IBM InfoSphere Optim for z/OS
Version 11 Release 3

Move Introduction

IBM

IBM InfoSphere Optim for z/OS
Version 11 Release 3

Move Introduction

Note

Before using this information and the product it supports, read the information in "Notices" on page 99.

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 provides an overview of Move and Optim™ Legacy, and demonstrates key product features. You can use this document as a tutorial. The scenarios discussed are based on the sample database distributed with Optim.

For detailed information, see the *Move User Manual*.

Note: In general, when this document refers to Move, it includes the Optim Legacy functions. When the document describes specific Optim Legacy functions, it refers to Optim Legacy rather than Move.

Chapter 1. Overview

IBM® Optim for z/OS® 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 IBM 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. The Optim components include Access, Archive, Compare, and Move.

Move includes Optim Legacy. Both are relational copy facilities that migrate relationally intact sets of data from multiple sources to corresponding destinations. The underlying object definitions can also be migrated.

Note: In general, when this manual refers to Move, it includes the Optim Legacy functions. When this manual describes specific Optim Legacy functions, it refers to Optim Legacy, rather than Move.

Move is indispensable for:

- Creating test data that represents a relationally intact subset of an existing production database.
- Copying a set of related data from production to a "work area" to resolve any problems. Subsequently, corrected data can be re-introduced into production.
- Migrating subsets of data that require data transformations as part of the migration.

Note: Any application-managed relational structure in your legacy data can be replicated, using information in the Optim Directory.

Since Move runs as a TSO/ISPF application, the Help and Tutorial facility in ISPF is also supported. Menu-driven prompt screens or panels are used to specify the data to obtain and to direct processing of the extracted data. Intelligent screen-handling technology provides simultaneous display of data in multiple tables. Pop-up windows, cursor-sensitive on-line help, and tutorials aid you in using Move.

Other Optim Components

In addition to Move, the components of Optim are Access, Archive, and Compare.

Access
Access is a relational facility that lets you browse and edit related data residing in multiple DB2 tables. Use Access to verify the presence of test cases and to create additional test cases. See the *Access User Manual* for detailed information.

Archive
Archive is a relational archiving facility that allows you to identify and archive sets of relationally intact data before removing selected data from your database. Archived data is indexed and stored. You can browse, search, or restore selected subsets of archived data. See the *Archive User Manual* for detailed information.

Compare
Compare is a relational comparison facility that compares two sets of relationally intact data. The data to be compared may reside in a database or have been extracted prior to the comparison. See the *Compare User Manual* for detailed information.

Sample Database

The examples in this manual use the sample database distributed with Optim, which is described in this topic.

Move

The database contains the following DB2 tables with the Creator ID FOPDEMO.

Note: When used in the examples, the tables are shown without the OPTIM_ prefix.

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

Optim Legacy

If Optim Legacy is installed, the sample data also includes several legacy files. Like the sample database, these files are created as part of the installation process. A sequential file, FOPDEMO.BKORDER, two VSAM data sets, FOPDEMO.VENDITEM and FOPDEMO.VENDOR, and four IMS data sets, FOPDEMO.FOPDEPDB, FOPDEMO.FOPDEPIX, FOPDEMO.FOPJOBDB, and FOPDEMO.FOPJOBIX, are provided.

Note: These files are named with high-order qualifiers determined at installation. Ask your site administrator for the high-order qualifiers.

For each legacy record, Optim Legacy needs a user-defined object, called a Legacy Table that describes the legacy data. Legacy Tables for records in the IMS and VSAM data sets are included in the sample data. These Legacy Tables are:

VSAM Legacy Tables

FOPDEMO.VENDITEM

FOPDEMO.VENDOR

IMS Legacy Tables

FOPDEMO.DEPARTMENT

FOPDEMO.EMPLOYEE

FOPDEMO.JOBCODE

FOPDEMO.POSITION

You can create a Legacy Table using COBOL or PL1 copybooks. Three COBOL copybooks are provided as part of the sample data and stored in the Installation Library. The distributed copybooks are:

- FOP9BKOR
- FOP9DEPT
- FOP9EMP

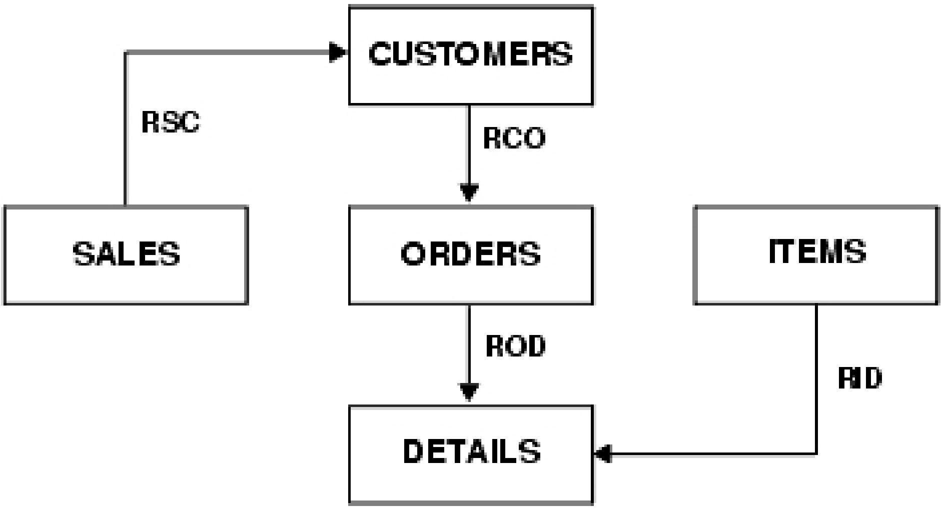
- FOP9ITEM
- FOP9JOBC
- FOP9POS
- FOP9VEND

The legacy data has a relational structure that must be replicated with user-defined relationships in order to use the legacy data in Move processes. The Optim relationship RVV (between the VENDOR and VENDITEM VSAM Legacy Tables) and the IMS relationships I1_2 (between the DEPARTMENT and EMPLOYEE IMS Legacy Tables) and I2_3 (between the EMPLOYEE and POSITION IMS Legacy Tables) are distributed with the sample data. Other relationships involving the sample legacy data are created in Chapter 5, “Scenario 3 – Legacy Tables,” on page 55.

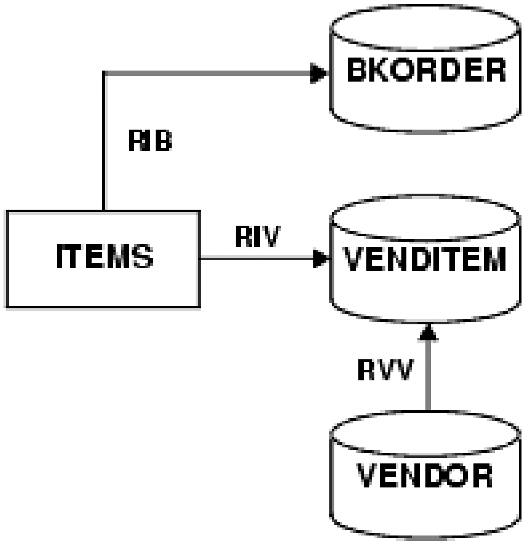
Only the tables listed in the following charts are used in this manual. (The OPTIM_ prefix for each table name is not shown in the charts.) The charts show the tables and the relationships between them for each scenario listed in this manual. The arrows indicate the flow from parent to child.

The tables and DB2 relationships used in Scenarios 1 and 2 are depicted in the following diagram. The five tables used in the scenarios are Customers, Sales, Orders, Items, and Details. The four relationships used in the scenarios are RSC, RCO, ROD, and RID:

- In the RSC relationship, Sales is the parent of Customers.
- In the RCO relationship, Customers is the parent of Orders.
- In the ROD relationship, Orders is the parent of Details.
- In the RID relationship, Items is the parent of Details.



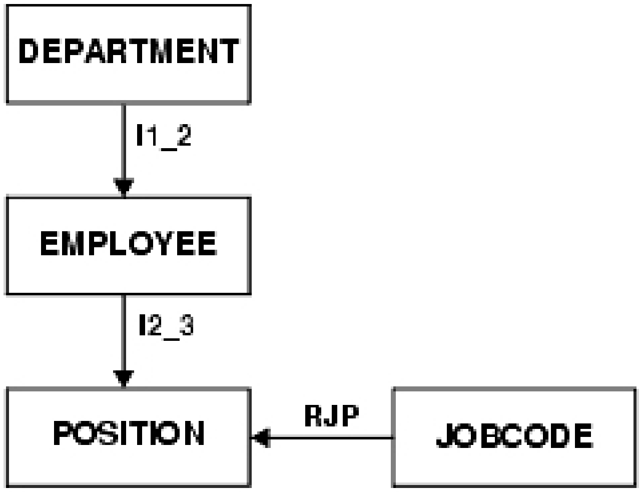
The tables, files, and Optim relationships used in Scenario 3a are depicted in the following diagram. In this scenario, you will use sample data to extract back orders from the sequential file BKORDER, related items from the DB2 table ITEMS, and related records from the VSAM files VENDITEM and VENDOR. The DB2 table ITEMS is the parent of the Legacy Tables BKORDER and VENDITEM through the Optim relationships RIB and RIV, and VENDOR is the parent of VENDITEM through the RVV relationship.



The tables, IMS data sets, and relationships (Optim and IMS) used in Scenario 3b are depicted in the following diagram. The tables and IMS data sets used in the scenario are Department, Employee, Position, and Job Code. The three relationships used in the scenarios are I1_2, I2_3, and RJP:

- In the I1_2 relationship, Department is the parent of Employee.
- In the I2_3 relationship, Employee is the parent of Position.
- In the RJP relationship, Job Code is the parent of Position.

In this scenario, you will use sample data, distributed with Optim, to extract employee records from the IMS data sets, FOPDEPDB and FOPJOBDB.

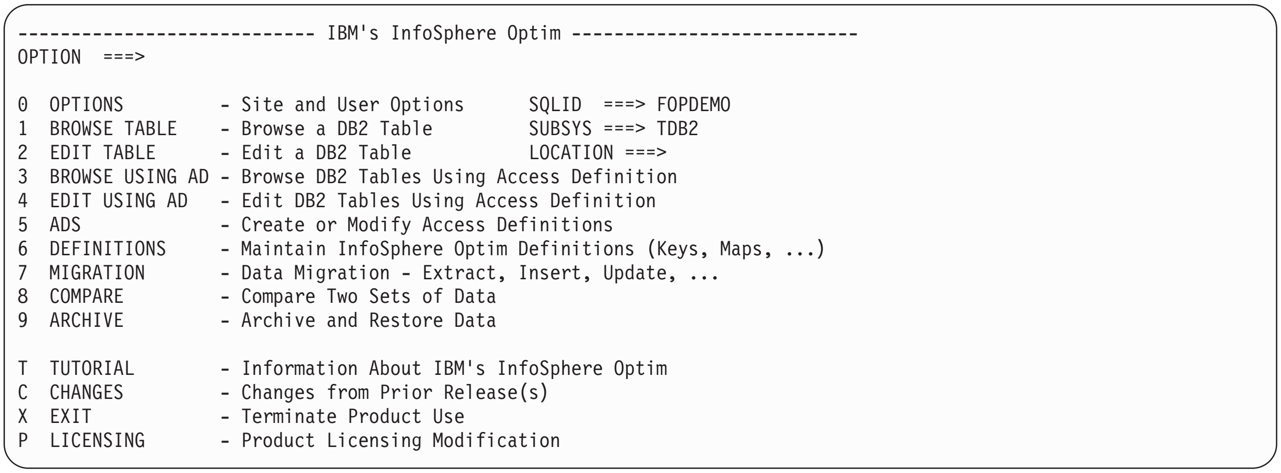


If you use this manual as a tutorial, relationships may have been added in other training functions or by other users. Also, the data in DB2 tables or legacy files may have been changed in other sessions, or by other users.

Chapter 2. Getting Started

This topic discusses getting started using Optim.

When you invoke Move, the Main Menu is displayed. The following figure shows the Main Menu that is displayed when Access, Archive, and Compare are also installed.



```
---------------------------- 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. Optim Main Menu

The panel prompts for two site-specific values, an SQLID and SUBSYS. The DB2 subsystem name, SUBSYS, must be supplied the first time you log on to Move. These values are profiled. If remote access is available, a LOCATION prompt is also displayed for the remote location. For example, you can change the displayed values for SQLID, SUBSYS, and LOCATION to extract data from one database and insert it into another.

Panel Options

The Main Menu options are:

0 OPTIONS
Specify product options, including site, user, editor and display, job card, and Legacy options.

1 BROWSE TABLE
Use Access to browse data from a DB2 table.

2 EDIT TABLE
Use Access to edit data from a DB2 table.

3 BROWSE USING AD
Use Access to browse data from DB2 using an Access Definition.

4 EDIT USING AD
Use Access to edit data from DB2 using an Access Definition.

5 ADS
Create and maintain Access Definitions.

6 DEFINITIONS
Define and maintain Legacy Tables, IMS™ Environment Definitions, and IMS Retrieval

Definitions. You can also define and maintain Optim primary keys, relationships, Access Definitions, Column Maps, Table Maps, and Archive Collections, or invoke utilities to export and import these objects.

7 MIGRATION
Perform the Move processes for extracting, inserting, loading, creating, converting, and browsing DB2 or Legacy data, or the Compare processes for extracting and comparing data.

8 COMPARE
Use Compare to compare two sets of related data and browse the results.

9 ARCHIVE
Perform the Archive processes for archiving data, browsing and searching the archives, and selectively restoring archived data.

T TUTORIAL
Display the online Tutorial.

C CHANGES
Display the enhancements for the current release.

X EXIT
Terminate the session.

P LICENSING
Display a list of the Optim components and their status (In Evaluation, Permanently Licensed, Disabled).

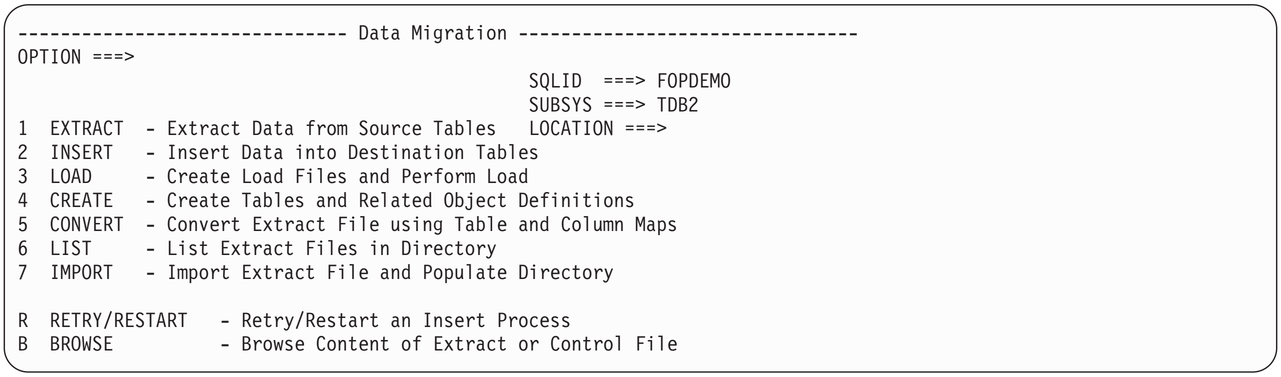
This information focuses on Option 7 MIGRATION, which can be performed online or in batch. (If you intend to execute the migration processes in batch, select Option 0 OPTIONS and specify the necessary job card information on the Job Card and Print Options panel, displayed using Option 3 JOBCARD on the Choose Option Type menu.)

Additional Information

Many of the Main Menu options are available for all or most components. See the *Common Elements Manual* for information about these common options. For information about the Access, Archive, or Compare menu options, refer to the appropriate user manual.

Data Migration Menu

Select Option 7 MIGRATION from the Main Menu to display the following panel.



```
------------------------------ Data Migration --------------------------------
OPTION ===>
                                              SQLID  ===> FOPDEMO
                                              SUBSYS ===> TDB2
1  EXTRACT  - Extract Data from Source Tables   LOCATION ===>
2  INSERT   - Insert Data into Destination Tables
3  LOAD     - Create Load Files and Perform Load
4  CREATE   - Create Tables and Related Object Definitions
5  CONVERT  - Convert Extract File using Table and Column Maps
6  LIST     - List Extract Files in Directory
7  IMPORT   - Import Extract File and Populate Directory

R  RETRY/RESTART   - Retry/Restart an Insert Process
B  BROWSE          - Browse Content of Extract or Control File
```

Figure 2. Data Migration Menu

The Data Migration menu lists options used to migrate or copy data and object definitions from a source location to a destination.

Extract Process

The first step in migrating data is to identify the source data to be extracted in the Extract Process. Normally, this source data is in several related tables. You must specify:

- The Start Table, or table from which data is first extracted.
- The tables related to the Start Table from which data is extracted.
- Optional – selection criteria for data in the tables. You can select data using column predicates, SQL WHERE clauses, or by visually choosing rows in the Start Table.
- Optional – object definitions for the tables. You can choose to extract primary key and relationship definitions, indexes, views, aliases, synonyms, column field procedure names, triggers, user-defined types and functions, and stored procedures. (DB2 and Legacy table and column definitions are always extracted.)

If you choose to extract object definitions, Move extracts them first. Move then extracts the selected Start Table rows and related rows from other tables. The extracted data and any object definitions are copied to an "Extract File."

An Extract File is a sequential file that contains the extracted data and associated information that defines the characteristics of the data. By using an Extract File, you can process the same source data repeatedly, without having to recreate it and without being concerned about changes to the original data. You can use the Extract File as input to the Insert, Load, Create, and Convert Processes, as well as the Browse facility.

Insert Process

The Insert Process inserts data from the Extract File into destination tables. You must define a Table Map to match source tables in the Extract File to destination tables, and you can define one or more Column Maps to match source columns to destination columns. You can also select insert processing to insert new rows, update processing to update existing rows, or both. This selection can apply to all tables, or you can select the type of processing on a table-by-table basis.

Load Process

The Load Process creates load files from the Extract File. The load files can be used for input to the DB2 Load Utility or another third-party load facility (for example, BMC LOADPLUS) to load large sets of data.

Create Process

The Create Process creates DB2 and Legacy Tables and related objects for which definitions are contained in the Extract File.

Convert Process

The Convert Process transforms the source data in the Extract File by applying Table Map and Column Map specifications to obtain a new or revised Extract File. Conversion is useful for masking sensitive data or for converting data to a comma separated values (CSV) format to be used with any application that supports CSV files.

List

Displays a list of Extract files that match selection criteria you specify. The List process allows you to manage extract files registered in the Optim Directory. Refer to the *Move User Manual* for full details.

Import

Select Import to create entries to register extract files in the current Optim directory. An extract file stored on tape must be registered in the Optim Directory before it can be used in a Convert, Create, Insert, or Report process. For details refer to the *Move User Manual*.

Retry/Restart

You can use the Retry or Restart Process if an Insert Process does not complete successfully. The Retry Process allows you to process rows that have been discarded (e.g., due to RI constraints or conflicting data types). Any discarded rows are flagged and can be retried without reprocessing any successfully processed rows. The Restart Process allows you to “restart” a process from the last commit point. The process can be restarted if it terminates unexpectedly due to an internal error, resource restriction, or user request.

Browse

You can use the Browse facility to view the contents of an Extract File or a Control File. Browsing an Extract File is useful prior to inserting data into the database. Browsing a Control File is useful in examining discarded rows when a process does not complete successfully.

Sample Scenarios

The scenarios in this manual focus on defining and extracting a set of source data and inserting that data at the destination. Chapter 3, “Scenario 1 – A Set of Customers,” on page 9 highlights the basic procedures used to create a relationally intact test database. Chapter 4, “Scenario 2 – A Set of Items,” on page 33 uses a more complex scenario to introduce other significant features. Chapter 5, “Scenario 3 – Legacy Tables,” on page 55 includes two scenarios that demonstrate the processes for migrating VSAM, sequential, and IMS legacy data.

Chapter 3. Scenario 1 – A Set of Customers

In Scenario 1 you extract a selection of customers and related data from the DB2 tables in the Sample Database. You then use the extracted data to create a test database.

In addition to the OPTIM_CUSTOMERS rows (no more than 1000), all related OPTIM_ORDERS and OPTIM_DETAILS rows are extracted. The data also includes all OPTIM_ITEMS rows related to the extracted OPTIM_DETAILS.

Defining the Source

Only a few steps are required to obtain the desired set of data that is then used as the source for the test database. You must:

1. Identify the Start Table (OPTIM_CUSTOMERS, in this example).
2. Provide names of related database tables.
3. Specify a selection factor and a maximum number of rows.
4. Select the relationships to be traversed.

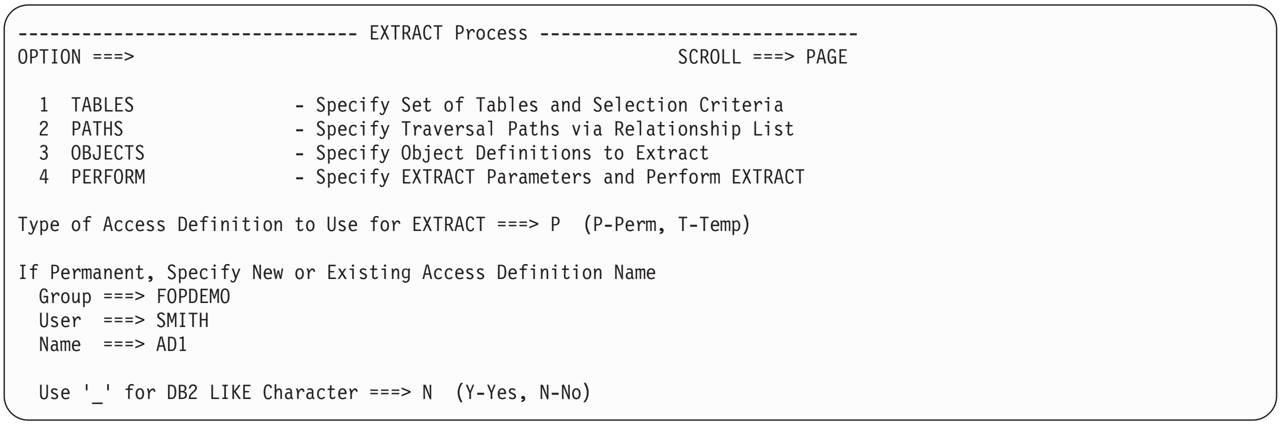
To begin, select Option 1 EXTRACT from the Data Migration menu. This option prompts you to define the source data and to extract it from the database.

Defining the Destination

After the data is extracted, you must define the destination, i.e., the test database itself. Only the names of the destination tables are needed and, if destination table names are the same as source table names, Move provides them for you. Use Option 2 INSERT from the Data Migration menu to insert the source data at the destination.

Defining the Source

Select Option 1 EXTRACT from the Data Migration menu to display the EXTRACT Process menu.



```
-------------------------------- EXTRACT Process ------------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  TABLES              - Specify Set of Tables and Selection Criteria
  2  PATHS               - Specify Traversal Paths via Relationship List
  3  OBJECTS             - Specify Object Definitions to Extract
  4  PERFORM             - Specify EXTRACT Parameters and Perform EXTRACT

Type of Access Definition to Use for EXTRACT ===> P  (P-Perm, T-Temp)

If Permanent, Specify New or Existing Access Definition Name
  Group ===> FOPDEMO
  User  ===> SMITH
  Name  ===> AD1

  Use '_' for DB2 LIKE Character ===> N  (Y-Yes, N-No)
```

Figure 3. EXTRACT Process Menu

The options on this menu correspond to the specifications you must provide for the data to be extracted.

1 TABLES
Specify the names of the tables from which data is extracted, and selection criteria for data in those tables.

2 PATHS
Specify the traversal path used to obtain data for the Extract Process. The traversal path is defined by relationships selected between tables.

3 OBJECTS
Select the type of object definitions to extract with the tables. The extracted objects can include primary keys and relationships, indexes, views, aliases, synonyms, column field procedure names, triggers, user defined types and functions, and stored procedures associated with the extracted tables. (The definitions for DB2 and Legacy Tables and columns are always extracted so that you can create these objects when they do not exist at the destination.)

4 PERFORM
Specify the parameters needed to execute the Extract Process.

Access Definition

The information describing data to be extracted (table names, relationships, selection criteria, etc.) is in an Access Definition. The Access Definition is copied to the Extract File with the data and objects.

Temporary or Permanent

You can define a temporary Access Definition for a single Extract Process. When the Extract Process is completed, the temporary Access Definition is discarded. A permanent Access Definition can be reused as needed and can be used interchangeably among the components. For this scenario, the Access Definition is permanent. Select P (for Perm) at the **Type of Access Definition to Use for EXTRACT** prompt.

Access Definition Name

A permanent Access Definition must have an identifying name. As the prompts indicate, an Access Definition name is in three parts:

```
Group
User
Name
```

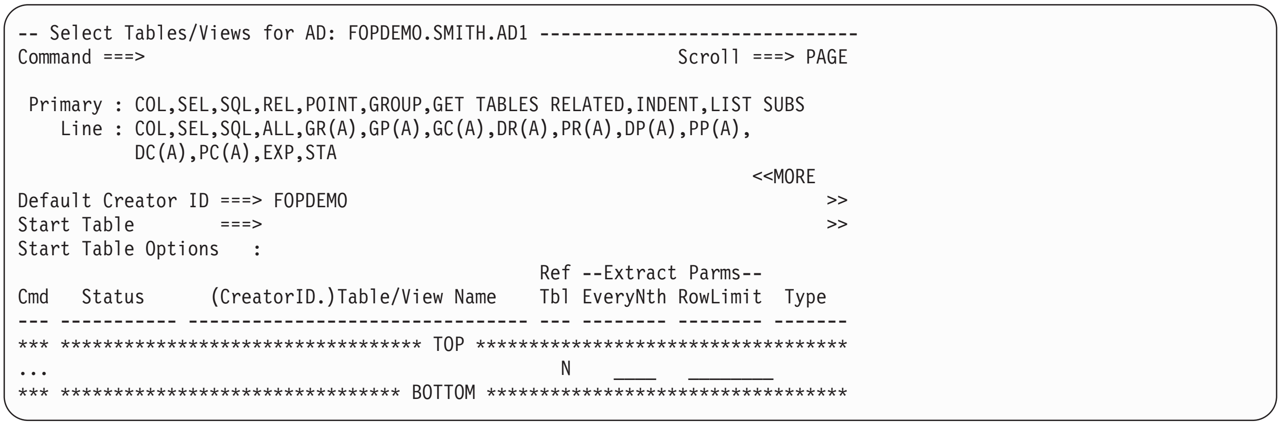
To extract data, you can use an existing Access Definition or create a new one.

- To use an existing Access Definition, type the name at the prompts, or obtain a selection list by using DB2 LIKE syntax or leaving one or more prompts blank.
- To create a new Access Definition, type the name.

For this scenario, you create an Access Definition named FOPDEMO.SMITH.AD1. Type this name or a name of your choice.

Select Tables

The data used as a source for the test database is extracted from the tables you list. To edit the Table List, select Option 1 TABLES on the EXTRACT Process menu to display the Select Tables/Views for AD panel.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                                     <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table       ===>                                                      >>
Start Table Options   :
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------ --- -------- -------- -------
*** ******************************** TOP *********************************
...                                                N    ____   ________
*** ****************************** BOTTOM ********************************
```

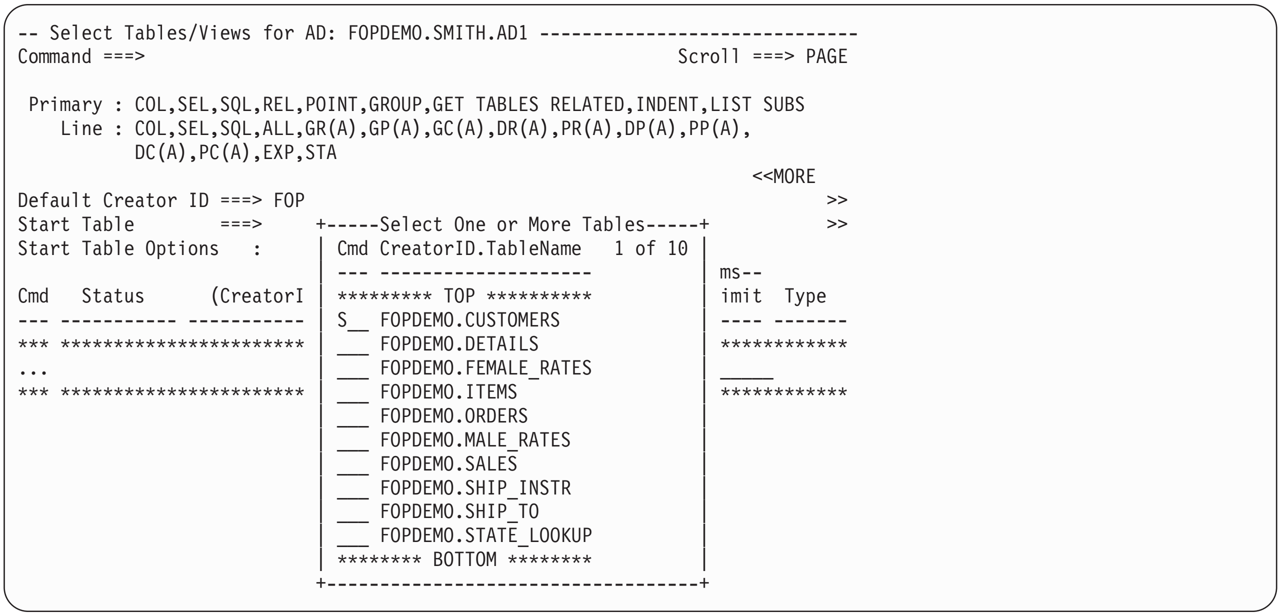
Figure 4. Select Tables/Views for AD Panel

Default Creator ID

Note the SQLID specified on the Main Menu is provided as the default Creator ID. Change **Default Creator ID** to FOPDEMO to use tables in the sample database.

LIST TABLES Command

Space on the panel allows you to type the names of the tables from which data is extracted. To save time, or if you are not sure of the names, you can request a selection list, using the LIST TABLES command. A pop-up window lists the tables with names prefixed with the **Default Creator ID**. (The width of the pop-up will vary depending on whether the listed tables include Long Object Names or LONs.)



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                                     <<MORE
Default Creator ID ===> FOP                                                 >>
Start Table       ===>    +-----Select One or More Tables-----+          >>
Start Table Options   :    | Cmd CreatorID.TableName   1 of 10 |
                           | --- --------------------          | ms--
Cmd   Status      (CreatorI | ********* TOP **********          | imit  Type
--- ----------- ----------- | S__ FOPDEMO.CUSTOMERS             | ---- -------
*** ********************** | ___ FOPDEMO.DETAILS               | ************
...                        | ___ FOPDEMO.FEMALE_RATES          | _____
*** ********************** | ___ FOPDEMO.ITEMS                 | ************
                           | ___ FOPDEMO.ORDERS                |
                           | ___ FOPDEMO.MALE_RATES            |
                           | ___ FOPDEMO.SALES                 |
                           | ___ FOPDEMO.SHIP_INSTR            |
                           | ___ FOPDEMO.SHIP_TO               |
                           | ___ FOPDEMO.STATE_LOOKUP          |
                           | ******** BOTTOM ********          |
                           +-----------------------------------+
```

Figure 5. Table Selection List

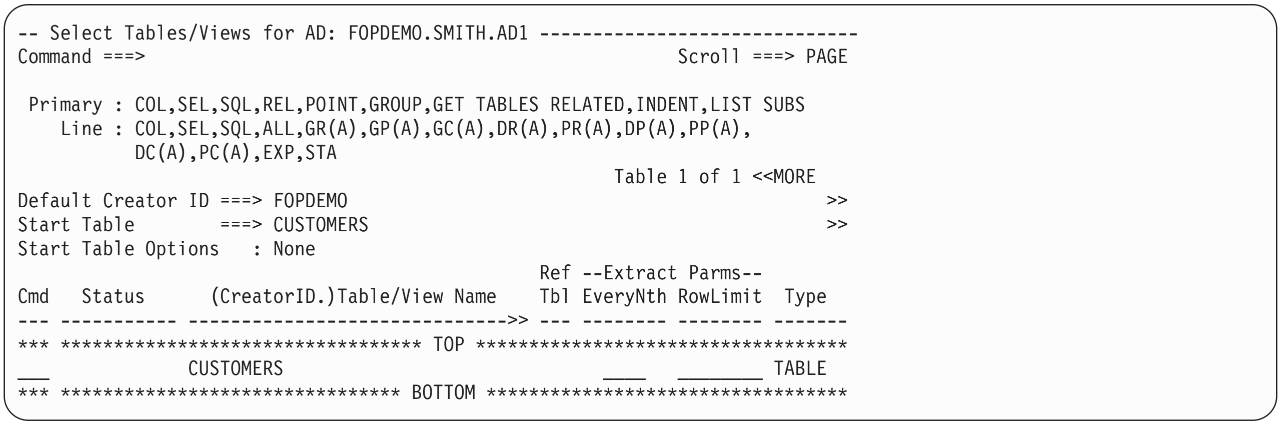
Typically, a database has many tables and you will want to keep the selection list to a manageable size. To do this, you can use DB2 LIKE syntax with the LIST TABLES command. For example, to limit the list to tables with the Creator ID FOPDEMO and a base name beginning with S, enter:

```
LIST TABLES FOPDEMO.S%
```

Applying the command in the example to tables in the sample database, the selection list includes FOPDEMO.SALES, FOPDEMO.SHIP_INSTR, FOPDEMO.SHIP_TO, and FOPDEMO.STATE_LOOKUP.

Select Table Names

You can select one or more tables from the list using the Select line command. Type S under **Cmd** for the CUSTOMERS table to select that table and press ENTER to return to the Select Tables/Views for AD panel.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 1 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> CUSTOMERS                                           >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************** TOP ***********************************
___             CUSTOMERS                              ____   ________ TABLE
*** ******************************** BOTTOM **********************************
```

Figure 6. Selected Table Name Inserted

The name of the CUSTOMERS table is inserted in the list and automatically designated as the Start Table. The Start Table is the first table from which data is extracted.

Insert Related Tables

After one or more tables have been listed, you no longer need to type names or choose from a selection list. Using the GET TABLES RELATED command, you can direct Move to automatically insert the names of tables that are related to a table on the list. This command automatically lists any parent and child tables.

You can specify the name of the table that is the subject of the command or designate it by cursor position. However, when a single table is listed, as in this example, you can use the command without designating a subject table, as in:

```
GET TABLES RELATED
```

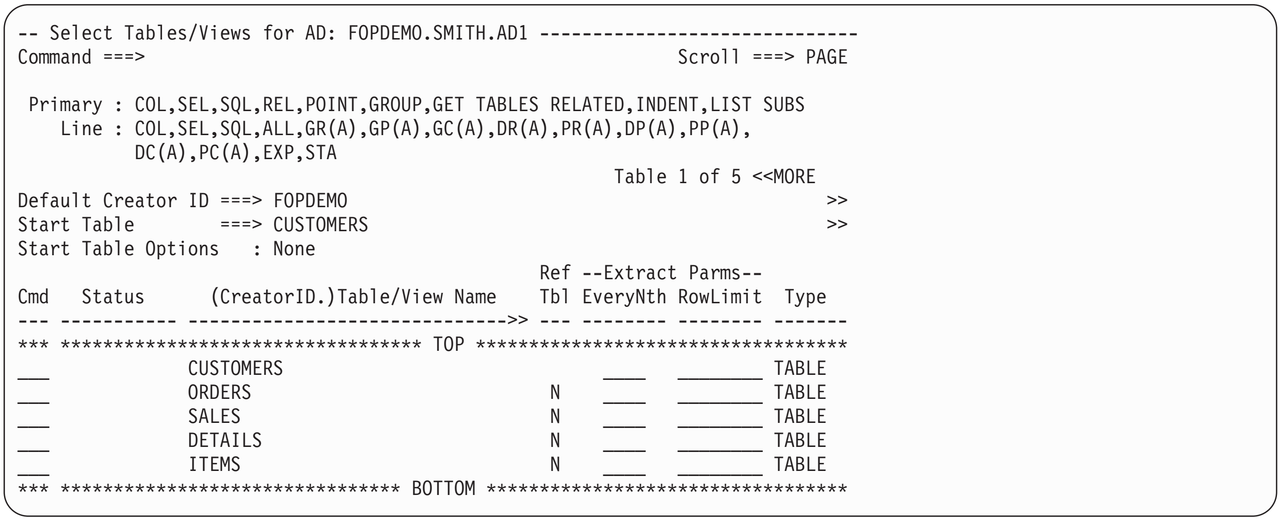
The ORDERS table and the SALES table are directly related to the CUSTOMERS table and their names are inserted.

Using the ALL operand, you can use this command to insert names of *all* related tables, including parents, children, grandparents, grandchildren, and so on. Other operands allow you to insert the names of parent or child tables only and limit the levels to be included. For example, to list two levels of children and grandchildren, enter the command as:

```
GET TABLES RELATED 2 CHILD
```

Get Related line commands are also available. Enter GR (Get Related) or GRA (Get Related All) in **Cmd** for a listed table to achieve the same results as the GET TABLES RELATED primary command. Variations of this line command limit the function to parents or children and a desired level of relationships.

For this scenario, enter the GET TABLES RELATED command with the ALL operand. Move lists the names of the SALES and ORDERS tables, which are directly related to the CUSTOMERS table in the Access Definition. Also, DETAILS is related to ORDERS and ITEMS is related to DETAILS. No other tables are related to SALES. The following figure shows the results.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 5 <<MORE
Default Creator ID ===> FOPDEMO                                            >>
Start Table        ===> CUSTOMERS                                          >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************* TOP *********************************
___            CUSTOMERS                               ____   ________ TABLE
___            ORDERS                             N    ____   ________ TABLE
___            SALES                              N    ____   ________ TABLE
___            DETAILS                            N    ____   ________ TABLE
___            ITEMS                              N    ____   ________ TABLE
*** ******************************** BOTTOM *******************************
```

Figure 7. All Related Tables Included

Manipulating the List

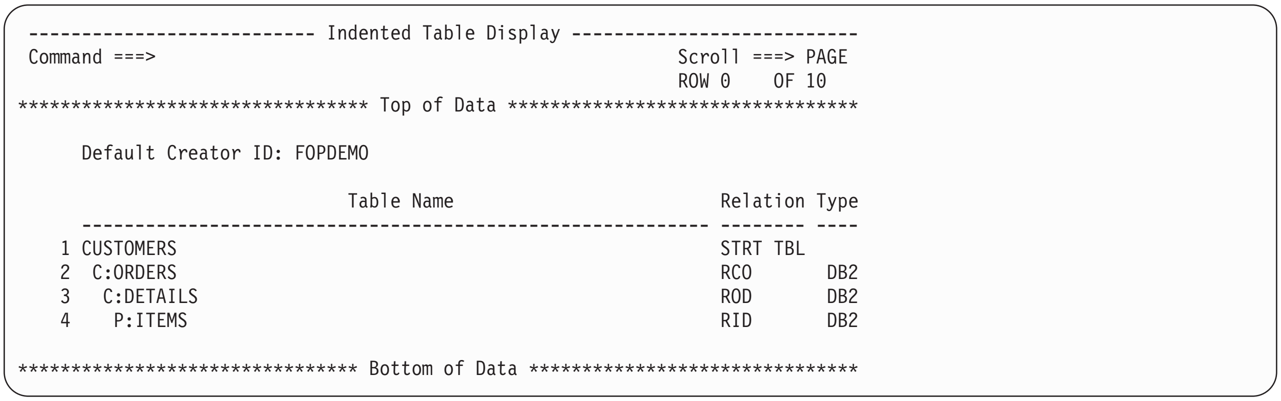
You can scroll and edit the Table List, using standard ISPF operations.

Delete Unused Table

The SALES table is not used in this scenario. Use the D (Delete) line command to remove the SALES table from the Table List.

Indented Display by Relationships

You can list the tables in an indented format that shows all relationships and illustrates the data model for listed tables. Use the INDENT command on the Select Tables/Views for AD panel to view the Indented Table Display. The four tables in the Access Definition are listed in a browse-only display with the Start Table identified and listed first. All other listed tables are identified with a prefix as either the child, C, or the parent, P, of the preceding table. The name of the relationship with the preceding table is also displayed.



```
 --------------------------- Indented Table Display ---------------------------
 Command ===>                                                 Scroll ===> PAGE
                                                              ROW 0    OF 10
********************************* Top of Data ********************************

      Default Creator ID: FOPDEMO

                         Table Name                        Relation Type
      ---------------------------------------------------- -------- ----
    1 CUSTOMERS                                            STRT TBL
    2  C:ORDERS                                            RCO      DB2
    3   C:DETAILS                                          ROD      DB2
    4    P:ITEMS                                           RID      DB2

******************************** Bottom of Data *******************************
```

Figure 8. Indented Table Display

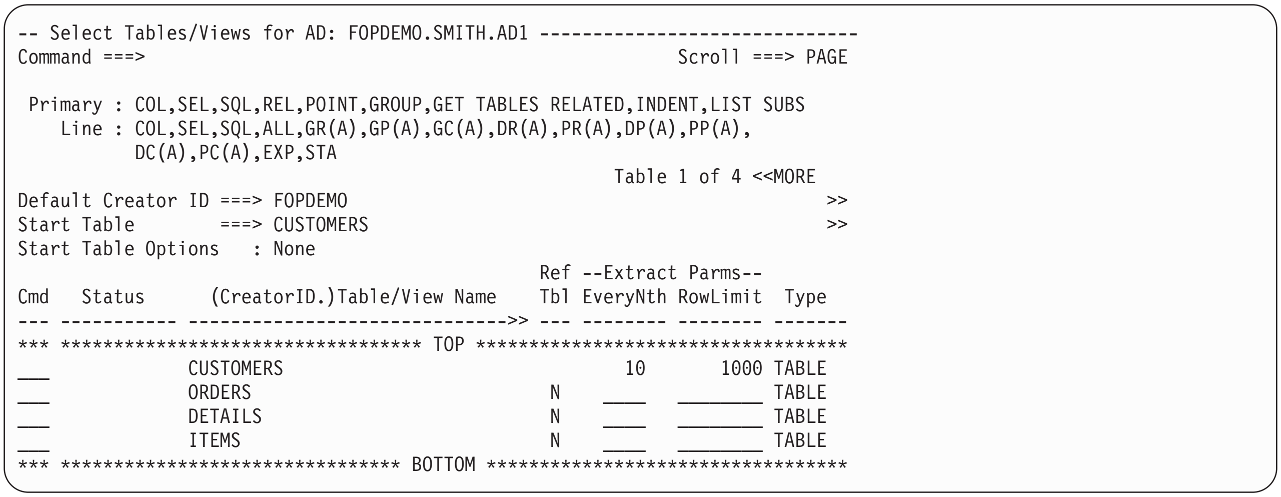
After viewing the Indented Table Display, use END to return to the Select Tables/Views for AD panel.

Table Specifications and Criteria

The Select Tables/Views for AD panel is also used to specify various table criteria. If specifying selection criteria, you must request the prompts from this panel.

Start Table

The extract for this scenario begins with a set of rows from the CUSTOMERS table. As shown in Figure 9, CUSTOMERS is the Start Table. Also, a selection factor (**EveryNth**) and a maximum for the number of extracted Start Table rows (**RowLimit**) are specified. You can enter values for **EveryNth** and **RowLimit** for any listed table.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> CUSTOMERS                                           >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************** TOP **********************************
___             CUSTOMERS                             10      1000 TABLE
___             ORDERS                            N    ____   ________ TABLE
___             DETAILS                           N    ____   ________ TABLE
___             ITEMS                             N    ____   ________ TABLE
*** ******************************** BOTTOM *********************************
```

Figure 9. Selection Factor and Row Limit

Random Factor

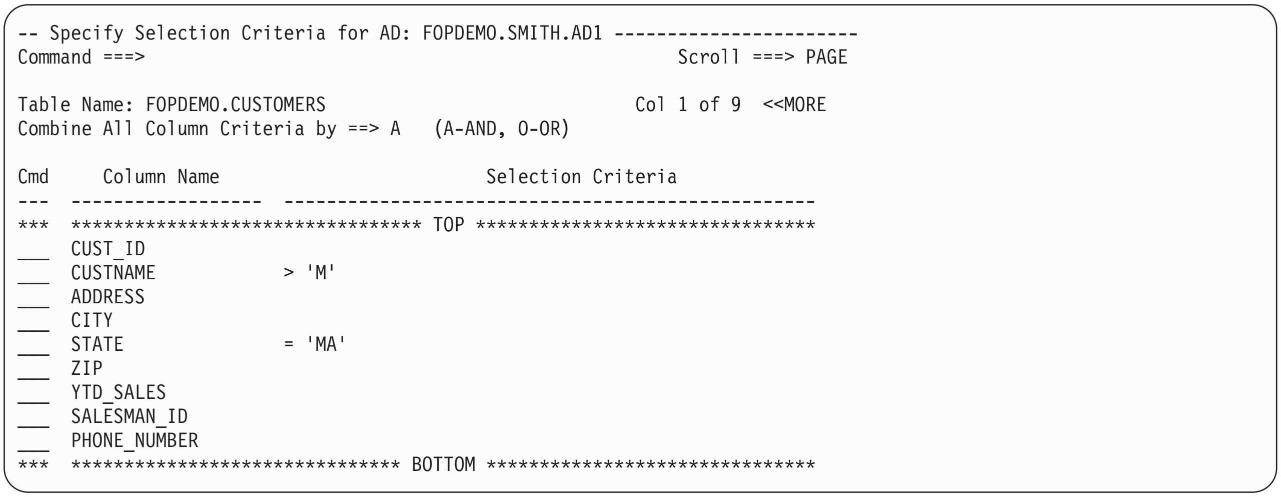
You apply a selection factor by entering a numeric value in **EveryNth**. This value directs Move to extract every nth row from the set of rows that satisfy any selection criteria. For example, to extract every tenth CUSTOMERS row, type 10 in **EveryNth**. Move extracts the tenth, twentieth, thirtieth, etc., rows from the selected set of CUSTOMERS rows.

Row Limit

To limit the number of rows extracted from a table, enter a value in **Row Limit**. In this example, 1000 is the maximum number of rows extracted from the CUSTOMERS table.

Selection Criteria

Use the Specify Selection Criteria for AD panel to enter column⌂specific criteria for each table. To display this panel, use the SEL primary or line command on the Select Tables/Views for AD panel. To specify selection criteria for the CUSTOMERS table, use the SEL line command.



```
-- Specify Selection Criteria for AD: FOPDEMO.SMITH.AD1 -----------------------
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           > 'M'
___  ADDRESS
___  CITY
___  STATE              = 'MA'
___  ZIP
___  YTD_SALES
___  SALESMAN_ID
___  PHONE_NUMBER
***  ******************************* BOTTOM *******************************
```

Figure 10. Selection Criteria for CUSTOMERS Table

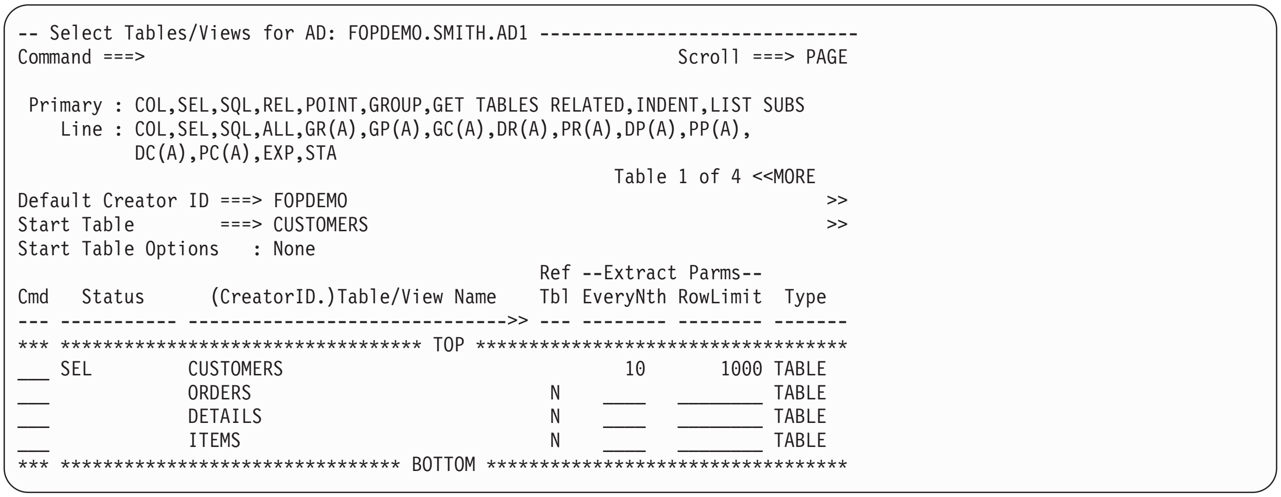
The notation at the side of the Table Name, **Col 1 of 9**, indicates the position of the first displayed column name in relation to all columns. The display in this scenario begins with the first of nine columns.

Specifying the Criteria

Selection criteria for the CUSTOMERS table is specified by supplying an appropriate operator with a corresponding value or a list of values, as shown in Figure 10. For example, entering `> 'M'` under **Selection Criteria** for the column CUSTNAME selects all customers with names that are alphabetically greater than M. Similarly, entering `= 'MA'` under **Selection Criteria** for the column STATE selects all customers from the state of Massachusetts. Further, if you specify that all criteria are combined by OR, all customers satisfying either condition are included. If the criteria are combined by AND, only customers that satisfy both conditions are included. That is, only customers whose names are alphabetically greater than M and are from Massachusetts are selected.

