IBM InfoSphere Optim for z/OS
Version 11 Release 3

Access User Manual

IBM InfoSphere Optim for z/OS
Version 11 Release 3

Access User Manual

Note

Before using this information and the product it supports, read the information in “Notices” on page 119.

Version 11 Release 3

This edition applies to version 11, release 3 of IBM InfoSphere Optim for z/OS and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication

This document explains the use of Access to edit or browse data in one or more related tables.

Chapter 1. Introduction

IBM® InfoSphere® Optim™ Solution for z/OS® (hereafter referred to as Optim), manages enterprise data throughout every stage of the information lifecycle. Optim enables you to assess, classify, subset, archive, store, and access enterprise application data.

Optim uses the relationships defined in the DB2® Catalog, where available, and supplements these relationships with those defined in the Optim Directory. Optim runs as a TSO/ISPF application and incorporates familiar ISPF commands. Optim handles any number of tables and any number of relationships, regardless of the complexity.

Optim helps you achieve these benefits with the following components: Access, Archive, Move, and Compare. You can use these Optim components for test data management, data privacy, data retention, application retirement, and data growth management. This manual describes how to use Compare in test data management.

Test Data Management

The Optim test data management capabilities provide an efficient alternative to database cloning, allowing you to create development and testing environments that are sized appropriately.

For information about the test data management functions of Optim, see the *Move User Manual*, the *Compare User Manual*, and this user manual.

Access is an online productivity tool for browsing and editing related data residing in multiple IBM DB2 tables. This interactive utility is useful for programmers, database administrators, DB2 support technicians, or anyone needing a global view of data in a DB2 database.

A major reason that companies invest in relational technology is to have greater flexibility in accessing corporate information. In DB2, information is normalized into simple elements and stored in multiple tables. Application programs access the related tables to “re-assemble” the information. However, faster, more flexible, and dynamic access to the data is needed at many times during the development, testing, and maintenance of applications.

Access provides this access to related data in multiple DB2 tables and simplifies many daily maintenance tasks, such as:

- Editing test data to incorporate special test cases and erroneous data in order to verify all logic paths in an application.
- Browsing one or more tables to ensure that data is as expected. This includes inspecting specific rows or columns that may be causing problems.
- Correcting data problems.
- Verifying SQL statements embedded in programs.

A programmer or database administrator can easily inspect and edit sets of related data at one time on one screen. Access eliminates the time-consuming efforts of manually “assembling” data from different tables.

Data Privacy

Data privacy is a licensed function of test data management.

For information about the general test data management functions of Optim, see the *Move User Manual*, the *Compare User Manual*, and this user manual. Data transformations for privacy are accomplished through the use of Optim column maps. For information needed to transform data using a column map, see the *Common Elements Manual*, Column Maps Section.

Data Retention, Application Retirement, and Data Growth Management

You can use the archiving features in Optim to:

- Isolate historical data from current activity and safely remove it to a secure archive.
- Access archived data easily, using familiar tools and interfaces.
- Restore archived data to its original business context when it requires additional processing.

For information about the archive functions of Optim, see the *Archive User Manual*.

Processing Order

In general, information is processed during an Access session in a set order.

Information is processed the following order:

1. All data editing is evaluated.
2. Line commands are processed.
3. Function key requests and primary commands are processed.

If data editing introduces a syntax error, all command processing is suspended and an appropriate error message is displayed. You must correct the error to resume processing.

If a line command has a syntax error, all command processing is suspended and an appropriate error message is displayed. You must correct or delete the line command to resume processing.

If a primary command is in error, all commands before the error are processed. All other processing is suspended and an appropriate error message is displayed. You must correct or delete the erroneous command to resume processing.

Exceptions

Certain primary commands take precedence over all other processing and are performed first. These commands include CANCEL, HELP, RESET, ROLLBACK, and RESTART.

Also, when using a line command that requires more than one entry, such as a block command or the destination for a copy or move operation, you can scroll to complete the command specification or to enter a destination. Pending block commands or Move/Copy commands must be resolved prior to processing other line commands.

General Information

This section presents general information about Access, including a description of the elements common to the Optim components, terminology used to describe Access, and the sample database.

Note: For general information about naming conventions and screen format and handling, refer to the *Common Elements Manual,* Introduction.

Common Elements

The components of Optim provide varied functions.

Access is the relational facility that lets you browse and edit related data residing in multiple DB2 tables. Archive enhances database performance by facilitating the removal of infrequently referenced data. Compare is the relational comparison facility that lets you compare sets of related data from two database structures. Move is the relational copy facility that lets you extract sets of related data from DB2 or Legacy tables, and insert that data into destination databases and files. Features used throughout the Optim components are discussed in the *Common Elements Manual*.

To carry out their functions, the Optim components rely upon user-defined objects that supplement objects defined to the database (for example, tables, primary keys, relationships, stored procedures). These user-defined objects (collectively, Optim objects) are stored in the Optim Directory.

Optim objects that are common to the Optim components include:
- Access Definitions
- Primary Keys
- Relationships
- Column Maps
- Table Maps

The following processes and facilities are common to the Optim components:
- Export/Import
- Convert
- Retry/Restart
- Browse

The *Common Elements Manual*, Options section, describes the various options that allow you to manage Optim.

Terminology

The following paragraphs describe some common terms that are used in this manual.

Access Definitions

An **Access Definition** describes a set of DB2 data. For Access, this is the data to be accessed during an edit or browse session. An Access Definition incorporates a variety of specifications, including
- The list of tables.
- Optional instructions for retrieving and presenting the data in the tables. These instructions include
  – The columns to be displayed.
  – The format of the display.
  – The rows to be selected.
  – The sequence of the rows.
- Other browse and edit parameters.

You can save an Access Definition for repeated future use. The saved definition can be modified and re-saved under the same or a new name. Once saved, the same Access Definition can be used by Archive to archive and restore data, by Compare to compare data, and by Move to extract data.

An Access Definition name consists of three parts:

```
group.user.name
```

The *group* and *user* portions of the name are useful for organizing Access Definitions. For example, you can assign a unique group name for Access Definitions used in each project and, within each project, a unique value for each person (user) in the group.

Primary Keys

A **primary key** identifies a column or set of columns that uniquely identifies each row in a table. For example, the CUSTOMERS table has a column named CUST_ID that contains a unique value for each row in the table. The CUST_ID column is an acceptable primary key for the table.

Optim Directory

The Optim Directory contains information needed to access DB2 data. This information includes user-specified

- Access Definitions
- Primary keys
- Relationships.

Access Definitions are unique to Optim, but primary keys and relationships are often available from the DB2 Catalog. Access uses information in the DB2 Catalog whenever possible. However, if DB2 Catalog information is not available, you can specify and store the information in the Optim Directory.

Referential Integrity

Access uses referential integrity information and table and column information from the DB2 Catalog. When the DB2 Catalog does not provide needed relationship information, user-specified objects, stored in the Optim Directory, supplement the DB2 Catalog.

Relationships

A **relationship** determines how two tables are related. A relationship can be defined in the DB2 Catalog or the Optim Directory. Whenever a list of relationships is presented, the source is indicated. You can create or modify an Optim relationship using Optim, or you can browse DB2 relationships. You can also use DB2 relationships as a “model” for new Optim relationships.

In the DB2 Catalog, a relationship is defined by a primary key/foreign key pairing. The foreign key is the set of columns in a child table that describes the correspondence with the primary key columns in the parent table. For example, the ORDERS table contains a column CUST_ID that can be related to the primary key column CUST_ID in the CUSTOMERS table. The column in the ORDERS table is the foreign key.

However, Optim relationships do not require primary key/foreign key pairing. You can define relationships in the Optim Directory that pair any compatible columns between two tables. You can also define Optim relationships using substring and concatenation functions for columns, as well as literal and constant values. Optim relationships are defined by and available to all components of Optim interchangeably. You cannot define an Optim relationship for a Materialized Query Table.

Table Levels

The set of tables described by an Access Definition do not have to be related in a hierarchical manner. Rather, tables are assigned levels on the basis of the sequence in which they are displayed when the Access Definition is used to retrieve data. The first table in the display is the highest-level table, the next table is the second highest-level, and so forth. The last table in the display is the lowest-level table.

Tables

Throughout this document, the term **tables** refers to DB2 tables, views, joined views, aliases, and synonyms. These objects are manipulated similarly. The differences in handling are noted where pertinent.

Data from Materialized Query Tables (MQTs) can be browsed. You can edit data in a User-Maintained MQT. System-maintained MQTs are protected from modification.

Although data from a joined view can be displayed, it cannot be edited. This adheres to the DB2 rules governing joined views. However, because Access allows you to browse and edit related data from multiple tables simultaneously, the use of joined views can be avoided.

Aliases are supported, but remote aliases are not.

Sample Database

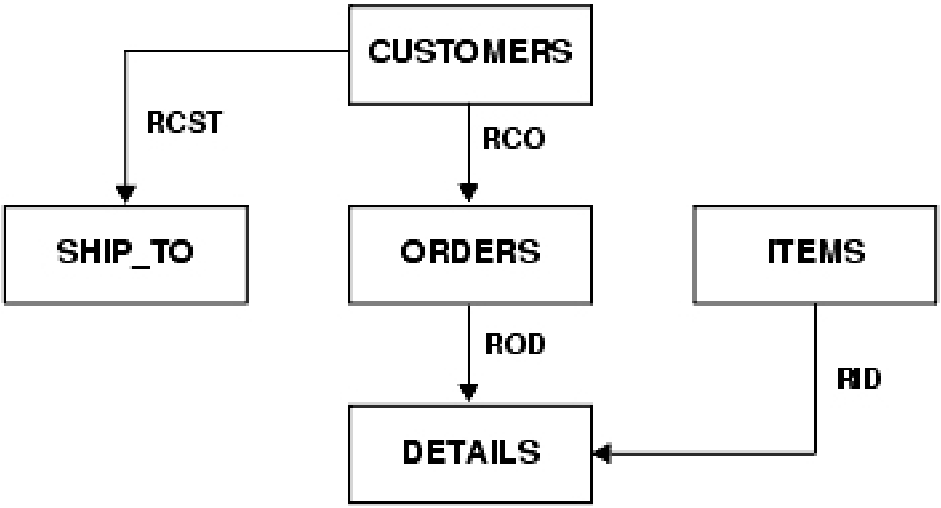
A sample database is distributed with Optim.

The sample database is created as part of the installation and is described fully in the *Common Elements Manual*, Appendix A. Sample Database Tables and Structure. It provides data for training and allows you to experiment with Optim without fear of disrupting your production database.

The sample database is used in the sample session in Chapter 2, "A Sample Session," on page 7and in other examples in this manual. This database includes the following tables (names are prefixed with the Creator ID FOPDEMO):

- OPTIM_CUSTOMERS
- OPTIM_ORDERS
- OPTIM_DETAILS
- OPTIM_SALES
- OPTIM_ITEMS
- OPTIM_SHIP_TO
- OPTIM_SHIP_INSTR
- OPTIM_FEMALE_RATES
- OPTIM_MALE_RATES
- OPTIM_STATE_LOOKUP

The tables in the following chart are used in this manual. The chart shows these tables and the relationships among them. The arrows indicate the flow from parent to child. (The OPTIM_ prefix for each table name is not shown in the chart.)



If you use the Sample Session as a tutorial, note that relationships may have been added to the sample database at your facility during training or other activities. The table names in the sample scenario are shown without the OPTIM_ prefix. To use the scenarios, prefix the table names with OPTIM_

Chapter 2. A Sample Session

This sample session demonstrates the basics of selecting and editing data with Access.

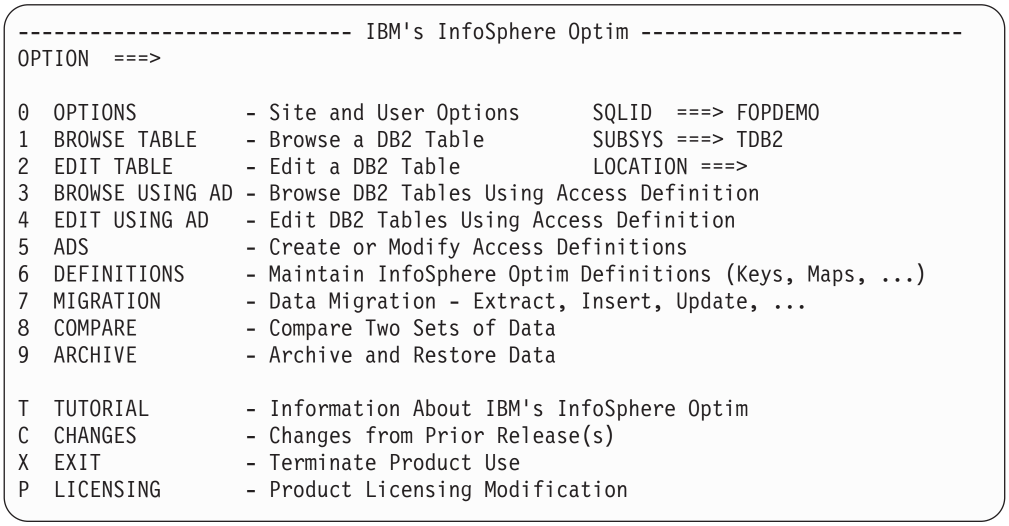
It shows how to specify selection and sort criteria and to scroll and edit the data. This session also demonstrates how to display related data from other tables.

This section provides an overview of the Access facilities. For details on defining Access Definitions, see the *Common Elements Manual,* section for Access Definitions. For details on editing and scrolling data, see Chapter 3, “Editing and Browsing DB2 Data,” on page 19 in this user manual.

The Main Menu

When Access is invoked, the **Main Menu** is displayed, as in the following figure.

The options available on the **Main Menu** may vary according to the components that are installed. Option 7 MIGRATION is available only if Move, Optim Legacy, or Compare is installed; Option 8 COMPARE is available only if Compare is installed; and Option 9 ARCHIVE is available only if Archive is installed. An option that is not available is marked with an asterisk.



```
---------------------------- IBM's InfoSphere Optim ---------------------------
OPTION  ===>

0  OPTIONS         - Site and User Options      SQLID  ===> FOPDEMO
1  BROWSE TABLE    - Browse a DB2 Table         SUBSYS ===> TDB2
2  EDIT TABLE      - Edit a DB2 Table           LOCATION ===>
3  BROWSE USING AD - Browse DB2 Tables Using Access Definition
4  EDIT USING AD   - Edit DB2 Tables Using Access Definition
5  ADS             - Create or Modify Access Definitions
6  DEFINITIONS     - Maintain InfoSphere Optim Definitions (Keys, Maps, ...)
7  MIGRATION       - Data Migration - Extract, Insert, Update, ...
8  COMPARE         - Compare Two Sets of Data
9  ARCHIVE         - Archive and Restore Data

T  TUTORIAL        - Information About IBM's InfoSphere Optim
C  CHANGES         - Changes from Prior Release(s)
X  EXIT            - Terminate Product Use
P  LICENSING       - Product Licensing Modification
```

Figure 1. Main Menu

Panel Options

To select an option, type the corresponding one-character identifier and press ENTER. The options are:

0 OPTIONS

Specify product options, including user options, editor and display options, job card and print options, Compare options, Archive options, and Legacy options. For details, see the *Common Elements Manual,* section for Options.

1 BROWSE TABLE

Browse data in a DB2 table.

2 EDIT TABLE

Edit data in a DB2 table.

3 BROWSE USING AD

Browse DB2 data defined by an Access Definition.

4 EDIT USING AD

Edit DB2 data defined by an Access Definition.

5 ADS

Create and maintain Access Definitions. For details, see the *Common Elements Manual,* section for Access Definitions.

6 DEFINITIONS

Define and maintain Access Definitions and primary keys and relationships that are not available in the DB2 Catalog. If other components are installed, you can also use this option to define and maintain Table Maps and Column Maps. This option is also used to Export and Import Optim object definitions, and maintain Archive Collections. For details, see the *Common Elements Manual*, section on Choose a Definition Option.

If Move or Compare for IMS, VSAM or sequential data is installed, this option allows you to create and maintain Legacy Tables, IBM IMS™™ Environment Definitions, and IMS Retrieval Definitions. For details, see the *Move User Manual,* sections for Legacy Tables, IMS Environment Definition, and IMS Retrieval Definition or *Compare for IMS/VSAM/Sequential File Data*.

7 MIGRATION

Perform the Move processes for extracting, inserting, updating, loading, creating, converting, and browsing DB2 or Legacy data, or the Compare Extract and Browse Processes. These processes are documented in the *Move User Manual*, section for Data Migration, and the *Compare User Manual,* sections for Extract Data and Browse Compare File.

8 COMPARE

Compare one set of tables with another and browse the results. This facility is documented in the *Compare User Manual*, section for Compare Process.

9 ARCHIVE

Perform the Archive processes for archiving data, browsing and searching the archives, and selectively restoring the archived data. This facility is documented in the *Archive User Manual*, section on Session Overview.

T TUTORIAL

Display the online Tutorial.

C CHANGES

Display a list of enhancements for the current release.

X EXIT

Terminate Optim.

P LICENSING

Display a list of the Optim components and their releases. The status for each component is identified as “In Evaluation: n Days Left” or “Not Installed.” Administrator privileges are required to enable or disable a component. This facility is documented in the *Customization Guide*, section for Enable and Disable Products.

Panel Prompts

Values in the panel prompts are profiled. The prompts are:

SQLID
The current SQLID. Modify this value to connect using a different SQLID.

SUBSYS
The current DB2 subsystem. Modify this value to connect to a different DB2 subsystem.

When connecting to a remote subsystem, this value should be the local subsystem where the remote location is defined.

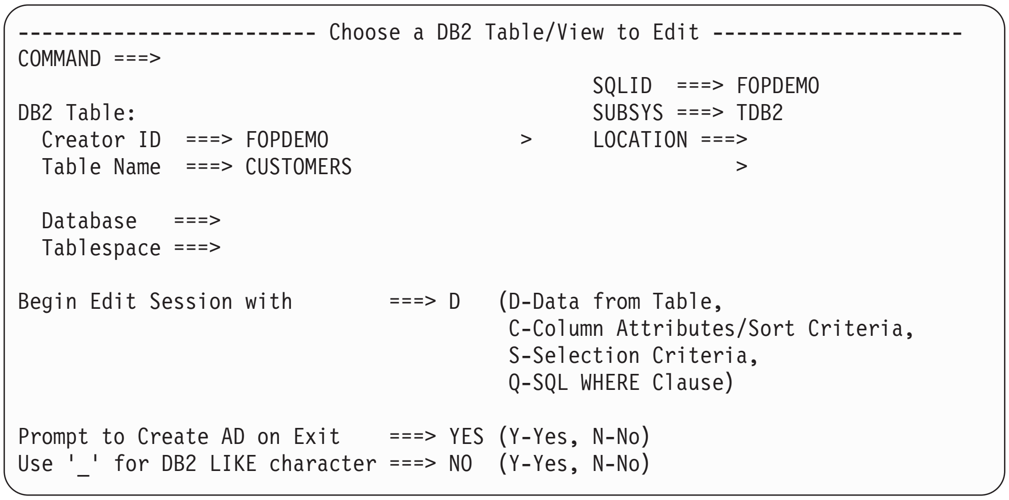
LOCATION
The remote location. This prompt is displayed if remote access is available. Specify a value to connect to a remote DB2 subsystem. You can use a percent sign (%) to obtain a selection list of available locations.

Note: If you leave this prompt blank, the local subsystem is assumed.

Initiating an Edit Session

When Option 2 EDIT TABLE is selected on the **Main Menu**, the **Choose a DB2 Table/View to Edit** panel is displayed to prompt for the name of a table to be edited.

You specify the table name by supplying the **Creator ID** and the **Table Name** as shown in the following figure. In this example, the table is named FOPDEMO.CUSTOMERS.



```
------------------------- Choose a DB2 Table/View to Edit ---------------------
COMMAND ===>
                                                SQLID  ===> FOPDEMO
DB2 Table:                                      SUBSYS ===> TDB2
  Creator ID  ===> FOPDEMO                >     LOCATION ===>
  Table Name  ===> CUSTOMERS                                >

  Database   ===>
  Tablespace ===>

Begin Edit Session with        ===> D   (D-Data from Table,
                                         C-Column Attributes/Sort Criteria,
                                         S-Selection Criteria,
                                         Q-SQL WHERE Clause)

Prompt to Create AD on Exit    ===> YES (Y-Yes, N-No)
Use '_' for DB2 LIKE character ===> NO  (Y-Yes, N-No)
```

Figure 2. Table FOPDEMO.CUSTOMERS is Specified

You can obtain a selection list by entering blanks or using DB2 LIKE syntax at either or both **DB2 Table** prompts. You supply a value in **Database** or **Tablespace** to limit the list to tables in a specific database or tablespace.

Initial Session Display

Use the **Begin Edit Session with** prompt on the Choose a DB2 Table/View to Edit panel to control the initial display when you start the edit session. You can specify:

D Display the data from the specified table.

C Display a prompt for column specifications.

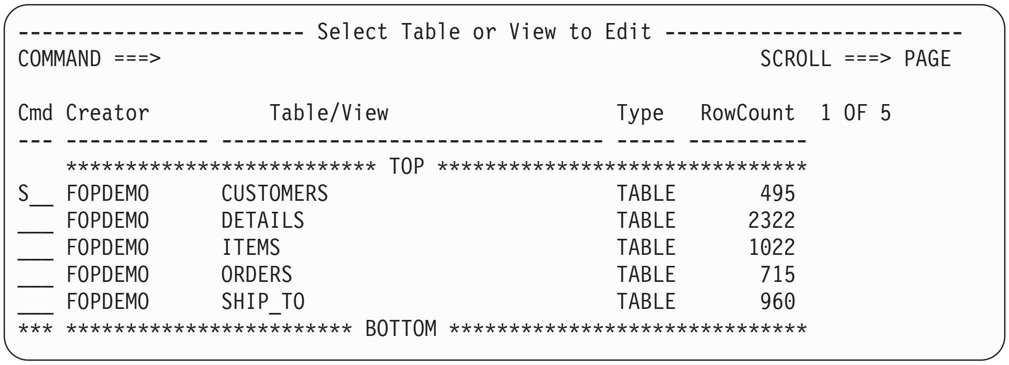
S Display a prompt for selection criteria.

Q Display a prompt for an SQL WHERE clause.

These prompts are discussed in detail later in this manual. For this demonstration, specify D in response to the **Begin Edit Session with** prompt.

Selecting Table from List

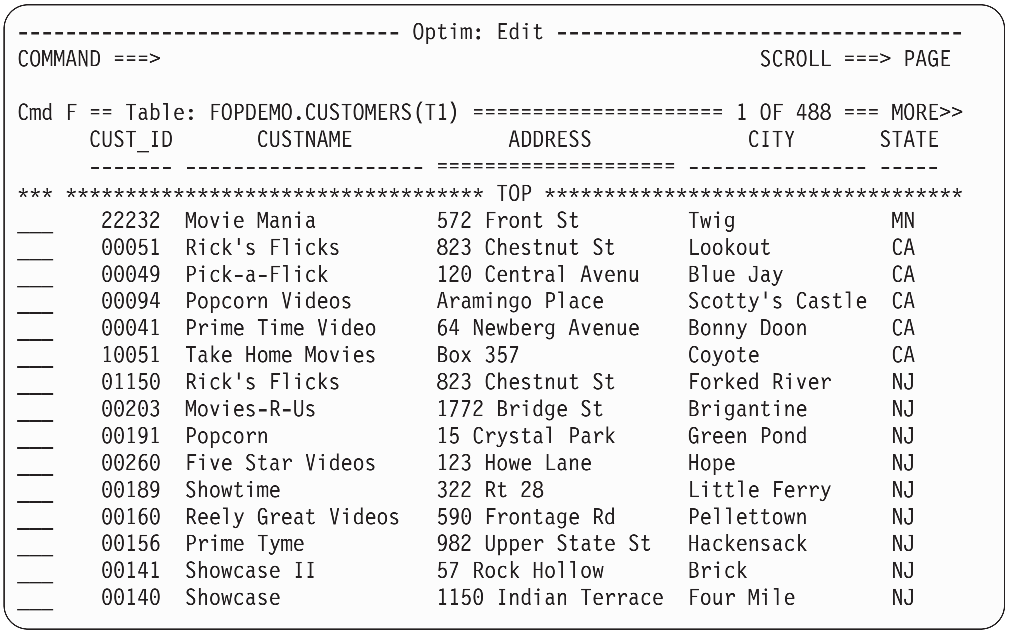
If only the Creator ID, FOPDEMO, is supplied on the Choose a DB2 Table/View to Edit panel, the following selection list is displayed. The list includes the Creator ID, table or view name, type, and row count. **RowCount** contains the number of rows in the table if the IBM RUNSTATS utility has been executed against the table. If not, **RowCount** is blank. **RowCount** is also blank for views.



```
------------------------- Select Table or View to Edit -------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd Creator          Table/View                    Type   RowCount  1 OF 5
--- ------------ -------------------------------- ----- ----------
    ************************** TOP *******************************
S__ FOPDEMO      CUSTOMERS                        TABLE       495
___ FOPDEMO      DETAILS                          TABLE      2322
___ FOPDEMO      ITEMS                            TABLE      1022
___ FOPDEMO      ORDERS                           TABLE       715
___ FOPDEMO      SHIP_TO                          TABLE       960
*** ************************ BOTTOM ******************************
```

Figure 3. CUSTOMERS Table is Selected

Use the S line command to select the CUSTOMERS table. The display begins with the data from the table, as shown in the following figure.



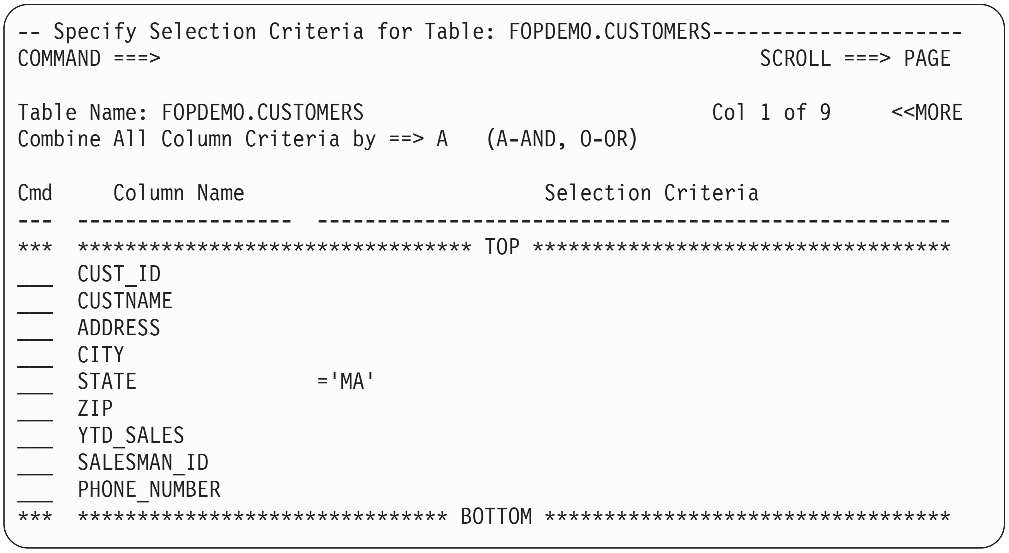
```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ==================== 1 OF 488 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    22232  Movie Mania          572 Front St         Twig            MN
___    00051  Rick's Flicks        823 Chestnut St      Lookout         CA
___    00049  Pick-a-Flick         120 Central Avenu    Blue Jay        CA
___    00094  Popcorn Videos       Aramingo Place       Scotty's Castle CA
___    00041  Prime Time Video     64 Newberg Avenue    Bonny Doon      CA
___    10051  Take Home Movies     Box 357              Coyote          CA
___    01150  Rick's Flicks        823 Chestnut St      Forked River    NJ
___    00203  Movies-R-Us          1772 Bridge St       Brigantine      NJ
___    00191  Popcorn              15 Crystal Park      Green Pond      NJ
___    00260  Five Star Videos     123 Howe Lane        Hope            NJ
___    00189  Showtime             322 Rt 28            Little Ferry    NJ
___    00160  Reely Great Videos   590 Frontage Rd      Pellettown      NJ
___    00156  Prime Tyme           982 Upper State St   Hackensack      NJ
___    00141  Showcase II          57 Rock Hollow       Brick           NJ
___    00140  Showcase             1150 Indian Terrace  Four Mile       NJ
```

Figure 4. Selected Table Display

Selecting and Sorting a Set of Rows

Use selection criteria to limit the set of selected rows. Type the SELECTION CRITERIA command (or SEL) at the command prompt and press ENTER to display the **Specify Selection Criteria for Table** panel.

When the Selection Criteria for Table panel is displayed, supply the selection criteria on the appropriate line. For example, to select rows for the customers in Massachusetts, type = 'MA' for the STATE column.



```
-- Specify Selection Criteria for Table: FOPDEMO.CUSTOMERS---------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Table Name: FOPDEMO.CUSTOMERS                             Col 1 of 9     <<MORE
Combine All Column Criteria by ==> A   (A-AND, O-OR)

Cmd     Column Name                        Selection Criteria
---  -----------------  ---------------------------------------------------
***  ******************************** TOP ***********************************
___  CUST_ID
___  CUSTNAME
___  ADDRESS
___  CITY
___  STATE              ='MA'
___  ZIP
___  YTD_SALES
___  SALESMAN_ID
___  PHONE_NUMBER
***  ******************************* BOTTOM **********************************
```

Figure 5. Selecting Customers in Massachusetts

Combining Criteria

If desired, you can use selection criteria for more than one column in a table and combine the criteria using a logical AND or OR. For example, you might add criteria to select rows for customers whose names begin with M. Use the default setting, A (for AND), at the **Combine All Column Criteria by** prompt to obtain rows for all customers in Massachusetts whose names begin with M. Alternatively, specify O (for OR) to obtain rows for all customers in Massachusetts and rows for all customers, regardless of location, whose names begin with M.

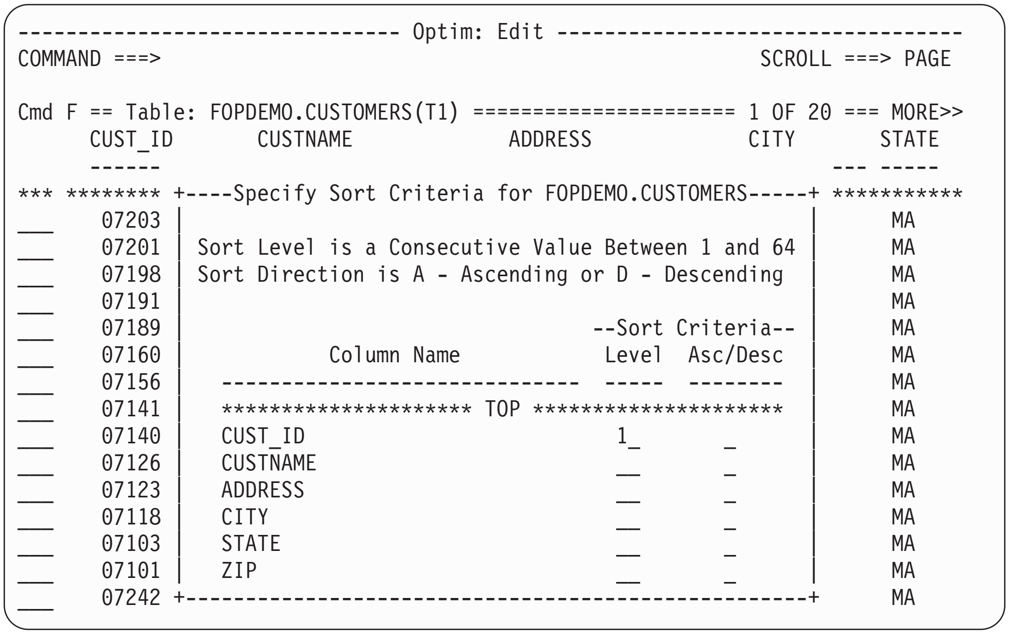
COL Command

To customize the display and review information about columns in the table, you can scroll the panel or use the COLUMNS command (or COL) to display the Describe Columns for Table panel. Details about this panel are discussed in the *Common Elements Manual*, section for Manage Data Displays.

Use the END command to return to the editor. The displayed data reflects the selection criteria.

Sort Criteria

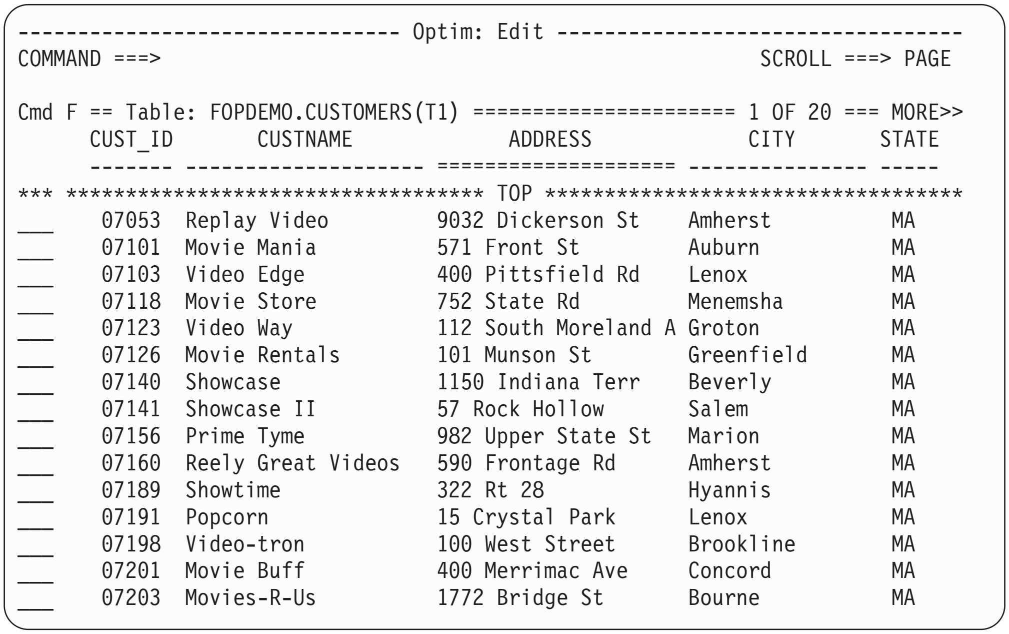
The rows are not displayed in any particular order unless you use sort criteria to organize the data. To arrange the rows, type the SORT CRITERIA command (or SOR) at the command prompt and press ENTER to display the **Specify Sort Criteria** pop-up window. This pop-up window lists all columns in the table and prompts for sort criteria. Current® sort criteria, if any, are also displayed. In this sample session, you will sort the rows from the CUSTOMERS table by customer ID values, as shown in Figure 6 on page 12.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------                                                        --- -----
*** ******** +----Specify Sort Criteria for FOPDEMO.CUSTOMERS-----+ ***********
___    07203 |                                                    |      MA
___    07201 | Sort Level is a Consecutive Value Between 1 and 64 |      MA
___    07198 | Sort Direction is A - Ascending or D - Descending  |      MA
___    07191 |                                                    |      MA
___    07189 |                                  --Sort Criteria-- |      MA
___    07160 |           Column Name             Level  Asc/Desc  |      MA
___    07156 |   ------------------------------  -----  --------  |      MA
___    07141 |   ********************* TOP *********************  |      MA
___    07140 |   CUST_ID                         1_        _      |      MA
___    07126 |   CUSTNAME                        __        _      |      MA
___    07123 |   ADDRESS                         __        _      |      MA
___    07118 |   CITY                            __        _      |      MA
___    07103 |   STATE                           __        _      |      MA
___    07101 |   ZIP                             __        _      |      MA
___    07242 +----------------------------------------------------+      MA
```

Figure 6. Specifying Sort Criteria

Use **Level** to assign a sort order priority to a column. You can assign priority levels for up to 64 columns. Priority level 1 is the highest and is sorted first. In this example, specify **Level** 1 for the CUST_ID column. Use **Asc/Desc** to specify ascending or descending sort order. By default, rows are sorted in ascending order, so leave **Asc/Desc** blank in this example. Once specifications are complete, use END to return to the editor.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          571 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
___    07118  Movie Store          752 State Rd         Menemsha         MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07156  Prime Tyme           982 Upper State St   Marion           MA
___    07160  Reely Great Videos   590 Frontage Rd      Amherst          MA
___    07189  Showtime             322 Rt 28            Hyannis          MA
___    07191  Popcorn              15 Crystal Park      Lenox            MA
___    07198  Video-tron           100 West Street      Brookline        MA
___    07201  Movie Buff           400 Merrimac Ave     Concord          MA
___    07203  Movies-R-Us          1772 Bridge St       Bourne           MA
```

Figure 7. Selected Rows are Sorted

Based on the criteria, customers from Massachusetts are displayed and the rows are sorted in ascending order by customer ID.

COUNT Command

When the number of rows that satisfy the criteria exceeds the **Maximum Fetch Rows** value, established as an Editor and Display Option, a message indicates that all possible rows have not been retrieved. You can use the COUNT command to determine the total number of rows in a named table that satisfies the search criteria. You can change the selection criteria to retrieve fewer rows.

MAX ROWS Command

If you prefer to display all rows that satisfy the current search criteria, you can use the MAX ROWS command. For example, if the **Maximum Fetch Rows** value is 200, and the COUNT command shows that 300 rows satisfy the criteria, use the MAX ROWS command to display the 300 rows:

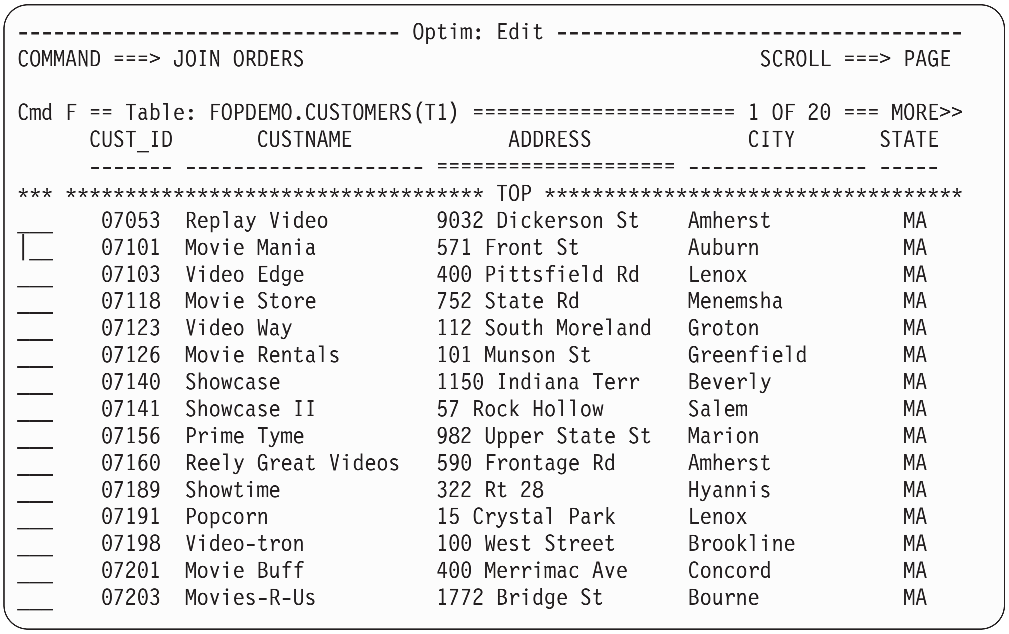
MAX ROWS 300

This command affects only the current session. To increase the maximum number of fetch rows for all of your sessions, use the **Editor and Display Options panel**, which is described in the *Common Elements Manual*, section for Editor and Display Options.

Displaying Rows From A Joined Table

The related data from another table can be displayed using the Join facility. When you know the name of the table, you can explicitly enter the name with the JOIN command.

The JOIN command in the following figure obtains rows from the ORDERS table that are related to a row in the CUSTOMERS table. A specific row from the CUSTOMERS table is indicated by the cursor location. For this sample session, type JOIN ORDERS at the command prompt, position the cursor on customer 07101, and press ENTER.



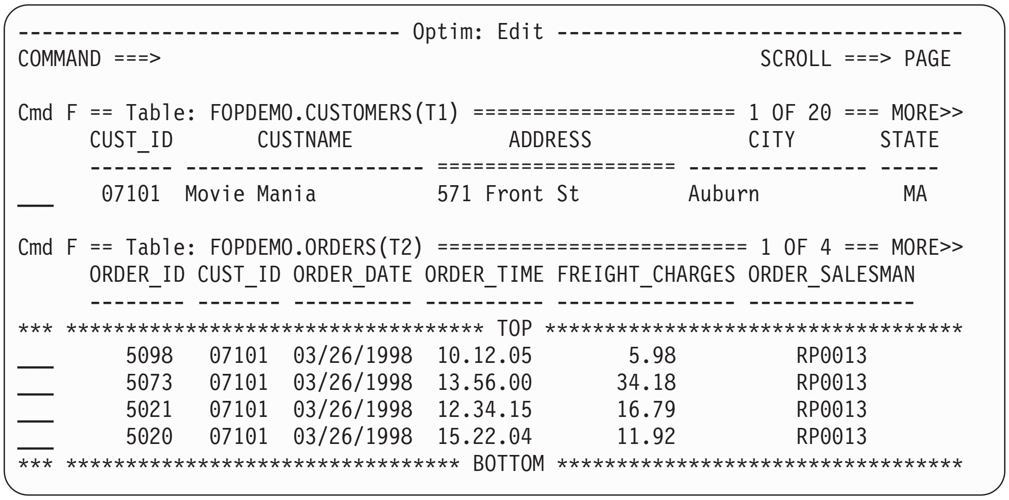
```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===> JOIN ORDERS                                      SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst           MA
|__    07101  Movie Mania          571 Front St         Auburn            MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox             MA
___    07118  Movie Store          752 State Rd         Menemsha          MA
___    07123  Video Way            112 South Moreland   Groton            MA
___    07126  Movie Rentals        101 Munson St        Greenfield        MA
___    07140  Showcase             1150 Indiana Terr    Beverly           MA
___    07141  Showcase II          57 Rock Hollow       Salem             MA
___    07156  Prime Tyme           982 Upper State St   Marion            MA
___    07160  Reely Great Videos   590 Frontage Rd      Amherst           MA
___    07189  Showtime             322 Rt 28            Hyannis           MA
___    07191  Popcorn              15 Crystal Park      Lenox             MA
___    07198  Video-tron           100 West Street      Brookline         MA
___    07201  Movie Buff           400 Merrimac Ave     Concord           MA
___    07203  Movies-R-Us          1772 Bridge St       Bourne            MA
```

Figure 8. Join Request

The orders for the selected customer are retrieved based on the relationship defined for the CUSTOMERS and ORDERS tables.

In this example, the relationship is defined in the DB2 Catalog. The tables are related by values in the column CUST_ID, which is the primary key for the CUSTOMERS table and the foreign key for the ORDERS table.

Access automatically displays the related rows from the ORDERS table with the selected row from the original CUSTOMERS table.

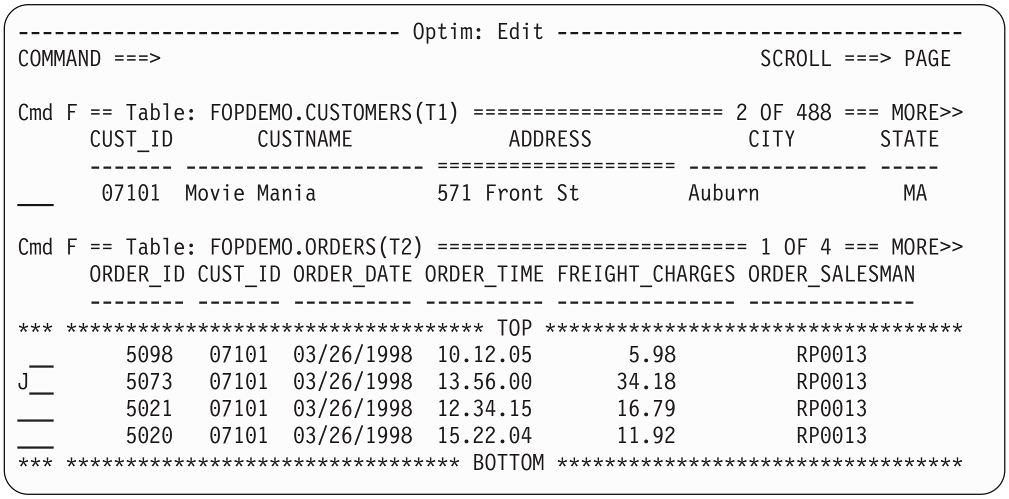


```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07101  Movie Mania          571 Front St         Auburn           MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP **********************************
___       5098   07101  03/26/1998  10.12.05       5.98          RP0013
___       5073   07101  03/26/1998  13.56.00      34.18          RP0013
___       5021   07101  03/26/1998  12.34.15      16.79          RP0013
___       5020   07101  03/26/1998  15.22.04      11.92          RP0013
*** ********************************* BOTTOM *********************************
```

Figure 9. Related Orders

Selection List of Related Tables

Rows from any related table can be displayed even if you are not sure of the exact name of the desired table. When you enter the JOIN command with no operands, or use the Join line command by entering J on a specific row, Access provides a selection list. As in the following figure, enter the J line command for the order 5073.



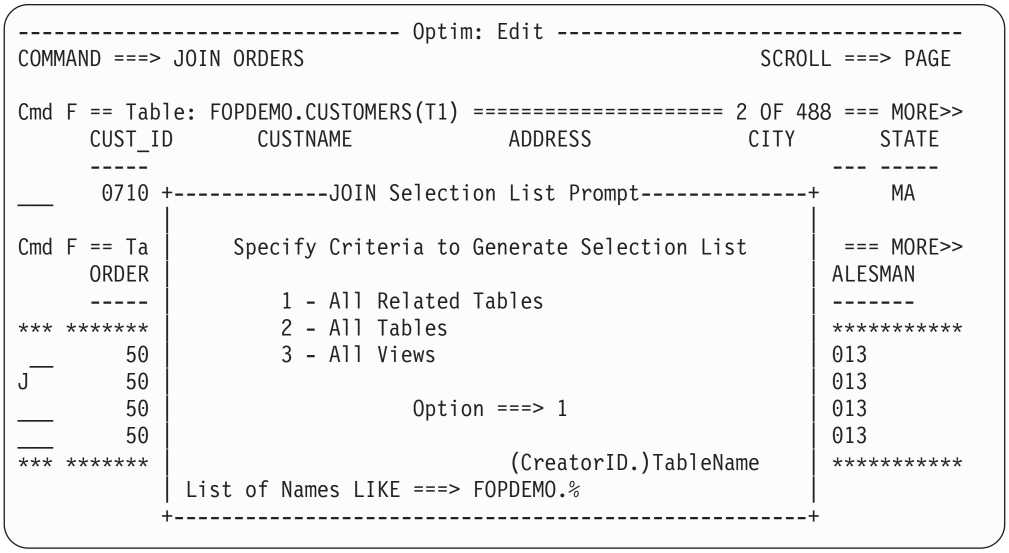
```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 2 OF 488 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07101  Movie Mania          571 Front St         Auburn           MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP **********************************
 __       5098   07101  03/26/1998  10.12.05       5.98          RP0013
J__       5073   07101  03/26/1998  13.56.00      34.18          RP0013
___       5021   07101  03/26/1998  12.34.15      16.79          RP0013
___       5020   07101  03/26/1998  15.22.04      11.92          RP0013
*** ********************************* BOTTOM *********************************
```

Figure 10. Joining to the ORDERS Table

Request Selection List

Access prompts for instructions regarding the contents of the selection list, which may include all related tables, all tables, or all views. The list excludes tables already joined. Also, you can limit the names on the selection list by specifying DB2 LIKE syntax in **List of Names LIKE**.

