Operating System/2
Extended Edition
Version 1.3

User's Guide
Volume 3: Database Manager

First Edition (December 1990)

This publication 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.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorised Dealer, your IBM Marketing Representative or your IBM Retailer.

No part of this publication may be reproduced or distributed in any form or by any means without prior permission in writing from the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1990.
All rights reserved.

Special Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

Application System/400
C/2
Etherand
FORTRAN/2
OS/2
Personal Computer AT
PS/2
SAA
Systems Application Architecture
AS/400
Personal Decision Series
IBM
Operating System/2
PC Network
AT
Presentation Manager
COBOL/2
Display Write

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Lotus 1-2-3	Lotus Development Corporation
Microsoft	Microsoft Corporation
Symphony	Lotus Development Corporation
Lotus	Lotus Development Corporation

About This Volume

The *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 3: Database Manager* (referred to as the *User's Guide, Volume 3: Database Manager* in the remainder of this volume) provides information for you to accomplish the basic tasks provided by the Database Manager component of IBM* Operating System/2* Extended Edition (referred to as the OS/2* program in the remainder of this volume).

For example, this volume contains information and procedures for using User Profile Management, starting and stopping Query Manager, logging on and logging off your workstation, querying a database, formatting reports, defining tables and views, editing data in tables, maintaining your database, and creating a customized interface.

As you are working through these basic tasks, using the procedures provided in this volume, keep in mind that if a step requires you to press two keys such as the Form (Shift+F5) key, you must press and hold the Shift key, and then press the F5 key.

Before You Begin

You should have already installed the OS/2 program as described in *IBM Operating System/2 Extended Edition Version 1.3 Getting Started* (referred to as *Getting Started* in the remainder of this volume). It is recommended that you view the online overview, *Introducing OS/2*, before continuing in this volume.

The illustrations in this volume may not be identical to what is displayed on your screen.

Who Should Use This Volume

The *User's Guide, Volume 3: Database Manager* is written primarily for you, the computer user, who needs to use databases.

You should already have installed the OS/2 program as described in *Getting Started* and become familiar with the Extended Edition interface by reviewing the information provided in that book, and viewing the online overview, *Introducing OS/2.*

Experienced users and programmers can also use this volume along with the more advanced information in the *OS/2 Command Reference.*

Administrator Roles in the OS/2 Program

The following definitions define four administrator roles. These four roles can be filled by one or more individuals, depending upon your particular organization. For instance, the Communications Manager system administrator and network administrator may be the same person or different people, and they may be called by different titles in your organization. In some cases, an organization may have many people responsible for different aspects of one of the following roles listed here.

These definitions of roles are provided to clarify how they are used throughout the OS/2 program library:

Communications Manager system administrator (referred to as system administrator): This role, performed by technically skilled individuals, includes helping users of the OS/2 program plan for and install, configure, and use the functions of the Communications Manager component of the OS/2 program.

Network administrator: This role includes installing, managing, controlling, and configuring a network or a local area network. The network administrator defines resources to be shared and user access to the shared resources and determines the type of access those users have.

Database Manager system administrator: This role, performed by technically skilled individuals familiar with databases, includes helping users plan for, install, configure, and use the Database Manager component of the OS/2 program. Tasks include creating and controlling databases, deciding where databases will be stored, establishing users and groups, and helping users understand database server and requester concepts and use. To perform many of these tasks, an individual must have *SYSADM* (system administrator) authority for Database Manager.

Database administrator: This role includes designing, developing, operating, safeguarding, and maintaining a single database.

Service coordinators: This role, performed by technically skilled individuals, includes helping users of the OS/2 program with problem determination and is responsible for ensuring problem determination support for the OS/2 program and is the registered contact for IBM.

How This Volume Is Structured

Chapter 1 through Chapter 16 contains information and steps to:

- Begin to use Database Manager by using User Profile Management, installing the sample database, starting and stopping Query Manager, logging on and logging off your workstation, and opening a database. The sample database can be used throughout this volume to aid you in understanding how to use Query Manager.
- Write queries, if you already have tables in your database.
- Generate and format reports, if you already have tables in your database.
- Define tables for your database and specify indexes for tables.
- Edit data.
- Create views in the database.
- Maintain your database using Query Manager System Tools, Database Tools, and Object Tools to import and export data, back up a database to preserve the data, and create different profiles for running Database Manager. You can control access to Database Manager databases, tables and views, through use of Authorizations.
- Use SQL statements within Query Manager to manipulate and manage information stored in a database.
- Use Query Manager commands, such as RUN and PRINT, within Query Manager from a command line or within menus, panels, and procedures.

- Customize your application by designing an interface to access your database and reflecting the specific requirements of a specific application through menus, panels, and procedures.
- Use variables with the menus, panels, and procedures in a customized interface.

Appendix A through Appendix K contains information about:

- Naming rules for Database Manager as well as a list of SQL reserved words
- Personalizing your Query Manager start-up, either through the Group – Main window or by typing the command to start Query Manager from the OS/2 command prompt
- Importing and exporting data
- Downloading and importing data from a DB2 or SQL/DS table
- Migrating databases from previous versions of the OS/2 program to Version 1.3
- Resolving Database Manager messages that do not have online help
- Using edit codes
- Writing procedures using Procedures Language 2/REXX.

An index is located at the back of this volume.

About the Online Information

You can display additional information about the OS/2 program on your screen after installation. This *online* information is available in the following form.

- An online overview about the OS/2 program called *Introducing OS/2*

  Introducing OS/2 contains topics describing the components of the OS/2 program. It is recommended that you use this overview to familiarize yourself with the OS/2 program.

- The *OS/2 Command Reference*

  This online reference provides you with information about the base operating system commands, configuration commands, Communications Manager commands, Query Manager commands, Database Services commands, Query Manager procedure statements, Query Manager Callable Interface commands, and User Profile Management commands.

- The *LAN Command Reference*

  This online reference provides you with information about the LAN commands.

- The *Online Reference* for LAN Requester and IBM OS/2 LAN Server

  The *Online Reference* for LAN Requester and OS/2 LAN Server provides detailed information about LAN requester and server functions.

  Note: LAN Requester is a component of the OS/2 program. If you want to use LAN functions, you must also have OS/2 LAN Server installed on a workstation in the network. OS/2 LAN Server is available separately from the OS/2 program.

- Comprehensive, context-sensitive Help information.

  Context-sensitive help provides information to help you use the OS/2 program. You can select **F1 = Help** or Help with a pointing device (for example, a mouse) or press the Help (F1) key to display information related to the current task, panel, menu, item, or message.

Prerequisite Publications

Getting Started and *IBM Operating System/2 Version 1.3 Database Manager Exercises* (referred to as *Database Manager Exercises* for the remainder of the volume) are the only prerequisite publications for operations in this volume.

Getting Started contains an introduction to the OS/2 program and provides the steps and basic information needed to install it, as well as how to use the online overview, *Introducing OS/2*, and the *OS/2 Command Reference*.

Database Manager Exercises provide step by step instructions on performing basic tasks with Database Manager and Query Manager. You should complete the tasks in *Database Manager Exercises* before using this guide.

Related Publications

Related publications are listed on the following pages. For additional information on these publications, refer to the *IBM Operating System/2 Extended Edition Version 1.3 Guide to Information*. For information on ordering these publications, contact your IBM authorized dealer or your IBM marketing representative.

The following are related publications that are included in the Extended Edition library:

- *IBM Operating System/2 Extended Edition Version 1.3 Guide to Information*
- *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 1: Base Operating System*
- *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 2: Communications Manager and LAN Requester*
- *IBM Operating System/2 Extended Edition Version 1.3 Structured Query Language (SQL) Concepts*
- *IBM Operating System/2 Extended Edition Version 1.3 Keyboard Layouts*
- *IBM Operating System/2 Extended Edition Version 1.3 Terminal Emulation Keyboard Templates*
- *IBM Operating System/2 Extended Edition Version 1.3 Glossary.*

The following is a related publication for the Communications Manager component of the Extended Edition library that you can purchase separately:

> *IBM Operating System/2 Extended Edition Version 1.3 System Administrator's Guide for Communications.*

The following are related publications for the Database Manager component of the Extended Edition library that you can purchase separately:

- *IBM Operating System/2 Extended Edition Version 1.3 Database Manager Programming Guide and Reference*
- *&sqlref.*
- *IBM Operating System/2 Extended Edition Version 1.3 Database Manager Administrator's Guide.*

The following are related publications for Extended Edition that you can purchase separately:

- *IBM Operating System/2 Extended Edition Version 1.3 Problem Determination Guide for the Service Coordinator*
- *IBM Operating System/2 Extended Edition Version 1.3 Commands Reference.*

The following is a related publication for the Standard Edition library that you can purchase separately:

IBM Operating System/2 Version 1.2 Programming Tools and Information.

The following is a related publication for the IBM Systems Application Architecture that you can purchase separately:

IBM Systems Application Architecture Common Programming Interface Database Reference.

The following are related publications for the Procedures Language 2/REXX programs that you can purchase separately:

- *OS/2 IBM Operating System/2 Procedures Language 2/REXX User's Guide*
- *OS/2 IBM Operating System/2 Procedures Language 2/REXX Reference.*

Contents

Chapter 1. User Profile Management

This chapter describes how User Profile Management is used to control access to Database Manager and how the user IDs and passwords created with User Profile Management interact with Database Manager.

Refer to *IBM Operating System/2 Extended Edition Version 1.3 Database Manager Administrator's Guide* (referred to as the *Database Manager Administrator's Guide* in the remainder of this volume) for more information and steps to creating and managing User Profile Management user IDs and passwords.

About User Profile Management

User Profile Management provides a way for you to log on and log off your workstation to access Database Manager. User Profile Management verifies the user ID and password you type in the User Profile Management logon panel against a list of authorized user IDs defined for that workstation. User Profile Management also provides a way for you to create, view, and manage individual and group user IDs.

You must have an authorized user ID defined in User Profile Management to access Database Manager. This means that you define authorized user IDs in User Profile Management on each workstation you use, unless you are using the OS/2 LAN Server Program Version 1.3 (a separately available program that works in conjunction with the LAN Requester component of the OS/2 program). For details, see the LAN Server publications.

User Profile Management is automatically installed at each workstation as part of the OS/2 program installation process. Follow the steps described in *Getting Started* for logging on to your workstation through User Profile Management and defining a user ID for the first time after installation.

Authorization

User Profile Management provides one of three *user types*: administrator, local administrator, or user. Each of these user types allows you a different level of authority within User Profile Management.

As an *administrator* user type in User Profile Management, you can perform user and group management tasks. These tasks include adding, updating, and erasing user and group IDs. You can specify and change the user type and password option, and allow or deny access authority for each user ID. You can view every user or group ID profile defined for the individual workstation. When you are assigned an administrator user type in User Profile Management, you automatically have SYSADM (system administrator) authority for all Database Manager databases, tables, views, and other Query Manager objects. This means, for example, that you can create databases, grant access to these databases to other users, catalog remote databases that you want to access, and change the database configuration file (all of which require SYSADM authority).

As a *local administrator*, you can view your profile, change your password, and change your user comments. You do not have administrator authority to perform User Profile Management management tasks, but you do have SYSADM (system administrator) authority for all Database Manager databases, tables, views, and other Query Manager objects on your particular workstation.

As a *user*, you have the same authority within User Profile Management as a local administrator. You do not, however, have SYSADM (system administrator) authority within Database Manager, and you cannot, for example, create a database.

If you are adding a user ID, it is important to consider that user's Database Manager access needs when assigning the user type. Refer to "Database Manager Authorizations" on page 2-3 for additional information about Database Manager authorization levels.

To Log On to User Profile Management

1. If the Desktop Manager is not displayed, press the Task List (TskLst) (Ctrl+Esc) key and select **Desktop Manager** from the list of active programs displayed.
2. Select **User Profile Management Services** from the Desktop Manager window. The Group – User Profile Management Services menu is displayed.
3. Select **Logon** from the Group – User Profile Management Services menu. The Logon panel is displayed.
4. Type your user ID and password (if required) in the Logon panel and select Enter. The Group – User Profile Management Services menu is displayed again.

Hints:

- User Profile Management lets you log on with multiple user IDs. To enable your workstation to support multiple logons (and to control selective logoffs), you must type commands at the OS/2 command prompt. Refer to the LOGON and LOGOFF commands in the *Database Manager Administrator's Guide* for additional information about multiple logons.
- Logons are active until you log off or turn off your workstation. For example, even if you had to log on to use Query Manager or the 5250 Work Station Feature, exiting them does not automatically log you off. If you plan to leave your workstation unattended, you should log off to prevent unauthorized use of your user ID.

If you are the only user of databases at your workstation, and you do not want to require yourself and other users to use the Logon panel, you may want to consider the following:

- In User Profile Management, indicate that a password for your user ID is *not* required. Ensure the user ID type is administrator (so you will have full authority over the database).

- Add the following to the STARTUP.CMD file so the logon will be automatic when the workstation is first turned on:

  ```
  LOGON userid
  rem   Automatic logon---userid is an administrator and does
  rem   not require a password.  Anyone with access to this
  rem   workstation can use Database Manager without a user ID
  rem   or password.
  exit
  ```

Information about using a STARTUP.CMD file is contained in the *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 1: Base Operating System.*

To Log Off User Profile Management

You can log off User Profile Management Services in one of two ways:

- Issue the LOGOFF command from the OS/2 command prompt.
- Select **User Profile Management Services** from the Desktop Manager window and select **Logoff** from the Group – User Profile Management Services menu. Then select a user ID and **Logoff**, or select **Logoff all**, from the the Logoff menu.

You are also automatically logged off whenever your workstation is turned off.

To Manage User and Group IDs

Note: Your user ID must be assigned an administrator user type to be authorized to perform management tasks within User Profile Management.

1. In the Desktop Manager window, select **User Profile Management Services**. The Group – User Profile Management Services menu is displayed.
2. Select **User Profile Management** from the Group – User Profile Management Services menu. The User Profile Management Services menu is displayed.
3. Select **Manage** from the action bar.
4. In the Management pull-down, make the appropriate selection and follow the prompts to add, update, or erase user and group IDs.

Hints:

- The user ID you use to log on to a workstation through User Profile Management does not display in the User Management menu. For example, if you used the default user ID and password as described in *Getting Started* to log on to your workstation for the current session, the default user ID will not display in the User Management menu. Should you want to erase the default user ID and password, you must create another user ID profile and assign it an administrator user type. You can then log onto your workstation using the second user ID. Since the second user ID is defined as an administrator, you can change or erase the default user ID profile. You cannot change the authorization specified for the profile of the user ID that was used to log on to the workstation for the current session. You are, therefore, protected from not having at least one user ID defined as an administrator.
- If you want to add your user ID to a group, select **Select groups for user ID** from the Actions pull-down in the User Profile Management menu and follow the prompts to add your user ID to the listed groups.
- For user ID and password naming rules, see Appendix A.

To Update Your User ID

Note: Use this procedure to view your access authority, change your password, and change your user comments.

1. In the Desktop Manager window, select **User Profile Management Services**. The Group – User Profile Management Services menu is displayed.
2. Select **User Profile Management** from the Group – User Profile Management Services menu. The User Profile Management menu is displayed.
3. Select **Actions** from the action bar.
4. In the Actions pull-down, make the appropriate selection and follow the prompts to update your user ID.

Chapter 2. Beginning to Use Database Manager

This chapter provides information and steps about installing the sample database, starting and stopping Query Manager, and opening a database. The sample database can be used with other chapters in this volume to aid you in understanding how to use Query Manager. An overview of Database Manager authorizations and a discussion of qualifiers is also provided.

About Database Manager

Database Manager is based on the relational model of data. This means that the information in the database is presented to you in tables, which are simple tabular forms of rows and columns. You define data or access data in terms of tables and operations on those tables. This allows you to design, create, and access your databases without having to understand complex physical data structures and access methods.

A database can store one table or many different tables. You can create several databases, each based on a different topic. Each database can contain tables of data pertaining to that topic. For example, you can create a personnel database that contains two tables: one with employee names and addresses and another with employee names and salary data. You could also create a merchandise database that contains several tables with data describing all the merchandise sold by your company.

A Database Manager database contains tables, views, and indexes, as well as Query Manager *objects* that you create to access or manipulate the data in the tables. These other objects include queries, forms, customized menus, customized panels, and procedures. Each of these objects is described in this section.

Database Manager consists of *Database Services*, *Remote Data Services*, and *Query Manager*. Database Services is the engine of Database Manager; it actually manages the data that is stored in the database and provides lock management, transaction management, and data integrity. Database Services provides an application programming interface that allows you to write programs in the IBM C/2*, FORTRAN/2*, COBOL/2* and PASCAL/2 programming languages to access Database Manager databases. Database Services also supports the Procedures Language 2/REXX, which is a high-level interpretive language that does not have to be compiled. Software developers can use Database Services as an application programmer interface to write their own precompilers for Database Manager applications.

Remote Data Services provides users access to remote databases across an IBM Token-Ring Network, PC Network* and ETHERAND*, X.25, or Synchronous Data Link Control networks. Multiple workstations or applications can access a single database at the same time. Access to remote and local databases is transparent once database cataloging has been performed. This volume addresses using Database Manager databases on a standalone workstation. Refer to the *Database Manager Administrator's Guide* for information on using remote databases in a distributed environment.

Query Manager allows you to access and manipulate the database with a Presentation Manager* interface that provides menus and panels with associated contextual help. Each of these menus and panels provides information about the process of selecting tables, columns, and rows from a Database Manager database and performing the desired database operations. Help is available by pressing the Help (F1) key or by selecting Help with a mouse. This volume of the *User's Guide* describes how to use Query Manager.

Before you can use Query Manager, you must first install both Database Services and Query Manager on your workstation. For information on installation, see *Database Manager Excercises.*

Database Manager Authorizations

Controlling access over a database is an important consideration. For example, suppose you are the accountant for a large company and you are in charge of the monthly payroll and overseeing all of the data processing for your department. There are four clerks in your department who type information into different tables on the database. You want to control who can create reports, perform queries, and create tables on the database.

You would want only the manager for the company and yourself to have authorization to type and retrieve information from the table containing employee salaries. You want the clerks to be authorized to type information and to query the database using only certain tables.

You can control how users access Database Manager databases, tables, and views by granting and revoking *authorizations*. After you create a Query Manager object (query, form, menu, panel, or procedure), Query Manager provides a way to save, name, and control access to the object.

User Profile Management controls access *outside* of Database Manager by maintaining and verifying a list of authorized user IDs and passwords used to log on to a workstation. (Refer to Chapter 1 for more information on adding and managing individual and group user IDs.) You must have an authorized user ID defined in User Profile Management to access Database Manager through a User Profile Management logon panel or the OS/2 command prompt. Access controls are provided *within* Database Manager through the use of authorizations granted to users and groups of users (identified by their user ID).

An authorization provides the ability to access, create, or control a database, table, view, plan, or other Query Manager object. Authority to open a database, create a table, create a new access plan, and assign another user as a database administrator is known as a *database* authorization. The authorization to Query rows, Add rows, Change rows, Delete rows, Add an index, Alter a table, Reference a table, and have table Control authority are the different types of *table* authorizations. *View* authorizations provide controls on how a view can be accessed and used. You can control how users can select and use queries, forms, and other Query Manager objects by setting **Share** to **Yes** or **No**.

Direct and Indirect Authorizations

Authorizations are granted to individual user and group IDs directly or indirectly. (A group ID includes one or more individual user IDs.) You can *directly* grant authorization to a specific user ID or specific group ID. Whenever you grant authorization to a group ID, you are *indirectly* granting authorizations to every individual user ID included in that group. PUBLIC is another type of group ID used within Database Manager. When you grant an authorization to PUBLIC, you are indirectly granting the authorization to every user ID authorized in User Profile Management for the local workstation or known to Database Manager.

Levels of Authority

Your user ID is granted different *levels* of authority in Database Manager. User Profile Management provides the first set of access controls to Database Manager by maintaining a list of authorized user IDs assigned different user *types*. If your user ID is assigned an *administrator* or a *local administrator* user type in User Profile Management, you automatically have SYSADM (system administrator) authority over Database Manager for the selected workstation.

SYSADM authority provides you top level authority over Database Manager. You have access and control over databases and objects within Database Manager. You can grant or revoke authorizations to all users for *any* database, plan, table, view, or Query Manager object. Only a user with SYSADM authority can create or erase a database, change the Database Manager configuration file, or reinstall Database Manager. In addition, you can use the *Authorizations* function to directly grant or revoke to other user and group IDs, the second level of administrative authority within Database Manager: *database administrator*. Users with SYSADM authority are automatically database administrators over any database they create.

As a database administrator, you have access and control over a selected database. This means that you could be a database administrator for one database but not have the same level of authority for another. You can only be granted database administrator authority by a user with SYSADM authority.

With either Database Manager administrative authority level (SYSADM authority or database administrator), you also automatically have *Control* authority over any table, view, or index for the selected database. If you do not have an administrative authority level within Database Manager, you can be granted Control authority by a user with SYSADM authority or by a database administrator. You are also automatically granted Control authority over any table, view, or index you create. With Control authority over a selected table or view, you can use the Authorizations function to control access over the table or view you create.

When you create a Query Manager object, you automatically become the *owner* of the object. As the owner of the Query Manager object (or as a user with an administrative authority level), you can control access to the object when you save and name the object by setting **Share** to **Yes** or **No** in the Save panel. When **Share** is set to **Yes**, other users with appropriate database and table authorization can select and run, edit, export, or import the object. How you can use the Share option is discussed in each chapter in this volume that provides information to creating and using a Query Manager object.

Your user ID can be granted top level (administrative) authority or a combination of different authority levels over each database, table, view, or plan. Your ability to grant or revoke an authorization depends upon your user ID authority level for the selected database or object. For example, a user with SYSADM authority can grant or revoke another user database administrator authority. A user with database administrator authority, however, cannot grant another user database administrator authority.

The following tables illustrate users with different authorization levels in Query Manager. For more information on performing steps to grant or revoke database or table and view authorizations, see "Using Database Authorizations" on page 8-17 and "Using Tables and Views Authorizations" on page 8-28.

Grant/Revoke authorization to:	SYSADM Authority	(DBADM) Database Admin.	Control Authority	User (no authority)
SYSADM authority		X	X	X
Database Admin.			X	X
Control authority				X
User				

Note: You must be defined as an *administrator* or a *local administrator* user type in User Profile Management to have SYSADM authority in Database Manager. A user with database administrator authority cannot grant or revoke database administrator authority to another user. A user with Control authority cannot grant or revoke another user Control authority.

Installing the Sample Database

The exercises in *Database Manager Exercises* use a sample database. If you want to do these exercises, you must install the sample database before starting Query Manager. It is recommended that you install the sample database and use the exercises to help you learn how to use Database Services and Query Manager. The sample database is also used in examples throughout this volume.

To use the sample database to do the exercises, you need to have installed the base operating system, Query Manager, and Database Services.

If you perform the exercises in *Database Manager Exercises* and another person in your organization wants to go through the exercises later on, you may need to erase the SAMPLE database tables so they can recreate them. If this is the case, follow the steps "To Erase Tables From the SAMPLE Database" described in *Database Manager Exercises*, or follow the steps in "To Erase a Table" on page 5-21 in this volume.

If you have no further use for the sample database, you should erase it. Follow the steps in "To Erase the Sample Database" on page 2-7 to erase the entire sample database.

To Install the Sample Database

At the OS/2 command prompt, type `sqlsampl` and press Enter.

Notes:

1. If you are not logged on to your workstation through User Profile Management, Database Manager will prompt you to do so.
2. If you receive any error messages, see "Sample Database Installation Messages" on page H-3 for error-recovery information.

When the sample database installation program completes, the OS/2 command prompt is displayed.

To Erase the Sample Database

1. After you have started Query Manager, select **SAMPLE** from the Databases primary menu.
2. Select **Actions** from the action bar or press the Switch to Action Bar (F10) key and press Enter.
3. Select **Erase** from the Actions pull-down.
4. Select **Yes** when the confirmation message is displayed and select Enter.

   The Databases primary menu is displayed and the name SAMPLE is removed.

Starting and Stopping Query Manager

Once Database Manager is installed, you can start Query Manager in one of two ways:

- The most direct method, and initially the simplest, is from the Group – Main window. Refer to the steps under "To Start Query Manager" on page 2-9 for instructions.
- From the OS/2 command prompt. For information on starting Query Manager from the OS/2 command prompt, see Appendix B.

Notes:

1. If you are the only user of databases at your workstation, you may not want the Logon panel from User Profile Management to be displayed the first time you access a Database Manager database or other protected Query Manager object. To prevent the Logon panel from being displayed, follow the instructions under *Hints* on changing the STARTUP.CMD file under "To Log On to User Profile Management" on page 1-3.

2. Before you can access remote databases on other workstation, you must have Communications Manager installed on your local and remote workstations. In addition, the appropriate Advanced Program-to-Program Communications Interface (APPC) profiles within a Communications Manager configuration file must be created on the local and remote workstations. For more information, see your database administrator or the *Database Manager Administrator's Guide.*

   After Communications Manager is installed, a configuration file is created. You may also need to contact the person (for example, your system or database administrator) responsible for the remote workstation to perform the appropriate startup procedure at the workstation. At the Database Server or the Database Requester workstation, you should always start Communications Manager before starting Query Manager. For information about starting Communications Manager, see the *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 2: Communications Manager and LAN Requester* (referred to as the *User's Guide, Volume 2: Communications Manager and LAN Requester* in the remainder of this volume).

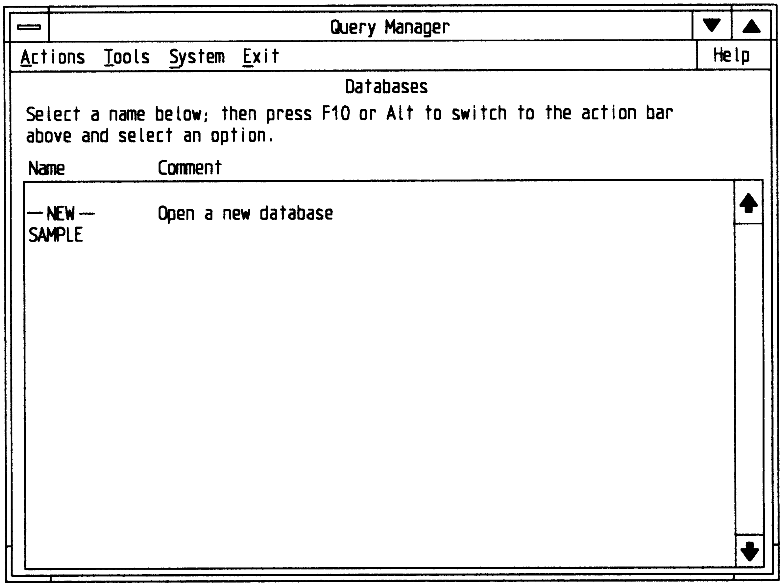
When you have completed your work with Query Manager, you can return to the Group – Main window or the OS/2 command prompt, depending on which one was your starting point.

To Start Query Manager

1. From the Group – Main window, select **Query Manager**.

   Note: If you are using the keyboard, in the Group – Main window, press the Up Arrow (↑) or Down Arrow (↓) keys to scroll through this list until the cursor is on **Query Manager** and then press the Enter key.

   The Databases primary menu is displayed.



Databases

Select a name below; then press F10 or Alt to switch to the action bar above and select an option.

Name	Comment
—NEW—	Open a new database
SAMPLE	

Hints:

- If you want to use the sample database now, refer to *Getting Started.*
- If you are using Remote Data Services, make sure you start Communications Manager before Query Manager.

To Stop Query Manager

1. In the Databases primary menu, select **Exit** or press the Exit Query Manager (F3) key.

   Notes:

   a. You can also exit Query Manager from each object names primary menu or from the Main Selection menu.

   b. If you are using the keyboard in the Databases primary menu, press the Switch to Action Bar (F10) key to switch to the action bar at the top of the screen. Press the Left Arrow (←) or Right Arrow (→) keys to position the cursor on **Exit** and press Enter.

   c. The fast path for stopping Query Manager is to press the Exit Query Manager (F3) key from the Databases primary menu.

2. Select **Exit Query Manager** from the Exit pull-down.
3. Select **Yes** from the confirmation message.

   Either the Group – Main window or the OS/2 command prompt is displayed, depending on which one you used to start Query Manager.

Opening a Database

In order to type and retrieve data from a database, you must first open the database you want to use. You must be a user with SYSADM (system administrator) authority to open a *new* database. You can only create the database on your local workstation.

You can open any *existing* database that you have been granted authority to use.

To Open a New Database

When you start Query Manager from the Group – Main window, the Databases primary menu is normally the first menu that is displayed.

1. In the Databases primary menu, select —NEW—.
2. Select **Actions** from the action bar or press the Switch to Action Bar (F10) key, and press Enter.

3. Select **Open** from the Actions pull-down and press Enter or press the Open (F6) key.

   Note: The fast path for selecting **Open** is to press the Open (F6) key from the Databases primary menu. Another fast path is to select Enter twice. The first time Enter is selected, the pull-down is displayed; the second time, **Open** is selected.

4. In the Open Database panel, type the name for the database and drive. Type a comment for the database and select Enter. Select Help for information about typing database names.

   The Main Selection menu is displayed.

Hint:

When a database is created using Query Manager, the database is assigned the code page and country code active for that OS/2 session. The code page and country code for a session is normally the same as those contained in the CONFIG.SYS file, unless they are not specified in the CONFIG.SYS file (indicating that system defaults were used); or, in the case of the code page, it was explicitly changed by you.

To Open an Existing Database

1. In the Database primary menu, select the name of the database you want to open.

   Note: If you are using the keyboard, in the Database primary menu, press the Up Arrow (↑) or the Down Arrow (↓) keys until the name of the database you want to open is highlighted, then press Enter.

2. Select **Actions** from the action bar, or press the Switch to Action Bar (F10) key, and press Enter.

3. Select **Open** from the Actions pull-down and press Enter, or press the Open (F6) key.

   The Main Selection menu is displayed.

Hints:

- For information on the action bar options **Tools** and **System**, both of which are available in the Databases primary menu, see "Using Query Manager System Tools" on page 8-3 and "Using Query Manager Database Tools" on page 8-7.

- In order to open an existing database, the code page of the OS/2 session must match that of the database. The code page for Query Manager sessions is normally that of the system, and can be found in the CONFIG.SYS file. The base operating system uses a default if no code page is specified.

Printing with Query Manager

If you have an IBM printer, you can print reports, object definitions, or data from a panel as soon as you begin using Query Manager. All the reports and object definitions are printed in 10 pitch. If you want to print in either 15 or 17 pitch (compressed), you should create a printer nickname file.

In order to use the printer nickname file, you must either change your current profile or create a new profile that points to that printer nickname. You can then you specify that profile name when you start Query Manager.

For information on creating a printer nickname file, see "Using Query Manager System Tools" on page 8-3. For information on creating profiles, see "Using Profiles" on page 8-35. For information on specifying a profile name when you start Query Manager, see Appendix B.

Using Qualifiers

All of the objects, such as tables, queries, and forms that you create using Query Manager, are prefixed with a unique qualifier. Qualifiers are provided in Query Manager so that you can name and display groups of objects under a particular category or user ID. Whenever you select an item from the Main Selection menu, an object names primary menu is displayed. The title line for that primary menu will display the object type and the *current active qualifier* for the objects listed. Query Manager will only list an object whose qualifier exactly matches the current active qualifier that the user has authority to use.

In previous versions of Database Manager, you were instructed to follow the steps in *Getting Started* to set SQLUSER to a unique value in the CONFIG.SYS file. This unique value was used by Query Manager to prefix the names of the objects you created. This value is now known as the qualifier.

Database Manager Version 1.3 sets the default value for the current active qualifier in one of three ways:

- From the OS/2 command prompt or from the Properties panel (Refer to Appendix B for more information.)
- From the Query Manager profile whenever you start Query Manager
- From the user ID used to log on to a workstation.

Query Manager will use the active qualifier specified from the OS/2 command prompt or the Properties panel as the first default value for the current qualifier. If you do not specify a value for the qualifier, Query Manager will use the value set in the Query Manager profile as the current active qualifier. If no qualifier value is specified when Query Manager is started, Query Manager automatically sets the current active qualifier to your user ID.

For example, suppose you have not set a value for the qualifier from either the OS/2 command prompt or in the profile. If you log on to your workstation using the user ID of BLAKE and create a query named QCLERK, the query name is displayed in the Queries primary menu as QCLERK when the qualifier is equal to BLAKE.

If you want to create and group several queries, you can select **Change qualifier for lists** from the Actions pull-down. The qualifier you changed to now becomes the *current active* qualifier and any other objects you create are prefixed with this new qualifier. You can use **Change qualifier for lists** to review or access objects created under different qualifiers. **Change qualifier for lists** is available from each object names primary window.

Note: If NULLID is displayed in the list of qualifiers, you have created a Query Manager object using Database Manager Version 1.0 or 1.1.

If you select **Change qualifier for lists** to view objects using another qualifier, you will be able to review and access only those objects for which *your* user ID is allowed to select.

Consider the following example: you are the accountant for a large company and you have database administrator authority over a database named ACCOUNTS. You have created two tables: CASH and PAYROLL. You named the tables with a qualifier equal to your user ID of BLAKE.

Since you want only yourself and your manager to be able to access information stored on the PAYROLL table, you use the Authorizations function to grant table authorizations on the PAYROLL table to your manager but you do not grant authorization to the PAYROLL table to other users. You grant the clerks in your department authorization to access the CASH table. You will learn how to do this later in this volume. You then proceed to create several queries from each table with the active current qualifier still set to your user ID of BLAKE.

You request a clerk working in your department to run a query named BLAKE.CREDIT against the CASH table. The clerk logs on to a workstation using their user ID and selects **Change qualifier for lists** to display queries with a qualifier equal to BLAKE. Database Manager checks the clerk's authorization and lists only the queries with the specified qualifier the clerk has authority to select. In this example, the CREDIT query is displayed with other queries the clerk has access to. The clerk can only view and select those queries you saved with **Share** set to **Yes**.

If the clerk did not use the **Change qualifier for lists** to change the current active qualifier to equal BLAKE, none of the queries created under this qualifier would display. When the clerk logs off the workstation and logs back on or *changes* databases, the active current qualifier resets back to the default value originally specified from the OS/2 command prompt, Query Manager profile, or the user ID used to log on to the workstation through the logon panel in User Profile Management.

Note: Query Manager displays up to 1085 objects in each of the Query Manager objects primary menu for a single qualifier. For example, you could display up to 1085 forms under the qualifier SUSAN, and then use the qualifier SUSAN again to display up to 1085 queries. Query Manager does not limit the number of objects you can create for a particular qualifier; only the number to be displayed.

Chapter 3. Querying a Database

This chapter provides the steps and accompanying information you need to write queries. Chapter 4 provides the information you need to format report forms. It is assumed that you have completed the exercises in *Database Manager Exercises* and do not need to be guided through all of the same steps again.

About Querying a Database

Query Manager provides *prompted query* to allow you to request specific information from a database. Prompted query guides you through a series of steps to define your query. Once your query is defined, you can run it and display the results on your screen or print the results in a report.

The first time a query is run without specifying a form, a default report form is used. A *form* is a definition of how the result of a query, or the report, is to be displayed or printed. To allow you to display or print reports in the format of your choice, you can define a different form for the report. The report can be formatted with a personalized title, different headings, double-line spacing, summary information, and other options.

Once you run your query and the report is displayed, a few keystrokes take you to the Form panel, where you can make changes to the report form. When you display the report again, the changes to the report form are reflected. You can switch back and forth between the form, report, and query until your report is formatted to meet your needs. This is called the *query-report-form triangle*. If necessary, you can also use the triangle to change and run the query again if different data is required. In addition, you can have several different report forms for one query, or one report form for several different queries, depending on your needs.

Note: When you create a prompted query, Query Manager automatically creates an equivalent SQL SELECT statement to actually access the data in the database. However, you do not need an in-depth knowledge of SQL (Structured Query Language) to write Query Manager prompted queries. If you want to use SQL to write queries, see Chapter 9 and then see *IBM Operating System/2 Extended Edition Version 1.3 Structured Query Language (SQL) Concepts* (referred to as *Structured Query Language (SQL) Concepts* in the remainder of this volume) for more information.

Authorization

Queries are instructions to Query Manager that request some subset of data that exists in the database. To write a query, you need to know how to describe the subset of data you want. You also need certain authorizations granted to you to retrieve data from a database. You cannot create a prompted query unless you can access a database and have *Query rows* authorization for a selected table. (See "Using Tables and Views Authorizations" on page 8-28 for an explanation of Query rows authorization.)

If you are a user with SYSADM authority (system administrator), a database administrator for the selected database, or a user with table Control authority for the selected table, you automatically have access to the table and have Query rows authorization. (See "Authorizations" on page 5-1 for an explanation of table Control authority.)

With either administrative authority level or table Control authority, you can use the table **Authorizations** function to grant or revoke other users Query rows authorization. When you are granted Query rows authorization for a selected table, you can define and run a query to access the selected table.

When you create a query, you are considered to be the *owner* of the query. When you are finished defining a query, you can select to exit and save the query. A Save panel is then displayed that prompts you to type a name and comment and make a **Share** selection. As owner of the query, you can choose to grant or revoke other users the authority to access and run the query by setting **Share** to **Yes** or **No** in the Save panel.

If the query is saved with **Share** set to **Yes**, any user who can access the database can *select* and *view* the query. You can *run, edit,* and *print* the query when you have Query rows authorization for the table or view that the query accesses. You can also *export* or *import* the query and assign it a new name. You can also select **Get** to retrieve a copy of the prompted query and then edit and save it under a new name. You cannot edit a query and save it under the same name unless you are the owner of the query or a user with an administrative authority level.

When the query is saved with **Share** set to **No**, then only the owner, a user with SYSADM authority, or a database administrator for the selected database can select and run the query, or import or export the query.

You can check the authorization granted your user ID to query rows for a table or view by selecting **Authorizations** from the Tools pull-down in the Tables and Views primary menu. For more information and a discussion on steps to grant or revoke table and view authorizations, see "Using Tables and Views Authorizations" on page 8-28.

Writing Queries

Query Manager prompted query provides menus and panels that guide you through the steps to describe the subset of data you want to retrieve from the database.

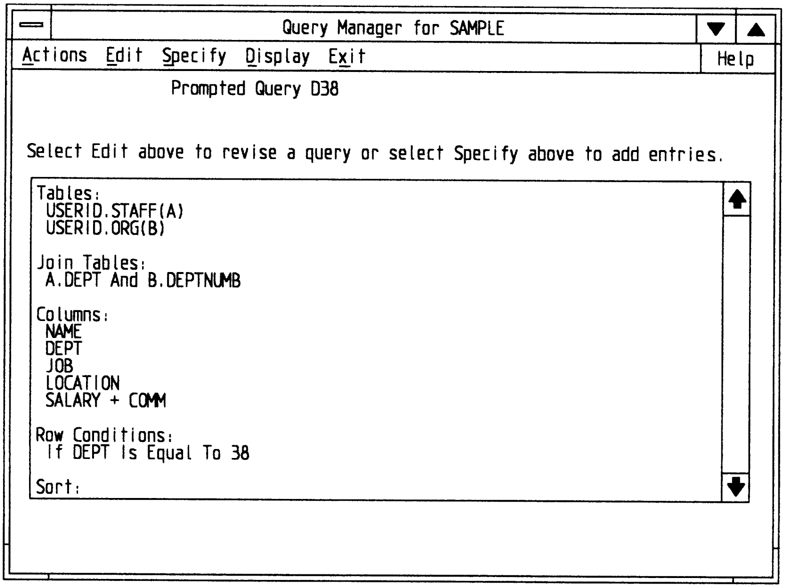
In *Database Manager Exercises*, there was an exercise in which you created a query to retrieve data from the STAFF table stored within the SAMPLE database. This exercise showed you how to retrieve data from a single table.

In this chapter you will define a more complex query that retrieves data from the STAFF *and* the ORG table named D38. This query retrieves the data for employee name, department, job, location, and salary plus commission for each employee in department 38, then puts the data in alphabetical order by employee name. (Remember this query; the report is displayed and discussed in Chapter 4.)

To write this query, you need to specify the following:

- Table names (STAFF, ORG)
- Join columns (A.DEPT, B.DEPTNUMB)
- Column names (NAME, DEPT, JOB, LOCATION, SALARY+COMM)
- Row condition for selecting the output (DEPT = 38)
- Sort value as ascending (for NAME).

The STAFF and ORG tables are shown in Chapter 5. The completed D38 query is displayed as follows:



Prompted Query D38

Select Edit above to revise a query or select Specify above to add entries.

```
Tables:
 USERID.STAFF(A)
 USERID.ORG(B)

Join Tables:
 A.DEPT And B.DEPTNUMB

Columns:
 NAME
 DEPT
 JOB
 LOCATION
 SALARY + COMM

Row Conditions:
 If DEPT Is Equal To 38

Sort:
```

When you define a query, the menus are presented in the appropriate order so that you can define each part of the query in the correct sequence. The prompted query menus provide the following sequence:

Tables Define the table names to be used in the query.

Join Tables Define how the tables are to be joined and on which columns to join them. If you specify more than one table name, prompted query automatically provides menus in which you specify how the tables are joined. The tables must have a connecting column containing matching values to provide a way for Query Manager to link rows from one table to the corresponding rows in another table. This is called *joining* the tables because the corresponding rows are actually joined together to form a single row in the report. Joining is only required if more that one table is specified for the query.

In example query D38, both the STAFF and ORG table have columns for department number. Since both tables contain a column for department number, you would select this column from each table to join the tables. This column may not be named the same in each table, but it must have matching column length and data type *and* contain data with matching values to link the specified tables together.

When two or more table names are selected, the table names are displayed in the Prompted Query panel under the heading **Tables:**. Each table name is followed by a letter in parentheses that qualifies or identifies the table. The columns used to join the tables have a letter prefix that corresponds to the letter identifier of the table. For example, the column A.DEPT is in the table USERID.STAFF(A), and the column B.DEPTNUMB is in the table USERID.ORG(B).

Columns Define the columns that represent the data produced. A column for query output can be a single column (NAME), a column that represen ents an arithmetic expression (SALARY+COMM), or a column that represents a summary of the data in a column (SUM, AVG, and so on).

Row Conditions Define the qualifying conditions for selecting rows of data to be retrieved. Only data from the specified rows is retrieved. A row condition has three parts: a left side, a comparison operator, and a right side. For example, in the query D38 previously described, the row condition is `If DEPT Is Equal To 38`. `DEPT` is the left side of the condition, `Is Equal To` is the comparison operator, and 38 is the right side.