Status Revised

When column specifications for the current table are complete, use END to return to the Select Tables/Views for AD panel. The SEL in **Status** indicates that selection criteria is defined for the table.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 -----------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> CUSTOMERS                                           >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************** TOP ***********************************
___ SEL         CUSTOMERS                                10       1000 TABLE
___             ORDERS                            N    ____   ________ TABLE
___             DETAILS                           N    ____   ________ TABLE
___             ITEMS                             N    ____   ________ TABLE
*** ******************************** BOTTOM **********************************
```

Figure 11. Specifications Indicated

ALL command

You can remove the selection criteria for a column by redisplaying the Specify Selection Criteria for AD panel, and clearing the values. Also, you can use the ALL primary or line command on the Select Tables/Views for AD panel to remove all selection criteria defined for a table. For this scenario, enter the ALL line command in **Cmd** for the CUSTOMERS table to remove the selection criteria. The SEL indicator is removed from **Status**.

Reference Tables

Some tables are used as reference or lookup tables. All rows in a reference table are extracted without regard to relationships to rows in other tables. For example, the ITEMS table contains the complete inventory of items. It is a reference table. By default, the extract will include only the items ordered by the selected customers. However, to facilitate inserting and modifying rows in the test database, you can extract the entire ITEMS table by overtyping the N in **Ref Tbl** with Y to indicate that the ITEMS table is a reference table. For this scenario, however, the ITEMS table is not defined as a reference table so that it can be used later in this document to explain the traversal path based on the relationships.

Other Commands

Other primary and line commands, unique to Optim, are available when defining an Access Definition. Listed at the beginning of the Select Tables/Views for AD panel, these commands display panels from which you can specify additional Access Definition information. See Access Definitions in the *Common Elements Manual* for details.

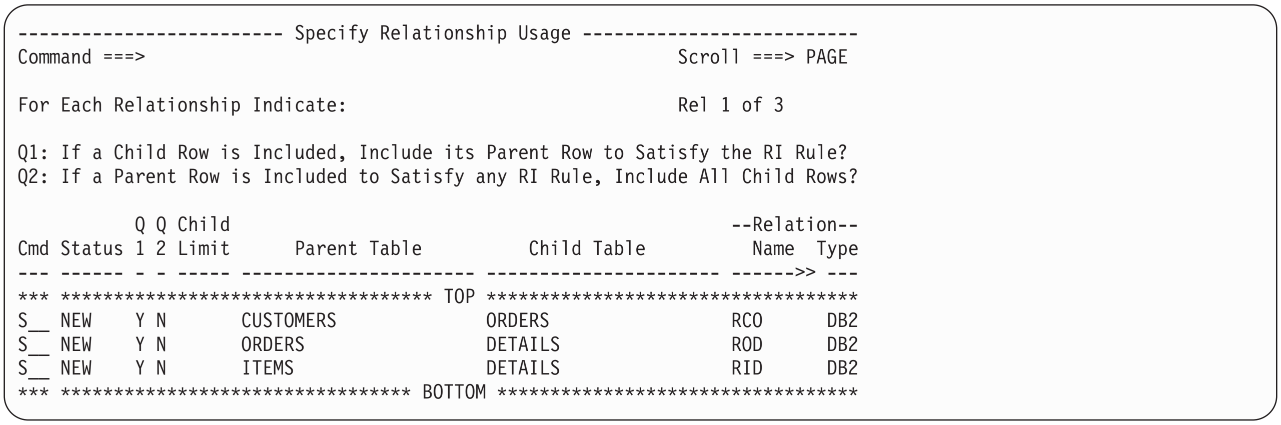
Use END to return to the EXTRACT Process menu.

Paths

After listing the table names, you can specify the traversal path. That is, you can select the relationships to be used and the direction in which relationships are traversed (i.e., from parent to child, from child to parent, or in both directions).

Move automatically lists any relationships between any pair of tables listed on the Select Tables/Views for AD panel. These relationships may be defined in the DB2 Catalog, the Optim Directory used to supplement the Catalog, or in IMS.

Select Option 2 PATHS on the EXTRACT Process menu to display the Specify Relationship Usage panel.



```
------------------------- Specify Relationship Usage ---------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                               --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- ---------------------- ---------------------- ------>> ---
*** *********************************** TOP ***********************************
S__ NEW    Y N       CUSTOMERS              ORDERS                 RCO      DB2
S__ NEW    Y N       ORDERS                 DETAILS                ROD      DB2
S__ NEW    Y N       ITEMS                  DETAILS                RID      DB2
*** ********************************* BOTTOM **********************************
```

Figure 12. Selecting Relationships

The names of all relationships between pairs of tables listed on the Select Tables/Views for AD panel are displayed. The names of parent and child tables for each relationship are provided.

The name of each relationship is shown in **Name** and the source of the relationship is shown in **Type**. All relationships in Figure 12 on page 16 are defined in the DB2 Catalog. (A relationship defined in the Optim Directory is identified as OPT.)

Status

When this panel is first displayed, all relationships have a NEW **Status** (see Figure 12 on page 16). You can use the S and U line commands to select or unselect any relationship. For this scenario, all relationships are selected. In instances in which more than one relationship exists between two tables, you can select one or more to be traversed in the Extract Process.

The Start Table is always included in the Extract Process. After rows are extracted from the Start Table, Move traverses selected relationships from parent to child. In this scenario, Move extracts the CUSTOMERS rows, and the related ORDERS rows and, finally, the DETAILS related to the extracted ORDERS.

Because ITEMS is a parent of the DETAILS table, the process must traverse from child to parent to obtain ITEMS rows related to the DETAILS. You must direct Move to traverse a relationship from child to parent by designating Y (Yes) to **Q1** for the relationship. Because it is generally desirable to select the parents of child rows in order to maintain referential integrity, the default designation for **Q1** is Y. You need not change anything on the panel.

Child Limit

At your option, you can designate a maximum number of child table rows to be extracted for each parent row in a relationship (e.g., no more than five ORDERS rows for each extracted CUSTOMERS row using the relationship RCO).

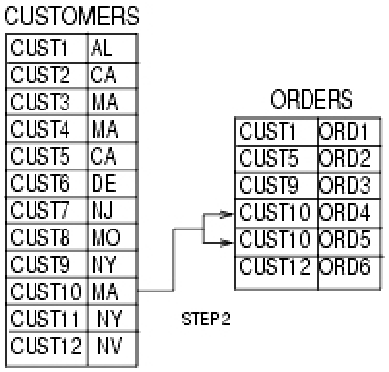
Extract Sequence

Consider the steps that Move performs to extract the data according to the selection factors, selected relationships and specifications for **Q1** and **Q2**. Simplified data for each table is used in the following example to aid in the explanation.

Sequence

CUSTOMERS to ORDERS

Traversal



CUSTOMERS

CUST1	AL
CUST2	CA
CUST3	MA
CUST4	MA
CUST5	CA
CUST6	DE
CUST7	NJ
CUST8	MO
CUST9	NY
CUST10	MA
CUST11	NY
CUST12	NV

ORDERS

CUST1	ORD1
CUST5	ORD2
CUST9	ORD3
CUST10	ORD4
CUST10	ORD5
CUST12	ORD6

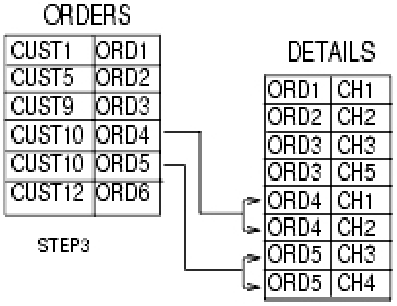
Steps

Step 1 Every tenth CUSTOMERS row is extracted based on the random selection factor.

Step 2 The ORDERS rows for these CUSTOMERS rows are extracted because of the parent to child traversal of relationship RCO.

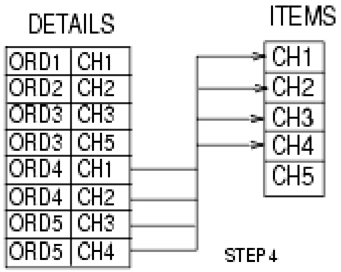
Sequence **Traversal** **Steps**

ORDERS to DETAILS



Step 3 The DETAILS that are related to the extracted ORDERS, ORD4 and ORD5, are extracted by traversing the relationship ROD from parent to child.

DETAILS to ITEMS



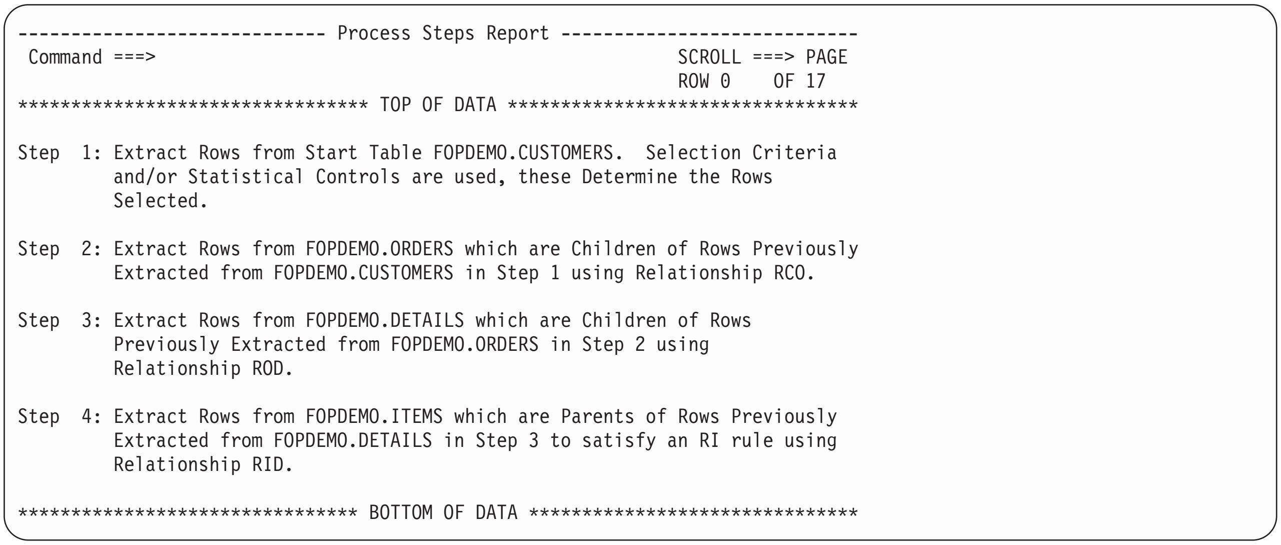
Step 4 The ITEMS related to the DETAILS are extracted by traversing the relationship RID from child to parent because **Q1** is Y.

You can select an option to follow a selected relationship from parent to child after a parent row has been extracted on the basis of **Q1**. In this example, rows are extracted from the ITEMS table because **Q1** is Y. The extracted ITEMS rows are CH1, CH2, CH3, and CH4. These ITEMS are related to unextracted DETAILS rows related to other ORDERS rows. You can direct Move to extract the additional DETAILS rows by specifying Y to **Q2** for the relationship RID. This directs Move to extract the additional children for any parent rows that have been extracted because the response to **Q1** is Y.

The specifications for the set of data, however, to select a sampling of CUSTOMERS rows with related ORDERS, DETAILS and ITEMS rows, do not require this. Therefore **Q2** is set to N. Since N is the default, you need not change anything on the panel.

SHOW STEPS Command

To request an analysis of the traversal path for the process, use the SHOW STEPS command on the Specify Relationship Usage panel. Move evaluates the traversal path on the basis of the Table List, the selected relationships, and **Q1** and **Q2** settings to document the series of steps that would be performed for the Extract Process. SHOW STEPS is a powerful way to check your specifications without actually extracting data. After reviewing the process steps, you can add tables to the Table List, select or unselect relationships, create relationships, or change **Q1** and **Q2** settings, as needed, to obtain the required set of relationally intact data.



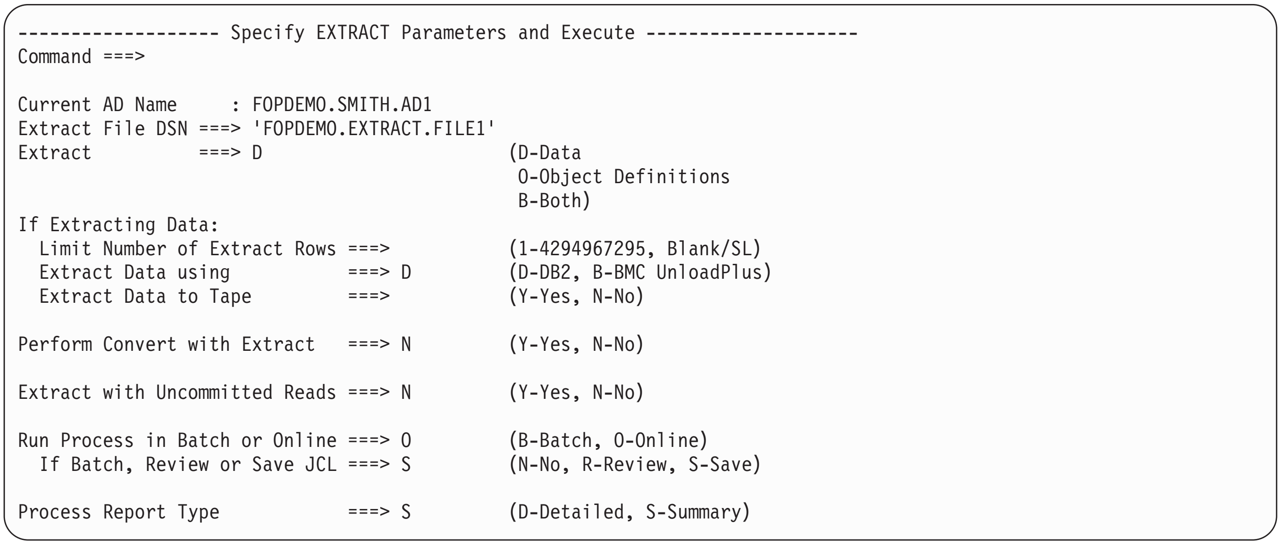
```
----------------------------- Process Steps Report ----------------------------
 Command ===>                                                 SCROLL ===> PAGE
                                                              ROW 0    OF 17
********************************* TOP OF DATA ********************************

Step  1: Extract Rows from Start Table FOPDEMO.CUSTOMERS.  Selection Criteria
         and/or Statistical Controls are used, these Determine the Rows
         Selected.

Step  2: Extract Rows from FOPDEMO.ORDERS which are Children of Rows Previously
         Extracted from FOPDEMO.CUSTOMERS in Step 1 using Relationship RCO.

Step  3: Extract Rows from FOPDEMO.DETAILS which are Children of Rows
         Previously Extracted from FOPDEMO.ORDERS in Step 2 using
         Relationship ROD.

Step  4: Extract Rows from FOPDEMO.ITEMS which are Parents of Rows Previously
         Extracted from FOPDEMO.DETAILS in Step 3 to satisfy an RI rule using
         Relationship RID.

******************************** BOTTOM OF DATA *******************************
```

Figure 13. Process Steps Report

When you are satisfied with the specifications for the Extract Process, use END until you return to the EXTRACT Process menu.

Perform the Extract

Select Option 4 PERFORM on the EXTRACT Process menu to display the Specify EXTRACT Parameters and Execute panel. This panel prompts for several parameters needed to execute the Extract Process.



```
------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE1'
Extract          ===> D                      (D-Data
                                              O-Object Definitions
                                              B-Both)
If Extracting Data:
  Limit Number of Extract Rows ===>          (1-4294967295, Blank/SL)
  Extract Data using           ===> D        (D-DB2, B-BMC UnloadPlus)
  Extract Data to Tape         ===>          (Y-Yes, N-No)

Perform Convert with Extract   ===> N        (Y-Yes, N-No)

Extract with Uncommitted Reads ===> N        (Y-Yes, N-No)

Run Process in Batch or Online ===> O        (B-Batch, O-Online)
  If Batch, Review or Save JCL ===> S        (N-No, R-Review, S-Save)

Process Report Type            ===> S        (D-Detailed, S-Summary)
```

Figure 14. Extract Process Parameters

Note that the name of the Access Definition is provided as **Current AD Name**. You must provide the name of the sequential file that is to contain the extracted data. Type the fully qualified file name in **Extract File DSN**. Use single quotes to delimit the name. Move automatically prefixes a name that is not delimited—a user option determines the prefix that is used.

Note: If the Extract File does not exist, Move prompts for information and allocates the file for you.

You can choose to extract the data, object definitions for the data, or both. The object definitions for the extracted tables and their columns are always extracted with the data so that Move can create the tables if they do not exist at the destination. However, the definitions for objects such as primary keys, indexes, views, and so forth must be requested explicitly. For this scenario, only data is extracted.

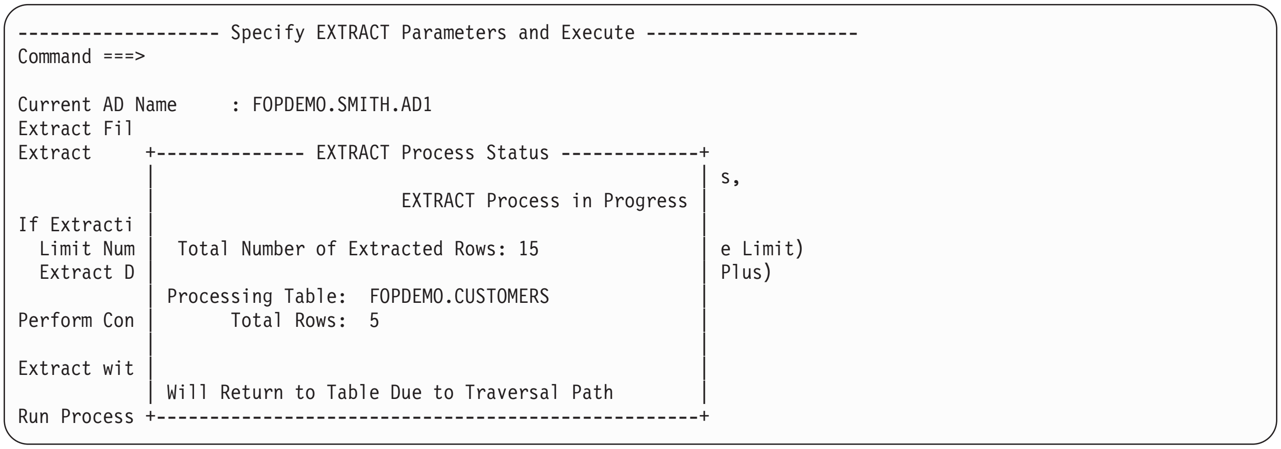
The **Perform Convert with Extract** option allows you to convert the data before storing it in the Extract File. Conversion is a useful way to protect sensitive data. These scenarios do not use the Convert option; however, you can apply the same functionality when inserting the data. For information about converting the data, see Convert Archive and Extract Files in the *Common Elements Manual*.

Other prompts on this panel allow you to limit the size of the Extract File, extract data using DB2 or an unload program, extract data that has not been committed to the database, execute the process in batch or online, create the extract file on tape, and, if in batch, review the JCL and utility control statements. After specifying a name for the Extract File, press ENTER to invoke the Extract Process online.

Move checks your specifications for the data to be extracted and, if any conditions require attention, displays warnings and error messages. Error conditions prevent the Extract Process from proceeding. Warnings do not prevent the Extract Process from executing, however you may want to modify your specifications. A warning is displayed, for example, when a table on the list will not be included in the extract because one or more relationships are unselected. If a warning prompt is displayed, press ENTER to proceed with the extract or END to return to the EXTRACT Process menu from which you can select the options to correct the condition.

Extract Status

Move provides status information during the process. When the Extract Process is executed online, a pop-up window displays execution status information that includes the name of the table being processed and the number of processed rows. This information is refreshed periodically as the extract proceeds.



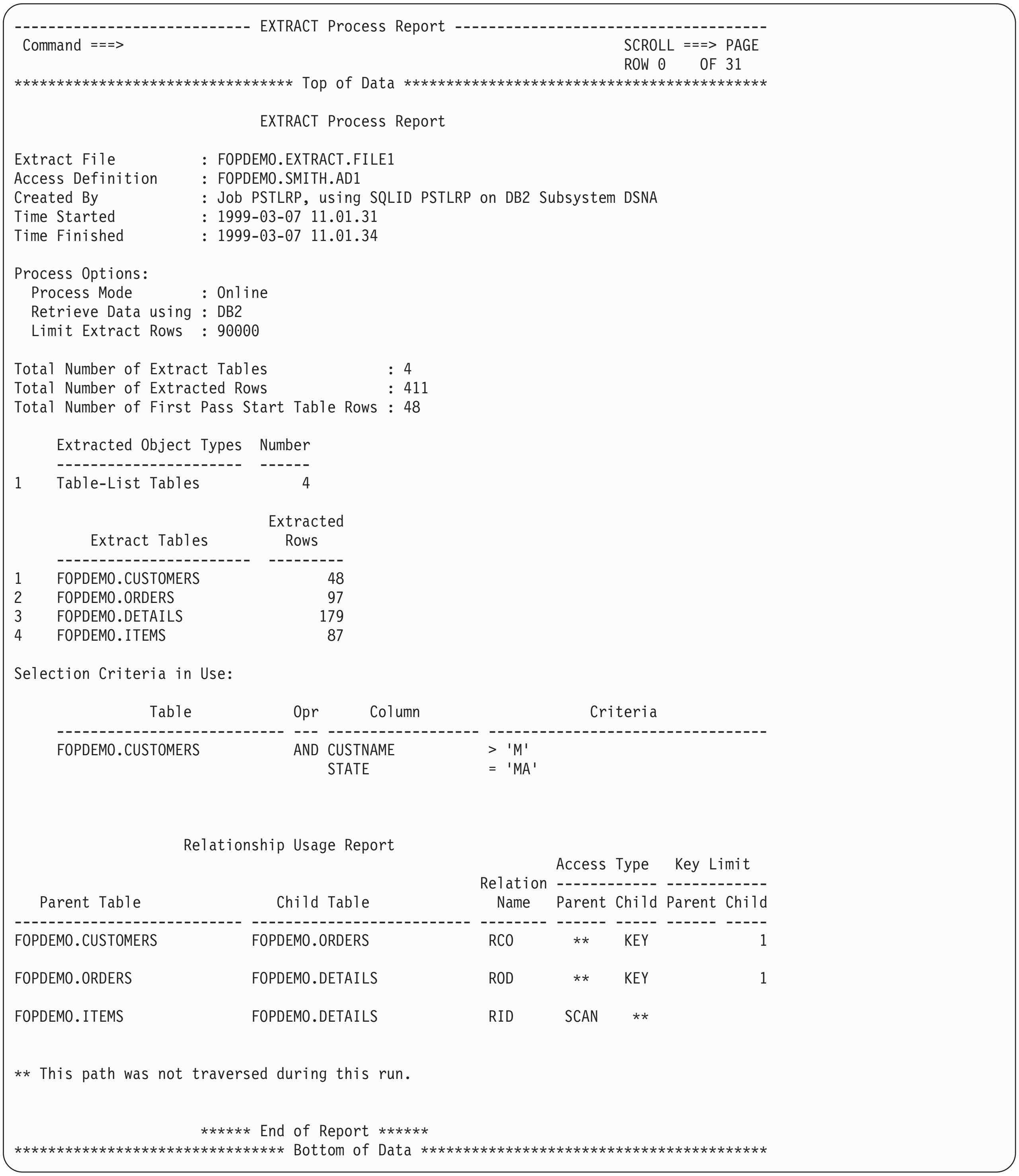
```
------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract Fil
Extract     +-------------- EXTRACT Process Status -------------+
            |                                                   | s,
            |                         EXTRACT Process in Progress |
If Extracti |                                                   |
  Limit Num |  Total Number of Extracted Rows: 15               | e Limit)
  Extract D |                                                   | Plus)
            | Processing Table:  FOPDEMO.CUSTOMERS              |
Perform Con |       Total Rows:  5                              |
            |                                                   |
Extract wit |                                                   |
            | Will Return to Table Due to Traversal Path        |
Run Process +---------------------------------------------------+
```

Figure 15. Extract Process Status

EXTRACT Process Report

Move displays an EXTRACT Process Report automatically when online execution is complete. (This report is available in SYSOUT or a designated file as specified in the JCL for batch execution.)

A sample of this report is shown in the following figure:

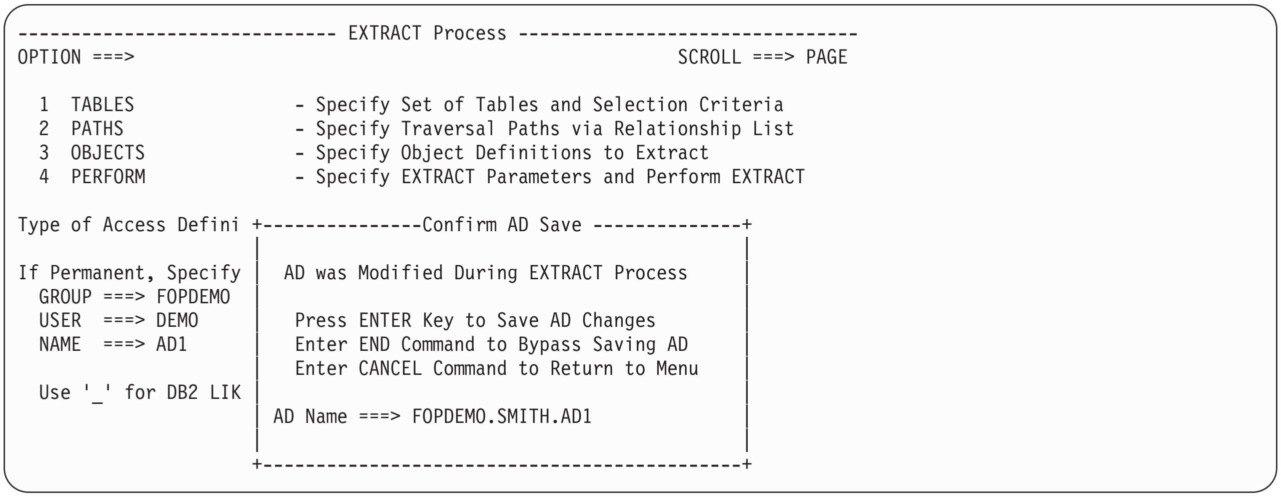


```
---------------------------- EXTRACT Process Report -------------------------------------
 Command ===>                                                           SCROLL ===> PAGE
                                                                        ROW 0    OF 31
********************************* Top of Data *******************************************

                             EXTRACT Process Report

Extract File          : FOPDEMO.EXTRACT.FILE1
Access Definition     : FOPDEMO.SMITH.AD1
Created By            : Job PSTLRP, using SQLID PSTLRP on DB2 Subsystem DSNA
Time Started          : 1999-03-07 11.01.31
Time Finished         : 1999-03-07 11.01.34

Process Options:
  Process Mode        : Online
  Retrieve Data using : DB2
  Limit Extract Rows  : 90000

Total Number of Extract Tables              : 4
Total Number of Extracted Rows              : 411
Total Number of First Pass Start Table Rows : 48

     Extracted Object Types  Number
     ----------------------  ------
1    Table-List Tables           4

                              Extracted
         Extract Tables         Rows
     -----------------------  ---------
1    FOPDEMO.CUSTOMERS               48
2    FOPDEMO.ORDERS                  97
3    FOPDEMO.DETAILS                179
4    FOPDEMO.ITEMS                   87

Selection Criteria in Use:

                Table           Opr     Column                   Criteria
     --------------------------- --- ------------------ ---------------------------------
     FOPDEMO.CUSTOMERS           AND CUSTNAME           > 'M'
                                     STATE              = 'MA'



                    Relationship Usage Report
                                                                Access Type   Key Limit
                                                       Relation ------------ ------------
   Parent Table                Child Table               Name   Parent Child Parent Child
-------------------------- -------------------------- -------- ------ ----- ------ -----
FOPDEMO.CUSTOMERS           FOPDEMO.ORDERS             RCO       **    KEY             1

FOPDEMO.ORDERS              FOPDEMO.DETAILS            ROD       **    KEY             1

FOPDEMO.ITEMS               FOPDEMO.DETAILS            RID      SCAN    **


** This path was not traversed during this run.


                      ****** End of Report ******
******************************** Bottom of Data *****************************************
```

Figure 16. EXTRACT Process Report Format

Use END to return to the EXTRACT Process menu. Use END again to return to the Data Migration menu.

Move prompts you to confirm that the Access Definition is to be saved. You can specify a new name or update the current Access Definition. In this sample scenario, you press ENTER to save the updated Access Definition and return to the Data Migration menu.



```
------------------------------ EXTRACT Process --------------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  TABLES              - Specify Set of Tables and Selection Criteria
  2  PATHS               - Specify Traversal Paths via Relationship List
  3  OBJECTS             - Specify Object Definitions to Extract
  4  PERFORM             - Specify EXTRACT Parameters and Perform EXTRACT

Type of Access Defini +----------------Confirm AD Save --------------+
                      |                                              |
If Permanent, Specify |  AD was Modified During EXTRACT Process      |
  GROUP ===> FOPDEMO  |                                              |
  USER  ===> DEMO     |   Press ENTER Key to Save AD Changes         |
  NAME  ===> AD1      |   Enter END Command to Bypass Saving AD      |
                      |   Enter CANCEL Command to Return to Menu     |
  Use '_' for DB2 LIK |                                              |
                      | AD Name ===> FOPDEMO.SMITH.AD1               |
                      |                                              |
                      +----------------------------------------------+
```

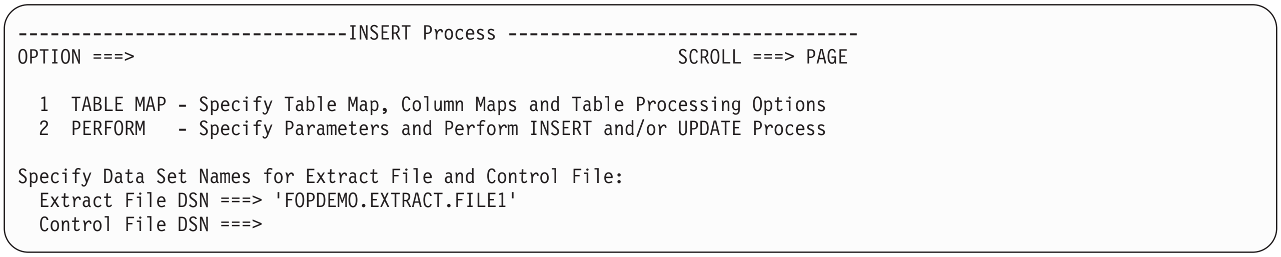
Figure 17. Confirm AD Save

Defining the Destination

After Move generates the Extract File containing the data for the test database, you can use the Insert Process to specify the destination.

INSERT Process Menu

Select Option 2 INSERT on the Data Migration menu to display the INSERT Process menu.



```
-------------------------------INSERT Process ---------------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  TABLE MAP - Specify Table Map, Column Maps and Table Processing Options
  2  PERFORM   - Specify Parameters and Perform INSERT and/or UPDATE Process

Specify Data Set Names for Extract File and Control File:
  Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE1'
  Control File DSN ===>
```

Figure 18. Scenario 1 – INSERT Process Menu

The options on this panel are:

1 TABLE MAP
Define the destination tables and table-level processing options.

2 PERFORM
Specify parameters and perform the Insert Process. Depending on the processing options you specify, the Insert Process inserts source data rows when the primary key value does not match that of a destination row, updates any destination rows when the primary key value matches that of a source row, or does both.

Two files are needed for an Insert Process. This panel prompts for the names of those files:

Extract File DSN
The name of the Extract File that contains the data to be inserted. **Extract File DSN** is automatically populated with the name from the most recent Extract Process.

Control File DSN
The name of a Control File for processing information.

Extract File for Source Data Only

The Extract File is used solely to provide the data and object definitions as a source for Move processing. It can be used by multiple users simultaneously and repeatedly.

Control File

Because the Extract File is not modified in an Insert Process, a Control File is needed to track processing. To distinguish the execution of multiple Move processes using the same Extract File, a Control File is generated for each Insert Process. The Control File includes information about the Insert Process parameters and the destination tables, indicates the success or failure of processing for each row in the Extract File, and shows whether the process completed.

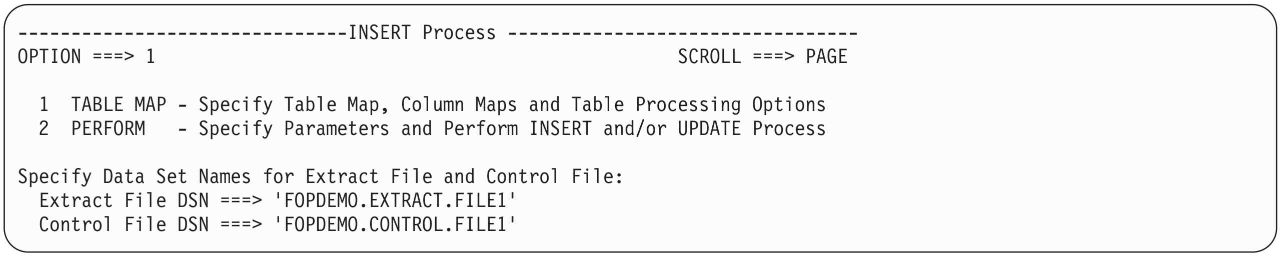
Specify the Tables

For this scenario, you create a test database in the image of the production system.

The source data, extracted earlier in the scenario, is stored in the Extract File, FOPDEMO.EXTRACT.FILE1. The new tables are identical to the tables from which the source data was extracted. To identify the new set of tables, a different Creator ID is assigned.

In order to create the test database, you must provide the names of the Extract File and the Control File on the INSERT Process menu. In this scenario, FOPDEMO.EXTRACT.FILE1 is the name of the Extract File and FOPDEMO.CONTROL.FILE1 is the name of the Control File. Use names that comply with naming conventions at your site.

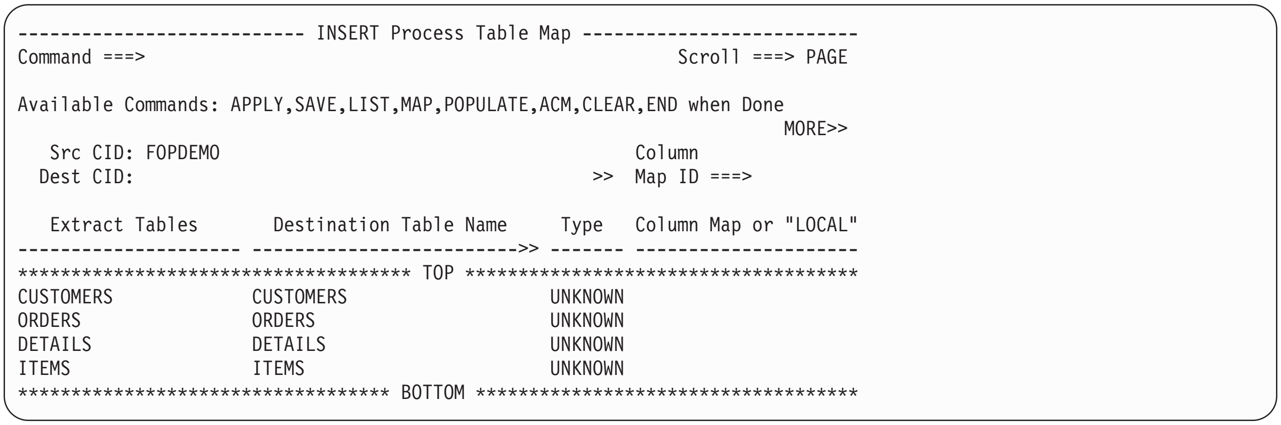
The Control File, like the Extract File, is a sequential file. If the named file does not exist, Move will prompt for necessary information before performing the Insert Process and allocate the file for you.



```
-------------------------------INSERT Process ---------------------------------
OPTION ===> 1                                                 SCROLL ===> PAGE

  1  TABLE MAP - Specify Table Map, Column Maps and Table Processing Options
  2  PERFORM   - Specify Parameters and Perform INSERT and/or UPDATE Process

Specify Data Set Names for Extract File and Control File:
  Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE1'
  Control File DSN ===> 'FOPDEMO.CONTROL.FILE1'
```

Figure 19. Option 1 on the INSERT Process Menu

Table Map parameters direct the insertion of source data into destination tables. Thus, you must have a Table Map to create the test database. Select Option 1 TABLE MAP from the INSERT Process menu to display the Table Map editor.



```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID:                                           >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
--------------------- ------------------------>> ------- ---------------------
************************************* TOP *************************************
CUSTOMERS             CUSTOMERS                   UNKNOWN
ORDERS                ORDERS                      UNKNOWN
DETAILS               DETAILS                     UNKNOWN
ITEMS                 ITEMS                       UNKNOWN
*********************************** BOTTOM ************************************
```

Figure 20. Initial Display of Table Map

Table Map Panel

Move automatically inserts the Source Creator ID (**Src CID**) and the names of all Extract File tables (**Extract Tables**) in the Table Map Editor and inserts these same names in **Destination Table Name**. You must supply the Destination Creator ID (**Dest CID**) and can edit the inserted names of destination tables.

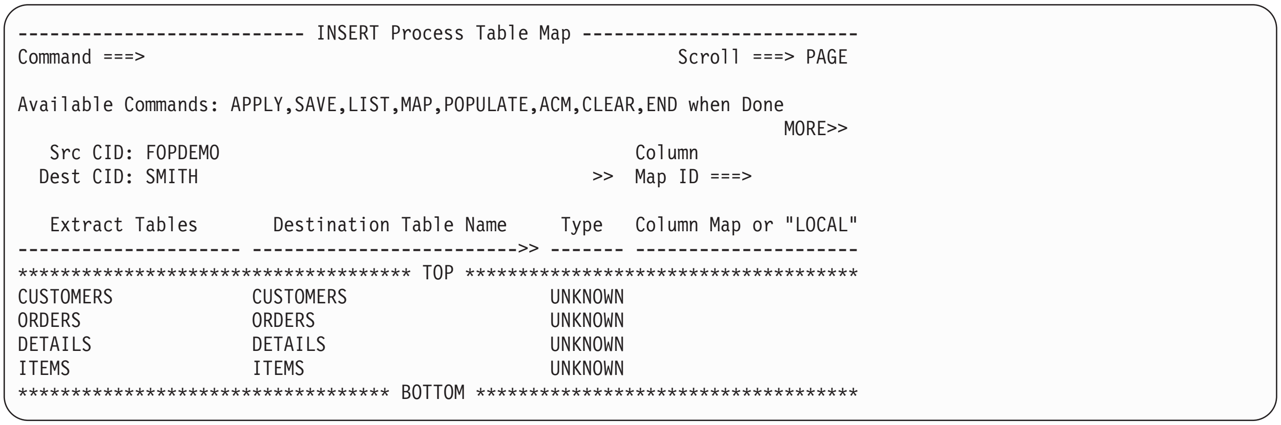
Type is supplied by Move and indicates whether the destination is a DB2 or Legacy Table, a view, or unknown. (UNKNOWN indicates that the table does not exist at the destination or that there is not enough information to identify the table; for example, if **Dest CID** is not supplied.) If destination tables do not exist when the Insert Process is executed, Move creates them. (Note that you must have authorization to create tables.) The structure of the created tables duplicates that of the source tables.

You can specify the **Dest CID** and modify the **Destination Table Names** to specify the names you want. Move adjusts the **Type** accordingly.

Changing Dest CID

For this scenario, you will create a test database containing tables identical to those in the production database. Only the Creator ID distinguishes the two sets of tables. (The test database may be on the same or a different DB2 subsystem.) To insert the extracted data into test tables with the same names, enter your Creator ID as the **Dest CID**.

Move automatically identifies the **Type** for each destination table indicated by name and the new **Dest CID**. In this example, when the **Dest CID** is changed to SMITH, the **Type** for each destination table remains unknown. That is, SMITH.CUSTOMERS, SMITH.ORDERS, SMITH.DETAILS, and SMITH.ITEMS do not exist.



```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID: SMITH                                     >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
-------------------- ------------------------->> ------- ---------------------
************************************* TOP ************************************
CUSTOMERS             CUSTOMERS                   UNKNOWN
ORDERS                ORDERS                      UNKNOWN
DETAILS               DETAILS                     UNKNOWN
ITEMS                 ITEMS                       UNKNOWN
*********************************** BOTTOM ***********************************
```

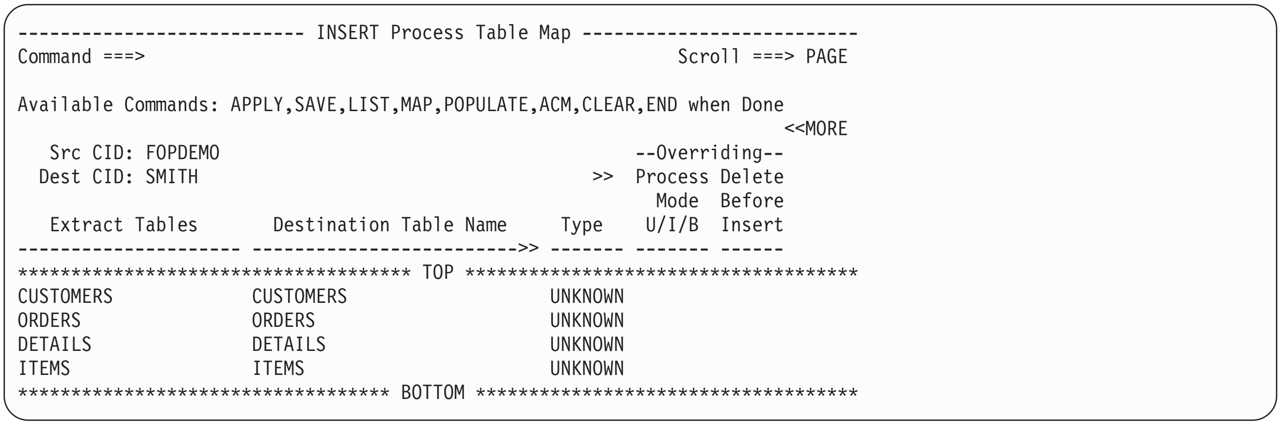
Figure 21. Dest CID Specified on Table Map

You can choose a different destination table by overtyping the listed name. You can also explicitly prefix any destination table name to override the **Dest CID** default.

Processing Overrides

The INSERT Process Table Map is presented on facing pages.**MORE**, preceded or followed by two arrows, indicates the presence of a facing page. Use the primary commands LEFT and RIGHT or the assigned function keys to scroll the page horizontally. (Note that you must enter the **Dest CID** before you can scroll the page horizontally.)

To provide processing overrides for selected tables, you can scroll to **Overriding** on the INSERT Process Table Map panel.



```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                         <<MORE
   Src CID: FOPDEMO                                       --Overriding--
  Dest CID: SMITH                                     >>  Process Delete
                                                            Mode  Before
   Extract Tables       Destination Table Name     Type    U/I/B  Insert
-------------------- ------------------------->> ------- ------- ------
************************************* TOP ************************************
CUSTOMERS             CUSTOMERS                   UNKNOWN
ORDERS                ORDERS                      UNKNOWN
DETAILS               DETAILS                     UNKNOWN
ITEMS                 ITEMS                       UNKNOWN
*********************************** BOTTOM ***********************************
```

Figure 22. Table Map – Processing Overrides

Processing overrides are displayed on the panel. These prompts allow you to set the processing options for any table on an individual basis. The default processing options from the Specify INSERT Parameters and Execute panel are used for tables with no overriding parameters. (For a description of the default processing options, see “Process Options” on page 29.) The following prompts are displayed on the panel and are blank by default.

Process Mode
The overriding processing mode for selected tables. Specify one of the following:

U Update only.

I Insert only.

B Both update and insert.

Delete Before Insert

Specify whether to delete all rows in the destination table prior to inserting data from the Extract File. Specify one of the following:

Y Yes, delete all rows prior to the Insert. (**Process Mode** must be I.).

N No, do not delete all rows prior to the Insert.

For Scenario 1, leave the overriding parameters blank for all tables. Use END to return to the INSERT Process menu.

Storing Table Maps

You can store and reuse a Table Map. For example, the Table Map defined for this Insert Process can be saved, using the SAVE *mapname* primary command. The saved Table Map is then available for any process and to any users.

You can use the APPLY command to insert specifications from a saved Table Map into the current Table Map. Operands allow you to indicate the specifications that are inserted–the **Dest CID** and the **Destination Table Name**, or other values from a stored Table Map. Destination table names are inserted when the **Extract Tables** in the Table Maps match.

In addition to defining a Table Map as part of a process, such as the Insert Process described here, you can define a Table Map using Option 6 DEFINITIONS on the Main Menu. For more information, see Table Maps in the *Common Elements Manual*.

Perform the Insert

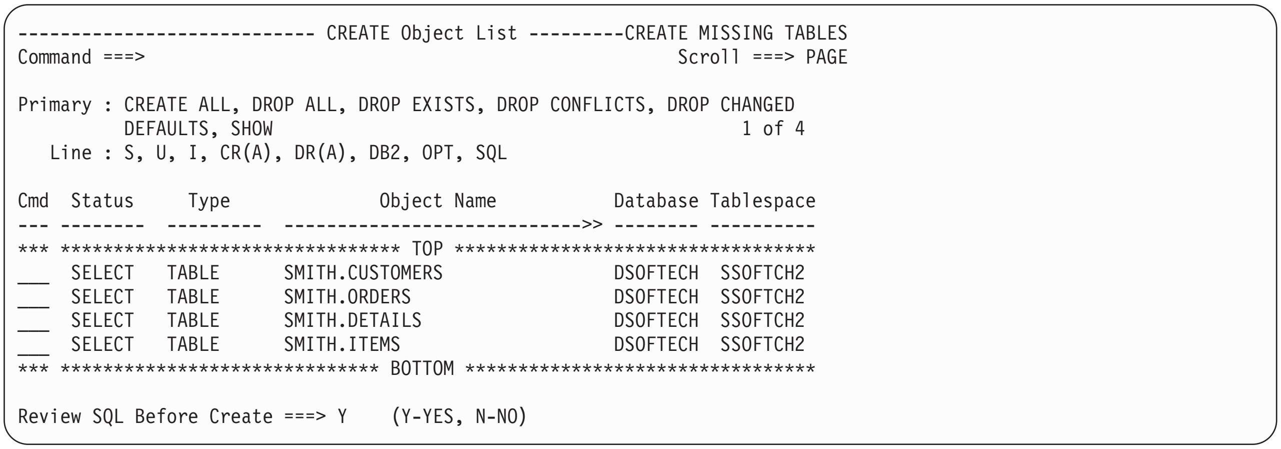
Select Option 2 PERFORM on the INSERT Process menu to perform the Insert. The Insert Process can be performed in several ways. You can specify insert only, update only, or both.

If you specify insert only, only new rows are inserted. Thus, a source row is bypassed or discarded when the primary key value matches the value in an existing destination row. If you specify update only, existing destination rows are updated. Therefore, when the primary key value of the source row matches the primary key value in an existing destination row, the destination row is updated. If you specify both, new rows are inserted and existing destination rows are updated.

Prompt to Create Tables

Move first checks for any UNKNOWN destination tables and prompts for information needed to create those tables. For this scenario, the Extract File contains no objects other than the definitions for tables (which are always extracted). Therefore, only tables are listed in the prompt. However, if additional object definitions were extracted, they would also be listed.

The following figure shows the CREATE Object List. The names of the tables are listed in the order specified in the Access Definition.