In Figure 11, Access is instructed to check the database for all tables with the Creator ID FOPDEMO that are related to the ORDERS table. The DB2 LIKE character % is used. (For more information, see DB2 LIKE Syntax in the *Common Elements Manual*.)

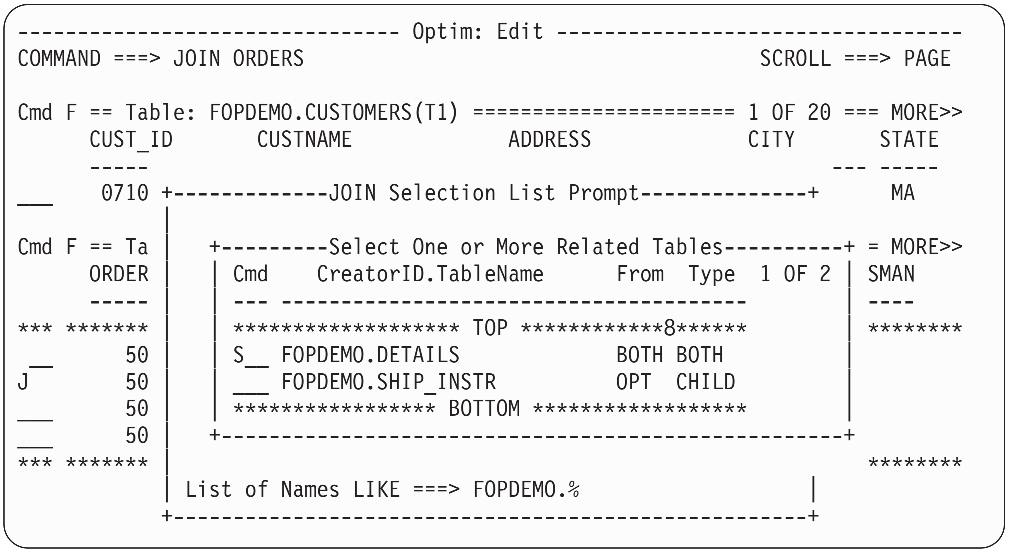


```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===> JOIN ORDERS                                      SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 2 OF 488 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      -----                                                         --- -----
___    0710 +-------------JOIN Selection List Prompt--------------+      MA
            |                                                     |
Cmd F == Ta |     Specify Criteria to Generate Selection List     |  === MORE>>
      ORDER |                                                     | ALESMAN
      ----- |         1 - All Related Tables                      | -------
*** ******* |         2 - All Tables                              | ***********
 __      50 |         3 - All Views                               | 013
J        50 |                                                     | 013
___      50 |                    Option ===> 1                    | 013
___      50 |                                                     | 013
*** ******* |                            (CreatorID.)TableName    | ***********
            | List of Names LIKE ===> FOPDEMO.%                   |
            +-----------------------------------------------------+
```

Figure 11. JOIN Selection List Prompt

Select Table

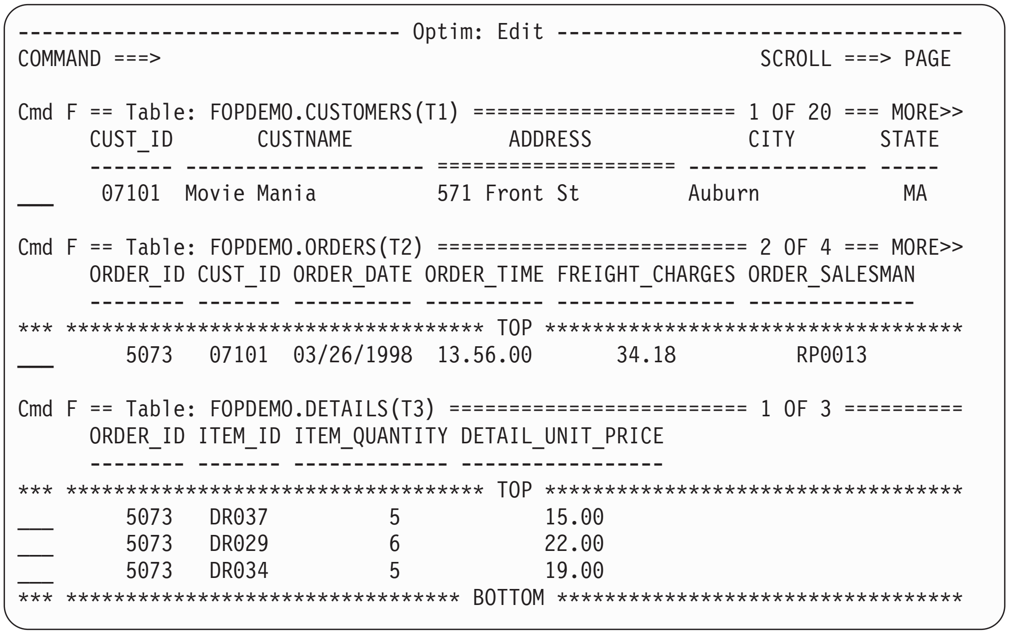
If only one table or view matches the name qualification, it is automatically joined to the display. However, for this example, assume that two tables in the database are related to the ORDERS table. Since either table will satisfy the join request, Access presents a selection list from which you can select the desired table.



```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===> JOIN ORDERS                                      SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      -----                                                         --- -----
___    0710 +-------------JOIN Selection List Prompt--------------+      MA
            |
Cmd F == Ta |   +---------Select One or More Related Tables----------+ = MORE>>
      ORDER |   | Cmd    CreatorID.TableName      From  Type  1 OF 2 | SMAN
      ----- |   | --- ---------------------------------------        | ----
*** ******* |   | ******************* TOP ************8******        | ********
 __      50 |   | S__ FOPDEMO.DETAILS             BOTH BOTH          |
J        50 |   | ___ FOPDEMO.SHIP_INSTR          OPT  CHILD         |
___      50 |   | ***************** BOTTOM ******************        |
___      50 |   +----------------------------------------------------+
*** ******* |                                                          ********
            | List of Names LIKE ===> FOPDEMO.%                   |
            +-----------------------------------------------------+
```

Figure 12. Select Table to Join

To select from this list, type an S next to the desired table names. In this example, select the DETAILS table only. Based on the specifications in the DB2 Catalog, the DETAILS table is joined and displayed. The related key columns are labeled ORDER_ID in both the ORDERS table and the DETAILS table.



```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID      CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07101  Movie Mania          571 Front St         Auburn           MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 2 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP **********************************
___       5073   07101  03/26/1998  13.56.00       34.18          RP0013

Cmd F == Table: FOPDEMO.DETAILS(T3) ======================== 1 OF 3 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP **********************************
___       5073   DR037          5            15.00
___       5073   DR029          6            22.00
___       5073   DR034          5            19.00
*** ********************************* BOTTOM *********************************
```

Figure 13. DETAILS Table Displayed

Display Levels

Throughout this manual, the term “display level” indicates the hierarchy in which tables are joined and displayed.

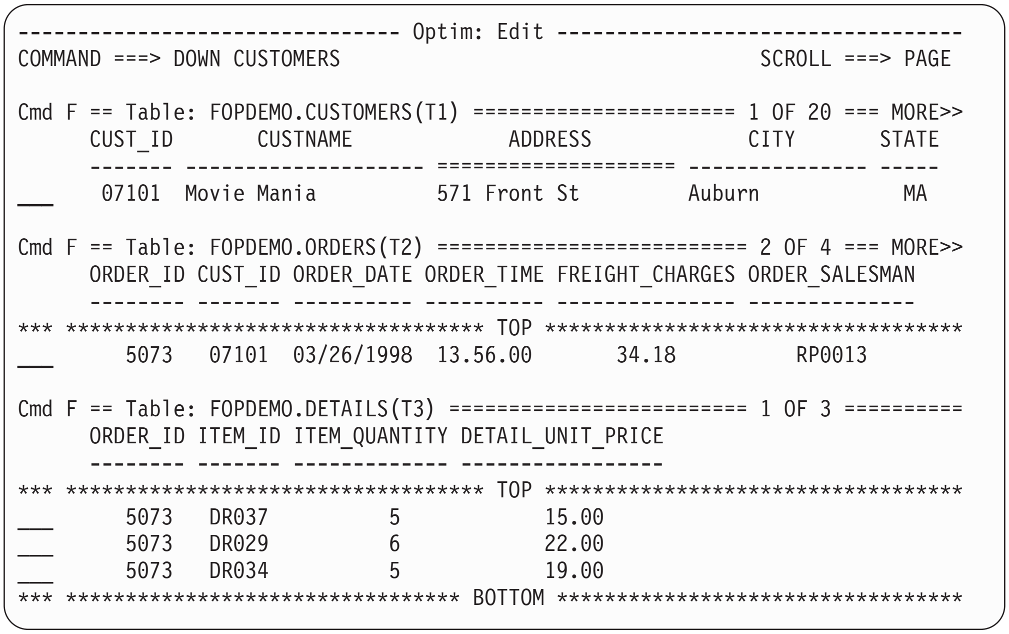
The first displayed table, or start table, is the first table that is used to select and process data. Each joined table is at a subordinate level to the start table. In Figure 13, the CUSTOMERS table is the start table, the ORDERS table is subordinate to the CUSTOMERS table, and the DETAILS table is subordinate to the ORDERS table.

To reflect the table level, Access assigns short names that are displayed in parentheses next to the table name. In this example, the CUSTOMERS table is T1, the ORDERS table is T2, and the DETAILS table is T3. These short names can be used in place of the table name as a command operand.

Scrolling the Tables

When any table is scrolled, tables at subordinate levels are also scrolled to show the data related to the newly displayed row. For example, when you scroll the CUSTOMERS table to display the next row, the ORDERS table scrolls to display the first related order and the DETAILS table scrolls to display the details related to that order. Similarly, if you then scroll the ORDERS table, the DETAILS table scrolls to display the details for the currently displayed ORDERS row.

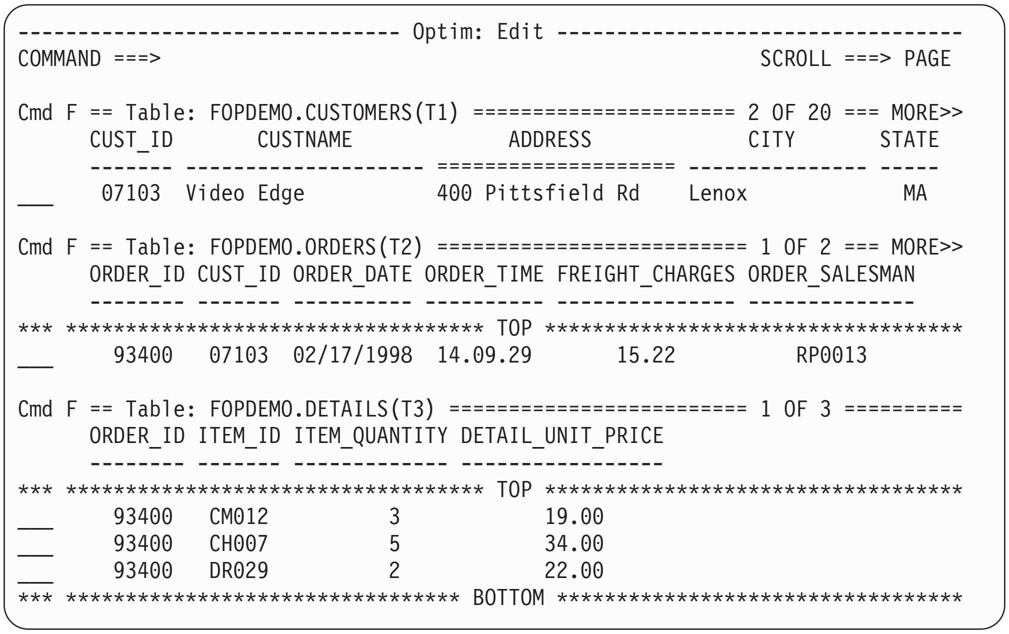
All familiar ISPF commands are available for scrolling. You can indicate the table to scroll by specifying the table name or the Access-assigned short name on the command line, or by positioning the cursor. For example, enter the DOWN command as DOWN T1 or DOWN CUSTOMERS to scroll the CUSTOMERS table. In the following figure, the table name is supplied with the DOWN command.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===> DOWN CUSTOMERS                                   SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07101  Movie Mania          571 Front St         Auburn           MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 2 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP ***********************************
___      5073   07101  03/26/1998  13.56.00       34.18          RP0013

Cmd F == Table: FOPDEMO.DETAILS(T3) ======================== 1 OF 3 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP ***********************************
___      5073   DR037         5            15.00
___      5073   DR029         6            22.00
___      5073   DR034         5            19.00
*** ********************************* BOTTOM **********************************
```

Figure 14. Scroll Request Entered

The scrolled display includes the next customer, the first related order for that customer, and the details for that order.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 2 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 2 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP ***********************************
___     93400   07103  02/17/1998  14.09.29       15.22          RP0013

Cmd F == Table: FOPDEMO.DETAILS(T3) ======================== 1 OF 3 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP ***********************************
___     93400   CM012         3            19.00
___     93400   CH007         5            34.00
___     93400   DR029         2            22.00
*** ********************************* BOTTOM **********************************
```

Figure 15. Coordinated Scrolling

Editing the Data

Any displayed data can be edited by directly typing values or using the editing commands. The commands include primary editing commands and the line commands for insert, delete, copy, and repeat. The ability to edit data is controlled by DB2-assigned security privileges. All Referential Integrity rules are respected. (See Chapter 3, “Editing and Browsing DB2 Data,” on page 19 for details on editing.)

Committing Changes

Changes to data are committed to the database with the SQL COMMIT statements. AUTOCOMMIT is in effect by default, according to an option in Editor and Display Options, and a COMMIT statement is executed for every screen interaction. However, if AUTOCOMMIT is not in effect, you must use the Access COMMIT command to explicitly invoke a COMMIT statement.

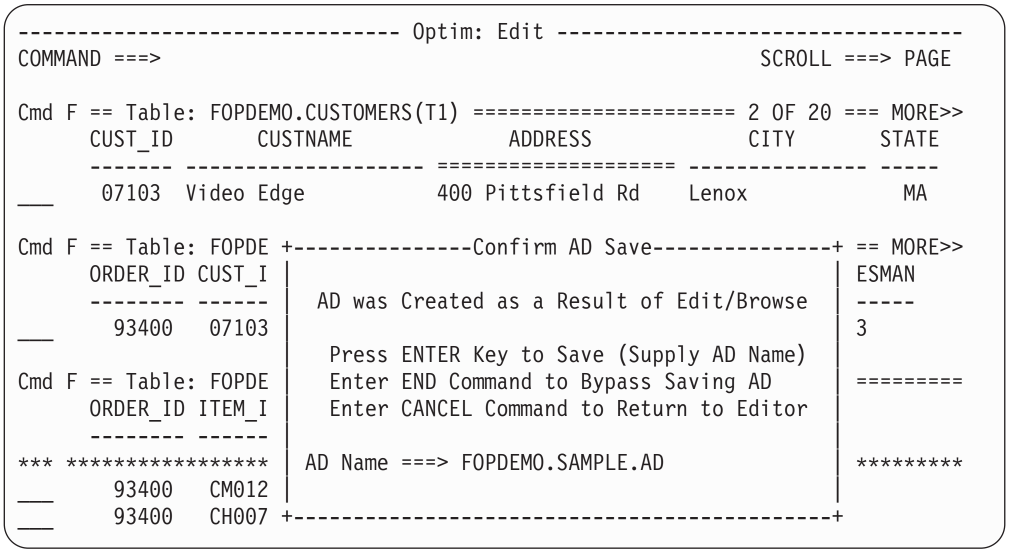
In addition to supporting DB2 ROLLBACK, an Access UNDO capability restores the data in a single row, a block of rows, or all rows independent of commit points. UNDO is also used to restore data when an error condition arises.

Saving the Access Definition

Use the END command when you are finished editing.

If you specified Yes to the **Prompt to Create AD on Exit** option on the Choose a DB2 Table/View to Edit panel, Access prompts you to determine whether to save the specifications from the current edit session as an Access Definition. If you specify No for this option and later decide that you would like to save the specifications from this session as an Access Definition, use the CREATE AD YES command to request a prompt.

When the **Confirm AD Save** prompt is displayed, type the three-part name in the form *user,group,name*, and press ENTER. In the following figure, the Access Definition is to be saved as FOPDEMO.SAMPLE.AD. (If an Access Definition with the specified name already exists, you are prompted to replace the existing Access Definition with the current session specifications.) As instructed on the following confirmation prompt, use ENTER to save the Access Definition.



```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 2 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07103  Video Edge           400 Pittsfield Rd    Lenox             MA

Cmd F == Table: FOPDE +---------------Confirm AD Save---------------+ == MORE>>
      ORDER_ID CUST_I |                                             | ESMAN
      -------- ------ |  AD was Created as a Result of Edit/Browse  | -----
___     93400   07103 |                                             | 3
                      |   Press ENTER Key to Save (Supply AD Name)  |
Cmd F == Table: FOPDE |   Enter END Command to Bypass Saving AD     | =========
      ORDER_ID ITEM_I |   Enter CANCEL Command to Return to Editor  |
      -------- ------ |                                             |
*** ***************** | AD Name ===> FOPDEMO.SAMPLE.AD              | *********
___     93400   CM012 |                                             |
___     93400   CH007 +---------------------------------------------+
```

Figure 16. Confirmation Prompt

The saved Access Definition includes the names of tables that have been accessed, any specified sort and selection criteria, and any specified column attributes. Once stored, multiple users can use the Access Definition repeatedly to browse or edit the same set of data.

Chapter 3. Editing and Browsing DB2 Data

The section explains methods used to initiate an Access edit or browse session, and discusses editor basics such as panel layout and available display modes. Detailed information on joining related data in other tables, handling multiple tables, editing data, and committing and undoing changes to the data is also provided.

Initiating Edit/Browse

You can initiate an edit or browse session using one of several methods.

- From the Main Menu

  You can directly invoke an edit or browse session using one of four Main Menu options:

  1 Browse a DB2 table

  2 Edit a DB2 table

  3 Browse DB2 tables using an Access Definition

  4 Edit DB2 tables using an Access Definition

- While editing an Access Definition

  You can use the current Access Definition to edit or browse data by entering the EDIT or BROWSE command from any panel used to define the Access Definition.

- Recursively, while editing or browsing data

  You can nest edit or browse sessions by using the EDIT or BROWSE command with a table name or an Access Definition name operand. For example, to edit the data in table FOPDEMO.CUSTOMERS, enter:

  ```
  EDIT TABLE FOPDEMO.CUSTOMERS
  ```

  Use DB2 LIKE syntax to display a selection list of tables or Access Definitions. Omit the Creator ID to use the default Creator ID.

- By using a CLIST

  You can use a CLIST to invoke an edit or browse session. For more information, see the *Customization Guide*.

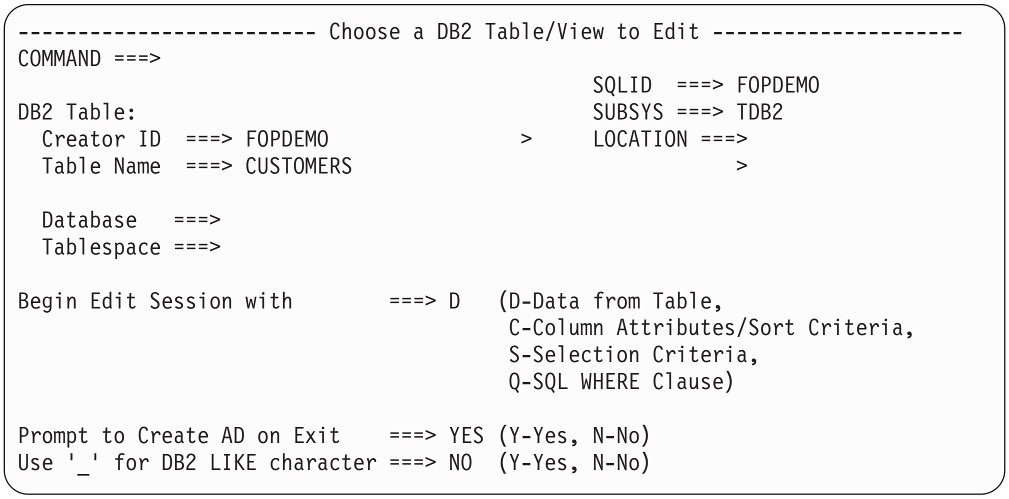
- While using the BMC Catalog Manager

  You can invoke an edit or browse session from any list of tables or views displayed by the Catalog Manager. Line commands are available to select the desired table or view. See Appendix D. Interfacing to BMC Catalog Manager for more information on the interface.

Choosing a Table or View

When you select Option 2 EDIT on the **Main Menu**, the **Choose a DB2 Table/View to Edit** panel prompts for the name of a table for the edit session.

You specify the table name by supplying the **Creator ID** and the **Table Name** as shown in the following figure. In this example, the table is named FOPDEMO.CUSTOMERS.



```
------------------------ Choose a DB2 Table/View to Edit ---------------------
COMMAND ===>
                                                SQLID  ===> FOPDEMO
DB2 Table:                                      SUBSYS ===> TDB2
  Creator ID  ===> FOPDEMO                >     LOCATION ===>
  Table Name  ===> CUSTOMERS                                >

  Database   ===>
  Tablespace ===>

Begin Edit Session with        ===> D   (D-Data from Table,
                                         C-Column Attributes/Sort Criteria,
                                         S-Selection Criteria,
                                         Q-SQL WHERE Clause)

Prompt to Create AD on Exit    ===> YES (Y-Yes, N-No)
Use '_' for DB2 LIKE character ===> NO  (Y-Yes, N-No)
```

Figure 17. Choose a DB2 Table/View to Edit

Change DB2 Access

The following prompts allow you to change the SQLID, subsystem, or location with which you access the database table.

SQLID
The current SQLID. Modify this value to connect using a different SQLID.

SUBSYS
The current DB2 subsystem. Modify this value to connect to a different DB2 subsystem.

When connecting to a remote subsystem, this value should be the local subsystem where the remote location is defined.

LOCATION
The remote location. This prompt is displayed if remote access is available. Specify a value to connect to a remote DB2 subsystem. You can use a percent sign (%) to obtain a selection list of available locations.

Note: If you leave this prompt blank, the local subsystem is assumed.

Database and Tablespace

The **Database** and **Tablespace** prompts on the Choose a DB2 Table/View to Edit panel allow you to further qualify the list of tables. Any value you specify is profiled.

Begin Edit Session with

You may want to limit the amount of data displayed during an edit or browse session or to customize the display. Use the **Begin Edit Session with** prompt to set up the initial display for the session. By default, the prompt is set to D to display all data from the specified table. If desired, you can change this to:

C Display a prompt for column attributes and sort criteria. At the start of the session, you can indicate the columns to be displayed, the order in which rows will be sorted, and choose column names or labels for the headings. This prompt is discussed in the *Common Elements Manual*, section for Manage Data Displays.

S Display a prompt for selection criteria. At the start of the session, you can specify valid SQL predicates used as selection criteria for one or more columns. Criteria can be ANDed or ORed. This prompt is discussed in the *Common Elements Manual*, section for Selection Criteria.

Q Display a prompt for an SQL WHERE clause. At the start of the session, you can specify a full WHERE clause for complex selection criteria (for example, to combine AND and OR logical operators). This prompt is discussed in the *Common Elements Manual,* section for SQL WHERE Clause Specifications.

Regardless of the setting you choose for beginning your edit or browse session, you can use the COL, SEL, or SQL commands during a session to display these prompts.

Dynamic Access Definitions

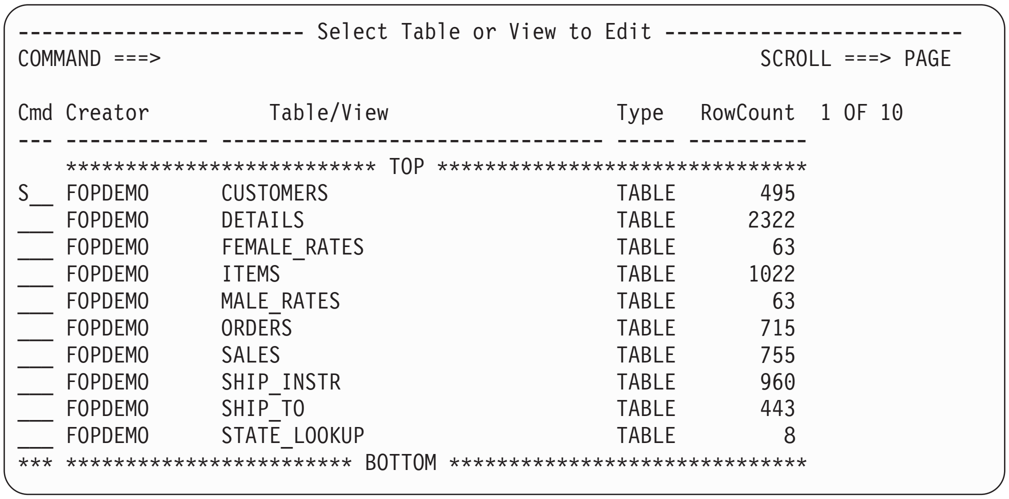
When you begin an edit or browse session, Access dynamically creates an Access Definition that reflects your editing or browsing activity. For example, if you specify column attributes or selection criteria during an edit session, they are incorporated into the Access Definition. If you join to other tables, those table names are added to the Access Definition Table List.

You can save or discard this dynamically created Access Definition at the end of your session. In order to save the Access Definition for future use, you must specify Y for the **Prompt to Create AD on Exit** option. At the end of the session, Access prompts you to name the Access Definition or to discard it. If **Prompt to Create AD on Exit** is N, the Access Definition is automatically discarded.

You can “recreate” the edit session that produced the Access Definition by selecting Option 3 BROWSE USING AD or Option 4 EDIT USING AD and reusing the saved Access Definition. For more information, see the *Common Elements Manual*, section for Access Definitions.

Selecting a Table from a Selection List

To obtain a selection list of tables, enter a partial table name using DB2 LIKE syntax or leave **Creator ID** or **Table Name** blank. For example, specifying only the Creator ID, FOPDEMO, on the previous Choose a DB2 Table/View to Edit panel invokes the following selection list.



```
------------------------ Select Table or View to Edit -------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd Creator          Table/View                   Type   RowCount  1 OF 10
--- ------------ -------------------------------- ----- ----------
    ************************* TOP ******************************
S__ FOPDEMO      CUSTOMERS                        TABLE       495
___ FOPDEMO      DETAILS                          TABLE      2322
___ FOPDEMO      FEMALE_RATES                     TABLE        63
___ FOPDEMO      ITEMS                            TABLE      1022
___ FOPDEMO      MALE_RATES                       TABLE        63
___ FOPDEMO      ORDERS                           TABLE       715
___ FOPDEMO      SALES                            TABLE       755
___ FOPDEMO      SHIP_INSTR                       TABLE       960
___ FOPDEMO      SHIP_TO                          TABLE       443
___ FOPDEMO      STATE_LOOKUP                     TABLE         8
*** *********************** BOTTOM *****************************
```

Figure 18. Select Table or View to Edit

The selection list includes the Creator ID, name, type (TABLE, S-MQT, U-MQT or VIEW), and row count. **RowCount** is the number of rows in the table as determined by the most recent execution of the IBM RUNSTATS utility for the table. If RUNSTATS has not been run for a table, **RowCount** is blank. **RowCount** is also blank for a view. Use the S line command to select a table.

Once you have selected a table or view from the selection list or explicitly named one on the Choose a Table/View to Edit panel, the session begins.

About this Section

The concepts and figures presented in the following sections focus on an edit session invoked using Option 2 EDIT on the **Main Menu**.

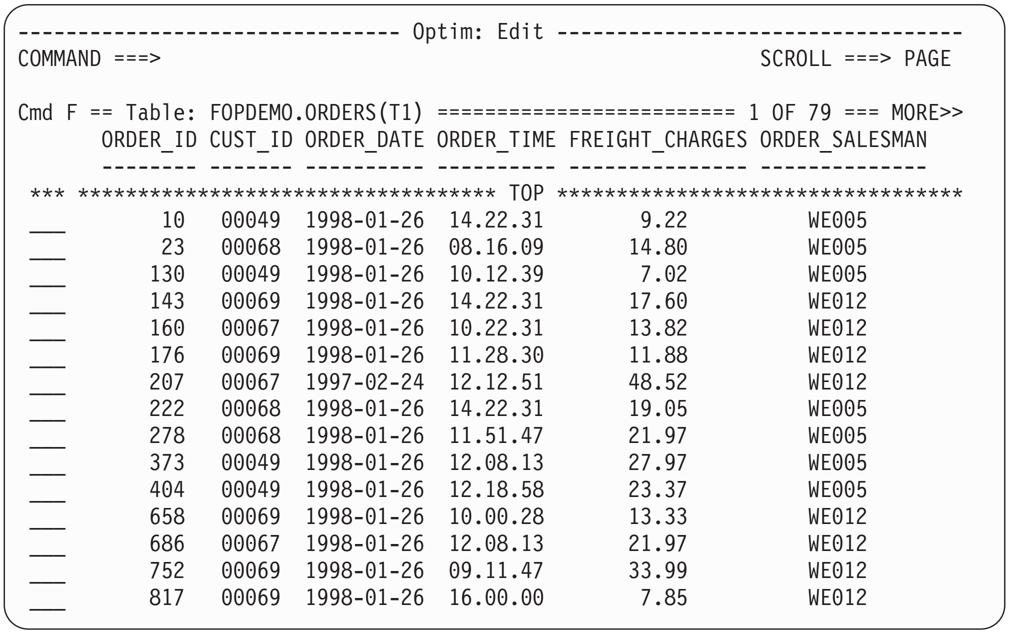
Any differences between the edit and browse facilities are noted. Also, features unique to the edit and browse facilities using an Access Definition (Options 3 and 4 on the **Main Menu**) are discussed in "Using an Explicit Access Definition" on page 41.

Editor Basics

An Access edit or browse session resembles an ISPF/PDF edit session. This section discusses the display modes, screen elements, and commands available during an Access edit or browse session.

Display Format

There are two basic display formats: columnar and sidelabels. Columnar format is the default. In columnar format, headings and data are displayed in vertical columns. The columnar format allows you to look at multiple rows in a table and join to related tables. The following figure shows data from the ORDERS table displayed in columnar format.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ======================== 1 OF 79 === MORE>>
       ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
       -------- ------- ---------- ---------- --------------- --------------
 *** ********************************** TOP *********************************
 ___        10   00049  1998-01-26  14.22.31        9.22          WE005
 ___        23   00068  1998-01-26  08.16.09       14.80          WE005
 ___       130   00049  1998-01-26  10.12.39        7.02          WE005
 ___       143   00069  1998-01-26  14.22.31       17.60          WE012
 ___       160   00067  1998-01-26  10.22.31       13.82          WE012
 ___       176   00069  1998-01-26  11.28.30       11.88          WE012
 ___       207   00067  1997-02-24  12.12.51       48.52          WE012
 ___       222   00068  1998-01-26  14.22.31       19.05          WE005
 ___       278   00068  1998-01-26  11.51.47       21.97          WE005
 ___       373   00049  1998-01-26  12.08.13       27.97          WE005
 ___       404   00049  1998-01-26  12.18.58       23.37          WE005
 ___       658   00069  1998-01-26  10.00.28       13.33          WE012
 ___       686   00067  1998-01-26  12.08.13       21.97          WE012
 ___       752   00069  1998-01-26  09.11.47       33.99          WE012
 ___       817   00069  1998-01-26  16.00.00        7.85          WE012
```

Figure 19. Columnar Screen Display

Screen Elements

The screen elements in the columnar and sidelabels display formats are the same; only the positioning is different. For more information about sidelabels format, see "Wide Data Displays" on page 55. The elements are:

Command
: Prompt for primary commands. See "Available Commands" on page 23 for a summary of available primary commands.

Line Cmd
: Area for line commands. The label is **Cmd** in columnar format and **LineCmd** in sidelabels format.

See “Available Commands” for a summary of available line commands.

Status Flag
Identifies the status of the row. The possible values are:

D Deleted row.

E Row contains an error.

I Inserted row.

U Updated row.

A blank indicates that the displayed data is unchanged from its original fetched state.

In columnar format, the label is **F**. In sidelabels format, the label is **Row Status**.

Table Name
Fully-qualified name of the table or view. The value is truncated if the Creator ID and Table Name are more than 22 characters. This value is inserted by ACCESS and cannot be edited.

Short Name
An identifier showing the level for each table (T*n*) or view (V*n*). In a multi-table display, T1 identifies the first level, T2, the second, etc. Several tables may be joined at any level except T1. Tables joined at the same level (i.e., stacked tables) have the same level identifier. (See “Special Considerations for Multi-way Joins” on page 36 for additional information about multi-way joins and stacked tables.) The level identifier can be substituted for the table name as a command operand.

x OF y
The position of the first data row on the screen (*x*) in relation to the total number of displayed rows (*y*) from the table.

Horizontal Scroll Indicator
Indicator, in the form **MORE** or **<<MORE**, that additional data can be displayed by scrolling LEFT or RIGHT. (See “Scroll” on page 47 for additional information about scrolling.)

Column Headings
By default, Access uses the DB2 column names as headings for the rows from each table. Alternatively, you can use the DB2 column labels as headings, as described in the *Common Elements Manual*, section for managing the data display.

Column Count
(Displayed in sidelabels format only.) Indicates the current position of the first column within the set of columns.

Available Commands

Both primary and line commands are available in an edit or browse session.

Many of these commands are introduced in the following sections. See the *Command Reference Manual*, section for Primary Commands for detailed information on these commands.

Primary Commands

The following primary commands are available for both editing and browsing. (The INDENT, JOIN, UNJOIN, START, and SWITCH commands are not available when using sidelabels format.)

- ANCHOR
- ATTRIBUTES
- BOTTOM
- BROWSE
- CANCEL

- CAPS
- COLUMNS
- COUNT
- CREATE AD
- DOWN
- EDIT
- END
- EXCLUDE
- EXPAND
- FIND
- HEX
- INDENT
- JOIN
- LEFT
- LIST
- LOCK
- MAX ROWS
- ONLY
- OPTIONS
- REPORT
- RESET
- RESORT
- RESTART
- RFIND
- RIGHT
- SEL CRIT
- SIDELABELS
- SHOW
- SHOW SQL
- SORT
- SQL
- START
- SWITCH
- TOP
- UNJOIN
- UNLOCK
- UP
- ZOOM

The following primary commands are available for editing only:

- AUTOCOMMIT
- CHANGE
- COMMIT
- DELETE
- GO

- HIDE
- RCHANGE
- RETRY
- ROLLBACK
- SAVE
- UNDO

Line Commands

The following line commands are available for both editing and browsing. (The J and UNJ line commands are not available in sidelabels format.)

Line Cmd	Function	Description
F, F*n*	First	Redisplay the first row, or the first *n* rows, in a block of excluded rows.
J	Join	Join related data in other tables.
L, L*n*	Last	Redisplay the last row, or the last *n* rows, in a block of excluded rows.
S, S*n*, SS	Show	Redisplay a row, *n* rows, or all rows in a block of excluded rows.
SID	Sidelabels	Display the row in sidelabels format, or return the display to columnar format.
UNJ	Unjoin	Remove joined tables from the display.
X, X*n*, XX	Exclude	Temporarily remove a row, *n* rows, or a block of rows from the display.
Z	Zoom	Display all rows in the table, or return to the multi-table display.

The following line commands are available for editing only:

Line Cmd	Function	Description
C, C*n*, CC	Copy	Copy a row, *n* rows, or all rows in a block of rows to a new location. Use A (after) or B (before) to indicate the destination.
D, D*n*, DD	Delete	Delete a row, *n* rows, or a block of rows.
I, I*n*	Insert	Insert a row or *n* rows for data entry after the specified row.
LC, LC*n*, LLC	Lowercase	Translate a row, *n* rows, or a block of rows to lowercase.
R, R*n*, RR*n*	Repeat	Duplicate a row or a block of rows once, or *n* times.
RP, RP*n*, RRP	Repeat pending	Duplicate a row or a block of rows once, or *n* times, and leave in insert pending status until you press ENTER.
U, U*n*, UU	Undo	Restore a row, *n* rows, or a block of rows to a previous state.
UC, UC*n*, UUC	Uppercase	Translate a row, *n* rows, or a block of rows to uppercase.

Join Tables

Both an edit and a browse session begin with a display of rows from a table specified on the Choose a DB2 Table/View panel.

Related rows from other tables may be added to the display using a Join command. Data in the display can be viewed using various navigation aids, allowing you to browse or edit a set of related data at one time.

Materialized Query Tables cannot be joined.

The table for which the join is executed, the *anchor table*, is joined to a related table and any related rows from that table are displayed. To perform a join, Access requires the name of the table to join and a relationship between the named table and the anchor table. The relationship can be a DB2 or an Optim relationship. If the relationship is not defined as either, Access prompts for the information to create an Optim relationship. The responses to those prompts are stored in the Directory and used as a supplement to the DB2 Catalog.

Note: This section focuses on joining tables and assumes the relationships have been defined. See the *Common Elements Manual*, section on Relationships for information on defining relationships.

When joining tables, it does not matter if the anchor table is the parent or the child table. You can join from the parent to the child or vice versa.

JOIN Command

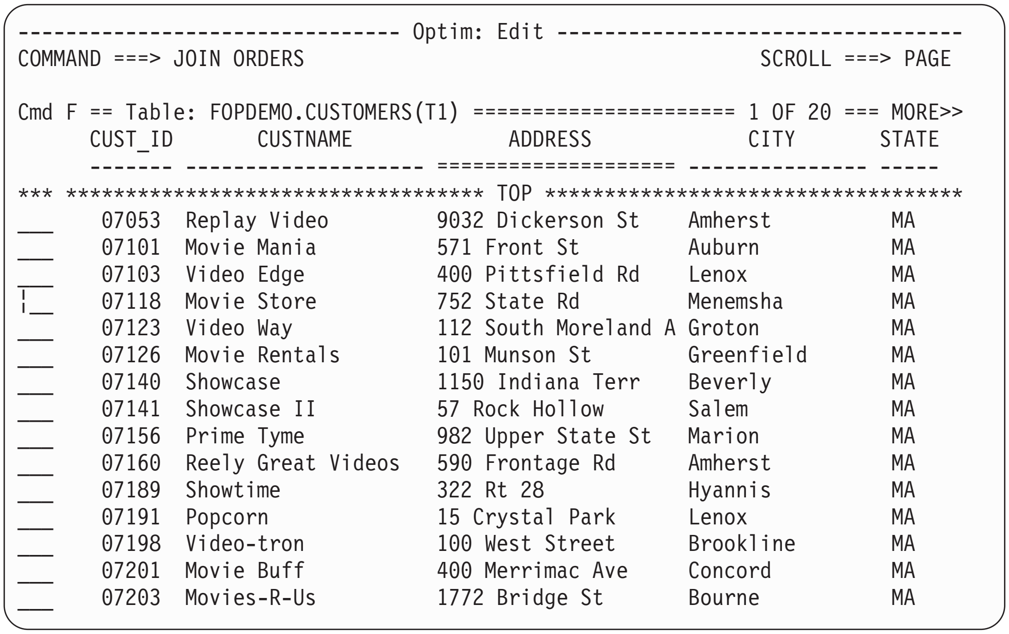
You can use the JOIN primary command or the J line command to display simultaneously, a row from the anchor table and related rows from the joined table.

By default, a JOIN primary command operates on the first displayed row in the table at the lowest level in the display. This table becomes the anchor table for that operation. To specify the anchor table, use the JOIN command with the FROM operand and anchor table name. For example, JOIN FROM ORDERS or JOIN FROM T1. Additionally, you can designate the anchor table and row in the anchor table by using the JOIN primary command and positioning the cursor on a row in the desired table, or by using the J line command. To specify the table to join to the anchor table, you can specify a table name with the JOIN command, or omit the table name to display a selection list.

One Related Table

When only one relationship exists between the anchor table and the table named with the JOIN command, the join occurs automatically.

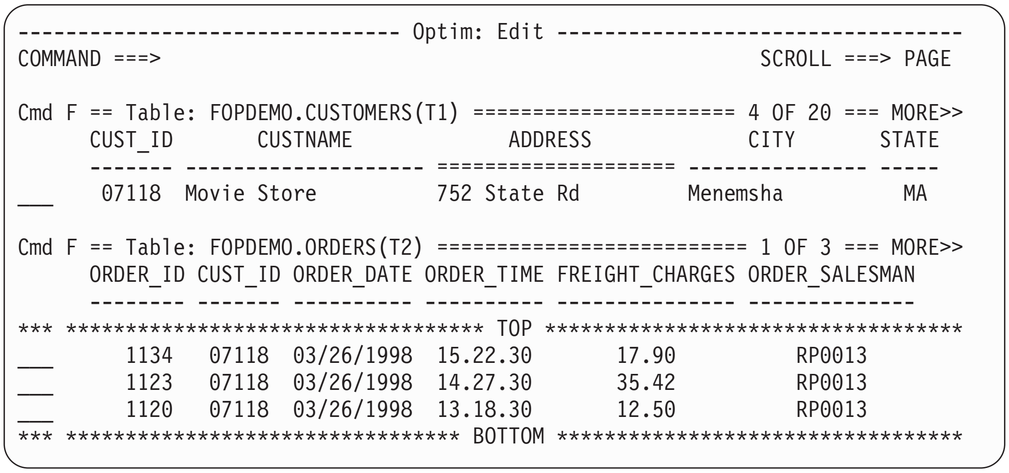
In the following figure, JOIN ORDERS is entered on the command line and the cursor is positioned on the row for CUST_ID 07118. In this example, only one relationship is defined between the CUSTOMERS and ORDERS tables. The CUST_ID column is the primary key in the CUSTOMERS table and the foreign key in the ORDERS table.



```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===> JOIN ORDERS                                      SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ************************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          571 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
¦__    07118  Movie Store          752 State Rd         Menemsha         MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07156  Prime Tyme           982 Upper State St   Marion           MA
___    07160  Reely Great Videos   590 Frontage Rd      Amherst          MA
___    07189  Showtime             322 Rt 28            Hyannis          MA
___    07191  Popcorn              15 Crystal Park      Lenox            MA
___    07198  Video-tron           100 West Street      Brookline        MA
___    07201  Movie Buff           400 Merrimac Ave     Concord          MA
___    07203  Movies-R-Us          1772 Bridge St       Bourne           MA
```

Figure 20. JOIN Command with One Related Table

When you press ENTER, ORDERS is joined to CUSTOMERS, as shown in the following figure. Since the join was specified for CUST_ID 07118, all rows in the ORDERS table with that CUST_ID are displayed.



```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 4 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    07118  Movie Store          752 State Rd         Menemsha          MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================== 1 OF 3 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- -------------- --------------
*** *********************************** TOP ************************************
___      1134   07118  03/26/1998  15.22.30       17.90          RP0013
___      1123   07118  03/26/1998  14.27.30       35.42          RP0013
___      1120   07118  03/26/1998  13.18.30       12.50          RP0013
*** ********************************* BOTTOM ***********************************
```

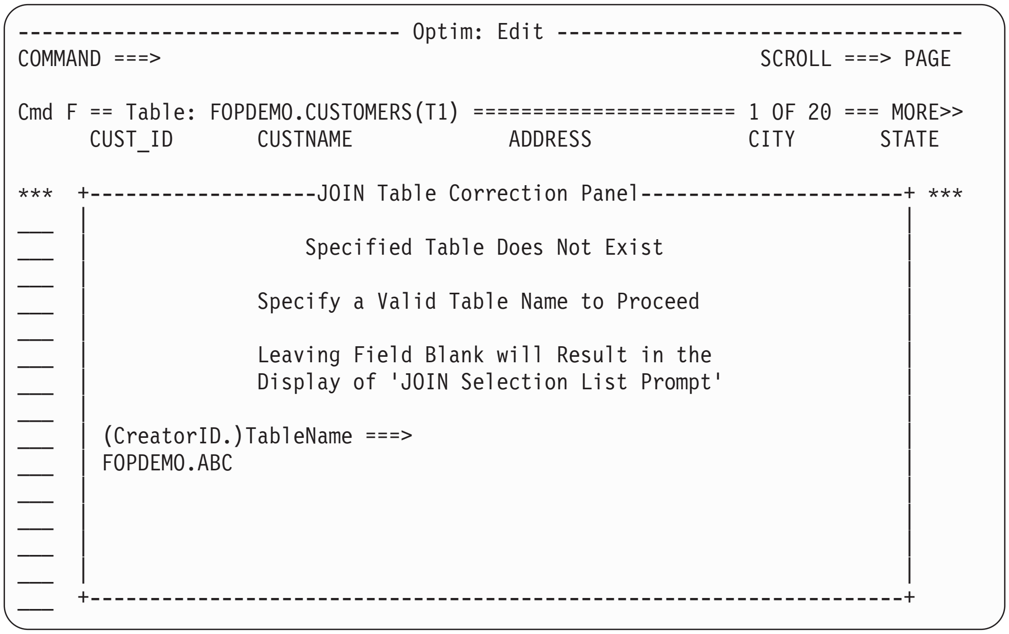
Figure 21. Result of Joining

This example uses cursor position to indicate the anchor row, or the row on which to perform the join. However, if the cursor is not positioned on a specific row, the anchor row is the first displayed row in the CUSTOMERS table.

See the *Command Reference Manual* for details on the JOIN command syntax.

Specified Table Does Not Exist

If you use the JOIN command with the name of a table that is not defined in the DB2 Catalog, Access displays the following pop-up window to allow you to correct the table name.