Sort Define the sort order in which the output rows are to be displayed. Sort order can be ascending (A to Z or 0 to 9) or descending (Z to A or 9 to 0).

Duplicate Rows Define whether duplicate rows from the set of retrieved rows in the query should be kept or discarded. Duplicate rows occur when a selected row matches another selected row. The default is to keep duplicate rows. Duplicate rows would be set to **Keep**, for example, when a query selects the columns NAME and DEPT and two employees have the same last name and are in the same department. In this case, you would probably want to keep duplicate rows.

During the prompting sequence, the current item being defined (**Tables, Join Tables, Columns, Row Conditions, Sort, Duplicate Rows**) is displayed in the Prompted Query panel followed by an ellipsis (...). The ellipsis indicates that you are still defining the query. After the item is defined, the entire selection is displayed in the Prompted Query panel.

Within the columns and row conditions you specify for a query, you can use *expressions*. An expression is limited to 254 characters and can consist of the following:

- A column name defined for any of the tables that are being queried; for example, DEPT.
- A character string; for example, `'Clerk'`.
- An arithmetic expression consisting of a numeric column, numeric constant, or a summary function, optionally preceded by an addition (+) symbol or a subtraction (−) symbol; for example, AVG(SALARY) or SUM(COMM).
- An arithmetic expression connected to another arithmetic expression by an arithmetic operator to form a larger arithmetic expression. Arithmetic operators and symbols are addition (+), subtraction &l par.−), multiplication (*), and division (/); for example, (SALARY+COMM).
- An arithmetic expression that contains another arithmetic expression enclosed in parentheses; for example, SALARY+(2*COMM).

You can also specify *summary functions* for columns in your queries. A summary function can consist of a function and an argument. The function can be SUM, AVG, MIN, MAX, or COUNT. These are defined in *Structured Query Language (SQL) Concepts.*

The argument must meet the following rules:

- The argument must immediately follow the function. The argument must be enclosed in parentheses.

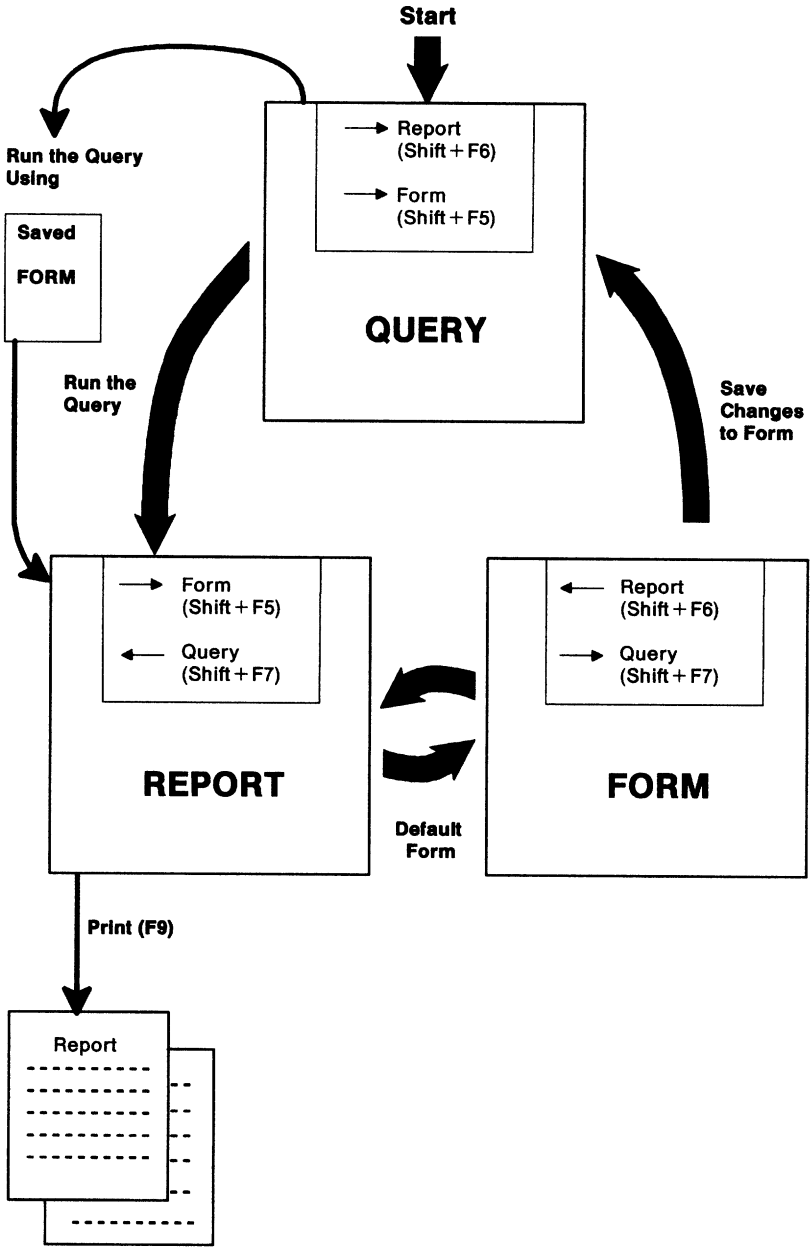
- The argument must contain at least one column name (except for COUNT (*), where you are counting all the qualifying rows in the query).
- SUM and AVG must use an arithmetic expression (that is, numeric data). MAX and MIN can contain an arithmetic expression or the name of a column containing character data.
- An argument cannot contain another summary function.

If you specify summary functions in your query, the results of the query are summary information for one or more groups of rows. For example, if you specify only AVG(SALARY) and SUM(COMM), the results of the query are the average salary and the sum of the commissions of all the rows selected by the query. If you specify column names in addition to summary functions, the results of your query are summary information for each unique group of column names. For example, if you specify DEPT, AVG(SALARY), and SUM(COMM), the results of you r query are the average salaries and sum of the commissions for each department. If you specify DEPT, JOB, AVG(SALARY), and SUM(COMM), the re sults are the average salaries and sum of the commissions for each unique job type within each department. If you specify DEPT, NAME, AVG(SALARY), and SUM(COMM), the results of your query are the average salary and sum of the commissions of each person within each department. This may not be what you want; since each person has only one salary and one commission, the results of this query would not be very meaningful.

The Query-Report-Form Triangle

When you run a query, a default form is used to generate a report. If the format of the report is not what you want, you can use the query-report-form triangle to go to the Form panel and modify the default form. Then, you can return to the Report panel and the report is displayed using the modified form. If you want to keep the modified form, you should return to the Form panel and save it. The next time you run the query, run it using the modified form to generate the correctly formatted report. You can also start the query-report-form triangle using the RUN co mmand. Refer to the *OS/2 Command Reference* for more information on using this command.

The following diagram illustrates the query-report-form triangle:



If the data generated by the query is not quite what you want, use the query-report-form triangle to return to the Prompted Query panel, then change the query and run it again. To delete or insert items in a query, select the item you want to work on and select **Edit** from the action bar. You can repeat this process until you have defined the query to generate all the data you want from the table. Then, you can save the query for use at a later time.

Since you have already been through the steps for writing a query in *Introducing OS/2* and in "Exercise 1 – Querying a Database and Displaying a Report" in *Database Manager Exercises*, the steps for writing a query should be somewhat familiar to you. Therefore, the steps that follow for writing a query are divided into sections to provide more detailed information for you as you write your own queries. However, the illustrations will represent the example D38 query found in "Writing Queries" (on 3-3). Write a query using the following procedures:

- To Specify Tables for a Query
- To Specify Columns to Join Tables for a Query
- To Specify Columns for a Query
- To Specify Row Conditions for a Query
- To Specify Sort Order for a Query
- To Discard Duplicate Rows in a Query
- To Run and Save a Query
- To Edit a Query.

These procedures are meant to be followed in order, but you can skip the steps that do not apply to your work.

Note: The names for all the queries you define are prefixed with the current active qualifier. If you want to access and use copies of queries defined under another qualifier, you can select **Change qualifiers for lists** from the Actions pull-down from any object's name primary menu. Any changes you make to the query are saved under the qualifier you changed to. If you want to use tables in your query that were created with a different qualifier, you can select **Change qualifiers for lists** to specify a new qualifier. Query Manager will list all of the tables you are authorized to access. You can also use a table in your query that was created with a different qualifier by simply specifying the fully qualified name; for example, USERID.TABLENAME.

To Specify Tables for a Query

1. In the Main Selection menu, select **Queries**.
2. Select **—NEW—** from the Queries primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.
4. Type the table names you want to use in the Tables menu and select Enter.

   Notes:

   a. You can select List to display a list of the table names which you have authorization to access. Position the cursor on the table name and select Enter to select a table name from the list. To select more than one table, position the cursor in the next entry field. Select List again, and then select another table name.

   b. Up to 15 tables can be joined; however, if the tables you are using are very large (in terms of the number of rows and columns), the join process can use up all the memory available and the query cannot be run.

   c. As you define the query, the definition is reflected in the Prompted Query panel.

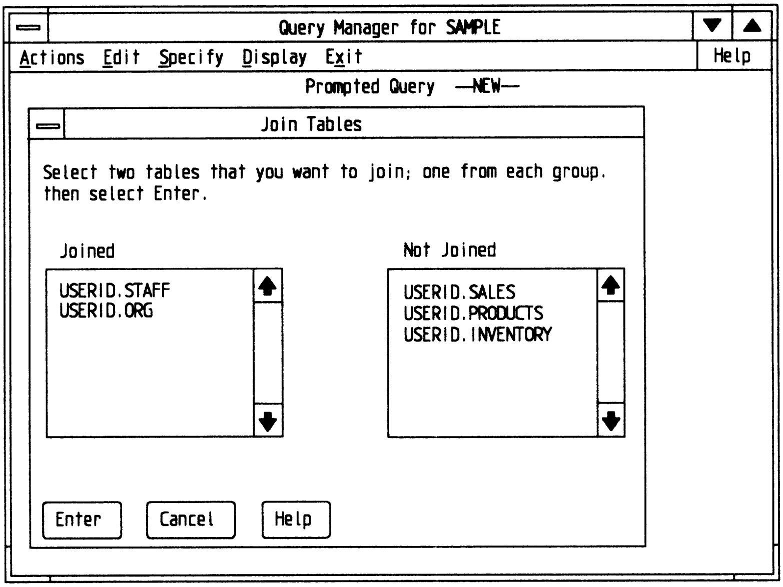
   d. When two or more table names are selected, the table names are displayed in the Prompted Query panel followed by a letter identifier in parentheses.

5. Determine the steps to use based on the number of tables you selected:

   - If you selected one table, skip the steps that follow and continue with "To Specify Columns for a Query" on page 3-13.
   - If you selected two tables, continue with step 7 in this procedure.
   - If you selected more than two tables, continue with step 6 in this procedure.

6. Select a table from each group in the Join Tables menu and select Enter.

Note: The **Joined** group indicates which tables have already been joined; the default is the first selected table. The **Not Joined** group indicates the tables that have not yet been joined; the table names being displayed are those that you selected in the Tables panel.



Select two tables that you want to join; one from each group. then select Enter.

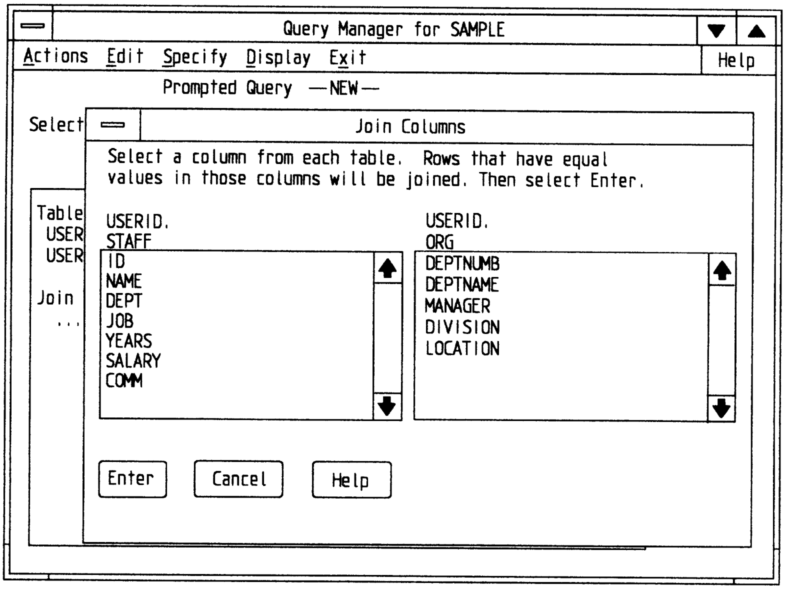
Joined

USERID.STAFF
USERID.ORG

Not Joined

USERID.SALES
USERID.PRODUCTS
USERID.INVENTORY

7. Select a column with matching values, data type, and column length from each group in the Join Columns menu and then select Enter.

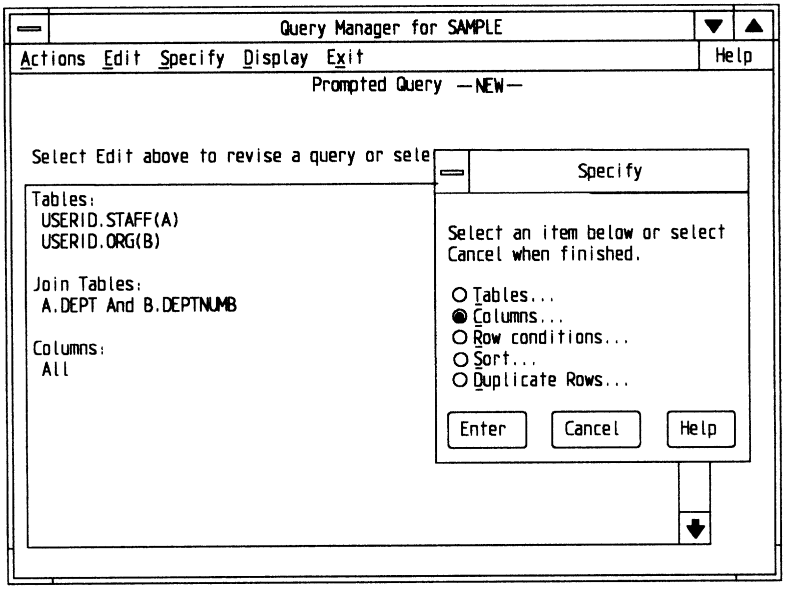


Select a column from each table. Rows that have equal values in those columns will be joined. Then select Enter.

Note: The column names displayed are those from the two selected tables.

8. If you only selected two tables, continue with "To Specify Columns for a Query" on page 3-13. Otherwise, continue with step 9.

9. If you selected more than two tables, repeat steps 6 and 7 until you have joined all the tables to establish a link from table to table in your query.



Query Manager for SAMPLE

Actions Edit Specify Display Exit Help

Prompted Query —NEW—

Select Edit above to revise a query or sele

Tables:
USERID.STAFF(A)
USERID.ORG(B)

Join Tables:
A.DEPT And B.DEPTNUMB

Columns:
All

Specify

Select an item below or select Cancel when finished.

○ Tables...
● Columns...
○ Row conditions...
○ Sort...
○ Duplicate Rows...

Enter Cancel Help

10. Continue with "To Specify Columns for a Query" that follows.

Hint:

> If you made a mistake in creating this query, see "To Edit a Prompted Query" on page 3-27 for steps on how to edit the query.

To Specify Columns for a Query

In the steps that follow, you can only select **Column Names**, **Summary functions**, or **Expression** one at a time from the Columns menu. If you need to use more than one of these in order to specify the columns in your query, you must select **Columns** again in the Specify menu.

1. In the Specify menu, select **Columns**. If the Specify menu is not displayed, select it from the action bar.
2. Use **Column Names** as follows:
   - To select all the columns in a table, select **tablename – all**, where **tablename** is the name of the table, and select Enter.
   - To use individual columns in a table or tables, select each column name using the Spacebar and select Enter after selecting all the columns you need.

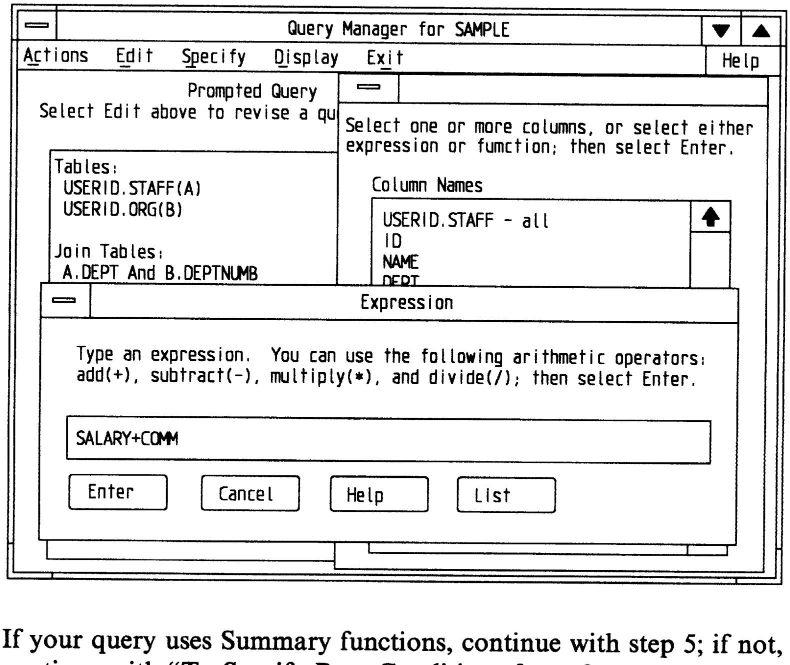
Note: If you want the columns to display in a particular order, you may need to select a column, select Enter to exit the Columns menu, select **Columns** again from the Specify menu, and select another column, and so on. Or, you can also use the report form to change the order in which the columns are displayed in the report.

3. Select **Expression** if you want to define a column as one or more columns or constants in an arithmetic expression, as follows:

   a. In the Columns menu, select **Expressions** and then select Enter to select it.

      If you are using the keyboard, use the Tab key to move to **Expressions**, and press the Enter key.

   b. Type the expression in the Expression panel and select Enter. Select List for a list of column names that can be included as part of the expression. Only numeric columns are listed for expressions.



Select Edit above to revise a qu

Tables:
USERID.STAFF(A)
USERID.ORG(B)

Join Tables:
A.DEPT And B.DEPTNUMB

Select one or more columns, or select either expression or fumction; then select Enter.

USERID.STAFF - all
ID
NAME
DEPT

Type an expression. You can use the following arithmetic operators: add(+), subtract(-), multiply(*), and divide(/); then select Enter.

SALARY+COMM

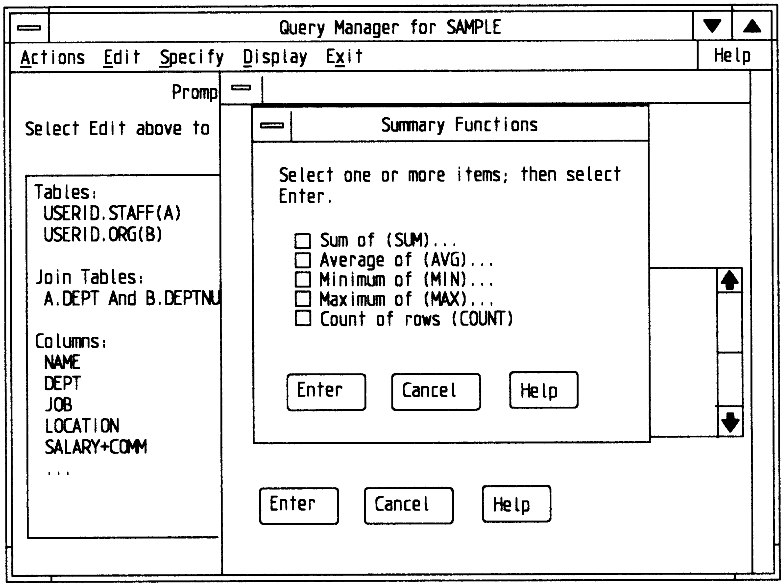
4. If your query uses Summary functions, continue with step 5; if not, continue with "To Specify Row Conditions for a Query" on page 3-17.

5. Select **Summary functions** if you want to define a column as a summary of the data in a column or as a summary of the data generated by an expression, as follows:

   a. In the Columns menu, select **Summary functions** and then select Enter.

      If you are using the keyboard, use the tab keys to move to **Summary functions**, and press the Enter key.

   b. Select the function or functions from the Summary Functions menu and select Enter.



Select Edit above to

Tables:
USERID.STAFF(A)
USERID.ORG(B)

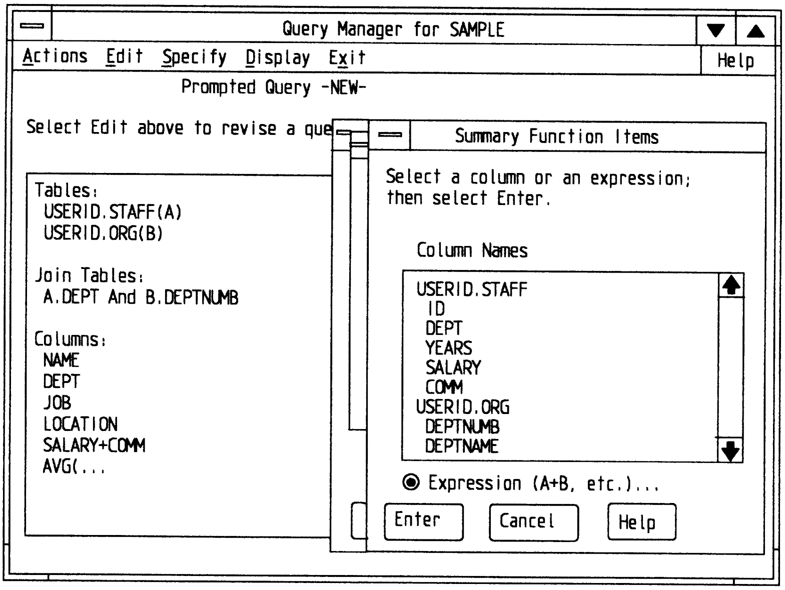
Join Tables:
A.DEPT And B.DEPTNU

Columns:
NAME
DEPT
JOB
LOCATION
SALARY+COMM
...

Select one or more items; then select Enter.

□ Sum of (SUM)...
□ Average of (AVG)...
□ Minimum of (MIN)...
□ Maximum of (MAX)...
□ Count of rows (COUNT)

Note: When you select a function, that function is displayed as **FUNCTION(ARGUMENT)**. The argument can be a column name or an expression; for example, **AVG(COMM + 10)**.



```
Query Manager for SAMPLE

Prompted Query -NEW-

Select Edit above to revise a que

Tables:
 USERID.STAFF(A)
 USERID.ORG(B)

Join Tables:
 A.DEPT And B.DEPTNUMB

Columns:
 NAME
 DEPT
 JOB
 LOCATION
 SALARY+COMM
 AVG(...
```

```
Summary Function Items

Select a column or an expression;
then select Enter.

 Column Names
 USERID.STAFF
  ID
  DEPT
  YEARS
  SALARY
  COMM
 USERID.ORG
  DEPTNUMB
  DEPTNAME

◉ Expression (A+B, etc.)...
```

c. Determine the argument for the selected function or functions:

- To select a column as the argument, select the column from the Summary Function Items menu and select Enter. Continue with step 6.

- To define an expression as the argument for the function, select **Expression (A+B, etc.)...** from the Summary Function Items menu. Continue with step d.

d. Type the expression in the Expression panel and select Enter.

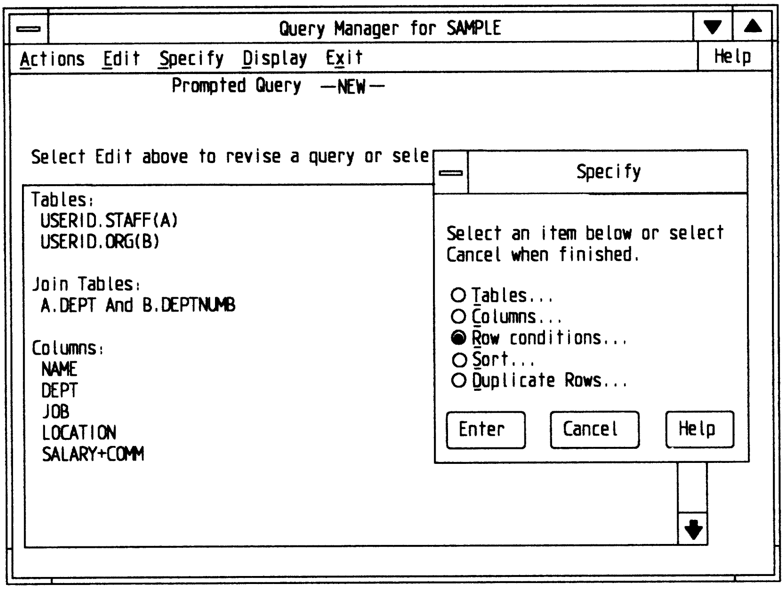
e. If you want to deselect the **Summary** or **Expressions**, press the Tab key to move back to Columns and make another selection.

The columns you define are reflected in the Prompted Query panel.

6. Repeat steps 2, 3, 4, and 5, as applicable, until all columns for your query are defined.

7. Continue with "To Specify Row Conditions for a Query" on page 3-17.

To Specify Row Conditions for a Query



Select Edit above to revise a query or sele

Tables:
USERID.STAFF(A)
USERID.ORG(B)

Join Tables:
A.DEPT And B.DEPTNUMB

Columns:
NAME
DEPT
JOB
LOCATION
SALARY+COMM

Select an item below or select Cancel when finished.

○ Tables...
○ Columns...
● Row conditions...
○ Sort...
○ Duplicate Rows...

1. In the Specify menu, select **Row conditions**. If the Specify menu is not displayed, select **Specify** from the action bar.

   Note: Specifying a row condition allows you to indicate which rows to retrieve. A row condition consists of a left side, a comparison operator, and a right side. Defining the left side is very similar to defining a column, except you can only specify a column name or an expression.

2. Determine the column you want to use for the row condition. For a single column, continue with step 3; for an expression, continue with step 4.

   Note: The row condition you define is reflected in the Prompted Query panel.

3. Select **Column Names** if you want a row condition that contains only a column, as follows:

   a. Position the cursor on the column name and select Enter. Notice that the column expressions you defined for the columns in your query are included in the list of column names.

b. Continue with step 4.

4. Select **Expression** if you want a row condition that contains an expression, as follows:

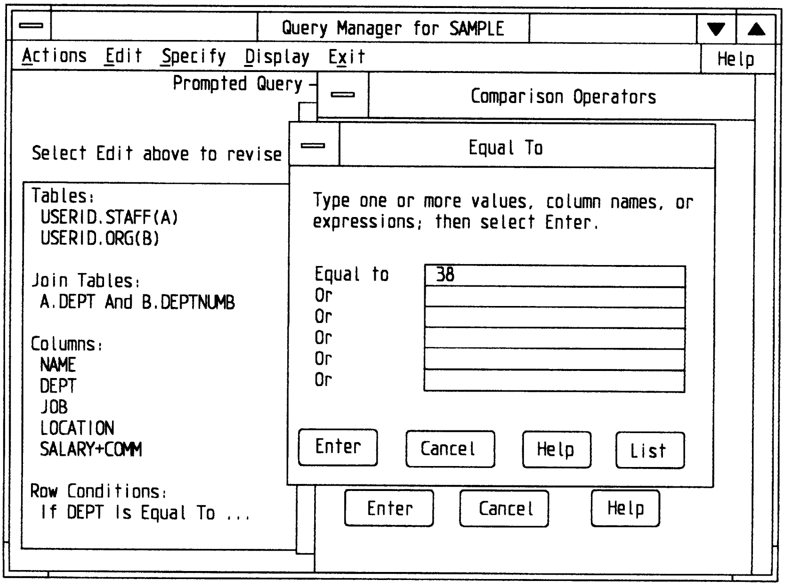
   a. In the Row Conditions menu, select **Expression (A+B, etc.)...**

      If you are using the keyboard, press the Tab key to move to **Expression (A+B, etc.)...** and press the Enter key.

   b. Type the expression in the Expression panel and select Enter. Select List for a list of column names that can be included as part of the expression. Only numeric columns are listed for expressions.

5. Select the appropriate choice for **Verb** and **Comparison** from the Comparison Operators menu and then select Enter.

   Note: NULL means no information exists in that entry field. Null is not equivalent to spaces or zeros. For example, if the row condition SALARY+COMM is NULL, the row is to be retrieved if the expression SALARY+COMM evaluates to NULL. An expression evaluates to null if either value is null. If you select NULL, continue with step 7.



Select Edit above to revise

Tables:
USERID.STAFF(A)
USERID.ORG(B)

Join Tables:
A.DEPT And B.DEPTNUMB

Columns:
NAME
DEPT
JOB
LOCATION
SALARY+COMM

Row Conditions:
If DEPT Is Equal To ...

Type one or more values, column names, or expressions; then select Enter.

Equal to 38
Or
Or
Or
Or
Or

6. Type the value or values for the comparison in the appropriate comparison operator panel and select Enter.

   Notes:

   a. Select List to determine which column names are valid for that comparison operator. **Equal to**, **Less than**, **Greater than**, and **Between** can be a value or column name. Values include expressions or constants.

   b. The values for **Starting with**, **Ending with**, and **Containing** can be character strings up to a maximum of 254 characters. See the *Hints:* that follow these steps for information on typing character strings or constants.

   c. Remember, the characters you type are case-sensitive. Note whether or not the character string or constant is, for example, all uppercase, or uppercase and lowercase.

7. If you want to define another row condition, select **Row Conditions** from the Specify menu and continue with step 8. Otherwise, continue with "To Specify Sort Order for a Query" on page 3-22 or "To Discard Duplicate Rows in a Query" on page 3-24.

8. Select your choice from the Condition Connectors menu to determine how the row conditions are connected. Select **Or** if either row condition can be true or select **And** if both row conditions must be true.

9. Repeat steps 1 through 6 to define the next row condition.

   The results of a query will change as a direct result of changes made to the data rows within the selected table or view. After you have added, changed, or deleted data rows, you can run the same prompted query again for an updated set of results on changes made to the table.

10. Continue with "To Specify Sort Order for a Query" on page 3-22 or "To Discard Duplicate Rows in a Query" on page 3-24, depending on your needs.

Hints:

- You can specify two other comparisons as follows:

Comparison Operation	**Symbols**	**Specify**
Greater than or equal to	>=	**Not less than**
Less than or equal to	<=	**Not greater than**

- When you type character strings or constants, keep in mind the following:
  - Character, date, time, and system date and time constants can be enclosed in single quotes in the query. You can type in these constants according to the default edit code specified in the active profile with or without quotes. When the constant is displayed in the query definition panel, Query Manager automatically supplies the enclosing quotes.
  - You can *enter* data for columns defined with date, time, and decimal character edit codes according to the default edit codes for the current active profile, or the edit code format for SQL. Columns defined with date, time, and decimal character edit codes are displayed in the query definition panel in the SQL edit code format.

    For example, if you use the Profiles task to specify the decimal character data format option to be a comma (,), you can type a numeric constant as `1,10`. When the expression is saved as part of the prompted query definition or SQL query definition, the constant is displayed in the SQL edit code format. In this example, the data is displayed as `1.10`. The SQL edit code formats are as follows:

Edit Code	**Format**
Date	TDY- `(YYYY-MM-DD)`
Time	TTS. `(HH.MM.SS)`
Decimal Character	Period (.)

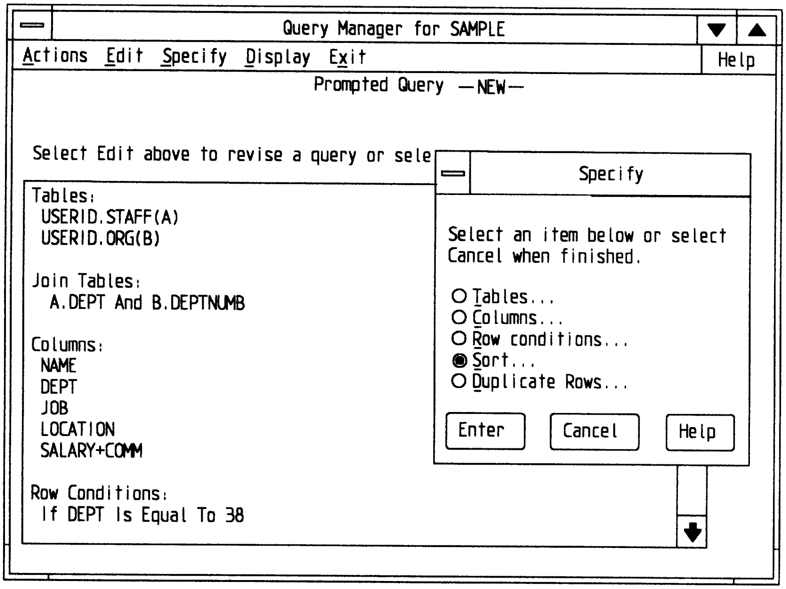
(See “Using Profiles” on page 8-35 for more information on using the Profiles task.)

- If you are using a character constant that could be confused with a numeric constant, or if the character constant is the same as a column name, enclose the character constant in single quotes (' ').
- If you enclose a character constant in single quotes, which itself contains single quotes, be sure to double the internal quotes (two single quotes represent one single quote inside a quoted string). For example, you would type O'Brien as follows:

  ```
  'O''Brien'
  ```
- Single quotes are not necessary for the comparison operators Starts With, Ends With, and Contains.
- If you want to use a column name that could be confused with a constant or an expression, enclose the column name in double quotes (" ").
- You can use SQL global search characters (special characters) such as an underscore (_) or percent sign (%) to search for rows when using the comparison operators Starts With, Ends With, and Contains. An (_) means that any character is allowed to fill that position. A (%) means that zero or more characters are allowed to fill that position. For example, when using Starts With, you can type S for a name column to find all the rows where the name begins with S, or you can type _s to find all the rows where the second character of the name is s.

  Starts With automatically adds % to the end of the character string you type. Ends With automatically adds % to the beginning of the character string you type. Contains automatically encloses the character string you type with percent signs (%).

To Specify Sort Order for a Query



Prompted Query —NEW—

Select Edit above to revise a query or sele

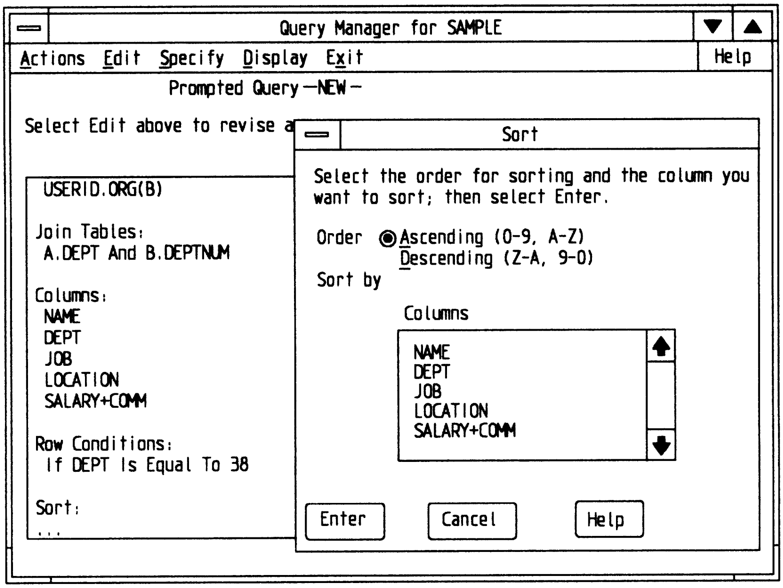
```
Tables:
 USERID.STAFF(A)
 USERID.ORG(B)

Join Tables:
  A.DEPT And B.DEPTNUMB

Columns:
 NAME
 DEPT
 JOB
 LOCATION
 SALARY+COMM

Row Conditions:
 If DEPT Is Equal To 38
```

Select an item below or select Cancel when finished.

○ Tables...
○ Columns...
○ Row conditions...
● Sort...
○ Duplicate Rows...

1. If you want to sort the rows based on the values in a column in your query, select **Sort** from the Specify menu. If the Specify menu is not displayed, select **Specify** from the action bar.

   Note: Sort allows you to specify the order in which rows are displayed in a report.



Prompted Query —NEW—

Select Edit above to revise a

USERID.ORG(B)

Join Tables:
A.DEPT And B.DEPTNUM

Columns:
NAME
DEPT
JOB
LOCATION
SALARY+COMM

Row Conditions:
If DEPT Is Equal To 38

Sort:
...

Select the order for sorting and the column you want to sort; then select Enter.

Order ● Ascending (0-9, A-Z)
○ Descending (Z-A, 9-0)

Sort by

Columns
NAME
DEPT
JOB
LOCATION
SALARY+COMM

2. Select your choice for **Order** and **Columns** from the Sort menu and select Enter. The columns you can select are only those column names, expressions, and summary functions you defined as the columns in your query.

3. Repeat steps 1 and 2 if you want to sort more columns.

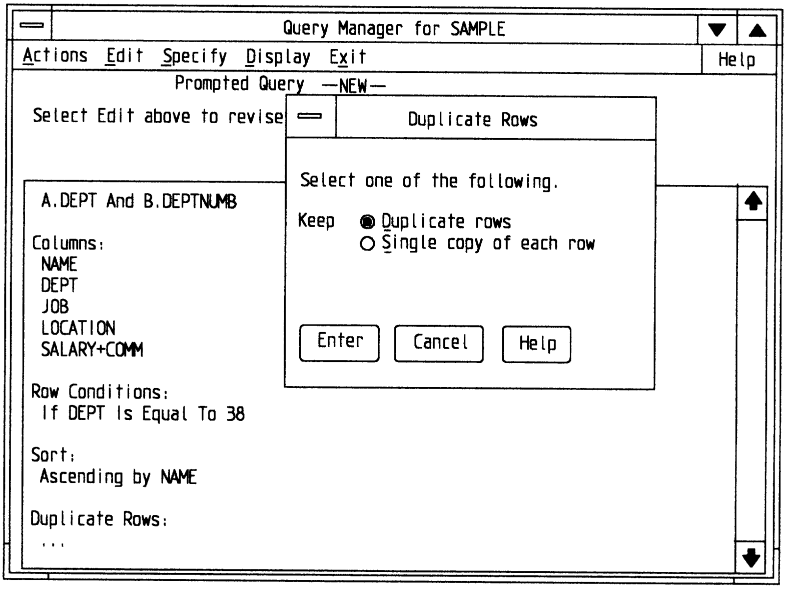
   Note: You can set up a sort hierarchy by sorting more than one column. For example, to sort by last name and then by first name within last name, specify your first column as LASTNAME, then specify a second column as FIRSTNAME. The rows are first sorted by last name, and within like last names, are sorted by first name. Therefore, Brenda Smith is listed in the report before Reed Smith.

To Discard Duplicate Rows in a Query

1. Select **Duplicate rows** from the Specify menu. If the Specify menu is not displayed, select **Specify** from the action bar.

   Note: To be duplicates, the data within all the selected columns of the selected rows must match. For example, if a query selected the columns NAME and DEPT, and two employees had the same last name and were in the same department, duplicate rows would be selected. In this case, you would probably want to keep duplicate rows.

2. Set **Keep** to **Duplicate rows** from the Duplicate Rows menu and select Enter. This is the default.



Select Edit above to revise

A.DEPT And B.DEPTNUMB

Columns:
NAME
DEPT
JOB
LOCATION
SALARY+COMM

Row Conditions:
If DEPT Is Equal To 38

Sort:
Ascending by NAME

Duplicate Rows:
...

Select one of the following.

Keep ● Duplicate rows
○ Single copy of each row

The Prompted Query panel is displayed with your query definition.

3. Continue with "To Run and Save a Query" on page 3-25.

To Run and Save a Query

1. In the Prompted Query panel, select **Actions** from the action bar and then select **Run** from the Actions pull-down, or press the Run (Shift+F1) key.

   Notes:

   a. The data generated by the query is displayed in the Report panel.

   b. If the report does not contain the data you want, you can return to the Prompted Query panel, insert or delete items in the query, and then run it again.

   c. If you have already defined a form for the report generated by a query, you can select **Run using** from the Actions pull-down instead of **Run**. You can then specify the form to use to format the report. For information on defining forms, see Chapter 4.

2. If you want to save or edit the query, select **Display** from the action bar in the Report panel.

3. Select **Query** from the Display pull-down. If you want to save the query, continue with step 6. If you want to edit the query, continue with step 4.

4. Follow the steps in "To Edit a Prompted Query" on page 3-27 to insert, change, or delete any entries in the query.

5. Repeat steps 1 through 3 to run the query again and return to the Prompted Query panel.

6. Select **Exit** from the action bar and then select **Exit Query** from the Exit pull-down, or press the Exit Query (F3) key. The confirmation message is displayed.

7. Select **Save and exit** from the confirmation message. The Save panel is displayed.

   Notes:

   a. If you select **Exit without saving**, the Queries primary menu is displayed again without saving the query.

   b. If you select **Resume**, the Prompted Query panel is displayed again so you can resume the task of defining or changing the query.

8. Type the query name and an optional comment in the Save panel, set **Share** to **Yes** or **No**, and then select Enter.

   Notes:

   a. For information on choosing the Share options, see page 3-2.

   b. For information on naming rules for query names, see Appendix A.

Hints:

- To print the prompted query definition from the Prompted Query panel, select **Print** from the Actions pull-down, or press the Print (F9) key.
- To clear the Prompted Query panel, select **Refresh** from the Actions pull-down, or press the Refresh (F5) key. The Tables panel is displayed so you can begin to redefine the query, and the name is reset to **—NEW—**.
- To retrieve a copy of an existing prompted query into the Prompted Query panel, select **Get** from the Actions pull-down, or press the Get (F2) key. Get replaces the current contents of the Prompted Query panel. Get allows a fully qualified name, for example, USERID.QUERYNAME.
- To become familiar with SQL, you can display the prompted query as the equivalent SQL SELECT statement, select **Show SQL** from the Actions pull-down in the Prompted Query panel, or press the Show SQL (Shift+F7) key. With Show SQL, you can only browse the SQL equivalent of the prompted query. You can still save the query as a prompted query.
- To convert the prompted query into an SQL SELECT statement, select **Convert to SQL** from the Actions pull-down, or press the Convert to SQL (Shift+F4) key. Using SQL, you can make the query more complex by adding items that are not supported in prompted query; for example, UNION and SUBSELECT statements. If you want to convert the prompted query to an SQL SELECT statement *and* save the prompted query version, be sure to save the prompted query before converting; otherwise, you lose the prompted version of the query.