```
---------------------------- CREATE Object List ---------CREATE MISSING TABLES
Command ===>                                                  Scroll ===> PAGE

Primary : CREATE ALL, DROP ALL, DROP EXISTS, DROP CONFLICTS, DROP CHANGED
          DEFAULTS, SHOW                                            1 of 4
   Line : S, U, I, CR(A), DR(A), DB2, OPT, SQL

Cmd  Status     Type             Object Name          Database Tablespace
--- --------  ---------  ---------------------------->> -------- ----------
*** ******************************** TOP *********************************
___  SELECT   TABLE      SMITH.CUSTOMERS               DSOFTECH  SSOFTCH2
___  SELECT   TABLE      SMITH.ORDERS                  DSOFTECH  SSOFTCH2
___  SELECT   TABLE      SMITH.DETAILS                 DSOFTECH  SSOFTCH2
___  SELECT   TABLE      SMITH.ITEMS                   DSOFTECH  SSOFTCH2
*** ****************************** BOTTOM ********************************

Review SQL Before Create ===> Y    (Y-YES, N-NO)
```

Figure 23. Prompt to Create Destination Table

You must provide a Database name for each table. Tablespace names are optional. If default values have been established, they are displayed in **Database** and **Tablespace**. Type the names to be used at your site.

The status of all listed tables in this scenario is SELECT. This status indicates that the tables do not exist at the destination and are selected to be created. The status for tables that exist at the destination is EXISTS. (Several other statuses are possible and are documented in the *Move User Manual*.)

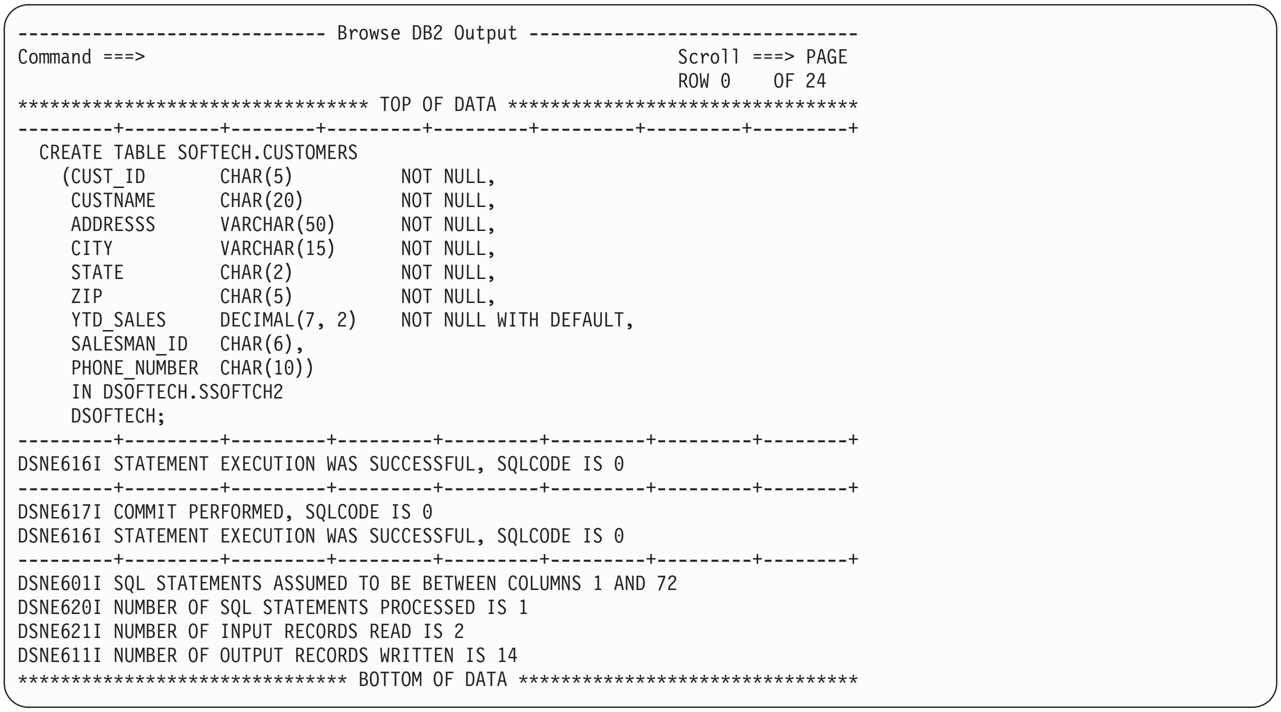
A prompt on the panel allows you to display the SQL used to create the tables prior to execution. If you specify YES, the generated SQL is displayed in the ISPF editor, where you can modify and save it.

CREATE ALL Command

Use the CREATE ALL command to generate and execute the SQL DDL statements needed to create tables with SELECT status. You may optionally use the CR line command to create the tables individually.

Use RUN to execute the SQL and return to the Move session. Use END or CANCEL to abort execution and return to the Move session. For this scenario, do not modify the SQL; use RUN to execute.

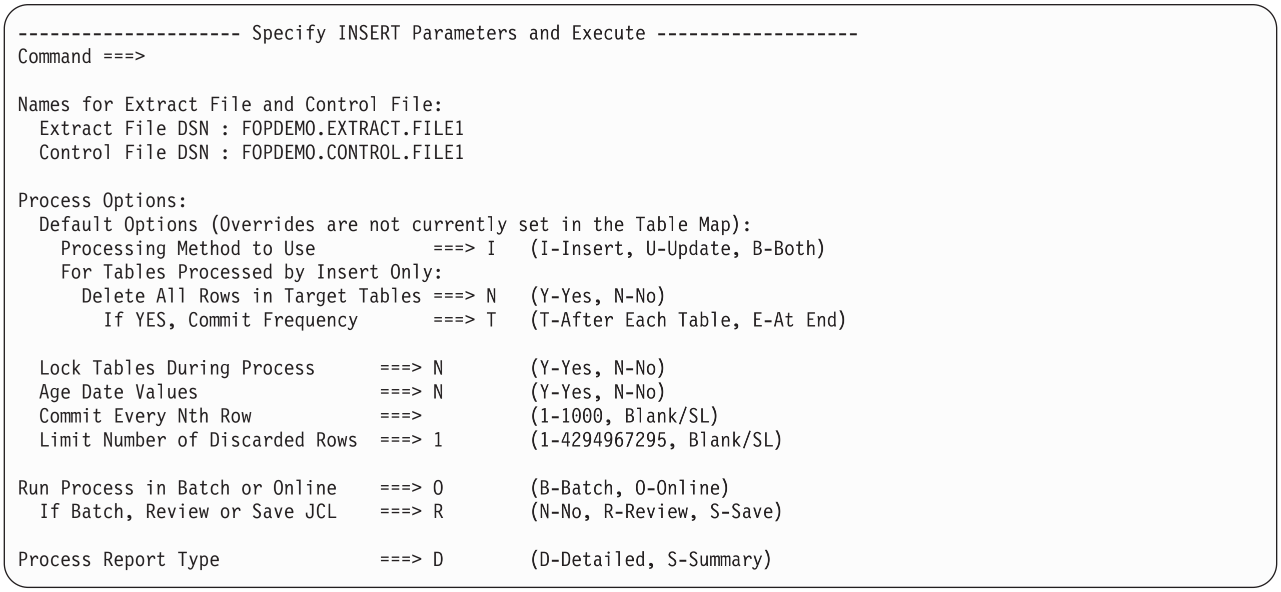
Move displays, in read-only form, the DB2 information generated for the create request. The following figure shows the output for the CUSTOMERS table as an example:



```
----------------------------- Browse DB2 Output -------------------------------
Command ===>                                                  Scroll ===> PAGE
                                                              ROW 0    OF 24
******************************** TOP OF DATA **********************************
---------+---------+---------+---------+---------+---------+---------+---------+
  CREATE TABLE SOFTECH.CUSTOMERS
    (CUST_ID       CHAR(5)          NOT NULL,
     CUSTNAME      CHAR(20)         NOT NULL,
     ADDRESSS      VARCHAR(50)      NOT NULL,
     CITY          VARCHAR(15)      NOT NULL,
     STATE         CHAR(2)          NOT NULL,
     ZIP           CHAR(5)          NOT NULL,
     YTD_SALES     DECIMAL(7, 2)    NOT NULL WITH DEFAULT,
     SALESMAN_ID   CHAR(6),
     PHONE_NUMBER  CHAR(10))
     IN DSOFTECH.SSOFTCH2
     DSOFTECH;
---------+---------+---------+---------+---------+---------+---------+--------+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+--------+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+--------+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
DSNE621I NUMBER OF INPUT RECORDS READ IS 2
DSNE611I NUMBER OF OUTPUT RECORDS WRITTEN IS 14
****************************** BOTTOM OF DATA *********************************
```

Figure 24. DB2 Output from CREATE

Use END to return to the CREATE Object List panel. Use END a second time to continue the Insert Process. The following panel is displayed.



```
--------------------- Specify INSERT Parameters and Execute -------------------
Command ===>

Names for Extract File and Control File:
  Extract File DSN : FOPDEMO.EXTRACT.FILE1
  Control File DSN : FOPDEMO.CONTROL.FILE1

Process Options:
  Default Options (Overrides are not currently set in the Table Map):
    Processing Method to Use           ===> I   (I-Insert, U-Update, B-Both)
    For Tables Processed by Insert Only:
      Delete All Rows in Target Tables ===> N   (Y-Yes, N-No)
        If YES, Commit Frequency       ===> T   (T-After Each Table, E-At End)

  Lock Tables During Process      ===> N        (Y-Yes, N-No)
  Age Date Values                 ===> N        (Y-Yes, N-No)
  Commit Every Nth Row            ===>          (1-1000, Blank/SL)
  Limit Number of Discarded Rows  ===> 1        (1-4294967295, Blank/SL)

Run Process in Batch or Online    ===> O        (B-Batch, O-Online)
  If Batch, Review or Save JCL    ===> R        (N-No, R-Review, S-Save)

Process Report Type               ===> D        (D-Detailed, S-Summary)
```

Figure 25. Specify INSERT Parameters and Execute

Move supplies the Extract File and Control File data set names; they cannot be modified.

Several parameters allow you to control the Insert Process. For example, you can age dates using the **Age Date Values** option. You can also specify whether the process is to be performed in batch or online.

Process Options

Process Options allow you to specify default values for tables that do not have overrides set in the Table Map Editor. A parenthetical note on the **Default Options** line indicates whether overriding values have been established. To set the default **Processing Method to Use**, specify whether to perform the Insert Process as insert only (I), update only (U), or both insert and update (B). If you specify I, you can choose to delete all rows in destination tables prior to inserting data from the Extract File and specify how often to issue commits during processing.

Note: If site management does not allow user specification, the **Delete All Rows in Target Tables** and **Commit Frequency** lines may be omitted from this panel.

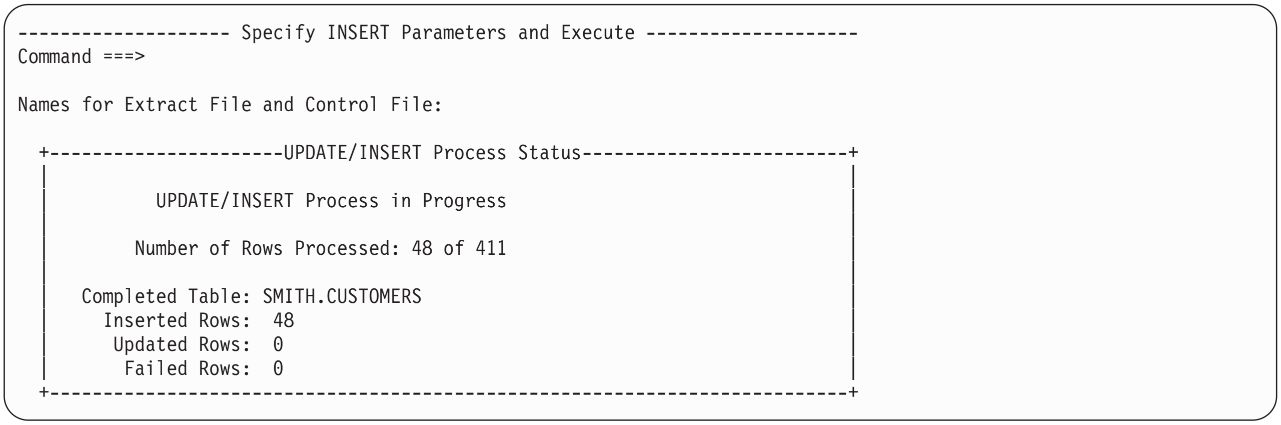
In this scenario, specify I for **Processing Method to Use** and N for **Delete All Rows in Target Tables**.

Limit Discarded Rows

You can cause an Insert Process to terminate if the number of discarded rows reaches a specified limit. For example, discarding more than 50 rows may indicate a problem with RI rules that prevents the insertion of rows. Rather than allow a lengthy and unproductive process to continue, you can cause it to terminate. You can then browse the discarded rows, determine the problem, correct it and restart the Insert Process.

In this scenario, enter “1” at the **Limit Number of Discarded Rows** prompt to terminate the process if a row is discarded. Since the destination tables duplicate the source tables, all source data should be inserted successfully.

When specifications are complete, press ENTER to perform the Insert Process. The status of an online Insert Process is displayed as it executes.



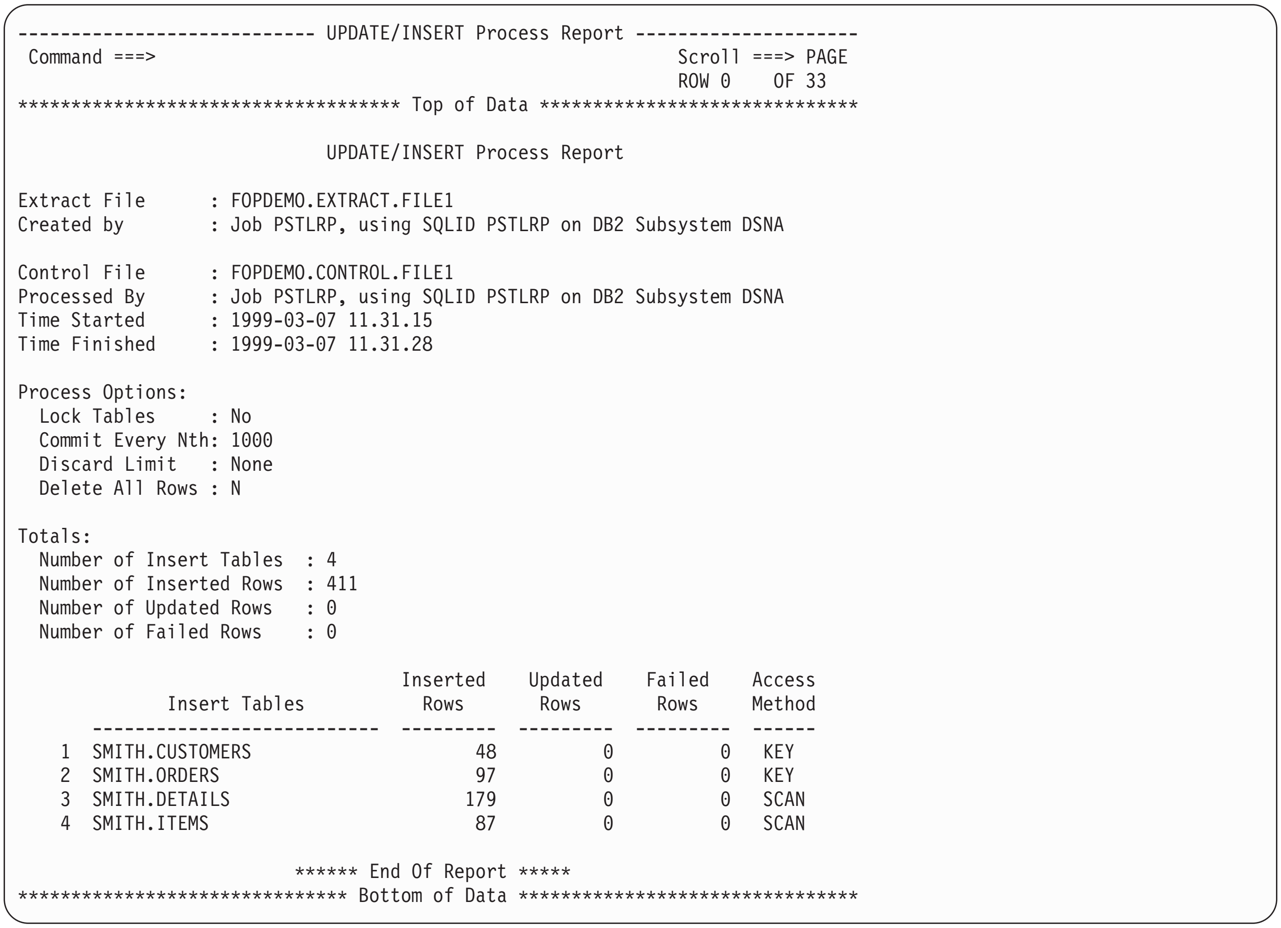
```
-------------------- Specify INSERT Parameters and Execute --------------------
Command ===>

Names for Extract File and Control File:

  +----------------------UPDATE/INSERT Process Status-------------------------+
  |                                                                           |
  |          UPDATE/INSERT Process in Progress                                |
  |                                                                           |
  |        Number of Rows Processed: 48 of 411                                |
  |                                                                           |
  |   Completed Table: SMITH.CUSTOMERS                                        |
  |     Inserted Rows:  48                                                    |
  |      Updated Rows:  0                                                     |
  |       Failed Rows:  0                                                     |
  +---------------------------------------------------------------------------+
```

Figure 26. UPDATE/INSERT Process Status

During processing, the total number of rows processed and the total number to be processed are displayed. Also, the name of the currently processing table and the total number of rows that have been inserted, updated, or failed for that table are displayed. This display is updated periodically as the Insert Process is executed.

Process Report

An INSERT Process Report, generated at the conclusion of the process, contains information about the process. When the process is executed online, the INSERT Process Report is displayed automatically.



```
---------------------------- UPDATE/INSERT Process Report ---------------------
 Command ===>                                                  Scroll ===> PAGE
                                                               ROW 0    OF 33
************************************ Top of Data *****************************

                             UPDATE/INSERT Process Report

Extract File      : FOPDEMO.EXTRACT.FILE1
Created by        : Job PSTLRP, using SQLID PSTLRP on DB2 Subsystem DSNA

Control File      : FOPDEMO.CONTROL.FILE1
Processed By      : Job PSTLRP, using SQLID PSTLRP on DB2 Subsystem DSNA
Time Started      : 1999-03-07 11.31.15
Time Finished     : 1999-03-07 11.31.28

Process Options:
  Lock Tables     : No
  Commit Every Nth: 1000
  Discard Limit   : None
  Delete All Rows : N

Totals:
  Number of Insert Tables  : 4
  Number of Inserted Rows  : 411
  Number of Updated Rows   : 0
  Number of Failed Rows    : 0

                                    Inserted    Updated    Failed    Access
              Insert Tables           Rows       Rows       Rows     Method
       ---------------------------  ---------  ---------  ---------  ------
    1  SMITH.CUSTOMERS                     48          0          0   KEY
    2  SMITH.ORDERS                        97          0          0   KEY
    3  SMITH.DETAILS                      179          0          0   SCAN
    4  SMITH.ITEMS                         87          0          0   SCAN

                          ****** End Of Report *****
****************************** Bottom of Data ********************************
```

Figure 27. INSERT Process Report Format

Report Contents

The report is formatted with headings to identify the information. This information includes the Extract File name, the Control File name, the user initiating the Insert Process, the date and time the process was executed, various process options, the number of tables processed, the number of successfully inserted or updated rows, and the number of failed or discarded rows. Standard ISPF scrolling functions are available.

The tables are listed in the order in which they were stored in the Extract File. The number of rows processed successfully and unsuccessfully is shown for each table.

Print Report

While browsing the INSERT Process Report online, you can use the OUTPUT command to direct the contents of the report to an output file or SYSOUT. A panel prompts for information according to the specified output destination.

Summary

This section discussed how to specify the set of relational data to be extracted and stored in an Extract File. An Extract File is a sequential data set that contains a copy of the specified data. In order to move the data to destination tables, this file is used as input for the Insert Process.

This section also discussed how to insert the source data from the Extract File into a set of newly created tables. Move automatically created the destination tables as exact copies of the original source tables and inserted the source data as specified.

Chapter 4. Scenario 2 – A Set of Items

Scenario 2 uses the same set of tables used in Scenario 1, but this scenario is more complex. You will extract a set of data that includes DETAILS rows related to ORDERS for specific ITEMS.

To obtain the desired data, the ITEMS table is the Start Table and you will select specific ITEMS rows for the Extract Process. To ensure that extracted data is referentially intact, it must also include the CUSTOMERS that placed the ORDERS, and all DETAILS rows for any ORDER for a selected ITEMS row.

To demonstrate a set of functions different from those in the first scenario, the processing method for the Insert Process is both update and insert. Any rows that may have been modified in the time between the original insert and this process, are to be updated with the extracted data. Also, only the destination tables were created in the first scenario. For Scenario 2, the primary keys, relationships, and indexes will be created.

Defining the Source

Just a few steps are performed to obtain the desired set of data. You will:

- Specify the Start Table as ITEMS.
- Select individual rows from the ITEMS table.
- Choose the relationships that are traversed.
- Select the object definitions that are extracted.

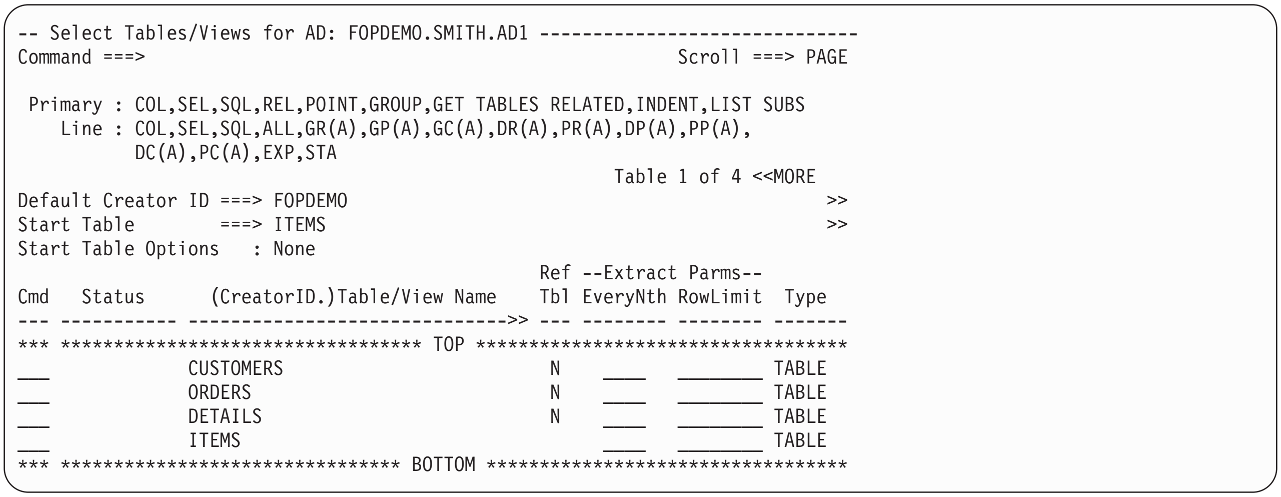
Defining the Destination

The destination tables are the same as those used in Scenario 1. However, a Column Map is used to insert a literal value into one column and the values of DB2 special registers into other columns.

Defining the Source

The Access Definition created for the first scenario must be edited for this scenario. Specify the Access Definition, FOPDEMO.SMITH.AD1, and select Option 1 TABLES on the EXTRACT Process menu to display the Select Tables/Views for AD panel.

Change the **Start Table** name on the Select Tables/Views for AD panel to ITEMS and clear **EveryNth** and **RowLimit** entries for the CUSTOMERS table.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> ITEMS                                               >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************** TOP ***********************************
___             CUSTOMERS                         N    ____   ________ TABLE
___             ORDERS                            N    ____   ________ TABLE
___             DETAILS                           N    ____   ________ TABLE
___             ITEMS                                  ____   ________ TABLE
*** ******************************** BOTTOM **********************************
```

Figure 28. ITEMS as Start Table

Row Selection using Point-and-Shoot

You can select rows from the Start Table individually, while browsing the data. The primary key values for the rows you select are copied to a “row list” or Point-and-Shoot File and used to select database rows during the Extract Process.

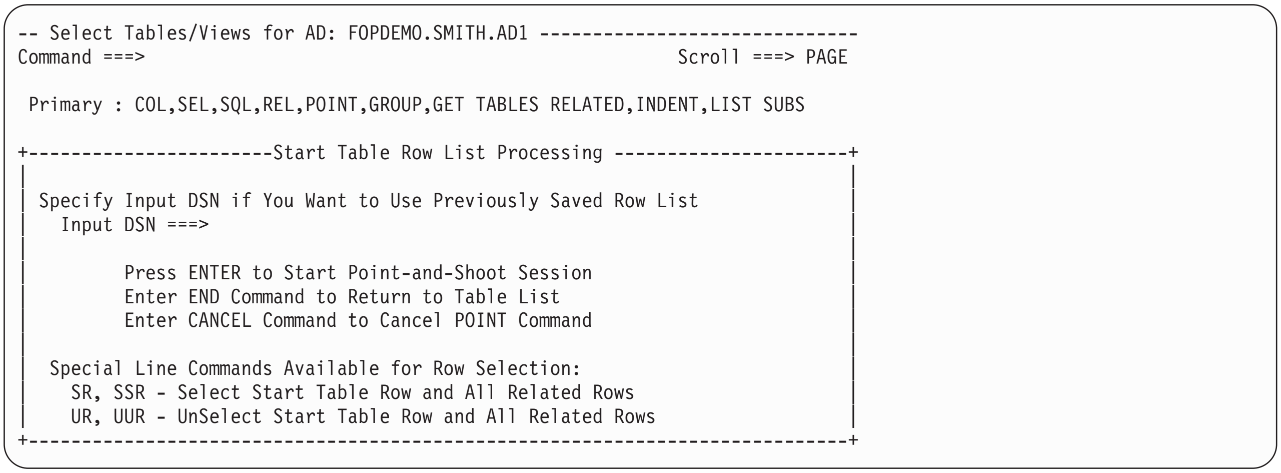
For this scenario, you select rows from ITEMS, the Start Table, using the Point-and-Shoot facility. The Extract Process will extract the selected ITEMS rows, the DETAILS rows related to those ITEMS, the ORDERS rows related to the DETAILS, and so on.

POINT Command

Use the POINT command to invoke the Point-and-Shoot facility. (You can specify selection criteria, as described in “Specifying the Criteria” on page 15 to limit the Start Table rows that are retrieved and displayed for Point-and-Shoot. However, the ITEMS table in the sample database is very small and all rows are displayed for this scenario.)

Input DSN

Before displaying the data, Move prompts for an **Input DSN**. The input data set contains primary key values for any rows selected previously.



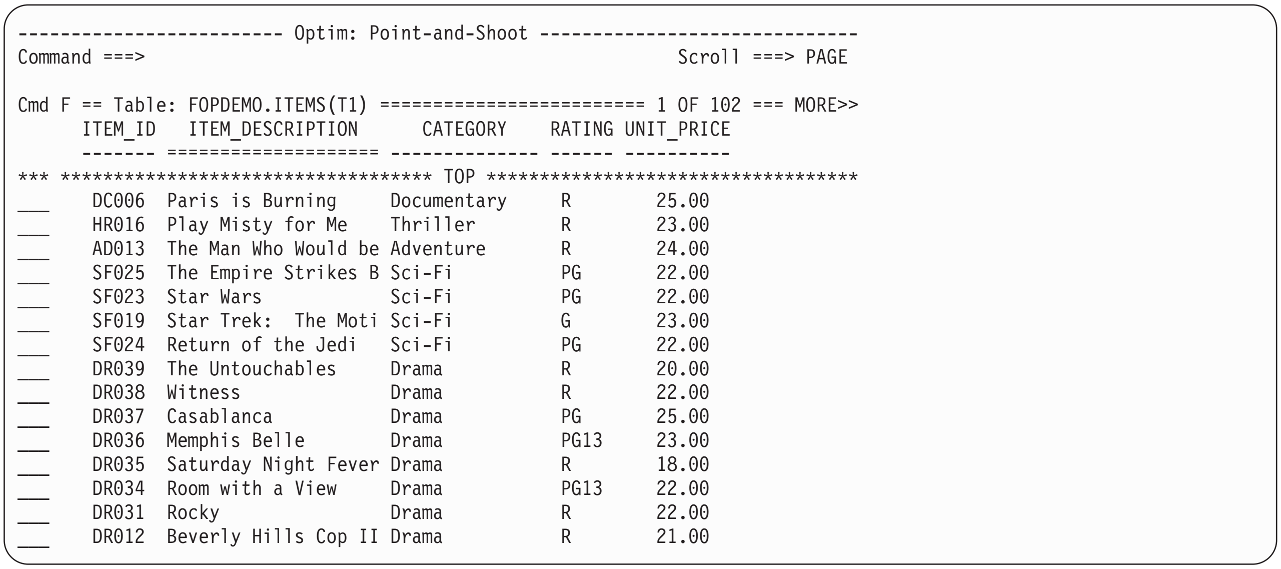
```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS

+-----------------------Start Table Row List Processing ----------------------+
|                                                                             |
|  Specify Input DSN if You Want to Use Previously Saved Row List             |
|    Input DSN ===>                                                           |
|                                                                             |
|          Press ENTER to Start Point-and-Shoot Session                       |
|          Enter END Command to Return to Table List                          |
|          Enter CANCEL Command to Cancel POINT Command                        |
|                                                                             |
|   Special Line Commands Available for Row Selection:                        |
|     SR, SSR - Select Start Table Row and All Related Rows                   |
|     UR, UUR - UnSelect Start Table Row and All Related Rows                 |
+-----------------------------------------------------------------------------+
```

Figure 29. Start Table Row Selection

Since a previously created Point-and-Shoot data set is not used for this scenario, leave **Input DSN** blank. Press ENTER to begin the Point-and-Shoot session.

Point-and-Shoot Display

Data from all rows in the Start Table are displayed for the Point-and-Shoot session.



```
------------------------ Optim: Point-and-Shoot ------------------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 1 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** ********************************** TOP ***********************************
___     DC006  Paris is Burning     Documentary     R        25.00
___     HR016  Play Misty for Me    Thriller        R        23.00
___     AD013  The Man Who Would be Adventure       R        24.00
___     SF025  The Empire Strikes B Sci-Fi          PG       22.00
___     SF023  Star Wars            Sci-Fi          PG       22.00
___     SF019  Star Trek:  The Moti Sci-Fi          G        23.00
___     SF024  Return of the Jedi   Sci-Fi          PG       22.00
___     DR039  The Untouchables     Drama           R        20.00
___     DR038  Witness              Drama           R        22.00
___     DR037  Casablanca           Drama           PG       25.00
___     DR036  Memphis Belle        Drama           PG13     23.00
___     DR035  Saturday Night Fever Drama           R        18.00
___     DR034  Room with a View     Drama           PG13     22.00
___     DR031  Rocky                Drama           R        22.00
___     DR012  Beverly Hills Cop II Drama           R        21.00
```

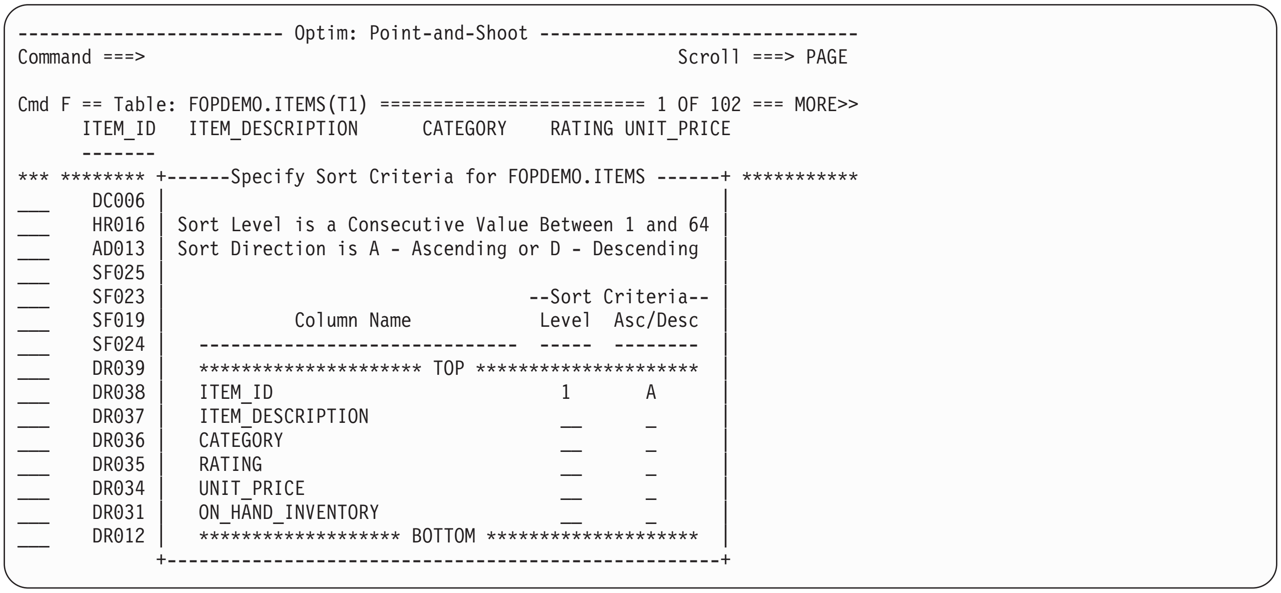
Figure 30. Point-and-Shoot Display

The panel includes Table Information, Column Headings, and Table Data.

- Table Information: The name of the table with a Move-generated identifier. In Figure 30, the ITEMS table is identified as T1. A count of the number of lines of data is provided and the position of the first displayed line in relation to all lines is noted. In the figure, 1 of 102 indicates that the first line of data from the ITEMS table is displayed at the beginning of the list and a total of 102 rows are available for display.
- Column Headings: Headings for each column are displayed.
- Table Data: Rows from the table are listed with the headings. Markers indicate the first and last rows of data.

Specifying Sort Criteria

The rows are retrieved in an arbitrary order. You can use the SORT command to display the **Specify Sort Criteria** prompt, from which you can specify sort criteria to arrange the data.



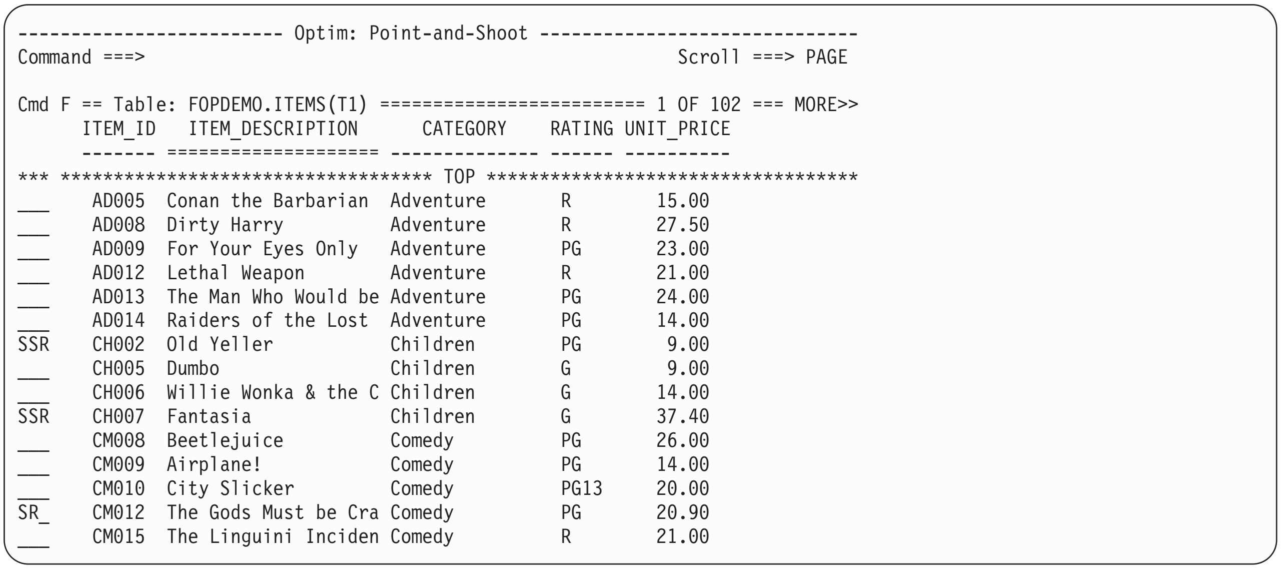
```
------------------------- Optim: Point-and-Shoot ------------------------------
Command ===>                                                   Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 1 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      -------
*** ******** +------Specify Sort Criteria for FOPDEMO.ITEMS ------+ ***********
___    DC006 |                                                     |
___    HR016 | Sort Level is a Consecutive Value Between 1 and 64  |
___    AD013 | Sort Direction is A - Ascending or D - Descending   |
___    SF025 |                                                     |
___    SF023 |                                  --Sort Criteria--  |
___    SF019 |            Column Name            Level  Asc/Desc   |
___    SF024 |   ------------------------------  -----  --------   |
___    DR039 |   ********************* TOP *********************   |
___    DR038 |   ITEM_ID                           1       A       |
___    DR037 |   ITEM_DESCRIPTION                  __      _       |
___    DR036 |   CATEGORY                          __      _       |
___    DR035 |   RATING                            __      _       |
___    DR034 |   UNIT_PRICE                        __      _       |
___    DR031 |   ON_HAND_INVENTORY                 __      _       |
___    DR012 |   ******************* BOTTOM ********************   |
             +-----------------------------------------------------+
```

Figure 31. Specify Sort Criteria

To arrange the data according to ascending values in ITEM_ID, enter 1 as the **Level** and A as the direction for ITEM_ID, as shown in the figure, and press ENTER or use END to return to the data display.

Selecting Rows

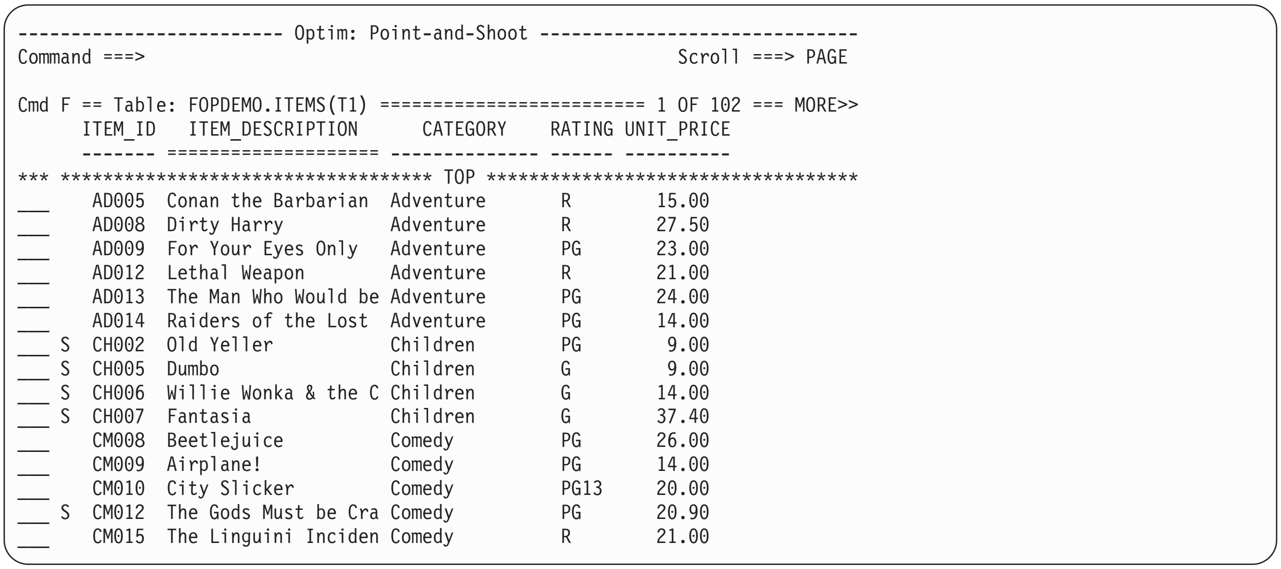
Use the Select Related line command to make your row selections. You can use the SR single line form or the SSR block form. In the following figure, all ITEMS rows with the CATEGORY “Children” are selected using the block form, SSR. A single ITEMS row, ITEM_ID CM012, is selected using the single line form SR.



```
------------------------ Optim: Point-and-Shoot ------------------------------
Command ===>                                                 Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ======================== 1 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** ********************************** TOP **********************************
___    AD005  Conan the Barbarian  Adventure      R        15.00
___    AD008  Dirty Harry          Adventure      R        27.50
___    AD009  For Your Eyes Only   Adventure      PG       23.00
___    AD012  Lethal Weapon        Adventure      R        21.00
___    AD013  The Man Who Would be Adventure      PG       24.00
___    AD014  Raiders of the Lost  Adventure      PG       14.00
SSR    CH002  Old Yeller           Children       PG        9.00
___    CH005  Dumbo                Children       G         9.00
___    CH006  Willie Wonka & the C Children       G        14.00
SSR    CH007  Fantasia             Children       G        37.40
___    CM008  Beetlejuice          Comedy         PG       26.00
___    CM009  Airplane!            Comedy         PG       14.00
___    CM010  City Slicker         Comedy         PG13     20.00
SR_    CM012  The Gods Must be Cra Comedy         PG       20.90
___    CM015  The Linguini Inciden Comedy         R        21.00
```

Figure 32. Selecting Rows on Point-and-Shoot Display

Press ENTER to process your selections. An S is placed under the **F** (the status flag) for each selected row. The total number of rows selected is noted and is revised as selections are made.



```
------------------------ Optim: Point-and-Shoot ------------------------------
Command ===>                                                 Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ======================== 1 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** ********************************** TOP **********************************
___    AD005  Conan the Barbarian  Adventure      R        15.00
___    AD008  Dirty Harry          Adventure      R        27.50
___    AD009  For Your Eyes Only   Adventure      PG       23.00
___    AD012  Lethal Weapon        Adventure      R        21.00
___    AD013  The Man Who Would be Adventure      PG       24.00
___    AD014  Raiders of the Lost  Adventure      PG       14.00
___ S  CH002  Old Yeller           Children       PG        9.00
___ S  CH005  Dumbo                Children       G         9.00
___ S  CH006  Willie Wonka & the C Children       G        14.00
___ S  CH007  Fantasia             Children       G        37.40
___    CM008  Beetlejuice          Comedy         PG       26.00
___    CM009  Airplane!            Comedy         PG       14.00
___    CM010  City Slicker         Comedy         PG13     20.00
___ S  CM012  The Gods Must be Cra Comedy         PG       20.90
___    CM015  The Linguini Inciden Comedy         R        21.00
```

Figure 33. Selected Rows Identified

The data cannot be edited in a Point-and-Shoot session. However, available commands help you navigate the data display. These commands include:

Scroll Use ISPF-like commands to scroll vertically or horizontally. You can use the LOCK command to “lock” individual columns so that they remain in the display when you scroll back and forth.

Locate Locate specific data using the FIND command.

Exclude
Exclude or display one or more rows of data using the EXCLUDE and SHOW primary and line commands.

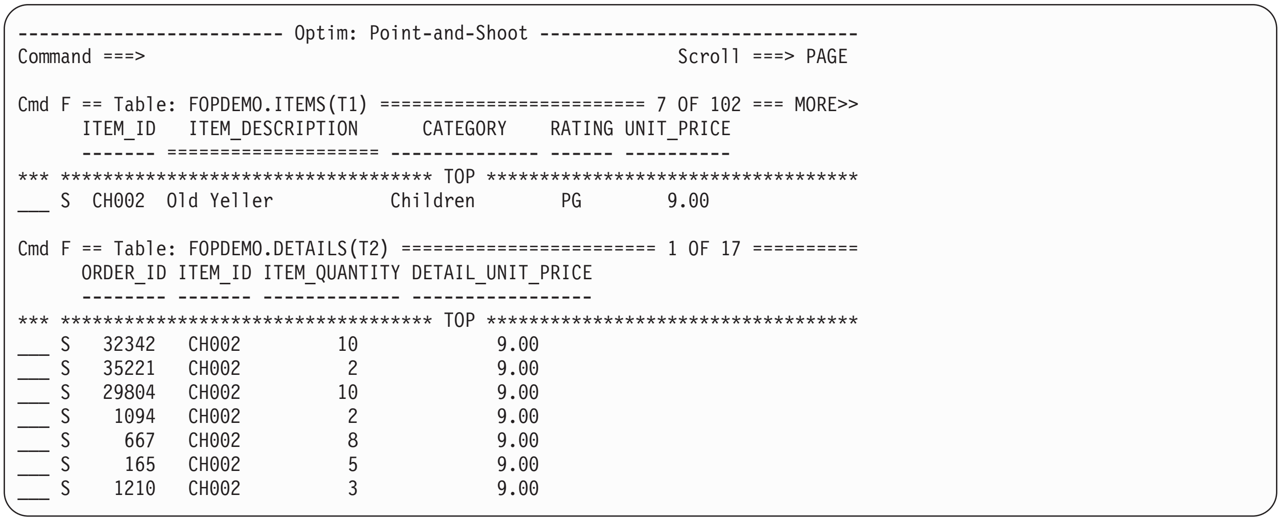
Specify Criteria
Specify selection and sort criteria to choose rows and the order in which they are displayed.

Joining to Other Tables

A Point-and-Shoot session begins with the set of rows from the Start Table. Related rows may be added to the display using the JOIN primary command or the Join line command.

The Join facility allows you to join a displayed row to related rows in another table and display the rows from the related table with the row in the current table. The Join facility provides a useful way to inspect the related data so that you can select the appropriate sets of data.

To display the DETAILS rows related to an ITEMs row, join the DETAILS table to the ITEMS table. Type the J line command in **Cmd** for ITEM_ID CH002 in the ITEMS table and press ENTER; then use the S line command on the resulting pop-up to select the DETAILS table and join that table to the ITEMS table.



```
------------------------- Optim: Point-and-Shoot ------------------------------
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 7 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- ==================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___ S  CH002  Old Yeller           Children        PG        9.00

Cmd F == Table: FOPDEMO.DETAILS(T2) ======================== 1 OF 17 ==========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP ***********************************
___ S   32342   CH002         10            9.00
___ S   35221   CH002          2            9.00
___ S   29804   CH002         10            9.00
___ S    1094   CH002          2            9.00
___ S     667   CH002          8            9.00
___ S     165   CH002          5            9.00
___ S    1210   CH002          3            9.00
```

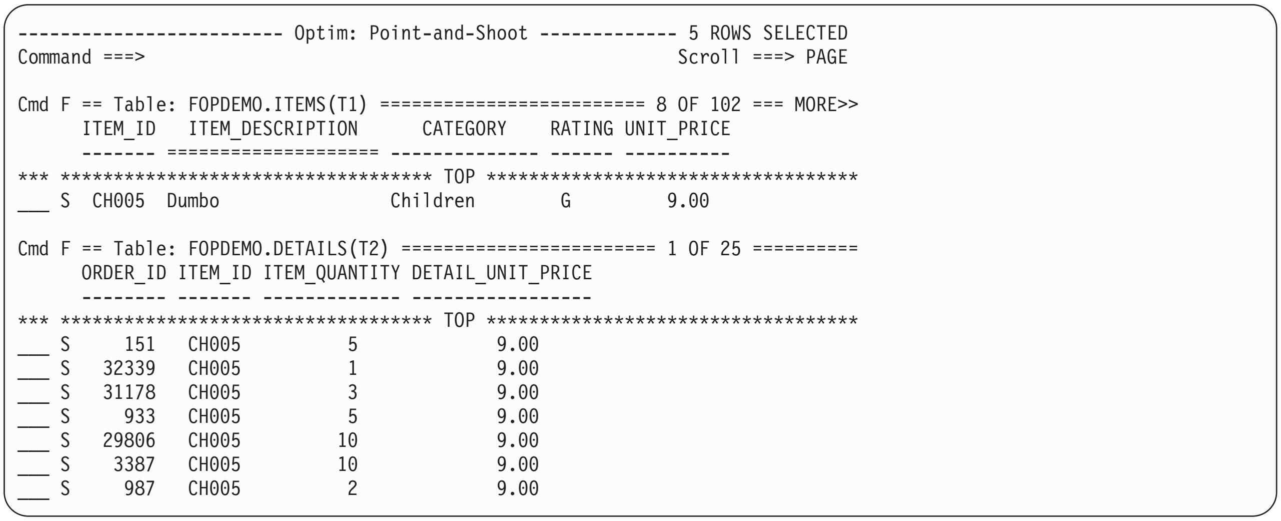
Figure 34. Multi-table Display

This item has 17 child rows in the DETAILS table. Note the S in **F** (the status flag) for the DETAILS rows. These rows will be selected with the ITEMS row when the data is extracted from the database.

Coordinated Scroll

All ISPF scrolling facilities are supported and coordinated when more than one table is displayed. When rows in a table are scrolled up or down, any lower-level displays scroll so that related rows are displayed at all times.

To display the DETAILS rows related to the next ITEMS row, position the cursor on the ITEMS table entry and press the DOWN function key (usually PF8). Twenty-five DETAILS rows are related to the next ITEMS row.

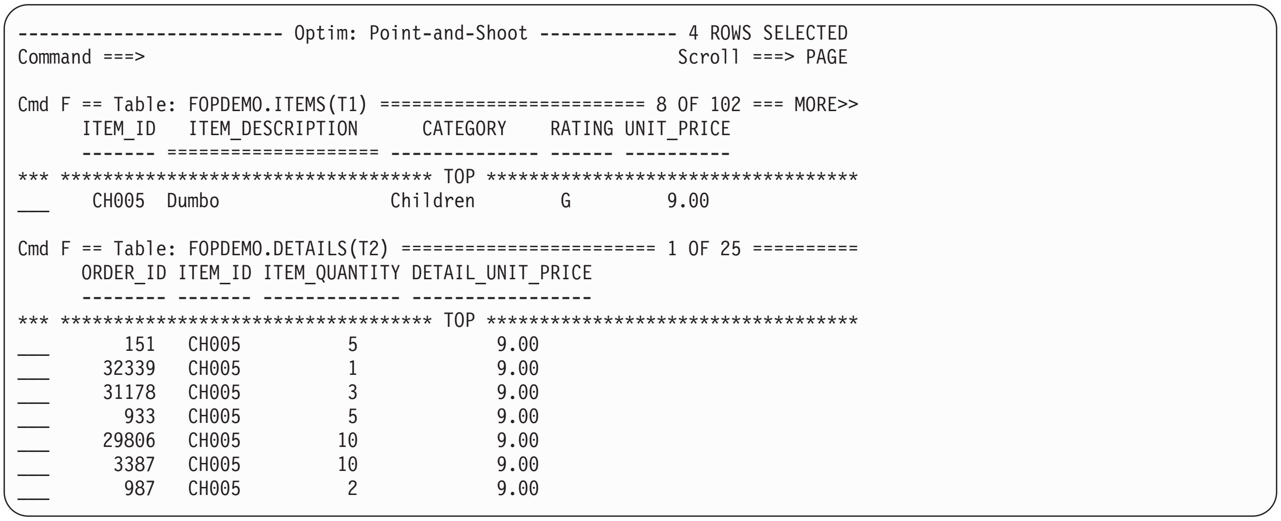


```
------------------------- Optim: Point-and-Shoot ------------- 5 ROWS SELECTED
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 8 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___ S  CH005  Dumbo                Children       G         9.00

Cmd F == Table: FOPDEMO.DETAILS(T2) ======================== 1 OF 25 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP ***********************************
___ S     151   CH005         5            9.00
___ S   32339   CH005         1            9.00
___ S   31178   CH005         3            9.00
___ S     933   CH005         5            9.00
___ S   29806   CH005        10            9.00
___ S    3387   CH005        10            9.00
___ S     987   CH005         2            9.00
```

Figure 35. Scrolling Multi-table Display

Unselecting Rows

If you unselect an ITEMS row, all related DETAILS rows are also unselected. For example, type UR (unselect) in **Cmd** for the ITEMS row CH005 and press ENTER. As shown in the following figure, the S indicator for both tables is removed from the **F** status flag. The rows are no longer selected. Note that the number of selected rows indicator is also revised to indicate that 4 ROWS are now SELECTED, instead of 5.



```
------------------------- Optim: Point-and-Shoot ------------- 4 ROWS SELECTED
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 8 OF 102 === MORE>>
      ITEM_ID   ITEM_DESCRIPTION      CATEGORY    RATING UNIT_PRICE
      ------- =================== -------------- ------ ----------
*** *********************************** TOP ***********************************
___    CH005  Dumbo                Children       G         9.00

Cmd F == Table: FOPDEMO.DETAILS(T2) ======================== 1 OF 25 =========
      ORDER_ID ITEM_ID ITEM_QUANTITY DETAIL_UNIT_PRICE
      -------- ------- ------------- -----------------
*** *********************************** TOP ***********************************
___       151   CH005         5            9.00
___     32339   CH005         1            9.00
___     31178   CH005         3            9.00
___       933   CH005         5            9.00
___     29806   CH005        10            9.00
___      3387   CH005        10            9.00
___       987   CH005         2            9.00
```

Figure 36. Unselected Rows

Joining Additional Tables

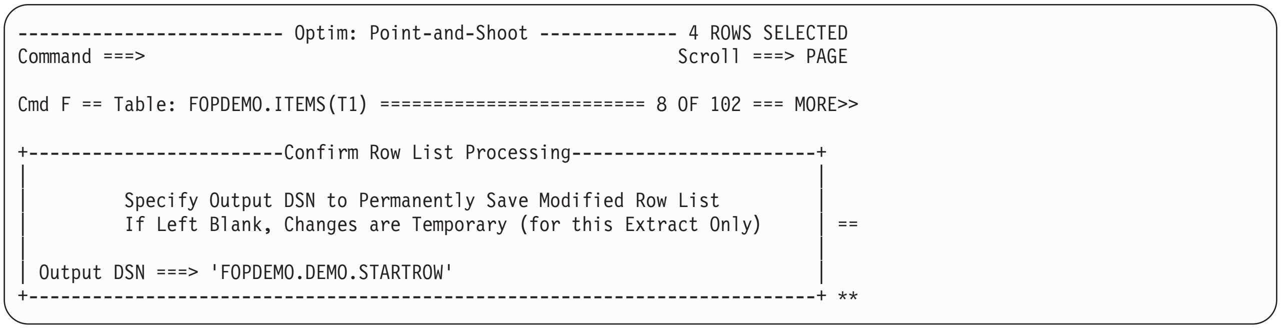
You can also join from DETAILS to ORDERS and from ORDERS to CUSTOMERS. Related data for each joined table is always displayed.