```
-------------------------------- Optim: Edit -----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE

***  +-------------------JOIN Table Correction Panel----------------------+ ***
___  |                                                                    |
___  |                  Specified Table Does Not Exist                    |
___  |                                                                    |
___  |              Specify a Valid Table Name to Proceed                 |
___  |                                                                    |
___  |              Leaving Field Blank will Result in the                |
___  |              Display of 'JOIN Selection List Prompt'               |
___  |                                                                    |
___  | (CreatorID.)TableName ===>                                         |
___  | FOPDEMO.ABC                                                        |
___  |                                                                    |
___  |                                                                    |
___  |                                                                    |
___  |                                                                    |
___  +--------------------------------------------------------------------+
___
```

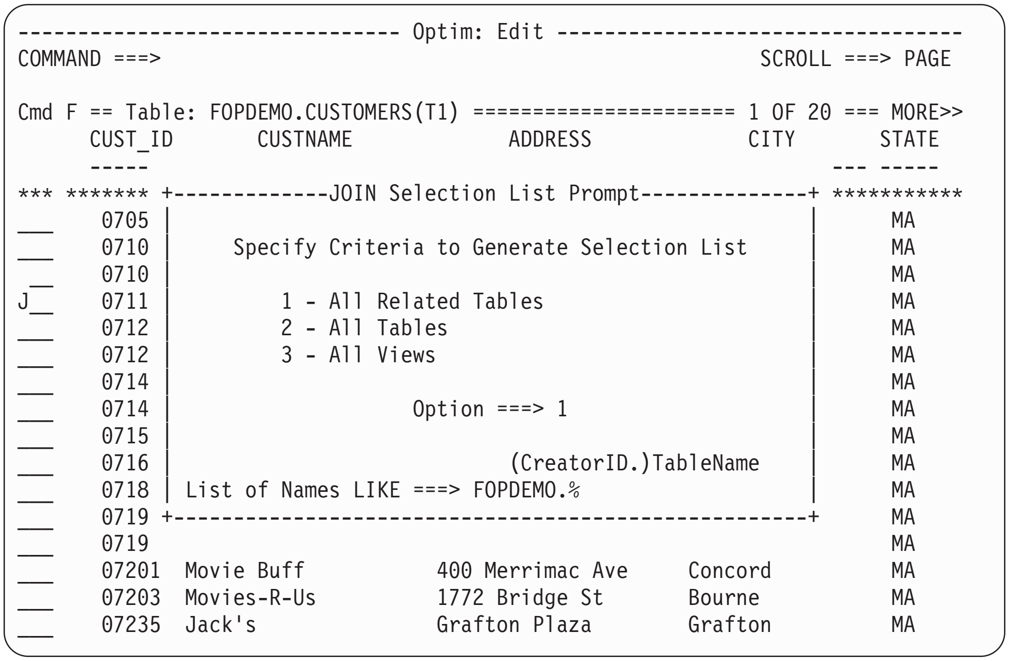
Figure 22. JOIN Table Correction Panel

The **(CreatorID.)TableName** prompt displays the invalid table name supplied with the JOIN command (see FOPDEMO.ABC). If the table name was spelled incorrectly, you can correct it by typing over the displayed name. If you omit the Creator ID, the Creator ID of the initial table is used.

Press ENTER to proceed or use END or CANCEL to cancel the join request.

No Table Name

If you do not specify a table name with the JOIN command, or if you use the J line command, Access displays a selection list. An example of the **JOIN Selection List Prompt** is shown in the following figure. In the example, the user displayed the prompt using the J line command.



```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS            CITY       STATE
      -----                                                          --- -----
*** ******* +-------------JOIN Selection List Prompt--------------+ ***********
___    0705 |                                                     |      MA
___    0710 |     Specify Criteria to Generate Selection List     |      MA
 __    0710 |                                                     |      MA
J__    0711 |         1 - All Related Tables                      |      MA
___    0712 |         2 - All Tables                              |      MA
___    0712 |         3 - All Views                               |      MA
___    0714 |                                                     |      MA
___    0714 |                    Option ===> 1                    |      MA
___    0715 |                                                     |      MA
___    0716 |                            (CreatorID.)TableName    |      MA
___    0718 | List of Names LIKE ===> FOPDEMO.%                   |      MA
___    0719 +-----------------------------------------------------+      MA
___    0719                                                              MA
___    07201  Movie Buff          400 Merrimac Ave     Concord           MA
___    07203  Movies-R-Us         1772 Bridge St       Bourne            MA
___    07235  Jack's              Grafton Plaza        Grafton           MA
```

Figure 23. JOIN Selection List Prompt

The options on this prompt are:

1 All Related Tables
Generates a selection list of tables that match the specified DB2 LIKE syntax and are related to the anchor table by either DB2 or Optim relationships. This is the default when browsing or editing without using an Access Definition (using Options 1 or 2 on the **Main Menu**).

2 All Tables
Generates a selection list of all tables in the database that match the specified DB2 LIKE syntax. This is the default when browsing or editing using an Access Definition (using Options 3 or 4 on the **Main Menu**).

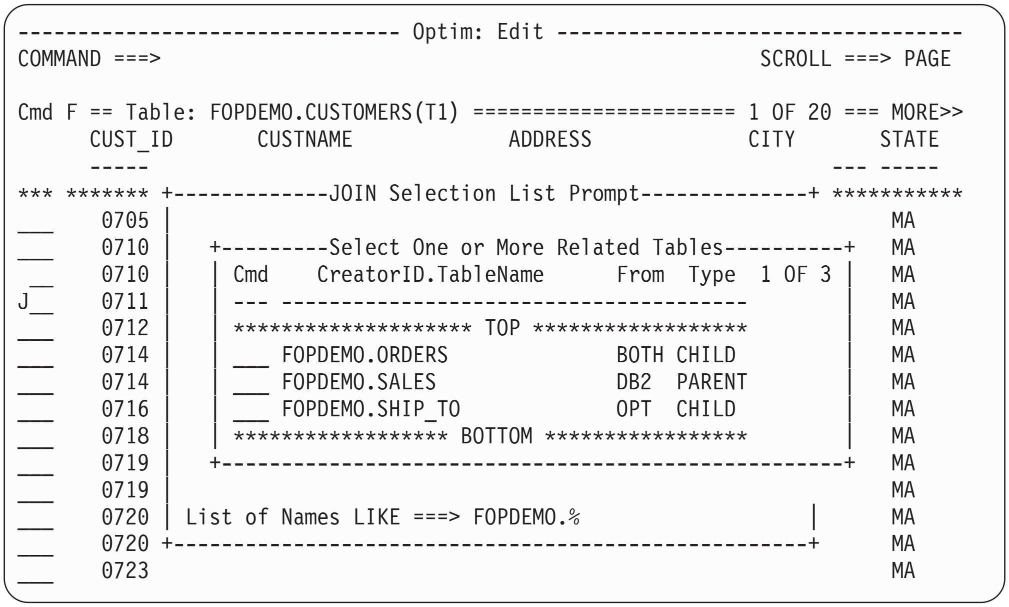
3 All Views
Generates a selection list of all views in the database that match the specified DB2 LIKE syntax.

Use the **List of Names LIKE** prompt to specify DB2 LIKE syntax to qualify the selection list and obtain a range of names. The Creator ID is optional. If omitted, the Creator ID of the initial table is assumed. Access supplies the Creator ID of the initial table and the DB2 LIKE syntax character %. This value may be overtyped. For example, specify PS% to display a list of tables with the default Creator ID and beginning with PS.

Selection List of Tables

If Option 1 is specified and several related tables match the specified DB2 LIKE syntax, Access displays a selection list. However, if only one table meets the criteria, Access joins and displays it without displaying a selection list.

In Figure 23, Option 1 is specified to generate a selection list of all related tables. When you press ENTER, the following selection list is displayed.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      -----                                                         --- -----
*** ******* +-------------JOIN Selection List Prompt--------------+ ***********
___    0705 |                                                            MA
___    0710 |   +---------Select One or More Related Tables----------+   MA
 __    0710 |   | Cmd    CreatorID.TableName      From  Type  1 OF 3 |   MA
J__    0711 |   | --- ---------------------------------------        |   MA
___    0712 |   | ******************** TOP ******************        |   MA
___    0714 |   | ___ FOPDEMO.ORDERS              BOTH CHILD         |   MA
___    0714 |   | ___ FOPDEMO.SALES               DB2  PARENT        |   MA
___    0716 |   | ___ FOPDEMO.SHIP_TO             OPT  CHILD         |   MA
___    0718 |   | ****************** BOTTOM *****************        |   MA
___    0719 |   +----------------------------------------------------+   MA
___    0719 |                                                            MA
___    0720 | List of Names LIKE ===> FOPDEMO.%                   |      MA
___    0720 +-----------------------------------------------------+      MA
___    0723                                                              MA
```

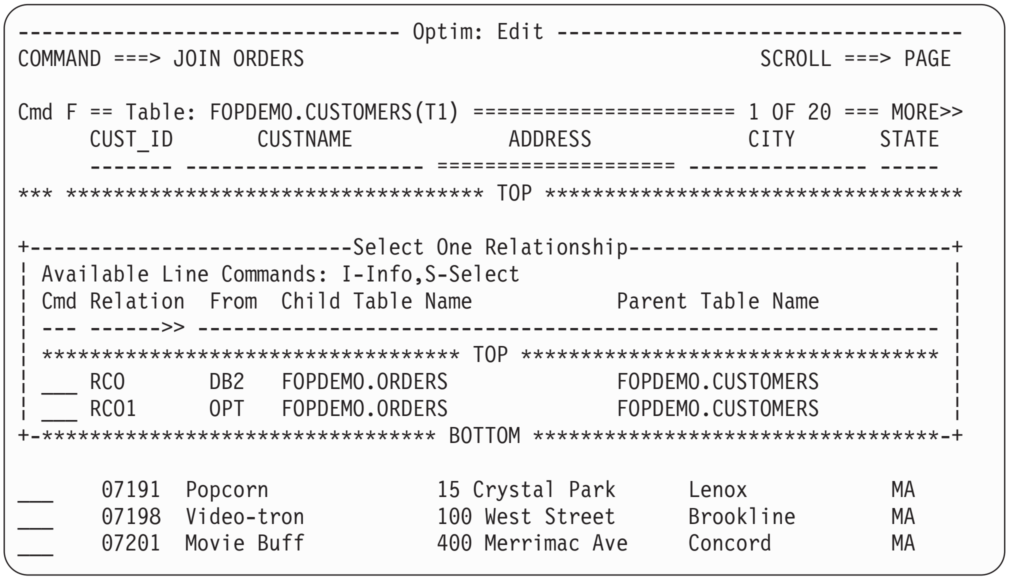
Figure 24. Select One or More Related Tables

Each line in the list indicates the table name and the relationship to the anchor table. The **From** value indicates whether the table is related to the anchor table by a DB2 or Optim relationship, or both. The **Type** value indicates whether the table is a child or parent of the anchor table. Use the S line command to select one or more tables from the list, and press ENTER or use END.

If only one relationship exists between the anchor table and the selected table (e.g., the ORDERS table), the tables are joined as shown in Figure 21 on page 27. If more than one relationship exists between the anchor table and the selected table, Access displays a selection list of relationships. For more information about this selection list, see “Select Relationships.”

Select Relationships

If there are two or more relationships between the anchor table and a selected table, Access displays a selection list of relationships, as in the following figure.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===> JOIN ORDERS                                      SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************

+----------------------------Select One Relationship--------------------------+
¦ Available Line Commands: I-Info,S-Select                                    ¦
¦ Cmd Relation  From  Child Table Name            Parent Table Name           ¦
¦ --- ------>> ---------------------------------------------------------------- ¦
¦ *********************************** TOP *********************************** ¦
¦ ___ RCO       DB2   FOPDEMO.ORDERS              FOPDEMO.CUSTOMERS            ¦
¦ ___ RCO1      OPT   FOPDEMO.ORDERS              FOPDEMO.CUSTOMERS            ¦
+-******************************** BOTTOM *********************************-+

___    07191  Popcorn             15 Crystal Park      Lenox            MA
___    07198  Video-tron          100 West Street      Brookline        MA
___    07201  Movie Buff          400 Merrimac Ave     Concord          MA
```

Figure 25. Select One Relationship

Panel Elements

The panel includes:

Cmd The line command entry area. Valid commands are:

- **I** Display the Browse Relationship panel, allowing you to view the names of the columns in the selected relationship.
- **S** Select the relationship to be used to join the anchor table and the selected table.

Relation
The name of the relationship.

From The type of relationship.

- **DB2** The relationship is defined to the DB2 Catalog.
- **OPT** The relationship is defined to the Optim Directory.

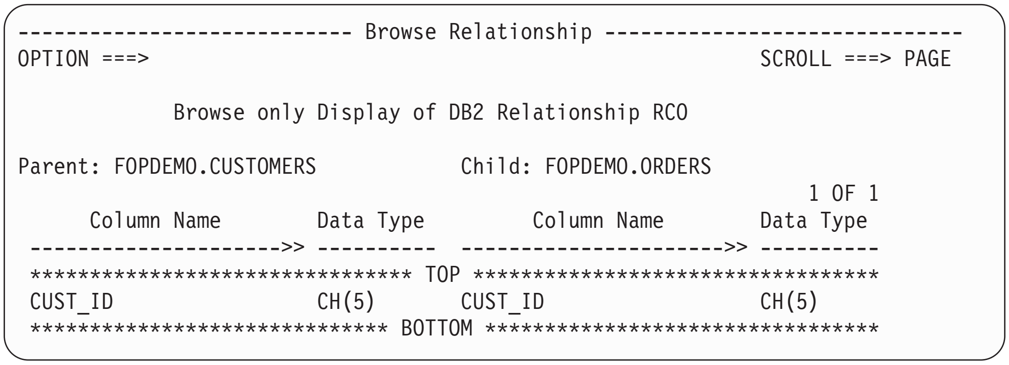
Child Table Name
The fully qualified name (Creator ID and Table Name) of the child table in the relationship.

Parent Table Name
The fully qualified name (Creator ID and Table Name) of the parent table in the relationship.

After you select a table, press ENTER or use END to display the joined tables.

Browse Relationship Panel

In the following figure, information about the relationship RCO is displayed in the panel.



```
----------------------------- Browse Relationship ------------------------------
OPTION ===>                                                    SCROLL ===> PAGE

             Browse only Display of DB2 Relationship RCO

Parent: FOPDEMO.CUSTOMERS            Child: FOPDEMO.ORDERS
                                                                   1 OF 1
      Column Name        Data Type         Column Name        Data Type
 --------------------->> ----------  ---------------------->> ----------
 ******************************** TOP **********************************
 CUST_ID                CH(5)       CUST_ID                  CH(5)
 ****************************** BOTTOM *********************************
```

Figure 26. Browse Relationship

Panel

The **Browse Relationship** panel includes:

Parent The fully qualified name of the parent table.

Child The fully qualified name of the child table.

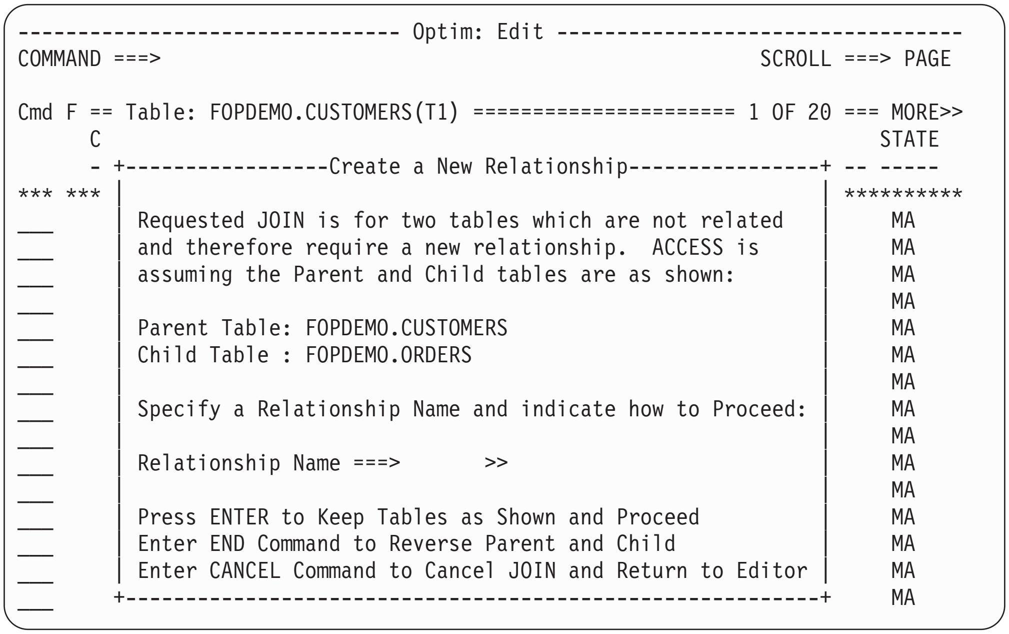
Column Name
The expressions and names of columns for each table that are used in the relationship.

Data Type
The data type of the matched columns. In addition to the standard DB2 data types, **Data Type** may be EXPR, to indicate that the relationship is based on an expression for which the data type is determined when the relationship is used.

Use END or press ENTER to return to the Select One Relationship panel.

No Relationship

When you use a JOIN command and no relationship between the anchor table and the specified table exists, Access prompts for relationship information, as shown in the following figure.



```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>
      C                                                                STATE
      - +-----------------Create a New Relationship---------------+ -- -----
*** *** |                                                          | **********
___     | Requested JOIN is for two tables which are not related   |     MA
___     | and therefore require a new relationship.  ACCESS is     |     MA
___     | assuming the Parent and Child tables are as shown:       |     MA
___     |                                                          |     MA
___     | Parent Table: FOPDEMO.CUSTOMERS                          |     MA
___     | Child Table : FOPDEMO.ORDERS                             |     MA
___     |                                                          |     MA
___     | Specify a Relationship Name and indicate how to Proceed: |     MA
___     |                                                          |     MA
___     | Relationship Name ===>       >>                          |     MA
___     |                                                          |     MA
___     | Press ENTER to Keep Tables as Shown and Proceed          |     MA
___     | Enter END Command to Reverse Parent and Child            |     MA
___     | Enter CANCEL Command to Cancel JOIN and Return to Editor |     MA
___     +----------------------------------------------------------+     MA
```

Figure 27. Create a New Relationship

Access assumes the currently displayed table is the parent table and the table specified with the JOIN command is the child table. Press ENTER if this is correct, or use END to reverse the names of the parent table and the child table.

Supply the name for the relationship you are defining in **Relationship Name**, and press ENTER. The relationship editor, described in the *Common Elements Manual* in the section for Edit or Browse a Relationship, is displayed.

Join All Command

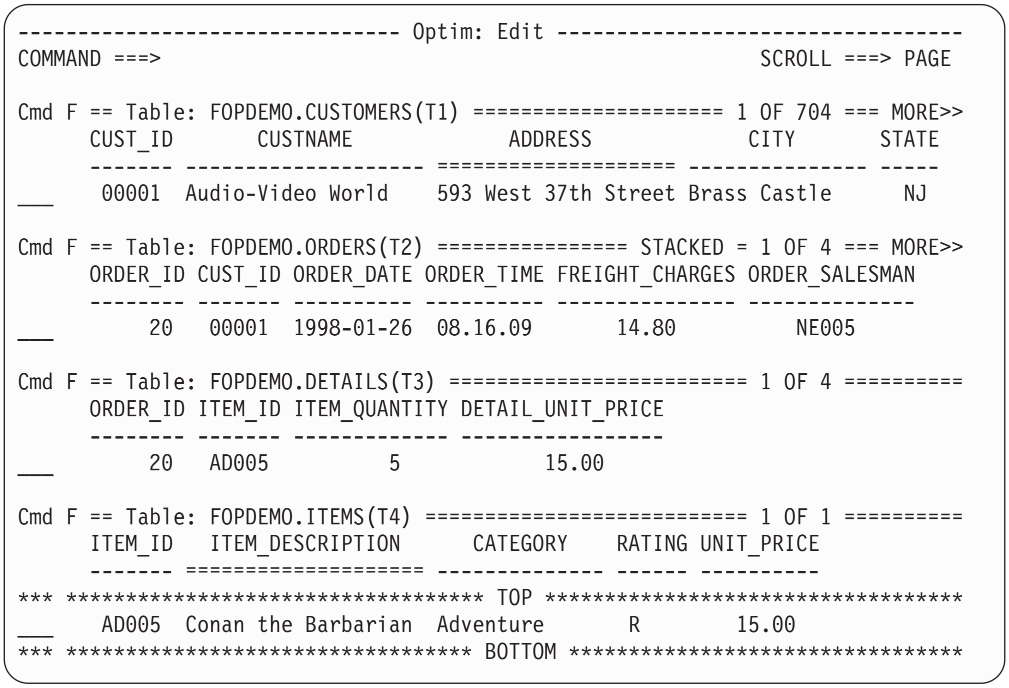
A temporary Access Definition is used during a browse or edit session invoked using Option 1 or 2 on the Main Menu.

As you Join tables in a browse or edit session, those tables are added to the temporary Access Definition, and, even if unjoined during the session, remain in the Access Definition. Instead of manually joining each table in the display, you can use the JOIN ALL command to join all tables in the temporary Access Definition.

Note: During an edit or browse session with an explicit Access Definition (Option 3 or 4 on the Main Menu), you can use the JOIN ALL command to join all related tables listed in the Access Definition.

With a temporary or explicit Access Definition, you cannot use the JOIN ALL command if tables are currently joined in the display. If you enter JOIN ALL when one or more tables are joined, an error message is displayed.

The following figure shows the results of executing the JOIN ALL command for the CUSTOMERS table, when the related ORDERS, DETAILS, ITEMS, and SALES tables are in the temporary Access Definition.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 704 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    00001  Audio-Video World    593 West 37th Street Brass Castle      NJ

Cmd F == Table: FOPDEMO.ORDERS(T2) ================ STACKED = 1 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
___        20   00001  1998-01-26  08.16.09       14.80          NE005

Cmd F == Table: FOPDEMO.DETAILS(T3) ========================= 1 OF 4 ==========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
___        20   AD005          5            15.00

Cmd F == Table: FOPDEMO.ITEMS(T4) =========================== 1 OF 1 ==========
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___    AD005  Conan the Barbarian  Adventure       R        15.00
*** ********************************** BOTTOM *********************************
```

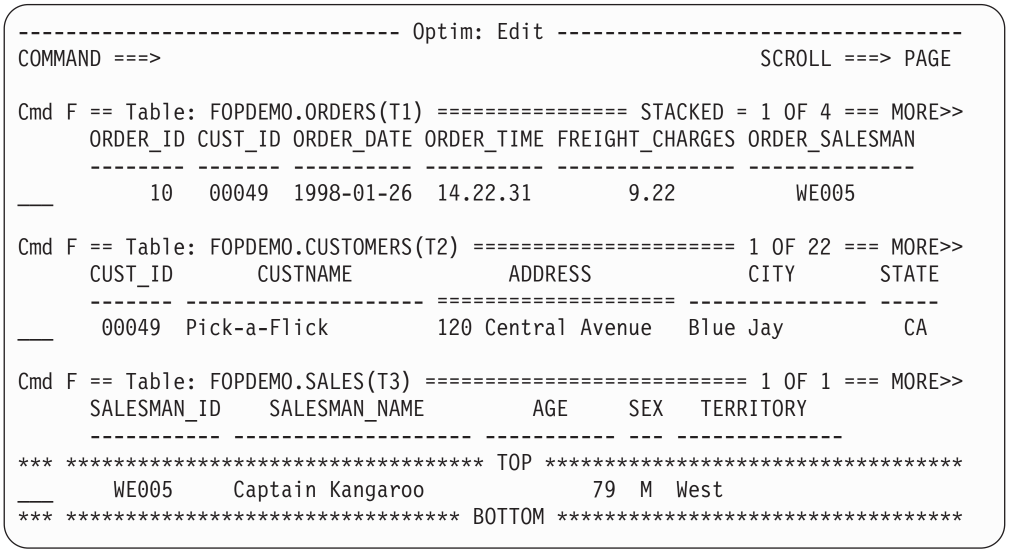
Figure 28. Results of JOIN ALL

If there are two or more relationships between a parent and child table, Access prompts you to select the relationship to use for the join. See Figure 25 on page 31 for details.

In Figure 28, the SALES table is not displayed because it is “stacked.” For information about multi-way joins and “stacked” tables, read “Special Considerations for Multi-way Joins” on page 36. For information about the multiple table display, read “Multiple Table Display.”

Multiple Table Display

After one or more joins, the display shows related rows from multiple tables. In the following figure, the CUSTOMERS table is joined to the ORDERS table, and the SALES table is joined to the CUSTOMERS table.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ================ STACKED = 1 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
___        10   00049  1998-01-26  14.22.31        9.22          WE005

Cmd F == Table: FOPDEMO.CUSTOMERS(T2) ====================== 1 OF 22 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
___    00049  Pick-a-Flick         120 Central Avenue   Blue Jay          CA

Cmd F == Table: FOPDEMO.SALES(T3) =========================== 1 OF 1 === MORE>>
      SALESMAN_ID    SALESMAN_NAME         AGE     SEX   TERRITORY
      ----------- -------------------- ----------- --- --------------
*** *********************************** TOP ***********************************
___     WE005     Captain Kangaroo              79  M  West
*** ********************************* BOTTOM **********************************
```

Figure 29. Sample Multiple Table Display

An information line is provided for each table in the multi-table display that includes the following information:

Included Information
    Example

Name of the Table
    FOPDEMO.ORDERS

Identifier supplied by Access
    (T1)

Row number relative to the total number of rows selected for display
    4 OF 78

The TOP and BOTTOM markers are displayed before the first line and after the last line of the SALES table in Figure 29. These markers indicate the first and last related rows in the table.

If more than one table is joined at one level (i.e., a multi-way join), the STACKED indicator is displayed. (See “Special Considerations for Multi-way Joins” on page 36 for details.)

The maximum number of tables that can be joined is 64. If necessary, higher-level tables are scrolled from the display to make room for newly joined tables. Use the UNJOIN command to redisplay higher-level tables by removing subordinate tables. For more information about the UNJOIN command, see “Unjoin Tables” on page 40.

Note: Additionally, you can use the ZOOM command to focus the display on a single, higher-level table in the display, without unjoining the table. For more information about the ZOOM command, see “Zoom a Joined Table Display” on page 39.

Change Highest Display Levels

You can use the START command to place a specified table first on the display, temporarily removing tables that are before the specified table in the hierarchy. The table name or identifier for a table currently active in the display may be specified with the START command (e.g., START SALES or START T3), as long as there is sufficient screen space to display at least one line of the lowest-level table.

Note: The START command does not change the Start Table or remove tables from the Access Definition.

The START command is useful when you want to display additional rows from the lowest-level table without zooming the display. Changing the highest-level table is also useful when you change from split screen to single screen and you want to focus on one or more specific tables.

You can also use the **Indented Table Display** to view or navigate the tables. For more information, read the information on Display Traversal Paths in "Special Considerations for Multi-way Joins."

Special Considerations for Multi-way Joins

In many cases, a database table is related to two or more other tables. When multiple tables are related, several navigational paths are available for joining data.

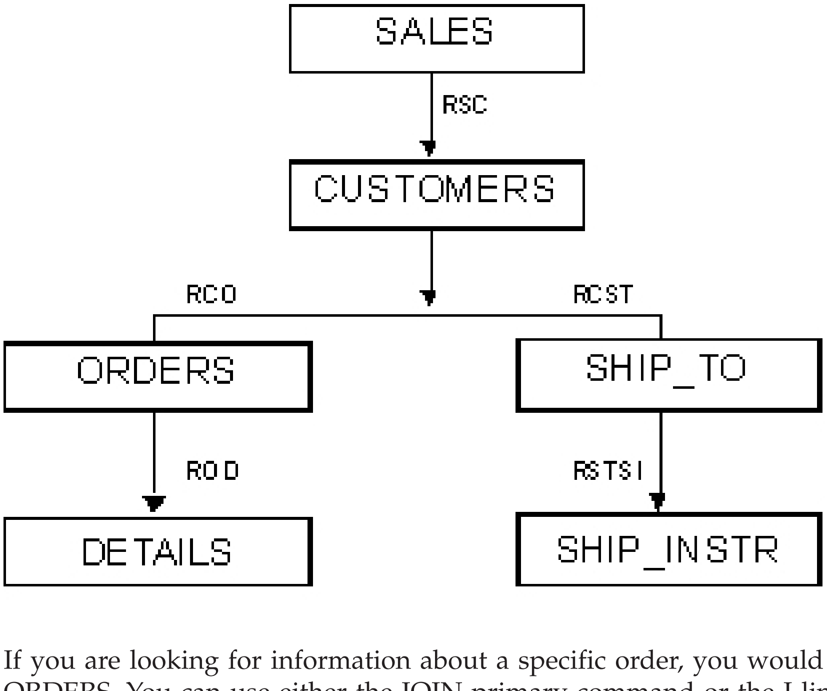
You may find it useful to display and edit simultaneously the related data obtained by traversing several of these paths. You can use multi-way joining to access multiple related tables without unjoining any tables.

Stacked Tables

When multiple tables are joined to a single table, the tables are "stacked" in the order they were joined. The most recently joined table is stacked first, the second most recently joined table is stacked second, and so on. The table that is stacked first is displayed; the other tables are hidden.

You can stack multiple tables at any display level, except T1. If tables are joined to a stacked table, these subordinate tables are hidden when the stacked table is hidden.

Consider the following example. A database contains the tables shown in the following diagram with the indicated relationships:



If you are looking for information about a specific order, you would start the edit or browse session with ORDERS. You can use either the JOIN primary command or the J line command to display the **JOIN Selection List Prompt**. The selection list of all related tables contains CUSTOMERS and DETAILS. You can select both tables, which are then joined in a stack with the last joined table, DETAILS, displayed. The other table in the stack, CUSTOMERS, is hidden.

Any Level Join

You can stack tables at any display level. To specify the display level for the join request, use the JOIN command with either the FROM operand and the anchor table name or identifier, or the cursor position indicating the anchor table.

For example, assume ORDERS is the lowest-level table and is the only table joined to CUSTOMERS. You can join another table to CUSTOMERS, without unjoining ORDERS, by specifying:

```
JOIN SHIP_TO FROM CUSTOMERS
```

The SHIP_TO table is joined to CUSTOMERS. The SHIP_TO table is stacked first and is displayed. ORDERS is stacked last and is hidden.

You can also use DB2 LIKE syntax to display a selection list of tables to stack at any display level. For example, assume ORDERS is the only table joined to CUSTOMERS. To display a selection list of all related tables beginning with S to join to CUSTOMERS, specify:

```
JOIN S% FROM CUSTOMERS
```

You can select one or more tables to add to the stack.

Switch the Displayed Table

You can display any table in a stack using the SWITCH command.

- If only two tables are joined in the stack, the hidden table is displayed and the previously displayed table is hidden.
- If several tables are joined, SWITCH displays a selection list.
- If a single table is joined, a message is displayed.

By default, the SWITCH command operates on the lowest level of the display. You can use the cursor position to indicate the target level for the SWITCH command. You may also find it useful to assign the SWITCH command to a PF key and use the cursor position to indicate the target level.

Using SWITCH with several tables

If several (three or more) tables are joined in a stack, you can use operands to display a specific table in a stack at a specific level. To display a selection list of joined tables for a stack at the second level, specify:

```
SWITCH T2
```

To display the next table in a stack on the second level, specify:

```
SWITCH T2 NEXT
```

You can also specify the name of a table. To display the ORDERS table in a stack on the second level, specify:

```
SWITCH T2 ORDERS
```

Related lower-level tables

Any table in a stack can be joined to lower-level tables. However, only related data is displayed. Any subordinate tables or stacks are hidden or displayed with the stacked table to which they are joined. If you switch the display from one table in a stack to another, the related lower-level tables also switch.

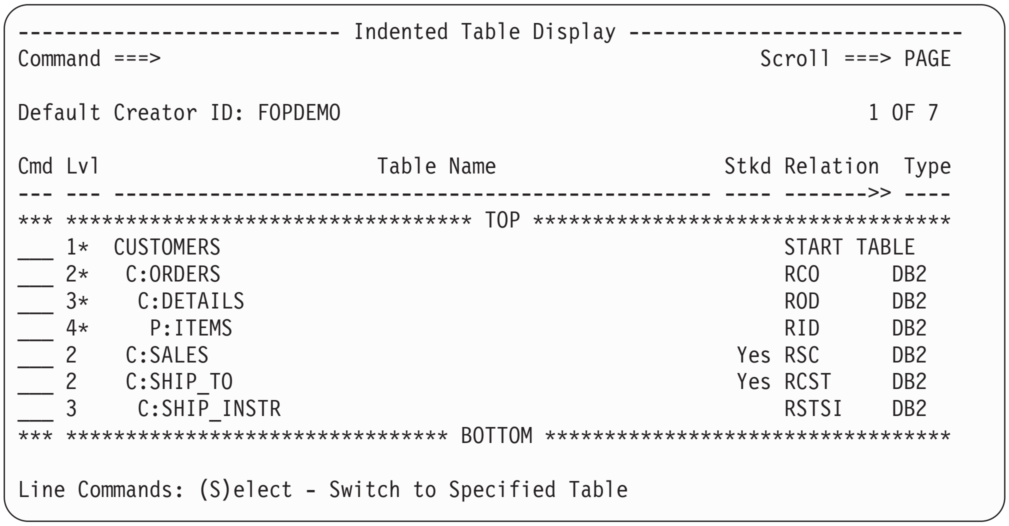
Display Traversal Paths

The edit or browse session can become very complex with multi-way joining. You may have joined to tables on subordinate levels for one or more tables within the stack and you may have “nested stacks” (i.e., stacked tables within a stack). Even if there are no stacks, it may be difficult to keep track of joined tables and the relationships between them, especially if all levels cannot be displayed on the screen at one time.

You can use the INDENT primary command to obtain the **Indented Table Display**, which provides an indented list of all active tables. The format of this list identifies the display-level hierarchy of all joined tables and all stacks.

Note: The INDENT command is available during an edit or browse session, regardless of whether you have stacked or joined tables.

In the following figure, CUSTOMERS is the Start Table. All active tables are listed, and asterisks are used to identify the tables currently displayed on the main panel from which the indented listed was generated. The display level of each table is indicated at the beginning of each line. Tables in a stack are on the same level and, except for the active table, are identified by the word “Yes” in the **Stkd** column, as shown for the SALES and SHIP_TO tables.



```
--------------------------- Indented Table Display ----------------------------
Command ===>                                                  Scroll ===> PAGE

Default Creator ID: FOPDEMO                                            1 OF 7

Cmd Lvl                       Table Name                   Stkd Relation  Type
--- --- -------------------------------------------------- ---- ------->> ----
*** ********************************** TOP ***********************************
___ 1*  CUSTOMERS                                               START TABLE
___ 2*   C:ORDERS                                               RCO      DB2
___ 3*    C:DETAILS                                             ROD      DB2
___ 4*     P:ITEMS                                              RID      DB2
___ 2    C:SALES                                            Yes RSC      DB2
___ 2    C:SHIP_TO                                          Yes RCST     DB2
___ 3     C:SHIP_INSTR                                          RSTSI    DB2
*** ******************************** BOTTOM **********************************

Line Commands: (S)elect - Switch to Specified Table
```

Figure 30. Indented Table Display

Panel Elements

The panel includes:

Default Creator ID
The Default Creator ID specified on the **Choose a DB2 Table/View** panel.

x OF y
The relative position of the first displayed table (*x*) and the total number of active tables (*y*).

Cmd The line command entry area. Use the S line command to switch the display to a different table in a stack.

For example, in Figure 30, the ORDERS, SALES, and SHIP_TO tables are stacked at level 2, and the ORDERS table is currently displayed. You can display the SHIP_TO table by typing “S” in **Cmd** for the SHIP_TO table, or any subordinate table of the SHIP_TO table (e.g., SHIP_INSTR), and pressing ENTER.

Lvl The display level for the table. Tables at the same level are in a stack, and the currently displayed table is identified by an asterisk next to its level number. Any stacked tables that are not currently displayed are identified by the word “Yes” in **Stkd**.

Table Name

The name of the table. If the Creator ID is different from the default Creator ID, it is included with the name.

The Start Table is listed first, with other tables listed in the order displayed. Indentations reflect the display level. Tables in a stack are listed in stack order.

The names of all tables, except the Start Table, are prefixed with **P:** or **C:** to indicate whether a table is the parent or child in the relationship with the table under which it is indented.

Stkd Any stacked tables that are not currently displayed are identified by the word “Yes.”

Relation

The name of the relationship between the table and the table under which it is indented. START TABLE is always specified for the Start Table.

Type Indicates whether the relationship is defined in the DB2 Catalog or the Optim Directory.

Use END on the **Indented Table Display** panel to return to the edit or browse session.

Zoom a Joined Table Display

The Zoom command switches the display between a multi-table display and a single table format without altering the joins. You can use the Zoom command when multiple tables are joined and you want to display additional rows from a higher-level table.

Zoom Line Command

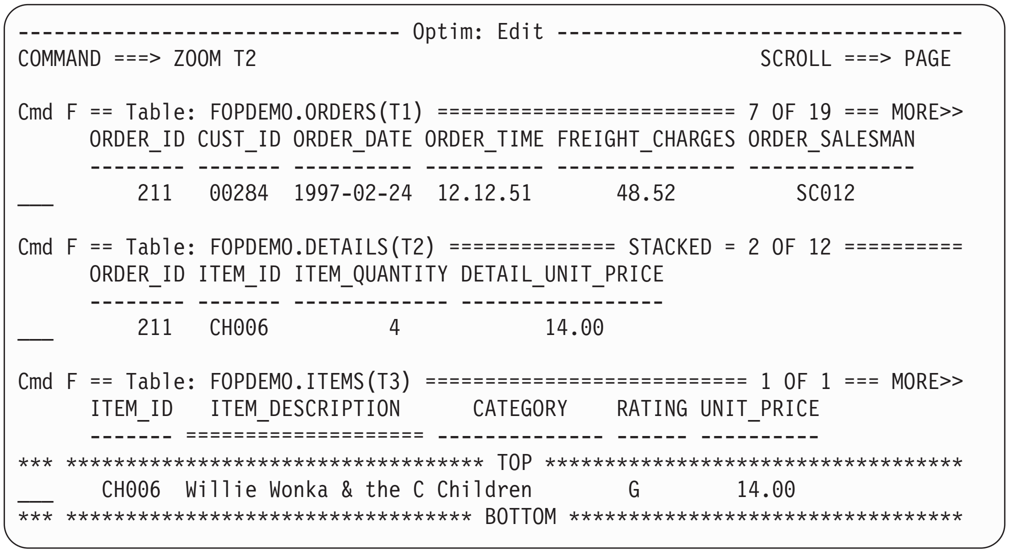
To switch from a multi-table display to focus on a single table, enter the Z line command for a row in the table. To return to a multi-table display, enter the Z line command for any row. The row on which the command is entered is displayed first in the multi-table display.

ZOOM Primary Command

You can select a table to zoom using an operand (e.g., ZOOM ORDERS or ZOOM T1), or by positioning the cursor. If the cursor is positioned on a row, that row is the first line of data in the zoomed display. If ZOOM is executed without positioning the cursor or specifying a table name operand, the lowest-level table is displayed in single table format.

Similarly, if the cursor is positioned on a row in the zoomed display when ZOOM is requested again, the selected row is scrolled to the first position in the multi-table display. If the cursor is not positioned, the first displayed row in the zoomed display is shown in the multi-table display.

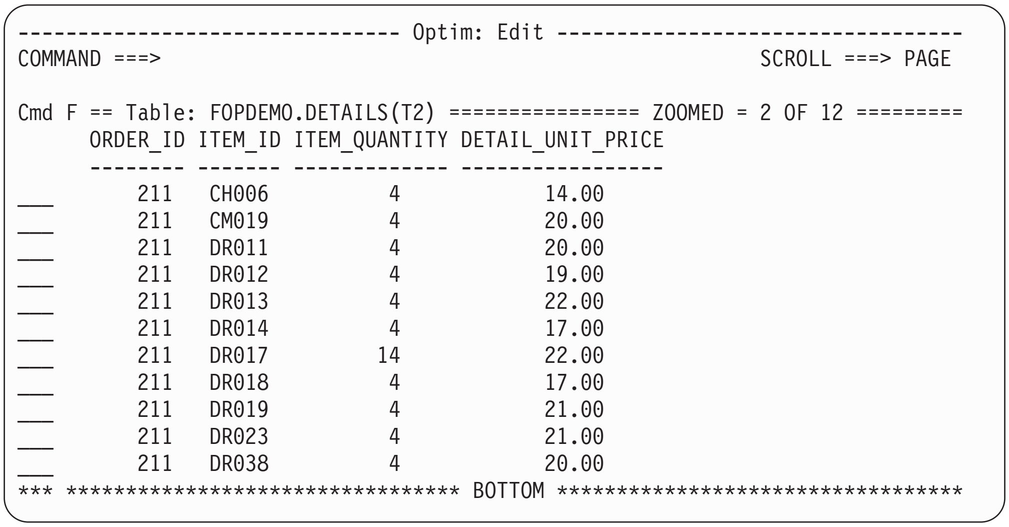
The primary command can be entered as ZOOM DETAILS or ZOOM T2, as shown in the following figure.



```
--------------------------------- Optim: Edit ----------------------------------
COMMAND ===> ZOOM T2                                          SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 7 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
___       211   00284  1997-02-24  12.12.51       48.52          SC012

Cmd F == Table: FOPDEMO.DETAILS(T2) ============== STACKED = 2 OF 12 ==========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
___       211   CH006          4           14.00

Cmd F == Table: FOPDEMO.ITEMS(T3) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___     CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM *********************************
```

Figure 31. Zoom Multi-Table Display

In the following figure, the display is changed from multi-table to a single-table display of the DETAILS table.



```
--------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.DETAILS(T2) ================ ZOOMED = 2 OF 12 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
___       211   CH006          4           14.00
___       211   CM019          4           20.00
___       211   DR011          4           20.00
___       211   DR012          4           19.00
___       211   DR013          4           22.00
___       211   DR014          4           17.00
___       211   DR017         14           22.00
___       211   DR018          4           17.00
___       211   DR019          4           21.00
___       211   DR023          4           21.00
___       211   DR038          4           20.00
*** ********************************* BOTTOM *********************************
```

Figure 32. Zoomed Details

Note: Access returns to the multi-table columnar display even if the zoom is executed while in sidelabels format. (See "Wide Data Displays" on page 55 for a description of sidelabels format.)

Impact on Joined Tables

All joins between tables are retained when zoom is used. The single-table zoomed display can be scrolled and edited. If a table is scrolled during a zoomed display, any tables joined at a subordinate level are logically scrolled. When the multi-table display is restored, the scrolled data is displayed.

Unjoin Tables

In a multi-table display, the UNJOIN command severs the join between the specified table and the next higher-level table. The specified table and any related lower-level tables are removed from the display.

When you enter the UNJOIN command, you can either position the cursor on the desired table, regardless of level, or you can specify the table name or identifier as an operand (e.g., UNJOIN ORDERS or UNJOIN T2). The UNJOIN command removes the specified table and all related lower-level tables. If you do not specify a table, the lowest-level table is unjoined.

You can also use the UNJ line command on a row in the table you want to unjoin. Entering the UNJ command removes the specified table and any related lower-level tables from the display.

If the unjoined table is not stacked, the remaining joined tables are repositioned and the display of the lowest-level table expands to include any additional rows.

Unjoining Stacked Tables

If the unjoined table is stacked, the next table in the stack is displayed along with any related lower-level tables.

You can use operands with the UNJOIN command to unjoin all tables in a stack. UNJOIN ALL removes all tables in a stack at the lowest level. If there is no stack at the lowest level, the table at the lowest level is unjoined. When there are multiple stacks, you can specify the target stack using the cursor position or the level indicator (e.g., UNJOIN T2 ALL).

Using an Explicit Access Definition

This section highlights the features unique to performing a join operation during an edit or browse session initiated using an explicit Access Definition.

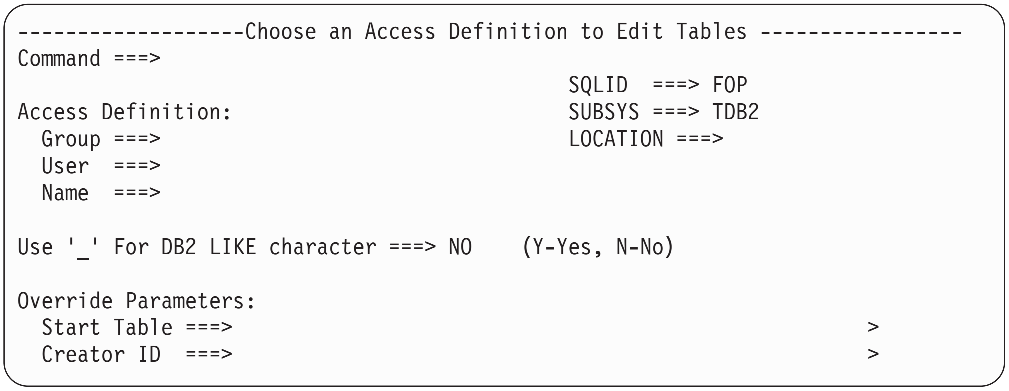
Joining tables during a session initiated using an Access Definition (Option 3 or 4 on the **Main Menu**) is similar to joining tables during an edit or browse session without an Access Definition (Option 1 or 2 on the **Main Menu**). However, Options 3 and 4 can limit access to tables that are not included in the Access Definition.

After you create and save an Access Definition, you can use it to initiate an edit or browse session by selecting Option 3 BROWSE USING AD or Option 4 EDIT USING AD from the **Main Menu**. (Refer to the *Common Elements Manual,* section on Access Definitions, for details about creating Access Definitions.)

Choose an Access Definition

The Choose an Access Definition panel prompts for the name of the Access Definition, which must exist. When this panel is displayed from Option 3, it is called the Choose an Access Definition to Browse Tables panel. When it is displayed from Option 4, it is called the Choose an Access Definition to Edit Tables panel. The two panels are identical in appearance and operation in all other respects. (Note that you cannot edit or browse using an Access Definition that contains substitution variables.)

You can provide the name of an Access Definition or request a selection list.



```
-------------------Choose an Access Definition to Edit Tables -----------------
Command ===>
                                              SQLID  ===> FOP
Access Definition:                            SUBSYS ===> TDB2
  Group ===>                                  LOCATION ===>
  User  ===>
  Name  ===>

Use '_' For DB2 LIKE character ===> NO    (Y-Yes, N-No)

Override Parameters:
  Start Table ===>                                                     >
  Creator ID  ===>                                                     >
```

Figure 33. Choose an Access Definition to Edit Data

Panel Prompts

The prompts on the panel are:

Group Specify the 1- to 8-character group name. The default is the previously entered value or, if **Group** has never been specified, the TSO prefix for the current user. Leave **Group** blank or use DB2 LIKE syntax to obtain a selection list.

User Specify the 1- to 8-character user name. The default is the previously entered value, or if **User** has never been specified, the DB2 SQLID for the current user. The DB2 SQLID is specified on the **Main Menu** to establish the connection to DB2. Leave **User** blank or use DB2 LIKE syntax to obtain a selection list.

Name Specify the 1- to 12-character base name for the Access Definition. This value is retained and redisplayed the next time this panel is displayed. Leave **Name** blank or use DB2 LIKE syntax to obtain a selection list.

Use '_' for DB2 LIKE character
Indicator for use of the underscore (_) to represent any single character in a name. Specify Y to use the underscore as a DB2 LIKE character (default), or N to use it literally as part of the name. Note that when the ′%′

For example, A_B might be a three-character name containing the characters 'A_B' as entered or a three-character name that begins with “A” and ends with “B” with any valid character in the middle.

Note: When the ′_′ character is used in conjunction with the ′%′ wildcard character, the ′_′ is treated as a DB2 LIKE character, and not as a literal.

Start Table
Specify the name of a table to override the Start Table specified in the Access Definition. This table is displayed first in an edit or browse session. The table must be referenced in the Access Definition.

Creator ID
Specify a Creator ID to override the default Creator ID specified in the Access Definition.

SQLID
The current SQLID. Modify this value to connect using a different SQLID.

SUBSYS
The current DB2 subsystem. Modify this value to connect to a different DB2 subsystem.

When connecting to a remote subsystem, this value should be the local subsystem where the remote location is defined.

LOCATION

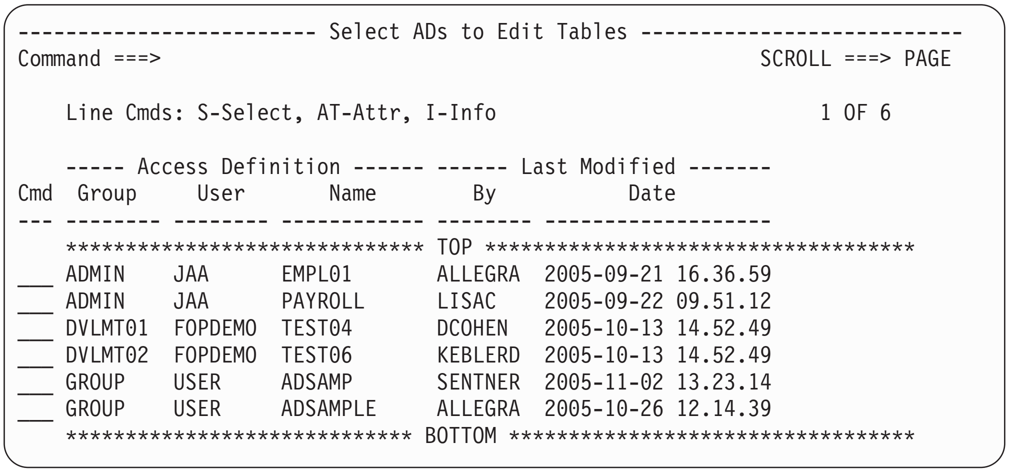
The remote location. This prompt is displayed if remote access is available. Specify a value to connect to a remote DB2 subsystem. You can use a percent sign (%) to obtain a selection list of available locations. If the connection fails, the session is restarted and the **Main Menu** is displayed.

Note: If you leave this prompt blank, the local subsystem is assumed.

Selection List

You can display a selection list of Access Definitions from which to choose. Either leave a prompt blank or use DB2 LIKE syntax to obtain the desired selection list.

Assume Access Definitions have been defined and **Group**, **User**, and **Name** on the **Choose an Access Definition** panel are blank or contain DB2 LIKE syntax. A selection list is displayed.



