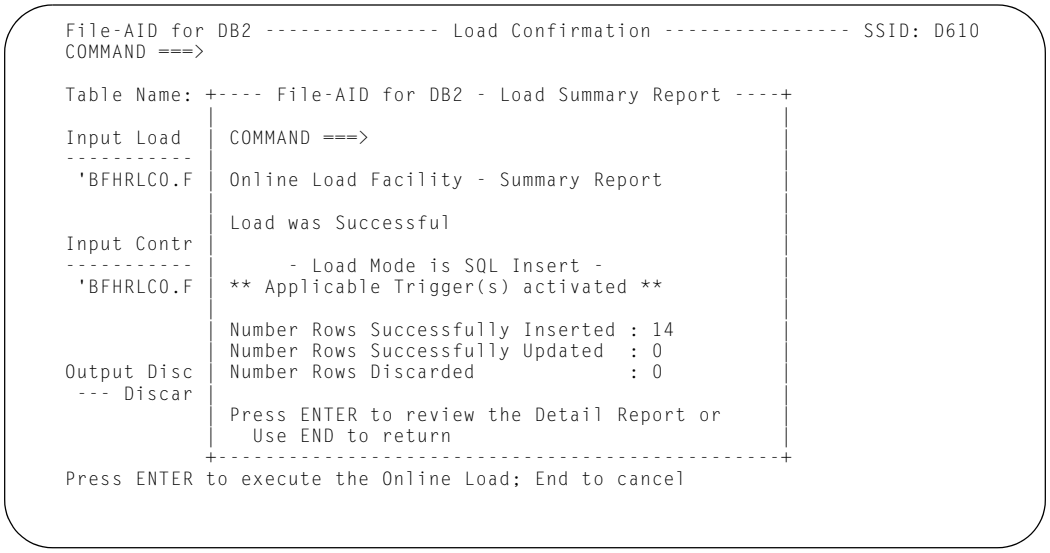
Online Load Summary

When online load processing has been completed, a window displays in the center of the Load Confirmation screen (Figure 14-22 on page 14-27). This window advises whether the load was successful or unsuccessful, and displays basic statistics on the load process.

If you press Enter, you are presented with a detailed load report (Figure 14-23 on page 14-28). This detailed load report resides in a temporary dataset. The dataset is discarded unless you save it using the appropriate ISPF commands (CREATE or REPLACE).

If one or more triggers exist on the table being loaded, the message ‘Applicable Trigger(s) activated’ displays in the upper right-hand corner.

Figure 14-22. Load Confirmation Screen with Report Pop-Up



Load was Successful/Unsuccessful

The load process is successful if rows are inserted or updated. If a DB2 insert error is encountered, the load is unsuccessful.

The load process is also unsuccessful if the Maximum Rows To Discard limit is reached (Figure 14-17 on page 14-22). For example, if Maximum Rows to Discard was set to 5, File-AID *for DB2* processes data until it encounters 5 errors. File-AID *for DB2* then terminates the load process and flags the load as unsuccessful.

Number of Rows Successfully Inserted

The number of rows inserted into the target object.

Number of Rows Successfully Updated

The number of rows updated in the target object. This field is only updated if the Add (Replace Duplicate Keys) option of the Load Table screen (Figure 14-17 on page 14-22), was selected.

Number of Rows Discarded

The number of rows in which a DB2 insert error condition occurred.

Detailed Load Report

The detailed load report contains specific information about the load process. The report is composed of three sections. The first section (Figure 14-23 on page 14-28), contains the control card input that was used to build the SQL INSERT statements for the load. The second section (Figure 14-24 on page 14-28), contains specific error information about each input record. The third section (Figure 14-25 on page 14-29), contains the same information displayed in the summary window.

The report dataset has a record length of 80, and may be saved in any standard ISPF dataset. Standard ISPF primary and line commands may be used to work with this report.

Figure 14-23. Detailed Load Report: Control Card Input

```
EDIT ---- SYS92310.T073841.RA000.USERID.R0022174 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
000001 .
000002    ONLINE Load Facility - Control Card Input
000003 .
000004 LOAD DATA RESUME YES INTO TABLE USERID.EMP2
000005 (EMPNO  POSITION (21) CHAR  (6) ,
000006  FIRSTNME POSITION (27) VC ,
000007  MIDINIT POSITION (41) CHAR  (1) ,
000008  LASTNAME POSITION (42) VC ,
000009  WORKDEPT POSITION (59) CHAR  (3)  NULLIF (1:2)=X'FFFF' ,
000010  PHONENO POSITION (62) CHAR  (4)  NULLIF (3:4)=X'FFFF' ,
000011  HIREDATE POSITION (66) DATE EXTERNAL  (10)  NULLIF (5:6)=X'FFFF' ,
000012  JOB POSITION (76) CHAR  (8)  NULLIF (7:8)=X'FFFF' ,
000013  EDLEVEL POSITION (84) SMALLINT  NULLIF (9:10)=X'FFFF' ,
000014  SEX POSITION (86) CHAR  (1)  NULLIF (11:12)=X'FFFF' ,
000015  BIRTHDATE POSITION (87) DATE EXTERNAL  (10)  NULLIF (13:14)=X'FFFF' ,
000016  SALARY POSITION (97:101) DEC  NULLIF (15:16)=X'FFFF' ,
000017  BONUS POSITION (102:106) DEC  NULLIF (17:18)=X'FFFF' ,
000018  COMM POSITION (107:111) DEC  NULLIF (19:20)=X'FFFF' )
000019 ------------------------------------------------------------------------
```

Figure 14-24. Detailed Load Report: Input Error Information

```
EDIT ---- SYS92310.T073841.RA000.USERID.R0022174 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
000019 ------------------------------------------------------------------------
000020
000021  ONLINE LOAD FACILITY - ERROR REPORT
000022
000023        INPUT      DISCARD        ERROR    TABLE
000024       RECORD      RECORD         TYPE     NAME
000025       ------      -------        -------  ----------------------------
000026           22            1        -803     USERID.EMP2
000027           23            2        -803     USERID.EMP2
000028           24            4        -803     USERID.EMP2
000029           25            5        -803     USERID.EMP2
000030
000031 ------------------------------------------------------------------------
```

INPUT RECORD

The location of the record in error in the input load dataset.

DISCARD RECORD

The relative location of the record in the discard dataset.

ERROR TYPE

The SQL code identifying the type of error that occurred.

TABLE NAME

The name of the object that was being loaded.

Figure 14-25. Detailed Load Report: Summary Window Information

```
EDIT ---- SYS92310.T073841.RA000.USERID.R0022174 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
000031 ------------------------------------------------------------------------
000032 .
000033    ONLINE Load Facility - Summary Report
000034 .
000035 Number Rows Inserted     - 14
000036 Number Rows Updated      - 0
000037 Number Rows Discarded    - 4
000038 ------------------------------------------------------------------------
****** **************************** BOTTOM OF DATA ****************************
```

Batch Processing

Load Option

When the SQL LOAD option is selected on the Load Options screen, but the load is to be executed in batch mode, enter **2** for BATCH in the PROCESSING MODE option field on the Load Table screen (Figure 14-17 on page 14-22).

Specify the LOAD PROCESSING options and the optional discard processing for batch execution in the same manner described for online execution. After all parameters have been specified and Enter is pressed, you are presented with the Discard Dataset Allocation screen (if the discard dataset is new).

Once the parameters for the discard dataset have been defined, File-AID *for DB2* will automatically generate the JCL for the batch load. The JCL is displayed in a temporary dataset. If necessary, modify the JCL and submit the job. Standard ISPF primary and line commands may be used to work with the JCL.

The following section describes the DB2 Batch Load Table screen and generating the batch load.

DB2 Batch Load Table

The Batch Load Table screen displays when **DB2** is specified for LOAD OPTION on the Load Options screen (Figure 14-15 on page 14-20), and after the load and control card datasets have been specified. When this option is selected, the DB2 Load Utility is used to load the target object.

If triggers exist on the extract table, the message 'Trigger(s) exist' displays in the upper right-hand corner of the EXTRACT Selection Template screen.

Figure 14-26. Batch Load Table Screen

```
File-AID for DB2 - LOAD table - USERID.EMP2 ----------------------- SSID: DSN
COMMAND ===>

Specify Load Processing Option (select one) :
  _   Target Tablespace is Empty (RESUME NO)
  _   Add New Data (RESUME YES)
  _   Delete Existing Data and Add New Data (REPLACE)

Specify DB2 LOAD Processing Options :
 Log during the LOAD           ===>  NO         (Yes or No)
 Enforce Referential Integrity ===>  NO         (Yes or No)
 Discard Processing            ===>  NO         (Yes or No)
 Discard Limit                 ===>  0          (0 for no maximum)
 Dataset for Discarded Rows    ===>
```

Specify Load Processing Option Area

Select only one of the options listed.

Target Tablespace is Empty (RESUME NO)

Select this option only if the target tablespace is empty. If the tablespace is not empty, the utility will stop.

Add New Data (RESUME YES)

Add dataset data to the table. If the tablespace is empty, a warning message is generated, but the load process continues.

Delete Existing Data and Add New Data

Reset the tablespace and all tables within it to empty. Associated indexes are also reset if this option is selected.

DB2 Load Processing Options Area

Log During the Load

YES Logs the load activity in the DB2 file.

NO Default. Load activity is not logged. The tablespace is placed in a COPY PENDING state.

Enforce Referential Integrity

YES The rules of referential integrity are enforced when loading data.

NO Default. The rules of referential integrity are bypassed when loading data.

Discard Processing

Discarded rows are rows that cannot be loaded into the table due to a DB2 insert error.

YES Saves discarded rows in the dataset specified in the Dataset for Discarded Rows field. If you enter YES, you must also enter a discard dataset.

NO Default. Discarded rows are not saved.

Discard Limit

Maximum number of From Table rows you want discarded before the load process is halted. Enter zero, or any value up to 99999999. A value of zero allows the entire input dataset to be discarded if errors occur. If a non-zero value is entered, load processing is halted and updates rolled back when the number of errors exceeds the specified value. The default is zero.

Dataset for Discarded Rows

Enter the name of a dataset if discarded rows are to be saved. The name may be fully-qualified or non-qualified.

Generating the DB2 Batch Load

File-AID *for DB2* automatically generates the DB2 batch load JCL after all appropriate options have been selected on the DB2 Batch Load Table option screen (Figure 14-26 on page 14-29), and you have pressed Enter. Review and, if necessary, modify the JCL using standard ISPF primary and line commands. After the JCL has been reviewed, it may be submitted and/or saved.

Option 3 - Extract and Load

The Extract and Load Options screen displays when you select option 3 from the Extract/Load menu (Figure 14-1 on page 14-2). This option allows you move tables from one location to another smoothly.

To use this option, the target table must already exist and the target table must be identical in definition to the source table. Column matching is not supported.

The only new screen used for option 3 is the Extract and Load Options screen.

Figure 14-27. Extract and Load Options Screen

```
File-AID for DB2 --------- Extract and Load Options ----------------------------
COMMAND ===>                                                    SSID ===> D610

Specify Processing Option:
 Processing Mode ===>_                     (1 = online, 2 = Batch)

From Table:                              To Table:
   Location ID ===>                         Location ID ===>
   Creator     ===>                         Creator     ===>
   Table Name  ===>                         Table Name  ===>
```

From Table Area:

Location ID

Location of the table from which to copy. If the subsystem is local, this field can be left blank.

Creator

ID of the person who created the DB2 table. Enter an asterisk (*) for a selection list.

Table Name

Name of the DB2 table, synonym, view or alias from which data is extracted. Enter an asterisk (*) for a selection list.

To Table Area

Location ID

Location of the table to which to copy. If the subsystem is local, this field can be left blank.

Creator

ID of the person who created the DB2 table. Enter an asterisk (*) for a selection list.

Table Name

Name of the DB2 table, synonym, view or alias to which data is copied. Enter an asterisk (*) for a selection list.

Processing

After this initial screen, processing follows the normal path. The screens all base upon previous screens in the series, just as in normal processing, except that information in the Load screens is now based upon previous Extract screens. Which screens are accessed will depend on whether you selected batch or online as your processing mode:

Online

1. Extract Options screen (Figure 14-2 on page 14-3)
2. Extract Selection Template (Figure 14-5 on page 14-7)

3. Extract Table screen (Figure 14-9 on page 14-11)
4. Extract Confirmation (Figure 14-14 on page 14-18)
5. Load Table screen (Figure 14-17 on page 14-22)
6. Load Confirmation screen (Figure 14-21 on page 14-26)
7. Load Summary Report screen (Figure 14-25 on page 14-29).

Batch

1. Extract Options screen (Figure 14-2 on page 14-3)
2. Extract Selection Template (Figure 14-5 on page 14-7)
3. Extract Table screen (Figure 14-9 on page 14-11)
4. Extract Confirmation (Figure 14-14 on page 14-18)
5. Load Table screen (Figure 14-17 on page 14-22)
6. Load Confirmation screen (Figure 14-21 on page 14-26)
7. JCL to submit.

Chapter 15.
Batch Facility

This chapter describes the File-AID *for DB2* Batch Facility. The Batch Facility allows you to execute SQL in batch mode under the control of File-AID *for DB2*. Any SQL that can be executed dynamically can execute in File-AID *for DB2* batch. From the Batch Facility, you can identify the location of existing SQL, or invoke an ISPF Edit session and update SQL or code new SQL to then be executed. File-AID *for DB2* does not provide any assistance in SQL creation.

When the SQL contains UPDATE or DELETE processing, you can specify the interval at which COMMIT processing is invoked. No change is required to the SQL; File-AID *for DB2* will transform the SQL before execution so that the COMMITs can be executed as required. This COMMIT processing reduces the impact of running an SQL statement that affects many rows.

Note: Restart is not provided when running the Batch Facility. Compuware recommends that before the Batch Facility job is run, you take an image copy of all tables to be updated. If the job fails, you can use the image copy to restore the tables.

Create Trigger Statement

If you issue a CREATE TRIGGER statement, you must precede it with a #SET TERMINATOR control statement to set the SQL terminator to a value other than semi-colon. The pound sign (#) is recommended. Then, follow it with another #SET TERMINATOR control statement to reset the SQL terminator back to semi-colon.

Example:

```
--#SET TERMINATOR #

CREATE TRIGGER userid.trigger_name

            AFTER UPDATE

            OF column1, column2

            ON userid.table_name

            FOR EACH ROW MODE DB2SQL

            WHEN when-clause

            BEGIN ATOMIC

                SQL statement;

                SQL statement;

             END #

            --#SET TERMINATOR ;
```

CAUTION:

If you do not precede and follow the CREATE TRIGGER statement with a --#SET TERMINATOR control statement, the CREATE TRIGGER statement may produce unpredictable results.

Refer to the *IBM Database 2 SQL Reference*, CREATE TRIGGER section, for more information.

Accessing the Batch Facility

Access the Batch Facility from the File-AID *for DB2* Primary Option Menu or the Utility Selection Menu. To start the Batch Facility from the File-AID *for DB2* Primary Option Menu, select option **3.9**. To start the Batch Facility from the Utility Selection Menu, select option **9**.

Figure 15-1 illustrates option 3.9 on the File-AID *for DB2* Primary Option Menu.

Figure 15-1. File-AID *for DB2* Primary Option Menu

```
File-AID for DB2 - 4.8 ------ Primary Option Menu ------------------------------
OPTION  ===>
                                                                More:     +
   0  DEFAULTS and USER PARMS                          USERID   -    PFHCAB0
   1  BROWSE                                           TIME     -    09:08
   2  EDIT                                             TERMINAL -    3278
   3  UTILITIES                                        PF KEYS  -    24
      3.2 - Create, Drop, Alter DB2 Objects            DB2 SSID ===> D701
      3.3 - Copy Rows Between Tables
      3.4 - Object List Processing
      3.5 - SQL Command Manager
      3.6 - Display, Grant, Revoke or Modify DB2 Privileges
      3.7 - Extract and Load Tables
      3.8 - DBA-XPERT Column Impact Analysis
      3.9 - Batch Execute SQL with Auto-COMMIT Processing
   4  SQL ANALYSIS  - SQL Development and Analysis
   5  PRINT         - Print table Data or Audit Trail
   6  RELATIONSHIPS - Relationship Facilitator
   R  File-AID/RDX  - Transfer to File-AID/Related Data XPERT
   XT XPEDITER      - Transfer to XPEDITER
   T  TUTORIAL      - Display information about File-AID for DB2
   C  CHANGES       - Display summary of File-AID for DB2 changes
      Copyright (c) 1989 - 2005 Compuware Corporation. All Rights Reserved.
  Unpublished - rights reserved under the Copyright Laws of the United States.
```

Batch Execute SQL

When you select option **3.9** from the File-AID *for DB2* Primary Option Menu, the Batch Execute SQL screen is displayed. The Batch Execute SQL screen lets you specify the location of the SQL, processing options, and print options. See Figure 15-2.

Figure 15-2. Batch Execute SQL

```
File-AID for DB2 -------------- Batch Execute SQL -----------------------------
COMMAND ===>                                                    SSID ===> D701

Specify SQL Dataset: (Leave Dataset Name blank for temporary dataset)
   Edit SQL         ===> NO               (Yes or No)
   Dataset Name     ===>
   Member           ===>                  (Blank or Pattern for list)

Specify Processing Options:
   Automatic COMMIT processing  ===>        (Yes or No)
   Auto commit row interval     ===>        (Rows to update before commit)
   Auto commit time interval    ===>        (Maximum seconds before commit)
   SQL Terminator               ===> ;      (SQL statement terminator)

Specify Print Options for SELECT statements:
   Print Mode             ===> 1           (1 - Table, Multiple rows per screen
                                             2 - Row, One row per screen)
   Maximum Rows to Select ===> *           (* for all rows)
```

Specify SQL Dataset

Edit SQL

Specify whether or not you want to make changes to the SQL dataset.

If you enter **NO** in this field, you must have a location where the SQL is saved in a dataset. If you enter **YES** and you leave the Dataset name blank, a temporary PDS member along with a warning will display. In the temporary PDS member you can enter the SQL The warning message indicates that the SQL you have entered will not be saved. If you enter **YES** and a dataset name and an existing member name, then you are able to edit the PDS member.

Dataset name

Name of the dataset in which the SQL resides. The dataset must be partitioned and have a record length of 80. If the record length is not 80, an error message is displayed. Type the Dataset name completely with or without quotes. You can also type the member name, following the Dataset name. For example: 'userid.my.sql(member)' or my.sql(member).

If you leave Dataset name blank, and you entered NO in the Edit SQL field, then an error message is displayed.

Member

The field is prefilled with the last specified member name. If you leave this field blank or specify an asterisk (*****), File-AID *for DB2* displays the Member List for the dataset. Specify a pattern, for example PART*****, to narrow down the member list to the matching member names. The asterisk (*****) is the only valid pattern character.

Specify Processing Options

Automatic COMMIT Processing

Specify **YES** to change the frequency (how often) of commits when an UPDATE or DELETE statement is processed. Specify **NO** for no commits, other than the one commit after each UPDATE or DELETE statement is processed.

Auto commit row interval

Specify the maximum number of rows before changes are committed. You can enter up to six numeric digits, 0 through 9. The default is 999999.

Auto commit time interval

Specify the number of seconds before changes are committed. You can enter up to four numeric digits, 0 through 9. The default is 9999.

Note: If you enter both auto commit number of rows AND number of seconds, then the commit will occur at whichever interval is reached first.

SQL Terminator

Character that terminates the SQL statement. This can be any alphanumeric character except for a blank, comma, single or double quote, left or right parenthesis, or underscore. The default is a semicolon.

Specify Print Options for SELECT Statements

Print Mode:

Enter the desired presentation mode for printing. This field is updated with the last mode in use.

1 Presents multiple rows per screen or page. Column names are listed across the top of the screen. This is called *table mode*.

2 Presents one row per screen or page. Column names are listed down the left side of the screen or page and column data is listed down the right side of the screen or page. This is called *row mode*.

See “Display Modes” on page 6-14 for more information.

Maximum rows to select

Specify a numeric value indicating the maximum number of rows to be selected, or specify an asterisk (*) to select all rows. The default value is retrieved from User Parms (Option 0). See “Option 1, Browse/Edit/SQL Analysis Display Options” on page 5-2 for information on changing the default.

Editing SQL

Once you press Enter from the Batch Execute SQL screen, a temporary dataset or existing SQL is displayed. Use ISPF edit to modify existing SQL or enter new SQL

The SQL dataset can include SQL statements, comments, and control statements. SQL statements can span multiple lines. If line numbers are present, they will be ignored. SQL statements must be terminated with a semicolon or the SQL terminator that you specified. You can enter any number of SQL statements.

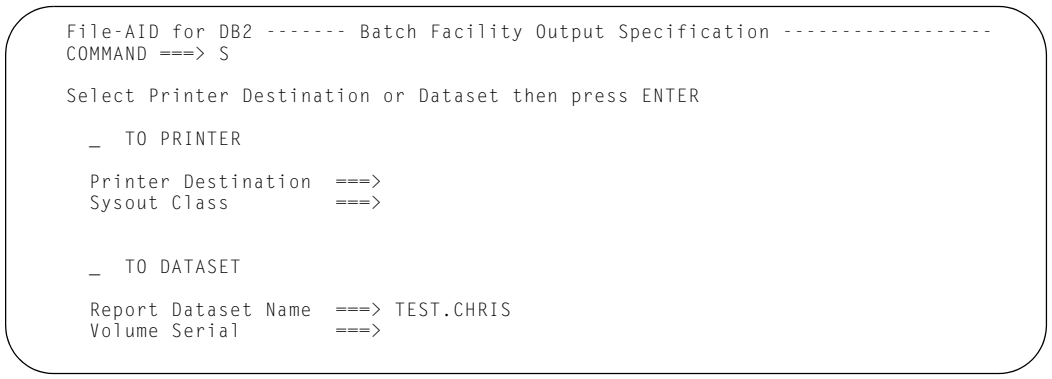
Comments start with an asterisk (*) in column 1, or two hyphens (--) anywhere on a line. Control statements start with two hyphens anywhere on a line. Comments and control statements cannot span lines.

The control statement ROWS_FETCH controls the number of rows that are processed by the SQL statements that follow. If you do not include a ROWS_FETCH statement, or if you specify the default of -1, then all rows are processed.

Batch Facility Output Specification

The Batch Facility Output Specification screen lets you route the output to a printer or a file. See Figure 15-3.

Figure 15-3. Batch Facility Output Specification



You cannot elect to send the report to both destinations. Use the S line command to select the destination and enter the required fields for the appropriate destination. Set the default print options and job statement by selecting option 0, Defaults and User Parameters, from the Primary Option Menu. See “Option 7, Print Options” on page 5-10 and “Option 3, Batch Options” on page 5-6.

Once all the parameters have been defined, File-AID *for DB2* automatically generates the JCL for the Batch Facility (see Figure 15-4).

Figure 15-4. Sample Batch Facility JCL (1 of 2)

```
 //JOBCARD
//*
//*
//*
//*-------------------------------------------------------------------------------------------
------------------------
//UTIL EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=8192K,COND=(7,LT)
//*-------------------------------------------------------------------------------------------
------------------------
//* This is the customized JCL to run TSO from batch.
//* It is imbedded in the skeletons for F2EXTR, F2LOAD, F2ANLBCH,
//* and F2IMPBCH
//*-------------------------------------------------------------------------------------------
------------------------
//STEPLIB  DD DISP=SHR,
// DSN=FADB2.V4R8Mx.ISPLLIB
//         DD DSN=DSN710.D701.SDSNEXIT,
// DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,
// DISP=SHR
//*
//ISPPLIB  DD DISP=SHR,
// DSN=ISP.SISPPENU
//         DD DISP=SHR,
// DSN=FADB2.V4R8Mx.ISPPLIB
//*
//ISPMLIB  DD DISP=SHR,
// DSN=FADB2.V4R8Mx.ISPMLIB
//         DD DISP=SHR,
// DSN=ISP.SISPMENU
//*
//ISPSLIB  DD DISP=SHR,
// DSN=ISP.SISPSENU
//         DD DISP=SHR,
// DSN=FADB2.V4R8Mx.ISPSLIB
//*
//ISPTLIB  DD DISP=SHR,
// DSN=FADB2.V4R8Mx.ISPTLIB
//         DD DISP=SHR,
// DSN=ISP.SISPTENU
//*
//ISPPROF  DD DSN=&BATPROF,DISP=(NEW,DELETE,DELETE),
//            DCB=(BLKSIZE=3120,LRECL=80,DSORG=PO,RECFM=FB),
//            SPACE=(TRK,(1,,1)),UNIT=SYSDA
//*
//ISPCTL1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(TRK,(5,5,0)),
//            DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB)
//SYSTMP01 DD DSN=&TMP,
//            DISP=NEW,UNIT=SYSALLDA,SPACE=(TRK,(5,5,0)),
//            DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB)
//SYSTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//F2SYSOUT DD SYSOUT=*,HOLD=YES
```

Figure 15-5. Sample Batch Facility JCL (2 of 2)

```
//SYSIN    DD *

          *** User entered SQL will be inserted here ***

/*
//SYSTSIN  DD *
ISPSTART PGM(F2FADBAT) NEWAPPL(F2DE)
//PARM     DD *
F2SSID       D701
FIPLAN       FD460052
CURSQLID     PFHPVM0
MODEOPT      T
FRGNDISP     NO
UPKEYEDI     NO
PRINTOPT     YES
PRDSNAME     "''"
PRDSVOL      ""
PRPSW        ""
PRLSPAG      55
FDB2VERS     4.6.0.52
SHOWSQL      2
NONDISP      ~
/*
```

The JCL is displayed in a temporary dataset. If necessary, modify the JCL using standard ISPF primary and line commands, and then submit the job. The following parameters are included in the JCL:

Parameter	Description
F2SSID	Subsystem ID
FIPLAN	Current plan
CURSQLID	Current SQLID
MODEOPT	Type of printing: T = Table R = Row
FRGNDISP	Foreign key display
UPKEYEDI	Update with unique
PRINTOPT	Print option: Y = to printer N = to dataset
PRDSNAME	Print dataset name (ignored if PRINTOPT field is Y)
PRDSVOL	Print dataset volser
PRPSW	Print dataset password
PRLSPAG	Print lines per page.

Parameter	Description
FDB2VERS	Version of File-AID *for DB2*
SHOWSQL	Type of logging in effect
NONDISP	Character used to represent non-displayable characters

Chapter 16.
SQL Development and Analysis

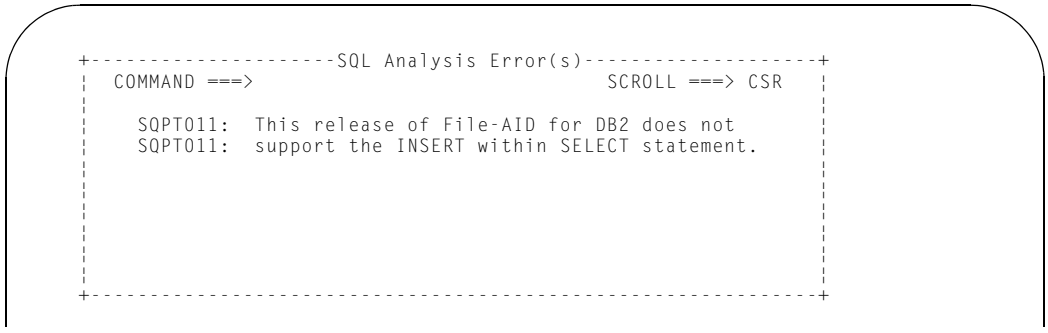
This chapter describes the SQL Development and Analysis function (SQL Analysis) of File-AID *for DB2*. The following features are described:

- Toleration of DB2 Version 8 SQL Clause or Syntax
- Maintain Explain Table(s)
- SQL Plan Analysis
- Explaining Objects in Batch
- Review and Print DB2 Explain Reports
- SQL Source Development and Analysis

Toleration of DB2 Version 8 SQL Clause or Syntax

DB2 Version 8 SQL is not supported in File-AID *for DB2*. When DB2 Version 8 clause or syntax is detected, an error message is displayed. This message indicates that the SQL clause or syntax is not supported. The associated SQL statement will not be processed.

Figure 16-1. SQL Clause or Syntax Not Supported



In Option 4.1, SQL Plan Analysis, additional SQL statements will continue to be processed, if they exist.

Accessing the SQL Analysis Function

To access the SQL Analysis function, select option **4** on the File-AID *for DB2* Primary Option menu. The SQL Analysis and Development Menu (Figure 16-2) displays. Select the option to be used.

Figure 16-2. SQL Analysis and Development Menu

```
File-AID for DB2 ----------- SQL Analysis and Development Menu ----
 OPTION  ===>                                                SSID:701

    0  Create or Maintain Explain Table(s)

    1  SQL Plan Analysis

    2  Review or Print DB2 Explain Reports

    3  ISPF/PDF  SQL Source Development and Analysis

    L  LIBRARIAN SQL Source Development and Analysis

    P  PANVALET  SQL Source Development and Analysis
```

OPTION

Enter the desired option.

0 Displays the Create or Maintain Explain Table screen ("Option 0, Create or Maintain Explain Table(s)" on page 16-2).

1 Displays the SQL Plan Analysis Specification screen ("Option 1, SQL Plan Analysis" on page 16-7).

2 Displays the SQL Explain Report Specification screen ("Option 2, Review or Print DB2 Explain Reports" on page 16-41).

3 Displays ISPF/PDF SQL Source Development and Analysis screen ("Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44).

L Initiates CA-LIBRARIAN SQL Source Development and Analysis (see "Option L, LIBRARIAN Source Development and Analysis" on page 16-59). If you do not have CA-LIBRARIAN, this option may not appear on your screen. Contact the person who installed File-AID *for DB2* if you think this option should appear on your screen.

P Initiates CA-PANVALET SQL Source Development and Analysis (see "Option P, PANVALET Source Development and Analysis" on page 16-59). If you do not have CA-PANVALET, this option may not appear on your screen. Contact the person who installed File-AID *for DB2* if you think this option should appear on your screen.

Option 0, Create or Maintain Explain Table(s)

On this screen you can create Plan, Statement, and Function tables, as well as maintain and delete rows from them.

Select option **0** from the SQL Analysis and Development menu. The Create or Maintain Explain Table(s) screen displays.

Figure 16-3. Create or Maintain Explain Table(s) Screen

```
File-AID for DB2 ----- Create or Maintain Explain Table(s) --------------------
OPTION  ===>                                                        SSID: D701


Plan_Table AUTHID ===> USERID    Authorization ID for all EXPLAIN tables

   1  Create Explain Table(s)

   2  Alter or Upgrade Explain Table(s)

   3  Delete rows from Explain Table(s)


     (Explain Tables: PLAN_TABLE, F2PLAN_TABLE, DSN_STATEMNT_TABLE,
                      DSN_FUNCTION_TABLE)
```

1, Create PLAN_TABLE, DSN_STATEMNT_TABLE, DSN_FUNCTION_TABLE

Select option **1** from the Create or Maintain Explain Table(s) screen. The Create Explain Table(s) screen displays.

On this screen you can create the Plan table, as well as the Statement and Function tables. The Plan and F2Plan tables are required. You must explicitly specify all database names.

Wildcards are not valid on this screen.

Figure 16-4. Create Explain Table(s) screen

```
File-AID for DB2 ----------- Create Explain Tables ---------------------------
COMMAND ===>                                                        SSID: D701

Select the tables below to be created.

Plan_Table AUTHID ===> USERID    Authorization ID for all EXPLAIN tables
Required Tables:
  Specify the Database and Tablespace for PLAN_TABLE
    Database   ===> USERIDDB
    Tablespace ===> USERIDTS      (blank for implicit Tablespace)
  Specify the Database and Tablespace for F2PLAN_TABLE
    Database   ===> USERIDDB
    Tablespace ===> USERIDTS      (blank for implicit Tablespace)

Optional Tables:                  (Release DB2 6.1 and higher)
_ Specify the Database and Tablespace for DSN_STATEMNT_TABLE
    Database   ===> USERIDDB
    Tablespace ===>               (blank for implicit Tablespace)
_ Specify the Database and Tablespace for DSN_FUNCTION_TABLE
    Database   ===> USERIDDB
    Tablespace ===>               (blank for implicit Tablespace)
Instructions:
   Press ENTER to continue or END to cancel.
```

2, Alter or Upgrade PLAN_TABLE

Select option **2** from the Create or Maintain Explain Table(s) screen. The Alter Plan Table screen displays.

On this screen you can upgrade your plan table from an older version of DB2 to a more current version, and also drop/recreate your plan table and F2 plan table. The example shows upgrading to version 7.

Figure 16-5. Alter Plan Table Screen

```
File-AID for DB2 ----------- Alter Plan Table ---------------------------------
 COMMAND ===>                                                    SSID: D701


 Plan_Table AUTHID ===> USERID    Authorization ID for all EXPLAIN tables

   1  Upgrade EXPLAIN tables to DB2 Version 7

   2  Drop Existing EXPLAIN tables and recreate them in DB2 Version 7
```

Upgrade Plan Table

Select 1 from the Alter Plan Table screen. The Upgrade Plan Table screen displays (Figure 16-6 on page 16-4).

The Upgrade Plan Table screen shows all the changes to be done to the plan table you are upgrading, based on the version you request. If you are upgrading from version 3 to version 5, the screen shows version 4 and version 5 changes. If you are upgrading from version 3 to version 6, the screen shows version 4, version 5 and version 6 changes. Figure 16-6 on page 16-4 shows an example of upgrading from version 3 to version 7.

Any changes made to the plan table also are made to the F2 plan table.

You cannot change anything on this screen - all fields are protected and for information only.

Figure 16-6. Upgrade Plan Table Screen

```
File-AID for DB2 ----------- Upgrade Plan Table ------------------- SSID: D701
COMMAND ===>
                                                          Sqlid: USERID1
  Alter PLAN_TABLE                                       to V7
    Add Columns:
       SORTC_PGROUP_ID    SMALLINT                        (V4)
       SORTN_PGROUP_ID    SMALLINT                        (V4)
       PARALLELISM_MODE   CHAR(1)                         (V4)
       MERGE_JOIN_COLS    SMALLINT                        (V4)
       CORRELATION_NAME   CHAR(18)                        (V4)
       PAGE_RANGE         CHAR(1)      NOT NULL WITH DEFAULT  (V4)
       JOIN_TYPE          CHAR(1)      NOT NULL WITH DEFAULT  (V4)
       GROUP_MEMBER       CHAR(8)      NOT NULL WITH DEFAULT  (V4)
       IBM_SERVICE_DATA   VARCHAR(254) NOT NULL WITH DEFAULT  (V4)
       WHEN_OPTIMIZE      CHAR(1)      NOT NULL WITH DEFAULT  (V5)
       QBLOCK_TYPE        CHAR(6)      NOT NULL WITH DEFAULT  (V5)
       BIND_TIME          TIMESTAMP    NOT NULL WITH DEFAULT  (V5)
       OPTHINT            CHAR(8)      NOT NULL WITH DEFAULT  (V6)
       HINT_USED          CHAR(8)      NOT NULL WITH DEFAULT  (V6)
       PRIMARY_ACCESSTYPE CHAR(1)      NOT NULL WITH DEFAULT  (V6)
       PARENT_QBLOCK      SMALLINT     NOT NULL WITH DEFAULT  (V7)
       TABLE_TYPE         CHAR(1)                         (V7)

Press ENTER to continue or END to cancel.
```

Drop and Recreate Plan Table and F2 Plan Table

Select 2 to drop or recreate a plan table and F2 plan table. Enter the database name and, optionally, a tablespace name.

3, Delete Rows from Explain Table

Select option 3 to perform maintenance on the plan, statement and function tables. The Explain Table(s) Maintenance Specification screen displays. On this screen you can delete obsolete plan, statement, and function table information. This function should be performed frequently when File-AID *for DB2* Explains are used extensively.

Performance of SQL and Plan analysis features are negatively affected when the plan table and alternate table become too large. In addition, this function's performance is negatively affected when you delete on a row by row basis; that is, when you enter YES in the Preview Rows Prior to Delete field.

Figure 16-7. Explain Table(s) Maintenance Specification Screen

```
File-AID for DB2 --- Explain Table(s) Maintenance Specification---  SSID: D701
COMMAND ===>

  Explain_Table AUTHID ===> FDDBA       Authorization ID of Explain Tables

  Specify Processing Options:
    Preview rows prior to delete ===> YES   (Yes or No)
    Explain_Table rows to delete ===> FD    (Any or FD) * See note below *

  Specify the range of Explain dates/times for delete:
    Starting Date ===> 2004-05-14      Starting Time ===> 00.00.01
    Ending Date   ===> 2004-05-14      Ending Time   ===> 15.26.31

  Note: This function will delete rows from the DB2 AUTHID.PLAN_TABLE,
        AUTHID.F2PLAN_TABLE, AUTHID.STATEMNT_TABLE and AUTHID.FUNCTION_TABLE.
        The F2PLAN_TABLE is only used by File-AID for DB2. Rows that
        Were created within the date/time range specified will be deleted.

        If you specify FD for Rows to delete, only Explain Table rows created
        Thru File-AID for DB2 will be deleted.

        If you specify ANY for Rows to delete, the deleted rows may have been
        Created outside of File-AID for DB2.
```

Explain_Table_AUTHID

Enter the creator ID or authorization ID of the selected table(s).

Preview Rows Prior to Delete

Specify whether the SQL Explain Tables Row Deletion screen displays.

YES SQL Explain Tables Row Deletion screen displays, which lists those rows created by File-AID *for DB2* and meet the specified criteria. From this screen, you can select which rows to delete, or which rows to provide more detail. The plan, statement and function table rows generated outside of File-AID *for DB2* are not listed or deleted by this function.

NO SQL Explain Tables Row Deletion screen is not displayed. The deletion process occurs immediately upon pressing Enter when NO is specified. All rows which qualify based on the criteria specified are deleted.

Explain_Table rows to delete

Enter **ANY** or **FD**. FD specifies to delete Explain Table rows that were created with File-AID *for DB2*. ANY specifies to delete Explain Table rows created within and outside of File-AID *for DB2*. Deletes rows from the DB2 AUTHID.PLAN_TABLE, AUTHID.F2PLAN_TABLE, AUTHID.STATEMNT_TABLE and AUTHID. FUNCTION_TABLE.

Starting Date/Ending Date

Enter the range of dates for all Explain Table rows to be deleted. Only those within the date range will be deleted. This is a required field. It contains the current system date as a default.

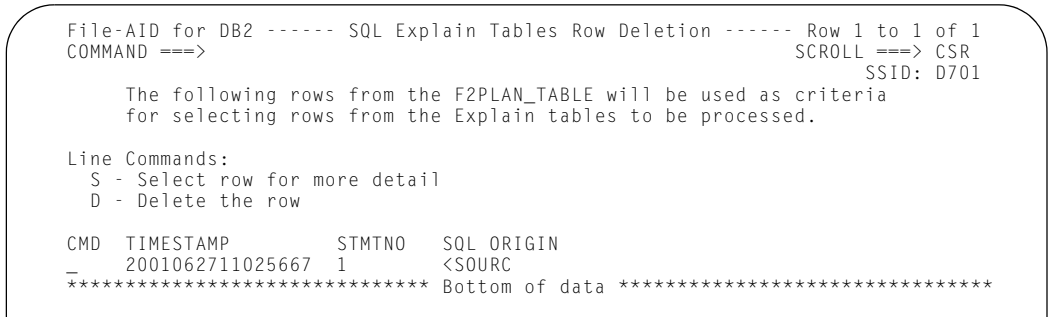
Starting Time/Ending Time

Enter the range of time for all Explain Table rows to be deleted. Only those within the time range will be deleted. This is a required field. It contains a time one second past midnight as a default.

Row Deletion

Enter **YES** on the Preview Rows Prior to Delete field on the Explain Table(s) Maintenance Specification screen. The SQL Explain Tables Row Deletion screen displays. On this screen you can delete individual rows using the D line command. Enter an **S** in the CMD field to display the Maintenance Detail screen.

Figure 16-8. SQL Explain Tables Row Deletion Screen



CMD

Specify the line command to be performed on the selected row.

S Displays the Maintenance Detail screen. Different screens appear depending on whether you are selecting rows from the plan table, function table, or statement table. Maintenance Detail screens allow you to review the contents of the row prior to its deletion.

D Deletes a row from the plan table. The message **Delete Pending** displays in the upper right corner of the screen. The D remains in the CMD field until END is pressed, at which time the row is deleted from the plan table. To remove a D, space over it.

Maintenance Detail

The Maintenance Detail screen is displayed when you select a statement on the SQL Plan Table Row Deletion screen. This screen allows you to review individual rows prior to deleting them. The SQL statement displays in the upper section of the screen and each referenced table is listed in the lower portion (Plan Table only). The list of referenced table(s) can be scrolled (DOWN/UP) if needed. Enter the **END** command to return to the Row Deletion screen.

Figure 16-9. Maintenance Detail Screen

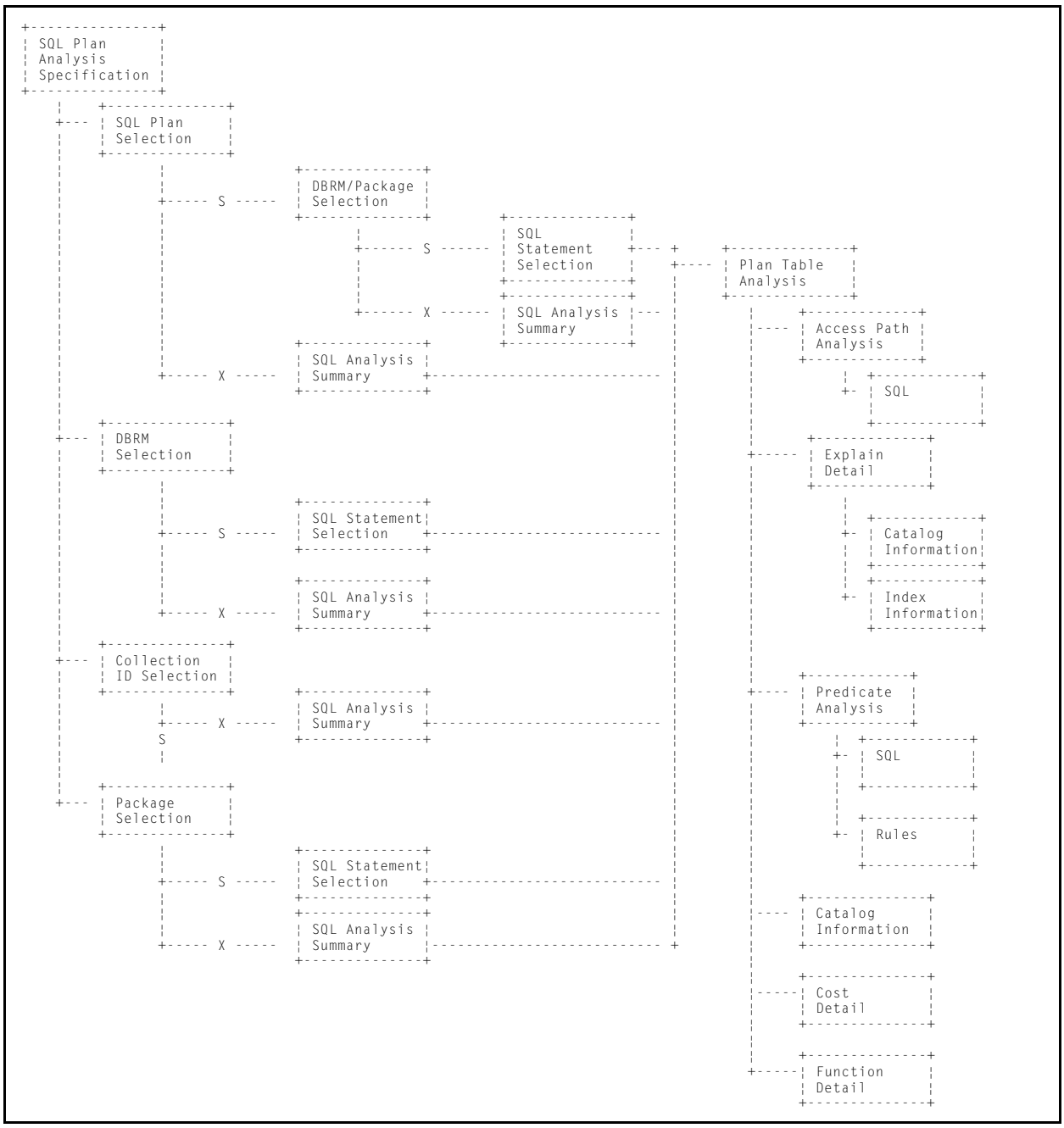
```
File-AID for DB2 -------- Maintenance Detail  ---------------- Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> CSR
                                                                    SSID: DSN
SQL Statement:

DECLARE PLAN31_CSR2 CURSOR FOR
SELECT NAME , CREATOR , BINDDATE , VALIDATE , ISOLATION , VALID ,
OPERATIVE , BINDTIME , PLSIZE , IBMREQD , AVGSIZE , ACQUIRE , RELEASE ,
EXPLAN , BOUNDBY , QUALIFIER , CACHESIZE , PLENTRIES , DEFERPREP ,
CURRENTSERVER , SYSENTRIES , DEGREE , SQLRULES , DISCONNECT
FROM SYSIBM . SYSPLAN
WHERE CREATOR = :SQLCCREA
AND NAME LIKE :SQLCPPLN FOR FETCH ONLY


  Qblk Plan        Access Match Index  TS  SortN SortC Table  Pre  Col  Mix
   No   No  Method  Type  Cols  Only  Lock UJOG  UJOG   No   Fetch Eval Seq
    1    1     0      I     1     N     IS NNNN  NNNN    1               0
  Access  Access     Join    Join       Table: SYSIBM.SYSPLAN
  Degree  Pgroup ID  Degree  Pgroup ID  Index: SYSIBM.DSNPPH01
  <NULL>   <NULL>    <NULL>   <NULL>
```

Option 1, SQL Plan Analysis

Select option 1 from the SQL Analysis and Development menu (Figure 16-2 on page 16-2). The SQL Plan Analysis Specification screen displays. On this screen you can explain a plan, DBRM, collection ID, or package and obtain rules-based analysis in plain language. An SQL predicate analysis is also available.

The flow chart in Figure 16-10 shows you how to navigate through the screens when processing the analysis in online mode. See “Explaining Objects in Batch” on page 16-36 for information on batch processing.

Figure 16-10. SQL Plan Analysis Flow Chart — Online Processing Mode



SQL Plan Analysis Specification

The SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9) displays when you select option **1** from the SQL Analysis and Development menu. On this screen you can select an object type to analyze, specify batch or online processing mode, and specify the default qualifier used for unqualified tables and views.

Figure 16-11. SQL Plan Analysis Specification Screen

```
File-AID for DB2 ------- SQL Plan Analysis Specification ----------
COMMAND  ===>                                                 SSID: 701

 Select One Object Type to be Analyzed:
  s Plan
  _ DBRM
  _ Collection ID
  _ Package

 Specify Selection Criteria:
   Object Name  ===> SQLDEMO            (* or pattern for selection list)
   Use Filters  ===> NO                 (Yes or No, Not valid for Collection ID)

 Specify Processing Option:

   Processing Mode            ===> 1      (1 = Online, 2 = Batch)
   Host variable Substitution ===> 1      (1 = Literal, 2 = Parameter Marker)


 Specify Object Qualifier Options:
   Qualify by                 ===> 1      (1 = Current SQLID,
                                           2 = Bind Qualifier,
                                           3 = Default Qualifier)
   Default Qualifier          ===> TSOID01
```

Select One Object Type to be Analyzed

Enter **S** next to the object type to be analyzed. Only one object type can be selected at a time.

Plan Displays the SQL Plan Selection screen shown in Figure 16-13 on page 16-12.

DBRM Displays the DBRM Selection screen shown in Figure 16-14 on page 16-13.

Collection ID Displays the Collection ID Selection screen shown in Figure 16-15 on page 16-14.

Package Displays the Package Selection screen shown in Figure 16-16 on page 16-15.

Specify Selection Criteria

Object Name Enter the name of the object to be analyzed. An asterisk (*****) can be entered. If a single asterisk is entered, all objects of the type selected are displayed. An asterisk can also be used in conjunction with a partial object name.

Use Filters Enter YES to use a wild card character to specify a creator and filter the search. Figure 16-12 on page 16-11 displays. This field is ignored if you have selected a type of Collection ID.

Specify Processing Option

Processing Mode

Select the mode in which the analysis is to be performed.

1 Performs the analysis in online mode. Requested information displays on the screen.

2 Performs the analysis in batch mode. JCL is generated and submitted to produce a report of the requested information.

Host Variable Substitution

This is where you specify the type of host variable substitution to use for an explain:

1 Use literals generated by File-AID *for DB2* by using data type information obtained from the DB2 catalog. The appropriate data type value will be substituted for a host variable.

2 Use parameter markers. A question mark will be substituted for a host variable where allowed by DB2, otherwise the product will generate a literal.

For example: You have a Plan or Package containing the following SQL statement, where IXNAME is defined as VARCHAR(18), and IXCREATOR is defined as CHAR(8):

```
SELECT COUNT ( * )
INTO :SQLINPAR
FROM SYSIBM . SYSINDEXPART
WHERE IXNAME = :SQLINAME
AND IXCREATOR = :SQLICREA
```

If you enter **1** (for Literal), then the following SQL is generated:

```
EXPLAIN PLAN SET QUERYNO = 3519 FOR SELECT COUNT(*)   FROM
SYSIBM.SYSINDEXPART
WHERE IXNAME = '                  ' AND IXCREATOR = '        '
```

If you enter **2** (for Parameter Marker), then the following SQL is generated:

```
EXPLAIN PLAN SET QUERYNO = 3519 FOR SELECT COUNT(*)   FROM
SYSIBM.SYSINDEXPART
WHERE IXNAME = ? AND IXCREATOR = ?
```

Specify Object Qualifier Option

Qualify By

Specify which qualifier option to use for the specified table name.

1 (Default) Current SQLID. Enter 1 to use the SQLID as specified with the primary command "SQLID" on page 3-55 or in the "User Parms Menu" on page 5-1.

2 Bind Qualifier. Enter **2** to use the qualifier that was specified when the plan was bound.

3 Default Qualifier. Enter **3** to use the qualifier that is shown in the Default Qualifier field.

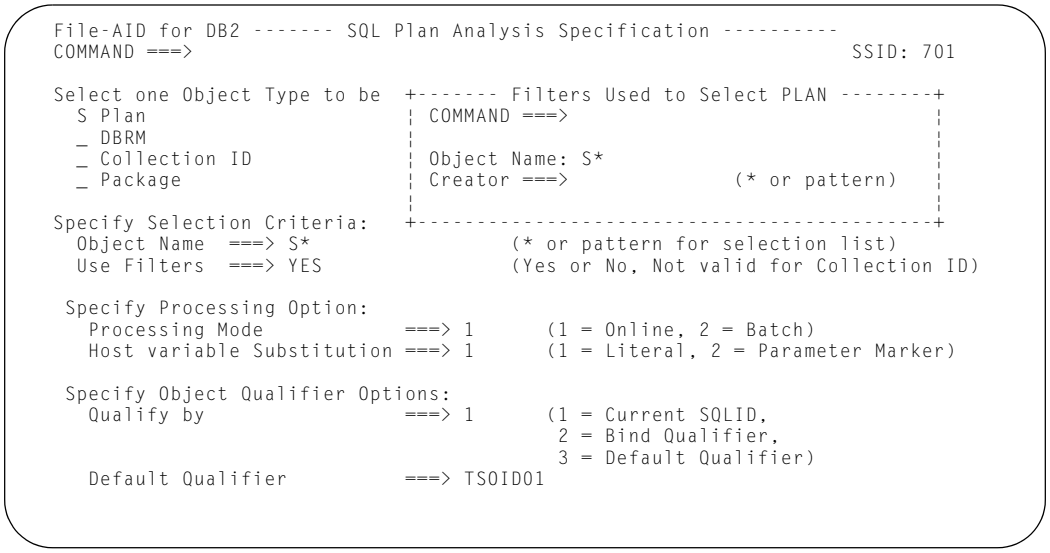
Default Qualifier

This is where you specify the default qualifier for Option 3 in the Qualify By field.