  Once converted, the prompted query is displayed in the SQL Query panel as an SQL SELECT statement. You can save this SQL SELECT statement under the same name as the prompted query just converted or specify a different name. You can also edit the SQL query version; for more information on using SQL queries, see Chapter 9.

- To save the results of a query as a table after you run the query and the report is displayed in the Report panel, select **Save data** from the Actions pull-down, or press the Save Data (Shift+F2) key. If you save the data into a new table, the original column names from the original tables are used as the new column names, unless there are duplicate column names; then, a number is added to the duplicate column name. Before you can save data into a new table, you must be *authorized* to create a table. Turn to "Authorizations" on page 5-1 for information on authorization to create a table.

  If you save the data into an existing table, you can specify if you want to *replace existing data* or *append to existing data.* The existing table must have the correct number of columns and the correct data type for each column. Query Manager will not allow you to save data into a table defined as a parent table since deleting rows in a parent table could result in changing or deleting rows in other tables. Turn to Chapter 5 for information on table relationships.

- The Queries primary menu not only lists prompted queries, but also lists queries created using SQL query. If you select an SQL query, the SQL Query panel is displayed; otherwise, if you select —NEW— or an existing prompted query, the Prompted Query panel is displayed. For more information on SQL queries, see Chapter 9.

To Edit a Prompted Query

1. In the Main Selection menu, select **Queries**.
2. Select the name of the prompted query you want to change from the Queries primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.

Notes:

a. As you edit the query, the changes are reflected in the Prompted Query panel.

b. If you select an SQL query, the SQL Query panel is displayed instead of the Prompted Query panel.

4. To insert a new entry in the query:

   a. Select the existing entry from the Prompted Query panel after which you want to insert the new entry.

   b. Select **Edit** from the action bar and then select **Insert** from the Edit pull-down, or press the Insert (Ctrl+F2) key.

   c. Respond to the menus and panels that follow to insert the new entry. Use the steps for defining queries as you insert the new entry.

5. To change an entry in the query:

   a. Select the existing entry from the Prompted Query panel that you want to change.

   b. Select **Edit** from the action bar and then select **Change** from the Edit pull-down, or press the Change (Ctrl+F1) key.

   c. Respond to the menus and panels that follow to change the entry.

6. To delete an entry from a query:

   a. Select the existing entry in the Prompted Query panel that you want to delete.

   b. Select **Edit** from the action bar and then select **Delete** from the Edit pull-down, or press the Delete (Ctrl+F9) key.

7. Follow the steps in “To Run and Save a Query” on page 3-25 to continue with your work.

Hint:

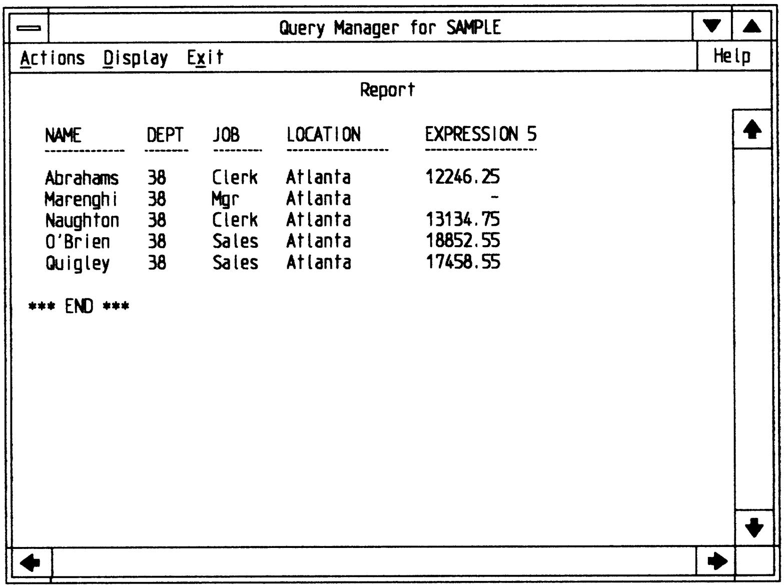
To edit or modify a query you are currently creating, use steps 4 through 6 to insert, change, and delete entries in the query.

Chapter 4. Formatting Reports

This chapter provides detailed information and steps about how to format a report. When you run a query, data is produced and displayed as a *report* in the Report panel. The report is a visual image of that data.

About Formatting Reports

Reports are initially formatted based on a default form that is generated by Query Manager when you are running a query. For example, if you run the simple query, D38, from "Writing Queries" on page 3-3, the report is displayed as follows:

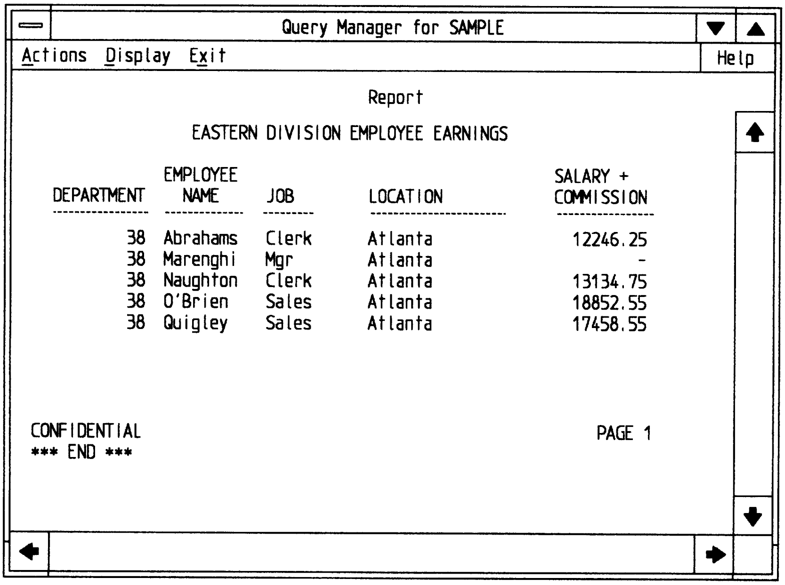


```
NAME          DEPT   JOB     LOCATION        EXPRESSION 5
------------  ------ ------- --------------  ---------------

Abrahams      38     Clerk   Atlanta         12246.25
Marenghi      38     Mgr     Atlanta                -
Naughton      38     Clerk   Atlanta         13134.75
O'Brien       38     Sales   Atlanta         18852.55
Quigley       38     Sales   Atlanta         17458.55

*** END ***
```

Note: Notice the hyphen (-) in the EXPRESSION 5 column for the name Marenghi. The hyphen indicates a null value; reports always use hyphens for null values, even if you have indicated another symbol for null in your profile. One of the values in the EXPRESSION 5 (SALARY+COMM) column evaluated to null because one of the values in the expression (COMM) is null in the original table. In an expression, if one value is null, the expression computes to null.

The *data* displayed in the report cannot be changed except by modifying this query or changing the data in the table and running the query again. However, the way the data is displayed or printed in the report is determined by the report form. When the preceding report was run, the *default form definition* was used to format the data. You can change the appearance of that data, or of the report, by changing the default form definition or by creating a new form.

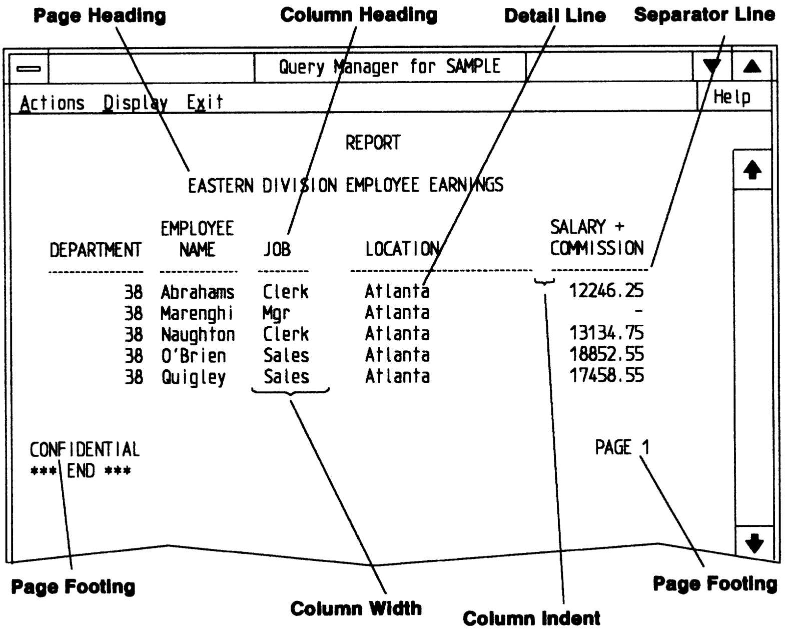
Since the report for query D38, generated using the default form definition, does not provide enough descriptive information, you can modify the default form so that the report appearance meets your needs. The purpose of formatting data in a report is to make the data easier to use and to draw your attention to the value, comparison, or contrast you are most interested in. The following report illustrates the benefits of report forms. This report was also generated using the query D38 from “Writing Queries” on page 3-3 with a form defined especially for the data. Compare this report to the previous report.



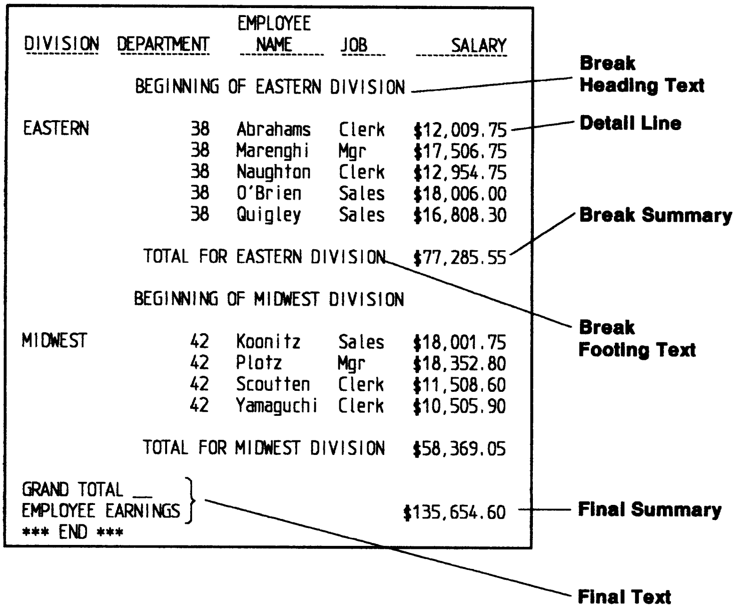
```
               EASTERN DIVISION EMPLOYEE EARNINGS

              EMPLOYEE                                    SALARY +
  DEPARTMENT  NAME       JOB        LOCATION              COMMISSION
  ----------  --------   -----      ----------------      ----------
          38  Abrahams   Clerk      Atlanta                 12246.25
          38  Marenghi   Mgr        Atlanta                        -
          38  Naughton   Clerk      Atlanta                 13134.75
          38  O'Brien    Sales      Atlanta                 18852.55
          38  Quigley    Sales      Atlanta                 17458.55



CONFIDENTIAL                                                 PAGE 1
*** END ***
```

After you modify the default form, you can display the report again in its changed format. If the report is still not quite what you wanted, you can return to the Form panel, change the form, and display the report again. You can repeat this process until you have modified the form enough to generate the report you want. Then, you can save the form for use at a later time. When you run the query the next time, select **Run using** from the Actions pull-down and then specify the particular form you want to use to format the report. If necessary, you can have many different forms for one set of data. You can also define a form without first writing a query and generating a report.

You can enhance the appearance of a report in many different ways. The following illustrations show how you can enhance the format of a report. Each part of the report is represented.



```
                              REPORT

                 EASTERN DIVISION EMPLOYEE EARNINGS

              EMPLOYEE                                  SALARY +
  DEPARTMENT    NAME      JOB       LOCATION            COMMISSION
  ----------  --------  ------  ---------------------  -----------
          38  Abrahams  Clerk   Atlanta                   12246.25
          38  Marenghi  Mgr     Atlanta                          -
          38  Naughton  Clerk   Atlanta                   13134.75
          38  O'Brien   Sales   Atlanta                   18852.55
          38  Quigley   Sales   Atlanta                   17458.55


CONFIDENTIAL                                           PAGE 1
*** END ***
```



```
                             EMPLOYEE
DIVISION  DEPARTMENT           NAME       JOB        SALARY

          BEGINNING OF EASTERN DIVISION

EASTERN             38     Abrahams    Clerk   $12,009.75
                    38     Marenghi    Mgr     $17,506.75
                    38     Naughton    Clerk   $12,954.75
                    38     O'Brien     Sales   $18,006.00
                    38     Quigley     Sales   $16,808.30

          TOTAL FOR EASTERN DIVISION         $77,285.55

          BEGINNING OF MIDWEST DIVISION

MIDWEST             42     Koonitz     Sales   $18,001.75
                    42     Plotz       Mgr     $18,352.80
                    42     Scoutten    Clerk   $11,508.60
                    42     Yamaguchi   Clerk   $10,505.90

          TOTAL FOR MIDWEST DIVISION         $58,369.05

GRAND TOTAL __     }
EMPLOYEE EARNINGS  }                        $135,654.60
*** END ***
```

You can also use the business graphics interface that permits you to install a separately purchased business graphics program that has been written to work with Query Manager. Graphics support can present report data in graph forms such as the bar, pie, and line charts. The business graphics program must be one that is compatible with Query Manager. See "To Graph a Report" on page 4-26 for instructions on how to generate data to graph your report.

When you modify or create a report form, use the information in this chapter as a reference to determine what you need to specify when formatting the report.

Authorization

Before you can display or print the results of a query using a report form, you need authorization to access a database and query a table or view. If the query that uses a saved report form accesses data from several tables, you must be granted Query rows authorization for each table included in the query. Query rows authorization allows you to *run* a query prior to displaying or printing a report.

If you are a user with SYSADM (system administrator) authority, a database administrator for the selected database, or a user with table Control authority for the selected table, you are automatically granted access to the table and have query row authorization. With either administrative authority level or table Control authority, you can use the table Authorizations function to grant or revoke other user's Query rows authorization.

When you create a report form, you become the *owner* of the report form. When you are finished creating and modifying a report and want to exit and save the report form, a Save panel will display to prompt you to type a name and comment and make a **Share** selection. As owner of the report form, you can choose to grant or revoke other users the authority to access the report form by setting **Share** to **Yes** or **No**.

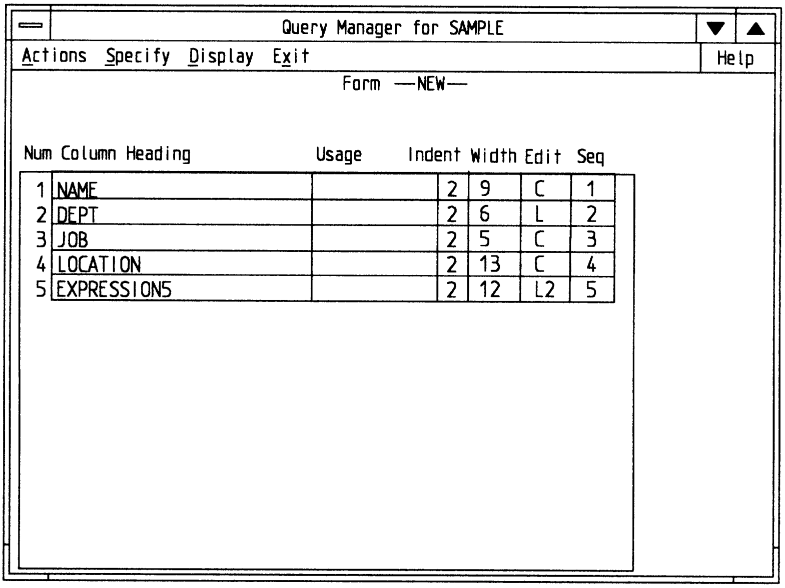
If the report form is saved with **Share** set to **Yes**, any user who can access the database can select and view the report form. If you have Query rows authorization for the selected table or view, you can display the results of a query using a saved report form, export the form, or import the form and assign it a new name. When the report form is saved with **Share** set to **No**, then only the owner, a user with SYSADM authority, or a database administrator for the selected database can select and print the report form, export the form, or import the form with a new name.

You can also select **Get** to retrieve a copy of the report form to edit it and create a new report form saved under a new name. You cannot edit the report form and save it under the same name unless you are the owner of the report form or a user with an administrative authority level.

You can check the authorization granted your user ID to query rows for a table or view by selecting **Authorizations** from the Tools pull-down in the Tables and Views primary menu. For more information and discussion on steps to grant or revoke table and view authorization, see "Using Tables and Views Authorizations" on page 8-28.

Modifying the Default Form

After running the query D38 (from Chapter 3), the Form panel containing the default form is displayed as follows:



Form —NEW—

Num	Column Heading	Usage	Indent	Width	Edit	Seq
1	NAME		2	9	C	1
2	DEPT		2	6	L	2
3	JOB		2	5	C	3
4	LOCATION		2	13	C	4
5	EXPRESSIONS		2	12	L2	5

The headings on the Form panel are used as follows:

Num The order in which each column was selected by the query.

Column Heading The heading for each column in the report. The default headings are the column names from the query.

Usage How each column is used in the report; for example, you could type the usage value OMIT to allow a column to be used in a query but exclude the data from being displayed in the report. Or, you could cause a break point when data is formatted in the report by typing the usage value BREAK1, as shown in the example on page 4-12.

Indent The spacing between the columns in a report. The indent before the first column represents the spacing between the left paper edge or the screen and the beginning of the left edge of the first column.

Width The output character width for a column. Width indicates how many characters should be reserved for displaying the column heading and data.

Edit How data is formatted for displaying or printing.

Seq The order in which each column is displayed or printed in the report.

In the Form panel itself, you indicate how you want the columns and the data for each column to be displayed. You can also use the Specify pull-down (from the action bar) to format headings, footings, and breaks in the report, type final summarizations, specify line spacing, and so on. The Specify pull-down selections are as follows:

- **Page,** for specifying headings and footings
- **Final,** for final summaries and final text
- **Breaks,** for defining headings, footings, and summaries for each break in the report
- **Options,** for specifying line spacing, column wrapping, separators for column headings, break and final summaries, and so on.

Use the following information on column headings, column usage, indent, width, edit codes, and column sequence as you define your report forms.

Column Heading Rules

The following are rules and examples for specifying column headings in report forms.

- A heading can be up to 40 characters long.
- Underscores can be embedded in the heading to indicate a line end for multiple line headings. The maximum number of underscores allowed in one heading is 8. Leading and trailing underscores produce blank lines before and after the column heading. Consecutive underscores (up to 8) produce blank lines. If more

than 8 underscores are in a heading, the remaining underscores are treated as underscore characters.

- When multiple line headings are specified using underscores, the smaller lines are centered within the space of the longest line. For example, `AMOUNT_LAST_INCREASE` is displayed as:

  ```
   AMOUNT
    LAST
  INCREASE
  ```

- Headings for character columns are left-justified and headings for numeric columns are right-justified, within the space allotted for the columns width.

- Default column names must be unique within the first 18 characters. If they are not, Query Manager appends a number to the end of the name. The first occurrence of the column name remains unchanged; the number is appended to all other occurrences of the name. The numbers are assigned sequentially across all column names. For example, if the column names

  ```
  ID DEPT JOB ID DEPT JOB
  ```

  are specified in a query, the default form shows

  ```
  ID DEPT JOB ID1 DEPT2 JOB3
  ```

  as the default column headings.

- When the first 18 characters cannot be made unique, Query Manager assigns `COLn` as the column headings. For example, if the column names

  ```
  ABCDEFGHIJKLMNOPQR ABC ABCDEFGHIJKLMNOPQR
  ```

  are specified in a query, the default form shows

  ```
  ABCDEFGHIJKLMNOPQR ABC COL1
  ```

  as the default column headings.

Column Usage Rules

The following is a chart defining each of the possible values for column usage in report forms. Following the chart are rules and examples for using the usage values.

Note: Only one usage value can be specified for each column in the report. If you need a column to have more than one usage, you need to be sure your query accesses the column more than once. Then, there are multiple entries in the Form panel for the column and you can specify a different usage for each.

Usage Value	Definition
[blank]	Includes this column in the report.
OMIT	Excludes this column from the report.
AVERAGE (AVG)	Averages the values in the column. AVG is the abbreviation used in the Form panel.
COUNT	Counts the number of non-null values in the column.
FIRST	Shows the value of the first row in the column.
LAST	Shows the value of the last row in the column.
MAXIMUM (MAX)	Shows the maximum value in the column. MAX is the abbreviation used in the Form panel.
MINIMUM (MIN)	Shows the minimum value in the column. MIN is the abbreviation used in the Form panel.
SUM	Totals or sums the values in the column.
BREAK1	Breaks the report at this column first. For example, a department number would be a logical break for a report that listed department number, job, and names for the STAFF table. BREAK also allows you to show subtotals when the value of a column changes.

Usage Value	Definition
BREAK2	Breaks the report at this column second. BREAK2 is generated when the column it is defined for changes or when BREAK1 is generated. For example, the job category would be a logical second break for a report that listed department number, job, and names for the STAFF table.
BREAK3—BREAK6	Breaks the report at the columns identified as BREAK3, BREAK4, BREAK5, and BREAK6 (third, fourth, fifth, and sixth, respectively).
BREAK1X—BREAK6X	Breaks the report at the columns identified in the same manner as BREAK1—BREAK6 but *omits* the column data from the report. For example, the salary amount would be a logical break in a report that included department number, job, and names for the STAFF table but also where you would not want to print the actual salary amounts. The *X* indicates that the column data will not print.

Using AVG, COUNT, FIRST, LAST, MAX, MIN, SUM

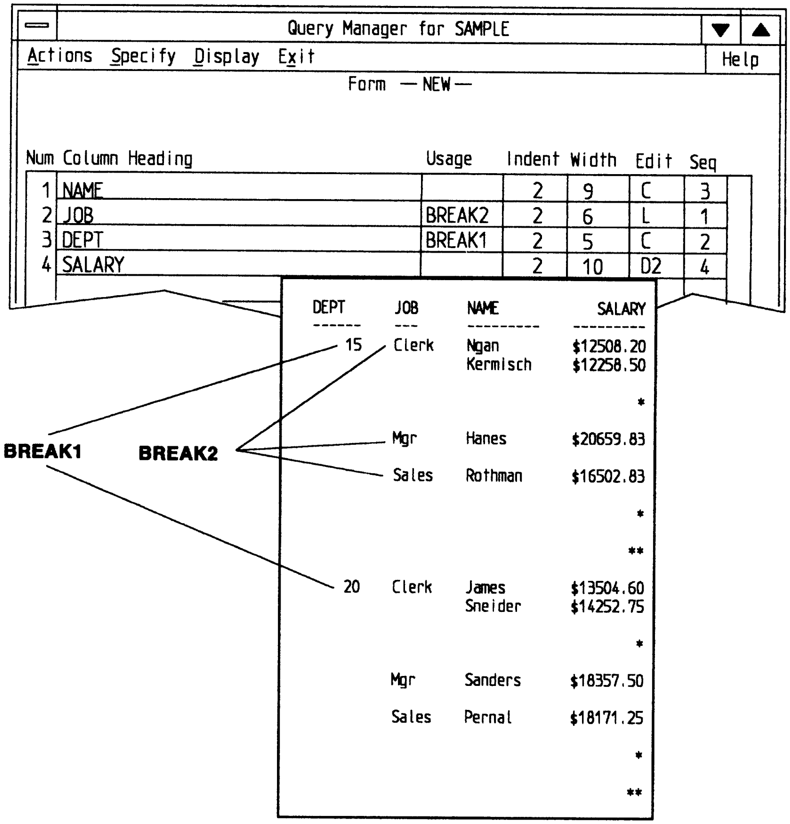
- The result of usage values AVG, COUNT, FIRST, LAST, MAX, MIN, or SUM is provided at the end of the report. They can be displayed in the middle of a report, as subtotals, when another column is defined as a break column. For more information on break columns, see "Using BREAK1—BREAK6" on page 4-11.
- AVG and SUM only work on numeric data. COUNT, FIRST, LAST, MAX, and MIN work on numeric and character data.
- For character data comparisons using MAX and MIN, the short character string is padded with blanks and the character strings are compared based on existing characters. For example, the maximum value of a column containing character strings *ab* and *aaa* would be *ab*.
- Null column values are ignored, unless the usage value for the column is FIRST or LAST.
- If the value in a column of a particular row is null, the report shows the null symbol, a hyphen (-), for that column.

- If the column width is too small for the value generated for the column, the report shows the following: For numeric data, the symbol asterisk (*) for the entire width of the column; for character data, the data is removed on the right.

Using BREAK1—BREAK6

The following is an example of using BREAK. If your query orders a set of rows of employees by department number and job title, a BREAK1 can be used to group the employees by department and a BREAK2 can be used to group the employees by job title within the department. Each time a row with a different job title is read, a BREAK2 would be generated and the appropriate data displayed. Each time a row with a different department number is read, a BREAK2 and BREAK1 would be generated and both sets of the appropriate data would be displayed.

The following report illustrates this use of BREAK1 and BREAK2:

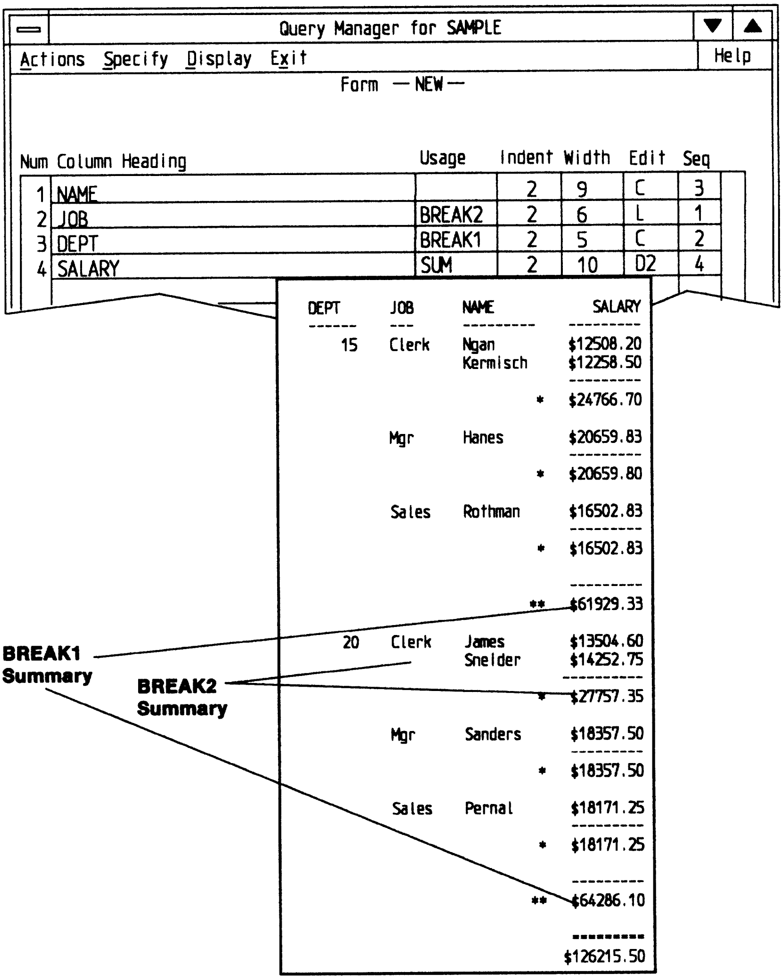


Form — NEW —

Num	Column Heading	Usage	Indent	Width	Edit	Seq
1	NAME		2	9	C	3
2	JOB	BREAK2	2	6	L	1
3	DEPT	BREAK1	2	5	C	2
4	SALARY		2	10	D2	4

```
DEPT     JOB     NAME          SALARY
------   ---     ---------   ----------
   15    Clerk   Ngan         $12508.20
                 Kermisch     $12258.50

                                      *

         Mgr     Hanes        $20659.83

         Sales   Rothman      $16502.83

                                      *

                                     **

   20    Clerk   James        $13504.60
                 Sneider      $14252.75

                                      *

         Mgr     Sanders      $18357.50

         Sales   Pernal       $18171.25

                                      *

                                     **
```

You can also use the same grouping, or breaks, to show the total salaries of all employees within the departments (BREAK1) and use BREAK2 to total the salaries by job title within each department. The following report illustrates this use of BREAK1 and BREAK2:

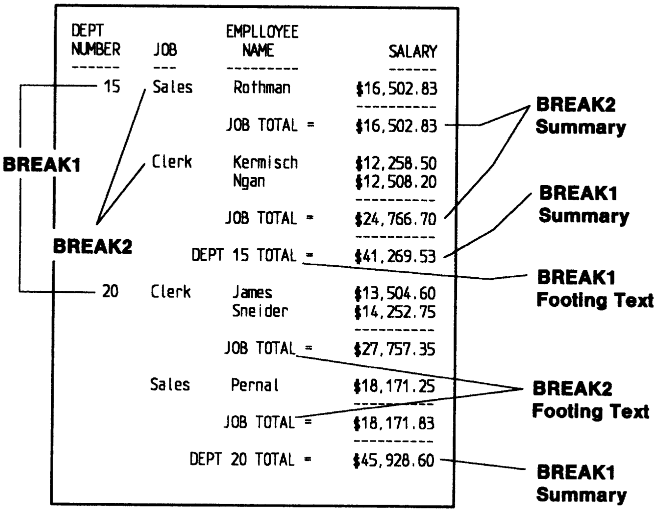


Form — NEW —

Num	Column Heading	Usage	Indent	Width	Edit	Seq
1	NAME		2	9	C	3
2	JOB	BREAK2	2	6	L	1
3	DEPT	BREAK1	2	5	C	2
4	SALARY	SUM	2	10	D2	4

```
DEPT     JOB      NAME            SALARY
------   ---      ---------    ---------
    15   Clerk    Ngan         $12508.20
                  Kermisch     $12258.50
                               ---------
                            *  $24766.70

         Mgr      Hanes        $20659.83
                               ---------
                            *  $20659.80

         Sales    Rothman      $16502.83
                               ---------
                            *  $16502.83

                               ---------
                           **  $61929.33

    20   Clerk    James        $13504.60
                  Snelder      $14252.75
                              ----------
                            *  $27757.35

         Mgr      Sanders      $18357.50
                               ---------
                            *  $18357.50

         Sales    Pernal       $18171.25
                               ---------
                            *  $18171.25

                               ---------
                           **  $64286.10

                               =========
                              $126215.50
```

You can define the same break level on different columns. Also, you do not have to start with BREAK1; you can start with BREAK3 or BREAK5, and so on. The following shows the use of BREAK in a report, including the break summary text:



```
DEPT                EMPLLOYEE
NUMBER   JOB          NAME              SALARY
------   ---        ---------           ------
    15   Sales      Rothman         $16,502.83
                                    ----------
                  JOB TOTAL =       $16,502.83

         Clerk      Kermisch        $12,258.50
                    Ngan            $12,508.20
                                    ----------
                  JOB TOTAL =       $24,766.70
                                    ----------
             DEPT 15 TOTAL =        $41,269.53

    20   Clerk      James           $13,504.60
                    Sneider         $14,252.75
                                    ----------
                  JOB TOTAL =       $27,757.35

         Sales      Pernal          $18,171.25
                                    ----------
                  JOB TOTAL =       $18,171.83
                                    ----------
             DEPT 20 TOTAL =        $45,928.60
```

BREAK1
BREAK2
BREAK2 Summary
BREAK1 Summary
BREAK1 Footing Text
BREAK2 Footing Text
BREAK1 Summary

Notes:

1. If you want each detail line to show the department number and job category, then set **Outlining for break columns** to **No** in the Form Options menu. **Yes** is the default.

2. If you want the BREAK1 footing text to include the department number (DEPT 15 TOTAL and DEPT 20 TOTAL), specify a variable within the footing text by typing `DEPT &N TOTAL` in the Footing Text Line panel, where `N` represents the column number (Num) in the query. In this example, the department number column is 1, so you type `&1` to include the department number in the footing text.

Before each break summary is shown in a report, a line is placed in the report consisting of a row of hyphens (-) under any column with an AVG, COUNT, FIRST, LAST, MAX, MIN, or SUM usage. You can suppress the row of hyphens by setting **Break summary separators** to **No** in the Form Options menu.

Column Indent Rules

The following information can help you when specifying column indent values in report forms:

- You can specify a value for column indent or accept the default.
- The units used for indent are the number of blank characters between the right edge of the immediately previous column or the left edge of the paper or screen and the left edge of the column.
- The default value for indent is 2.
- Indent can be set to a value from 0 to 999.

Column Width Rules

The following information can help you when specifying column width values in report forms:

- You can specify a value for column width.
- The maximum width is 32767 characters.
- If the length of the value to be displayed or printed exceeds the specified width, the value is replaced with a row of asterisks (****) if it is numeric data, or excess characters are removed (truncated) on the right, if it is character data. You can then change the width and display the report again.
- The default for a column width is as long as either the column heading or the column length. The following is a list of the data types and the associated column length:

Data Type	**Default**
Character (fixed)	The length of the column that was defined in the table.
Character (variable)	The (maximum) length of the column that was defined in the table.
Small integer	6.
Large integer	11.

Decimal
The length of the column that was defined in the table, plus 3. For example, if the column length is defined as 7.2, the default width for reports is 10.2.

Date
The width determined by the default date edit code in the Profile panel for the active profile.

Time
The width determined by the default time edit code in the Profile panel for the active profile.

Special data
Only the column heading is displayed. No data is displayed.

System date and time
26.

Scientific notation
10.

Note: For small integer, large integer, and decimal, the left negative sign and right negative sign are assumed to have a length of 1 character. The negative sign is specified in your profile. If the total length for negative signs is greater than 1 character, the default for small integer, large integer, and decimal is increased accordingly. For information on profiles, see "Using Profiles" on page 8-35.

Column Edit Rules

Edit codes are used to format data for displaying and printing. Every field must have an edit code. A blank is not valid for the edit code field. Turn to Appendix I for information on each of the edit codes you can use to specify report forms and to specify panels.

Column Sequence Rules

The column sequence can be specified to set the order in which columns are displayed in the generated report. The following information can help you when specifying column sequence values in report forms:

- When a default form is created, **Seq** defaults to the value of the **Num**.
- The range of sequence values is 1 to 999.
- The numbers do not need to be consecutive.
- Columns with the same sequence value are displayed or printed in the report in the same order that those columns are displayed in the form. The lower sequence number is displayed or printed first, then, if there are two columns that have the same sequence number, they are displayed or printed in the sequential order they display in the form. For example, if the Form panel is filled in as follows:

```
Num  Column heading              ........  Seq
---  --------------                       ---
 1         Column 1                        3
 2         Column 2                        2
 3         Column 3                        3
```

The column headings in the report would be displayed in the following order:

```
Column 2    Column 1    Column 3
--------    --------    --------
```

Using a Default Report Form

The steps that follow assume you have accessed the Report panel after running a query. See "To Run and Save a Query" on page 3-25 for more information on running a query.

To Use and Modify a Default Report Form

1. In the Report panel, select **Display** from the action bar, then select **Form** from the Display pull-down to access the Form panel after you have run a query.

   The Form panel is displayed. Notice that for every column in a report, there is a corresponding item in the Form panel.

2. Type the column heading, usage, indent amount, column width, edit code, and sequence for the columns you want to include in the report.

   Note: Press the Insert Line (Ctrl+F2) key to insert a line after the current line. Press the Delete Line (Ctrl+F9) key to delete a line on the form. These keys cause the **Num** column to be adjusted so it always contains consecutive numbers beginning with 1.

3. To define the format for headings, footings, breaks, summarizations, detail line spacing, and so on, select **Specify** from the action bar.

   Note: You may not need to use all the items from the Specify pull-down to format your report. Use only the ones you need.

4. To specify a page heading for your report, select **Page** from the Specify pull-down.

5. Set **Heading text** to **Yes** in the Page Text panel and select Enter.

6. Type the heading in the Heading Text Lines panel and select Enter. You can also select a different alignment for the title.

   Note: If you want the date, time, or system date and time included in the report text, type `&DATE`, `&TIME`, or `&TIMESTAMP` in the Heading Text Lines panel. For more information on these system variables, see Chapter 15. If you want the page number included in the report text, type `&PAGE` in the Heading Text Lines panel.

   You can also use the `&n` variable to display the first field as it is displayed in a specified column of the report to be displayed in the report text. For example, suppose that the `DEPT NUMBER` column in your report has a sequence of 2 and the first department number to be displayed on the report is 15. By typing `Department &2` in the Heading Text panel, the heading of your report will display `Department 15`.

7. To specify a page footing for your report, select **Page** from the Specify pull-down.

8. Set **Footing text** to **Yes** in the Page Text panel and select Enter.

9. Type the footer text in the Footing Text Lines panel and select Enter.

   Note: If you want the date, time, or system date and time included in the footing text, type `&DATE`, `&TIME`, or `&TIMESTAMP` in the Footing Text Lines panel. For more information on these system variables, see Chapter 15. If you want the page number included in the footing text, type `&PAGE` in the Footing Text Lines panel.

   You can also use the `&n` variable to display the last field, as it is displayed in a specified column of the report, to display in the footing text. For example, suppose that you run a query to find all of the employees for the Accounting department hired for the current year, and you specify a sort in ascending order by starting date. You use a report form that lists the starting date of each employee in the department. The `START DT` column in your report has a sequence of 5 and the last Start Date to display on the report is 10/10/89. By typing `ENDED HIRING FOR DEPARTMENT &5` in the Footing Text panel, the footing text of your report will display `ENDED HIRING FOR DEPARTMENT 10/10/89`.

10. Select **Specify** from the action bar, then select **Final** from the Specify pull-down to specify how you want any summary text and

final summary data to be displayed. Select Enter when you have completed your changes.

Note: You can use up to 12 lines for final text. Special variables can be used to display different values in the final text of your report. By including a column sequence number in the special variable, you can specify &AVGn, &COUNTn, &FIRSTn, &LASTn, &MINIMUMn, &MAXIMUMn, and &SUMn in the Final Text Lines panel. For example, suppose that the Salary column in your report has a column sequence of 3. You would specify &AVG3 in the Final Text Lines panel to display the average for all reported salaries on the bottom of your report.

You can also use the &n variable to display the last field as it is displayed in a specified column of the report to be displayed in the final text of your report.

11. Select **Specify** from the action bar, then select **Breaks** from the Specify pull-down to specify how you want the breaks to be displayed. Select Enter when you have completed your changes.

    Note: You can use the &n variable to display the last field from the break row as it is displayed in the specified column of the report to appear as the Break heading of each Break. For example, suppose the seventh column in your report is Department Name. By typing &7 in the Heading Text Lines panel or in the Heading and Footing Text Lines panel, a break and the department name will display every time the department name changes. (The Heading Text Lines panel is displayed when you set **Heading text** to **Yes** in the Break Options panel. The Heading and Footing Text Lines panel is displayed when you set both **Heading text** and **Footing text** to **Yes** in the Break Options panel.)

    You can also use the &n variable in the Footing Text Lines panel to display a specified column name as the footing text for each page.

12. Select **Specify** from the action bar, then select **Options** from the Specify pull-down to specify how you want miscellaneous form options to be displayed (for example, detail line spacing). Select Enter when you have completed your changes.

    Note: You can use the query-report-form triangle at this time to review the results of the form you have defined. If you first run a query and display a report, then go to the Form panel, you can first select **Display**, then select **Report**, or press the Report

(Shift+F6) key to return to the Report panel and see the reformatted data. You can view the last page of the report by pressing the Last Page (Ctrl+End) key. You can return to the first page by pressing the First Page (Ctrl+Home) key.

You can select **Query** from the Display pull-down, or press the Display Query (Shift+F7) key, to return to the Prompted Query panel (or the SQL Query panel) and generate different data.

13. Select **Exit** from the action bar and then select **Exit Forms** from the Exit pull-down, or press the Exit Form (F3) key. The confirmation message is displayed.

14. Select **Save and exit** from the confirmation message. The Save panel is displayed.

    Notes:

    a. If you select **Exit without saving**, and you began creation from the —**NEW**— specification panel, the Forms primary menu is displayed without saving the form. If you began creating your form from a Queries edit session, you are prompted to return to the Queries primary panel to save the query.

    b. If you select **Resume**, the Form panel is displayed and you can resume the task of editing or defining the form.

15. Type the form name and an optional comment in the Save panel, set **Share** to either **Yes** or **No**, and then select Enter.

    Notes:

    a. You can grant or revoke access to the form you saved with **Share** set to **Yes**. For more information on setting **Share**, see page 4-5.

    b. If you are modifying an existing form, a form name and comment are displayed in the Save panel rather than —**NEW**—. To accept this name and comment, select Enter.

    c. For information on the naming rules for form names, see Appendix A.

Hints:

- To access forms from the Main Selection menu, select **Forms**, then select —**NEW**— from the Forms primary menu and then press the Open (F6) key to display the Form panel.

- To reset the Form panel to the default form values if data exists (as in a query), or to a blank form if no data exists (no query has been run to display the form), select **Refresh** from the Actions pull-down, or press the Refresh (F5) key.
- To retrieve a copy of an existing form into the Form panel, select **Get**, or press the Get (F2) key. A confirmation message is displayed and selecting **Yes** causes the current contents of the Form panel to be replaced.
- To print the form definition from the Form panel, select **Print** from the Actions pull-down, or press the Print (F9) key.
- To access and use copies of forms created under another qualifier, select **Change qualifier for lists** from the Actions pull-down in the Forms primary menu. You can save your changes to the form under the qualifier you changed to. Any forms you create are also prefixed with the qualifier you changed to.

To Run a Query Using a Saved Form

1. After you have saved a form, access the Queries primary menu.

   Notes:

   a. You can access the Queries primary menu from the Main Selection menu.

   b. You can also run the query from the Prompted Query panel while the query is displayed. Access the Prompted Query panel by using the query-report-form triangle (by selecting **Display** from the action bar in either the Form panel or the Report panel).

2. Select the query you want to run.
3. Select **Actions** from the action bar and then select **Run using** from the Actions pull-down.
4. Type the name of the form you want to use to format the report in the Run Using panel.

   The report is displayed in the Report panel.