Unjoining Tables

The UNJOIN primary command or the UNJoin line command severs a join between tables and removes unjoined tables from the display.

Selection Complete

When you have completed your selections, use END to return to the Select Tables/Views for AD panel. (You do not have to unjoin the joined tables prior to returning to the Select Tables/Views for AD panel.) Move prompts for the name of a data set in which to store your Point-and-Shoot selections or Row List.



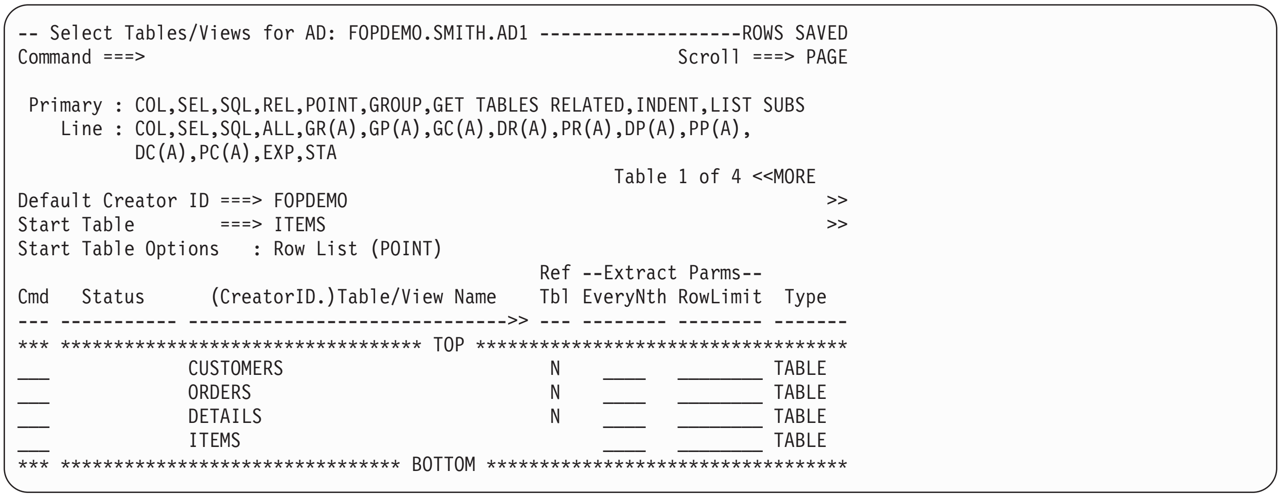
```
------------------------- Optim: Point-and-Shoot ------------- 4 ROWS SELECTED
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: FOPDEMO.ITEMS(T1) ========================= 8 OF 102 === MORE>>

+------------------------Confirm Row List Processing-----------------------+
|                                                                          |
|         Specify Output DSN to Permanently Save Modified Row List         |
|         If Left Blank, Changes are Temporary (for this Extract Only)     | ==
|                                                                          |
| Output DSN ===> 'FOPDEMO.DEMO.STARTROW'                                  |
+--------------------------------------------------------------------------+ **
```

Figure 37. Saving Row List Confirmation Prompt

Output DSN

If you supply an Input DSN at the beginning of the Point-and-Shoot session, the name is shown in **Output DSN**. Since no Input DSN is used in this sample scenario, **Output DSN** is blank. Type the name of the data set you want to use. (This scenario uses 'FOPDEMO.DEMO.STARTROW'.) Press ENTER to update the named data set. If the data set does not exist, Move prompts for necessary information and allocates it before storing your selections and redisplaying the Select Tables/Views for AD panel.

Note that **Start Table Options** indicates rows that have been selected in a Point-and-Shoot session and saved in a Row List.



```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 -------------------ROWS SAVED
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> ITEMS                                               >>
Start Table Options   : Row List (POINT)
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************** TOP ***********************************
___             CUSTOMERS                         N    ____   ________ TABLE
___             ORDERS                            N    ____   ________ TABLE
___             DETAILS                           N    ____   ________ TABLE
___             ITEMS                                  ____   ________ TABLE
*** ******************************** BOTTOM **********************************
```

Figure 38. Specifications Indicated

Group Selection Processing

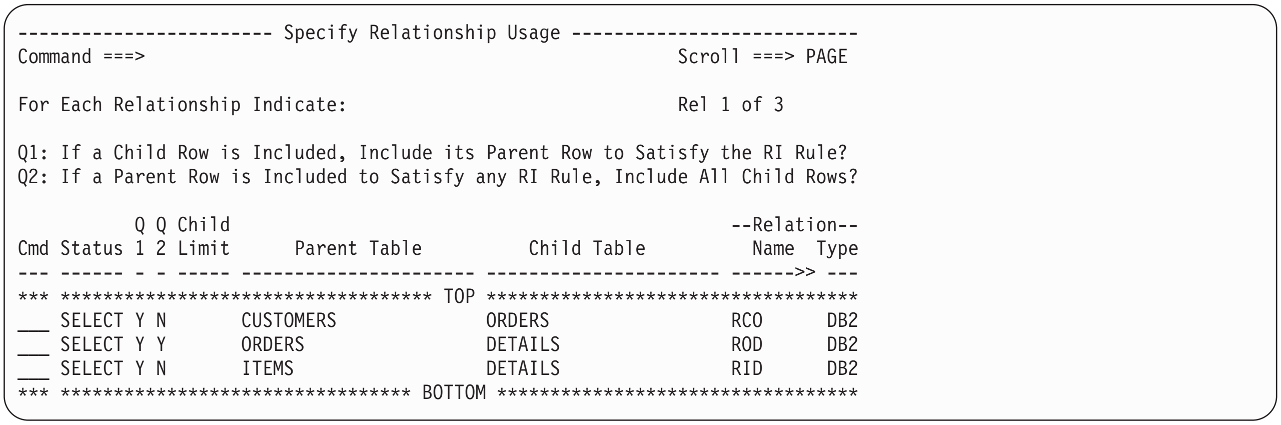
Rather than select Start Table rows individually, you can use group selection processing to extract rows with selected values in a column to a maximum number of rows per value. Use the GROUP command for this function, as described in the *Common Elements Manual* under Group Selection Processing.

Paths

As for Scenario 1, you should review the list of relationships before executing the Extract Process. Rather than return to the EXTRACT Process menu however, use the REL primary command to display the Specify Relationship Usage panel.

Three relationships are to be traversed in this scenario. The extract begins with the rows from the ITEMS table and proceeds to the related child rows in the DETAILS table. To ensure that the relationships between DETAILS and ORDERS and between ORDERS and CUSTOMERS are traversed from child to parent, confirm that **Q1** is Y for these relationships. (This is the default setting.)

So that complete orders are extracted, you must retrieve any additional DETAILS rows related to the ORDERS rows. For the relationship ROD, between ORDERS and DETAILS, specify Y for **Q2**, as in the following figure. Note that **Status** for all relationships is SELECT.



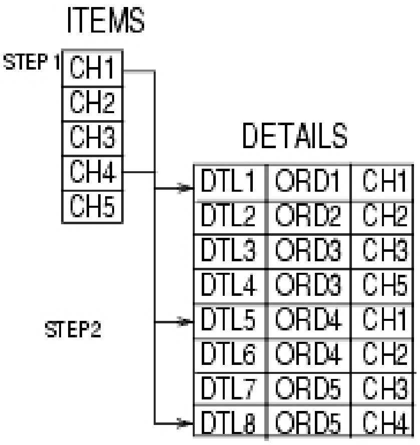
```
------------------------ Specify Relationship Usage ---------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                               --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- --------------------- --------------------- ------>> ---
*** ************************************** TOP *********************************
___ SELECT Y N       CUSTOMERS              ORDERS                 RCO      DB2
___ SELECT Y Y       ORDERS                 DETAILS                ROD      DB2
___ SELECT Y N       ITEMS                  DETAILS                RID      DB2
*** ********************************** BOTTOM **********************************
```

Figure 39. Selecting Relationships for Move Functions

Use END twice to return to the EXTRACT Process menu.

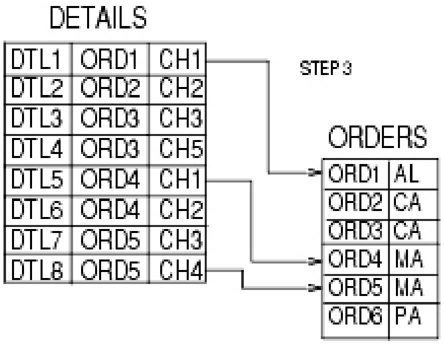
Extract Sequence

Consider the steps that Move performs to extract the data according to the selection criteria, selected relationships and responses to **Q1** and **Q2**. Simplified data is used in the following example for clarity.

Sequence	**Traversal**	**Steps**
ITEMS to DETAILS	 ITEMS STEP 1 CH1 CH2 CH3 CH4 CH5 STEP2 DETAILS DTL1 ORD1 CH1 DTL2 ORD2 CH2 DTL3 ORD3 CH3 DTL4 ORD3 CH5 DTL5 ORD4 CH1 DTL6 ORD4 CH2 DTL7 ORD5 CH3 DTL8 ORD5 CH4	**Step 1** The CH1 and CH4 Start Table rows are selected using Point-and-Shoot. **Step 2** DTL1, DTL5, and DTL8, the DETAILS related to the extracted ITEMS row, are extracted by traversing the relationship RID from parent to child.

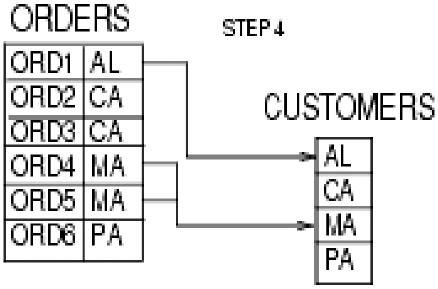
Sequence **Traversal** **Steps**

DETAILS to ORDERS



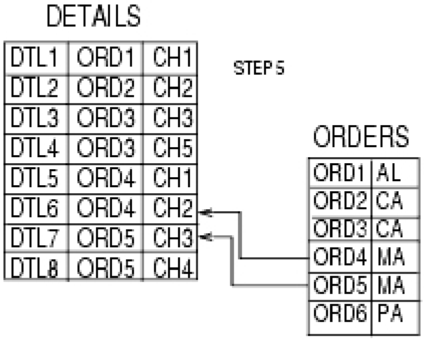
Step 3 ORD1, ORD4, and ORD5, the ORDERS related to the extracted DETAILS rows, are extracted by traversing the relationship ROD from child to parent.

ORDERS to CUSTOMERS



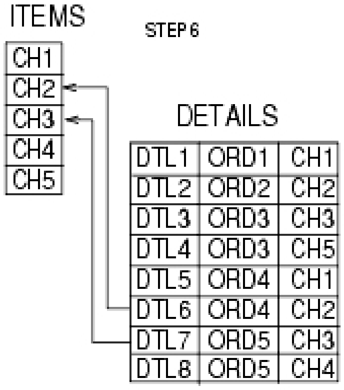
Step 4 The CUSTOMERS related to the ORDERS are extracted by traversing the relationship RCO from child to parent.

ORDERS to DETAILS



Step 5 DTL6 and DTL7, the additional DETAILS related to the extracted ORDERS rows, are extracted by traversing the relationship ROD from parent to child because **Q2** is Y.

DETAILS to ITEMS



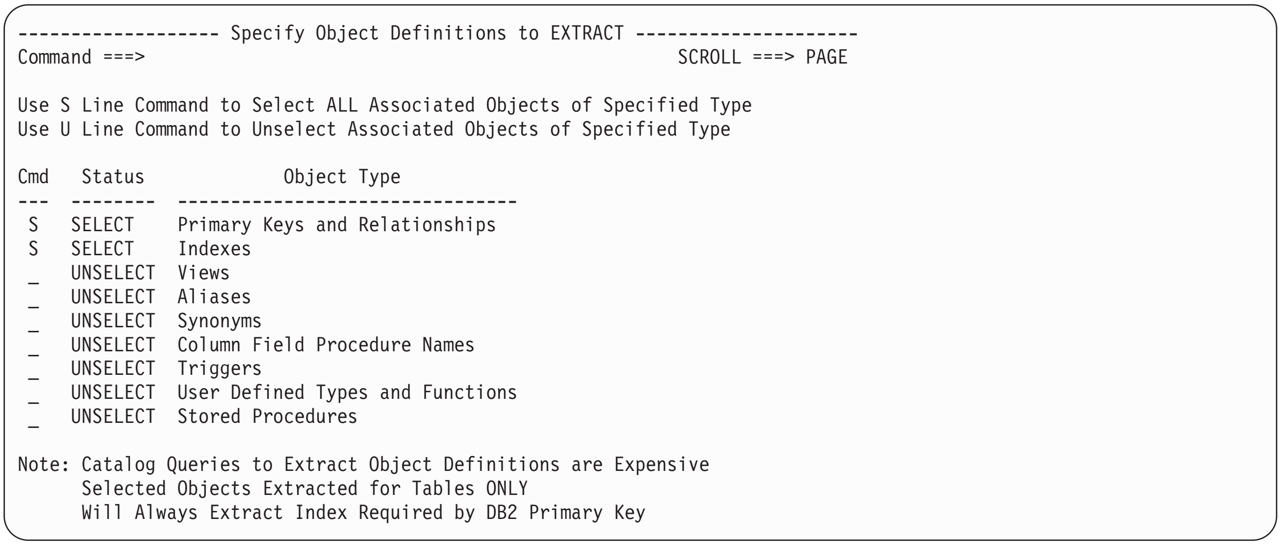
Step 6 CH2 and CH3, the ITEMS rows related to the additional DETAILS rows, are extracted by traversing the relationship RID from child to parent.

Specify Object Definitions

For this scenario, primary keys, relationships, and indexes are extracted. Move needs the definitions for primary keys, relationships, and indexes in order to create them at the destination.

Select Option 3 OBJECTS on the EXTRACT Process menu to display the Specify Object Definitions to EXTRACT panel.

By default, primary keys, relationships, and indexes are selected when the panel is displayed. The **Status** for other object definitions is UNSELECT.



```
------------------- Specify Object Definitions to EXTRACT ---------------------
Command ===>                                                  SCROLL ===> PAGE

Use S Line Command to Select ALL Associated Objects of Specified Type
Use U Line Command to Unselect Associated Objects of Specified Type

Cmd   Status            Object Type
---  --------  --------------------------------
 S   SELECT    Primary Keys and Relationships
 S   SELECT    Indexes
 _   UNSELECT  Views
 _   UNSELECT  Aliases
 _   UNSELECT  Synonyms
 _   UNSELECT  Column Field Procedure Names
 _   UNSELECT  Triggers
 _   UNSELECT  User Defined Types and Functions
 _   UNSELECT  Stored Procedures

Note: Catalog Queries to Extract Object Definitions are Expensive
      Selected Objects Extracted for Tables ONLY
      Will Always Extract Index Required by DB2 Primary Key
```

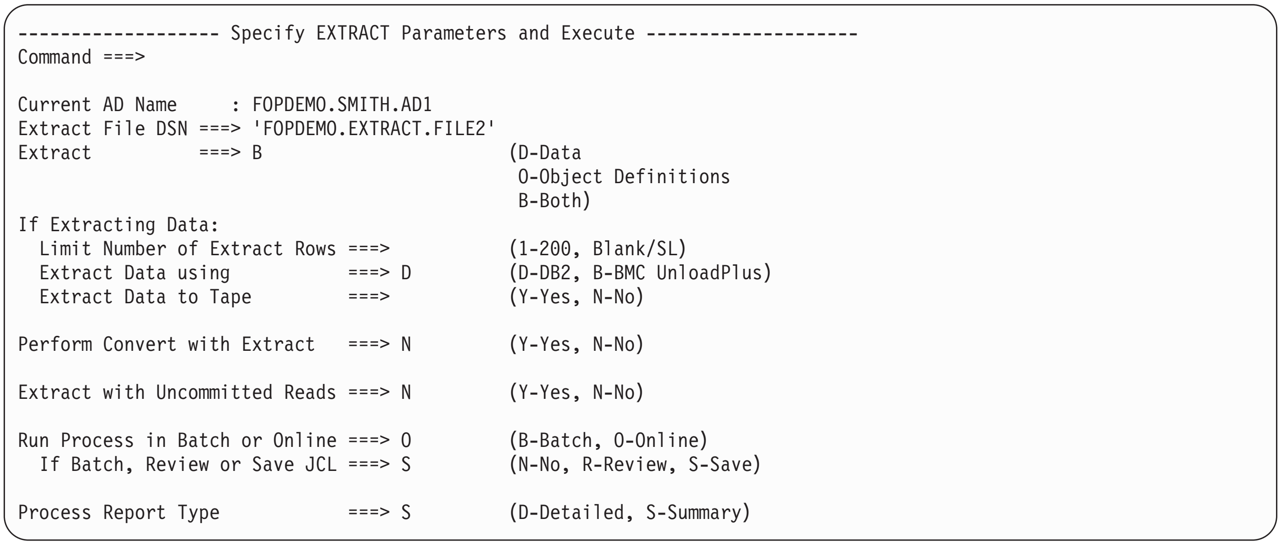
Figure 40. Specify Object Definitions

To select or unselect the object definitions, you use the S and U line commands. For this scenario, confirm that only Primary Keys and Relationships and Indexes are selected (as indicated in the figure), and use END to return to the EXTRACT Process menu.

Perform the Extract

To execute the Extract Process, select Option 4 PERFORM on the EXTRACT Process menu to display the Specify EXTRACT Parameters and Execute panel.

The name of the Extract File you specified for the previous extract has been profiled and is displayed in **Extract File DSN**. You can reuse this file or change the name in order to create a new Extract File.



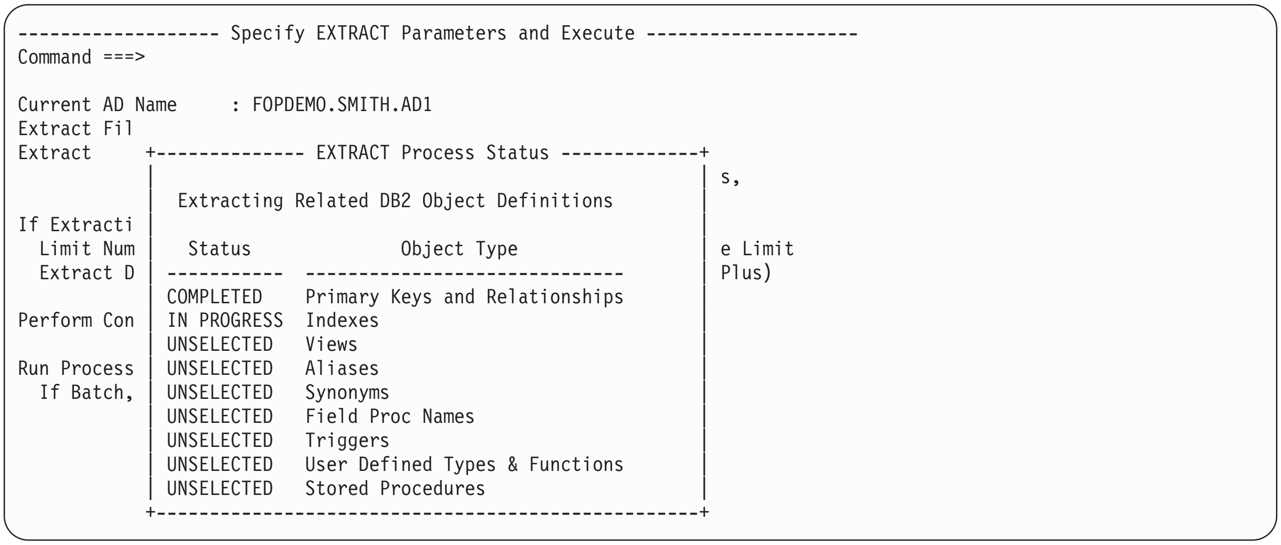
```
------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE2'
Extract          ===> B                      (D-Data
                                              O-Object Definitions
                                              B-Both)
If Extracting Data:
  Limit Number of Extract Rows ===>          (1-200, Blank/SL)
  Extract Data using           ===> D        (D-DB2, B-BMC UnloadPlus)
  Extract Data to Tape         ===>          (Y-Yes, N-No)

Perform Convert with Extract   ===> N        (Y-Yes, N-No)

Extract with Uncommitted Reads ===> N        (Y-Yes, N-No)

Run Process in Batch or Online ===> O        (B-Batch, O-Online)
  If Batch, Review or Save JCL ===> S        (N-No, R-Review, S-Save)

Process Report Type            ===> S        (D-Detailed, S-Summary)
```

Figure 41. Extract Process Parameters

For this scenario, the name of the Extract File is changed to FOPDEMO.EXTRACT.FILE2. Specify B at the **Extract** prompt in order to extract both the data and the selected object definitions. Press ENTER.

As in Scenario 1, the Extract Process is performed online. If necessary, Move prompts for information to allocate the Extract File data set. Also, Move will display warnings or messages about any errors detected when evaluating the Access Definition before performing the Extract Process. Press ENTER to continue with the process or use END to revise.

The status of the Extract Process is displayed as the object definitions are extracted. **Status** is shown as IN PROGRESS for a type of object definition as the objects are extracted. After each type of object is extracted, the **Status** is COMPLETED. The **Status** is SELECTED for any selected object definition that has not yet been extracted and UNSELECTED for any object definition that is not to be extracted in the process.



```
------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract Fil
Extract     +-------------- EXTRACT Process Status -------------+
            |                                                   | s,
            |  Extracting Related DB2 Object Definitions        |
If Extracti |                                                   |
  Limit Num |   Status             Object Type                  | e Limit
  Extract D | -----------  ------------------------------       | Plus)
            | COMPLETED    Primary Keys and Relationships       |
Perform Con | IN PROGRESS  Indexes                              |
            | UNSELECTED   Views                                |
Run Process | UNSELECTED   Aliases                              |
  If Batch, | UNSELECTED   Synonyms                             |
            | UNSELECTED   Field Proc Names                     |
            | UNSELECTED   Triggers                             |
            | UNSELECTED   User Defined Types & Functions       |
            | UNSELECTED   Stored Procedures                    |
            +---------------------------------------------------+
```

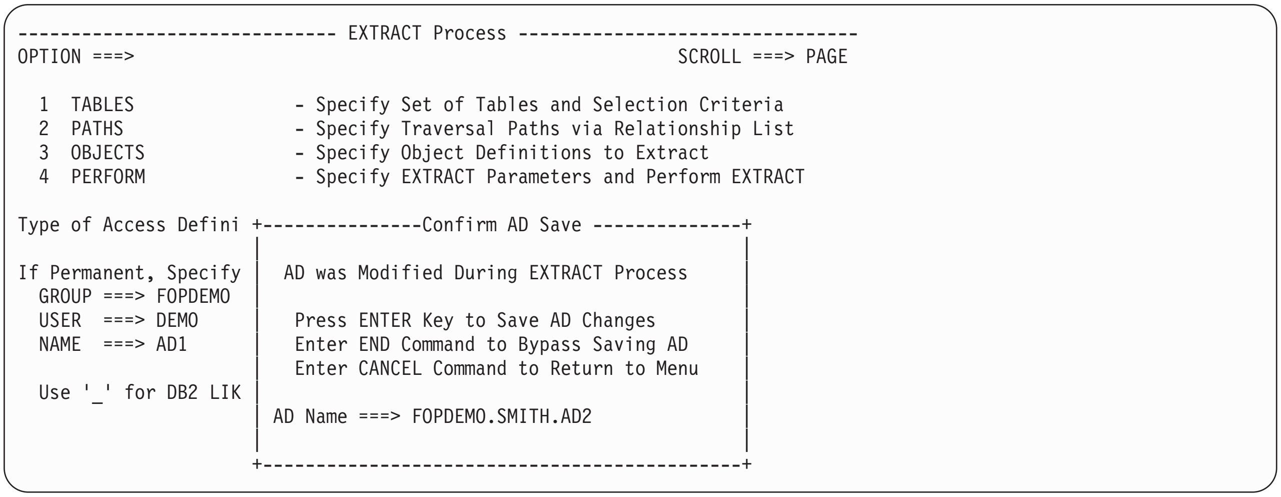
Figure 42. Extract Process Parameters

The **Status** is updated when Move begins to extract the next object type on the list. Once all selected object definitions are extracted, the status pop-up window is removed from the display.

As the data is extracted, a second status pop-up window is displayed to indicate the table that is being processed. When the process is complete, a summary report is displayed. The summary report includes information about the extracted object definitions and data. For example, the total number of extracted object definitions for each type is listed and the number of data rows extracted from each table is provided on a table-by-table basis. (For information about the status display while the data is extracted and the summary report, see “Perform the Extract” on page 19.)

Use END twice to return to the Data Migration menu.

Because the Access Definition was modified, a confirmation prompt is displayed. You can specify a new name in order to create a new Access Definition or update the current Access Definition. For this sample scenario, specify a new name at this prompt, FOPDEMO.SMITH.AD2.



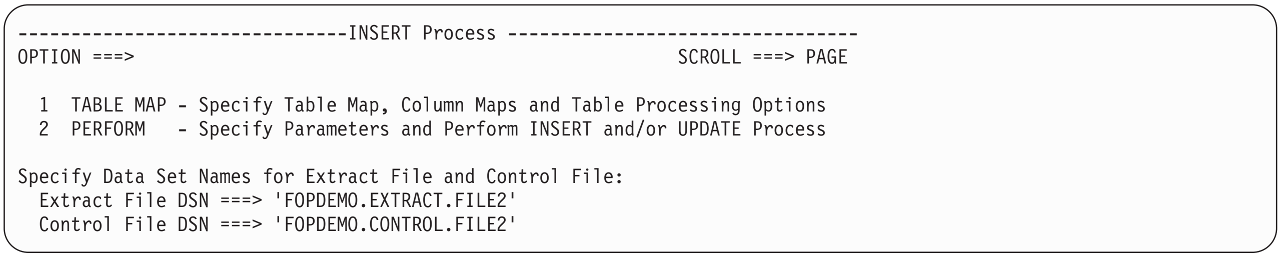
```
----------------------------- EXTRACT Process -------------------------------
OPTION ===>                                                 SCROLL ===> PAGE

  1  TABLES              - Specify Set of Tables and Selection Criteria
  2  PATHS               - Specify Traversal Paths via Relationship List
  3  OBJECTS             - Specify Object Definitions to Extract
  4  PERFORM             - Specify EXTRACT Parameters and Perform EXTRACT

Type of Access Defini +---------------Confirm AD Save -------------+
                      |                                            |
If Permanent, Specify |  AD was Modified During EXTRACT Process    |
  GROUP ===> FOPDEMO  |                                            |
  USER  ===> DEMO     |   Press ENTER Key to Save AD Changes       |
  NAME  ===> AD1      |   Enter END Command to Bypass Saving AD    |
                      |   Enter CANCEL Command to Return to Menu   |
  Use '_' for DB2 LIK |                                            |
                      | AD Name ===> FOPDEMO.SMITH.AD2             |
                      |                                            |
                      +--------------------------------------------+
```

Figure 43. Access Definition Confirmation Prompt

Press ENTER to save the Access Definition.

Defining the Destination

After Move has generated the Extract File, you can use Option 2 INSERT on the Data Migration menu to specify the destination and display the INSERT Process menu.



```
-------------------------------INSERT Process --------------------------------
OPTION ===>                                                 SCROLL ===> PAGE

  1  TABLE MAP - Specify Table Map, Column Maps and Table Processing Options
  2  PERFORM   - Specify Parameters and Perform INSERT and/or UPDATE Process

Specify Data Set Names for Extract File and Control File:
  Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE2'
  Control File DSN ===> 'FOPDEMO.CONTROL.FILE2'
```

Figure 44. Scenario 2 – INSERT Process Menu

The details of this panel are discussed in “Defining the Destination” on page 22.

For this scenario, you create a test database using data from the production system. The source data has been extracted and is stored in the Extract File, FOPDEMO.EXTRACT.FILE2. As in Scenario 1, the destination tables match the source tables and are uniquely identified by the Creator ID. In addition, a Column Map prevents some of the source data from being inserted at the destination. The primary keys and relationships and the indexes from the production database are copied into the test database.

To perform this process, type the name of the Extract File on the INSERT Process menu. The name of the current or most recent Extract File is provided automatically. For this scenario, FOPDEMO.EXTRACT.FILE2 is the name of the Extract File used to create the test database.

A Control File is also required for the Insert Process. The name of the current or most recent Control File is provided automatically. You may use the same file or specify a different one. For this scenario, the Control File is FOPDEMO.CONTROL.FILE2.

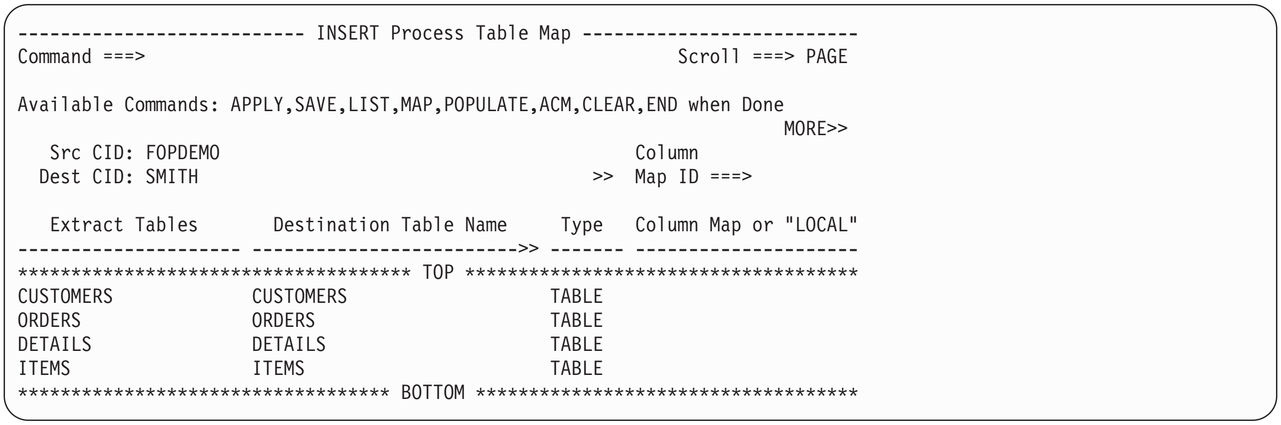
Specify the Tables

Select Option 1 TABLE MAP from the INSERT Process menu to display the INSERT Process Table Map panel.

The details of the INSERT Process Table Map panel are provided in “Specify the Tables” on page 23.

Table Map Panel

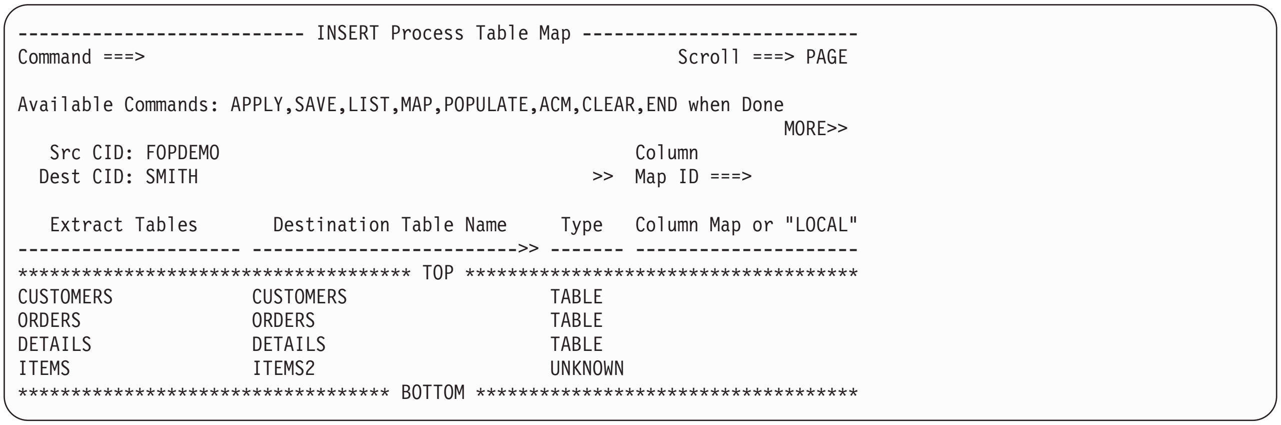
Specify the **Dest CID**. If you have performed the Insert Process used in Scenario 1 and enter the same value for the **Dest CID**, all destination tables are known and Move inserts the type, as in the following figure.



```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID: SMITH                                     >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
-------------------- ------------------------->> ------- ---------------------
************************************* TOP *************************************
CUSTOMERS            CUSTOMERS                  TABLE
ORDERS               ORDERS                     TABLE
DETAILS              DETAILS                    TABLE
ITEMS                ITEMS                      TABLE
*********************************** BOTTOM ************************************
```

Figure 45. Insert Process Table Map

Move automatically creates any tables that do not exist, as well as any other object definitions in the Extract File. However, Move does not automatically create object definitions if the tables exist at the destination. (You can use the CREATE option on the **Data Migration** menu to create extracted objects, regardless of whether the extracted tables exist at the destination.) For this scenario, change the **Destination Table Name** for the ITEMS table to preserve the data in the SMITH.ITEMS table and to provide an unknown table.



```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID: SMITH                                     >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
--------------------- --------------------------->> ------- ---------------------
************************************* TOP *************************************
CUSTOMERS             CUSTOMERS                    TABLE
ORDERS                ORDERS                       TABLE
DETAILS               DETAILS                      TABLE
ITEMS                 ITEMS2                       UNKNOWN
*********************************** BOTTOM ************************************
```

Figure 46. One Table UNKNOWN

Because of the unknown table, Move will automatically prompt you to create it and any other extracted objects when you execute the Insert Process.

In this scenario, you want to change the sales representative and several dates in the source data. Thus, before proceeding to the Insert Process you must define a Column Map.

Column Maps

When the names of source and destination columns match, the Insert proceeds automatically. However, if the source and destination do not match or if you want to influence the data that is inserted at the destination, you must use a Column Map.

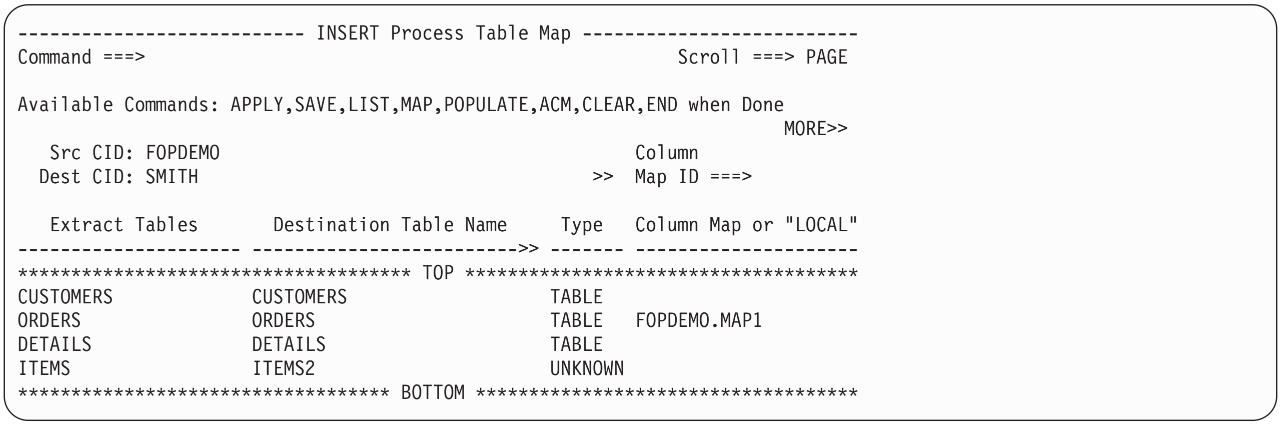
A Column Map is a set of instructions that defines the source for each destination column. Use a Column Map when:

- The names of source columns are different from the names of destination columns.
- A literal value or the result of an expression is inserted in a destination column instead of the source data. This functionality is useful to mask sensitive data, generate unique values, and so forth, when creating a test database. You can also age test data.
- A DB2 default value is inserted into the destination column. (This is especially useful when inserting DATE, TIME, and TIMESTAMP values.)
- A site-defined exit routine is used to derive a value that is inserted into a destination column. (An exit routine is useful when the data to be inserted requires an arithmetic operation or is derived from multiple columns. Note that the entire source row is passed to the exit routine.)
- A source column contains the source data for more than one destination column.

A named Column Map is stored in the Optim Directory and can be used repeatedly and simultaneously by multiple users. Alternatively, you can create a local Column Map that is available only to the current process or Table Map and is not stored in the Optim Directory.

Creating a Column Map

To display the prompts needed to create a Column Map from the INSERT Process Table Map panel, type the name for the destination table that is to have a map in **Column Map or "LOCAL"** and press ENTER. For this scenario, a Column Map named FOPDEMO.MAP1 is named for the ORDERS table.

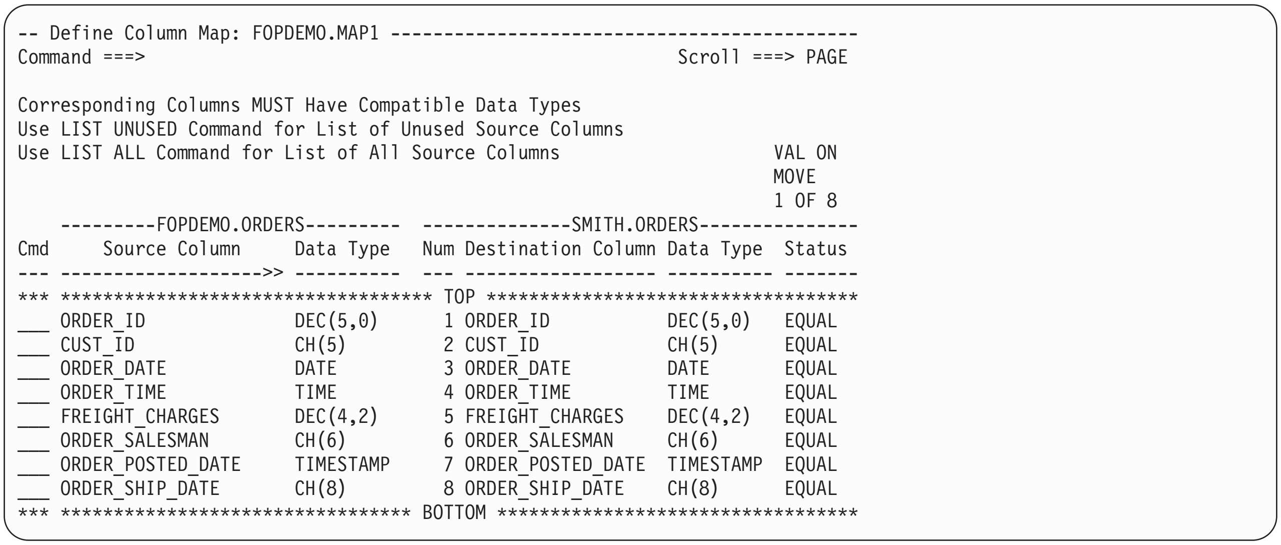


```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID: SMITH                                     >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
--------------------- ------------------------>> ------- ---------------------
************************************* TOP ************************************
CUSTOMERS             CUSTOMERS                TABLE
ORDERS                ORDERS                   TABLE   FOPDEMO.MAP1
DETAILS               DETAILS                  TABLE
ITEMS                 ITEMS2                   UNKNOWN
*********************************** BOTTOM ***********************************
```

Figure 47. Specifying a Column Map

To ensure that you typed the name correctly, Move displays a confirmation prompt before creating the Column Map. Use ENTER to create the Column Map or END to abort the request and redisplay the INSERT Process Table Map panel.

When you press ENTER, the Define Column Map panel is displayed. The source columns are listed on one side of the panel, and the destination columns are listed on the other. Move automatically matches source and destination columns that have the same name and a compatible data type. Matched columns have the **Status** EQUAL or MAPPED, as appropriate.



```
-- Define Column Map: FOPDEMO.MAP1 --------------------------------------------
Command ===>                                                  Scroll ===> PAGE

Corresponding Columns MUST Have Compatible Data Types
Use LIST UNUSED Command for List of Unused Source Columns
Use LIST ALL Command for List of All Source Columns                   VAL ON
                                                                       MOVE
                                                                       1 OF 8
    ---------FOPDEMO.ORDERS---------  --------------SMITH.ORDERS---------------
Cmd     Source Column     Data Type   Num Destination Column Data Type  Status
--- ------------------->> ----------  --- ------------------ ---------- -------
*** ********************************** TOP ***********************************
___ ORDER_ID              DEC(5,0)      1 ORDER_ID           DEC(5,0)   EQUAL
___ CUST_ID               CH(5)         2 CUST_ID            CH(5)      EQUAL
___ ORDER_DATE            DATE          3 ORDER_DATE         DATE       EQUAL
___ ORDER_TIME            TIME          4 ORDER_TIME         TIME       EQUAL
___ FREIGHT_CHARGES       DEC(4,2)      5 FREIGHT_CHARGES    DEC(4,2)   EQUAL
___ ORDER_SALESMAN        CH(6)         6 ORDER_SALESMAN     CH(6)      EQUAL
___ ORDER_POSTED_DATE     TIMESTAMP     7 ORDER_POSTED_DATE  TIMESTAMP  EQUAL
___ ORDER_SHIP_DATE       CH(8)         8 ORDER_SHIP_DATE    CH(8)      EQUAL
*** ******************************** BOTTOM **********************************
```

Figure 48. Define Column Map - Initial Column List

In this example, the destination table matches the source table exactly. Therefore, all columns match and the **Status** is EQUAL for each pair.

The data type for each source or destination column is automatically supplied and cannot be modified. Although the data type of the source and destination do not have to match exactly, they must be compatible. (See Compatibility Rules in the *Common Elements Manual* for further information.)

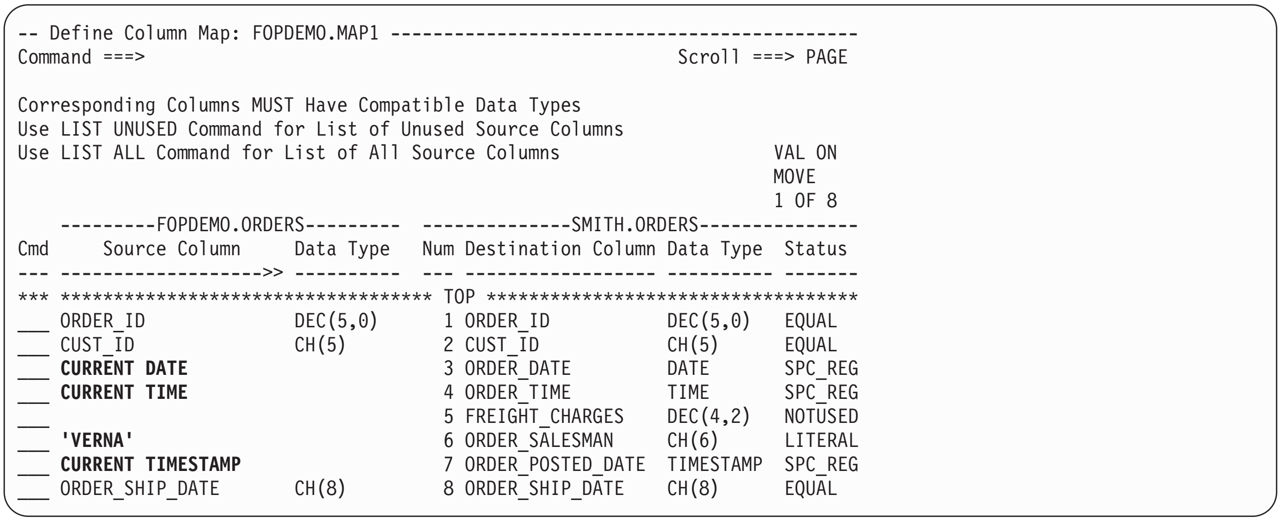
Specifying Other Destination Values

For this scenario, values other than the extracted source data are to be inserted at the destination.

- The columns ORDER_DATE and ORDER_TIME must reflect the date and time the row is inserted into the destination table. DB2 special register values are inserted in these destination columns.

- The column FREIGHT_CHARGES is not used. That is, the source data is not inserted into the destination FREIGHT_CHARGES column when the Insert Process is executed. The DB2 default value for the column is inserted in this destination column.
- The destination column ORDER_SALESMAN contains a literal value in order to mask sensitive data.

You can overtype any Move-inserted **Source Column** entry with a desired entry. In the following figure, the Define Column Map panel shows the **Source Column** entries needed for the changes listed. Note that literals other than numerics must be in single quotes. Special registers can be used. Although not shown here, expressions can also be specified.



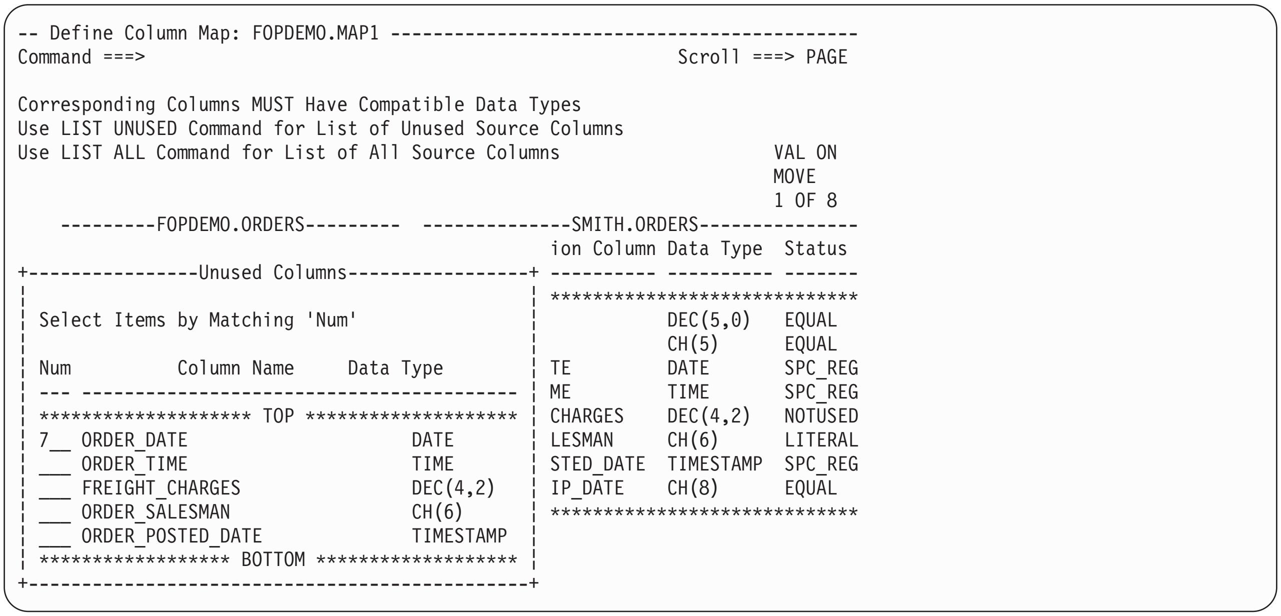
```
-- Define Column Map: FOPDEMO.MAP1 -------------------------------------------
Command ===>                                                  Scroll ===> PAGE

Corresponding Columns MUST Have Compatible Data Types
Use LIST UNUSED Command for List of Unused Source Columns
Use LIST ALL Command for List of All Source Columns                    VAL ON
                                                                       MOVE
                                                                       1 OF 8
    ---------FOPDEMO.ORDERS---------  --------------SMITH.ORDERS---------------
Cmd     Source Column     Data Type   Num Destination Column Data Type  Status
--- ------------------->> ----------  --- ------------------ ---------- -------
*** *********************************** TOP ***********************************
___ ORDER_ID              DEC(5,0)      1 ORDER_ID           DEC(5,0)   EQUAL
___ CUST_ID               CH(5)         2 CUST_ID            CH(5)      EQUAL
___ CURRENT DATE                        3 ORDER_DATE         DATE       SPC_REG
___ CURRENT TIME                        4 ORDER_TIME         TIME       SPC_REG
___                                     5 FREIGHT_CHARGES    DEC(4,2)   NOTUSED
___ 'VERNA'                             6 ORDER_SALESMAN     CH(6)      LITERAL
___ CURRENT TIMESTAMP                   7 ORDER_POSTED_DATE  TIMESTAMP  SPC_REG
___ ORDER_SHIP_DATE       CH(8)         8 ORDER_SHIP_DATE    CH(8)      EQUAL
```

Figure 49. Define Column Map - Specifying Source

Move automatically supplies the source column **Data Type**, when appropriate, and revises the **Status**.