```
------------------------ Select ADs to Edit Tables ---------------------------
Command ===>                                                 SCROLL ===> PAGE

    Line Cmds: S-Select, AT-Attr, I-Info                          1 OF 6

    ----- Access Definition ------ ------ Last Modified -------
Cmd  Group     User      Name        By           Date
--- -------- -------- ------------ -------- -------------------
    ****************************** TOP ***********************************
___ ADMIN    JAA      EMPL01       ALLEGRA  2005-09-21 16.36.59
___ ADMIN    JAA      PAYROLL      LISAC    2005-09-22 09.51.12
___ DVLMT01  FOPDEMO  TEST04       DCOHEN   2005-10-13 14.52.49
___ DVLMT02  FOPDEMO  TEST06       KEBLERD  2005-10-13 14.52.49
___ GROUP    USER     ADSAMP       SENTNER  2005-11-02 13.23.14
___ GROUP    USER     ADSAMPLE     ALLEGRA  2005-10-26 12.14.39
    ***************************** BOTTOM *********************************
```

Figure 34. Select an Access Definition

The selection list includes the full name of the Access Definition (**Group.User.Name**), the TSO User ID for the person who created or last modified the Access Definition, and the date and time of the last update. A user option determines if the description of each Access Definition is displayed on this panel. See the section for User Options, in the *Common Elements Manual* for information about the **Selection List Format** option.

Line Commands

The following line commands are available on the **Select ADs to Edit Tables** panel.

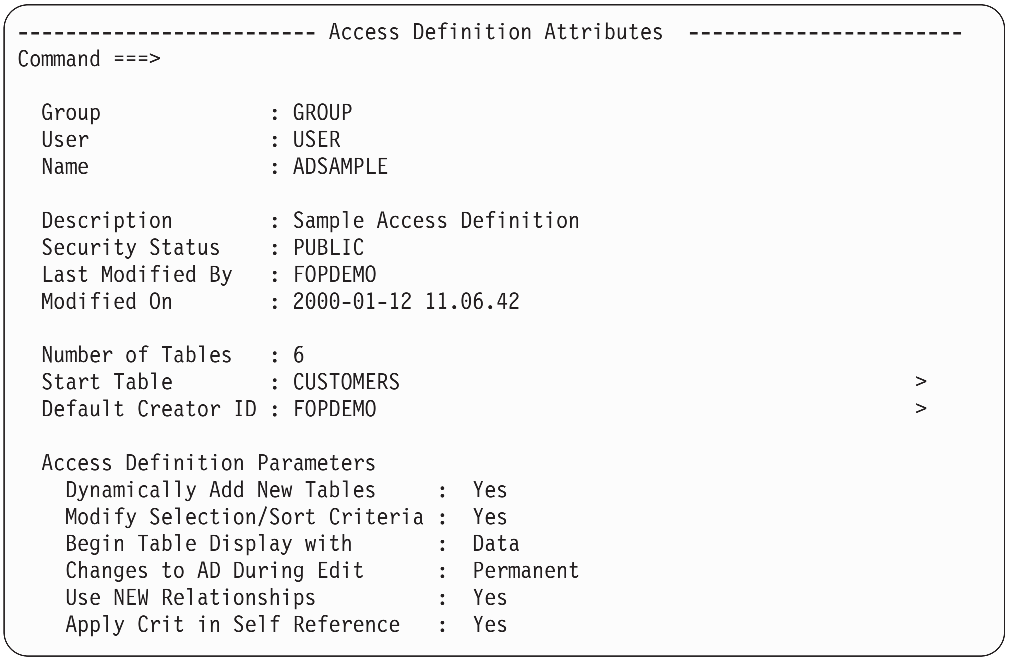
S Select an Access Definition for processing on the Optim: Edit using AD panel, which will include the name of the Access Definition you selected in the panel's name. For example, if the Access Definition you selected is named ADMIN.TOM.PAYROLL, the panel's name will appear as **Optim: Edit using AD ADMIN.TOM.PAYROLL**.

I Display information about attributes for an Access Definition.

AT Display the Object Attributes panel to modify the attributes of an Access Definition. This panel provides a 40-character area for the description of an Access Definition. Site management determines whether this panel also displays a prompt for Security Status.

Access Definition Attributes

To display the attributes of an Access Definition, type I in **Cmd** next to the name of the Access Definition. The following figure shows the read-only Access Definition Attributes panel.



```
------------------------ Access Definition Attributes  -----------------------
Command ===>

  Group              : GROUP
  User               : USER
  Name               : ADSAMPLE

  Description        : Sample Access Definition
  Security Status    : PUBLIC
  Last Modified By   : FOPDEMO
  Modified On        : 2000-01-12 11.06.42

  Number of Tables   : 6
  Start Table        : CUSTOMERS                                        >
  Default Creator ID : FOPDEMO                                          >

  Access Definition Parameters
    Dynamically Add New Tables     :  Yes
    Modify Selection/Sort Criteria :  Yes
    Begin Table Display with       :  Data
    Changes to AD During Edit      :  Permanent
    Use NEW Relationships          :  Yes
    Apply Crit in Self Reference   :  Yes
```

Figure 35. Access Definition Attributes

Access generates and maintains all information on this panel from your specifications for the Access Definition. The parameters listed at the end of the access definition attributes are the settings specified on the Access Definition Parameters panel. (See the *Common Elements Manual*, section for Access Definitions, for more information on these parameters.)

Primary Commands

The following primary commands are available on the Select ADs to Edit Tables panel.

- BOTTOM
- CANCEL
- DOWN
- END
- FIND
- LOCATE
- OPTIONS
- RESET
- RFIND
- SELECT
- SHOW
- SORT
- TOP
- UP

You can use the SORT, LOCATE, and FIND primary commands to manage the list. The SORT command sorts the list by the specified column heading. The LOCATE command locates and scrolls to the specified string in the column that was last used in a SORT. (If a SORT has not been performed, LOCATE searches values in **Group**.) The FIND command locates a specific character string anywhere in the selection list.

Select an Access Definition

Use the S line command or the SELECT primary command to select an Access Definition. The Access Definition named on the SELECT command does not have to be displayed on the selection list, but it must exist.

The session begins with a display of rows from the Start Table specified on the Choose an Access Definition panel. Related rows from other tables can be added to the display using the JOIN command.

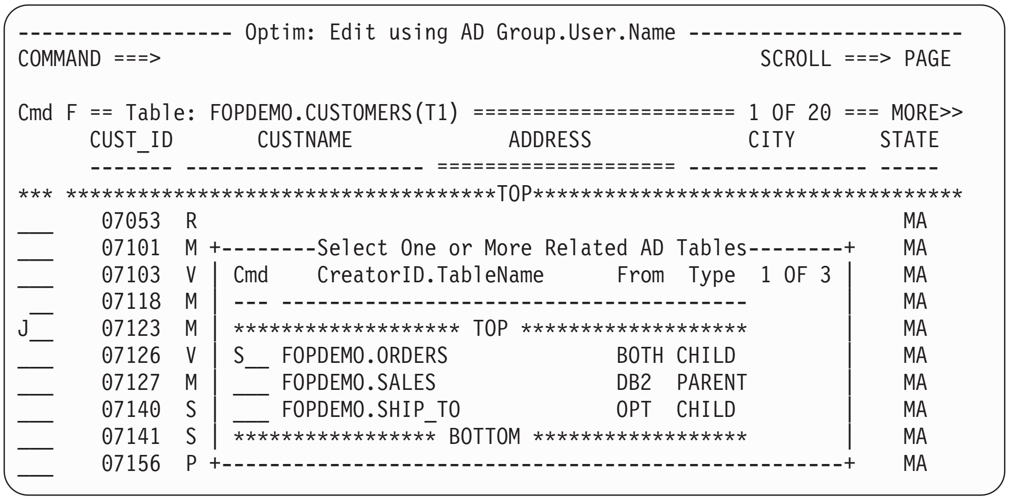
One Related Table

Assume the Access Definition lists only one table related to the anchor table. If you enter the JOIN command with no operands, the join occurs automatically, with no prompting. In this instance, only the tables listed in the Access Definition are considered. For example, assume you are viewing the CUSTOMERS table using an Access Definition that references the CUSTOMERS and ORDERS tables. The JOIN command with no operands automatically joins the ORDERS table to the CUSTOMERS table—if ORDERS is the only table referenced in the Access Definition that is related to CUSTOMERS.

Multiple Related Tables

Often the anchor table is related to more than one table referenced in the Access Definition. If you enter the JOIN command without entering the name of a specific table when more than one table is related, a selection list of related tables is presented. The related tables are listed in the order in which they are referenced in the Access Definition. For an edit session, this list does not include any tables in the current display. Since cycles are supported during a browse session, all related tables, including those in the current display, are listed.

Note: When the Optim: Edit using AD panel is displayed, its name will include the name of the Access Definition you selected, as indicated in the following figure by the AD name “Group.User.Name.” This name is used in this manual to denote the three parts that make up all Access Definition names.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ***********************************TOP***********************************
___    07053  R                                                          MA
___    07101  M +--------Select One or More Related AD Tables--------+    MA
___    07103  V | Cmd    CreatorID.TableName      From  Type  1 OF 3 |    MA
 __    07118  M | --- ----------------------------------------       |    MA
J__    07123  M | ******************* TOP *******************        |    MA
___    07126  V | S__ FOPDEMO.ORDERS              BOTH CHILD         |    MA
___    07127  M | ___ FOPDEMO.SALES               DB2  PARENT        |    MA
___    07140  S | ___ FOPDEMO.SHIP_TO             OPT  CHILD         |    MA
___    07141  S | ***************** BOTTOM ******************        |    MA
___    07156  P +----------------------------------------------------+    MA
```

Figure 36. Join Using Selection List

Use the S line command to select a table from the list. Use END or press ENTER to terminate the selection list. Data from the joined table is related to the anchor row for the join request. If you prefer, you can bypass the selection list and enter the JOIN command with the name of the table to be joined, as in:

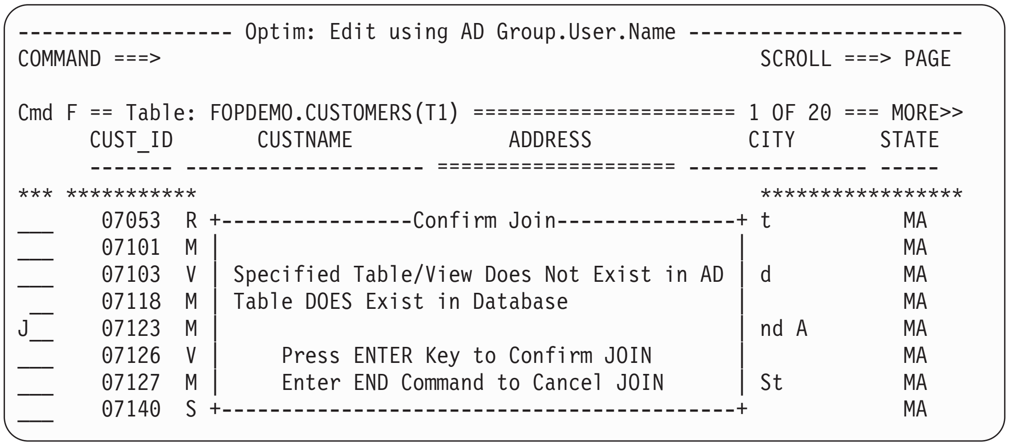
```
JOIN ORDERS
```

No Related Tables

At times, the anchor table is not related to any tables referenced in the Access Definition. In this case, if you enter the JOIN command with no operands, Access displays the JOIN Selection List Prompt, shown in Figure 23 on page 29.

Table Not in the Access Definition

If you enter the JOIN command with the name of a table that is not referenced in the Access Definition, Access searches the DB2 Catalog for the table and, if it exists, displays a confirmation prompt before joining.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                              SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ***********                                          *****************
___    07053  R +----------------Confirm Join---------------+ t           MA
___    07101  M |                                           |             MA
___    07103  V | Specified Table/View Does Not Exist in AD | d           MA
 __    07118  M | Table DOES Exist in Database              |             MA
J__    07123  M |                                           | nd A        MA
___    07126  V |     Press ENTER Key to Confirm JOIN       |             MA
___    07127  M |     Enter END Command to Cancel JOIN      | St          MA
___    07140  S +-------------------------------------------+             MA
```

Figure 37. JOIN Confirmation Prompt

When you press ENTER, the join is performed and the name of the new table is automatically added to the Access Definition. If relationship information is required or more than one relationship is defined, Access prompts for the necessary information. Use END to cancel the join request.

Note: You can join a table not referenced in the Access Definition only if the Access Definition has been defined to allow tables to be added dynamically. (See Restrictions to Dynamic Join for more information.)

JOIN NEW

You can also use the JOIN NEW command to join a table not referenced in the Access Definition, but present in the database. JOIN NEW can be used to join directly to a known table or to obtain a selection list to select a table. The newly joined table is added to the Access Definition.

You can include the table name with the command. For example, to join to a table named SHIP_INSTR that is not in the Access Definition but is present in the database, enter:

```
JOIN NEW SHIP_INSTR
```

The default Creator ID defined for the Access Definition is assumed.

If NEW is not specified on the JOIN SHIP_INSTR command, the **Confirm Join** pop-up window is displayed to confirm the request to join a table not in the Access Definition. Press ENTER to confirm. With JOIN NEW, a confirmation prompt is not displayed because Access assumes the specified table is not defined in the Access Definition.

If there is only one relationship between the anchor table and the specified table, the join is performed. If more than one relationship exists, the **Select One Relationship** pop-up window is displayed. If no relationships exist, a prompt to define a relationship is displayed (see "No Relationship" on page 32 for details).

If the name of a table is not specified on the JOIN NEW command, the **JOIN Selection List Prompt** is displayed (see "No Table Name" on page 28).

Restrictions to Dynamic Join

The Access Definition parameter, **Dynamically Add Tables**, determines whether you can join tables not referenced in the Access Definition while editing or browsing data. If you specify No for this parameter, you can join only to tables listed in the Access Definition. If you attempt to join to a table not defined in the Access Definition, a message is displayed and the join is not performed. If you specify Yes to **Dynamically Add Tables**, you can join new tables and add new relationships not listed currently in the Access Definition. When you terminate the edit or browse session, you can update the Access Definition to include the newly joined tables.

The Access Definition Parameter, **Changes to AD During Edit**, determines whether changes made to an Access Definition during an edit or browse session are permanent or temporary. See the *Common Elements Manual,* section for Access Definition Parameters, for more information.

Manage the Display

A database may contain extensive data from hundreds of tables, each with dozens of columns and thousands of rows. Navigating this data to locate a specific subset of related rows can be daunting.

Access provides many techniques and features to help you:

- Navigate large lists of data easily and efficiently.
- Find specific data and manage the display.
- Review information about columns or data.
- Arrange and format displays of wide data to accommodate editing and browsing.

Scroll

Access uses familiar ISPF commands to scroll data, including DOWN, UP, BOTTOM, TOP, LEFT, and RIGHT. These commands may be entered on the command line or assigned to program function keys. ISPF SCROLL field values, CSR, PAGE, DATA, HALF, MAX, and *n* (where *n* is the number of lines to scroll) are supported.

Vertical Scroll

Use DOWN, UP, BOTTOM, and TOP to scroll through the rows in the table when the number of rows retrieved exceeds the number of lines available on the screen. DOWN scrolls the display to succeeding table rows. UP scrolls the display to preceding table rows. Operands for the DOWN and UP commands allow you to scroll by page, numeric value, or cursor position. You can also use the BOTTOM command to scroll to the last row and the TOP command to scroll to the first row in the table.

When a single table is displayed, vertical scrolling is exactly like vertical scrolling in ISPF. Scrolling a multi-table display is discussed in Coordinated Scroll.

Horizontal Scroll

Use LEFT and RIGHT to scroll through the columns in the table when the combined width of the columns exceeds the width of the screen. Operands for these commands allow you to scroll by column name, numeric value, or cursor position. MORE, with an appropriate direction arrow, is displayed on the information line for a table to indicate that you can scroll horizontally.

When multiple tables are displayed, indicate the table to scroll by specifying a table name or identifier with the column name (for example, T1.CUST_ID) or by positioning the cursor. If a table is not indicated, the lowest-level table is scrolled.

Access maintains the “insert-pending” status when you scroll horizontally. When inserting data into a table with a partitioned index, this allows you to display any non-displayed key columns and enter data to complete the insert. This is helpful since you can insert data into the index columns of these tables, but cannot update them.

Coordinated Scroll

Scrolling a table vertically in a multi-table display automatically scrolls any tables joined at subordinate levels. For example, scrolling the DETAILS table in Figure 31 on page 40 automatically scrolls the ITEMS table to display the related row, if any. Scrolling the ORDERS table scrolls the DETAILS table and the ITEMS table to the related rows.

You can indicate the table you want to scroll by including its name or identifier as an operand for the command or by positioning the cursor. For example, UP T2 scrolls the table at level T2 backwards. If the cursor is not positioned in a window and no operands are given with the scroll command, the table displayed at the lowest level is scrolled.

You can also scroll by positioning the cursor on a specific row in any table and pressing a PF key assigned to scroll. All lower-level tables are scrolled accordingly.

Note: When scrolling a table also causes subordinate tables to be scrolled, a new set of rows is fetched for the subordinate tables. Access retains any modifications to the original set of rows in the subordinate tables. These modifications cannot be undone. For more information about the impact of scrolling when editing, see “Restoring Data” on page 81.

Navigate and Manage Display of Data

If working with many rows, you may find it difficult to find the rows you want to edit or browse. The following commands allow you to search for specific rows and adjust the display according to your needs:

Find Locate a row containing a specified value and position the cursor to that row.

Exclude
Temporarily remove specified rows from the display.

Only Display only rows that match defined criteria, excluding rows that do not.

Show Redisplay rows excluded with the EXCLUDE or ONLY commands.

Operands for these commands allow you to direct a search (e.g., where the search begins, the direction in which it proceeds, etc.). Detailed information about the operands available for each command is provided in the *Command Reference Manual* in the section for Primary Commands.

Working with Multiple Tables

In a multi-table display, the FIND, EXCLUDE, and ONLY commands operate on the lowest-level table, by default. To specify a particular table, enter the command and do one of the following:

- Use the cursor position to indicate the desired table.
- Use the IN operand with *creatorid.tablename.columnname*. For example:

```
EXCLUDE CH006 IN FOPDEMO.DETAILS.ITEM_ID
```

In a multi-table display, the SHOW command operates on all currently displayed tables.

Find Specific Data

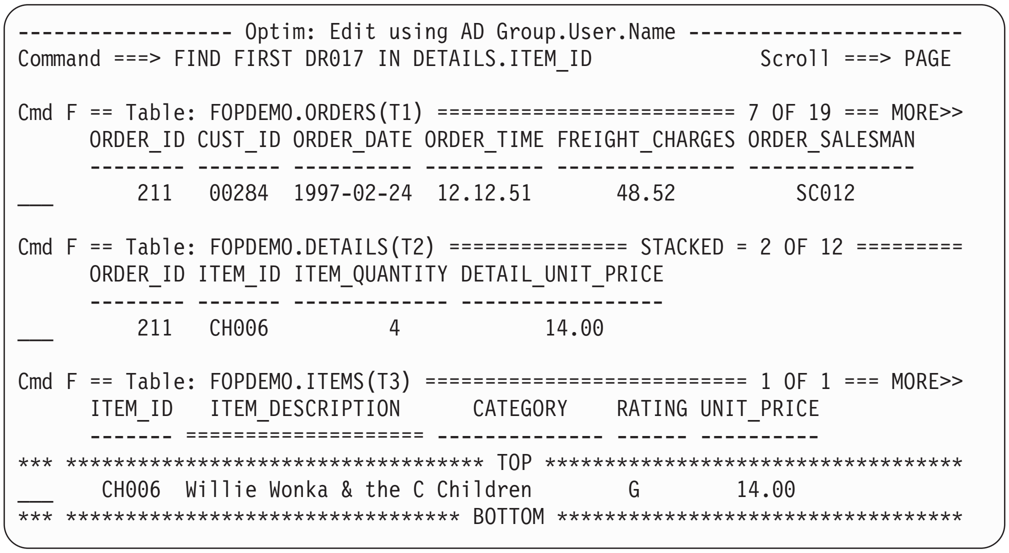
Use the FIND command to locate a row containing a specified value and position the cursor on the row and column where the value is located. A row that is not currently visible is scrolled to the beginning of the display in the window of the searched table. In a multi-table display, any subordinate tables are automatically scrolled to related rows, if any. To locate the next occurrence of the search value, use the RFIND command (usually assigned to PF5).

Several operands are available for the FIND command that allow you to specify where a search begins, the direction in which it proceeds, and whether it includes excluded or deleted rows. The Access FIND command is similar to the ISPF FIND command. The Access FIND command, however, is extended to support DB2 data, such as null values and floating point data types.

For example, to locate the first occurrence of the ITEM_ID DR017 in DETAILS, enter:

```
FIND FIRST DR017 IN DETAILS.ITEM_ID
```

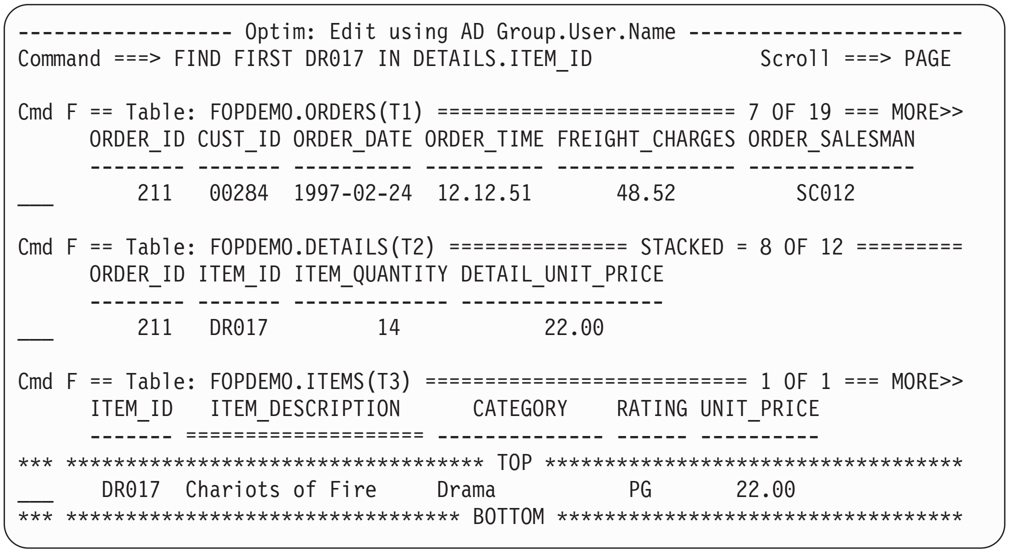
As shown in the following figure:



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===> FIND FIRST DR017 IN DETAILS.ITEM_ID              Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 7 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
___       211   00284  1997-02-24  12.12.51       48.52          SC012

Cmd F == Table: FOPDEMO.DETAILS(T2) =============== STACKED = 2 OF 12 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
___       211   CH006         4           14.00

Cmd F == Table: FOPDEMO.ITEMS(T3) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM **********************************
```

Figure 38. FIND Command

The results of the FIND are:



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===> FIND FIRST DR017 IN DETAILS.ITEM_ID              Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ======================== 7 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
___        211   00284  1997-02-24  12.12.51       48.52          SC012

Cmd F == Table: FOPDEMO.DETAILS(T2) =============== STACKED = 8 OF 12 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
___        211   DR017         14            22.00

Cmd F == Table: FOPDEMO.ITEMS(T3) ========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** *********************************** TOP **********************************
___     DR017  Chariots of Fire     Drama          PG       22.00
*** ********************************* BOTTOM *********************************
```

Figure 39. FIND Results

The FIRST operand specifies that the search is to begin with the first row of data. Otherwise, the search begins at the cursor location. The IN operand specifies the search is limited to the named column. In this example, the search is limited to the ITEM_ID column. If a column name operand is omitted, all columns with a data type appropriate to the specified value are searched. If the specified column name is in more than one table (for example, ITEM_ID, in the preceding figure) and is not prefixed with the table name, the lowest-level table containing the specified column is searched.

Case-sensitive Searching

For a case-sensitive search, qualify the search string with `C` and enclose it in quotes, as in `C'DR017'`. If the value includes blanks or apostrophes, you must use double quotes, as in `"DR' 017"`. If the value includes quotation marks, use single quotes, as in `'DR"017'`.

FIND ALL

The FIND ALL command is useful in combination with EXCLUDE ALL, to display all occurrences of the specified character string. For example, to display all rows containing “Drama”, specify:

```
EXCLUDE ALL
FIND ALL DRAMA
```

Manage Data Display

When a display includes many rows, you may use the EXCLUDE and ONLY commands to temporarily remove from the display any rows that you select or that meet specified criteria, allowing you to concentrate on the remaining rows. The SHOW command allows you to redisplay temporarily removed rows.

Note: Although excluded rows are not displayed, global editing with commands, such as CHANGE or DELETE, may affect these rows.

EXCLUDE Command

When working with a large number of rows, you can use the EXCLUDE command to temporarily remove rows from the display. The EXCLUDE primary command and the Exclude line command (X) allow you to exclude rows. Excluded rows are retained in the table, but are not displayed.

EXCLUDE Primary Command

You can use the EXCLUDE primary command to exclude rows based on specific search criteria. You must specify a search value when using the EXCLUDE primary command. Several operands for the EXCLUDE command allow you to specify the direction in which the search proceeds, the name of the column to search, and whether to include all rows. For example, to exclude the ITEMS table rows for items with a PG rating, enter:

```
EXCLUDE ALL PG IN ITEMS.RATING
```

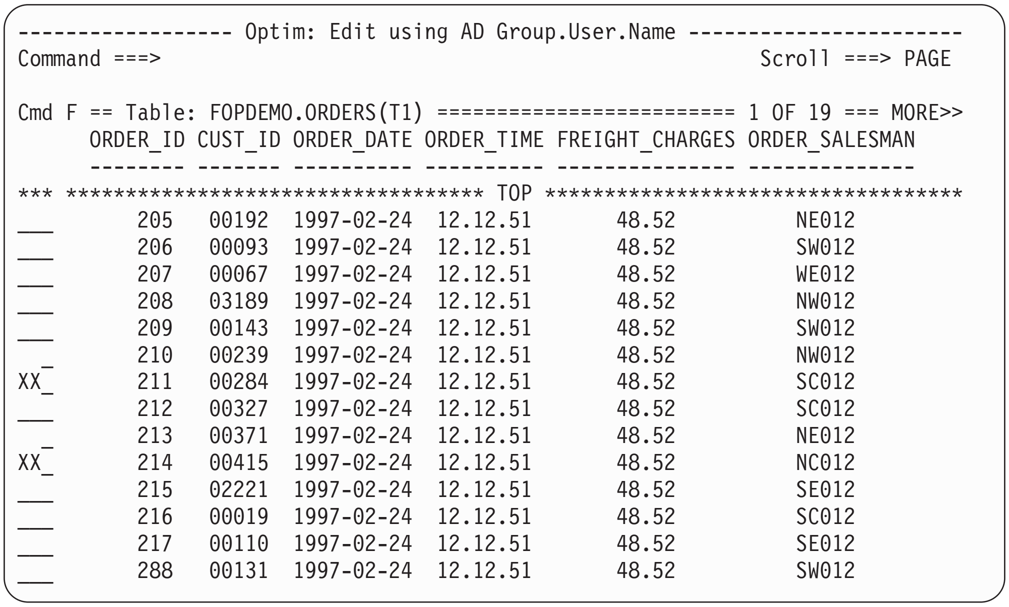
The EXCLUDE ALL command with no other parameters excludes all rows in a table.

Exclude Line Command

The Exclude line command allows you to select individual rows to exclude from the display.

- To exclude a single row, type X in **Cmd** for that row.
- To exclude a block of rows, type XX in **Cmd** for the first and last rows you want to remove from the display.

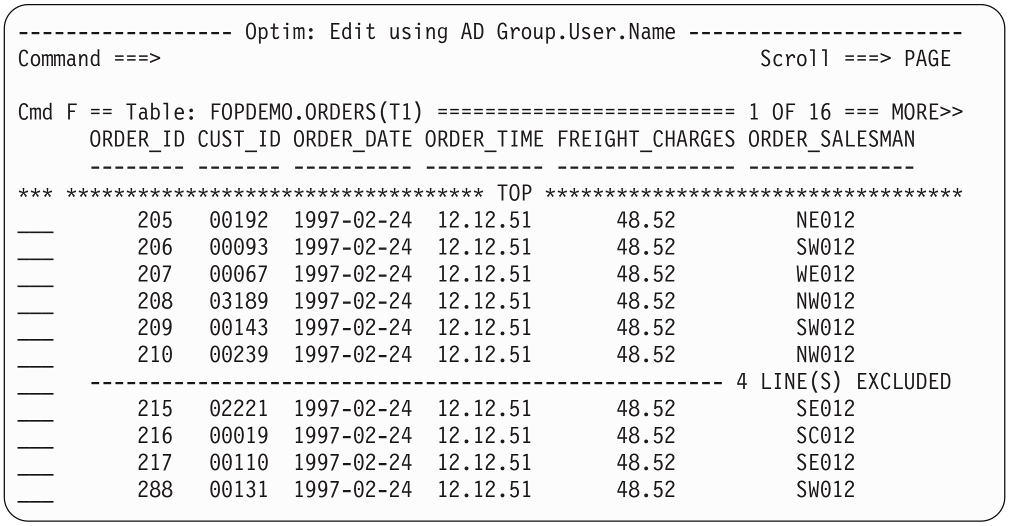
In the following example, the block form of the Exclude line command is used to identify the first and last rows of a block of rows to be excluded.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 1 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** ********************************** TOP ***********************************
___       205   00192  1997-02-24  12.12.51       48.52          NE012
___       206   00093  1997-02-24  12.12.51       48.52          SW012
___       207   00067  1997-02-24  12.12.51       48.52          WE012
___       208   03189  1997-02-24  12.12.51       48.52          NW012
___       209   00143  1997-02-24  12.12.51       48.52          SW012
  _       210   00239  1997-02-24  12.12.51       48.52          NW012
XX_       211   00284  1997-02-24  12.12.51       48.52          SC012
___       212   00327  1997-02-24  12.12.51       48.52          SC012
  _       213   00371  1997-02-24  12.12.51       48.52          NE012
XX_       214   00415  1997-02-24  12.12.51       48.52          NC012
___       215   02221  1997-02-24  12.12.51       48.52          SE012
___       216   00019  1997-02-24  12.12.51       48.52          SC012
___       217   00110  1997-02-24  12.12.51       48.52          SE012
___       288   00131  1997-02-24  12.12.51       48.52          SW012
```

Figure 40. Exclude Rows

The excluded rows are retained in the table, but are replaced with a marker indicating the location and number of excluded rows.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ======================== 1 OF 16 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP **********************************
___        205   00192  1997-02-24  12.12.51       48.52          NE012
___        206   00093  1997-02-24  12.12.51       48.52          SW012
___        207   00067  1997-02-24  12.12.51       48.52          WE012
___        208   03189  1997-02-24  12.12.51       48.52          NW012
___        209   00143  1997-02-24  12.12.51       48.52          SW012
___        210   00239  1997-02-24  12.12.51       48.52          NW012
___   ---------------------------------------------------- 4 LINE(S) EXCLUDED
___        215   02221  1997-02-24  12.12.51       48.52          SE012
___        216   00019  1997-02-24  12.12.51       48.52          SC012
___        217   00110  1997-02-24  12.12.51       48.52          SE012
___        288   00131  1997-02-24  12.12.51       48.52          SW012
```

Figure 41. Excluded Rows Marker

The excluded rows are no longer included in the row count; however, the excluded lines marker counts as a row. In this example, four consecutive rows are excluded and replaced with a marker, so the row count decreases by three, from 19 to 16. (You can redisplay the excluded rows one at a time with the S line **Cmd**, or you can redisplay all four rows by typing S4 in **Cmd**.)

ONLY Command

Use the ONLY command to limit the display to rows that match criteria you specify, excluding all rows that do not. The excluded rows are replaced with a marker. For example, to display only the rows that contain the value NE012, specify:

```
ONLY NE012
```

To limit the selection to rows with the value in the ORDER_SALESMAN column, specify:

```
ONLY NE012 IN ORDER_SALESMAN
```

The ONLY primary command provides the same results obtained by executing EXCLUDE ALL followed by FIND ALL. This ONLY command is equivalent to the EXCLUDE ALL command, followed by:

```
FIND ALL NE012 IN ORDER_SALESMAN
```

SHOW Command

Use the SHOW primary command to redisplay all rows previously hidden with the EXCLUDE or ONLY commands. When multiple tables are displayed, the SHOW command affects all currently displayed tables.

In addition, you can use the following Show line commands to redisplay specific rows:

F, F*n*
Redisplay the first row in a block of excluded rows. If *n* is specified, the first *n* rows in a block of excluded rows are redisplayed.

L, L*n*
Redisplay the last row in a block of excluded rows. If *n* is specified, the last *n* rows in a block of excluded rows are redisplayed.

S, S*n***, SS**
Redisplay one row in a block of excluded rows. If *n* is specified, *n* rows in a block of excluded rows are redisplayed. The block form of the line command, SS, redisplays all excluded rows within the specified block of rows.

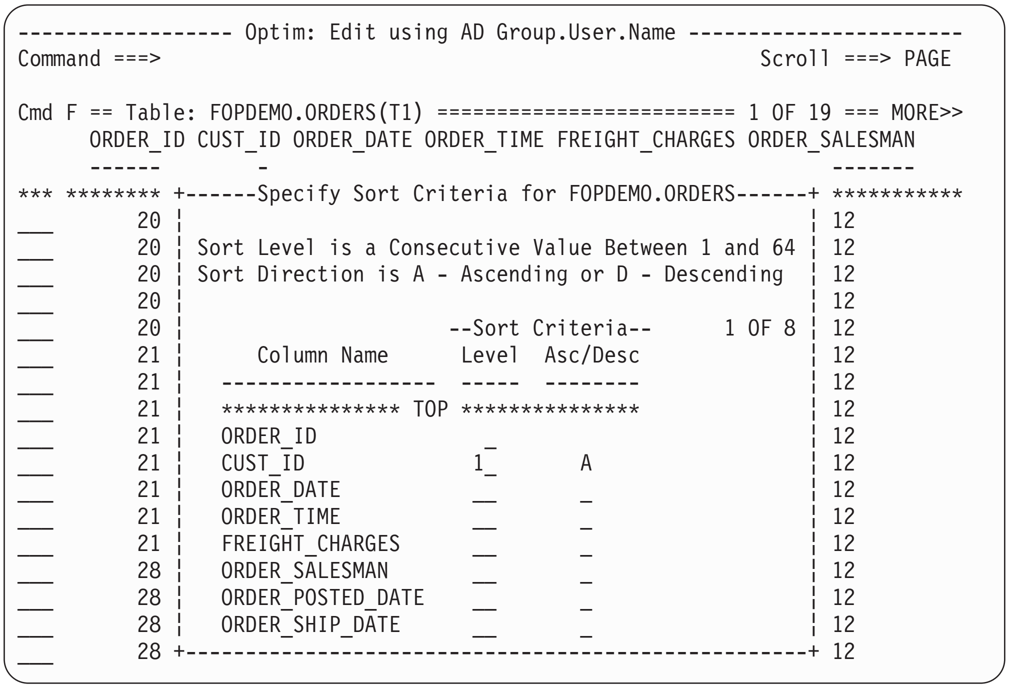
Sort Criteria

Rows in a table are displayed in no particular order. When editing or browsing data, you can use the SORT command to arrange rows in the display according to values in any column, noting that limits on the DB2 ORDER BY clause apply.

You can apply sort criteria to a table indicated by cursor position or by specifying a table name or identifier (e.g., SORT ORDERS or SORT T2). (Refer to **Access Definitions** in the *Common Elements Manual*, in the section titled Manage Data Displays, for information about defining column specifications for a table.) If you do not specify a table, the sort criteria are applied to the lowest-level table. The SORT command displays the Specify Sort Criteria panel, shown in Figure 42.

The Specify Sort Criteria panel lists all columns in the table, prompts for sort criteria, and displays any previously specified sort criteria. You can use **Level** to specify a numeric value, 1 through 64, indicating the sort priority of a column. The value for each column must be unique and consecutive starting with 1, which is the highest priority. You can use **Asc/Desc** to specify the sort order as ascending or descending. By default, rows are sorted in ascending order.

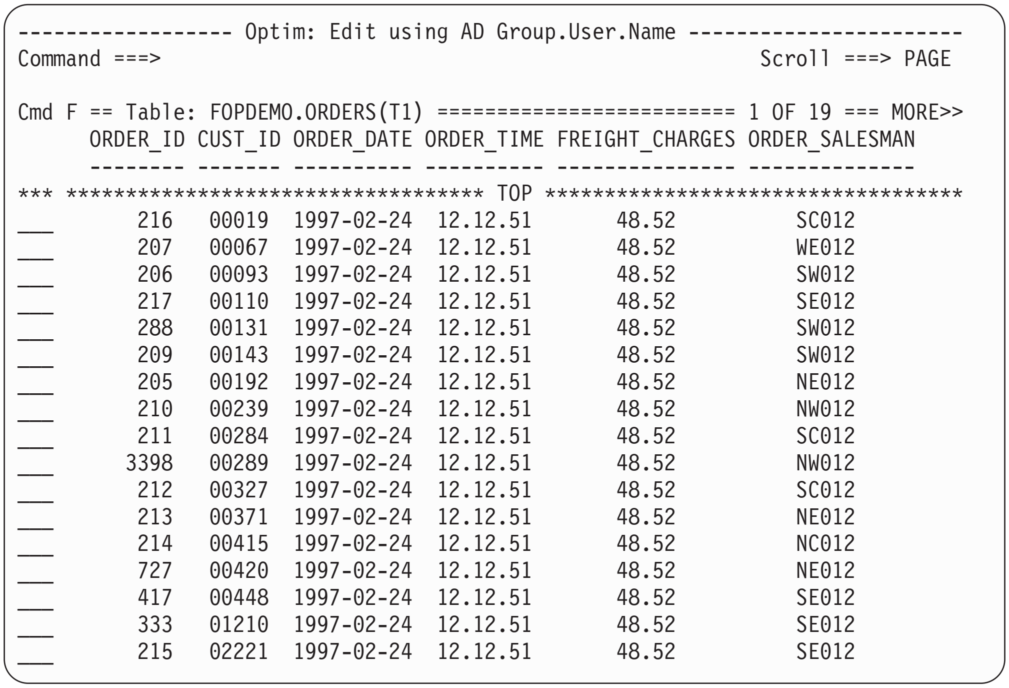
For example, to sort rows in the ORDERS table by customer ID values, specify 1 for **Level** and A for **Asc/Desc** for the CUST_ID column, as shown in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 1 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      ------        -                                               -------
*** ******** +------Specify Sort Criteria for FOPDEMO.ORDERS------+ ***********
___       20 ¦                                                     ¦ 12
___       20 ¦ Sort Level is a Consecutive Value Between 1 and 64 ¦ 12
___       20 ¦ Sort Direction is A - Ascending or D - Descending   ¦ 12
___       20 ¦                                                     ¦ 12
___       20 ¦                      --Sort Criteria--      1 OF 8  ¦ 12
___       21 ¦      Column Name      Level  Asc/Desc               ¦ 12
___       21 ¦   ------------------  -----  --------               ¦ 12
___       21 ¦   *************** TOP ***************               ¦ 12
___       21 ¦   ORDER_ID              _                           ¦ 12
___       21 ¦   CUST_ID              1_       A                   ¦ 12
___       21 ¦   ORDER_DATE           __       _                   ¦ 12
___       21 ¦   ORDER_TIME           __       _                   ¦ 12
___       21 ¦   FREIGHT_CHARGES      __       _                   ¦ 12
___       28 ¦   ORDER_SALESMAN       __       _                   ¦ 12
___       28 ¦   ORDER_POSTED_DATE    __       _                   ¦ 12
___       28 ¦   ORDER_SHIP_DATE      __       _                   ¦ 12
___       28 +-----------------------------------------------------+ 12
```

Figure 42. Specifying Sort Criteria

Once you specify the sort criteria, press ENTER or use END. Based on the sort criteria, the ORDERS rows are sorted in ascending order by customer ID, as shown in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 1 OF 19 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** *********************************** TOP ***********************************
___       216   00019  1997-02-24  12.12.51       48.52         SC012
___       207   00067  1997-02-24  12.12.51       48.52         WE012
___       206   00093  1997-02-24  12.12.51       48.52         SW012
___       217   00110  1997-02-24  12.12.51       48.52         SE012
___       288   00131  1997-02-24  12.12.51       48.52         SW012
___       209   00143  1997-02-24  12.12.51       48.52         SW012
___       205   00192  1997-02-24  12.12.51       48.52         NE012
___       210   00239  1997-02-24  12.12.51       48.52         NW012
___       211   00284  1997-02-24  12.12.51       48.52         SC012
___      3398   00289  1997-02-24  12.12.51       48.52         NW012
___       212   00327  1997-02-24  12.12.51       48.52         SC012
___       213   00371  1997-02-24  12.12.51       48.52         NE012
___       214   00415  1997-02-24  12.12.51       48.52         NC012
___       727   00420  1997-02-24  12.12.51       48.52         NE012
___       417   00448  1997-02-24  12.12.51       48.52         SE012
___       333   01210  1997-02-24  12.12.51       48.52         SE012
___       215   02221  1997-02-24  12.12.51       48.52         SE012
```

Figure 43. Sort Criteria Results

For additional information on the SORT (or SORT CRITERIA) command, see the *Command Reference Manual*, section for Primary Commands.

Maximum Fetch Limit

When the total number of rows that satisfy the criteria for a table exceeds the maximum fetch limit specified on the Editor and Display Options panel, Access displays a message indicating that all possible rows have not been retrieved.

COUNT Command

Use the COUNT command to determine the number of rows in a table that satisfy the search criteria. You can then change the selection criteria to retrieve fewer rows or use the MAX ROWS command to increase the number of rows fetched.

Note: The COUNT command is not available when browsing a Compare File.

MAX ROWS Command

To view rows in excess of the maximum fetch limit, use the MAX ROWS command to increase the maximum number of retrieved rows. For example, if the maximum fetch limit is 200 rows, and the COUNT command shows that there are 300 rows to view, specify MAX ROWS 300 and press ENTER to display the 300 rows.

Increase Maximum Fetch Limit

The MAX ROWS command increases the maximum fetch limit temporarily, until you terminate the edit or browse session. To increase the default maximum number of fetch rows, use the OPTIONS EDITOR command to display the Editor and Display Options panel. On this panel, you can change the specification for Maximum Fetch Rows.

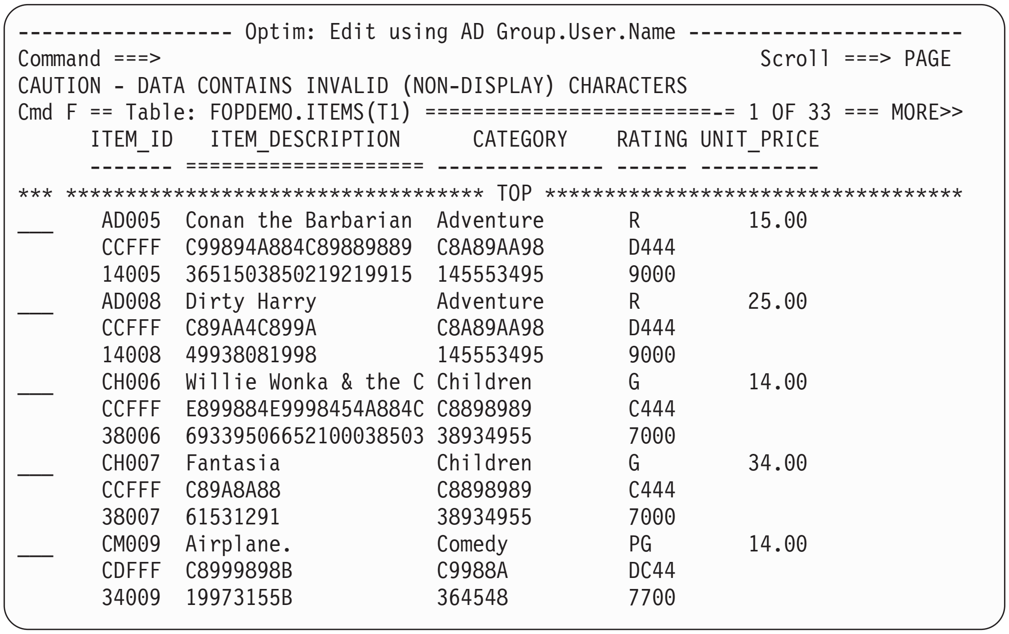
Display Hexadecimal Data

Under certain circumstances, you may need to display the hexadecimal (or hex) value of data during an edit or browse session.

For example, you may need to edit non-displayable characters, such as binary data. If a Warning indicates that the data contains non-displayable characters, a hexadecimal display may be useful.

You can use the HEX primary command to display each row of data in standard text format along with its hexadecimal format. The HEX command displays each row of data on three lines. One line is in the standard text representation of the data and two are used for the hexadecimal representation. The hexadecimal representation is shown as two digits directly under each EBCDIC character. The hexadecimal display remains in effect until you use the HEX command again to return the display to character representation.

In the following figure, note that the hexadecimal representation is displayed only for columns with character data type. The Warning in the figure refers to the character with a hexadecimal value of BB. This character is found in the ITEM_DESCRIPTION column, after the word “Airplane.”



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                   Scroll ===> PAGE
CAUTION - DATA CONTAINS INVALID (NON-DISPLAY) CHARACTERS
Cmd F == Table: FOPDEMO.ITEMS(T1) ========================-= 1 OF 33 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___    AD005  Conan the Barbarian  Adventure       R          15.00
       CCFFF  C99894A884C89889889  C8A89AA98       D444
       14005  3651503850219219915  145553495       9000
___    AD008  Dirty Harry          Adventure       R          25.00
       CCFFF  C89AA4C899A          C8A89AA98       D444
       14008  49938081998          145553495       9000
___    CH006  Willie Wonka & the C Children        G          14.00
       CCFFF  E899884E9998454A884C C8898989        C444
       38006  69339506652100038503 38934955        7000
___    CH007  Fantasia             Children        G          34.00
       CCFFF  C89A8A88             C8898989        C444
       38007  61531291             38934955        7000
___    CM009  Airplane.            Comedy          PG         14.00
       CDFFF  C8999898B            C9988A          DC44
       34009  19973155B            364548          7700
```

Figure 44. Hexadecimal Values Displayed

The same primary and line commands that are available in standard display are available in the hexadecimal display. Some commands, such as the FIND and CHANGE commands, have operands specifically for a hexadecimal display. For information on editing data while in hexadecimal display, refer to “Editing in Hexadecimal Mode” on page 77.