Use Filters Window

The Filters Used to Select window (Figure 16-12) displays when you enter YES in the Use Filters field on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9).

Figure 16-12. Filters Used to Select Object Window



Creator

Specify a Creator to use to filter the search for the object. A wild card character can be entered. See "Wildcard Characters" on page 2-9. If a single asterisk is entered, all creators of the object selected are displayed. An asterisk can also be used in conjunction with a partial creator name.

SQL Plan Selection

The SQL Plan Selection screen (Figure 16-13) displays when a Plan object type is selected on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9). This screen is only displayed when the catalog contains more than one plan matching the specified object name. Otherwise, the DBRM/Package Selection screen (Figure 16-17 on page 16-16) displays. On this screen you can select a plan to be analyzed or explained.

Figure 16-13. SQL Plan Selection

```
File-AID for DB2 ----------- SQL Plan Selection -----------  ROW 1 TO 14 OF 14
COMMAND ===>                                                 SCROLL ===> CSR
                                                                   SSID: DSN

Line Commands:
  S - Select Plan for more detail
  X - Explain entire plan

CMD   Plan Name   Creator  Bound By   Bind Date   Bind Time
---   ---------   -------  --------   ---------   ---------
 _    SQLP2000    USERID1   USERID1    920227     15213063
 _    SQLP1160    USERID1   USERID1    920313     07213183
 _    SQLP2PKG    USERID2   USERID2    920331     11061644
 _    SQLP3PKG    USERID2   USERID2    920331     17141217
 _    SQLSAMP     USERID2   USERID2    920417     16472267
 _    SQLROBIN    USERID2   USERID2    920420     10391270
 _    SQLTEST     USERID2   USERID2    920421     11191392
 _    SQLP1302    USERID2   USERID2    920424     16492837
 _    SQLLONG     USERID2   USERID2    920619     17142798
 _    SQLP2010    USERID1   USERID1    920729     12243255
 _    SQLBIG      USERID3   USERID3    920916     09585582
 _    SQL407      USERID3   USERID3    920930     14490213
 _    SQLSAMP1    USERID2   USERID2    920923     14314954
 ******************************* BOTTOM OF DATA ********************************
```

CMD

Specify the line command to be performed on the selected plan.

S Displays detailed DBRM/Package information for the selected plan. The DBRM/Package Selection screen (Figure 16-17 on page 16-16) displays.

X Performs an Explain of the entire plan.

If you entered **1** for online processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the SQL Analysis Summary screen (Figure 16-19 on page 16-18) displays.

If you entered **2** for batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the Batch Plan Analysis Reports window (Figure 16-46 on page 16-36) displays. You may explain more than one plan by entering multiple X line commands. See "Explaining Objects in Batch" on page 16-36 for more information.

Note: To explain the plan, it must contain one of the following SQL statements: DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE. If the plan cannot be explained, the error message **NONE EXPLAINABLE** displays.

Sorting Plan Entries

The SORT primary command can be used to change the order in which the plan entries are displayed. Multiple column names are not accepted. The selected columns are sorted in ascending sequence. The valid column names that can be used with the SORT command are listed below. Abbreviations are noted within parentheses.

- Name (NA)
- Creator (CR)
- Boundby (BO)
- Date (DA)
- Time (TI)

For example, to sort the display by the bind time, enter the command **SORT TI**.

DBRM Selection

The DBRM Selection screen (Figure 16-14) displays when a DBRM object type is selected on the SQL Plan Analysis Specification screen. DBRMs are listed in the same order as they are retrieved from the DB2 catalog.

Figure 16-14. DBRM Selection Screen

```
File-AID for DB2 ----------- DBRM Selection ----------------  ROW 1 TO 4 OF 89
COMMAND ===>                                                 SCROLL ===> CSR
                                                                   SSID: DSN

Line Commands:
  S - Select DBRM for more detail
  X - Explain entire DBRM

CMD
_    DBRM Name       :     OSSQCOMT
     Plan Name       :     SQLP2000
     Precompile Time :     67-07-71-09.42.20.69

_    DBRM Name       :     DB2SQLXX
     Plan Name       :     SQLP2000
     Precompile Time :     67-07-71-09.42.20.69

_    DBRM Name       :     DB2SQLAA
     Plan Name       :     SQLP2000
     Precompile Time :     67-07-71-09.42.20.69

_    DBRM Name       :     DB2SQLBB
     Plan Name       :     SQLP2000
```

CMD

Specify the line command to be performed on the selected DBRM.

S Displayed detailed information for the selected DBRM. The SQL Statement Selection screen shown in Figure 16-18 on page 16-17 displays.

X Performs an Explain of the entire DBRM.

If you entered **1** for online processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the SQL Source Analysis screen shown in Figure 16-19 on page 16-18 displays.

If you entered **2** for batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the Batch Plan Analysis Reports window shown in Figure 16-46 on page 16-36 displays. See "Explaining Objects in Batch" on page 16-36 for more information.

To explain the DBRM, it must contain one of the following SQL statements: DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE. If the DBRM cannot be explained, the error message **NONE EXPLAINABLE** displays.

Sorting DBRM Entries

The SORT primary command can be used to change the order in which DBRM entries are displayed. Multiple column names are not accepted. The selected columns are sorted in ascending sequence. The valid column names that can be used with the SORT command are listed below. Abbreviations are noted within parentheses.

- Name (NA)
- Plan (PL)
- Time (TI)

For example, to sort the display by DBRM name, enter the command **SORT NA**.

Collection ID Selection

The Collection ID Selection screen (Figure 16-15) displays when a Collection ID object type is selected on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9).

Figure 16-15. Collection ID Selection Screen

```
File-AID for DB2 --------- Collection ID Selection ----------
 ROW 1 TO 16 OF 20
COMMAND ===>                                          SCROLL ===> CSR
                                                           SSID: DSN

Line Commands:
  S - Select Collection ID for more detail
  X - Explain entire Collection ID

CMD  Collection ID
_    AADB2COL
_    AASQLCOLL
_    AASQLPAK
_    AASQLPKG
_    CPKREMOTE
_    DBXPKG
_    DC3O2PAK
_    DSNESPCS
_    DSNESPRR
_    DSNQCATV
_    MIKEPK
_    NORM
_    PLATEST
_    Q
S    SQLPCID1
```

CMD

Specify the line command to be performed on the selected collection ID.

S Displays detailed information for the selected collection ID. A Package Selection screen similar to the one shown in Figure 16-16 on page 16-15 displays. The only difference is that the packages displayed are for a specific collection ID which is noted at the top of the screen.

X Performs an Explain of the entire collection ID.

If you entered **1** for online processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the SQL Analysis Summary screen shown in Figure 16-19 on page 16-18 displays.

If you entered **2** for batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the Batch Plan Analysis Reports window shown in Figure 16-46 on page 16-36 displays. See "Explaining Objects in Batch" on page 16-36 for more information.

Note: To explain the Collection ID, it must contain one of the following SQL statements: DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE. If it cannot be explained, the error message **NONE EXPLAINABLE** displays.

Package Selection

The Package Selection screen (Figure 16-16 on page 16-15) displays when Package is selected on the SQL Plan Analysis Specification screen, or when a Collection ID is selected on the Collection ID Selection screen (Figure 16-15 on page 16-14). This screen lists detailed package information, including collection ID, package name, bind timestamps, and version.

CMD

Specify the line command to be performed on the selected package.

S Displays specific package information. The SQL Statement Selection screen (Figure 16-18 on page 16-17) displays. If no detailed information is available for the package, the error message **NONE EXPLAINABLE** displays.

X Performs an Explain of the entire package.

If you entered **1** for online processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the SQL Analysis Summary screen (Figure 16-19 on page 16-18) displays.

If you entered **2** for batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the Batch Plan Analysis Reports window (Figure 16-46 on page 16-36) displays. See "Explaining Objects in Batch" on page 16-36 for more information.

Note: To explain the package, it must contain one of the following SQL statements: DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE. If it cannot be explained, the error message **NONE EXPLAINABLE** displays.

Figure 16-16. Package Selection Screen

```
File-AID for DB2 ------------ Package Selection -------------- Row 1 to 3 of 6
COMMAND ===>                                                 SCROLL ===> CSR
                                                                   SSID: D610

Line Commands:
  S - Select Package for more detail
  X - Explain entire Package

Cmd
 _   Package Name    : TRIGDEPT
     Bind Timestamp  : 2001-05-07-13.05.32.091432
     Collection ID   : BFHRUSER1
     Version         :

 _   Package Name    : TRIGDEP2
     Bind Timestamp  : 2001-09-26-18.02.29.961110
     Collection ID   : BFHUSER2
     Version         :

 _   Package Name    : TABRTE
     Bind Timestamp  : 2001-09-26-18.25.14.784215
     Collection ID   : BFHUSER1
     Version         :
```

Sorting Package Entries

The SORT primary command can be used to change the order in which package entries are displayed. Multiple column names are not accepted. The selected columns are sorted in ascending sequence. The valid column names that can be used with the SORT command are listed below. Abbreviations are noted within parentheses.

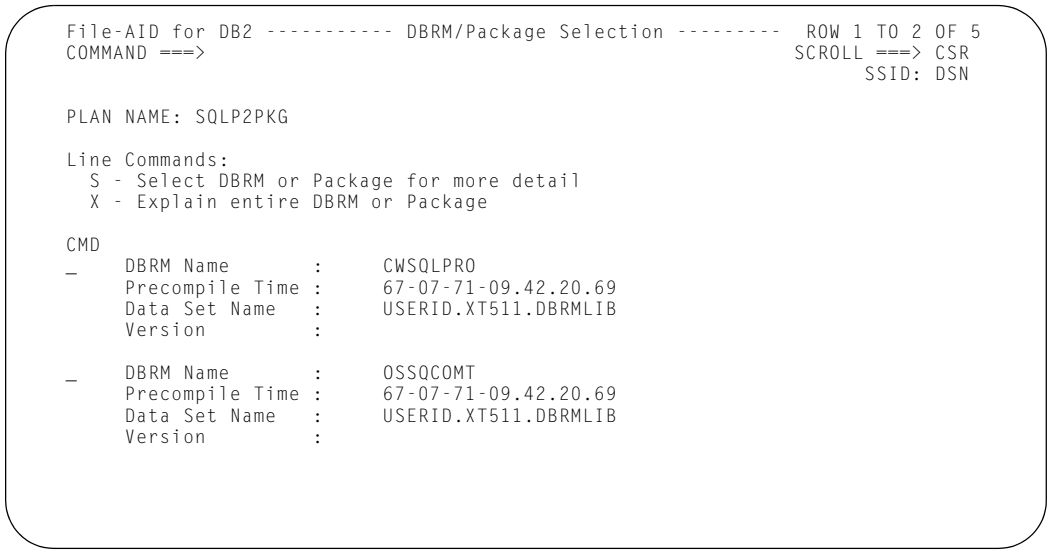
- Name (NA)
- Time (TI)
- Collid (CI)
- Version (VER)

For example, to sort the display by the bind timestamp, enter the command **SORT TI**.

DBRM/Package Selection

The DBRM/Package Selection screen (Figure 16-17 on page 16-16) displays when a plan is selected with the S line command on the SQL Plan Selection screen (Figure 16-13 on page 16-12). This screen displays detailed DBRM or Package information.

Figure 16-17. DBRM/Package Selection Screen



CMD

Specify the line command to be performed on the selected DBRM/Package.

S Displays detailed information for the selected DBRM or package. The SQL Statement Selection screen (Figure 16-18 on page 16-17) displays.

X Performs an Explain of the entire DBRM or package.

If you entered **1** for online processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the SQL Analysis Summary screen (Figure 16-19 on page 16-18) displays.

If you entered **2** for batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9), the Batch Plan Analysis Reports window (Figure 16-46 on page 16-36) displays. See "Explaining Objects in Batch" on page 16-36 for more information.

Note: To explain the DBRM or package, it must contain one of the following SQL statements: DECLARE CURSOR, DELETE, INSERT, SELECT, or UPDATE. If the DBRM or package cannot be explained, the error message **NONE EXPLAINABLE** displays.

Sorting DBRMs or Packages

The SORT primary command can be used to change the order in which the DBRM or package entries are displayed. Multiple column names are not accepted. The selected columns are sorted in ascending sequence. The valid column names that can be used with the SORT command are listed below. Abbreviations are noted within parentheses.

- Name (NA)
- Time (TI)
- Dname (DN)
- Version (VER)

For example, to sort the display by dataset name, enter the command **SORT DN**.

Note: Use the SORT command before using the LOCATE command.

SQL Statement Selection

The SQL Statement Selection screen (Figure 16-18) displays when a DBRM or package is selected with the S line command on the DBRM/Package Selection screen (Figure 16-17 on page 16-16) or the DBRM Selection screen (Figure 16-14 on page 16-13).

Figure 16-18. SQL Statement Selection Screen

```
File-AID for DB2 --------- SQL Statement Selection ---------------- SSID: DSN
COMMAND ===>                                                  SCROLL ===> CSR

 PLAN NAME/COLLID : SQLP2PKG    PRECOMPILE TIMESTAMP: 92-03-33-11.06.16.44
 DBRM/PACKAGE NAME: OSSQLC22

 Line Commands:
   X - Explain the selected SQL statement

CMD  STMT  SQL TEXT
_     488   DECLARE ALIAS_CURSOR1 CURSOR FOR
            SELECT NAME , CREATOR , TYPE , LOCATION , TBCREATOR , TBNAME
            FROM SYSIBM . SYSTABLES
            WHERE CREATOR = : SQLCCREA
            AND NAME = : SQLCNAME
            AND TYPE = 'A'

_     497   DECLARE ALIAS_CURSOR2 CURSOR FOR
            SELECT NAME , CREATOR , TYPE , LOCATION , TBCREATOR , TBNAME
            FROM SYSIBM . SYSTABLES
            WHERE CREATOR = : SQLCCREA
            AND NAME LIKE : SQLCNAME
            AND TYPE = 'A'
```

CMD

Enter **X** to perform an Explain of the selected SQL statement. The Plan Table Analysis screen (Figure 16-20 on page 16-19) displays.

SQL Analysis Summary

The SQL Analysis Summary screen provides a quick reference of your SQL statements and enables you to easily identify the SQL you want to analyze. It displays the DB2 cost threshold, enabling you to identify SQL requiring performance improvements. From this screen, you can select a DBRM/Package for more analysis.

The screen displays when you request an online explain by entering the X line command on the screens listed below. It is not displayed if there is only one SQL statement.

- SQL Plan Selection screen (Figure 16-13 on page 16-12).

- DBRM Selection screen (Figure 16-14 on page 16-13).
- Collection ID Selection screen (Figure 16-15 on page 16-14).
- Package Selection screen (Figure 16-16 on page 16-15).
- DBRM/Package Selection screen (Figure 16-17 on page 16-16).

This screen is also displayed when you request an online report from the SQL Explain Report Specification screen (Figure 16-50 on page 16-41).

Note: In the event that File-AID *for DB2* detects a syntax error, the original unmodified SQL statement is passed to DB2 for an explain. No literal substitution or object qualification is done in this case. This can cause different explain results.

Figure 16-19. SQL Analysis Summary

```
File-AID for DB2 -------- SQL Analysis Summary ------------- ROW 1 TO 13 OF 84
COMMAND ===>                                                  SCROLL ===> CSR

 Plan Name: DB2XPERT         DB2 Cost Threshold: 1200  (* = Threshold exceeded)

 Enter 'S' to select a SQL statement for more analysis

 CMD  DBRM/Package  Statement   Type   DB2 Cost    Access Type
 ---  ------------  ---------  ------  --------    ----------------------
  _   DB2SQLXX         1698     SELECT      1250 *  Non-Matching Index Scan
  _   DB2SQLXX         3198     SELECT      1225 *  TableSpace Scan
  _   DB2SQLXX         2298     SELECT      1225 *  Non-Matching Index Scan
  _   DB2SQLXX         8638     SELECT      1217 *  Non-Matching Index Scan
  _   DB2SQLXX         3498     SELECT      1210 *  Non-Matching Index Scan
  _   DB2SQLXX         2658     SELECT      1208 *  Non-Matching Index Scan
  _   DB2SQLXX         4698     SELECT      1163    Non-Matching Index Scan
  _   DB2SQLXX         1698     SELECT      1150    Non-Matching Index Scan
  _   DB2SQLXX         2698     SELECT      1140    Non-Matching Index Scan
  _   DB2SQLXX         3698     SELECT      1135    Non-Matching Index Scan
  _   DB2SQLXX         5698     SELECT      1120    Non-Matching Index Scan
  _   DB2SQLXX         6698     SELECT      1080    Non-Matching Index Scan
  _   DB2SQLXX         7698     SELECT      1050    Non-Matching Index Scan
  _   DB2SQLXX         5698     SELECT      1030    Non-Matching Index Scan
```

CMD

Enter **S** to select a SQL statement for more analysis. The Plan Table Analysis screen (Figure 16-20 on page 16-19) displays.

DBRM/Package

Name of the DBRM/Package for the plan.

Statement

Statement number within the plan or package.

Type

Type of SQL statement.

DB2 Cost

If the cost exceeds the threshold for the plan, an asterisk (*****) displays next to this field. The value of the DB2 cost threshold is determined at the time File-AID *for DB2* is installed.

Access Type

Access type.

Sorting DBRM/Package Entries

The SORT primary command can be used to change the order in which the DBRM/Package entries are displayed. Multiple column names are not accepted. The selected columns are sorted in ascending sequence. The valid column names that can be used with the SORT command are listed below. Abbreviations are noted within parentheses.

- DBRM/Package (DBRM)
- Statement (ST)
- Type (TYPE)
- DB2 Cost (COST)
- Access Type (AC)

For example, to sort the display by statement, enter the command **SORT ST**.

Plan Table Analysis

The Plan Table Analysis screen displays when you:

- Select a DBRM/Package on the SQL Analysis Summary screen (Figure 16-19 on page 16-18).
- Select a statement on the SQL Statement Selection screen (Figure 16-18 on page 16-17).
- Issue the XP line command in SQL Source Development and Analysis or select option 3 from the Host Variable Resolution screen. See "Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44 for more information.

It displays the entire SQL statement and enables you to receive an access path analysis, explain detail, a predicate analysis, and catalog information for the SQL statement. Scroll to the previous or next SQL statements with the PREV and NEXT primary commands.

Figure 16-20. Plan Table Analysis Screen

```
File-AID for DB2 - Analysis for Plan :MJMPROB2 ------------------ Enter option
OPTION ===>                                                  SCROLL ===> PAGE
                                                                   SSID: D610
    1 - Access Path Analysis         4 - Object Information
    2 - Explain Detail               5 - Cost Detail
    3 - Predicate Analysis           6 - Function Detail

    DBRM/Package  Statement   Type   DB2 Cost       Access Type
    ------------  ---------  ------  ---------      ---------------------------
    OSSQLCAT      6037       SELECT  7              Matching Index Scan

SQL Statement: 1 OF 1

    DECLARE COLAUTH_CURSOR CURSOR FOR
    SELECT GRANTEETYPE , COLNAME
    FROM SYSIBM . SYSCOLAUTH
    WHERE :SQLGTABNAME = TNAME
    AND :SQLGTABCREA = CREATOR
    AND :SQLGTABTIME = TIMESTAMP
****************************** Bottom of data ********************************
```

OPTION

Enter the desired option.

1 Displays an Access Path Analysis for the plan (Figure 16-21 on page 16-20).

2 Displays the Explain Detail for the plan (Figure 16-25 on page 16-24).

3 Displays a SQL Predicate Analysis for the plan (Figure 16-27 on page 16-26).

4 Displays catalog information for the package (Figure 16-32 on page 16-29).

5 Displays Cost Detail information for the package (Figure 16-44 on page 16-35).

6 Displays Function Detail information for the package (Figure 16-45 on page 16-36).

Access Path Analysis

The Access Path Analysis screen displays when you select option 1 from the Plan Table Analysis screen (Figure 16-21 on page 16-20). It provides a detailed analysis of how the SQL statement is analyzed and executed by DB2. For example, the sequence and manner in which various tables are accessed, the locking method that is used, and a comprehensive description of both consequences and recommendations.

Figure 16-21. Access Path Analysis Screen

```
 File-AID for DB2 ------- PLAN_TABLE Analysis Report for: <SOURCE>
 COMMAND ===>                                              SCROLL ===> CSR

 Use SQL to View/Save/Re-explain the SQL Source

 DB2 VERSION: 7.1
 QUERY NO   : 1

 Conditions
   Query Block No:        1
   Plan No:               1

   Query Block Type:      SELECT

   01 - Table SYSIBM.SYSSTMT will be the first table accessed
        in this Query Block.
   06 - Table SYSIBM.SYSSTMT will be accessed
        through a tablespace scan of
        DSNDB06.SYSPLAN.
   07 - The tablespace contains 2160 pages.
   08 - The tablespace contains multiple tables and is not segmented.
   33 - A sequential PREFETCH will be used to read a series of
        consecutive pages.

Consequences
  03 - A scan of a tablespace of this size will significantly add to:
       - the processing cost of the query because the
         number of rows scanned is a contributing factor,
       - the I/O cost of the query because the number
         of data pages read is also a factor.
  04 - In a tablespace that is not segmented, DB2 reads every row in
       tablespace DSNDB06.SYSPLAN.  The total time required to
       execute the query depends on the number of pages to be read
       in the tablespace.  This will include storage for other
       tables, space left free for expansion, and unreclaimed
       storage after delete operations.

Recommendations
  07 - Table SYSIBM.SYSSTMT has no available indexes. Consider
       creating an index for frequently used columns in the table
       and incorporating the index into the query.
  08 - Consider segmenting tablespace DSNDB06.SYSPLAN,
       so that DB2 will scan only segments containing
       rows of table SYSIBM.SYSSTMT.
F5=PRINTP
```

The report contains header information and a Conditions, Consequences, and Recommendations section for each table and/or index used by the SQL statement. The text in these sections can be modified to meet your site's needs. See the *File-AID for DB2 Installation Guide* for information on modifying the text.

Header Information

The top of the report contains header information, including the DB2 Version and the Query Number. The DB2 Version is the DB2 release for the subsystem upon which the query is executed. The Query Number identifies the EXPLAINed statement. File-AID *for DB2* assigns the program statement number as the query number unless the statement has a QUERYNO clause specified. The explain command may generate one or more rows in the PLAN_TABLE. If such is the case, all rows with the same query number refer to the same SQL statement. The PLAN_TABLE only contains the QUERYNO, not the statement number.

Conditions

The Conditions section of the report identifies the Query Block Number and Plan Number for the statement. The Query Block Number (QBlock Number) identifies the position of the query within the SQL statement. For example:

```
SELECT * FROM ABC WHERE COL1 IN

SELECT COL1 FROM DEF WHERE COL2 = 0
```

The outer SELECT will have a Query Block Number of 1 and the second SELECT a Query Block Number of 2.

The Plan Number identifies the sequence in which the steps within the Query Block Number specified are processed.

Note: The following data can also display in the Conditions section, depending on the type of statement being explained: Query Block Type, Parent Query Block Type, Method of Join, and Join Type.

The Conditions section also contains information on any locking that may be performed as well as any index utilization.

Consequences

The Consequences section of the report outlines the performance ramifications of the SQL and the effects of object volume. The analysis is performed using actual catalog statistics in effect at the time the analysis operation was performed. To maximize the potential of the analysis operation, use objects with characteristics that closely resemble the objects that the SQL will ultimately be used against.

Recommendations

The Recommendations section of the report offers suggestions for improving the performance of the SQL statement. The recommendations are based on actual, not potential, object statistics. You may need to scroll with the DOWN primary command to view this section.

Printing the Access Path Analysis

Use PRINTP (using the associated PF key) to print the Access Path Analysis to a printer or dataset. When you exit the screen, the SQL Analysis Output Destination Specification screen (Figure 16-22) displays. Select a destination with the S line command. Enter the required fields for the selected destination. The default print options and job statement are set by selecting option 0, Defaults and User Parameters, from the Primary Option Menu. See "Option 7, Print Options" on page 5-10 and "Option 3, Batch Options" on page 5-6.

Note: This screen is also displayed when printing Explain Detail and Predicate Analysis reports.

Figure 16-22. SQL Analysis Output Destination Specification Screen

```
File-AID for DB2 -------- SQL Analysis Output Destination Specification -------
COMMAND ===>

Select Printer Destination or Dataset then press ENTER

  _  TO PRINTER

  Printer Destination  ===>
  Sysout Class         ===>
  Lines Per Page       ===>

  _  TO DATASET

  Report Dataset Name  ===>
  Volume Serial        ===>
```

Executing the SQL Command

When you execute the SQL primary command, a temporary dataset containing the SQL source is created. An ISPF/PDF edit session is initiated enabling you to edit the SQL. If you want to save the dataset, you must use the ISPF/PDF CREATE or COPY command before exiting the window.

Figure 16-23. SQL Source — After Executing the SQL Command

```
File-AID for DB2 -- Access Path Analysis for: DB2AAAA ----------- SSID: DSN
+-------------------------------------------------------------------------------+
¦ EDIT ---- SYS94252.T153243.RA000.PFHGRH0.R0034117 --------- COLUMNS 001 068 ¦
¦ COMMAND ===>                                              SCROLL ===> CSR  ¦
¦ ****** ************************* TOP OF DATA ***************************** ¦
¦ =NOTE= SQL may be modified. If the SQL is modified it will automatically    ¦
¦ =NOTE= be re-explained when you exit this window. The Plan, Package or      ¦
¦ =NOTE= DBRM will not be updated with the modified SQL.                      ¦
¦ 000001  DECLARE DBAUTH_CURSOR CURSOR FOR                                    ¦
¦ 000002  SELECT GRANTOR , GRANTEE , NAME , GRANTEETYPE ,CREATETABAUTH ,      ¦
¦ 000003    DBCTRLAUTH , DBMAINTAUTH , DISPLAYDBAUTH , DROPAUTH ,             ¦
¦ 000004    LOADAUTH , REORGAUTH , RECOVERDBAUTH , REPAIRAUTH , STARTDBAUTH , ¦
¦ 000005    STATSAUTH , STOPAUTH                                              ¦
¦ 000006  FROM SYSIBM . SYSDBAUTH                                             ¦
¦ 000007  WHERE COLUMN1 > :VALUE1                                             ¦
¦ 000008  AND COLUMN1 < :VALUE2;                                              ¦
¦ ****** ********************** BOTTOM OF DATA ***************************** ¦
¦                                                                             ¦
¦                                                                             ¦
¦                                                                             ¦
+-------------------------------------------------------------------------------+
       tablespace DSNDB06.SYSPLAN.  The total time required to execute the

F5=PRINTP
```

When you press END to exit the window, the SQL is automatically re-explained and the SQL Analysis Summary screen displays (Figure 16-24).

The modified SQL is inserted directly after the original statement. The same DBRM/Package name and Statement number are used. The new DB2 cost value and access type are displayed. The new row is identified by an asterisk (*) in front of the DBRM/Package name, enabling you to compare the results of the changes immediately.

Note: In the modified SQL, the host variables are always substituted with parameter markers prior to the re-explain. This may affect the resulting DB2 cost.

Figure 16-24. SQL Analysis Summary Screen — After a Re-explain

```
File-AID for DB2 -------- SQL Analysis Summary ------------- ROW 1 TO 13 OF 84
COMMAND ===>                                                 SCROLL ===> CSR


 Plan Name: DB2AAAA          DB2 Cost Threshold: 1200  (* = Threshold exceeded)

 Enter 'S' to select a SQL statement for more analysis

     DBRM/Package  Statement   Type   DB2 Cost    Access Type
     ------------  ---------  ------  --------    ------------------------
 _   DB2SQLXX         1698     SELECT      1250 *  Non-Matching Index Scan
 _   DB2SQLXX         3198     SELECT      1225 *  TableSpace Scan
 _   DB2SQLXX         2298     SELECT      1225 *  Non-Matching Index Scan
 _   DB2SQLXX         8638     SELECT      1217 *  Non-Matching Index Scan
 _   DB2SQLXX         3498     SELECT      1210 *  Non-Matching Index Scan
 _   DB2SQLXX         2658     SELECT      1208 *  Non-Matching Index Scan
 _ * DB2SQLXX         2658     SELECT      1163    Non-Matching Index Scan
 _   DB2SQLXX         1698     SELECT      1150    Non-Matching Index Scan
 _   DB2SQLXX         2698     SELECT      1140    Non-Matching Index Scan
 _   DB2SQLXX         3698     SELECT      1135    Non-Matching Index Scan
 _   DB2SQLXX         5698     SELECT      1120    Non-Matching Index Scan
 _   DB2SQLXX         6698     SELECT      1080    Non-Matching Index Scan
 _   DB2SQLXX         7698     SELECT      1050    Non-Matching Index Scan
 _   DB2SQLXX         5698     SELECT      1030    Non-Matching Index Scan
```

Explain Detail

The Explain Detail screen displays when you select option 2 from the Plan Table Analysis screen (Figure 16-20 on page 16-19). It provides a formatted display of the explain information. You can receive additional table or index information from this screen as well as a detailed explanation of each field. Use PF8 (DOWN) and PF7 (UP) to navigate.

Figure 16-25. Explain Detail Screen

```
F2REP300 for DB2 ----- Explain Detail for:  FD470001              Top of data
 COMMAND ===>

 Use 'S' for additional information                  Plan_Table Row 1 OF 1
  _ Table information for: SYSIBM.SYSCOLAUTH
        Correlation name:
  _ Index information for: PFHJGDO.SYSCOLAUTH_IDX

 TAB to any field and press ENTER for an explanation of that field

 QBlock Type:   SELECT                      Bind time:   2004-06-29 14.04.55.83

 Statement           :   7333   Table Detail ----------------------------
 Query number        :   7333   Table number :   1    Encoding Scheme:
 QBlock number       :   1      Table Type   :   N/A   DBCS CCSID    :
 Plan number         :   1                             SBCS CCSID    :
 Parent QBlock Number:   N/A    CTE reference:         Mixed CCSID   :

 Performance ------------------------------------- JOIN Detail--------------
  ACCESS TYPE        :   I       Method      :   0    JOIN Type       :
  Index Only         :   N                            Merge Join Cols :   NULL
  Match columns      :   2       Prefetch    :        JOIN degree     :   NULL
  Number of index col:   3                            JOIN PGroupID   :   NULL
  Multiple index seq :   0       TS lock mode:   IS
  Primary access type:

 New Table Sort ---------------------------- Composite Table Sort -------------
  SortN Unique    :   N                       SortC Unique        :   N
  SortN Join      :   N                       SortC Join          :   N
  SortN Orderby   :   N                       SortC Orderby       :   N
  SortN Groupby   :   N                       SortC Groupby       :   N

 Concurrency ------------------------------- Processing ----------------------
  Parallelism mode:   NULL                    Page range screening:
  Access degree   :   NULL                    Column function eval:
  Access PGroupID :   NULL
  SortN PGroupID  :   NULL
  SortC PGroupID  :   NULL

 EXPLAIN Execution ------------------------- Optimization ---------------------
  Program Name    :   OSSQLCAT                Hint         :
  Application name:                           Hint Used    :
  Group Member    :                           When optimize:
```

Additional Information

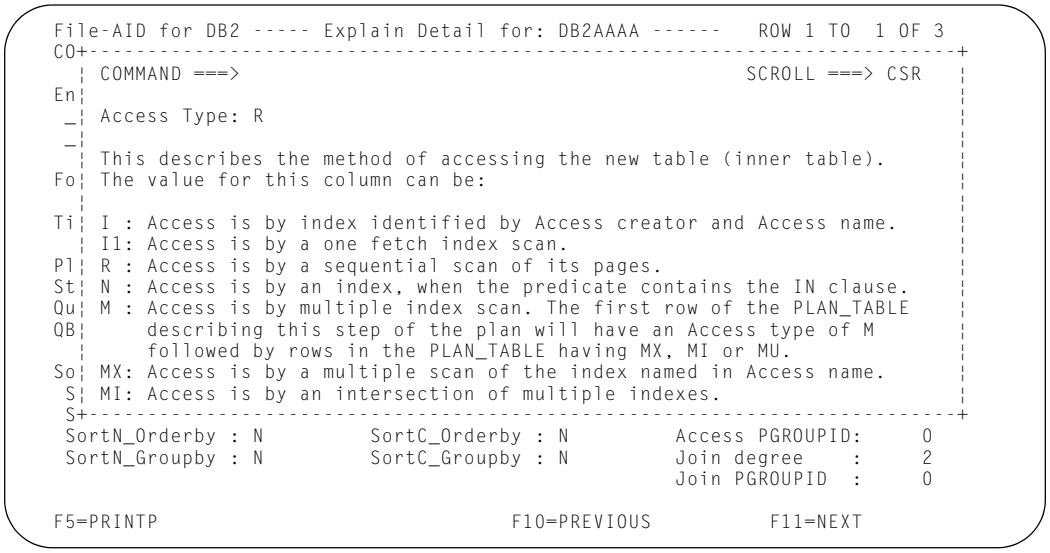
Enter an **S** to select one of the following:

Table Information for	The Table Information screen (Figure 16-32 on page 16-29) displays.
Index Information for	The Index Information screen (Figure 16-38 on page 16-32) displays.

Detailed Field Explanations

The Explain Detail screen is cursor sensitive. Tab to any field and press Enter to receive more information. Figure 16-26 shows detail information for the Access Type field.

Figure 16-26. FIeld Explanation



Printing the Explain Detail

Use PRINTP (using the associated PF key) to print the Explain Detail to a printer or dataset. When you exit the screen, the SQL Analysis Output Destination Specification screen (Figure 16-22 on page 16-22) displays.

Predicate Analysis

The SQL Predicate Analysis screen (Figure 16-27 on page 16-26) displays when you select option 3 from one of the Analysis screens. (See Figure 16-20 on page 16-19 for more information on the Plan Table Analysis screen.) The predicates are analyzed to determine if they are indexable, stage I, or both. Based on the analysis, recommendations for improving the SQL performance are made. A long message is also provided for more detailed information. See Appendix A, “Predicate Analysis Rules Abbreviations” for a complete list of the short and long messages.

Note: The Predicate Analysis function does not yet fully support functionality introduced with DB2 Release 7.

The SQL statements for which there is a recommendation is highlighted in white. Recommendations for performance improvement immediately follow the SQL statement. The recommendations are highlighted and suffixed according to their severity as follows:

- **Mandatory:** highlighted green and suffixed with an M
- **Severe:** highlighted red and suffixed with an S
- **Warning:** highlighted yellow and suffixed with a W
- **Informational:** highlighted white and suffixed with an I

Figure 16-27. SQL Predicate Analysis

```
File-AID for DB2 ----------- SQL Predicate Analysis --------------- SSID: DSN
COMMAND ===>                                                  SCROLL ===> PAGE

Use SQL to View/Save/Re-explain the SQL source

TAB to -- Recommendation and press ENTER for additional information

  SELECT LASTNAME, FIRSTNME, SALARY, HIREDATE
    ( -- RULE0011W:  Use a FOR FETCH if the cursor is retrieval only -- )
    ( -- RULE0014I:  Use the Optimize For n Rows Clause -- )
  FROM TSOID01.EMP
  WHERE WORKDEPT
  IN ('A00', 'B01')
  AND SALARY > 10000
  AND HIREDATE < '1994-01-01'
  ORDER BY LASTNAME, FIRSTNME
    ( -- RULE0013I:  List only necessary columns in Order By clause -- )
```

Additional Recommendation Information

The Predicate Analysis screen is cursor sensitive. Tab to any rule and press Enter to receive more information (Figure 16-28 on page 16-26).

Figure 16-28. SQL Predicate Analysis — Detail Recommendation Information

```
File-AID for DB2 ----------- SQL Predicate Analysis --------------- SSID: D701
COMMAND ===>                                                  SCROLL ===> PAGE

Use SQL to Vie+------------- File-AID for DB2 - RULE0011 -------------------+
              | COMMAND ===>                                SCROLL ===> PAGE  |
TAB to -- Reco|                                                               |
              | Best to use FOR FETCH ONLY if only reading                    |
  DECLARE C2 C|                                                               |
  WITH HOLD FO|      Consider using the FOR FETCH ONLY clause on your         |
  SELECT DEPTN| DECLARE CURSOR statement.  This clause enables DB2's          |
    ( -- RULE0| block fetch processing.                                       |
    ( -- RULE0|                                                               |
    ( -- RULE0| Block fetch is a mechanism where by DB2 retrieves data        |
  FROM DSN8710| before the application requests it.  Block fetch forces       |
  WHERE ADMRDE| the cursor to be "read only".  This means the applica-        |
              | tion cannot issue a DELETE WHERE CURRENT OF or UPDATE         |
              | WHERE CURRENT OF using this cursor, unless the cursor         |
              | is a SENSITIVE STATIC scrollable cursor.                      |
              |                                                               |
              | Note: FOR READ ONLY is comparable to FOR FETCH ONLY.          |
              |                                                               |
              | Revised:  8/19/04                                             |
              +---------------------------------------------------------------+
```

Executing the SQL Command

When you execute the SQL primary command, a temporary dataset containing the SQL source is created (Figure 16-29). An ISPF/PDF edit session is initiated enabling you to edit the SQL. If you want to save the dataset, you must use the ISPF/PDF CREATE or COPY command before exiting the window.

When you END from this screen, the SQL is automatically reexplained and the SQL Analysis Summary screen displays.

Figure 16-29. Edit session initiated after executing the SQL command

```
EDIT ---- SYS95143.T102452.RA000.DFHMSH0.R0152152 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
****** ***************************** TOP OF DATA ******************************
=NOTE=
=NOTE=  SQL MAY BE MODIFIED. IF THE SQL IS MODIFIED IT WILL
=NOTE=  AUTOMATICALLY BE REEXPLAINED WHEN YOU END. THE PLAN,
=NOTE=  PACKAGE OR DBRM WILL NOT BE UPDATED WITH THE MODIFIED SQL.
=NOTE=
000001        SELECT LASTNAME, FIRSTNME, SALARY, HIREDATE
000002        FROM TSOID01.EMP
000003        WHERE WORKDEPT
000004        IN ('A00', 'B01')
000005        AND SALARY > 10000
000006        AND HIREDATE < '1994-01-01'
000007        ORDER BY LASTNAME, FIRSTNME
****** **************************** BOTTOM OF DATA ****************************
```

Printing Predicate Analysis Reports

Use PRINTP (using the associated PF key) to print the Predicate Analysis reports. The Predicate Analysis Report window (Figure 16-30 on page 16-27) displays. Select the desired report and press Enter. When you exit the Analysis screen (Figure 16-27 on page 16-26), the SQL Analysis Output Destination Specification screen (Figure 16-22 on page 16-22) displays. This screen enables you to specify whether you want the report printed to a printer or a dataset.

Figure 16-30. Predicate Analysis Reports Window

```
File-AID for DB2 ----------- SQL Predicate Analysis --------------- SSID: DSN
C +---- Predicate Analysis Reports ----+                     SCROLL ===> PAGE
  ¦ COMMAND ===>                       ¦
U ¦                                    ¦ ource
  ¦ Use 'S' to Select Report           ¦
T ¦   _ Predicate Analysis Summary     ¦ for additional information
  ¦   _ Predicate Analysis Detail      ¦
  ¦                                    ¦ DATE
  +------------------------------------+ he cursor is retrieval only -- )
    ( -- RULE0014I:  Use the Optimize For n Rows Clause -- )
  FROM TSOID01.EMP
  WHERE WORKDEPT
  IN ('A00', 'B01')
  AND SALARY > 10000
  AND HIREDATE < '1994-01-01'
  ORDER BY LASTNAME, FIRSTNME
    ( -- RULE0013I:  List only necessary columns in Order By clause -- )
```

Select one of the following reports:

Predicate Analysis Summary

Shows the rules and the embedded SQL statement(s).

Predicate Analysis Detail

Shows the rules with the embedded SQL statement(s) as well as the recommendation text.

Short and Long Recommendations

For information on the short and long messages you may receive when using Predicate Analysis, see Appendix A, “Predicate Analysis Rules Abbreviations”.

Catalog Information

The Object Selection screen (Figure 16-31 on page 16-28) displays when you select option 4, Catalog Information, from the SQL Development and Analysis Summary screen.

Figure 16-31. Object Selection Screen

```
File-AID for DB2 -------------- Object Selection ------------- Row 1 to 3 of 3
 COMMAND ===>                                                  SCROLL ===> CSR

   Enter 'S'  to select the table for Additional information:

     OBJECT                TYPE   NPAGES   PCTPAGES   CARDINALITY
     -------               ----   ------   --------   ------------
  _  DSN8610.EMP           T      2        1          42
  _  FADB2610.ALTDATE      F      ---      ---        ---
  _  FADB2610.ALTDATE      F      ---      ---        ---
****************************** Bottom of data ********************************
```

TABLE

The fully qualified table name.

TYPE

The type of object. T indicates Table, and F indicates UDF.

NPAGES

The total number of pages on which rows of the table exist. This field can contain a value of -1 if statistics are not available.

PCTPAGES

The percentage of total pages of the table space that contain rows of the table. If the table space is segmented, the value shown is the percentage of the total pages in the set of segments assigned to the table. This field can contain a value of -1 if there are no statistics available.

CARDINALITY

The number of rows in the table. This field can contain a value of -1 if there are no statistics available.

To select an object, enter an **S** line command in the area to the left of the desired table. If you select a table, the Table Information screen displays (Figure 16-32 on page 16-29). If you select a user defined function, the Function Detail screen displays. For more information about the Function Detail screen, refer to “Function Detail” on page 16-35.

The Table Information screen provides detailed information about the selected object. From this screen you can receive additional column, index, and related table information, and table partition statistics.

Note: You can use the tab key to go between selection list items by row between columns (Column Information, then Table Partition Statistics, then back to Index information). Alternatively, you can use the new line key (consult your keyboard mapping for the location of this key) to go between selection list items for the first column only (Column Information, then Index Information, etc.).

Figure 16-32. Table Information screen

```
File-AID for DB2 ------------- Table Information ------------------ SSID: D610
COMMAND ===>

Use 'S' to select more information for Table: USERID1.TRIGTAB
 _  Column Information                  _ Table Partition Statistics
 _  Index Information                   _ Table Check Constraints
 _  Related Table Information           _ Remarks/Comment On
 S  Related Trigger Information

CREATEDBY: BFHUSER1   PCTPAGES: -1      AUDITING  : N
COLCOUNT : 2          KEYCOLS : 0       CHECKRID  : <NONE>
DBNAME   : BFHUSRDB   KEYOBID : 0       CHECKFLAG : <NONE>
TSNAME   : BFHUSRTS   STATUS  : <NONE>  RBA1      : 0003F58716AC
CARD     : -1         PARENTS : 0       RBA2      : 0003F58716AC
DBID     : 474        CHILDREN: 0       CREATEDTS : 2001-04-25-11.45.15.525438
OBID     : 170        DATACAPT: N       ALTEREDTS : 2001-04-25-11.45.15.525438
TYPE     : T          PCTRWCMP: -1      STATSTIME : 0001-01-01-00.00.00.000000
RECLENGTH: 53         EDPROC  : <NONE>     CHECKS : 0
NPAGES   : -1         VALPROC : <NONE>  LOCATION  : XXXXXXXXXXXXXXXX
ENCSCHEME: X          RELCRETD: X       CARDF     : -1.000000E+00
TABSTATUS: XXXXXXXXXX                   NPAGESF   :
AVGROWLEN: XXXXXXXXXXX                  SPACEF    :
LABEL    : <NONE>
```

Select one for more information:

Column Information

Displays the Column Information screen (Figure 16-33 on page 16-30).

Table Index Selection

Displays the Table Index Selection screen (Figure 16-37 on page 16-32).

Related Table Information

Displays the Related Table Information screen (Figure 16-41 on page 16-34).

Related Trigger Information

Displays the Trigger Information screen (Figure 10-7 on page 10-23).

Table Partition Statistics

Displays the Table Partition Statistics screen (Figure 16-42 on page 16-34).

Table Check Constraints

Displays the Check Constraints screen (Figure 16-43 on page 16-35).

Remarks/Comment On

Displays the Comment On screen (Figure 10-9 on page 10-25).

Column Information

The Column Information screen (Figure 16-33 on page 16-30) displays when the Column Information option is selected on the Table Information screen (Figure 16-32). This screen contains detailed information about each column in the selected table, including column name, column type, length, scale, and default values. To view RUNSTATS statistics, enter RIGHT, or the associated PF key. Enter RIGHT again to view the COLCARDF values. The scroll amount field is ignored when scrolling right. To return to the original display, enter LEFT or the associated PF key.

Enter an **S** in the Column Partition Statistics field to display the Column Partition Statistics screen (Figure 16-36 on page 16-31).

Figure 16-33. Column Information Screen

```
File-AID for DB2 ------------ Column Information ---------- Row 1 to  9 of 13
COMMAND ===>                                                 SCROLL ===> CSR

Enter 'S' for additional Column information
_ Column Partition Statistics

Creator   : SYSIBM             Database  : DATABASE
Table Name: SYSDBAUTH          Tablespace: TBLSPACE                  More: >

Column Name       Number  Type    Length Scale Null  Default Value
----------------- ------  ------- ------ ----- ----  --------------------------
NAME                   1  VARCHAR    18     0  N
CREATOR                2  CHAR        8     0  N    USER
TYPE                   3  CHAR        1     0  N
DBNAME                 4  CHAR        8     0  N
TSNAME                 5  CHAR        8     0  N
DBID                   6  SMALLIN     2     0  N
OBID                   7  SMALLIN     2     0  N
COLCOUNT               8  SMALLIN     2     0  N
EDPROC                 9  CHAR        8     0  N
****************************** BOTTOM OF DATA ******************************
```

Figure 16-34. Column Information Screen — After Scrolling RIght

```
File-AID for DB2 ------------ Column Information ---------- Row 1 to  9 of 13
COMMAND ===>                                                 SCROLL ===> CSR

Enter 'S' for additional Column information
_ Column Partition Statistics

Creator   : SYSIBM             Database  : DSNDB06
Table Name: SYSDBAUTH          Tablespace: SYSDBASE                  More: <>

Column Name         Colcard   High2key Low2key  Stats Time
-----------------   -------   -------- -------  --------------------------
NAME                     -1                     0000-00-00-00.00.00.000000
CREATOR                  94   XDB      BFHLIH0  1994-01-04-12.05.00.000100
TYPE                     -1                     0000-00-00-00.00.00.000000
DBNAME                   -1                     0000-00-00-00.00.00.000000
TSNAME                   -1                     0000-00-00-00.00.00.000000
DBID                     77                     1994-01-04-12.05.00.000100
OBID                     -1                     0000-00-00-00.00.00.000000
COLCOUNT                 -1                     0000-00-00-00.00.00.000000
EDPROC                   -1                     0000-00-00-00.00.00.000000
****************************** BOTTOM OF DATA ******************************
```

Figure 16-35. Column Information Screen — After Scrolling Right Twice

```
File-AID for DB2 ------------ Column Information ---------- Row 1 to  9 of 13
COMMAND ===>                                                 SCROLL ===> CSR

Enter 'S' for additional Column information
_ Column Partition Statistics

Creator   : SYSIBM             Database  : DSNDB06
Table Name: SYSDBAUTH          Tablespace: SYSDBASE                  More: <

Column Name         ColcardF
------------------  ----------
NAME                N/A
CREATOR             N/A
TYPE                N/A
DBNAME              N/A
TSNAME              N/A
DBID                N/A
OBID                N/A
COLCOUNT            N/A
EDPROC              N/A
****************************** BOTTOM OF DATA ********************************
```

Column Partition Statistics

The Column Partition Statistics screen (Figure 16-36) displays when an **S** is entered next to the Column Partition Statistics field on the Column Information screen (Figure 16-33 on page 16-30).

Figure 16-36. Column Partition Statistics Screen

```
File-AID for DB2 ------- Column Partition Statistics ------ Row 1 to 12 of 13
COMMAND ===>                                                 SCROLL ===> CSR

Creator   : SYSIBM             Database  : DATABASE
Table Name: SYSDBAUTH          Tablespace: SYSDBASE
Partition :  1 OF 36

Column Name         Colcard  High2key Low2key  Stats Time
------------------  -------  -------- -------  --------------------------
NAME                     -1                   0000-00-00-00.00.00.000000
CREATOR                  94  XDB      BFHLIH0  1994-01-04-12.05.00.000100
TYPE                     -1                   0000-00-00-00.00.00.000000
DBNAME                   -1                   0000-00-00-00.00.00.000000
TSNAME                   77                   1994-01-04-12.05.00.000100
DBID                     -1                   0000-00-00-00.00.00.000000
OBID                     -1                   0000-00-00-00.00.00.000000
COLCOUNT                 -1                   0000-00-00-00.00.00.000000
EDPROC                   -1                   0000-00-00-00.00.00.000000
VALPROC                  -1                   0000-00-00-00.00.00.000000
CLUSTERTYPE              -1                   0000-00-00-00.00.00.000000
SALARY                   -1                   0000-00-00-00.00.00.000000
****************************** BOTTOM OF DATA ********************************
F10=PREVIOUS   F11=NEXT
```

Index Selection

The Table Index Selection screen (Figure 16-37 on page 16-32) displays when the Index Information option is selected on the Table Information screen (Figure 16-32 on page 16-29) and more than one index exists for the selected table. If only one index exists for the selected table, the Index Information screen (Figure 16-38 on page 16-32) displays. If no index for the selected table exists, the message **Index not found** displays in the upper right corner of the screen.

To request additional index information, enter **S** in the area provided to the left of the CREATOR column.

Figure 16-37. Table Index Selection Screen

```
File-AID for DB2 ---------- Table Index Selection -----------  ROW 1 TO 2 OF 2
COMMAND ===>                                                 SCROLL ===> CSR

Creator   : SYSIBM             Database  : DSNDB06
Table Name: SYSTABLES          Tablespace: SYSDBASE

  Select one of the following for additional information:

     CREATOR   NAME     UNIQUERULE  CLUSTERING CLUSTERED CLUSTERRATIO
_    SYSIBM    DSNDTX02      U           N          N     81
_    SYSIBM    DSNDTX01      P           N          N     66
****************************** BOTTOM OF DATA ********************************
```

Index Information

The Index Information screen (Figure 16-38) contains detailed information about the selected table's index. If only one index exists for the table, this screen displays immediately after the Index Information option is selected on the Table Information screen (Figure 16-32 on page 16-29).

This screen lists detailed information about the selected index, including index name, creator ID, and buffer pool. From this screen, you can request additional index key information or index partition information.