LIST

When the list of columns is extensive, you can use the LIST UNUSED command to select from unmapped source columns. LIST ALL provides a list of all source columns. With either selection list, you type a value in **Num** for the source column equal to the **Num** value for the matching destination column. In the following figure, the ORDER_DATE source column is matched with the ORDER_POSTED_DATE destination column. (The data types are compatible.)



```
-- Define Column Map: FOPDEMO.MAP1 --------------------------------------------
Command ===>                                                  Scroll ===> PAGE

Corresponding Columns MUST Have Compatible Data Types
Use LIST UNUSED Command for List of Unused Source Columns
Use LIST ALL Command for List of All Source Columns                    VAL ON
                                                                       MOVE
                                                                       1 OF 8
    ---------FOPDEMO.ORDERS---------  --------------SMITH.ORDERS---------------
                                                  ion Column Data Type  Status
+----------------Unused Columns-----------------+ ---------- ---------- -------
¦                                               ¦ *****************************
¦ Select Items by Matching 'Num'                ¦            DEC(5,0)   EQUAL
¦                                               ¦            CH(5)      EQUAL
¦ Num          Column Name     Data Type        ¦ TE         DATE       SPC_REG
¦ --- ----------------------------------------- ¦ ME         TIME       SPC_REG
¦ ******************** TOP ******************** ¦ CHARGES    DEC(4,2)   NOTUSED
¦ 7__ ORDER_DATE                     DATE       ¦ LESMAN     CH(6)      LITERAL
¦ ___ ORDER_TIME                     TIME       ¦ STED_DATE  TIMESTAMP  SPC_REG
¦ ___ FREIGHT_CHARGES                DEC(4,2)   ¦ IP_DATE    CH(8)      EQUAL
¦ ___ ORDER_SALESMAN                 CH(6)      ¦ *****************************
¦ ___ ORDER_POSTED_DATE              TIMESTAMP  ¦
¦ ****************** BOTTOM ******************* ¦
+-----------------------------------------------+
```

Figure 50. Matching Unused Source Columns

END List

When you have finished using this list to match the columns, use END or press ENTER to return to the Define Column Map panel. Move checks the mapping you have selected and updates the **Status**.

Map to Source

After specifying a literal, expression, special register, or other special source column entry for the Column Map, you may decide to revert to the name of the source column. The SRC line command automatically inserts the name of the source column, if available, mapping it to the destination column. For this scenario, use SRC to match the destination column ORDER_POSTED_DATE with the original source column.

Additional Features

You frequently need to age dates to create test data. You can use the AGE function to force a specific date, increment a date value, or identify a column as containing a date to be aged when the Insert is performed. When the Insert Process is initiated, a prompt asks for global aging parameters. These parameters are used to age the identified columns for which the aging parameters are not explicitly specified in the Column Map. (For information about aging, see Age Date Values in the *Move User Manual* and the Move Age Function in the *Common Elements Manual*.)

In addition, you can use the **EXPAND** command to display a 75-character area in which to type a value for the destination column. Expressions and functions can be used. Useful functions that are available include a function to generate a random number, a function to generate sequential numbers and a function to substring a character string. (For details about defining Column Maps, see the *Common Elements Manual*.)

Completed Column Map

When you have completed the Column Map, use END to redisplay the INSERT Process Table Map panel. You can create a Column Map for any destination table. Use END to return to the INSERT Process menu.

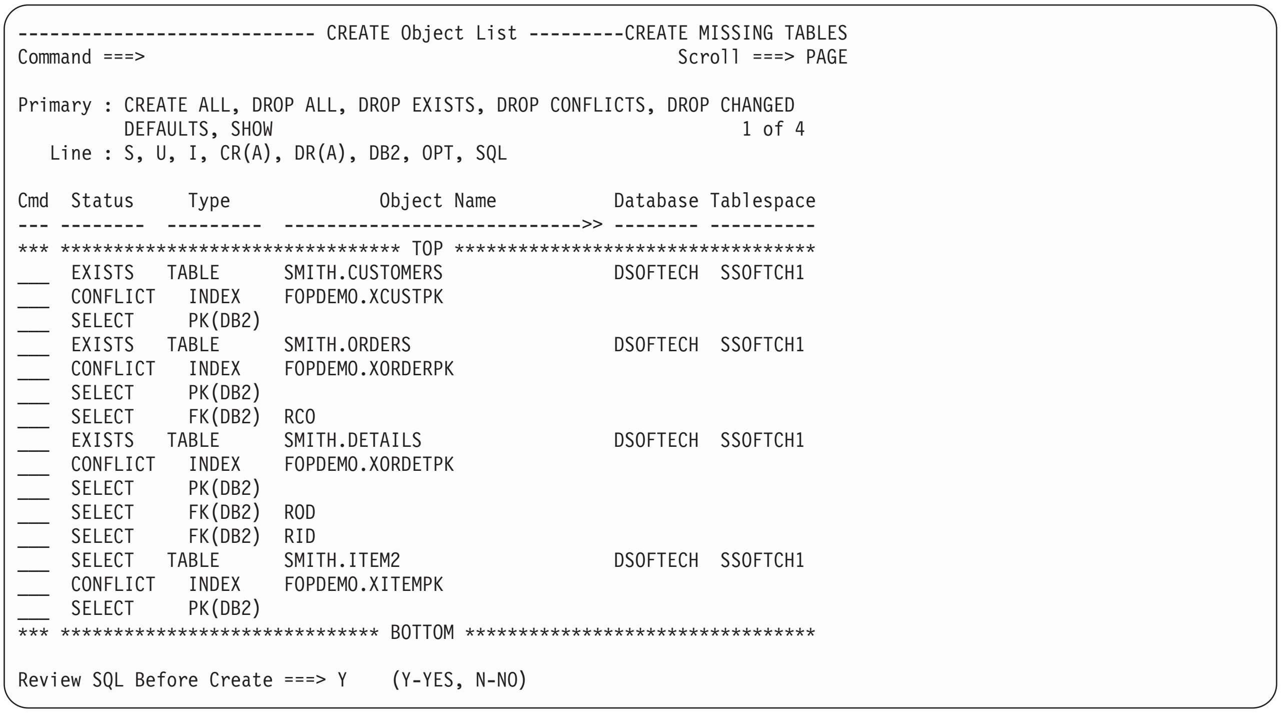
Perform the Insert

Select Option 2 PERFORM on the INSERT Process menu to execute the Insert Process. In this scenario, you will specify Both (insert and update) for **Processing Method to Use** on the Specify INSERT Parameters and Execute panel.

New rows are inserted and existing destination rows are updated. Thus, if the primary key value of the source row does not match the value in a row in the destination table, the row is inserted. Also, if the primary key value of the source row matches the value in an existing destination row, the destination row is updated.

Select Object Definitions

Since the ITEM2 table does not exist, Move displays the CREATE Object List panel, as shown in the following figure.



```
---------------------------- CREATE Object List ---------CREATE MISSING TABLES
Command ===>                                                  Scroll ===> PAGE

Primary : CREATE ALL, DROP ALL, DROP EXISTS, DROP CONFLICTS, DROP CHANGED
          DEFAULTS, SHOW                                            1 of 4
   Line : S, U, I, CR(A), DR(A), DB2, OPT, SQL

Cmd  Status     Type             Object Name           Database Tablespace
--- --------  ---------  ---------------------------->> -------- ----------
*** ******************************** TOP *********************************
___  EXISTS   TABLE      SMITH.CUSTOMERS               DSOFTECH  SSOFTCH1
___  CONFLICT   INDEX    FOPDEMO.XCUSTPK
___  SELECT     PK(DB2)
___  EXISTS   TABLE      SMITH.ORDERS                  DSOFTECH  SSOFTCH1
___  CONFLICT   INDEX    FOPDEMO.XORDERPK
___  SELECT     PK(DB2)
___  SELECT     FK(DB2)  RCO
___  EXISTS   TABLE      SMITH.DETAILS                 DSOFTECH  SSOFTCH1
___  CONFLICT   INDEX    FOPDEMO.XORDETPK
___  SELECT     PK(DB2)
___  SELECT     FK(DB2)  ROD
___  SELECT     FK(DB2)  RID
___  SELECT   TABLE      SMITH.ITEM2                   DSOFTECH  SSOFTCH1
___  CONFLICT   INDEX    FOPDEMO.XITEMPK
___  SELECT     PK(DB2)
*** ****************************** BOTTOM ********************************

Review SQL Before Create ===> Y    (Y-YES, N-NO)
```

Figure 51. Create Object List

In addition to the tables in the Extract File, any extracted primary keys, relationships, and indexes or other objects are also listed. The **Status** for tables that were created in the first scenario is EXISTS. Move assigns this status to objects for which the definition in the database is the same as the definition in the Extract File. In this example, the **Status** of other objects, except for indexes, is SELECT (i.e., the objects are to be created).

Note: It is possible that the test database that you are using has been modified so that more objects exist or are in conflict, or no objects are in conflict. The following discussion demonstrates how to handle conflicts, if any are present.

CONFLICT Status

CONFLICT **Status** is assigned when an object definition in the Extract File conflicts with an object definition of the same name at the destination. A conflict arises, for example, when the name of an object definition applies to a different type of object. In the figure, source and destination indexes have the same names but are for different tables. You can:

- Use the DR line command to drop the existing index and allow you to create the index from the Extract File.
- Overtype each index name to create an index of a different name for which a conflict does not exist.
- Specify a default Creator ID to be used for every index that is to be created. This default, which is specified on the Index Defaults panel, can be a literal value or an implicit value, such as the Creator ID used for the table to be created.

For this scenario, if any listed object definition (except a table) is in CONFLICT, simply modify the name. However, if a table is in conflict, you must review the INSERT Process Table Map panel. The **Dest CID** or table name may require modification.

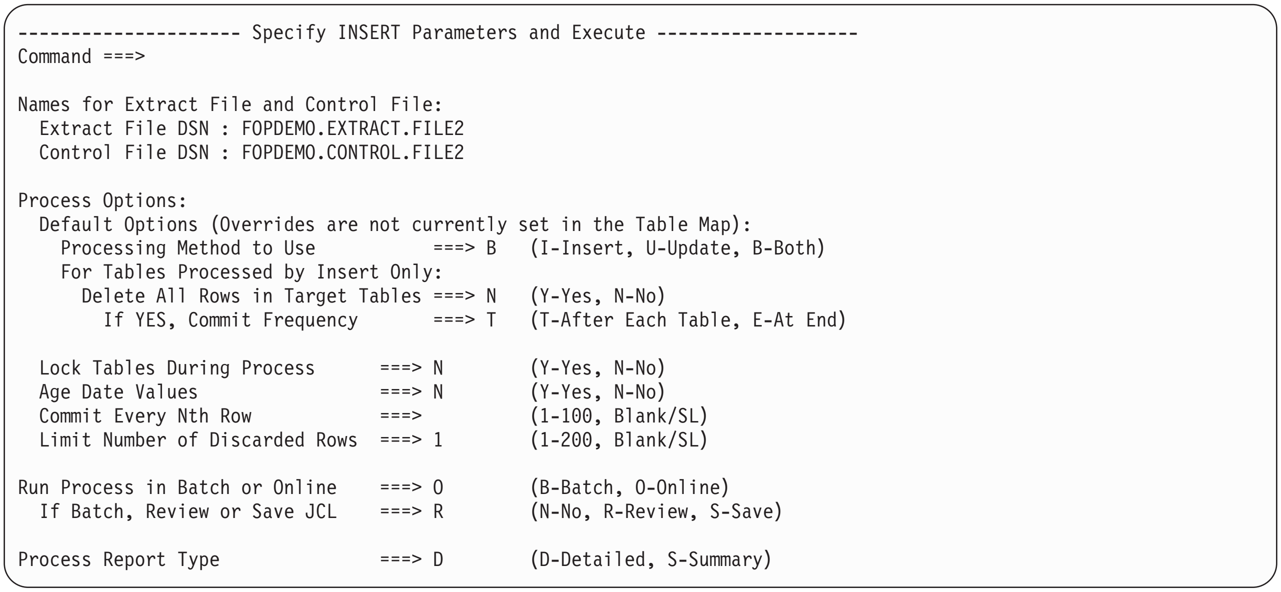
Panels prompt for the default Creator ID for indexes and aliases, and default Database and Tablespace names. See the *Move User Manual* for information about defining default values for object definitions.

Create ITEM2

For this scenario, create the table ITEM2. If the INDEX for ITEM2 is in CONFLICT status (same name as the index for ITEMS), overtype the name. Then use the CRA line command for the ITEM2 table to create the table and all related objects. To display the SQL before execution, you can type Y at the **Review SQL Before Create** prompt.

Use the CREATE ALL primary command to create the primary keys and relationships in SELECT status for the other tables in a single step. As for the ITEM2 table, you can display the SQL if desired.

If displaying the generated SQL, use RUN to execute. After the SQL executes, use END to return to the CREATE Object List panel, and use END a second time to display the Specify INSERT Parameters and Execute panel.



```
--------------------- Specify INSERT Parameters and Execute -------------------
Command ===>

Names for Extract File and Control File:
  Extract File DSN : FOPDEMO.EXTRACT.FILE2
  Control File DSN : FOPDEMO.CONTROL.FILE2

Process Options:
  Default Options (Overrides are not currently set in the Table Map):
    Processing Method to Use          ===> B   (I-Insert, U-Update, B-Both)
    For Tables Processed by Insert Only:
      Delete All Rows in Target Tables ===> N   (Y-Yes, N-No)
        If YES, Commit Frequency       ===> T   (T-After Each Table, E-At End)

  Lock Tables During Process      ===> N       (Y-Yes, N-No)
  Age Date Values                 ===> N       (Y-Yes, N-No)
  Commit Every Nth Row            ===>         (1-100, Blank/SL)
  Limit Number of Discarded Rows  ===> 1       (1-200, Blank/SL)

Run Process in Batch or Online    ===> O       (B-Batch, O-Online)
  If Batch, Review or Save JCL    ===> R       (N-No, R-Review, S-Save)

Process Report Type               ===> D       (D-Detailed, S-Summary)
```

Figure 52. Specify INSERT Process Parameters

The Extract File and Control File data set names are supplied and cannot be modified. For this scenario, specify B (Both) for **Processing Method to Use**. Press ENTER to execute.

If primary keys have not been defined for the destination tables, you are prompted to create them before the Insert Process is performed. If the Insert Process is performed online, status information is provided and upon completion, an UPDATE/INSERT Process Report is automatically displayed. The report documents the created objects and the updated data.

Summary

This section discussed how to modify an existing Access Definition to specify source data to be extracted and to extract that data. It also discussed selecting object definitions and storing them in an Extract File. The Point-and-Shoot facility was demonstrated as a convenient means of identifying the desired source data.

The Extract File was then used to update data in a set of corresponding destination tables created in Scenario 1. A new copy of the ITEMS table, ITEM2, was created. Also, primary keys, relationships, and indexes were created. Column Maps were used to specify a literal value and use DB2 special registers for the destination data.

The last part of this section demonstrated using the Insert Process, with Both (insert and update) specified as the processing method, to refresh an existing test database with source data.

Periodically, you may need to refresh the data in destination tables. For example, after testing your application against the test database, you can restore the original extracted data to the tables. Since the source data is stored in the Extract File, you can use the Extract File to update existing data in the destination tables as well as to insert the source data into newly created tables.

Chapter 5. Scenario 3 – Legacy Tables

Other scenarios demonstrate the use of Move with data in DB2 tables. Optim Legacy can provide the same functionality for IMS, VSAM, and sequential legacy data, if the legacy data has a relational structure. This functionality allows you to create federated test environments (including data from both DB2 and legacy sources) or to migrate legacy data to DB2 tables.

Legacy Data and Legacy Tables

To migrate legacy data, Move requires a Legacy Table. A Legacy Table is a user-defined object that describes legacy data. The Legacy Table imposes a table-like structure on the legacy data that allows Move to process the data as it would a DB2 table. Additionally, an IMS Legacy Table also references the DBD and PSB segment name to be associated with the Legacy Table.

- When you extract VSAM and sequential data, Optim Legacy requires a VSAM Legacy Table. You do not need to create any special Optim objects, other than those needed to extract DB2 data.
- However, when you extract IMS data, Optim Legacy requires an IMS Legacy Table and an Environment Definition. An Environment Definition allows you to specify the DBD, PSB and IMS Program Libraries and, if the data is online to a control region, the IMS System ID and AGN. Optionally, you can also create a Retrieval Definition, which specifies the default data source for Extract or Insert processing.

Data Sources

Before processing, each Legacy Table must reference a data source. You can optionally define a default data source for each Legacy Table:

- The data source for a VSAM Legacy Table is the associated VSAM or sequential data set. You can specify a default data source when you are creating the VSAM Legacy Table.
- The data source for an IMS Legacy Table includes the PSB, PCB, and associated IMS data set. The default data source information is specified in the IMS Retrieval Definition. You can create an IMS Retrieval Definition from an IMS Legacy Table or from the Choose a Definition Option menu.

You can override the default data source information for a Legacy Table, even if no data source information has been specified. An Access Definition allows you to override the default data source information for the Extract Process. A Table Map allows you to override the default data source information for the Insert process.

Legacy Relationships

While there are no explicit relationships between VSAM and sequential legacy records, data in a VSAM or sequential legacy file may be logically related to other legacy data or to data in DB2 tables. These relationships are generally managed by the application. You can replicate this relational structure by creating Optim relationships.

As an example of an application-managed relational structure in VSAM legacy data, assume a database includes the personnel and payroll records for employees. Payroll information is stored in a VSAM file, while the personnel information is in a sequential file. Each data set uses the employee Social Security number as a key. By defining Legacy Tables that describe the records in each file, and a relationship based on values in the Social Security keys, you can use Move to extract related data for a group of employees and use it to create test data. The extracted data and object definitions can be used to create new legacy files or the extracted data can be inserted into pre-existing DB2 tables that match the structure of the Legacy Tables and associated data.

Both explicit and logical relationships can exist between IMS segments. An explicit relationship between IMS segments exists if both segments are referenced in the same DBD, and a logical relationship between IMS segments exists if each segment is referenced in a unique DBD. If no explicit or logical relationship has been defined, you can create an Optim relationship to replicate this relational structure.

You can define an Optim relationship between a pair of Legacy Tables or between a Legacy Table and a DB2 table. The combination of Legacy Tables and Optim relationships enables Move to migrate sets of logically related legacy data.

Legacy Scenarios

In this section, two scenarios demonstrate how Optim Legacy handles Legacy data.

- In Scenario 3a, you will extract backorders from the sequential file, BKORDER, related items from the DB2 table, ITEMS, and related records from the VSAM files, VENDITEM and VENDOR. After extracting the data, you will use the Extract File to insert the data into empty DB2 tables.
- In Scenario 3b, you will extract employee records from the IMS segment DEPARTMENT and related IMS segments EMPLOYEE and POSITION, all in the DBD FOPDEPDB, as well as JOBCODE in the DBD FOPJOBDB. After extracting the data, you will use the Extract File to insert the data into empty IMS database data sets.

Scenario 3a: VSAM Legacy Tables

In this scenario, you will use sample data, distributed with Optim, to extract back orders from the sequential file BKORDER, related items from the DB2 table ITEMS, and related records from the VSAM files VENDITEM and VENDOR.

Before performing the Extract you must:

- Create the FOPDEMO.BKORDER Legacy Table from information in the COBOL copybook, FOP9BKOR.
- Create an Access Definition, referencing the FOPDEMO.ITEMS DB2 table, and the FOPDEMO.BKORDER, FOPDEMO.VENDITEM and FOPDEMO.VENDOR Legacy Tables.
- Associate each Legacy Table with the appropriate VSAM or sequential data set.
- Create Optim relationships to reflect application-managed relationships in which the DB2 table, ITEMS, is the parent of the Legacy Tables, BKORDER and VENDITEM. (See "Sample Database" on page 2.)

You will then create an Extract File containing the Legacy and DB2 tables referenced in the Access Definition.

Once the Extract File has been created, you will use it to create a test database. To test your application, you will insert the extracted backorder data into the empty DB2 table, FOPDEMO2.BKORDER, insert related items into the DB2 table, FOPDEMO2.ITEMS, and create VSAM files, FOPDEMO2.VENDITEM and FOPDEMO2.VENDOR.

Defining a VSAM Legacy Table

You can easily create a VSAM Legacy Table from one or more COBOL or PL/I copybooks:

- Name the new Legacy Table.
- Select the copybook library and member name used to define the Legacy Table.
- Edit column names, field details, etc. (Optional)
- Use criteria to match the Legacy Table to specific records in the source file. (Optional)
- Specify the default data source for the Legacy Table. (Optional)

Defining the Source

Just a few steps are performed to obtain the desired set of data. You will:

- Modify the Access Definition by specifying BKORDER as the Start Table for the Extract Process, and adding the other Legacy Tables to it.
- Assign data sources to the Legacy Tables.
- Define relationships to be traversed.
- Extract the data.

Defining the Destination

After you extract the data, it must be prepared for insertion into the DB2 destination table:

- Modify the INSERT Process Table Map.
- Associate Legacy Tables with destination data sets.
- Create the destination data sets using the Create Process.
- Insert the data into DB2 tables and data sets associated with Legacy Tables.

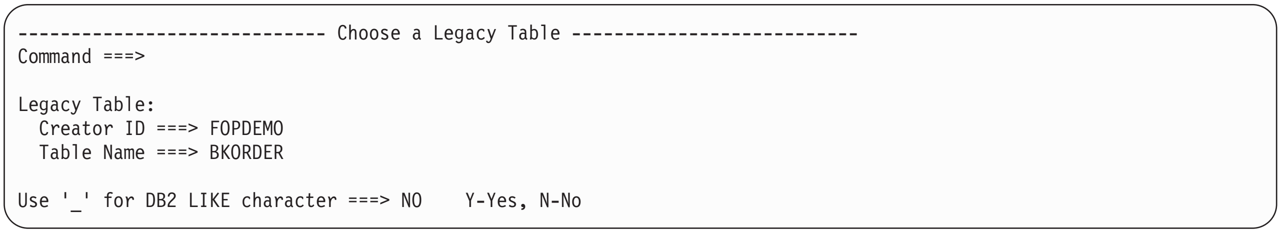
Defining a Legacy Table

Optim Legacy uses a Legacy Table to extract data from a legacy file. The Legacy Table is a user-defined object that reflects the structure of records in a file and can be used and reused to extract similarly structured records from any file with which it is associated. Since a legacy file may include several record structures, you can associate more than one Legacy Table with a file.

For this scenario, you will extract data from three legacy files: BKORDER, VENDITEM, and VENDOR. The sample data includes Legacy Tables for records in VENDITEM and VENDOR. However, you must create a Legacy Table for BKORDER.

Create a Legacy Table

To begin, select Option 6 DEFINITIONS from the Main Menu, then Option 6 LEGACY TABLES from the Choose a Definition Option menu to display the Choose a Legacy Table panel.



```
---------------------------- Choose a Legacy Table ---------------------------
Command ===>

Legacy Table:
  Creator ID ===> FOPDEMO
  Table Name ===> BKORDER

Use '_' for DB2 LIKE character ===> NO    Y-Yes, N-No
```

Figure 53. Choose a Legacy Table

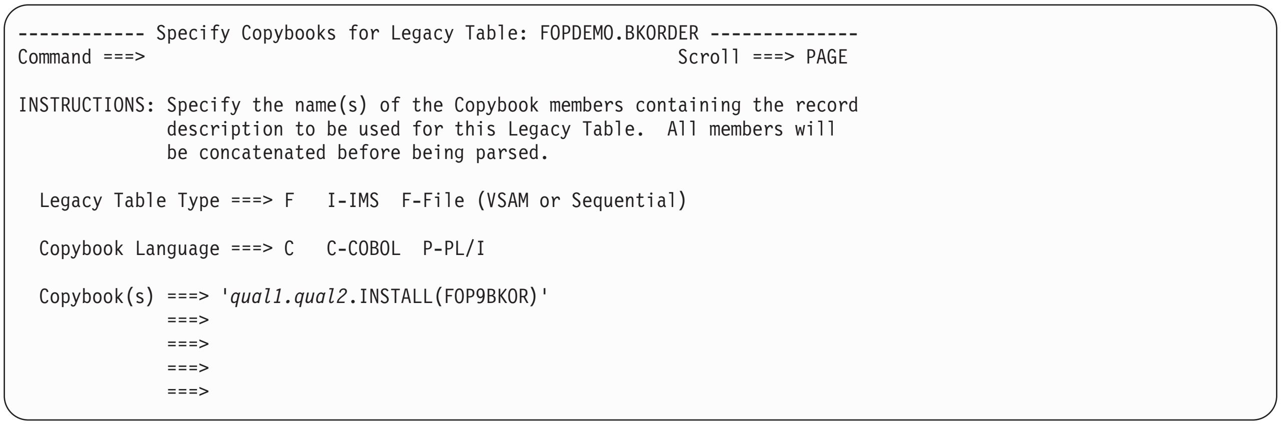
Legacy Table Name

A Legacy Table has a two-part name. You can:

- Review or edit an existing Legacy Table by entering the Creator ID and Table Name.
- Obtain a selection list by using DB2 LIKE syntax or leaving one part blank.
- Create a new Legacy Table by entering an unused name.

For this scenario, you will create the BKORDER Legacy Table using a copybook provided as part of the sample data. Enter FOPDEMO for **Creator ID** and BKORDER as the **Table Name** on the Choose a Legacy Table panel.

When you enter the name of a Legacy Table that does not exist, Move displays the Specify Copybooks for Legacy Table panel.



```
------------ Specify Copybooks for Legacy Table: FOPDEMO.BKORDER --------------
Command ===>                                                  Scroll ===> PAGE

INSTRUCTIONS: Specify the name(s) of the Copybook members containing the record
              description to be used for this Legacy Table.  All members will
              be concatenated before being parsed.

  Legacy Table Type ===> F   I-IMS  F-File (VSAM or Sequential)

  Copybook Language ===> C   C-COBOL  P-PL/I

  Copybook(s) ===> 'qual1.qual2.INSTALL(FOP9BKOR)'
              ===>
              ===>
              ===>
              ===>
```

Figure 54. Specify Copybooks for Legacy Table

Legacy Table Type

First, you need to specify the type of data to be used with this Legacy Table.

- An F entry means the data is stored in sequential or VSAM files.
- An I entry means the data is stored in an IMS database.

The FOPDEMO.BKORDER Legacy Table references sequential data, so enter F as the **Legacy Table Type**.

Copybook Language

Specify the language used to create the copybooks. Specify C (COBOL) or P (PL/I). For this scenario, enter C as the **Copybook Language**.

Copybook(s)

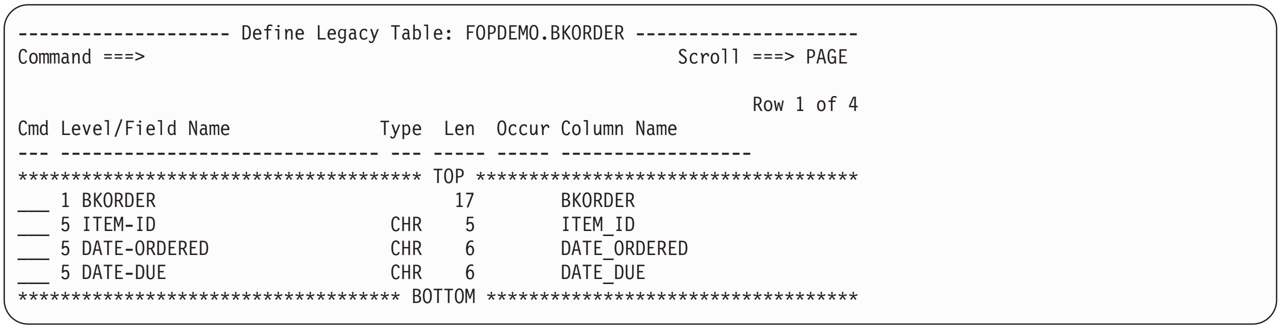
To create a Legacy Table manually, leave all Copybook entries blank. Move displays the Define Legacy Table panel in Edit mode, without populating any Field names. This allows you to manually create the Legacy Table.

You may enter as many as five copybook names on the Specify Copybooks for Legacy Table panel. If a desired copybook is not in the default copybook library, you must provide both the copybook data set and member names (e.g., '*qual1.qual2*.INSTALL(FOP9BKOR)'). You can also use a wildcard character with a fully-qualified data set name (e.g., 'FOPDEMO.RT.*') to display a more limited selection list.

Note:

- If the copybook member does not exist in the default or specified copybook library, an error message is displayed.
- If you do not use single quotes (' ') around the copybook library (and member) name, Move opens the Default Copybook Library, specified as a Legacy Option, in the Member Selection List panel. If there is no Default Copybook Library, the Data Set Prefix, specified as a User Option, is used as the prefix.

For this scenario, enter '*qual1.qual2*.INSTALL(FOP9BKOR)' as the copybook library and member name. Press ENTER to display the Legacy Table editor, populated with information from the copybook.



```
-------------------- Define Legacy Table: FOPDEMO.BKORDER ---------------------
Command ===>                                                 Scroll ===> PAGE

                                                                     Row 1 of 4
Cmd Level/Field Name              Type  Len  Occur Column Name
--- ----------------------------- --- ----- ----- ------------------
************************************* TOP *************************************
___ 1 BKORDER                            17       BKORDER
___ 5 ITEM-ID                     CHR     5       ITEM_ID
___ 5 DATE-ORDERED                CHR     6       DATE_ORDERED
___ 5 DATE-DUE                    CHR     6       DATE_DUE
************************************ BOTTOM ***********************************
```

Figure 55. Define Legacy Table

Edit Column Names

Move generates DB2-compatible column names from the names of fields in the copybook. The column names are used to identify and reference legacy fields and the data in them. Where possible, these generated column names are the same as those in the copybook. However, COBOL field names can be as many as 30 characters and PL/I field names can be 31 characters, while Legacy Table column names must be unique and cannot exceed 18 characters. Thus, Move may truncate or change COBOL and PL/I field names to generate column names. In addition, dashes in a field name are translated to underscore characters in the column name.

EDIT Command

From the Modify Legacy Table panel, you can edit the column names to make them more useful or descriptive. However, you must use the EDIT primary command to enable editing of other information displayed on the Modify Legacy Table panel. Alternatively, you can also use the Z (Zoom) line command to display and edit **Field Details**.

Other Commands

Use the FIND command to locate field names that include a specific character string. After using FIND, RFIND continues the search from the current cursor location.

The ONLY command is useful when you are working with a large Legacy Table and want to focus on a subset of fields. ONLY lists fields that satisfy command parameters (e.g., limiting the display to fields containing the string “-ID”). Use the SHOW command to redisplay all fields after using ONLY.

While editing the Legacy Table, you can use VERIFY to check for errors without saving the Legacy Table or terminating the session. SAVE verifies the edited Legacy Table. If errors are found, you are prompted to correct them before the Legacy Table can be saved.

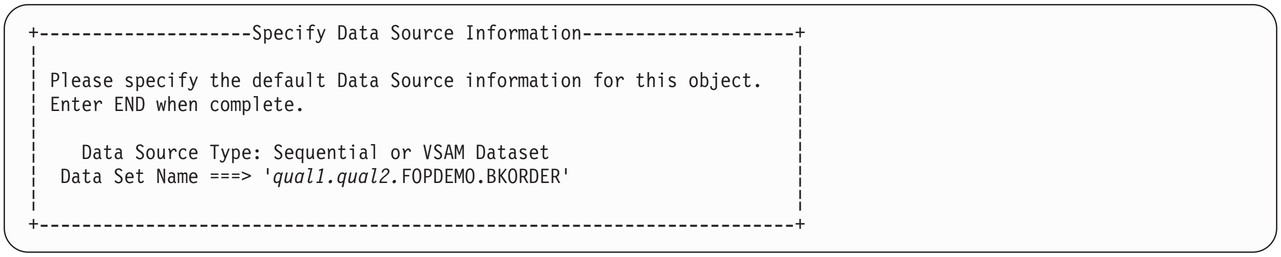
CRITERIA Command

If a legacy file contains records with different layouts, you can use criteria to match a Legacy Table with specific records in the source file. For example, a file might contain three types of records of the same length, but different formats. The applications that use this file recognize the record types by checking a code in the first field. You can use criteria to link a Legacy Table to the correct records.

The CRITERIA primary command invokes the Modify Criteria for Legacy Table panel. When defining or editing criteria, you can use the LIST COLUMNS command to display a selection list of column variables for the Legacy Table. Your selection(s) are inserted onto the Modify Criteria for Legacy Table panel. (For further information about this and other commands mentioned in this scenario, see the CRITERIA Command in the *Move User Manual* and Primary Commands in the *Command Reference Manual*.)

Default Data Source Association

To be used in Move processes, a Legacy Table must be linked with a file containing the data to be processed. You can establish a default data set name to be used unless overridden during processing. Use the DS primary command to display the Specify Data Source Information pop-up window.



```
+--------------------Specify Data Source Information--------------------+
¦                                                                       ¦
¦ Please specify the default Data Source information for this object.  ¦
¦ Enter END when complete.                                              ¦
¦                                                                       ¦
¦    Data Source Type: Sequential or VSAM Dataset                       ¦
¦  Data Set Name ===> 'qual1.qual2.FOPDEMO.BKORDER'                     ¦
¦                                                                       ¦
+-----------------------------------------------------------------------+
```

Figure 56. Specify Data Source Information

To save time or if you do not know the exact name of the data source, you can use wildcards to obtain a selection list. Using % as the last element of the name (e.g., 'FOPRT.PROD.%') will provide a selection list of all data sets beginning with FOPRT.PROD. The selected data source name is inserted at the **Data Set Name** prompt.

The data source for the BKORDER Legacy Table was created when Move was installed at your site. Check with your site administrator for information about the location of the sample data and enter the fully qualified name of that data set, enclosed in quotes. After you use END, the message “DATA SOURCE UPDATED” is displayed on the Modify Legacy Table panel. Use END again and the message “LEGACY TABLE MODIFIED” is displayed on the Choose a Legacy Table panel.

This completes editing the BKORDER Legacy Table for this scenario. Use END twice more to return to the Main Menu.

Defining the Source

After Legacy Tables describing records in the legacy files are created, you can define the source data for extraction using the normal steps: create or modify the Access Definition, establish the path to follow when extracting data, select object definitions to be extracted, and perform the extract.

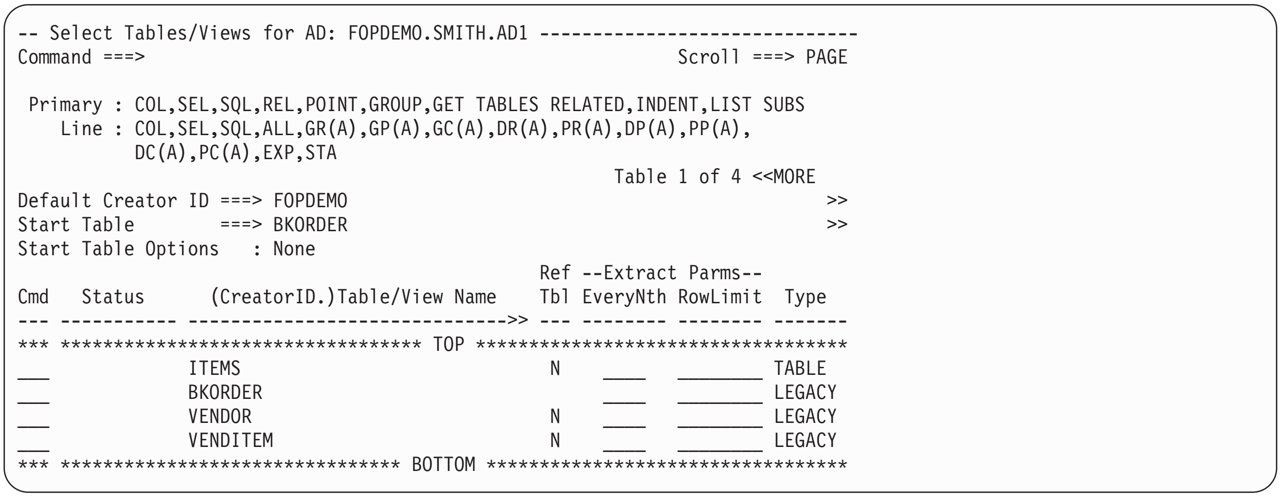
From the Main Menu, select Option 7 MIGRATION. On the Data Migration menu, select Option 1 EXTRACT.

The Extract Process for this scenario uses the Access Definition from previous scenarios. To edit the Access Definition, confirm that FOPDEMO.SMITH.AD1 is specified and select Option 1 TABLES from the EXTRACT Process menu. You will edit the Access Definition, retaining the DB2 table FOPDEMO.ITEMS on the Table List and adding the three Legacy Tables BKORDER, VENDOR, and VENDITEM.

Edit Table List

Overtype the names of CUSTOMERS, ORDERS and DETAILS tables with the names of the Legacy Tables BKORDER, VENDOR, and VENDITEM.

Because you are extracting a set of backordered items, the Extract Process must begin with the records in BKORDER. Replace ITEMS with BKORDER at the **Start Table** prompt. Since all BKORDER records are copied to the Extract File, you do not need to specify any selection criteria. The completed Access Definition is displayed as follows:

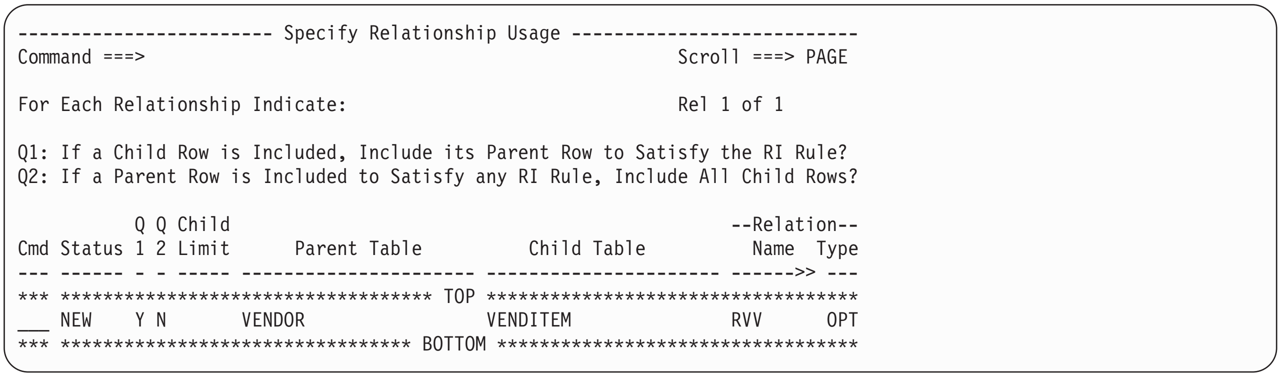


```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> BKORDER                                             >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ------------------------------>> --- -------- -------- -------
*** ********************************* TOP ***********************************
___             ITEMS                             N    ____    ________ TABLE
___             BKORDER                                ____    ________ LEGACY
___             VENDOR                            N    ____    ________ LEGACY
___             VENDITEM                          N    ____    ________ LEGACY
*** ******************************** BOTTOM *********************************
```

Figure 57. Completed Access Definition

Paths or Relationships

After completing the Table List, use the REL primary command to display the Specify Relationship Usage panel.



```
----------------------- Specify Relationship Usage ---------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 1

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                               --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- --------------------- --------------------- ------>> ---
*** ********************************** TOP **********************************
___ NEW    Y N      VENDOR               VENDITEM              RVV      OPT
*** ******************************** BOTTOM *********************************
```

Figure 58. Specify Relationship Usage

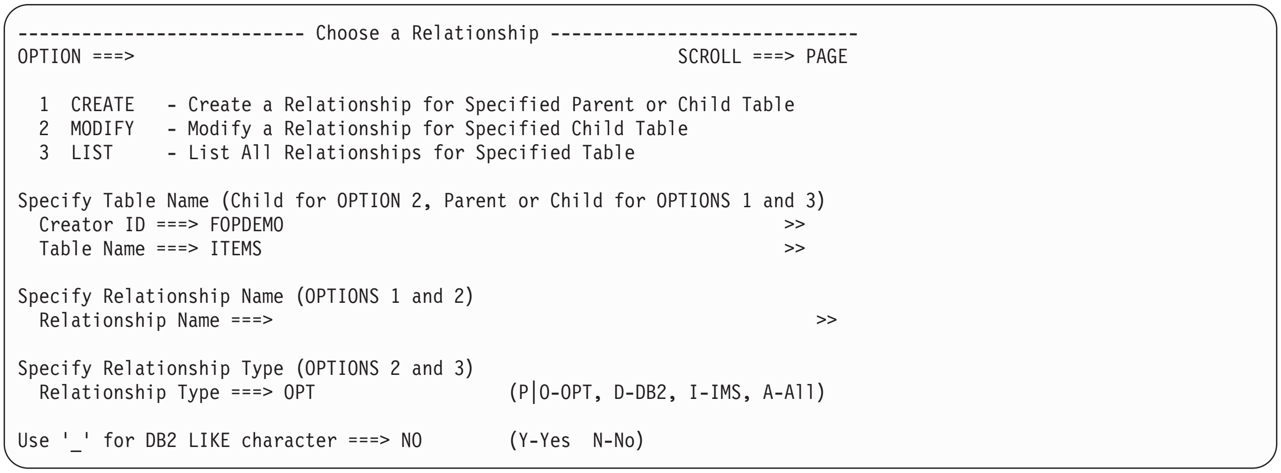
The Specify Relationship Usage panel lists an Optim relationship between VENDITEM and VENDOR. This relationship is included with the sample data. However, to extract the ITEMS rows related to the records in BKORDER and the VENDITEM records related to the selected ITEMS rows, you must create the needed relationships. When defining Optim relationships:

- Primary keys and foreign keys are not required.
- Corresponding columns need not be identical, but must be compatible. (Compatible column types are described in the *Common Elements Manual* under Compatibility Rules.)
- Expressions that may include string literals, numeric constants, concatenation, and substrings can be used.

You can define an Optim Relationship between Legacy Tables, between DB2 tables, or between a DB2 table and a Legacy Table. Move stores all Legacy Table definitions and Optim relationships in the Optim Directory.

First, you will define the relationship between the DB2 table, ITEMS, and the Legacy Table, BKORDER. To do this, use the CREATE REL primary command to display the Choose a Relationship panel.

You must provide the name of one table in the relationship. You can use DB2 LIKE syntax or leave one part of the table name blank in order to obtain a selection list. Since all relationships for this scenario involve tables with names prefixed with FOPDEMO, type FOPDEMO at **Creator ID**. Also, because ITEMS is involved in both relationships, type ITEMS at **Table Name**. Type OPT as the **Relationship Type**. The completed panel is shown in the following figure:

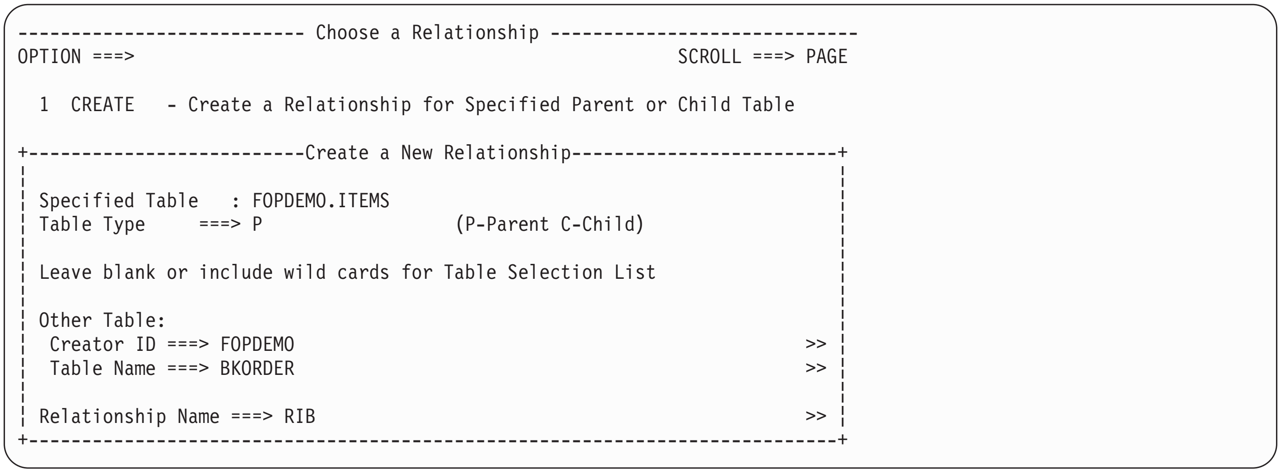


```
--------------------------- Choose a Relationship -----------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  CREATE   - Create a Relationship for Specified Parent or Child Table
  2  MODIFY   - Modify a Relationship for Specified Child Table
  3  LIST     - List All Relationships for Specified Table

Specify Table Name (Child for OPTION 2, Parent or Child for OPTIONS 1 and 3)
  Creator ID ===> FOPDEMO                                              >>
  Table Name ===> ITEMS                                                >>

Specify Relationship Name (OPTIONS 1 and 2)
  Relationship Name ===>                                                  >>

Specify Relationship Type (OPTIONS 2 and 3)
  Relationship Type ===> OPT                  (P|O-OPT, D-DB2, I-IMS, A-All)

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

Figure 59. Choose a Relationship

Define Relationship RIB

Select Option 1 CREATE and press ENTER to display the Create a New Relationship pop-up window.



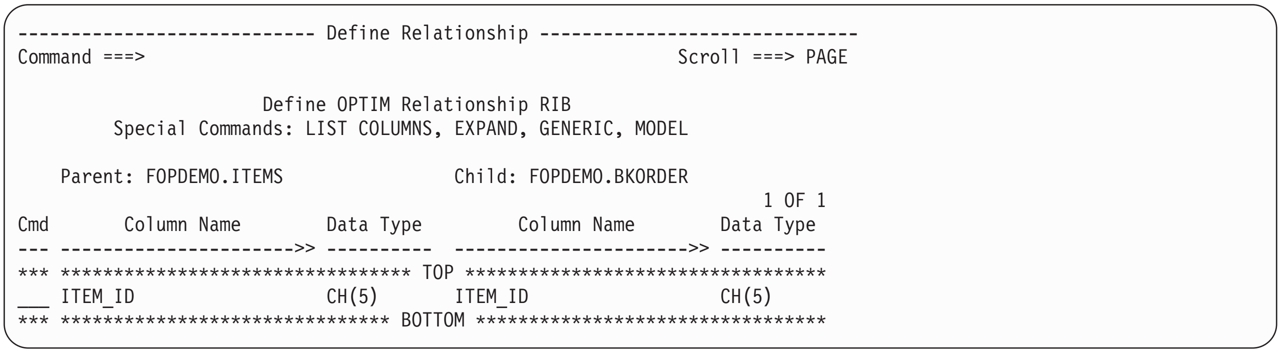
```
--------------------------- Choose a Relationship -----------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  CREATE   - Create a Relationship for Specified Parent or Child Table

+--------------------------Create a New Relationship------------------------+
|                                                                            |
| Specified Table   : FOPDEMO.ITEMS                                          |
| Table Type     ===> P                 (P-Parent C-Child)                   |
|                                                                            |
| Leave blank or include wild cards for Table Selection List                 |
|                                                                            |
| Other Table:                                                               |
|  Creator ID ===> FOPDEMO                                                >> |
|  Table Name ===> BKORDER                                                >> |
|                                                                            |
| Relationship Name ===> RIB                                              >> |
+----------------------------------------------------------------------------+
```

Figure 60. Create a New Relationship

Type P at **Table Type** to designate the ITEMS table as the parent. Move populates **Creator ID** for the second table in the relationship with the Creator ID for ITEMS. Enter the name of the child table, BKORDER, at **Table Name** and type RIB (i.e., Relationship ITEMS to BKORDER) as the **Relationship Name**.

Note: When creating a relationship, you can provide a name on the Choose a Relationship panel or on the Create a New Relationship pop-up window.

Press ENTER to display the Define Relationship panel.



```
---------------------------- Define Relationship ------------------------------
Command ===>                                                  Scroll ===> PAGE

                       Define OPTIM Relationship RIB
         Special Commands: LIST COLUMNS, EXPAND, GENERIC, MODEL

    Parent: FOPDEMO.ITEMS                Child: FOPDEMO.BKORDER
                                                                      1 OF 1
Cmd       Column Name        Data Type         Column Name        Data Type
--- ---------------------->> ----------  ---------------------->> ----------
*** ********************************* TOP *********************************
___ ITEM_ID                  CH(5)       ITEM_ID                  CH(5)
*** ******************************* BOTTOM ********************************
```

Figure 61. Define Relationship

If the parent table has a primary key, as ITEMS does, the names of primary key columns and data types are displayed on the parent side of the panel. The names of any columns in the child table that have the same names and compatible data types are displayed on the child side of the panel. ITEM_ID is the primary key for ITEMS. Since ITEM_ID in the BKORDER Legacy Table is the same type and length, Move populates both sides of the Define Relationship panel for the relationship RIB.

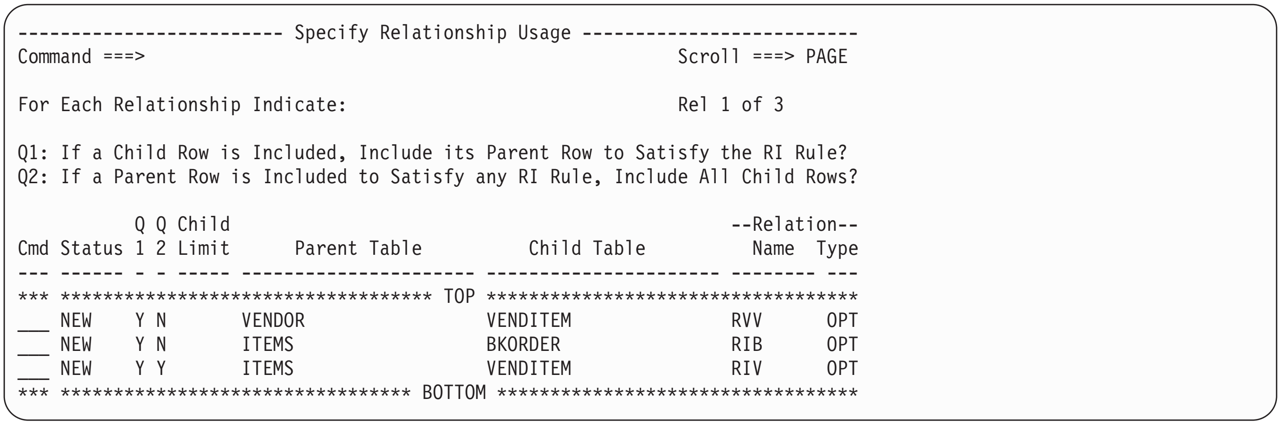
Use END to save the relationship and return to the Choose a Relationship panel.

Define Relationship RIV

For the second relationship needed in this scenario, retain ITEMS as **Table Name** and OPT as **Relationship Type**. Enter RIV at **Relationship Name** and again select Option 1 CREATE. Press ENTER. On the Create a New Relationship popup window, type VENDITEM at **Table Name**. Press ENTER.

As for relationship RIB, Move populates the parent side of the Define Relationship panel with the name of the primary key for the parent ITEMS table. When names of child table columns do not match the names of primary key columns in the parent table, Move checks the child table for a single column with identical attributes, regardless of name. If only one column has identical attributes, the name is inserted. Because VEND_ITEM_ID matches the data type and length of ITEM_ID, Move inserts VEND_ITEM_ID as the child **Column Name**.

Use END twice to save the relationship and return to the Specify Relationship Usage panel.