Note: Trailing blanks in CHAR columns are stored in the database and are displayed in hex mode as the hexadecimal value 40. Trailing blanks in VARCHAR columns are not stored in the database and are not displayed.

Wide Data Displays

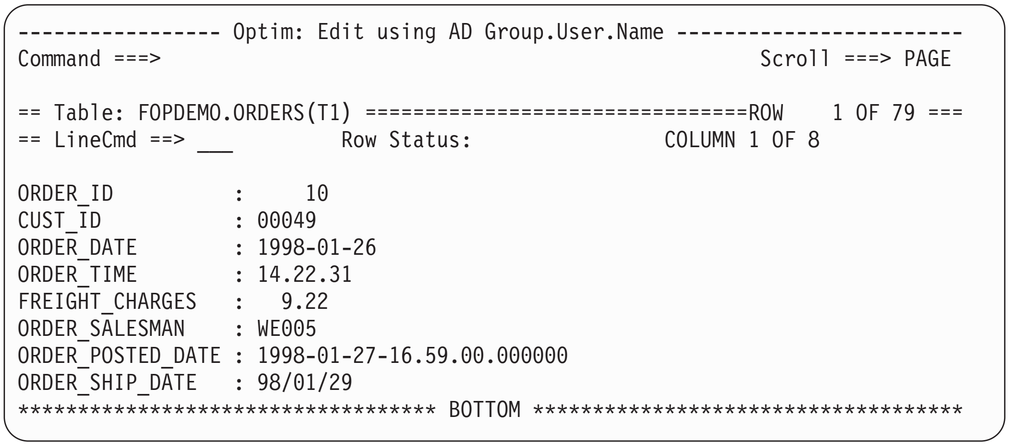
Data is sometimes too wide for display on the Access edit or browse display. A table column may be wider than the maximum display width or a table may include more columns than can be displayed at once. Several features and techniques help you edit or browse wide data easily.

Sidelabels Format

When editing or browsing tables containing multiple or wide columns, you may find it useful to use the sidelabels format instead of the standard columnar format. In sidelabels format, the panel displays one table row at a time. Each row of the panel displays the heading and data for a column. Generally, the sidelabels format displays more columns per panel and more data per row than columnar format.

To view a specific table in sidelabels format, use the Sidelabels primary or line command. When editing or browsing, you can also specify the sidelabels display format on the Editor and Display Options panel. Once specified, either by command or from the options panel, sidelabels format remains in effect until you change it.

The following figure shows a row from the ORDERS table in sidelabels format.



```
----------------- Optim: Edit using AD Group.User.Name ------------------------
Command ===>                                                  Scroll ===> PAGE

== Table: FOPDEMO.ORDERS(T1) ================================ROW    1 OF 79 ===
== LineCmd ==> ___         Row Status:               COLUMN 1 OF 8

ORDER_ID          :     10
CUST_ID           : 00049
ORDER_DATE        : 1998-01-26
ORDER_TIME        : 14.22.31
FREIGHT_CHARGES   :   9.22
ORDER_SALESMAN    : WE005
ORDER_POSTED_DATE : 1998-01-27-16.59.00.000000
ORDER_SHIP_DATE   : 98/01/29
*********************************** BOTTOM ***********************************
```

Figure 45. Sidelabels Format

SIDELABELS Primary Command

You can view a specific table in sidelabels format by typing the SIDELABELS primary command and either specifying a table name operand (e.g., SIDELABELS ITEMS or SID T3) or positioning the cursor on the table. If the cursor is positioned on a row, that row is displayed in sidelabels format; otherwise the first row in the table is displayed. If you do not specify a table name, the first row from the lowest-level table is displayed.

Note: When in sidelabels format, you can use the SIDELABELS command with a table name operand to display a different active table. During sidelabels display, joins between tables are retained and redisplayed when you return to columnar format.

To return to columnar format, type the SIDELABELS or SID primary command.

Sidelabels Line Command

To switch from columnar format to sidelabels format using the line command, type SID in **Cmd** for the desired row. The selected row is then displayed in sidelabels format. To return to columnar format, type SID on the command line.

Screen Elements

The screen elements in the columnar and sidelabels formats are the same: only the positioning is different. However, only one data row can be displayed at a time in sidelabels format. In addition to the elements included in columnar format, sidelabels format shows the position of the first displayed column and the total number of columns in the table. (Refer to Figure 19 on page 22 for a description of the screen elements.)

Available Commands

Except for the INDENT, JOIN, UNJOIN, START, and SWITCH commands, the primary and line commands that are available in columnar format are also available in sidelabels format. However, since only a single row from a single table is displayed in sidelabels format, certain commands are better suited to columnar format.

Scroll- Sidelabels

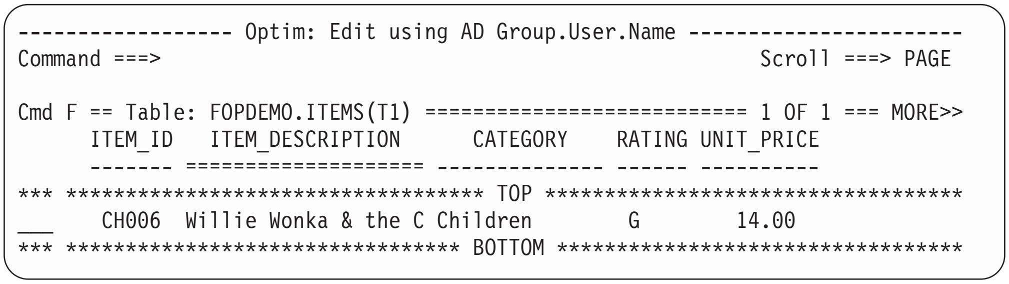
In sidelabels format, you can scroll using DOWN, UP, BOTTOM, and TOP to display another row in the table. You can also use LEFT and RIGHT to view additional columns in the same row. (See Scroll – Wide Tables for further discussion of horizontal scrolling in sidelabels format.)

Wide Columns

Data is sometimes too wide to be displayed on the panel. When the width of a column exceeds the user-specified maximum column display width, the data in that column is truncated. In columnar format, this truncation is indicated by equal signs (=) that separate the column heading from the data; in sidelabels format, an equal sign is displayed after the column heading.

Note: During an edit session, truncated data in a wide column is also protected.

In the following figure, the data column ITEM_DESCRIPTION is wider than the maximum column display width. Note the equal signs that separate the ITEM_DESCRIPTION heading from the data.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM ***********************************
```

Figure 46. Wide Column in Edit Session

The maximum number of characters that can be displayed per column is determined by the value specified as the **Columnar Max Display Width** and the **Sidelabel Max Display Width** on the Editor and Display Options panel. If the display of data from a column is truncated because of the maximum display width, several techniques allow you to view the complete data. You can:

- Use the OPTIONS EDITOR primary command to display the Editor and Display Options panel, where you can increase the maximum column display width.
- Use the EXPAND primary command to invoke a scrollable pop-up display of all data in the column.
- If in columnar format, use the SIDELABELS primary command or the SID line command to display a row of data in a vertical format that allows a maximum display width (of 50 to 32,767 characters per column) greater than the columnar format.

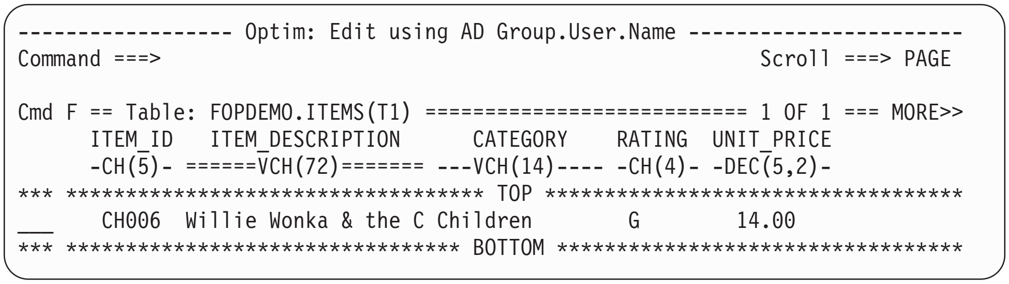
Display Attributes

It may be helpful to first review information about the displayed columns. The attributes of a column determine the type of editing you can perform on a column. For example, you cannot enter character data in a DECIMAL column, or null values in a NOT NULL column.

If the **Display Column Attributes** prompt on the Editor and Display Options panel is Y, column attributes are shown automatically. If attributes are not shown, use the ATTRIBUTES primary command to display information about data type, length, and nullability for each column in each table in the display.

Attributes can be displayed in any display mode (i.e., columnar, sidelabels, or hexadecimal mode). Attributes are displayed on the line after the column heading, when in columnar format. In sidelabels format, attributes are displayed before the column name. Note that, in sidelabels format, a column name is truncated if insufficient space is available to display both the attributes and the full column name.

In the following figure, attributes for each column are displayed directly beneath the column heading.



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING  UNIT_PRICE
      -CH(5)- ======VCH(72)======= ---VCH(14)---- -CH(4)- -DEC(5,2)-
*** *********************************** TOP ************************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM ***********************************
```

Figure 47. Column Attributes Displayed

The attribute information is in the form *type(n,n)*:N, where *type* is the data type and *n,n* is the width of the column. An N indicates the column is nullable. To conserve space on the screen, certain data type indicators are abbreviated when attributes are displayed. The following list shows the data type indicators and the DB2 data types that they represent.

Indicator	DB2 Data Type
CH	CHAR/ CHAR FOR MIXED DATA
VCH	VARCHAR
LVR	LONG VARCHAR
DEC	DECIMAL
INT	INTEGER
SMALLINT	SMALLINT
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
TIMESTMPZ	TIMESTAMP WITH TIME ZONE
SNGL FLOAT	SINGLE FLOAT
DBL FLOAT	DOUBLE FLOAT
GR	GRAPHIC

VGR VARGRAPHIC/
LONG VARGRAPHIC

CLOB CLOB FOR MIXED DATA

DBCLOB
DBCLOB

BIN BINARY

VARBIN
VARBINARY

BIGINT
BIG INTEGER

DECFLOAT
DECFLOAT

Note: See "Editing and Browsing with Unsupported Data Types" on page 69 for information on unsupported data types.

In Figure 47 on page 58, the attributes for the ITEM_DESCRIPTION column indicate it is 72 characters wide. Thus, you cannot display the entire column in columnar format by increasing the **Columnar Max Display Width** on the Editor and Display Options panel, which has a maximum width of 70.

To remove the column attributes from the display, use the ATTRIBUTES OFF command. You can also use the ATTRIBUTES command with no operand to toggle between displaying and not displaying column attributes.

Maximum Column Display Width

The maximum column display width is specified by the **Columnar Max Display Width** and the **Sidelabel Max Display Width** options on the Editor and Display Options panel. You can use the OPTIONS EDITOR command to display this panel. In columnar format, you can specify the maximum column display width in the range of 2 to 70 characters. In sidelabels format, you can specify the maximum column display width in the range of 50 to 32,767 characters. (See the *Common Elements Manual,* section for Editor and Display Options, for additional information.)

For VARCHAR columns, the entire width of the column, padded with trailing blanks, is displayed regardless of the actual length of the data (see Figure 50 on page 61). Therefore, when a table has several wide columns that do not contain correspondingly wide data, you may want to decrease the maximum display width so that more columns from the table are displayed per screen.

Expand Column

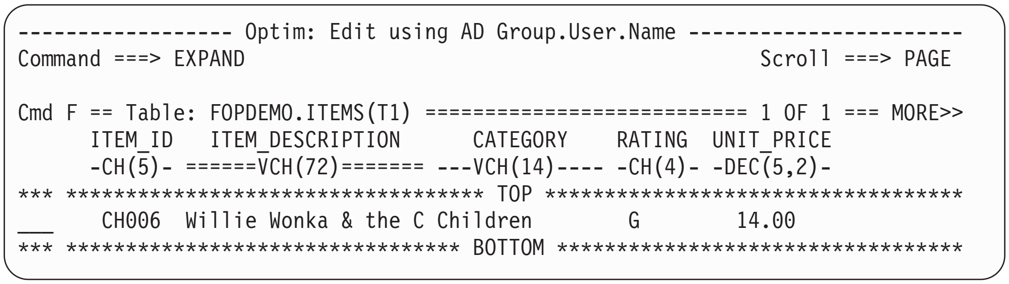
When the maximum display width does not provide sufficient display space for a specific column or when changing the maximum display width is undesirable, the EXPAND primary command can be used to display all data in that column. You can use this command on columns with data types of CHAR, VARCHAR, LONG VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, BINARY, VARBINARY, regardless of whether the column width exceeds the maximum display width. You can use this command on columns with data types of TIMESTAMP, TIMESTAMP WITH TIME ZONE, and DECIMAL only if the column width exceeds the maximum display width.

Note: During an edit session, the EXPAND command allows you to expand a protected column for editing.

When you enter the EXPAND command, you can either position the cursor on the desired table, regardless of level, or you can specify the column name. In a multi-table display, the EXPAND command operates on the lowest-level table by default. You can indicate the table in which the column you want to expand resides by including its tablename or identifier in the command. For example, to expand the

ITEM_DESCRIPTION column in the ITEMS table, type EXPAND ITEMS.ITEM_DESCRIPTION or EXPAND T1.ITEM_DESCRIPTION. You can also assign EXPAND to a program function key.

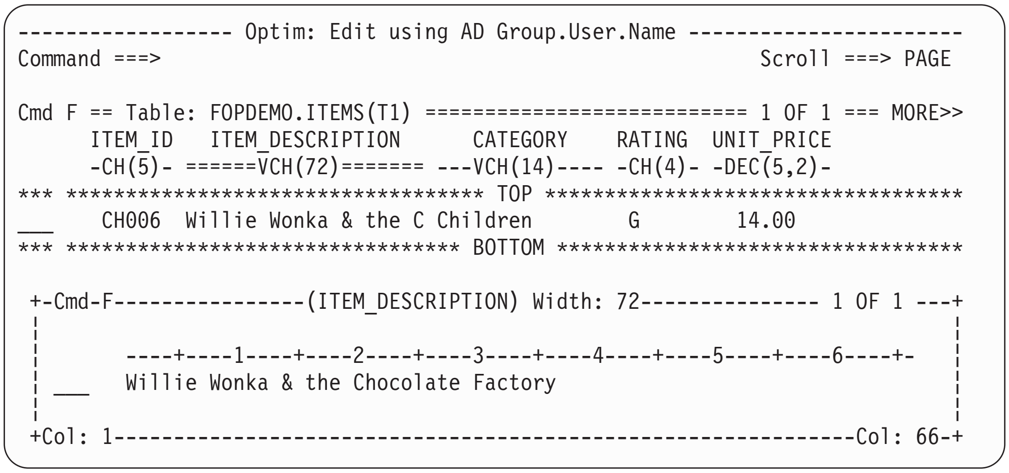
When trying to browse or edit a table with Optim, you may encounter an "Invalid name" error. This can be caused by invalid characters in the Creator ID or Table Name fields. To resolve the problem, on the command line type **EXPAND**, position the cursor on the field for Creator ID and hit Enter. Delete any characters after the field. Then Save and retry. Use the same procedure on the field for Table Name.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===> EXPAND                                           Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING  UNIT_PRICE
      -CH(5)- ======VCH(72)======= ---VCH(14)---- -CH(4)- -DEC(5,2)-
*** *********************************** TOP **********************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM *********************************
```

Figure 48. EXPAND When Column Exceeds Display Width

If the EXPAND command is entered at the **Command** prompt and the cursor is positioned on the ITEM_DESCRIPTION column of the item CH006, the following pop-up is displayed.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING  UNIT_PRICE
      -CH(5)- ======VCH(72)======= ---VCH(14)---- -CH(4)- -DEC(5,2)-
*** *********************************** TOP **********************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ********************************* BOTTOM *********************************

 +-Cmd-F----------------(ITEM_DESCRIPTION) Width: 72--------------- 1 OF 1 ---+
 |                                                                            |
 |      ----+----1----+----2----+----3----+----4----+----5----+----6----+-    |
 | ___  Willie Wonka & the Chocolate Factory                                  |
 |                                                                            |
 +Col: 1---------------------------------------------------------------Col: 66-+
```

Figure 49. Expanded Column Display

The column name and the actual width are shown in the header, as well as the position of the row and the number of rows in the edit or browse display.

In sidelabels format, positioning the cursor in a wrapped segment when executing the EXPAND command designates both the column to expand and the segment with which the expanded display begins.

You can overtype the expanded data or use available line commands. Refer to “Editing Protected Columns” on page 70 for information about the line commands available for editing expanded data.

Scroll- Expanded Column Display

The currently displayed column positions are indicated in the last border of the window. For example, if **Col: 1** and **Col: 66** are indicated in the last border of the window, characters 1 through 66 are displayed. When the column width exceeds 66 characters, the expanded data can be viewed by using the LEFT and RIGHT commands.

The following line commands can also be used:

>*nn* View next *nn* characters.

+*nn* View next *nn* characters.

<*nn* View previous *nn* characters.

-*nn* View previous *nn* characters.

Use the UP and DOWN commands to display the data in the expanded window by row. The row counter indicates the row that is currently displayed.

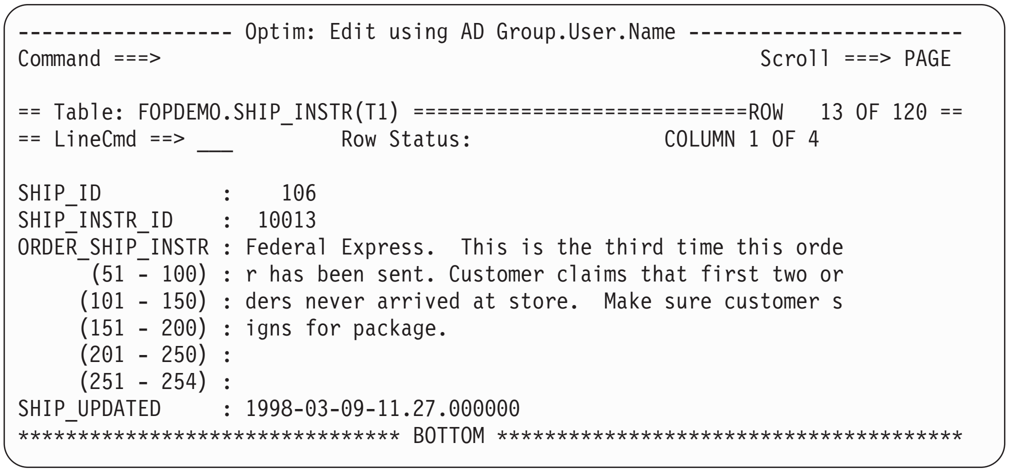
When you have finished editing or browsing the expanded data, use END to terminate the expanded column display and save your changes. Use CANCEL to terminate the expanded column display without saving your changes.

Wide Columns in Sidelabels Format

When a table contains many columns or wide columns, you may find it useful to use the sidelabels format instead of the standard columnar format. The sidelabels format displays more data per row, allowing the maximum number of characters per column to range from 50 to 32,767 characters.

When the maximum sidelabels display width permits, wide columns are automatically wrapped. The data is divided into 50-character segments and presented on multiple lines. If a column is wider than the maximum display width, it is displayed with an equal sign (=) after the column heading. As in columnar format, data in a truncated column can be displayed by using the EXPAND command or by increasing the maximum display width. (See "Editing Protected Columns" on page 70 for details about the EXPAND command.)

In the following figure, the maximum display width is 254 characters and data in the ORDER_SHIP_INSTR column is wrapped.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                Scroll ===> PAGE

== Table: FOPDEMO.SHIP_INSTR(T1) ============================ROW   13 OF 120 ==
== LineCmd ==> ___         Row Status:                COLUMN 1 OF 4

SHIP_ID          :    106
SHIP_INSTR_ID    :  10013
ORDER_SHIP_INSTR : Federal Express.  This is the third time this orde
      (51 - 100) : r has been sent. Customer claims that first two or
     (101 - 150) : ders never arrived at store.  Make sure customer s
     (151 - 200) : igns for package.
     (201 - 250) :
     (251 - 254) :
SHIP_UPDATED     : 1998-03-09-11.27.000000
****************************** BOTTOM ***************************************
```

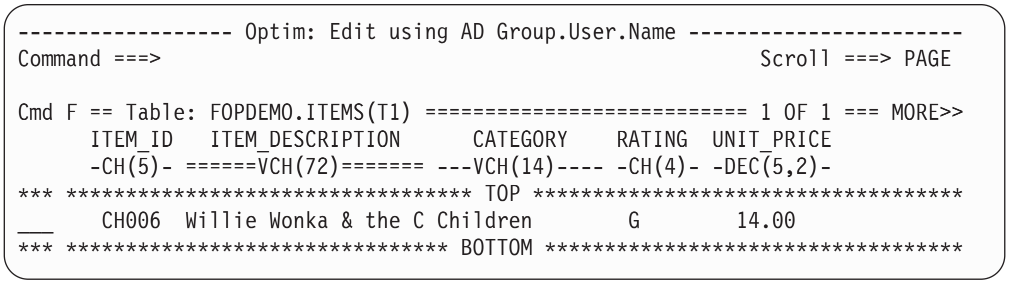
Figure 50. Column Wrapping

The first 50 characters of column data is displayed after the column name. The next five lines contain the remaining column data in 50-character segments. The beginning and ending character positions of each segment are displayed under the column heading.

Wide Tables

Many database tables have more columns than can be displayed at once in columnar format. In addition, columns may exceed the user-specified maximum width for columnar or sidelabels displays. For example, in the following figure, the table FOPDEMO.ITEMS has more columns than can be displayed on the screen at one time, as indicated by the MORE>> designation after the row count. Also, the column

ITEM_DESCRIPTION is wider than both the user-specified maximum column display width and the 70 character maximum limit for the columnar format parameter. (Note the length attribute under the ITEM_DESCRIPTION heading.)



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) =========================== 1 OF 1 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING  UNIT_PRICE
      -CH(5)- ======VCH(72)======= ---VCH(14)---- -CH(4)- -DEC(5,2)-
*** ********************************** TOP *************************************
___    CH006  Willie Wonka & the C Children        G        14.00
*** ******************************** BOTTOM ************************************
```

Figure 51. Wide Table in Edit Session

Editing or browsing may be difficult when you can view the data in no more than a few columns at a time. If the display of data in a table is truncated because of the screen width, several techniques and features make it possible for you to view the complete data for each row or to juxtapose columns of primary interest. You can:

- Use LEFT and RIGHT to scroll a columnar or sidelabels display to view additional columns.
- Use the LOCK (or ANCHOR) primary command to freeze one or more columns in a columnar or sidelabels display while scrolling the remaining columns. (The LOCK command and the ANCHOR command are synonyms.)
- Use the COLUMNS primary command to arrange columns in a columnar or sidelabels display so that the columns of interest are displayed together.
- If browsing data in columnar format, use the SIDELABELS primary command or the SID line command to display a row of data in a vertical format that allows a greater maximum display width (from 50 through 32,767 characters per column) and provides room for a greater number of columns.

Scroll – Wide Tables

In columnar format, a column is not displayed when the display width of the column exceeds the available space for display on the screen. For example, if the display width for a column is 20 characters, but only 10 spaces remain on the screen, the column is not displayed. You can view columns that are not displayed by using the LEFT and RIGHT commands (usually assigned to PF10 and PF11).

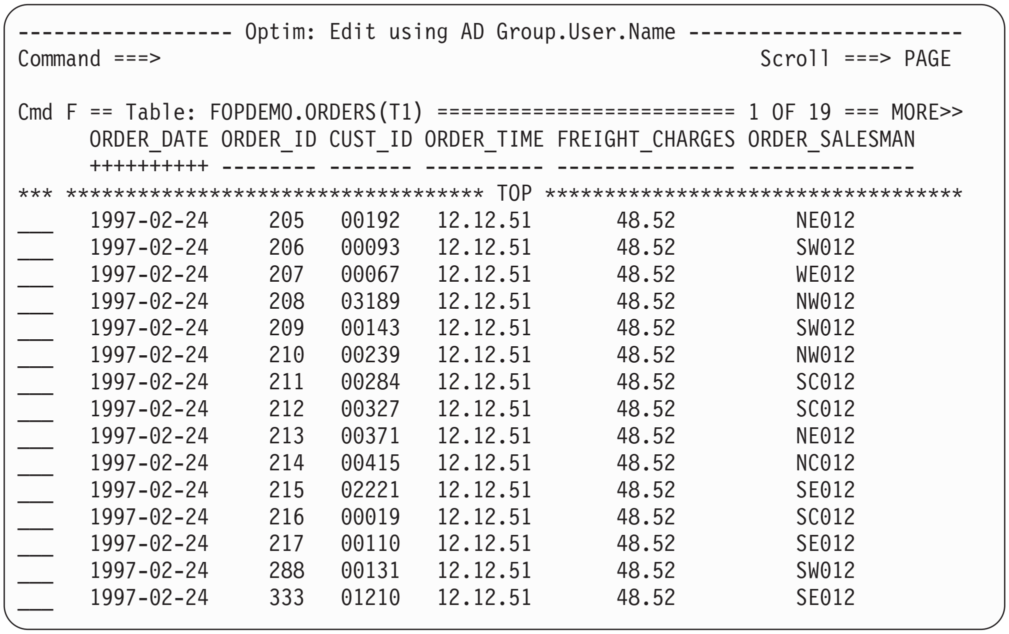
In sidelabels format, you can also use the LEFT and RIGHT commands to scroll through the columns in a table row. Use the RIGHT command to scroll the display to succeeding columns in the table row, and use the LEFT command to scroll the display to preceding columns in the table row.

Lock Columns

Use the LOCK (or the ANCHOR) command to position one or more columns at the beginning of the display. The locked columns are retained on the screen when you scroll the remaining columns.

For example, assume the combined width of the column data for the ORDERS table exceeds the width of the screen in columnar format. Thus, the display must be scrolled horizontally to view all columns. By locking the ORDER_DATE column, ORDER_DATE is repositioned at the beginning of each line and remains in that position on the screen when scrolling to view the remaining columns. In sidelabels format, locked columns are positioned as the first columns on the display.

In the following figure, assume the LOCK command has been entered to retain the ORDER_DATE column on the display.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ORDERS(T1) ========================= 1 OF 19 === MORE>>
      ORDER_DATE ORDER_ID CUST_ID ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      ++++++++++ -------- ------- ---------- --------------- --------------
*** ********************************** TOP **********************************
___   1997-02-24     205   00192   12.12.51       48.52          NE012
___   1997-02-24     206   00093   12.12.51       48.52          SW012
___   1997-02-24     207   00067   12.12.51       48.52          WE012
___   1997-02-24     208   03189   12.12.51       48.52          NW012
___   1997-02-24     209   00143   12.12.51       48.52          SW012
___   1997-02-24     210   00239   12.12.51       48.52          NW012
___   1997-02-24     211   00284   12.12.51       48.52          SC012
___   1997-02-24     212   00327   12.12.51       48.52          SC012
___   1997-02-24     213   00371   12.12.51       48.52          NE012
___   1997-02-24     214   00415   12.12.51       48.52          NC012
___   1997-02-24     215   02221   12.12.51       48.52          SE012
___   1997-02-24     216   00019   12.12.51       48.52          SC012
___   1997-02-24     217   00110   12.12.51       48.52          SE012
___   1997-02-24     288   00131   12.12.51       48.52          SW012
___   1997-02-24     333   01210   12.12.51       48.52          SE012
```

Figure 52. Repositioned Locked Column

Identify Locked Columns

In columnar format, a locked column is identified by a series of plus signs (+) under the column heading. The first locked column is displayed as the first column, regardless of how the columns are actually ordered in the table. Each succeeding locked column is placed after the previously locked columns.

A column that exceeds the maximum column display width is indicated by equal signs (=) under the column heading. If the truncated column is also locked, a series of dots under the column heading distinguish it from locked columns that are not truncated. (For more information on truncated columns, see Wide Columns in this section.)

In sidelabels format, the first locked column is displayed on the first row. Each succeeding locked column is placed after the previously locked columns. A locked column is marked with a plus sign (+) after the column name.

Lock Multiple Columns

In columnar format, any number of columns may be locked, as long as enough space remains to display at least one unlocked column. The number of positions reserved for unlocked columns is determined by the user-defined maximum display width specified on the Editor and Display Options panel. For example, if the maximum display width is 20 characters, then the last 21 character positions in each row are reserved for unlocked columns to ensure that at least one unlocked column is displayed. More columns may be displayed if they fit, in their entirety, in the remaining area.

Use the LOCK KEY command to lock all columns that comprise the primary key.

Unlock Columns

Use the UNLOCK command with no operands to unlock all locked columns. Use the UNLOCK command with a column name operand to unlock a single column.

Arrange Columns

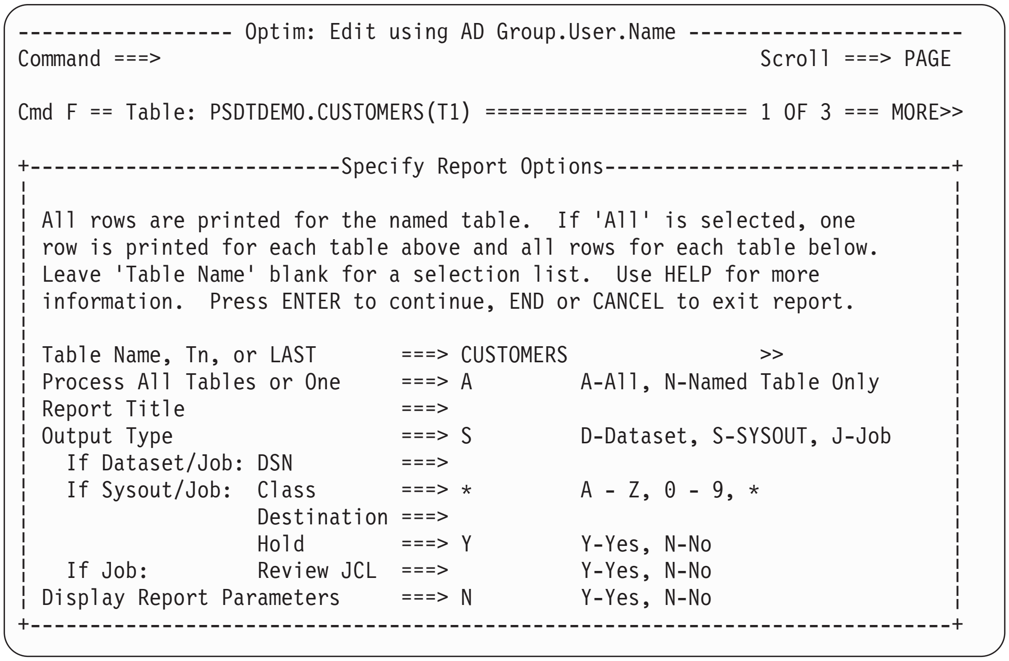
To rearrange the display order of the columns, use the COLUMNS command. The COLUMNS command displays the Describe Columns panel, which is described in the *Common Elements Manual*, section on Manage Data Displays.

On the Describe Columns panel, you can change the order of columns by using the Move (M) line command with the Before (B) and After (A) commands to designate the destination.

Reporting

At any time during an edit or browse session, you can produce a report about the session.

Use the reporting facility to generate a report that includes some or all retrieved rows in a variety of formats. To invoke the report facility, use the REPORT command. Access prompts for the report specifications on the Specify Report Options panel.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: PSDTDEMO.CUSTOMERS(T1) ====================== 1 OF 3 === MORE>>

+--------------------------Specify Report Options-----------------------------+
¦                                                                             ¦
¦ All rows are printed for the named table.  If 'All' is selected, one        ¦
¦ row is printed for each table above and all rows for each table below.      ¦
¦ Leave 'Table Name' blank for a selection list.  Use HELP for more           ¦
¦ information.  Press ENTER to continue, END or CANCEL to exit report.        ¦
¦                                                                             ¦
¦ Table Name, Tn, or LAST       ===> CUSTOMERS                >>              ¦
¦ Process All Tables or One     ===> A         A-All, N-Named Table Only      ¦
¦ Report Title                  ===>                                          ¦
¦ Output Type                   ===> S         D-Dataset, S-SYSOUT, J-Job     ¦
¦   If Dataset/Job: DSN         ===>                                          ¦
¦   If Sysout/Job:  Class       ===> *         A - Z, 0 - 9, *                ¦
¦                   Destination ===>                                          ¦
¦                   Hold        ===> Y         Y-Yes, N-No                    ¦
¦   If Job:         Review JCL  ===>           Y-Yes, N-No                    ¦
¦ Display Report Parameters     ===> N         Y-Yes, N-No                    ¦
+-----------------------------------------------------------------------------+
```

Figure 53. Specify Report Options

Panel Prompts

The prompts on the panel are:

Table Name, Tn or LAST

The name of the table that directs the report. You can specify the name of any table in the display by explicit name, assigned T*n* or V*n*, or the word LAST (for the lowest-level displayed table). Leave blank to choose from a selection list.

This specification, combined with **Process All Tables or One**, determines the set of data in the report.

For more information about the effect of this specification on the report contents, refer to “Determining Report Contents” on page 65.

Process All Tables or One

Indicate whether one or all tables are included in the report. Specify:

A All tables.

N Only the named table.

Report Title
Specify a title to be included in the heading for the report. Any characters, including uppercase and lowercase letters, are valid.

Output Type
Indicate the output type. Specify:

D Dataset

S SYSOUT

J Job (a batch job that produces both Dataset and SYSOUT output)

If Dataset/Job
If the **Output Type** is D or J, specify the name of the dataset. If the dataset does not exist, Access prompts for allocation information and allocates the file before generating the report.

If SYSOUT/Job
If the **Output Type** is S or J, specify the SYSOUT parameters:

Class The output class for the printed output. Specify an alphabetic or numeric character (A – Z, 0 – 9) or an asterisk (*).

Destination
The SYSOUT destination. Specify a valid local or remote terminal, a node in the JES network, a local or remote printer or workstation, or a TSO User ID.

Hold Specify Y or N.

If Job: Review JCL
If the **Output Type** is J, specify:

Y Review JCL.

N Do not review JCL.

Display Report Parameters
Indicates whether you want to review the prompts for report parameters. (Refer to "Report Format Parameters" on page 66 for details.) Specify:

Y Review prompts.

N Do not review prompts.

Determining Report Contents

The **Table Name** and **Process All Tables or One** specifications determine the contents and layout of the report. For example, assume you are currently editing four tables joined for display in the following order:

```
CUSTOMERS
  C:ORDERS
    C:DETAILS
      P:ITEMS
```

1. To generate a report on the rows from a single table, specify the table name and "N" for **Process All Tables or One**. For example, to include only the DETAILS rows in the report, specify:

   ```
   Table Name, Tn, or LAST   ===> DETAILS
   Process All Tables or One    ===> N
   All the rows in the DETAILS table, regardless of the ORDERS rows
   (the next highest table) that are displayed, are included in the report.  No
   other tables are included in this report.
   ```

   a. To limit the report to a specific level, specify the name of the table for which all rows on the current level are included. For reference, the related row from each higher level table is included. The rows from lower-level tables are always included with this specification.

For example, to report on the DETAILS rows for the currently displayed ORDERS row and the related ITEMS rows for each of the DETAILS rows, specify:

```
Table Name, Tn, or LAST   ===> DETAILS
Process All Tables or One    ===> A
```

The report lists the displayed row for the CUSTOMERS and ORDERS table and each DETAILS row, one at a time, along with the related ITEMS rows for each DETAILS row.

b. To include the rows from all the tables in the current session, specify the name of the Start Table and "A" for **Process All Tables or One**. For example, assume the CUSTOMERS table is the Start Table, specify:

```
Table Name, Tn, or LAST   ===> CUSTOMERS
Process All Tables or One    ===> A
```

If you want to include all information for a single CUSTOMERS row, specify ORDERS as the table name. Then, the current CUSTOMERS row is included with all current ORDERS rows, related DETAILS, and ITEMS rows.

Format

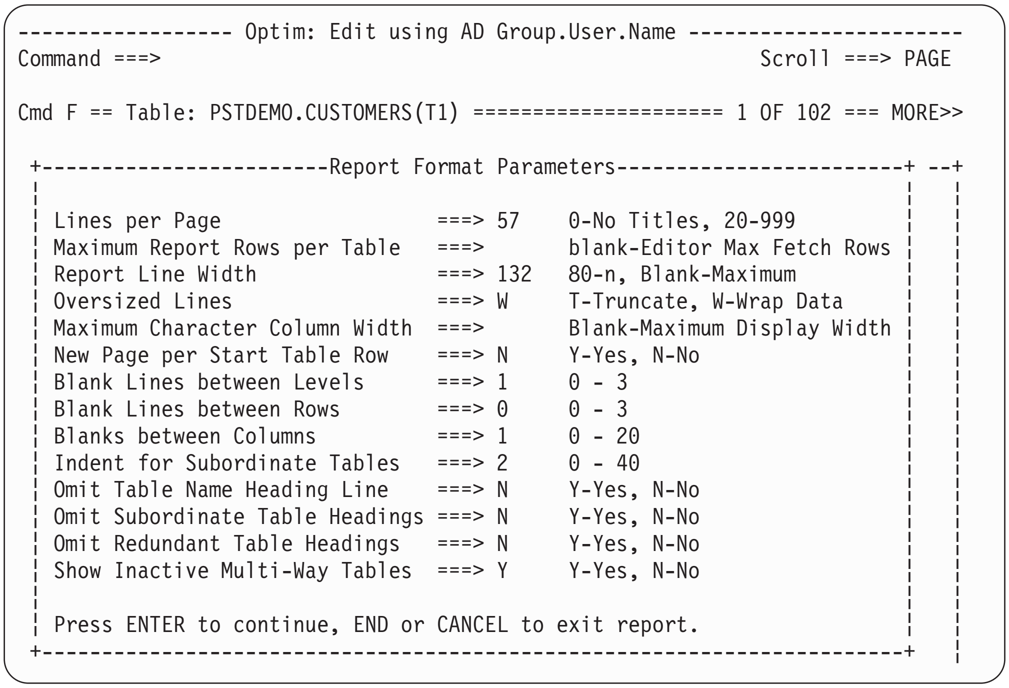
The format of the rows in the report conforms to the display format in the editor. A header contains the table name and column headings, and the data is provided after the header. You can specify the parameters for the report format.

When there are multiple subordinate tables, the report display is similar to that of the editor.

- For each table that is higher than the lowest-level table, only one row is displayed.
- For the lowest-level table, all rows are displayed.
- When multiple rows are included in the report for a higher-level table, each row is displayed in order with the related, lower-level rows.

Report Format Parameters

When you specify Y to **Display Report Parameters**, prompts for the parameters are displayed. Normally, you establish these parameters once, and use them consistently for all reporting.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
Command ===>                                                Scroll ===> PAGE

Cmd F == Table: PSTDEMO.CUSTOMERS(T1) ==================== 1 OF 102 === MORE>>

 +------------------------Report Format Parameters------------------------+ --+
 |                                                                        |   |
 | Lines per Page                 ===> 57    0-No Titles, 20-999          |   |
 | Maximum Report Rows per Table  ===>       blank-Editor Max Fetch Rows  |   |
 | Report Line Width              ===> 132   80-n, Blank-Maximum          |   |
 | Oversized Lines                ===> W     T-Truncate, W-Wrap Data      |   |
 | Maximum Character Column Width ===>       Blank-Maximum Display Width  |   |
 | New Page per Start Table Row   ===> N     Y-Yes, N-No                  |   |
 | Blank Lines between Levels     ===> 1     0 - 3                        |   |
 | Blank Lines between Rows       ===> 0     0 - 3                        |   |
 | Blanks between Columns         ===> 1     0 - 20                       |   |
 | Indent for Subordinate Tables  ===> 2     0 - 40                       |   |
 | Omit Table Name Heading Line   ===> N     Y-Yes, N-No                  |   |
 | Omit Subordinate Table Headings ===> N    Y-Yes, N-No                  |   |
 | Omit Redundant Table Headings  ===> N     Y-Yes, N-No                  |   |
 | Show Inactive Multi-Way Tables ===> Y     Y-Yes, N-No                  |   |
 |                                                                        |   |
 | Press ENTER to continue, END or CANCEL to exit report.                 |   |
 +------------------------------------------------------------------------+   |
```

Figure 54. Report Format Parameters

Panel Prompts

The following prompts are displayed on this panel:

Lines per Page

The maximum number of lines printed on a page. Normally, this is the number of lines that physically fit on a page, excluding lines for the margins. The default is 57. Specify:

0 Print the title line at the start of the report and do not insert page breaks. Use 0 to write to a dataset without inserting page breaks.

20-999 Print the indicated number of lines per page. When the number of lines have been printed, Access inserts a page break, prints the title at the beginning of the new page, and increments the page number by 1.

Maximum Report Rows per Table

The maximum number of rows fetched for each table in the report. The value must be from 1 through the fetch limit specified by site management. Leave blank to use the value specified for **Maximum Fetch Rows** in the **Editor and Display Options**.

Report Line Width

The maximum number of characters on a line in the report. If a row exceeds this value, it is wrapped or truncated according to the **Oversized Lines** specification.

The minimum value is 80. A value less than 80 is automatically reset to 80. Use 80 if the report is directed to a data set for subsequent browsing or printing. Leave blank to set the line width equal to the longest row of any table in the report.

If the report is directed to a printer, this value is normally the line length of the printer (132 for most printers), excluding the carriage control byte in the first position of the line.

Oversized Lines

Indicates the handling of lines that exceed the maximum width. Specify:

T Data is truncated. Any columns that do not fit entirely on the line are omitted.

W Data is wrapped. Each row occupies as many lines as needed. Line breaks occur between columns when the column width exceeds the remaining line width.

Maximum Character Column Width

The maximum number of bytes that can be included in a report column.

Note: For GRAPHIC and DBCLOB columns, the Shift Out and Shift In characters are included in the maximum width. Therefore, a report displays the specified maximum number of GRAPHIC and DBCLOB bytes minus the number of Shift Out and Shift In characters.

Leave blank to use the value specified for **Columnar Max Display Width** on the Editor and Display Options panel.

New Page per Start Table Row

Indicator for a new page for each Start Table row. (If the report is for a single table, this is ignored.) Specify:

Y Print on a new page.

N Continue printing on the current page.

Blank Lines between Levels

Indicates the number of blank lines inserted between table levels in the report. Specify a value from 0 through 3.

Blank Lines between Rows
The number of blank lines inserted between rows. This parameter applies only to the tables for which all relevant rows are included; only a single line is included for other tables. Specify a value from 0 through 3.

Blank between Columns
Indicates the number of blanks inserted between columns. Specify a value from 0 through 20.

This value is the minimum number of blanks between any two columns. The report may display more blanks between some columns, depending on the contents of the column and the justification of that data in relation to the column heading.

Indent for Subordinate Tables
The number of positions to indent data from a subordinate table at each control break to distinguish it from the immediately preceding table. Use this indentation to present a clear visual representation of the display levels.

Specify a value from 0 through 40. This value is subtracted from the **Report Line Width** to determine the actual number of characters in the indented table that can fit on a line.

Omit Table Name Heading Line
Indicator for including the line containing the table name and level number for each level in the report. Specify:

Y Omit heading.

N Include heading.

Omit Subordinate Table Headings
Indicator for including header lines for all tables other than the Start Table in the report. If the subordinate table names and column titles are not necessary, use this option to eliminate them. Specify:

Y Omit heading.

N Include heading.

Omit Redundant Table Headings
Indicator for including table headings more than once on a page. That is, if a table is included on a page more than once, the headings are included with the first occurrence only. Specify:

Y Omit redundant table headings.

N Include table headings for all occurrences of a table on a page.

Show Inactive Multi-Way Tables
Indicator for including all joined tables in a multi-way join. Although all tables cannot be currently active, all must have been displayed during the session. Specify:

Y Include all tables in a multi-way join.

N Include only the currently displayed table.

Additional Notes

The following notes apply to reports:

- Excluded rows are included in the report and are redisplayed when you return to the session after generating a report.
- If you specify a table that is involved in a multi-way join and only one table is printed, only the active table is included in the report. The hidden tables on that level are not printed, regardless of the **Show Inactive Multi-Way Tables** option.
- All locked column specifications are respected. Therefore, locked columns are positioned as in the edit or browse session.

- The specifications on the Describe Columns panel for column and label handling are used to format the report lines.
- The REPORT command is not available in a zoomed display.

Editing and Browsing with Unsupported Data Types

During an Edit or Browse session, DB2 tables that contain columns with unsupported data types are managed as follows.

Edit

When using the Edit function, if the table contains any columns with unsupported data types, the Edit function is disabled and the message “UNSUPPORTED DATA TYPE” appears.

Browse

When using the Browse function,

- If the table contains columns with both supported and unsupported data types, the supported data types are displayed and the unsupported data types are listed as “UNSUPPORTED.”
- If the table contains columns with unsupported data types only, the Browse function is disabled and the message “UNSUPPORTED DATA TYPE” appears.

Editing Data

This section discusses how to edit data, including typing over the data, and using line commands and primary commands.

Status Flags

The result of all editing activity is identified by the status flag, labeled **F**, on the display. The possible values are:

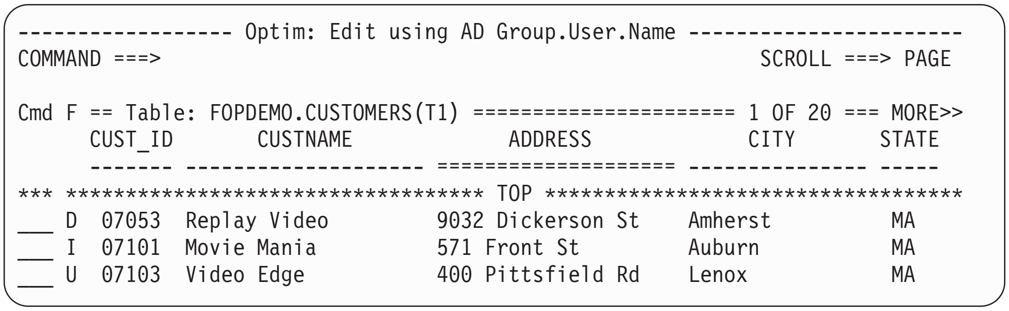
D The row has been deleted.

I The row has been inserted.

U The row has been updated.

E The row is in error.

For example, in the following screen segment the first row was deleted, the second was inserted, and the third was updated.



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
COMMAND ===>                                                   SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS              CITY       STATE
      ------- -------------------- =================== --------------- -----
*** *********************************** TOP ************************************
___ D  07053  Replay Video         9032 Dickerson St    Amherst          MA
___ I  07101  Movie Mania          571 Front St         Auburn           MA
___ U  07103  Video Edge           400 Pittsfield Rd    Lenox            MA
```

Figure 55. Editing Status Flags

The value in **F** reflects the last action to affect a row. For example, if you insert a row, the value in **F** is I. If you later update the same row, the value U replaces the I.

Various ways to edit data are discussed in the following sections.