Figure 16-38. Index Information screen

```
File-AID for DB2 ------------ Index Information -------------------- SSID: DSN
COMMAND  ===>

Use 'S' for additional Index Information:

 _ Index Key Information
 _ Index Partition Information

Creator   : SYSIBM             Database  : DATABASE
Table Name: EMP                Tablespace: TBLSPACE

Index Information for: SYSIBM.DSNDTX01

CREATEDBY    : FRED       INDEXSPACE : DSNDTX01   NLEAF     : 54
UNIQUERULE   : PRIMARY    DBNAME     : DSNDB06    NLEVELS   : 2
COLCOUNT     : 2          BPOOL      : BP0        PAGESIZE  : 4096
CLUSTERING   : N          DBID       : 6          FRSTKYCRD : 51
CLUSTERED    : N          OBID       : 61         FULLKYCAR : 1030
CLUSTERRATIO : 66         ISOBID     : 93         ERASERULE : NO
DASDSPACE    : 0          DSETPASSW  :            CLOSERULE : NES
STATSTIME    : 1994-01-04-12.05.32.000100        INDEXTYPE:
FIRSTKEYCARDF:                                   PIECESIZE:
FULLKEYCARDF :                                   COPY:
```

Index Key Information

The Index Key Information screen (Figure 16-39 on page 16-33) displays when an **S** is entered in the Index Key Information field on the Index Information screen (Figure 16-38). This screen contains specific information about the keys defined in the selected index.

Figure 16-39. Index Key Information

```
File-AID for DB2 ------------ Index Key Information ---------  ROW 1 TO 2 OF 2
COMMAND ===>                                                  SCROLL ===> CSR

   CREATOR: SYSIBM               DATABASE: DSNDB06
TABLE NAME: SYSTABLES          TABLESPACE: SYSDBASE

COLUMN NAME         COLNO   COLSEQ   ORDERING
CREATOR                 2        1    ASCENDING
NAME                    1        2    ASCENDING
****************************** BOTTOM OF DATA ********************************
```

Index Partition Information

The Index Partition Information screen (Figure 16-40) displays when an S is entered in the Index Partition Information field on the Index Information screen (Figure 16-38 on page 16-32).

The top portion of the screen describes the current index and related table. The lower portion of the screen is a scrollable list of partitions defined for the indexspace. A partitioned index will have one set of attributes for each partition. A single set of attributes with a partition of 0 indicates the index is non-partitioned.

Figure 16-40. Index Partition Information Screen

```
File-AID for DB2 ---------- Index Partition Information -----  ROW 1 TO 1 OF 1
COMMAND ===>                                                  SCROLL ===> CSR

Creator   : TSOID01            Database  : DATABASE
Table Name: EMP                Tablespace: TBLSPACE

Index Partition Information for: TSOID01.XEMP1
PARTITION    : 1 OF 1    FREEPAGE  :           FAROFFPOS    :
PRIQTY       :           PCTFREE   :           NEAROFFPOS   :
SECQTY       :           CARD      :           LEAFDIST     :
STOGROUP     :           NLEVELS   : 1         CLUSTERRATIO : 97
FIRSTKEYCARD : 34        NLEAF     : 1         KEYCOUNT     : 50
FULLKEYCARD  : 34        CARDF     : -1        FAROFFPOSF   :
LIMITKEY     : 0                               NEAROFFPOSF  :
GBPCACHE     : N/A       STATSTIME : 1994-01-04-12.05.32.000100

****************************** BOTTOM OF DATA ********************************
```

Related Table Information

The Related Table Information screen (Figure 16-41 on page 16-34) displays when the Related Table Information option is selected on the Table Information screen (Figure 16-32 on page 16-29). This screen lists each table linked to the selected table (RI relationships only).

Figure 16-41. Related Table Information Screen

```
File-AID for DB2 ---------- Related Table Information -------- Row 1 to 7 of 7
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID:

   CREATOR: EFHLAWO              DATABASE: EFHLAWDB
TABLE NAME: EMP                TABLESPACE: EFHLAWTS

-------- Parent Tables --------
CREATOR  TABLE NAME         RELNAME  COLNAME         COLSEQ   DELETE RULE
EFHLAWO  EMP2               PT124262 EMPNO                1   CASCADE

-------- Dependent Tables -----
CREATOR  TABLE NAME         RELNAME  COLNAME          COLNO   DELETE RULE
EFHLAWO  ALTER-TEST         TEST_REL EMPNO                1   CASCADE
****************************** Bottom of data ********************************
```

Related Trigger Information

The Trigger Information screen (Figure 10-7 on page 10-23) displays when the Related Trigger Information option is selected on the Table Information screen (Figure 16-32 on page 16-29). The screen lists information about triggers on the table.

Table Partition Statistics

The Table Partition Statistics screen (Figure 16-42) displays when the Table Partition Statistics option is selected on the Table Information screen (Figure 16-32 on page 16-29). This screen lists tablespace statistics for each partition in the table.

Figure 16-42. Table Partition Statistics Screen

```
File-AID for DB2 --------- Table Partition Statistics ---------ROW 1 TO 2 OF  2
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID: DSN
Creator   : SYSIBM            Database  : DSNDB06
Table Name: SYSDBAUTH         Tablespace: SYSDBASE

PARTITION :  1 OF  2    PCTPAGES  : 100
CARD      : 50          NACTIVE   : 2
NPAGES    : 2           PCTROWCOMP: 95      GBPCACHE:
STATSTIME : 1994-01-04-12.05.32.000100

PARTITION :  2 OF  2    PCTPAGES  : 100
CARD      : 50          NACTIVE   : 2
NPAGES    : 2           PCTROWCOMP: 95
STATSTIME : 1994-01-04-12.05.32.000100
****************************** BOTTOM OF DATA ********************************
```

Table Check Constraints

The Check Constraints screen (Figure 16-43 on page 16-35) displays when the Table Check Constraints option is selected on the Table Information screen (Figure 16-32 on page 16-29). The Check Constraints screen displays all defined check constraints for the selected table. The WHERE field only displays the first 55 characters of the WHERE clause.

Figure 16-43. Check Constraints Screen

```
File-AID for DB2 ------------ Check Constraints --------------- ROW 1 TO 5 OF 5
COMMAND ===>                                                   SCROLL ===> PAGE
                                                                     SSID: DSN

Check Constraints for table:  TSOID01.PROB1

CONSTRAINT NAME     WHERE
------------------  ------------------------------------------------------------
ASSIGN              ASSIGN IS NULL    OR ASSIGN IN('Robey',    'Lisa', 'K
NEW PRIORITY        PRTY IS NULL    OR PRTY    IN ('1',    '2',    '3',  '
REP                 REP IN('Brian',    'Robey',    'Lisa',    'Kenny', 'K
STAT                STAT IN('Open',    'Clos',    'Fix',    'Brok')
Type                TYPE IN('Abend',    'Appear',    'Result','Msg','Enh','D
****************************** BOTTOM OF DATA *******************************
```

Remarks/Comment On

The Comment On screen (Figure 10-9 on page 10-25) displays when the Remarks/Comment On option is selected on the Table Information screen (Figure 16-32 on page 16-29). The screen lists remarks and comments associated with the table.

Cost Detail

The Cost Detail Information screen displays when you select option 5, Cost Detail, from the Plan Table Analysis screen (Figure 16-20 on page 16-19). It provides detailed cost information about the selected table. Information on this screen is taken from the DB2 DSN_STATEMNT_TABLE. The DSN_STATEMNT_TABLE is optional and can be created under option 4.0.

Figure 16-44. Cost Detail Information Screen

```
File-AID for DB2 ------------- Dsn Statement Table -----------------------------
  COMMAND ===>
   Query Number     : 1
   Plan Name        : C215
   Program Name     : C2158001
   Collection ID    : Prod_Online
   Group Member     :
   Statement Type   : Select
   Cost Category    : B
   Explain Time     : 2001-08-06 10.43.06.06001

   Processor Cost
    Milliseconds    : 4410
    Service Units   : 3702
   Reason for Cost Category B
       Host Variables,
       Referential contraints
```

Function Detail

The Function Detail Information screen displays when you select option 6, Function Detail, from the Plan Table Analysis screen (Figure 16-20 on page 16-19). It provides detailed function information about the selected table. Information on this screen is taken from the DB2 DSN Function table.

Figure 16-45. Function Detail Information Screen

```
File-AID for DB2 -----------Function Detail -----------------------------------
 COMMAND ===>

 Use 'S' for additional information
    Catalog information for this UDF

                                Explain Time: 2004-07-06-14.19.02.100000
 Function Deatil ---------------------------- SYSROUTINES Data ---------------
 Query Number    : 1                          Cardinality     : -1.000000E+00
 Query Block No  : 1                          ISO per Invoc   : -1.000000E+00
 Schema          : FADB2610                   Insts per Invoc : -1.000000E+00
 Function Name   : WEATHER                    Initial IOS     : -1.000000E+00
 Specific Name   : DSN8DUWFV                  Initial Insts   : -1.000000E+00
 View Creator    :
 View Name       :
 Collection ID   : FD470013
 Program Name    : OSSQLCAT
 Plan Name       :
 Group Member    :
 Function Type   : TU
 Function Text   : FADB2610.WEATHER(' ')                                      >
 Path            : "SYSIBM","SYSFUN","SYSPROC","FDDBA"                        +

  F5=PRINT F10=PREV  F11=NEXT
```

Explaining Objects in Batch

The SQL Plan Analysis can be processed in Batch mode if you specify Batch processing on the SQL Plan Analysis Specification screen (Figure 16-11 on page 16-9). (Enter **2** in the Processing Mode field.) The Batch Plan Analysis Reports window (Figure 16-46) displays when an X is entered in the CMD field on the SQL Plan Selection screen (Figure 16-13 on page 16-12). You can select more than one plan.

When you press Enter, the SQL Analysis Batch Output Specification screen displays (Figure 16-47).

Figure 16-46. Batch Plan Analysis Reports Window

```
File-AID for DB2 --------- Collection ID Selection ----------  ROW 1 TO 2 OF 2
C +-------- Batch Plan Analysis Reports --------+             SCROLL ===> PAGE
  | COMMAND ===>                                |                   SSID: DSN
  |                                             |
L | Enter 'S' to Select One or More Reports     |
  |   _ Access Path Analysis                    |
  |   _ Explain Detail                          |
  |   _ Predicate Analysis Summary              |
C |   _ Predicate Analysis Detail               |
_ |                                             |
X |                                             |
* |                                             | ****************************
  +---------------------------------------------+
```

Enter an **S** next to the reports you want generated.

Access Path Analysis

See “Access Path Analysis” on page 16-20 for more information.

Explain Detail

See “Explain Detail” on page 16-23 for more information.

Predicate Analysis Summary

See “Predicate Analysis” on page 16-25 for more information.

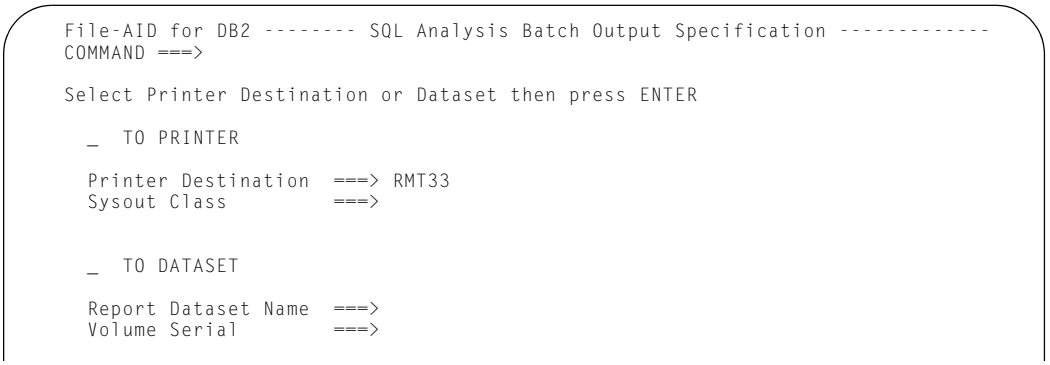
Predicate Analysis Detail

See "Predicate Analysis" on page 16-25 for more information.

SQL Analysis Batch Output Specification

On the SQL Analysis Batch Output Specification screen you can route the selected reports to a printer or a file. You cannot elect to send the report to both destinations. Use the S line command to select the destination and enter the required fields for the appropriate destination. Set the default print options and job statement by selecting option 0, Defaults and User Parameters, from the Primary Option Menu. See "Option 7, Print Options" on page 5-10 and "Option 3, Batch Options" on page 5-6.

Figure 16-47. SQL Analysis Batch Output Specification Screen



After you enter the appropriate information, File-AID *for DB2* generates a temporary dataset containing the JCL. Use the SUBMIT primary command to submit the job. If you want to save the dataset, use the ISPF/PDF CREATE or COPY command before exiting the screen. Once submitted, the SQL Analysis report is routed to the selected destination.

Explaining Objects in Batch via a Source Listing File

If you created a source file listing with XPEDITER or Abend-AID, you can explain the SQL in the file. Your program must be processed with the postprocessor.

Create the JCL below and provide the appropriate dataset names and parameters. Typically, the explain will be the last step in a job that performs a compile, link-edit, and bind.

Figure 16-48. JCL to Explain via SLS

```
//JOBCARD
//*      .
//*      .
//*   COMPILE
//*      .
//*      .
//*  LINK EDIT
//*      .
//*      .
//*    BIND
//*      .
//*      .
//*  EXPLAIN:
//*--------------------------------------------------------------------
//UTIL EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=8192K,COND=(7,LT)
//STEPLIB  DD DISP=SHR,DSN=FADB2.V4R8Mx.LOADLIB
//         DD DSN=DSN610.D610.SDSNEXIT,DISP=SHR
//         DD DSN=DSN610.SDSNLOAD,DISP=SHR
//*
//ISPPLIB  DD DISP=SHR,DSN=ISP.SISPPENU
//         DD DISP=SHR,DSN=FADB2.V4R8Mx.ISPPLIB
//*
//ISPMLIB  DD DISP=SHR,DSN=FADB2.V4R8Mx.ISPMLIB
//         DD DISP=SHR,DSN=ISP.SISPMENU
//*
//ISPSLIB  DD DISP=SHR,DSN=ISP.SISPSENU
//         DD DISP=SHR,DSN=FADB2.V4R8Mx.ISPSLIB
//*
//ISPTLIB  DD DISP=SHR,DSN=FADB2.V4R8Mx.ISPTLIB
//         DD DISP=SHR,DSN=ISP.SISPTENU
//*
//SLSF001  DD DISP=SHR,DSN=FADB2.SLSFILE
//*
//ISPPROF  DD DSN=&BATPROF,DISP=(NEW,DELETE,DELETE),
//            DCB=(BLKSIZE=3120,LRECL=80,DSORG=PO,RECFM=FB),
//            SPACE=(TRK,(1,,1)),UNIT=SYSDA
//*
//ISPCTL1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(TRK,(5,5,0)),
//            DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB)
//*
//CURRPATH DD *
“SYSIBM”,”SYSFUN”,”SYSPROC”,”SCHEMA1”,”SCHEMA2”,”SCHEMA3”,
”SCHEMA4”,”SCHEMA5”,”SCHEMA6”,”SCHEMA7”,”SCHEMA8”,”SCHEMA9”,
“SCHEM10”,”SCHEM11”,”SCHEM12”,”SCHEM13”,”SCHEM14”,”SCHEM15”,
“SCHEM16”,”SCHEM17”,”SCHEM18”,”SCHEM19”,”SCHEM20”,”SCHEM21”,
”SCHEM22”,”SCHEM23”,”SCHEM24”
/*
//SYSTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//F2SYSOUT DD SYSOUT=*,DEST=LOCAL,HOLD=YES
//SYSTSIN  DD *
ISPSTART PGM(F2ANLYZ) NEWAPPL(F2DE) PARM(subsystem plan-name +
        creator qualifier SOURCE member +
        LISTEND +
        ALL +
        LISTEND +
        dsn volume  +
        print-option ALL +
        analysis-options, rule-datasetname,
        object qualifier option, trace, plan table creator)

/*
```

DD Statements

STEPLIB

Step library

ISPPLIB

Panel library

ISPMLIB

Message library

ISPSLIB

Skeleton library

ISPTLIB

Table library

SLSF00 *n*

Where *n* is a value between 1 and 8. A maximum of eight source list files can be specified.

ISPPROF

ISPF profile dataset definition for the batch TSO session

ISPCTL1

Temporary dataset used by ISPF

SYSTSPRT

Required DD statement

SYSTERM

Required DD statement

SYSPRINT

Required DD statement

SYSUDUMP

Required DD statement

F2SYSOUT

Printer information. Required if *print-option* is Y.

SYSTSIN

Required DD statement

JCL Input

PGM

Specify the File-AID *for DB2* program F2ANLYZ

NEWAPPL

Specify the File-AID *for DB2* application identification

CURRPATH

Specify the schemas to replace the CURRENT PATH special register value. Can be up to 254 characters.

Parameters

Subsystem

Subsystem name

Plan Name

Plan name of File-AID for DB2

Creator

Userid of the creator

Qualifier

Qualifier used for unqualified tables

SOURCE

Required keyword. Specifies that the SQL is in the source listing file.

Member

Specify the member in the source listing file containing the program to be explained.

LISTEND

Required keyword

ALL

Required keyword

LISTEND

Required keyword

DSN

Output dataset name. If Print-option is N, specify a dataset set name in single quotes and double quotes. If Print-option is Y, specify two sets of double quotes.

Volume

Volume of the dataset name specified for the *DSN* parameter. Enclose the volume in double quotes. If writing to a printer, specify double quotes.

Print-option

Y Writes the report to the printer specified in the DD F2SYSOUT

N Writes the report to the dataset specified in the *DSN* parameter

ALL

Required keyword. Performs an Explain for all statements.

Analysis Options

Specifies which report you want printed. The available reports are listed below. Enter a Y or N for each report. Do not separate with blanks.

- Access Path Analysis
- Explain Detail
- Predicate Analysis Summary
- Predicate Analysis Detail

Rule Dataset Name

Identifies the name of the dataset containing the rules for predicate analysis.

Object Qualifier Option

Specify which qualifier option to use (see also “Qualify By” on page 16-10).

1 Current SQLID

2 Bind Qualifier

3 Default Qualifier

Trace

Specify which logging option to use

1 Write to ISPF log.

2 Write to SYSPRINT.

3 Write to ISPF log and SYSPRINT.

4 No logging.

Plan Table Creator

Specify the Creator ID for the plan table you want to use.

Sample Explain JCL

The following JCL will perform an explain on the SQL in member SQLDEMO. The output will be written to 'TSOID01.XP.OUTPUT'. The following reports will be created: Access Path Analysis and Predicate Analysis. The rule dataset FADB2.V4R8Mx.ISPSLIB will be used as well as the current SQLID for the object qualifier, logging to SYSPRINT, and the plan table of creator TSOID01.

Figure 16-49. Sample JCL to Explain with SLS

```
  .
  .
  .
ISPSTART PGM(F2ANLYZ) NEWAPPL(F2DE) PARM(  DSNA FDxxxxxx +
        CREATOR QUALIF SOURCE +
        SQLDEMO +
        LISTEND +
        ALL +
        LISTEND +
        "'TSOID01.XP.OUTPUT'" "" +
        N ALL    +
        YNYN FADB2.V4R8Mx.ISPSLIB 1 2 TSOID01)
/*
/*
```

Option 2, Review or Print DB2 Explain Reports

Select option **2** on the SQL Analysis and Development screen. The SQL Explain Report Specification screen (Figure 16-50) displays. On this screen you can print or review a history of SQL Analysis Reports by entering a range of Explain dates and times.

Figure 16-50. SQL Explain Report Specification Screen

```
File-AID for DB2 ----- SQL Explain Report Specification -----------
COMMAND ===>                                                SSID: 701

Specify Processing Options:
  Processing Mode ===> 1         (1 = Online, 2 = Batch)
Plan_Table AUTHID ===> USERID    Authorization ID of Plan_Table

Specify the range of Explain dates/times for review:
  Starting Date ===> 1992-11-05      Starting Time ===> 00.00.01
  Ending Date   ===> 1992-11-05      Ending Time   ===> 08.32.37
```

Processing Mode

Specifies the destination of output. The following options are available:

1 Displays the output on the screen. The SQL Analysis Summary screen displays allowing you to select a SQL statement for more analysis. See "SQL Analysis Summary" on page 16-17 for more information

2 Generates JCL to print the job. The SQL Analysis Batch Output Specification screen (Figure 16-47 on page 16-37) displays, where you can specify if you want the output to be sent to a dataset or a printer. See "SQL Analysis Batch Output Specification" on page 16-37 for more information.

Plan_Table Authid

Enter the plan table authorization ID.

Starting Date/Ending Date

Enter the range of dates to select the explain reports. Each time this screen is accessed, the starting and ending date default to the current system date.

Starting Time/Ending Time

Enter the range of times to select the explain reports. Each time this screen is accessed, the starting time defaults to one second past midnight and the ending time defaults to the current system time.

Changing Report Format Skeletons

You can modify certain parts of the base report format skeleton, including changing titles, moving elements, or deleting elements of the report. You may want to modify report format skeletons to consistently produce reports with only certain elements, or to create reports that show the most important elements first.

To modify report format skeletons, comment out or modify lines in the F2PTROW member included with File-AID *for DB2* at installation.

You can add lines to the report format skeletons, but they cannot include new variables. You can change variable headings, but not the variables themselves, or their values.

F2PTROW

Elements in bold can be modified or deleted. Please note that you CANNOT change the value of the element (with exception of the title), merely whether or where the element displays.

```
 File-AID for DB2       EXPLAIN DETAIL REPORT          DATE  &CENDATE
                                                       TIME  &ZTIME
 PLAN NO:         &PLANNO
 STATEMENT NO:    &QUERYNO
 QUERY BLOCK NO: &QBLOCKNO
)BLANK
 TABLE NO:        &TABNO
)SEL &CREATOR ¬= &Z
 TABLE ACCESSED: &CREATOR..&TNAME
)ENDSEL
)SEL &CREATOR = &Z
 TABLE ACCESSED: <<NONE>>
)ENDSEL
)SEL &ACCESSCR ¬= &Z
 INDEX ACCESSED: &ACCESSCR..&ACCESSNA
)ENDSEL
)SEL &ACCESSCR = &Z
 INDEX ACCESSED: <<NONE>>
```

```
)ENDSEL
)BLANK
)TB 35
 METHOD       :   &METHOD        !COLUMN FUNCTION EVALUATED:  &COLUMNFN
 INDEX ONLY   :   &INDEXONL      !TABLESPACE LOCK MODE:       &TSLOCKMO
 ACCESS TYPE  :   &ACCESSTY      !ACCESS DEGREE:              &ACSSDGRE
 MATCH COLS   :   &MATCHCOL      !ACCCESS PGROUPID:           &ACSSPGID
 MULT IX SEQ  :   &MIXOPSEQ      !JOIN DEGREE:                &JOINDGRE
 PREEFETCH    :   &PREFETCH      !JOIN PGROUPID:              &JOINPGID
)BLANK
)TB 44
 SORT COMPOSITE TABLE PARALLEL GROUP ID: !&SORTCPGR
 SORT NEW TABLE PARALLEL GROUP ID:       !&SORTNPGR
)BLANK
)TB 35
 PARALLELISM MODE: &PRLMMODE     !JOIN TYPE:        &JOINTYPE
 MERGE JOIN COLS : &MRGEJOIN     !GROUP MEMBER:     &GROUPMEM
 CORRELATION NAME: &CORRNAME     !WHEN OPTIMIZE:    &WHENOPT
 PAGE RANGE      : &PAGERNGE     !QUERY BLOCK TYPE: &QBLKTYPE
)BLANK
)SEL &DB2VERS GE 6.0
 HINT NAME GIVEN : &OPTHINT
 HINT NAME USED  : &HINTUSED
 PRIMARY ACCESS  : &PRIMACCS
)ENDSEL
)SEL &DB2VERS GE 7.0
 PARENT QBLOCK NO: &PRQBLKNO
 TABLE TYPE      : &TBTYPEXP
)ENDSEL
)BLANK
)TB 28 37 47 59 67
                       UNIQUE    JOIN    ORDER BY    GROUP BY    PGROUP
                       ------    ----    --------    --------    ------
        SORT NEW TBL: !&SORTNUN!&SORTNJO!&SORTNOR!&SORTNGR!&SORTNPGR
  SORT COMPOSITE TBL: !&SORTCUN!&SORTCJO!&SORTCOR!&SORTCGR!&SORTCPGR
)BLANK 2
)CM ***************************************************
)CM  Get and display values for the Table(s) and Indexes
)CM  associated with the current explain row
)CM ***************************************************
 File-AID for DB2       CATALOG INFORMATION REPORT      DATE  &CENDATE
                                                        TIME  &ZTIME
)BLANK 2
 TABLE INFORMATION FOR: &CREATOR..&TNAME
)BLANK
)TB 35
 CREATED BY: &CREATOR   !DATABASE ID  : &BDBID
 COLUMNS   : &BCOLCNT   !OBJECT ID    : &BOBID
 DATABASE  : &XDBNAME   !TABLE TYPE   : &BTABTYP
 TABLESPACE: &XTSNAME   !RECORD LENGTH: &BRECLEN
)BLANK
)TB 35
 NPAGES    : &BNPAGES   !CARD          : &BCARD
 PCTPAGES  : &BPCTPAGE  !# OF PARENTS  : &BPARENT
 KEYCOLS   : &BKEYCOLS  !# OF CHILDREN : &BCHILD
 KEYOBID   : &BKEYOBID  !DATA CAPTURE  : &BDATACAP
 ENCSCHEME : &BENCSCHM  !STATUS        : &BSTATUS
)BLANK
)TB 35
 PCT ROW COMPRESSED: &BPCTROW !# OF CHECKS: &BCHECKS
 EDIT PROCEDURE    : &BEDPROC !CHECKRID   : &BCKRID
 VALID PROC        : &BVALPR  !CHECKFLAG  : &BCKFLAG
 AUDITING          : &BAUDIT  !LOCATION   : &BLOCATN
 CARDF             : &BCARDF
)BLANK
 RBA AT CREATE TB  : &BRBA1
 RBA AT ALTER TB   : &BRBA2
)BLANK
```

```
 TABLE CREATED: &BTABCRTS
 TABLE ALTERED: &BTABALTS
 LAST RUNSTATS: &BRUNSTAT
)BLANK
)BLANK 2
)ENDSEL
)SEL &BXCOUNT GE &BXCOUNTB
 INDEX INFORMATION FOR: &BXCREATR..&BXNAME
)BLANK
)TB 35
 CREATED BY   :&BXCRTRD     !DBID         : &BXDBID
 INDEX SPACE  :&BXSPACE     !OBID         : &BXOBID
 INDEX TYPE   :&BXTYPE      !IXSPACE OBID : &BXSOBID
 COLCOUNT     :&BXCOLCNT    !UNIQUE RULE  : &BXUNIQUE
 COLCOUNT     :&BXCOLCNT    !UNIQUE RULE  : &BXUNIQUE
 DATABASE     :&BXDBNAME    !CLUSTERING   : &BXCLUSTR
)BLANK
)TB 35
 CLUSTERED    : &BXCLSTRD   !FIRST KEY CARD : &BXFIRST
 CLUSTER RATIO: &BXCLSTRA   !FULL KEY CARD  : &BXFULL
 NLEAF        : &BXNLEAF    !FIRST KEY CARDF: &BXFIRSTF
 NLEVELS      : &BXNLEVEL   !FULL KEY CARDF : &BXFULLF
)BLANK
)TB 35
 BUFFERPOOL      : &BXBP       !PAGE SIZE  : &BXPAGESZ
 DATASET PASSWORD: &BXDSNPSW   !PIECE SIZE : &BXPIECSZ
 ERASE RULE      : &BXERASE    !COPY       : &BXCOPY
 CLOSE RULE      : &BXCLOSE    !DASD SPACE : &BXDASD
)BLANK
 LAST RUNSTATS: &BXRUNSTA
)ENDSEL
)BLANK
)ENDDOT
```

Option 3, ISPF/PDF SQL Source Development and Analysis

You can access SQL Source Development and Analysis function through File-AID *for DB2*, as well as an edit session in ISPF/PDF, Librarian, or Panvalet. This section describes accessing the function through option 4.3 from the File-AID *for DB2* Primary Option menu. However, once you begin an edit session from the other sources, all the procedures described in this section are available, with the exception of the EX (Execute) line command. It can only be used when accessed directly through File-AID *for DB2*

Select option 3 from the Analysis and Development screen (Figure 16-2 on page 16-2). The SQL Analysis Edit Entry screen displays. Enter the dataset name that contains the SQL you want to analyze.

The dataset name can be a sequential dataset or a partitioned dataset (PDS). The source code can be COBOL, PL/I, or C. Datasets entered on this screen cannot be CA-PANVALET or CA-LIBRARIAN. For information on accessing the interface to these packages, see “Option P, PANVALET Source Development and Analysis” on page 16-59 and “Option L, LIBRARIAN Source Development and Analysis” on page 16-59.

Figure 16-51. SQL Analysis Edit Entry Screen

```
File-AID for DB2 -------------- SQL Analysis Edit Entry -----------
COMMAND ===>                                                  SSID: 701

ISPF LIBRARY:
   PROJECT ===> COMPSQL
   GROUP   ===> COBOL
   TYPE    ===> SOURCE
   MEMBER  ===> SQLPROG        (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
   DATASET NAME  ===>
   VOLUME SERIAL ===>           (If not cataloged)

DATASET PASSWORD  ===>          (If password protected)
INITIAL MACRO     ===>
```

ISPF LIBRARY Area

If the dataset to be analyzed is an ISPF library, enter the required dataset name in the PROJECT, GROUP, TYPE, and MEMBER fields, and press Enter. If you omit the member name, a member list displays from which you can select a member.

OTHER PARTITIONED OR SEQUENTIAL DATASET Area

A dataset name specified in the OTHER PARTITIONED OR SEQUENTIAL DATASET fields takes precedence over a dataset specified in the ISPF LIBRARY fields.

DATASET NAME

If the dataset to be analyzed is not an ISPF library, enter the dataset name, enclosed in single quotes; otherwise, your user ID is prefixed to the name.

VOLUME SERIAL

Optional. If the volume serial is specified, the system catalog is not used.

DATASET PASSWORD

Optional. Used only with password protected datasets. RACF or ACF uses your TSO logon password for user identification. If entered, the password is not displayed on the screen.

INITIAL MACRO

Optional. Used only when you want to use your own macro, instead of the one supplied by File-AID *for DB2*, F2EDNOTE. An initial macro can contain any ISPF editing commands, and is executed after the data is read but before the data is displayed. Using an initial macro allows several commands to be executed together. The initial macro allows you to group commands used to set ISPF variables so that the same options are always used.

After you enter the dataset name on the SQL Analysis Edit Entry screen, the Edit screen (Figure 16-52 on page 16-46) displays.

Figure 16-52. Edit Screen

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
00351     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNO, LASTNAME, FIRSTNAME
00353       FROM VSQLSAMP
00354       WHERE CITY = :CITY-NAME
00355       AND STATE = :STATE-NAME
00356     END-EXEC.
00357     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQL-CODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
00366     EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00369     END-EXEC.
F16=FADB2
```

From the Edit screen, you can use the full power of the ISPF editor. For example, use the FIND command to move to the SQL statement you want to analyze, or use the I (Insert) line command to insert a new SQL statement to be analyzed.

With the introduction of long names in DB2 Version 8, the bounds setting of the data set type you are editing are very important in determining when names span multiple lines. File-AID *for DB2* uses the default bounds as defined in the IBM publication, *ISPF Edit and Edit Macros*. To check your bounds setting, enter the FADB2 BOUNDS command. To change the current bounds setting, enter FADB2 BOUNDS left_col,right_col. (e.g., FADB2 BOUNDS 1,72).

Note: Because the ISPF editor is in use, the disposition of any changes made to the source code are governed and controlled by the user's ISPF profile options. To exit the Edit screen, press END. In this editing mode you are operating in an ISPF Edit session and the ISPF profile options set are in effect. Commands such as SAVE and CANCEL have the same effect on any source code changes made as if you had edited the source using ISPF/PDF edit.

Using the SQL Analysis Line Commands

In addition to the ISPF/PDF commands, File-AID *for DB2* provides four line commands that enable you to analyze SQL. When the SQL statement to be analyzed displays, select the action for the statement by entering the desired line command in the line number area. The valid commands are:

EX Execute — only valid when accessed directly through File-AID *for DB2*.

XP Explain

BR Browse

ED Edit

The EX, XP, BR, and ED line commands must be placed on an SQL statement that begins with either an EXEC SQL statement or an SQL verb and ends with either a semicolon (;) or END-EXEC statement. Any SQL call can be selected for processing *except* the following:

- BEGIN DECLARE
- END DECLARE
- INCLUDE
- WHENEVER
- EXECUTE
- EXECUTE IMMEDIATE
- DESCRIBE
- PREPARE

To initiate the action after entering a line command on an SQL statement, type **FADB2** on the command line and press Enter, or use the associated PF key. See Chapter 3, "Primary Commands" for more information on the FADB2 command.

Figure 16-53 shows an example of executing a DECLARE CURSOR SQL statement where the EX line command is entered for statement number 351, the FADB2 command is entered at the command line, and the Enter key is pressed.

Figure 16-53. Executing a DECLARE CURSOR SQL Statement Dynamically

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------- COLUMNS 001 072
COMMAND ===> FADB2                                          SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
EX 51    EXEC SQL DECLARE SAMPLE CURSOR FOR
00352      SELECT SSNO, LASTNAME, FIRSTNAME
00353      FROM VSQLSAMP
00354      WHERE CITY = :CITY-NAME
00355      AND STATE = :STATE-NAME
00356    END-EXEC.
00357    EXEC SQL
00358      OPEN SAMPLE
00359    END-EXEC.
00360    PERFORM FETCH UNTIL SQL-CODE NOT EQUAL ZERO.
00361 PROG-END.
00362    EXEC SQL
00363      CLOSE SAMPLE
00364    END-EXEC.
00365 FETCH.
00366    EXEC SQL
00367      FETCH SAMPLE
00368        INTO :SSNR, :LASTNAME, :FIRSTNAME
00369    END-EXEC.
F16=FADB2
```

Entering Host Variable Values

If the SQL statement contains host variables, and you enter an EX, BR, or ED line command, you must enter values for the host variables before you can complete the action.

Note: You can generate an Explain report without entering host variable values. When you enter an XP line command on the Edit screen, you bypass the Host Variable Resolution screen.

You can provide values for the host variables in one of two ways:

- Enter the actual values on the Edit screen as shown in Figure 16-54. When you use this method, you are making an actual change to the SQL. If you do not want to save the change, you must enter the CANCEL primary command to exit the Edit session.

Figure 16-54. Entering Host Variable Values on the Edit Screen

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
00351     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNO, LASTNAME, FIRSTNAME
00353       FROM VSQLSAMP
00354       WHERE CITY = 'CHICAGO'
00355       AND STATE = 'IL'
00356     END-EXEC.
00357     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQLCODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
00366     EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00369     END-EXEC.
F16=FADB2
```

or

- Use the Host Variable Resolution screen (Figure 16-55 on page 16-48). This is a temporary method of providing values. The SQL is not modified.

  The Host Variable Resolution screen displays if you enter an EX, BR, or ED line command on the Edit screen without entering values for the host variables in the SQL statement. This screen enables you to see and enter values for the host variables in the SQL statement.

Figure 16-55. Host Variable Resolution Screen Prompting for Values

```
File-AID for DB2 --------------- Host Variable Resolution --------------------
OPTION ===>                                                 SCROLL ===> CSR

   1 - Browse Table                3 - Explain
   2 - Edit Table                  4 - Execute SQL

Selected SQL Statement                          Host Variable Values
-----------------------------------------   ----+---10----+---20----+---30
DECLARE GARY CURSOR FOR60
SELECT DEPTNO DEPTNAME MGRNO ADMRDEPT
FROM USERID.DEPT
WHERE DEPTNO >
:DVALUE1                                 CHAR(3)   ?
AND DEPTNO <
:DVALUE2                                 CHAR(3)   ?
GROUP BY DEPTNO
ORDER BY DEPTNO;
```

OPTION

Enter the desired option.

1 Initiates a Browse session for the results table (Figure 16-63 on page 16-54).

2 Initiates an Edit session for the results table. The screen displayed is similar to Figure 16-63 on page 16-54, the difference being that it is an Edit, not a Browse session.

3 Explains the SQL and displays the Plan Table Analysis screen (Figure 16-67 on page 16-57).

4 Executes the SQL and displays proper SQL return code in the upper left corner. If the SQL statement is a FETCH, File-AID *for DB2* automatically displays the row retrieved from the DB2 results table (Figure 16-62 on page 16-53 is a sample screen).

Selected SQL Statement

The selected SQL statement displays on the screen, along with its associated host variables. Each host variable is presented on a separate line.

Host Variable Values

The question marks (?) indicate that a value for the host variable must be specified. You can scroll to the right and left to enter host variable values for character strings with more than 30 bytes; maximum length is 256 bytes. Enter **NULL** to indicate NULL values.

Enter values for the host variables (Figure 16-56 on page 16-49), then enter the option associated with the action you want to take. For example, enter **4** to execute the SQL DECLARE statement shown in the figure. You must enter values for all options except option 3, Explain.

When entering values, single quotation marks should be used to delimit character data. No quotation marks should be used for numeric values.

Figure 16-56. Entering Values on the Host Variable Resolution Screen

```
File-AID for DB2 --------------- Host Variable Resolution --------------------
OPTION  ===> 4                                              SCROLL === > CSR

   1 - Browse Table                3 - Explain
   2 - Edit Table                  4 - Execute SQL

Selected SQL Statement                          Host Variable Values
----------------------------------------------  ----+---10----+---20----+---30

EXEC SQL DECLARE SAMPLE CURSOR FOR
SELECT SSNO, LASTNAME, FIRSTNAME FROM VSQLSAMP
WHERE CITY =
:CITY-NAME                                      'CHICAGO'
AND STATE =
:STATE-NAME                                     'IL'
END-EXEC.
```

Prototyping New SQL Statements

You can use ISPF/PDF Edit functions from the SQL Analysis Edit screen to insert and analyze new SQL statements. An SQL statement can be added in any of the following formats:

- Precede the SQL statement with EXEC SQL and follow it with END-EXEC., as if it was embedded in a COBOL program. For example,

```
EXEC SQL
     SELECT * FROM VSQLSAMP
END-EXEC.
```

- Precede the SQL statement with EXEC SQL and follow it with a semicolon, as if it was imbedded in a PL/1 program. For example:

```
EXEC SQL SELECT * FROM VSQLSAMP;
```

- Follow the SQL statement with a semicolon. For example:

```
SELECT * FROM VSQLSAMP;
```

You may use a host variable within your inserted SQL, but it must be preceded by a colon.

Once the new statement has been inserted, you can perform any of the valid analysis line commands on the statement: EX (Execute), XP (Explain), BR (Browse), or ED (Edit).

Prototyping Cursor Processing

In order to prototype cursor processing, each of the following SQL statements must be executed in order:

```
DECLARE CURSOR
OPEN CURSOR
FETCH CURSOR
```

Executing a DECLARE CURSOR SQL Statement

To execute the DECLARE CURSOR SQL statement shown at line 351 in Figure 16-57 on page 16-50, you must enter values for the host variables. Enter the required host variable values on the Edit screen or the Host Variable Resolution screen as described in "Entering Host Variable Values" on page 16-47.

Figure 16-57. Entering Host Variable Values on the Edit Screen

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
***** ***************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
00351     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNR, LASTNAME, FIRSTNAME
00353       FROM VSQLSAMP
00354       WHERE CITY = :CITY-NAME
00355       AND STATE = :STATE-NAME
00356     END-EXEC.
00357     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQL-CODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
00366     EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00369     END-EXEC.
F16=FADB2
```

SQL Analysis dynamically executes the SQL statement and displays the proper SQL return code at the message line (Figure 16-58).

Figure 16-58. Message Indicating Successful Execution of the SQL Statement

```
File-AID for DB2 ------------ Host Variable Resolution -----------------------
OPTION ===>                                                  SCROLL ===> CSR

SQIKE5 = 1, SUCCESSFUL EXECUTION
   1 - Browse Table                  3 - Explain
   2 - Edit Table                    4 - Execute SQL

Selected SQL Statement                          Host Variable Values
----------------------------------------------  ----+---10----+---20----+---30

EXEC SQL DECLARE SAMPLE CURSOR FOR
SELECT SSNO, LASTNAME, FIRSTNAME FROM VSQLSAMP
WHERE CITY =
:CITY-NAME                                      'CHICAGO'
AND STATE =
:STATE-NAME                                     'IL'
END-EXEC.
```

Executing an OPEN CURSOR SQL Statement

You can select other statements to be executed by returning to the Edit screen and selecting them.

For example, you can select the OPEN SQL statement for execution by entering the EX line command at line 357, (Figure 16-59 on page 16-51), typing **FADB2** in the command field, and pressing Enter.

Figure 16-59. Executing an OPEN CURSOR SQL

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
.EX       EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNR, LASTNAME, FIRSTNAME
00353       FROM VSQLSAMP
00354       WHERE CITY = :CITY-NAME
00355       AND STATE = :STATE-NAME
00356     END-EXEC.
EX 57     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQLCODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
00366     EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00369     END-EXEC.
F16=FADB2
```

If a DECLARE CURSOR was previously executed, SQL Analysis dynamically executes the OPEN CURSOR and displays the proper SQL return code. If a DECLARE CURSOR was not previously executed, the CURSOR cannot be opened, and an error message results.

In this case, we have already executed the DECLARE CURSOR, indicated by the .EX in the statement number area for statement 351.

Executing a FETCH SQL Statement

You can also execute a FETCH SQL statement. For example, you can select a FETCH for execution (Figure 16-60) for line 366.

Figure 16-60. Executing an SQL FETCH

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
SQLCODE = 0, SUCCESSFUL EXECUTION
00350 SQL-PROTOTYPE-START.
00351     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNR, LASTNAME, FIRSTNAME
00353       FROM VSQLSAMP
00354       WHERE CITY = :CITY-NAME
00355       AND STATE = :STATE-NAME
00356     END-EXEC.
.EX       EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQLCODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
EX 66     EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00369     END-EXEC.
F16=FADB2
```

SQL Analysis dynamically executes the FETCH statement and automatically displays the row retrieved (fetched) from the DB2 results table. Figure 16-61 on page 16-52 shows both the FETCH statement and the row that is fetched into storage.

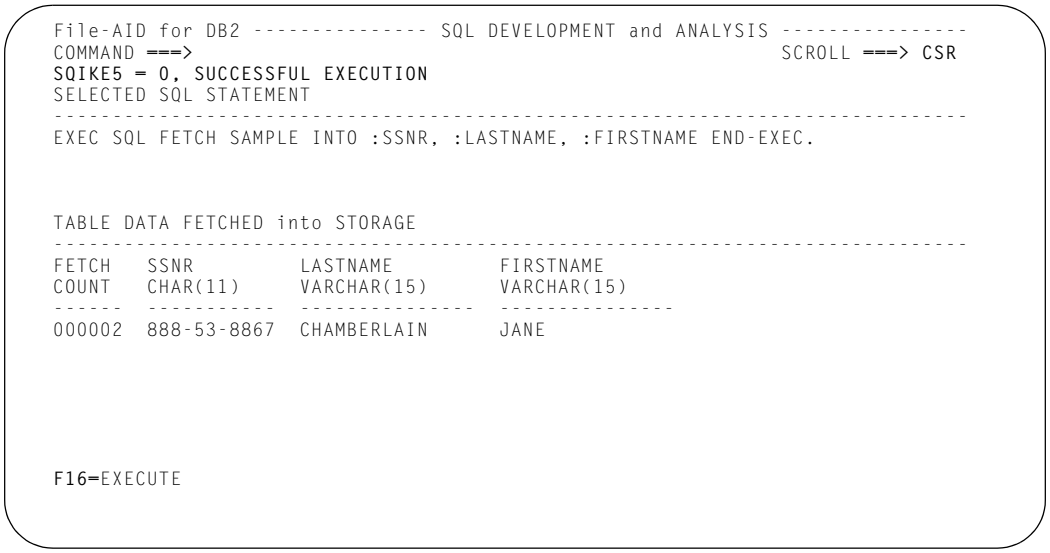
Figure 16-61. Display Showing SQL FETCH Statement and Table Data

```
File-AID for DB2 ---------------- SQL DEVELOPMENT and ANALYSIS -----------------
COMMAND ===>                                                  SCROLL ===> CSR
SQIKE5 = 0, SUCCESSFUL EXECUTION
SELECTED SQL STATEMENT
------------------------------------------------------------------------------
EXEC SQL FETCH SAMPLE INTO :SSNR, :LASTNAME, :FIRSTNAME END-EXEC.


TABLE DATA FETCHED into STORAGE
------------------------------------------------------------------------------
FETCH   SSNR         LASTNAME         FIRSTNAME
COUNT   CHAR(11)     VARCHAR(15)      VARCHAR(15)
------  -----------  ---------------  ---------------
000001  111-23-5438  DOE              JOHN




F16=EXECUTE
```

You can continue executing the FETCH statement using the EXECUTE command until you receive a message indicating there are no more rows to fetch. Figure 16-62 shows the second row of data fetched into storage.

Press END to return to the Edit screen.

Figure 16-62. Second Row Fetched Into Storage



Browsing the Results Table

To browse the DB2 results table generated by a SELECT or DECLARE CURSOR statement, do one of the following:

- From the Edit screen, replace the host variables with actual values, enter the BR line command on the SQL statement referencing the table to be browsed, then type **FADB2** on the command line and press Enter.

  Note: If you use this method, you are actually editing the SQL source. If you don't want to permanently replace the host variables, enter the CANCEL command before ending from the SQL Analysis Edit screen, or use the Host Variable Resolution screen as outlined below.

- From the Host Variable Resolution screen, enter option **1** for Browse after you enter the required host variable values.

You are taken directly to File-AID *for DB2* Browse for the results table (Figure 16-63). See Chapter 6, "Browse and Edit" for information on using the Browse function.

Press END to return to the ISPF/PDF SQL Source Development and Analysis screen from which you requested the Browse function. The SQL statement is marked with .BR in the line number area.

Figure 16-63. Browsing the Results Table

```
File-AID for DB2 -------- BROWSE - ABC245.SQLPROG ----------------------------
COMMAND ===>                                                  SCROLL ===> PAGE
         SSNR         LASTNAME         FIRSTNAME
         CHAR(11)     VARCHAR(15)      VARCHAR(15)
         -----------  ---------------  ---------------
******   *************************** TOP OF DATA ****************************
000001   111-23-5438  DOE              JOHN
000002   888-53-8867  CHAMBERLAIN      JANE
000003   151-04-8735  LINCOLN          FRED
000004   181-73-7496  KENNEDY          MIKE
000005   214-58-3869  ROGERS           ROBERT
000006   216-97-4975  ADAMS            DONALD
******   *************************** BOTTOM OF DATA *************************
```

Editing the Results Table

To edit the DB2 results table generated by a SELECT or DECLARE CURSOR statement, do one of the following:

- From the Edit screen, replace the host variables with actual values, enter the ED line command on the SQL statement referencing the table to be edited, then type **FADB2** on the command line and press Enter.

  Note: If you use this method, you are actually editing the SQL source. If you do not wish to permanently replace the host variables, enter the CANCEL command before ending from the SQL Analysis Edit screen, or use the Host Variable Resolution screen as outlined below.

- From the Host Variable Resolution screen, enter option **2** for Edit after you enter the required host variable values.

You are taken directly to File-AID *for DB2* Edit for the results table. See Chapter 6, "Browse and Edit" for information on using the Edit function. See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2 for more information on Edit display options.

Press END to return to the ISPF/PDF SQL Source Development and Analysis screen from which you requested the Edit function. The SQL statement is marked with .ED in the line number area.

Dynamically Changing and Testing SQL Statements

From the SQL Analysis Edit screen, you can dynamically change the SQL statement and execute it to see the results of the change.

For example, you can change the SQL DECLARE CURSOR statement to add the "ZIPCODE" variable (Figure 16-64).

Figure 16-64. Dynamically Changing an SQL Statement

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
00351     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNR, LASTNAME, FIRSTNAME, CITY, STATE, ZIPCODE
00353       FROM VSQLSAMP
00355       WHERE STATE = :STATE-NAME
00355       AND ZIPCODE = :ZIP-CODE
00356     END-EXEC.
00357     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQLCODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
.EX   EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00364     END-EXEC.
F16=FADB2
```

You can also enter values for the host variables on the Edit screen (Figure 16-65 on page 16-55). There you can dynamically execute, explain, browse, or edit the SQL statement, without being prompted to enter values on the Host Variable Resolution screen. If you use this method, you are actually editing the SQL source. If you do not want to permanently replace the host variables, enter the CANCEL command before ending from the Edit screen. In this example, the EX line command was entered to execute the DECLARE CURSOR statement.

Figure 16-65. Entering Host Variable Values on the Edit Screen

```
EDIT ----- COMPSQL.COBOL.SOURCE(SQLPROG) - 01.03 ------------ COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
***** *************************** TOP OF DATA ********************************
00350 SQL-PROTOTYPE-START.
EX 51     EXEC SQL DECLARE SAMPLE CURSOR FOR
00352       SELECT SSNR, LASTNAME, FIRSTNAME, CITY, STATE, ZIPCODE
00353       FROM VSQLSAMP
00355       WHERE STATE = 'FL'
00355       AND ZIPCODE = '33442'
00356     END-EXEC.
00357     EXEC SQL
00358       OPEN SAMPLE
00359     END-EXEC.
00360     PERFORM FETCH UNTIL SQLCODE NOT EQUAL ZERO.
00361 PROG-END.
00362     EXEC SQL
00363       CLOSE SAMPLE
00364     END-EXEC.
00365 FETCH.
.EX       EXEC SQL
00367       FETCH SAMPLE
00368         INTO :SSNR, :LASTNAME, :FIRSTNAME
00364     END-EXEC.
F16=FADB2
```

When the execution label (the line command preceded by a period) displays on the selected line, you can enter another line command to perform a different action or dynamically change and test other SQL statements.

Explaining SQL Statements

From the SQL Analysis Edit screen, you can dynamically explain SQL statements with the **XP** line command. Enter **XP** on the SQL statement you wish to explain, then type **FADB2** on the command line as shown in Figure 16-66 on page 16-56 and press Enter to process the command.

Note: To explain SQL, it must contain one of the following statements:

- DECLARE CURSOR
- DELETE
- INSERT
- SELECT
- UPDATE

Figure 16-66. Explaining SQL Statements with the XP Line Command

```
EDIT ---- USERID.DB2.SOURCE(SQLSAMP) - 01.00 ----------------- COLUMNS 001 072
COMMAND ===> FADB2                                            SCROLL ===> PAGE
000186            IF INPUT-LAST-NAME = SPACES
000187               MOVE '%' TO INPUT-LAST-NAME.
000188            UNSTRING INPUT-LAST-NAME
000189               DELIMITED BY SPACE
000190               INTO     LNAME-WORKC
000191               COUNT IN LNAME-WORKL.
000192       *-------------------------------------------------------------*
000193       * DEFINE SQL CURSOR FOR DIRECTORY TABLE                       *
000194       *-------------------------------------------------------------*
000195            EXEC SQL DECLARE DIRECTORY CURSOR FOR
.XP             SELECT *
000197                     FROM   VSQLSAMP
000198                     WHERE  LASTNAME  LIKE :LNAME-WORK
000199                       AND  FIRSTNAME LIKE :FNAME-WORK
000200                  END-EXEC.
000201            EXEC SQL OPEN  DIRECTORY                   END-EXEC.
000202            EXEC SQL FETCH DIRECTORY INTO :PDIRECTORY END-EXEC.
000203            IF SQLCODE = SQL-NOT-FOUND
000204         MOVE 'NO ENTRY FOUND IN DIRECTORY  ' TO RPT-ERROR-TEXT
000205         WRITE RPT-REC FROM RPT-ERROR-LINE AFTER ADVANCING 2 LINES
000206     ELSE
F16=FADB2
```

Press Enter as shown above to display the Plan Table Analysis screen (Figure 16-67). For information on using this screen, see “Plan Table Analysis” on page 16-19.

Figure 16-67. Plan Table Analysis Screen Resulting from an XP Line Command.