To Save Data

1. Select **Actions** from the action bar and then select **Save data** from the Actions pull-down. The Save Data panel is displayed.

2. Type your choices and make your selections in the Save Data panel. For information on each item in the Save Data panel, select Help.

   Notes:

   a. You can save the data into a new table by specifying the name of a non-existing table and setting **Data** to **Replace existing data**.

      If you save the data into a new table, the original column names from the original tables are used as the new column names, unless there are duplicate column names; then, a number is added to the duplicate column name. Before you can save data into a new table, you must be *authorized* to create a table. See "Authorizations" on page 5-1 for information on authorization to create a table.

   b. If you save data into an existing table, you can set **Data** to **Replace existing data** or **Append to existing data**. The existing table must have the correct number of columns and the correct data type for each column.

      Since deleting rows from a parent table could result in changing or deleting rows in other tables, Query Manager will only allow you to save data in parent tables when there are no rows present. See Chapter 5 for a discussion of tables defined with referential constraints.

   The data that generated the report is saved.

To Print a Report

1. Access the Report panel after you have run a query. The Report panel is displayed after a prompted query or an SQL query containing a SELECT statement is run. You can define a form for the report as well and use the query-report-form triangle (by selecting **Display** from the action bar) to get to the Report panel.

2. Select **Actions** from the action bar and then select **Print** from the Actions pull-down, or press the Print (F9) key. The Print menu is displayed.

3. Select **To printer** from the Print menu.

4. Type your choices and make your selections in the Print Options panel. For more information on each item in the Print Options panel, select Help.

   Notes:

   a. The items in the Print Options panel correspond to items in the Profile and Printer Nickname panels. See "Using Profiles" on page 8-35 for more information.

   b. If you want the date and time to print on the last line of each page in your report, be sure **Date and time** is set to **Yes**. (**Yes** is the default.)

5. Select Enter to print the report. The Report panel is displayed again.

6. To edit the specified report form or query, select **Display** from the action bar. The Display pull-down is displayed.

7. Select **Form** from the Display pull-down.

   Note: Press the Form (Shift+F5) key to return to the Form panel, or press the Query (Shift+F7) key to return to the Prompted Query Panel (or SQL Query panel).

8. If you want to exit the Form panel, select **Exit** from the action bar and then select **Exit Form** from the Exit pull-down, or press the Exit (F3) key.

   Notes:

   a. If the report form or query was *not* modified, you will return to the query, forms, or Main Selection menu from which you started.

   b. If you modified the query or report form, Query Manager will prompt you with a confirmation message. If you choose to save the query or report form, the Form panel, the Prompted Query, or the SQL Query Panel is displayed.

Hints:

- If you select **To file** from the Print menu, the formatted data for the print image is placed in the OS/2 file you specify. You can then print the OS/2 file using the OS/2 operating system PRINT command.
- To run a query and print a report *without* displaying the report, use the Query Manager RUN command and set the REPORT parameter to NO. Then, use the PRINT REPORT command to print the report. You can issue the RUN and PRINT commands from the Command Line panel. For information on the Command Line panel, see Chapter 10. For information on the RUN and PRINT commands, see your operating system documentation.
- You may notice some differences between the way the report is displayed and the way it is printed. The displayed report is considered to occupy an infinitely large page while the printed report is restricted to a physical paper size.

  Query Manager uses an X(Y) format to number pages when the report extends beyond the available paper width. The Y value contains the page segment number. For example, suppose you have a report that occupies a width of 132 characters and you have defined the width of your physical paper size in Profiles equal to 215 mm (or 8.5 inches). Since this paper width is equal to 80 characters of print, Query Manager will print the first segment of each page of the report and number them 1(1), 2(1), and 3(1) and then print the second segment of each page numbering them 1(2), 2(2) and 3(2). To create the completed report, you would match page 1(1) to page 1(2). If your report, when printed, does not extend beyond the width of your paper, the page segment number and parentheses are not needed and will not print.

  You can set **Print type** to **Compressed** in the Print Options panel to print reports with up to 132 characters onto paper containing only 8.5 inches of width. In some instances, by compressing the print size, you can avoid having pages of your report being split into segments.

To Edit a Report Form

1. In the Main Selection menu, select **Forms**.
2. Select the name of the form you want to change from the Forms primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.

   The Form panel is displayed.
4. Follow steps 2 through 12 in "To Use and Modify a Default Report Form" on page 4-18 to edit the report form.

To Graph a Report

1. Access the report panel after you have run a query. The Report panel is displayed after a prompted query or an SQL query containing a SELECT statement is run.
2. Press the Graph (Shift+F4) key, or select **Actions** from the action bar and then select **Graph** from the Actions pull-down.
3. If you have properly installed the appropriate vendor graphics program, Query Manager will automatically pass the report data for graphing.

   Note: For information concerning business graphics programs, see your IBM Marketing Representative.

Chapter 5. Defining a Table

This chapter contains information and provides steps about the less-frequently performed task of defining tables for your database. In addition, information about specifying indexes is also discussed here.

About Tables

A database can contain one or many tables. You can define and organize groups of tables for your database that *reference* one another. You can think of the data in your database as being arranged in tables consisting of columns and rows. When you define a table, you are defining the columns for that table. You organize table relationships and specify column definitions through the use of *constraints.*

The complete steps to defining a table also include procedures to grant or revoke table authorizations and defining an index.

See Appendix G for information on accessing tables created for migrated Database Manager Version 1.0 or Version 1.1 databases.

Authorizations

Before you can define tables for a database, you must be *authorized* to open a database. See "Opening a Database" on page 2-10 for information and steps on opening a new database.

If you are a user with one of the two administrative authority levels (a user with SYSADM authority or a database administrator for the selected database), you are automatically authorized to open a database and define a table. If you are granted the *Create table* database authorization, you can define a table. To define a table that references another table, you must also have *Reference table* authorization.

Whenever you define a table, you are also automatically granted *table Control* authority. This authority level allows you to control, alter, or delete the table definition and provides the authority to grant or revoke table authorizations. You must, however, be a user with an administrative authority level to use the table Authorizations function to grant or revoke table Control authority. When you are granted an administrative authority level, you also have table Control authority over any existing table on the selected database.

When you have completed defining a table, you should select **Authorizations** from the Tools pull-down in the Table primary menu to grant or revoke other user's authorization to access the defined table in specific ways. You control whether other user's can query rows, add rows, or change data rows stored on the table. You can grant other users the authority to add an index or alter the table definition. You will also use the table Authorizations function to specify whether the defined table can be referenced by another table.

You can check your database authorizations to *create* a table by selecting **Authorizations** from the Tools pull-down in the Databases primary menu. You can check your table authorizations by selecting **Authorizations** from the Tools pull-down in the Tables and Views primary menu. For more information on database and table authorizations, see Chapter 8.

Organizing Table Relationships

With Database Manager, you can define and organize tables that reference one another. Tables that reference one another have a *relationship*. In a relational database, you can express several types of relationships. Consider the relationship between the ORG and STAFF tables as they exist in the sample database.

The ORG Table

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lake	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

The STAFF Table

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sanders	20	Mgr	7	18357.50	-
20	Pernal	20	Sales	8	18171.25	612.45
30	Marenghi	38	Mgr	5	17506.75	-
40	O'Brien	38	Sales	6	18006.00	846.55
50	Hanes	15	Mgr	10	20659.80	-
60	Quigley	38	Sales	-	16808.30	650.25
70	Rothman	15	Sales	7	16502.83	1152.00
80	James	20	Clerk	-	13504.60	128.20
90	Koonitz	42	Sales	6	18001.75	1386.70
100	Plotz	42	Mgr	7	18352.80	-

The ORG table contains the department number, department name, manager, division, and location for each department within the organization. The STAFF table contains the ID number, name, department number, job title, years of service, salary, and commission for each employee. Since any given employee can work in only one department, this relationship is *single-valued* for employees. On the other hand, since one department can have many employees, the relationship is *multi-valued* for departments. The relationship between employees (single-valued) and departments (multi-valued) is a *one-to-many* relationship.

The following are the relationships that can exist.

- One-to-many
- Many-to-one
- One-to-one
- Many-to-many.

The type of a given relationship can vary, depending on the specific environment. If employees of a company can belong to several departments, the relationship between employees would be many-to-many. You should define separate tables for different types of relationships.

When you define tables to contain different organized groups of data, you are *normalizing* relations between tables. Normalization is one method of organizing data into tables with the least amount of repetition and the most flexibility. There are several levels of normalization. For an in-depth discussion of table relationships and normalization, see the *Database Manager Administrator's Guide.*

Each table you create will contain a particular group of data. The relationship between tables will determine the order in which they are created. You should plan to define groups of tables that reference one another using a series of logical steps. To understand the reason behind these steps, you must first understand how you go about creating tables using data and referential *constraints.*

A constraint forces restrictions on columns and tables. Constraints can be used to guarantee that data entry rules are followed and that database relationships are maintained. Tables that properly cross-reference one another are known to have *referential integrity*. The following definitions are useful to understanding referential integrity.

Primary key A primary key is a column or an *ordered* collection of columns whose values uniquely identify a row. To be unique, a value cannot be duplicated in any other row.

Foreign key A foreign key is a column or an *ordered* collection of columns whose values are required to match those of the primary key of a different table.

Parent table A parent table is the table containing the primary key. You can have one or more parent tables within a single database.

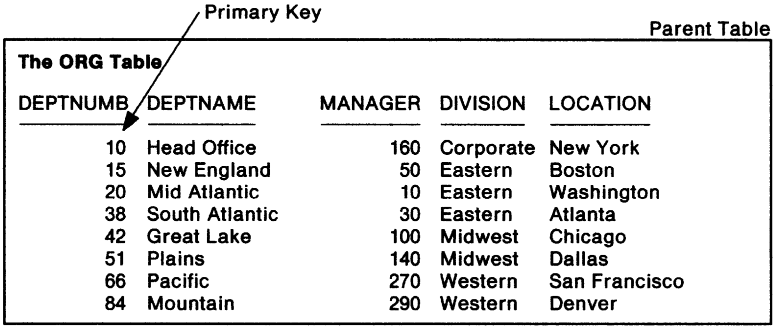
Dependent table A dependent table is a table containing a foreign key. You can have several dependent tables within a single database.

Descendent table A table is a descendent when it references a dependent table containing a foreign key that in turn references a parent table. You can also consider a dependent table to be a descendent table.

Dependent row A dependent row is a row within a dependent table containing a foreign key that matches the primary key value of the corresponding parent table.

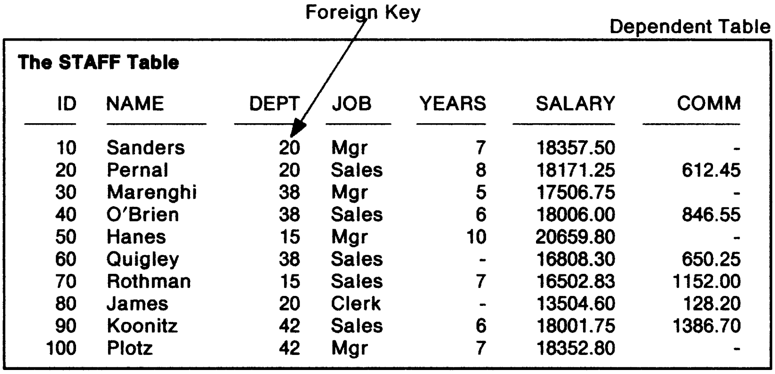
Parent row A parent row is a row of a table that has at least one dependent table row referencing it.

The following illustration shows how the ORG and STAFF table could be defined as a parent and dependent table with matching primary and foreign keys:


Primary Key
Parent Table

The ORG Table

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lake	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver


Foreign Key
Dependent Table

The STAFF Table

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sanders	20	Mgr	7	18357.50	-
20	Pernal	20	Sales	8	18171.25	612.45
30	Marenghi	38	Mgr	5	17506.75	-
40	O'Brien	38	Sales	6	18006.00	846.55
50	Hanes	15	Mgr	10	20659.80	-
60	Quigley	38	Sales	-	16808.30	650.25
70	Rothman	15	Sales	7	16502.83	1152.00
80	James	20	Clerk	-	13504.60	128.20
90	Koonitz	42	Sales	6	18001.75	1386.70
100	Plotz	42	Mgr	7	18352.80	-

The ORG table is created first. You would consider the kind of data you want to record within the table and define the appropriate columns. Since the DEPTNUMB column will represent a single occurrence of a unique value, you would logically specify it as the primary key. The STAFF table is defined next. When you define the columns for the STAFF table, you specify DEPT column with the same attributes as the DEPTNUMB column in the ORG table. When you have finished defining the columns for the STAFF table, you would specify the DEPT column as the foreign key and the ORG table as the parent table containing the matching primary key. By specifying that the foreign key must match the value in the primary key, you are enforcing a *referential constraint* on the ORG and STAFF tables. In this example, this means that for every department number typed for each employee in the STAFF table, there must be a department number with the same value in the ORG table. In other words, you could not assign an employee a nonexistent department number.

Before you begin defining tables, you should plan for data integrity *within* tables and referential integrity *between* tables. You plan for data integrity by specifying column data type, the length of the column, whether data is required, and whether the data typed in is text data. In considering the relationship between tables, you should plan how to use referential constraints. You should define parent tables with specified primary keys before you specify a foreign key in a dependent table. For more information on designing tables with referential integrity, see Appendix C.

Understanding the relationship between tables will help you determine how to define table columns.

Defining Table Columns

The first step to defining a single table or groups of tables that reference one another is to decide what data each table should contain. In a table, each column of a row is related in some way to all the other columns for that row. Columns of a table identify particular types of information. Each column must be of a particular data type, such as character, integer, decimal, date, time, and so on. When you define columns, you are enforcing a set of entry rules to help maintain data integrity.

Remember, rows contain the actual data of the table. The data in a row must conform to the column data types defined for the table. Each row is a sequence of column values that conform to the column data types defined for the table.

Before you can define a table for your database, you need to have some idea of the types of data you want in the table. For example, in the STAFF table the NAME column contains character data, the ID column contains integer data, and the SALARY column contains decimal data. After you have determined the particular types of data the columns in your table should contain, you are ready to begin defining it. For more information on things to consider when defining columns, see Appendix C or the *Database Manager Administrator's Guide*.

The following represents the column descriptions and attributes for the STAFF table as it is displayed within Query Manager:

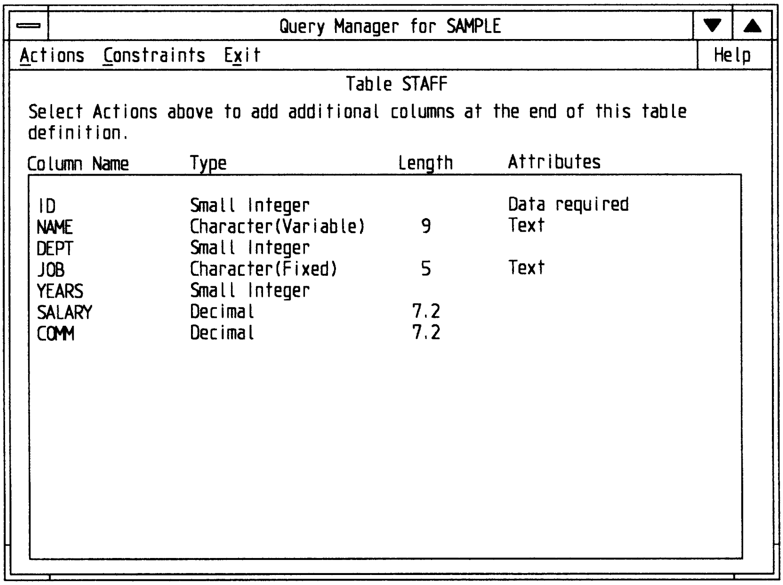


Table STAFF

Select Actions above to add additional columns at the end of this table definition.

Column Name	Type	Length	Attributes
ID	Small Integer		Data required
NAME	Character(Variable)	9	Text
DEPT	Small Integer		
JOB	Character(Fixed)	5	Text
YEARS	Small Integer		
SALARY	Decimal	7.2	
COMM	Decimal	7.2	

By defining the column name, data type, length, and attributes, you are indicating the kind of data your table can contain for each column and what operations can be performed using this data. For example, since numeric data can have arithmetic operations performed on it, you can calculate SALARY plus COMM for a report.

While you are defining a table, you can change a column definition or delete an entire column. Once all the columns are defined and the table definition is saved, you cannot change the column definitions. However, you can add new column definitions to a table that already exists. For example, you can add a column definition for MGRNAME to the ORG table, then edit the ORG table and add a manager's name to each row in the table.

You can also use the definition of an existing table to help you create a new table by using an action called **Get template**. **Get template** places a copy of the table definition of the table name you specify into the Table panel. The existing table is not affected. You can change, delete, or add columns to this table definition as if you were creating a new table definition.

In addition, if a table is no longer useful, you can erase the entire table. When the table is erased, both the table definition and the data in the table are removed.

When you define a column, you must specify the following:

- A column name (up to 18 characters), which must be unique within the table being defined. Columns in different tables can have the same name. The column name must begin with a letter, and can be followed by zero or more letters, digits, or special characters. For information on naming columns and a list of reserved words that should not be used when naming columns, see Appendix A.
- A data type for each column. The data type must be one of the following:

Data Type	**Definition**
Character (fixed length)	Character data that has a fixed number of characters. Each entry is not required to have the same number of characters, but the same number of characters is reserved for each entry whether they are actually used. The maximum length is 254.
Character (variable length)	Character data that varies in the number of characters. No specific number of characters is reserved for each entry. The maximum length is 4000.
Small integer	Small positive or negative whole numbers. The number can be in the range of −32,768 to 32,767.
Large integer	Large positive or negative whole numbers. The number can be in the range of −2,147,483,648 to 2,147,483,647.
Decimal	Numeric data that contains a decimal point. There can be a maximum of 31 digits in the number. If you specify an even number of digits, Database Services rounds up to the next highest odd number when the table is created.

Data Type	**Definition**
Date	A three-part value (year, month, and day) that designates a point in time in accordance with the Gregorian calendar. The range of the values for the year is 1 to 9999, for the month is 1 to 12, and for the day is 1 to n, where n depends on the month: • For months 4, 6, 9, and 11, n is 30. • For months 1, 3, 5, 7, 8, 10, and 12, n is 31. • For month 2, n is 28 except in a leap year, in which case n is 29. • A year that is a multiple of 100 is a leap year only if it is a multiple of 400. • A year that is not a multiple of 100 is a leap year only if it is a multiple of 4.
Time	A three-part value (hour, minute, and second) that designates a time of day in accordance with a 24-hour clock: • The range of hours is 0 to 24. • The range of minutes and seconds is 0 to 59. • If the hour is 24, the other values are 0.
Special data	Data that varies in the number of characters. The maximum number of characters is 32,700 for a column. This type is provided to allow programs to store long amounts of data within a single column of a row table. The data can be non-ASCII (or binary) information. You cannot add, change, display, print, or search special data using Query Manager. This data type is provided so that other programs can make use of these fields.
System date and time (timestamp)	A seven-part value that designates a date and a time as defined previously for date and time, except that the time includes a fractional part. The seven parts are year, month, day, hour, minute, second, and microsecond.

Data Type	Definition
Scientific notation (float)	Floating-point numbers. The number can have 15 significant digits and a value: • Of 0 (typed as `0E0` or `0.0E0`) • In the positive range of +2.225E−307 to +1.79769E+308 • In the negative range of −1.79769E+308 to −2.225E−307.

Note: You can specify how you want these data types to be displayed and printed by using edit codes when formatting reports. For information on edit codes, see Appendix I. For date and time, the edit code specified in the Profile panel for the active profile determines how the data can be typed into a table and the defaults for displaying and printing the data.

Depending on the column data type selected, you may have to specify the following:

- The length of the column.
- Whether data is required (for example, an employee ID or Social Security number would be required, but a middle initial would not). If you indicate that data is not required and nothing is entered, the value is null. You can search for null values in the row condition of a query.
- Whether the data is text (or ASCII) data. This item is used when exchanging data with other non-ASCII (binary) based programs. If the data contains binary information, set **Text** to **No**. For more information on importing and exporting data, see "Using Query Manager Object Tools" on page 8-22.

In addition, you may want to define the columns for the table in a particular order (as in the STAFF table). If you plan to use Query Manager to edit the data in the table, you should consider an order appropriate for the person entering the data.

Note: The name for all tables you define is prefixed with your current active qualifier. If you want to access and use copies of the table definitions defined under another qualifier, you can select **Change qualifier for lists** from the Actions pull-down in the Tables and Views primary menu.

Defining Primary Keys and Foreign Keys

A *primary key* is the column, or set of columns that provide unique identifiers of rows within a table. You must follow certain rules when defining a primary key:

- A table can have only one primary key.
- The values in the primary key must be unique. No two rows of a table can have the same key values.
- When you define a column, you can specify whether or not data is required. Columns must be specified to require data to be used in a primary key definition. If a column is specified to not require data and nothing is entered, the value is null. Null means no information exists in that column. A primary key cannot be null.
- A primary key can be a composite key. A *composite* key is an ordered set of columns.

A column or set of columns that refer to a primary key is called a *foreign key*. For example, the column of department numbers in the STAFF table is a foreign key; it refers to the primary key of the ORG table.

When you define a foreign key, you are applying a referential constraint between two tables. You will be enforcing the rule that a value of a foreign key must be displayed as a value in the primary key of the specified table. The foreign key within the dependent table *references* the primary key within the parent table. The following rules apply to foreign key definitions:

- A table can have many foreign keys.
- A foreign key value is null if any part is null.
- The number of columns in the foreign key must be the same as the number of columns in the primary key of the parent table. In other words, if the primary key is a *composite key*, then the foreign key must also be a composite key with the same number and corresponding sets of column definitions.

When you specify a foreign key, you will also need to specify *delete rules* for the parent table. These rules specify what can happen to dependent rows in dependent tables whenever an attempt is made to delete data in a parent table. You should give careful consideration when choosing a delete rule.

Restrict deletes You cannot delete any rows of the parent table that have dependent rows. For example, in the department-employee relationship that exists between the ORG and STAFF tables, using Restrict deletes would require that you reassign every employee for a department in the STAFF table before you delete the corresponding department value in the ORG table.

Cascade deletes When you delete a row of the parent table, any related rows in dependent table are also deleted.

It is important to note that the *Cascade deletes* is a very powerful rule. Unlike *Restrict deletes* and *Set to null*, Cascade deletes can potentially trickle down many levels of descendents if those descendents also have Cascade delete. Use this rule with caution.

Set to null When you delete a row of a parent table, the corresponding values of the foreign key in any dependent rows are set to null.

Defining Indexes

An *index* provides a way to order the data in a table based on a particular column or columns in the table. The index allows Database Manager to access the data in a table more efficiently by allowing you to define a frequently used sort order for a column or columns.

Indexes are used to access data. For example, you might have an employee ID that you wanted to identify with a particular employee. Given only an employee roster (with names in alphabetical order), the task of locating the appropriate name would be long and tedious; you would need to look at each employee ID, starting at the top of the roster, until you found the right one. If the entries in the roster could be reordered (or indexed) according to employee ID, however, the task could be accomplished more efficiently. Indexes in Query Manager have this effect; they allow direct access to data so that Query Manager can retrieve it without a lengthy sequential search.

You can add, review, or erase indexes in Database Manager. When you use indexes, keep in mind the following:

- Index names follow the same naming rules as tables.
- Indexes are based on the values of data in one or more columns.
- Indexes are used by Database Manager whenever the index presents an efficient path to the data, and therefore you are not required to specify whether an index should be used.
- Any time you edit data in tables, indexes for that table are automatically updated.
- You can define more than one index for a table. However, the more indexes there are for a table, the more indexes Database Manager has to update; a large number of indexes can have a detrimental effect on the performance of editing data.
- You can have a maximum of 16 column names in one index. Column names can be used only once in the same index. Specifying more than one column is useful when you frequently sort data in a particular order; for example, by last name then by first name. You could define an index that specified last name and then first name.
- The order in which you specify column names for an index determines the order in which the information is sorted. If you want to sort data by last name and then by first name, you must type the the column name for the last name before you type the column name for the first name.
- Columns can be indexed in ascending or descending order.
- When you specify an index, you can indicate whether to allow duplicate values in the collection of columns specified by the index. If duplicate values are not allowed, all the rows for the specified collection of columns must contain unique values. For example, to ensure that employee IDs are not duplicated in rows of the STAFF table, you could define a unique index based on the ID column:
  - If a unique index is defined for a table, the uniqueness constraint is enforced whenever rows are added or changed in the table.
  - If a unique index is defined for a column that contains null values, the column can contain no more than one null value.

To Define a Table

Note: The following steps require that your user ID has the necessary authorization to create a table definition.

When defining tables that reference one another, you should define parent tables with primary keys first. It is recommended that you define dependent tables that contain foreign keys after you have completed the parent table definition.

1. In the Main Selection menu, select **Tables and Views**.
2. Select **—NEW—** from the Tables and Views primary menu.
3. Select **Actions** from the action bar and then select **Open definition** from the Actions pull-down, or press the Open Definition (F6) key.
4. Select **Table** from the Open menu to display the Table panel.
5. Select **Actions** from the action bar and then select **Add a column** from the Actions pull-down, or press the Add a Column (Ctrl+F2) key.

   Note: Remember, with Query rows authorization, you can also get the definition of an existing table to use as a template for creating the new table by selecting **Get template** from the Actions pull-down. You can select List to view all the tables you have authorization to select and query.
6. Select a data type for the column from the Column Data Type menu.
7. Type the column name in the Add Column panel.

   Note: A column name must be unique within the table being defined. Columns in different tables can have the same name. For more information on the naming rules for columns, see Appendix A.
8. Indicate your choices for **Length**, **Data required**, and **Text**, if applicable and select Enter. Select Help for information on these items.
9. Repeat steps 6 through 8 until you have defined all the columns in the table. The column definitions are reflected in the Table panel as they are defined.

Note: If you need to change or delete a column definition, you must do so before saving the table definition. For information on changing or deleting a column definition, see "To Change a Column Definition" on page 5-18 or "To Delete a Column Definition" on page 5-19.

10. To specify a primary key for the table definition, continue with steps 11 through 12. To specify a foreign key, perform step 11, and continue with steps 13 through 16.

11. In the Table primary menu, select **Constraints** from the action bar.

    Note: You must have Reference table authority to define a constraint. For more information on Table authorizations, refer to "Using Tables and Views Authorizations" on page 8-28.

12. To add the primary key, select **Add primary key**, or press the Add Primary Key (Ctrl+F5) key.

13. Type the name of the column selected for the primary key in the Add Primary Key panel and select Enter.

    Note: Select List to display a list of column names defined for the selected table that you can select as a primary key.

14. To add a foreign key, select **Add foreign key**, or press the Add Foreign Key (Ctrl+F6) key.

15. Type a name for the foreign key and the parent table in the Add Foreign Key panel and select Enter.

16. To select a delete rule in the Add Foreign Key panel, select **Restrict deletes**, **Cascade deletes**, or **Set to null** and select Enter. The Foreign Key Columns panel is displayed.

    Notes:

    a. You should assign a name to the foreign key that distinguishes it as the foreign key in a particular relationship. Database Manager references this key name when displaying error messages.

    b. You do not need to specify the qualifier for the parent table name.

    c. Select List to display a list of the table names defined for the selected database that you can choose from.

17. Type the column name or the names of a set of columns for the foreign key that matches the primary key in the parent table in the

Foreign Key Columns panel and select Enter. The Table panel is displayed.

Notes:

a. The Foreign Key Columns panel displays the column name or set of column names for the primary key for the specified parent table. If the primary key is a composite key, the foreign key must also be a composite key with the exact same order of columns and corresponding column definitions.

b. Select List to display a list of column names defined for the selected (dependent) table that you can choose from.

18. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

    Note: If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

19. Select **Save and exit** from the confirmation message to save the table definition and select Enter. The Save panel is displayed.

    Notes:

    a. If you select **Exit without saving**, the table definition is not saved and the Tables and Views primary menu is displayed again.

    b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

20. Type the table name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

    Note: For information about the naming rules for table names, see Appendix A.

21. Grant and revoke table authorizations to other users on the table you have defined. See Chapter 8 for steps on using the table Authorizations function available from the Tools pull-down in the Tables and Views primary menu.

    Note: The table cannot be specified as a parent table within a relationship without granting *Reference table* authorization.

22. Define an index for the table. See "To Define an Index" on page 5-28 for steps on defining indexes.

Note: If you defined a primary key for the table, a primary index is automatically created using the specified primary key.

Hints:

- If you want to view the table definition in its SQL statement form, select **Show SQL** from the Actions pull-down, or by pressing the Show SQL (Shift+F7) key.
- You can convert the table definition to an SQL CREATE TABLE statement by selecting **Convert to SQL** from the Actions pull-down or by pressing the Convert (Shift+F4) key. For more information on SQL statements and SQL query, see Chapter 9.
- To print the table definition from the Table panel, select **Print** from the Actions pull-down or press the Print (F9) key.
- If you are planning to import the data for the table, the column definitions should be created in the same order as that of the imported data. For more information on importing data into tables, see "Using Query Manager Object Tools" on page 8-22.
- If you want to have a column in your table that contains zip codes or phone numbers, you should define the column as a character data type. If you define the column as small integer, any zeros at the beginning of the zip code are removed.
- If you have saved the table definition and discover that you need to delete or change columns within the table, you can quickly recreate another table by opening a **—NEW—** table and then selecting **Get template**. Specify the name of the table you want to change as the template for the new table definition, make the desired changes, and then save the table under a new name.

To Change a Column Definition

Note: You cannot change a column definition after it has been saved as part of the table definition. The hint on using Get template described above describes how to quickly create a new table definition.

1. While the Table panel is displayed with the columns that were just defined, select **Actions** from the action bar and then select **Change a column** from the Actions pull-down and select Enter, or press the Change a Column (Ctrl+F1) key.
2. Select the name of the column you want to change from the Change Column menu.

3. If applicable, change the data type for the column in the Column Data Type menu. The current data type is selected otherwise.

4. Make the necessary changes for **Column name**, **Length**, **Data required**, and **Text**, if applicable, in the Change Column panel and select Enter.

5. Repeat steps 1 through 4 until you have changed all the columns you want to change in the table. The changed column definitions are reflected in the Table panel as they are changed.

6. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary is displayed again and you can continue editing the table definition.

7. Select **Save and exit** from the confirmation message to save the changes to the table definition and select Enter. The Save panel is displayed.

   Notes:

   a. If you select **Exit without saving**, the table definition is not saved and the Tables and Views primary menu is displayed again.

   b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

8. Type the table name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

To Delete a Column Definition

Note: You cannot delete a column definition after it has been saved as part of the table definition. The hint on using Get template on page 5-18 describes how to quickly create a new table definition.

1. While the Table panel is displayed with the columns just defined, select **Actions** from the action bar and then select **Delete a column** from the Actions pull-down and select Enter, or press the Delete a Column (Ctrl+F9) key.

2. Select one or more columns to delete from the Delete Columns menu and select Enter. The Table panel is updated to show that the columns were deleted.

3. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

4. Select **Save and exit** from the confirmation message to save the changes to the table definition and select Enter. The Save panel is displayed.

   Notes:

   a. If you select **Exit without saving**, the table definition is not saved and the Tables and Views primary menu is displayed again.

   b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

5. Type the table name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

To Add Columns to a Table Definition

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Tables and Views**.

2. Select the name of the table you want to add columns to from the Tables and Views primary menu.

3. Select **Actions** from the action bar and then select **Open definition** from the Actions pull down and select Enter, or press the Open Definition (F6) key.

4. After the Table panel is displayed, select **Actions** from the action bar and select **Add a column** from the Actions pull-down, or press the Add a Column (Ctrl+F2) key.

5. Select a data type for the column from the Column Data Type menu.

6. Type the column name in the Add Column panel. In addition, indicate your choices for **Length** and **Text**, if applicable, and select Enter.

Note: All columns being added to an existing table definition are defined with **Data required** set to **No**. You cannot change this.

7. Repeat steps 5 and 6 until you have defined all the columns you want to add to the table. The column definitions are added following those in the Table panel as they are defined.

   Note: If you need to change or delete a column definition that you have just added, you must do so before you save the changes to the table definition. For information on changing or deleting a column definition, see "To Change a Column Definition" on page 5-18 or "To Delete a Column Definition" on page 5-19.

8. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

9. Select **Save and exit** from the confirmation message to save the changes to the table definition and select Enter. The Save panel is displayed.

   Notes:

   a. If you select **Exit without saving**, the changes to the table definition are not saved and the Tables and Views primary menu is displayed again.

   b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

10. Type an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

    Note: If you want to add data to the new column, follow the procedure "To Change Data Rows" on page 6-8.

To Erase a Table

Note: The following steps require that your user ID has the necessary authorization to erase a table.

1. In the Main Selection menu, select **Tables and Views**.
2. Select the name of the table you want to delete from the Tables and Views primary menu.

3. Select **Actions** from the action bar and then select **Erase** from the Actions pull-down, or press the Erase (Shift+F8) key. The confirmation message is displayed.

4. Select **Yes** from the confirmation message and select Enter. The table is erased and the table name is deleted from the Tables and Views primary menu.

   Note: When you erase a table, the table definition and the data in the table are removed. All views and indexes dependent on the erased table are also erased. Also, referential constraints held on the primary table by any dependent tables are erased. The data in the dependent table is not erased. If you have any queries that access the table you erased, you must explicitly erase those queries yourself. When you erase a view, only the view definition is removed. The data remains in the original table or tables.

To Add a Primary Key

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Table and Views**.

2. Select the name of the table that you want to add a primary key to from the Tables and Views primary menu.

3. In the Table primary menu, select **Constraints** from the action bar.

4. To add a primary key, select **Add primary key** or press the Add Primary Key (Ctrl+F5) key.

5. Type the primary key column name or the names that comprise the primary key in the Add Primary Key panel and select Enter.

   Note: Select List to display a list of the column names defined for the table you can choose from.

6. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary is displayed again and you can continue editing the table definition.

7. Select **Save and exit** from the confirmation message to save the new primary key in the table definition and select Enter. The table name is displayed in the Save panel.

Notes:

a. If you select **Exit without saving**, the changes to the table definition are not saved and the Tables and Views primary menu is displayed again.

b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

8. Type an optional comment in the Save panel and then select Enter. The Tables and Views primary menu is displayed again.

To Change a Primary Key

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Table and Views**.
2. Select the name of the table for which you want to change a primary key in the Tables and Views primary menu.
3. In the Table primary menu, select **Constraints** from the action bar.
4. Select **Change primary key** from the Constraints pull-down and select Enter.

   The Change Primary Key panel is displayed, listing the column names currently defined as the primary key for the selected table.
5. Position the cursor on the column name you want to change and type the name of another column to specify a new primary key in the Change Primary Key panel and select Enter. The Table primary menu is displayed. The confirmation message is displayed.

   Note: Select List to display a list of the column names defined for the table that you can choose from.
6. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary is displayed again and you can continue editing the table definition.
7. Select **Save and exit** from the confirmation message to save the new primary key in the table definition and select Enter. The table name is displayed in the Save panel.

Notes:

a. If you select **Exit without saving**, the changes are not saved and the Tables and Views primary menu is displayed again.

b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

8. Type an optional comment in the Save panel and then select Enter. The Tables and Views primary menu is displayed again.

To Erase a Primary Key

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Table and Views**.
2. Select the name of the table defined with the primary key you want to erase in the Tables and Views primary menu.
3. In the Table primary menu, select **Constraints** from the action bar.
4. Select **Erase primary key** from the Constraints pull-down and select Enter. The confirmation message is displayed.
5. Select **Yes** from the confirmation message to erase the primary key. Select **No** to return to the Table primary menu to continue editing the selected table.

   Note: If you erase a primary key on an *existing* table, any existing foreign keys used to create a referential constraint against the selected table are erased. The data rows within the dependent tables are not erased. When you erase a primary key previously defined in the same edit session, only the primary key selected is erased.

To Add a Foreign Key

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Table and Views**.
2. Select the name of the table to which you want to add a foreign key in the Tables and Views primary menu.
3. In the Table primary menu, select **Constraints** from the action bar.

4. To add a foreign key, select **Add foreign key** and select Enter, or press the the Add Foreign Key (Ctrl+F6) key.

5. Type a name for the foreign key and the parent table in the Add Foreign Key panel and select Enter. The Foreign Key Columns panel is displayed.

   Notes:

   a. You should assign a name to the foreign key that distinguishes it as the foreign key in a particular relationship. Database Manager references this key name when displaying error messages.

   b. You do not need to specify the qualifier for the parent table name.

   c. In the Add Foreign Key panel, select List to display a list of the table names defined for the database that you can choose from.

6. To select a delete rule in the Add Foreign Key panel, select **Restrict deletes**, **Cascade deletes**, or **Set to null** and select Enter. The Foreign Key Columns panel is displayed.

7. Type the column name or the names of a set of columns of the foreign key that matches the primary key in the parent table in the Foreign Key Columns panel and select Enter. The Table primary menu is displayed.

   Notes:

   a. The Foreign Key Columns panel displays the column name or set of column names of the primary key for the specified parent table.

      If the primary key is a composite key, the foreign key must also be a composite key with the exact same order of columns and corresponding column definitions.

   b. Select List to display a list of column names defined for the selected (dependent) table that you can choose from.

8. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

9. Select **Save and exit** from the confirmation message to save the new foreign key in the table definition and select Enter. The table name is displayed in the Save panel.

   Notes:

   a. If you select **Exit without saving**, the changes to the table definition are not saved and the Tables and Views primary menu is displayed again.

   b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

10. You can type a comment, then select Enter to display the Tables and Views primary menu again.

To Change a Foreign Key

Note: The following steps require that your user ID has the necessary authorization alter a table definition.

1. In the Main Selection menu, select **Tables and Views**.

2. Select the name of the table defined with the foreign key you want to change in the Tables and Views primary menu.

3. In the Table primary menu select **Constraints** from the action bar.

4. Select **Change foreign key** from the Constraints pull-down or press the Change Foreign Key (Ctrl+F7) key. The Change Foreign Key panel is displayed.

   Note: The list contains the names of the foreign keys that are currently defined for the specified table. It does not contain the names of any foreign keys erased during this session.

5. Position the cursor on the foreign key you want to change and select Enter. The Foreign Key panel is displayed.

6. Type the necessary changes for the foreign key name, parent table name, or delete rules in the Foreign Key panel and select Enter. The Foreign Key Columns panel is displayed.

7. In the Foreign Key Columns panel, type the names of the foreign key columns to correspond to the listed primary keys for the selected parent table and select Enter. The Change Foreign Key panel is displayed again. Select Cancel. The Table primary menu is displayed.

Notes:

a. If the primary key is a composite key, you must list the corresponding columns for the composite foreign key in the same order.

b. Select List to display a list of column names defined for the selected table that you can choose from.

8. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

   Note: If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

9. Select **Save and exit** from the confirmation message to save the new foreign key in the table definition and select Enter. The table name is displayed in the Save panel.

   Notes:

   a. If you select **Exit without saving**, the changes are not saved and the Tables and Views primary menu is displayed again.

   b. If you select **Resume**, the Table primary menu is displayed again and you can continue editing the table definition.

10. You can type a comment, then select Enter to display the Tables and Views primary menu again.

To Erase a Foreign Key

Note: The following steps require that your user ID has the necessary authorization to alter a table definition.

1. In the Main Selection menu, select **Table and Views**.
2. Select the name of the table defined with the foreign key that you want to erase in the Tables and Views primary menu.
3. In the Table primary menu, select **Constraints** from the action bar.
4. Select **Erase foreign key** from the Constraints pull-down and select Enter. The Erase Foreign Key panel is displayed.
5. Select the foreign key that you want to erase and select Enter. The Table primary menu is displayed.

To Define an Index

Note: The following steps require that your user ID has the necessary authorization to create an index.

1. In the Main Selection menu, select **Tables and Views**.
2. Select the name of the table for which you want to define an index from the Tables and Views primary menu.
3. Select **Action** from the action bar, and then select **Add index** from the Actions pull-down.
4. Type the column name or names you want in the index and the order for sorting in the Add Index panel. The default sort order is ascending.

   Note: Select List to display a list of the columns defined in the selected table or to display the selections for order that you can choose from.
5. When you have completed typing all the column names you want and their accompanying order, select Enter.
6. Type the index name and select **Duplicates allowed** from the Save Index panel and select Enter.

   Note: For information on the naming rules for index names, see Appendix A.

   The Tables and Views primary menu is displayed.

Hints:

- To review an index for a table from the Tables and Views primary menu, select **Show index** from the Actions pull-down, or press the Show Index (Shift+F7) key. Type the index name in the Show Index panel and select Enter. Then, you can browse the definition of the index. You can also select List to see the indexes defined for that table.
- To erase an index for a table from the Tables and Views primary menu, select **Actions** from the action bar and then select **Erase index** from the Actions pull-down. Then, select the name of the index to be erased from the Erase Index menu. You must have an administrative authority level or be a user with index Control authority to erase an index.

- You should use the Run Statistics tool after you have created or erased an index for a table. For information on Run Statistics, see "Using Query Manager Object Tools" on page 8-22.

Chapter 6. Editing the Data in Tables

This chapter discusses editing data in your tables and provides steps for adding to or changing data in these tables.

About Editing Data

You need to update the data in your database on a regular basis in order to keep it current. For example, you need to update the STAFF table when certain employees get a raise, are transferred, or new, employees are hired. When you edit the data in a table, you are either changing, deleting, or adding information. Query Manager provides several methods to easily search and change the data in tables.

Authorization

Of course, before you can update the data in your database, you or someone in your organization must have created a database and defined tables. You will also need to be *authorized* to access the database and select a table or view. As discussed throughout this volume, you can have one or a combination of several levels of authorization granted to your user ID over a selected database, table, or view.

Your ability to add, change, or delete information is determined by the authorization granted to your user ID for the selected table, view, or other Query Manager object. You may be authorized to select one table on a database but not another. At the same time, you could have authorization to *add* data rows but not have authorization to *delete* data rows for a selected table.

Each chapter of this volume that discusses Database Manager databases, tables, or other Query Manager objects, also provides a section on authorization to create and access the database or object. For an overview on Database Manager authorizations see "Database Manager Authorizations" on page 2-3. For a discussion on steps to granting and revoking authorization, refer to Chapter 8.