```
------------------------- Specify Relationship Usage --------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                               --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- --------------------- --------------------- -------- ---
*** ************************************ TOP **********************************
___ NEW    Y N       VENDOR                 VENDITEM               RVV      OPT
___ NEW    Y N       ITEMS                  BKORDER                RIB      OPT
___ NEW    Y Y       ITEMS                  VENDITEM               RIV      OPT
*** ******************************** BOTTOM **********************************
```

Figure 62. Specify Relationship Usage

Specifying Relationship Usage

The relationships needed to extract the full set of data are now present. The listed relationships must be traversed to obtain related data from all tables referenced in the Access Definition.

The first time a relationship is listed, it is assigned the NEW status, as shown in Figure 62 on page 63. A relationship in NEW status will be used in the Extract Process, if the **Use NEW Relationships** parameter for Access Definitions is YES. (For further information, see Access Definition Parameters in the *Common Elements Manual*.)

YES is the default setting for the **Use NEW Relationships** parameter. Because this setting may have been changed in training or other use of the product, use the S line command to select the relationships RIB, RIV, and RVV. If other relationships are listed, use the U line command to unselect them.

Q1 and Q2

The **Q1** and **Q2** options determine how relationships are traversed. When **Q1** is Yes, Move follows a relationship from child to parent and a parent row is extracted for every child row to ensure the relational integrity of the extracted data. (**Q1** and **Q2** are explained in greater detail in the previous scenarios. Additional information is available in the *Common Elements Manual* under Use Relationships.)

The second option, **Q2**, determines whether additional child rows are extracted if a parent row was extracted in a traversal from child to parent. If a parent row is extracted to ensure relational integrity (i.e., because a child row was extracted), any additional children of the parent row are also extracted when **Q2** is Yes.

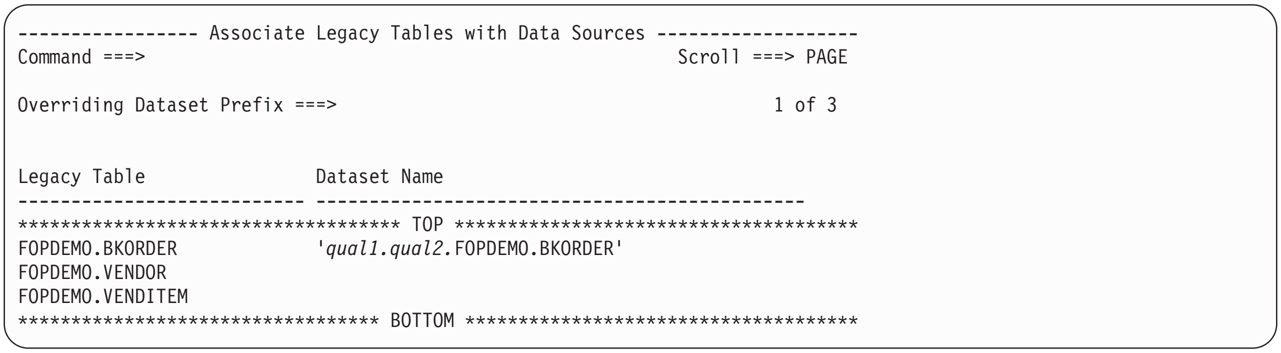
For this scenario:

- Using the relationship RIB, all Legacy BKORDER records are extracted with any related rows from the parent ITEMS table. **Q1** must be Y to traverse the relationship RIB in this direction.
- Using the relationship RIV, VENDITEM records that are children of the extracted ITEMS rows are then extracted. **Q2** must be Y to traverse the relationship RIV.
- Finally, using the relationship RVV, VENDOR records that are parents of extracted VENDITEM records are extracted. **Q1** must be Y to traverse the relationship RVV in this direction.

Note: You can use the SHOW STEPS command to review the extract path (see "SHOW STEPS Command" on page 65). However, if the Access Definition references Legacy Tables, you must assign data sources to the Legacy Tables before you can use the SHOW STEPS command.

Associate Legacy Tables with Data Sources

Use END to close the Access Definition editor. Before returning to the EXTRACT Process menu, Move prompts for information about the sources for the data described by the Legacy Tables.



```
----------------- Associate Legacy Tables with Data Sources -------------------
Command ===>                                                 Scroll ===> PAGE

Overriding Dataset Prefix ===>                                        1 of 3


Legacy Table               Dataset Name
-------------------------- ---------------------------------------------
*********************************** TOP **************************************
FOPDEMO.BKORDER            'qual1.qual2.FOPDEMO.BKORDER'
FOPDEMO.VENDOR
FOPDEMO.VENDITEM
********************************* BOTTOM *************************************
```

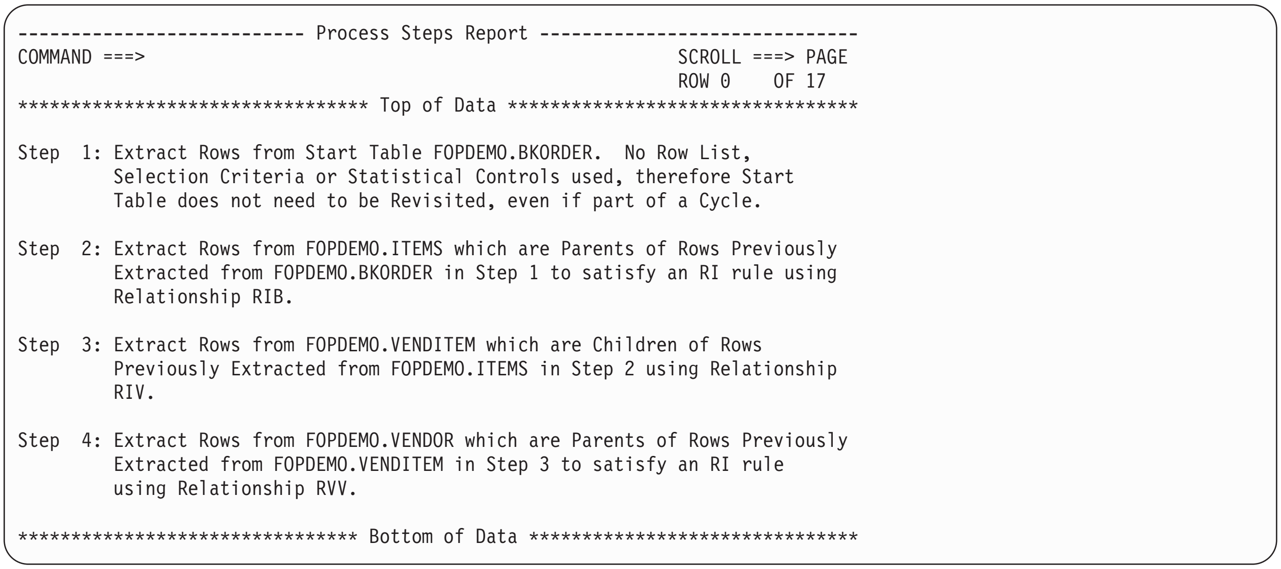
Figure 63. Associate Legacy Tables with Data Sources

Earlier in this scenario, you associated a data set as the default data source for the BKORDER Legacy Table. As shown in the figure, Move inserts any default names on the Associate Legacy Tables with Data Sources panel. You can edit the default name to specify a different data source for a process, if desired. For this scenario, however, you will use the default.

Since no default has been specified for the remaining Legacy Tables, they must also be associated with source data sets. To insert the names, you can type the complete data set name, or use the "=" shortcut to duplicate the data set name for the preceding Legacy Table and edit the duplicated data set name. Using either method, enter the names of the VENDOR and VENDITEM data sets. Use END to return to the EXTRACT Process menu.

SHOW STEPS Command

After assigning data sources to the Legacy Tables, select Option 2 PATHS on the EXTRACT Process menu to redisplay the Specify Relationship Usage panel. You can use the SHOW STEPS command to review the extract path.



```
--------------------------- Process Steps Report ------------------------------
COMMAND ===>                                                SCROLL ===> PAGE
                                                            ROW 0    OF 17
********************************* Top of Data *********************************

Step  1: Extract Rows from Start Table FOPDEMO.BKORDER.  No Row List,
         Selection Criteria or Statistical Controls used, therefore Start
         Table does not need to be Revisited, even if part of a Cycle.

Step  2: Extract Rows from FOPDEMO.ITEMS which are Parents of Rows Previously
         Extracted from FOPDEMO.BKORDER in Step 1 to satisfy an RI rule using
         Relationship RIB.

Step  3: Extract Rows from FOPDEMO.VENDITEM which are Children of Rows
         Previously Extracted from FOPDEMO.ITEMS in Step 2 using Relationship
         RIV.

Step  4: Extract Rows from FOPDEMO.VENDOR which are Parents of Rows Previously
         Extracted from FOPDEMO.VENDITEM in Step 3 to satisfy an RI rule
         using Relationship RVV.

******************************** Bottom of Data *******************************
```

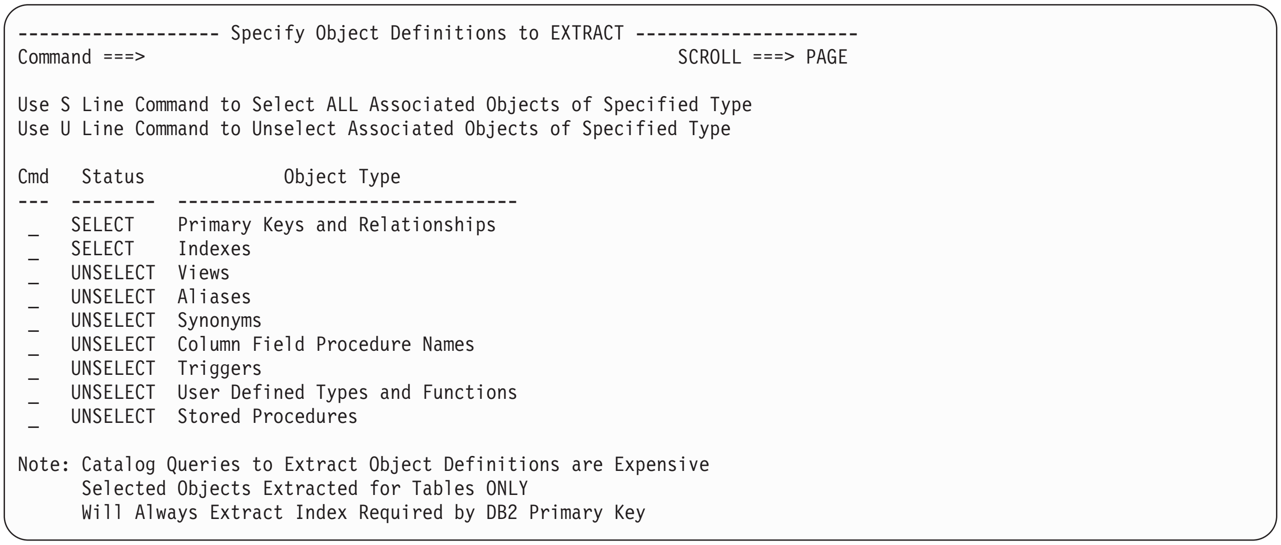
Figure 64. Process Steps Report

After reviewing the steps to be used in the process, use END twice to return to the EXTRACT Process menu.

Object Definitions

Although Move always extracts DB2 and Legacy Table definitions, and subordinate column definitions with the data, you will also need the Optim relationships used to extract the data in this scenario. To extract the relationships, select Option 3 OBJECTS on the EXTRACT Process menu. Move displays the Specify Object Definitions to EXTRACT panel.

By default, primary keys, relationships, and indexes are selected when the panel is displayed. The **Status** for other object definitions is UNSELECT.



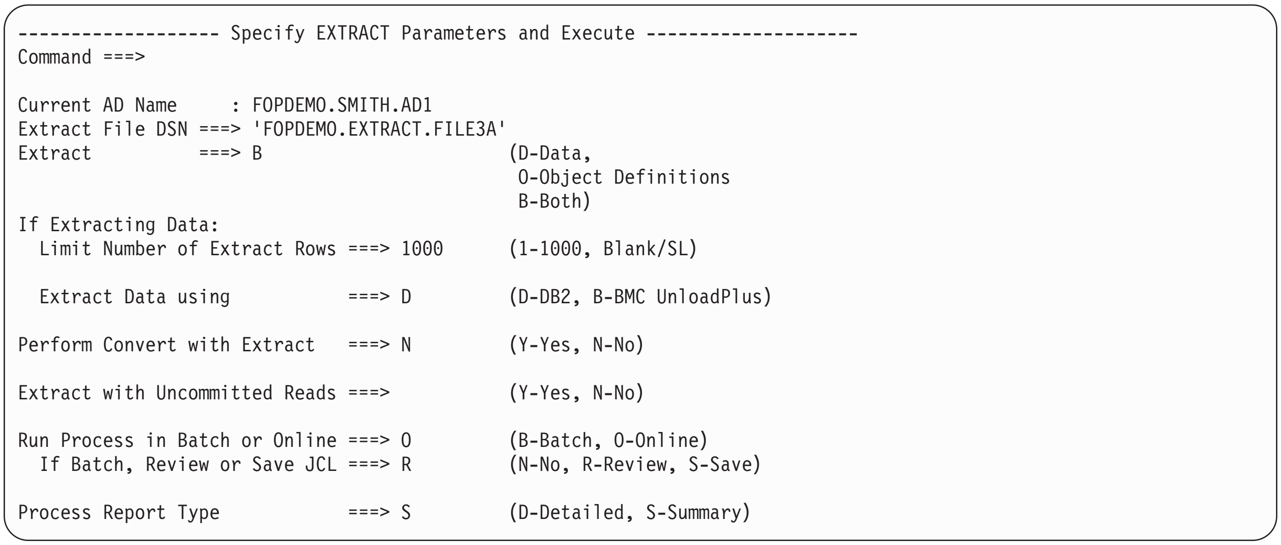
```
------------------- Specify Object Definitions to EXTRACT ---------------------
Command ===>                                                  SCROLL ===> PAGE

Use S Line Command to Select ALL Associated Objects of Specified Type
Use U Line Command to Unselect Associated Objects of Specified Type

Cmd   Status            Object Type
---  --------  -------------------------------
 _   SELECT    Primary Keys and Relationships
 _   SELECT    Indexes
 _   UNSELECT  Views
 _   UNSELECT  Aliases
 _   UNSELECT  Synonyms
 _   UNSELECT  Column Field Procedure Names
 _   UNSELECT  Triggers
 _   UNSELECT  User Defined Types and Functions
 _   UNSELECT  Stored Procedures

Note: Catalog Queries to Extract Object Definitions are Expensive
      Selected Objects Extracted for Tables ONLY
      Will Always Extract Index Required by DB2 Primary Key
```

Figure 65. Specify Object Definitions

Although indexes are not required for this scenario, use the default settings for Primary Keys, Relationships, and Indexes. Use END to return to the EXTRACT Process menu.

Perform the Extract

Select Option 4 PERFORM on the EXTRACT Process menu to display the Specify EXTRACT Parameters and Execute panel. To perform the extract, you must provide the Extract File data set name and indicate whether to extract data, object definitions, or both.

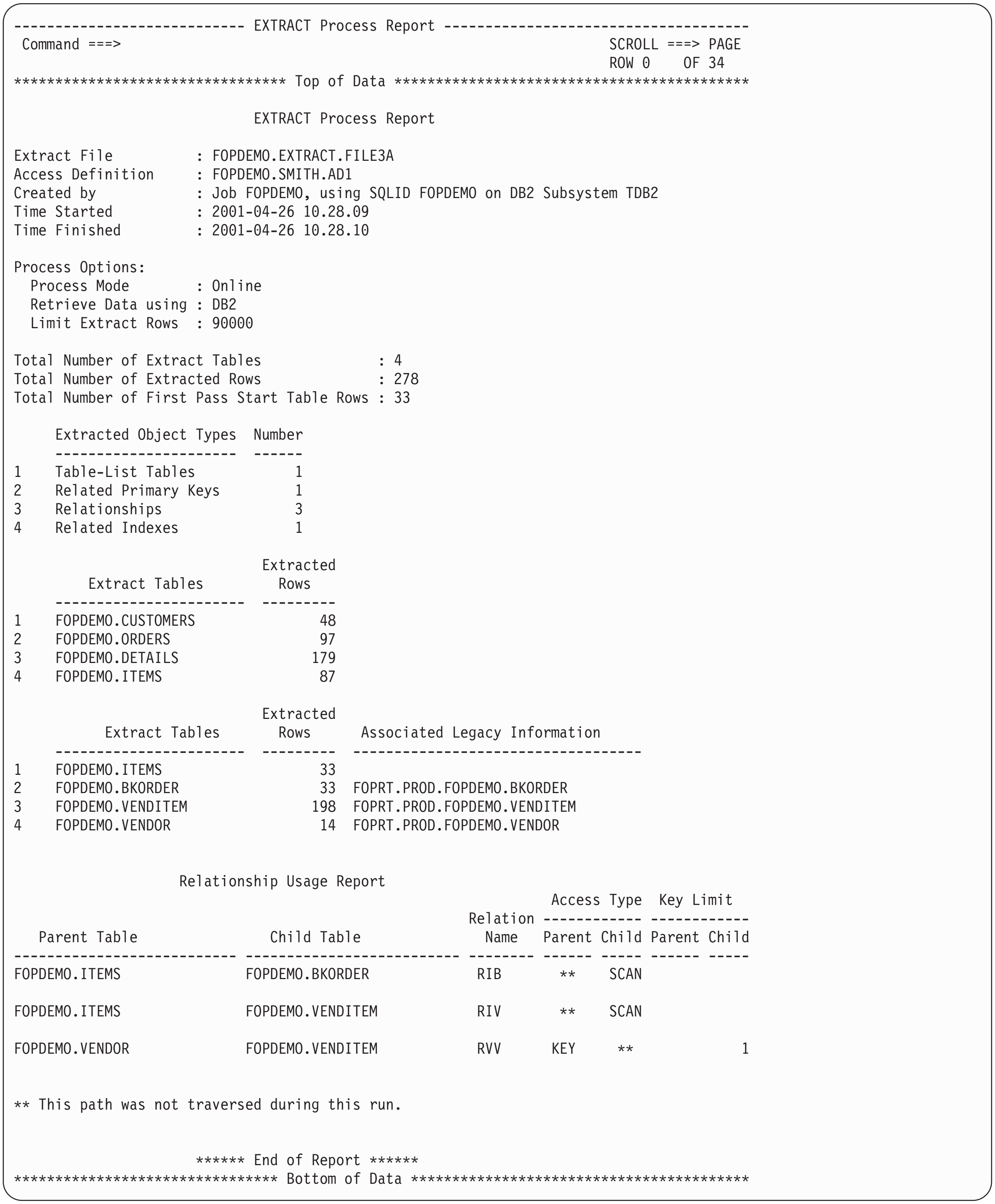


```
------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE3A'
Extract          ===> B                      (D-Data,
                                              O-Object Definitions
                                              B-Both)
If Extracting Data:
  Limit Number of Extract Rows ===> 1000      (1-1000, Blank/SL)

  Extract Data using           ===> D         (D-DB2, B-BMC UnloadPlus)

Perform Convert with Extract   ===> N         (Y-Yes, N-No)

Extract with Uncommitted Reads ===>           (Y-Yes, N-No)

Run Process in Batch or Online ===> O         (B-Batch, O-Online)
  If Batch, Review or Save JCL ===> R         (N-No, R-Review, S-Save)

Process Report Type            ===> S         (D-Detailed, S-Summary)
```

Figure 66. Specify EXTRACT Parameters and Execute

The name of the Extract File you specified for the previous extract has been profiled and is displayed in **Extract File DSN**. For this scenario, change the name to FOPDEMO.EXTRACT.FILE3A. Both data and object definitions are extracted. Confirm that B is specified at the **Extract** prompt. (Other details of this panel are discussed in “Perform the Extract” on page 19.)

Press ENTER to begin the extract. If the Extract File does not exist, Move prompts for the necessary information and allocates the file. Informational messages are displayed in pop-up windows as the extract progresses. When finished, Move displays the EXTRACT Process Report.



```
---------------------------- EXTRACT Process Report -------------------------------------
 Command ===>                                                           SCROLL ===> PAGE
                                                                        ROW 0    OF 34
********************************* Top of Data *******************************************

                             EXTRACT Process Report

Extract File          : FOPDEMO.EXTRACT.FILE3A
Access Definition     : FOPDEMO.SMITH.AD1
Created by            : Job FOPDEMO, using SQLID FOPDEMO on DB2 Subsystem TDB2
Time Started          : 2001-04-26 10.28.09
Time Finished         : 2001-04-26 10.28.10

Process Options:
  Process Mode        : Online
  Retrieve Data using : DB2
  Limit Extract Rows  : 90000

Total Number of Extract Tables              : 4
Total Number of Extracted Rows              : 278
Total Number of First Pass Start Table Rows : 33

     Extracted Object Types  Number
     ----------------------  ------
1    Table-List Tables           1
2    Related Primary Keys        1
3    Relationships               3
4    Related Indexes             1

                              Extracted
         Extract Tables         Rows
     -----------------------  ---------
1    FOPDEMO.CUSTOMERS               48
2    FOPDEMO.ORDERS                  97
3    FOPDEMO.DETAILS                179
4    FOPDEMO.ITEMS                   87

                              Extracted
           Extract Tables       Rows      Associated Legacy Information
     -----------------------  ---------  -----------------------------------
1    FOPDEMO.ITEMS                   33
2    FOPDEMO.BKORDER                 33  FOPRT.PROD.FOPDEMO.BKORDER
3    FOPDEMO.VENDITEM               198  FOPRT.PROD.FOPDEMO.VENDITEM
4    FOPDEMO.VENDOR                  14  FOPRT.PROD.FOPDEMO.VENDOR


                    Relationship Usage Report
                                                                 Access Type  Key Limit
                                                       Relation ------------ ------------
   Parent Table               Child Table                Name   Parent Child Parent Child
--------------------------- -------------------------- -------- ------ ----- ------ -----
FOPDEMO.ITEMS               FOPDEMO.BKORDER             RIB       **    SCAN

FOPDEMO.ITEMS               FOPDEMO.VENDITEM            RIV       **    SCAN

FOPDEMO.VENDOR              FOPDEMO.VENDITEM            RVV      KEY     **            1


** This path was not traversed during this run.


                      ****** End of Report ******
******************************** Bottom of Data *****************************************
```

Figure 67. EXTRACT Process Report

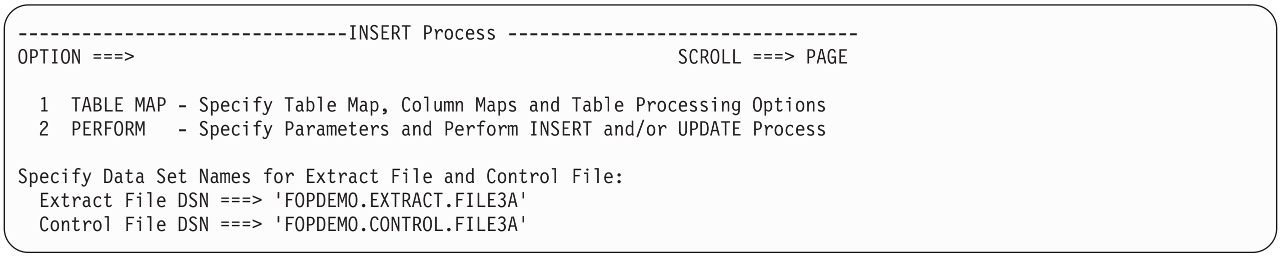
Note that when Legacy Tables are involved in the Extract Process, the associated Legacy data sets are listed in the EXTRACT Process Report. After reviewing the report, use END twice to return to the Data Migration menu. Move prompts you to save the edited Access Definition. Press ENTER to save the Access Definition under its current name and return to the Data Migration menu.

The Extract Process is complete.

Defining the Destination

With the conclusion of the Extract Process, all the data and necessary object definitions are copied to the Extract File, which is the source of data for the Insert Process.

Select Option 2 INSERT on the Data Migration menu to specify the destination. (Details of this panel are discussed in “Defining the Destination” on page 22.)



```
------------------------------INSERT Process --------------------------------
OPTION ===>                                                 SCROLL ===> PAGE

  1  TABLE MAP - Specify Table Map, Column Maps and Table Processing Options
  2  PERFORM   - Specify Parameters and Perform INSERT and/or UPDATE Process

Specify Data Set Names for Extract File and Control File:
  Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE3A'
  Control File DSN ===> 'FOPDEMO.CONTROL.FILE3A'
```

Figure 68. Scenario 3a – INSERT Process Menu

To create the data you need to test your application in this scenario, you will insert the extracted backorder data into the empty DB2 table, FOPDEMO2.BKORDER, insert related items rows into the DB2 table, FOPDEMO2.ITEMS, and create VSAM files, FOPDEMO2.VENDITEM and FOPDEMO2.VENDOR.

As in previous scenarios, the destination table names match the source table names and are uniquely identified by the Creator ID. The primary keys and relationships and the indexes from the production database are copied into the test database.

Move provides the name of the current or most recent Extract File automatically. For this scenario, FOPDEMO.EXTRACT.FILE3A is the name of the Extract File used to create the test database. If this name is not in **Extract File DSN**, enter it now. The name of the current or most recent Control File is also provided automatically. For this scenario, type FOPDEMO.CONTROL.FILE3A at the **Control File DSN** prompt.

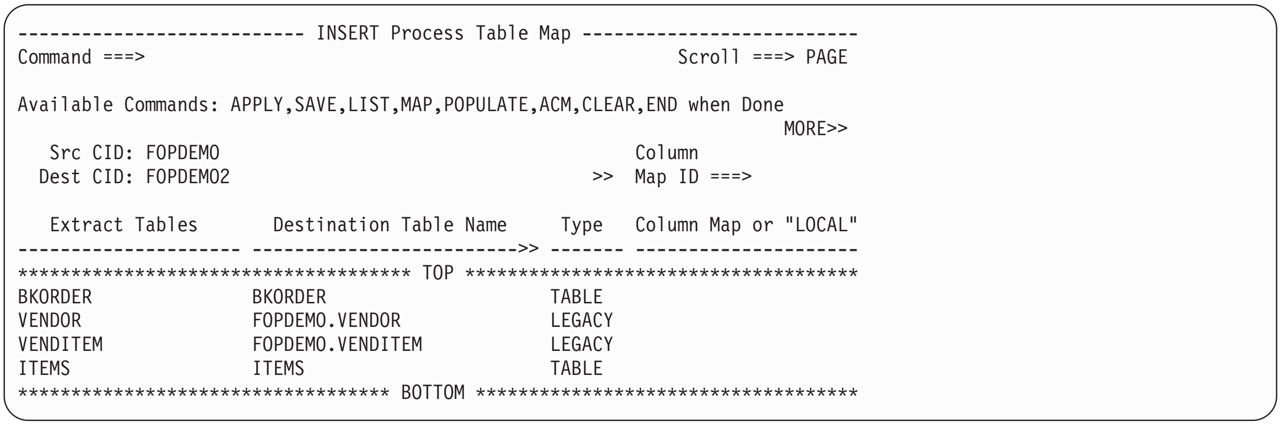
Specify the Tables

Select Option 1 TABLE MAP from the INSERT Process menu to display the INSERT Process Table Map panel. (A more detailed discussion of this panel is provided in “Specify the Tables” on page 23.)

Modify Table Map

Enter the **Dest CID**, FOPDEMO2. The destination **Type** for FOPDEMO2.ITEMS and FOPDEMO2.BKORDER is TABLE because the sample data includes DB2 tables with this Creator ID and name.

The **Type** for VENDOR and VENDITEM is displayed as UNKNOWN because no Legacy or DB2 tables named FOPDEMO2.VENDOR or FOPDEMO2.VENDITEM exist. To create new VENDOR and VENDITEM data sets, you will use the same Legacy Tables used in the Extract. Insert “FOPDEMO.” ahead of VENDOR and VENDITEM in **Destination Table Name**. Press ENTER to see that Move updates the **Type** from UNKNOWN to LEGACY.



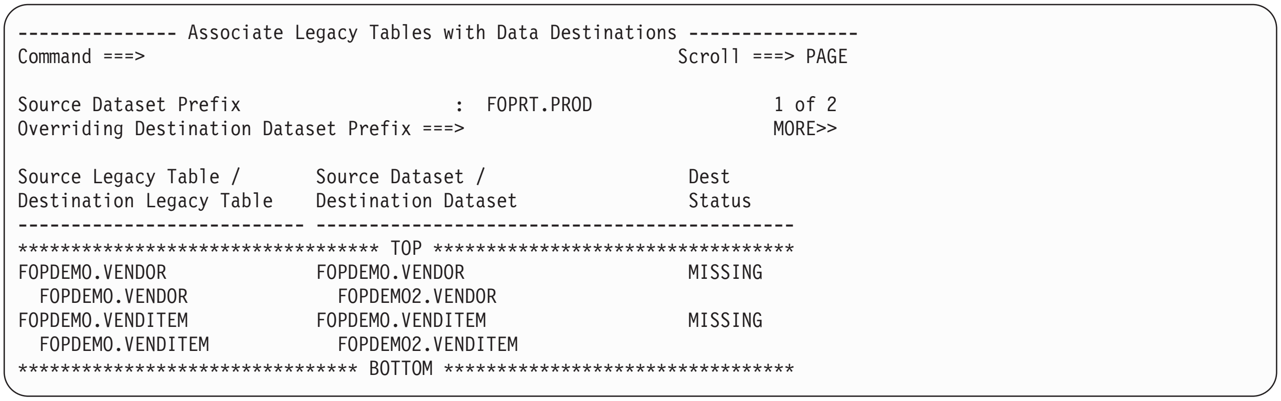
```
--------------------------- INSERT Process Table Map --------------------------
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY,SAVE,LIST,MAP,POPULATE,ACM,CLEAR,END when Done
                                                                        MORE>>
   Src CID: FOPDEMO                                       Column
  Dest CID: FOPDEMO2                                  >>  Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
--------------------- ------------------------->> ------- ---------------------
************************************* TOP *************************************
BKORDER               BKORDER                     TABLE
VENDOR                FOPDEMO.VENDOR              LEGACY
VENDITEM              FOPDEMO.VENDITEM            LEGACY
ITEMS                 ITEMS                       TABLE
*********************************** BOTTOM ************************************
```

Figure 69. INSERT Process Table Map

If you leave **Destination Table Name** blank for a table, it is not processed in the Insert. For example, to create new legacy files for VENDOR and VENDITEM without inserting data into the BKORDER and ITEMS tables, you would remove BKORDER and ITEMS from **Destination Table Name**. For this scenario, however, you will:

- Create new legacy files associated with FOPDEMO.VENDOR and FOPDEMO.VENDITEM.
- Insert data from the legacy file FOPDEMO.BKORDER into the DB2 table FOPDEMO2.BKORDER.
- Insert data from the DB2 table FOPDEMO.ITEMS into the DB2 table FOPDEMO2.ITEMS.

When you use END, the Associate Legacy Tables with Data Destinations panel is displayed. On this panel, Move inserts the names of any default data sources for listed Legacy Tables as the **Destination Dataset**. Since there are no default data sets for the FOPDEMO.VENDOR and FOPDEMO.VENDITEM Legacy Tables, **Destination Dataset** is initially blank.



```
--------------- Associate Legacy Tables with Data Destinations ----------------
Command ===>                                                  Scroll ===> PAGE

Source Dataset Prefix                    :  FOPRT.PROD                 1 of 2
Overriding Destination Dataset Prefix ===>                             MORE>>

Source Legacy Table /       Source Dataset /                   Dest
Destination Legacy Table    Destination Dataset                Status
--------------------------- ---------------------------------------------
********************************** TOP **********************************
FOPDEMO.VENDOR              FOPDEMO.VENDOR                     MISSING
  FOPDEMO.VENDOR              FOPDEMO2.VENDOR
FOPDEMO.VENDITEM            FOPDEMO.VENDITEM                   MISSING
  FOPDEMO.VENDITEM            FOPDEMO2.VENDITEM
******************************** BOTTOM *********************************
```

Figure 70. Associate Legacy Tables with Data Destinations

For this scenario, you will create the data sets FOPDEMO2.VENDOR and FOPDEMO2.VENDITEM to go with the DB2 tables FOPDEMO2.BKORDER and FOPDEMO2.ITEMS.

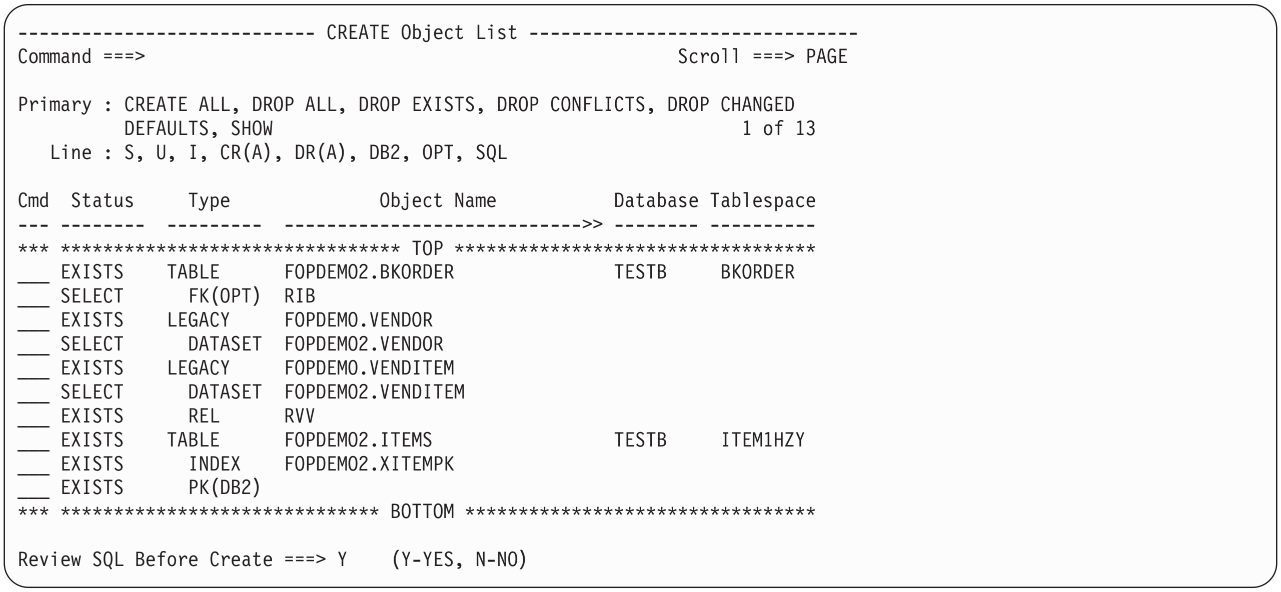
Since the new **Destination Dataset** names are similar to the **Source Dataset** names, you can use a shortcut to enter the names on the panel. Type “=s” in **Destination Dataset** for VENDOR and VENDITEM. When you press ENTER, the **Source Dataset** names are copied to **Destination Dataset** and you can edit them to reflect the FOPDEMO2 qualifier. The completed panel is shown in Figure 70.

Note: The **Destination Dataset** names are not enclosed in single quotes and will be prefixed with the **Data Set Prefix**, specified as a user option.

Use END to return to the INSERT Process menu.

Create the Objects and Perform the Insert

Select Option 2 PERFORM on the INSERT Process menu. Since several of the objects do not exist at the destination, Move displays the CREATE Object List panel, listing all objects.



```
---------------------------- CREATE Object List -------------------------------
Command ===>                                                  Scroll ===> PAGE

Primary : CREATE ALL, DROP ALL, DROP EXISTS, DROP CONFLICTS, DROP CHANGED
          DEFAULTS, SHOW                                            1 of 13
   Line : S, U, I, CR(A), DR(A), DB2, OPT, SQL

Cmd  Status     Type             Object Name           Database Tablespace
--- --------  ---------  ---------------------------->> -------- ----------
*** ****************************** TOP ************************************
___ EXISTS    TABLE      FOPDEMO2.BKORDER              TESTB     BKORDER
___ SELECT      FK(OPT)  RIB
___ EXISTS    LEGACY     FOPDEMO.VENDOR
___ SELECT      DATASET  FOPDEMO2.VENDOR
___ EXISTS    LEGACY     FOPDEMO.VENDITEM
___ SELECT      DATASET  FOPDEMO2.VENDITEM
___ EXISTS      REL      RVV
___ EXISTS    TABLE      FOPDEMO2.ITEMS                TESTB     ITEM1HZY
___ EXISTS      INDEX    FOPDEMO2.XITEMPK
___ EXISTS      PK(DB2)
*** ****************************** BOTTOM *********************************

Review SQL Before Create ===> Y    (Y-YES, N-NO)
```

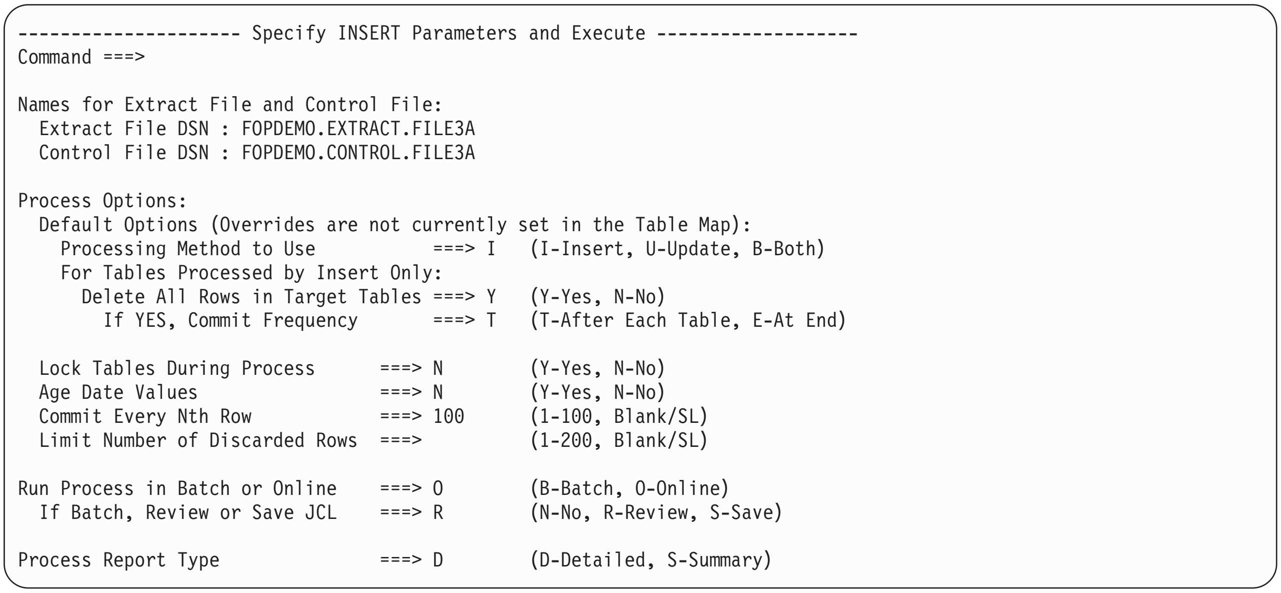
Figure 71. CREATE Object List

(Details of this panel are discussed in “Perform the Insert” on page 26 for Scenario 1 and “Perform the Insert” on page 51 for Scenario 2.) Note that the **Status** for DB2 and Legacy Tables is EXISTS; you will create only the Legacy data sets and the foreign key for the BKORDER table (SELECT status).

Use the CREATE ALL command to begin the process.

Move prompts for information needed to allocate the new data sets. Provide the allocation information and press ENTER to allocate the first data set and display the second allocation panel. When you press ENTER after completing the second allocation panel, Move redisplays the CREATE Object List panel, where the **Status** for the data sets is CREATED. A message on the panel indicates that the allocation was successful.

Now that the data sets and the foreign key are created, you must insert the data. Use END to display the Specify INSERT Parameters and Execute panel.

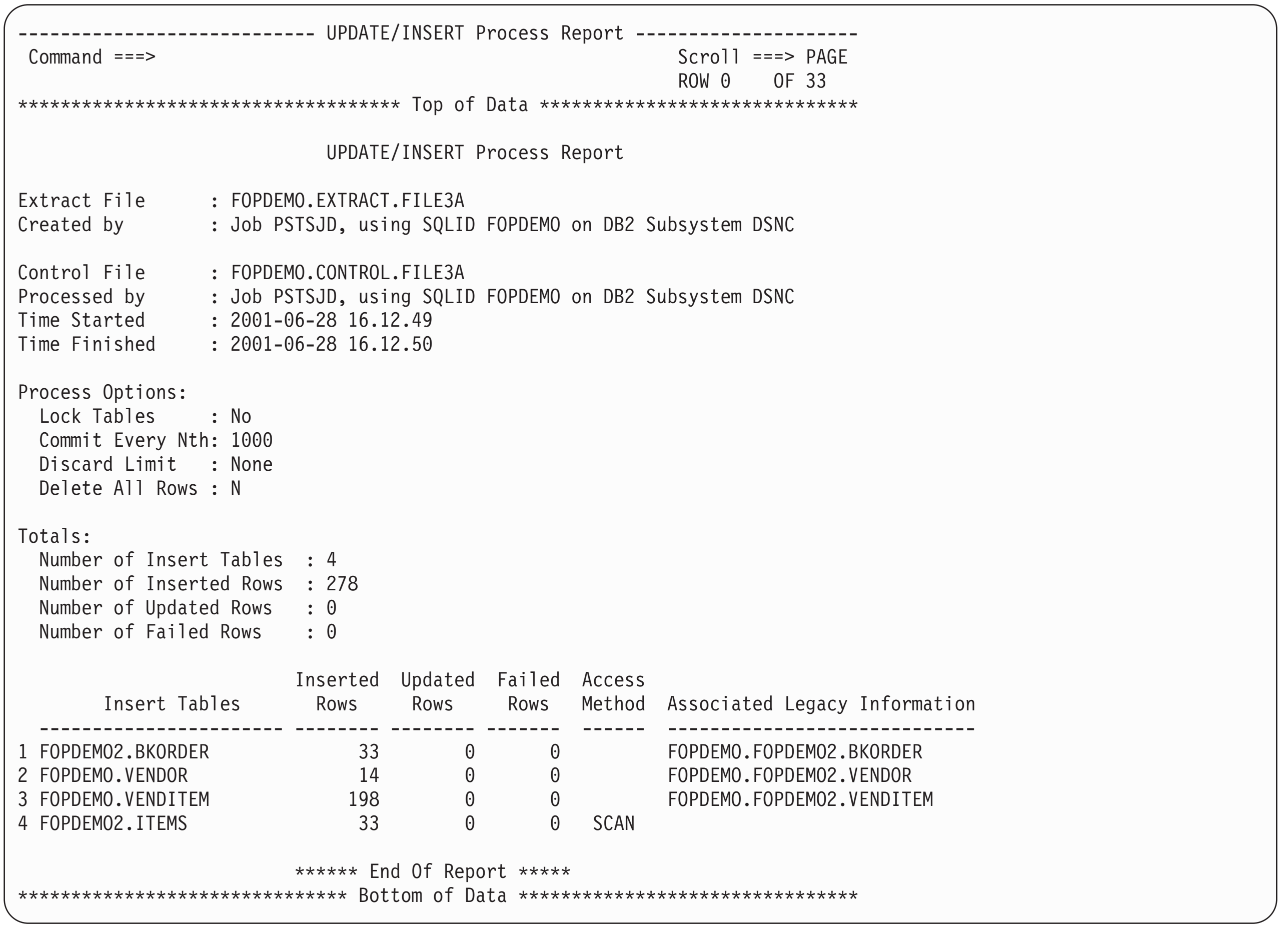


```
--------------------- Specify INSERT Parameters and Execute -------------------
Command ===>

Names for Extract File and Control File:
  Extract File DSN : FOPDEMO.EXTRACT.FILE3A
  Control File DSN : FOPDEMO.CONTROL.FILE3A

Process Options:
  Default Options (Overrides are not currently set in the Table Map):
    Processing Method to Use          ===> I   (I-Insert, U-Update, B-Both)
    For Tables Processed by Insert Only:
      Delete All Rows in Target Tables ===> Y   (Y-Yes, N-No)
        If YES, Commit Frequency       ===> T   (T-After Each Table, E-At End)

  Lock Tables During Process       ===> N       (Y-Yes, N-No)
  Age Date Values                  ===> N       (Y-Yes, N-No)
  Commit Every Nth Row             ===> 100     (1-100, Blank/SL)
  Limit Number of Discarded Rows  ===>          (1-200, Blank/SL)

Run Process in Batch or Online    ===> O       (B-Batch, O-Online)
  If Batch, Review or Save JCL    ===> R       (N-No, R-Review, S-Save)

Process Report Type               ===> D       (D-Detailed, S-Summary)
```

Figure 72. Specify INSERT Parameters and Execute

Specify I (Insert) for **Processing Method to Use**. When creating a test database, you may want to clear any data from existing DB2 tables before inserting new data. To be certain that no data is currently in FOPDEMO2.ITEMS and FOPDEMO2.BKORDER, type Y to **Delete All Rows in Target Tables** and enter T as the **Commit Frequency**. Press ENTER to execute the Insert Process.

The Insert Process is complete, and Move automatically displays an INSERT Process Report documenting the created objects and inserted data, similar in format to the EXTRACT Process Report.



```
---------------------------- UPDATE/INSERT Process Report ---------------------
 Command ===>                                                 Scroll ===> PAGE
                                                              ROW 0    OF 33
************************************ Top of Data *****************************

                             UPDATE/INSERT Process Report

Extract File      : FOPDEMO.EXTRACT.FILE3A
Created by        : Job PSTSJD, using SQLID FOPDEMO on DB2 Subsystem DSNC

Control File      : FOPDEMO.CONTROL.FILE3A
Processed by      : Job PSTSJD, using SQLID FOPDEMO on DB2 Subsystem DSNC
Time Started      : 2001-06-28 16.12.49
Time Finished     : 2001-06-28 16.12.50

Process Options:
  Lock Tables     : No
  Commit Every Nth: 1000
  Discard Limit   : None
  Delete All Rows : N

Totals:
  Number of Insert Tables  : 4
  Number of Inserted Rows  : 278
  Number of Updated Rows   : 0
  Number of Failed Rows    : 0

                          Inserted  Updated  Failed  Access
        Insert Tables       Rows     Rows     Rows   Method  Associated Legacy Information
  ----------------------- -------- -------- -------  ------  -----------------------------
1 FOPDEMO2.BKORDER              33        0       0          FOPDEMO.FOPDEMO2.BKORDER
2 FOPDEMO.VENDOR                14        0       0          FOPDEMO.FOPDEMO2.VENDOR
3 FOPDEMO.VENDITEM             198        0       0          FOPDEMO.FOPDEMO2.VENDITEM
4 FOPDEMO2.ITEMS                33        0       0   SCAN

                          ****** End Of Report *****
****************************** Bottom of Data ********************************
```

Figure 73. INSERT Process Report

Scenario 3b: IMS Legacy Tables

In this scenario, you will use sample data to extract employee records from the IMS data sets FOPDEPDB and FOPJOBDB.

You will do this by creating Legacy Tables, Environment Definitions, and Retrieval Definitions that reference the IMS Control Blocks (DBDs, PSBs, PCBs, etc.) and the associated data sets for IMS databases containing the data you want to extract. Using an Access Definition, you will extract the legacy data into an Extract File. Finally, you will create an IMS test database by inserting the data from the Extract File into empty data sets, included with the sample data.

To extract data from the IMS data sets you will:

- Create an Environment Definition (FOPDEMO) that identifies the IMS Program Libraries, the DBD and PSB Program Libraries, and the DFSVSAMP data set and member name.
- Create an IMS Legacy Table (FOPDEMO.EMPLOYEE) and specify the name of the Copybook (SAMPDEPT) as well as the IMS DBD name (FOPDEPDB) and the segment within the DBD (EMPLOYEE). In this scenario, you will also use three sample IMS Legacy Tables created during installation: FOPDEMO.DEPARTMENT and FOPDEMO.POSITION (which are associated with segments DEPT and POSITION in the DBD FOPDEPDB), and FOPDEMO.JOBCODE (which is associated with segment JOBS in the DBD FOPJOBDB).

- Create a Retrieval Definition providing the default PSB Name (FOPDEPPR), PCB number (1), and IMS ID (IMSA) for the DBD FOPDEDB, and associating the data set for the IMS database with each DD (data definition) within the DBD.
- Modify an existing Access Definition, adding the IMS Legacy Tables and creating an Optim relationship between FOPDEMO.JOBCODE and FOPDEMO.POSITION. Logical relationships between the other Legacy Tables already exist, so there is no need to create an Optim relationship for these tables.
- Once the Access Definition is modified, you will extract the legacy data and relationships into an Extract File.
- Create a Table Map to specify the destination Legacy Tables, and associate the Legacy Tables with destination IMS data sets. The destination Legacy Tables and destination data sets are created during installation.
- Insert the legacy data from the Extract File into the specified IMS data sets.

Defining an Environment Definition

First, you must create the FOPDEMO Environment Definition to be used by all Legacy Tables that reference IMS data. When you create an Environment Definition, you specify:

- The IMS Program Libraries to be used during processing.
- The DBD and PSB Libraries.
- The DFSVSAMP DSN/Member Name.
- The IMS System ID and the AGN (Application Group Name), if the data is online to a control region.

Note: The sample files will usually not be online to a control region.

Defining an IMS Legacy Table

You can easily create the IMS Legacy Table from one or more COBOL or PL/I copybooks distributed with Optim Legacy. By creating the FOPDEMO.DEPARTMENT Legacy Table, you will:

- Select the copybook library and member name used to define the Legacy Table.
- Specify the IMS DBD Name and Segment associated with the Legacy Table.
- Edit column names, field details, etc. (Optional)
- Use criteria to match the Legacy Table to specific records in the source file. (Optional)
- Specify the default data source for the Legacy Table. (Optional)

Defining a Retrieval Definition (Optional)

A Retrieval Definition allows you to define default settings for processing IMS data. You can change these specifications for an individual process (e.g., an Extract or Insert process). For this scenario, you will:

- Name the Retrieval Definition, using an Environment Definition and DBD Name.
- Specify the name of the PSB and the number of the PCB that allow you to access and manipulate IMS data.
- Associate each DD within the DBD with a data set for the IMS database.

Defining the Source

Just a few steps are needed to obtain the desired set of data. You will:

- Modify the Access Definition by specifying DEPARTMENT as the Start Table for the Extract Process, and adding the other Legacy Tables to it.
- Review the default PSBs, PCBs, and source data sets, populated by the Retrieval Definitions, associated with the Legacy Tables.

- Define relationships to be traversed.
- Extract the data.

Defining the Destination

After you extract the data, it must be prepared for insertion into the IMS destination data sets. You will:

- Modify the INSERT Process Table Map.
- Associate the IMS Legacy Tables with data sets for the destination IMS database.
- Insert the destination data sets using the INSERT Process.
- Insert the data into DB2 tables and data sets associated with Legacy Tables.

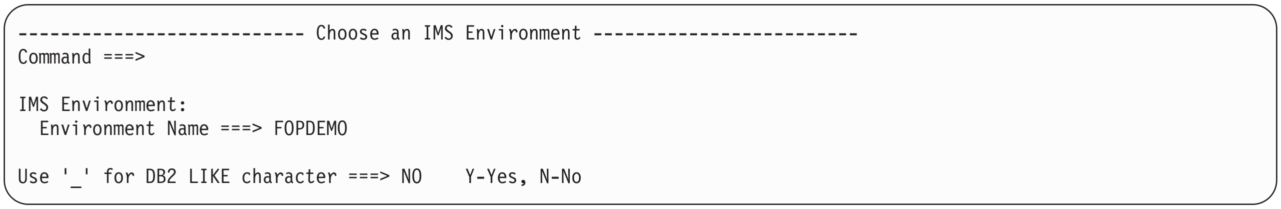
Defining an Environment Definition

Optim Legacy uses an Environment Definition to provide the information needed to access IMS data. An Environment Definition is a user-defined object that simplifies the definition of Optim objects (IMS Legacy Tables and Retrieval Definitions).

For this scenario, you will reference the FOPDEMO Environment Definition, using the DBD, PSB, and IMS Program Libraries, provided as part of the sample data. The FOPDEMO Environment Definition is referenced by all source Legacy Tables and Retrieval Definitions used in this scenario.

Create an Environment Definition

Select Option 6 DEFINITIONS from the Main Menu, then Option 7 ENVIRONMENT DEFINITION from the Choose a Definition Option menu to display the Choose an IMS Environment panel.