```
File-AID for DB2 -- Plan Table Analysis for: DB2AAAA -------------------------
OPTION  ===> 3                                              SCROLL ===> CSR

    1 - Access Path Analysis         4 - Object Information
    2 - Explain Detail               5 - Cost Detail
    3 - Predicate Analysis           6 - Function Detail

  DBRM/Package  Statement  Type     DB2 Cost    Access Type
  ------------  ---------  ------   --------    ----------------------
  DB2SQLXX        1698     SELECT       1250 *  Non-Matching Index Scan

SQL Statement  1 OF 84:
  DECLARE DBAUTH_CURSOR CURSOR FOR
  SELECT GRANTOR, GRANTEE, NAME, GRANTEETYPE, CREATETABAUTH, DBADMAUTH,
    DBCTRLAUTH, DBMAINTAUTH, DISPLAYDBAUTH, DROPAUTH, IMAGCOPYAUTH,
    LOADAUTH, REORGAUTH, RECOVERDBAUTH, REPAIRAUTH, STARTDBAUTH,
    STATSAUTH, STOPAUTH
  FROM SYSIBM.SYSDBAUTH
  WHERE COLUMN1 > :VALUE1
  AND COLUMN1 < :VALUE2



F10=PREVIOUS          F11=NEXT
```

Resolving SQL Coding Errors

If an SQL error is encountered, an error message displays in the message area and the action is aborted. For example, suppose you entered the ED (Edit Table command) on an SQL statement, but DB2 returned an SQL return code of -206 (Figure 16-68 on page 16-57).

Figure 16-68. Error Was Encountered During Execution of the SQL Statement

```
EDIT ---- FD.SHOW.SQL(GCO#1) - 01.02 -------------------------Columns 001 072
COMMAND ===>
ISRZ001 SQLCODE = -206, DOB
.ED     SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO,
000002    HIREDATE, DOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM
000003    INTO :EMPNO, :FIRSTNME, :MIDINIT, :LASTNAME, :WORKDEPT, :PHONENO,
000004    :HIREDATE, :JOB, :EDLEVEL, :SEX, BIRTHDATE, :SALARY, :BONUS, :COMM
000005    FROM FADB2610.EMP
000006    WHERE EDLEVEL = 17
000007    AND SEX = 'M'
000008    ORDER BY EMPNO;
******   *************************** BOTTOM OF DATA ************************
```

The HELP/TUTORIAL screen gives you information when you encounter an error. Type **HELP** and press Enter, or press **HELP** to access this screen. Entering **HELP** for the -206 return code gives you the Help/Tutorial (Figure 16-69). This Help screen provides the DB2 SQL error text returned from DB2.

Note: There are some instances while using SQL development and analysis that extended help may not be available. For more information about SQL error codes, refer to *IBM Database 2 Messages and Codes*.

Figure 16-69. Help Screen for an SQL Return Code of -206

```
             File-AID for DB2 - HELP / TUTORIAL
COMMAND ===>

DSNT408I SQLCODE: -206: DOB IS NOT A COLUMN OF AN INSERTED TABLE
         UPDATED TABLE, OR AN Y TABLE IDENTIFIED IN A FROM CLAUSE, OR IS NOT
         A COLUMN OF THE TRIGGERING TABLE OF A TRIGGER
DSNT418I SQLSATE    = 42703 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXORSO SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -100  0  0 -1  0  0 SQL DIAGNOSTICE INFORMATION
DSNT416I SQLERRD    = X'FFFFFF9C' X'00000000' X'00000000' X'FFFFFFFF'
         X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
```

Unqualified Objects Window

If you issued one of the FADB2 line commands, EX, XP, ED, or BR, and your SQL contains an unqualified object, this window displays pre-filled with the last saved profile variable value. If you leave the field blank, the SQLID will be used and stored in the profile.

Figure 16-70. Unqualified Objects WIndow

```
EDIT ---- USERID1.DEMO.CNTL(DEPTSQL)- 01.03 ------------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> PAGE
****** ****+---------------  Unqualified Objects  --------------+**************
000100  DEC|                                                    |DEPT FROM
000200  DEP| COMMAND ===>                      SQL ID: USERID1  |
000300  :DV|                                                    |
000400  AND| Selected objects are unqualified, i.e. Tables,     |
000500  :DV| Views and Alias's. If not specified they will      |
000600  GRO| default to the SQL ID.                             |
000700   DE|                                                    |
****** ****| Specify a default qualifier:                       |**************
           |    Default Qualifier ===> USERID                   |
           |                                                    |
           | Press ENTER to continue, END to return.            |
           |                                                    |
           +----------------------------------------------------+




F4=FADB2
```

Modifying Column Boundaries

You can modify the column boundaries from which the FADB2 command obtains an SQL statement to be processed so the line numbers, comments, etc. are eliminated. Enter **FADB2 col1 col2** on the command line of the SQL Analysis Edit screen, where **col1** is the beginning boundary and **col2** is the ending boundary. You will receive the following message:

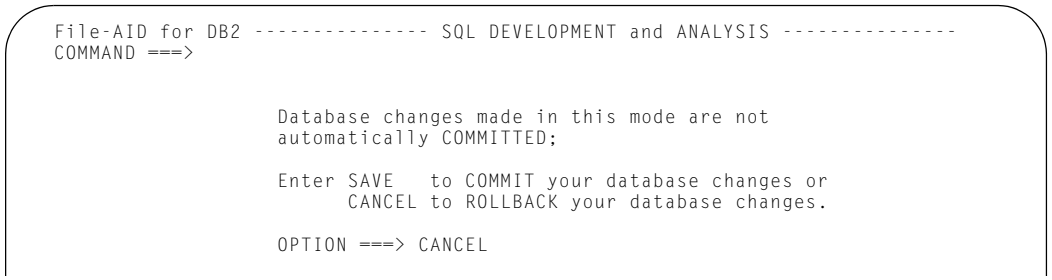
```
FADB2 BOUNDS ARE NOW SET FOR COLUMNS col1 TO col2
```

Any data outside those boundaries is ignored by the FADB2 line commands (BR, ED, EX, and XP).

COMMIT/ROLLBACK Prompt

If you executed any SQL while in SQL Analysis and Development, you receive the COMMIT/ROLLBACK prompt (Figure 16-71).

Figure 16-71. COMMIT/ROLLBACK Prompt



You may have database changes pending when you exit the Edit screen. You must either COMMIT or ROLLBACK those changes. Type **SAVE** to commit your database changes, or **CANCEL** to roll back your database changes. CANCEL is the default.

Your database changes are committed or rolled back to the last commit or rollback point.

Note: An SQL COMMIT or ROLLBACK statement may be executed within SQL Analysis.

Option L, LIBRARIAN Source Development and Analysis

The LIBRARIAN interface screen displays when **L** is selected on the SQL Analysis and Development Menu (Figure 16-2 on page 16-2). Depending on the manner in which File-AID *for DB2* is installed, the actual screen displayed may differ from site to site. Because of this variation, a sample screen is not shown.

Once the LIBRARIAN interface screen displays, select your SQL source as you normally would with this facility. Once the dataset has been displayed containing the SQL to be analyzed, you can perform all File-AID *for DB2* functions as described in "Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44.

Option P, PANVALET Source Development and Analysis

The PANVALET interface screen displays when **P** is selected on the SQL Analysis and Development Menu (Figure 16-2 on page 16-2). Depending on the manner in which File-AID *for DB2* is installed, the actual screen displayed may differ from site to site. Because of this variation, a sample screen is not shown.

Once the PANVALET interface screen displays, select your SQL source as you normally would with this facility. Once the dataset has been displayed containing the SQL to be analyzed, you can perform all File-AID *for DB2* functions as described in "Option 3, ISPF/PDF SQL Source Development and Analysis" on page 16-44.

Chapter 17.
Print Table Data or Audit Trail

This chapter describes how to use Option 5, Print Table Data or Audit Trail. It explains how you can use selection criteria to select the table data to be printed. This allows you to tailor your reports to your specific needs and have them printed in either table or row mode. Sample reports appear in Figure 17-11 on page 17-11 and Figure 17-12 on page 17-12.

This chapter also describes how an audit trail report is printed in File-AID *for DB2*. An audit trail report contains a record of all successful modifications (inserts, deletes, updates, commits and rollbacks) to DB2 table data during a File-AID *for DB2* edit session. A sample report appears in Figure 17-15 on page 17-15. For information on producing an audit trail dataset, see "Creating an Audit Trail Dataset" on page 6-31.

Note: If your installation chose the SMF audit trail option, please see *Chapter 5. Audit Trail Facility* in the *File-AID for DB2 Installation Guide* for printing instructions.

Print Menu

To access the Print Menu screen (Figure 17-1), select option **5** from the Primary Options menu (Figure 2-1 on page 2-1). In the Print Menu screen enter either Option **1** to print table data or Option **2** to print an audit trail report.

Figure 17-1. Print Menu Screen

```
File-AID for DB2 ----------------- Print Menu ---------------------------------
OPTION  ===>                                          SSID ===> DSN

   1 - Print table Data

   2 - Print Audit Trail
```

Printing Table Data

Enter option **1** from the Print menu to access the Print Options screen.

Print Options Screen

The Print Options screen (Figure 17-2 on page 17-2) is used to specify the DB2 table and the data selection criteria. You also indicate whether the table data is to be printed in row or table mode.

Figure 17-2. Print Options Screen

```
File-AID for DB2 ---------------- Print Options -------------------------------
COMMAND ===>                                                  SSID ===> DSN

Specify DB2 Table (* or pattern for list)  Optional:
   Creator     ===> USERID                   Database    ===>
   Table Name  ===> *                        Tablespace  ===>

Specify Print Template Information:
   Use Existing Criteria ===> NO       (Yes or No)
   Edit Criteria Dataset ===> PRINT.SQL
   Member                ===>          (Blank or Pattern for list)

Specify Print Options:
   Print Mode             ===> 1           (1 - Table, Multiple rows per screen
                                            2 - Row, One row per screen)
   Maximum Rows to Select ===> 2000        (* for all rows)
```

Specify DB2 Table Area

Creator

ID of the person who created the table or view you plan to print. Wild card characters (*****, _, %) are valid. See "Wildcard Characters" on page 2-9 for more information.

If you enter a wild card character in this field, Figure 17-4 on page 17-4 appears.

Table Name

Name of the object you want to edit. Tables, aliases, synonyms and views are valid. Wild card characters (*****, _, %) are also valid. See "Wildcard Characters" on page 2-9 for details.

If you enter a wild card character in this field, the Print Table Selection window (Figure 17-4 on page 17-4) appears.

Optional Area

Note: Specifying an optional database and/or tablespace is only in effect when the Creator field or the Table Name field contain an asterisk.

Database

Name of the database to which you want to confine your selection list.

Tablespace

Name of the tablespace to which you want to confine your selection list.

Specify Print Template Information Area

Use Existing Criteria

Enter whether you want to use a previously saved Edit/Print Selection Template dataset. Edit/Print Selection Criteria consists of conditions to limit and order column selection and/or row selection based on column values. This field is prefilled with the last saved entry (YES or NO).

NO Don't use a previously saved Edit/Print Selection Criteria dataset.

YES Apply the criteria stored in the dataset specified in the **Edit Criteria Dataset** field. Then you can verify and/or modify the selection criteria in the "PRINTP Selection Template" on page 17-5.

Edit Criteria Dataset

If you specified **Yes** in the **Use Existing Criteria** field, specify the name of the dataset that contains the print selection template criteria you want to use. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Note: The criteria dataset must contain a valid SELECT statement. File-AID *for DB2* only uses the first SELECT statement of the dataset. If a valid SELECT is missing, or the table name of the SELECT statement does not match the entered name in the above Table Name field, File-AID *for DB2* gives an error (see "Selection Template Not Prefilled" on page 17-7). When File-AID *for DB2* processes the SELECT statement to prefill the selection template, it ignores extraneous clauses such as WITH, OPTIMIZE, FOR FETCH, and CURRENT OF.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to use. If you leave this field blank or specify an asterisk (*****), File-AID *for DB2* displays the Member List for the dataset. The asterisk (*****) is the only valid pattern character.

Specify Print Options Area

Print Mode:

Enter the desired presentation mode for printing. This field is updated with the last mode in use.

1 Presents multiple rows per screen or page. Column names are listed across the top of the screen. This is called *table mode*.

2 Presents one row per screen or page. Column names are listed down the left side of the screen or page and column data is listed down the right side of the screen or page. This is called *row mode*.

See "Display Modes" on page 6-14 for more information.

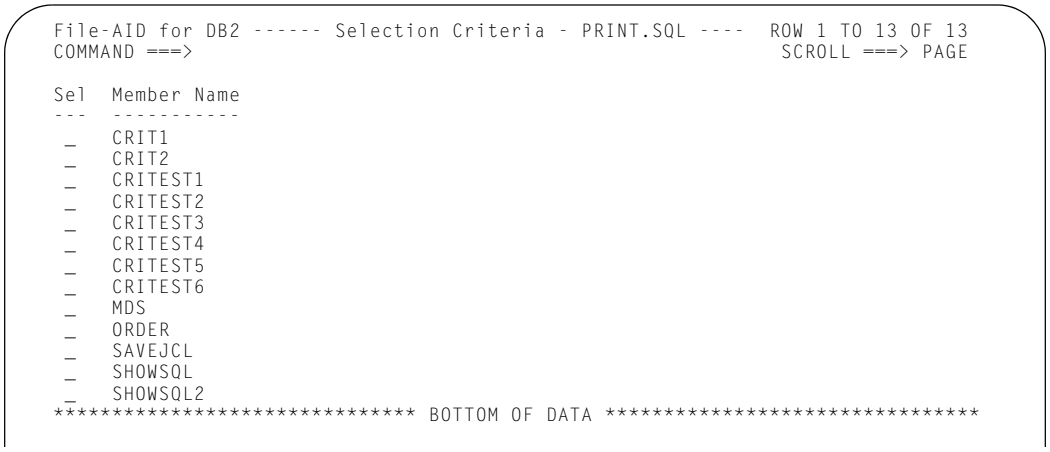
Maximum Rows to Select

Specify a numeric value indicating the maximum number of rows to be selected, or specify an asterisk (*****) to select all rows. The default value is retrieved from User Parms (Option 0). See "Option 1, Browse/Edit/SQL Analysis Display Options" on page 5-2 for information on changing the default. See "Maximum Rows to Select" on page 6-3 for more information on this field.

Selection Criteria Member List

The Selection Criteria Member List appears if you left the Selection Criteria Member field blank (or entered a *****) on the Edit Options screen. It provides a list of all members of the specified Selection Criteria dataset.

Figure 17-3. Print Selection Criteria Member List



Sel

Enter the **S** line command to select one of the listed members as your selection criteria.

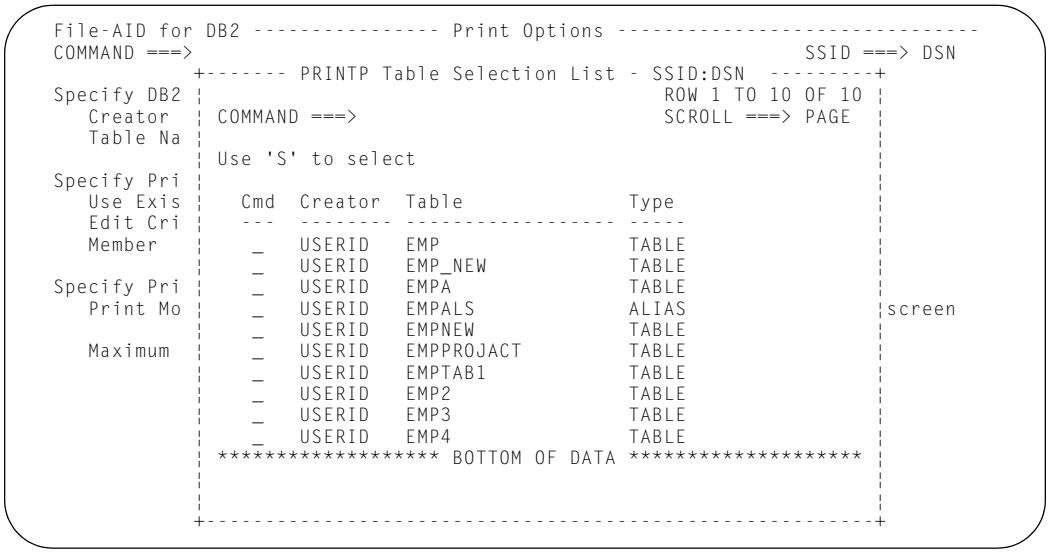
Member Name

Lists all member names of the selection criteria dataset.

Table Selection

The Table Selection window appears when you enter a wild card character in the Creator and/or Table Name field on the Print Options screen. It provides a list of tables or views to be selected for printing.

Figure 17-4. Print Table Selection Window



Cmd

Enter **S** to select the table or view.

Creator

Lists creator IDs of the tables to be selected.

Table

Lists the tables to be selected.

Type

Object type for each object listed.

PRINTP Selection Template

The PRINTP Selection Template screen appears after you complete the Print Options screen. On this screen you can specify the rows or columns of the object you want to print, the order in which the columns are printed, and the rows you want to use and the order in which they are printed.

If you are using an existing Edit/Print Criteria Dataset the SEL column is already prefilled with the selections from the Edit/Print Criteria Dataset. You can then either accept or modify them as desired.

Note: You achieve "Full Table Print" by NOT selecting any columns in the SEL column when entering the EXECUTE command.

Figure 17-5. Print Selection Template

```
File-AID for DB2 -------- PRINTP Selection Template ----------  ROW 1 TO 9 OF 14
COMMAND ===>                                                  SCROLL ===> PAGE
                                                                    SSID:  DSN
Use SQL to view/edit the SQL source for USERID.TABLENAME

Save Edit Criteria Information:
   Save Template Criteria On Execute  ===> NO       (Yes or No)
   Edit Criteria Dataset  ===> PRINT.SQL
   Member                 ===>

Max Rows to Select ===> *

SEL  COLUMN NAME        TYPE(LEN)        ORDER (A/D)  WHERE            More: >
---  ------------------ ------------------ --    -    ----+---10----+---20----
4    EMPNO              CHAR(6)            1     A    > '000110'
2    FIRSTNME           VARCHAR(12)
3    MIDINIT            CHAR(1)
1    LASTNAME           VARCHAR(15)
     WORKDEPT           CHAR(3)
     PHONENO            CHAR(4)
     HIREDATE           DATE
S    JOB                CHAR(8)                       >= 'FIELDREP' OR <= 'PRE'
     EDLEVEL            SMALLINT
F18=EXECUTE
```

Save Edit Criteria Information Area

Save Template Criteria On Execute

Enter **YES** to save the current Edit/Print Selection Criteria to a dataset.

Note: The selection criteria is saved when you enter the EXECUTE command on this screen. It is, however, NOT saved if you enter the SQL command and execute the SQL directly; The executed SQL is saved in the dataset as specified with "Allocate SHOW SQL Dataset" on page 5-5.

Enter **NO** (default) when you don't want to save the current Edit/Print Selection Criteria to a dataset.

Edit Criteria Dataset

Enter the name of the dataset you want the Edit/Print Selection Criteria to be saved to. File-AID *for DB2* prefills this field with the name of the last specified selection criteria dataset name.

Member

If the edit criteria dataset is a PDS, specify the member of the dataset you want to save the selection criteria to. If you leave this field blank or specify a *****, File-AID *for DB2* displays the Member List of the dataset. The only valid pattern character is an asterisk (*****).

Max Rows to Select

Maximum number of rows to be returned based on your selection criteria. See "Maximum Rows to Select" on page 6-3 for details.

SEL

Type **S** (select) beside the column or columns you want to edit. If you want the columns displayed in a particular order, type the appropriate integer beside each column. Mixing integers and "S" selections results in the integer-selected columns being displayed first, with the S-selected columns following. If you do not indicate any specific columns for selection, ***all*** columns are selected and presented in the order listed.

Column Name

Names of all columns within the table or view.

Type(Length)

Type and length of data in the column.

Distinct types are identified by their names.

Note: Columns with type LOB, or DISTINCT based on LOB do not appear on the template selection screen as they cannot be edited or browsed.

Order

Specify an integer to define how the returned rows should be sorted on the screen, based on the data content of the columns. Entering **1** indicates the primary sort column.

A/D

Specify whether the rows on this column data are sorted in ascending (A) or descending (D) order. If neither is specified, the columns are sorted in ascending order.

Where

Specify values limiting the rows displayed. The WHERE delimiter is analogous to the SQL WHERE clause in a SELECT statement. Up to 256 characters are available by scrolling right. You may use the following delimiters in your WHERE clauses:

>	OR	< >
<	> =	BETWEEN
=	<=	IN
¬ =	¬ >	LIKE
AND	¬ <	NOT

Note: Special considerations for DISTINCT (UDT) data type columns: File-AID *for DB2* generates the appropriate CAST function for simple WHERE clauses that contain just one of the following operators:

>	> =	< >
<	< =	¬ <
=	¬ =	¬ >

For all other WHERE clauses, you must enter the CAST function manually.

Selection Template Not Prefilled

The Selection Template screen cannot be prefilled when there is a mismatch between the table in the Use Existing Criteria and the one specified in the Table field of the Print Options screen, or the specified criteria does not contain a valid SELECT statement.

Figure 17-6. Print Selection Template Not Prefilled

```
File-AID for DB2 ------- PRINTP Selection Template ---- From tables don't match
COMMAND ===>                                              SCROLL ===> PAGE
                                                               SSID:  DSN
Use SQL to view/edit the SQL source for USERID.TABLENAME

Save Edit Criteria Information:
   Save Existing Criteria ===> NO       (Yes or No)
   Edit Criteria Dataset  ===> PRINT.SQL
   Member                 ===>

Max Rows to Select ===> 2000

SEL  COLUMN NAME        TYPE(LEN)        ORDER (A/D)  WHERE            More: >
---  ------------------ ------------------ --    -    ----+---10----+---20----
The existing selection criteria cannot be displayed in this format or
is in error.  To view or edit the criteria enter the SQL command. To
use the criteria enter the EXECUTE command.




F18=EXECUTE
```

Now you have several choices:

- Execute the SQL of the existing Selection Criteria Dataset. If it contains a valid SELECT statement you can then print the table specified in the selection criteria.
- Enter the SQL command to view and edit the SQL of the existing Selection Criteria Dataset before executing it.
- Return to the Print Options screen and select a different creator and table combination.
- Return to the Print Options screen and specify NO in the Use Existing Criteria field.
- Return to the Print Options screen and specify a different Edit/Print Criteria dataset member combination.

SQL Entry Panel

The SQL Entry Panel appears if you enter the **SQL** primary command on the Printp Selection Template screen whenever the screen could not be prefilled. It prompts you to name and describe a dataset. The dataset can have a previously saved SELECT statement or it can be empty, allowing you to enter a new SELECT statement. If the dataset contains

more than one SELECT statement, only the first statement is executed. Figure 17-7 shows a sample SQL Entry Panel.

Figure 17-7. SQL Entry Panel

```
File-AID for DB2 ------------------ SQL ENTRY PANEL ----------------- SSID: DSN
COMMAND ===>

Specify the input dataset name:         (Can be sequential or partitioned)
 1  DATASET NAME  ... ===> PRINT.SQL(MYSELECT)
 2  VOLUME SERIAL ... ===>              (Enter if not cataloged)
 3  DATASET PASSWORD  ===>              (Enter if password protected)
 4  MAX SELECT ROWS . ===> 2000         (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT ...... ===> YES          (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

Dataset Name

Name of the dataset (either partitioned or sequential) in which the SQL query resides or will reside.

Volume Serial

Disk pack on which the dataset resides, if it is not cataloged.

Dataset Password

Password for the dataset, if it is password protected.

Max Select Rows

Maximum number of rows to be returned for the SELECT. (An asterisk in this field indicates that all rows should be returned. See "Maximum Rows to Select" on page 6-3 for more information.) Only the first SELECT command in a dataset is executed.

Edit Input

Specify whether you want to edit the SQL before it is executed.

If the value of the Edit Input field is YES, pressing Enter displays the Edit SPUFI Selection screen (Figure 17-8 on page 17-9). If the value of the Edit Input field is NO, pressing Enter executes the SQL statement and displays the Edit mode screen.

Edit SPUFI Selection Screen

You access the Edit SPUFI selection screen from one of the following:

- SQL Entry Panel when you enter **YES** in the Edit Input field on the SQL Entry Panel.
- Prefilled EDIT Selection Template screen when you enter the **SQL** command. In that case the SQL is saved in the default SQL dataset as specified with "Allocate SHOW SQL Dataset" on page 5-5.

On this screen, you can enter a new SELECT statement or alter the existing one. The statement should be in standard SPUFI format.

Figure 17-8. Edit SPUFI Selection Screen

```
EDIT -------------- USERID.PRINT.SQL(MYSELECT) - COLUMNS  001 07
COMMAND ===>                                                SCROLL ===> PAGE
****** *************************** TOP OF DATA ******************************
000001 SELECT LASTNAME, FIRSTNAME, MIDINIT, EMPNO, JOB, BIRTHDATE, PHONENO,
000002   HIREDATE
000003   FROM USERID.EMP
000004   WHERE EMPNO > '000110'
000005   ORDER BY USERID.EMP.EMPNO, USERID.EMP.JOB;
****** ************************** BOTTOM OF DATA ******************************
```

Text Entry Area

Change or add to the SELECT statement using ISPF/PDF Edit. For more information concerning SPUFI, see the *IBM SQL Reference Guide*.

When you are satisfied with your SQL, press END to end the edit session.

- If you entered from the SQL Entry screen you return to the SQL Entry screen and are advised to press Enter to execute the SQL statement (Figure 17-9).
- If you entered from the PRINTP Selection Template screen and modified the displayed SQL, you are asked **Execute modified SQL?**. YES executes the SQL statement and you enter Edit or Browse; NO returns to the Selection Template screen.

Figure 17-9. SQL Entry Panel

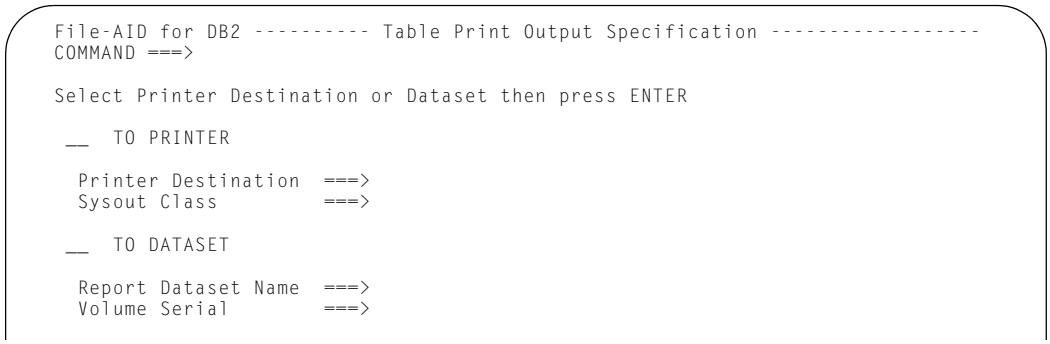
```
File-AID for DB2 ------------------ SQL ENTRY PANEL ----------------- SSID: DSN
COMMAND ===>
Edit session has completed. Press enter to execute SQL statement
Enter the input dataset name:           (Can be sequential or partitioned)
 1  DATASET NAME  ... ===> PRINT.SQL(MYSELECT)
 2  VOLUME SERIAL ... ===>              (Enter if not cataloged)
 3  DATASET PASSWORD  ===>              (Enter if password protected)
 4  MAX SELECT ROWS . ===> 2000         (Maximum number of rows to be
                                         returned from a SELECT)
 5  EDIT INPUT ...... ===> *            (YES to edit dataset,
                                         NO  to execute SQL immediately)
```

Table Print Output Specification Screen

The Table Print Output Specification screen (Figure 17-10 on page 17-10) appears when you complete the PRINTP Selection Template screen. You need to identify where File-AID *for DB2* directs the print output, either to a printer or a dataset. After you have specified the output destination, press ENTER.

File-AID *for DB2* generates a temporary JCL dataset which is presented on an Edit screen. Modify it as necessary, then submit the job to process the print request.

Figure 17-10. Table Print Output Specification



TO PRINTER Area

Enter the S command when you want the print output to go directly to a printing device.

Printer Destination

Use the LOCAL Printer ID or JES queue.

Sysout Class

Enter a valid sysout class.

TO DATASET Area

Enter the S command when you want the print output to be stored in a dataset.

Report Dataset Name

Enter the name of the designated output dataset.

Volume Serial

Enter the name of the disk pack on which the dataset is to reside.

Sample Row Mode Report

Figure 17-11. Example Row Mode Table Data Print to Dataset

```
File-Aid for DB2                                    TABLE PRINT REPORT                                    PAGE           1
                                                    DETAIL TABLE DATA                                      DATE  2000-07-12
                                                                                                         TIME 18.04.54



DB2 SUBSYSTEM: D31A
   TABLE/VIEW: FR.ORDER_TABLE


ROW SEQ NO:  1


---COLUMN NAME---- --FORMAT-- -KEY-- ----------------------------------------COLUMN VALUE----------------------------------------
ORDER_NUMBER       CHAR(6)    PRIM   AA2222
CUST_NUM           CHAR(6)           CN0001
ORD_TYPE           CHAR(2)           WS
ORD_STAT           CHAR(2)           01
PRIORITY_CODE      CHAR(1)           B
RD_DESCRIPTION    VARCHAR(40        INDUSTRIAL GRADE COAXIAL CABLE
** END OF ROW ***


ROW SEQ NO:  2


---COLUMN NAME---- --FORMAT-- -KEY-- ----------------------------------------COLUMN VALUE----------------------------------------
ORDER_NUMBER       CHAR(6)    PRIM   AA4444
CUST_NUM           CHAR(6)           CN0033
ORD_TYPE           CHAR(2)           HT
ORD_STAT           CHAR(2)           03
PRIORITY_CODE      CHAR(1)           A
ORD_DESCRIPTION    VARCHAR(40        HIGH PERFORMANCE COAX
*** END OF ROW ***



File-Aid for DB2                                    TABLE PRINT REPORT                                    PAGE           7
                                                    EXECUTION SUMMARY                                      DATE  2000-07-12
                                                                                                              TIME  18.04.54



DB2 SUBSYSTEM: D31A
   TABLE/VIEW: FR.ORDER_TABLE


             PRINT FORMAT: ROW
               HEX FORMAT: NO

             ROWS PRINTED: 20
          COLUMNS PRINTED: 6

TABLE LOCKED DURING PRINT: NO
        MAX ROWS TO PRINT: 20
        MAX COLS TO PRINT: 6
  KEY INFORMATION PRINTED: NO
```

Sample Table Mode Report

Figure 17-12. Example Table Mode Table Data Print to Database

```
 File-Aid for DB2 4.6.x.xx                          TABLE PRINT REPORT                                  PAGE          1
                                                    DETAIL TABLE DATA                                 DATE 2000-07-12
                                                                                                    TIME 14.50.10


     DB2 SUBSYSTEM: D31A
        TABLE/VIEW: FR      .ORDER_TABLE

  ROW     PRIORITY_CODE  ORDER_NUMBER  CUST_NUM  ORD_DESCRIPTION
 SEQ NO   CHAR(1)        CHAR(6)       CHAR(6)   VARCHAR(40)
-------  -------------  PRIMARY       --------  ----+---10----+---20----+---30----+---40
      1   B             AA2222        CN0001    INDUSTRIAL GRADE COAXIAL CABLE
      2   A             AA4444        CN0033    HIGH PERFORMANCE COAX
      3   C             BB1111        CN0044    VDT COVERS - BLACK
      4   B             BB3333        CN0001    RCA-BNC ADAPTERS
      5   B             BB9999        CN0033    52-PIN AMPHENOL CONNECTORS
      6   R             CC1111        CN0099    COAXIAL CABLE - BULK
      7   B             CC6666        CN0022    CRT TERMINAL - MONOCHROME
      8   B             DD4444        CN0001    WIRING HARNESS
      9   A             DD8888        CN0001    COILED TRANSISTOR
     10   C             EE2222        CN0044    SOLDERED WIRING GUIDE
     11   A             EE3333        CN0033    4800 BAUD DIAL-UP MODEM
     12   C             FF6666        CN0001    CONTROL UNIT
     13   B             FF7777        CN0001    LASER PRINTER
     14   B             GG1111        CN0025    4800 BAUD DIAL-UP MODEM
     15   B             GG7777        CN0035    768-C REMOTE PRINTER
     16   C             II2222        CN0033    PLOTTER PRINT JETS
     17   B             GG4444        CN0025    4800 BAUD DIAL-UP MODEM
     18   B             HH6666        CN0050    HIGH-PERFORMANCE MAG-TAPE
     19   C             HH8888        CN0022    PRINTER RIBBON CARTRIDGE
     20   B             II3333        CN0011    MICRO-DISK PACKS
     21   B             JJ4444        CN0099    LCD STATUS BOARDS
     22   B             JJ9999        CN0022    WWZ-20 MB HARD CARD
     23   F             KK5555        CN0011    LX-435 LASER JET PRINTER
     24   A             LL6666        CN0025    ACCESS DIODE
     25   C             NN4444        CN0011    DOUBLE STRUNG - COAX


 File-Aid for DB2 4.6.x.xx                          TABLE PRINT REPORT                                  PAGE          2
                                                     EXECUTION SUMMARY                                DATE 2000-07-12
                                                                                                    TIME 14.50.10


      DB2 SUBSYSTEM: D31A
         TABLE/VIEW: FR      .ORDER_TABLE


                   PRINT FORMAT:  TABLE
                     HEX FORMAT:  NO

                   ROWS PRINTED:  25
                COLUMNS PRINTED:  4

      TABLE LOCKED DURING PRINT:  NO
             MAX ROWS TO PRINT:  25
            MAX COLS TO PRINT:  4
       KEY INFORMATION PRINTED:  OFF
```

Printing an Audit Trail Report

To access the Print Audit Trail Specification screen, select option **5** from the Primary Options menu.

Print Audit Trail Specification Screen

On the Print Audit Trail Specification screen (Figure 17-13) you can specify the printer destination and sysout class for printing an audit trail dataset, as well as optional selection criteria. The name of the dataset and your job card information also appear on the screen for you to verify. Enter either **SUB** on this screen to submit the batch job or **JCL** to edit the JCL and manually submit the job. To exit this screen without printing the report, enter **END**.

Figure 17-13. Print - Audit Trail Facility Screen

```
File-AID for DB2 -------- Print - Audit Trail Specification ------- SSID: DSN
COMMAND ===>

Specify Audit Trail Dataset:
  Dataset Name ===> 'USERID.FADB2.D930208.T092123'

Specify Printer Characteristics:
 Printer Destination ===>
 Sysout Class        ===>
 Lines Per Page      ===> 55

Optional Selection Criteria ===> NO        (Yes or No)



Job Card Information:
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
 //USERIDX JOB ('ACCT INFO',ROOM),'NAME',
 //             CLASS=A,MSGCLASS=G,TIME=(1),NOTIFY=USERID
 //*
 //*

 Enter SUB to submit the batch job; JCL to edit the generated JCL
 Enter END or press PF3 to exit
```

Dataset Name

Specifies the audit trail dataset name. Defaults to the most current audit trail dataset. You may override the dataset name.

Printer Destination

Required. Enter the printer destination for the audit trail report.

Sysout Class

Required. Enter the sysout class for the audit trail report.

Lines Per Page

Required. Enter the maximum number of lines to print per page.

Optional Selection Criteria

Specify whether the Audit Trail Selection Criteria screen appears.

NO

Submits the job without selection criteria.

YES

Displays the Audit Trail Selection Criteria screen (Figure 17-14 on page 17-14).

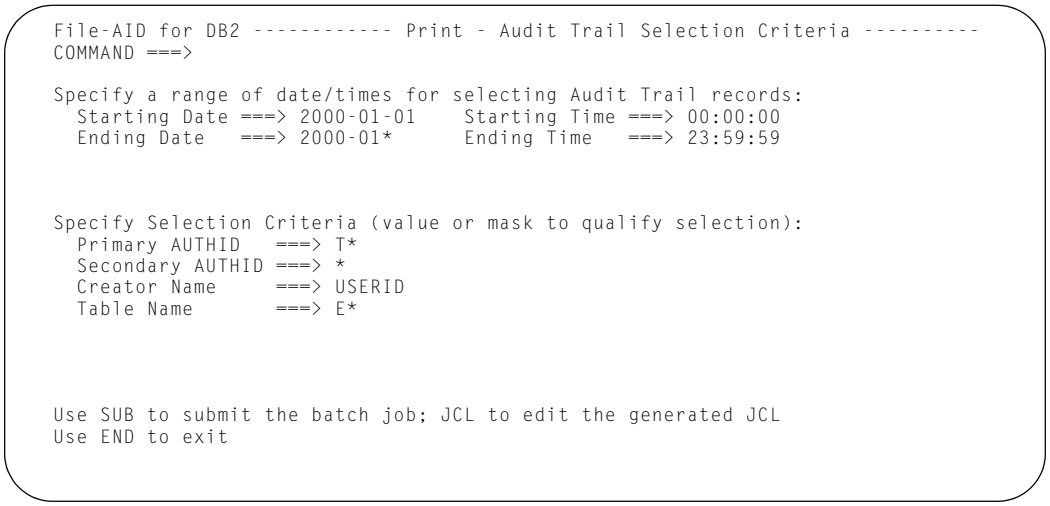
Job Card Information

Enter a valid job card. See "Option 3, Batch Options" on page 5-6 for entering a default job card.

Print Audit Trail Selection Criteria Screen

The Print Audit Trail Selection Criteria screen (Figure 17-14 on page 17-14) appears when **YES** is entered in the OPTIONAL SELECTION CRITERIA field on the Print Audit Trail Specification screen. You can print selected records based on starting and ending dates, starting and ending times, primary AUTHID, secondary AUTHID, creator name, and table name. Since a separate dataset is created for every audit trail generated, this option is most useful if you have combined your audit trail datasets. Enter **SUB** to submit the JCL. Enter **JCL** to edit the JCL and submit manually. Enter END to exit without printing the report.

Figure 17-14. Print — Audit Trail Selection Criteria



Starting/Ending Date

Enter a range of dates for selecting audit trail records. Dates default to the present date. A wild character (*) is valid at the end of a partial date.

Starting/Ending Time

Enter a range of times for selecting audit trail records. Starting time defaults to 00:00:00. Ending time defaults to 23:59:59. A wild character (*) is valid at the end of a partial time.

Primary AUTHID

Enter a primary AUTHID. A wild character (*) is valid at the end of a partial string.

Secondary AUTHID

Enter a secondary AUTHID. A wild character (*) is valid at the end of a partial string.

Creator Name

Enter a creator name. A wild character (*) is valid at the end of a partial string.

Table Name

Enter a table name. A wild character (*) is valid at the end of a partial string.

Sample Audit Trail Report

Note: Whenever a character field contains a non-displayable character the hexadecimal representation for that field is always printed on the Audit Trail Report.

Figure 17-15. Example Audit Trail Report

```
OBJECT NAME : USERID.EMP                                  DATE - 02/08/2000
OBJECT TYPE : TABLE                                       Time - 09:21:23
LOCATION :
CREATOR  :                                  PRIMARY AUTHID   : USERID
NAME     :                                  SECONDARY AUTHID :

***
ACTION : UPDATE

COLUMN NAME          KY  TYPE         OLD DATA          NEW DATA
-------------------  --  -----------  ---------------   ------------------
EMPNO                PK  CH(6)        000020
FIRSTNME                 VC(12)       MICHAEL
LASTNAME                 VC(15)       THOMPSON
PHONENO                  CH(4)        3476
HIREDATE                 DATE         1973-10-10
JOB                      CH(8)        MANAGER
                                      DCDCCCD           01234545
                                      4151759           49953369
SALARY                   DEC(9,2)     41250.00          45100.00
BONUS                    DEC(9,2)       800.00           1000.00
COMM                     DEC(9,2)      3300.00              0.00

***
ACTION : COMMIT                       DATE - 02/08/1993, TIME - 09:23:10
***
EDIT SESSION TERMINATED               DATE - 02/08/1993, TIME - 09:23:12

*** COMMENTS :
USERID.EMP has been edited
```

Chapter 18.
Relationship Facilitator

This chapter describes Option 6, Relationships - Relationship Facilitator.

Overview

The Relationship Facilitator enables you to view and manipulate the relationship file to:

- view all relationships defined to File-AID *for DB2*.
- load DB2-defined referential integrity (RI) relationships from the DB2 catalog table, SYSIBM.SYSRELS, into the relationship file.
- refresh the relationship file when its RI information does not match the DB2 catalog.
- delete RI and user-defined application relationship information from the relationship file.
- add application relationship information to the relationship file.
- modify existing application relationship information.
- print formatted contents of a relationship file.

Relationship File Compatibility Notes

1. **Relationship files can be shared** between File-AID *for DB2* and File-AID/RDX. Relationship files created or converted with File-AID/RDX Release 2.0, or higher, or File-AID *for DB2* Release 3.9.1, or higher, are Version 2 relationship files and are not downward compatible with earlier product releases.
2. **Version 1 Relationship files**, those created with earlier releases of either product, must be converted to a Version 2 relationship file. See Appendix C of the File-AID *for DB2* Installation Guide for more information on the creation and conversion procedures.
3. **DB2 Version 8 long name support** requires File-AID/RDX Release 4.3, or higher, or File-AID *for DB2* Release 4.8.1, or higher. Once a DB2 long name has been added to a Version 2 relationship file, File-AID/RDX updates it to a Version 3 relationship file. Version 3 relationship files are not downward compatible with earlier product releases.

Relationship File Contents

The relationship file (relationship repository) is a VSAM-KSDS file in which File-AID *for DB2* stores the following types of information:

DB2 Referential Integrity (RI) relationships

This relationship is one in which the dependent table rows contain foreign keys, the values of which are the primary index of the parent table. The rules that define RI relationships are enforced by DB2.

A batch job run during installation builds the initial relationship file using information collected from the DB2 catalog table, SYSIBM.SYSRELS, about referential integrity (RI) relationships in your subsystem, if your site installed File-AID *for DB2* with the DB2 portion. You can run this batch job whenever you need to refresh the relationship file.

Refer to "Relationship File Utilities" on page C-1 in the File-AID *for DB2* Installation Guide for more information.

Application Relationships (AR)

This relationship is one in which values from dependent MVS file fields, IMS database root segments, or DB2 table or view columns match the values in corresponding parent MVS file fields or DB2 table or view columns. The rules that define application relationships are enforced by user applications, not by DB2 or other databases.

To define an application relationship File-AID *for DB2* needs to know the parent and dependent object names along with the names of the related column(s) or field(s). File-AID *for DB2* can assist you with the application relationship definition through its SUGGEST feature: File-AID *for DB2* will suggest relationships when column or field name, type, and length in the parent and dependent objects are identical.

When defining an application relationship for an MVS object, you must always identify the specific record layout or XREF information that defines the related field within the file.

In addition, you can define relationships in File-AID *for DB2*, version 3.5 and above, and store them in the same relationship file you use with File-AID/RDX. You can view and modify these relationships using File-AID/RDX Option 1. File-AID *for DB2*, however, does not support application relationships involving IMS or MVS objects in its Related Edit/Browse function.

Considerations for MVS Objects

File-AID *for DB2* supports defining application relationships for MVS objects as either parent and/or dependent objects. Before defining a relationship keep in mind:

- file type must be SEQ, KSDS, or ESDS
- MVS object must be defined with a COBOL or PLI record layout
- only one record layout per object in a relationship file (layout usage S); if more than one is required, use XREF (layout usage X)
- only one XREF file per object in a relationship file
- the LRCL must be less than or equal to 32,756 in order for the data records to be extracted

Use of DB2 Views

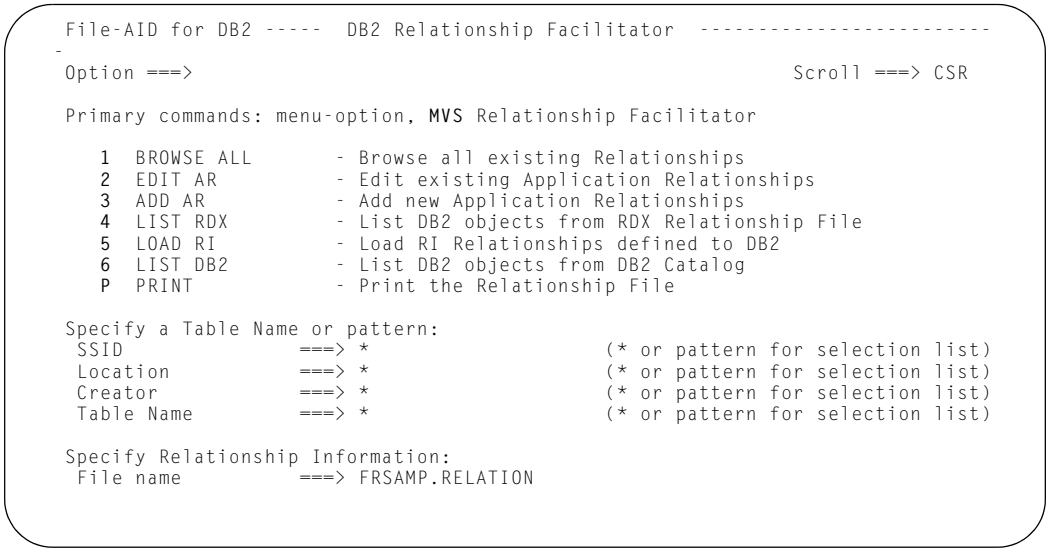
File-AID *for DB2* supports defining application relationships for views in the suboptions of the Relationships Options function (Primary Menu Option 6). Views are treated just like tables and can be entered in any **Table** field. For example, you can define an application relationship for a view to any other dependent object (MVS, DB2 or IMS).

When you specify a view in either Relationships suboption, File-AID *for DB2* will display all relationships defined for the view. If no relationships are defined for the view, File-AID *for DB2* will not display any relationships. To see the relationships for the view's base table you must enter the base table name.

DB2 Relationship Facilitator

The "DB2 Relationship Facilitator" screen shown in Figure 18-1 is displayed when you select Option 6, Relationships - Relationship Facilitator, from the File-AID *for DB2* Primary Option Menu. The "DB2 Relationship Facilitator" screen also appears when you enter the **DB2** command in the "MVS Relationship Facilitator" screen.

Figure 18-1. DB2 Relationship Facilitator



Use this screen to specify the name of the table or view for which you want to maintain relationships and the File-AID *for DB2* relationship file that contains the information about those relationships. Then enter one of the menu options to specify the action you want to perform. When pressing Enter without selecting an option, File-AID *for DB2* selects option 4, List DB2, and lists all matching entries found in the relationship file. File-AID *for DB2* issues a message when the relationship file does not contain any entries for the specified DB2 object.

Menu Options

The following File-AID *for DB2*-specific menu options are valid on this screen:

1 - Browse All

Browse Relationship File. Select Option 1 to search the relationship file for previously stored relationships that match the specified object.

Option 1, Browse All, enables you to view a list of all objects defined to your relationship file that are "related" to the object (MVS file or DB2 table or view) you specify. You can then select a single relationship about which you want to view detailed information from the list. See "Relationship Detail" on page 18-15 for more detail.

When browsing existing relationships you can also modify, delete, and add new application relationships.

2 - Edit AR

Edit Application Relationships. Select Option 2 to search the relationship file for previously stored application relationships.

Option 2, Edit AR, enables you to view a list of all objects defined to your relationship file that are "related" to the object (MVS file or DB2 table or view) you

specify. From the list, you can then select a single relationship about which you want to view detailed information. See "Application Relationship Detail" on page 18-31 for more detail.

3 - Add AR

Add Application Relationship. Select Option 3 to add an application relationship. Displays the Add Relationships — Specify Object Types so you can specify the parent and dependent object types for the new relationship. See "Add Relationships — Specify Object Types" on page 18-34.

4 - List RDX

(Default) List all DB2 objects from the Relationship File. Select Option 4 to search the relationship file for matching DB2 objects.

Option 4, List RDX, enables you to view a list of all DB2 objects defined to your relationship file that "match" the object specification. See "DB2 Object List" on page 18-5.

5 - Load RI

Select Option 5 to maintain the relationship file for DB2 RI information. Option 5 enables you to load or refresh DB2-defined RI information in your relationship file using information from the DB2 catalog. In addition, you can delete RI relationship data from the relationship file.

6 - List DB2

Search the DB2 Catalog and Relationship File. Select Option 6 to find objects in the DB2 catalog that match your specified pattern.

Option 6 enables you to view a list of all objects in the DB2 catalog that match the specified pattern. It also checks the relationship file and informs you whether it includes an entry for the object. See "DB2 Object List" on page 18-5.

P - Print

Print Relationships. Enables you to print a formatted report that contains information from the File-AID *for DB2* relationship file that you specify. You can also save this report to an existing or new file.

Specify a Table Name or Pattern Area

Use the fields in this area to specify the DB2 object for which you want to retrieve relationship information. When specifying patterns in the fields, File-AID *for DB2* presents selection lists. These fields don't apply for the P, Print Relationship file, menu option.

SSID

Specify the DB2 subsystem ID of the DB2 subsystem on which the table resides for which you want to view relationship information.

Location

Specify the DB2 remote location ID of the DB2 subsystem on which the table resides for which you want to view relationship information. Leave blank if your site does not use DDF.

Special Notes for SSID and Location fields

When you enter a pattern and select option 1, 2, or 4 at the same time File-AID *for DB2* will search the relationship file for matching entries.

For option 3, 5, or 6 you must select the one subsystem ID in the SSID field that is valid for the remote location ID you specify/select in the Location field.

Creator

Specify the authorization ID of the owner of the table for which you want to view relationship information. To view a list of tables, specify a pattern. Valid pattern

characters are %, _, and ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) for which you want to see relationship information. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Notes for Creator and Table fields

Which list File-AID *for DB2* will display depends on the option you select:

1
Table List in Relationship File

2
Table List in Relationship File

3
Table List - SYSIBM.SYSTABLES

4
DB Object List

5
Table List - SYSIBM.SYSTABLES

6
DB2 Object List

If you specify a pattern and press the Enter key without selecting an option, the DB2 Object List screen shown in Figure 18-2 on page 18-6 is displayed. See "DB2 Object List" on page 18-5 for more information. In the other table lists, select the desired Creator.Table combination with the S line command.

Specify Relationship File Area

File Name

Specify the name of the File-AID *for DB2* relationship file that contains the relationship information you want to display or maintain. Refer to "Relationship File Contents" on page 18-1 for more information.

Primary Commands

The following File-AID *for DB2*-specific primary commands are valid on this screen:

MVS

Switch to the "MVS Relationship Facilitator" panel to specify an MVS driving object for your Relationship task.

DB2 Object List

The "DB2 Object List" screen shown in Figure 18-2 is displayed when you select Option 4, List RDX, or Option 6, List DB2, and specify a pattern in the **Creator** or **Table Name** field on the "DB2 Relationship Facilitator" screen, Figure 18-1 on page 18-3.