Overtyping Data

If the entire column is displayed and the data is not defined as read-only, you can type over it.

Read-Only Data

Data in a column is read-only, and cannot be overtyped if:

- The DB2 specification defines the column as read-only. For example, if the column is displayed as part of a joined view, consists of an expression, or is a DB2 Catalog column.
- The column is defined as read-only on the Describe Columns panel (displayed by using the COLUMNS command).

If one or both conditions are true, Access protects the data and you cannot edit it. However, the data is not protected for any condition that Access cannot detect. For example, Access may allow a user to edit a column that the user does not have explicit authority to edit. If an attempt is made to modify such data, a DB2 error message is displayed.

Editing Protected Columns

When the actual data length for a column exceeds the user-specified maximum column display width, values in that column are protected and cannot be edited directly. The column heading of a protected column is separated from the data by a series of equal signs (=) instead of by a series of dashes (-).

For example, the ADDRESS column is defined to DB2 as 50 characters, which exceeds the maximum display width of 20 characters defined on the Editor and Display Options panel. To edit the ADDRESS column, you can change the column display width or expand the column.

Change Maximum Display Width

You can change the maximum column display width on the Editor and Display Options panel. Use the OPTIONS EDITOR command to invoke the Editor and Display Options panel. In columnar format, the maximum display width must be a value from 2 through 70 characters. In sidelabels format, the maximum display width must be a value from 50 through 32,767 characters. (See the *Common Elements Manual*, section on Editor and Display Options for additional information.)

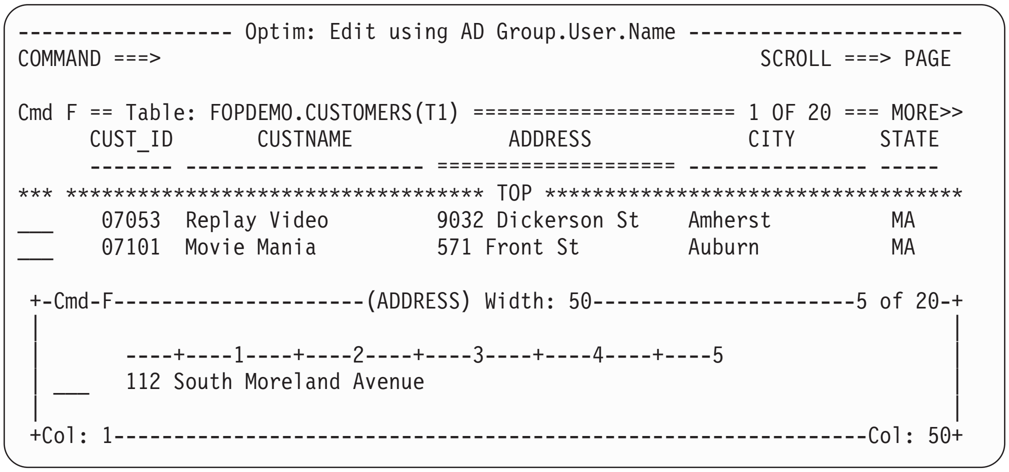
EXPAND Command

When the maximum display width is insufficient for a column, or changing the maximum display width is undesirable, you can use the EXPAND primary command to display data in the column.

You can use the EXPAND command on columns with data types of CHAR, VARCHAR, LONG VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, BINARY, VARBINARY, regardless of whether the column width exceeds the maximum display width. However, the EXPAND command can be used on columns with data types of TIMESTAMP and DECIMAL only if the column width exceeds the maximum display width.

You can enter the EXPAND command and either position the cursor on the desired table, regardless of level, or specify the column name. In a multi-table display, the EXPAND command operates on the lowest-level table by default. You can indicate the table in which the column you want to expand resides by including its tablename or identifier in the command. Press ENTER to display the expanded column. You can also assign EXPAND to a program function key.

The expanded column data is displayed in a pop-up window or full screen as specified on the User Options panel. In the following example, a pop-up window is displayed when you enter the EXPAND command at the **Command** prompt and position the cursor on the ADDRESS column for the customer named Video Way.



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ********************************** TOP ***********************************
___    07053  Replay Video        9032 Dickerson St    Amherst          MA
___    07101  Movie Mania         571 Front St         Auburn           MA

 +-Cmd-F---------------------(ADDRESS) Width: 50----------------------5 of 20-+
 |                                                                            |
 |       ----+----1----+----2----+----3----+----4----+----5                   |
 | ___   112 South Moreland Avenue                                            |
 |                                                                            |
 +Col: 1---------------------------------------------------------------Col: 50+
```

Figure 56. Expanded Column Display Pop-Up Window

The column name and width and a row count indicator are displayed in the border before the expanded column data.

Scrolling

The starting and ending positions in the display are indicated in the border after the expanded column data. When the column width exceeds the screen width, you can scroll the expanded data horizontally using the LEFT and RIGHT commands. Use the UP and DOWN commands to scroll the data in the expanded window by row. The row counter displayed in the border before the expanded column data indicates which row is currently displayed.

Editing Expanded Column Data

You can overtype the data in the expanded window directly. Several line commands are available to edit expanded character data, including:

>*nn* Display next *nn* characters.

+*nn* Display next *nn* characters.

<*nn* Display previous *nn* characters.

-*nn* Display previous *nn* characters.

Hex Display and edit hexadecimal value of the data. Available only for character data types. Hexadecimal is useful for editing any non-displayable characters that are protected in text mode and can only be modified by typing over the hexadecimal value. A full-screen hexadecimal display mode is also available (see “Editing in Hexadecimal Mode” on page 77 for details).

LC Translate to lowercase.

UA Undo all changes.

UC Translate to uppercase.

UL Undo the last change.

Update Flags

When editing is complete, use END to terminate the expanded column display and save your changes. As with any editing, a modified row is indicated by a U (Update) for the status flag.

Adding Rows

You can add rows using the following line commands:

C[*n*] or CC
Copy

R[*n*] or RR
Repeat

RP[*n*] or RRP
Repeat Pending

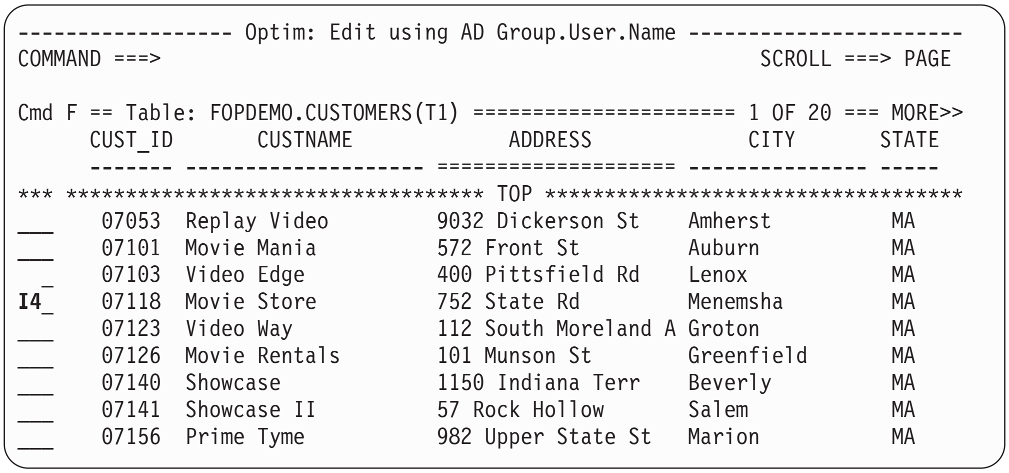
I[*n*] Insert

The Copy, Repeat, and Repeat Pending line commands duplicate source rows. The Repeat Pending line command repeats rows and leaves them in insert-pending status until you press ENTER. This line command allows you to work with partitioned indexes to repeat and edit rows before inserting them into the database.

The Insert line command adds the specified number of rows for data entry. The displayed columns in the inserted rows are empty and unprotected to allow you to enter data.

If all columns do not fit on the screen, you can scroll the display horizontally. Based on options for default column values, Access inserts values in any columns that are not directly edited or prompts the user for input for columns that are not defined to DB2 as candidates for a default value. The values that may be automatically inserted, based on the column data type and DB2 defaults, include a user-defined value, NULL, spaces, zeros, current date, and current time. (For more information on setting the option that controls this behavior, see the *Common Elements Manual* section for Editor and Display Options.)

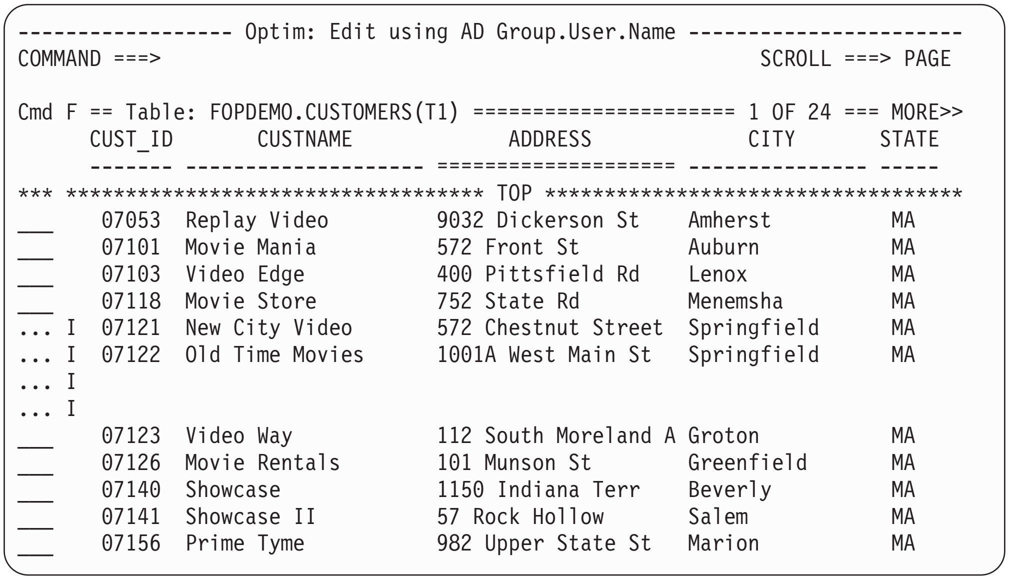
The I line command is used in the following example to insert four lines in the CUSTOMERS table.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          572 Front St         Auburn           MA
  _    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
I4_    07118  Movie Store          752 State Rd         Menemsha         MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07156  Prime Tyme           982 Upper State St   Marion           MA
```

Figure 57. Using Insert Line Command

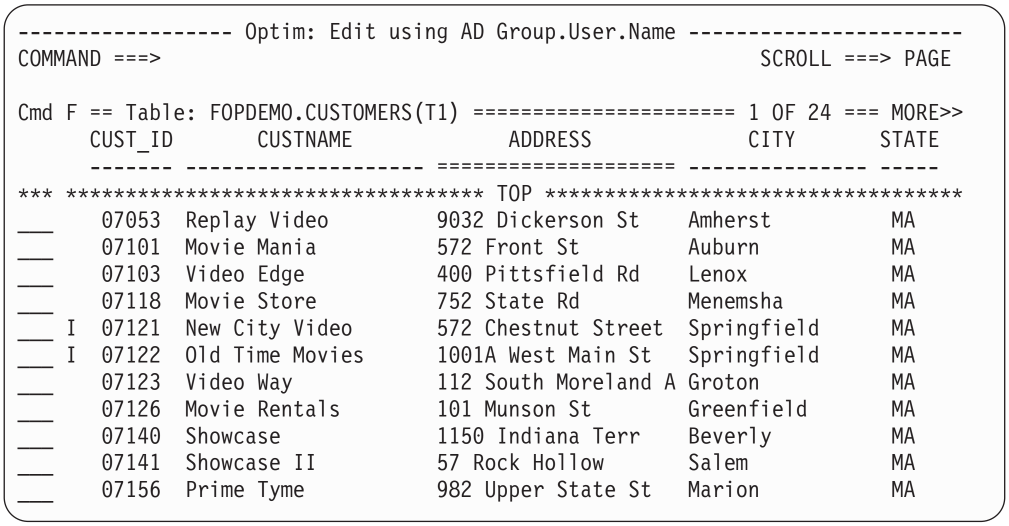
Access displays four blank lines for data entry. In the following example, data has been entered on the first two inserted lines and the third and fourth remain blank.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                   SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 24 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          572 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
___    07118  Movie Store          752 State Rd         Menemsha         MA
... I  07121  New City Video       572 Chestnut Street  Springfield      MA
... I  07122  Old Time Movies      1001A West Main St   Springfield      MA
... I
... I
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07156  Prime Tyme           982 Upper State St   Marion           MA
```

Figure 58. Data Typed on Inserted Lines

Access retains only lines on which data is entered. If you press ENTER, only the first two inserted lines are retained. The line command area for insert-pending lines is protected until the row has been successfully inserted in the database.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                   SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 24 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          572 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
___    07118  Movie Store          752 State Rd         Menemsha         MA
___ I  07121  New City Video       572 Chestnut Street  Springfield      MA
___ I  07122  Old Time Movies      1001A West Main St   Springfield      MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07156  Prime Tyme           982 Upper State St   Marion           MA
```

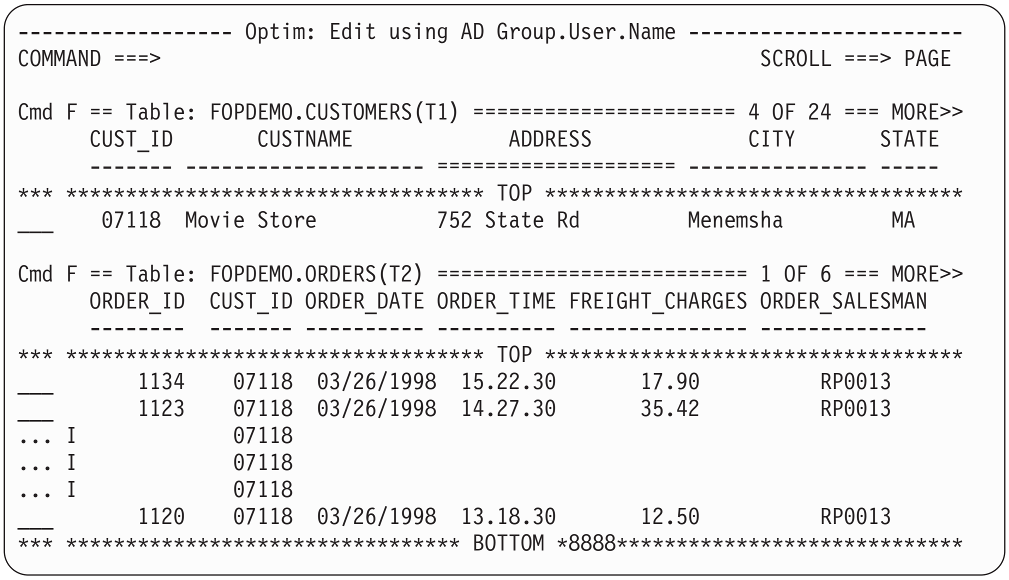
Figure 59. I to Identify Inserted Lines

An I (Insert) status flag indicates rows added to the table, including copied and repeated rows.

Multiple Tables

If you insert rows in a lower-level table during a multi-table display, Access automatically inserts the appropriate value in the foreign key columns (if using a DB2 relationship) or in the corresponding columns (if using an Optim relationship).

In the following figure, the ORDERS table is joined to the CUSTOMERS table, and three rows have been inserted in the ORDERS table using the I line command.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 4 OF 24 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07118  Movie Store          752 State Rd         Menemsha         MA

Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 6 === MORE>>
      ORDER_ID  CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      --------  ------- ---------- ---------- --------------- --------------
*** *********************************** TOP ***********************************
___       1134    07118  03/26/1998  15.22.30       17.90          RP0013
___       1123    07118  03/26/1998  14.27.30       35.42          RP0013
... I             07118
... I             07118
... I             07118
___       1120    07118  03/26/1998  13.18.30       12.50          RP0013
*** ********************************* BOTTOM *8888*****************************
```

Figure 60. Data Inserted when Adding Rows

Access automatically inserts the foreign key or corresponding column value in the CUST_ID column of the ORDERS table. The cursor is positioned to the first modifiable data column.

Rows in Error

A row is not inserted into the database when doing so causes an error condition. An E (Error) status flag indicates an error condition. The error must be resolved before the row can be inserted into the database. For example, if an added row causes a duplicate value in a unique index, the row is marked as an error.

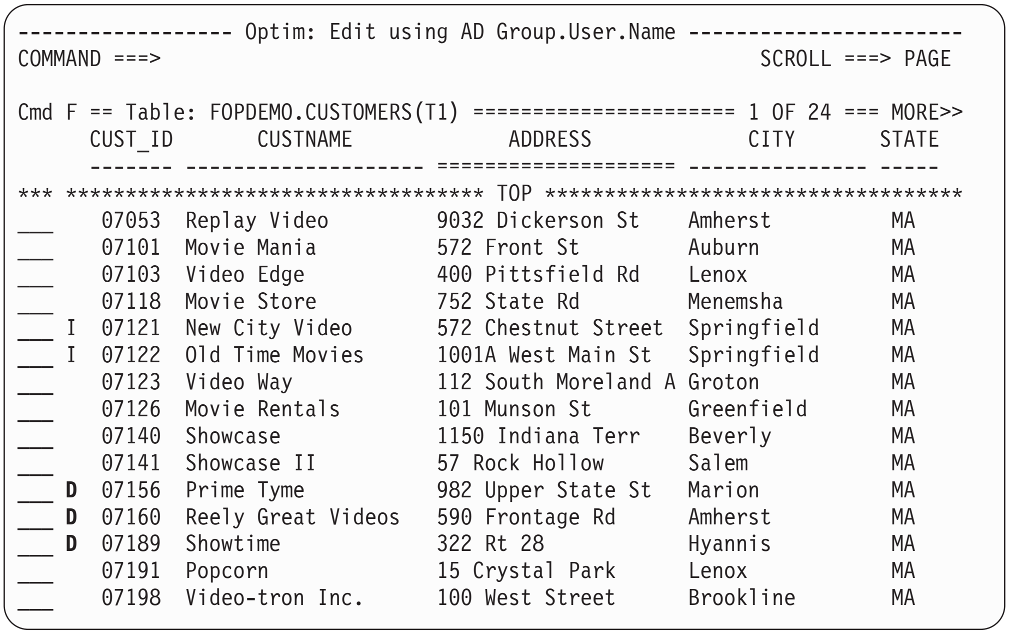
You can use the FIND ERROR command to locate and scroll to the error. You can display additional information about a specific error by positioning the cursor on the line containing the error and pressing the HELP key or entering the HELP command.

Deleting Rows

You can delete rows using the Delete line command or the DELETE primary command.

Access passes the delete request to DB2, which processes the delete operation according to the Referential Integrity rules defined to DB2 (see “DELETE Rules” on page 76).

A ‘D' (Delete) status flag indicates deleted rows, as in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 24 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          572 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
___    07118  Movie Store          752 State Rd         Menemsha         MA
___ I  07121  New City Video       572 Chestnut Street  Springfield      MA
___ I  07122  Old Time Movies      1001A West Main St   Springfield      MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___ D  07156  Prime Tyme           982 Upper State St   Marion           MA
___ D  07160  Reely Great Videos   590 Frontage Rd      Amherst          MA
___ D  07189  Showtime             322 Rt 28            Hyannis          MA
___    07191  Popcorn              15 Crystal Park      Lenox            MA
___    07198  Video-tron Inc.      100 West Street      Brookline        MA
```

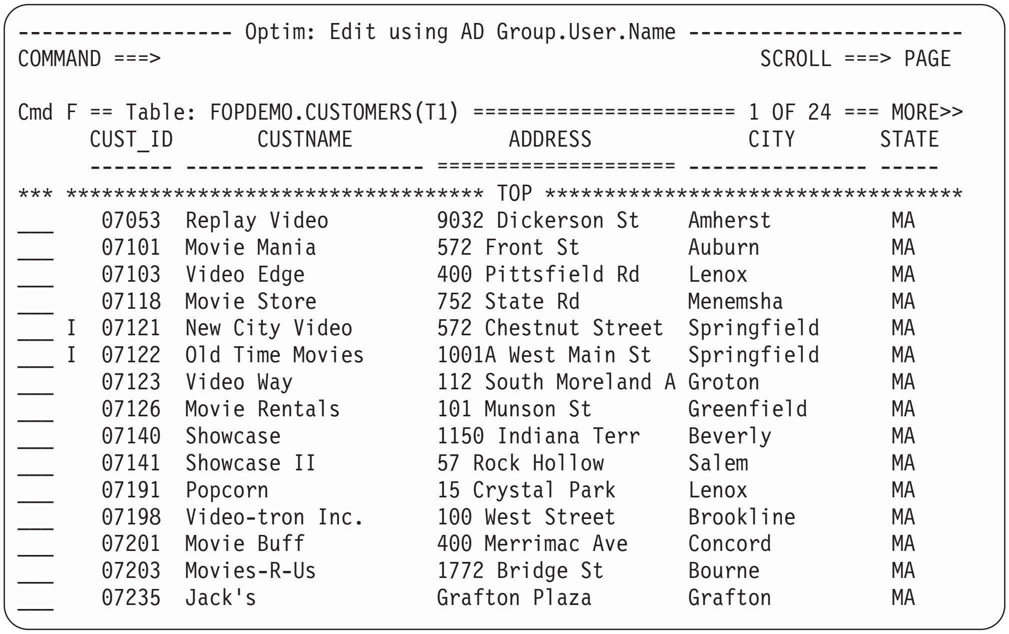
Figure 61. Deleted Rows Identified by D

By retaining deleted rows in the display, Access helps you monitor your changes. This is especially useful if you need to restore a deleted row.

Access provides an Undo capability to restore deleted rows. For example, you can restore a deleted row by typing the U line command for that row. See “Restoring Data” on page 81for additional Undo functions.

HIDE DELETES Command

You can use the HIDE DELETES command to remove deleted rows from the display. An option on the Editor and Display Options panel is also available to control whether deleted rows are displayed by default.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 24 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP **********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
___    07101  Movie Mania          572 Front St         Auburn           MA
___    07103  Video Edge           400 Pittsfield Rd    Lenox            MA
___    07118  Movie Store          752 State Rd         Menemsha         MA
___ I  07121  New City Video       572 Chestnut Street  Springfield      MA
___ I  07122  Old Time Movies      1001A West Main St   Springfield      MA
___    07123  Video Way            112 South Moreland A Groton           MA
___    07126  Movie Rentals        101 Munson St        Greenfield       MA
___    07140  Showcase             1150 Indiana Terr    Beverly          MA
___    07141  Showcase II          57 Rock Hollow       Salem            MA
___    07191  Popcorn              15 Crystal Park      Lenox            MA
___    07198  Video-tron Inc.      100 West Street      Brookline        MA
___    07201  Movie Buff           400 Merrimac Ave     Concord          MA
___    07203  Movies-R-Us          1772 Bridge St       Bourne           MA
___    07235  Jack's               Grafton Plaza        Grafton          MA
```

Figure 62. Deleted Rows Hidden from Display

To redisplay deleted rows, use the SHOW DELETES command.

DELETE Rules

When you delete a row from a parent table, Access passes the delete request to DB2. If DELETE rules defined to DB2 allow the delete, Access adjusts the display accordingly. When you delete a row from a parent table with a DELETE RESTRICT rule, delete operations may result in errors. When you delete a row from a parent table with a DELETE CASCADE or DELETE NULL rule defined, the related rows in the child table will be adjusted appropriately, whether or not they are displayed.

For example, if a table is defined with a DELETE CASCADE rule, certain rows you select for deletion may be the parents in a relationship defined with cascade delete. In that case, Access warns you that descendant rows in tables not included in the Access Definition may be deleted. Access displays a pop-up to confirm the cascade delete.

For additional information, see “Restoring Data” on page 81, and see the UNDO and ROLLBACK commands in the *Command Reference Manual*.

Applying Global Changes

Use the CHANGE command to apply changes to the data in one or more rows.

For example, the CUSTOMERS table includes a column that contains a code to identify the sales representatives. If sales representative RS replaces sales representative RP, you must change all customers assigned to RP. The SALESMAN_ID column identifies the sales representative. Enter the CHANGE command as:

```
CHANGE RP RS IN SALESMAN_ID ALL
```

If you omit the column name, all occurrences of RP in all columns are changed to RS. In other words, the command would also change all occurrences of RP in the CITY column and any other column of the CUSTOMERS table. Using the column name ensures that only the target data is changed.

Specify a table name if the same column name occurs in more than one table in the display. If you do not fully qualify a column name, the search for the column begins at the lowest-level table and proceeds through the tables in order, lowest to highest, until a matching column is found. Only the column located in a lower-level table is affected. If a matching column is not found, Access displays a message.

To perform a case-sensitive search, use the form C‘string' where string is the desired search string. For example, enter the command as:

```
CHANGE C'RP' RS ALL
```

Only occurrences of RP are changed. Any other case combination, such as Rp, rP, or rp, is not changed.

To include spaces, embedded quotation marks, and so forth, use apostrophes on the change string. For example:

```
CHANGE 'R P' 'R S' ALL
```

This changes all occurrences of “R P” to “R S”.

The CHANGE command supports keywords to change the first, last, next, or previous occurrence of a string. The RCHANGE command re-executes the previous CHANGE command.

Translating Case

When you edit data, the CAPS primary command governs the translation of alphabetic characters to uppercase. CAPS ON causes data in any updated column to be translated to uppercase. Use CAPS OFF to retain the data as typed.

You can also use an option on the User Options panel to automatically set CAPS ON or OFF.

The case line commands can be used to modify the case of an existing line of data. The UC line command translates all alphabetic characters in the row to uppercase. The LC line command translates all alphabetic characters in the row to lowercase. Single row and block forms of these commands are available.

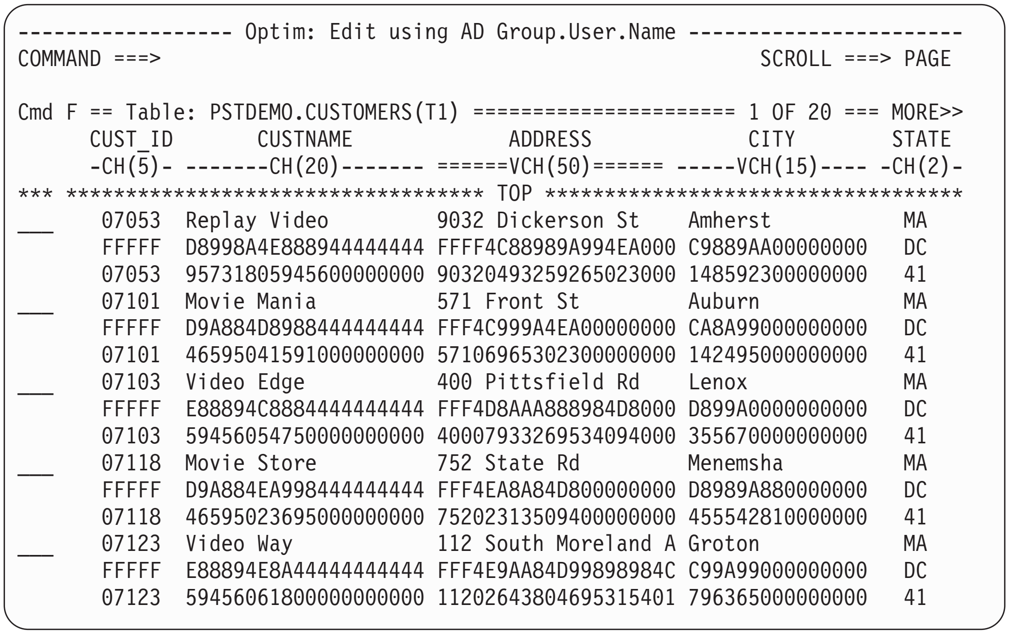
For further information on the primary and line commands, see CAPS and Editing Case in the *Command Reference Manual*.

Editing in Hexadecimal Mode

Under certain circumstances, you may need to display the hexadecimal (or hex) value of data while editing or browsing.

Use the HEX primary command to display each row of data in the standard text format with its hexadecimal representation. Note that the hexadecimal representation is displayed only for columns with character data type.

When data is displayed in hexadecimal mode, each row of data uses three lines on the display. The first line provides the text representation of the data. The second and third lines present the same data in hexadecimal. The hexadecimal representation is shown as two digits directly under each EBCDIC character, as in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: PSTDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY        STATE
      -CH(5)- -------CH(20)------- ======VCH(50)====== -----VCH(15)---- -CH(2)-
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst           MA
       FFFFF  D8998A4E888944444444 FFFF4C88989A994EA000 C9889AA00000000   DC
       07053  95731805945600000000 90320493259265023000 148592300000000   41
___    07101  Movie Mania          571 Front St         Auburn            MA
       FFFFF  D9A884D8988444444444 FFF4C999A4EA00000000 CA8A99000000000   DC
       07101  46595041591000000000 57106965302300000000 142495000000000   41
___    07103  Video Edge           400 Pittsfield Rd    Lenox             MA
       FFFFF  E88894C8884444444444 FFF4D8AAA888984D8000 D899A0000000000   DC
       07103  59456054750000000000 40007933269534094000 355670000000000   41
___    07118  Movie Store          752 State Rd         Menemsha          MA
       FFFFF  D9A884EA998444444444 FFF4EA8A84D800000000 D8989A880000000   DC
       07118  46595023695000000000 75202313509400000000 455542810000000   41
___    07123  Video Way            112 South Moreland A Groton            MA
       FFFFF  E88894E8A44444444444 FFF4E9AA84D99898984C C99A99000000000   DC
       07123  59456061800000000000 11202643804695315401 796365000000000   41
```

Figure 63. Hexadecimal Mode

Overtyping

You can type over data as text or as the hexadecimal representation. If you change both a text character and the hexadecimal representation, the hexadecimal change overrides the change to the text. Entering a noncharacter representation on a hexadecimal line results in a warning message.

The line and primary commands that are available in standard display mode are also available in hexadecimal mode. Some commands, such as the FIND and CHANGE commands, have operands specifically for hexadecimal mode. (See the *Command Reference Manual* for the complete syntax for the FIND and CHANGE commands.)

FIND Command

To locate a specific hexadecimal value anywhere in the data, use the FIND command with the HEX operand. HEX is useful when you want to locate an invalid (non-displayable) character value. For example, to locate the first occurrence of the hex value 00, enter:

```
FIND HEX 00
```

To locate the next occurrence of the search value, use the RFIND command (usually assigned to PF5).

CHANGE Command

To replace an occurrence of a hexadecimal value with another hexadecimal value, use the CHANGE command with the HEX operand. For example, to replace all occurrences of the hex value 00 with 40, enter:

```
CHANGE HEX 00 40 ALL
```

Trailing Blanks and Nulls

Trailing blanks in CHAR columns are stored in the database as blanks. They are displayed in hexadecimal mode as the value 40. These values can be modified like any other displayed value and the modification will be stored in the database.

Generally, trailing blanks in VARCHAR columns are not stored in the database and are not displayed. To include trailing blanks in a VARCHAR column value, use the DB2 VarChar Delimiter character to mark the end of the value. The default DB2 VarChar Delimiter character is a semi-colon (;), but you can change the default on the Editor and Display Options panel.

Handling Rows in Error

A row that contains an error condition is not inserted into the database in an edit session. An E status flag value indicates rows containing a DB2 error.

An error can occur, for example, when you use repeat or copy to insert new rows and values for unique columns or required columns have not been edited appropriately. An error can also occur when you supply values that do not conform to the DB2 table constraints.

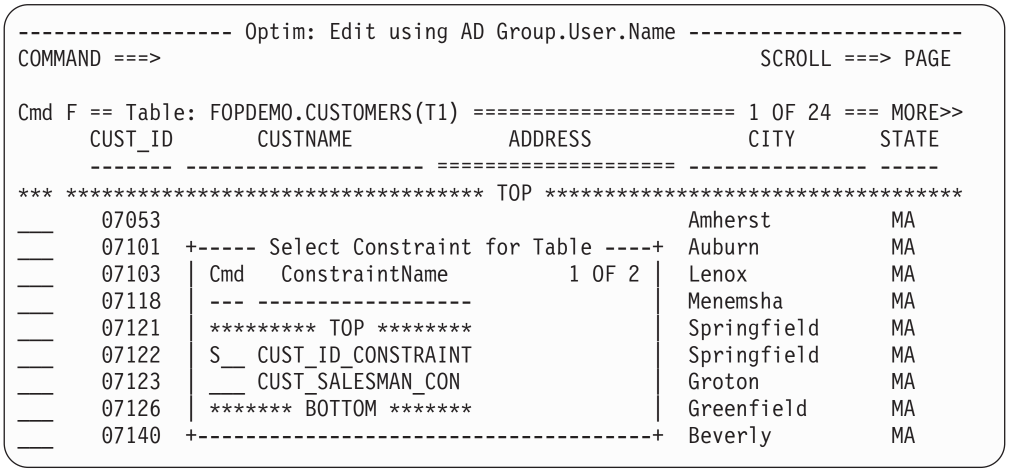
FIND ERROR

You can use the FIND ERROR command to locate and scroll to the error. Display information about the error by positioning the cursor on the line in error and pressing the HELP key or by entering the HELP command.

LIST CONSTRAINTS

If an error is the result of a table constraint, HELP displays the constraint text. However, it may be useful to display the table constraints before you begin editing. Use the LIST CONSTRAINTS command. If only one constraint is defined for the table, the text is displayed. If more than one constraint is defined, a selection list of the constraints for a specific table is displayed. From this list, you can display the text of any listed constraints. You can use LIST CONSTRAINTS for any active table at any time during the session.

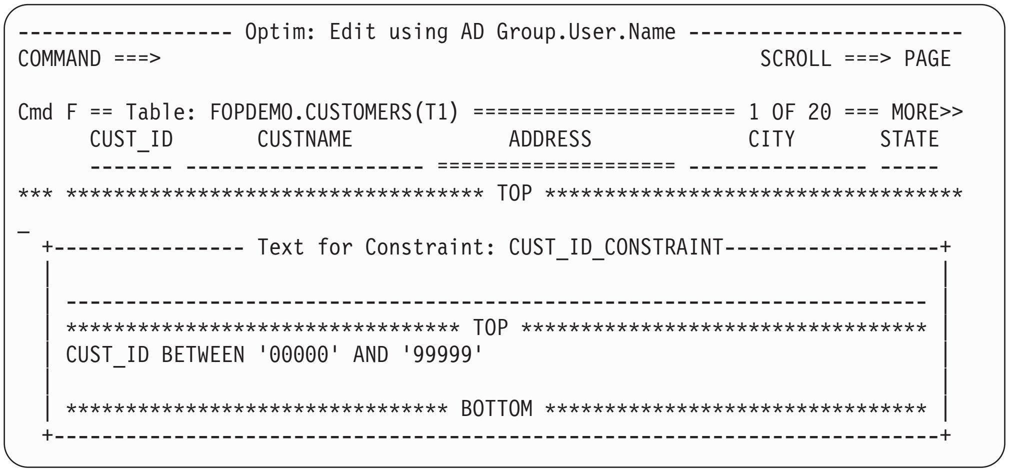
For example, assume two constraints apply to the CUSTOMERS table. If you specify the LIST CONSTRAINTS command, the following selection list is displayed.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 24 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053                                            Amherst          MA
___    07101  +----- Select Constraint for Table ----+  Auburn           MA
___    07103  | Cmd   ConstraintName         1 OF 2 |  Lenox            MA
___    07118  | --- ------------------              |  Menemsha         MA
___    07121  | ********* TOP ********              |  Springfield      MA
___    07122  | S__ CUST_ID_CONSTRAINT              |  Springfield      MA
___    07123  | ___ CUST_SALESMAN_CON               |  Groton           MA
___    07126  | ******* BOTTOM *******              |  Greenfield       MA
___    07140  +--------------------------------------+  Beverly          MA
```

Figure 64. Select Constraint for Table

If necessary, you can scroll the **Select Constraint for Table** list. Type S in the **Cmd** column before CUST_ID_CONSTRAINT to display the constraint text.



```
------------------ Optim: Edit using AD Group.User.Name ------------------------
COMMAND ===>                                                   SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ************************************
_
  +---------------- Text for Constraint: CUST_ID_CONSTRAINT------------------+
  |                                                                          |
  | ------------------------------------------------------------------------ |
  | ********************************* TOP ********************************** |
  | CUST_ID BETWEEN '00000' AND '99999'                                      |
  |                                                                          |
  | ******************************** BOTTOM ******************************** |
  +--------------------------------------------------------------------------+
```

Figure 65. Constraint Text Displayed

The constraint name is displayed after **Text for Constraint**. The text of the constraint follows on subsequent lines. Up to 50 lines of up to 75 characters are available for text display. A line of more than 75 characters is wrapped with line breaks occurring on word boundaries. If all text cannot be displayed at one time, you can scroll the display UP and DOWN.

Use END to return to the **Select Constraint for Table** pop-up window or, if you have selected more than one constraint, use END to display the text of the next selected constraint. Use CANCEL to return to the **Select Constraint for Table** pop-up window without viewing any additional constraints. When the **Select Constraint for Table** pop-up window is redisplayed, you can select another constraint to display its text or use END or CANCEL to return to the edit session.

Committing Changes to the Database

Each time a change is made to a data row during editing, the database is updated. Locks are applied to the page containing the modified row. This causes that row and any other row on that page to be locked.

The COMMIT command or the SAVE command is used to commit all outstanding changes to the database.

The ROLLBACK command is used to remove changes since the last commit point. Any outstanding locks are released when the data is committed or a ROLLBACK is requested.

When a single user uses the database, the locks do not present a problem. When several users use the database, locking must be considered, even when the users are modifying different tables. Other users cannot access any rows on a page locked by another user.

Multi-User Considerations

In a multi-user environment, it is essential that consideration be given to the number of rows locked and the length of time those rows are locked. To minimize contention, use the command AUTOCOMMIT ON to specify that the changes are automatically committed every time ENTER or any function key is pressed.

This does, however, affect the impact of a ROLLBACK request. ROLLBACK restores the database to the last commit point. If the data is committed each time ENTER is pressed, the last commit point is the state of the data after the last ENTER.

Simply viewing the data does not automatically cause the rows to be locked. Read-only functions that fetch data, such as scrolling, do not modify the data. Locks are only applied when the data is actually changed.

Concurrent Updates

A concurrent update occurs when another user makes a change to the row between the time it is fetched and the update is requested. Access provides error recovery processing for this situation. For more information about the recovery processing, see Chapter 6, "Error Recovery for Concurrent Updates," on page 109.

Restoring Data

You can use the ROLLBACK command to return the database to the state of the last commit point.

Access refetches the data presented to the user to reflect the current status of the database.

In addition to ROLLBACK, Access provides the UNDO command to remove editing changes that have occurred since the last set of rows was fetched. You can use the UNDO primary command or the Undo line command. The primary command affects all currently fetched data; whereas, the line command affects specific rows. UNDO does not refetch the data.

The advantage of the UNDO command is that it is not affected by COMMITs. The UNDO command can be used to back out changes since the data was last fetched.

Fetch Set

The set of rows that Access reads from DB2 for each table or view is referred to as a "fetch set." A fetch set includes rows from a single table or view. For example, in a multi-table display of the CUSTOMERS and ORDERS tables, the CUSTOMERS fetch set contains all selected CUSTOMERS rows. The ORDERS fetch set contains only the ORDERS rows related to the currently displayed CUSTOMERS row.

A fetch set is retrieved when you:

- Request a join. The related rows in the joined table must be fetched.
- Request a ROLLBACK, causing all data to be refetched for all tables to ensure current data.
- Use the SORT CRITERIA command to redefine sort criteria for a displayed table. The rows must be fetched for DB2 to be able to sort the data. The rows for any subordinate tables are also refetched.
- Use the RESORT command to refetch the rows based on the sort criteria defined. The rows for any lower-level tables are also refetched.
- Use the SELECTION CRITERIA command to redefine selection criteria for a displayed table. The newly selected rows must be fetched. The rows for any lower-level tables will also be refetched.
- Scroll a table that has subordinate tables. A new fetch set is required for all subordinate tables.
- Use the SWITCH command to display a different table in a stack. Each time SWITCH is invoked, the edited data is committed and cannot be undone. The data is refetched when the table is redisplayed.
- Modify the maximum number of fetched rows using the MAX FETCH ROWS command or the **Maximum Fetch Rows** prompt on the **Editor and Display Options** panel.
- Use a command that causes another row in a higher-level table to be displayed, thus requiring a fetch for any lower-level tables. These commands include the CHANGE, DELETE, HIDE, UNDO, EXCLUDE, FIND, and SHOW commands.

Single Table Editing

To clarify the difference between ROLLBACK and UNDO, assume a single table is being edited. During the session, AUTOCOMMIT is in effect. That means a commit is automatically performed each time the ENTER key is pressed.

Assume several changes and interactions have been performed. If ROLLBACK is executed, only the changes made since the last commit, in this case since the ENTER key was pressed, are restored. If the UNDO primary command is entered as either UNDO LAST or UNDO with no operands, the changes made during the last screen interaction are restored. If the UNDO ALL primary command is executed, Access attempts to restore the original fetch set and remove all changes to the data.

Sample Scenario–Single Table

1. Fetch Set contains the row:

```
Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         9032 Dickerson St    Amherst          MA
```

2. The ADDRESS column is changed:

```
Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         105 Main Street      Amherst          MA
```

3. ENTER is pressed with AUTOCOMMIT ON.
4. A second change is made, this time to the CITY column:

```
Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay Video         105 Main Street      Boston           MA
```

If ROLLBACK is requested before ENTER is pressed, the CITY Amherst is restored, but the ADDRESS has been committed and remains. However, if ENTER is pressed, thus committing the changes to CITY, and then ROLLBACK is requested, no data is restored.

If UNDO LAST is requested, the CITY Amherst is restored. That is the last change that was made. If UNDO ALL is requested, Access attempts to re-update the row to its original fetch set values. All changes (in this example, the changes to the ADDRESS and CITY) are restored. UNDO ALL will restore all data because it functions on the fetched set and is independent of commit points.

It is important to remember that ROLLBACK causes a fetch and UNDO does not. That means UNDO ALL can be executed to remove all previous changes even after UNDO LAST has been executed on the same fetch set. As long as a new fetch set has not been acquired, UNDO ALL can be executed to remove all changes.

In this example, only one row has been modified. When changes have been made to several rows, use the UNDO primary command to restore the data from the current rows in all displayed tables.

Using UNDO ALL, changes to all rows in the current fetch set of each table are removed. Using UNDO LAST, the changes applied to each table in the last screen interaction in which changes were made are undone.

The Undo line commands, UL and UA, are available to undo the modifications to a single row without impacting the other changes. The UL (U is a valid abbreviation) line command removes the last change made to the specified row regardless of the number of screen interactions since the change was made. The UA line command will remove all changes to the specified row regardless of how many changes were made with any number of screen interactions.

Multiple Table Editing

When editing multiple tables, a fetch set is present for each table. Assume that the CUSTOMERS table is displayed along with the ORDERS table. Each time the CUSTOMERS table is scrolled, a new set of rows is fetched for the ORDERS table.

The UNDO primary command and Undo line commands can be used to remove changes only to the current ORDERS fetch set and CUSTOMERS fetch set.

Sample Scenario–Multiple Tables

1. The CUSTOMERS and ORDERS tables are displayed:

   ```
   Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
         CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
         ------- -------------------- ==================== --------------- -----
   *** *********************************** TOP ***********************************
   ___    07053  Replay Video         9032 Dickerson St    Amherst          MA

   Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 3 === MORE>>
         ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
         -------- ------- ---------- ---------- --------------- --------------
   *** *********************************** TOP ***********************************
   ___       5214   07053  03/26/1998  10.15.33       15.90          RP0013
   ___       5213   07053  03/26/1998  14.23.59       37.92          RP0013
   ___       5212   07053  03/26/1998  12.57.33       22.32          RP0013
   *** ********************************* BOTTOM **********************************
   ```

2. A change is made to the ADDRESS in the CUSTOMERS table and the ORDER_DATE for order 5212 in the ORDERS table:

   ```
   Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
         CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
         ------- -------------------- ==================== --------------- -----
   *** *********************************** TOP ***********************************
   ___    07053  Replay Video         105 Main Street      Amherst          MA

   Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 3 === MORE>>
         ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
         -------- ------- ---------- ---------- --------------- --------------
   *** *********************************** TOP ***********************************
   ___       5214   07053  03/26/1998  10.15.33       15.90          RP0013
   ___       5213   07053  03/26/1998  14.23.59       37.92          RP0013
   ___       5212   07053  03/28/1998  12.57.33       22.32          RP0013
   *** ********************************* BOTTOM **********************************
   ```

3. ENTER is pressed with AUTOCOMMIT ON.
4. A change is made to the CITY in the CUSTOMERS table and the ORDER_SALESMAN for order 5212 in the ORDERS table:

   ```
   Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
         CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
         ------- -------------------- ==================== --------------- -----
   *** *********************************** TOP ***********************************
   ___    07053  Replay Video         105 Main Street      Boston           MA

   Cmd F == Table: FOPDEMO.ORDERS(T2) ========================= 1 OF 3 === MORE>>
         ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
         -------- ------- ---------- ---------- --------------- --------------
   *** *********************************** TOP ***********************************
   ___       5214   07053  03/26/1998  10.15.33       15.90          RP0013
   ```

```
___      5213   07053  03/26/1998  14.23.59       37.92          RP0013
___      5212   07053  03/28/1998  12.57.33       22.32          RP0024
*** ******************************** BOTTOM *********************************
```

If ROLLBACK is requested before the ENTER key is pressed, the CITY is restored, but the ADDRESS has been committed and will remain. Similarly, the first change to the last order remains and only the last changed column, ORDER_SALESMAN is restored.

If, however, ENTER is pressed and the commit is performed, only the UNDO primary command will restore the data because it functions on the fetch set, independent of the commit points. In this case, the primary command UNDO ALL will restore the original rows from the ORDERS table and the CUSTOMERS table. If the primary command UNDO LAST is requested, CITY is restored and the ORDER_SALESMAN column is restored.

If the CUSTOMERS table is scrolled, a new fetch set is obtained for the ORDERS table. At that point, changes made to the previous customer's orders cannot be undone. In other words, UNDO ALL will restore only the current fetch set for each displayed table. Thus, only the CUSTOMERS table is restored. In contrast, if the data has not been committed to the database, ROLLBACK can be executed to back out all changes and restore the data.

Undo Line Command

The Undo line command can be used to restore the data for specific rows. The following occurs when using the Undo line commands with the multi-table sample scenario.

- If UL, indicating undo last, is requested on the displayed CUSTOMERS row, CITY is restored to Amherst. Changes to the ORDERS table are retained. UL can be entered for the changed row in the ORDERS table to restore the values prior to the last change. (For example, ORDER_SALESMAN is restored to RP0013.)
- If UA, indicating undo all, is requested on the CUSTOMERS row, the original fetched CUSTOMERS row is restored and all changes are removed. Again, the ORDERS table is not affected. UA can be entered for the changed row in the ORDERS table to restore the row to its original fetched state.

ROLLBACK vs. UNDO

It is important to understand the distinction between ROLLBACK and UNDO. ROLLBACK performs as commonly expected within the DB2 environment. UNDO is unique to Access. It provides additional flexibility and power not available with ROLLBACK. UNDO can be used to:

- Remove only the last change — The UNDO LAST primary command removes the last changes made to any current fetch sets in a single screen interaction. There may be one or more intervening screen interactions in which no changes are made.

  In addition, the UL line command affects only the targeted rows. UL will undo the last change to the row on which it was entered.
- Remove all changes — The UNDO ALL primary command removes all changes made to the data since the last fetch.

  The UA line command affects only the row on which it is entered.
- Restore rows in error — The UNDO ERRORS primary command is used to restore all rows in error to their last valid state. This state is the last valid value the row contained.

  The UE line command restores a single row in error. For example, assume the Repeat line command has duplicated a row three times. All column values are duplicated including those that should be unique. These values must be edited or an error condition results. Use the UNDO ERRORS primary command or the UE line command to “undo” or remove the repeated rows containing errors.

UNDO Consideration

The UNDO commands actually perform a series of update, insert, and delete requests. On rare occasions, these database requests may fail due to Referential Integrity constraints enforced by DB2. If so, they are marked as an error.

Displaying SQL

During an edit or browse session, you can enter the SHOW SQL command to display a generalized form of the SQL that was used to fetch the rows from any displayed tables.

By default, the SQL for the lowest-level table is displayed.

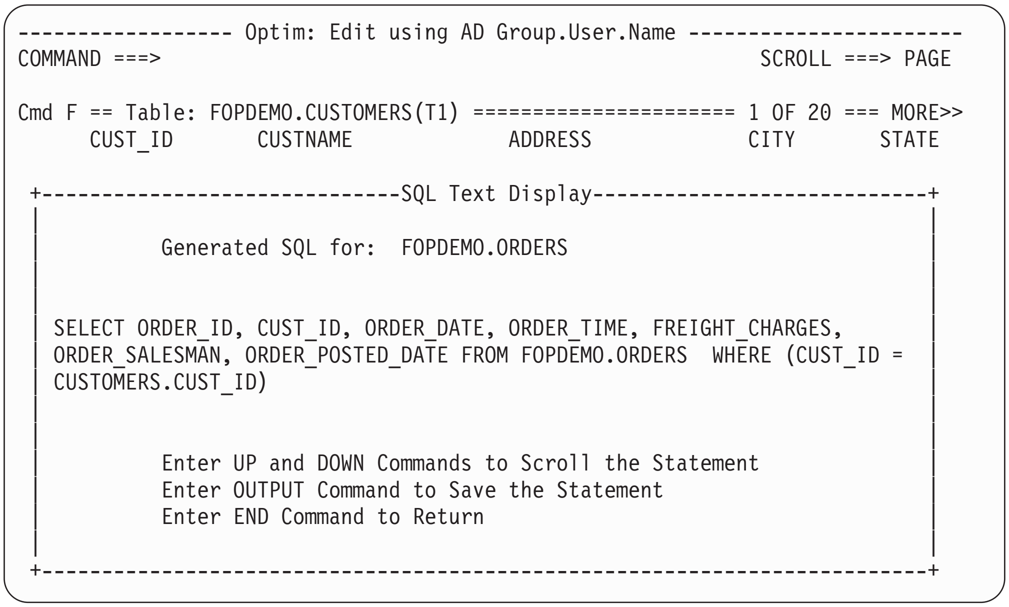
To display the SQL for another table, either supply the table name or identifier with the command or position the cursor to the desired table. For example, to display the SQL for the highest-level table, enter:

```
SHOW SQL T1
```

The table name is shown in the window heading. The generated SQL may contain three parts:

- Host variables are declared if the corresponding columns have different data types or dimensions.
- Pseudocode is generated to populate the host variables if they are declared.
- The “generalized” SELECT statement, which incorporates the host variables.

Assume the CUSTOMERS table and the ORDERS table are displayed. The SHOW SQL command is entered to display the SQL used to fetch the rows from the ORDERS table. The SQL statement is displayed as:



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE

 +------------------------------SQL Text Display----------------------------+
 |                                                                          |
 |          Generated SQL for:  FOPDEMO.ORDERS                              |
 |                                                                          |
 |                                                                          |
 | SELECT ORDER_ID, CUST_ID, ORDER_DATE, ORDER_TIME, FREIGHT_CHARGES,       |
 | ORDER_SALESMAN, ORDER_POSTED_DATE FROM FOPDEMO.ORDERS  WHERE (CUST_ID =  |
 | CUSTOMERS.CUST_ID)                                                       |
 |                                                                          |
 |                                                                          |
 |          Enter UP and DOWN Commands to Scroll the Statement              |
 |          Enter OUTPUT Command to Save the Statement                      |
 |          Enter END Command to Return                                     |
 |                                                                          |
 +--------------------------------------------------------------------------+
```

Figure 66. Text Display of Generated SQL

Generalized WHERE Clause

Access generates SQL that contains the specific data values for columns defined in the relationship. Therefore, the actual SQL that was executed based on the SQL in Figure 66 contained a value for CUST_ID, as in:

```
SELECT ...
FROM FOPDEMO.ORDERS WHERE CUST_ID = '07053'
```

However, this SQL would not be as useful to the user as the generalized form that is shown:

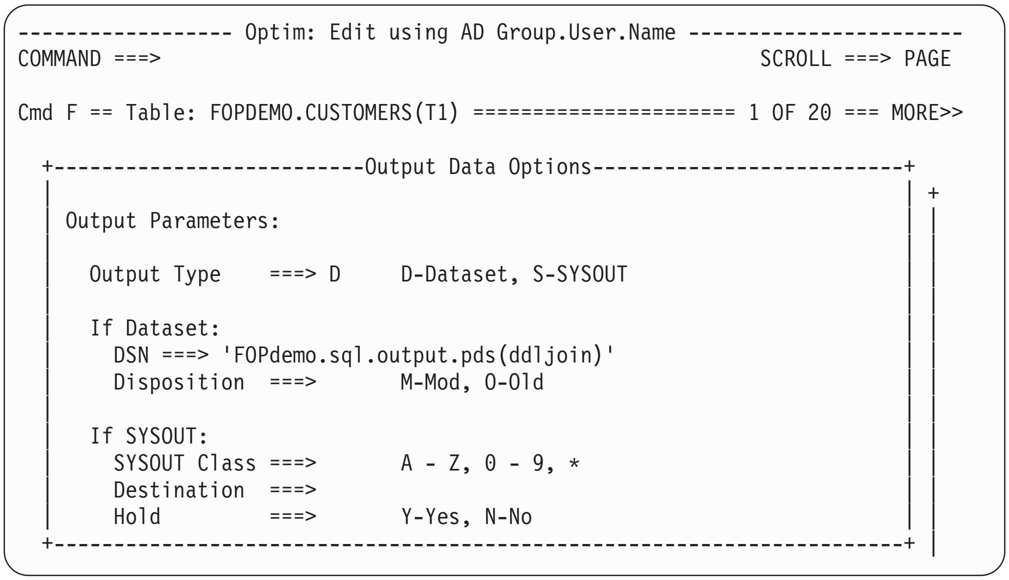
```
SELECT ...
FROM FOPDEMO.ORDERS WHERE CUST_ID =
:CUSTOMERS.CUST_ID
```

The last part of the predicate in the SQL statement is shown as a host variable (note the colon) with a meaningful name that represents the processing. Although this generalized SQL is incomplete, it is displayed because this generalized statement is more meaningful than the actual executed SQL statement.