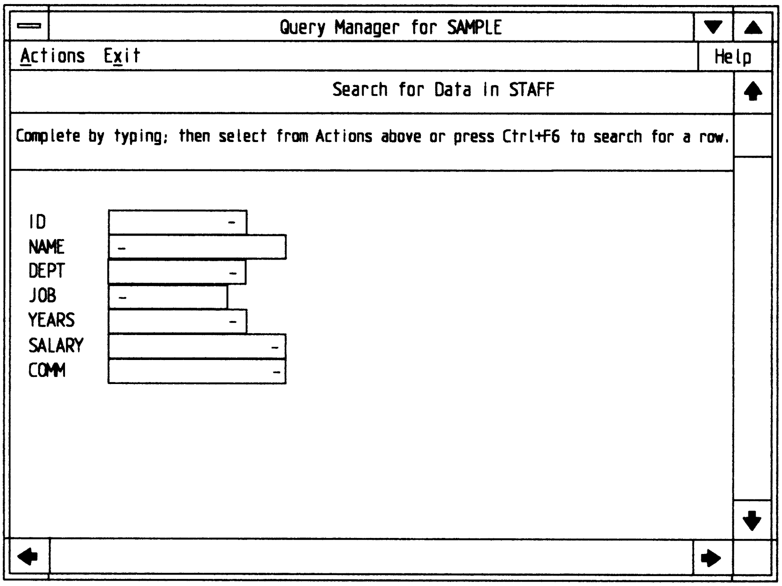
Adding, Changing, and Deleting Data

Data is stored in rows in a table. You can add data rows to a table, change data rows in a table, or delete data rows from a table. With Query Manager, a data row is displayed using a panel. The panel fields are based on the definition of the columns in the table or view. Each column name is displayed with a corresponding entry field for the data itself. If you are adding a data row, the column names are displayed with entry fields that contain only the null character (which by default is a hyphen (-)). You then type the values for each column in that row.

How the tables within your database are defined will determine, in many ways, how you make updates. The enforcement of referential constraints between tables can restrict you from deleting or updating data. The effect of performing deletions on a *parent* table is specified when you define a *foreign key* for a *dependent* table and select a *delete rule* against the selected parent table. You can delete rows from a dependent table without any effect on the parent table. Data can be added to a parent table without concern for the corresponding dependent table. Data cannot be added to a dependent table unless there is a row in the parent table with a *primary key* value equal to the foreign key value you want to add. You may receive an error message when you try to change a column defined as a primary or foreign key.

For more discussion on updating, adding, or deleting data on tables defined with referential constraints, see the *Database Manager Administrator's Guide*. If you need information on *defining* tables or changing the definition of a table (for example, changing a foreign key definition), see Chapter 5.

The following is an example of a panel for searching a row in the STAFF table:



Query Manager for SAMPLE

Search for Data in STAFF

Complete by typing; then select from Actions above or press Ctrl+F6 to search for a row.

```
ID      [          -]
NAME    [-             ]
DEPT    [          -]
JOB     [-        ]
YEARS   [          -]
SALARY  [             -]
COMM    [             -]
```

Note: The null character (displayed in this panel as a hyphen (-)) should not be erased if you do not type a value into the columns. You should only erase the null character from the columns when you type a value into the field to search for data.

When you need to change a single data row, you can type a value for one or more columns and press the Perform Search (Ctrl+F6) key. Each value you type is used as a *search argument*. Query Manager searches the database for rows based on the search arguments. For example, if you needed to change the salary for Rothman, you would type `Rothman` in the **NAME** entry field in the Search for Data in the STAFF panel and then press the Perform Search (Ctrl+F6) key. Remember, the search on characters is case sensitive. If you typed in the search argument for Rothman as `ROTHMAN` when the data was stored in your table as `Rothman`, the row would not be retrieved. The table is searched using the search argument of Rothman, and the row for Rothman is displayed. You can then make the necessary changes. The row for Rothman is a *panel instance*; a panel instance corresponds to a single occurrence of a row in a table.

Search Arguments

You can also use search arguments to change a set of rows, or a *panel set*, in a table. A panel set is one or more panel instances retrieved from the database due to one set of search arguments. The panel set is presented to you for editing one panel instance at a time. For example, if you needed to change all members of department 10, you would type 10 in the **DEPT** entry field and press the Perform Search (Ctrl+F6) key. The table is searched using the search argument of 10 and the first row in the panel set with the department number of 10 is displayed. To change each row in that set one row at a time, type your changes to the row to update that row in the table and press the Change and Next (Ctrl+F1) key.

Extended Search

You may need to identify more flexible search conditions for the rows you want to edit in a table. For example, you want to give everyone that has worked five years or more a bonus commission. You can write a query to retrieve only the rows from the table where the value for YEARS is greater than or equal to five. With *extended search*, you can specify this existing query to retrieve a panel set from the table you are editing. You should ensure that this is a valid query before using it as part of an extended search.

If you create an SQL SELECT statement using SQL query, you can set a variable value to be used by the query when it is run to further tailor the extended search. For example, you could create an SQL SELECT statement that specified a variable for department number, so that each time you edited the table, you could use extended search to select the rows for a particular department. (You cannot use variables in prompted query.) The SQL SELECT statement would be:

```
SELECT *
FROM STAFF
WHERE DEPT=&DEPT
```

When you specify a prompted query or SQL SELECT statement for extended search, keep in mind that Query Manager actually uses only the part of the query beginning with the row conditions or the WHERE clause, respectively.

If the prompted query you want to use for extended search was created with any of the following attributes, it cannot be used for extended search:

- In the Duplicate Rows panel, the **Keep** item is set to **Duplicate rows** (see Chapter 3 for more information).
- A summary function is specified for a column.
- More than one table or view is specified.
- A view that cannot be updated. For more information on views that cannot be updated, see "Read-Only Views" on page 7-7.

If the query was created using SQL query, it cannot be used if the definition contains any of the following:

- An SQL correlation name in the WHERE clause
- The keyword DISTINCT
- A GROUP BY, HAVING or ORDER BY clause
- A FOR UPDATE OF clause following the WHERE clause
- A subquery in the WHERE clause that identifies this table in its FROM clause
- A read-only view. For information on read-only views, see "Read-Only Views" on page 7-7.

  For more information on using SQL queries, refer to *SQL Concepts*.

Valid Values for Editing Data

The valid value you can type for each column in a row is determined by the data type, length, and attributes of that column; these are specified in the definition of the table. Each field has an edit code associated with it based on the column data type. These edit codes determine how you can *input*, or type, data into the fields.

If you type the data incorrectly, you get an error message that indicates you have typed the wrong kind of data for this field. For example, if your table is defined to accept a date formatted according to edit code TDMA- (08-11-88), and you type 08/11/88, an error message is displayed. The hyphen(-) that follows the edit code indicates that you must use a hyphen when entering the data, and not a slash (/).

You can have an edit code of C for *character* data and L and E edit codes for *numeric* data. A field containing dates can have TDM, TDD, TDY, TDYA, TDMA, and TDDA edit codes.

The *default* edit codes for columns within table definitions are displayed in Profiles. These edit codes are known as the *OS/2 default* edit codes. You can create a new profile with different default edit codes. When using the Profiles task (as described in "Using Profiles" on page 8-35) you can activate a new profile containing specified edit code formats. Refer to Appendix I for more detailed information on the use of edit codes for columns within a table definition.

Edit codes are also used for forms to designate how the data is to be *output* in displayed and printed reports. Edit codes for forms are described in Appendix I; read this appendix for more specific information on edit code formats.

When you are adding data rows in tables, remember the following:

- The value for each column in a row is initially displayed as the null character, which, by default, is a hyphen (-). The null character indicates that there is no value for the field. Null is not equivalent to spaces or zeros. Null is only valid for character fields defined with **Data Required** set to **No.** You can change the null character to a pound sign (#) by specifying the # as the null character in your profile. For more information on profiles, see "Using Profiles" on page 8-35.
- The size of the entry field for a column name is determined by the data type and length of the column. Column data type and length are specified in the table definition. For more information on table definitions, see Chapter 5.
- In order to view data in a field that has been defined as character (variable), which has a length greater than 254 bytes, you must position the cursor in the entry field and select **Show field** from the Actions pull-down, or press the Show Field (Shift+F7) key.

To Add Data Rows

1. In the Main Selection menu, select **Tables and Views.**
2. In the Tables and Views primary menu, select the table or view name to which you want to add rows.
3. Select **Actions** from the action bar and then select **Add data rows** from the Actions pull-down, or press the Add Data Rows (Ctrl+F2) key.
4. Type the values for each column in the row in the Add Data into *TABLENAME* panel, where *TABLENAME* is the name of the table or view you selected.

   Note: To print the current row displayed on the screen, select **Print** from the Actions pull-down, or press the Print (F9) key.
5. Select **Add and next** from the Actions pull-down, or press the Add and Next (Ctrl+F2) key.

   Note: An informational message is displayed, indicating the data row was added to the table. This message is automatically removed when you move the cursor to the next entry field.
6. Repeat steps 4 and 5 until all the rows have been added. Remember to select **Add and Next** after you have typed the data for the last row you want to add.
7. Select **Exit** from the action bar and then select **Exit Panel** from the Exit pull-down, or press the Exit Panel (F3) key.

   Note: Be sure you saved the last data entry by selecting **Add and next** from the Actions pull-down prior to performing the next step. If you exit before saving the data, the confirmation message is displayed. If you do not want to save the last entry, select **Exit without saving** from the confirmation message and select Enter. The Tables and Views primary menu is displayed again.

   You can select **Resume** to continue the task of adding data rows in the Add Data into *TABLENAME* panel.

Hints:

- If you do not know the format of the data you need to add to the table, and the table already contains data, select **Change data rows** from the Actions pull-down to access and browse the data in the table.

- To clear the entry fields for the row to the null character, select **Blank panel** from the Actions pull-down or press the Blank Panel (Shift+F10) key.

- To keep a copy of the row values as a template when adding the values for the *next* row, select **Add and keep** from the Actions pull-down, or press the Add and Keep (F7) key to display the row values you just specified again.

- You can also get data into a table by using the Import tool or by using a customized panel. For more information on the Import tool, see "Using Query Manager Object Tools" on page 8-22. For more information on customizing panels, see Chapter 13.

- You can add data to a view that is based on one table. Views based on multiple tables cannot be updated. For more information on views, see Chapter 7.

To Change Data Rows

1. In the Main Selection menu, select **Tables and Views**.

2. In the Tables and Views primary menu, select the table or view name in which you want to change or delete rows.

3. Select **Actions** from the action bar and then select **Change data rows** from the Actions pull-down, or press the Change Data Rows (Ctrl+F1) key.

4. Type a value for one or more of the columns in the Search for Data in *TABLENAME* panel, where *TABLENAME* is the name of the table or view you selected. These values serve as the search arguments for the panel set you want to change.

   Note: You can type special characters like an underscore (_) and a percent sign (%) to search for a panel set. An _ means that any character is allowed to fill that position. A % means that 0 or more characters are allowed to fill that position. Special characters can be used only for character data type fields. For example, you can type `S%` for a NAME column to find all the rows where the name begins with `S`, or you can type `%man` to find all the rows where the name ends with `man`. You can also type `R_thman` to find all the rows where the name is `Rothman`, `Ruthman`, and so on.

5. Select **Perform Search** from the Actions pull-down, or press the Perform Search (Ctrl+F6) key.

The first row found with a value matching the search criteria is displayed. Notice that the instructions in the panel have changed; in the previous steps you were searching for data, now you are changing that data.

6. Type your changes to the row and select **Change and Next** from the Actions pull-down, or press the Change and Next (Ctrl+F1) key.

   If you do not want to change this row, but you do want to display the next row in the panel set, select **Next** from the Actions pull-down, or press the Next (F8) key.

   If you want to delete this row and display the next row in the panel set, select **Delete and Next** from the Actions pull-down, or press the Delete and Next (Ctrl+F9) key.

7. Repeat step 6 until you have changed all rows in the panel set.

   Notes:

   a. If you want to stop making changes before reaching the end of the panel set and search again, select **Search** from the Actions pull-down, or press the Search (Ctrl+F6) key and then repeat steps 4 through 7.

   b. Remember to press the Change and Next (Ctrl+F1) key after you have typed the data for the last row you want to change, or to press the Delete and Next (Ctrl+F9) key to delete the last row that is displayed.

8. Repeat steps 4 through 7 to search and update data that matches another search criteria.

9. Select **Exit** from the action bar and then select **Exit Panel** from the the Exit pull-down, or press the Exit Panel (F3) key. A confirmation message is displayed.

10. Select **Exit without saving** from the confirmation message. the Tables and Views primary menu is displayed again.

    Note: You can select **Resume** to continue the task of changing data rows in the Add Data into *TABLENAME* panel.

Hints:

- If you have added a column to an existing table definition, and you want to add data to that column, you must use the preceding steps. For information on adding columns, see "To Add Columns to a Table Definition" on page 5-20.
- To clear any previous search arguments, select **Blank panel** from the Actions pull-down, or press the Blank Panel (Shift+F10) key. Then, type the next search arguments and press the Perform Search (Ctrl+F6) key.
- To print the current row displayed on the screen, press the Print (F9) key. This corresponds to the **Print** item from the Actions pull-down.

To Use Extended Search

Note: Be sure you have defined the query you want to use for the extended search before you begin. If the query is an SQL SELECT statement, the query can contain a variable. You are prompted for the variable value when you specify the query name. For information on SQL query, see Chapter 9. For information on variables, see Chapter 15.

1. Type the values for each column in the row in the Search for Data in *TABLENAME* panel, where *TABLENAME* is the name of the table or view you selected.
2. Select **Extended Search** from the Actions pull-down, or press the Extended Search (Ctrl+F3) key.
3. Type the query name in the Extended Search panel and select Enter.

   The first row retrieved is displayed.
4. Type your changes to the row and select **Change and next** from the Actions pull-down, or press the Change and Next (Ctrl+F1) key.

   If you do not want to change this row, but you do want to display the next row in the panel set, select **Next** from the Actions pull-down, or press the Next (F8) key.

   If you want to delete this row and display the next row in the panel set, select **Delete and next** from the Actions pull-down, or press the Delete and Next (Ctrl+F9) key.

5. Repeat step 3 until you have changed all the rows in the panel set. Remember to press the Change and Next (Ctrl+F1) key after you have typed the data for the last row you want to change, or to press the Delete and Next (Ctrl+F9) key to delete the last row that is displayed.

6. Repeat steps 1 through 4 if you want to specify another query to search for data.

7. Select **Exit** from the action bar and then select **Exit Panel** from the Exit pull-down, or press the Exit Panel (F3) key.

Chapter 7. Defining Views

This chapter contains information about creating views on your database.

About Views

Query Manager provides views to allow flexibility in the way you see data. A *view* (similar to a window) allows you to access only the data you need. When you define a view of a table, the table is displayed to you as though it contains only the columns and rows you want.

You process the view as though it were a table, even though the view is totally dependent on one or more tables for data. A view can have column names different from the actual names of the corresponding columns in the table. Using a view, you can also order the columns differently than in the table definition. A view contains no data of its own; it is a stored definition of a set of rows and columns.

As the data in a table changes, the data accessible through any views based on the table also changes. You can update the data you see with a view if the view is not read-only. Because a view is derived from a table, when you edit the data in the view, you are really editing data in the table. You may be restricted from editing or adding data to views derived from tables defined with referential constraints. You can use a view, by its name, as you would a table.

For information on views that are read-only, see "Read-Only Views" on page 7-7. For information on updating or editing the data in tables, see Chapter 6.

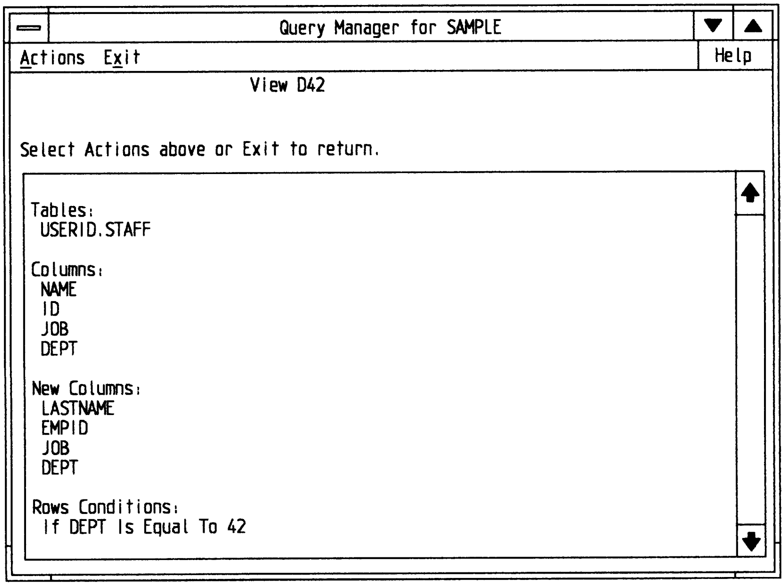
With a view, you can:

- Make a subset of the data in a table available. For example, a view based on the STAFF table might contain rows for a particular department only.
- Combine data from two or more tables and make the combined data available. By specifying a view that matches values in one table to those in another table, you can create a view that presents data from both tables. However, data defined by this type of view

can only be queried. *You cannot add, change, or delete data from a view that joins two tables.*

- Perform functions or operations on data in a table and make the resulting data available. For example, resulting data might be one of the following:
  - Sum of the values in a column
  - Maximum value in a column
  - Average of the values in a column.
- Define a view on another view. A maximum of 15 view and table names (including the name of the view you are creating) can be referenced within one view.

A view is created using procedures similar to those used when creating a prompted query. The selected columns and rows become the columns and rows of the view. In the example view that follows, the columns NAME, ID, and JOB from the STAFF table become the columns LASTNAME, EMPID, and JOB respectively. The following illustration shows the definition for the view that is going to be created.



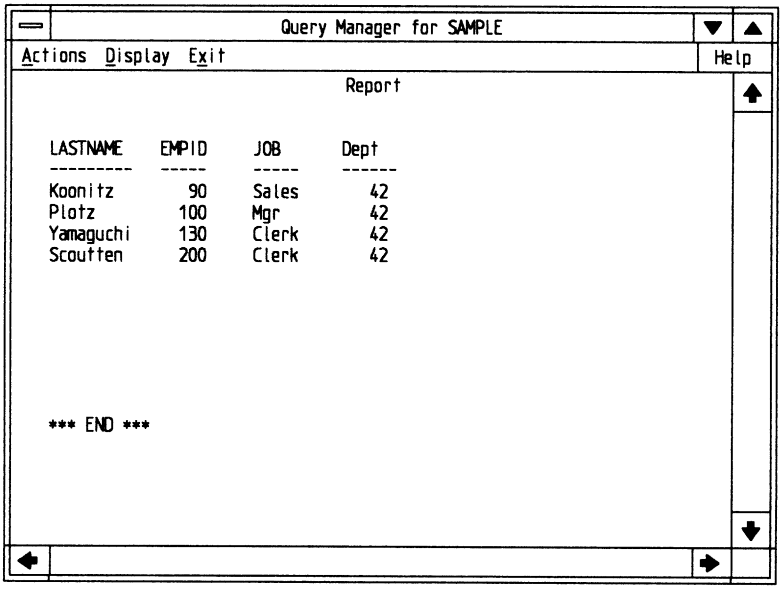
View D42

Select Actions above or Exit to return.

```
Tables:
 USERID.STAFF

Columns:
 NAME
 ID
 JOB
 DEPT

New Columns:
 LASTNAME
 EMPID
 JOB
 DEPT

Rows Conditions:
 If DEPT Is Equal To 42
```

Note: Notice that **Sort** is not available from the Specify menu, as in prompted query and that a new entry for **New Columns** is part of the View panel.

The result of a view for department 42 is displayed as follows when used in a query.



```
                              Report

LASTNAME   EMPID    JOB     Dept
---------  -----    -----   ------
Koonitz       90    Sales       42
Plotz        100    Mgr         42
Yamaguchi    130    Clerk       42
Scoutten     200    Clerk       42



*** END ***
```

To display the results of a query, you must first run an existing query or create and run a query. The data retrieved and displayed in the preceding report are the results of a query that retrieves the rows from a view where the value for department number is 42. You could create another query and select **All** rows. The results of the query would display all the rows of data in the columns defined for the selected view.

Authorization

Before you can define a view, you must be *authorized* to open a database and select a table. As a user with an administrative authority level (a user with SYSADM authority or a database administrator for the selected database), or a user with *table* Control authority, you are automatically authorized to select a table and define a view. In addition, you are automatically authorized to define a view derived from any table for which you are granted *Query rows* authorization.

If you are a user with an administrative authority level, you also have *view* Control authority over any existing view on the selected database. When you create a view, you are granted view Control authority when you have table Control authority for all the tables the view accesses. You cannot increase your authority to access tables as a result of creating a view.

With view Control authority, you can control or delete the view definition. In addition, you can select **Authorizations** from the Tools pull-down in the Tables and Views primary menu to grant and revoke view authorizations. You must, however, be a user with an administrative authority level to use the view Authorizations function to grant or revoke view Control authority.

When you have completed defining a view, you should use the view authorizations function to grant or revoke other user's authorization to access the defined view in specific ways. When you grant a user view authorizations, you allow access only to the particular data columns included in the view. This means you can grant *Query rows, Add rows, Change rows,* or *Delete rows* authorizations on the view derived from a table that users are otherwise denied access to.

If you are not authorized to create a prompted view, you can still have authority to select a view if you are granted Add rows, Change rows, or Delete rows authorization for the selected view.

You can check your table or view authorization by selecting **Authorizations** from the Tools pull-down in the Tables and Views primary menu. For more discussion on table and view authorization, see "Using Tables and Views Authorizations" on page 8-28.

To Define a View

1. In the Main Selection menu, select **Tables and Views**.
2. Select **—NEW—** from the Tables and Views primary menu.
3. Select **Actions** from the action bar and then select **Open definition** from the Actions pull-down, or press the Open Definition (F6) key.
4. Select **View** from the Open menu.

5. To define the tables for the view, see "To Specify Tables for a Query" on page 3-10.

6. To define the columns for the view, see "To Specify Columns for a Query" on page 3-13.

   Note: After you leave the Columns menu, the New Columns panel is displayed so you can rename the selected columns for the view.

7. Type the new column names you want to use in the New Columns panel and select Enter. The columns listed under **Column Name** are the columns you defined in step 6.

   Notes:

   a. A new column name is only required for columns that are defined as summary functions or expressions, or when joining tables that have duplicate column names. If you do not type a new column name, the column name used is the one under **Column Name**.

   b. See Appendix A for the naming rules for column names and a list of SQL reserved words that should not be used when naming columns.

8. To define the row conditions for the view, see "To Specify Row Conditions for a Query" on page 3-17.

9. To specify whether duplicate rows are allowed for the view, see "To Discard Duplicate Rows in a Query" on page 3-24.

10. Select **Exit** from the action bar and then select **Exit view** from the Exit pull-down, or press the Exit View (F3) key. The confirmation message is displayed.

    Note: If you select **Resume**, the View panel is displayed and you can continue editing the view definition.

11. Select **Save and exit** from the confirmation message.

    Notes:

    a. If you select **Exit without saving**, the view definition will not be saved and the Tables and Views primary menu is displayed again.

    b. If you select **Resume**, the View panel is displayed again and you can continue editing the view definition.

12. Type the view name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

    Notes:

    a. For more information on the naming rules for view names, see Appendix A.

    b. Be sure the view is defined as you want it since a view definition cannot be changed once it is saved; after a view is saved, it must be erased and created again if you need to change it.

    c. To use the definition of an existing prompted view to help you create a new view, select **Get template** from the Actions pull-down, or press the Get Template (F2) key.

13. Grant and revoke view authorizations to other users on the view you have defined. See "Using Tables and Views Authorizations" on page 8-28 for steps on using the table and views Authorizations function available from the Tools pull-down in the Tables and Views primary menu.

Hints:

- To clear the View panel, select **Refresh** from the Actions pull-down, or press the Refresh (F5) key. The Tables panel is displayed so you can begin to redefine the view. **Get template** places a copy of the view definition you specify into the View panel. **Get template** is only allowed when creating a new view.

- To browse the view definition in its SQL statement form, select **Show SQL** from the Actions pull-down, or press the Show SQL (Shift+F7) key.

- To optionally convert the new view definition to an SQL CREATE VIEW statement, select **Convert to SQL** from the Actions pull-down, or press the Convert to SQL (Shift+F4) key. You can save this SQL CREATE VIEW statement under the same name as the prompted view you just converted, but you cannot convert it back to a prompted view. For more information on SQL statements and SQL queries, see Chapter 9.

- To print the view definition from the View panel, select **Print** from the Actions pull-down, or press the Print (F9) key.

- Unlike tables, which can only be modified by adding columns to the end, an existing view cannot be modified at all. You can only browse the view definition.
- The name for all views you define is prefixed with the current active qualifier. If you want to browse views defined under another qualifier, you can select the **Change qualifier for lists** item from the Actions pull-down in the Tables and Views primary menu.
- To erase a view, see "To Erase a Table" on page 5-21.
- To import data into a view, see "To Import or Export Tables or Views" on page 8-25.

Read-Only Views

Depending on how a view is defined, you can add, change, and delete data rows in the base table of the view. If a view is defined with any of the following attributes, the view cannot be updated (it is read-only):

- Elimination of duplicate rows
- Usage of a column function
- Joining of two or more tables
- References to a view with any or all of the above attributes.

If the view was created using SQL query, then it is read-only if its definition involves any of the following:

- The first FROM clause identifies more than one table or view.
- The keyword DISTINCT.
- Set operators such as UNION, EXCEPT, or INTERSECT.
- A GROUP BY clause.
- A HAVING clause.
- A column function.
- A subquery such that the base object of the SELECT statement and of the subquery is the same table.
- The first FROM clause identifies a read-only view.

Note: The conditions that define a read-only view using SQL query do not apply to subqueries of the SELECT statement.

You cannot import into a view that is read-only.

Chapter 8. Maintaining Your Database

This chapter provides information about database maintenance tasks that are performed using Query Manager System Tools, Database Tools, and Object Tools. These tasks include importing and exporting data, backing up a database to preserve the data, and creating different profiles for running Database Manager. This chapter also provides information about controlling access to databases, tables, and views through use of the Authorizations function.

About Maintenance

With Query Manager, you can perform tasks that aid in maintaining an efficient database in terms of performance and space utilization. For example, you can reorganize a heavily modified table to aid in accessing the data and then run statistics on the table so that the reorganization is effective. You can back up a database to preserve the data and, if necessary, restore the data later. You can create different profiles for running Query Manager, depending on your needs. Also, you can import data from other database or spreadsheet programs to use in Database Manager, and export Database Manager data for use with another database or spreadsheet program. These tasks can be performed using Query Manager tools and profiles.

In addition, you can control which users can open selected databases, add data, change data, or create and alter tables and views. These tasks can be performed using Database Authorizations and Table and View Authorizations.

Every database created with Database Manager Version 1.0 or Version 1.1 must be migrated before use with Database Manager Version 1.3. A database created with Version 1.2 does not require migration. Query Manager will automatically prompt you to run the migration utility the first time you select a database created under a previous version of Database Manager. See Appendix G for information on using this utility.

Authorization

Your user ID will need certain authorization levels to use the different kinds of Query Manager Tools and Authorization functions. You will need, for example, an administrative authority level (a user with SYSADM authority or a database administrator) to select the Show operational status System Tool. Each section of this chapter that describes a Query Manager tool or Authorization function also discusses the prerequisite level of authority to select the Tool or Authorization function. For additional information on Database Manager authorization levels, see "Database Manager Authorizations" on page 2-3.

Maintenance Tasks

Query Manager tools provide an interface to database utilities, configuration, and printer defaults. Three types of tools are available:

- **System Tools**, from the action bar in the Databases primary menu: These are for optimizing the performance of your database system and for specifying printer parameters. System Tools are not related to a specific database; rather, they are related to the environment that makes up your database system. Depending upon your level of authorization, you have the ability to reconfigure Database Manager, catalog or uncatalog a remote workstation, or view the operational status of a database. For a description of each of the System Tools and the required authorization level required to use them, see "Using Query Manager System Tools" on page 8-3.
- **Database Tools**, from the action bar in the Databases primary menu (after a database is selected): With the required authorization level, you can reconfigure, restore defaults, backup, or restore the local database. Tools are available to catalog and uncatalog both local and remote databases. For a description of each of these Database Tools and the authorization level required to use them, see "Using Query Manager Database Tools" on page 8-7.
- **Object Tools**, from the action bar in primary menus for objects (queries, forms, tables, views, menus, panels, and procedures): These are related to the particular type of object and allow you to import or export that object. The Import and Export tools allow you to exchange objects from one database to another. The

Reorganize and Run Statistics tools can be used to store a table more efficiently and to determine optimal access to data within a table. For a description of each of these Object Tools and the authorization level required to use them, see "Using Query Manager Object Tools" on page 8-22.

Authorizations, which are also available from the Tools pull-down. in the Databases primary menu and the Tables and Views primary menu, provide a way for you to control access over Database Manager databases, tables, and views, or view your authorizations. With the Database Authorization task, you can add or change other user and group IDs authorizations to open a database, create an access plan, and create a table. As a user with SYSADM authority, you can also name another user as a database administrator. With the Table or View Authorization task, you can add or change other user and group IDs authorizations to access a table or view in several ways. With an administrative authority level, you can grant or revoke table and view Control authority. For more information on each Database authorization, see "Using Database Authorizations" on page 8-17. For more information on each Table or View Authorization, see "Using Tables and Views Authorizations" on page 8-28.

With the Query Manager profiles task, you can modify the profile shipped with Query Manager and save the changes in a different profile. Profiles contain information on sign-on parameters, default printing options, data format options, and import and export options. You can also personalize your Query Manager start-up by specifying a profile name. For more information on personalizing your Query Manager start-up, see Appendix B.

Using Query Manager System Tools

System Tools, available only from the Databases primary menu, provide a way for you to change the configuration of your database system. You must be a user with SYSADM (system administrator) authority to reconfigure the Database Manager configuration or restore the Database Manager defaults.

As a user with SYSADM authority, or as a database administrator, you can select the Show operational status System Tool to review the activity of one or several databases. You can use the System Tools to open or erase a printer nickname.

System Tools are also available to catalog and uncatalog remote workstations within a distributed environment. The *User's Guide, Volume 3: Database Manager* addresses configuration for a standalone workstation. Refer to the *Database Manager Administrator's Guide* for discussion and instructions about using System Tools within a distributed environment.

The Database Manager Configuration

The Database Manager configuration contains parameters used by Database Services for tuning the entire database system. These parameters are concerned with the amount of memory available to Database Services, as well as the number of databases that can be used at one time.

If you are working in a distributed environment, Database Manager will display additional parameters in the Reconfigure Database Manager panel. You should refer to the *Database Manager Administrator's Guide* for an in-depth discussion and description of how to define Database Manager Configuration parameters for a distributed environment.

You would want to change the Database Manager configuration if you are planning to use more than three databases at a time. If you need to run four different Query Manager sessions and use a *different* database for each session, then you need to increase the **Maximum number of concurrently active databases** from 3 to 4. If you increase that parameter, you also need to increase the parameter **Maximum number of shared segments** from 20 to at least 26 (6 more for each additional database).

The parameters contained in the Database Manager configuration are as follows: (The name for the parameter used in the *Database Manager Programming Guide and Reference* is in parentheses.)

- **Maximum number of shared segments** (*sqlenseg*)

  This parameter allows you to increase or decrease the number of base operating system shared-memory segments to be used by Database Manager. The number of segments can range from 7 to 802. The default is 20 for standalone workstations. The first database activated needs a minimum of 7 shared segments. Each additional activated database requires at least 6 additional shared segments.

- **Maximum number of concurrently active database** (*numbd*)

  This parameter allows you to increase or decrease the number of concurrently active databases allowed in Database Manager. An active database is one that is currently being used. The value can be 1 to 8. The default is 3.

Printer Nickname Files

A printer nickname file, along with allowing you to assign a nickname for the physical printer, contains printer-specific parameters for your printer; for example, the parameters include the device, type, page size supported, and an optional comment. If you do not want to use the shipped default settings for printer nickname, you should have a printer nickname for each printer you plan to use with Query Manager, and your profile should reference the nickname. The active printer nickname is the one specified in the profile you are currently using. You can create, revise (open), or erase a printer nickname file.

Show Operational Status

As a user with SYSADM authority or as a database administrator, you are authorized to monitor database activity upon the database by use of the Operational Status tool. The database name, nickname, location, and last date and time of backup, along with the number of current connects are listed for review. You can display detailed information for each user connected to the selected database and view the status of the system containing the requested database.

To Change the Database Manager Configuration

1. In the Databases primary menu, select **System** from the action bar.
2. Select **Reconfigure Database Manager** from the System pull-down.
3. Type your changes in the Reconfigure Database Manager panel and select Enter.

Hint:

Select **Restore Database Manager defaults** from the System pull-down if you want to return the Database Manager configuration file to the default settings.

To Specify a Printer Nickname File

1. In the Databases primary menu, select **System** from the action bar.
2. Select **Open printer nickname** from the System pull-down.
3. Type the printer nickname you want to create or revise in the Open Printer Nickname panel and select Enter. (Query Manager automatically appends an extension of .QPN to the nickname.)
4. Make the necessary selections, optionally type a comment in the Printer Nickname panel and then select Enter.

   Notes:

   a. If you set **Printer type** to **Other** in the Printer Nickname panel, you need to manually set the switch settings on your printer for page length, line spacing, and pitch. Query Manager only sets the printer controls for carriage return, line feed, and form feed. You should set the line spacing and pitch values to match those set in your profile.

   b. If you set **Page size** to **Other** in the Printer Nickname panel, the Other Page Sizes panel is displayed. Type your choices for **Page length** and **Page width** and select Enter.
5. Select Cancel from the Open Printer Nickname panel to return to the Databases primary menu.
6. Reactivate the profile that contains this printer nickname. For information on profiles, see "Using Profiles" on page 8-35.

   Note: Be sure the profile you are using references the printer nickname you want to use.

Hints:

- Query Manager does not support manual paper insertion devices. Be sure your printer has continuous paper feed or some type of sheet feed device to handle cut sheet paper.
- If you want to erase a printer nickname file, select **Erase printer nickname** from the System pull-down.

To Show Operational Status

1. In the Databases primary menu, select **System** from the action bar.
2. Select **Show operational status** from the System pull-down.
3. Select an item from the Show Operational Status menu.

   Notes:

   a. **Selected** indicates that you want to view the status of the database selected from the Databases primary menu.

   b. **All local** allows all the databases cataloged to the local workstation to be displayed.

   c. **All on specific drive** allows you to type the drive location for all the databases on a specific drive in the Drive Selection panel. The database status for all specified databases is displayed.
4. Select **Actions** from the action bar.

   Notes:

   a. **Show system status** allows you to view the system status of the workstation at which the specified database resides.

   b. **Show user detail** allows you to view the information for each user connected to the specified database.

Hint:

To print the operational status information, select **Print** from the Operational Status panel, or press the Print (F9) key.

Using Query Manager Database Tools

Database Tools, available only from the Databases primary menu, provide a way for you to reconfigure, backup and restore a local database, restore a local database configuration *defaults* and to catalog and uncatalog a database, and to perform a variety of utilities for database maintenance. The Authorizations function, also available from the Tools pull-down. in the Databases primary menu, allows you to grant or revoke database authorizations to other users or view your own authorizations. See "Using Database Authorizations" on page 8-17 for a discussion of this function.

With the exception of Backup, you must be a user with SYSADM (system administrator) authority to use the following Database Tools:

Reconfigure You should change the parameters of the local database if you want to increase the number of active applications, increase the number of database files open per application, or if you want to modify additional parameters that aid in database performance.

Restore defaults Restore the local database defaults if you want to return the database configuration file back to the default settings.

Backup Back up a database to keep a copy. By keeping a backup copy of a database, you are ensuring a means of recovery from any kind of environment failure. You can back up the entire database or only files that have changed since the last backup. You should back up your databases regularly to ensure that you always have a safe copy of them.

After you have backed up a database, you can create another database on a second workstation, and then *restore* the backed up database in order to move it to another workstation.

Restore database

Restore a database if you want to return the database to a prior backed-up state (for example, if the current database is damaged). You should restore the database beginning with the first diskette in a set of backup diskettes and ending with the last diskette in the set of backup diskettes. If the *original* database has been deleted, it must be recreated before it can be restored.

Catalog Databases created at your workstation are automatically cataloged from the Databases primary menu and are available for use. You should catalog a database if that database was not created at your workstation or if you previously uncataloged that database but did not erase it. To catalog a local database, you must select **—NEW—** from the Databases primary menu before selecting **Catalog database** from the Tools menu. An entry for the database is added to the Databases primary menu. This menu lists all cataloged databases.

Uncatalog Uncatalog a database to remove that database name from the Databases primary menu. You can catalog that database again if necessary.

Object table Maintenance on the Query Manager object table should be performed when extensive creation and deletion of Query Manager objects have occurred. All Query Manager objects (queries, forms, menus, panels, and procedures) are stored in the Query Manager object table. It is suggested that you routinely select **Object table maintenance**. This database tool will automatically re-order the table, eliminate stored deletions, and update the information about accessing the data in the table. By routinely selecting this tool, you may improve system performance.

Query Manager will automatically run the migration utility on databases created under previous versions of Database Manager and accessed for the first time through use of the object table maintenance utility. See Appendix G for more information about migration of Database Manager databases.

Database Configuration

There are three types of database configurations: the *General Database Configuration*, *Database Log Configuration*, and the *Change Heap Segments Configuration*, (available when you select **Reconfigure local database** from the Tools pull-down in the Databases primary menu).

The General Database Configuration contains parameters used by Database Services for accessing and using the selected database. The Change Heap Segments Configuration contains parameters that define the amount of private storage that Database Services can use for each local and remote application program connected to the database. The Database Log Configuration contains parameters that define the size, location, and number of log files used to store records of changes made to the database.

With SYSADM authorization, you can change the defaults for either of these files to meet changing needs; you can also restore the database defaults if necessary. You should not change these defaults if you do not understand their purpose. It is recommended that you refer to the *Database Manager Administrator's Guide* before making any modifications to the database configuration and concurrency considerations when running customized applications in a distributed environment.

The parameters for the General Database Configuration are as follows (the name for the parameter used in the *Database Manager Programming Guide and Reference* is in parentheses):

- **Maximum number of active applications** (*maxappls*)

  This parameter allows you to increase or decrease the number of applications that can concurrently connect to the database. The value can be from 1 to 117 applications. The default is 8 applications.

  If the parameter *maxappls* is set to 1 and an application program is already connected to the database, the application program cannot execute any utilities. See the parameters *maxtotfilop* and *buffpage* for other considerations.

- **Maximum number of database files open per application** (*maxfilop*)

  This parameter allows you to indicate the maximum number of base operating system files that can be used by this database. Database Manager opens and closes files as necessary, but never maintains more open files than the value you specify. The greater the value you specify, the less frequently Database Manager must open and close files. The value can be from 2 to 235 database files open per application. The default is 20 database files open per application.

  The total number of open files for all concurrently running OS/2 application programs on Database Manager can be no more than the parameter *maxtotfilop*. The smaller the number of files specified by the parameter *maxfilop*, the more application programs that can run concurrently.

  See the parameter *maxtotfilop* when changing this parameter.

- **Maximum number of total files open per application** (*maxtotfilop*)

  This parameter allows you to indicate the total number of OS/2 files that can be open per application. This includes all database and all user files. Database Manager opens and closes files as necessary and uses this value as the maximum. The greater the value you specify, the less frequently Database Manager must open and close files. The value can be from 25 to 32700 total files open per application. The default is 255 total files open per application. The value of the parameter *maxtotfilop* cannot be more than allowed by the OS/2 version number.

- **Buffer pool size (# of 4K pages)** (*buffpage*)

  This parameter allows you to increase or decrease the number of 4K pages used for a buffer pool. A buffer pool is an area of memory into which database records (rows) are read and updated. Records stay in the buffer pool until the database is no longer being used, the last active write transaction commits, or the space is needed in the buffer pool for other records. The purpose of the buffer pool is to improve database system performance. Because data can be accessed much faster in memory than it can on disk, the fewer times Database Manager has to read from or write to a disk, the better the performance.

  The greater the number of pages, the more data Database Manager can keep in memory. This may improve performance. The value can be from 2 times the maximum number of active applications (*maxappls*) to 1500. The default is 16 pages.

- **Time interval for checking deadlock (# of seconds)** (*dlchktime*)

  This parameter allows you to increase or decrease the time interval in seconds for checking deadlocks. The deadlock interval indicates how often deadlocks are checked. Increasing this parameter decreases the frequency of checking for deadlocks, thereby increasing the time applications must wait for the deadlock to be resolved. Decreasing this parameter increases the frequency of checking for deadlocks, thereby decreasing the time applications must wait for the deadlock to be resolved but increasing the time the database system takes to check for deadlocks. If the value you specify is too small, runtime performance can be affected since Database Manager frequently performs deadlock detection. The value can be from 1 to 600 seconds. The default is 10 seconds.

- **Maximum storage for lock lists (# of 4K pages)** (*locklist*)

  This parameter indicates the amount of storage in 4KB pages that is allocated to the lock list.

  This parameter allows you to increase or decrease the number of table locks and row locks all applications concurrently connected to the database can lock. For example, when more than one application (or session of Query Manager) is accessing the same database at the same time, each application is running in a separate process. A *lock* prevents two applications from updating the same record (row) at the same time. When one user is updating a row, the other user is locked out.

  The size of the lock list affects the maximum number of locks available for each application concurrently connected to the database. If this number is small, each application is allocated fewer locks; then escalation can occur more frequently resulting in more table locks, fewer row locks, and possibly less concurrency on shared objects in the database. If the lock list value is too small or there are several active applications concurrently connected to the same database, you may use up all the locks. If this occurs, increase the value for the lock list or decrease the number of concurrently active applications. The value can be from 4 to 250 pages. The default is 8 pages.

- **Maximum percent of lock lists allowed per application** (*maxlocks*)

  This parameter indicates the amount of lock lists permitted for an application program. The maximum value of lock lists can be from 1 to 100 percent. The default value for each application within a database is 200 /(*maxappls* + 1).

  Changing the value of the parameter *maxappls* without changing the parameter *maxlocks* in the same call causes the system to change the parameter *maxlocks* to the default value, based on the new value of the parameter *maxappls*.

- **Copy protection** (*sqlf_dets*)

  This parameter allows the user to enable (Yes) and disable (No) the copy-protection attribute for the database. When copy protection is enabled, a database cannot be used on a workstation other than the one on which it was created. When copy protection is disabled, a database can be moved to another workstation and used on that workstation. The default is set to enable (Yes).