Figure 18-2. DB2 Object List

```
File-AID for DB2 --------  DB2 Object List  ---------------- Row 1 to 6 of 6
Command ===>                                                 SCROLL ===> CSR

Line Commands: A = Add Application Relationship    B = Browse Relationship
               E = Edit Application Relationship   L = Load RI Relationships

                                                  Relation
CMD            Object Name                          Type    SSID
---  ------------------------------------  ------   ----
 _   DSNLOCAN.TSOID01.CONTACT_TABLE           RI      DSN
 _   DSNLOCAN.TSOID01.CUSTOMER_TABLE          AR/RI   DSN
 _   DSNLOCAN.TSOID01.ORDER_LINE_TABLE        RI      DSN
 _   DSNLOCAN.TSOID01.ORDER_TABLE             AR/RI   DSN
 _   DSNLOCAN.TSOID01.PART_TABLE              RI      DSN
 _   DSNLOCAN.TSOID01.SUPPLIER_TABLE          CATLG   DSN
****************************** Bottom of data ********************************
```

This screen displays a list of objects retrieved from either the DB2 catalog (Option 6) or the relationship file (Option 4). It also displays whether an object is already involved in relationships defined in the relationship file (AR or RI) or not (CATLG). If there are more objects that match the pattern you specified on the previous screen than will fit on the screen, you can scroll the list using standard scrolling commands. Enter one of the available line commands for an object.

CMD

Enter a line command in this column next to the object for which you want to maintain relationships. The available line commands are shown in the top part of this screen.

Object Name

Displays the names of either DB2 or MVS objects that match the value specified in the Creator and Table name fields on the preceding screen.

Relation Type

Displays the Relationship Type of the displayed objects.

AR	One or more Application Relationships exist for this object in the relationship file.
RI	One or more DB2 RI relationships exist for this object in the relationship file.
AR/RI	Both AR and RI exist for this object in the relationship file.
catalg	There are no relationships defined for this object in the relationship file.

SSID

Displays the ID of the DB2 subsystem from which the relationship had been defined or loaded.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen:

A

Add Application Relationship. Enter the A line command to add an application relationship. Displays the Add Relationships — Specify Object Types so you can

specify the parent and dependent object types for the new relationship. See “Add Relationships — Specify Object Types” on page 18-34.

B

Browse Relationship File. Enter the B line command to search the relationship file for previously stored relationships.

B enables you to view a list of all objects defined to your relationship file that are “related” to the object (MVS file or DB2 table or view) you specify. You can then select a single relationship about which you want to view detailed information from the list. See “Relationship Detail” on page 18-15 for more detail.

The message "**Selection not viewable**" appears if no relationships exist in the relationship file for the object (Relation Type CATLG).

E

Edit Application Relationships. Enter the E line command to search the relationship file for previously stored application relationships for the selected object.

E displays a list of all application relationships defined to your relationship file that are “related” to the selected object (MVS file or DB2 table or view). You can then maintain (modify, add, delete) application relationships or display more detail information about any of the application relationships from the list.

The message "Selection not viewable" appears if no relationships exist in the relationship file for the object (Relation Type CATLG).

The message "Can’t edit RI relationship" appears if only RI relationships exist in the relationship file for the object (Relation Type RI).

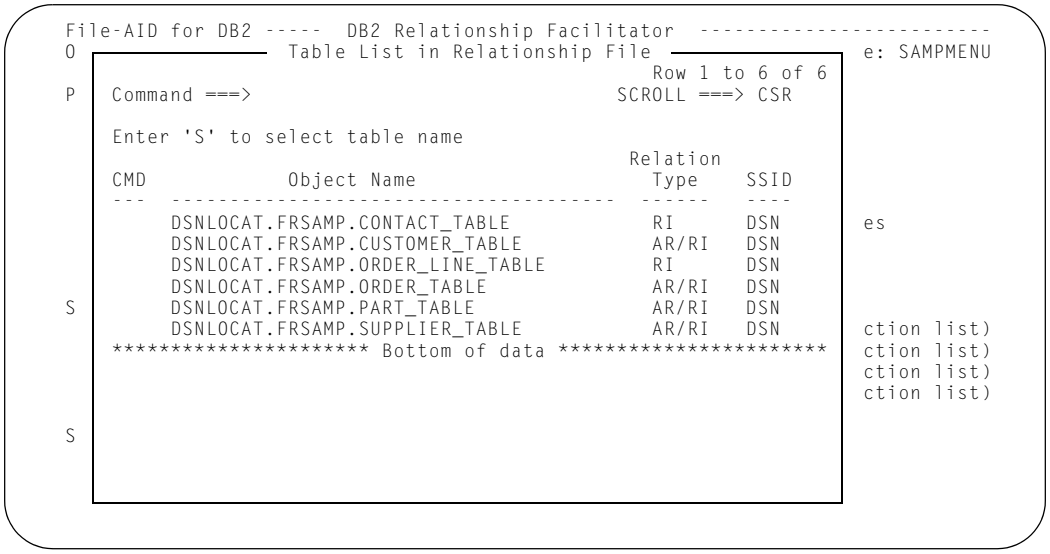
L

Load RI. Enter the L line command to maintain the relationship file for DB2 RI information for the selected object. L enables you to load or refresh DB2-defined RI information in your relationship file using information from the DB2 catalog. In addition, you can delete RI relationship data from the relationship file.

Table List in Relationship File

The "Table List in Relationship File" screen shown in Figure 18-3 is displayed when you select Option 1, Browse All, or Option 2, Edit AR, and specify a pattern in the **Creator** or **Table Name** field on the "DB2 Relationship Facilitator" screen, Figure 18-1 on page 18-3.

Figure 18-3. Table List in Relationship File



This screen displays a list of objects involved in relationships defined in the relationship file. If there are more relationships that match the pattern you specified on the previous screen than will fit on the "Table List in Relationship File" window, you can scroll the list using standard scrolling commands. Use this window to select the object for which File-AID *for DB2* will search the relationship file. The resulting list of relationships associated with the selected object is displayed on the "Relationship Detail" screen.

CMD

Enter an S line command in this column next to the object for which you want to see a list of relationships. The S line command is the only valid line command for this screen.

Object Name

Displays the names of DB2 objects that match the value specified in the "DB2 Relationship Facilitator" screen.

Relation Type

Displays the Relationship Type of the displayed objects.

AR	One or more Application Relationships exist for this object in the relationship file.
RI	One or more DB2 RI relationships exist for this object in the relationship file.
AR/RI	Both AR and RI exist for this object in the relationship file.

SSID

Displays the ID of the DB2 subsystem from which the relationship had been defined or loaded.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen:

S (Select)

Use the S line command to select an object for which you want to display relationship information. Relationship information about the object you select is displayed on the "Relationship Detail" screen as shown in Figure 18-8 on page 18-15.

Table List - SYSIBM.SYSTABLES

The "Table List - SYSIBM.SYSTABLES" screen shown in Figure 18-4 is displayed when you select Option 3, Add AR, or Option 5, Load RI, and specify a pattern in the **Creator** or **Table Name** field on the "DB2 Relationship Facilitator" screen, Figure 18-1 on page 18-3.

Figure 18-4. Table List - SYSIBM.SYSTABLES

```
File-AID for DB2 -----  DB2 Relationship Facilitator  -------------------------
O ┌──────────────────  Table List - SYSIBM.SYSTABLES  ──────────────────┐  e: SAMPMENU
  │                                                  Row 1 to 12 of 12 │
P │ Command ===>                                    SCROLL ===> CSR    │
  │                                                                    │
  │ Enter 'S' to select table name                                     │
  │                                                  Relation          │
  │ CMD            Object Name                         Type    SSID    │
  │ ---  ------------------------------------------  ------   ----    │
  │      DSNLOCAT.FRSAMP.CONTACT_TABLE                RI      DSN     │ es
  │      DSNLOCAT.FRSAMP.CONTACT_VIEW                 CATLG   DSN     │
  │      DSNLOCAT.FRSAMP.CUSTOMER_TABLE               AR/RI   DSN     │
  │      DSNLOCAT.FRSAMP.CUSTOMER_VIEW                CATLG   DSN     │
S │      DSNLOCAT.FRSAMP.ORDER_LINE_TABLE             RI      DSN     │
  │      DSNLOCAT.FRSAMP.ORDER_LINE_VIEW              CATLG   DSN     │ ction list)
  │      DSNLOCAT.FRSAMP.ORDER_TABLE                  AR/RI   DSN     │ ction list)
  │      DSNLOCAT.FRSAMP.ORDER_VIEW                   CATLG   DSN     │ ction list)
  │      DSNLOCAT.FRSAMP.PART_TABLE                   AR/RI   DSN     │ ction list)
  │      DSNLOCAT.FRSAMP.PART_VIEW                    CATLG   DSN     │
S │      DSNLOCAT.FRSAMP.SUPPLIER_TABLE               AR/RI   DSN     │
  │      DSNLOCAT.FRSAMP.SUPPLIER_view                CATLG   DSN     │
  │ ********************** Bottom of data ***********************    │
  └────────────────────────────────────────────────────────────────────┘
```

This screen displays a list of DB2 objects defined in the DB2 catalog that match the pattern. If there are more objects that match the pattern you specified on the previous screen than will fit on the "Table List - SYSIBM.SYSTABLES" window, you can scroll the list using standard scrolling commands. Use this window to select the object for which File-AID *for DB2* will add an application relationship or load the RI information to the relationship file. The resulting list of relationships associated with the selected object is displayed on the "Relationship Detail" screen.

CMD

Enter an S line command in this column next to the object for which you want to see a list of relationships or add a relationship.The S line command is the only valid line command for this screen.

Object Name

Displays the names of DB2 objects that match the value specified in the "DB2 Relationship Facilitator" screen.

Relation Type

Displays the Relationship Type of the displayed objects.

AR	One or more Application Relationships exist for this object in the relationship file.
RI	One or more DB2 RI relationships exist for this object in the relationship file.
AR/RI	Both AR and RI exist for this object in the relationship file.
catalg	There are no relationships defined for this object in the relationship file. This status only applies when selecting option 3, Add AR.

SSID

Displays the ID of the DB2 subsystem from which the relationship had been defined or loaded.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen:

S (Select)

Use the S line command to select an object for which you want to display relationship information or add a relationship. When loading RI for the object you select the "The RI Relationship Detail screen shown in Figure 18-13 is displayed when you select the Load RI option or enter the L line command for a specified DB2 object." screen as shown in Figure 18-13 on page 18-25 is displayed. When adding an AR for the selected object, the "Object Type Selection Screen for Adding Application Relationship" screen as shown in Figure 18-17 on page 18-34 is displayed.

MVS Relationship Facilitator

The "MVS Relationship Facilitator" screen shown in Figure 18-5 is displayed when you enter the **MVS** command in the "DB2 Relationship Facilitator" screen.

Figure 18-5. MVS Relationship Facilitator

```
File-AID for DB2 -----  MVS Relationship Facilitator  -------------------------
Option ===>

Primary commands: menu-option, DB2 Relationship Facilitator

   1  BROWSE ALL       - Browse all existing Relationships
   2  EDIT AR          - Edit existing Application Relationships
   3  ADD AR           - Add new Application Relationships
   4  LIST RDX         - List MVS objects from RDX Relationship File
   P  PRINT            - Print the Relationship File


Specify a File Name or Pattern:
 File name             ===> 'TSOID01.FRSAMP.*'
 Record layout usage   ===> S          (S = Single; X = XREF;)
 Record layout file    ===>
 Member name           ===>            (Blank or pattern for member list)
 XREF file name        ===>
 Member name           ===>
 I/O exit name         ===>

Specify Relationship Information:
  File name            ===> FRSAMP.RELATION
```

Use this screen to specify the name of the MVS file for which you want to maintain relationships and the File-AID *for DB2* relationship file that contains the information

about those relationships. Then enter one of the menu options to specify the action you want to perform. When pressing Enter without selecting an option, File-AID *for DB2* selects option 4, List RDX, and lists all matching entries found in the relationship file. File-AID *for DB2* issues a message when the relationship file does not contain any entries for the specified MVS object.

Menu Options

The following File-AID *for DB2*-specific menu options are valid on this screen:

1 - Browse All

Browse Relationship File. Select Option 1 to search the relationship file for previously stored relationships that match the specified object.

Option 1, Browse All, enables you to view a list of all objects defined to your relationship file that are "related" to the object (MVS file or DB2 table or view) you specify. You can then select a single relationship about which you want to view detailed information from the list. See "Relationship Detail" on page 18-15 for more detail.

When browsing existing relationships you can also modify, delete, and add new application relationships.

2 - Edit AR

Edit Application Relationships. Select Option 2 to search the relationship file for previously stored application relationships.

Option 2, Edit AR, enables you to view a list of all objects defined to your relationship file that are "related" to the object (MVS file or DB2 table or view) you specify. From the list, you can then select a single relationship about which you want to view detailed information. See "Application Relationship Detail" on page 18-31 for more detail.

3 - Add AR

Add Application Relationship. Select Option 3 to add an application relationship. Displays the Add Relationships — Specify Object Types so you can specify the parent and dependent object types for the new relationship. See "Add Relationships — Specify Object Types" on page 18-34.

4 - List RDX

(Default) List all MVS objects from Relationship File. Select Option 4 to search the relationship file for matching objects.

Option 4, List RDX, enables you to view a list of all objects defined to your relationship file that "match" the object specification. See "MVS Object List" on page 18-12.

P - Print

Print Relationships. Enables you to print a formatted report that contains information from the File-AID *for DB2* relationship file that you specify. You can also save this report to an existing or new file.

Specify a File Name or Pattern Area

Use the fields in this area to specify the MVS file for which you want to retrieve or add relationship information from the relationship file. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

File Name

Specify the MVS file name of the object for which you want to view relationship information. To view a list of files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

If you leave this field blank and you specify a pattern in the Record layout file or XREF file name fields together with menu options 1 or 2, File-AID *for DB2* searches the relationship file for any objects with a matching layout or XREF definition.

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.
You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object for which you want to view relationship information. To view a list of layout files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS object. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS object for which you want to view relationship information. To view a list of XREF files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the MVS object. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the MVS object for which you want to view relationship information.

You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing above MVS file. Not needed to search the relationship file.

Specify Relationship File Area

Relationship File Name

Specify the name of the File-AID *for DB2* relationship file that contains the relationship information you want to display. Refer to "Relationship File Contents" on page 18-1 for more information.

Primary Commands

The following File-AID *for DB2*-specific primary commands are valid on this screen:

DB2

Switch to the "DB2 Relationship Facilitator" panel to specify a DB2 driving object for your Relationship task.

MVS Object List

The "MVS Object List" screen shown in Figure 18-6 is displayed when you select Option 4 and specify a pattern in the **File Name** field on the "MVS Relationship Facilitator" screen, Figure 18-5 on page 18-10.

Figure 18-6. MVS Object List

```
File-AID for DB2 ------------  MVS Object List  -------------- Row 1 to 6 of 6
Command ===>                                                SCROLL===> CSR

Line Commands: A = Add Application Relationships   B = Browse Relationship
               E = Edit Application Relationship

CMD            Object Name
---  ------------------------------------------------------
     TSOID01.FRSAMP.CONTACTT
     TSOID01.FRSAMP.CUSTOMER
     TSOID01.FRSAMP.ORDERLIN
     TSOID01.FRSAMP.ORDERTAB
     TSOID01.FRSAMP.PARTTABL
     TSOID01.FRSAMP.SUPPLIER
****************************** Bottom of data ********************************
```

This screen displays a list of objects retrieved from the relationship file (Option 4, List RDX). If there are more objects that match the pattern you specified on the previous screen than will fit on the screen, you can scroll the list using standard scrolling commands. Enter one of the available line commands for an object.

CMD

Enter a line command in this column next to the object for which you want to maintain relationships. The available line commands are shown in the top part of this screen.

Object Name

Displays the names of MVS objects that match the pattern specified in the **File Name** field on the preceding screen.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen:

A

Add Application Relationship. Enter the A line command to add an application relationship. Displays the Add Relationships — Specify Object Types so you can specify the parent and dependent object types for the new relationship. See “Add Relationships — Specify Object Types” on page 18-34.

B

Browse Relationship File. Enter the B line command to search the relationship file for previously stored relationships.

B enables you to view a list of all objects defined to your relationship file that are “related” to the object (MVS file or DB2 table or view) you specify. You can then select a single relationship about which you want to view detailed information from the list. See “Relationship Detail” on page 18-15 for more detail.

The message "**Selection not viewable**" appears if no relationships exist in the relationship file for the object (Relation Type CATLG).

E

Edit Application Relationships. Enter the E line command to search the relationship file for previously stored application relationships for the selected object.

E displays a list of all application relationships defined to your relationship file that are “related” to the selected object (MVS file or DB2 table or view). You can then maintain (modify, add, delete) application relationships or display more detail information about any of the application relationships from the list.

The message "**Selection not viewable**" appears if no relationships exist in the relationship file for the object (Relation Type CATLG).

The message "**Can’t edit RI relationship**" appears if only RI relationships exist in the relationship file for the object (Relation Type RI).

Object List in Relationship File

The “Object List in Relationship File” window shown in Figure 18-7 is displayed when you specify a pattern in the **MVS file name** field and selected Options 1, Browse All, or 2, Edit AR, on the “MVS Relationship Facilitator” screen, Figure 18-5 on page 18-10.

Figure 18-7. Object List in Relationship File

```
File-AID for DB2 -----  MVS Relationship Facilitator  -------------------------
O ┌──────────────── Object List in Relationship File ──────────────────┐ e: SAMPMENU
  │                                                  Row 1 to 6 of 6 │
P │ Command ===>                                 SCROLL ===> CSR    │
  │                                                                 │
  │ Enter 'S' to select table name                                  │
  │                                                                 │
  │ CMD            Object Name                                      │
  │ ---  ------------------------------------------------------     │
  │  _   TSOID01.FRSAMP.CONTACTT                                    │
  │  _   TSOID01.FRSAMP.CUSTOMER                                    │
  │  _   TSOID01.FRSAMP.ORDERLIN                                    │
S │  _   TSOID01.FRSAMP.ORDERTAB                                    │
  │  _   TSOID01.FRSAMP.PARTTABL                                    │
  │  _   TSOID01.FRSAMP.SUPPLIER                                    │
  │ ********************** Bottom of data ***********************  │
  │                                                                 │  list)
  │                                                                 │
  │                                                                 │
  │                                                                 │
  │                                                                 │
S │                                                                 │
  └─────────────────────────────────────────────────────────────────┘
```

This screen displays a list of objects involved in relationships defined in the relationship file. If there are more relationships that match the pattern you specified on the previous screen than will fit on the “Object List in Relationship File” window, you can scroll the list using standard scrolling commands. Use this window to select the object for which File-AID *for DB2* will search the relationship file. The resulting list of relationships associated with the selected object is displayed on the “Relationship Detail” screen.

CMD

Enter an S line command in this column next to the object for which you want to see a list of relationships. The S line command is the only valid line command for this screen.

Object Name

Displays the names of MVS objects that match the value specified in the MVS File name field on the Process MVS Objects screen.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen:

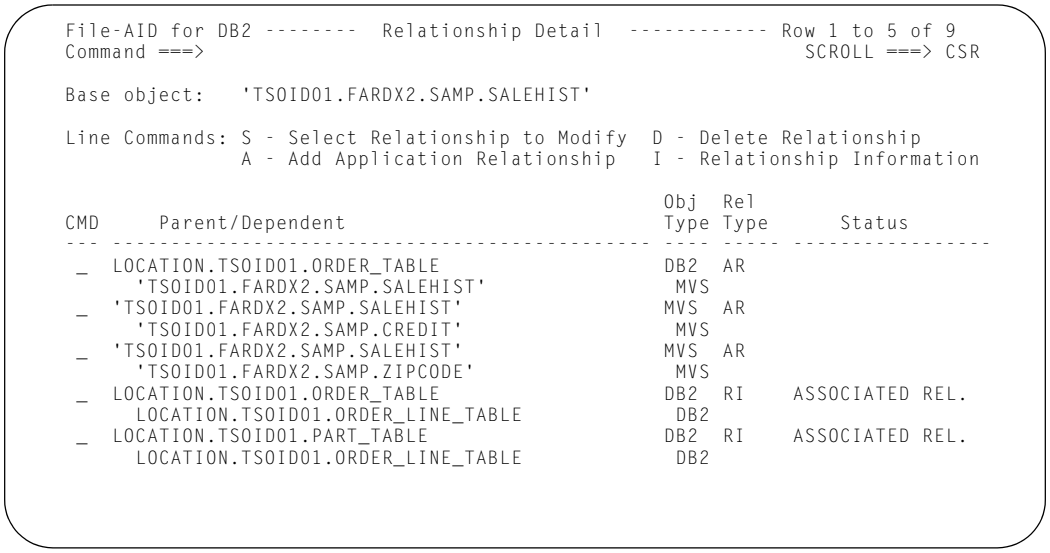
S (Select)

Use the S line command to select an object for which you want to display relationship information. Relationship information about the object you select is displayed on the “Relationship Detail” screen as shown in Figure 18-8 on page 18-15.

Relationship Detail

The “Relationship Detail” screen shown in Figure 18-8 is displayed when you selected Options 1, Browse All, or 2, Edit AR, on the DB2 or MVS Relationship Facilitator screen for a single object, or entered the B or E line command in the DB2 or MVS Object List, or after selecting a single object from a table or object list window.

Figure 18-8. Relationship Detail



```
File-AID for DB2 --------  Relationship Detail  ------------ Row 1 to 5 of 9
Command ===>                                                  SCROLL ===> CSR

Base object:   'TSOID01.FARDX2.SAMP.SALEHIST'

Line Commands: S - Select Relationship to Modify  D - Delete Relationship
               A - Add Application Relationship   I - Relationship Information

                                                   Obj  Rel
CMD     Parent/Dependent                           Type Type      Status
--- ---------------------------------------------- ---- ----- ----------------
 _  LOCATION.TSOID01.ORDER_TABLE                   DB2  AR
      'TSOID01.FARDX2.SAMP.SALEHIST'                MVS
 _  'TSOID01.FARDX2.SAMP.SALEHIST'                 MVS  AR
      'TSOID01.FARDX2.SAMP.CREDIT'                  MVS
 _  'TSOID01.FARDX2.SAMP.SALEHIST'                 MVS  AR
      'TSOID01.FARDX2.SAMP.ZIPCODE'                 MVS
 _  LOCATION.TSOID01.ORDER_TABLE                   DB2  RI    ASSOCIATED REL.
      LOCATION.TSOID01.ORDER_LINE_TABLE             DB2
 _  LOCATION.TSOID01.PART_TABLE                    DB2  RI    ASSOCIATED REL.
      LOCATION.TSOID01.ORDER_LINE_TABLE             DB2
```

Use this screen to view a list of all relationships associated with the specified object and to select a relationship about which you want to display detailed information with the I line command.

Other line commands and the ADD primary command allow you to add, change or delete relationships.

The Relationship Detail screen displays a scrollable list of relationships involving the object specified with a previous Browse or Edit option or line command.

A relationship will appear more than once on this screen if there is more than one relationship between a given pair of objects.

CMD

Enter line commands in this column. See “Line Commands” on page 18-20 for information about commands that are valid for this screen.

Parent/

Displays the name of the object defined as the parent in a relationship in the upper row (the row with the CMD input field).

/Dependent

Displays the name of the object defined as the dependent in a relationship in the lower row, under its parent.

Note: For long object names, you scroll Parent and Dependent entries separately: Place the cursor in a Parent entry to scroll all Parent entries to the left or right. Place the cursor in a Dependent entry to scroll all Dependent entries to the left or right.

Obj Type

Indicates the object type of the parent and dependent.

IMS IMS database.

DB2 DB2 table.

MVS MVS file.

Rel Type

Indicates the kind of relationship that exists between the corresponding parent and dependent objects.

RI DB2-defined referential integrity.

AR User-defined application relationship. Created and maintained by the user.

Status

Indicates the condition of each relationship. This field contains a message only if there are inconsistencies in the relationship file or as a result of line or primary command processing. Refer to "Status" on page 18-27 for a list of possible messages.

Line Commands

The following File-AID *for DB2*-specific line commands are valid on this screen:

A (Add)

Adds an application relationship for this parent object. Displays the Add Relationships — Specify Object Types screen so you can specify the parent and dependent object types for the new relationship. See "Add Relationships — Specify Object Types" on page 18-34.

D (Delete)

Deletes an AR or RI relationship from the relationship file.

I (Relationship Information)

Invokes the Relationship Information screen, which displays detailed information about the selected RI or application relationship. See "Relationship Information Screen" on page 18-20 for more information on the contents of the screen.

S (Select)

Selects an AR relationship for modification and displays the Column/Field Relationships screen. Not valid for RI relationships. See "Building Column/Field Relationships" on page 18-56 for more information.

Primary Commands

The following File-AID *for DB2*-specific primary commands are valid on this screen:

ADD

Enter the ADD command in the COMMAND field to add an application relationship. Displays the Add Relationships — Specify Object Types so you can specify the parent and dependent object types for the new relationship. See "Add Relationships — Specify Object Types" on page 18-34.

GRAPHIC (G)

Enter the GRAPHIC command in the COMMAND field to invoke the Graphical Relationship Display screen and show RI and application relationships for the table specified on the Browse Relationship File screen. See "Graphical Relationship Display" on page 18-18 for more information on the Graphical Relationship Display screen. There are no operands associated with this command.

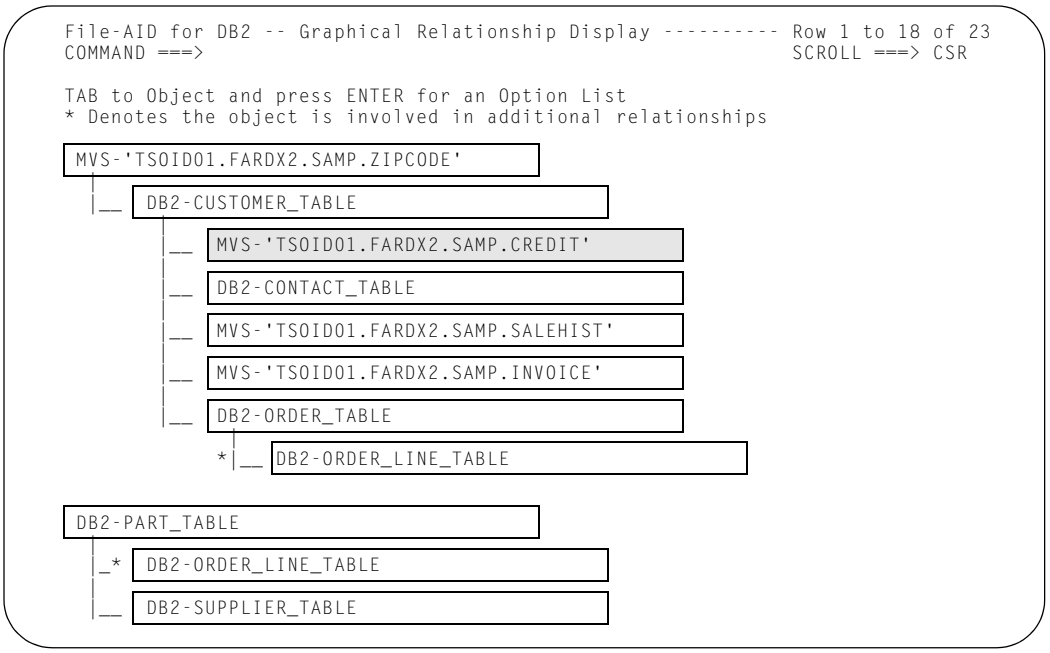
SQLID

Enter the SQLID primary command to change the current DB2 SQLID. Enter the SQLID command without an operand to change the current SQLID to match the current userid. Use the new-sqlid operand to specify any SQLID.

Graphical Relationship Display

The Graphical Relationship Display screen shown in Figure 18-9 is displayed when you enter the GRAPHIC command on the Relationship Detail screen, the RI Relationship Detail screen, or the AR Relationship Detail screen.

Figure 18-9. Graphical Relationship Display Screen



Understanding the Graphical Relationship Display

The body of this screen consists of a graphical representation of the relationships in which the object you selected on the Browse Table Specification screen or Browse File Specifications screen is involved. Read the Graphical Relationship Display as follows:

- Objects are identified by their:
  - Object type, either DB2, MVS, or IMS
  - Object name
    - DB2 objects include the table or view name. The Creator name is displayed only if necessary to have a unique name.
    - MVS objects include the fully qualified file name (quoted).
    - IMS objects include the fully qualified IMS database file name (quoted).
- Relationships are arranged such that each object that has no parents is at the top of a group. Each group consists of an object that has no parents, its dependents, their dependents, etc. File-AID *for DB2* displays one group of relationships for each object that has no parents. For example, in Figure 18-9, **MVS-'TSOID01.FARDX2.SAMP.ZIPCODE'** and **DB2-PART_TABLE** have no parents. They are at the top of their groups, respectively. There is one group displayed for **MVS-'TSOID01.FARDX2.SAMP.ZIPCODE'** and one group displayed for **DB2-PART_TABLE**.

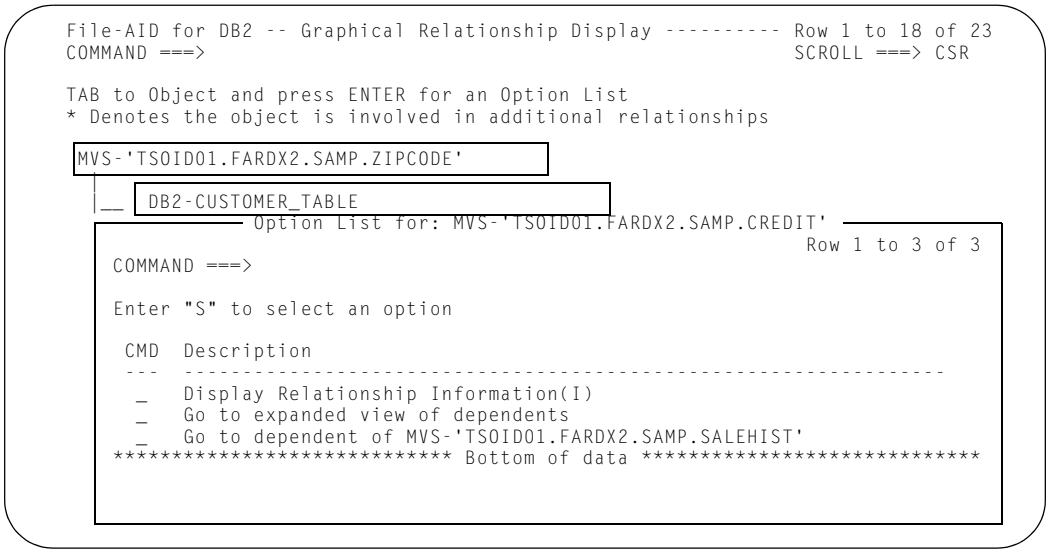
  Note: For situations involving cyclical relationships, File-AID *for DB2* puts the object for which the most detailed dependent display can be shown at the top of the group.

- The object you specified on the Browse Specifications screens is highlighted. In Figure 18-9, **MVS-'TSOID01.FARDX2.SAMP.CREDIT'** is the object specified and is therefore highlighted.
- Relationships are displayed as lines that connect the boxes. Lines start at the bottom of parent objects and end at the left side of dependent objects. In Figure 18-9, **MVS-'TSOID01.FARDX2.SAMP.ZIPCODE'** is a parent of **DB2-CUSTOMER_TABLE**. **DB2-CUSTOMER_TABLE** itself is a parent of **MVS-'TSOID01.FARDX2.SAMP.CREDIT'** and four other objects.
- Objects that are displayed more than once are denoted by an asterisk (*****). In Figure 18-9 on page 18-18, **DB2-ORDER_LINE_TABLE** is displayed with an asterisk next to it because it is displayed twice (once as a dependent of **DB2-ORDER_TABLE** and again as a dependent of **DB2-PART_TABLE**).

You can scroll UP, DOWN, LEFT, and RIGHT to see information that does not fit on a single screen.

Use the TAB key to move the cursor from object name to object name. When the cursor is next to an object name, you can press ENTER to display a list of options available as shown in Figure 18-10. In addition, you can enter the I line command to display the Relationship Information screen. See "Line Commands" on page 18-20 for more information about the I line command.

Figure 18-10. Graphical Relationship Display with Option List



Option List

To select an option from the Option List, type an **S** in the **CMD** column next to an option and press **Enter**.

The options in the Option List are as follows:

Display Relationship Information (I)

Displays the Relationship Information pop-up window. See "Relationship Information Screen" on page 18-20 for more information about the Relationship Information window.

Go to expanded view of dependents

Scrolls the display such that the first occurrence of the selected object is positioned at the top of the display. The first occurrence of a given object in the Graphical Relationship Display has an expanded display of its dependents (if it has any). Subsequent occurrences of a given object do not repeat expanded detail.

In Figure 18-9 on page 18-18, for example, if the cursor was positioned on the occurrence of **DB2-ORDER_LINE_TABLE** under **DB2-PART_TABLE**, selecting **Go to expanded view of dependents** would position the first occurrence of **ORDER_LINE_TABLE** at the top of the display. The first occurrence of **ORDER_LINE_TABLE** under **ORDER_TABLE** would display any dependents if **ORDER_LINE_TABLE** had any.

Go to dependent of *objectname*

Scrolls the display such that the occurrence of the selected object that is a dependent of *objectname* is at the top of the display. Use this option to determine where the object is a dependent in other relationships.

For example, if the cursor was positioned on the second occurrence of ORDER_LINE_TABLE when you pressed ENTER, the following entry would be displayed:

```
Go to dependent of ORDER_TABLE
```

Selecting this entry would position the occurrence of **ORDER_LINE-TABLE** under **ORDER_TABLE** at the top of the display. If **ORDER_LINE_TABLE** had additional parents, there would be an entry on the Option List for each parent.

Line Commands

The following File-AID *for DB2*-specific line command is valid on this screen.

I (Relationship Information)

Displays the Relationship Information screen. See “Relationship Information Screen” for more information about the Relationship Information screen.

Relationship Information Screen

The Relationship Information screen contains information about the relationship you specified on the Relationship Detail screen. If there are more columns comprising the relationship than can be displayed in this screen at one time, you can scroll the **Parent Columns** and **Dependent Columns** fields using standard scrolling commands.

Depending on the object types File-AID *for DB2* displays slightly different information. When you select a DB2 RI relationship, the Relationship Information window is displayed as shown in Figure 18-11.

Figure 18-11. Relationship Information Screen—RI relationship information

```
File-AID for DB2          Relationship Information       DB2  Row 1 to 2 of 2
COMMAND ===>                                             SCROLL   ===> CSR

 Parent Name:    LOCATION.TSOID01.PART_TABLE                 Type: DB2



 Dependent Name: LOCATION.TSOID01.SUPPLIER_TABLE             Type: DB2



Relationship Details:
 Description:    RI -  SUPPPART
 Status:
 Created:        1996-01-09 14.55.16.425499                  User: TSOID01
 Last Modified   1996-01-09 14.55.16.425499                  User: TSOID01
Parent Columns  /Type(Length)            Dependent Columns  /Type(Length)
---------------------------------------- ----------------------------------------
PART_NUMBER                                PART_NUMBER
   CHAR(6)                                    CHAR(6)
*** End Of Columns ***                     *** End Of Columns ***
```

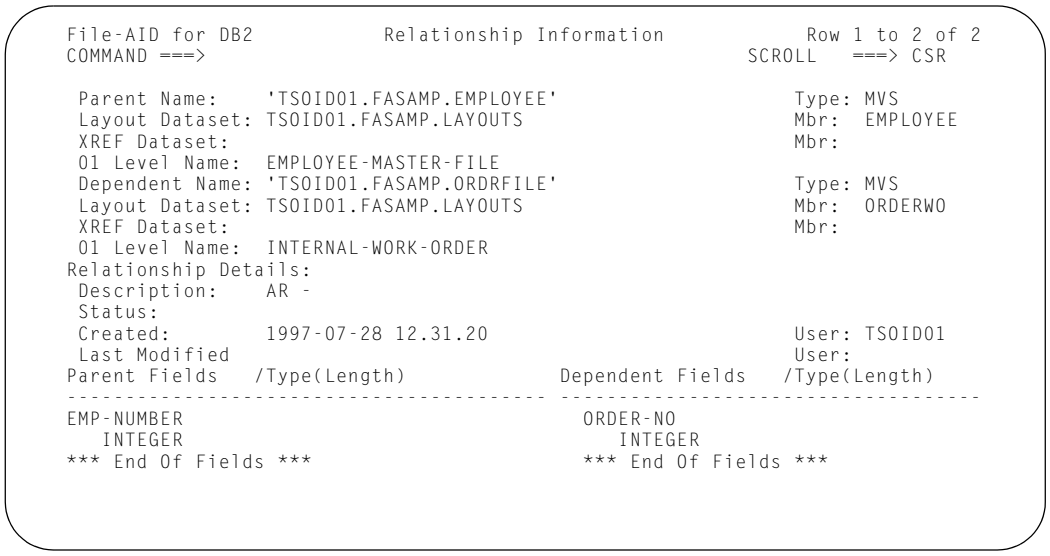
When you select an application relationship between an MVS and DB2 object, the Relationship Information window is displayed as shown in Figure 18-12 on page 18-21.

Figure 18-12. Relationship Information Screen—MVS to DB2 Application Relationship Information

```
File-AID for DB2          Relationship Information            Row 1 to 2 of 2
COMMAND ===>                                             SCROLL   ===> CSR

 Parent Name:    'TSOID01.FARDX2.SAMP.ZIPCODE'               Type: MVS
 Layout Dataset: TSOID01.FARDX2.SAMP.LAYOUT                  Mbr:  SMPZIP
 XREF Dataset:                                               Mbr:
 01 Level Name:
 Dependent Name: LOCATION.TSOID01.CUSTOMER_TABLE             Type: DB2



Relationship Details:
 Description:    AR -  ZIPCODE TO CUSTOMER
 Status:
 Created:                                                    User:
 Last Modified                                               User:
Parent Fields   /Type(Length)            Dependent Columns  /Type(Length)
---------------------------------------- ----------------------------------------
PARENT-KEY                                 CHILD-KEY
   INTEGER                                    INTEGER
*** End Of Fields ***                      *** End Of Columns ***
```

When you select an MVS to MVS AR relationship, the Relationship Information screen is displayed as shown in Figure 18-13.

Figure 18-13. Relationship Information Screen—MVS to MVS AR Relationship Information



The Relationship Information screen consists of the fields described below. Fields that vary according to relationship or object type are described under separate headings.

Parent and Dependent Object Name Area

The fields in this area provide information about the parent and dependent objects.

Parent Name

Displays the fully qualified name of the parent object according to the parent's object type.

Type

Indicates the object type of the parent and dependent object.

DB2
DB2 table.
MVS
MVS file.
IMS
IMS database (dependent only).

Layout File

Displays only if parent or dependent object is an MVS file. Identifies the record layout file for the MVS object used for this relationship.

XREF File

Only displays if layout usage is X. Identifies the XREF file name for the MVS object in this relationship.

01 Level Name

Only displays for MVS object. Identifies the 01 Level name for the MVS object in this relationship.

Mbr

Displays only if parent or dependent object is an MVS file. Each Mbr field identifies the member of the record layout file or the XREF file that contains the layout information for the parent or dependent MVS object in this relationship.

Dependent Name

Displays the fully qualified name of the dependent object according to the dependent's object type.

Relationship Details Area

The fields in this area provide information about the relationship such as the Relationship Description, the Status of the relationship, creation and modification dates.

Description

A description of the selected relationship. The contents of this field vary according to the relationship type as follows:

RI

The RELNAME from the DB2 catalog table, SYSIBM.SYSRELS (see Figure 18-11 on page 18-21).

AR

The description entered by the user when adding an application relationship (see Figure 18-12 on page 18-21).

Status

Indicates the condition of each relationship. For DB2 RI relationships, this column is **blank** if the information in the File-AID *for DB2* relationship file is consistent with the DB2 catalog for the particular relationship. If the relationship file is not consistent with the DB2 catalog, File-AID *for DB2* displays one of the following messages:

Message	Explanation
Not in File	Indicates that this relationship exists in the DB2 catalog, but does not exist in the relationship file specified on the RI Relationships screen. May indicate that the relationship did not exist the last time the relationship file was refreshed.
Not in DB2 Cat	Indicates that this relationship exists in the relationship file specified on the RI Relationships screen, but does not exist in the DB2 catalog.
Out of Sync 1	Indicates that the relationship exists in both the relationship file and the DB2 catalog, but the timestamps on each table do not match each other. May also indicate that the columns that comprise the foreign key are inconsistent.
ASSOCIATED REL	At least one of the objects involved in this relationship is a descendent of the driving object, or the driving object is a descendent of at least one of the objects in this relationship.

In addition, File-AID *for DB2* displays the following messages as a result of line command processing:

Message	Explanation
Deleted	Displayed after you enter the D line command. Indicates that this relationship has been deleted from the relationship file. It does **not** however, indicate that the relationship has been deleted from the DB2 catalog.
Loaded	Displayed after you enter the S line command. Indicates that this relationship has been loaded into the relationship file.
Refreshed	Displayed after you enter the S line command for a relationship that had the message **Out of Sync 1** displayed. Indicates that the DB2 catalog and the relationship file are now consistent with respect to this relationship.

For AR application relationships one of the following messages may be displayed:

Modified	Displayed when you enter the S line command and complete the Field Relationships screen. Indicates that this relationship has been modified.
Loaded	Displayed after you enter the S line command. Indicates that this relationship has been loaded into the relationship file.
Refreshed	Displayed after you enter the S line command for a relationship that had the message **Out of Sync 1** displayed. Indicates that the DB2 catalog and the relationship file are now consistent with respect to this relationship.

For AR application relationships one of the following messages may be displayed.

Modified	Displayed when you enter the S line command and complete the Field Relationships screen. Indicates that this relationship has been modified.
Added	Displayed when you enter the A line command and complete the Column/Field Relationships screen. Indicates that this relationship has been added to the relationship file.
Deleted	Displayed when you enter the D line command. Indicates that this relationship has been deleted from the relationship file.

In addition, File-AID *for DB2* could display the following error messages:

Object not Found	File-AID *for DB2* cannot find the parent object in the MVS or DB2 catalog. This may indicate that the object not found has been dropped. Use the D line command to delete the invalid relationships.
Column Errors	An invalid field relationship exists within the relationship. Invalid field relationships may be caused by eliminating or changing a column from a table defined in the DB2 catalog. Use the S line command to display the Column/Field Relationships screen where the invalid column/field relationships are indicated by the **ERR>** flag.
DBD not Found	File-AID *for DB2* cannot find the dependent IMS database. This may indicate that the database not found has been deleted or renamed. Use the D line command to delete the invalid relationship.
Segment Errors	An invalid root segment relationship exists within the IM relationship. Invalid relationships may be caused b eliminating or changing a root segment in the IMS database. Use the S line command to display the Column/Segment Relationships screen where the invalid relationships are indicated by the **ERR>** flag.

Created

Displays the date and time that the relationship was created. For DB2 RI relationships it represents when it was loaded into the DB2 catalog. File-AID *for DB2* displays date and time in the format specified at the time of DB2 installation.

User

Displays the user ID of the person who created the AR relationship or who loaded the RI relationship into the relationship file using either the initial load batch job or with the Load RI option or line command.

Last Modified

Display date, time, and user ID indicating when the relationship was last modified and who modified it. File-AID *for DB2* displays date and time in the format specified at the time of DB2 installation.

Parent Columns or Parent Fields

Displays the names of the columns from the DB2 parent table or the names of the fields from the MVS parent file that are involved in the relationship.

Type(Length)

The line directly below the parent field's or column's name displays the field's or column's data type. The length of the data is displayed in parentheses immediately following the data type for CHAR, VCHAR, GRPH, VGRPH, and DEC data types only. If an extended column relationship has been defined, File-AID *for DB2* indicates this by displaying the start position and length of the string from the parent column in parentheses instead of the column's length. For example, for character data types:

```
CHAR(2,3)
```

indicates a string in a column with a data type of CHAR has an extended column relationship defined beginning in position 2 for a length of 3.

Note: If the DB2 column data type is distinct (UDT), File-AID *for DB2* displays only the underlying built-in data type.

Dependent Columns or Dependent Fields

Displays the names of the columns from the dependent table or the names of the fields from the MVS dependent file that are involved in the relationship.

Type(Length)

The line directly below the dependent field's or column's name displays the field's or column's data type. The length of the data is displayed in parentheses immediately following the data type for CHAR, VCHAR, GRPH, VGRPH, and DEC data types only. If an extended column relationship has been defined, File-AID *for DB2* indicates this by displaying the start position and length of the string from the parent column in parentheses instead of the column's length. For example, for character data types:

```
CHAR(2,3)
```

indicates a string in a column with a data type of CHAR has an extended column relationship defined beginning in position 2 for a length of 3.

Note: If the DB2 column data type is distinct (UDT), File-AID *for DB2* displays only the underlying built-in data type.

Root Segment

Displays the name of the Root Segment from the IMS database that correspond to the columns or fields from the parent object (see Figure 18-20 on page 18-40).

RI Relationship Detail Screen

The RI Relationship Detail screen shown in Figure 18-13 is displayed when you select the Load RI option or enter the L line command for a specified DB2 object.

Figure 18-14. RI Relationship Detail Screen

```
File-AID for DB2
         RI Relationship Detail              Row 1 to 3 of 3
COMMAND ===>                                            SCROLL ===> CSR


Relationships for:   LOCATION.TSOID01.ORDER_TABLE

Line Commands: S - Select Relationship to Modify  D - Delete Relationship
               I - Relationship Information

                                                    Obj  Rel
CMD     Parent/Dependent                            Type Type      Status
--- ------------------------------------------------ ---- ----- -----------------
 _  LOCATION.FRSAMP.ORDER_TABLE                     DB2  RI
      LOCATION.FRSAMP.ORDER_LINE_TABLE               DB2
 _  LOCATION.FRSAMP.PART_TABLE                      DB2  RI    ASSOCIATED REL.
      LOCATION.FRSAMP.ORDER_LINE_TABLE               DB2
 _  LOCATION.FRSAMP.PART_TABLE                      DB2  RI    Not in Dataset
      LOCATION.FRSAMP.SUPPLIER_TABLE                 DB2
****************************** Bottom of data ********************************
```

This screen lists all DB2 RI relationships associated with the driving table, the DB2 table specified in the **Relationships for:** field, the table for which you had specified the Load RI option or L line command.

Use this screen to load and refresh relationships in the File-AID *for DB2* relationship file (S line command), delete relationships from the relationship file (D line command), and select relationships about which you want to view more detailed information (I line command).

This screen displays information from the DB2 catalog and File-AID *for DB2*'s relationship file about all RI relationships involving the specified table.

Relationships in which the driving table is either the parent or the dependent are listed first. Those relationships that involve the dependents of the driving table are listed below and are marked **ASSOCIATED REL**.

For each relationship, the following columns are displayed:

CMD

Enter line commands in this column. See "Line Commands" on page 18-20 for information about commands that are valid for this screen.

Parent/Dependent

Displays the name of the DB2 table defined as the parent in an RI relationship in the upper row (the row with the CMD input field). The foreign keys in the dependent table refer to this table. Table names are displayed in the format ***location.creator.table***.

Displays the name of the DB2 table defined as the dependent in an RI relationship in the lower row. This table contains the foreign keys. Table names are displayed in the format ***location.creator.table***.

Obj Type

Indicates the object type of the parent and dependent. For RI relationships, the object type should always be DB2.

Rel Type

Indicates the kind of relationship that exists between the corresponding parent and dependent objects.

RI DB2-defined referential integrity.

Status

Indicates the condition of each RI relationship. This column is blank if the relationship file is consistent with the DB2 catalog for the particular relationship. If the relationship file is not consistent with the DB2 catalog, File-AID *for DB2* displays one of the following messages:

Message	Explanation
Not in File	Indicates that this relationship exists in the DB2 catalog, but does not exists in the relationship file specified on the RI Relationship screen. May indicate that the relationship did not exist the last time the relationship file was refreshed.
Not in DB2 Cat	Indicates that this relationship exists in the relationship but does not exist in the Db2 catalog. Indicates that the parent or dependent tables, or a foreign key has been dropped.
Out of Sync1	Indicates the relationship exists in both the relationship file and the DB2 catalog, but the timestamps on each table do not match each other. May also indicate that the columns that comprise the foreign key are inconsistent.

In addition, File-AID *for DB2* displays the following messages as a result of line command processing:

Deleted	Displayed after you enter the D line command. Indicates that this relationship has been deleted from the relationship file. It does not, however, indicate that the relationship has been deleted from the DB2 catalog.
Loaded	Displayed after you enter the S line command. Indicates that this relationship has been loaded into the relationship file.
Refreshed	Displayed after you enter the S line command for a relationship that had the message Out of Sync 1 displayed. Indicates that the DB2 catalog and the relationship file are now consistent with respect to this relationship.

Line Commands

The following File-AID *for DB2*-specific line commands are valid on this screen:

D (Delete)

Deletes an RI relationship only from the relationship file and displays the message Deleted in the Status column. Use this command to delete a relationship from the relationship file when it no longer exists in the DB2 catalog.

I (Relationship Information)

If you enter the I line command on a relationship for which a message is displayed in the Status column, the RI Information window is displayed (see Figure 18-15 on page 18-28). If you enter the I line command on a relationship for which the Status column is blank (RI already in relationship file), the Relationship Information window is displayed (see Figure 18-11 on page 18-21).

S (Select)

Selects a relationship to be loaded or refreshed. File-AID *for DB2* retrieves information about the RI relationship from the DB2 catalog. If the relationship does not yet exist in the relationship file, File-AID *for DB2* loads the relationship and displays the

message Loaded in the Status column. If the relationship does exist but is inconsistent with the DB2 catalog, File-AID *for DB2* uses the information from the DB2 catalog to refresh the relationship file and displays the message Refreshed in the Status column. File-AID *for DB2* also displays the messages Loaded and Refreshed in the Status column.

Primary Commands

The following File-AID *for DB2*-specific primary command is valid on this screen:

GRAPHIC (G)

Enter the GRAPHIC command to invoke the Graphical Relationship Display screen and show RI relationships for the selected table. See "Graphical Relationship Display" on page 18-18 for more information on the Graphical Relationship Display screen. There are no operands associated with this command.

SQLID

Enter the SQLID primary command to change the current DB2 SQLID. Enter the SQLID command without an operand to change the current SQLID to match the current userid. Use the new-sqlid operand to specify any SQLID.

RI Information Screen

The RI Information screen is displayed when you enter the I line command on the RI Relationship Detail screen on a relationship and press **Enter**. This window contains detailed information about the relationship on which you invoked the I line command on the RI Relationship Detail screen.

Figure 18-15. RI Information Screen

```
File-AID for DB2        Relationship Information          DB2  Row 1 to 2 of 2
COMMAND ===>                                              SCROLL   ===> CSR

 Parent Name:    LOCATION.FRSAMP.EMP                          Type: DB2



 Dependent Name: LOCATION.FRSAMP.DEPT                         Type: DB2



Relationship Details:
 Description:    RI -  REP
 Status:         Not in Dataset
 Created:                                                     User:
 Last Modified                                                User:
Parent Columns  /Type(Length)            Dependent Columns  /Type(Length)
---------------------------------------- ------------------------------------
LNAME                                      REP

*** End Of Columns ***                     *** End Of Columns ***
```

Parent and Dependent Object Name Area

The fields in this area provide information about the parent and dependent objects.

Parent Name

Displays the name of the table defined as the parent table in an RI relationship. This table's primary index contains values that match the foreign keys defined in the corresponding dependent table. Tables are displayed in the format ***location.creator.table***.

Type

Indicates the object type of the parent and dependent object.

DB2 indicates a DB2 table.