Saving the SHOW SQL Output

You can save or print the output of the SHOW SQL command by using the OUTPUT command.

Entering the OUTPUT command displays the Output Data Options panel, as shown in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>

  +--------------------------Output Data Options--------------------------+
  |                                                                       | +
  | Output Parameters:                                                    | |
  |                                                                       | |
  |   Output Type    ===> D     D-Dataset, S-SYSOUT                       | |
  |                                                                       | |
  |   If Dataset:                                                         | |
  |     DSN ===> 'FOPdemo.sql.output.pds(ddljoin)'                        | |
  |     Disposition  ===>       M-Mod, O-Old                              | |
  |                                                                       | |
  |   If SYSOUT:                                                          | |
  |     SYSOUT Class ===>       A - Z, 0 - 9, *                           | |
  |     Destination  ===>                                                 | |
  |     Hold         ===>       Y-Yes, N-No                               | |
  +-----------------------------------------------------------------------+ |
```

Figure 67. Output Data Options Panel

Panel Prompts

The prompts on this panel are:

Output Type
Indicate whether the SQL is to be saved in a dataset or as a SYSOUT class for printing.

If Dataset:
If the **Output Type** is D, specify the following:

DSN Specify the name of a new or existing dataset. The dataset must be either a sequential file, with a record format of fixed or fixed block, or a Partitioned Data Set (PDS) with a member name enclosed in parentheses.

The dataset name is automatically prefixed unless it is enclosed in single quotes. The prefix is determined by the **Data Set Prefix** option on the User Options panel.

Alternatively, you may specify an asterisk (*) or the DB2 LIKE character (%) at the end of the DSN specification to generate a selection list. For example:

'FOPDEMO.DDL*' or 'FOPDEMO.DDL%'

Both generate a selection list of all datasets having a DSN beginning with FOPDEMO.DDL and having a valid format (that is, PDS or sequential file with a record format of fixed or fixed block).

If the specified dataset does not exist, Access prompts for allocation information. See the *Common Elements Manual*, Appendix B. Allocating External Files, for a description of the allocation prompts.

Disposition
For an existing, sequential dataset only. Specify:

M Modify. The SQL data is appended to the dataset.

O Old. The contents of the file are replaced with the SQL data.

If SYSOUT:
If the **Output Type** is S, specify the following:

SYSOUT Class
Specify the output class to which the printed output belongs. This can be a character from A through Z, 0 through 9, or an asterisk (*).

Destination
Specify a valid SYSOUT destination. The destination can be a local or remote terminal, a node in the JES network, a local or remote printer or workstation, or a TSO user ID.

Hold Specify whether the output is to be held. Specify:

Y Output is held until released or deleted by an operator.

N Output is not held.

When you have entered all necessary information on this panel, press ENTER to continue processing. To return to the previous panel without specifying output data options, use END or CANCEL.

Terminating a Session

An edit or browse session can be terminated using:

CANCEL
Terminates the current session and returns control to the location from which it was invoked. Any changes that have not been committed to the database are backed out via ROLLBACK.

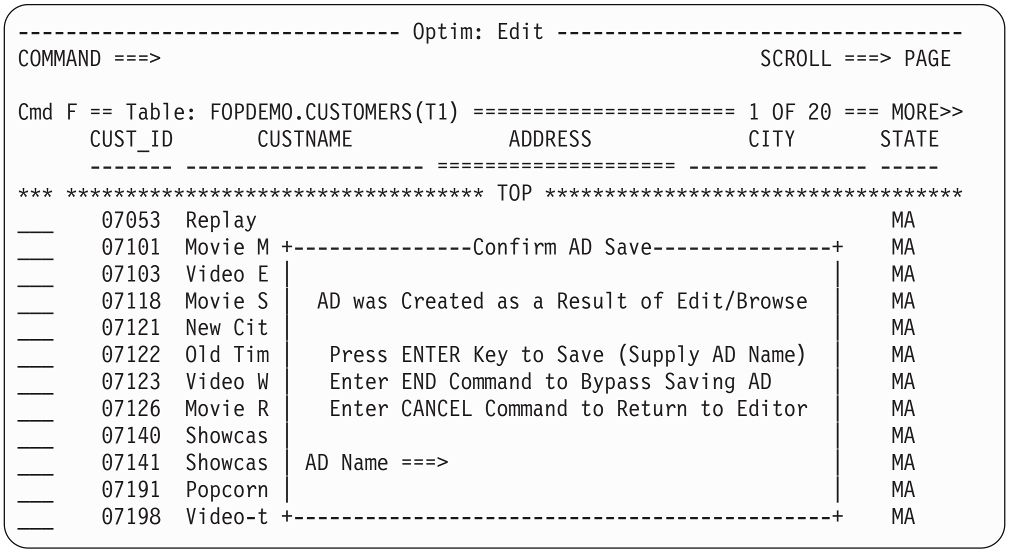
END Terminates the current session and returns control to the location from which it was invoked. Any outstanding changes are committed to the database.

RESTART
Terminates the current session and restarts a new session. If there are any uncommitted changes and pending functions, you are prompted to determine whether to ROLLBACK or COMMIT the changes.

The CANCEL, END, and RESTART functions can be assigned to program function keys.

Storing Dynamic Changes

If you specified Yes for **Prompt to Create AD on Exit** on the Choose a DB2 Table/View to Edit panel, you can save the Access Definition. If you choose to save the Access Definition, the complete three-part name, *group.user.name*, is required. The prompt is displayed as:



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay                                                     MA
___    07101  Movie M +---------------Confirm AD Save---------------+    MA
___    07103  Video E |                                             |    MA
___    07118  Movie S |  AD was Created as a Result of Edit/Browse  |    MA
___    07121  New Cit |                                             |    MA
___    07122  Old Tim |   Press ENTER Key to Save (Supply AD Name)  |    MA
___    07123  Video W |   Enter END Command to Bypass Saving AD     |    MA
___    07126  Movie R |   Enter CANCEL Command to Return to Editor  |    MA
___    07140  Showcas |                                             |    MA
___    07141  Showcas | AD Name ===>                                |    MA
___    07191  Popcorn |                                             |    MA
___    07198  Video-t +---------------------------------------------+    MA
```

Figure 68. Confirmation Prompt to Save Changes

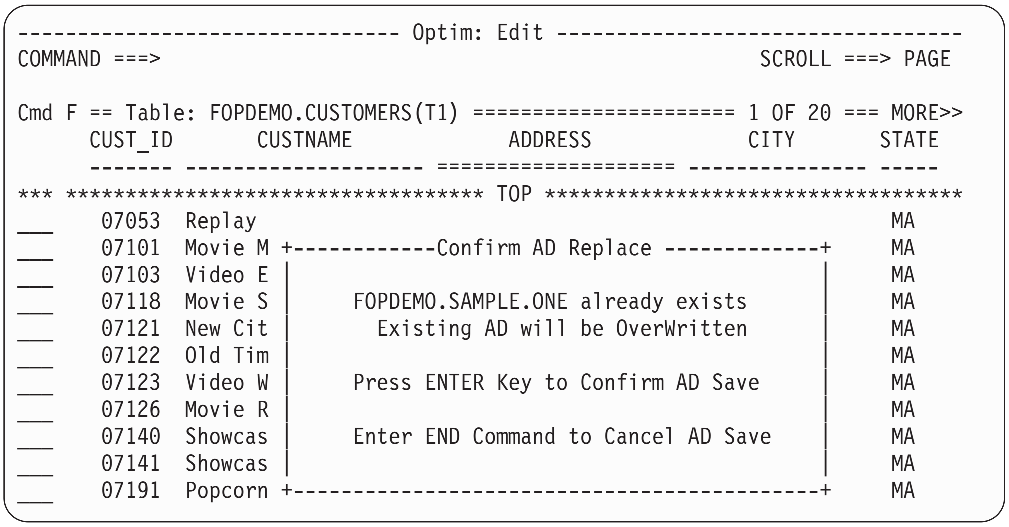
CREATE AD Command

If you specified No to **Prompt to Create AD on Exit** and want to save the specifications from this session as an Access Definition, use the CREATE AD YES command to request that Access prompt you to save the Access Definition.

Use the CANCEL command to abandon the SAVE operation and the END request. If you enter CANCEL while the **Confirm AD Save** prompt is displayed, the edit session is not terminated, and you are returned to the data display.

Replace an Access Definition

You can specify the name of an Access Definition other than the one used to invoke the Access Definition edit or browse session. If you specify the name of an existing Access Definition, Access prompts to confirm that the definition is to be replaced. Assume an Access Definition named FOPDEMO.SAMPLE.ONE exists and will be replaced. The prompt is displayed as:



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay                                                     MA
___    07101  Movie M +------------Confirm AD Replace -------------+     MA
___    07103  Video E |                                            |     MA
___    07118  Movie S |     FOPDEMO.SAMPLE.ONE already exists      |     MA
___    07121  New Cit |       Existing AD will be OverWritten      |     MA
___    07122  Old Tim |                                            |     MA
___    07123  Video W |     Press ENTER Key to Confirm AD Save     |     MA
___    07126  Movie R |                                            |     MA
___    07140  Showcas |     Enter END Command to Cancel AD Save    |     MA
___    07141  Showcas |                                            |     MA
___    07191  Popcorn +--------------------------------------------+     MA
```

Figure 69. Confirmation Prompt to Replace Access Definition

Use ENTER to replace the existing Access Definition. Use END to cancel the SAVE request.

Recursive Sessions

When another session is invoked using the EDIT or BROWSE primary commands, the current session is suspended. All changes are retained while the original session is suspended. The suspended session resumes when the new session is terminated.

You can recursively edit or browse a table directly or using an Access Definition. You cannot recursively edit or browse an Access Definition that is currently suspended in a nested session.

Note: Each nested execution of an Access Definition can cause multiple locks. If AUTOCOMMIT is not in effect and COMMIT is not executed, the locks applied by the suspended edit session or sessions remain until a COMMIT is executed.

Multiple Users

Several users may use an Access Definition simultaneously. For more information on sharing Access Definitions, see Chapter 5, "Sharing Access Definitions," on page 107.

Chapter 4. Processing Embedded SQL

Access provides commands for processing SQL statements that are embedded in a source program or displayed alone in the ISPF editor. These commands allow you to execute:

- A DB2 PREPARE to check the SQL syntax.
- A DB2 EXPLAIN to analyze the execution of an SQL statement and display the findings without exiting the ISPF edit session.
- A DB2 EXECUTE to execute the targeted SQL statement. For SELECT statements, execution automatically invokes an Access edit session.

You can use these commands while you are displaying the SQL statements during an ISPF edit session. Simply identify the target SQL statement and enter the appropriate Access command.

Note: The maximum size of an SQL statement is 32,767 bytes.

The remainder of this section discusses how to:

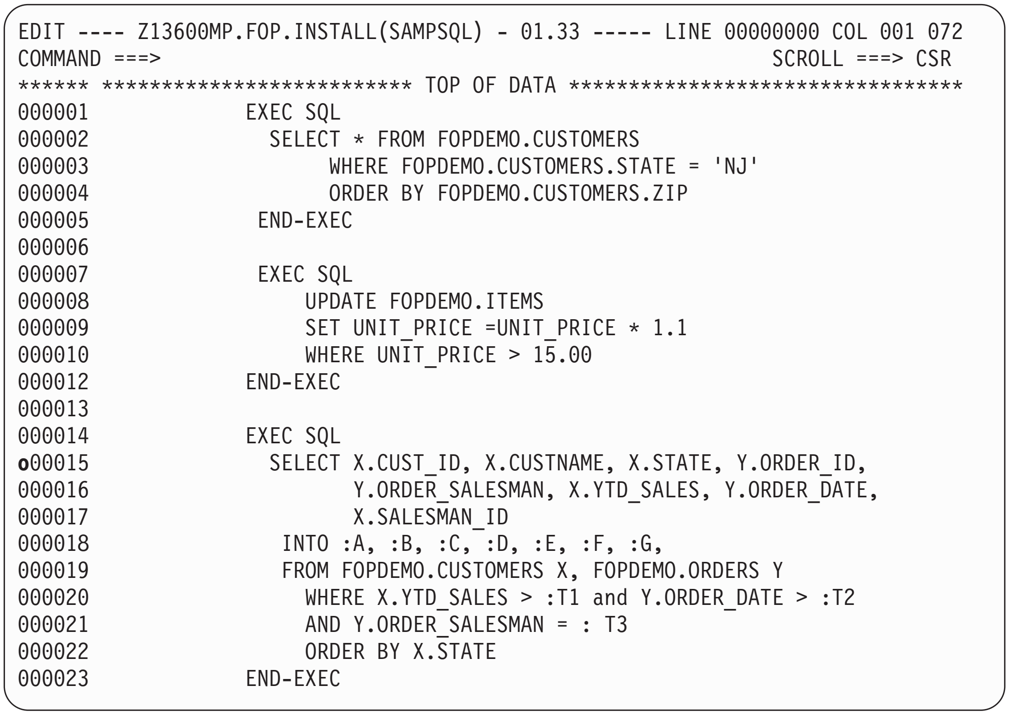
- Identify the target SQL statement.
- Use the FOPFAD2P command to execute the DB2 PREPARE statement.
- Use the FOPD2EXP command to execute the DB2 EXPLAIN statement.
- Use the FOPD2EXE command to execute the DB2 EXECUTE statement.
- Address SQL syntax errors.

Identifying the SQL Statement

To test an embedded SQL statement with the Access facilities, you must first identify the target statement in your program source code. Use the O line command or position the cursor to identify the target statement.

Using the O Line Command

Type a single O in the line command area for a desired SQL statement to identify the entire statement as the target for a command. In the following figure, the O on line 000015 identifies the SQL statement beginning on line 000014 and ending on line 000023 as the target.



```
EDIT ---- Z13600MP.FOP.INSTALL(SAMPSQL) - 01.33 ----- LINE 00000000 COL 001 072
COMMAND ===>                                                   SCROLL ===> CSR
****** ************************** TOP OF DATA ********************************
000001             EXEC SQL
000002               SELECT * FROM FOPDEMO.CUSTOMERS
000003                    WHERE FOPDEMO.CUSTOMERS.STATE = 'NJ'
000004                    ORDER BY FOPDEMO.CUSTOMERS.ZIP
000005              END-EXEC
000006
000007              EXEC SQL
000008                  UPDATE FOPDEMO.ITEMS
000009                  SET UNIT_PRICE =UNIT_PRICE * 1.1
000010                  WHERE UNIT_PRICE > 15.00
000012             END-EXEC
000013
000014             EXEC SQL
o00015               SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID,
000016                      Y.ORDER_SALESMAN, X.YTD_SALES, Y.ORDER_DATE,
000017                      X.SALESMAN_ID
000018                INTO :A, :B, :C, :D, :E, :F, :G,
000019                FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
000020                  WHERE X.YTD_SALES > :T1 and Y.ORDER_DATE > :T2
000021                  AND Y.ORDER_SALESMAN = : T3
000022                  ORDER BY X.STATE
000023             END-EXEC
```

Figure 70. Identifying Target SQL Statement

The O line command is handled like any ISPF line command. Thus, you can overtype the O or use the RESET primary command to cancel identification of the target statement.

Block forms of the O line command are also available. You can target a group of SQL statements by typing the OO line command before the beginning of the first SQL statement and after the end of the last SQL statement in the group of statements or by using *Onnnnnn*. Access displays a prompt to confirm each target statement before executing the specified Access command. When the entire block is not displayed on a single screen, you can scroll the display to complete your entry.

Targeting a Portion

You can also use a block form of the O line command when SQL statements are nested (for example, nested sub-selects) to target a portion of the entire structure. Specify OO on the first and last lines of the block of lines in the target SQL statement.

Using the Cursor Position

Another way to specify an SQL statement is to position the cursor on the target statement.

If you do not select the statement with the cursor or the O line command, the first valid SQL statement in the member is used as the starting position, provided it is a valid SQL statement.

Determining the Target SQL

When you specify a single O line command or position the cursor to target an SQL statement, Access scans the source member to identify the bounds of the target SQL. The scan proceeds as follows:

1. Beginning at the specified line, Access scans backward one line to locate “EXEC SQL,” which identifies the first line in the target statement. If EXEC SQL is not located, the first line in the member is used as the start of the target SQL statement.
   a. Access then scans forward using the rules for statement boundaries, continuation, and end-of-statement that are appropriate for the language in use. For example, END-EXEC indicates

the end of the target statement in a COBOL program and a semicolon indicates the end in a PL1 program. (Information about determining the language follows.) If the end of an SQL statement is not found, the last line in the member is used as the end of the target statement.

b. Access then determines if the original target line is within the range of lines that comprise the SQL statement. If the target line is not in the range, Access displays an error message, “SQL STATEMENT NOT MARKED.”

Determining the Language

When scanning the member to determine the boundaries of the SQL statement, Access enforces the various rules for statement boundaries, continuation, and end-of-statement documented in the *IBM Application Programming and SQL Guide*. This rule enforcement ensures that embedded SQL statements are in the correct columns and that the appropriate parsing rules for the language are used.

Using the rules appropriate for the language is important. For example, a hyphen in column 7 indicates a continuation in a COBOL program, but it represents a minus sign in a PL1 or C program.

You can use the LANG operand with the commands FOPFAD2P, FOPD2EXP, and FOPD2EXE to specify the language as COBOL, PL1, C, Fortran, or Assembler. However, the operand is necessary only if the language is Fortran or Assembler.

If the LANG operand is not used, Access attempts to ascertain the language in the following order: COBOL, PL1, and C. Access begins the scan using the rules for COBOL. If the statement is not formatted according to COBOL rules, Access reprocesses the statement using PL1 rules. If the format does not follow PL1 rules, Access reprocesses the statement using C rules. If a language cannot be determined, an error message is displayed.

Using the FOPD2EXE Command

Use the FOPD2EXE command to execute the target SQL statement.

The syntax for the FOPD2EXE command is:

```
FOPD2EXE [SSN ( ssn ) | DB2LOC (locname)]
[ SQLID ( sqlid ) ] [ QUAL ( qual ) ]
 [ LANG ( lang ) ] [ WIDTH ( width ) ]
 [ APOST | QUOTE ] [ PERIOD | COMMA ]
```

The operands are available to override the defaults specified in your Access profile. You can specify:

SSN(*ssn*)
The subsystem name. If a subsystem name is omitted, the name from your most recent Access session is used. If subsystem is not profiled, Access prompts you to re-enter the command with a subsystem name.

DB2LOC(*locname*)
To execute the target SQL on a remote DB2 subsystem, use the name of the subsystem, as it is defined to Optim.

SQLID(*sqlid*)
The SQL ID. If an SQL ID is omitted, the SQL ID from your most recent Access session is used. If SQL ID is not profiled, Access prompts for one.

QUAL(*qual*)
The qualifier used as a prefix for any unqualified table, view, or alias names in SELECT, DELETE, INSERT, and UPDATE SQL statements. If **QUAL** is omitted, DB2 uses the current SQL ID.

LANG(*lang*)
The language of the target SQL statement specified as ASM, PLI, FOR, COB, or C. The language

determines the format and character interpretation for the SQL statement. If the value for LANG is inconsistent with the current data, an error message is displayed.

By default, Access determines the language when FOPD2EXE is executed, testing for COBOL, PLI, or C, in that order. If the language cannot be determined, an error message is displayed. For more information on how Access determines the language, see page “Determining the Language” on page 93.

WIDTH(*width***)**
The size of the host variable prompt. Specify a value from 10 through 70 that indicates the maximum number of characters for values substituted for host variables in the SQL statement.

If **WIDTH** is omitted, DB2 uses the Maximum Display Width specified on the Editor and Display Options panel.

[APOST|QUOTE]
The string delimiter used in the target SQL. The default is APOST.

[PERIOD|COMMA]
The decimal point character used in numbers in the target SQL statement. The default is PERIOD.

Processing a Valid SQL Statement

The action taken for a valid SQL statement is based on the SQL statement to be executed. Processing for the various SQL statements is discussed in the following sections. If the SQL statement contains host variables, Access prompts for the values.

Using the FOPD2EXP Command

Use the Access FOPD2EXP command to execute a DB2 EXPLAIN with the target SQL statement as input.

This Access facility provides a convenient way of evaluating the performance of any SQL statement.

The DB2 EXPLAIN information is displayed as note lines after the target SQL statement. Access adjusts the screen as necessary to display the SQL statement and the EXPLAIN information. You can edit the SQL statement and re-execute the FOPD2EXP command. The previous note lines are replaced with the new DB2 information. Thus, you can see how DB2 will execute the alternative SQL statements. The syntax for the FOPD2EXP command is:

```
FOPD2EXP [ SSN ( ssn ) | DB2LOC (locname)] [ SQLID ( sqlid ) ] [ QUAL ( qual ) ]
 [ LANG ( lang ) ] [ APOST | QUOTE ]
 [ PERIOD | COMMA ]
```

Use the operands to override defaults specified in your Access profile.

SSN(*ssn***)**
The subsystem name. If a subsystem name is omitted, the name from your most recent Access session is used. If subsystem is not profiled, Access prompts you to re-enter the command with a subsystem name.

DB2LOC(*locname***)**
To execute the target SQL on a remote DB2 subsystem, use the name of the subsystem, as it is defined to Optim.

SQLID(*sqlid***)**
The SQL ID. If an SQL ID is omitted, the SQL ID from your most recent Access session is used. If SQL ID is not profiled, Access prompts for one.

QUAL(*qual***)**
The qualifier used as a prefix for any unqualified table, view, or alias names in SELECT, DELETE, INSERT, and UPDATE SQL statements. If **QUAL** is omitted, DB2 uses the current SQL ID.

LANG(*lang*)

The language of the target SQL statement specified as ASM, PLI, FOR, COB, or C. The language determines the format and character interpretation for the SQL statement. If the value for LANG is inconsistent with the current data, an error message is displayed.

By default, Access determines the language when FOPD2EXP is executed, testing for COBOL, PLI, or C, in that order. If the language cannot be determined, an error message is displayed. For more information on how Access determines the language, see page "Determining the Language" on page 93.

[APOST|QUOTE]

The string delimiter used in the target SQL statement. The default is APOST.

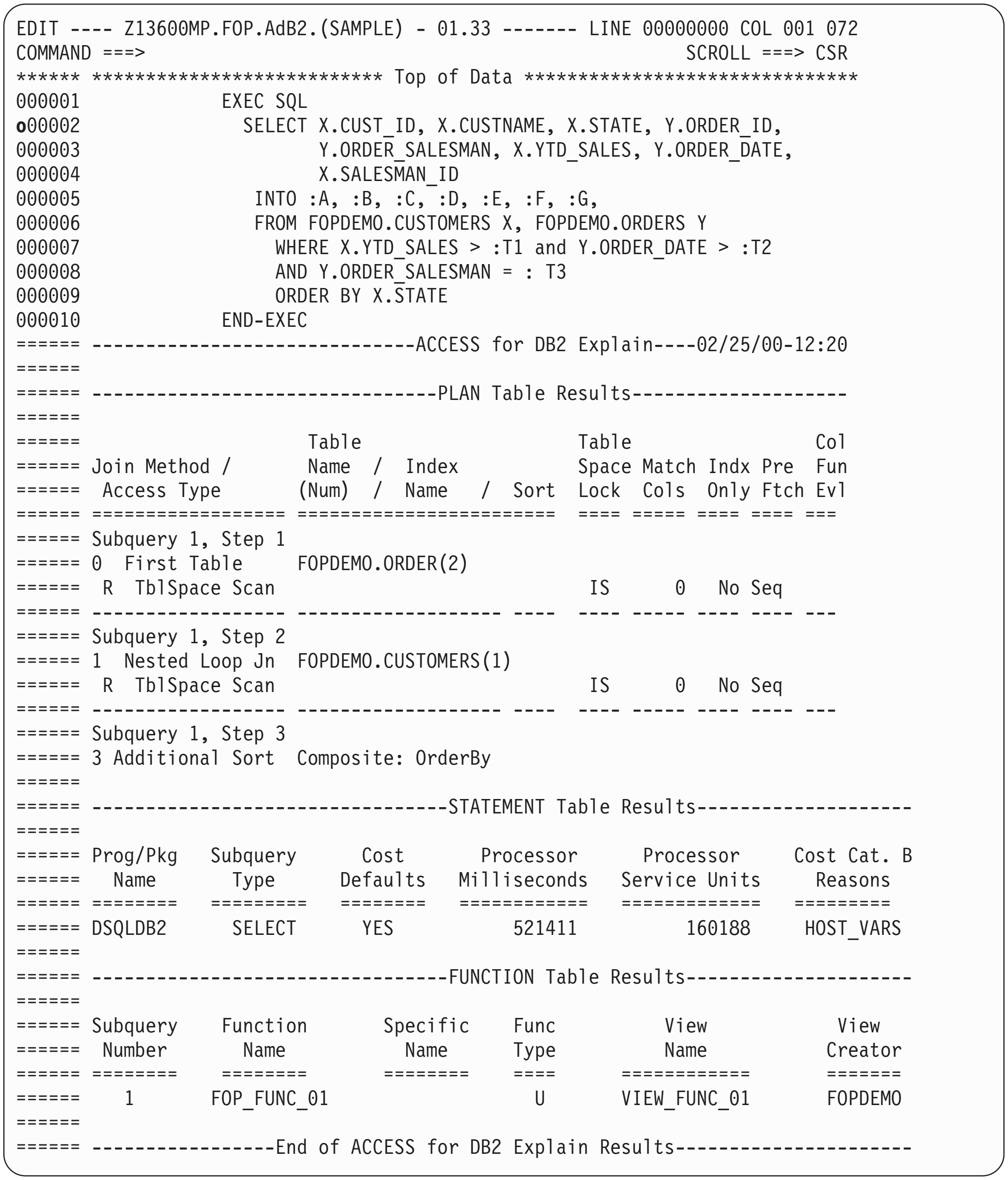
[PERIOD|COMMA]

The decimal point character used in numbers in the target SQL statement. The default is PERIOD.

When FOPD2EXP is executed, the current SQL ID must have a PLAN Table or an error occurs and a message indicates that a PLAN Table does not exist. For information about defining a PLAN Table, see the *DB2 Application Programming and SQL Guide*.

FOPD2EXP Example

Assume an embedded SQL statement is identified and the FOPD2EXP command is entered. The EXPLAIN information is presented in a logical manner.



```
EDIT ---- Z13600MP.FOP.AdB2.(SAMPLE) - 01.33 ------- LINE 00000000 COL 001 072
COMMAND ===>                                                  SCROLL ===> CSR
****** *************************** Top of Data ******************************
000001             EXEC SQL
o00002               SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID,
000003                      Y.ORDER_SALESMAN, X.YTD_SALES, Y.ORDER_DATE,
000004                      X.SALESMAN_ID
000005                INTO :A, :B, :C, :D, :E, :F, :G,
000006                FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
000007                  WHERE X.YTD_SALES > :T1 and Y.ORDER_DATE > :T2
000008                  AND Y.ORDER_SALESMAN = : T3
000009                  ORDER BY X.STATE
000010             END-EXEC
====== ------------------------------ACCESS for DB2 Explain----02/25/00-12:20
======
====== --------------------------------PLAN Table Results--------------------
======
======                    Table                    Table                Col
====== Join Method /      Name  /  Index           Space Match Indx Pre  Fun
======  Access Type      (Num)  /  Name   /  Sort  Lock  Cols  Only Ftch Evl
====== ================= =======================  ==== ===== ==== ==== ===
====== Subquery 1, Step 1
====== 0  First Table    FOPDEMO.ORDER(2)
======  R  TblSpace Scan                            IS      0   No Seq
====== ------------------ ------------------- ----  ---- ----- ---- ---- ---
====== Subquery 1, Step 2
====== 1  Nested Loop Jn  FOPDEMO.CUSTOMERS(1)
======  R  TblSpace Scan                            IS      0   No Seq
====== ------------------ ------------------- ----  ---- ----- ---- ---- ---
====== Subquery 1, Step 3
====== 3 Additional Sort  Composite: OrderBy
======
====== ---------------------------------STATEMENT Table Results--------------------
======
====== Prog/Pkg   Subquery      Cost      Processor      Processor     Cost Cat. B
======   Name       Type     Defaults  Milliseconds   Service Units    Reasons
====== ========   =========   ========   ============   ============   ========
====== DSQLDB2      SELECT      YES          521411         160188     HOST_VARS
======
====== ---------------------------------FUNCTION Table Results---------------------
======
====== Subquery    Function       Specific    Func         View          View
======  Number       Name           Name      Type         Name         Creator
====== ========    ========       ========    ====      ============      =======
======    1       FOP_FUNC_01                  U       VIEW_FUNC_01      FOPDEMO
======
====== -----------------End of ACCESS for DB2 Explain Results----------------------
```

Figure 71. EXPLAIN Report Displayed

Each step is documented and the Join Method and Access Type are presented in one column while the appropriate table name, index name or sort, as applied to the Join Method or Access Type, is presented in the next column. For this text, the corresponding column name in the PLAN Table is displayed in uppercase characters after the Access heading.

See the *DB2 Application Programming and SQL Guide* for information about displayed values. The following text provides a brief description of PLAN Table Results.

For each step, the following information is in the **Join Method/Access Type** column:

Subquery QBLOCKNO

Query or subquery identifier. This value reflects the query's order of appearance in the statement being explained. The outer level is assigned 1, the next 2, and so on.

Step
PLANNO

The step of the processed subquery. This value indicates the order of execution for the steps.

Join Method
METHOD

The join method used for the step, shown as:

0	First table accessed
1	Nested loop
2	Merge scan join
3	Sort
4	Hybrid join

Access Type
ACCESSTYPE

The method of accessing the table, shown as:

I	By index
I1	One-fetch scan
N	Index scan
R	Table space scan
M	Multiple index scan
MX	Index scan named in ACCESSNAME
MI	Intersection of multiple indexes
MU	Union of multiple indexes
blank	Not applicable

The Table Name (Num)/Index Name/Sort column, includes one of the following to correspond to the information on the same line in the first column.

Table Name(Num)/

Name of the table accessed in this step and its position in the FROM Clause.

Index Name/Sort

Name of the index used for this step. Indicators showing the type of sort that was performed for this step.

The information in the second column is taken from the CREATOR.TNAME, ACCESSCREATOR.ACCESSNAME, or one of several sort columns of the plan table. The remaining columns are:

Table Space Lock TSLOCKMODE

The lock mode of the table space, shown as:

IS	Intent share
IX	Intent exclusive
S	Share
U	Update
X	Exclusive
SIX	Share with intent exclusive

Match Cols
MATCHCOLS

The number of index keys used in an index scan when the Type is I, I1, N or MX. For other types, the value is 0.

Indx Only
INDEXONLY

Index access indicator, shown as:

Yes Access to the index is sufficient for the entire step. Data is not accessed.

No Data and the index are accessed.

Pre Ftch
PREFETCH

Type of prefetch, shown as:

Seq Sequential prefetch

List

Prefetch through page list

blank Unknown or no prefetch

Col Fun Evl
COLUMN_FN_EVAL

Timing of column function evaluation. Column Functions are evaluated:

R When data is retrieved.

S When data is sorted.

blank After data is retrieved and sorted.

STATEMENT Table Results

Prog/Pkg Name

Name of the program or package containing the statement. Blank if not applicable.

Subquery Type

Type of statement, shown as:

SELECT
SELECT statement.

INSERT
INSERT statement.

UPDATE
UPDATE statement.

DELETE
DELETE statement.

SELUPD
SELECT with FOR UPDATE OF.

DELCUR
DELETE WHERE CURRENT OF CURSOR.

UPDCUR
UPDATE WHERE CURRENT OF CURSOR.

Cost Defaults

Default values indicator.

A DB2 could estimate costs without using default values.

B DB2 forced to use default values to estimate costs.

Processor Milliseconds
Estimated processor cost, in milliseconds, for the SQL statement.

Processor Service Units
Estimated processor cost, in service units, for the SQL statement.

Cost Cat. B Reasons
Reason for using default values to estimate costs, shown as:
HOST VARIABLES
TABLE CARDINALITY
UDF (User defined functions)
TRIGGERS
REFERENTIAL CONSTRAINTS

FUNCTION Table Results

Subquery Number
Query or subquery identifier. This value reflects the query's order of appearance in the statement being explained. The outer level is assigned 1, the next 2, and so on.

Function Name
The name of the function invoked in the statement.

Specific Name
The specific name of the invoked function.

Func Type
The type of function invoked. Possible values are:

S Scalar function

T Table function

View Name
If the named function is referenced in a view definition, the name of the view. Blank, if there is no view reference.

View Creator
If the named function is referenced in a view definition, the creator of the view. Blank, if there is no view reference.

SQL with Multiple Subqueries

In the previous example, the SQL contained a single subquery. If multiple subqueries are included, they are named as Subquery 2, Subquery 3, and so on. The same information is displayed for each independently.

Retaining FOPD2EXP Information

You can retain the note lines inserted by FOPD2EXP using the ISPF Make Data, MD, line command. Once retained, you can save the notes as comments within your program or copy them to another member.

Using the FOPFAD2P Command

The Access FOPFAD2P command executes a DB2 PREPARE with the target SQL statement as input.

The DB2 PREPARE command checks the embedded SQL for syntax errors and returns the results. The ISPF edit session remains active during the PREPARE. When the syntax is correct, an appropriate message is displayed in the message area at the beginning of the ISPF edit session screen. The syntax of the FOPFAD2P command is:

```
FOPFAD2P   [ SSN ( ssn ) | DB2LOC (locname)]  [ SQLID ( sqlid ) ]  [ QUAL ( qual )]
      [ LANG ( lang ) ]  [ APOST | QUOTE ][ PERIOD | COMMA ]
```

Use the operands to override defaults specified in your Access profile.

SSN(*ssn*)
The subsystem name. If a subsystem name is omitted, the name from your most recent Access session is used. If subsystem is not profiled, Access prompts you to re-enter the command with a subsystem name.

DB2LOC(*locname*)
To execute the target SQL on a remote DB2 subsystem, use the name of the subsystem, as it is defined to Optim.

SQLID(*sqlid*)
The SQL ID. If an SQL ID is omitted, the SQL ID from your most recent Access session is used. If SQL ID is not profiled, Access prompts for one.

QUAL(*qual*)
The qualifier for any unqualified table, view, or alias names in SELECT, DELETE, INSERT, and UPDATE SQL statements. If **QUAL** is omitted, DB2 uses the current SQL ID.

LANG(*lang*)
The language of the target SQL statement specified as ASM, PLI, FOR, COB, or C. The language determines the format and character interpretation for the SQL statement. If the value for LANG is inconsistent with the current data, an error message is displayed.

By default, Access determines the language when FOPFAD2P is executed, testing for COBOL, PLI, or C, in that order. If the language cannot be determined, an error message is displayed. For more information on how Access determines the language, see page “Determining the Language” on page 93.

[APOST|QUOTE]
The string delimiter used in the target SQL. The default is APOST.

[PERIOD|COMMA]
The decimal point character used in the target SQL statement. The default is PERIOD.

SQL Statements that Directly Affect Data

Several SQL statements, such as INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, GRANT, and REVOKE, directly affect the data.

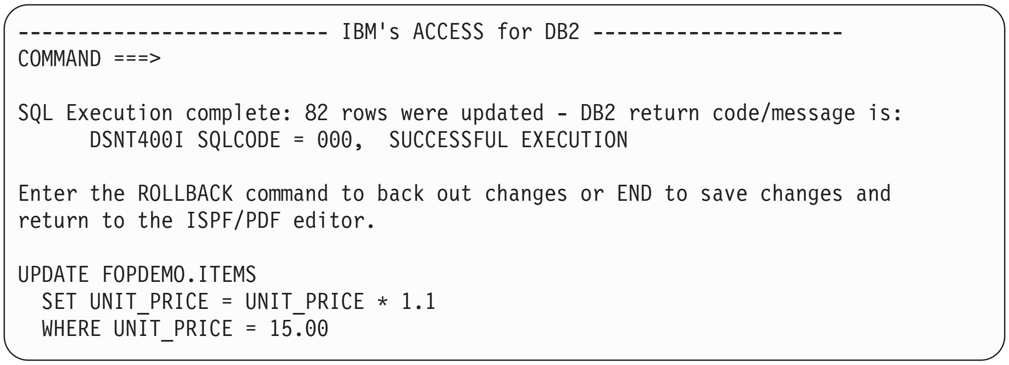
When these SQL statements are executed using FOPD2EXE, Access displays a message showing the return code, the number of rows affected, and the SQL statement executed. You are prompted to specify whether you want to ROLLBACK the changes resulting from the SQL statement or return to the ISPF/PDF editor after saving your changes.

For example, you may want to ROLLBACK your changes if they affected more rows than expected. When executing the SQL with host variables iteratively, you may use COMMIT or ROLLBACK following each iteration or after the final iteration.

When you use END to return to the ISPF/PDF editor, all pending database activity is committed.

Note: Access supports WHERE search-condition requests for UPDATE and DELETE, but does not support WHERE CURRENT OF cursor requests.

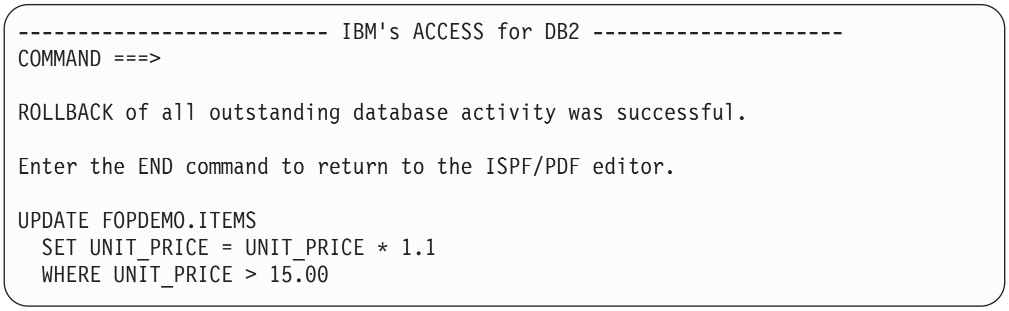
The following figure shows the display after the successful execution of an UPDATE statement.



```
-------------------------- IBM's ACCESS for DB2 ---------------------
COMMAND ===>

SQL Execution complete: 82 rows were updated - DB2 return code/message is:
      DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION

Enter the ROLLBACK command to back out changes or END to save changes and
return to the ISPF/PDF editor.

UPDATE FOPDEMO.ITEMS
  SET UNIT_PRICE = UNIT_PRICE * 1.1
  WHERE UNIT_PRICE = 15.00
```

Figure 72. Successful UPDATE Execution by FOPD2EXE

If you request COMMIT or ROLLBACK, Access displays a message specifying whether the request was successful. The following figure shows a successful ROLLBACK message.



```
-------------------------- IBM's ACCESS for DB2 ---------------------
COMMAND ===>

ROLLBACK of all outstanding database activity was successful.

Enter the END command to return to the ISPF/PDF editor.