The following are the parameters for the Change Heap Segments Configuration.

- **Default application heap (# of segments)** (*applheapsz*)

  This item allows you to increase or decrease the amount of memory that this database needs or uses. The application heap is the amount of memory allocated for use by Database Services for an application currently connected to the database. The value can be from 2 to 20 segments. The default is 3 segments.

- **Application agent heap (# of segments)** (*agentheap*)

  The agent heap is a private data area on the Database Server allocated to each remote application connected to the database. This parameter applies only if you are working in a distributed environment. The value can be from 2 to 85 segments. The default value is 2 segments. Turn to the *Database Manager Administrator's Guide* for additional information on setting this parameter.

- **Sort list heap (# of segments)** (*sortheap*)

  This parameter allows you to increase or decrease the heap size for the sort list in each segment. The value can be from 1 to 20. The default is 2 segments.

- **Database heap (# of segments)** (*dbheap*)

  This parameter allows you to increase or decrease the available memory shared by all applications currently connected to the database. The value can be from 1 to 45 segments. The default is 1 segment.

- **SQL statement heap size (# of segments)** (*stmtheap*)

  This parameter indicates the amount of storage used for compiling SQL statements. The value can be from 8 to 255 segments. The default value is 64 segments.

Previous versions of Database Manager used a linear type of recovery log file. Database Manager Version 1.3 uses circular recovery log files. The configuration parameters affecting the recovery log files have changed. Database Manager Version 1.3 converts the log file parameters for the different versions of Database Manager.

The following list contains the Database Manager Version 1.0 and Version 1.1 configuration parameters that affect the recovery log files:

- Initial log file size (*logfile*)
- Log file extension size (*logext*)
- Number of log file extensions (*logmaxet*).

Refer to the *Database Manager Administrator's Guide* for a description of the algorithms used by Database Manager Version 1.3 to convert previous versions of log file parameter values.

The following Database Log Configuration parameters affect the recovery log files for Database Manager Version 1.3:

- **Log file size (# of 4K pages)** (*logfilsiz*)

  This parameter allows you to increase or decrease the amount of disk storage allocated for each log file.

  The size of the log files can affect the amount of changes that can be made by each application program concurrently connected to the database. As changes to a table are made by an application program, records are written to the recovery log file, which contains a record of the changes made. If the application program has been completed successfully, the changes are committed to the database. If the application program fails, the changes are rolled back so that the table remains as it was before the application programs began. If there are many changes or many concurrent application programs, the log files can become full.

  When a database is created, each primary log file is created and allocated the space indicated by the parameter *logfilsiz*. When the primary log files become full, records are stored in the secondary log files. The secondary log files are created as empty files. The secondary files are extended to the size specified by the parameter *logfilsiz* as they are needed, and they are truncated back to empty when the space is no longer needed.

  The value can be from 15 to 4095 pages. The default value is 50.

- **Number of primary log files** (*logprimary*)

  This parameter allows you to increase or decrease the number of primary log files. The primary log files establish a fixed amount of storage dedicated to the recovery log files. When a database is created, the primary log files are created and allocated the storage indicated by *logfilsiz*. Changes to data are recorded in the recovery log files until they are committed to the database. When the changes are committed, the primary log files remain the same size.

The number of primary log files can be set to 0 for databases that are not frequently used. More time will be required each time the database is opened, since the secondary log files will need to be extended. The number of primary and secondary log files must comply with the following equation:

$0 < (logprimary + logsecond) < 64$

The value can be from 0 to 63 files. The default value is 2 files.

- **Number of secondary log files:** (*logsecond*)

  This parameter allows you to increase or decrease the number of secondary log files. The secondary log files establish an amount of storage that is used for the recovery log files as needed. When a database is created, the secondary log files are created as empty files. Changes to the data are recorded in the primary log files until they become full. Then the *secondary* log files are extended to the size specified by *logfilsiz* as needed. When the changes are committed, the secondary log files are truncated to empty files. The value can be from 0 to 63 files. The default value is 3 files.

- **Log records to write before a soft checkpoint** (*softmax*)

  This parameter allows you to increase or decrease the frequency of a soft checkpoint. *Soft checkpoints* are taken periodically by Database Services. A soft checkpoint updates the log file with information that determines the point in the log file where forward recovery processing can begin. If a database system failure occurs, then you must start Query Manager again. The log is used to restore the database to the point of failure.

  This parameter sets the frequency at which checkpoints are taken by specifying the number of records to be written to the log between checkpoints. Increasing this value increases the duration between soft checkpoints, which slows restart time. Decreasing this value lessens the duration between soft checkpoints, which may slow normal runtime. The value can be from 0 to 65535 records. The default is 100 records. The value of 0 means that no soft checkpoint is made.

- **Current location of log** (*logpath*)

  When a database is created, the recovery log file for it is created in the directory containing the database. The parameter *logpath* is set to null, indicating the default path and the parameter *newlogpath* is set to null.

- **New location of log** (*newlogpath*)

  The log path must not be changed for remote workstations that have the LAN server and that use remote program execution to function as Database Standalone workstations.

  The log path can be changed for other workstation configurations. If you change the log path through a configuration change, the new path is stored in the parameter *newlogpath*. After all users have stopped using the database, the next time the database is accessed, the log file is moved to the new location. The parameter *logpath* is changed to the new path, and the parameter *newlogpath* is reset to null. If the database has to be selected again, the parameters *logpath* and *newlogpath* are not changed.

  Database Manager initially places the database log files in the same subdirectory as the database. If you relocate a database log file to another disk, you may improve performance since the disk heads will no longer need to move between data and log recordings. The log file and database may also have increased capacity since they are not limited by one file system.

Several of the parameters are interrelated. If the Database Heap, the Maximum Storage for Lock Lists, or the Buffer Pool Size for a particular database is changed in the database configuration file, the Maximum Segments Allowed for Database Services in the Database Manager configuration file may have to be increased. The Maximum Segments Allowed for Database Services must be greater than or equal to:

```
(Maximum Number of Concurrently Active Databases) * (number of
shared segments per database) + 1
```

The number of shared segments per database is computed at:

```
dbheap + (locklist + 15) / 16 + (buffpage + 15) / 16 + 3
```

Using Database Authorizations

Database authorizations provide a way for you to control access to Database Manager databases. The following are the four types of database authorizations:

Open Database By granting this authorization, the user will have authority to access a specified database. You should determine which users need to access a selected database to perform Query Manager tasks when granting or revoking this authorization.

Create Table You can allow other users the ability to create tables from the selected database with this authorization. Users granted this authorization automatically receive *table Control* authority over any table they create. Users with table Control authority can select **Authorizations** from the Tools pull-down. in the Tables and Views primary menu to grant or revoke other users' *table* authorizations. For more information about table authorizations, see "Using Tables and Views Authorizations" on page 8-28.

Create Plan You can grant other users the ability to create an application program in IBM C/2, COBOL/2, FORTRAN/2, PASACAL/2, or languages that use Database Services application programming interface to access to Database Manager databases.

Database Administrator
The database administrator authorization allows a user an administrative authority level over a *selected* database. A user named as a database administrator has authority over every table, view, query, form, procedure, panel, and menu for a selected database. With database administrator authorization, you can also grant and revoke database and table authorizations. You cannot, however, assign another user SYSADM authority.

Query Manager will prompt you to run the migration utility on databases created under previous versions of Database Manager and accessed for the first time through use of database Authorizations. See

Appendix G for more information about migration of Database Manager databases.

As a user with SYSADM authority, you can grant or revoke authorization to Open a Database, Create a Table, or Create a Plan. Only a user with SYSADM authority can name another user as a database administrator. A group ID can be named as a database administrator and every user ID included in the group will be indirectly granted database administrator authority. PUBLIC cannot, however, be named as a database administrator. You cannot change your own authorization.

As a database administrator, you can grant or revoke all database authorizations, except database administrator authority, to a new or existing user ID for a *selected* database. In other words, you can be a database administrator over one database but not another. You can grant or revoke other users the Open a Database, Create a Table, or Create a Plan authorization. You cannot, however, assign another user database administrator authority or revoke another user's database administrator authority.

Direct and Indirect Authority

You can grant and revoke authorizations on a direct and indirect basis. When you grant or revoke authorizations to a specific user or group ID, it is a *direct* authorization. Whenever you grant or revoke a direct authorization to a group ID, you are also granting or revoking an *indirect* authorization to every user ID belonging to that group.

You can grant, change, and revoke direct authorizations. If you receive an indirect authorization as a result of being included in a group, you will lose the authorization if the group's authorization is revoked. If you hold a direct authorization, and you are also a member of a group, you will maintain the direct authorization even when the group's authorization is revoked.

A user ID revoked from having *any* authorizations will not be displayed in the Database Authorizations menu. When you grant an authorization to PUBLIC, every user known to Database Manager is indirectly granted the specified authorization.

Checking Your Database Authorizations

If you do not have authority to grant or revoke database authorizations, you can still select **Authorizations** from the the Tools pull-down in the database primary menu to display and review the database authorizations granted to you.

To Use Database Tools

1. In the Databases primary menu, select a database name.

   Note: If you are cataloging a database, you must select **—NEW—**.

2. Select **Tools** from the action bar.
3. Select the tool you want to use from the Tools pull-down..
4. If necessary, follow the prompts to type the appropriate information for the tool you selected.

   The Databases primary menu is displayed.

Hints:

- To change the database configuration file for the selected database, select **Reconfigure local database** from the Tools pull-down..
- To return the database configuration file to the default settings for the selected database, select **Restore local database defaults** from the Tools pull-down..
- For Backup and Restore, you are prompted to insert diskettes as required to perform the Backup or Restore operation.

To Change Database Authorizations for an Existing User

Note: You must be a user with SYSADM (system administrator) authority or a database administrator for the selected database to change the database authorizations of an existing user.

1. In the Databases primary menu, select a database name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down..
4. In the Database Authorizations menu, to change a personal authorization, position the cursor on a user ID and select Enter. If you want to change a group authorization, continue with step 6 on page 8-20.

   Note: An asterisk indicates a group user ID.

5. Make your selections in the Change Database Authorizations for User menu to grant or revoke database authorizations for a single user and select Enter. The Database Authorizations menu is displayed. Continue with step 8.

   Notes:

   a. You can grant or revoke authorizations by making selections in the Direct Authority column.

      The Indirect Authority column displays the names of group IDs granting you that privilege. Your user ID can be granted indirect authorization by belonging to one or several groups. Up to two Group IDs can be displayed at one time. An ellipsis following the second Group ID indicates that your user ID belongs to more than two groups.

   b. Select Show to list all the groups that grant the individual user indirect authorization.

   c. You must be a user with SYSADM (system administrator) authority to grant database administrator authority to another user.

6. In the Database Authorization menu, to change a group authorization, position the cursor on PUBLIC or a group ID and select Enter.

7. Make your selections in the Change Database Authorizations for Group menu to grant or revoke database authorizations for a group and select Enter. The Database Authorizations menu is displayed.

   Notes:

   a. Select Show to view a list of users belonging to the selected group ID in the Change Database Authorizations for Group panel

   b. If you are changing authorizations for PUBLIC, by selecting Show, you can view all the users known to Database Manager and User Profile Management.

   c. You must be a user with SYSADM (system administrator authority) or a database administrator for the selected database to change the database authorizations of an existing user.

8. Select Cancel to display the Databases primary menu again.

To Specify Database Authorizations for a New User

1. In the Databases primary menu, select a database name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.
4. Select –NEW– from the Database Authorizations menu and select Enter.
5. Type the user or group ID in the Select User or Group panel and select Enter.

   Note: Select List to view all of the users and groups granted authorizations through Database Manager and defined in User Profile Management.
6. Make your selections in the Change Database Authorizations for User menu to grant or revoke authorizations for a new user ID. Select Enter to display the Database Authorizations menu again.
7. Select Cancel to display the Databases primary menu again.

   Note: You cannot use Database Manager until your user ID is defined in User Profile Management. See Chapter 1 for a discussion on using User Profile Management.

To Show Database Authorizations

Note: You can use the following procedure without an administrative authority level to display your Database authorizations.

1. In the Databases primary menu, select a database name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.

   The Show Database Authorizations for User panel is displayed.
4. Select Cancel to display the Databases primary menu again.

Using Query Manager Object Tools

Object Tools are utilities you can use from the Query Manager object menu (queries, forms, procedures, panels, and menus) and from the Tables and Views primary menu (for tables and views). These utilities allow you to perform four kinds of tasks:

- Exchange Query Manager *objects*, including queries, forms, menus, panels, and procedures, between Database Manager databases using the Import and Export utilities.
- Exchange the *data* in tables and views between Database Manager databases, as well as other workstation database or spreadsheet programs, using the Import and Export utilities.
- Reorder a *table* and eliminate stored deletions using the Reorganize utility.
- Update the information about accessing the data in a table using the Run Statistics utility.

The Authorizations function, available from the Tables and Views primary menu, provides a way to protect access to table and view definitions and a way to control how other users can select a table or view to add, change, or edit data. You can also use the Authorizations function to view your own authorizations. See "Using Tables and Views Authorizations" on page 8-28 for a discussion of this function.

When you exchange Query Manager objects between Database Manager databases, the following occurs:

- Export causes an object, selected from the primary menu for an object, to be copied into an OS/2 file. (You cannot select a **—NEW—** object to export.)
- Import causes an object in an OS/2 file to be copied into the database and saved as an object.

You must be authorized to export or import a Query Manager object. As a user with SYSADM authority or as a database administrator, you are automatically authorized to import or export an object existing on the selected database. To import or export a Query Manager object without an administrative authority level, you must be the *owner* of the object, or the object (query, form, panel, menu, or procedure) must be saved with **Share** set to **Yes**.

You can also exchange data between Query Manager tables and views and another personal computer database or spreadsheet program. The data file formats supported for tables and views are:

Format	Description
IXF	Personal Computer version, the preferred method for exporting a table so that it can be imported later into the same table or into a new Database Manager table.
WSF	Work sheet formats, a subset of the formats supported in common by the following LOTUS products: • LOTUS 1-2-3**, versions 1.0, 1A, and 2.0 • Symphony**, versions 1.0 and 1.1.
DEL	Delimited ASCII, for exchange with one of the following: • dBASE II, III • BASIC programs • IBM Personal Decision Series* • DB2 and SQL/DS (import only). For information on importing DB2 and SQL/DS data, see Appendix E.
ASC	Nondelimited ASCII, for *importing* data from other applications; for example, from the IBM DisplayWrite* series and the IBM Personal Editor.

When you exchange data from tables and views with other database or spreadsheet programs, the following occurs:

Export Export places the data from an existing table or view into an OS/2 file formatted according to the data format you specify (PC/IXF, work sheet formats (WSF), or DEL). ASC is not a supported file type for export.

Import For DEL, WSF, and ASC data file formats, the table with its column names and specific data types must be defined before Import can occur. For the PC/IXF data file format, the table can be –NEW–; the table is automatically created when the data is imported. If the table or view receiving the imported data already contains data, Import will replace or append the data in the existing table or view with the data in the OS/2 file.

Degradation in performance can occur when a table has many modifications and the rows within it are no longer optimally stored with respect to the sort order of the indexes on the table. Using the reorganize utility, you can store a table more efficiently, possibly resulting in improved performance. You can specify an index to physically reorder the records in the table (for example, if you run queries that consistently access a particular column, your reorganization should reflect that).

Select **Run statistics** to update the statistics about the physical characteristics of a table or its indexes (for example, number of rows, average row length, and so on). These statistics are used by Database Manager in order to determine optimal access paths to the data in tables. You should run statistics after creating an index, after using the Reorganize tool to reorganize a table, and after extensive additions, deletions, and changes to a table.

To Import or Export Query Manager Objects

Note: Query Manager objects include queries, forms, procedures, panels, and menus.

1. In a primary object menu (for example, in the Queries primary menu), select an object name.
2. Select **Tools** from the action bar.
3. Select **Import** or **Export** from the Tools pull-down.

   Note: If the object is being imported, in the Import panel, you can set **Share** to **Yes** or **No**.

   If **Share** is set to **Yes**, then the object is available for use by any user with authorization to access the selected database. If **Share** is set to **No**, then only the owner of the object, or a user with an administrative authority level, can access the object.
4. Follow the prompts to type the appropriate information for the utility you selected.

Hint:

> If you do not specify an extension when importing or exporting Query Manager objects, a default extension is appended based on the object type (.FRM for forms, .PRC for procedures, .QRY for queries, .MNU for menus, and .PNL for panels).

To Import or Export Tables or Views

Note: For IXF data, the table you import is automatically created if necessary; it can be **—NEW—**. Before you can import DEL, WSF, or ASC data, you must define the table or view to which you plan to import the data. See Appendix D for more information on importing and exporting tables and views.

1. In the Tables and Views primary menu, select a table or view name, or select **—NEW—** if you are importing IXF data and the table does not exist.
2. Select **Tools** from the action bar.
3. Select **Import** or **Export** from the Tools pull-down.
4. Follow the prompts to type the appropriate information for the utility you selected.

   Notes:

   a. If the OS/2 file is not in the current directory, provide a complete path when you specify the OS/2 file name; for example, type:

      ```
      a:\delfiles\mydata.del
      ```

   b. For IXF data, you can specify the qualifier prefix of the table name you are importing to.
5. If the object is being imported into an existing table or view, the existing data of the table in the database is either *replaced* or *appended to* the existing data of the OS/2 file.

   Notes:

   a. You can import and *append* data to an existing parent table referenced by dependent tables, or to a view based on a table referenced by dependent tables.

   b. Since deleting data rows from a referenced table, or deleting rows from a view based on a referenced table could result in changing or deleting rows in other tables, Query Manager will not allow you to import and *replace* data to an existing parent table containing data rows. You can only import and replace data into an empty parent table.

6. If you are importing ASC data, the Import ASCII Data panel is displayed. This panel shows the name of the table or view you are importing into, the OS/2 file name that is the source file, and the table definition (column names, data types, and lengths). You must specify the starting and ending positions of each column you want to import. The starting position indicates the first character of the column for the data you are importing. The ending position indicates the last character or space to be used by the column. The starting position up to and including the ending position defines the width of the column. For example, in an 80 character row, a column with a starting position of 5 and an ending position of 15 would identify a column 11 characters wide beginning 4 spaces from the left of the ASCII file.

   When specifying starting and ending positions, you can also specify overlapping positions. For example, suppose you have a 12 character column containing phone numbers, and you want to show the data by area code and then by the entire phone number. You can specify the first column to begin with position 1 and end with position 3; then specify the second column to begin with position 1 and end with postition 12. Query Manager will automatically separate out the data for you.

   When you have completed this panel, select **Perform import** from the Actions pull-down, or press the Perform Import (Shift+F1) key.

7. If necessary, follow the prompts to insert the appropriate diskettes.

Hints:

- If you do not specify an extension for the OS/2 file name for export or import of data from a table or view, an extension is appended based on the data file format. For example, for the extension .IXF, .DEL, .ASC, and for WSF, either .WKS, .WRK, .WK1, or .WR1 is appended, depending on what is specified in the active profile.

- For PC/IXF files, the data from tables and views can span diskettes. You are not asked for more than one diskette unless you are importing or exporting a table or view that takes up more space than there is on one diskette. The first diskette used for exporting to PC/IXF files must contain at least 97 bytes of free space to ensure proper diskette processing.

- If the object you are importing was produced for a database running under a different code page or country code than the one you are using, you can get unexpected results using the object. For example, code page differences can cause constant strings to represent different characters; country code differences can cause the constants defining the dates to be interpreted differently.
- You can override the default character for the column delimiter, character string delimiter, and decimal point character for the DEL file format in your profile. See "Using Profiles" on page 8-35 for more information.
- Error messages received during import or export processing are stored in a file named QRWIMPRT.LOG or QRWEXPRT.LOG, respectively. These files are stored in the current directory. Review the contents of these files using a base operating system editor or using the OS/2 TYPE or PRINT commands. Record the message number. Return to Query Manager, select Help, select Index, and select **Import/Export Messages**. In the Import/Export Messages index, select the message number that matches the number you recorded. The help panel for the message is displayed.
- Before you begin importing ASC data, you can save the file specification in the Import ASCII Data panel into a table by pressing the Save (Shift+F3) key. This corresponds to the **Save** item from the Actions pull-down. Then, you can get this file specification to use for an import at another time by pressing the Get (F2) key. This corresponds to the **Get** item from the Actions pull-down.

  The *file specification* is a table that contains information about the layout of the OS/2 file. This information includes the column names, the starting position, and the ending position for the columns in the OS/2 file.
- For more information, refer to Appendix D.

To Reorganize a Table

1. In the Tables and View primary menu, select a table name.
2. Select **Tools** from the action bar.
3. Select **Reorganize** from the Tools pull-down.
4. Follow the prompt to complete the task.

Hints:

- While the table is being reorganized, the hourglass icon is displayed.
- If a table is very large, reorganizing may not increase performance significantly.
- **Reorganize** only reorganizes a single table, not a database.
- It is recommended that you do not press the Break (Ctrl+Break) key to stop the Reorganize utility. (Pressing the Break key during a reorganize of a table may require you to use a recovery procedure.) After reorganization is successfully completed, you will be prompted to run Statistics on the table.

To Run Statistics on a Table

1. In the Tables and Views primary menu, select a table name.
2. Select **Tools** from the action bar.
3. Select **Run statistics** from the Tools pull-down.
4. Follow the prompts to complete the task.

Using Tables and Views Authorizations

If you are a user with SYSADM (system administrator) authority, a database administrator for the selected database, or if you have table or view Control authority, you can select **Authorizations** from the Tools pull-down in the Tables and Views primary menu to grant or revoke other user's specific authorizations. You should make using the Authorizations function the last step to creating a table or view.

The following is a list of *table* and *view* authorizations:

Query rows Allows you to retrieve data rows from a table or view, create a view, and use the Export tool.

Add rows Allows you to add rows to a table or view and use the Import tool.

Change rows Allows you to update a table or view.

Delete rows Allows you to delete rows from a table or view.

The following authorizations apply only to tables:

Add index Allows you to create an index on a table. When you are authorized to create an index, you automatically receive index Control authority. With index Control authority, you can later choose to delete the index.

Alter table This authorization allows you to change a table definition and to add or change a comment on the table.

Reference table Allows you to create a referential constraint referencing the table as the parent in the relationship.

Table control This authorization grants you table Control authority. You can use all of the table authorizations. You can grant or revoke all table authorizations, with the exception of table Control authority, and use the Statistics tool on the table and indexes. When you create a table, you automatically receive this authorization.

This authorization applies only to views:

View control This authorization grants you view Control authority. You can use all of the view authorizations. You can grant or revoke all view authorizations, with the exception of view Control authority, and use the Statistics tool on the view. You must have table Control or query rows authorization on each table a view accesses to create a view.

Direct and Indirect Authority

You can grant and revoke table or view authorizations on a direct or indirect basis. When you grant or revoke authorizations to a specific user ID, it is a *direct* authorization. Whenever you grant or revoke a direct authorization to a group ID, you are also granting or revoking an *indirect* authorization to every user belonging to that group.

You can grant, change, and revoke direct authorizations. If you receive an indirect authorization as a result of being included in a group, you will lose the authorization if the group's authorization is revoked. If you hold a direct authorization, and you are also a member of a group, you will maintain the direct authorization even when the group's authorization is revoked.

When granting and revoking table and view authorizations, keep in mind that you can grant very specific authorizations. For example, you could grant a user the authority to Add data rows and query rows authority for a selected table but not grant Alter table authority. You would thereby allow users to enter data using the selected table but deny users the ability to alter the table definition.

Checking Your Table and View Authorizations

If you do not have authority to grant or revoke table or view authorizations, you can still select **Authorizations** from the the Tools pull-down in the Tables or Views primary menu to show the authorizations granted to you.

To Change Table Authorizations

1. In the Tables and Views primary menu, select a table name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.
4. To change a personal authorization, position the cursor on a user ID in the Table Authorizations menu and select Enter. If you want to change a group authorization, continue with step 6 on page 8-31.
5. Make your selections in the Change Table Authorizations for User menu to grant or revoke table authorizations for a single user and select Enter. Continue with step 8 on page 8-31.

Notes:

a. You can grant or revoke authorizations by making selections in the Direct Authority column.

   The Indirect Authority column displays the names of group IDs. Your user ID can be granted indirect authorization by belonging to one or several groups. Up to two group IDs can be displayed at a time. An ellipsis (...) following the second group ID indicates that your user ID belongs to more than two groups.

   If the Indirect Authority column displays that a group ID has DBA* authority, each user ID included in the group ID has database administrator authority and, in addition, is granted all of the table authorizations for the selected table.

b. Select Show to list all the group ID names that grant you indirect authorization.

c. You must be a user with an administrative authority level (a user with SYSADM authority or a database administrator) to be able to grant table or view Control authority to another user.

6. In the Table Authorizations menu, to change a group authorization, position the cursor on a PUBLIC or group ID and select Enter.

7. Make your selections in the Change Table Authorizations for Group menu to grant or revoke table authorizations for a group.

   Notes:

   a. Select Show to view a list of users belonging to the selected group ID in the Change Table Authorizations panel.

   b. If you are changing authorizations for PUBLIC, select Show to view a list of all users known to Database Manager.

8. Select Cancel to display the Tables Authorizations menu again.

Hint:

In the Table Authorizations menu, the columns displayed include Query rows, Add rows, Change rows, and Delete rows authorizations for user and group IDs. The Misc. Auth column represents whether or not the user or group ID was granted one or more of the following authorizations: Alter table, Reference a table, or Add an index. You can change or view each of these table authorizations in the the Change Table Authorizations for User menu, or the Change Table Authorizations for Group menu.

To Specify Table Authorizations for a New User

1. In the Tables and Views primary menu, select a table name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.
4. Select —**NEW**— from the Tables Authorization menu and select Enter.
5. Type the user or group ID in the Select User or Group panel and select Enter.

   Note: Select List to view all of the users known to Database Manager.
6. Make your changes in the Change Table Authorizations for User menu grant or revoke table authorizations for a new user. Select Enter to display the Tables Authorization menu again.
7. Select Cancel to display the Tables and Views primary menu again.

To Show Table Authorizations

Note: You can use the following procedure without table Control authority or an administrative authority level to display your Table authorizations.

1. In the Tables and Views primary menu, select a table name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.

   The Show User Tables Authorizations panel is displayed.
4. Select Cancel to display the Tables and Views primary menu again.

To Change View Authorizations

1. In the Tables and Views primary menu, select a view name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.
4. To change a personal authorization, position the cursor on a user ID in the View Authorizations menu and select Enter. If you want to change a group authorization, continue with step 6 on page 8-33.
5. Make your selections in the Change View Authorizations for User menu to grant or revoke view authorizations for a single user ID and select Enter. Continue with step 9 on page 8-34.

   Notes:

   a. You can grant or revoke view authorizations by making selections in the Direct Authority column.

      The Indirect Authority column displays the names of group IDs. Your user ID can be granted indirect authorization by belonging to one or several groups. Up to two group IDs can be displayed at a time. An ellipsis (...) following the second group ID indicates that your user ID belongs to more than two groups.

      If the Indirect Authority column displays that a group ID has `DBA*` authority, each user ID included in the group ID has database administrator authority and, in addition, is granted all of the view authorizations for the selected view.

   b. Select Show to list all group ID names that grant your user ID indirect authorization.

   c. You must be a user with an administrative authority level (a user with SYSADM authority or a database administrator) to be able to grant table or view Control authority to another user. In addition, you must also have table Control authority or Query rows authorization for any table a view accesses to create the view.

6. In the View Authorizations menu, to change a group authorization, position the cursor on a PUBLIC or group ID and select Enter.

7. Make your selections in the Change View Authorizations for Group menu to grant or revoke view authorizations for a group.

   Notes:

   a. Select Show to view a list of users belonging to the selected group ID in the Change View Authorizations for Group menu.

   b. If you are changing authorizations for PUBLIC, select Show to view all users known to Database Manager.

   c. You must be a a user with an administrative authority level (a user with SYSADM authority or a database administrator) to be able to grant table or view Control authority to another user.

8. Select Enter to display the View Authorizations menu again.

9. Select Cancel to display the Tables and Views primary menu again.

To Specify View Authorizations for a New User

1. In the Tables and Views primary menu, select a view name.

2. Select **Tools** from the action bar.

3. Select **Authorizations** from the Tools pull-down.

4. Select —NEW— from the View Authorizations menu and select Enter.

5. Type the user or group ID in the Select User or Group panel.

   Note: Select List to view all of the users known to Database Manager.

6. Make your changes in the Change View Authorization for User menu (for an individual user) or the Change View Authorization for Group menu (for a group) to grant or revoke table or view authorizations for a new user.

7. Select Enter to display the View Authorizations menu again.

8. Select Cancel to display the Tables and Views primary menu again.

To Show View Authorizations

Note: You can use the following procedure without view Control authority or an administrative authority level to display your View authorizations.

1. In the Tables and Views primary menu, select a view name.
2. Select **Tools** from the action bar.
3. Select **Authorizations** from the Tools pull-down.

   The Show User View Authorizations panel is displayed.
4. Select Cancel to display the Tables and Views primary menu again.

Using Profiles

With the Profiles task, you can create and maintain one or more *profiles* to use when running Query Manager. A profile contains information on sign-on options, default printing options, data format options, and import and export options. If the values in the default Query Manager profile do not meet your needs, you can create a customized profile based on the default values by opening a new profile.

Profiles contain four categories of parameters:

Sign-on options Allow you to specify values to use when Query Manager is started. These include a database name (so that you can automatically connect to the database of your choice and bypass the Databases primary menu), a default qualifier for lists (to be used for listing object names so you can access objects created using another qualifier ID), and the amount of memory to allocate for retrieving rows from the database.

Printing options Include the printer nickname and parameters for lines per inch, normal or compressed print (pitch), number of copies, and whether to include a page number or the date and time on your printed materials. The printer nickname points to the printer nickname file you specify through Query Manager System

Tools. For information on specifying printer nickname files, see "To Specify a Printer Nickname File" on page 8-6.

Data format options Use to format the data in reports and to create default edit codes for columns within a table definition. You can specify different values for the decimal character, thousands separator, the rounding rule, negative signs, currency symbols, default date and time edit codes, and null character. The null character is displayed when editing or viewing the data in tables to indicate a null value in a row. The null character can also be used to indicate a null comment.

See Appendix I for a description of each of these edit codes.

Import/export options Use to specify options for importing and exporting tables and views between Query Manager and another database or spreadsheet program. For more information on import and export, see "To Import or Export Tables or Views" on page 8-25 and Appendix D.

You can personalize your Query Manager start-up by specifying a profile to use when you select Query Manager from the Group – Main window. For information on personalizing your Query Manager start-up, see Appendix B. You can also activate a different profile while you are using Query Manager. When a profile is activated, all of the options are active with the exception of the Sign-on options. Since Sign-on options are established when a database is accessed, you must exit and restart Query Manager to have the changed Sign-on options take effect.

The Data format options display OS/2 defaults. The OS/2 default edit code formats represent the characters and formats specified by the selected OS/2 country code information. For example, for U.S. country code 001, the default format for date is TDM- and the default format for decimal character is a period (.). See the *User's Guide, Volume 1: Base Operating System* for a description of country code information.

To Use Profiles

1. In the Main Selection menu, select **Profiles.**
2. Select an existing profile name or **—NEW—** from the Profiles primary menu.
3. Select **Actions** from the Action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.

   The first screen of the Profile panel is displayed. Each item is set to the values for the default profile.
4. Type your choices and make your selections for **Sign-On Options** and **Printing Options** in the Profile panel and then press the Page Down (PgDn) key.

   Notes:

   a. For a list of the available database names and printer nicknames, select the List key, or select **Actions** from the action bar and then select **List** from the Actions pull-down.

   b. Select Help for information on each item in the Profile panels.
5. Type your choices and make your selections for **Data Format Options** in the Profile panel, and then press the PgDn key.
6. Type your choices and make your selections for **Import/Export Options** in the Profile panel.
7. Select **Exit** from the action bar and **Exit Profile** from the Exit pull-down, or press the Exit Profile (F3) key. The confirmation message is displayed.
8. Select **Save and exit** from the confirmation message to save the entries in Profile panel and select Enter.

   Notes:

   a. If you select **Exit without saving**, the changes to the parameters in the Profile panel are not saved and the Profile primary menu is displayed again.

   b. If you select **Resume**, the Profiles panel is displayed again and you can continue to make changes to the displayed parameters.
9. Type a name for the profile and select Enter to save the profile. Query Manager appends the extension .QPF to the profile name.

10. In the Profiles primary menu, select the name of the profile you just created.

11. Select **Activate** from the Actions pull-down, or press the Activate (Shift+F1) key to activate the new profile.

    Note: In order to use the **Sign-On Options** in the new profile, you must exit and restart Query Manager, using the new profile name, before the changes take effect. For information on personalizing your Query Manager start-up, see Appendix B.

Hint:

To erase a profile, select the profile name from the Profiles primary menu and select **Erase** from the Actions pull-down, or press the Erase (Shift+F8) key.

Chapter 9. Using SQL Query

This chapter provides information about using IBM Structured Query Language (SQL) statements within Query Manager to manipulate and manage information stored in a database.

About SQL

IBM Structured Query Language (SQL) is an established set of statements that can be used to manage information that is stored in a Database Manager database. SQL statements can be used to query a database; update, insert, and delete data in tables; create tables and views; and grant and revoke authorizations.

Using SQL Statements in a Query

To practice writing an SQL statement, you can take the D38 query from "Writing Queries" on page 3-3 and show it as an SQL statement. The objective of the query was to retrieve the data from the STAFF table for employee name, department, job, salary, commission, and salary plus commission for each employee in department 38 and put the data in alphabetical order by employee name. The corresponding SQL statement would be specified as follows:

```
SELECT ALL NAME, DEPT, JOB, LOCATION, SALARY+COMM
FROM USERID.STAFF A USERID.ORG B
WHERE (DEPT=DEPTNUMB) AND
((DEPT=38))
ORDER BY NAME
```

If you create an object without qualifying the object name, Database Services will add a qualifier to the object when the object is run. The qualifier will equal the user ID of the person who runs the object.

You can type or edit SQL statements within Query Manager through SQL query. If you understand SQL, you can perform some Query Manager functions with fewer steps by using SQL query. You can also create queries that are more complex than those allowed through prompted query. The supported SQL statements are:

- COMMENT
- CREATE/DROP/ALTER TABLE
- CREATE/DROP VIEW
- CREATE/DROP INDEX
- DELETE
- DROP PROGRAM
- GRANT
- INSERT
- SELECT
- REVOKE
- UPDATE.

For information on writing and using SQL statements, see *SQL Concepts*.

As you type SQL queries into Query Manager, keep the following rules in mind:

- Only one statement can be typed and run from the SQL Query panel at a time.
- Comments can be used within an SQL statement. A comment begins with two consecutive dashes (− −) and occupies all of the line from the dashes to the end of the line. A line can begin with a comment. An example of a comment follows:

  ```
  SELECT NAME FROM STAFF     --Retrieve the employee name
  ```

- A character string surrounded by single quotes can span multiple lines. When you type a quoted character, be careful of where you put the spaces. If you put a space before the carriage return, the space will become part of the character string.
- Lines can be inserted or deleted or the entire SQL Query panel can be cleared.

- In Query Manager, variables can be included within SQL statements. A variable should begin with an ampersand (&). An example follows:

  ```
  SELECT NAME, JOB FROM &TABLE
  ```

  Query Manager prompts you for the value of the variable when you run the query if the value is not previously assigned. Query Manager substitutes that value into the statement before processing the statement.

To Use SQL Query

1. In the Main Selection menu or in primary menus for objects, select **Commands** from the action bar and then select **SQL Query** from the Commands pull-down, or press the SQL Query (Shift+F4) key.

   Note: You can also access the SQL Query panel by selecting **Convert to SQL** from the Actions pull-down while in prompted query, prompted view, or while defining a table.

2. Type the SQL statement in the SQL Query panel.

   Note: Press the Enter key to go to the new line and continue typing. The Enter key works as a line-advance key.

   For more information on the available editing keys in SQL query, select Help and then select Keys.

3. Select **Run** from the Actions pull-down to run the SQL statement displayed in the SQL Query panel, or press the Run (Shift+F1) key.

   Notes:

   a. If the SQL statement is a SELECT statement, the data generated is displayed in the Report panel using the default form.

   b. To run an SQL SELECT statement and specify a form to use, select **Run using** from the Actions pull-down.

4. If you want to save the SQL statement displayed in the SQL Query panel, select **Exit** from the actions bar and then select **Exit SQL Query** from the Exit pull-down, or press the Exit SQL Query (F3) key. The confirmation message is displayed.

5. Select **Save and exit** from the confirmation message. The Save panel is displayed.

Notes:

a. If you select **Exit without saving**, the primary object or Main Selection menu from where you started is displayed again.

b. If you select **Resume**, the SQL Query panel is displayed again so you can resume the task of defining or changing the SQL query.

6. Type the SQL query name and an optional comment in the Save panel, set **Share** to **Yes** or **No**, and then select Enter.

   Notes:

   a. For information on naming conventions for the query name, see Appendix A.

   b. For more information on using the **Share** option, see "Authorization" on page 3-2.

Hints:

- The name of a saved query is listed in the Queries primary menu along with the prompted queries. If you select this name from the Queries primary menu, you would go to the SQL Query panel, *not* the Prompted Query panel.
- To clear the SQL Query panel, select **Refresh** from the Actions pull-down, or press the Refresh (F5) key. The SQL Query panel is cleared so you can type the statement or another statement again, and the name is reset to **—NEW—**.
- To retrieve a copy of an existing SQL query or an existing prompted query, select **Get** from the Actions pull-down, or press the Get (F2) key. If you get a prompted query, **Get** replaces the contents of the SQL Query panel. The name is replaced with **—NEW—**.
- To print the SQL query from the SQL Query panel, select **Print** from the Actions pull-down, or press the Print (F9) key.
- If the SQL query is a SQL SELECT statement, you can use the query-report-form triangle from the SQL Query panel just as you can from the Prompted Query panel. Select **Display** from the action bar and the corresponding item in the Display pull-down, or press the Form (Shift+F5) key, the Report (Shift+F6) key, or the Query (Shift+F7) key, depending on where you are. For more information on the query-report-form triangle, see Chapter 3.

- If you receive an error message while running an SQL query, select Help. The help panel for the message is displayed.

Chapter 10. Using Query Manager Commands

This chapter contains information about using Query Manager commands, such as RUN and PRINT, within Query Manager. These commands can be processed from a command line (the Command Line panel) or from within menus, panels, procedures, and the Query Manager Callable Interface. Query Manager commands correspond to functions available through the Query Manager prompted interface; however, depending on your needs, using the command itself can provide a faster path to accomplish your task.

About Query Manager Commands

Query Manager commands allow you to manipulate and manage Query Manager objects (including queries, forms, panels, procedures, and menus). You can use start commands that provide a *callable* interface into *prompted* Query Manager sessions. This chapter contains the steps for specifying Query Manager commands in the Command Line panel.

Since the command language uses an *action-verb object-type object-name* syntax, a command would be displayed as:

```
RUN QUERY DEPT
```

In the previous example, RUN is the action-verb, QUERY is the object-type (the kind of object to be manipulated), and DEPT is the object-name (the name you assign to that object).

The following two tables illustrate the Query Manager commands that can be processed from the command line (the Command Line panel), the Query Manager Callable Interface or from within menus, panels, and procedures. For an alphabetical list of the commands, along with syntax, examples, parameter descriptions, and limitations, see the *OS/2 Command Reference*. For more information on Query Manager Callable Interface, see the *Database Manager Programming Guide and Reference*.

Query Manager Commands	**Procedures**	**Query Manager Callable Interface**
BEGIN WINDOW	X	X
BEGIN WORK	X	X
CANCEL WORK	X	X
CONVERT QUERY	X	X
DEFINE TABLE	X	X
DISPLAY	X	X
EDIT TABLE	X	X
END WINDOW	X	X
END WORK	X	X
ERASE	X	X
EXIT	X	X
EXPORT	X	X
GET variable	X	X
IMPORT	X	X
LIST FORM LIST MAIN LIST MENU LIST PANEL LIST PROC LIST QUERY LIST TABLE		X X X X X X X
MESSAGE	X	X
PRINT	X	X
RESET	X	X
RUN	X	X
SAVE DATA AS	X	X
SET variable	X	X
START		X

Query Manager Commands	Command Line Panel	Menus	Panels
BEGIN WORK	X	X	X
CANCEL WORK	X	X	X
CONVERT QUERY	X	X	X
DEFINE TABLE	X		
END WORK	X	X	X
ERASE	X	X	X
EXPORT	X	X	X
GET variable	X	X	X
IMPORT	X	X	X
LIST FORM LIST MAIN LIST MENU LIST PANEL LIST PROC LIST TABLE LIST QUERY	X X X X X X X		
MESSAGE	X	X	X
PRINT	X	X	X
RUN	X	X	X
SAVE DATA AS	X	X	X
SET variable	X	X	X

Issuing Commands from the Command Line Panel or within Procedures

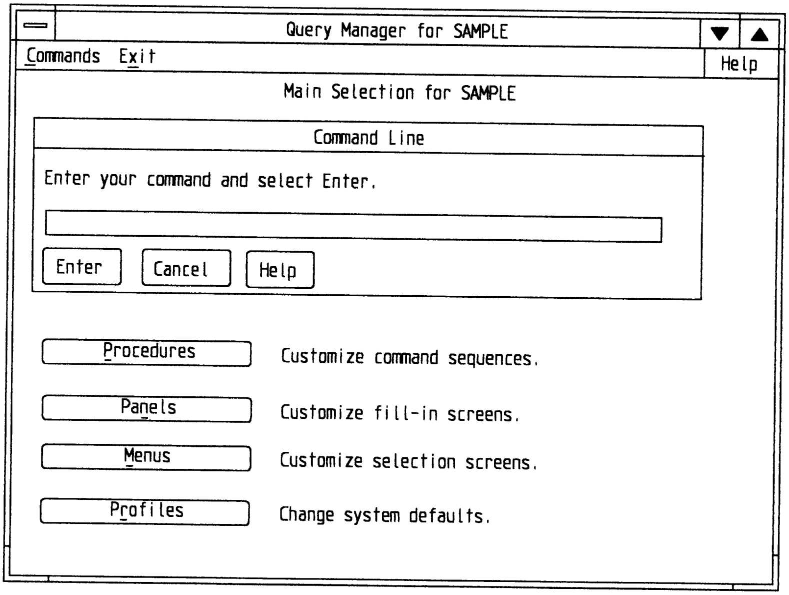
To use Query Manager commands:

- Type the command in the Command Line panel accessed by selecting **Commands** from the action bar in the Main Selection menu or the primary menus for objects.
- Place the command within a procedure.
- Place the command within a menu action and then select that action when the menu is used.