Dependent Name

Displays the name of the table defined as the dependent table in an RI relationship. This table contains the foreign key, the values of which match the parent table's primary index. Tables are displayed in the format ***location.creator.table***.

Relationship Details Area

The fields in this area provide information about the relationship such as the Relationship Description, the Status of the relationship, creation and modification dates.

Description

A description of the selected relationship.

RI The RELNAME from the DB2 catalog table, SYSIBM.SYSRELS (see Figure 18-11 on page 18-21).

Status

Displays the message displayed in the **Status** column on the RI Relationship Detail screen.

Message	Explanation
NOT IN FILE	Relationship in DB2 catalog is not loaded in the relationship file.
NOT IN DB2 CAT	Relationship cannot found in the DB2 catalog.
OUT OF SYNC 1	Timestamp is inconsistent between DB2 catalog and relationship file.

Created

Displays the date and time that the relationship was created. For DB2 RI relationships it represents when it was loaded into the DB2 catalog. File-AID *for DB2* displays date and time in the format specified at the time of DB2 installation.

User

Displays the user ID of the person who loaded the RI relationship into the relationship file using either the initial load batch job or the Load RI option or line command.

Last Modified

Display date, time, and user ID indicating when the relationship was last modified and who modified it. File-AID *for DB2* displays date and time in the format specified at the time of DB2 installation.

Parent Columns

Displays the names of the columns from the parent table that are referenced by the foreign key.

Dependent Columns

Displays the names of the columns from the dependent table that comprise the foreign key.

Type(Length)

The line directly below the parent column's name displays the column's data type. The length of the data is displayed in parentheses immediately following the data type for CHAR, VCHAR, GRPH, VGRPH, and DEC data types only. Note that the length value only displays after the RI has been loaded into the relationship file.

Note: If the DB2 column data type is distinct (UDT), File-AID *for DB2* displays only the underlying built-in data type.

Application Relationship Detail

The Application Relationship Detail screen shown in Figure 18-16 is displayed when you select the Edit AR option or enter the E line command for an object that already has an application relationship defined in the current relationship file.

Figure 18-16. Application Relationship Detail Screen

```
File-AID for DB2       Application Relationship Detail         Row 1 to 1 of 1
COMMAND ===>                                                   SCROLL ===> CSR

Relationships for:   LOCATION.TSOID01.BOOKSUBJECTS

Line Commands: S - Select Relationship to Modify  D - Delete Relationship
               A - Add Application Relationship   I -
Relationship Information

                                                    Obj  Rel
CMD     Parent/Dependent                            Type Type      Status
--- ------------------------------------------------ ---- ----- ----------------
  _  LOCATION.TSOID01.BookSubjects                   DB2  AR
       LOCATION.TSOID01.DLETTBL                      DB2        COLUMN ERRORS
****************************** BOTTOM OF DATA ********************************
```

This screen displays a list of application relationships (AR) for the object and relationship file specified with the **Edit AR** option or **E** line command. It is similar to the "Relationship Detail" on page 18-15, except that it only displays application relationships for the selected object.

Use this screen to select application relationships from the relationship file that you want to modify (**S** line command) or delete (**D** line command). You can also use this screen to display detailed information about a relationship (**I** line command) or to begin the process of adding new application relationships (**A** line command or **ADD** primary command).

CMD

Enter a valid line command next to the appropriate relationship. See "Line Commands" on page 18-20 for information about valid line commands.

Parent/

Displays the name of the object defined as the parent in a relationship in the upper row (the row with the CMD input field).

/Dependent

Displays the name of the object defined as the dependent in a relationship in the lower row, under its parent.

Obj Type

Indicates the object type of the parent and dependent.

IMS IMS database.

DB2 DB2 table.

MVS MVS file.

Rel Type

Indicates the kind of relationship that exists between the corresponding parent and dependent objects. (Always AR as only application relationships are displayed.)

AR	User-defined application relationship. Created and maintained by the user.

Status

Indicates the condition of each relationship. This field contains a message only if there are inconsistencies in the relationship file or as a result of line or primary command processing. Refer to "Status" on page 18-29 for a list possible messages.

Message	Explanation
Modified	Displayed when you enter the S line command and complete the Relationships screen. Indicates that this relationship has been modified.
Added	Displayed when you enter the A line command or ADD primary command and complete the Relationships screen. Indicates that this relationship has been added to the relationship file.
Deleted	Displayed when you enter the D line command. Indicates that this relationship has been deleted from the relationship file.

In addition, File-AID *for DB2* could display the following error messages:

Message	Explanation
Object not found	cannot find one of the objects (dependent or parent) in the MVS or DB2 catalog. This may indicate that the object not found has been dropped. Use the D line command to delete the invalid relationship.
Column Errors	An invalid column relationship exists within the application relationship. Invalid column relationships may be caused by eliminating or changing a column from a table defined in the DB2 catalog. Use the S line command to display the Column/Field Relationships screen where the invalid column/field relationship is in indicated by the **ERR>** flag.

Line Commands

The following File-AID *for DB2*-specific line commands are valid on this screen:

A (Add)

Adds an application relationship for this parent object. Displays the Add Relationships screen. See "Add Relationships — Specify Object Types" on page 18-34.

D (Delete)

Deletes an application relationship from the relationship file.

I (Relationship Information)

Invokes the Relationship Information screen. See "Relationship Information Screen" on page 18-20 for more information.

S (Select)

Selects an application relationship for modification and displays the Column/Field Relationships screen. See "Building Column/Field Relationships" on page 18-56 for more information.

Primary Commands

The following File-AID *for DB2*-specific primary commands are valid on this screen:

ADD

Enter the ADD command in the COMMAND field to add an application relationship. Displays the Add Relationships screen. See "Add Relationships — Specify Object Types" on page 18-34.

GRAPHIC (G)

Enter the GRAPHIC command in the COMMAND field to invoke the Graphical Relationship Display screen and show application relationships for the selected table. See "Graphical Relationship Display" on page 18-18 for more information on the Graphical Relationship Display screen. There are no operands associated with this command.

Relationship Information

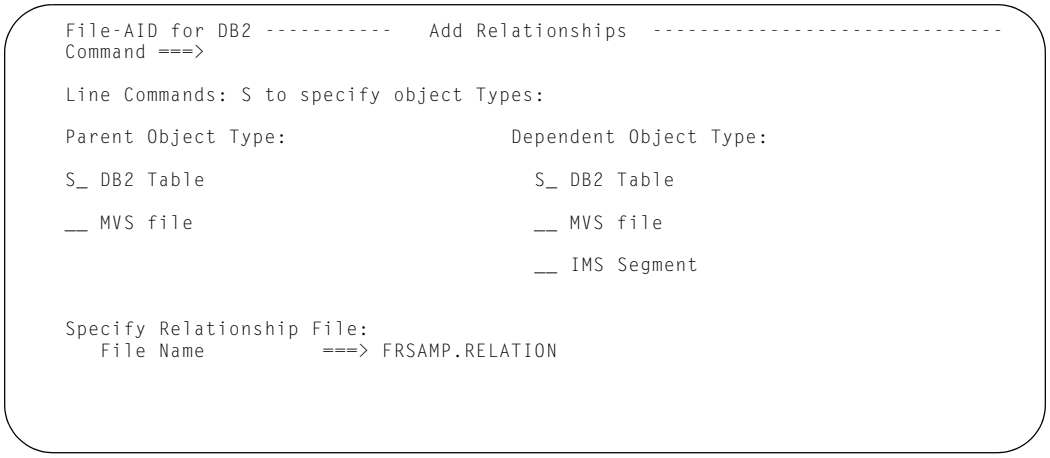
The Relationship Information screen is displayed when you enter the I line command on the Relationship Detail screen or the Application Relationship Detail screen. Use the Relationship Information screen to view detailed information about the relationship on which you entered the I line command. See "Relationship Information Screen" on page 18-20 for more information about the contents of this window.

Add Relationships — Specify Object Types

The Add Relationships — Specify Object Types screen shown in Figure 18-17 is displayed when you do one of the following:

- Enter the **Add AR** option or **A** line command for a specific object.
- Enter the **Edit AR** option or **E** line command for a specific object and no object match is found in the relationship file
- Enter the **ADD** primary command on the Relationship Detail screen
- Use the **A** (Add) line command on the Relationship Detail screen (see Figure 18-8 on page 18-15).

Figure 18-17. Object Type Selection Screen for Adding Application Relationship



Use this screen to specify the object type of the parent and the dependent object for the new relationship.

Parent Object Type

Select one of the two choices to indicate the object type of the parent:

DB2 Table	Place an **S** line command in front of this option if the parent object is a DB2 table.
MVS File	Place an **S** line command in front of this option if the parent object is an MVS file.

Dependent Object Type

Select one of the three choices to indicate the object type of the dependent:

DB2 Table	Place an **S** line command in front of this option if the dependent object is a DB2 table.
MVS File	Place an **S** line command in front of this option if the dependent object is an MVS file.
IMS Segment	Place an **S** line command in front of this option if the dependent object is an IMS Segment.

Press Enter and continue with one of the six possible parent-dependent combinations:

1. **DB2 ==> DB2**, see "Add Application Relationships — DB2 to DB2" on page 18-35
2. **DB2 ==> MVS**, see "Add Relationships — DB2 to MVS" on page 18-37
3. **DB2 ==> IMS**, see "Add Relationships — DB2 to IMS" on page 18-40
4. **MVS ==> DB2**, see "Add Relationships — MVS to DB2" on page 18-42

5. **MVS ==> MVS**, see "Add Relationships — MVS to MVS" on page 18-44
6. **MVS ==> IMS**, see "Add Relationships — MVS to IMS" on page 18-47

Add Application Relationships — DB2 to DB2

This Add Application Relationships screen shown in Figure 18-18 is displayed after you specified DB2 tables for both parent and dependent objects in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-18. Add Relationships Screen — DB2 to DB2

```
File-AID for DB2 -------------  Add Application Relationships  ----------------
Command ===>

Specify Parent DB2 Table for Relationship Definition:
 Location     ===> LOCATION
 Creator      ===> TSOID01              (* or pattern for selection list)
 Table Name   ===> EMP                  (* or pattern for selection list)

Specify Dependent DB2 Table for Relationship Definition:
 Location     ===> LOCATION
 Creator      ===> TSOID01              (* or pattern for selection list)
 Table Name   ===> EMP                  (* or pattern for selection list)

Suggest Relationships when column name/type/length are identical:
 Suggest      ===> N                    (Y = Yes; N = No)

SSID          ===> DSN                  (* or pattern for selection list)
Description   ===>
```

Use this screen to specify the parent and dependent table for the new relationship.

Specify Parent DB2 Table for Relationship Definition Area

Use the fields in this area to specify the DB2 table that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Location

Specify the DB2 remote location ID of the DB2 subsystem on which the table resides for which you want to add an application relationship.

Creator

Specify the authorization ID of the owner of the table for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Specify Dependent DB2 Table for Relationship Definition Area

Use the fields in this area to specify the DB2 table that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Location

Identifies the DB2 remote location ID of the DB2 subsystem on which the table resides which is to be the dependent table of the new application relationship. Protected field; same as the parent location by default.

Creator

Specify the authorization ID of the owner of the table that is to be the dependent for the new application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) that shall be the dependent (child) table for the new application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Suggest Relationships when column name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when column name, type, and length in the parent and dependent tables are identical.

Y

Yes, File-AID *for DB2* will select columns with identical names, type and length for this relationship definition and present them in the following Column Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then accept them or delete individual relationship definitions and/or add your own definitions.

N

No, File-AID *for DB2* will not suggest any application relationships when entering the following Column Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

SSID

Specify the DB2 subsystem ID of the DB2 subsystem on which the table resides for which you want to add a relationship. To view a list of subsystems, specify a pattern.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information window.

After specifying all required parent and dependent information, press Enter. You then continue with the Column Relationship screen to define the related columns, see "Relationships" on page 18-50.

Add Relationships — DB2 to MVS

This Add Relationships screen shown in Figure 18-19 is displayed after you specified DB2 table as parent object and MVS file as dependent object in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-19. Add Relationships Screen — DB2 to MVS

```
File-AID for DB2 -------------  Add Relationships  ----------------------------
Command ===>

Specify Parent DB2 Table for Relationship Definition:
 SSID         ===> DSN                  (* or pattern for selection list)
 Location     ===> LOCATION
 Creator      ===> TSOID01              (* or pattern for selection list)
 Table Name   ===> EMP                  (* or pattern for selection list)

Specify Dependent MVS File for Relationship Definition:
 Name                   ===>
 Record Layout usage    ===>            (S = Single; X = XREF;)
 Record layout file     ===>
 Member name            ===>            (Blank or pattern for member list)
 XREF file name         ===>
 Member name            ===>            (Blank or pattern for member list)
 I/O Exit Name          ===>

Suggest Relationships when column and field name/type/length are identical:
 Suggest       ===> N                    (Y = Yes; N = No)

Description            ===>
```

Use this screen to specify the DB2 parent table and the dependent MVS file for the new relationship.

Specify Parent DB2 Table for Relationship Definition Area

Use the fields in this area to specify the DB2 table that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

SSID

Specify the DB2 subsystem ID of the DB2 subsystem on which the table resides for which you want to add relationship information.

Note: If the dependent file is already involved in another relationship definition with a DB2 object, you must use the same SSID to add a new relationship.

Location

Specify the DB2 remote location ID of the DB2 subsystem on which the table resides for which you want to add an application relationship.

Creator

Specify the authorization ID of the owner of the table for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern

characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Specify Dependent MVS File for Relationship Definition Area

Use the fields in this area to specify the MVS file that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Note: If the file is already involved in another relationship definition, you must use the same layout, if usage=S, or use the same XREF file to add a new relationship.

File name

Specify the MVS file name of the dependent in the new relationship. To view a list of files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.
You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object that is to be the dependent in the new relationship. To view a list of layout files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS file. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS file that is to be the dependent in the new relationship. To view a list of XREF files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the dependent MVS file. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the dependent MVS file.

You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing the above MVS file. Not needed to search the relationship file.

Suggest Relationships when column and field name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when column and field name, type, and length in the parent and dependent tables are identical.

Y

Yes, File-AID *for DB2* will select columns and fields with identical names, type and length for this relationship definition and present them in the following Column/Field Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then

accept them or delete individual relationship definitions and/or add your own definitions.

N

No. File-AID *for DB2* will not suggest any application relationships when entering the following Column/Field Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information screen.

After specifying all required parent and dependent information, press Enter. You then continue with the Column/Field Relationship screen to define the related DB2 columns and MVS fields, see "Relationships" on page 18-50.

Add Relationships — DB2 to IMS

This Add Relationships screen shown in Figure 18-20 is displayed after you specified DB2 table as parent object and IMS Segment as dependent object in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-20. Add Relationships Screen — DB2 to IMS

```
File-AID for DB2 -------------  Add Relationships  ----------------------------
Command ===>

Specify Parent DB2 Table for Relationship Definition:
 SSID         ===> DSN                  (* or pattern for selection list)
 Location     ===> LOCATION
 Creator      ===> TSOID01              (* or pattern for selection list)
 Table Name   ===> EMP                  (* or pattern for selection list)

Specify Dependent IMS Database for Relationship Definition:
 DBD Library  ===> 'TSOID01.FISAMP.DBDLIB'
 DBD Member   ===> *                    (Blank or pattern for member list)

Suggest relationships when column and segment name/type/length are identical:
 Suggest       ===> N                    (Y = Yes; N = No)

Description   ===> DB2 to IMS relationship
```

Use this screen to specify the DB2 parent table and the dependent IMS database for the new relationship.

Specify Parent DB2 Table for Relationship Definition Area

Use the fields in this area to specify the DB2 table that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

SSID

Specify the DB2 subsystem ID of the DB2 subsystem on which the table resides for which you want to add relationship information.

Location

Specify the DB2 remote location ID of the DB2 subsystem on which the table resides for which you want to add an application relationship.

Creator

Specify the authorization ID of the owner of the table for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) that is to be the parent table in the new application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and * (see "Wildcard Characters for Relationship Option" on page 2-9).

Specify Dependent IMS Database for Relationship Definition Area

Use the fields in this area to specify the IMS database that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

DBD Library

Specify the DBD library for which you want to establish a relationship with the parent DB2 table.

Member

Specify the member of the DBD library for which you want to establish a relationship. If you leave this field blank or enter a pattern other than an asterisk (*), File-AID *for DB2* displays a Member List pop-up window containing a list of members that match the specified pattern.

Suggest Relationships when column and segment name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when column or segment name, type, and length in the parent and dependent objects are identical.

Y

Yes, File-AID *for DB2* will select columns with identical names, type and length for this relationship definition and present them in the following Column/Segment Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then accept them or delete individual relationship definitions and/or add your own definitions.

N

No. File-AID *for DB2* will not suggest any application relationships when entering the following Column/Segment Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information screen.

After specifying all required parent and dependent information, press Enter. You then continue with the Column/Segment Relationship screen to define the related DB2 columns and IMS root segment, see “Relationships” on page 18-50.

Add Relationships — MVS to DB2

This Add Relationships screen shown in Figure 18-21 is displayed after you specified MVS file as parent object and DB2 table as dependent object in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-21. Add Relationships Screen — MVS to DB2

```
File-AID for DB2 -------------  Add Relationships  ----------------------------
Command ===>

Specify Parent File for Relationship Definition:
 Name                  ===>
 Record Layout usage   ===>            (S = Single; X = XREF;)
 Record layout file    ===>
 Member name           ===>            (Blank or pattern for member list)
 XREF file name        ===>
 Member name           ===>            (Blank or pattern for member list)
 I/O Exit Name         ===>

Specify Dependent DB2 Table for Relationship Definition:
 SSID         ===> DSN                 (* or pattern for selection list)
 Location     ===> LOCATION
 Creator      ===> TSOID01             (* or pattern for selection list)
 Table Name   ===> EMP                 (* or pattern for selection list)

Suggest Relationships when field and column name/type/length are identical:
 Suggest      ===> N                   (Y = Yes; N = No)

Description   ===> MVS to DB2 relationship
```

Use this screen to specify the parent MVS file and the dependent DB2 table for the new relationship.

Specify Parent File for Relationship Definition Area

Use the fields in this area to specify the MVS file that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Note: If the file is already involved in another relationship definition, you must use the same layout, if usage=S, or use the same XREF file to add a new relationship.

File name

Specify the MVS file name of the parent in the new relationship. To view a list of files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Note: If your current relationship file has an existing relationship for this MVS file to a DB2 table, the new relationship must be to a DB2 table on the same DB2 subsystem.

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.
You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object that is to be the parent in the new relationship. To view a list of layout files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS file. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS file that is to be the parent in the new relationship. To view a list of XREF files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the parent MVS file. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the parent MVS file.
You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing the above MVS file. Not needed to search the relationship file.

Specify Dependent DB2 Table for Relationship Definition Area

Use the fields in this area to specify the DB2 table that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

SSID

Specify the DB2 subsystem ID of the DB2 subsystem on which the table resides for which you want to add relationship information.

Note: If the parent file is already involved in another relationship definition with a DB2 object, you must use the same SSID to add a new relationship.

Location

Specify the DB2 remote location ID of the DB2 subsystem on which the table resides for which you want to add an application relationship.

Creator

Specify the authorization ID of the owner of the table for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Table Name

Specify the name of the DB2 object (table or view) for which you want to add an application relationship. To view a list of tables, specify a pattern. Valid pattern characters are %, _, and ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Suggest Relationships when field and column name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when column/field name, type, and length in the parent and dependent objects are identical.

Y

Yes, File-AID *for DB2* will select columns and fields with identical names, type and length for this relationship definition and present them in the following Field/Column Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then

accept them or delete individual relationship definitions and/or add your own definitions.

N

No. File-AID *for DB2* will not suggest any application relationships when entering the following Field/Column Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information screen.

After specifying all required parent and dependent information, press Enter. You then continue with the Field/Column Relationship screen to define the related MVS fields and DB2 columns, see "Relationships" on page 18-50.

Add Relationships — MVS to MVS

This Add Relationships screen shown in Figure 18-22 is displayed after you specified MVS file as both parent and dependent object in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-22. Add Relationships Screen — MVS to MVS

```
File-AID for DB2 -------------  Add Relationships  ----------------------------
Command ===>

Specify Parent File for Relationship Definition:
 Name                   ===>
 Record Layout usage    ===>            (S = Single; X = XREF;)
 Record layout file     ===>
 Member name            ===>            (Blank or pattern for member list)
 XREF file name         ===>
 Member name            ===>            (Blank or pattern for member list)
 I/O Exit Name          ===>

Specify Dependent MVS File for Relationship Definition:
 Name                   ===>
 Record Layout usage    ===>            (S = Single; X = XREF;)
 Record layout file     ===>
 Member name            ===>            (Blank or pattern for member list)
 XREF file name         ===>
 Member name            ===>            (Blank or pattern for member list)
 I/O Exit Name          ===>

Suggest Relationships when field name/type/length are identical:
 Suggest                ===> N                    (Y = Yes; N = No)
Description             ===> MVS to MVS relationship
```

Use this screen to specify the parent MVS file and the dependent MVS file for the new relationship.

Specify Parent File for Relationship Definition Area

Use the fields in this area to specify the MVS file that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Note: If the file is already involved in another relationship definition, you must use the same layout, if usage=S, or use the same XREF file to add a new relationship.

File name

Specify the MVS file name of the parent in the new relationship. To view a list of files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.
You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object that is to be the parent in the new relationship. To view a list of layout files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS file. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS file that is to be the parent in the new relationship. To view a list of XREF files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the parent MVS file. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the parent MVS file.

You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing the above MVS file. Not needed to search the relationship file.

Specify Dependent MVS File for Relationship Definition Area

Use the fields in this area to specify the MVS file that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Note: If the file is already involved in another relationship definition, you must use the same layout, if usage=S, or use the same XREF file to add a new relationship.

File name

Specify the MVS file name of the dependent in the new relationship. To view a list of files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.
You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object that is to be the dependent in the new relationship. To view a list of layout files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS file. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS file that is to be the dependent in the new relationship. To view a list of XREF files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the dependent MVS file. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the dependent MVS file.

You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing the above MVS file. Not needed to search the relationship file.

Suggest Relationships when field name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when field name, type, and length in the parent and dependent objects are identical.

Y

Yes, File-AID *for DB2* will select fields with identical names, type and length for this relationship definition and present them in the following Field Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then accept them or delete individual relationship definitions and/or add your own definitions.

N

No. File-AID *for DB2* will not suggest any application relationships when entering the following Field Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information screen.

After specifying all required parent and dependent information, press Enter. You then continue with the Field Relationship screen to define the related MVS fields, see "Relationships" on page 18-50.

Add Relationships — MVS to IMS

This Add Relationships screen shown in Figure 18-23 is displayed after you specified MVS file as parent object and IMS Segment as dependent object in the previous screen, "Add Relationships — Specify Object Types" on page 18-34 or "Application Relationship Detail" on page 18-31.

Figure 18-23. Add Relationship Screen — MVS to IMS

```
File-AID for DB2 -------------  Add Relationships  ----------------------------
Command ===>

Specify Parent File for Relationship Definition:
 Name                  ===>
 Record Layout usage   ===>            (S = Single; X = XREF;)
 Record layout file    ===>
 Member name           ===>            (Blank or pattern for member list)
 XREF file name        ===>
 Member name           ===>            (Blank or pattern for member list)
 I/O Exit Name         ===>

Specify Dependent IMS Database for Relationship Definition:
 DBD Library  ===> 'TSOID01.FISAMP.DBDLIB'
 DBD Member   ===> *                   (Blank or pattern for member list)

Suggest Relationships when column and segment name/type/length are identical:
 Suggest      ===> N                   (Y = Yes; N = No)

Description   ===> MVS to IMS relationship
```

Use this screen to specify the parent MVS file and the dependent IMS database for the new relationship.

Specify Parent File for Relationship Definition Area

Use the fields in this area to specify the MVS file that is the parent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

Note: If the file is already involved in another relationship definition, you must use the same layout, if usage=S, or use the same XREF file to add a new relationship.

File name

Specify the MVS file name of the parent in the new relationship. To view a list of files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Record layout usage

Indicate whether a single record layout (usage "S"), or an XREF (usage "X") is to be used to define the selection criteria.

You must also supply a Record Layout file and either a layout member (Usage=S) or an XREF file and member (Usage=X).

Record layout file

Specify the record layout file for the MVS object that is to be the parent in the new relationship. To view a list of layout files, specify a pattern. Valid pattern character is ***** (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the record layout file that contains the layout information for the MVS file. Leave blank or use a pattern to select a member from a list.

XREF file name

Only needed if layout usage is X. Specify the XREF file name for the MVS file that is to be the parent in the new relationship. To view a list of XREF files, specify a pattern. Valid pattern character is * (see "Wildcard Characters for Relationship Option" on page 2-9).

Member name

Specify the member of the XREF file that contains the layout information for the parent MVS file. Leave blank or use a pattern to select a member from a list.

I/O Exit Name

Specify the I/O Exit Name for the parent MVS file.

You only need to fill in this field when you want File-AID *for DB2* to use an IO exit routine when accessing the above MVS file. Not needed to search the relationship file.

Specify Dependent IMS Database for Relationship Definition Area

Use the fields in this area to specify the IMS database that is the dependent in the new relationship. When specifying patterns in the fields, File-AID *for DB2* presents selection lists.

DBD Library

Specify the DBD library for which you want to establish a relationship with the parent MVS file

Member

Specify the member of the DBD library for which you want to establish a relationship. If you leave this field blank or enter a pattern other than an asterisk (*), File-AID *for DB2* displays a Member List pop-up window containing a list of members that match the specified pattern.

Suggest Relationships when field and segment name/type/length are identical Area

Suggest

Specify whether you want File-AID *for DB2* to select relationships when field and root segment name, type, and length in the parent and dependent objects are identical.

Y

Yes, File-AID *for DB2* will select fields and root segments with identical names, type and length for this relationship definition and present them in the following Field/Segment Relationships Screen (see Figure 18-24 on page 18-50). A message will inform you how many relationships have been suggested. You can then accept them or delete individual relationship definitions and/or add your own definitions.

N

No. File-AID *for DB2* will not suggest any application relationships when entering the following Field/Segment Relationships Screen (see Figure 18-24 on page 18-50).

Note: When parent and dependent objects are the same object, File-AID *for DB2* will bypass the automatic suggest functions as all columns/fields would be suggested. You can still issue the SUGGEST primary command in the Field/Column Relationships screen. Remember, File-AID *for DB2* only allows up to 20 field/column relationship definition.

Relationship Description Area

Description

Enter a description up to 128 characters in length of the application relationship to be created. This information is displayed in the Relationship Information screen.

After specifying all required parent and dependent information, press Enter. You then continue with the Field/Segment Relationship screen to define the related MVS fields and IMS root segment, see “Relationships” on page 18-50.

Relationships

The Relationships screen shown in Figure 18-24 is displayed after you

- specified the related objects in the Add Relationship screen.
- used the **S** (Select) line command on the Application Relationship Detail screen.

The screen title varies depending on the object types involved.

Table 18-1 shows the possible title combinations.

Table 18-1. Relationship Titles

Object Type Parent	Object Type Dependent	Title
DB2	DB2	Column Relationships
DB2	MVS	Column/Field Relationships
DB2	IMS	Column/Segment Relationships
MVS	MVS	Field Relationships
MVS	DB2	Field/Column Relationships
MVS	IMS	Field/Segment Relationships

Figure 18-24. Relationships Screen - Column Relationships not yet defined

```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                    SCROLL ===> PAGE
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.EMP - DB2
Dependent: LOCATION.TSOID01.ORDKFL - DB2
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
____ EMPNO                           CHAR(6)
____ FIRSTNME                        VCHAR(12)
____ MIDINIT                         CHAR(1)
____ LASTNAME                        VCHAR(15)
____ WORKDEPT                        CHAR(3)
____ PHONENO                         CHAR(4)
____ HIREDATE                        DATE
CMD  Dependent Columns               Type(Length)
---- ------------------------------- ------------
   1 ORDER_NUMBER                    CHAR(6)
   2 CUST_NUM                        CHAR(6)
   3 ORD_TYPE                        CHAR(8)
   4 ORD_DATE                        DATE
   5 ORD_STAT                        CHAR(8)
   6 ORD_AMOUNT                      DEC(9,2)
```

Figure 18-24 shows the Relationships screen as it appears when adding a DB2 to DB2 application relationship, before any column relationships have been defined or suggested.

The Relationships screen shown in Figure 18-25 on page 18-51 is displayed when maintaining an MVS to MVS application relationship, with field relationships already defined.

Figure 18-25. Relationships Screen — FIeld Relationships already defined

```
File-AID for DB2 ----------  Field Relationships  ----------------------------
Command ===>                                                 SCROLL ===> PAGE
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: TSOID01.FARDX2.SAMP.SALEHIST - MVS
Dependent: TSOID01.FARDX2.SAMP.ZIPCODE - MVS
CMD  Parent Fields                    Type(Length) Mapped Fields
---- -------------------------------- ------------ -------------------------------
____ FILLER                           CHAR(15)     6(1,11,1)
____ SH-DATE-SHIP                     CHAR(8)      3(1,4,1)
      - - - - - - - -  Relationships 1 to 2 of 3  - - - - - - - -
____ SH-CUST-NO                       CHAR(6)
____ SH-ORD-NO                        CHAR(6)
____ SH-ORD_TYPE                      CHAR(8)
____ SH-DATE-SHIP                     CHAR(8)
____ SH-SHIP-CUST-NO                  CHAR(6)
CMD  Dependent Fields                 Type(Length)
---- -------------------------------- ------------
   1 ZM-BASE-ZIP                      CHAR(5)
   2 ZM-ZIP-SUF                       CHAR(4)
   3 ZM-TIME-DIFF                     DEC(2,0)
```

Figure 18-25 shows the Relationships screen as it appears when modifying an MVS to MVS application relationship. Notice that the titles now include the MVS naming convention where Parent Fields and Dependent Fields are used in place of DB2 columns. Also, the Parent Portion of the screen includes another scrollable area — *Relationships* — that lists the defined relationships.

Use the relationship screen to create or update the column or field level relationship detail for the related objects in the relationship file. See "Building Column/Field Relationships" on page 18-56 for more information about how to use the Relationships screen.

The body of the Relationships screen consists of two portions: a parent portion and a dependent portion. The parent portion includes two scrollable areas, Defined Relationships and Parent Columns/Fields. When the parent and dependent portions contain more information than can be displayed at one time, use the scroll commands to scroll through the data. Each area can also be scrolled individually with scroll mode **CSR**: move the cursor into the area you want to scroll and use the UP or DOWN scroll commands.

Parent Portion

The upper portion of this screen represents the parent information for the relationship and contains the following fields:

Object Names

The top portion of this screen displays the object names for the relationship :

Parent

Displays the full name of the parent object followed by the object type. For example, DB2 tables are identified in the form ***location.creator.table***.

Dependent

Displays the full name of the dependent object followed by the object type. For example, DB2 tables are identified in the form ***location.creator.table***.

Parent Portion

The Parent portion of this screen represents the parent information for the relationship.

If relationships have already been defined, the Parent portion is divided into two scrollable areas by a dashed line, for example

```
- - - - - - - -  Relationships 1 to 3 of 4  - - - - - - - -
```

The part above the dashed line contains the names of the columns/fields participating in the application relationship with the Mapped Fields values filled in. The part below the dashed line contains the names of all of the columns/fields for the parent.

CMD

Enter line commands in this column. See "Line Commands" on page 18-20 for information about valid commands. This column displays the **ERR>** flag when you select an existing relationship from the Application Relationship Detail screen in the following situations:

- When File-AID *for DB2* cannot find a column or field that is defined as part of an application relationship. You should delete such a relationship before you exit from this screen.
- When there is a conflict between column information stored in the relationship file and column information stored in the DB2 catalog table, SYSIBM.SYSCOLS.

Parent Columns or Parent Fields

Displays the names of the columns in the DB2 parent table, or, if the parent is an MVS file, displays the names of the fields. If the field contains an occurrence of subscripts, the element notation follows in parentheses.

Type(Length)

Displays the parent column's or parent field's data type. The length of the data is displayed in parentheses immediately following the data type for CHAR, VCHAR, GRPH, VGRPH, DEC, FIXPIC, data types only.

See Table 18-2 for more information about data types displayed in this column. Table 18-3 on page 18-53 identifies the allowable column/field type mapping when building a column relationship.

Notes:

1. If a DB2 column is a distinct type (UDT) File-AID *for DB2* displays the built-in column type only.
2. Columns with Large Object Type (CLOB, BLOB, DBCLOB) and/or ROWID column types cannot be part of an application relationship.

Table 18-2. Data Type Values

Data Type	Value
CHAR	Fixed-length character strings less than or equal to 254 characters.
VCHAR	Variable-length character strings less than or equal to the field length specified when the column is created. Maximum allowable field length is 4056 characters for 4k pages and 32,714 characters for 32k pages.
LVCHAR	Variable-length character strings less than or equal to 4056 characters for 4k pages and 32,714 characters for 32k pages.
GRPH	Fixed-length graphic strings less than or equal to 127 characters.
VGRPH	Variable-length graphic strings less than or equal to the field length specified when the column is created. Maximum allowable field length is 2028 characters for 4k pages and 16,357 characters for 32k pages.
LVGRPH	Variable-length graphic strings less than or equal to 2028 characters for 4k pages and 16,357 characters for 32k pages.

Table 18-2. Data Type Values (Continued)

Data Type	Value
SMALLINT	Fixed-length, binary integers in the range of -32,768 to +32,767, requiring two bytes of storage.
INTEGER	Fixed-length, binary integers in the range of -2,146,483,648 to +2,147,483,467, requiring four bytes of storage.
DEC	Fixed-length, decimal numbers in the range of -10E15 to +10E15.
REAL	Fixed-length, single-precision, floating-point numbers in the approximate range of 5.4E-79 to 7.2E+75.
DOUBLE	Fixed-length, double-precision, floating-point numbers in the approximate range of 5.4E-79 to 7.2E+75.
DATE	A three-part value (year, month, day) whose format is determined by DB2.
TIME	A three-part value (hour, minute, second) whose format is determined by DB2.
TIMESTAMP	A seven-part value (year, month, day, hour, minute, second, microsecond) whose format is YYYY-MM-DD-HH.MM.SS.TTTTTT.
CLOB	A character large object (CLOB) is a varying-length string with a maximum length of 2 147 483 647 bytes (2 gigabytes minus 1 byte). (DB2 Version 6.1 and later only)
DBCLOB	A double-byte character large object (DBCLOB) is a varying-length string with a maximum length of 1 073 741 823 double-byte characters. (DB2 Version 6.1 and later only)
BLOB	A binary large object (BLOB) is a varying-length string with a maximum length of 2 147 483 647 bytes (2 gigabytes minus 1 byte). (DB2 Version 6.1 and later only)
ROWID	Row identifier. A value that uniquely identifies a row. If a table contains a large object it must also have a ROWID column. (DB2 Version 6.1 and later only)
HEXADEC	Fixed length string, supported as a string only.
FLTPIC	Fixed length character string representing a floating point value.
FIXPIC	Fixed length character string representing a decimal or integer value.

Table 18-3. Column/Field Types Mapping

TO ⇒ FROM ⇓	**CHAR**	**VCHAR**	**LVCHAR**	**GRPH**	**VGRPH**	**LVGRPH**	**SMALLINT**	**INTEGER**	**DEC**	**REAL**	**DOUBLE**	**DATE**	**TIME**	**TIMESTAMP**	**HEXADEC**	**FLTPIC**	**FIXPIC**
CHAR	X	X	X				X	X	X								
VCHAR	X	X	X				X	X	X								
LVCHAR	X	X	X				X	X	X								
GRPH				X	X	X											
VGRPH				X	X	X											
LVGRPH				X	X	X											
SMALLINT	X	X	X				X	X	X								
INTEGER	X	X	X				X	X	X								
DEC	X	X	X				X	X	X								
REAL										X							
DOUBLE											X						
DATE												X					
TIME													X				

Table 18-3. Column/Field Types Mapping

TO ⇒ FROM ⇓	CHAR	VCHAR	LVCHAR	GRPH	VGRPH	LVGRPH	SMALLINT	INTEGER	DEC	REAL	DOUBLE	DATE	TIME	TIMESTAMP	HEXADEC	FLTPIC	FIXPIC
TIMESTAMP														X			
HEXADEC															X		
FLTPIC																X	
FIXPIC																	X

Mapped Fields

Displays a value only for already defined relationships in the Parent portion. It lists

- the column/field number of the dependent column/field,
- the extended column information in parentheses (parent start position, length, dependent start position, length) where applicable
- the dependent column/field name.

Dependent Portion:

The dependent portion of the Relationships screen contains the following information:

CMD

Displays File-AID *for DB2*-generated sequence numbers that indicate the dependent column/field/segment numbers to be used as input in the above parent CMD field to establish a new relationship. This number also correlates to the dependent column/field/segment number in the Mapped Fields values for defined relationships.

Dependent Columns/Fields/Root Segment

Displays the names of the columns/fields or IMS root segment in the dependent object. If an MVS field contains an occurrence of subscripts, the element notations follows in parenthesis.

Type(Length)

Displays the dependent column's or field's data type. The length of the data is displayed in parenthesis immediately following the data type for CHAR, VCAR, GRPH, VGRPH, and DEC data types only.

See Table 18-2 on page 18-52 for more information about data types displayed in this column. Table 18-3 identifies the allowable column/field type mapping when building a column relationship.

Notes:

1. If a DB2 column is a distinct type (UDT) File-AID *for DB2* displays the built-in column type only.
2. Columns with Large Object Type (CLOB, BLOB, DBCLOB) and/or ROWID column types cannot be part of an application relationship.

Line Commands

The following File-AID *for DB2*-specific line commands are valid on the Relationships screen:

number

Establishes an application relationship (AR) between a parent column/field and a dependent column/field/root segment.

The panel names the number line command according to the object type of the dependent:

Column-Number	when the dependent is a DB2 table.
Field-Number	when the dependent is an MVS file.
Root Segment-Number	when the dependent is an IMS database.

Enter a File-AID *for DB2*-generated column/field/segment number (listed in the dependent **CMD** column) in the parent **CMD** column next to the parent column/field for which you want to establish a relationship.

For example, in Figure 18-26 on page 18-56 a Field-Number line command (1) has been entered to relate the dependent field, SH-CUST-NUM to the parent column, CUSTOMER_NUMBER. When you press **Enter**, the new relationship is established and the Column/Field Relationships screen is redisplayed with the new relationship above the dashed line as shown in Figure 18-28 on page 18-57.

D (Delete)

Enter **D** in the CMD column to remove an existing column/field relationship from the application relationship.

Primary Commands

The following File-AID *for DB2*-specific primary commands are valid on the Relationships screen:

CANCEL (CAN)

Quits the Relationship Definition without saving any changes.

SUGGEST (SUG)

Enter this command and File-AID *for DB2* will suggest a relationship when the parent and dependent column or field name, type and length are identical. Same as if you had entered the column or field number(s) manually for all identical column or field matches. You only need to enter the SUGGEST command when you didn't enable it in the previous "Add Relationships Screen" or when you want File-AID *for DB2* to suggest matching pairs again.

Use the **D** line command to delete any suggested relationships you don't want included. Use the **CANCEL** primary command to abort the relationship definition.

Building Column/Field Relationships

File-AID *for DB2* enables you to modify the relationship file by modifying existing application relationships or creating new ones. To modify or create application relationships, you modify or create column/field relationships on the Relationships screen. An application relationship can consist of one or more column/field mappings.

The parent and dependent portions of the Relationships screen comprise two lists. These lists contain names, data types, and lengths of columns or fields from the parent and dependent objects specified in the preceding screen. Use these lists to build the relationships at the column/field level.

Note: To build application relationships to be used with existing DB2 relationships, you must be connected to the subsystem for which the existing relationships are defined. You can specify a subsystem in the **SSID** field on the Application Relationships screen shown in Figure 18-21 on page 18-42.

To build column/field relationships:

1. Decide which parent and dependent columns or fields you want to be related. It is likely that these will be columns or fields that your application recognizes as containing data values that are used as an index into the corresponding object in the relationship.
2. Enter the number from the dependent **CMD** column that corresponds to the dependent column/field you chose in step 1 in the parent **CMD** column next to the name of the parent column/field you chose in step 1. See Figure 18-26.

   Refer to Table 18-3 on page 18-53 to see which data types can be paired to establish an application relation. For other pairings File-AID *for DB2* displays an error message and will not establish the relationship.

Figure 18-26. Relationships Screen — Creating a DB2 to MVS Relationship

```
File-AID for DB2 -----  Column/Field Relationships  --------------------------
Command ===>                                                  SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.CUSTOMER_TABLE - DB2
Dependent: TSOID01.FARDX2.SAMP.SALEHIST - MVS
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
1___ CUSTOMER_NUMBER                 CHAR(6)
____ COMPANY_NAME                    CHAR(30)
____ ADDRESS                         CHAR(30)
____ CITY                            CHAR(25)
____ STATE                           CHAR(2)
____ ZIP_CODE                        CHAR(9)
CMD  Dependent fields                Type(Length)
---- ------------------------------- ------------
   1 SH-CUST-NO                      CHAR(6)
   2 SH-ORD-NO                       CHAR(6)

 RDXI753 I753 RDX was unable to find parent/child columns with the same name,
 type, and length.

   6 SH-ORD-LINE-NO                  DEC(5,0)
```

3. Press **Enter**.

   If one or both parent and dependent fields in an MVS file contain subscripts, File-AID *for DB2* will need to know which occurrence of subscript to use for the relationship. In that case, File-AID *for DB2* continues with "Specify MVS Field Subscripts" on page 18-65.

When partial column mapping is allowed between the parent and dependent columns/fields, the Extended Column Relationship Definition pop-up window is displayed as shown in Figure 18-27 on page 18-57. See "Extended Column Relationship Definition" on page 18-58 for more information about extended column relationships.

Figure 18-27. Extended Column Relationship Definition Window

```
File-AID for DB2 -----  Column/Field Relationships  --------------------------
Command ===>                                                 SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.CUSTOMER_TABLE - DB2
          Extended Column Relationship Definition

 COMMAND ===>


 Enter Start Position and Length for a partial-column relationship definition.

 Parent Field:                    Dependent Field:
   Name        : CUSTOMER_NUMBER      Name        : SH-CUST-NO
   Type(Length): CHAR(6)              Type(Length): CHAR(6)



  Start Position ===> 1              Start Position ===> 1
  Length         ===> 6

 Enter CANCEL to end without defining the extended column relationship
```

If partial column mapping is not applicable for the selected parent to dependent pairing the relationship is established immediately. File-AID *for DB2* positions the name of the parent column/field of the relationship you just created to the end of the list of established column relationships as shown in Figure 18-28. Note that the new column relationship is located above the dashed "End of Related Columns" line. In addition, the Mapped Fields value is filled with the dependent column/field number that appeared in the **CMD** column next to the dependent column/field as well as any applicable extended column information in parentheses, followed by the dependent column/field name.

Figure 18-28. Relationships Screen — New DB2 Column to MVS Field Relationship Established.

```
File-AID for DB2 ----  Column/Field Relationships  --------------------------
Command ===>                                                 SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: LOCATION.TSOID01.CUSTOMER_TABLE - DB2
Dependent: TSOID01.FARDX2.SAMP.SALEHIST - MVS
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
____ CUSTOMER_NUMBER                 CHAR(6)      1/SH-CUST-NO
     - - - - - - - - - End of Related Columns - - - - - - - - -
____ CUSTOMER_NUMBER                 CHAR(6)
____ COMPANY_NAME                    CHAR(30)
____ ADDRESS                         CHAR(30)
____ CITY                            CHAR(25)
____ STATE                           CHAR(2)
____ ZIP_CODE                        CHAR(9)
CMD  Dependent fields                Type(Length)
---- ------------------------------- ------------
   1 SH-CUST-NO                      CHAR(6)
   2 SH-ORD-NO                       CHAR(6)
   3 SH-ORD_TYPE                     CHAR(8)
   4 SH-DATE-SHIP                    CHAR(8)
   5 SH-SHIP-CUST-NO                 CHAR(6)
```

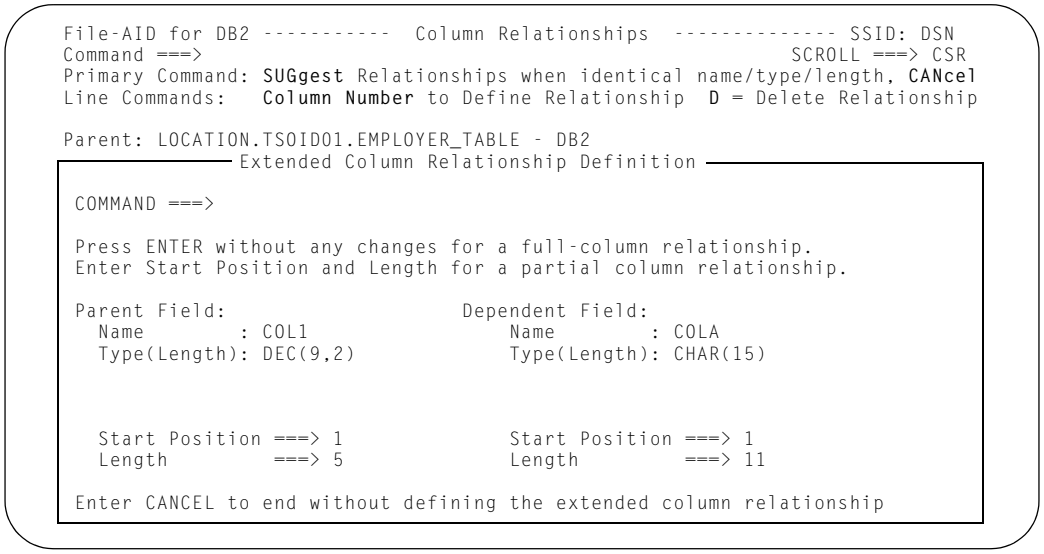
4. Repeat steps 1-3 for all columns/fields that you want to include in the application relationship for the pair of objects.

When you exit from this screen, File-AID *for DB2* writes a new or modified application relationship to the relationship file. File-AID *for DB2* redisplays the Application Relationship Detail screen, where the completed action is indicated in the **Status** column with the "Added" or "Modified" message.

Extended Column Relationship Definition

The Extended Column Relationship Definition pop-up window shown in Figure 18-29 is displayed when you attempt to map two columns/fields containing character (CHAR or VCHAR) or graphic (GRPH or VGRPH) data, or one column/field containing DECIMAL and its parent or dependent CHAR data.

Figure 18-29. Extended Column Relationship Definition Window



Use this window to create **partial-column/field** relationships.

If the columns you want to map are the same length, the fields in this window are prefilled and the following message is displayed:

```
Press ENTER without any changes for a full-column relationship
```

Complete this window and press **Enter** to proceed.

The left side of the body of this window contains fields that pertain to the **parent column/field**:

Name

Displays the name of the parent column/field.

Type(Length)

Displays the type of data contained in the parent column/field. The length, in parentheses, follows the data type. If an MVS field includes an occurrence of subscripts, the ARRAY SIZE and MAX LENGTH values are also displayed, see Figure 18-41 on page 18-67 for an example.

Start Position

Specify the starting position of the string in the parent column/field that will comprise the column/field relationship.

For DECIMAL columns/fields the Start Position value cannot be changed.

Length

Specify the length of the string contained in the parent column/field on which the column/field relationship will be based.

For DECIMAL data types the Length value represents its internal byte length and cannot be changed.

The right side of the body of this window contains fields that pertain to the **dependent column/field**:

Name

Displays the name of the dependent column/field.

Type(Length)

Displays the type of data contained in the dependent column/field. The length, in parentheses, follows the data type. If an MVS field includes an occurrence of subscripts, the ARRAY SIZE and MAX LENGTH values are also displayed, see Figure 18-41 on page 18-67 for an example.

Start Position

Specify the starting position of the string in the dependent column/field that will comprise the column/field relationship. If the right side does not display a Length field, the length of the dependent column/field is determined by the length of the string in the corresponding parent column/field.

For DECIMAL columns the Start Position value cannot be changed.

Length

(displays only for DECIMAL to CHAR relationships) Specify the length of the converted decimal string in the dependent character column/field. The maximum length of a DECIMAL to CHAR conversion is "precision + sign + decimal delimiter". For example, maximum CHAR length for DEC(9,2) is 11 (9 + 1 + 1).

Rules for DECIMAL and CHARACTER Columns/Fields

Mapping a decimal to a character column/field:

1. The decimal column/field is always mapped in its entirety. Start position and length cannot be modified.
2. The character column/field start position and length can be changed but cannot exceed the maximum displayable length of the decimal column/field. For example, if decimal parent format is DEC(9,2), the maximum map length of the dependent character column/field is 11 (9 plus 1 for the decimal point plus 1 for a possible minus sign).
3. The default dependent character column/field length is the lesser of the maximum displayable length of the parent decimal column/field or the dependent character column/field length.

Mapping a character to a decimal column/field:

1. The decimal column/field is always mapped in its entirety. Start position and length cannot be modified.
2. The character column/field start position and length can be changed but cannot exceed the maximum displayable length of the decimal column/field. For example, if

decimal parent format is DEC(9,2), the maximum map length of the dependent character column/field is 11 (9 plus 1 for the decimal point plus 1 for a possible minus sign).

3. The default parent character column/field length is the lesser of the maximum displayable length of the decimal column/field or the character column/field length.
4. The default parent character column/field start position is 1.

Converting a decimal to a character column/field:

1. The string values derived from decimal column values have these characteristics:
   - Leading zeroes are suppressed
   - For values less than 1, a leading zero is inserted before the decimal point (or comma), for example 0.25
   - For values less than 0, a minus sign is the left most character in the returned string
   - For values greater than 0, a plus sign is **not** returned
   - Special characters, such as monetary symbols, are not returned
   - Where the scale of the decimal column/field is zero, for example DEC(5,0), **no decimal point** is in the returned string
2. Where the dependent character column/field map length is less than the returned character string, truncation occurs on the left of the returned string.

Converting a character to a decimal column/field:

The string values to be converted to decimal values must comply with these constraints:

- The minus or plus sign must be the left most character of the string
- The placement of the decimal point in the string is interpreted, but the scale of the decimal value is derived from the decimal column/field definition. If, for example, the decimal definition is DEC(9,2) and the string to be converted is 99.999, the last decimal position is truncated.
- If the string does not contain a decimal point and the scale is greater than zero, zeroes will be put into the decimal positions of the decimal value. For example, character string 1234 would be converted to 1234.000 for a DEC(7,3) column.
- The character string cannot contain values other than numeric values, sign or decimal delimiter (point or comma)

Extended Column Relationships Examples

The following describes several situations in which the Extended Column Relationship Definition window is displayed.

Example 1

Assume you have entered a column-number line command (as illustrated in Figure 18-30) to create a DB2 to DB2 column relationship between two columns with data types of CHAR and lengths of 6.

Figure 18-30. Relationship Screen with column-number command

```
File-AID for DB2 ----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: Suggest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
Dependent: LOCATION.FRSAMP.ORDER_TABLE - DB2
CMD  Parent Columns                   Type(Length) Mapped Fields
---- -------------------------------- ------------ ------------------------------
2___ CUSTOMER_NUMBER                  CHAR(6)
____ COMPANY_NAME                     CHAR(30)
____ ADDRESS                          CHAR(30)
____ CITY                             CHAR(25)
____ STATE                            CHAR(2)
____ ZIP_CODE                         CHAR(9)
____ COUNTRY                          CHAR(20)
CMD  Dependent Columns                Type(Length)
---- -------------------------------- ------------
   1 ORDER_NUMBER                     CHAR(6)
   2 CUST_NUM                         CHAR(6)
   3 SOC_SEC_NUM                      CHAR(11)
   4 CREDIT_CARD_NUM                  CHAR(16)
   5 ORD_TYPE                         CHAR(8)
   6 ORD_DATE                         DATE
```

The Extended Column Relationship Definition pop-up window would be displayed as shown in Figure 18-31 with default values for starting position and length prefilled.