UPDATE FOPDEMO.ITEMS
  SET UNIT_PRICE = UNIT_PRICE * 1.1
  WHERE UNIT_PRICE > 15.00
```

Figure 73. Successful ROLLBACK Execution by FOPD2EXE

Use END to return to the ISPF/PDF editor. All pending database activity is committed.

Note: The INSERT SQL command is analyzed to determine the number of columns, if any, that were specified. If there are more columns than there are VALUES (as when a host structure is used), FOPD2EXE automatically includes additional prompts to enter a value for every column. The additional prompts follow the last host variable specified in the VALUES list.

OPEN, CLOSE, and FETCH

When the target SQL statement is a request to OPEN, CLOSE, or FETCH, the corresponding DECLARE CURSOR statement is located. The SELECT statement in the DECLARE is executed as described in the following section.

SELECT

When the target SQL statement is a SELECT request, the statement is executed and an Access edit session is automatically invoked for the selected data. If the SELECT statement references a joined view, the data is displayed as browse only.

When the edit session is invoked, the default Creator ID is the current user's SQLID or the SQLID specified on the FOPD2EXE command. If this value is different than the Creator ID on the table or tables being edited, the Creator ID must be specified with the table name.

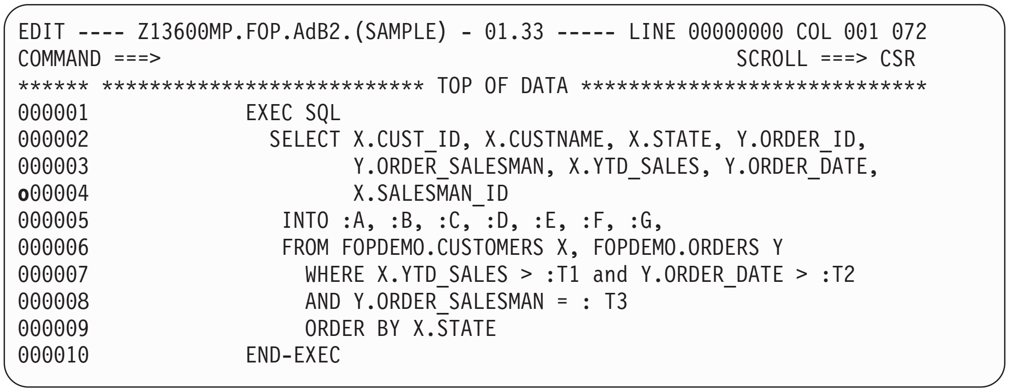
During the edit session, the data can be handled as if defined by an Access Definition. Data from other tables can be joined, and all data can be scrolled and edited. The only restriction is that the SORT, SELECTION, COLUMNS, and SHOW SQL commands cannot be used.

When you terminate the Access session using END, a COMMIT is automatically executed and the ISPF edit session resumes.

Host Variables

When the SQL statement contains host variables, Access prompts for the values for those variables prior to executing the SQL statement.

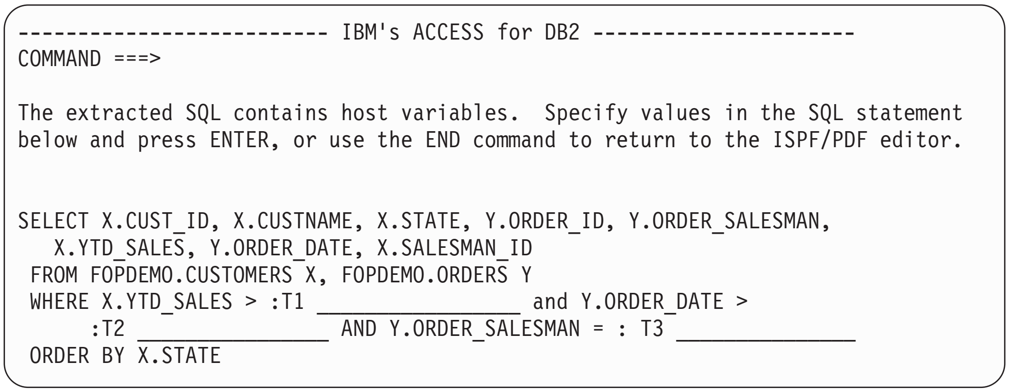
For example, assume a request has been made to execute a SELECT statement that contains three host variables as shown in the following screen segment.



```
EDIT ---- Z13600MP.FOP.AdB2.(SAMPLE) - 01.33 ----- LINE 00000000 COL 001 072
COMMAND ===>                                                  SCROLL ===> CSR
****** *************************** TOP OF DATA *****************************
000001            EXEC SQL
000002              SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID,
000003                     Y.ORDER_SALESMAN, X.YTD_SALES, Y.ORDER_DATE,
o00004                     X.SALESMAN_ID
000005               INTO :A, :B, :C, :D, :E, :F, :G,
000006               FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
000007                 WHERE X.YTD_SALES > :T1 and Y.ORDER_DATE > :T2
000008                 AND Y.ORDER_SALESMAN = : T3
000009                 ORDER BY X.STATE
000010            END-EXEC
```

Figure 74. Sample SQL Statement

Access prompts for the values with the following panel prior to executing the SQL statement.



```
-------------------------- IBM's ACCESS for DB2 ----------------------
COMMAND ===>

The extracted SQL contains host variables.  Specify values in the SQL statement
below and press ENTER, or use the END command to return to the ISPF/PDF editor.


SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID, Y.ORDER_SALESMAN,
   X.YTD_SALES, Y.ORDER_DATE, X.SALESMAN_ID
 FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
 WHERE X.YTD_SALES > :T1 ________________ and Y.ORDER_DATE >
      :T2 ________________ AND Y.ORDER_SALESMAN = : T3 _______________
 ORDER BY X.STATE
```

Figure 75. Host Variable Panel

Explanatory text is displayed on this panel. The host variables are requested within the context of the SELECT statement. When you enter alphabetic characters for host variables, they are translated to uppercase according to the setting in Personal Options. You can use the CAPS ON or CAPS OFF commands to change the Personal Options setting.

The space allocated for the values, indicated by underscores, is determined by the columnar display **Maximum Display Width** defined on the **ACCESS Editor and Display Options** panel or on the WIDTH parameter of the FOPD2EXE command.

If the executed SQL statement is SELECT and an Access edit session is invoked, the OPTIONS EDITOR command can be used to display the Access Editor and Display Options panel to modify the **Maximum Display Width** for either columnar or sidelabels display during the session.

Additional text is provided on the panel in Figure 75 only when one or more host variables are repeated. The text is as follows:

```
Host variables already referenced are marked ">Repeated<".  You may overtype
them with a value or you can leave the field alone (or blank it out) to use
the value specified in the first reference.
```

Data Type

Access assumes that any non-numeric data is a character string. Therefore, you need not supply apostrophes to mark a character string unless the string could be interpreted as a number.

If a string has trailing spaces, you should include apostrophes or use the **DB2 VarChar Delimiter** specified on the **ACCESS Editor and Display Options** panel. For example, to define the string “ABC ”, specify ABC ;.

Repeated Variable

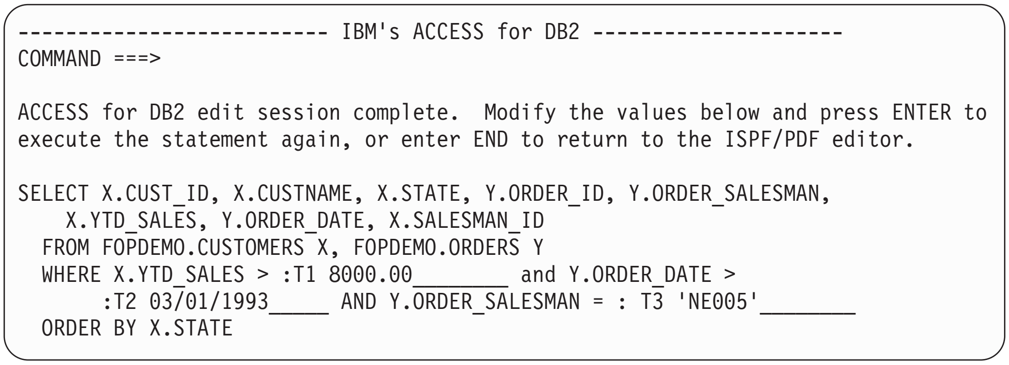
When a variable is repeated in a statement, the initial presentation of the Host Variable panel displays ">Repeated<" for each entry.

You may leave the value unedited or blank it to reuse the previously specified value or you may provide another value. The values in each repetition of a host variable do not have to be the same.

Iterative Prompting

If the SQL contains host variables, you are prompted again after the SQL is executed.

For example, assume the SQL is a SELECT statement. An Access edit session is automatically invoked. When you terminate the Access session, the Host Variable panel is redisplayed with the previous values. You can specify new values for any or all host variables.



```
-------------------------- IBM's ACCESS for DB2 ---------------------
COMMAND ===>

ACCESS for DB2 edit session complete.  Modify the values below and press ENTER to
execute the statement again, or enter END to return to the ISPF/PDF editor.

SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID, Y.ORDER_SALESMAN,
    X.YTD_SALES, Y.ORDER_DATE, X.SALESMAN_ID
  FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
  WHERE X.YTD_SALES > :T1 8000.00________ and Y.ORDER_DATE >
       :T2 03/01/1993_____ AND Y.ORDER_SALESMAN = : T3 'NE005'________
  ORDER BY X.STATE
```

Figure 76. Re-prompting for Host Variables

NULL Host Variable

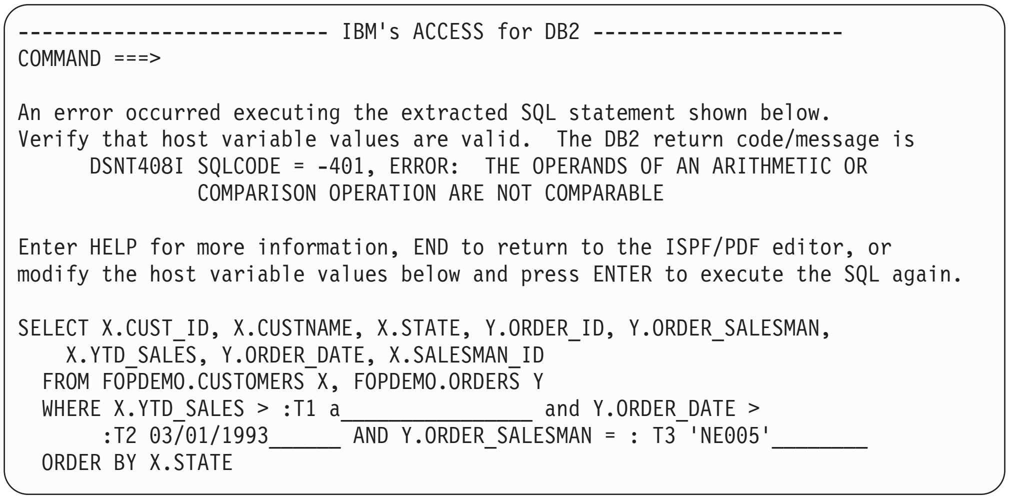
You can use the ACCESS NULL indicator character, ?, to specify NULL when the host variable is followed by a host indicator variable.

When you have completed specifying the host variable values, use ENTER to re-execute the SQL statement. Use END to return to the ISPF edit session.

Invalid Host Variable Values

If you do not specify a value for a host variable or you attempt to enter the ACCESS NULL indicator when an SQL indicator variable has not been specified, an error message is displayed. The value in error is highlighted and Access re-prompts you to specify a value. If values for the host variables are invalid, Access displays the DB2 return code and messages.

In the following figure, a non-numeric value was specified for the numeric field in the prompt for host variable T1.



```
-------------------------- IBM's ACCESS for DB2 ---------------------
COMMAND ===>

An error occurred executing the extracted SQL statement shown below.
Verify that host variable values are valid.  The DB2 return code/message is
      DSNT408I SQLCODE = -401, ERROR:  THE OPERANDS OF AN ARITHMETIC OR
               COMPARISON OPERATION ARE NOT COMPARABLE

Enter HELP for more information, END to return to the ISPF/PDF editor, or
modify the host variable values below and press ENTER to execute the SQL again.

SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID, Y.ORDER_SALESMAN,
    X.YTD_SALES, Y.ORDER_DATE, X.SALESMAN_ID
  FROM FOPDEMO.CUSTOMERS X, FOPDEMO.ORDERS Y
  WHERE X.YTD_SALES > :T1 a________________ and Y.ORDER_DATE >
       :T2 03/01/1993______ AND Y.ORDER_SALESMAN = : T3 'NE005'_______
  ORDER BY X.STATE
```

Figure 77. Invalid Value for Host Variable

You can use HELP to display the DB2 SQL ERROR panel, Figure 77, for more information about the error. You can change the values or use END to terminate the FOPD2EXE request.

Invoking the ISPF/PDF Editor

You can invoke the ISPF/PDF editor from within ACCESS by using the PDF command.

This command is useful when you have started an Access session and find that you need to execute embedded SQL in a program, but do not want to terminate your Access session.

The PDF command allows you to invoke the ISPF/PDF editor, execute the embedded SQL in your program, exit the edit session, and return to your Access session.

When used without any keywords, the PDF command displays a customized version of the ISPF/PDF EDIT entry panel. Access profiles the data set name that you enter on this panel for future use.

You can specify the command with an asterisk (*) to obtain a member selection list of the profiled data set, as shown here:

```
PDF*
```

To limit the member selection list, you can specify a partial member name with an asterisk as a wild card. For example, to obtain a selection list of members that begin with TEST, enter:

```
PDF TEST*
```

You can also specify the PDF command with a full member name to directly enter the editor for the member:

```
PDF TESTMEM
```

If you specify a member that contains SQL in the first line or lines, you can automatically execute the SQL by specifying the EXEC keyword:

```
PDF SQLMEM EXEC
```

The EXEC keyword executes the SQL using the Access FOPD2EXE command. If the member contains a SELECT statement, the FOPD2EXE command automatically invokes an Access edit session for the table(s) named in the SELECT statement.

For details about the PDF command, see the PDF command in the *Command Reference Manual*.

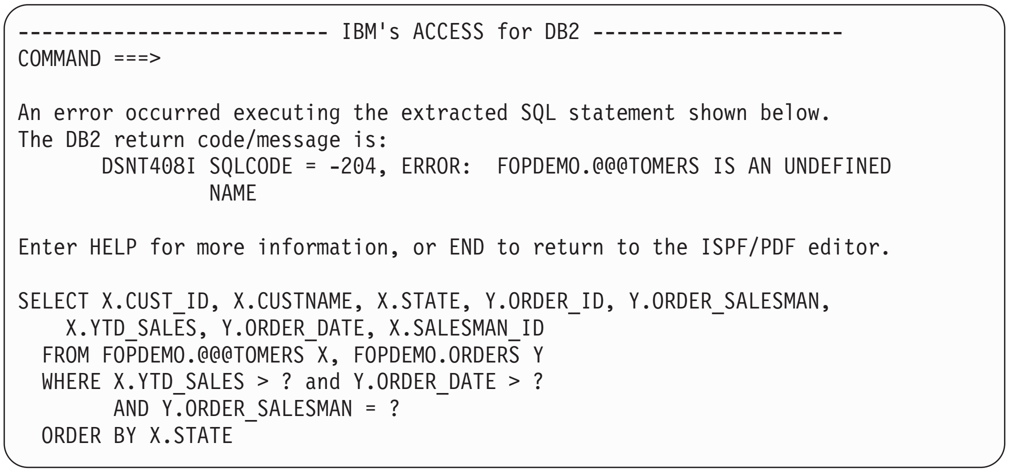
Syntax Errors

When you execute FOPFAD2P, FOPD2EXP, or FOPD2EXE, a syntax error may occur. When a syntax error is encountered, Access displays a message including the DB2 return code and messages. Additional information can be displayed.

Example

The following series of figures demonstrates handling a syntax error when FOPFAD2P is executed. (This sequence applies to FOPD2EXP and FOPD2EXE also.)

When a syntax error is encountered, a message is displayed, as shown in the following panel.

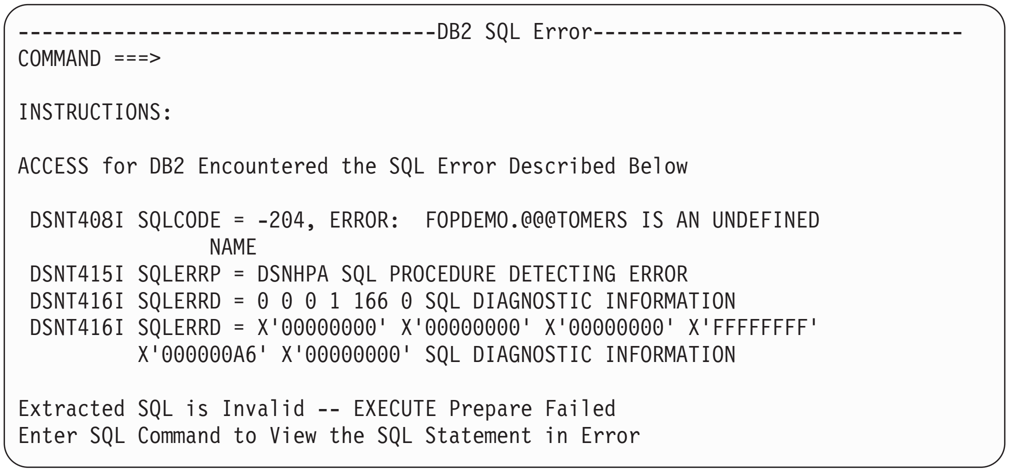


```
-------------------------- IBM's ACCESS for DB2 ---------------------
COMMAND ===>

An error occurred executing the extracted SQL statement shown below.
The DB2 return code/message is:
       DSNT408I SQLCODE = -204, ERROR:  FOPDEMO.@@@TOMERS IS AN UNDEFINED
                NAME

Enter HELP for more information, or END to return to the ISPF/PDF editor.

SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID, Y.ORDER_SALESMAN,
    X.YTD_SALES, Y.ORDER_DATE, X.SALESMAN_ID
  FROM FOPDEMO.@@@TOMERS X, FOPDEMO.ORDERS Y
  WHERE X.YTD_SALES > ? and Y.ORDER_DATE > ?
        AND Y.ORDER_SALESMAN = ?
  ORDER BY X.STATE
```

Figure 78. DB2 SQL Syntax Error Detected by FOPFAD2P

Note the question mark to represent the location of the host variables in the SQL statement.

You can use END to return to the ISPF editor or HELP to display the DB2 SQL Error panel, as shown in the following panel.



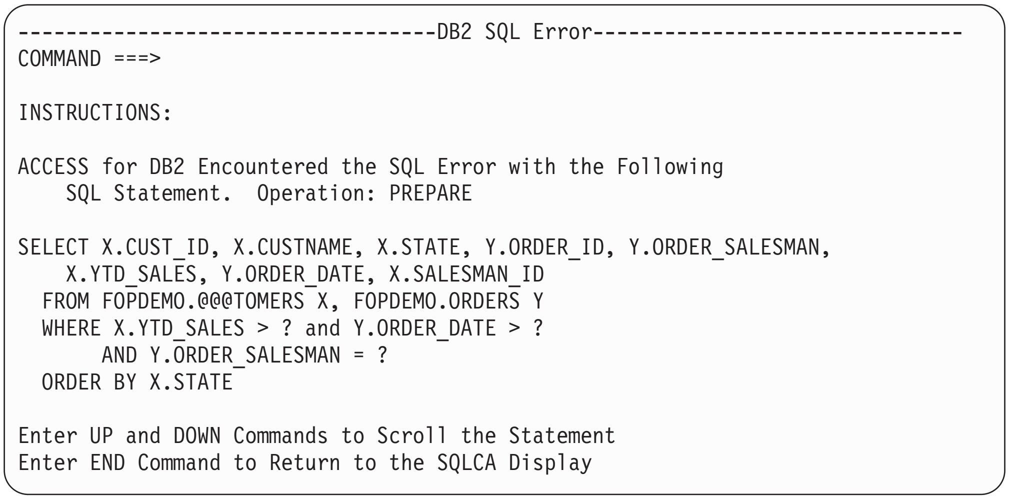
```
-----------------------------------DB2 SQL Error-------------------------------
COMMAND ===>

INSTRUCTIONS:

ACCESS for DB2 Encountered the SQL Error Described Below

 DSNT408I SQLCODE = -204, ERROR:  FOPDEMO.@@@TOMERS IS AN UNDEFINED
                NAME
 DSNT415I SQLERRP = DSNHPA SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD = 0 0 0 1 166 0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD = X'00000000' X'00000000' X'00000000' X'FFFFFFFF'
          X'000000A6' X'00000000' SQL DIAGNOSTIC INFORMATION

Extracted SQL is Invalid -- EXECUTE Prepare Failed
Enter SQL Command to View the SQL Statement in Error
```

Figure 79. DB2 SQL Error Panel

The DB2 SQL Error panel contains the complete error messages generated by DB2.

Use END to return to the ISPF edit session. Edit the embedded SQL statement, re-select the SQL statement, and re-execute the Access FOPFAD2P command.

Use the Access SQL command to display the SQL statement containing the error, as shown in the following panel.



```
-----------------------------------DB2 SQL Error-------------------------------
COMMAND ===>

INSTRUCTIONS:

ACCESS for DB2 Encountered the SQL Error with the Following
    SQL Statement.  Operation: PREPARE

SELECT X.CUST_ID, X.CUSTNAME, X.STATE, Y.ORDER_ID, Y.ORDER_SALESMAN,
    X.YTD_SALES, Y.ORDER_DATE, X.SALESMAN_ID
  FROM FOPDEMO.@@@TOMERS X, FOPDEMO.ORDERS Y
  WHERE X.YTD_SALES > ? and Y.ORDER_DATE > ?
       AND Y.ORDER_SALESMAN = ?
  ORDER BY X.STATE

Enter UP and DOWN Commands to Scroll the Statement
Enter END Command to Return to the SQLCA Display
```

Figure 80. Display SQL Containing Error

From this panel, use END to return to the previous panel.

Chapter 5. Sharing Access Definitions

Any number of users can share an Access Definition, regardless of how the Access Definition is obtained.

Thus, a user can select Option 5 to modify an Access Definition, while another user applies the same Access Definition from Option 4 to edit data, and yet another user requests the Access Definition using the EDIT command.

Modifying During Use

When using the Access Definition to display and edit data, you can modify it by joining to new tables and changing the sort or selection criteria.

From Option 5 ADS, the list of tables, the column specifications, and the parameter values of the Access Definition can be modified. These activities can occur simultaneously. Therefore, when an Access Definition that has been modified is terminated, Access must determine what action to take.

The action depends on certain circumstances. The value specified for **Changes to AD During Edit** option on the Access Definition Parameters panel determines whether the changes can be saved. (See the *Common Elements Manual*, section for Access Definitions, for further information on that option.)

- If that value is set to Temporary, changes are automatically abandoned when the session terminates. No further action is required by Access.
- If that value is set to Permanent, Access prompts the user to specify whether the changes should be saved. Press ENTER to save the changes, and use END to bypass saving the changes. CANCEL returns you to the editor session.

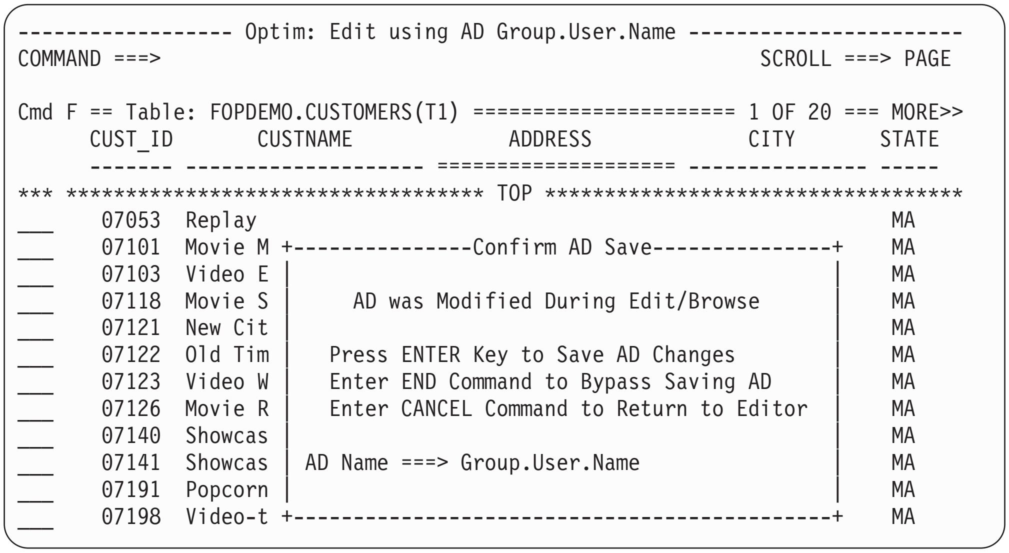
The **AD Name** value can be overtyped to specify a new name to create a new Access Definition containing the specifications of the current session. If the new name is unique, further action is not required. However, if an Access Definition of that name exists, you are prompted to confirm that the existing Access Definition is to be replaced with the current specifications.

If a new name is not specified, the changes are saved under the current name. Prior to updating the Access Definition, Access checks to ensure these changes do not conflict with changes made by other users.

Checking the Timestamp

To monitor the changes, Access Definitions are timestamped to indicate the time of the last modification. When changes to an Access Definition are saved and use of the Access Definition is terminated, Access checks to ensure that the timestamp has not changed from the time the session was initiated to the time the session terminated.

- If the timestamp has not changed, there have been no intervening changes and the requested update is performed.
- If the timestamp has changed, the user is prompted to specify whether the changes are to be saved, as shown in the following figure.



```
------------------ Optim: Edit using AD Group.User.Name -----------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ====================== 1 OF 20 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** *********************************** TOP ***********************************
___    07053  Replay                                                   MA
___    07101  Movie M +---------------Confirm AD Save---------------+    MA
___    07103  Video E |                                             |    MA
___    07118  Movie S |      AD was Modified During Edit/Browse     |    MA
___    07121  New Cit |                                             |    MA
___    07122  Old Tim |    Press ENTER Key to Save AD Changes       |    MA
___    07123  Video W |    Enter END Command to Bypass Saving AD    |    MA
___    07126  Movie R |    Enter CANCEL Command to Return to Editor |    MA
___    07140  Showcas |                                             |    MA
___    07141  Showcas | AD Name ===> Group.User.Name                |    MA
___    07191  Popcorn |                                             |    MA
___    07198  Video-t +---------------------------------------------+    MA
```

Figure 81. Confirm AD Save Prompt

The user may abandon the changes for this session or continue with the update to the Access Definition. If the user elects to update the Access Definition, any intervening changes by other users are abandoned.

Chapter 6. Error Recovery for Concurrent Updates

This appendix explains how Access handles error recovery processing when concurrent updates to the data occur. Access prompts you to determine whether your updates take precedence or are ignored.

Recovery Situations

Error recovery processing is needed to handle those situations in a multi-user environment when another user has modified one or more rows between the time you fetched the row and the time you modified it.

These changes can occur because Access does not apply locks to the rows until you actually modify the data. Fetching the data, regardless of whether you are browsing or editing, does not automatically cause locks to be applied.

Certain typical, concurrent update situations must be resolved. Error recovery is needed when you:

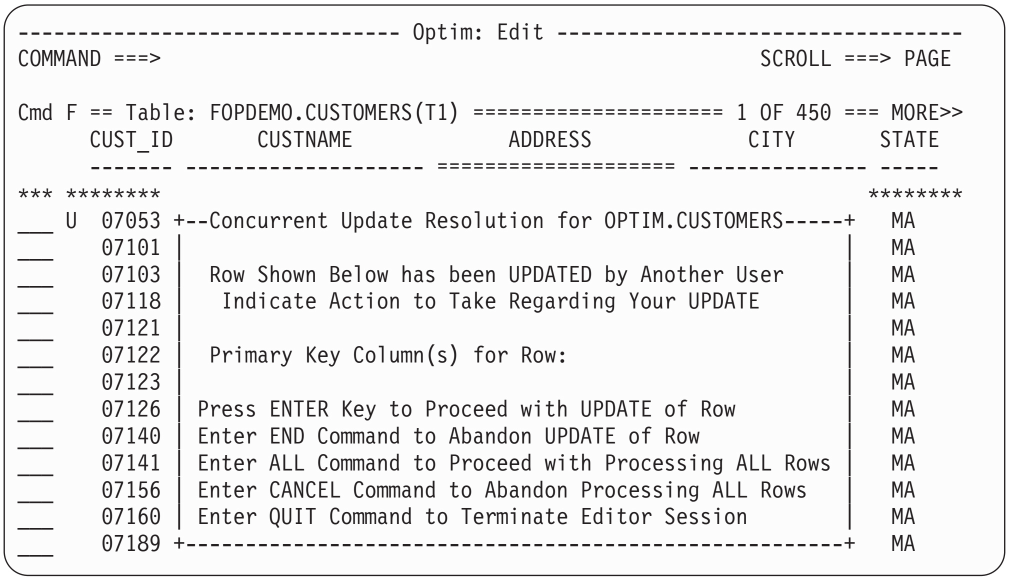
- Update a row that another user has updated.
- Update a row that another user has deleted.
- Delete a row that another user has updated.

If you delete a row that another user has deleted, Access realizes there is no conflict. There is no need to notify you and processing continues.

The error recovery prompts reflect the current activity. The possible situations are discussed in the following sections.

Update a Modified Row

If you attempt to update a row that has been modified since Access fetched the row, the following prompt is displayed.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ==================== 1 OF 450 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ********                                                          ********
___ U  07053 +--Concurrent Update Resolution for OPTIM.CUSTOMERS-----+   MA
___    07101 |                                                       |   MA
___    07103 |  Row Shown Below has been UPDATED by Another User     |   MA
___    07118 |   Indicate Action to Take Regarding Your UPDATE       |   MA
___    07121 |                                                       |   MA
___    07122 |  Primary Key Column(s) for Row:                       |   MA
___    07123 |                                                       |   MA
___    07126 | Press ENTER Key to Proceed with UPDATE of Row         |   MA
___    07140 | Enter END Command to Abandon UPDATE of Row            |   MA
___    07141 | Enter ALL Command to Proceed with Processing ALL Rows |   MA
___    07156 | Enter CANCEL Command to Abandon Processing ALL Rows   |   MA
___    07160 | Enter QUIT Command to Terminate Editor Session        |   MA
___    07189 +-------------------------------------------------------+   MA
```

Figure 82. Concurrent Update Resolution

The panel provides the following information:

Primary Key Column(s) for Row

The columns that comprise the primary key are listed. Space permitting, the entire set of columns in the primary key is displayed. Otherwise, all columns that can fit are displayed with the notation, “First n Columns of Primary Key for Row,” where ‘n' is the number of columns that are listed.

The first 25 characters of the data are displayed, along with the column name. You can use this information to determine which row that you are attempting to update has been modified.

Possible Actions

The possible actions are listed on the panel:

ENTER

Update this row with your changes and disregard any changes that occurred between the time the data was fetched and your changes. Continue processing.

END Abandon your changes to this row, but continue processing.

ALL Update all rows with your changes, regardless of future conflicts, without further prompting. ALL is useful when a global change has resulted in many updated rows and you do not want to be prompted for each changed row.

CANCEL

Abandon all updated rows that cause a conflict, without further prompting. Process the other changes.

QUIT Terminate the edit session without committing the changes.

Undo an Updated Row

Since the Access Undo command results in an update to the database to reinsert the original copy of a changed row, you may encounter a concurrent update when the Undo command is issued. Except for the wording, the prompt displayed is the same as that in Figure 82 on page 109.

```
Row Shown Below Has Been UPDATED By Another User
Indicate Action to Take Regarding Your UNDO
```

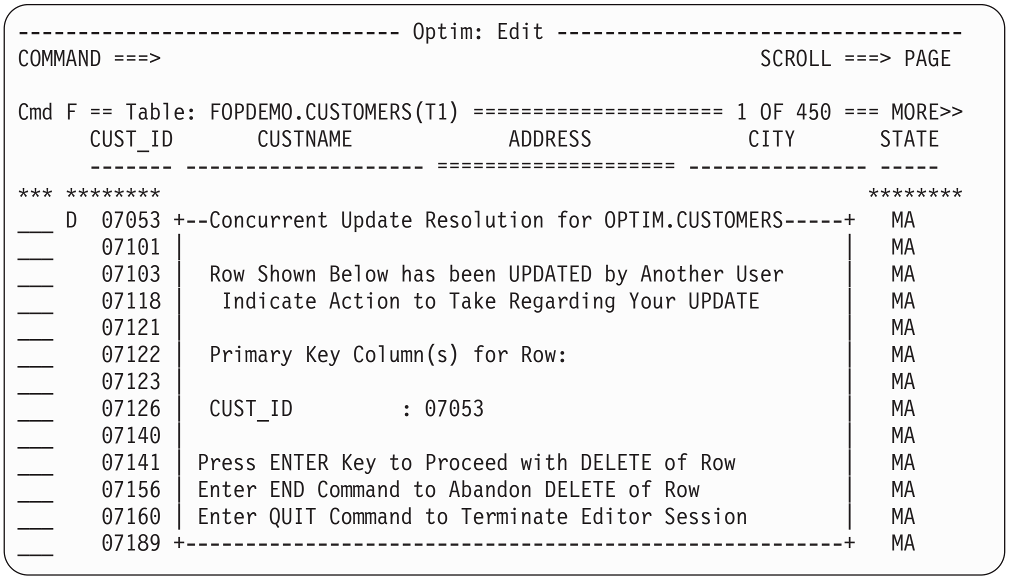
Update a Deleted Row

If you are attempting to update a row that has been deleted since Access fetched the row for you, Access prompts as in Figure 82 on page 109. It is handled the same as when updating a row modified by another user. The wording on the descriptive text reflects this state.

```
Row Shown Below Has Been DELETED by Another User
Indicate Action to Take Regarding Your UPDATE
```

Delete an Updated Row

If you attempt to delete a row that has been modified since Access fetched the row, the following prompt is displayed.



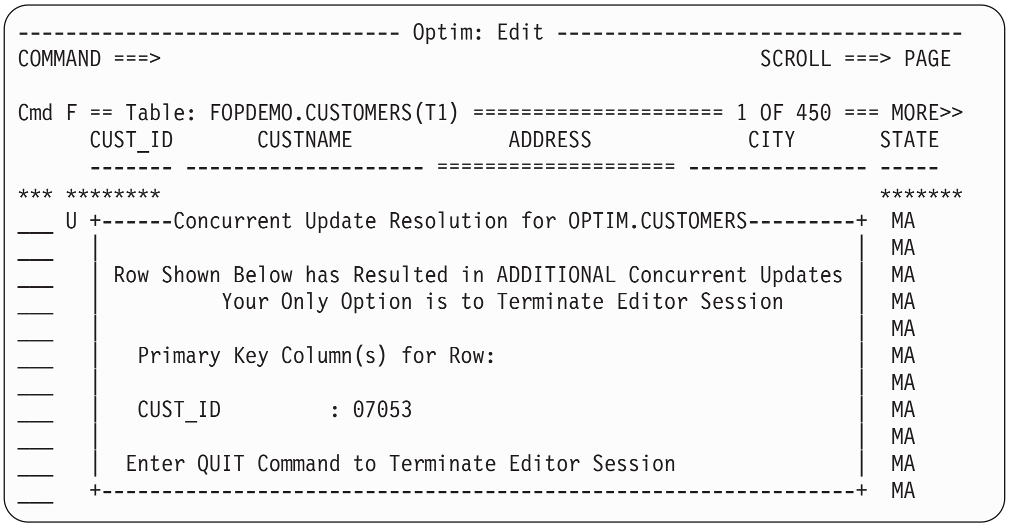
```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 450 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ********                                                          ********
___ D  07053 +--Concurrent Update Resolution for OPTIM.CUSTOMERS-----+   MA
___    07101 |                                                       |   MA
___    07103 |  Row Shown Below has been UPDATED by Another User     |   MA
___    07118 |   Indicate Action to Take Regarding Your UPDATE       |   MA
___    07121 |                                                       |   MA
___    07122 |  Primary Key Column(s) for Row:                       |   MA
___    07123 |                                                       |   MA
___    07126 |  CUST_ID         : 07053                              |   MA
___    07140 |                                                       |   MA
___    07141 | Press ENTER Key to Proceed with DELETE of Row         |   MA
___    07156 | Enter END Command to Abandon DELETE of Row            |   MA
___    07160 | Enter QUIT Command to Terminate Editor Session        |   MA
___    07189 +-------------------------------------------------------+   MA
```

Figure 83. Deleting an Updated Row

As shown in Figure 82 on page 109, the **Primary Key Column(s) for Row** prompt is displayed along with the first 25 characters of data for each column. The possible actions are listed on the panel.

Another Intervening Update

It is possible, although highly unlikely, that a second modification was made while you were responding to an error recovery prompt. In that case, Access considers the row too volatile and requests that you terminate the current edit session by displaying the following prompt.



```
-------------------------------- Optim: Edit ---------------------------------
COMMAND ===>                                                 SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 450 === MORE>>
      CUST_ID       CUSTNAME            ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----
*** ********                                                           *******
___ U +------Concurrent Update Resolution for OPTIM.CUSTOMERS---------+  MA
___   |                                                               |  MA
___   |  Row Shown Below has Resulted in ADDITIONAL Concurrent Updates |  MA
___   |           Your Only Option is to Terminate Editor Session     |  MA
___   |                                                               |  MA
___   |   Primary Key Column(s) for Row:                              |  MA
___   |                                                               |  MA
___   |   CUST_ID         : 07053                                     |  MA
___   |                                                               |  MA
___   |  Enter QUIT Command to Terminate Editor Session               |  MA
___   +---------------------------------------------------------------+  MA
```

Figure 84. Intervening Update

As shown in Figure 82 on page 109, the **Primary Key Column(s) for Row:** prompt is displayed along with the first 25 characters of data for each column. The only possible action is to QUIT.

Multiple Rows with Same Primary Key Value

Unlike DB2 primary keys, Access-defined primary keys do not have to be unique. When an Access-defined primary key is used, multiple rows with the same primary key value may exist. When you update or delete a row with a non-unique primary key, Access attempts to locate the row that matches the fetched row. If a matching row is located, the update or delete request is performed. If a matching row is not located, Access handles the update or delete request as for any concurrently updated row, except the message on the prompt is different. The first line of the message is:

```
Multiple Rows with same Primary Key Value Exist
```

The remainder of the prompt and the processing actions are as documented previously in this section.

Chapter 7. Using the Audit Facility

The Access Audit Facility tracks changes made to DB2 data during an Access edit session.

The audit data is stored in DB2 to provide a secure, centralized location. This also ensures that data in the audit table is consistent with the changes made to the actual data since all ROLLBACK and COMMIT processing affects the audit as well as the data.

Only changes made to the DB2 data during an Access session are tracked by the Audit Facility. This does not include any activity that results from executing an embedded SQL statement, unless it is a SELECT statement that invokes an Access edit session.

The Access Audit Facility is available to users if:

- The audit table, SYSFOP.ADB2AUDIT, has been created as part of the installation procedure.
- Site management has activated the facility.

You can determine whether the Audit Facility is available by displaying the Editor and Display Options panel from Option 0 on the **Main Menu**.

If **Audit Mode** is protected and set to ON, the facility is always active and you cannot deactivate it without proper authority. If **Audit Mode** is protected and set to OFF, the facility is not active and you cannot activate it without proper authority. If **Audit Mode** is unprotected, you can specify Yes to activate or No to deactivate the Audit Facility for your edit session. (See the *Common Elements Manual*, section for Editor and Display Options, for more information on **Audit Mode.)**

Displaying the Audit Information

You can display the data stored in the audit table the same way you display any other DB2 data.

You can create an Access Definition specifying the table SYSFOP.ADB2AUDIT or use **Main Menu** Option 1 to directly browse the table. Regardless of how you display the data, you will probably want to specify selection criteria before displaying the data since this table can be quite large.

SYSFOP.ADB2AUDIT Table Structure

SYSFOP.ADB2AUDIT contains the following columns:

TYPE The type of data:

HDR Header. One header row is generated for each table during an edit session. The header displays the names for each column in the audited row. Note that if the column specifications for a table are modified during the edit session, the header is generated again.

BEF Before image of a row. This indicates that the **IMAGE** column contains the original version of an updated or deleted row.

AFT After image of a row. This indicates that the **IMAGE** column contains the updated version of a row or an inserted row.

ACTION
The action that was applied to the data:

DEL The row has been deleted.

UPD The row has been updated. If the **ACTION** is UPD, two images are stored and two audit rows are created. The first contains the before image and the second contains the after image.

INS The row has been inserted.

TIMESTAMP
The time of the activity.

USERID
The TSO user ID of the user performing the activity.

TBCREATOR
The Creator ID of the table containing the edited row.

TBNAME
The name of the table containing the edited row.

SEQNO
The sequence number of the edited row. This contains a value other than 1 only when the length of the audited edited row is greater than 3950 characters. The **IMAGE** column, provided to contain an image of the row, is defined with a maximum length of 3950 characters. If the image exceeds this, ACCESS divides it over multiple audit rows and uses this column to sequence the audit rows in ascending order.

IMAGE
The image of the edited row. For columns that are expressions, ACCESS shows "n/a" on inserted rows since the value cannot be determined. This column can contain a maximum of 3950 characters.

Specifying Sort and Selection Criteria

SYSFOP.ADB2AUDIT has been designed to facilitate specifying selection and sort criteria. The following are some examples of these specifications.

- To display changes made during a specific edit session, specify selection criteria for **USERID** and **TIMESTAMP**.
- To display specific types of changes, such as all inserted rows, specify selection criteria for **ACTION**. You can sort the data by **TIMESTAMP** to display the activity chronologically, and sort by **TBCREATOR** and **TBNAME** to display by table.
- To display the activity for a specific table in chronological order, specify selection criteria for **TBCREATOR** and **TBNAME** and sort the rows by **TIMESTAMP**, **TYPE**, and **SEQNO**. To display the before image prior to the after image, specify the sort for **TYPE** as descending.

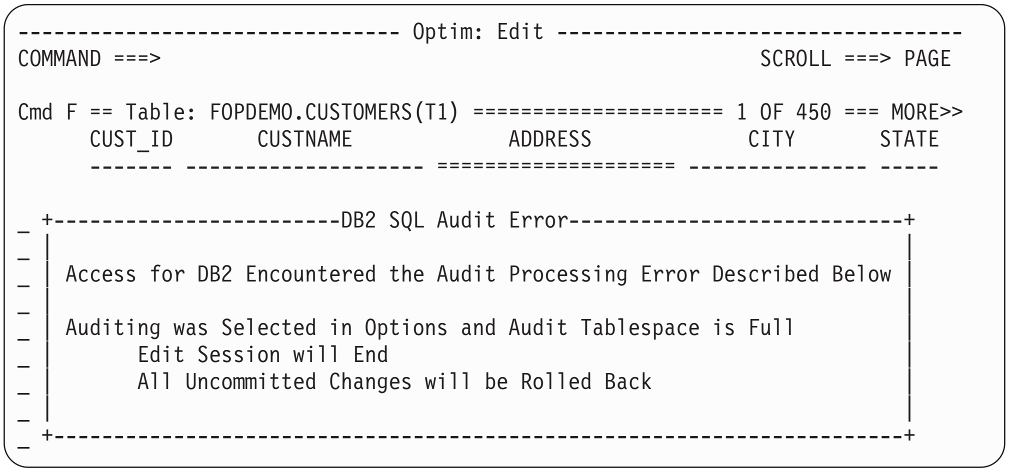
Expanding the Image Column

Since the length of the **IMAGE** column exceeds the display width for an individual column, use the EXPAND command to display the entire contents of the column.

Audit Processing Errors

If an error occurs while inserting into the audit table, Access displays an appropriate error message. Since DB2 may have performed a ROLLBACK for all uncommitted changes, the Access edit session is terminated.

In the following figure, a message is displayed because the audit table has become full during an edit session.



```
-------------------------------- Optim: Edit ----------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Cmd F == Table: FOPDEMO.CUSTOMERS(T1) ===================== 1 OF 450 === MORE>>
      CUST_ID       CUSTNAME             ADDRESS             CITY       STATE
      ------- -------------------- ==================== --------------- -----

_ +------------------------DB2 SQL Audit Error----------------------------+
_ |                                                                       |
_ | Access for DB2 Encountered the Audit Processing Error Described Below |
_ |                                                                       |
_ | Auditing was Selected in Options and Audit Tablespace is Full         |
_ |       Edit Session will End                                           |
_ |       All Uncommitted Changes will be Rolled Back                     |
_ |                                                                       |
_ +-----------------------------------------------------------------------+
```

Figure 85. Auditing Error Message

Chapter 8. Interfacing to BMC Catalog Manager

If the BMC Catalog Manager interface is installed at your site, you can use Access to browse and edit tables, views, synonyms, and local aliases from the lists displayed by the Catalog Manager.

As distributed, the interface consists of the following commands. In the following text, the term "table" refers to tables, views, synonyms, and local aliases.

BR Browse a table. When you browse a table with the Access Choose a DB2 Table/View to Browse panel, your response to the **Begin Session with** prompt on that panel is profiled. If your last response was:

- **D** Your session begins with table data.
- **C** Your session begins with a prompt for column attributes.
- **S** Your session begins with a prompt for selection criteria.
- **Q** Your session begins with a prompt for an SQL WHERE clause.

ED Edit a table. Your session begins with table data or a prompt for selection criteria or column attributes, depending on your last edit session.

BRS Specify selection criteria prior to browsing a table.

EDS Specify selection criteria prior to editing a table.

Since the command names can be changed during installation, verify the command names assigned at your site. Primary commands and line commands are available.

Primary Commands

The primary commands available to invoke Access can be entered from any list. Assume the commands BR and ED are in effect. To invoke an edit session for the CUSTOMERS table from a displayed list of tables, enter the command as:

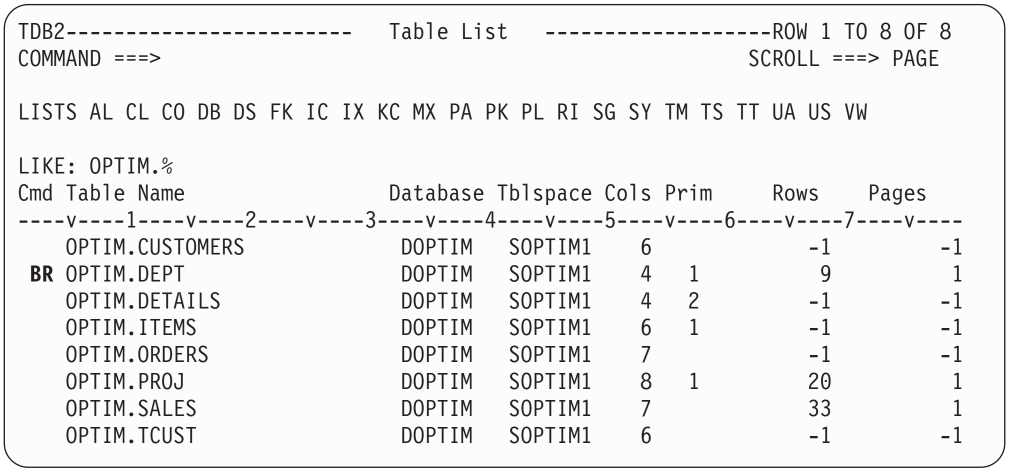
```
ED CUSTOMERS
```

To invoke Access for an object type other than that of the current list, enter the object type with the command. For example, to invoke an Access edit session for the CUSTOMERS table while displaying a list of databases, enter:

```
ED TABLE CUSTOMERS
```

Line Commands

Assume the line commands BR and ED are in effect. To browse the data in the table named OPTIM.DEPT, enter the line command as shown in the following figure.



```
TDB2------------------------   Table List   -------------------ROW 1 TO 8 OF 8
COMMAND ===>                                                 SCROLL ===> PAGE

LISTS AL CL CO DB DS FK IC IX KC MX PA PK PL RI SG SY TM TS TT UA US VW

LIKE: OPTIM.%
Cmd Table Name                 Database Tblspace Cols Prim     Rows    Pages
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
    OPTIM.CUSTOMERS             DOPTIM   SOPTIM1    6             -1          -1
 BR OPTIM.DEPT                  DOPTIM   SOPTIM1    4   1          9           1
    OPTIM.DETAILS               DOPTIM   SOPTIM1    4   2         -1          -1
    OPTIM.ITEMS                 DOPTIM   SOPTIM1    6   1         -1          -1
    OPTIM.ORDERS                DOPTIM   SOPTIM1    7             -1          -1
    OPTIM.PROJ                  DOPTIM   SOPTIM1    8   1         20           1
    OPTIM.SALES                 DOPTIM   SOPTIM1    7             33           1
    OPTIM.TCUST                 DOPTIM   SOPTIM1    6             -1          -1
```

Figure 86. Interfacing Catalog Manager with ACCESS

After the command is entered, an Access browse or edit session is invoked, as requested. All Access facilities are available. You can join tables, modify the selection and sort criteria, and save your specifications as an Access Definition.

An **>> ACCESS ENDED <<** message is displayed on the Catalog Manager table list when you terminate the Access session and return to the Catalog Manager.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator
Director of Engineering, Information Management
111 Campus Drive
Princeton, NJ 08540
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Java™ is a trademark of Sun Microsystems, Inc.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in USA