- Place the command within a panel action and then select that action when the panel is used.

For information on using commands within panels, procedures, and menus, see Chapter 11.

The Command Line panel is displayed as follows:



Enter your command and select Enter.

Customize command sequences.

Customize fill-in screens.

Customize selection screens.

Change system defaults.

To Use the Command Line Panel

1. In the Main Selection menu or in any primary names menus, select **Commands** from the action bar and then select **Command line** from the Commands pull-down, or press the Command Line (Shift+F9) key.
2. Type the Query Manager command in the Command Line panel.

   Notes:

   a. Only one command can be typed at a time.

   b. The command cannot be saved or retrieved.
3. Select Enter to run the command.
4. If necessary, respond to any prompts, menus, or panels that are displayed.

When you finish, the Command Line panel is displayed.

5. Select Cancel to exit the Command Line panel, or repeat steps 2 and 3.

Hint:

Commands can be saved within procedures. For information on writing procedures, see Chapter 14.

Chapter 11. Customizing an Application

This chapter contains information about customization tasks (designing panels, menus, and writing procedures) and discusses how you can design these objects to work together to create an application. These tasks allow you to design an interface to access your database. This interface is designed to reflect the requirements of a specific application.

Use this chapter in conjunction with the following chapters:

- Chapter 12
- Chapter 13
- Chapter 14
- Chapter 15
- Chapter 16.

Before You Begin

Designing menus and panels and writing procedures are described in the chapters that follow. These customization tasks should be performed by a more experienced database user who is familiar with application interface design.

As the designer of customized applications, you should be a user with SYSADM (system administrator) authority or a database administrator for the selected database. With either of these administrative authority levels, you have the necessary authorization to access and control the selected database, tables, views and other Query Manager objects to effectively create a customized application.

The application designer would design the customized menus, panels, and procedures that could be used by anyone from a data entry person to a company president. The actual user of the customized interface (the data entry person or company president) is only interested in interacting with the database in the area related to that person's job. When designing a customized application, you, as the designer, would take into account how much authorization each user requires to access the database and perform their job. In some instances, you will want to restrict access to selected databases or tables. By designing an interface for this user, you, as the designer, are providing a more efficient work environment.

This chapter contains a scenario that describes the customization tasks used to create a customized interface to the sample database.

About Customization Tasks

Customization tasks allow you to design a customized interface to access your database. This interface can be tailored to address the needs of your business. For example, if your enterprise is a catalog warehouse, you can design an interface with a customized menu that contains items for Accounts Receivable and Billing Notices. The Accounts Receivable item presents a panel used for order information, such as account numbers and unit quantities, and for computing unit costs based on quantities. The Billing Notices item would print a bill for each customer, indicating the amount owed for each account.

With customization tasks, you can design the following:

- *Menus*, lists of items from which the user can make a selection.
- *Panels*, customized screens that can be used for browse-only information or can contain one or more entry fields where you can type information, or a combination of the two. Panels can also be printed.
- *Procedures*, sequences of commands or statements to process information stored in the database.

To help you understand how customization of panels, procedures, and menus can aid you in your work, consider the following scenario. The next four chapters explain how you can create menus, panels, and procedures similar to the ones described in this scenario.

You are the administrative assistant to Mr. Daniels, the president of National Paper. National Paper distributes and sells paper products throughout the United States. Mr. Daniels wants to keep track of sales figures from each department and the job categories, *sales representative* and *clerk*. The sales representatives deal directly with wholesalers and the clerks sell paper products from the Paper Jam stores in each location. Mr. Daniels believes by analyzing departmental sales figures for each job category, he can determine which departments and which sales representatives and clerks are producing the most revenue, and thereby determine whether promotions or reorganizations are required.

To satisfy Mr. Daniels, you need to provide a monthly report that shows the salary plus commission for clerks and sales representatives. In addition, you need to be able to update the information you have about each department and each employee in the event of organizational changes.

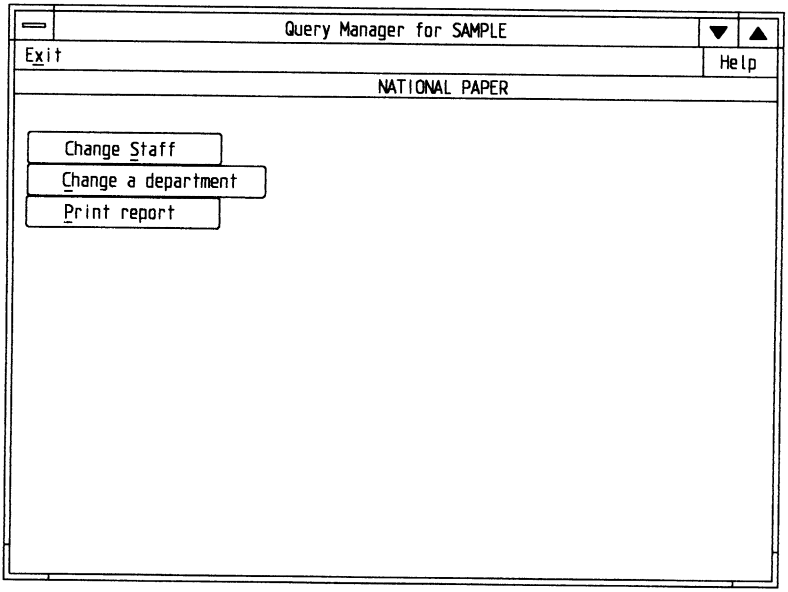
Fortunately you have already created a database that contains all pertinent data for National Paper. The database, named SAMPLE, contains two tables, STAFF and ORG. (You may be familiar with these two tables if you went through the exercises in *Database Manager Exercises.*) These tables are shown inChapter 5 . The ORG table contains the following information about each department within your organization: the department number, the department name, the manager of the department, the division the department is in, and the location (city) of the department. The STAFF table contains the following information about each employee at National Paper: ID, name, department number, job title, years in the company, salary, and commission.

Because Mr. Daniels requires the same reports monthly and because the changes you need to make to the database are predictable, you should design your own Query Manager panels, procedures, and menus so that you (or your data entry person) can accomplish these tasks more efficiently and quickly.

Your requirements for customized panels, procedures, and menus are as follows.

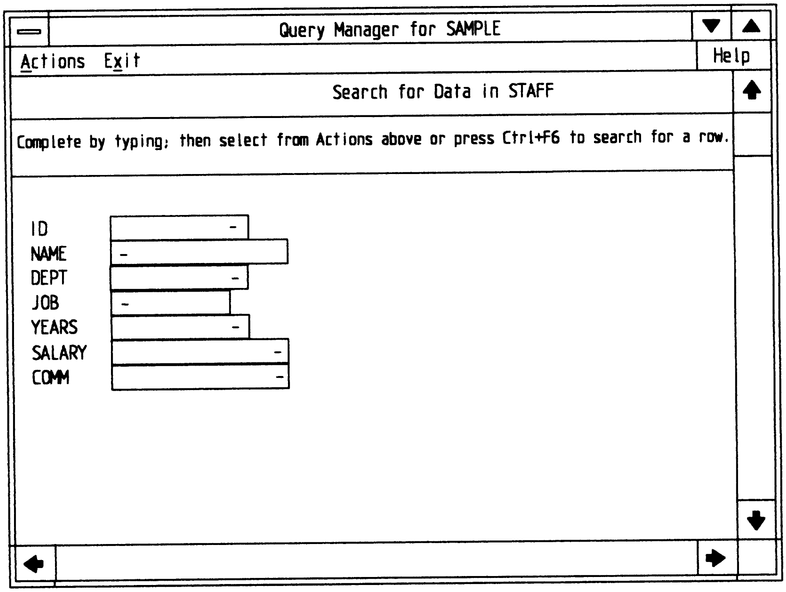
- Provide a simple method for updating the STAFF table when someone is hired, promoted, or their salary or commission changes. You want to be able to update STAFF based on the employee name.
- Provide a way to review information for a department, and change salary and commission at the same time.
- Print a monthly report to indicate salary plus commission for the job categories of sales representative and clerk.

To begin, determine which menu choices are appropriate based on the list of requirements. The following list represents the choices that can be displayed on your customized main menu:



Your first menu item, **Change Staff**, allows you to update the STAFF table. You need to be able to make the addition, change, or deletion based on an employee name. When **Change Staff** is selected, another panel is displayed instructing you to provide a value for the employee name.

If you do not type an employee name, a panel is displayed in which you can add a new employee to the STAFF table. If you type an employee name instead, the table is searched for that name and a panel is displayed that contains information about that employee. You can then make changes to the information in the panel. Since the column names in the STAFF table are exactly the same as the column names you want in the panel, you can use the STAFF table definition to create your panel. This is called the *default definition* for a panel. The panel would be displayed as follows.

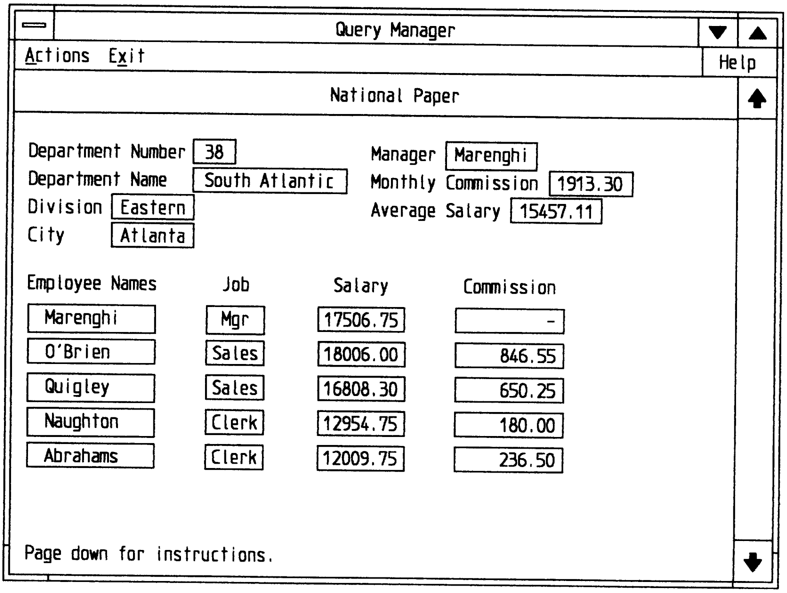


Complete by typing; then select from Actions above or press Ctrl+F6 to search for a row.

ID
NAME
DEPT
JOB
YEARS
SALARY
COMM

The second menu item, **Change a department**, allows you to retrieve data from both the STAFF and ORG tables and display this data at the same time. When **Change a department** is selected, another panel is displayed instructing you to provide a department number. When you type a department number, both the STAFF and ORG tables are searched for that department number and a panel is displayed that contains information about that department. You can then browse or make changes to the information in the panel. If you want to make changes to another department, you can select **Search** and type in a different department number in the **Department Number** entry field.

After the panel is designed, it should be displayed as follows.



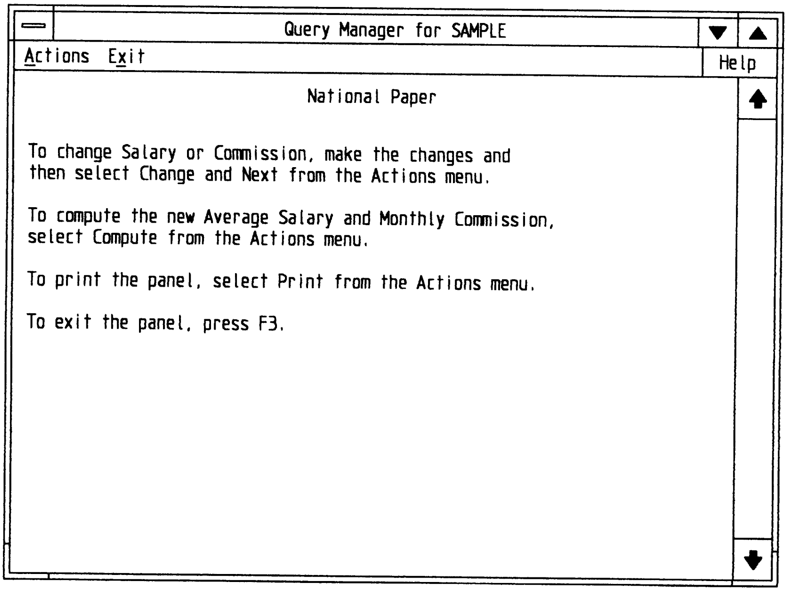
National Paper

Department Number 38
Department Name South Atlantic
Division Eastern
City Atlanta

Manager Marenghi
Monthly Commission 1913.30
Average Salary 15457.11

Employee Names	Job	Salary	Commission
Marenghi	Mgr	17506.75	-
O'Brien	Sales	18006.00	846.55
Quigley	Sales	16808.30	650.25
Naughton	Clerk	12954.75	180.00
Abrahams	Clerk	12009.75	236.50

Page down for instructions.



National Paper

To change Salary or Commission, make the changes and
then select Change and Next from the Actions menu.

To compute the new Average Salary and Monthly Commission,
select Compute from the Actions menu.

To print the panel, select Print from the Actions menu.

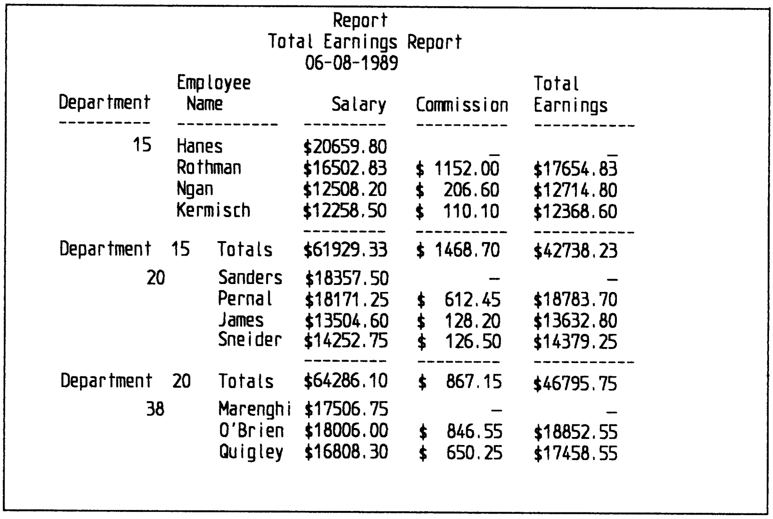
To exit the panel, press F3.

Notes:

1. Note that the first panel illustrated displays text to the user to page down for instructions. The second illustration is a continuation of the first panel.
2. When you type data into the **Salary** and **Commission** entry fields for one or more employees, the values displayed in the **Monthly Commission** and **Average Salary** fields are affected. To see the new values, you can use Query Manager to compute monthly commission and average salary for the department again based on the new data. To compute these values again, you select **Actions** from the action bar and **Compute** from the Actions pull-down to reflect this new data.

The final item in the NATIONAL PAPER main menu, **Print report,** runs a query and prints a report showing the salary and commission for each sales representative and clerk.

The QCLERK query is the same one you defined in the exercises in *Database Manager Exercises.* The query produces a report showing the employee name, department number, and salary plus commission for all the clerks and sales people in the company. The report has a title and appropriate headings. The report form, also created in "Exercise 2 – Revising the Report Form" in *Database Manager Exercises,* is named SALFORM. The following is the report as it would be printed.



```
                             Report
                      Total Earnings Report
                           06-08-1989
             Employee                                  Total
Department    Name             Salary   Commission    Earnings
----------   -----------   ---------    ----------    -----------
        15   Hanes         $20659.80             -              -
             Rothman       $16502.83    $ 1152.00     $17654.83
             Ngan          $12508.20    $  206.60     $12714.80
             Kermisch      $12258.50    $  110.10     $12368.60
                           ---------    ----------    -----------
Department  15   Totals    $61929.33    $ 1468.70     $42738.23
          20     Sanders   $18357.50             -              -
                 Pernal    $18171.25    $  612.45     $18783.70
                 James     $13504.60    $  128.20     $13632.80
                 Sneider   $14252.75    $  126.50     $14379.25
                           ---------    ---------     -----------
Department  20   Totals    $64286.10    $  867.15     $46795.75
          38     Marenghi  $17506.75             -              -
                 O'Brien   $18006.00    $  846.55     $18852.55
                 Quigley   $16808.30    $  650.25     $17458.55
```

To summarize, your NATIONAL PAPER menu does the following:

Change Staff Allows you to add new employees or change data about current employees contained in the STAFF table. (Runs a procedure named STAFFCHG that runs the ASTAFF panel and prepares the screen for data entry or data edit.)

Change a department Allows you to browse or change the data for an entire department. This data is retrieved from both the STAFF and ORG tables. (Runs a panel named DEPT with the mode equal to Change. You can then make further changes as required.)

Print report Allows you to print a report that shows the sales and commission for the clerks and sales representatives employed by National Paper. (Runs a procedure named PRINTREPT that runs the QCLERK query and prints the report using the SALFORM form.)

When you have finished designing the menus, panels, and procedures to meet your requirements, the user can run one procedure that runs the entire customized interface from within Query Manager, from the Group – Main window, or from the OS/2 command prompt. For more information on running procedures, refer to Appendix B. When the user runs this procedure, the NATIONAL PAPER menu is displayed and users can make selections to perform their work.

In the chapters that follow, the specific steps are given for designing customized menus, panels, and procedures. You can design simple or complex panels, procedures, and menus, depending on your needs. Each chapter explains the options available to you as you do your work. There is also a section within the chapters on menus, panels, and procedures that shows how the Query Manager menus and panels are filled in to create the scenario. Chapter 16 indicates the flow of the customized interface for the National Paper company described in this chapter. To practice using menus, panels, and procedures before you create your own customized interface, you can create the interface for the National Paper company.

Chapter 12. Designing Menus

This chapter contains information and steps for designing and running menus for use in a customized interface. Use this chapter in conjunction with the following chapters:

- Chapter 13
- Chapter 14
- Chapter 15
- Chapter 16.

This chapter assumes the primary reader is the menu designer. When it addresses *you* in the text, it is talking to menu designers. When it talks about the *user*, it is referring to the person who performs work at a computer using the menus that you, the menu designer, created. As you read this chapter, refer to "To Define a Menu" on page 12-6 for examples of the fill-in panels used to create the NATIONAL PAPER menu.

Before You Begin

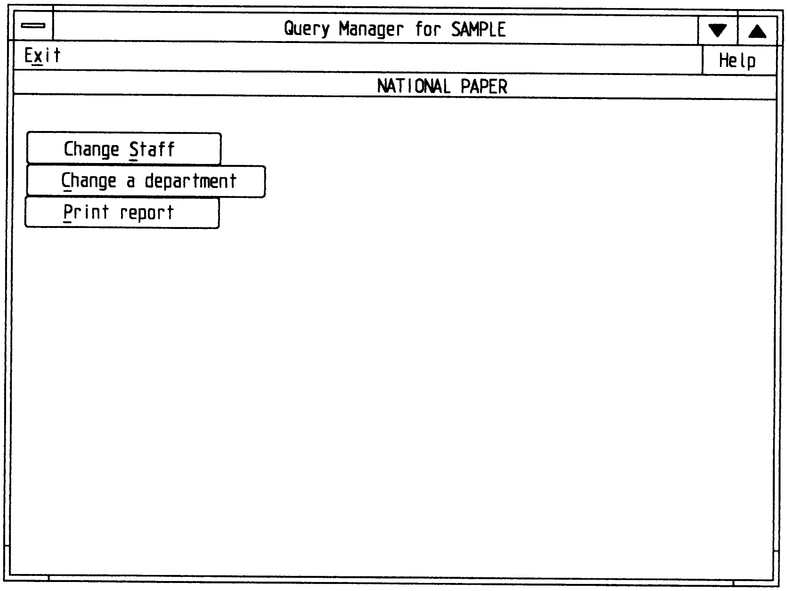
When you design a menu for a customized interface, you, the designer, should give some thought as to how the menu will best serve the needs of the user. You will need a thorough knowledge of the design and purpose of the database and applications you are working with. You should also have a working knowledge of how to define procedures and use Query Manager variables in customization tasks.

As the designer, you should be a user with SYSADM (system administrator) authority or a database administrator for the selected database. With either of these administrative authority levels, you are authorized to access and control panels, procedures, queries, and other objects that could be used in designing other parts of the customized interface.

You may also want to consider testing the menu in conjunction with the panels and procedures you may use in the other parts of the customized interface. It is recommended that you perform a backup for the selected database prior to designing a customized interface.

About Menus

To meet your specific database application needs, you can create customized menus to perform queries or data edit functions. These menus resemble other Query Manager menus in terms of presentation and user interaction and they can be tailored to your specific needs. Remember, a menu is a list of items from which you can make a selection. In the scenario described in Chapter 11, a customized menu is used to tie all the customized procedures, panels, and queries together. That customized menu, called NATIONAL PAPER, would be displayed as follows.



The items in a menu can run queries and generate reports, and call procedures, panels, and other menus, all by using Query Manager commands. For information on Query Manager commands, see Chapter 10 and the *OS/2 Command Reference*. As each menu item is defined, you assign a Query Manager command to that item. The menu from Chapter 11, and the action associated with each item on the menu, would do the following:

Change Staff Runs a procedure named STAFFCHG that runs the ASTAFF panel.

Change a department Runs a panel named DEPT with the mode equal to Change.

Print report Runs a procedure named PRINTREPT that runs the QCLERK query and prints the report using the SALFORM form.

There are two stages to designing a menu: *specify* and *layout*. First, you specify the text for the menu items, the Query Manager command run by the menu item, any variables required, and the title of the menu. Then, you lay out the menu items and any informational text on the Menu panel by marking where each menu item should be displayed in the final menu.

Note that the first time you access menus created under a previous version of Query Manager, the menu is automatically migrated to the Query Manager Version 1.3 format. See the Appendix G for information on how the menu may change as a result of migration.

Specifying a Menu

To specify a menu, you need to determine the following information:

Menu actions Type any phrase you want for the action text for the menu item, specify its associated mnemonic, and the Query Manager command that is processed when the menu item is selected from the menu. The action text is displayed in the final menu with the mnemonic highlighted. You can define up to 26 menu items.

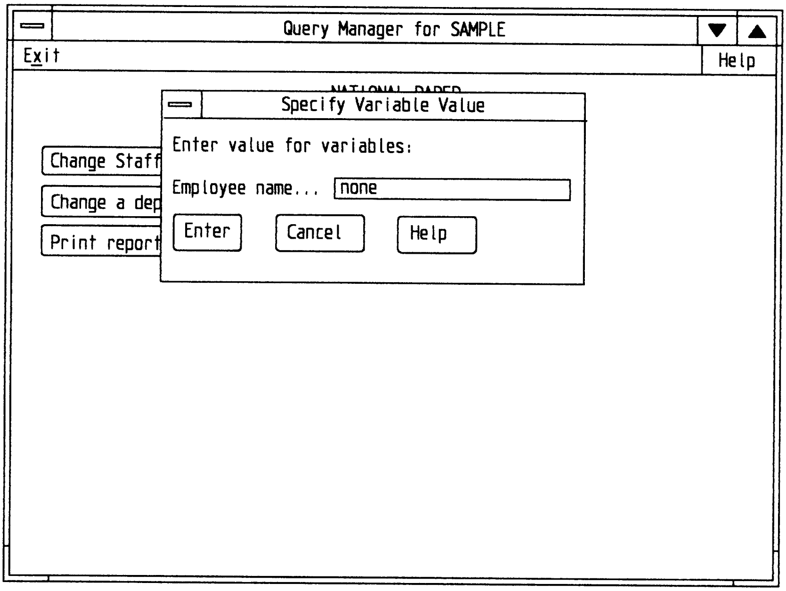
Variables The variable name, prompt text, and prompt value (default) for any variables that are used with the Query Manager command for a menu item. Variables are used to set a value that can be used by a query, form, menu, panel, or procedure. If you want a default value to display for the variable, you can specify that value as the prompt value. A prompt value can also be a variable. You can define up to 64 variables, but the maximum number of variables that can be associated with a single menu item is 10.

Title line The title of the menu, which is displayed centered on the first line of the menu when you run the menu.

Defining Variables

As part of menu design, you can use variables in your menu action commands. These variables are available to a query, form, menu, panel, or procedure. When the user selects an item, a panel is displayed that prompts for the variable, if applicable. In our scenario, a variable is defined, using the Specify Variables panel, for the **Change Staff** menu item: n for NAME. This variable is initialized to a default value of none.

If users select **Change Staff** from the NATIONAL PAPER menu, they are prompted for a name, as follows.



Enter value for variables:

Employee name... none

If **Change a department** is selected, a similar panel is displayed prompting for the department number. In our scenario, a variable is defined for the **Change a department** menu item: dept for DEPTNUMB. This variable is not initialized. If the user selects **Change a department** from the NATIONAL PAPER menu, the user must type a department number to identify the department in which they want to change data.

Laying Out a Menu

To lay out a menu, you should determine the order you want the menu items to be in when the menu is displayed and if you want to enhance the menu with instructions or informational text.

Marking Text and Menu Actions

Laying out a menu involves marking text and marking menu actions on the Menu definition panel. Each of these marked items are easily moved and sized using the mouse pointer or cursor movement keys.

When laying out a menu, **Adjust marks** can serve as an aid in aligning the marked text and menu actions. Adjust marks divides the panel area into an invisible grid and then allows you to move the selected text and menu actions in evenly blocked spaces. A check displayed beside Adjust marks in the Layout pull-down indicates that the option is turned on. Text and fields marked on the menu after this option is turned off are not adjusted.

When you are designing the menu, you can select **Check final menu** to view how the menu will be displayed when it is run.

Authorization

When you create a menu, you are considered the *owner* of the menu. When you are finished specifying a menu, you can select to exit and save the menu. A Save panel is then displayed that prompts you to type a name and comment and make a **Share** selection. As owner of the menu, you can choose to grant or revoke other user's authority to access and use the menu by setting **Share** to **Yes** or **No** in the Save panel.

If the menu is saved with **Share** set to **Yes**, any user who accesses the database can *select* the menu. You can also export or import the menu and assign it a *new* name.

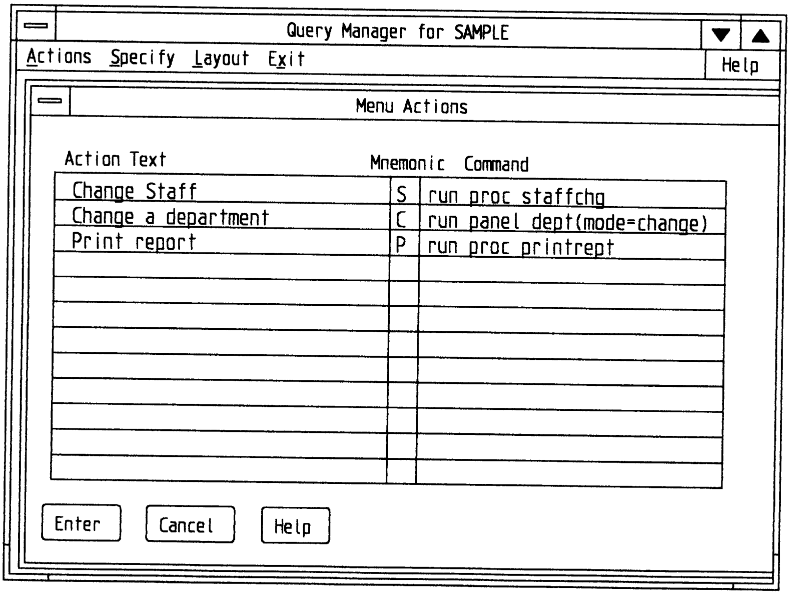
When the menu is saved with **Share** set to **No**, then only the owner, a user with SYSADM (system administrator) authority, or a database administrator for the selected database can select the menu, or import or export the menu.

You cannot change and save the menu under the same name, or select to erase the menu, unless you are the owner of the menu or a user with an administrative authority level.

To Define a Menu

The following steps show the panels for specifying and laying out the NATIONAL PAPER menu used in the scenario in Chapter 11.

1. In the Main Selection menu, select **Menus.**
2. Select **—NEW—** from the Menus primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.
4. In the Menu definition panel, select **Specify** from the action bar and then select **Menu actions** from the Specify pull-down.



Action Text	Mnemonic	Command
Change Staff	S	run proc staffchg
Change a department	C	run panel dept(mode=change)
Print report	P	run proc printrept

5. In the Menu Actions panel, for each menu item you select, type the **Action Text**, **Mnemonic**, and corresponding **Command** information and select Enter.

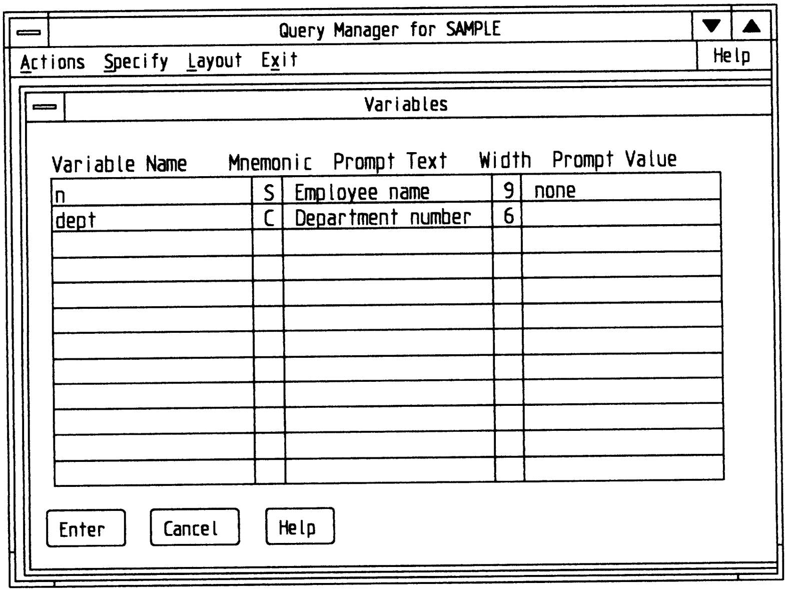
Notes:

a. The mnemonic is a character that must be present in the menu action text and that can be typed to select the menu item. It can be any alphanumeric character except a space and must be unique for each menu action.

b. If more than one command is required to define a menu action, you should create a procedure that contains those commands. Then, use that procedure as the menu action by assigning the RUN PROC command to the menu action. For information on Query Manager commands, see Chapter 10 and the *OS/2 Command Reference*. For information on procedures, see Chapter 14.

c. If the menu you are specifying does not require variables, you can skip steps 6 and 7.

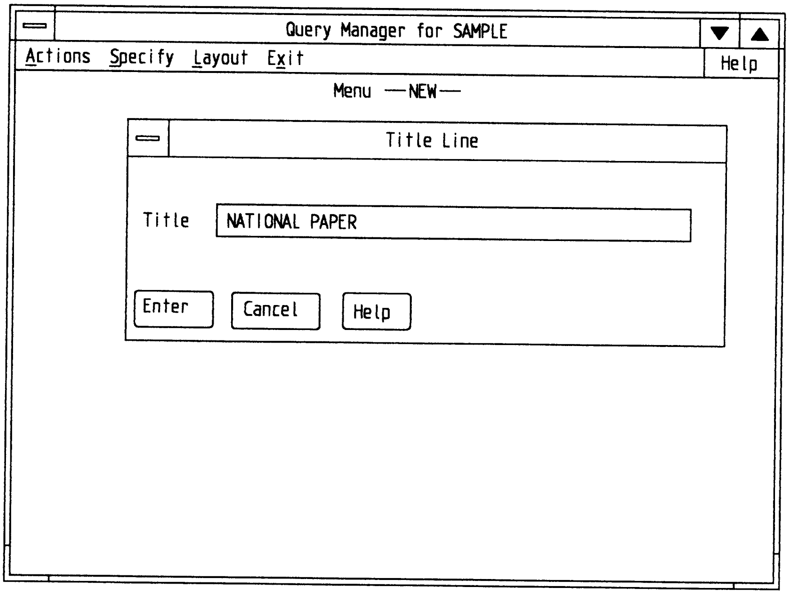
6. In the Menu definition panel, select **Specify** from the action bar and then select **Variables** from the Specify pull-down.



Variable Name	Mnemonic	Prompt Text	Width	Prompt Value
n	S	Employee name	9	none
dept	C	Department number	6	

7. Type the appropriate values for **Variable Name, Mnemonic, Prompt Text, Width,** and **Prompt Value** in the Variables panel and select Enter.

Note: The mnemonic provides the link to the mnemonic in the Menu Actions panel; this is how you tie the variable to the menu item.



Title NATIONAL PAPER

8. If you want the menu to have a title, select **Specify** from the action bar and then select **Title line** from the Specify pull-down.

   Note: The title is centered and placed on line 1 when the menu is running.

9. Type the title for the menu in the Title Line panel and select Enter.

   Note: Variables are identified by an ampersand (&) and can be included in the title line for a menu. For example, you could include the variables &DATE or &TIME. If the resolved variable plus the title text exceeds 50 characters, truncation occurs.

10. To mark text on the Menu definition panel, select **Layout** from the action bar and then select **Mark text** from the Layout pull-down, or press the Mark Text (Ctrl+F7) key. The tracking rectangle is displayed in the Menu definition panel.

11. Use the mouse pointer or cursor movement keys to move the tracking rectangle to select a position for the text area and press button 1 or press the Enter key. The Mark Text panel is displayed.

12. In the Mark Text panel, type the text and specify alignment and styles to display text and select Enter. The specified text is then displayed in the selected marked text in the Menu definition panel.

    a. Set **Horizontal** to **Left, Center**, or **Right** to specify the horizontal alignment of the text within the boundary of the text area as displayed by the selected marked text.

    b. Set **Vertical** to **Top, Center**, or **Bottom** to specify the vertical alignment of the text within the boundary of the text area as displayed by the selected marked text.

    c. Specify **Styles** for text. Select **Word break** to wrap words within the boundary of the text area as displayed by the selected marked text. Select **Halftone** to display *greyed* text.

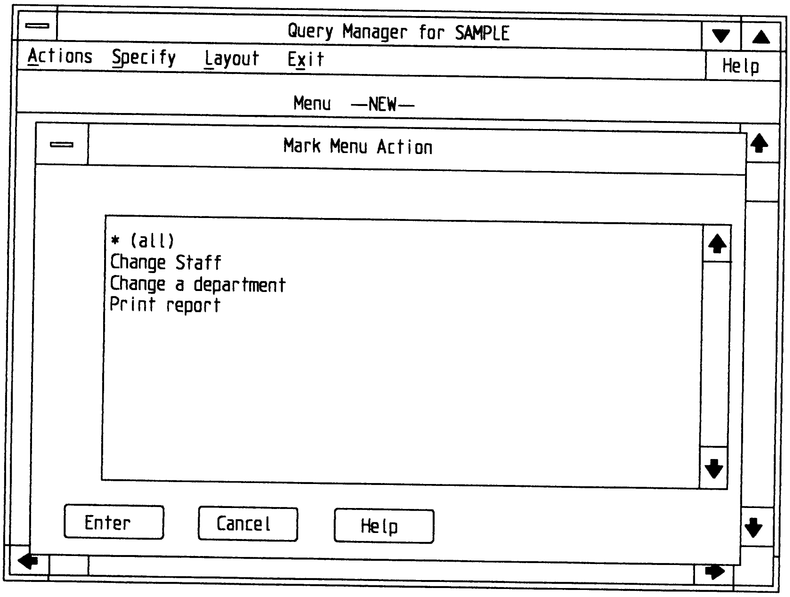
    Notes:

    a. You can display the Mark Text panel again to change the selected marked text and display style. Position the mouse pointer on the text and press button 1 twice. You can also select **Edit** from the Layout pull-down or press the Edit (Ctrl+F1) key.

    b. You can select **Mark text** to type any instructions or informational text you want to be displayed in the menu any time while you are defining a menu.

13. Size and move the selected marked text as desired.

14. To mark menu actions on the Menu definition panel, select **Layout** from the action bar and then select **Mark menu action** from the Layout pull-down, or press the Mark Menu Action (Ctrl+F4) key.



15. Select the menu action text from the Mark Menu Action menu. The tracking rectangle is displayed in the Menu definition panel.

    Notes:

    a. The list of menu action text corresponds to the **Action Text** entries in the Menu Actions panel under **Specify**.

    b. For a quick method to lay out all the actions, select * **(all)**. A tracking rectangle is displayed for the first menu action specified in the Mark Menu Action menu. Position the menu action as described in steps 16 and 17. When you select Enter, Query Manager will display another tracking rectangle for the next menu item until you have positioned every specified action.

    c. A menu action can be marked only once on the Menu panel. If you select an action that has already been marked, the action is moved to the cursor position.

16. In the Menu definition panel, use the mouse pointer or cursor movement keys to move the tracking rectangle to locate a position for the marked menu action. Press button 1 or select Enter to select a position.

17. Size the selected marked menu action if desired.

18. Repeat steps 10 through 17 until you have marked all the text and menu actions for the menu.

    Notes:

    a. If you need more information on the available editing keys in menus, select Help and then select Keys.

    b. If you want to delete a menu action or text, use the mouse pointer or cursor to select the menu action or text and select **Delete mark** from the Layout pull-down, or press the Delete Mark (Ctrl+F10) key.

    c. If no errors are detected, the menu is displayed as it looks when run.

    d. If a variable is defined in the menu title and no variable value is found, you are prompted for a value.

19. Select **Exit** from the action bar and then select **Exit Menu** from the Exit pull-down, or press the Exit Menu (F3) key. The confirmation message is displayed.

20. Select **Save and exit** from the confirmation message. The Save panel is displayed.

    Notes:

    a. If you select **Exit without saving**, the Menus primary menu is displayed again without saving the menu.

    b. If you select **Resume**, the Menu definition panel is displayed again so you can resume the task of defining or changing the menu.

21. Type the menu name and an optional comment in the Save panel, set **Share** to **Yes** or **No**, and then select Enter.

    Notes:

    a. For information on the naming conventions for menu names, see Appendix A.

    b. The menu is verified by Query Manager. If an error is encountered, the verification is ended and a message is displayed. You may correct the error before saving the menu. You can save the menu without correcting the errors, but you cannot run it.

c. If the menu is saved with **Share** set to **Yes**, any user who can access the database can select and view the menu.

Hints:

- To print the menu definition from the Menu definition panel, select **Print** from the Actions pull-down or press the Print (F9) key. The printed image of the menu is an approximation of the screen image. Menu items are adjusted to a print grid that prints one character for each cell of the grid. If you overlapped text and menu actions when defining your menu, the menu will print accordingly.
- To edit an existing menu, select the menu from the Menus menu, press the Open (F6) key, and select **Specify** and **Layout** from the action bar in the Menu definition panel to make the changes.

To Run a Menu

1. In the Main Selection menu, select **Menus**.
2. Select the name of the menu you want to run from the Menus primary menu.
3. Select **Actions** from the action bar and then select **Run** from the Actions pull-down, or press the Run (Shift+F1) key.
4. Use the menu as required for your work.
5. When you have finished using the menu, select **Exit** from the action bar and then select **Exit Menu** from the Exit pull-down, or press the Exit Menu (F3) key.

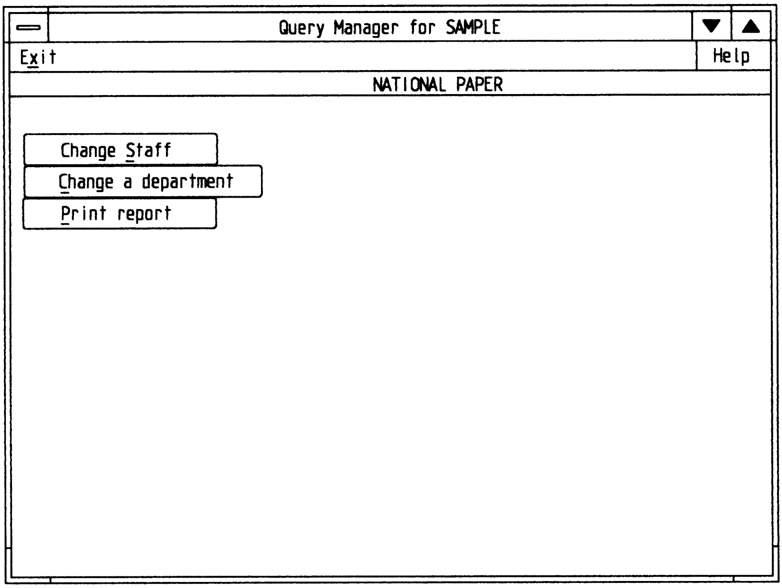
   Note: If you select **Resume** from the Exit pull-down, the Menu you selected is displayed again.

Hints:

- When the menu is run, the action bar contains **Exit** and **Help**. Selecting **Exit** allows the user to end the task, and selecting Help provides information on using menus.
- To run a menu from the Query Manager command line, select **Commands** from the action bar and then selecting **Command line** from the Commands pull-down from within the Main Selection menu or any primary menus for objects. Or, press the Command Line (Shift+F9) key. Type `RUN MENU` and the menu name in the Command Line panel.

- You can also run the menu you are designing from the Menu definition panel by selecting **Actions** from the action bar and then selecting **Run** from the Actions pull-down. Or, press the Run (Shift+F1) key.

- Even though you can verify the final menu by selecting **Check final menu**, you can only verify the variable prompting when the menu is run. To check for the variable prompting, select a menu item from the menu that requires the variable value. The following is an example of the Menu definition panel after Check Final Menu is selected.



- If you try to run a menu that contains definition errors (but was saved), you get an error message.

- A menu can be run from another menu, panel, or procedure.

Chapter 13. Designing Panels

This chapter provides information and steps for designing and running panels for use in a customized interface.

This chapter is intended for panels designers. When it addresses *you* in the text, it is referring to panel designers. The *user* refers to the person who performs work at a computer using the panels that the panel designer created.

Use this chapter in conjunction with the following chapters:

- Chapter 14
- Chapter 15
- Chapter 16.

Before You Begin

As the designer of customized display screens, called *panels*, you should have expert knowledge of the needs of the users who will use the panel. You should consider all the types of information to present on the panel and how the panel should function to answer the needs of users performing specific tasks. Designing a panel is one step in designing a customized application. In meeting the specific needs of your application, you must have a thorough knowledge and understanding of the tables and of the relationships between the tables of your database. You will also need to define procedures, menus, and use Query Manager commands.