Figure 18-31. Extended Column Relationship Definition Window

```
File-AID for DB2 -----------  Column Relationships  -------------- SSID: DSN
Command ===>                                                  SCROLL ===> PAGE
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Column Number to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
          Extended Column Relationship Definition
 COMMAND ===>

 Press ENTER without any changes for a full-column relationship
 Enter Start Position and Length for a partial-column relationship

 Parent Field:                    Dependent Field:
   Name        : CUSTOMER_NUMBER     Name        : CUST_NUM
   Type(Length): CHAR(6)             Type(Length): CHAR(6)



   Start Position ===> 1             Start Position ===> 1
   Length         ===> 6

 Enter CANCEL to end without defining the extended column relationship
```

Note: When you select parent and dependent columns with the same data type and length, the Extended Column Relationship Definition window is prefilled and the following message is displayed:

```
Press ENTER without any changes for a full-column relationship
```

Example 2

Assume you have entered a column-number line command (as illustrated in Figure 18-32) to create a DB2 to IMS relationship between columns with data types of CHAR and different lengths.

Figure 18-32. Column/Segment Relationships Screen with Segment-Number command

```
File-AID for DB2 ------  Column/Segment Relationships  ------------ SSID: DSN
Command ===>                                                  SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Root segment  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
Dependent: TSOID01.FISAMP.DBDLIB(PCUST) - IMS
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
1___ CUSTOMER_NUMBER                 CHAR(6)
____ COMPANY_NAME                    CHAR(30)
____ ADDRESS                         CHAR(30)
____ CITY                            CHAR(25)
____ STATE                           CHAR(2)
____ ZIP_CODE                        CHAR(9)
____ COUNTRY                         CHAR(20)
____ AREA_CODE                       CHAR(3)
____ TELEPHONE_NUM                   CHAR(7)
____ CONTACT_NAME                    CHAR(30)
____ CONTACT_TITLE                   CHAR(30)
____ CONTACT_ADDR                    CHAR(30)
CMD  Dependent Root segment          Type(Length)
---- ------------------------------- ------------
   1 CUST010                         CHAR(6)
```

The Extended Column Relationship Definition pop-up window would be displayed as shown in Figure 18-33 with default values for starting position and length prefilled.

Figure 18-33. Extended Column Relationship Definition Window

```
File-AID for DB2 -------  Column/Segment Relationships  ------------------------
Command ===>                                                  SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Root Segment  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
          Extended Column Relationship Definition
 COMMAND ===>


 Enter Start Position and Length for a partial column relationship.

 Parent Field:                    Dependent Field:
   Name        : CUSTOMER_NUMBER      Name        : CUST010
   Type(Length): CHAR(6)              Type(Length): CHAR(6)

                 MAX LENGTH 6

   Start Position ===> 1              Start Position ===> 1
   Length         ===> 6

 Enter CANCEL to end without defining the extended column relationship
```

Change the Dependent Root Segment's Start Position to 5 and the Parent Column Length to 2. Figure 18-34 on page 18-63 indicates that the first two characters in the parent column and the last two characters in the dependent root segment will make up the column/segment relationship.

Figure 18-34. Changed Dependent Start Position and Length

```
File-AID for DB2 -------  Column/Segment Relationships  ------------------------
Command ===>                                                 SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Root Segment  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
          Extended Column Relationship Definition

 COMMAND ===>


 Enter Start Position and Length for a partial column relationship.

 Parent Field:                    Dependent Field:
   Name        : CUSTOMER_NUMBER      Name        : CUST010
   Type(Length): CHAR(6)              Type(Length): CHAR(6)

                 MAX LENGTH 6

   Start Position ===> 1              Start Position ===> 5
   Length         ===> 2

 Enter CANCEL to end without defining the extended column relationship
```

When you press Enter the Column/Segment Relationship screen reflects the desired relationship (see Figure 18-35).

Figure 18-35. Partial Column/Segment Relation Established

```
File-AID for DB2 ------  Column/Segment Relationships  ------------------------
Command ===>                                                 SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Root Segment  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.CUSTOMER_TABLE - DB2
Dependent: TSOID01.FISAMP.DBDLIB(PCUST) - IMS
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
____ CUSTOMER_NUMBER                 CHAR(6)      1(1,2,5)/CUST010
     - - - - - - - - - End of Related Columns - - - - - - - - -
____ CUSTOMER_NUMBER                 CHAR(6)
____ COMPANY_NAME                    CHAR(30)
____ ADDRESS                         CHAR(30)
____ CITY                            CHAR(25)
____ STATE                           CHAR(2)
____ ZIP_CODE                        CHAR(9)
____ COUNTRY                         CHAR(20)
____ AREA_CODE                       CHAR(3)
____ TELEPHONE_NUM                   CHAR(7)
____ CONTACT_NAME                    CHAR(30)
CMD  Dependent Root segment          Type(Length)
---- ------------------------------- ------------
   1 CUST010                         CHAR(6)
```

Now the Mapped Fields value includes the extended column relationship information in parentheses (1,2,5) where 1 is the parent starting position, 2 is the length, and 5 is the dependent starting position.

Example 3

Assume you have entered a Field-Number line command (as illustrated in Figure 18-36) to create a DB2 to MVS relationship between a column with data type of DECIMAL and a field with data type CHAR.

Figure 18-36. Column/Field Relationships Screen Mapping DECIMAL to CHAR

```
File-AID for DB2 -------  Column/Field Relationships  -------------------------
Command ===>                                                  SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.ORDER_TABLE - DB2
Dependent: FR.BASE.V2R0M0.SAMP.INVOICE - MVS
CMD  Parent Columns                  Type(Length) Mapped Fields
---- ------------------------------- ------------ ------------------------------
____ SOC_SEC_NUM                     CHAR(11)
____ CREDIT_CARD_NUM                 CHAR(16)
____ ORD_TYPE                        CHAR(8)
____ ORD_DATE                        DATE
____ ORD_STAT                        CHAR(8)
10__ ORD_AMOUNT                      DEC(9,2)
____ ORD_DEPOSIT                     DEC(9,2)
CMD  Dependent fields                Type(Length)
---- ------------------------------- ------------
   8 IM-INVOICE-TOT                  DEC(9,2)
   9 IM-SHIP-CHG                     DEC(7,2)
  10 IM-SHIP-VIA                     CHAR(20)
  11 IM-SHIP-CODE                    CHAR(4)
  12 IM-SHIP-TO-CUST                 CHAR(6)
  13 IM-ORDER-NO                     CHAR(6)
```

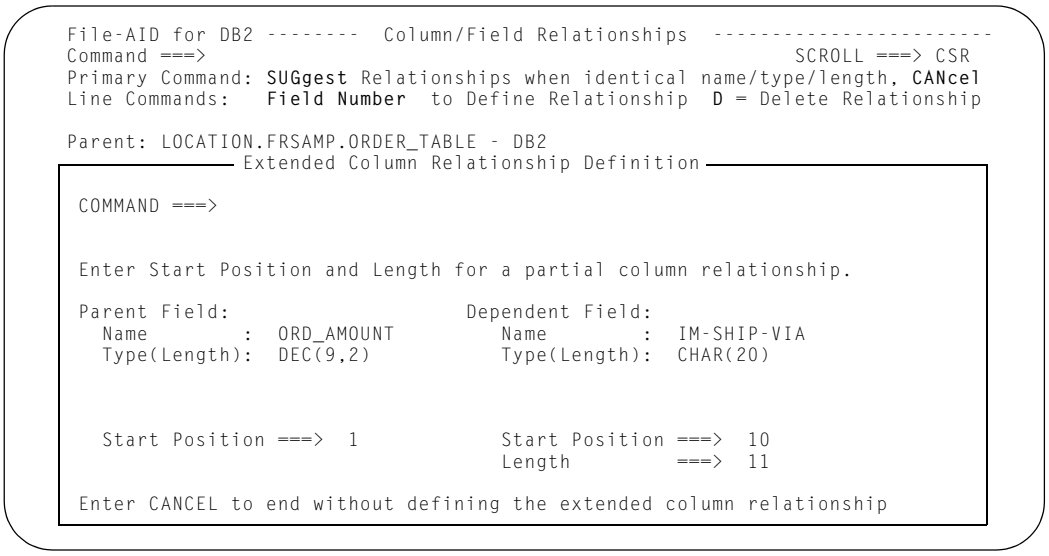
The Extended Column Relationship Definition pop-up window would be displayed as shown in Figure 18-37 with default values for starting position and length prefilled:

Figure 18-37. Extended Column Relationship Definition Window with DECIMAL to CHAR

```
File-AID for DB2 --------  Column/Field Relationships  -------------------------
Command ===>                                                  SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: LOCATION.FRSAMP.ORDER_TABLE - DB2
          Extended Column Relationship Definition
 COMMAND ===>


 Enter Start Position and Length for a partial column relationship.

 Parent Field:                    Dependent Field:
   Name        :  ORD_AMOUNT         Name        :  IM-SHIP-VIA
   Type(Length):  DEC(9,2)           Type(Length):  CHAR(20)



   Start Position ===>  1            Start Position ===>  1
                                     Length         ===>  11

 Enter CANCEL to end without defining the extended column relationship
```

The values entered in the window in Figure 18-38 on page 18-65 indicate that the complete parent decimal column and the last eleven characters in the dependent column (starting position 10, length 11) will make up the column/field relationship.

Figure 18-38. Extended Column Relationship Definition Window after Modification

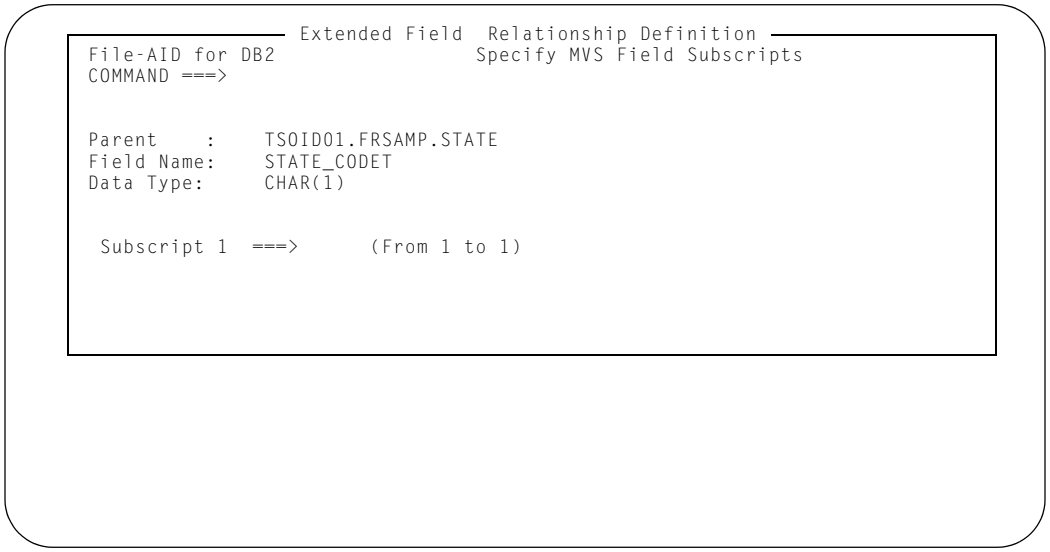


Press ENTER to complete the extended relationship definition.

Specify MVS Field Subscripts

The Specify MVS Field Subscripts pop-up window shown in Figure 18-39 is displayed when you attempt to map two fields where one or both fields contain an occurrence of subscript. You then specify the bounds of the array for each subscript. File-AID *for DB2* supports a maximum occurrence of a three dimensional array.

Figure 18-39. Specify MVS Field Subscripts for Parent Field



This window appears only if the selected parent field contains subscripts. Use it to specify the desired value for up to three subscripts, where applicable.

Parent

Displays the name of the parent file.

Field Name

Displays the name of the parent field to be mapped.

Data Type

Displays the type of data contained in the field. The length, in parentheses, follows the data type.

Subscript 1

Specify the starting position of substring 1 in the parent field that will comprise the column/field relationship.

Subscript 2

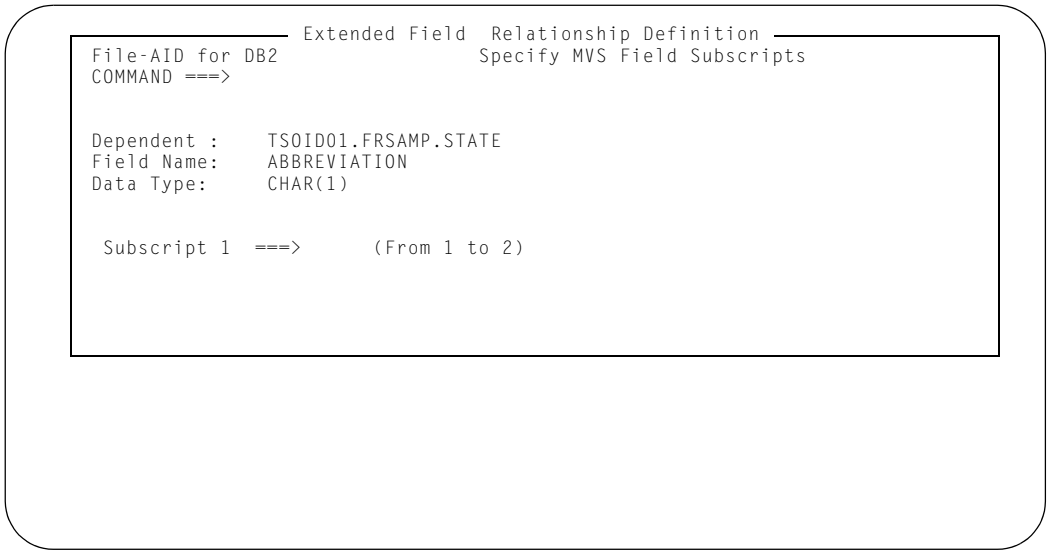
Specify the starting position of substring 2 in the parent field that will comprise the column/field relationship.

Subscript 3

Specify the starting position of substring 3 in the parent field that will comprise the column/field relationship.

After you press Enter, you continue either with the Dependent Field Subscripts window (Figure 18-40 on page 18-66), if the dependent field contains subscripts, or the Extended Field Relationship Definition window (Figure 18-41 on page 18-67), if the parent and dependent field types qualify for Extended Field Relationship Definition, or return directly to the Field/Column Relationship screen (see Figure 18-42 on page 18-68) when the relationship definition is completed.

Figure 18-40. Specify MVS Fleld Subscripts for Dependent Field



Use this window to specify the desired value for up to three subscripts, if applicable, for the dependent field.

Dependent

Displays the name of the dependent file.

Field Name

Displays the name of the dependent field to be mapped.

Data Type

Displays the type of data contained in the field. The length, in parentheses, follows the data type.

Subscript 1

Specify the starting position of substring 1 in the parent or dependent field that will comprise the column/field relationship.

Subscript 2

Specify the starting position of substring 2 in the parent or dependent field that will comprise the column/field relationship.

Subscript 3

Specify the starting position of substring 3 in the parent or dependent field that will comprise the column/field relationship.

After you press Enter, you continue either with the Extended Field Relationship Definition window (Figure 18-41 on page 18-67), if the parent and dependent field types qualify for Extended Field Relationship Definition, or return directly to the Field/Column Relationship screen (see Figure 18-42 on page 18-68) when the relationship definition is completed.

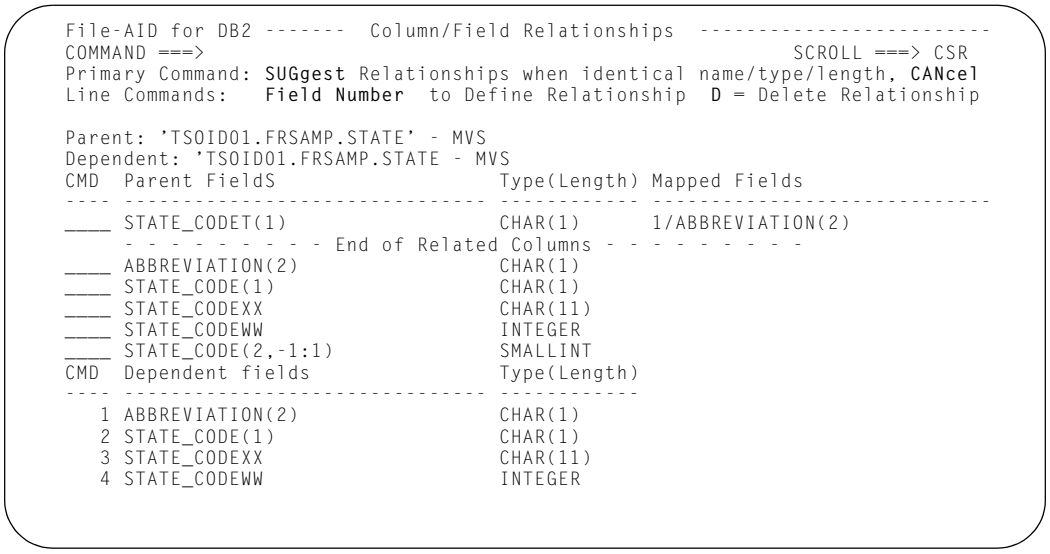
Figure 18-41. Extended Field Relationship Definition with Field Subscripts Defined

```
File-AID for DB2 -----------  Field Relationships  ------------------------------
Command ===>                                                   SCROLL ===> CSR
Primary Command: SUGgest Relationships when identical name/type/length, CANcel
Line Commands:   Field Number  to Define Relationship  D = Delete Relationship

Parent: 'TSOID01.FRSAMP.STATE' - MVS
              Extended Field  Relationship Definition
 COMMAND ===>


 Enter Start Position and Length for a partial-column relationship.

 Parent Field:                    Dependent Field:
   Name        :  STATE_CODET(1)     Name        :  ABBREVIATION(2)
   Type(Length):  CHAR(1)            Type(Length):  CHAR(1)
                  ARRAY SIZE(1)                     ARRAY SIZE(2)
                  MAX LENGTH 1

   Start Position ===>  1            Start Position ===>  1
   Length         ===>  1

 Enter CANCEL to end without defining the extended column relationship
```

See "Extended Column Relationship Definition" on page 18-58 for a description of the fields. Note that the Type(Length) field now also includes the array size and maximum length of the related elements.

Upon completion, File-AID *for DB2* displays the defined relationship in the Field Relationships screen as shown in Figure 18-42.

Figure 18-42. Extended Field Relationship with Defined Subscripted Relationship



See "Relationships" on page 18-50 for a description of the fields. Note that the element notation for the parent is enclosed in parentheses following the parent field name, while the element notation for the dependent is enclosed in parentheses following the dependent name in the Mapped Fields area.

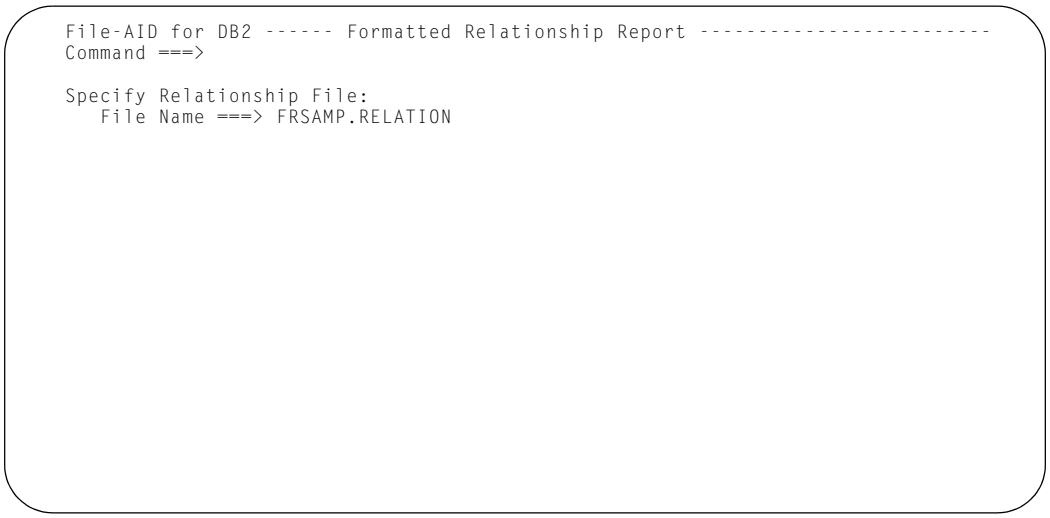
Print Relationship File

The Print Relationships function enables you to generate a formatted report containing the contents of a relationship file. You can print the report or write the report to a file.

Formatted Relationship Report Screen

The Formatted Relationship Report screen shown in Figure 18-43 is displayed when you select Option P in the "DB2 Relationship Facilitator" or "MVS Relationship Facilitator" menu.

Figure 18-43. Formatted Relationship Report Screen



Use this screen to specify the relationship file for which you want to generate a report.

The body of the Formatted Relationship Report screen consists of the following field:

File Name

Specify the name of the relationship file for which you want to generate a formatted report.

Print Options Screen

The Print Options screen shown in Figure 18-44 is displayed when you complete the Formatted Relationship Report screen and press **Enter**.

Figure 18-44. Print Options Screen

```
File-AID for DB2 -------------- Print Options ---------------------------------
Command ===>

Maximum lines per page ===> 55       (20-99)

Specify Print Options:
   Printer Destination ===> LOCAL
   Sysout Class        ===> R
      - OR -
   "TO" Print file     ===>
     Member            ===>
     VOLSER            ===>
     Disposition       ===>          (NEW, OLD, or MOD)

If disposition is NEW, specify the following allocation parameters:
   Generic Unit        ===> SYSDA
   Block Size          ===> 141      (DSORG=PS,RECFM=VBA,LRECL=137)
   Space Units         ===> TRK      (BLK, TRK, OR CYL)
   Primary QTY         ===> 10       (In above units)
   Secondary QTY       ===> 2        (In above units)

                       ENTER to view JCL
```

Use this screen to specify whether you want to route the report to a printer or a file. If you route the report to a file, use this screen to specify a file and its characteristics.

Maximum lines per page

Specify the maximum number of lines to be printed on each page. The value specified must be in the range 20-99.

If you want to route the report to a printer, specify appropriate values in the following two fields:

Printer Destination

Specify the destination printer.

SYSOUT Class

Specify the SYSOUT class.

If you want to route the report to a file, specify the appropriate values in the following fields:

"TO" Print file

Specify the name of the file to which the report will be written.

Member

If the file you specified in "TO" Print file is partitioned, you can enter a member name.

VOLSER

If the file specified in TO Print file is non-cataloged or new, specify the volume serial number of the file. For non-cataloged files, a volume serial number is required. For new files, a volume serial number is optional.

Disposition

Specify the disposition of the TO Print file. You can write to existing files with a disposition of OLD or MOD. Specify a disposition of NEW to create a new sequential file.

If you specify NEW in the Disposition field, you must specify the following allocation parameters:

Generic Unit

Specify the device type on which the new file will reside. The names of these devices are installation dependent.

Block Size

Specify the block size for the new file.

Space Units

Specify whether space for the new file will be allocated in blocks (BLK), tracks (TRK), or cylinders (CYL).

Primary QTY

Specify the initial number of space units that will be allocated for the new file.

Secondary QTY

Specify the number of space units that will be allocated for each secondary extent.

Edit Print JCL

The ISPF/PDF Edit screen shown in Figure 18-45 is displayed when you pressed **Enter** on the Print Options screen.

Figure 18-45. ISPF/PDF EDIT Screen — File-AID *for DB2*-generated JCL

```
EDIT - File-AID for DB2 Generated JCL                          COLUMNS 000 000
Command ===>                                                  SCROLL ===> CSR
    You are now editing the JCL that File-AID for DB2 has generated to perform
    the required function. Change this JCL as desired before submitting.

    You should review the generated DDL statements, if present, to ensure
    all user requirements have been met.
****** ***************************** Top of Data ******************************
000001 //*Your Jobcard information
000002 //*
000003 //*
000004 //*
000005 //*********************************************************************
000006 //* PRINT FORMATTED REPORT
000007 //*********************************************************************
000008 //RELPRINT     EXEC PGM=RELPRINT
000009 //STEPLIB  DD DISP=SHR,
000010 //            DSN=FADB2.V4R8Mx.ISPLLIB
000011 //SYSUDUMP DD SYSOUT=*
000012 //SYSPRINT DD SYSOUT=*
000013 //SYSTERM  DD SYSOUT=*
000014 //SYSOUT   DD SYSOUT=*
000015 //RDXPARMS DD  *
000016 LINES     55
```

Use this screen to edit and/or submit the print JCL generated by File-AID *for DB2*. If you want to save this JCL, use the ISPF/PDF CREATE or REPLACE command before you exit the panel. Otherwise, File-AID *for DB2* deletes this file.

Note: SYNCSORT users: In step SORTTMP, second step, change DDNAME DFSPARM to SYSIN.

Submit this JCL to perform the print.

Formatted Relationship Report

The Formatted Relationship Report shown in Figure 18-46 is a summary of the relationships contained in the relationship file you specified on the Formatted Relationship Report screen. The report is organized such that for each object involved in a relationship you can see information about the object and its direct relationships.

Figure 18-46. Formatted Relationship Report

```
File-AID for DB2                          Relationships Listing Report
                                                                                DATE 2000-01-07
                                                                                TIME   15.47.7

Relationship file:  'TSOID01.FRSAMP25.RELATE'
Obj                                                          Par/  Rel
Type    Object Name/Related Object Name                      Child Type   Description
--- ---------------------------------------------------------- -    -- ---------------------------------------------------

DB2 DSNLOCAN.TSOID01.table_name_up_to_128_characters_long_wi P    RI MIXRLS
    th_location_and_creator_the_complete_object_name_can_be_
    up_to_266_characters_long
        LOCATION.BFHDJS1.MIX_CASE_UE                           C    RI MIXRLS

MVS TSOID01.FARDX2.SAMP.SALEHIST                             P    AR
MVS     TSOID01.FARDX2.SAMP.CREDIT                           C    AR

MVS TSOID01.FARDX2.SAMP.ZIPCODE                              P    AR
IMS     FR.IXPSAMP.DBDLIB(PCUST)                             C    AR
DB2     .TSOID01.DEC30                                       C    AR

DB2 LOCATION.SYSIBM.SYSPACKAGE                               P    RI DSNKK@KD
DB2     LOCATION.SYSIBM.SYSPACKDEP                           C    RI DSNKK@KD
DB2     LOCATION.SYSIBM.SYSPACKSTMT                          C    RI DSNKK@KS
DB2     LOCATION.SYSIBM.SYSPKSYSTEM                          C    RI DSNKK@KY

DB2 LOCATION.SYSIBM.SYSCOLUMNS                               P    RI DSNDC@TN
DB2     LOCATION.SYSIBM.SYSCOLDIST                           C    RI DSNDC@TN
DB2     LOCATION.SYSIBM.SYSCOLDISTSTATS                      C    RI DSNDC@TP
DB2     LOCATION.SYSIBM.SYSCOLSTATS                          C    RI DSNDC@TC
DB2     LOCATION.SYSIBM.SYSFIELDS                            C    RI DSNDF#FD

------------------------------------------------------------------------------

DB2     LOCATION.SYSIBM.SYSCOLUMNS                           C    RI
DB2 LOCATION.SYSIBM.SYSTABLES                                P    RI

DB2     LOCATION.SYSIBM.SYSFIELDS                            C    RI
DB2 LOCATION.SYSIBM.SYSCOLUMNS                               P    RI

DB2     LOCATION.SYSIBM.SYSINDEXES                           C    RI
DB2 LOCATION.SYSIBM.SYSTABLES                                P    RI

MVS TSOID01.FARDX2.SAMP.SALEHIST                             P    AR
MVS     TSOID01.FARDX2.SAMP.CREDIT                           C    AR

MVS TSOID01.FARDX2.SAMP.ZIPCODE                              P    AR
IMS     FR.IXPSAMP.DBDLIB(PCUST)                             C    AR
DB2     LOCATION.TSOID01.DEC30                               C    AR

MVS PFHPACO.FASAMP.ORDRFILE                                  P    AR
```

The report groups related objects. In the first part, the groups are made up with the parent object listed first, followed by all of its child objects. In the second part, the groups are made up with the child object listed first, followed by all of its parent objects. If an object is listed more than once in a group as a parent or child, it reflects the number of relationships that exist between those objects. The groups are listed in alphabetical order.

The Formatted Relationship Report consists of the following fields:

Obj Type

Identifies the object type of the relationship objects:

DB2

IMS

MVS

Object Name/Related Object Name

Name of object. If it is indented it is a dependent (child). DB2 Version 8 long object names are wrapped within the Object Name column.

Par / Child

Identifies whether the object is the parent or the child in the relationship.

Rel Type

Displays the type of relationship that exists between the parent and dependent object.

RI Referential Integrity relationship

AR Application Relationship

Description

Displays the defined relationship description

Appendix A.
Predicate Analysis Rules Abbreviations

This appendix contains a complete list of the abbreviations used in the short and long messages you may receive when using Predicate Analysis.

Table A-1. Abbreviations Used in Rule Recommendations

Abbreviation	Explanation
LHS	left-hand side of a predicate
RHS	right-hand side of a predicate
RHS2	the second operand on the right-hand side of a predicate such as column C in the predicate A between B and C or A like B escape C.
col	column
hostvar, hv	host variable
expr	expression
IntegerArithmeticConstant	arithmetic expression that contains only integer constants - for example, 10*5*2
op	operator - often, this operator is restricted to an arithmetic operator. The detailed message has more information about any restrictions.
intConst	integer constant
colExpr	expression that includes columns - for example, A*10+2.0
noncolexpr, noncolexp	expression that does not include columns - for example, 10*2.0
(...)	an IN list
noncorr_subq	noncorrelated subquery
corr_subq	correlated subquery
quantifiers	ALL, ANY, SOME, or DISTINCT
(C1,C2,..Cn)	A list of columns for DB2.

Appendix B.
Action Bar and Pull-Downs Summary

The following figures and tables provide a summary of all Action Bar items and Pull-Down menu choices for File-AID *for DB2*. For a general overview of the Action Bar, see "Action Bar" on page 2-6.

Edit Action Bar Items and Pull-Downs

These Action Bar items are available for the Edit and Browse panels:

- File
- Edit
- View
- Options
- Open
- Menu
- Utilities
- Help

Figure B-1 shows the File-AID *for DB2* Edit action bar.

Figure B-1. Action Bar Example - Row Mode

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit   FR.ORDER_TABLE                  SSID: DSN
COMMAND ===> FIND                                              SCROLL ===> CSR
Row 1
    COLUMN NAME        TYPE(LEN)  KEY            COLUMN VALUE
--- ------------------ ---------- ---- ----+---10----+---20----+---30----+---40
001 ORDER_NUMBER       CHAR(6)    PRIM AA2222
002 CUST_NUM           CHAR(6)         CN0001
```

Figure B-2. Action Bar Example - Table Mode

```
  File  Edit  View  Options  Open  Menu  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2            Edit   FR.ORDER_TABLE                  SSID: DSN
COMMAND ===>                                                   SCROLL ===> CSR
         ORDER_NUMBER  CUST_NUM  ORD_TYPE  ORD_STAT  FIRST_ACT_DATE  LAST_ACT_
         CHAR(6)       CHAR(6)   CHAR(2)   CHAR(2)   CHAR(10)        CHAR(10)
         PRIMARY-----  --------  --------  --------  --------------  ----------
 000001  AA2222        CN0001    WS        01        19880830        19880914
 000002  AA4444        CN0033    HT        03        19890810        19880914
```

File Pull-Down

Figure B-3 shows the File pull-down menu.

Figure B-3. File Pull-down

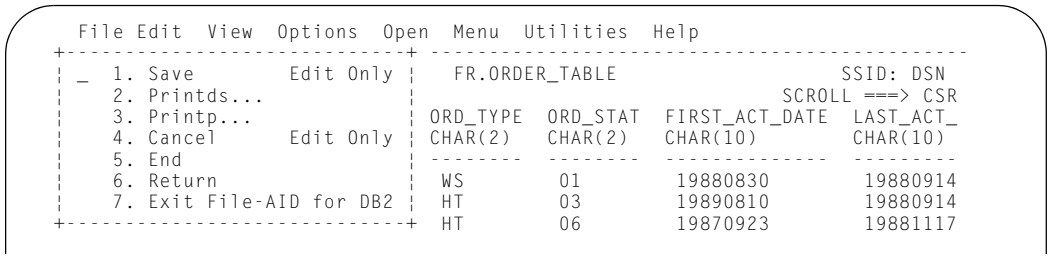


Table B-2 lists the File pull-down choices with a short description.

Table B-2. File Pull-Down

Menu Choices	Description
1. Save -- Edit Only	SAVE primary command. Not available in Browse mode. See "SAVE" on page 3-52 for a detailed description.
2. Printds...	PRINTDS primary command. Prints table data to a dataset. See "PRINTDS (PRTDS)" on page 3-38 for a detailed description.
3. Printp...	PRINTP primary command. Prints table data to a printer. See "PRINTP (PRTP)" on page 3-38 for a detailed description.
4. Cancel -- Edit Only	CANCEL primary command. Not available in Browse mode. See "CANCEL (CAN)" on page 3-4 for a detailed description.
5. End	END primary command. See "END" on page 3-18 for a detailed description.
6. Return	RETURN primary command. See "RETURN" on page 3-50 for a detailed description.
7. Exit File-AID for DB2	Option **X** from the Primary Option Menu. It terminates your current File-AID *for DB2* session.

Edit Pull-Down

Figure B-4 shows the Edit pull-down menu.

Figure B-4. Edit Pull-down in Table Mode

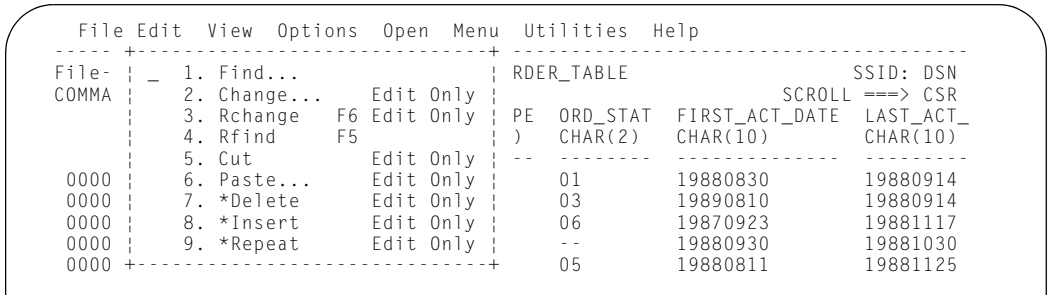


Table B-3 lists the Edit pull-down choices with a short description.

Table B-3. Edit Pull-Down

Menu Choices	Description
1. Find...	FIND primary command. See "FIND (F)" on page 3-19 for a detailed description.

Table B-3. Edit Pull-Down

Menu Choices	Description
2. Change... Edit Only	CHANGE primary command. Not available in Browse mode. See "CHANGE (C)" on page 3-5 for a detailed description.
3. Rchange F6 Edit Only	RCHANGE primary command. Repeats the last CHANGE command. Not available in Browse mode. See "RCHANGE" on page 3-42 for a detailed description.
4. Rfind F5	RFIND primary command. Repeats the last FIND command. See "RFIND" on page 3-50 for a detailed description.
5. Cut Edit Only	CUT primary command. Not available in row mode. See "CUT (CU)" on page 3-16 for a detailed description.
6. Paste... Edit Only	PASTE primary command. Not available in row mode. See "PASTE (PAS)" on page 3-34 for a detailed description.
7. Delete Edit Only	DELETE primary command. Not available in table mode. See "DELETE (DEL)" on page 3-17 for a detailed description.
8. Insert Edit Only	INSERT primary command. Not available in table mode. See "INSERT (INS)" on page 3-30 for a detailed description.
9. Repeat Edit Only	REPEAT primary command. Not available in table mode. See "REPEAT (REP)" on page 3-43 for a detailed description.

View Pull-Down

Figure B-5 shows the Edit pull-down menu.

Figure B-5. View Pull-down in Row Mode

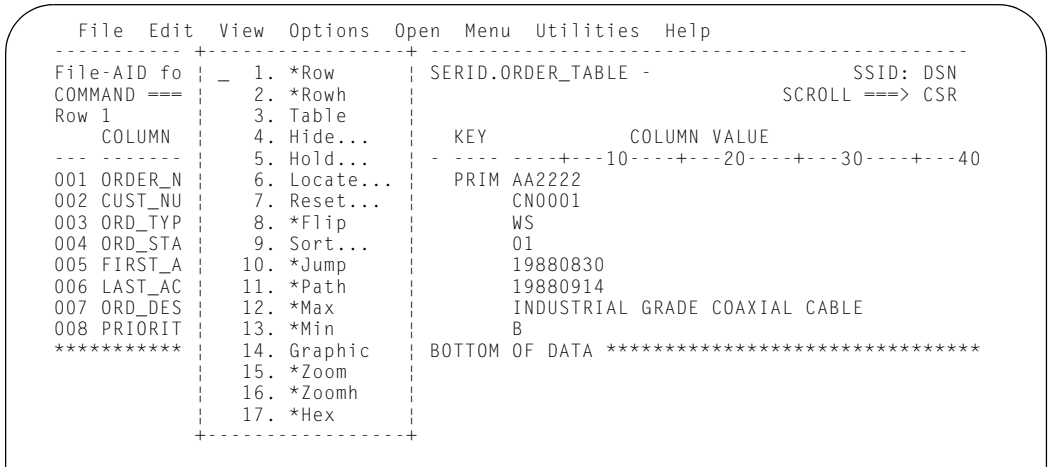


Table B-4 lists the View pull-down choices with a short description.

Table B-4. View Pull-Down

Menu Choices	Description
1. Row	ROW primary command. Switches display mode from table to row mode. See "ROW" on page 3-51 for a detailed description.
2. Rowh	ROWH primary command. Switches display mode from table to row mode in hexadecimal display. See "ROWH" on page 3-52 for a detailed description.
3. Table	TABLE primary command. Switches display mode from row to table mode. See "TABLE (TAB)" on page 3-56 for a detailed description.

Table B-4. View Pull-Down

Menu Choices	Description
4. Hide...	HIDE primary command. Displays the Hide/Hold/Reset Command window so you can manipulate the column display on your screen. See “HIDE” on page 3-25 for a detailed description.
5. Hold...	HOLD primary command. Displays the Hide/Hold/Reset Command window so you can manipulate the column display on your screen. See “HOLD” on page 3-27 for a detailed description.
6. Locate...	LOCATE primary command. See “LOCATE (L)” on page 3-32 for a detailed description.
7. Reset...	RESET primary command. Displays the Hide/Hold/Reset Command window so you can manipulate the column display on your screen. See “RESET (RES)” on page 3-48 for a detailed description.
8. Flip	FLIP primary command. Not available in row mode. Changes the status of all rows in respect to exclusion. See “FLIP” on page 3-24 for a detailed description.
9. Sort...	SORT primary command. Sorts table rows by columns. See “SORT” on page 3-52 for a detailed description.
10. Jump	JUMP primary command. Moves cursor from window to window in a multiple table edit or browse session. Not available in single table session. See “JUMP (J)” on page 3-30 for a detailed description.
11. Path	PATH primary command. Switches to the first window of the next related table path. Only available in related table sessions with more than 1 path open. See “PATH” on page 3-37 for a detailed description.
12. Max	MAX primary command. Enlarges a window to maximum screen size in a multiple table edit or browse session. Not available in single table session. See “MAX” on page 3-33 for a detailed description.
13. Min	MIN primary command. Reduces a window to minimum window size in a multiple table edit or browse session. Not available in single table session. See “MIN” on page 3-33 for a detailed description.
14. Graphic	GRAPHIC primary command. Displays a graph of the tables related to the current table in the edit or browse session. See “GRAPHIC (GR)” on page 3-25 for a detailed description.
15. Zoom	ZOOM primary command. Examines data from a specific column in character format. Not available in row mode. See “ZOOM” on page 3-60 for a detailed description.
16. Zoomh	ZOOMH primary command. Examines data from a specific column in hexadecimal format. Not available in row mode. See “ZOOMH” on page 3-60 for a detailed description.
17. Hex	HEX primary command. Displays all data in hexadecimal format. Not available in table mode. See “HEX” on page 3-25 for a detailed description.

Options Pull-Down

Figure B-6 shows the Options pull-down menu.

Figure B-6. Options Pull-Down

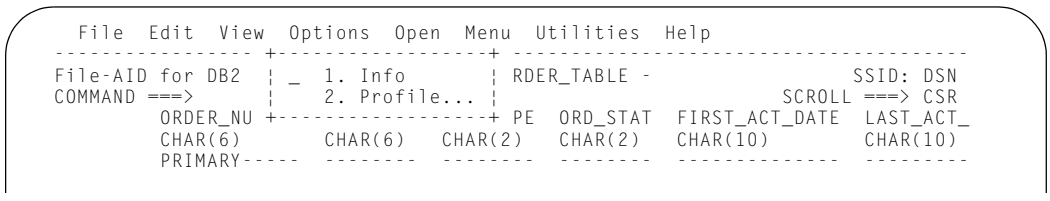


Table B-5 lists the View pull-down choices with a short description.

Table B-5. Options Pull-Down

Menu Choices	Description
1. Info	INFO primary command. Displays File-AID *for DB2* system information. See "INFO" on page 3-29 for a detailed description.
2. Profile...	PROFILE primary command. Displays your current user profile defaults. See "PROFILE (PRO)" on page 3-39 for a detailed description.

Open Pull-Down

Figure B-7 shows the Open pull-down menu.

Figure B-7. Open Pull-Down

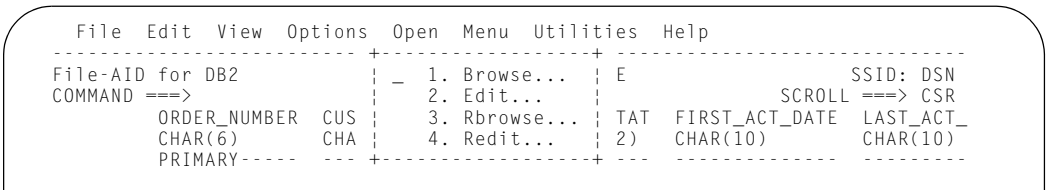


Table B-6 lists the View pull-down choices with a short description.

Table B-6. Open Pull-Down

Menu Choices	Description
1. Browse...	BROWSE primary command. Initiates a multiple browse session. See "BROWSE (BRO)" on page 3-3 for a detailed description.
2. Edit...	EDIT primary command. Initiates a multiple edit session. See "EDIT (ED)" on page 3-18 for a detailed description.
3. Rbrowse...	RBROWSE primary command. Initiates a related browse session. See "RBROWSE (RB)" on page 3-41 for a detailed description.
4. Redit...	REDIT primary command. Initiates a related edit session. See "REDIT (RE)" on page 3-42 for a detailed description.

Menu Pull-Down

Figure B-8 shows the Menu pull-down menu.

Note: When selecting an item in this menu, you **immediately terminate** the edit or browse session and switch to the selected function.

Figure B-8. Menu Pull-Down

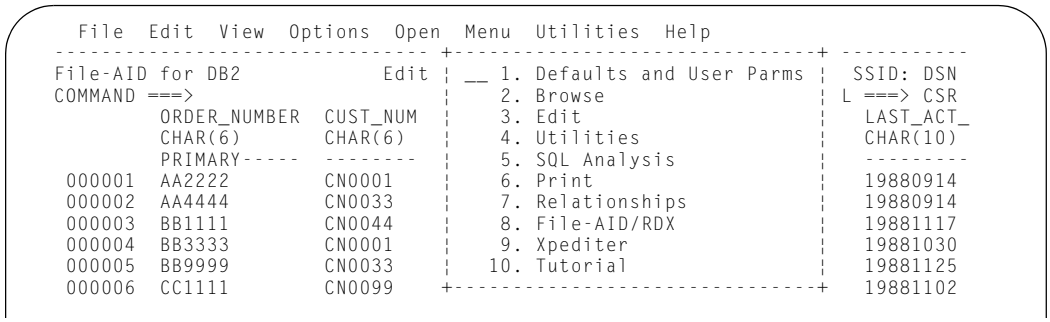


Table B-7 lists the pull-down choices with a short description. See also "Product Functions" on page 1-3.

Table B-7. Menu Pull-Down

Menu Choices	Description
1. Defaults and User Parms	Option 0 of the Primary Options Menu. Transfers to the "User Parms Menu" on page 5-1. See "User Parms" on page 5-1 for a detailed description.
2. Browse	Option 1 of the Primary Options Menu. Transfers to the Browse Options Menu to start a browse session. See "Starting a Browse or Edit Session" on page 6-4 for a detailed description.
3. Edit	Option 2 of the Primary Options Menu. Transfers to the Edit Options Menu to start an edit session. See "Starting a Browse or Edit Session" on page 6-4 for a detailed description.
4. Utilities	Option 3 of the Primary Options Menu. Transfers to the Utility Selection Menu. See "Utilities" on page 1-5 for a detailed description.
5. SQL Analysis	Option 4 of the Primary Options Menu. Transfers to the SQL Analysis and Development Menu. See "Accessing the SQL Analysis Function" on page 16-2 for a detailed description.
6. Print	Option 5 of the Primary Options Menu. Transfers to the Print Menu. See "Print Menu" on page 17-1 for a detailed description.
7. Relationships	Option 6 of the Primary Options Menu. Transfers to the Relationships Options Menu. See Chapter 18, "Relationship Facilitator" for a detailed description.
8. File-AID/RDX	Option R of the Primary Options Menu. Transfers to the "File-AID/Related Data XPERT" on page 1-6. See "File-AID/Related Data XPERT" on page 1-6 for a detailed description.
9. Xpediter	Option XP of the Primary Options Menu. Transfers to the "XPEDITER for DB2 Extension" on page 1-6. See "XPEDITER for DB2 Extension" on page 1-6 for a detailed description.
10. Tutorial	Option T of the Primary Options Menu. Transfers to the File-AID *for DB2* Tutorial System. See "Tutorials" on page 2-10 for a detailed description.

Utilities Pull-Down

Figure B-9 shows the Utilities pull-down menu.

Note: When selecting an item in this menu, you **immediately terminate** the edit or browse session and switch to the selected function.

Figure B-9. Utilities Pull-Down

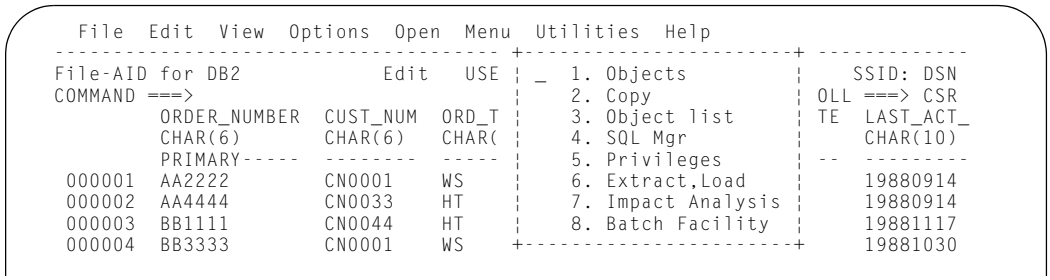


Table B-8 lists the pull-down choices with a short description. See also “Utilities” on page 1-5

Table B-8. Utilities Pull-Down

Menu Choices	Description
1. Objects	Option 2 of the Utility Selection Menu. Transfers to the “Objects - Create, Drop, Alter” on page 8-1 where you can create, alter or drop DB2 objects. See “Objects - Create, Drop, Alter” on page 8-1 for a detailed description.
2. Copy	Option 3 of the Utility Selection Menu. Transfers to the “Table Copy Selection” on page 9-1 where you can copy rows between DB2 tables. See Chapter 9, “Copy Utility” for a detailed description.
3. Object list	Option 4 of the Utility Selection Menu. Transfers to the “Access Object List Utility” on page 10-1 where you can list DB2 objects. See Chapter 10, “Object List Utility” for a detailed description.
4. SQL Mgr	Option 5 of the Utility Selection Menu. Enters the SQL Command Manager Utility. See Chapter 11, “SQL Command Manager Utility” for a detailed description.
5. Privileges	Option 6 of the Utility Selection Menu. Enters the Privileges Utility. See Chapter 12, “Privileges Utility” for a detailed description.
6. Extract,Load	Option 7 of the Utility Selection Menu. Enters the Extract and Load Utility. See Chapter 14, “Extract and Load Utility” for a detailed description.
7. Impact Analysis	Option 8 of the Utility Selection Menu. Enters the Impact Analysis Utility. See Chapter 13, “Impact Analysis Utility” for a detailed description.
8. Batch Facility	Option 9 of the Utility Selection Menu. Enters the Batch Facility. See Chapter 15, “Batch Facility” for a detailed description.

Help Pull-Down

Figure B-10 shows the Help pull-down menu.

Figure B-10. Help Pull-Down Menu

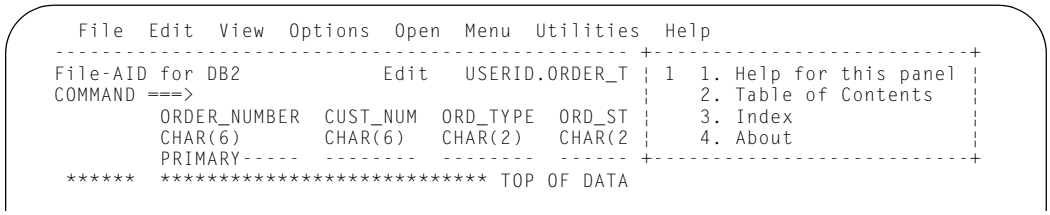


Table B-9 lists the pull-down choices for the Help action bar item with a short description.

Table B-9. Help Pull-Down

Menu Choices	Description
1. Help for this panel	Displays the tutorial for the current panel. Same as pressing HELP. See also "Tutorials" on page 2-10.
2. Table of Contents	Starts the File-AID *for DB2* tutorial with the Table of Contents. See also "Tutorials" on page 2-10.
3. Index	Starts the File-AID *for DB2* tutorial with the Tutorial Index. See also "Tutorials" on page 2-10.
4. About	Invokes the File-AID *for DB2* informational window.

Objects Utility Action Bar Items and Pull-Downs

These Action Bar items are available for the Objects Utility panels:

- Menu
- Utilities
- Help
- Define
- Actions

Figure B-11 shows the File-AID *for DB2* Edit action bar.

Figure B-11. Action Bar Example - Tables Option

Figure B-12 on page B-9 shows the File-AID *for DB2* Edit action bar.

Figure B-12. Action Bar Example - Tables Create Panel

```
  Menu  Define  Actions  Utilities  Help
--------------------------------------------------------------------------------
File-AID for DB2                  Table Create            Primary key copied
COMMAND ===>                                                SCROLL ===> PAGE
                                                                  SSID: DSN

Creator    => USERID1            Database   => DFHMSHDB  Editproc  =>
Table Name => NEWTAB             Tablespace => DFHMSHTS  Validproc =>
Restrict Drop => NO  (Yes or No)
```

The Menu, Utilities, and Help pull-downs are identical to the ones for the Edit and Browse panels, see "Menu Pull-Down" on page B-5, "Utilities Pull-Down" on page B-6, and "Help Pull-Down" on page B-8.

The Define and Actions pull-downs are very similar for all Objects functions. The listed items are always specific to the current panel and function, for example, the EXECUTE action in a create screen is called Execute Create, while in an alter screens it is called Execute Modify.

Define Pull-Down

Figure B-13 shows a typical Define pull-down menu.

Figure B-13. Define Pull-down Menu Example