```
--------------------------- Choose an IMS Environment -------------------------
Command ===>

IMS Environment:
  Environment Name ===> FOPDEMO

Use '_' for DB2 LIKE character ===> NO    Y-Yes, N-No
```

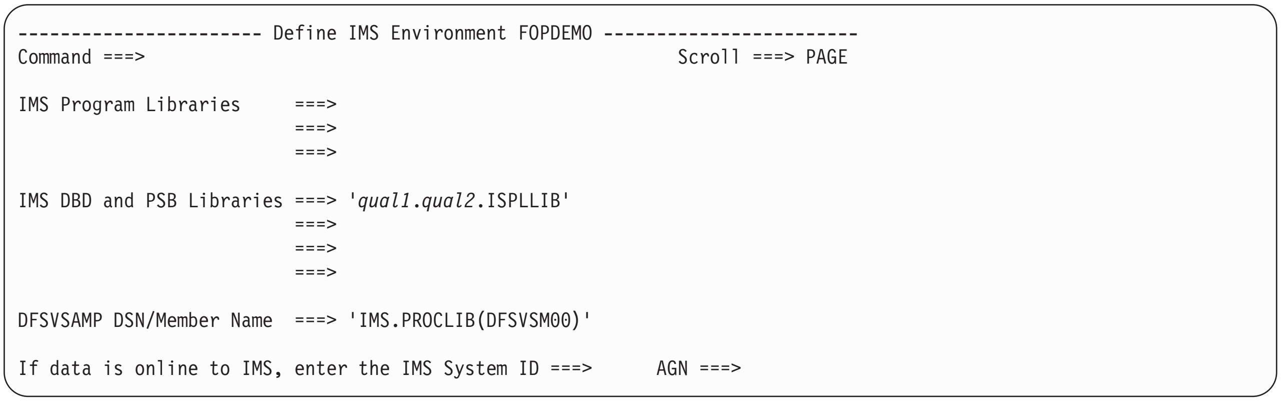
Figure 74. Choose an IMS Environment

An Environment Definition has a one-part name. You can:

- Review or edit an existing Environment Definition by entering the Environment Name.
- Obtain a selection list by using DB2 LIKE syntax.
- Create a new Environment Definition by entering an unused name.

Enter FOPDEMO as the **Environment Name** on the Choose an IMS Environment panel.

When you enter the name of an Environment Definition that does not exist, Move displays the Define IMS Environment panel.



```
----------------------- Define IMS Environment FOPDEMO ------------------------
Command ===>                                                  Scroll ===> PAGE

IMS Program Libraries     ===>
                          ===>
                          ===>

IMS DBD and PSB Libraries ===> 'qual1.qual2.ISPLLIB'
                          ===>
                          ===>
                          ===>

DFSVSAMP DSN/Member Name  ===> 'IMS.PROCLIB(DFSVSM00)'

If data is online to IMS, enter the IMS System ID ===>      AGN ===>
```

Figure 75. Define IMS Environment

The Define IMS Environment panel allows you to reference the IMS Program Libraries, the IMS DBD and PSB Libraries, and the DFSVSAMP data set and member. Also, you can specify how the data is processed when online to a control region. This information allows Move to access IMS data utilizing the proper libraries.

IMS Program Libraries

Enter the names of the IMS Program Libraries that define and load the libraries required to run IMS. You can type a name, or enter wildcard characters to select a library from a list.

You can enter up to three library names:

Note:

- All data sets defined as IMS program libraries must be APF-authorized.
- If you have authorized the Optim SFOPLLIB data set, add it to this list.
- If your site uses IMS dynamic allocation, include the name of the data set containing the dynamic allocation load modules.
- If the IMS Program Libraries are already available (e.g., by Linklist or other method), you do not need to enter the library names.

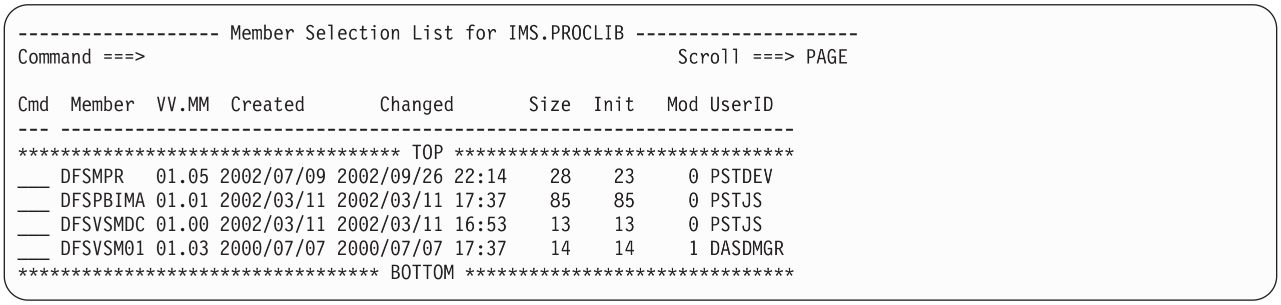
IMS DBD and PSB Libraries

Enter the names of the libraries for the DBD and PSBs that define the structure of the IMS database, as well as the method Move uses to access the data. Move requires that you enter at least one library name and allows you to enter as many as four. You can also use wildcard characters to select a library from a list.

For this scenario, enter '*qual1*.*qual2*.ISPLLIB' as the IMS DBD and PSB library name.

DFSVSAMP DSN/Member Name

Specify the DFSVSAMP data set and member name containing the IMS buffer parameters to be used when running in batch mode. You can also use wildcard characters to display the Select PDS Library pop-up window and select the DFSVSAMP data set. Or, enter the DFSVSAMP data set name to display the Member Selection List panel and select the DFSVSAMP member name.



```
------------------- Member Selection List for IMS.PROCLIB ---------------------
Command ===>                                                Scroll ===> PAGE

Cmd  Member  VV.MM  Created       Changed       Size  Init   Mod UserID
--- ------------------------------------------------------------------------
********************************** TOP *******************************
___ DFSMPR   01.05 2002/07/09 2002/09/26 22:14    28    23     0 PSTDEV
___ DFSPBIMA 01.01 2002/03/11 2002/03/11 17:37    85    85     0 PSTJS
___ DFSVSMDC 01.00 2002/03/11 2002/03/11 16:53    13    13     0 PSTJS
___ DFSVSM01 01.03 2000/07/07 2000/07/07 17:37    14    14     1 DASDMGR
******************************** BOTTOM ******************************
```

Figure 76. DFSVSAMP Member Selection List

IMS System ID and AGN

If the IMS database has been allocated to a control region, enter the IMS System ID and the IMS Application Group Name (AGN) required to process the data.

When you have completed the Environment Definition, use END to save it and return to the Choose an IMS Environment panel. Use END again to return to the Choose a Definition Option menu.

Defining a Legacy Table

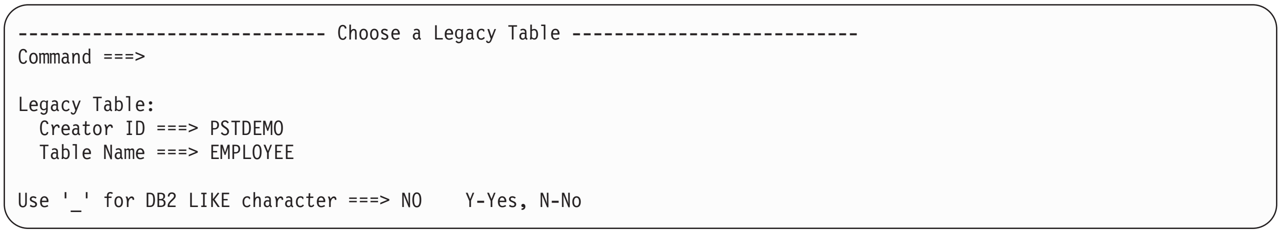
Optim Legacy uses an IMS Legacy Table to incorporate IMS legacy data into Move processes. The Legacy Table is a user-defined object that reflects the structure of segments in an IMS database and can be used and reused to extract similarly structured records from any database segment with which it is associated.

Since a legacy file may include several record structures, you can associate more than one Legacy Table with a segment. Once you have defined a Legacy Table, you can reference the Legacy Table in Optim objects (e.g., Access Definitions or Table Maps) as if it were a DB2 table.

For this scenario, you will extract data from four legacy tables, each referencing IMS data, DEPARTMENT, EMPLOYEE, JOBCODE, and POSITION. The sample data includes Legacy Tables for records in DEPARTMENT, POSITION, and JOBCODE. However, you must create a Legacy Table for EMPLOYEE.

Create a Legacy Table for IMS data

To begin, select Option 6 DEFINITIONS from the Optim Main Menu, then Option 6 LEGACY TABLES from the Choose a Definition Option menu to display the Choose a Legacy Table panel.



```
----------------------------- Choose a Legacy Table ---------------------------
Command ===>

Legacy Table:
  Creator ID ===> PSTDEMO
  Table Name ===> EMPLOYEE

Use '_' for DB2 LIKE character ===> NO    Y-Yes, N-No
```

Figure 77. Choose a Legacy Table

Legacy Table Name

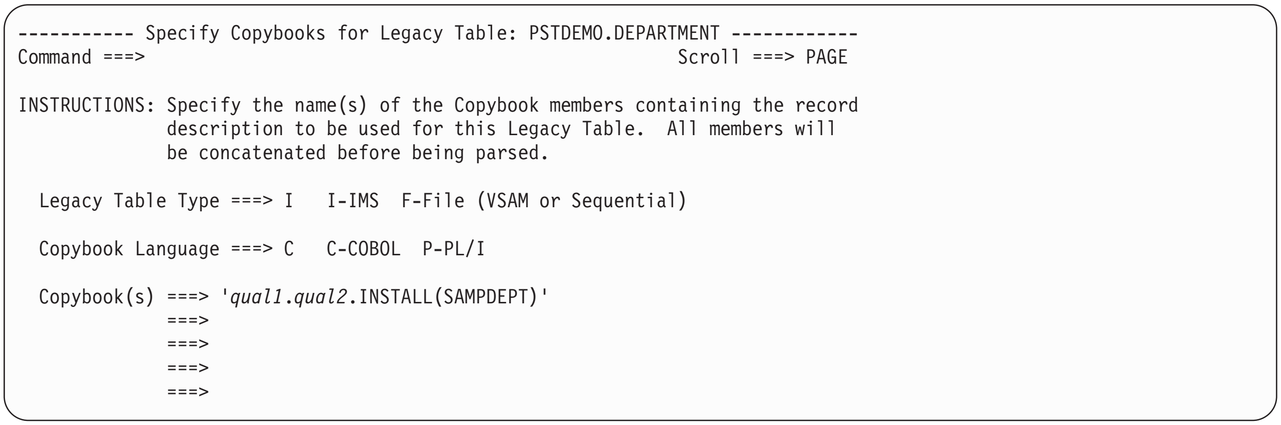
As for a VSAM Legacy Table, an IMS Legacy Table has a two-part name. You can:

- Review or edit an existing Legacy Table by entering the Creator ID and Table Name.
- Obtain a selection list by using DB2 LIKE syntax or leaving one part blank.
- Create a new Legacy Table by entering an unused name.

Note: Each IMS Legacy Table must have a matching Environment Definition. If a matching Environment Definition has not been predefined, you can create it by using the ENV command from the Specify Copybooks for Legacy Table panel.

For this scenario, you will create the EMPLOYEE Legacy Table for IMS data, using a copybook provided with the sample data. Enter `EMPLOYEE` as the **Table Name** on the Choose a Legacy Table panel. The **Creator ID** is `PSTDEMO`.

When you enter the name of a Legacy Table that does not exist, Move displays the Specify Copybooks for Legacy Table panel.



```
----------- Specify Copybooks for Legacy Table: PSTDEMO.DEPARTMENT ------------
Command ===>                                                  Scroll ===> PAGE

INSTRUCTIONS: Specify the name(s) of the Copybook members containing the record
              description to be used for this Legacy Table.  All members will
              be concatenated before being parsed.

  Legacy Table Type ===> I   I-IMS  F-File (VSAM or Sequential)

  Copybook Language ===> C   C-COBOL  P-PL/I

  Copybook(s) ===> 'qual1.qual2.INSTALL(SAMPDEPT)'
              ===>
              ===>
              ===>
              ===>
```

Figure 78. Specify Copybooks for Legacy Table

Legacy Table Type

First, you must specify the type of data to be used with this Legacy Table:

- An `F` entry means the data is stored in sequential or VSAM files.
- An `I` entry means the data is stored in an IMS database.

The FOPDEMO.DEPARTMENT Legacy Table references IMS data, so enter `I` as the Legacy Table Type.

Copybook Language

Specify the language used to create the copybooks. Specify C (COBOL) or P (PL/I). For this scenario, enter `C` as the Copybook Language.

Copybook(s)

To create a Legacy Table manually, you can leave all Copybook entries blank. Move displays the Define Legacy Table panel in Edit mode, without populating any Field names. This allows you to manually create the Legacy Table.

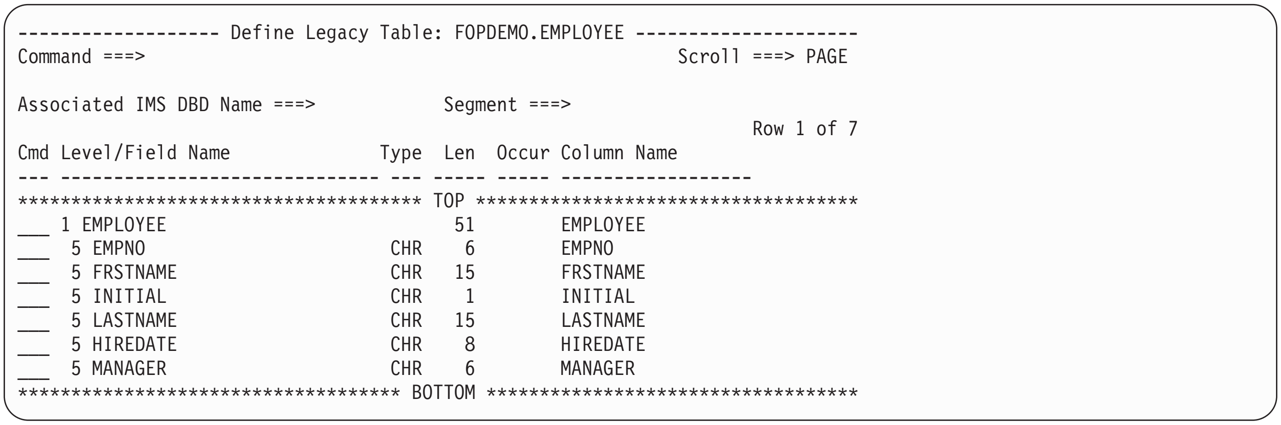
You may enter as many as five copybook names on the Specify Copybooks for Legacy Table panel. If a desired copybook is not in the default copybook library, you must provide both the copybook data set and member names (e.g., `'FOPDEMO.RT.INSTALL(SAMPLE)'`). You can also use a wildcard character with a fully-qualified data set name (e.g., `'FOPQA.RT.*'`) to display a more limited selection list.

Note:

- If the copybook member does not exist in the default or specified copybook library, an error message is displayed.

- If you do not use single quotes (' ') around the copybook library (and member) name, Move opens the Default Copybook Library, specified as a Legacy Option, in the Member Selection List panel. If there is no Default Copybook Library, the Data Set Prefix, specified as a User Option, is used as the prefix.

For this scenario, enter `'`*`qual1.qual2`*`.INSTALL(SAMPDEPT)'` as the copybook library and member name. Press ENTER to display the Legacy Table editor, populated with information from the copybook.



```
-------------------- Define Legacy Table: FOPDEMO.EMPLOYEE ---------------------
Command ===>                                                  Scroll ===> PAGE

Associated IMS DBD Name ===>            Segment ===>
                                                                     Row 1 of 7
Cmd Level/Field Name              Type  Len  Occur Column Name
--- ------------------------------ --- ----- ----- ------------------
************************************** TOP ***********************************
___ 1 EMPLOYEE                           51        EMPLOYEE
___  5 EMPNO                       CHR    6        EMPNO
___  5 FRSTNAME                    CHR   15        FRSTNAME
___  5 INITIAL                     CHR    1        INITIAL
___  5 LASTNAME                    CHR   15        LASTNAME
___  5 HIREDATE                    CHR    8        HIREDATE
___  5 MANAGER                     CHR    6        MANAGER
************************************ BOTTOM **********************************
```

Figure 79. Define Legacy Table

Associated IMS DBD Name

Specify the DBD name you want to associate with the Legacy Table. The DBD must be in a DBD Library referenced in the Environment Definition. Enter an asterisk to generate a selection list of DBDs in the referenced Environment Definition.

Segment

Specify the name of the segment in the specified DBD that you want to associate with the Legacy Table. Enter an asterisk to generate a selection list of segments in the specified DBD.

Edit Column Names

Move generates DB2-compatible column names from the names of fields in the copybook. The column names are used to identify and reference legacy fields and the data in them. Wherever possible, these generated column names are exactly the same as those in the copybook. However, COBOL field names can be as many as 30 characters and PL/I field names can be 31 characters, while Legacy Table column names must be unique and cannot exceed 18 characters. Thus, Move may truncate or change COBOL and PL/I field names to generate column names. In addition, dashes in a field name are translated to underscore characters in the column name.

EDIT Command

From the Modify Legacy Table panel, you can edit the column names to make them more useful or descriptive. However, you must use the EDIT primary command to enable editing of other information displayed on the Modify Legacy Table panel. Alternatively, you can also use the Z (Zoom) line command to display and edit **Field Details**.

Other Commands

Use the FIND command to locate field names that include a specific character string. After using FIND, RFIND continues the search from the current cursor location.

The ONLY command is useful when you are working with a large Legacy Table and want to focus on a subset of fields. ONLY lists only fields that satisfy command parameters (e.g., limiting the display to fields containing the string “-ID”). Use the SHOW command to redisplay all fields after using ONLY.

While editing the Legacy Table, you can use VERIFY to check for errors without saving the Legacy Table or terminating the session. SAVE verifies the edited Legacy Table. If errors are found, you are prompted to correct them before the Legacy Table can be saved.

CRITERIA Command

If an IMS DBD contains segments with different layouts, you can use criteria to match a Legacy Table with specific segments in the source file. For example, a file might contain three types of segments of the same length, but different formats. The applications that use this file recognize the segment types by checking a code in the first field. You can use criteria to link a Legacy Table to the correct segments.

The CRITERIA primary command invokes the Modify Criteria for Legacy Table panel. When defining or editing criteria, you can use the LIST COLUMNS command to display a selection list of column variables for the Legacy Table. Your selection(s) are inserted onto the Modify Criteria for Legacy Table panel. (For further information about this and other commands mentioned in this scenario, see Legacy Tables in the *Move User Manual*, and Primary Commands in the *Command Reference Manual*.)

Default Data Source Association

To be used in Move processes, a Legacy Table must be linked with an IMS database data set containing the data to be processed. You can establish a default data set name to be used unless overridden during processing.

When you are creating an IMS Legacy Table, use the DS primary command to define a new or edit an existing Retrieval Definition. For more information see “Defining a Retrieval Definition.”

This completes the creation of the EMPLOYEE Legacy Table for this scenario. Use END twice to return to the Choose a Definition Option menu.

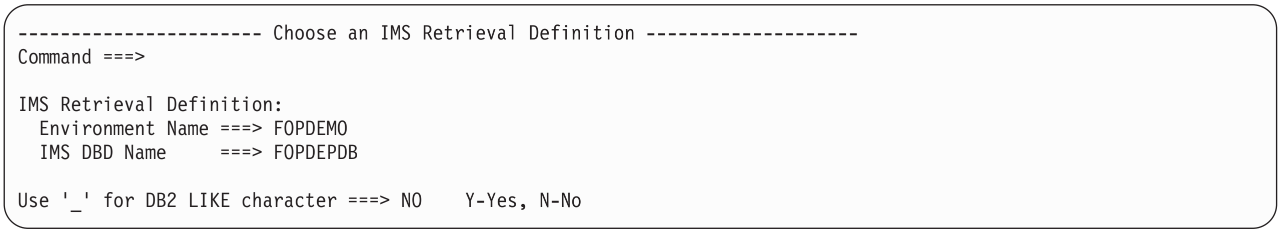
Defining a Retrieval Definition

Optim Legacy uses an IMS Retrieval Definition for default settings needed to process IMS data. You can change the specifications for an individual process (e.g., an Extract or Insert process).

By creating a Retrieval Definition, you provide the name of the default PSB and number of the PCB to be used to access the specified DBD, and associate segments within the DBD with default names for the database data sets to be processed.

Create a Retrieval Definition

To begin, select Option 6 DEFINITIONS from the Main Menu, then Option 8 IMS RETRIEVAL from the Choose a Definition Option menu to display the Choose an IMS Retrieval Definition panel.



```
----------------------- Choose an IMS Retrieval Definition --------------------
Command ===>

IMS Retrieval Definition:
  Environment Name ===> FOPDEMO
  IMS DBD Name     ===> FOPDEPDB

Use '_' for DB2 LIKE character ===> NO    Y-Yes, N-No
```

Figure 80. Choose an IMS Retrieval Definition

Retrieval Definition Name

A Retrieval Definition has a two-part name. You can:

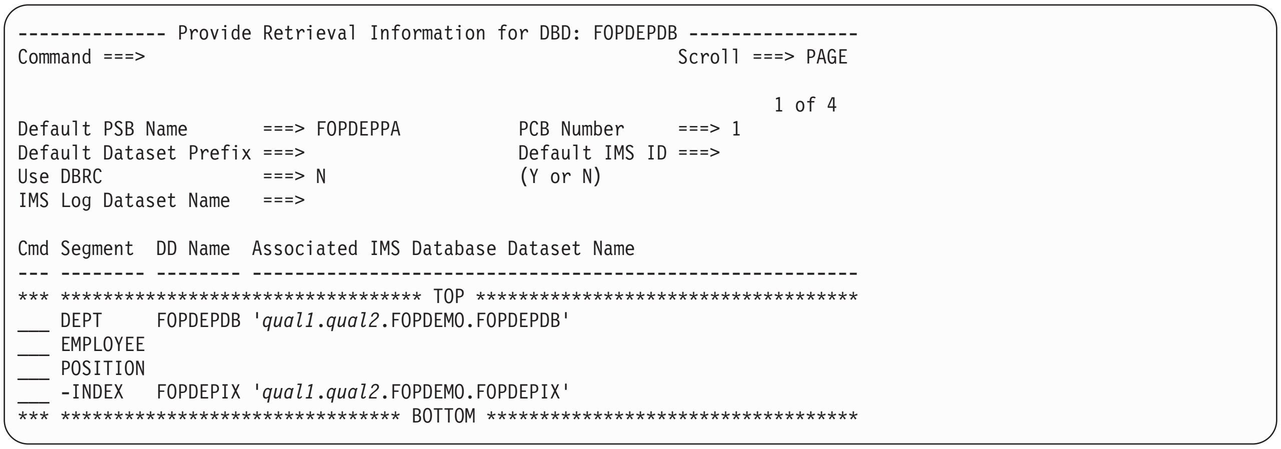
- Review or edit an existing Retrieval Definition by entering the Environment Name and IMS DBD Name.
- Obtain a selection list by using DB2 LIKE syntax or leaving one part blank.
- Create a new Retrieval Definition by entering an unused name.

Note: The two-part name of the Retrieval Definition is used to associate the default settings with one or more Legacy Tables. Any IMS Legacy Table that references the Environment Name, FOPDEMO, and the DBD, FOPDEPDB, is associated with this Retrieval Definition.

For this scenario, you will create the FOPDEMO.FOPDEPDB Retrieval Definition, referencing the FOPDEMO Environment Definition and the FOPDEPDB DBD. This Retrieval Definition provides default information for the DEPARTMENT, EMPLOYEE, and POSITION Legacy Tables when used in an Extract or Insert Process.

The sample data includes an additional Retrieval Definition, FOPDEMO.FOPJOBDB, that is referenced when the JOBCODE Legacy Table is used in an Extract or Insert Process.

When you enter the name of a Retrieval Definition that does not exist, Move displays the Provide Retrieval Information for DBD panel.



```
------------- Provide Retrieval Information for DBD: FOPDEPDB ----------------
Command ===>                                                  Scroll ===> PAGE

                                                                       1 of 4
Default PSB Name       ===> FOPDEPPA           PCB Number     ===> 1
Default Dataset Prefix ===>                    Default IMS ID ===>
Use DBRC               ===> N                  (Y or N)
IMS Log Dataset Name   ===>

Cmd Segment  DD Name  Associated IMS Database Dataset Name
--- -------- -------- ---------------------------------------------------------
*** ********************************** TOP ************************************
___ DEPT     FOPDEPDB 'qual1.qual2.FOPDEMO.FOPDEPDB'
___ EMPLOYEE
___ POSITION
___ -INDEX   FOPDEPIX 'qual1.qual2.FOPDEMO.FOPDEPIX'
*** ******************************* BOTTOM ************************************
```

Figure 81. Provide Retrieval Information for DBD

The Provide Retrieval Information for DBD panel allows you to specify the name of the PSB and number, of the PCB needed to access the referenced DBD, and to associate each DD with an IMS database data set. It also allows you to use DBRC and IMS logging for processing in DL/I mode.

Note: You can override any information in the Retrieval Definition when you create an Access Definition or Table Map that includes one or more Legacy Tables that reference the Retrieval Definition.

Default PSB Name

Enter the name of the default PSB, which must reside in a PSB library referenced in the specified Environment Definition. The PSB provides access to the IMS services that Move requires to access the database records.

For this scenario, specify FOPDEPPA for the **Default PSB Name**.

PCB Number

Enter the relative number of the default database PCB, which is within the specified PSB that grants Move the authorization to manipulate the data.

For this scenario, specify 1 for the **PCB Number**.

Default Dataset Prefix

The **Default Dataset Prefix** is an optional 1- to 8-character prefix used when specifying Associated IMS Database Dataset Names. You can enter a prefix (e.g., FOPQA), instead of typing it for each data set name (e.g., RT.FOPDEMO.FOPDEPDB).

Default IMS ID

Enter the default IMS System ID required to access the IMS data if allocated to a control region.

Use DBRC

This entry is valid only for IMS processing in DL/I mode (i.e, when an IMS ID is not specified). If appropriate, type **Y** for yes to use Database Recovery Control (DBRC) to control logging and perform database recovery; otherwise type **N** for no. IMS uses the online log datasets (OLDS) if the database is accessed in BMP or DBB mode.

The default for a HALDB (High Availability Large DataBase), is **Y**, and that entry cannot be changed.

DBRC use is optional for a non-HALDB, such as HIDAM, HDAM, HISAM, etc. Thus, you may specify **Y** for a non-HALDB, but it is not required.

IMS Log Dataset Name

If appropriate, specify the dataset name for the IMS Log used to perform database recovery. This dataset name is used to dynamically allocate the IEFRDER dataset. It is recommended that you provide a GDG dataset for the IMS log because Optim may make multiple calls to IMS while traversing through the legacy tables defined in an Access Definition or Table Map; this would result in the Log dataset being overwritten if a sequential dataset is used. If a GDG is used, Optim will allocate one GDS for each invocation of IMS, which will prevent the IMS log from being overwritten.

If a PSB with a Processing Option (PROCOPT) other than G (for Get) is used while accessing a HALDB in DL/I mode (i.e., an IMS region name is not specified), you must specify the name of the dataset to be allocated for DD Name IEFRDER.

If you specify an IMS Log Dataset Name, when you exit the Provide Retrieval Information for DBD panel, the Allocate Dataset panel automatically displays. You must provide sufficient Primary and Secondary space units on that panel to allocate the IEFRDER dataset. Failing to do so will cause IMS to abort processing and lock the database from further updates until a recover/rollback is done.

Segment

The segments within the specified DBD are listed here. This data cannot be modified.

DD Name

The names of each DD or Data Definition (i.e., the physical data sets) associated with each segment are listed here. This data cannot be modified.

Associated IMS Database Dataset Name

Enter the location of the default IMS Database Dataset associated with each data definition (DD) name in the DBD. This data is then associated with the named Legacy Table during processing.

Note: A Site Option (Require IMS Data Set Names) determines whether you can omit the data set name to allow IMS to dynamically allocate the data set. All users can specify '$MDA' as the data set name to choose dynamic allocation, regardless of this Site Option.

Note: You do not have to specify a dataset name for a DEDB or HALDB because the appropriate dataset name will already be known to the IMS subsystem.

In this scenario, associate DD FOPDEPDB with database data set '*qual1*.*qual2*.INSTALL.FOPDEPDB', and associate FOPDEPIX with database data set '*qual1*.*qual2*.INSTALL.FOPDEPIX'.

When you have completed defining the Retrieval Definition, use END to save it and return to the Choose an IMS Retrieval Definition panel. Use END again to return to the Choose a Definition Option menu.

Defining the Source

After Legacy Tables describing segments in the IMS DBDs are created, you can define the source data for extraction using the normal steps: create or modify the Access Definition, establish the path to follow when extracting data, select object definitions to be extracted, and perform the extract.

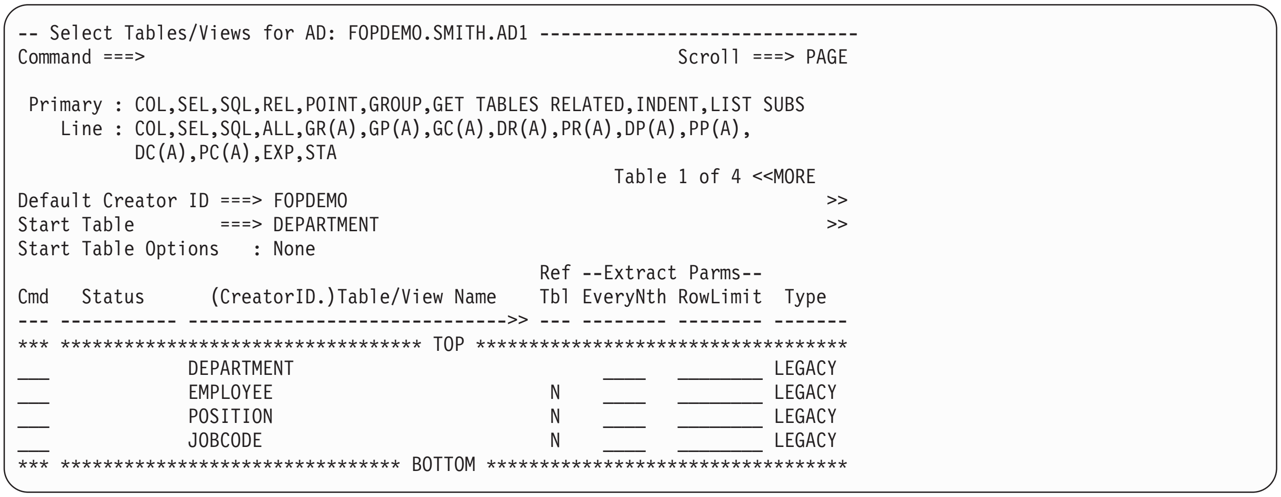
From the Main Menu, select Option 7 MIGRATION. On the Data Migration menu, select Option 1 EXTRACT.

The Extract Process for this scenario uses the Access Definition from previous scenarios. To edit the Access Definition, confirm that FOPDEMO.SMITH.AD1 is specified and select Option 1 TABLES from the EXTRACT Process menu. You will edit the Access Definition, removing all DB2 table names and adding the four IMS Legacy Tables, DEPARTMENT, EMPLOYEE, POSITION, and JOBCODE.

Edit Table List

Overtype the names of the CUSTOMERS, ORDERS, ITEMS, and DETAILS tables with the names of the IMS Legacy Tables, DEPARTMENT, EMPLOYEE, POSITION, and JOBCODE.

For this scenario, you will begin with the records in the DEPARTMENT Legacy Table. Replace ITEMS with DEPARTMENT at the **Start Table** prompt. For this scenario, all DEPARTMENT records are copied to the Extract File; you don't need to define selection criteria. The completed Access Definition is displayed.

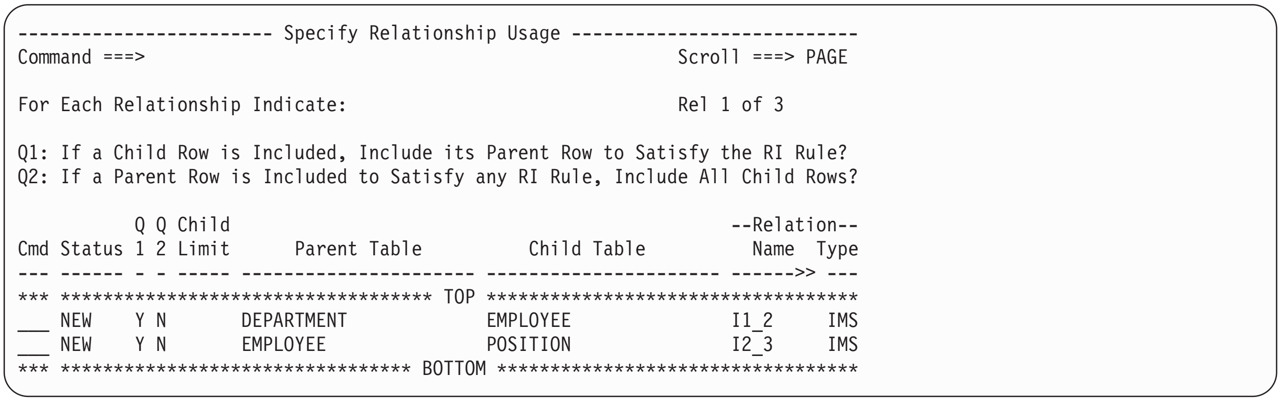


```
-- Select Tables/Views for AD: FOPDEMO.SMITH.AD1 ------------------------------
Command ===>                                                  Scroll ===> PAGE

 Primary : COL,SEL,SQL,REL,POINT,GROUP,GET TABLES RELATED,INDENT,LIST SUBS
    Line : COL,SEL,SQL,ALL,GR(A),GP(A),GC(A),DR(A),PR(A),DP(A),PP(A),
           DC(A),PC(A),EXP,STA
                                                        Table 1 of 4 <<MORE
Default Creator ID ===> FOPDEMO                                             >>
Start Table        ===> DEPARTMENT                                          >>
Start Table Options   : None
                                                 Ref --Extract Parms--
Cmd   Status      (CreatorID.)Table/View Name    Tbl EveryNth RowLimit  Type
--- ----------- ----------------------------->> --- -------- -------- -------
*** ********************************* TOP ***********************************
___             DEPARTMENT                             ____   ________ LEGACY
___             EMPLOYEE                          N    ____   ________ LEGACY
___             POSITION                          N    ____   ________ LEGACY
___             JOBCODE                           N    ____   ________ LEGACY
*** ******************************** BOTTOM *********************************
```

Figure 82. Completed Access Definition

Paths or Relationships

After completing the Table List, use the REL primary command to display the Specify Relationship Usage panel.



```
----------------------- Specify Relationship Usage ---------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                               --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- --------------------- --------------------- ------>> ---
*** ********************************* TOP **********************************
___ NEW    Y N       DEPARTMENT             EMPLOYEE               I1_2     IMS
___ NEW    Y N       EMPLOYEE               POSITION               I2_3     IMS
*** ******************************** BOTTOM ********************************
```

Figure 83. Specify Relationship Usage

The first time a relationship is listed in the Access Definition editor, it is assigned the NEW status, as shown in Figure 83. A relationship in NEW status will be used in the Extract Process, if the **Use NEW Relationships** parameter for Access Definitions is YES. (For further information, see Access Definition Parameter in the *Common Elements Manual*.)

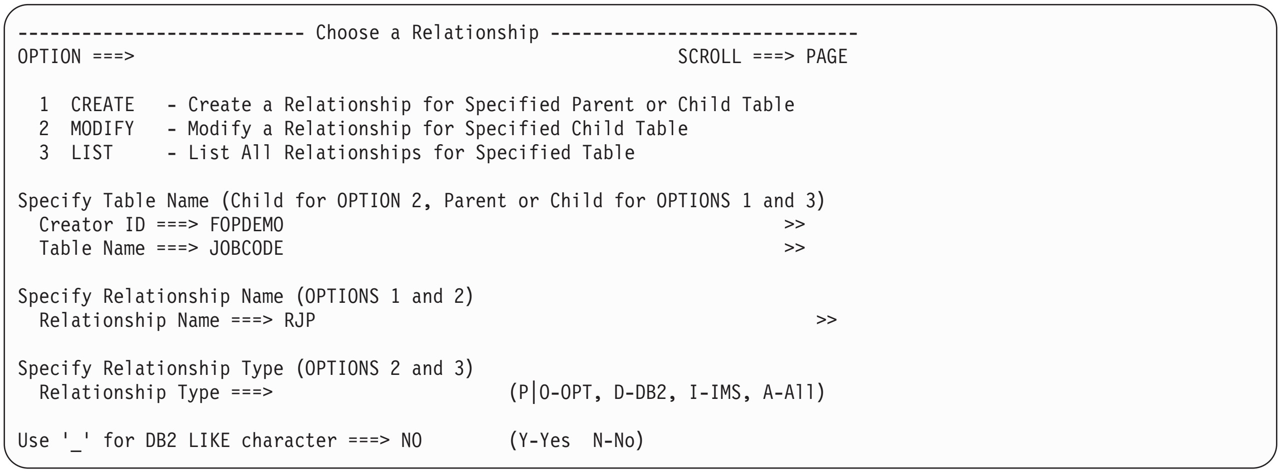
The Specify Relationship Usage panel lists an IMS relationship between the EMPLOYEE and POSITION tables, as well as one between the DEPARTMENT and EMPLOYEE tables. These are logical relationships created in the FOPDEPDB DBD. However, to extract the JOBCODE rows related to the records in POSITION, you must create the needed relationship. When defining Optim relationships:

- Primary keys and foreign keys are not required.
- Corresponding columns need not be identical, but must be compatible. (Compatible column types are described in the *Common Elements Manual* under Compatibility Rules.)
- Expressions that may include string literals, numeric constants, concatenation, and substrings can be used.

You can define an Optim relationship between a pair of Legacy Tables, between a pair of DB2 tables, or between a DB2 table and a Legacy Table. Move stores all Legacy Table definitions and Optim relationships in the Optim Directory.

For this scenario, you will define the relationship between the IMS Legacy Tables, POSITION and JOBCODE. To do this, use the CREATE REL primary command to display the Choose a Relationship panel.

You must provide the name of one table in the relationship. You can use DB2 LIKE syntax or leave one part of the table name blank in order to obtain a selection list. Since the relationship for this scenario involves tables with names prefixed with FOPDEMO, type FOPDEMO at **Creator ID**. Then type JOBCODE at **Table Name** and RJP (i.e., Relationship JOBCODE to POSITION) as the **Relationship Name**. The completed panel is shown in the following figure:

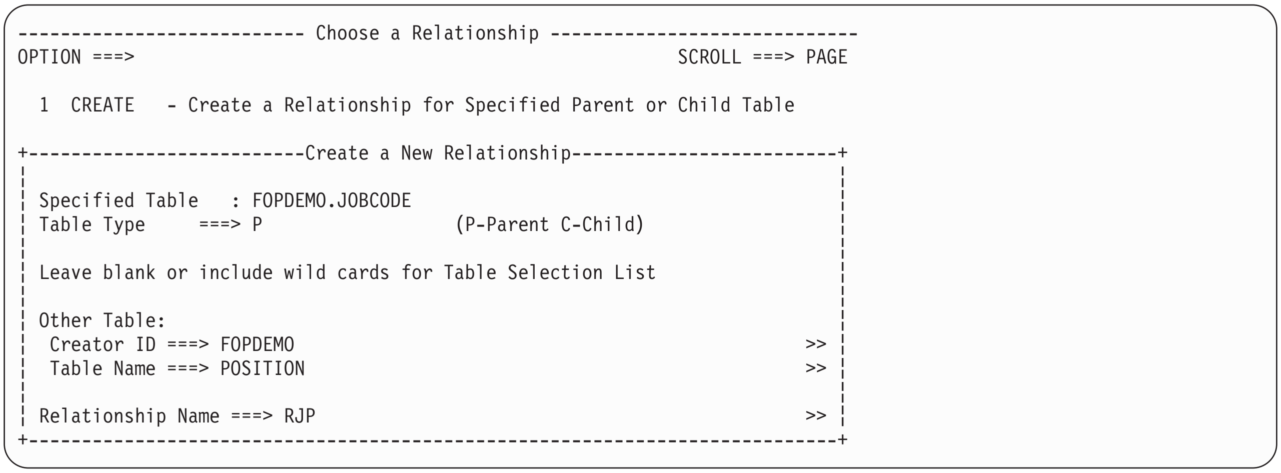


```
--------------------------- Choose a Relationship -----------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  CREATE   - Create a Relationship for Specified Parent or Child Table
  2  MODIFY   - Modify a Relationship for Specified Child Table
  3  LIST     - List All Relationships for Specified Table

Specify Table Name (Child for OPTION 2, Parent or Child for OPTIONS 1 and 3)
  Creator ID ===> FOPDEMO                                          >>
  Table Name ===> JOBCODE                                          >>

Specify Relationship Name (OPTIONS 1 and 2)
  Relationship Name ===> RJP                                          >>

Specify Relationship Type (OPTIONS 2 and 3)
  Relationship Type ===>                     (P|O-OPT, D-DB2, I-IMS, A-All)

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

Figure 84. Choose a Relationship

Define Relationship RJP

Select Option 1 CREATE and press ENTER to display the Create a New Relationship pop-up window.

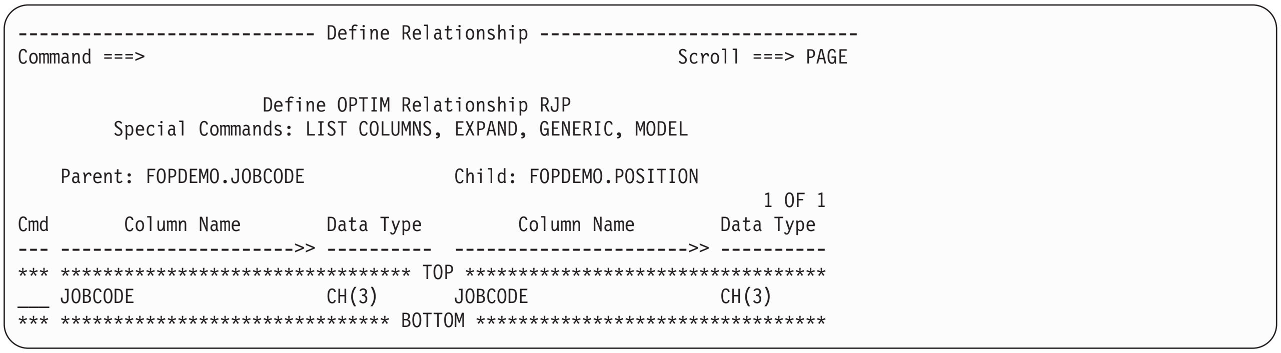


```
--------------------------- Choose a Relationship -----------------------------
OPTION ===>                                                   SCROLL ===> PAGE

  1  CREATE   - Create a Relationship for Specified Parent or Child Table

+--------------------------Create a New Relationship-------------------------+
¦                                                                            ¦
¦ Specified Table   : FOPDEMO.JOBCODE                                        ¦
¦ Table Type     ===> P                 (P-Parent C-Child)                   ¦
¦                                                                            ¦
¦ Leave blank or include wild cards for Table Selection List                 ¦
¦                                                                            ¦
¦ Other Table:                                                               ¦
¦  Creator ID ===> FOPDEMO                                                >> ¦
¦  Table Name ===> POSITION                                               >> ¦
¦                                                                            ¦
¦ Relationship Name ===> RJP                                              >> ¦
+----------------------------------------------------------------------------+
```

Figure 85. Create a New Relationship

Type P as the **Table Type** to designate the JOBCODE table as the parent. Move populates **Creator ID** for the second table in the relationship with the Creator ID for JOBCODE. Enter the name of the child table, POSITION, at **Table Name**.

Press ENTER to display the Define Relationship panel.

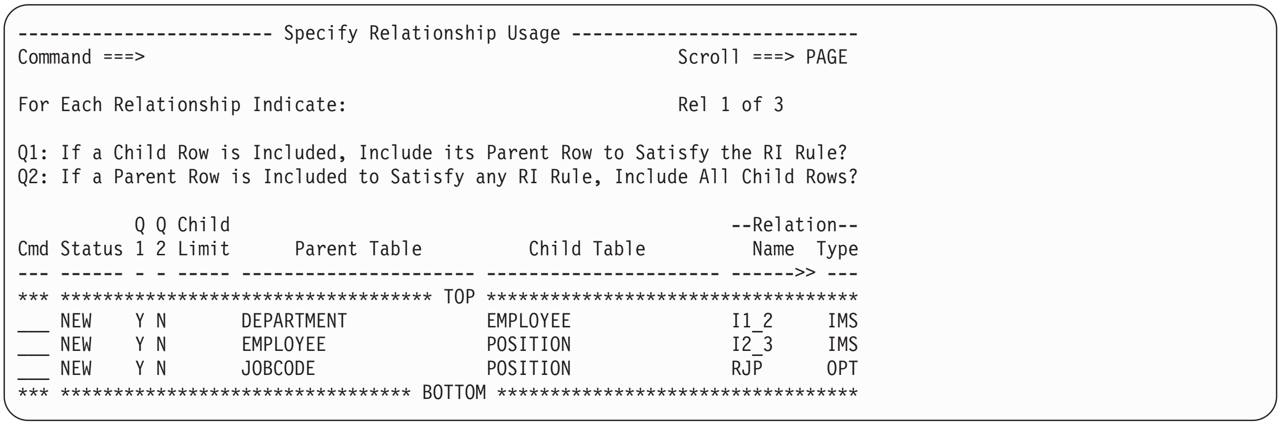


```
---------------------------- Define Relationship -----------------------------
Command ===>                                                  Scroll ===> PAGE

                       Define OPTIM Relationship RJP
         Special Commands: LIST COLUMNS, EXPAND, GENERIC, MODEL

    Parent: FOPDEMO.JOBCODE             Child: FOPDEMO.POSITION
                                                                        1 OF 1
Cmd       Column Name        Data Type         Column Name        Data Type
--- ---------------------->> ----------  ---------------------->> ----------
*** ********************************* TOP **********************************
___ JOBCODE                  CH(3)       JOBCODE                  CH(3)
*** ******************************* BOTTOM *********************************
```

Figure 86. Define Relationship

If the parent table has a primary key, as JOBCODE does, the names of primary key columns and data types are displayed on the parent side of the panel. The names of any columns in the child table that have the same names and compatible data types are displayed on the child side of the panel. JOBCODE is the primary key for the JOBCODE table. Since JOBCODE in the POSITION table is of the same type and length, Move populates both sides of the Define Relationship panel for the relationship RJP.

Use END twice to save the relationship and return to the Specify Relationship Usage panel.



```
------------------------ Specify Relationship Usage ---------------------------
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

          Q Q Child                                                --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table          Name  Type
--- ------ - - ----- --------------------- --------------------- ------>> ---
*** *********************************** TOP ***********************************
___ NEW    Y N       DEPARTMENT            EMPLOYEE              I1_2     IMS
___ NEW    Y N       EMPLOYEE              POSITION              I2_3     IMS
___ NEW    Y N       JOBCODE               POSITION              RJP      OPT
*** ********************************* BOTTOM **********************************
```

Figure 87. Specify Relationship Usage

Specifying Relationship Usage

The relationships needed to extract the full set of data are now present. The listed relationships must be traversed to obtain related data from all tables referenced in the Access Definition.

YES is the default setting for the **Use NEW Relationships** Access Definition parameter. Because this setting may have been changed in training or other use of the product, use the S line command to select relationship RJP. If other Optim relationships are listed, use the U line command to unselect them.

Q1 and Q2

The **Q1** and **Q2** options determine how relationships are traversed. When **Q1** is Yes, Move follows a relationship from child to parent and a parent row is extracted for every child row to ensure the relational integrity of the extracted data. (**Q1** and **Q2** are explained in greater detail in the previous scenarios. Additional information is available in the *Common Elements Manual* under Select Relationships.)