```
Menu  Define  Actions  Utilities  Help
----- +---------------------------+-------------------------------------------
File- | 7  1. Primary Keys        |Table Create               Row 1 to 5 of 8
COMMA |    2. Foreign Keys        |                         SCROLL ===> CSR
      |    3. Default Values      |                             SSID: D610
Creat |    4. Column Constraints  | tabase   => BFHRLCDB  Editproc  =>
Table |    5. Named Constraints   | blespace => BFHRLCTS  Validproc =>
Restr |    6. Identity Column     |   CCSID =>          (EBCDIC or ASCII)
Data C|                           |   Audit =>          (NONE/CHANGES/ALL)
Tempor+---------------------------+   OBID  =>          (integer)
```

Table B-10 lists the pull-down choices for the Define action bar item with a short description.

Table B-10. Define Pull-Down

Menu Choices	Description
1. Primary Keys	Invokes the PRIMARY primary command and displays "Primary Key Definition" on page 8-9. See also "PRIMARY (PRIM)" on page 3-37 for a description of the PRIMARY command.
2. Foreign Keys	Invokes the FOREIGN primary command and displays "Foreign Key Definition" on page 8-10. See also "FOREIGN (FOR)" on page 3-24 for a description of the FOREIGN command.
3. Default Values	Invokes the DEFAULT primary command and displays "Default Values" on page 8-12. See also "DEFAULT" on page 3-17 for a description of the DEFAULT command.
4. Column Constraints	Invokes the COLCHECK primary command and displays "Column Check Constraints" on page 8-13. See also "COLCHECK" on page 3-16 for a description of the COLCHECK command.
5. Named Constraints	Invokes the NAMCHECK primary command and displays "Named Check Constraints" on page 8-14. See also "NAMCHECK" on page 3-33 for a description of the NAMCHECK command.

Table B-10. Define Pull-Down

Menu Choices	Description
6. Identity Column	Invokes the IDENTITY primary command and displays "Identity Column" on page 8-15. See also "IDENTITY" on page 3-28 for a description of the IDENTITY command.

Actions Pull-Down

Figure B-14 shows a typical Actions pull-down menu.

Figure B-14. Actions Pull-down Menu Example

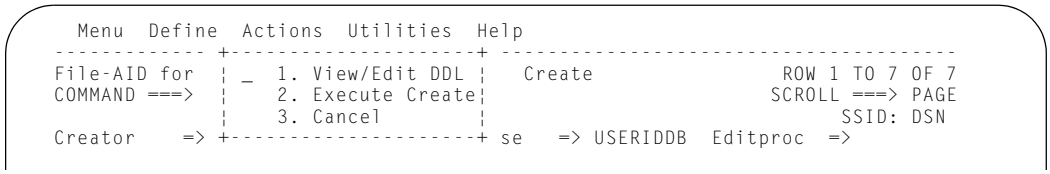


Table B-11 lists the pull-down choices for the Actions action bar item with a short description.

Table B-11. Actions Pull-Down

Menu Choices	Description
1. View/Edit DDL	Invokes the SQL primary command. See "SQL" on page 3-55 for a description of the SQL command.
2. Execute Create	Invokes the EXECUTE primary command. See "EXECUTE (EXEC)" on page 3-19 for a description of the EXECUTE command.
3. Cancel	Invokes the CANCEL primary command. See "CANCEL (CAN)" on page 3-4 for a description of the EXECUTE command.
4. Define Partitions	Invokes the PARTITN primary command and displays the "Partitioned Tablespace Create" on page 8-28. See "PARTITN (PART)" on page 3-34 for a description of the PARTITN command.
5. Define Index Values	Invokes the VALUES primary command and displays the "Index Create" on page 8-34. See "VALUES" on page 3-59 for a description of the VALUES command.

Index

A

A (Add) line command
  Application Relationship Detail screen, 18-16, 18-32
  Relationship Facilitator, 18-6, 18-13
A (After) line command, 4-2
access
  browse and edit, 6-4
  copy rows between tables utility, 9-1
  create, drop, alter DB2 objects utility, 8-1
  display, grant, revoke, and modify privileges utility, 12-1
  extract and load tables utility, 14-2
  File-AID *for DB2*, 2-1
  from source code edit session, 2-4
  IBM SPUFI, 11-1
  object list utility, 10-1
  path analysis, 16-20
  SQL analysis, 16-2
ACQUIRE, 10-15
action
  bar, B-1
  pull-down, B-10
active table, 7-9
add application relationship, 7-19
add AR, 18-11
  Relationship Facilitator, 18-4
add AR menu option
  Relationship Facilitator, 18-11
ADD command, 3-3
ADD primary command, 7-19
  Relationship Detail screen, 18-16, 18-33
adding relationships
  DB2 to DB2, 18-35
  DB2 to IMS, 18-40
  DB2 to MVS, 18-37
  MVS to DB2, 18-42
  MVS to IMS, 18-47
  MVS to MVS, 18-44
aging example, 3-14
alias, 8-47
  create, 8-49
  drop, 8-50
  information, 10-18
  modify, 8-48
allocate
  discard dataset, 14-24
  SHOW SQL dataset, 5-5
alter
  DB2 object, 8-1
  index, 8-39
  object, 10-3
  stored procedure, 8-83
  table, 8-20
  tablespace, 8-29
ALTER (ALT) command, 3-3
ALTEREDTS, 10-9, 10-29, 10-35
alternate catalog plan, 2-2
ANY, 3-6, 3-20
Application Relationship (AR), 18-31–18-58
  column relationships
    example, 18-56
  creating column relationships, 18-50
  deleting, 18-31
  deleting column relationships, 18-50
  line commands
    A (Add), 18-16, 18-32
    column-number, 18-54
    D (Delete), 18-16, 18-32, 18-55
    I (Information), 18-32
    S (Select), 18-16, 18-32
  messages, status, 18-32
  modifying, 18-31, 18-50
  modifying column relationships, 18-50
  overview, 18-31
  primary commands
    CANCEL, 18-55
    GRAPHIC, 18-33
    SUGGEST, 18-55
  viewing AR information, 18-31, 18-33
application relationship (AR)
  add, 7-19
  create column relationships, 7-20
  delete, 7-19
  overview, 7-2
  using in related table edit, 7-5
application relationships, 18-2
AR relationships, 18-2
assembler C, 8-75
associated RI relationships
  screen label
    RI Relationship Detail screen, 18-26
ASUTIME, 8-76, 10-27, 10-35
ATOMIC command, 3-3
audit trail
  audit trail print facility, 6-32
  create an audit trail dataset, 6-31
  print, 17-1
  print report, 17-13
  to SMF log, 6-32
AUDITING, 10-9
auto create, 9-4
autocast, 3-58, 5-4, 8-14
autocommit, 5-3
AUTOSAVE
  command, 7-40
  profile option, 3-40
autosave, 5-3
auxiliary tables, information, 10-22
AVGROWLEN, 10-8
AVGSIZE, 10-15, 10-19

B

B (Before) line command, 4-2
B (Browse) line command
  Relationship Facilitator, 18-7, 18-13
batch
  load table, 14-29

options, 5-6
output specification, 13-5
output specifications, 16-36
batch execute SQL, 15-2
specify PRINT options, 15-4
specify processing, 15-3
specify SQL dataset, 15-3
batch execute with Auto-COMMIT processing utility, 1-5
Batch Facility, 15-1
accessing, 15-2
editing SQL, 15-4
JCL, 15-5
editing, 15-7
output specification, 15-4
begin atomic...end, 8-90
BINDDATE, 10-14
BINDTIME, 10-15, 10-19
BOTTOM (BOT) command, 3-3
BOUND BY, 10-15
BOUNDTS, 10-15
BPOOL, 10-10, 10-13
BR line command, 4-2, 16-47, 16-53
browse, 1-4
access, 6-4
and edit, 6-1–6-4
object, 10-3
options, 6-4
Relationship Detail screen, 18-15
results table, 16-53
row mode, 6-17
selection criteria member list, 6-8
selection template, 6-9
SQL statements, 16-48
table mode, 6-14
table selection, 6-9
BROWSE (BRO) command, 3-3
browse menu option
Relationship Facilitator, 18-3
browse relationship file, 18-3
Browse Relationships
browsing information about a relationship, 18-20
primary commands
ADD, 18-33
browse/edit/SQL analysis, 5-2
BUFFERPOOL, 5-8–5-9
built-in data types, 8-80
built-in datatypes, 8-78

C

C line command, 4-3
CANCEL (CAN) command, 3-4, 7-40
CANCEL primary command
Column Relationships screen, 18-55
CAPS command, 3-4
CARD, 10-8
CARDF, 10-10
cardinality, 10-35
cast function, 10-32
catalog information, 16-28
CCSID, 8-67, 8-78
CHANGE (C) command, 3-5
ANY, 3-6
Date, 3-15
Time, 3-14
Timestamp, 3-11
with increment, 3-10
changing SQL statements, 16-54
CHARSET, 10-17, 10-22
CHECK, 10-14
check constraint, 16-34
error, 6-21
CHECKFLAG, 10-9
CHECKRID, 10-9
CHECKS, 10-10
CHILDREN, 10-9
CLASS, 10-36
CLOSE, 5-8–5-9
CLOSERULE, 10-11, 10-13
CLUSTERED, 10-12
CLUSTERING, 10-12
CLUSTERRATIO, 10-12
CMD, 10-2
COBOL, 8-75
COLCHECK command, 3-16
COLCOUNT, 10-7, 10-12, 10-14
collection
display privileges, 12-3
grant privileges, 12-4
ID, 10-26
modify privileges, 12-5
privileges, 12-3
revoke privileges, 12-5
collection ID
selection, 16-14
COLLID, 10-19
column
alignment, 6-16
allow update of identity column, 5-3
attributes, 8-8
check constraints, 8-13
check constraints, when creating a table, 8-13
create relationships, 7-20
identity, 3-28, 5-3, 8-15
impact analysis, 13-1
impact analysis report, 13-5
impact analysis utility, 1-5
information, 16-29
obscured, 6-2
partition statistics, 16-31
snap, 6-2
update identity column, 3-41
column relationships
creating, 18-50
example of building, 18-56
column-number line command
Column Relationships screen, 18-54
COMMA, 10-17, 10-21
commands, line, 4-1
comment on, 8-67, 8-82, 10-24, 10-26, 10-31, 16-35
COMMIT on return, 10-35
commit on return, 8-76, 10-27
comparing SQL results, 16-22
COMPJAVA, 10-2
concurrent update intercept, 6-20
confirm drop, 8-22
constraints
check, 16-34
check constraint error, 6-21
column check, 8-13
named check, 8-14
CONTOKEN, 10-19
control card dataset allocation, 14-15

convert data to uppercase, 5-3
COPY, 5-8, 10-14
copy
  column selection, 9-6
  row selection, 9-7
  rows between tables, 1-5
  rows between tables utility, 9-1–9-9
  table, 9-1
cost detail, 16-35
create
  alias, 8-49
  application relationship, 7-19
  auto create a table from copy, 9-2
  DB2 objects, 8-1
  distinct type, 8-51
  impact analysis, 13-3
  indexes, 8-34
  partitioned tablespace, 8-28
  stored procedure, 8-75
  synonym, 8-46
  table, 8-3
  tablespaces, 8-25
  view, 8-44
CREATE (CRE) command, 3-16
CREATE TRIGGER, 15-1
CREATED BY, 10-7, 10-11–10-12, 10-14, 10-18
CREATEDTS, 10-9, 10-18, 10-24, 10-29, 10-35
CREATOR, 10-2, 10-10, 10-14, 10-17–10-19
creator
  of table for relationship display, 18-4
  of table for relationship display, specifying, 18-37, 18-40, 18-43
current date and time, 2-8
current path, 16-39
CUT (CU) command, 3-16

D

D (Delete) line command
  Application Relationship Detail screen, 18-16, 18-32
  Column Relationships screen, 18-55
  RI Relationship Detail screen, 18-27
D (delete) line command, 4-3
DASDSPACE, 10-12
data, convert to uppercase, 5-3
DATACAPT, 10-9
datatypes
  built-in, 8-78, 8-80
  CHAR, 6-16
  DATE, 6-17
  DEC, 6-16
  defined, 6-16
  DISTINCT, 6-17
  FLOAT, 6-17
  GRAPH, 6-16
  INTEGER, 6-16
  LVARG, 6-16
  LVCHAR, 6-16
  ROWID, 6-17
  SMALLINT, 6-16
  TIME, 6-17
  TIMESTAMP, 6-17
  VARGRAPHIC, 6-16
  VCHAR, 6-16
date and time
  arithmetic example, 3-14
  current, 2-8
DB2, 7-3, 18-1
  adding application relationships, 18-35
  browsing results table, 16-53
  cursor, 6-3
  display, grant, revoke, and modify privileges, 1-5
  edit results table, 16-54
  results table, 16-53–16-54
  supported versions, -xxvi
  updating methods, 6-3
  using catalog to determine relationships, 7-4
DB2 catalog
  inconsistent with relationship file, 18-29
  loading RI relationships from, 18-27
DB2 object
  relationship facilitator, specifying a table, 18-3
DB2 primary command
  MVS Relationship Facilitator screen, 18-12
DB2 Version 8
  long names, 2-4
  SQL, 16-1
DB2 Version 8 long names
  on reports, 2-5
DB2SQL, 8-77, 10-28
DBA-XPERT column impact analysis utility, 1-5
DBCS and uppercase-only support, 1-3
DBID, 10-8, 10-10, 10-13, 10-24
DBINFO, 8-76, 10-27, 10-34
DBNAME, 10-7, 10-10, 10-13–10-14
DBPROTOCOL, 10-21
DBRM
  selection, 16-13, 16-16
DBRM NAME, 10-16
ddl
  generate statements, 10-4
  view and edit, 8-68, 8-82
DEC31, 10-17, 10-22
DECLARE CURSOR, 16-51, 16-54
DEFAULT, 8-67, 8-74, 8-78, 10-28, 10-34
DEFAULT command, 3-17
default values, defining, 8-12
default values, index, tablespace, and database, 5-7
DEFERPREPARE, 10-20
define pull-down, B-9
DEFINER, 8-65, 8-73, 8-78, 10-29, 10-34
DEGREE, 10-19
DELETE (DEL) command, 3-17
delete rows from explain table, 16-4
delimited format options, 14-12
detail, explain, 16-23
determine relationship, 7-4
deterministic, 8-76, 10-27, 10-33
discard
  allocate a discard dataset, 14-24
  rows, 14-23, 14-27, 14-30
display
  collection privileges, 12-3
  distinct type privileges, 12-23
  package privileges, 12-6
  plan privileges, 12-8
  schema privileges, 12-19
  stored procedure privileges, 12-21
  table/view privileges, 12-10
  UDF privileges, 12-25
display privileges, 12-1
distinct type, 3-57, 5-4, 8-6
  autocast, 3-58

create, 8-51
display privileges, 12-23
drop, 8-52, 8-74
grant privileges, 12-24
information, 10-25
modify privileges, 12-25
privileges, 12-23
revoke privileges, 12-25
selection, 8-9
DOWN command, 3-17
drop
alias, 8-50
confirm, 8-22
DB2 object, 8-1
distinct type, 8-52, 8-74
index, 8-43
object, 10-3
stored procedure, 8-84
synonym, 8-47
table, 8-22
tablespace, 8-33
triggers, 8-92
view, 8-45
DSETPASS, 5-8–5-9, 10-11
DSETPASSW, 10-13
dynamic result sets, 10-27
DYNAMIC RULES, 10-15, 10-21

E

E (Edit AR) line command
Relationship Facilitator, 18-7, 18-13
ED line command, 4-3, 16-47, 16-54
edit, 1-4, 6-1–6-4
access, 6-4
action bar, B-1
columns with defined default values, 6-4
considerations, 6-1
DB2 results table, 16-54
ddl, 8-68, 8-82
end confirmation, 6-19, 7-40
object, 10-3
options, 6-4, 6-9
pull-downs, B-1–B-2
row mode, 6-17
SELECT statement, 14-10
selection criteria member list, 6-8
selection template, 6-9
SPUFI selection, 6-13, 9-9, 17-8
SQL, 14-10
SQL statements, 16-48
table mode, 6-14
table selection, 6-9
views, 6-4
EDIT (ED) command, 3-18
edit AR, 18-3, 18-11
editing SQL
Batch Facility, 15-4
EDPROC, 10-9
ENCSCHEME, 10-8
END command, 3-18, 7-40
end confirmation window, 6-19
enhancing SQL efficiency, 1-3
ERASE, 5-7–5-8
ERASERULE, 10-10, 10-13
EX line command, 4-4, 16-47, 16-51
EXCLUDE (X) command, 3-18
EXECUTE (EXEC) command, 3-19
execute SQL, 16-50, 16-52–16-53
exit, File-AID *for DB2*, 2-1
Exiting, 2-1
EXPLAIN, 10-15, 10-20
explain
delete rows from explain table, 16-4
detail, 16-23
report, 16-41
report specification, 16-41
SQL statements, 16-12, 16-14–16-16, 16-48, 16-56
external action, 10-33
external name, 10-26
extract
and load, 14-31
and load tables utility, 14-1
confirmation, 14-17
data online, 14-10
data through batch JCL, 14-18
dataset allocation, 14-13
from table, 10-5
load tables utility, 1-5
select rows to extract, 14-6
select table to extract, 14-2
selection criteria member list, 14-5
through SPUFI, 14-9
extract and load
tables utility, 14-30
extract primary commands
SQLID, 18-17, 18-28

F

F2PTROW, 16-42
FADB2, 2-4
initiating File-AID *for DB2* from source edit session, 16-47
FADB2 command, 3-19
fenced, 10-27, 10-35
FETCH FIRST n ROWS, 6-3
FETCH SQL statement, 16-52
FETCH statement, 16-52
file pull-down, B-1
File-AID/Related Data XPERT, 1-6
final call, 10-33
FIND (F) command, 3-19
FIRSTKEYCARDF, 10-13
FLIP command, 3-24
FOREIGN (FOR) command, 3-24
FOREIGN command, 6-16
foreign keys, 3-41, 5-2
defining, 8-10
FREEPAGE, 5-8
FRGN, 6-18
FRSTKYCRD, 10-13
FULLKEYCARDF, 10-13
FULLKYCRD, 10-13
function
detail, 16-35
FUNCTIONTS, 10-19

G

GBPCACHE, 8-26, 8-29, 8-31, 8-33, 8-37, 8-39–8-40, 8-43
GENERAL, 8-77, 10-28
GENERAL WITH NULLS, 8-77
generate DDL statements, 10-4
grant
  collection privileges, 12-4
  distinct type privileges, 12-24
  package privileges, 12-6
  plan privileges, 12-8
  privileges, 12-1
  privileges on object, 10-4
  schema privileges, 12-20
  stored procedure privileges, 12-22
  table/view privileges, 12-12
  tablespace privileges, 12-17
  UDF privileges, 12-26
GRANT command, 3-24
Granularity, 10-24
graph relationship, 3-25, 7-42
GRAPHIC (GR) command, 3-25
GRAPHIC primary command
  Application Relationship Detail screen, 18-33
  Relationship Detail screen, 18-17
  RI Relationship Detail screen, 18-28
Graphical Relationship Display, 18-18
  Option List, 18-19
GROUP MEMBER, 10-15, 10-20

H

help, 2-10
HELP command, 3-25
help pull-down, B-8
HEX command, 3-25
HIDE command, 3-25
HOLD command, 3-27
host variable resolution, 16-48
  entering values for, 16-47, 16-55
  resolving values for, 16-47
HOSTLANG, 10-17, 10-20, 10-22

I

I (Information) line command
  Application Relationship Detail screen, 18-32
  Graphical Relationship Display screen, 18-20
  Relationship Detail screen, 18-16
  RI Relationship Detail screen, 18-27
I (Insert) line command, 4-4
IBM
  required release, -xxvi
IBMREQD, 10-15, 10-17, 10-33
IDENTITY, 6-16
identity column, 3-28, 5-3, 6-16, 6-18, 8-15, 8-17
  allow update, 5-3
  update, 3-41
IDFN, 6-19
IDNT, 6-18
IMMEDWRITE, 10-21
impact
  analysis, create, 13-3
  analysis, table, 13-3
  selection template, 13-4
IMPLICIT, 10-11
implicit tablespace confirmation, 8-17
IMS Root Segment
  column mapping to, 18-54
  example of building column/segment relationships, 18-61
IN, 8-80, 10-30
increment example, 3-14
Ind D/T, 8-80
index, 10-11
  alter, 8-39
  alter for nnpartitioned, 8-39
  alter for partitioned, 8-41
  confirm drop, 8-43
  create, 8-34
  create partition, 8-38
  create values, 8-36
  drop, 8-43
  information, 10-11, 16-32
  key information, 16-32
  options, 8-33
  partition information, 16-33
  selection, 16-31
index, tablespace, and database default values, 5-7
INDEXSPACE, 10-12
INDEXTYPE, 10-13
INFO command, 3-29
information display, 10-6, 10-22, 10-25, 10-31
INHERIT, 8-66, 8-74, 8-78, 10-28, 10-34
Initial Insts, 10-36
Initial IOS, 10-35
INSERT (INS) command, 3-30
Insts per Invoc, 10-35
invoke
  File-AID *for DB2*, 2-1
  from source code edit session, 2-4
IOS per Invoc, 10-35
ISOBID, 10-13
ISOLATION, 10-15, 10-20
ISPF
  invoking File-AID *for DB2*, 2-4
  keylists, 2-5
  log ISPF calls, 5-9
  required release, -xxvi
  source code edit session, 2-4

J

JAR ID, 10-36
JAR Schema, 10-36
JAVA, 10-2
JCL
  Batch Facility, 15-5
    editing, 15-7
    parameters, 15-7
JUMP (J) command, 3-30
JUMP command, 6-25, 7-31

K

KEEPDYN, 10-15
KEEPDYNAMIC, 10-21
key, index information, 16-32
KEYCOLS, 10-8
KEYLIST command, 3-31
KEYOBID, 10-8
KEYS command, 3-31

L

L (Load RI) line command
  Relationship Facilitator, 18-7
LABEL, 10-8, 10-18
language, 8-84
large object type (LOB), 7-1, 8-6, 14-1
LEFT command, 3-31
line commands, 4-1
  A (Add AR), 18-6
  A (Add), 18-13, 18-16, 18-32
  A (After), 4-2
  B (Before), 4-2
  B (Browse), 18-7, 18-13
  BR (BRowse), 16-47, 16-53
  BR (Browse), 4-2
  C (Copy), 4-3
  column-number, 18-54
  D (Delete), 18-16, 18-27, 18-32, 18-55
  D (delete), 4-3
  E (Edit AR), 18-7, 18-13
  ED (EDit), 16-47, 16-54
  ED (Edit), 4-3
  EX (EXecute), 16-47, 16-51
  EX (Execute), 4-4
  I (Information), 18-16, 18-27, 18-32
  I (Insert), 4-4
  L (Load RI), 18-7
  M (Move), 4-4
  R (Repeat), 4-4
  Relationship Facilitator, 18-6, 18-13
  RI (Referential Integrity), 4-5
  S (Select), 4-5, 18-16, 18-27, 18-32
  S (Show), 4-5
  where valid, 4-1
  X (Exclude), 4-6
  XP (Explain), 4-6
  XP (explain), 16-47, 16-56
list DB2 catalog, 18-4
  Relationship Facilitator, 18-4
list RDX
  Relationship Facilitator, 18-4, 18-11
load
  and extract, 14-31
  confirmation, 14-25
  data online, 14-19, 14-21
  data through batch JCL, 14-29
  data using SQL statements, 14-20
  dataset specification, 14-20
  options, 14-19
  table, 10-5
LOB, 7-1, 8-6, 14-1
LOBcolumns, 10-33
LOCATE (L) command, 3-32
  in an object list, 3-33
  in browse or edit, 3-32
LOCATION, 10-10, 10-18
location selection, 6-7
locked table, 6-3
LOCKMAX, 8-27, 8-30, 8-32, 10-11
LOCKRULE, 10-10
LOCKSIZE, 5-8
log
  ISPF calls, 5-9
  options, 5-9
  SQL information, 5-9
long name support, 7-3, 18-1
long names
  on reports, 2-5
  viewing, 2-4

M

M (Move) line command, 4-4
MAIN, 8-65, 8-73, 8-76, 10-27, 10-32
map copied columns, 9-6
MAX command, 3-33, 6-26, 7-33
maximum rows to select, 6-3
menu pull-down, B-5
MESSAGE, 10-2
messages, predicate analysis, A-1
MIN command, 3-33, 6-26, 7-33
MIXED, 10-17, 10-22
model
  relationship, 7-16
  table area, 8-8
  table selection, 8-7
modify
  collection privileges, 12-5
  distinct type privileges, 12-25
  package privileges, 12-7
  plan privileges, 12-10
  privileges, 12-1
  schema privileges, 12-21
  stored procedure privileges, 12-23
  table/view privileges, 12-16
  tablespace privileges, 12-18
  UDF privileges, 12-27
MODIFY (MOD) command, 3-33
multiple tables
  editing and browsing related tables, 7-1
  editing and browsing unrelated tables, 6-22
  using row mode, 6-26
  using the AUTOSAVE command, 3-18, 3-40, 7-40
  using the BROWSE (BRO) command, 3-3
  using the CANCEL (CAN) command, 3-4
  using the CANCEL command, 7-40
  using the EDIT (ED) command, 3-18
  using the JUMP (J) command, 3-30
  using the JUMP command, 6-25, 7-31
  using the MAX command, 3-33, 6-26, 7-33
  using the MIN command, 3-33, 6-26, 7-33
  using the PATH command, 7-38
  using the RBROWSE (RB) command, 3-41
  using the REDIT (RE) command, 3-42
  working with related tables, 7-34
MVS file name
  for relationship display, specifying, 18-38, 18-41–18-42, 18-45, 18-47–18-48
  for Relationship Facilitator, specifying, 18-12

MVS object
considerations for application relationships, 18-2
MVS primary command
Relationship Facilitator, 18-5

N

NACTIVE, 10-11
NAMCHECK command, 3-33
NAME, 10-2, 10-10, 10-14, 10-17–10-18
named check constraints, 8-14
when creating a table, 8-14
NEXT command, 3-34
NLEAF, 10-13
NLEVELS, 10-13
NPAGES, 10-8
NPAGESF, 10-10
NTABLES, 10-11
null
call, 10-26, 10-33
handling, 5-4

O

OBID, 10-8, 10-10, 10-13, 10-24
object
creator, 10-1
information, 10-6
name, 10-1
names, convert, 5-3
selection menu, 8-1
type, 10-1
types, privileges, 12-2
utility, 8-1
utility, action bar items, B-8
utility, pull-downs, B-8
Object List in Relationship File
line commands
S (Select), 18-14
selecting an object, 18-14
object list utility, 1-5, 10-1–10-21
(A)lter, 10-3
(B)rowse, 10-3
(D)rop, 10-3
(E)dit, 10-3
(G)rant privileges, 10-4
(GD)Generate DDL, 10-4
(I)nformation, 10-6
(I)nformation for alias, 10-18
(I)nformation for auxiliary tables, 10-22
(I)nformation for indexes, 10-11
(I)nformation for packages, 10-18
(I)nformation for plans, 10-14
(I)nformation for stored procedures, 10-25
(I)nformation for synonyms, 10-17
(I)nformation for tables, 10-6
(I)nformation for tablespaces, 10-10
(I)nformation for triggers, 10-22
(I)nformation for user defined function (UDF), 10-31
(I)nformation for views, 10-14
(L)oad, 10-5
(P)rivilege (M)odify, 10-6
(P)rivileges, display, 10-5
(R)evoke, 10-4
e(X)tract, 10-5
selection, 10-2
Object type
DB2 table, 18-15
IMS database, 18-15
MVS file, 18-15
specify for adding relationships, 18-34
obscured columns, 6-2
one-to-many table relationship, 7-12
one-to-one table relationship, 7-12
OPEN CURSOR SQL statement, 16-51
open pull-down, B-5
OPERATIVE, 10-15, 10-21
OPTHINT, 10-19
Option 3.9, 15-2
options pull-down, B-4
origin, 10-26, 10-32
OUT, 8-80
OWNER, 10-19
owner, 10-26, 10-31

P

package
display privileges, 12-6
grant privileges, 12-6
information, 10-18
modify privileges, 12-7
privileges, 12-6
revoke privileges, 12-7
selection, 16-15–16-16
PAGESIZE, 10-13
parallel, 10-35
parameter
for stored procedure, 8-78
information, 10-26, 10-29, 10-31, 10-36
parameters
JCL
Batch Facility, 15-7
parent relationship, 7-14
PARENTS, 10-9
parm count, 10-32
parm style, 10-28, 10-32
PARMS command, 3-34
partial column names, 2-9
partition
information, 16-33
statistics, 16-31, 16-34
PARTITIONS, 10-10
PARTITN (PART) command, 3-34
PASTE (PAS) command, 3-34
paste to dataset screen, 3-35
path, 7-36
access analysis, 16-20
current, 5-11
opening related, 7-36
switching to other, 7-38
PATH command, 3-37, 7-38
PCTFREE, 5-8
PCTIMESTAMP, 10-19
PCTPAGES, 10-8
PCTRWCMP, 10-9
PDSNAME, 10-16, 10-19
PEND, 6-2
pending state, 6-2

performance, show foreign keys, 3-41, 5-2
PGSIZE, 10-10
PIECESIZE, 8-19, 10-13
PKSIZE, 10-19
plan
  analysis specification, 16-8
  display privileges, 12-8
  grant privileges, 12-8
  information, 10-14
  modify privileges, 12-10
  privileges, 12-8
  revoke privileges, 12-9
  selection, 16-11
PLCREATOR, 10-16
PLI, 8-75
PLNAME, 10-16
PLSIZE, 10-15
PRECOMPDATE, 10-16
PRECOMPTIME, 10-16
PRECOMPTS, 10-16
predicate analysis, 16-25
  messages, A-1
  rules, A-1
PREV (P) command, 3-37
PRFI, 6-19
PRID, 6-19
PRIM, 6-18
PRIMARY (PRIM) command, 3-37
PRIMARY command, 6-15
primary commands, 2-7, 3-1
  ADD, 3-3, 18-4, 18-16, 18-33
  ALTER (ALT), 3-3
  ATOMIC, 3-3
  BOTTOM (BOT), 3-3
  BROWSE (BRO), 3-3
  CANCEL, 18-55
  CANCEL (CAN), 3-4
  CAPS, 3-4
  CHANGE (C), 3-5
  COLCHECK, 3-16
  CREATE (CRE), 3-16
  CUT (CU), 3-16
  DB2, 18-12
  DEFAULT, 3-17
  DELETE (DEL), 3-17
  DOWN, 3-17
  EDIT (ED), 3-18
  END, 3-18
  EXCLUDE (X), 3-18
  EXECUTE (EXEC), 3-19
  FADB2, 3-19
  FIND (F), 3-19
  FLIP, 3-24
  FOREIGN (FOR), 3-24
  GRANT, 3-24
  GRAPHIC, 18-17, 18-28, 18-33
  GRAPHIC (GR), 3-25
  HELP, 3-25
  HEX, 3-25
  HIDE, 3-25
  HOLD, 3-27
  IDENTITY, 3-28
  INFO, 3-29
  INSERT (INS), 3-30
  JUMP (J), 3-30
  KEYLIST, 3-31
  KEYS, 3-31
  LEFT, 3-31
  LOCATE (L), 3-32
  MAX, 3-33
  MIN, 3-33
  MODIFY (MOD), 3-33
  MVS, 18-5
  NAMCHECK, 3-33
  NEXT, 3-34
  PARTITN (PART), 3-34
  PASTE (PAS), 3-34
  PATH, 3-37
  PREV (P), 3-37
  PRIMARY (PRIM), 3-37
  PRINTDS (PRTDS), 3-38
  PRINTP (PRTP), 3-38
  PROFILE (PRO), 3-39
  RBROWSE (RB), 3-41
  RCHANGE, 3-42
  REDIT (RE), 3-42
  REPEAT (REP), 3-43
  RESET (RES), 3-48
  RETPARM, 3-50
  RETRIEVE, 3-50
  RETURN, 3-50
  REVOKE (REV), 3-50
  RFIND, 3-50
  RI, 3-50
  RIGHT, 3-51
  ROW, 3-51
  ROWH, 3-52
  RUNOPTS, 3-52
  SAVE, 3-52
  SORT, 3-52
  SPLIT, 3-55
  SQL, 3-55
  SQLID, 3-55, 18-17, 18-28
  SRCPARM, 3-56
  SUGGEST, 18-55
  SWAP, 3-56
  TABLE, 3-56
  TOP, 3-57
  UDFOPTS, 3-57
  UDT, 3-57
  UP, 3-59
  VALUES, 3-59
  WHENCOND, 3-59
  WRAP, 3-60
  ZOOM, 3-60
  ZOOMH, 3-60
primary key, 5-2, 8-9
primary options
  Relationship Facilitator, 18-1–18-73
    Application Relationship, ??–18-58
    Browse Relationships, 18-15
    Print Relationships, 18-68–18-73
    RI Relationship, ??–18-30
  relationship facilitator, 18-1
print
  audit trail, 17-1
  menu, 17-1
  options, 5-10, 17-1
  selection template, 17-5
  table data, 17-1
  table output specification, 17-9
  table selection, 17-4
print relationship file, 18-4, 18-11
Print Relationships, 18-68–18-73
  modifying print JCL, 18-70
  sample formatted relationship report, 18-71

specifying print options, 18-69
print relationships
specifying a relationship file, 18-68
PRINTDS (PRTDS) command, 3-38
PRINTP (PRTP) command, 3-38
PRIQTY, 5-7–5-8
privileges, 10-5
collection, 12-3
display collection, 12-3
display distinct type, 12-23
display package, 12-6
display plan, 12-8
display schema, 12-19
display stored procedure, 12-21
display table, 12-10
display UDF, 12-25
display view, 12-10
display, grant, revoke, and modify DB2, 12-1
distinct type, 12-23
general, 12-1
grant collection, 12-4
grant distinct type, 12-24
grant on object, 10-4
grant package, 12-6
grant plan, 12-8
grant schema, 12-20
grant stored procedure, 12-22
grant table, 12-12
grant tablespace, 12-17
grant UDF, 12-26
grant view, 12-12
modify, 10-6
modify collection, 12-5
modify distinct type, 12-25
modify package, 12-7
modify plan, 12-10
modify schema, 12-21
modify stored procedure, 12-23
modify table, 12-16
modify tablespace, 12-18
modify UDF, 12-27
modify view, 12-16
object types, 12-2
package, 12-6
plan, 12-8
revoke collection, 12-5
revoke distinct type, 12-25
revoke on object, 10-4
revoke package, 12-7
revoke plan, 12-9
revoke schema, 12-21
revoke stored procedure, 12-23
revoke table, 12-14
revoke tablespace, 12-18
revoke UDF, 12-27
revoke view, 12-14
schema, 12-19
stored procedure, 12-21
table, 12-10
tablespace, 12-16
view, 12-10
product
capabilities, 1-1
conventions, 2-1–2-10
functions, 1-3
overview, 1-1
PROFILE (PRO) command, 3-39
PROFILE (PROF) command, 6-3
prototyping SQL code, 1-3
PSID, 10-10
pull-down, B-1
actions, B-10
define, B-9
edit, B-2
file, B-1
help, B-8
menu, B-5
open, B-5
options, B-4
utilities, B-6
view, B-3

Q

QUALIFIER, 10-15, 10-19
QUOTE, 10-16, 10-21

R

R (Repeat) line command, 4-4
RBA1, 10-9
RBA2, 10-9
RBROWSE (RB) command, 3-41
RBROWSE command, 7-7
RCHANGE command, 3-42
RECLENGTH, 10-8
record layout
allocation, 14-16
generating, 14-10, 14-12
record layout file
MVS Relationship Facilitator screen, 18-12
REDIT (RE) command, 3-42
REDIT command, 7-7
Redit/Rbrowse
using views, 7-11
referential integrity
browse window, 6-29
resolving errors, 6-27
resolving errors in row mode, 6-30
resolving errors in table mode, 6-29
referential integrity (RI), 18-1
Referential Integrity (RI) relationship, 18-25–18-30
line commands
D (Delete), 18-27
I (Information), 18-27
S (Select), 18-27
load RI, 18-4
overview, 18-25
primary commands
GRAPHIC, 18-28
relationship file
modifying, 18-25
status messages, 18-24, 18-27, 18-29
viewing RI relationship information, 18-28
referential integrity (RI) relationship
overview, 7-1
RI line command, 4-5
using in related table edit, 7-5
refresh relationship file, 18-28
related table browse and edit, 7-1
accessing, 7-7
changing parent record, 7-42

deleting data, 7-41
graph a relationship, 7-42
in row mode, 7-34
in zoom mode, 7-35
inserting data, 7-40
model relationships, 7-12, 7-16
navigating through multiple windows, 7-31
one-to-many table relationship, 7-12
one-to-one table relationship, 7-12
opening additional related table paths, 7-36
parent relationship, 7-14
scrolling windows, 7-32
sizing windows, 7-33
switching to other paths, 7-38
using, 7-11
related table information, 16-33
relationship
determine, 7-4
determine relationship diagram, 7-5
model, 7-16
one-to-many table, 7-12
one-to-one table, 7-12
parent, 7-14
relationship dataset
characteristics, 7-3
creating and loading, 7-4
sample, 7-12, 7-18
specifying, 6-7, 7-7, 7-11
relationship definition
application relationship overview, 7-2
deleting an application relationship, 7-19
RI relationship overview, 7-1
RI relationship, using in related table edit, 7-5
Relationship Detail, 18-15
line commands
I (Information), 18-16
primary commands
ADD, 18-16
GRAPHIC, 18-17
selecting a relationship, 18-15
Relationship Facilitator
add AR, 18-4, 18-11
application relationship, 18-31, 18-58
building a column relationship, 18-56
viewing AR information, 18-33
application relationship overview, 18-31
Browse, 18-11
browse, 18-3
browsing information about a relationship, 18-20
edit AR, 18-3, 18-11
line commands, 18-6, 18-13
A (Add AR), 18-6
A (Add), 18-13
B (Browse), 18-7, 18-13
E (Edit AR), 18-7, 18-13
L (Load RI), 18-7
list DB2 catalog, 18-4
list RDX, 18-4, 18-11
load RI, 18-4
modifying an application relationship, 18-50
Object List in Relationship File
selecting a table, 18-14
print, 18-11
print relationships, 18-68–18-73
relationship detail
selecting a relationship, 18-15
RI relationship, 18-25, 18-28
deleting, 18-27
viewing RI relationship information, 18-28
RI relationships, 18-25
selecting a table, 18-5
selecting an application relationship to add or modify, 18-31
selecting an MVS object, 18-12
selecting an RI relationship to load, refresh, view, 18-25
specifying a DB2 object and relationship file, 18-3
specifying a table and relationship file, 18-3
specifying an MVS file and relationship file, 18-10
Table List - SYSIBM.SYSTABLES
selecting a table, 18-9
Table List in Relationship File
selecting a table, 18-8
viewing AR information, 18-31
relationship facilitator
overview, 18-1
Relationship Facilitator primary commands
DB2, 18-12
MVS, 18-5
relationship file, 7-3, 18-1
DB2 long name support, 7-3, 18-1
loading RI from DB2 catalog, 18-27
modifying, 18-50
RELBOUND, 10-21
RELCRETD, 10-9
RELEASE, 10-15, 10-20
REMARKS, 10-14, 10-18
remarks, 16-35
comment on, 10-7
REMARKS command, 3-43
REMOTE, 10-22
rename table, 8-23
REOPTVAR, 10-16, 10-21
REPEAT (REP) command, 3-43
repeat command prompt, 3-41
report
format skeletons, 16-42
report, print DB2 explain reports, 16-41
RESET (RES) command, 3-48
result
sets, 8-76, 10-27
table, 16-53–16-54
table, browsing, 16-53
table, editing, 16-54
result sets, 10-33
RETPARM command, 3-50
RETRIEVE command, 3-50
return code, 16-57
RETURN command, 3-50
return type, 10-32
review or print DB2 explain reports, 16-41
revoke
collection privileges, 12-5
distinct type privileges, 12-25
package privileges, 12-7
plan privileges, 12-9
privileges, 12-1
privileges on object, 10-4
schema privileges, 12-21
stored procedure privileges, 12-23
table/view privileges, 12-14
tablespace privileges, 12-18
UDF privileges, 12-27
REVOKE (REV) command, 3-50
RFIND command, 3-50
RI (referential integrity) line command, 4-5, 6-30

RI primary command, 3-50
RI relationship, 18-4
RI relationships, 18-1
RI, load, 18-4
RIGHT command, 3-51
ROOTCREATOR, 10-11
ROOTNAME, 10-11
routine type, 10-32
routineid, 10-29, 10-31
ROW command, 3-51
row mode
  browse, 6-17
  definition, 6-6
  definition for printing, 15-4, 17-3
  edit, 6-17
  editing multiple tables, 6-26
  resolving RI errors, 6-30
  working with, 6-17
ROWH command, 3-52
ROWID, 8-6
rowid, 6-18, 14-23
rows
  discarded, 14-23
  maximum to select, 5-3
  select rows to extract, 14-6
rules, predicate analysis, A-1
run time
  information, 10-26, 10-31
  option for stored procedure, 8-81
  option for user defined functions, 8-67
RUNOPTS command, 3-52

S

S (Select) line command, 4-5
  Relationship Detail screen, 18-16
S (select) line command
  Application Relationship Detail screen, 18-32
  Object List in Relationship File, 18-14
  RI Relationship Detail screen, 18-27
  Table List - SYSIBM.SYSTABLES, 18-9
  Table List in Relationship File, 18-8
S (Show) line command, 4-5
SAVE command, 3-52, 7-39
saving changes in REDIT, 7-39
schema
  display privileges, 12-19
  grant privileges, 12-20
  modify privileges, 12-21
  privileges, 12-19
  revoke privileges, 12-21
screen names
  Add Application Relationships DB2 to DB2, 18-35
  Add Relationships-Specify Object Types, 18-34
  DB2 Object List, 18-5
  Edit Print JCL, 18-70
  Formatted Relationship Report, 18-68
  Graphical Relationship Display, 18-18
  Option List (Graphical Relationship Display), 18-19
  Print Options, 18-69
  Relationship Detail, 18-15
  Relationship Information, 18-20
  Relationships, 18-50
SCROLL field, 2-8
SECQTY, 5-7–5-8
security, 8-77, 10-28, 10-34
SEGSIZE, 5-8, 10-11
select
  rows to extract, 14-6
  table to extract, 14-2
  table to extract/load, 14-6
selecting a file
  Relationship Facilitator, 18-12
selecting a table
  Relationship Facilitator, 18-5
selection
  criteria member list, 6-8
  template, 6-9
SHOW SQL
  allocate dataset, 5-5
SMF audit log
  browse, 6-32
  edit, 6-32
snap, column, 6-2
SORT command, 3-52
  in an object list, 3-54
  in browse or edit, 3-52
  in plan analysis, 16-12–16-13, 16-15, 16-17, 16-19
source code edit session, 2-4
SPACE, 10-11
SPACEF, 10-10
special register, 8-78, 10-28
SPLIT command, 3-55
SPUFI, 11-1
  edit SPUFI selection, 17-8
  extract through, 14-9
  file, 16-45
  selection, 6-13
SQL
  accessing analysis, 16-2
  allocate SHOW SQL dataset, 5-5
  analysis and development menu, 16-2
  analysis batch output specification, 16-36
  analysis edit, 16-47
  analysis edit entry, 16-44, 16-59
  analysis of statements, 16-1
  browse statements, 16-48
  changing statements, 16-54
  command, 3-55
  command manager utility, 1-5, 11-1–11-2
  data, 10-27, 10-33
  dataset allocation, 5-5
  development and analysis, 1-5, 16-1
  development of statements, 16-1
  dynamically changing statements, 16-54
  edit statements, 16-48
  entry, 6-12
  entry panel, 9-8, 14-9, 17-7
  execute, 16-50
  execute statements, 16-48, 16-51–16-53
  explain report specification, 16-41
  explain statements, 16-12, 16-14, 16-48
  log information, 5-9
  performing development and analysis, 16-49
  plan analysis specification, 16-8
  plan selection, 16-11
  plan table row deletion, 16-6
  predicate analysis, 16-25, A-1
  return code, 16-57
  selecting statements for analysis, 16-47
  statement selection, 16-17
  statements, 16-1, 16-49
SQL coding error, 16-57
  help, 16-57

SQL DATA, 8-77
SQLERROR, 10-20
SQLID, 8-75
SQLID command, 3-55
SQLID primary command
  Relationship Detail screen, 18-17, 18-28
SRCPARM command, 3-56
statement
  CREATE TRIGGER, 15-1
statements, SQL
  analysis of, 16-1
  development of, 16-1
  explain, 16-12
STATSTIME, 10-9, 10-12
STATUS, 10-9, 10-11
stay resident, 8-76, 10-28, 10-35
STOGROUP, 5-7–5-8
stored procedure, 8-74, 10-2
  alter, 8-83
  comment on, 8-82
  create, 8-75
  display privileges, 12-21
  drop, 8-84
  execute create, 8-68, 8-83
  grant privileges, 12-22
  information, 10-25
  modify privileges, 12-23
  parameter, 8-78
  privileges, 12-21
  revoke privileges, 12-23
  run time option, 8-81
  view/edit ddl, 8-68, 8-82
stored procedures, 8-1
SUB, 8-65, 8-73, 8-76, 10-27, 10-33
SUBPAGES, 5-9
subscript
  specify field subscripts, 18-65
subsystem identification, updating, 2-8
SUGGEST primary command
  Column Relationships screen, 18-55
suggesting relationships
  DB2 to DB2, 18-36
  DB2 to IMS, 18-41
  DB2 to MVS, 18-38
  MVS to DB2, 18-43
  MVS to IMS, 18-48
  MVS to MVS, 18-46
support
  operating environment, -xxvi
SWAP command, 3-56
SYNCSORT
  Note concerning print JCL, 18-71
synonyms, 8-45
  create, 8-46
  drop, 8-47
  information, 10-17
  modify, 8-46
SYSENTRIES, 10-20–10-21

T

table
  altering, 8-20
  create, 8-3, 8-8
  create screen, 8-3
  data, 17-1
  display privileges, 12-10
  drop, 8-22
  extract from, 10-5
  grant privileges, 12-12
  index selection, 16-31
  information, 10-6
  load, 10-5
  modify privileges, 12-16
  option, 8-2
  partition statistics, 16-34
  print data, 17-1
  print output specification, 17-9
  privileges, 12-10
  related table information, 16-33
  rename, 8-23
  revoke privileges, 12-14
  select table to extract, 14-2
  selection, 6-9
TABLE command, 3-56
table list
  copy from, 9-3
  copy to, 9-4
Table List - SYSIBM.SYSTABLES
  line commands
    S (Select), 18-9
  selecting an object, 18-9
Table List in Relationship File
  line commands
    S (Select), 18-8
  selecting an object, 18-8
table mode
  browse, 6-14
  definition, 6-6
  definition for printing, 15-4, 17-3
  edit, 6-14
  working with, 6-14
tablespace
  alter, 8-29
  alter partitioned, 8-31
  confirm drop, 8-33
  create, 8-25
  create partitioned, 8-28
  drop, 8-33
  grant privileges, 12-17
  information, 10-10
  modify privileges, 12-18
  options, 8-24
  privileges, 12-16
  revoke privileges, 12-18
TABSTATUS, 10-8
TBCREATOR, 10-18
TBNAME, 10-18
TBName, 10-23
TBOwner, 10-23
TEXT, 10-14
TIMESTAMP, 10-16, 10-19
TOP command, 3-57
TRACKMOD, 5-7
trigger, 8-85, 9-2, 10-7, 13-3, 14-26, 14-29
  begin atomic...end, 8-90
  comment on, 8-90, 10-24, 16-35
  create, 8-86
  drop, 8-92
  event, 10-23
  information, 10-22
  package, 16-15
  related information, 16-34
  remarks, 16-35

terminator, 5-11
text, 10-24
time, 10-23
update column, 8-88
view/edit ddl, 8-91
warning, 9-5
WHEN search condition, 8-89
triggers, 6-19, 7-1, 14-1, 14-6
TSNAME, 10-7, 10-14
tutorials, 2-10, 16-57
TYPE, 10-2, 10-8, 10-22

U

UDF, 8-1, 10-2
display privileges, 12-25
grant privileges, 12-26
modify privileges, 12-27
revoke privileges, 12-27
specific name, 10-2
UDFOPTS command, 3-57
UDT, 10-29, 10-36
autocast, 3-58
UDT columns, 7-1, 14-1
UDT command, 3-57
uncommitted read
use for browse/edit, 6-7
UNID, 6-19
UNIQ, 6-18
UNIQUE, 6-16
unique key, 5-2
UNIQUERULE, 10-12
unrelated tables, 6-22
UP command, 3-59
update
identity column, 3-41
verify, 6-20
Upgrade Plan Table, 16-4
USER, 8-65, 8-72, 8-77, 10-28, 10-34
user defined function (UDS)
information, 10-31
user defined functions (UDFs), 8-1, 8-52, 10-2
alter function, 8-68
comment on, 8-67
create external scalar, 8-53
external table, 8-57
function options, 8-63
parameter information, 8-70
parameters, 8-63
run time option, 8-67
sourced function, 8-59
specific name, 10-2
view/edit DDL, 8-68
user parameters, 5-1–5-11
batch options, 5-6
browse, edit, sql analysis display options, 5-2
current path options, 5-11
index, tablespace, and database default values, 5-7
log options, 5-9
print options, 5-10
SQL dataset allocation, 5-5
trigger terminator, 5-11
user parms, 1-4
utilities, 1-5
batch execute with Auto-COMMIT processing, 1-5
copy rows between tables, 1-5, 9-1
create, drop, alter DB2 objects, 1-5, 8-1
DBA-XPERT column impact analysis utility, 1-5
display, grant, revoke, and modify DB2 privileges, 1-5, 12-1
extract and load tables, 1-5
impact analysis, 13-1
object list, 1-5, 10-1
pull-down, B-6
selection menu, 8-1
SQL command manager, 1-5, 11-1
utilities, extract and load tables, 14-1
Utility Selection Menu, 8-1

V

VALID, 10-15, 10-21
VALIDATE, 10-14, 10-21
VALPROC, 10-9
VALUES command, 3-59
VCAT, 5-8
verify update, 6-20
VERSION, 10-16, 10-19
view
considerations for editing, 6-4
create, 8-44
ddl, 8-68, 8-82
display privileges, 12-10
drop, 8-45
editing, 6-4
grant privileges, 12-12
information, 10-14
modify privileges, 12-16
options, 8-43
privileges, 12-10
pull-down, B-3
revoke privileges, 12-14
use in related edit/browse, 7-11
use in relationship definition, 18-2

W

WHEN search condition, 8-89
WHENCOND command, 3-59
wildcard characters, 2-9
for PDS member fields, 2-9
for relationship option, 2-9
WLM
env nested, 10-29, 10-36
environment, 8-78, 10-29, 10-35
WRAP command, 3-60

X

X (Exclude) line command, 4-6
XP line command, 4-6, 16-47
XPEDITER for DB2, 1-6

Z

ZOOM command, 3-60

zoom mode, 7-35
ZOOMH command, 3-60