The second option, **Q2**, determines whether additional child rows are extracted if a parent row was extracted in a traversal from child to parent. That is if a parent row is extracted to ensure relational integrity (because a child row was extracted) any additional children of the parent row are also extracted when **Q2** is Yes.

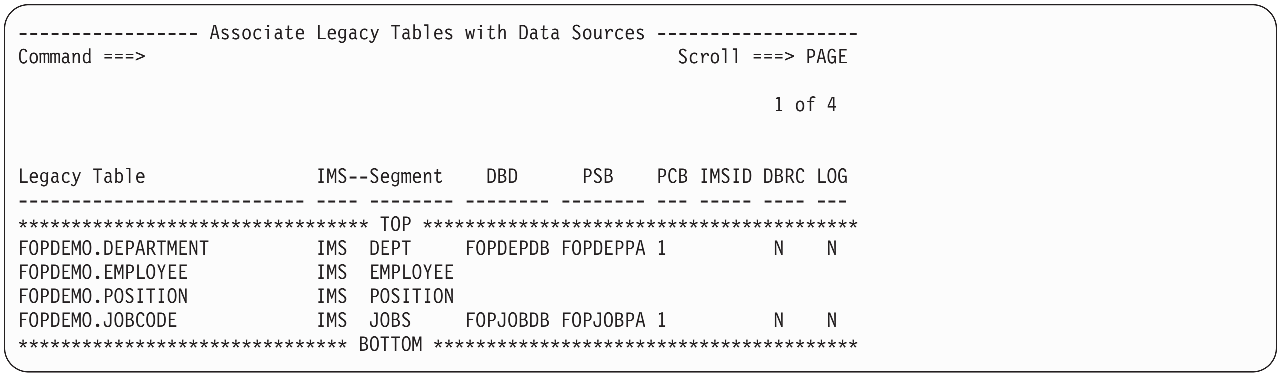
For this scenario:

- The extract begins with the rows from the DEPARTMENT table and proceeds to the related child rows in the EMPLOYEE table, using relationship I1_2.
- Using the IMS relationship I2_3, POSITION records that are children of the extracted EMPLOYEE rows are then extracted. **Q1** must be Y to traverse the relationship I2_3.
- Finally, using the Optim relationship RJP, JOBCODE records that are parents of extracted POSITION records are extracted. **Q1** must be Y to traverse the relationship RJP in this direction.

Note: You can use the SHOW STEPS command to review the extract path (see “SHOW STEPS Command” on page 87). However, if the Access Definition references Legacy Tables, you must associate Legacy Table information before you can use the SHOW STEPS command.

Associate Legacy Tables with Data Sources

After reviewing the relationships to be used in the process, use END to close the Access Definition editor. Before returning to the EXTRACT Process menu, Move prompts for information about the sources for the data described by the Legacy Tables.



```
----------------- Associate Legacy Tables with Data Sources -------------------
Command ===>                                                   Scroll ===> PAGE

                                                                        1 of 4


Legacy Table                IMS--Segment     DBD      PSB    PCB IMSID DBRC LOG
--------------------------- ---- -------- -------- -------- --- ----- ---- ---
********************************* TOP ******************************************
FOPDEMO.DEPARTMENT          IMS  DEPT     FOPDEPDB FOPDEPPA 1           N    N
FOPDEMO.EMPLOYEE            IMS  EMPLOYEE
FOPDEMO.POSITION            IMS  POSITION
FOPDEMO.JOBCODE             IMS  JOBS     FOPJOBDB FOPJOBPA 1           N    N
******************************* BOTTOM *****************************************
```

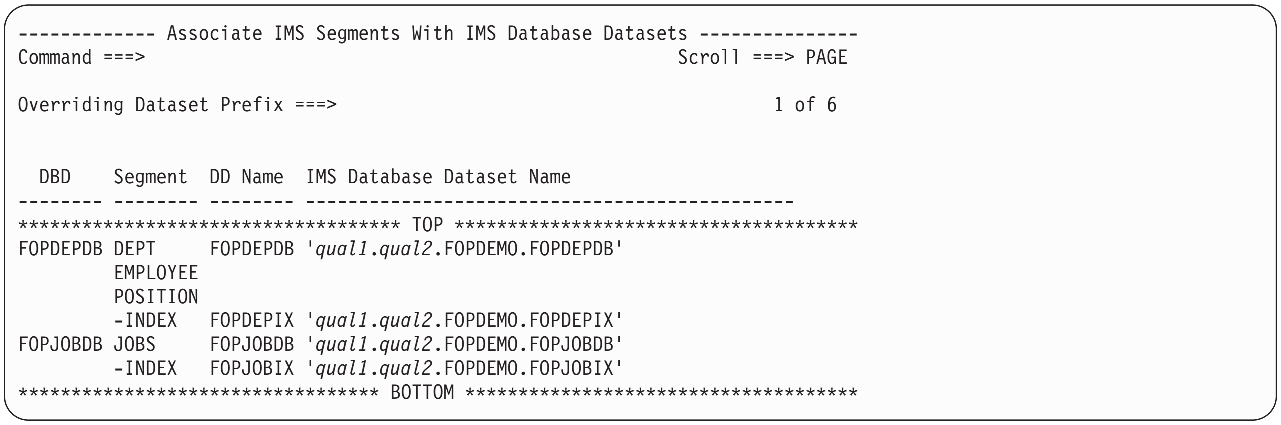
Figure 88. Associate Legacy Tables with Data Sources

Earlier in this scenario, you created a Retrieval Definition, FOPDEMO.FOPDEPDB, which associates a default PSB name (FOPDEPPA) and a PCB number (1) with the FOPDEPDB DBD. Additionally, the Retrieval Definition FOPDEMO.FOPJOBDB, included in the sample data, associates a default PSB name (FOPJOBDB) and a PCB number with the FOPJOBDB DBD.

As shown in the figure, Move inserts the default PSB names, PCB numbers, IMSID and DBRC entries on the Associate Legacy Tables with Data Sources panel. You can edit the defaults to use a different PSB, PCB, and IMS ID, if needed. If an IMSID is not specified, you also can override the displayed DBRC and LOG entries. For this scenario, however, you will use the defaults.

Associate IMS Segments with IMS Database Datasets

When finished, use END to display the Associate IMS Segments with IMS Database Datasets panel, unless you specified an IMSID for the appropriate tables, in which case the previous panel is redisplayed.



```
------------- Associate IMS Segments With IMS Database Datasets ---------------
Command ===>                                                  Scroll ===> PAGE

Overriding Dataset Prefix ===>                                         1 of 6


  DBD    Segment  DD Name  IMS Database Dataset Name
-------- -------- -------- --------------------------------------------
************************************ TOP *************************************
FOPDEPDB DEPT     FOPDEPDB 'qual1.qual2.FOPDEMO.FOPDEPDB'
         EMPLOYEE
         POSITION
         -INDEX   FOPDEPIX 'qual1.qual2.FOPDEMO.FOPDEPIX'
FOPJOBDB JOBS     FOPJOBDB 'qual1.qual2.FOPDEMO.FOPJOBDB'
         -INDEX   FOPJOBIX 'qual1.qual2.FOPDEMO.FOPJOBIX'
********************************** BOTTOM ************************************
```

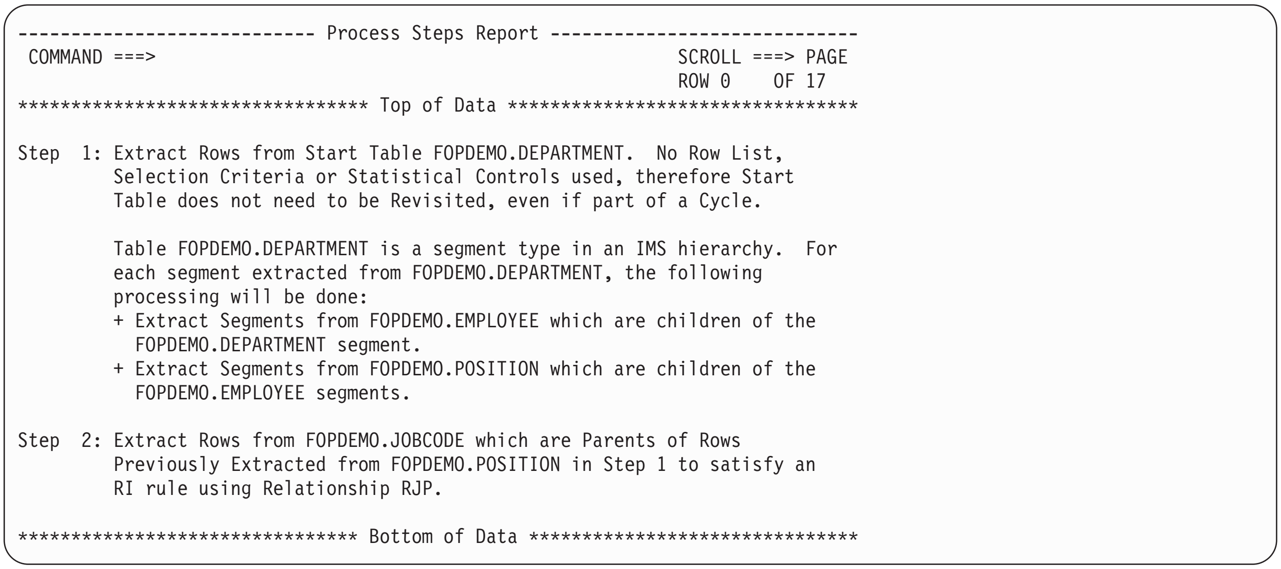
Figure 89. Associate IMS Segments With IMS Database Datasets

When you created the Retrieval Definition FOPDEMO.FOPDEPDB, you associated the listed DDs (FOPDEPDB and FOPDEPIX) with default IMS Database Datasets. Additionally, the Retrieval Definition FOPDEMO.JOBDB, included in the sample data, associates the DDs (FOPJOBDB and FOPJOBIX) with default data sets.

As shown in the figure, Move inserts the default names on the Associate IMS Segments with IMS Database Datasets panel. You can edit the defaults to specify different data sets, if desired. For this scenario, however, you will use the default data sets. Use END to return to the EXTRACT Process menu.

SHOW STEPS Command

Legacy TablesAfter assigning data sources to the Legacy Tables and associating IMS segments to IMS Database Datasets, select Option 2 PATHS on the EXTRACT Process menu to redisplay the Specify Relationship Usage panel. You can use the SHOW STEPS command to review the extract path.



```
---------------------------- Process Steps Report -----------------------------
 COMMAND ===>                                                 SCROLL ===> PAGE
                                                              ROW 0    OF 17
********************************* Top of Data ********************************

Step  1: Extract Rows from Start Table FOPDEMO.DEPARTMENT.  No Row List,
         Selection Criteria or Statistical Controls used, therefore Start
         Table does not need to be Revisited, even if part of a Cycle.

         Table FOPDEMO.DEPARTMENT is a segment type in an IMS hierarchy.  For
         each segment extracted from FOPDEMO.DEPARTMENT, the following
         processing will be done:
         + Extract Segments from FOPDEMO.EMPLOYEE which are children of the
           FOPDEMO.DEPARTMENT segment.
         + Extract Segments from FOPDEMO.POSITION which are children of the
           FOPDEMO.EMPLOYEE segments.

Step  2: Extract Rows from FOPDEMO.JOBCODE which are Parents of Rows
         Previously Extracted from FOPDEMO.POSITION in Step 1 to satisfy an
         RI rule using Relationship RJP.

******************************** Bottom of Data *******************************
```

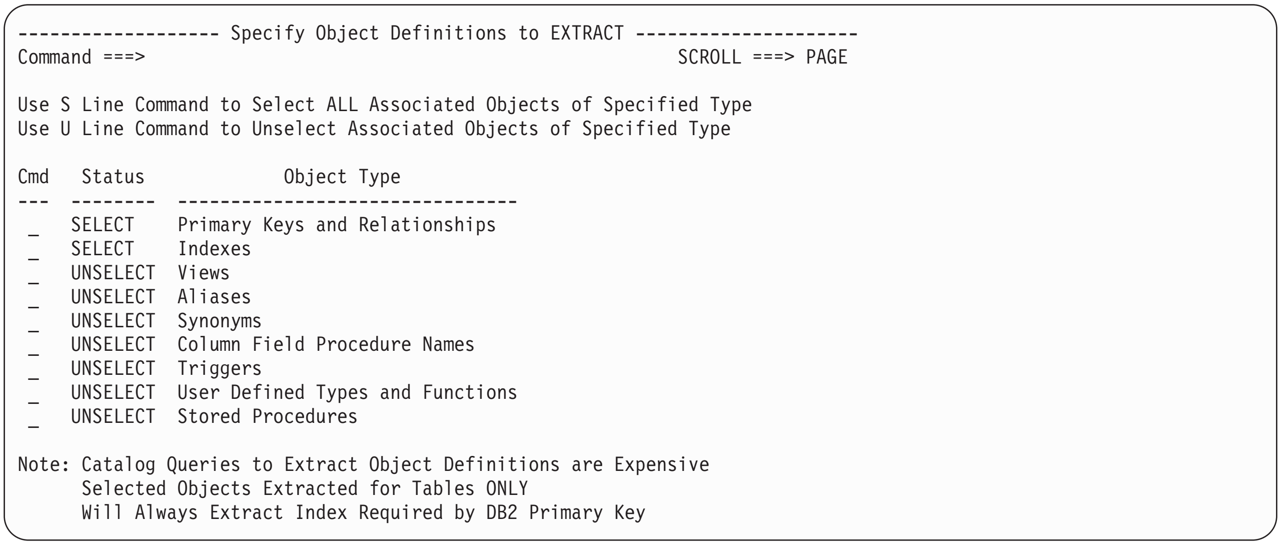
Figure 90. Process Steps Report

After reviewing the steps to be used in the process, use END twice to return to the EXTRACT Process menu.

Object Definitions

Although Move always extracts DB2 and Legacy Table definitions, and subordinate column definitions with the data, you will also need the Optim relationships used to extract the data in this scenario. To extract the relationships, select Option 3 OBJECTS on the EXTRACT Process menu. Move displays the Specify Object Definitions to EXTRACT panel.

By default, primary keys, relationships, and indexes are selected when the panel is displayed. The **Status** for other object definitions is UNSELECT.



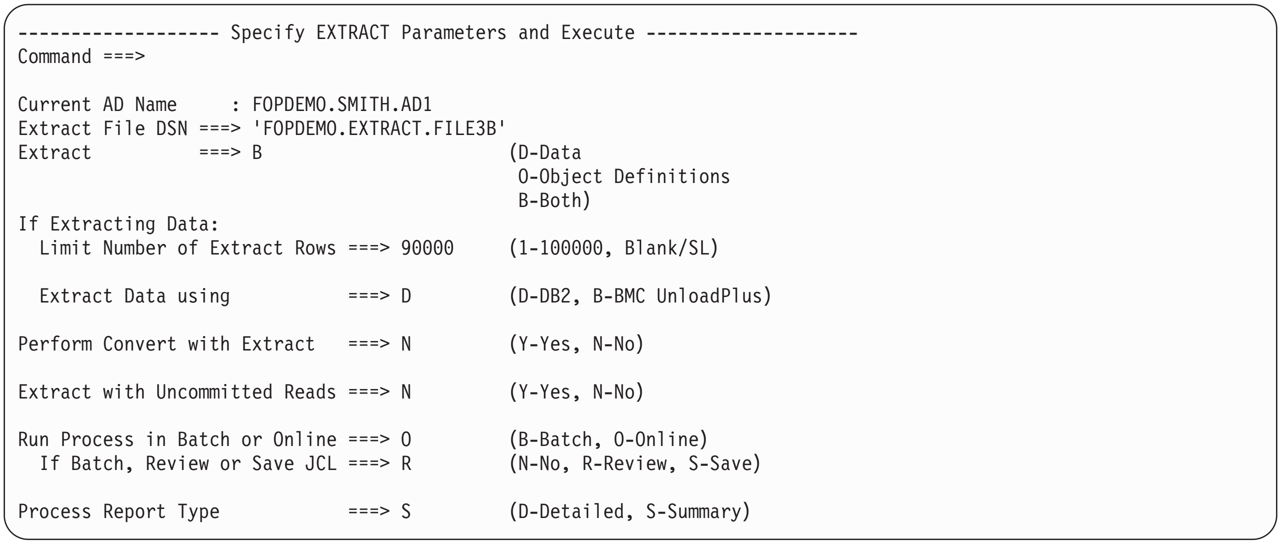
```
-------------------- Specify Object Definitions to EXTRACT ---------------------
Command ===>                                                  SCROLL ===> PAGE

Use S Line Command to Select ALL Associated Objects of Specified Type
Use U Line Command to Unselect Associated Objects of Specified Type

Cmd   Status            Object Type
---  --------  --------------------------------
 _   SELECT    Primary Keys and Relationships
 _   SELECT    Indexes
 _   UNSELECT  Views
 _   UNSELECT  Aliases
 _   UNSELECT  Synonyms
 _   UNSELECT  Column Field Procedure Names
 _   UNSELECT  Triggers
 _   UNSELECT  User Defined Types and Functions
 _   UNSELECT  Stored Procedures

Note: Catalog Queries to Extract Object Definitions are Expensive
      Selected Objects Extracted for Tables ONLY
      Will Always Extract Index Required by DB2 Primary Key
```

Figure 91. Specify Object Definitions to EXTRACT

Although indexes are not required for this scenario, use the default settings for Primary Keys, Relationships, and Indexes. Use END to return to the EXTRACT Process menu.

Perform the Extract

Select Option 4 PERFORM on the EXTRACT Process menu to display the Specify EXTRACT Parameters and Execute panel. To perform the extract, you must provide the Extract File data set name and indicate whether to extract data, object definitions, or both.



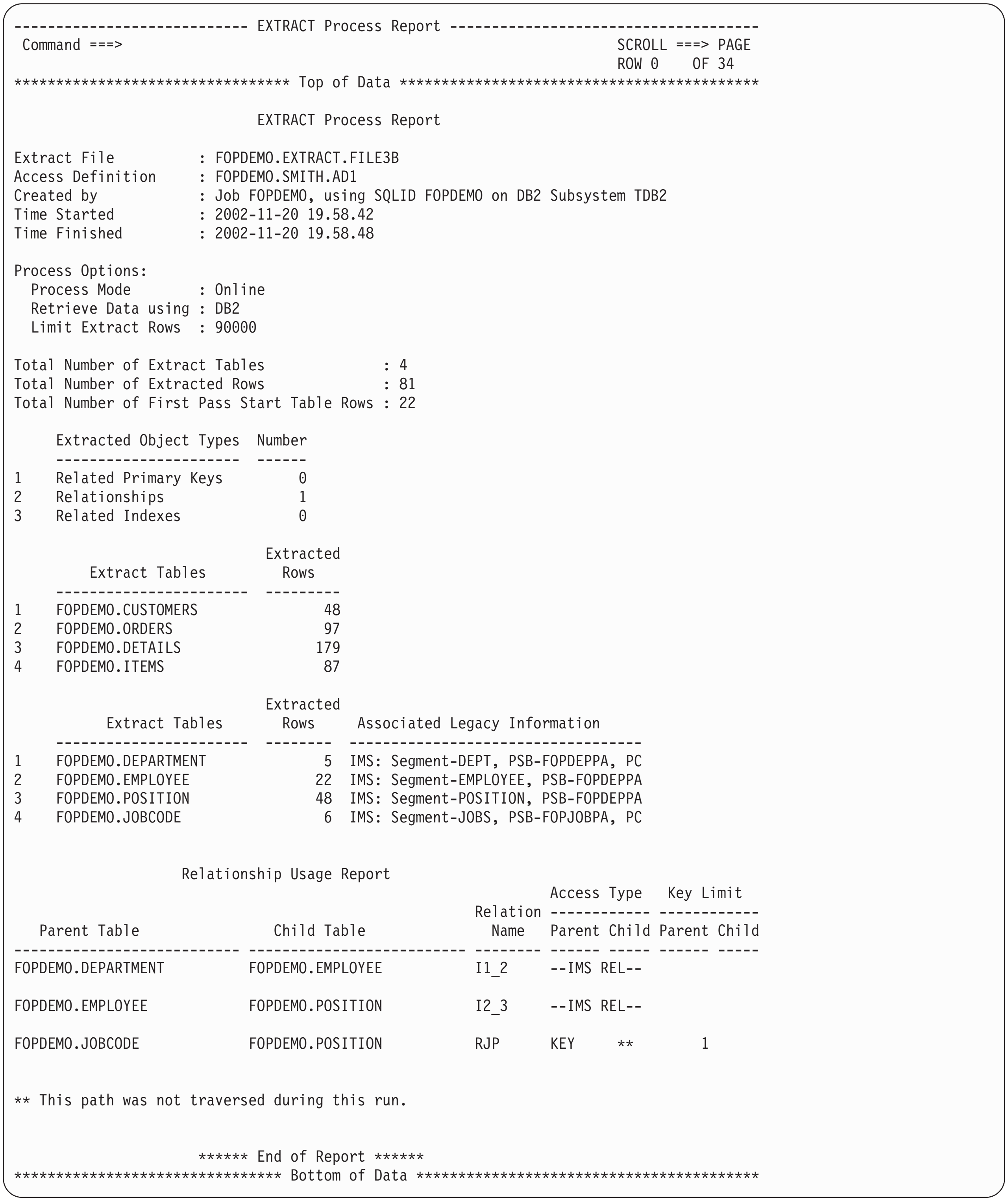
```
-------------------- Specify EXTRACT Parameters and Execute --------------------
Command ===>

Current AD Name     : FOPDEMO.SMITH.AD1
Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE3B'
Extract         ===> B                      (D-Data
                                             O-Object Definitions
                                             B-Both)
If Extracting Data:
  Limit Number of Extract Rows ===> 90000     (1-100000, Blank/SL)

  Extract Data using           ===> D         (D-DB2, B-BMC UnloadPlus)

Perform Convert with Extract   ===> N         (Y-Yes, N-No)

Extract with Uncommitted Reads ===> N         (Y-Yes, N-No)

Run Process in Batch or Online ===> O         (B-Batch, O-Online)
  If Batch, Review or Save JCL ===> R         (N-No, R-Review, S-Save)

Process Report Type            ===> S         (D-Detailed, S-Summary)
```

Figure 92. Specify EXTRACT Parameters and Execute

The name of the Extract File you specified for the previous extract has been profiled and is displayed in **Extract File DSN**. For this scenario, change the name to FOPDEMO.EXTRACT.FILE3B. Both data and object definitions are extracted. Confirm that B is specified at the **Extract** prompt. (Other details of this panel are discussed in "Perform the Extract" on page 19.)

Press ENTER to begin the extract. If the Extract File does not exist, Move prompts for the necessary information and allocates the file. Informational messages are displayed in pop-up windows as the extract progresses.

When finished, Move displays the EXTRACT Process Report.



```
---------------------------- EXTRACT Process Report -------------------------------------
 Command ===>                                                           SCROLL ===> PAGE
                                                                        ROW 0    OF 34
********************************* Top of Data *******************************************

                             EXTRACT Process Report

Extract File          : FOPDEMO.EXTRACT.FILE3B
Access Definition     : FOPDEMO.SMITH.AD1
Created by            : Job FOPDEMO, using SQLID FOPDEMO on DB2 Subsystem TDB2
Time Started          : 2002-11-20 19.58.42
Time Finished         : 2002-11-20 19.58.48

Process Options:
  Process Mode        : Online
  Retrieve Data using : DB2
  Limit Extract Rows  : 90000

Total Number of Extract Tables              : 4
Total Number of Extracted Rows              : 81
Total Number of First Pass Start Table Rows : 22

     Extracted Object Types  Number
     ----------------------  ------
1    Related Primary Keys         0
2    Relationships                1
3    Related Indexes              0

                              Extracted
         Extract Tables         Rows
     -----------------------  ---------
1    FOPDEMO.CUSTOMERS               48
2    FOPDEMO.ORDERS                  97
3    FOPDEMO.DETAILS                179
4    FOPDEMO.ITEMS                   87

                              Extracted
           Extract Tables       Rows     Associated Legacy Information
     -----------------------  --------  -----------------------------------
1    FOPDEMO.DEPARTMENT              5  IMS: Segment-DEPT, PSB-FOPDEPPA, PC
2    FOPDEMO.EMPLOYEE               22  IMS: Segment-EMPLOYEE, PSB-FOPDEPPA
3    FOPDEMO.POSITION               48  IMS: Segment-POSITION, PSB-FOPDEPPA
4    FOPDEMO.JOBCODE                 6  IMS: Segment-JOBS, PSB-FOPJOBPA, PC


                    Relationship Usage Report
                                                                Access Type   Key Limit
                                                       Relation ------------ ------------
   Parent Table              Child Table                 Name   Parent Child Parent Child
--------------------------- -------------------------- -------- ------ ----- ------ -----
FOPDEMO.DEPARTMENT          FOPDEMO.EMPLOYEE           I1_2     --IMS REL--

FOPDEMO.EMPLOYEE            FOPDEMO.POSITION           I2_3     --IMS REL--

FOPDEMO.JOBCODE             FOPDEMO.POSITION           RJP      KEY     **        1


** This path was not traversed during this run.


                      ****** End of Report ******
******************************** Bottom of Data *****************************************
```

Figure 93. EXTRACT Process Report

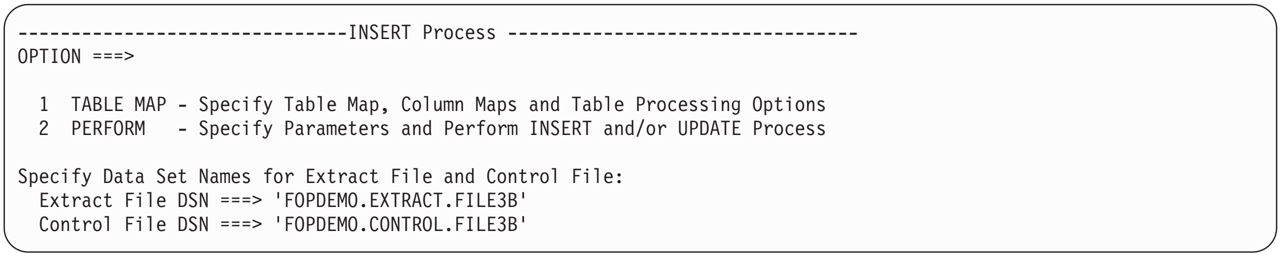
Note that when Legacy Tables are involved in an Extract Process, the **Associated Legacy Information** is listed in the EXTRACT Process Report. After reviewing the report, use END twice to return to the Data Migration menu. Move prompts you to save the edited Access Definition. Press ENTER to save the Access Definition under its current name and return to the Data Migration menu.

The Extract Process is complete.

Defining the Destination

With the conclusion of the Extract Process, all the data and necessary object definitions are copied to the Extract File, which is the source of data for the Insert Process.

Select Option 2 INSERT on the Data Migration menu to specify the destination. (Details of this panel are discussed in “Defining the Destination” on page 22.)



```
------------------------------INSERT Process --------------------------------
OPTION ===>

  1  TABLE MAP - Specify Table Map, Column Maps and Table Processing Options
  2  PERFORM   - Specify Parameters and Perform INSERT and/or UPDATE Process

Specify Data Set Names for Extract File and Control File:
  Extract File DSN ===> 'FOPDEMO.EXTRACT.FILE3B'
  Control File DSN ===> 'FOPDEMO.CONTROL.FILE3B'
```

Figure 94. Scenario 3b – INSERT Process

To create the data you need to test your application in this scenario, you will insert the extracted employee data into empty IMS destination data sets.

As in previous scenarios, the destination table names match the source table names and are uniquely identified by the Creator ID. The primary keys and relationships and the indexes from the production database are copied into the test database.

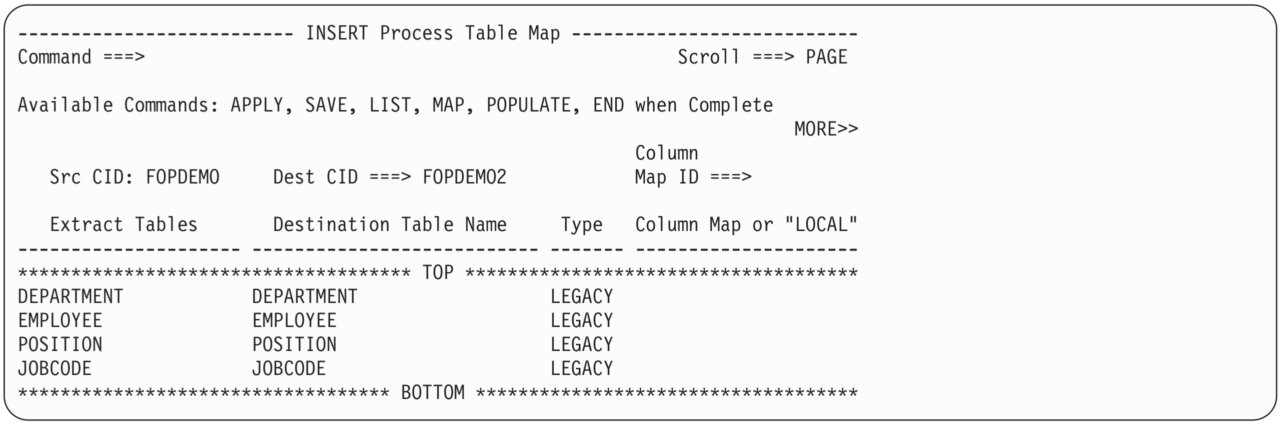
Move provides the name of the current or most recent Extract File automatically. For this scenario, FOPDEMO.EXTRACT.FILE3B is the name of the Extract File used to create the test database. If this name is not in **Extract File DSN**, enter it now. The name of the current or most recent Control File is also provided automatically. For this scenario, type 'FOPDEMO.CONTROL.FILE3B' at the **Control File DSN** prompt.

Specify the Tables

Select Option 1 TABLE MAP from the INSERT Process menu to display the INSERT Process Table Map panel. (A more detailed discussion of this panel is provided in “Specify the Tables” on page 23.)

Modify Table Map

Enter the **Dest CID**, FOPDEMO2. The sample data includes Legacy Tables with this Creator ID and matching names. The destination **Type** for all tables is LEGACY.



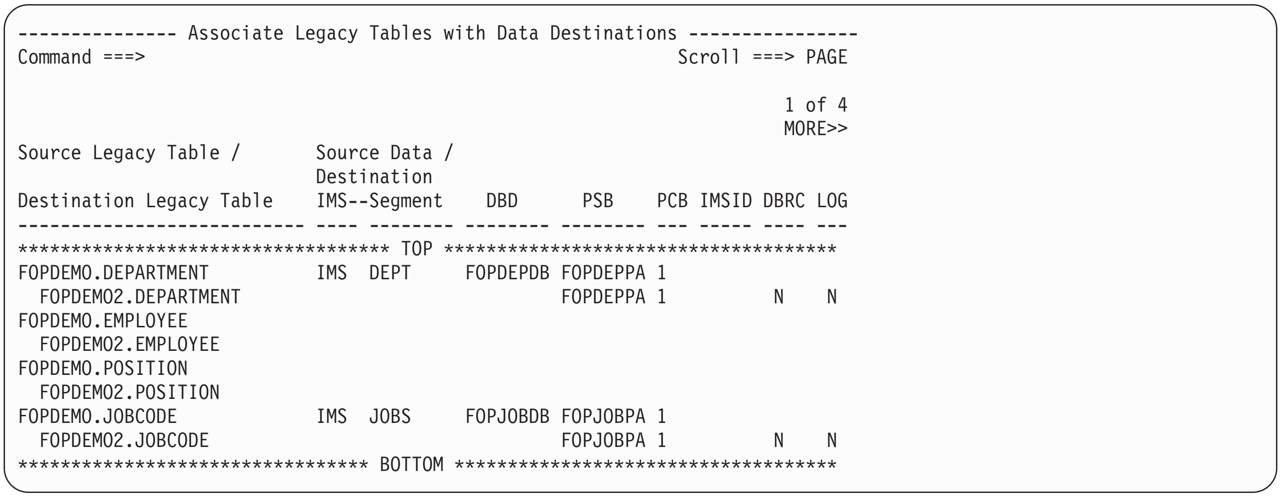
```
------------------------- INSERT Process Table Map ---------------------------
Command ===>                                              Scroll ===> PAGE

Available Commands: APPLY, SAVE, LIST, MAP, POPULATE, END when Complete
                                                                     MORE>>
                                                          Column
   Src CID: FOPDEMO     Dest CID ===> FOPDEMO2            Map ID ===>

   Extract Tables       Destination Table Name     Type   Column Map or "LOCAL"
--------------------- -------------------------- ------- ---------------------
************************************* TOP ************************************
DEPARTMENT            DEPARTMENT                 LEGACY
EMPLOYEE              EMPLOYEE                   LEGACY
POSITION              POSITION                   LEGACY
JOBCODE               JOBCODE                    LEGACY
********************************** BOTTOM ***********************************
```

Figure 95. INSERT Process Table Map

If you leave **Destination Table Name** blank, data is not inserted into the table. For example, to create new legacy files for DEPARTMENT and EMPLOYEE without inserting data into the POSITION and JOBCODE segments, you would remove POSITION and JOBCODE from **Destination Table Name**. However, for this scenario, all tables are included in the Insert Process.

Associate Legacy Tables with Data Destinations

When you use END, the Associate Legacy Tables with Data Destinations panel is displayed. The Retrieval Definitions FOPDEMO2.FOPDEPDB and FOPDEMO2.FOPJOBDB, included with the sample data, associate default PSBs and PCBs with the destination Legacy Tables.



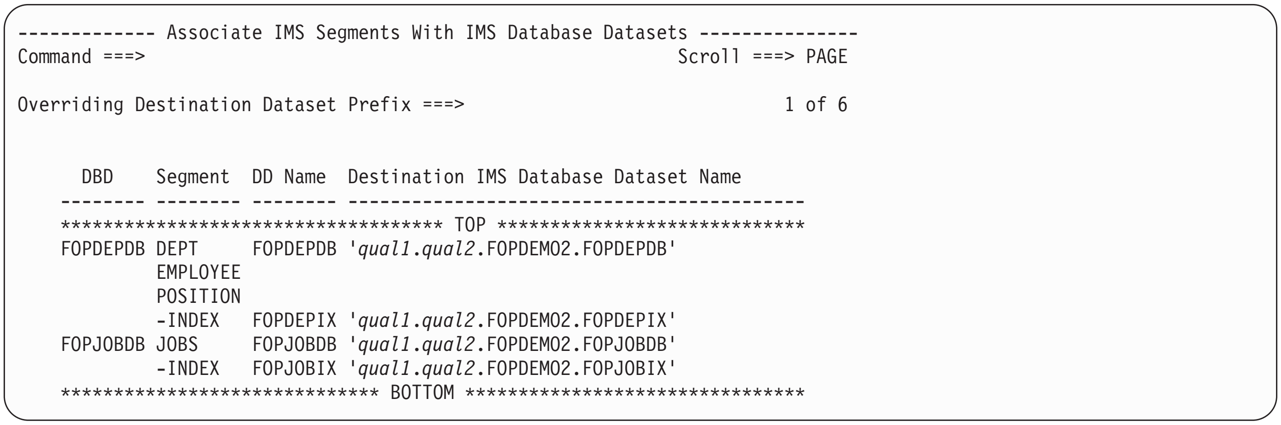
```
--------------- Associate Legacy Tables with Data Destinations ----------------
Command ===>                                                  Scroll ===> PAGE

                                                                        1 of 4
                                                                        MORE>>
Source Legacy Table /       Source Data /
                            Destination
Destination Legacy Table    IMS--Segment    DBD      PSB    PCB IMSID DBRC LOG
-------------------------- ---- -------- -------- -------- --- ----- ---- ---
*********************************** TOP *************************************
FOPDEMO.DEPARTMENT          IMS  DEPT     FOPDEPDB FOPDEPPA 1
  FOPDEMO2.DEPARTMENT                              FOPDEPPA 1          N    N
FOPDEMO.EMPLOYEE
  FOPDEMO2.EMPLOYEE
FOPDEMO.POSITION
  FOPDEMO2.POSITION
FOPDEMO.JOBCODE             IMS  JOBS     FOPJOBDB FOPJOBPA 1
  FOPDEMO2.JOBCODE                                 FOPJOBPA 1          N    N
********************************* BOTTOM ************************************
```

Figure 96. Associate Legacy Tables with Data Destinations

As shown in the figure, Move inserts the default PSB names, PCB numbers, DBRC and IMS Log entries on the Associate Legacy Tables with Data Destinations panel. You can edit the defaults, if desired. For this scenario, however, you will use the defaults.

Associate IMS Segments with IMS Database Datasets

When finished, use END to display the Associate IMS Segments with IMS Database Datasets panel, unless you specified an IMSID for the appropriate tables, in which case the previous panel is redisplayed. In the following example, FOPDEMO2.FOPDEPDB and FOPDEMO2.FOPJOBDB, the Retrieval Definitions included with the sample data, associate default IMS Database Datasets with each DD.



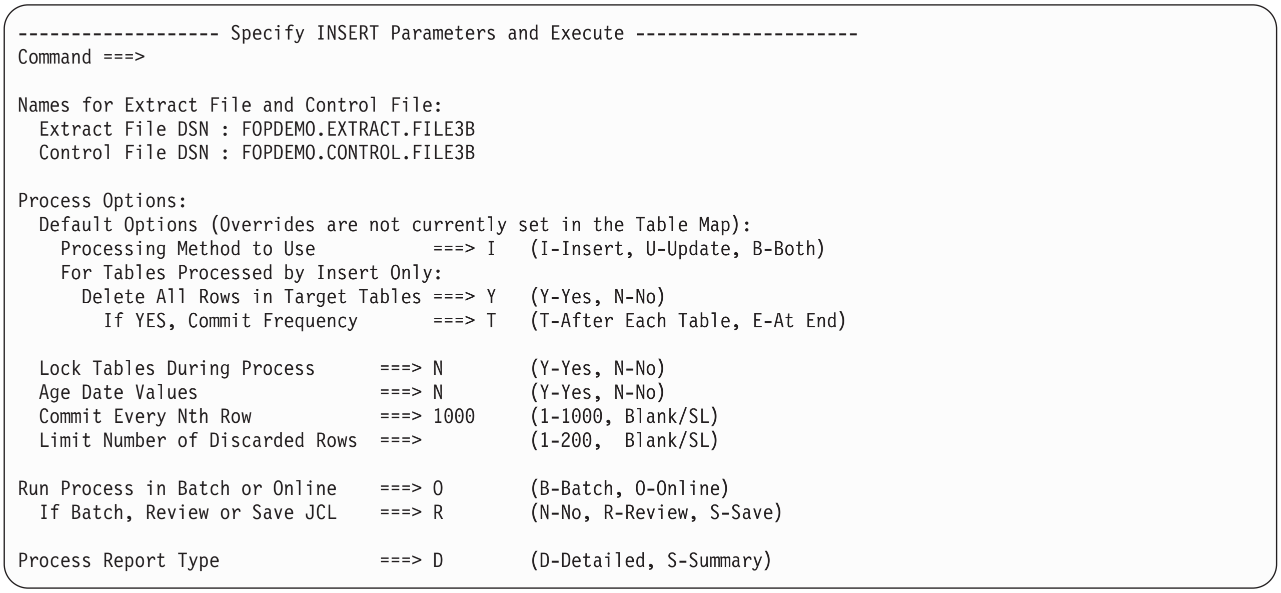
```
------------- Associate IMS Segments With IMS Database Datasets ---------------
Command ===>                                                  Scroll ===> PAGE

Overriding Destination Dataset Prefix ===>                              1 of 6


      DBD    Segment  DD Name  Destination IMS Database Dataset Name
    -------- -------- -------- ------------------------------------------
    ************************************* TOP ******************************
    FOPDEPDB DEPT     FOPDEPDB 'qual1.qual2.FOPDEMO2.FOPDEPDB'
             EMPLOYEE
             POSITION
             -INDEX   FOPDEPIX 'qual1.qual2.FOPDEMO2.FOPDEPIX'
    FOPJOBDB JOBS     FOPJOBDB 'qual1.qual2.FOPDEMO2.FOPJOBDB'
             -INDEX   FOPJOBIX 'qual1.qual2.FOPDEMO2.FOPJOBIX'
    ****************************** BOTTOM ********************************
```

Figure 97. Associate IMS Segments With IMS Database Datasets

As shown in the figure, Move inserts the default data set names on the Associate IMS Segments with IMS Database Datasets panel. You can edit the defaults to specify different data set names, if desired. For this scenario, however, you will use the default data sets. Use END to return to the INSERT Process menu.

Perform the Insert

Select Option 2 PERFORM on the INSERT Process menu. Since all the destination Legacy Tables exist, Move displays the Specify INSERT Parameters and Execute panel.

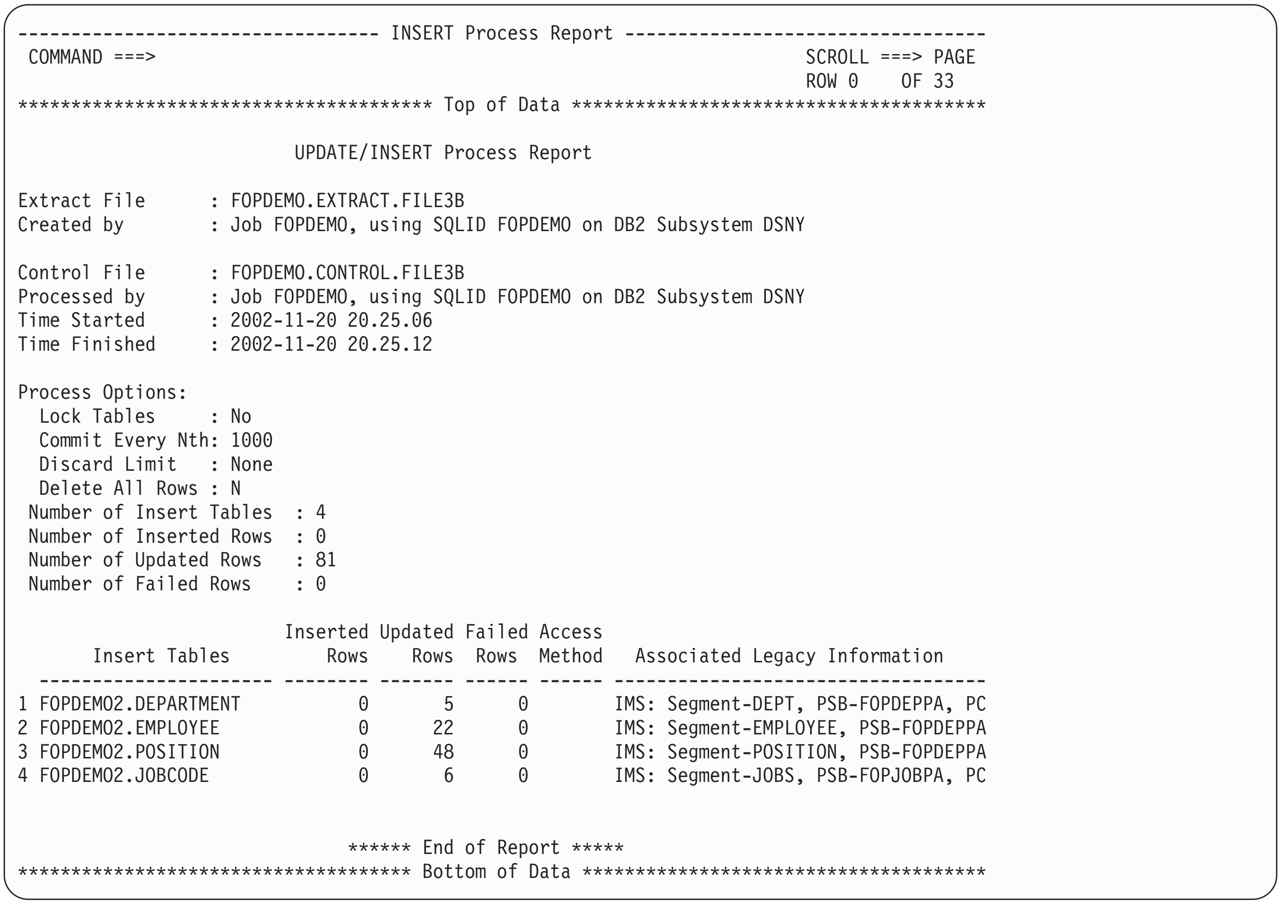


```
------------------- Specify INSERT Parameters and Execute ---------------------
Command ===>

Names for Extract File and Control File:
  Extract File DSN : FOPDEMO.EXTRACT.FILE3B
  Control File DSN : FOPDEMO.CONTROL.FILE3B

Process Options:
  Default Options (Overrides are not currently set in the Table Map):
    Processing Method to Use          ===> I   (I-Insert, U-Update, B-Both)
    For Tables Processed by Insert Only:
      Delete All Rows in Target Tables ===> Y   (Y-Yes, N-No)
        If YES, Commit Frequency       ===> T   (T-After Each Table, E-At End)

  Lock Tables During Process      ===> N       (Y-Yes, N-No)
  Age Date Values                 ===> N       (Y-Yes, N-No)
  Commit Every Nth Row            ===> 1000    (1-1000, Blank/SL)
  Limit Number of Discarded Rows  ===>         (1-200,  Blank/SL)

Run Process in Batch or Online    ===> O       (B-Batch, O-Online)
  If Batch, Review or Save JCL    ===> R       (N-No, R-Review, S-Save)

Process Report Type               ===> D       (D-Detailed, S-Summary)
```

Figure 98. Specify INSERT Parameters and Execute

Specify I (Insert) as **Processing Method to Use**. When creating a test database, you may want to clear any data from existing data sets before inserting new data. Type Y to **Delete All Rows in Target Tables** to be certain that no data is currently in any IMS data sets and type T as the **Commit Frequency**. Press ENTER to execute the Insert Process.

When the Insert Process is complete, Move automatically displays an INSERT Process Report documenting the inserted data. This report is similar in format to the EXTRACT Process Report.



```
---------------------------------- INSERT Process Report ---------------------------------
 COMMAND ===>                                                          SCROLL ===> PAGE
                                                                       ROW 0    OF 33
*************************************** Top of Data **************************************

                          UPDATE/INSERT Process Report

Extract File      : FOPDEMO.EXTRACT.FILE3B
Created by        : Job FOPDEMO, using SQLID FOPDEMO on DB2 Subsystem DSNY

Control File      : FOPDEMO.CONTROL.FILE3B
Processed by      : Job FOPDEMO, using SQLID FOPDEMO on DB2 Subsystem DSNY
Time Started      : 2002-11-20 20.25.06
Time Finished     : 2002-11-20 20.25.12

Process Options:
  Lock Tables     : No
  Commit Every Nth: 1000
  Discard Limit   : None
  Delete All Rows : N
 Number of Insert Tables  : 4
 Number of Inserted Rows  : 0
 Number of Updated Rows   : 81
 Number of Failed Rows    : 0

                         Inserted Updated Failed Access
       Insert Tables         Rows    Rows  Rows  Method   Associated Legacy Information
  ---------------------- -------- ------- ------ ------ ----------------------------------
1 FOPDEMO2.DEPARTMENT           0       5      0        IMS: Segment-DEPT, PSB-FOPDEPPA, PC
2 FOPDEMO2.EMPLOYEE             0      22      0        IMS: Segment-EMPLOYEE, PSB-FOPDEPPA
3 FOPDEMO2.POSITION             0      48      0        IMS: Segment-POSITION, PSB-FOPDEPPA
4 FOPDEMO2.JOBCODE              0       6      0        IMS: Segment-JOBS, PSB-FOPJOBPA, PC


                               ****** End of Report *****
************************************* Bottom of Data *************************************
```

Figure 99. INSERT Process Report

Summary

This section discussed how to create a Legacy Table, a user-defined object used to describe IMS, VSAM, and sequential data, and incorporate it into the Move processes used to extract, insert, and create relational data and objects.

The Legacy Table uses copybooks to describe the legacy data and map it to a pseudo DB2 table format, which allows you to process legacy data as you would data in DB2 tables.

- In Scenario 3a, VSAM Legacy Tables, you created a VSAM Legacy Table, used in Move processes to extract VSAM and sequential data. Additionally, you defined the default Data Source information for this VSAM Legacy Table.
- In Scenario 3b, IMS Legacy Tables, you first created an Environment Definition, referencing the libraries required by Move to create IMS Legacy Tables and Retrieval Definitions. Then you created an IMS Legacy Table to be used in Move processes to extract IMS data. Additionally, you defined a Retrieval Definition, providing the name of the default PSB, number of the default PCB, IMS ID, and data set for the segments in the DBD.

In both scenarios, after creating the Legacy Table, you incorporated it and other Legacy Tables in an Access Definition, established Optim Relationships between pairs of tables on the Table List (Legacy and DB2), associated Legacy Tables with the appropriate legacy files, and copied the related DB2 and legacy data to an Extract File.

Data in the Extract File was then inserted into destination tables and files.

- In Scenario 3a, extracted sequential legacy data associated with one Legacy Table was inserted into a DB2 table, new legacy files were created, and VSAM legacy data associated with other Legacy Tables was inserted into the new files.
- In Scenario 3b, extracted IMS legacy data associated with the Legacy Tables was inserted into sample destination data sets, created during installation.

Chapter 6. Conclusion

This manual has proceeded through a Move session to create a test database by specifying the source data and the destination of that data. Several key facilities have been discussed, as summarized in this topic.

- Specifying the source data using:
  - Table Lists
  - Selection Criteria
  - Point-and-Shoot Facility
  - Traversal Paths based on Relationships
  - Legacy Tables
- Specifying the destination tables using:
  - New Tables
  - Table Maps
  - Column Maps
  - Insert and Update

Details on these facilities and more are documented in the *Move User Manual* and the *Common Elements Manual*.

After you have created the test database, you can use Access to incorporate special test data into your test database. This relational editor allows you to display and edit all related data from multiple tables simultaneously.

Further, you can evaluate the results of testing your application using Compare to compare the “before” and “after” images of the test data. This comparison helps you focus on the changes to all the related data, as well as the individual rows. For example, when comparing the “before” version of your test data in the original Extract File with the “after” version of the data in the test database, you can direct your attention to the related changes in context.

You can also create test data using Optim Legacy to extract data from legacy files and migrate it to other legacy files, or DB2 tables.

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