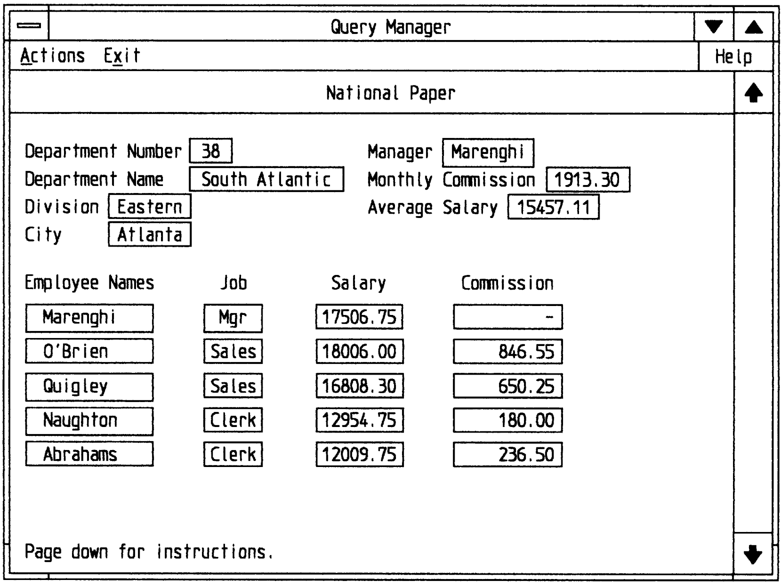
Before you begin the task of designing a panel, print each of the table definitions you intend to use in your design. You can use the print outs as a reference throughout the process of specifying the panel. If you intend to design a panel using several tables, note the referential relationships of the tables; is the table a parent or a dependent? Tables referenced as a parent will contain data rows that cannot be changed or deleted.

Back up the selected database prior to designing panels. You may also want to consider testing the panels for errors and performance requirements before making them available to users.

About Panels

Panels can be used to enter and edit data, search for data, and present information to the user. Panels can contain fields for data entry, fields for displaying data, instructional text, computed output fields, and user-defined actions (such as invoking a Query Manager command). Since you have used several panels in Query Manager, you know that a panel is a customized screen that can be used for browse-only information and that can contain one or more entry fields where you can type or enter information or a combination of the two. Panels are created by specifying the details of the fields you want on the panel and then by laying out the fields on the panel itself.

The following example of a customized panel named DEPT is from the scenario in Chapter 11. The DEPT panel includes fields for data entry (**Salary**, **Commission**), computed fields (**Monthly Commission**, **Average Salary**), output fields (**Manager**, **Employee Names**, **Job**, **Department Number**, **Department Name**, **Division**, **City**), and instructional text.



National Paper

Department Number 38
Department Name South Atlantic
Division Eastern
City Atlanta

Manager Marenghi
Monthly Commission 1913.30
Average Salary 15457.11

Employee Names	Job	Salary	Commission
Marenghi	Mgr	17506.75	-
O'Brien	Sales	18006.00	846.55
Quigley	Sales	16808.30	650.25
Naughton	Clerk	12954.75	180.00
Abrahams	Clerk	12009.75	236.50

Page down for instructions.

You can customize a panel so that a user can *add* or *change* data in tables and views and display and print data in the format of the panel layout. Panel actions are used when a panel is run to manipulate the panel. You can specify which panel operation or Query Manager command to use to tailor the set of actions used by the panel. You can design a panel that will allow an automatic execution of a procedure for each panel instance and process control at the end of the panel set. Because panels can use Query Manager commands to call menus, procedures, other panels and queries and to print reports, a user can perform a variety of database tasks without having to be directly aware of Query Manager.

Use the following sections to design panels. Once the panel is designed, the user can run the panel or you can create a procedure or menu that the user can use to access and run the panel.

Authorization

To design a panel, you should be a user with SYSADM (system administrator) authority or a database administrator for the selected database. With either of these administrative authority levels, you have access and control over all the objects within the selected database you can use or create when designing a panel.

When you create a panel, you are considered the *owner* of the panel. When you have specified the panel, a Save panel is displayed prompting you to enter a name and comment and make a **Share** selection. As owner of the panel, you can choose to grant or revoke other users the authority to access and run the panel by setting **Share** to **Yes** or **No** in the Save panel.

Users have a choice of running panels in one of two modes: *Add* or *Change*. A user accessing the panel in the Add or Change mode must have an administrative authority level, table or view Control authority, or query rows authorization for each table used in the panel design. If the user has only query rows authorization on the tables used by the panel, the user can browse the data in the panel but cannot issue panel actions that update data to the tables or change the panel definition. Users with the required authorization for each table used in the panel design can change the panel definition and save it under a *new* name.

As the owner of the panel, or as a user with an administrative authority level, you can edit the panel and save it under the *same name*. You can also import or export the panel and assign it a new name, or you can select to erase the panel.

As the designer of the panel, you can review the authorizations for each user by using the database and the table and view Authorizations functions. For more information on granting and revoking database and table and view authorizations, see Chapter 8.

Conceptualizing Panels

When you design a panel, think of the panel as an empty page upon which you can mark data fields and place text.

The user adds, changes, or deletes data in a panel by typing in *fields* in the panel. These fields are like the entry fields in Query Manager panels. Each field can have instructional text that indicates what kind of data should be typed.

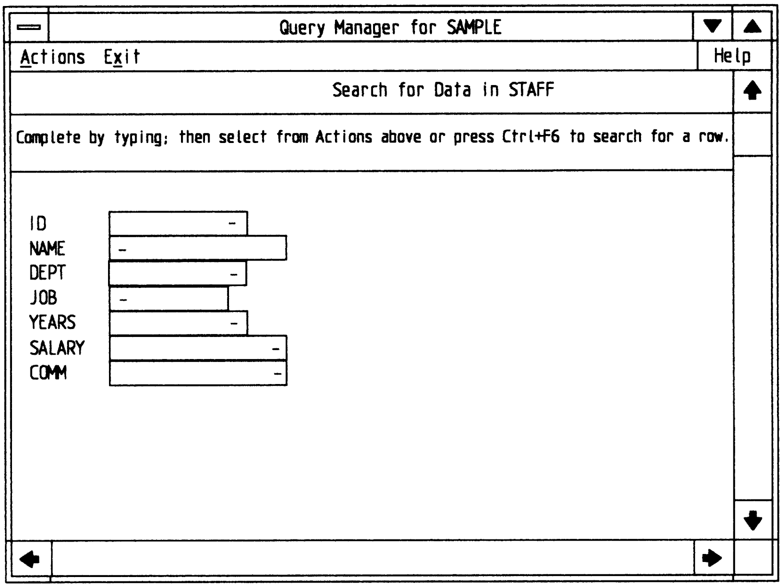
A panel can be designed to perform additions, as well as changes and deletions, to a database. There are two modes in which a panel can run: Add or Change. The Add mode corresponds to selecting **Add data rows** to add data to a table or tables. The Change mode corresponds to selecting **Change data rows** to change or delete data in a table or tables. You can also design a panel in which data can only be browsed.

Note: If the panel is based on a read-only view, no Add or Change operations are valid. For a definition of a read-only view, see "Read-Only Views" on page 7-7.

Single-Table Panels

The simplest type of panel is one designed for a single table. In the scenario in Chapter 11, a single-table panel is used to update the STAFF table, either by adding or changing data. You can lay out a single-table panel yourself by positioning the fields wherever you want on the panel, or you can use the *default definition* for the panel. The default definition panel is the same kind of panel you use to edit data in the table when you select **Add data rows** and **Change data rows** from the Tables and Views primary menu. The column names are vertically aligned under each other in the default definition panel. You can also use the default definition as a starting point for designing a customized panel for a single table or for multiple tables.

The default definition uses the column names in the table as information about each field. For example, the default definition panel, called ASTAFF, for *searching* data in the STAFF table is displayed as follows.



Search for Data in STAFF

Complete by typing; then select from Actions above or press Ctrl+F6 to search for a row.

ID
NAME
DEPT
JOB
YEARS
SALARY
COMM

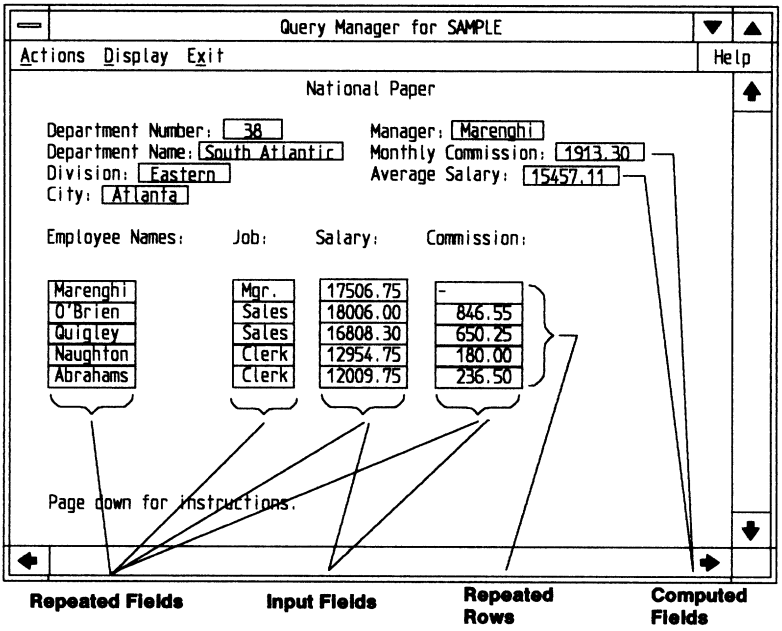
Multiple-Table Panels

You can also create a panel based on more than one table of data. To do this, you must specify the connecting relationship between all the tables accessed by the panel. In a panel, one table must be identified as the *root table*. The root table is at the top of the hierarchy of tables used in the panel. A second table can be identified as a *subtable* to the root table. Data rows can be added or changed in the root table and subtable only. One root table and one subtable can be used in a panel.

Other tables can be referenced by the panel as *lookup tables*. A lookup table is a table from which columns can be presented in the panel you are designing as output fields only. You can use data in lookup tables for computations, but you cannot add, change, delete, or search for the data in them. For example, if you had a database for a catalog of clothing, one table could contain the name of the clothing and the unit price. You could use this table as a lookup table for obtaining unit prices, the item, and the description, and then use the unit prices to compute total cost. You can use lookup tables whether you are using a subtable or not. Up to nine lookup tables can be used.

Connecting columns define the relationship between tables. There can only be one path defined, using connecting columns, to each table. All the tables you specify for the panel must be directly or indirectly connected to the root table by selecting **Connecting columns** from the Specify pull-down. A subtable must be directly connected to the root table; lookup tables can be connected to the root table, the subtable or other lookup tables. The connection between two tables consists of one column from each table. These columns must contain data values that match. Columns from the root table can be searched to create a panel set consisting of one or more instances of the panel that can be processed by the user one at a time.

The following diagram illustrates the relationships between the STAFF and ORG tables that is the basis of the DEPT panel.



National Paper

Department Number: 38
Department Name: South Atlantic
Division: Eastern
City: Atlanta

Manager: Marenghi
Monthly Commission: 1913.30
Average Salary: 15457.11

Employee Names:	Job:	Salary:	Commission:
Marenghi	Mgr.	17506.75	-
O'Brien	Sales	18006.00	846.55
Quigley	Sales	16808.30	650.25
Naughton	Clerk	12954.75	180.00
Abrahams	Clerk	12009.75	236.50

Page down for instructions.

(ORG)

ROOT	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
	38	South Atlantic	30	Eastern	Atlanta

ROOT TO SUB CONNECTING COLUMN

ROOT TO LOOKUP1 CONNECTING COLUMN

(STAFF) **Display only (No Edit)**

LOOKUP	ID	NAME
	30	Marenghi

(STAFF) **Change (Edit)**

SUB	DEPT	NAME	JOB	SALARY	COMM
	38	Marenghi	Mgr	17506.75	-
	38	O'Brien	Sales	18006.00	846.55
	38	Quigley	Sales	16808.30	650.25
	38	Naughton	Clerk	12954.75	180.00
	38	Abrahams	Clerk	12009.75	236.50

If a panel definition specifies the use of a subtable, the root table and subtable connection type must be defined for the panel. The following list identifies the four connection types that can exist between the root table and the subtable. It also provides design considerations on how table rows are processed for each connection type when rows are added or deleted.

One-to-One One row in the root table is connected to 0 or to one row in the subtable. The connection is made by using the value in the connecting column of the root table row to find the same value in the connecting column of the subtable row.

If there is a subtable row, it must only be connected to one root table row. The connecting columns must contain unique data values within each table (no duplicates). For example, a table of departments is related to a table of managers (the relationship being on department number); each department has one or no manager and each manager has one department.

If you provide the user with the **Delete and Next** panel action, then both the row in the root table and the row in the subtable (if one exists) are deleted when **Delete and Next** is selected.

If you provide the user with the **Add and Next** panel action, then both a root and a subtable row are added to their respective tables. If the value of the connecting column of the root table row or the subtable row to be added already exists in either table, the rows cannot be added.

If you provide the user with the **Change and Next** panel action and a row in the subtable does not exist, then only columns in the root table can be changed. Users receive a message when the panel is presented warning them of this condition (that no row from the subtable exists). The null character is displayed in all the fields for rows from the subtable in this case. If users type data into these fields and select **Change and Next**, they receive an error message.

One-to-Many One row in the root table can be connected to 0 or more rows in the subtable. The data value in the connecting column of the root table row must be unique with respect to all other rows in the root table. There can be duplicate data values in the subtable connecting column. For example, a table of departments is related to a table of employees (the relationship being on department number); each department has some or no employees.

If you provide the user with the **Delete and Next** panel action, then the row in the root table and all rows from the subtable (if any) are deleted when the **Delete and Next** action is selected. If the cursor is in a repeated row field and the user selects the **Delete and Next** action, only the row the cursor is in is deleted.

If you provide the user with the **Add and Next** panel action, then the values entered for the root table row and all subtable rows are added to their respective tables. If the value of the connecting column of the root table row to be added already exists in the table, the row cannot be added.

Many-to-One Many rows in the root table can be connected to 0 or one row in the subtable. The data value in the connecting column of the subtable row must be unique with respect to all other rows in the subtable. For example, a table of employees is related to a table of departments (the relationship being on department number); each employee is in a single department, but there are multiple employees in each department.

If you provide the user with the **Delete and Next** panel action, then only the row in the root table is deleted when the **Delete and Next** action is selected. If you want to delete rows from the subtable, create a single-table panel or change the connection type to One-to-Many to do so.

If you provide the user with the **Add and Next** action, and the value of the connecting column of the subtable row to be added already exists in the table, the rows cannot be added. If you want to add data to the root table only, design another panel. As in the example, a separate panel could be designed to make department table additions without reference to the employee table.

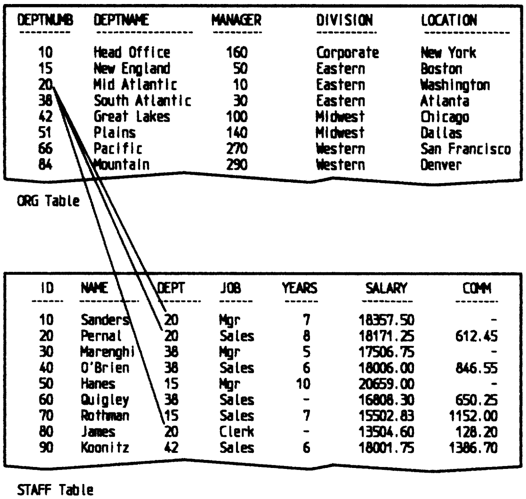
If you provide the user with the **Change and Next** panel action and a row in the subtable does not exist, then only columns in the root table can be changed. Users receive a message when the panel is presented warning them of this condition (that no row from the subtable exists). The null character is displayed in all the fields for rows from the subtable in this case. If users type data into these fields and select **Change and Next**, they will receive an error message.

Many-to-Many Many rows in the root table can be connected to 0 or more rows in the subtable. There are no uniqueness restrictions on the connecting columns. For example, an inventory table consisting of one entry per stocked item (stock number 12345, color blue) is related to a table of suppliers (the relationship being on stock number); each item can have multiple suppliers and there are multiple items since they are inventoried by color.

If you provide the user with the **Delete and Next** action, then only the row in the root table is deleted when the **Delete and Next** action is selected.

If you provide the user with the **Add and Next** action, then Query Manager attempts to add rows to the root table and to the subtable when this action is selected. If you want to add data to the root table or subtable only, design a single-table panel.

The following tables illustrate the One-to-Many connection between the DEPTNUMB column in the root table ORG and the DEPT column in the subtable STAFF that is used to define the DEPT panel described in Chapter 11.



DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lakes	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

ORG Table

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sanders	20	Mgr	7	18357.50	-
20	Pernal	20	Sales	8	18171.25	612.45
30	Marenghi	38	Mgr	5	17506.75	-
40	O'Brien	38	Sales	6	18006.00	846.55
50	Hanes	15	Mgr	10	20659.00	-
60	Quigley	38	Sales	-	16808.30	650.25
70	Rothman	15	Sales	7	15502.83	1152.00
80	James	20	Clerk	-	13504.60	128.20
90	Koonitz	42	Sales	6	18001.75	1386.70

STAFF Table

One-to-Many-Connection

If you specify a One-to-One, One-to-Many, or Many-to-One connection type, it is recommended that the table designated as the *One* have a unique index defined for the connecting column. This ensures that rows cannot be added by other programs or panels that violate the uniqueness requirements for panels based on the table. For information on defining an index, see "Defining Indexes" on page 5-13.

Defining Panels from Tables with Referential Constraints

When you select a table or set of tables to define a panel, you should take into consideration whether or not the table selected is defined with a referential constraint. There are certain restrictions in the use of panels when referential constraints are used between tables within a database. The table or tables you select to define a panel could also be designated as parent or dependent tables.

If you designed a panel using a table that is also specified as a *parent* within a table relationship, keep in mind that the the rules for deleting or changing a primary key on a parent table will also be enforced on a panel designed to run with a **Delete and Next** or **Change and Next** panel action.

If you design a panel using a table that is specified as a dependent table within a table relationship, the rules for *adding* a foreign key on a dependent table are also enforced on a panel designed to run with an **Add and Next** or **Add and Keep** panel action. The rules for *changing* a foreign key on a dependent table also apply to panels designed to run with a **Change and Next** panel action.

Database Manager will check that the rules for changing or deleting rows on parent or dependent tables are also enforced on a panel created to use a table or tables defined with referential constraints. For a definition of parent and dependent tables, and for more information on the rules for changing or deleting primary and foreign keys, see Chapter 5.

Use the following guidelines when specifying tables containing referential constraints:

- A self-referencing table should not be used as a subtable in a panel with repeated rows unless the design is such that the self-referencing constraint from a row will not affect other rows on the same panel instance. This restriction applies whether the self-reference is direct or through a cycle of references.
- A subtable in a one-to-many or many-to-many Change mode panel that is the primary key in a reference that can effect the panel root row should be avoided. This restriction applies whether the reference is direct or through a series of references.
- A Change mode panel should not have another panel nested, which can, through references, cause rows that have been accessed by the first panel to be modified or deleted.
- Changes could be applied to the wrong subtable row due to row keys having been changed on the database.
- The root table, as defined for the panel, must not be a read-only view.

Using Add or Change Mode for Panels

Once a panel is specified, laid out, and checked without errors, it can be run in either of two modes:

Add mode This mode is used when you want the user to insert new data into the database.

Change mode This mode is used when you want the user to *search for*, change, delete, or browse existing data from the database.

You select a mode by specifying the MODE=ADD or MODE=CHANGE keywords in the RUN PANEL command (either from the Command Line panel, from within a procedure, panel, or from a menu). If the MODE= keyword is not specified, the user is prompted to select the mode when the panel is run. You, or the user, can also run the panel from within the Panel definition panel or from the Panels primary menu.

Panel Instances, Panel Sets, and Search Conditions

Panel instance and *panel set* are terms used to describe how panels are presented to the user when a panel is run. Both panel instance and panel set are described in Chapter 6. A panel instance is a single occurrence of a row in the root table and any tables connected to the root table with matching connecting columns. A panel set is one or more panel instances, that are sequentially presented to the user, due to one set of search conditions.

If the user runs a panel in Add mode, each panel set contains only one panel instance. The rows to be added to the database are entered by the user; each panel instance always adds a new row to the root table and, possibly, to the subtable. If the root-table-to-subtable connection is One-to-Many, or Many-to-Many, many rows can be added to the subtable by one instance.

When the panel is displayed for adding rows to the database, all the fields on the panel are initialized to the null character, which, by default, is a hyphen (-). Once data is entered in the input fields of the panel, the user can select **Compute** (or an equivalent operation) from the Actions pull-down to retrieve any lookup tables and cause computed fields to be calculated.

If the user runs a panel in Change Mode, the number of panel instances in the panel set is equal to the number of rows in the root table that satisfy the set of search conditions. For example, if you are searching for all employees named Hayden in a table, each person named Hayden is considered one instance of the panel. There are three methods for setting search conditions:

- Selecting a **Panel search query** for the Begin Rule from the Rules for Panel menu from the Specify pull-down when you *define* the panel.
- Selecting the **Search** panel action (or an equivalent operation), which is valid when the user runs the panel. The user is placed into a Search mode where only those fields defined with a search usage will be input fields. The user must first type a value or values into the fields. These values are interpreted as search arguments. When **Perform search** is selected, the first panel instance in the panel set that satisfies the search conditions is presented.
- Selecting the **Extended search** panel action, which is valid when the user runs the panel. Extended search is similar to a panel search query, except that the user provides the name of a saved query in the Search Conditions panel, to determine the new panel set to be available for changes.

For the scenario in Chapter 11, both menu items, **Change Staff** and **Change a department**, can be used to *change* data in the database. For **Change Staff**, the menu is defined so that the user is prompted for a value for employee name. This value is used by the query, named Q1, defined for the initial Begin Rule panel search for changing data. The value typed by the user sets the n variable and the n variable is then used in the query. This query would select all the rows from STAFF where the value in the NAME column matches the value the user types. The SELECT statement for the query follows:

```
SELECT * FROM STAFF WHERE NAME = '&N'
```

Note: In a Begin Rule search query, the variable name cannot be the same as any table field name defined for the panel.

For **Change a department**, you can define a query to select all the rows from both the ORG and STAFF tables where the value in the DEPTNUMB and DEPT columns, respectively, match. This query, named Q2, is specified for the panel search query. When users select

Change a department, they are prompted for a department number. The SELECT statement for the query follows:

```
SELECT * FROM ORG WHERE DEPTNUMB = &DEPT
```

For information on defining SQL (Structured Query Language) statements, see Chapter 9.

A query specified for **Panel search query** or **Extended search** can be a prompted query or an SQL SELECT statement. The contents of the query from the row conditions or from the WHERE clause to the end of the query are incorporated into a SELECT statement that is used to do the actual accessing of the rows from the root table. A query without a WHERE clause is valid and results in the entire root table being in the panel set. If there are columns to be updated in the root table, the query is under the same constraints as a query specified when editing the data in table. For information on these constraints, see Chapter 6.

Repeated Fields and Repeated Rows

A field is one column from a table or one computed field. Not all fields have unique values within a panel; fields known as *repeated fields* can take on different values. A repeated field can be:

- A field that comes from the subtable in a One-to-Many or a Many-to-Many connection
- A field that comes from a lookup table connected to a subtable in a One-to-Many or a Many-to-Many connection
- A computed field using a repeated field (based on a formula).

For example, if a panel is designed to show employee names for a particular department, employee name is a repeated field since there is one department with several employees. Repeated fields can exist for a panel only when the relationship of the root table to the subtable is either One-to-Many or Many-to-Many. The repeated fields are actually in the subtable and any lookup tables connected to the subtable in a One-to-Many or Many-to-Many connection.

Since repeated fields can take on different values within a panel, there needs to be a way to show the fields multiple times. For example, in the scenario in Chapter 11, the fields for **Employee Names**, **Job**, **Salary**, and **Commission** need to be shown multiple times. This is done by specifying a *repeated field area* on the panel. A repeated row is a line on the panel that has one or more repeated fields marked on it.

The repeated field area contains a scrollable set of repeated fields; each repeated row represents one set of unique values for the repeated fields based on one connection from a root table row to a subtable row. For example, a field for department number 38 in the root table connecting to one corresponding row in the subtable would constitute one related set of values for the repeated fields. However, the root table row for department 38 is not part of the repeated rows. If a lookup table is connected to a subtable, its fields become repeated fields that can be used on repeated rows. Computed fields can also be repeated fields.

If you use repeated rows in your panel, keep in mind the following:

- The connection between the tables must be specified in the Connecting Columns panel as One-to-Many or Many-to-Many.
- If you mark a repeated field in a repeated row, the repeated field is marked in all the repeated rows. If you delete a repeated field from a repeated row, the repeated field is removed from all the repeated rows.
- Repeated rows are placed on consecutive lines on the panel by vertically sizing the repeated field area on the Panel definition panel.
- Only one set of repeated rows is allowed on a panel.
- Data from lookup tables connected through the subtable are repeated fields and so can only be used on repeated rows.
- Computed fields that use repeated fields, outside of summary functions, as part of the formula are repeated fields (depending on their use in a formula) and so can only be used in repeated rows.
- If there is a single repeated row on a panel, only one set of repeated field values is scrolled at a time.

Panel Actions

You can specify the menu items you want from the Actions pull-down, which is available in the action bar when the panel is run. These items are available to the user of the panel. Defaults are provided for the Actions pull-down. You can change the action text for each menu item to suit the needs of your application and the expertise of the user of the panel (by selecting **Panel actions** from the Specify pull-down).

When you specify panel actions, you are deciding how information can be entered and changed within the panel. A panel action is actually performed when you run the panel. When a new panel is opened, you are given a set of panel actions that can be customized. How you design a panel must take into account the mode in which you wish to run the panel as well as whether or not the panel is defined for single or multiple tables.

When you are specifying panel actions, keep in mind the following:

Mode Determines whether the action being defined is valid for Add or Change mode, or both. This field can have one of two values: **A** for **Add data rows** or **C** for **Change data rows**. If the **Mode** field is left empty, the action item is available for both Add and Change modes.

When the panel is run, the actions are in the sequence in which they are specified on the Panel Actions panel. Those actions valid for Add are present in Add mode. Those actions valid for Change are present in Change mode.

If you specify the panel mode as Change only, and no table fields in the panel are defined with a usage of C, the panel can be used only for browsing.

Action Text Determines the text that is displayed in the Actions pull-down when the panel is running, depending upon which mode of execution is selected (either Add or Change mode). The text can be from 1 to 21 characters. A maximum of 19 actions can be defined for each mode (Add or Change).

For the default definition panel, two sets of panel actions are pre-defined according to the two modes available (Add or Change). Both are displayed in the Panel Actions panel. These are similar to the Actions pull-down that is displayed when adding or changing data rows for a table. For information on adding or changing data rows for a table, see Chapter 6.

Mnemonic A required 1-character abbreviation for the action. It must be unique within each panel mode (Add or Change) and must be a character contained in the action text. When the action text is displayed in the Actions pull-down, the first occurrence of the mnemonic character is highlighted.

Action Key Assigns an action key to each action. Action keys can be any unused F-key, any unused Control (Ctrl) key, and unused Shift and F-key combination. Keys that are *reserved* by Query Manager are: F1, F3, F4, F10, F11, F12, Shift+F2, Shift+F3, Shift+F8, Shift+F11, Shift+F12, Ctrl+F8, Ctrl+F11, and Ctrl+F12. Examples of keys you can use are F2, F6, and Ctrl+F10. It is recommended that you make key assignments consistent with the use of the keys in the default set of panel actions.

Panel Operation Determines the panel operation or Query Manager command that is performed when the panel action item is selected. The maximum length of this field is 50 characters. A panel action can be either one Query Manager command or one panel action. For information on Query Manager commands, see Chapter 10. Panel actions are described in the following text.

Valid Panel Actions for Add Mode and Change Mode

The validity of each panel action is determined by the mode in which the panel is running (Add or Change). The following chart indicates the valid panel actions for Add mode, Change mode, and both Add and Change mode, as well as any runtime differences of the panel actions for panels based on single or multiple tables.

Note: If the panel is based on a read-only view, no Add or Change panel actions are valid. For a definition of read-only views, see "Read-Only Views" on page 7-7.

Panel Action	**Mode**	**Description**
Add and Keep	Add	Performs an addition to the database to add the rows specified in the panel, and displays once again the just-added panel values. These values can then be used as a template for another addition. • If either the root table or subtable is defined with a One connection, a check is made to ensure uniqueness of the connecting column data values in the table designated as having the One connection. • If the user does not type in a field that is present on the panel, the field remains null if it is not a required field for the table. If it is a required field, an error occurs. • The value for the connecting column field for the root table is automatically set into the subtable-connecting column field. If a procedure causes the subtable-connecting column to change, the panel still uses the root table value. • If the connecting column field for the subtable is on the panel, it is an output field. • You can define a procedure containing an Add and Keep panel action and define an action that will run the procedure. • If either the root table or subtable is designated as a dependent table, a check is made to ensure that the data being added to the foreign key in the dependent table exists in the specified primary key for the corresponding row in the parent table. See Chapter 5 for a discussion on the rules governing tables defined with referential constraints.

Panel Action	Mode	Description
Add and Next	Add	Performs an addition to the database to add the rows specified in the panel. For a panel based on multiple tables, additional rows are added to both the root table and subtable, as follows: • If either the root table or subtable is defined with a One connection, a check is made to ensure uniqueness of the connecting column data values in the table designated as having the One connection. • If the user does not type in a field that is present on the panel, the field remains null if it is not a required field for the table. If it is a required field, an error occurs. • The value for the connecting column field for the root table is automatically set into the subtable-connecting column field. If a procedure causes the subtable connecting column to change, the panel still uses the root table value. • If the connecting column field for the subtable is on the panel, it is an output field. • You can define a procedure containing an Add and Next panel action and define an action that will run the procedure. • If either the root table or subtable is designated as a dependent table, a check is made to ensure that the data being added to the foreign key in the dependent table exists in the specified primary key for the corresponding row in the parent table. See Chapter 5 for a discussion on the rules governing tables defined with referential constraints.
Blank panel	Add	Presents a panel instance with all null values in data fields (input and output) and computed fields while running a panel in Add mode.

Panel Action	**Mode**	**Description**
Change and Next	Change	Performs a change to all of the input data fields. Change and Next is performed on the row in the root table and all rows in the subtable, depending on the data fields to be changed. The cursor can be positioned anywhere on the panel when Change and Next is selected; all input fields are updated in both the root table and subtable rows. The next panel instance in the panel is displayed. If the end of the panel set is reached, the user is placed in the empty Search Mode panel, the Search Conditions menu is displayed, or the panel is exited, depending on the End Rule item selected from the Rules for Panel menu. The data values in the connecting column for the root table between the root table and the subtable cannot be changed and therefore are output fields on the panel. If either the root table or subtable is designated as a parent or dependent table, the rules for changing primary or foreign keys are enforced. Refer to Chapter 5 for the rules governing tables defined with referential constraints. You can define a procedure containing a Change and Next panel action and define an action that will run the procedure.
Compute	Add and Change	Causes computed fields to be calculated and displayed on the current panel instance if the field is marked. Any lookup table references are also accessed if the data field connecting the lookup tables changes. No changes are made to the database. You can define a procedure containing a Compute panel action and define an action that will run the procedure.

Panel Action	Mode	Description
Delete and Next	Change	If the cursor is positioned on a repeated row, Delete and Next applies only to that one row and that one row alone is deleted from the database. The user remains on the panel instance and all fields are shown with null values in the line that is deleted from the database. If the cursor is not positioned on a repeated row, the row in the root table represented on the current panel instance is deleted. If there is a subtable defined and the relationship is One-to-One or One-to-Many, the subtable rows are also deleted. For Many-to-One or Many-to-Many relationships, only the root table row is deleted. The next panel instance in the panel set is displayed. If the end of the panel set is reached, the empty Search panel is displayed, the Search Conditions menu is displayed, or the Panel is exited, depending upon the End Rule item selected from the Rules for Panel menu. If either the root table or subtable is designated as a parent table, the rules for deleting the parent table's primary key are enforced when Delete and Next is selected. Refer to Chapter 5 for information on the rules governing tables defined with referential constraints. You can define a procedure containing a Delete and Next panel action and define an action that will run the procedure.
Extended search	Change	Displays the Search Conditions menu to allow the user to select the name of a query to use to determine the new panel set to be available for changes. If no data satisfies the search conditions, the user is placed in the empty Search Mode panel, the Search Conditions menu is displayed, or the panel is exited, depending on the End Rule item selected from the Rules for Panel menu. You can define a procedure containing an Extended Search panel action and define an action that will run the procedure.

Panel Action	**Mode**	**Description**
Next	Change	Presents the next panel instance in the panel set. If the end of the panel set is reached, the user is placed in the empty Search mode panel, the Search Conditions menu is displayed, or the panel is exited, depending on the End Rule item selected in the Rules for Panels menu. You can define a procedure containing a Next panel action and define an action that will run the procedure.
Print	Add and Change	Prints the current panel instance. No changes are made to the displayed panel instance. You can define a procedure containing a Print panel action and define an action that will run the procedure.
Search	Change	Places the user in Search mode. Once in Search mode, the user can specify the search conditions by typing values in the fields. A field or fields that contain the null character will match on any character. Once a search is performed, the old panel set is closed and a new panel set is available. Even if no data meets the new search conditions, the old panel set is closed and the user is either placed in the empty Search Mode panel, the Search Conditions menu is displayed, or the panel is exited, depending on the End Rule item selected from the Rules for Panel menu. You can define a procedure containing a Search panel action and define an action that will run the procedure.
Show field	Add and Change	Presents a panel where the user can display the field that has a length greater than 254 bytes and has been defined as a character (variable) field. Show field is not valid for repeated fields.
Refresh	Change	Returns the current panel instance to its condition at the time of the last database action. Refresh allows the user to undo typing performed on this panel instance, but does not undo any deletions of repeated rows that were already performed.

Delayed Processing of Panel Actions

For some of the valid panel actions that can be run from a procedure, all or part of the processing is delayed until the procedure that is executing the panel action is finished and control has returned to the running panel. Panel actions that work differently when executed from a procedure are discussed as follows:

Add and Keep When an Add and Keep panel action is run from a procedure, an addition is made to the database to add the row or repeated rows specified on the panel. The data from the just added rows is then re-displayed in the panel. For a multiple table panel, additional rows are added to both the root tables and subtables depending upon the data values being added and the relationship between the root and subtables. If there is a subtable defined and the relationship is One-to-One or One-to-Many, a check is made to ensure the uniqueness of connecting column data values in the table designated as having a *One* connection.

Add and Next When an Add and Next panel action is run from a procedure, the root row and any repeated rows specified on the panel are added to the database at that time.

The *Next* part of the panel action, which will change the variable pool values to null and display a blank panel, is delayed until the procedure containing the Add and Next panel action is finished and control is passed back to the running panel.

Delete and Next When a Delete and Next panel action is run from a procedure, the row in the root table represented on the current panel instance will be deleted from the database. If there is a subtable defined and the relationship is One-to-One or One-to-Many, the subrows are also deleted. For Many-to-One or Many-to-Many relationships, only the root row will be deleted.

The cursor position does not affect processing when the Delete and Next panel action is run from a procedure. If the screen cursor is positioned on a repeated row, the Delete and Next panel action will still apply to the panel instance, and not just to that one row.

The *Next* part of the panel action, which sets the variable pool values and displays the next panel instance, is delayed until the procedure containing the Delete and Next panel action is finished and control is passed back to the running panel. If the *Next* part of the panel action causes the end of the panel set to be reached, then the specified End Rule is processed the same as it would be if the Delete and Next panel action was executed directly from the running panel. An exception occurs when the end of the panel set is reached and a Search, Extended Search, or Quit panel action is then run by the procedure. In this case, the End Rule is not processed.

Change and Next When a Change and Next panel action is run from a procedure, the row or repeated rows specified on the panel are changed in the database at that time.

The *Next* part of the panel action, which sets the variable pool values and displays the next panel instance, is delayed until the procedure containing the Change and Next panel action is finished and control is passed back to the running panel. If the Change and Next panel action causes the end of the panel set to be reached, then the specified End Rule is processed the same as it would if the Change and Next panel action was executed directly from the running panel. An exception occurs when the end of the panel set is reached and a Search, Extended Search, or Quit panel action is then run by the procedure. In this case the End Rule is not processed.

Next When a Next panel action is run from a procedure, nothing is done by the running panel at that time.

The Next panel action, which sets the variable pool values and displays the next panel instance, is delayed until the procedure containing the Next panel action. is finished and control is passed back to the running panel. If the Next panel action causes the end of the panel set to be reached, then the specified End Rule is processed the same as it would if the Next panel action was run directly from the running panel. An exception occurs when the end of the panel set is reached and a Search, Extended Search, or Quit panel action is then run by the procedure. In this case, the End Rule is not processed.

Search When a Search panel action is run from a procedure, the running panel does nothing at that time. Search, which places the user in the Search mode, is delayed until the procedure containing the Search is finished and control is passed back to the running panel. No other panel actions are allowed to be run in a procedure after a Search panel action.

Extended Search When an Extended Search panel action is run from a procedure, nothing is done by the running panel at that time. The Extended Search panel action, which presents the Search conditions menu to allow you to specify the name of the query to determine a new panel set, is delayed until the procedure containing the Extended Search panel action is finished and control is passed back to the running panel. No other panel actions are allowed to be executed in a procedure after an Extended search panel action.

Quit Panel When a Quit Panel panel action is run from a procedure, nothing is done by the running panel at that time. The Quit Panel panel action, which exits the running panel, is delayed until the procedure containing the Quit panel panel action is finished and control is passed back to the running panel. No other panel actions are allowed to be run in a procedure after a Quit Panel panel action.

The result of delayed processing is that the variable pool values for the current panel instance are available to the procedure after the execution of any panel actions.

Sequencing of Panel Actions within Procedures

You can only have a single panel action, which contains *Next* as in Add and Next, Delete and Next, and Change and Next. No panel actions are allowed after a Search, Extended Search, or Quit panel action.

Nesting

A panel action can cause another panel to be executed. When a panel action is run by a procedure and there are nested panels running, the most deeply nested panel is the one that causes the procedure to start and then process the panel operation.

You should not nest menus; for example, a menu that calls a procedure containing panel actions. Any problems encountered during the execution of the panel action will cause an error message to display on the menu and not on the panel.

Screen Display

There is no screen interaction with the running panel from the time the procedure, which contains the panel actions, is called until execution of the procedure is finished and control returns to the panel. Any error encountered during the execution of the procedure is displayed when control is returned back to the running panel.

Panel Actions in Initial Procedures

The initial procedure (Instance Rule) you specify in the Rules for Panels menu is allowed to run the same panel actions as any other procedure executed from a panel. When specified, initial procedures are run automatically before displaying each panel instance and can access variables from the current panel instance.

When an initial procedure contains an Add and Next, Add and Keep, Delete and Next, Change and Next, or Next panel operation, a looping effect is created. The *Next* part of the panel action executes the initial procedure for the next panel instance again. If you do not specify a Quit Panel panel action after an Add and Next or Add and Keep panel action, you will cause an *infinite loop*. In this case, you should create the procedure to contain logic that will manually stop execution; for example, by using a counter that is a global variable.

The initial procedure you specify can access variables for the panel instance and can therefore set any appropriate initialization. For example, you could specify a procedure to set variables based on the user ID of the user who logged on to Database Manager. The procedure could also set a value for a field that is not marked on the panel thereby not allowing you to change the field. The field could still be displayed by having a computed field that is set equal to the field. For example, you may not want the user to change time stamps or assigned order numbers.

It is also possible to access data from a procedure. For example, suppose you specify the initial procedure in the Rules for Panels as PROCEDURE1. This procedure is an initial procedure that needs a value from a table to assign a new order number; the value needs to be accessed and incremented by 1 so that the order numbers are always unique.

PROCEDURE1 can run another panel in Change mode provided that:

- The panel is defined against the table that contains the latest available number.
- The initial search query for the panel accesses the row containing the value.
- The initial procedure for the panel accesses the value from the variable pool (using the GET CURRENT command), increments the value, sets it back into the variable (SET CURRENT command), requests that the database be changed (CHANGE MESSAGE=NO command), and specifies that the panel is stopped (QUIT PANEL command).

When control returns to PROCEDURE1, the value for the new order number will be available in the panel variable. This value can be accessed (GET CURRENT command) and set into the appropriate field (SET CURRENT command). As previously stated, you should be familiar with how to use Query Manager commands when writing procedures. For more information on using Query Manager commands, see the *OS/2 Command Reference*.

Defining Procedures to Automate Panel Actions

You can make selections in the Rules for Panels menu to search out panel instances and then specify procedures to *automatically* access and update panel fields, and to change, add, or delete data rows for a running panel. You can also specify what should happen when the end of a panel set is reached.

The Begin Rule in the Rules for Panels menu is used to specify the initial **Panel search query**. When you are running a panel in the Change Mode, a Begin Rule panel search query can be used to find the panel set you want to change or delete.

The **Instance Rules** item in the Rules for Panels menu allows you to specify a procedure for a panel running in either Add or Change mode. You should define procedures keeping in mind the type of mode the panel will be running in. You do not have to specify a procedure every time you define a panel. Procedures have access to the panel variables and can set default values, update the current values, do input validation, or copy computed field values into panel fields.

The procedure specified can also use panel actions as commands. You can run a procedure as a result of a panel action or as an initial procedure. Refer to Chapter 14 for more information about how to write procedures to run panel actions.

Keep in mind that procedures are not part of the panel definition. You can therefore specify a procedure and then create the procedure after you have defined the panel. If you change a procedure after the panel is defined, the changes would take effect the next time you run the panel.

You can select List to view all the procedures matching the current active qualifier for lists. You can specify procedures for each mode as follows:

Procedure (Add) This item allows you to specify the procedure that is called just before each instance is displayed when a panel is running in Add mode. The procedure will be called initially before displaying the first instance, and will be called after an Add and Next or Add and Keep panel action is performed.

Procedure (Change) This item allows you to specify the procedure that will be called after each panel instance has been read from the database and just before each instance is displayed when a panel is running in the Change mode. The procedure will be called initially before displaying the first instance in the panel set, and will be called after the *Next* in a Change and Next, Delete and

Next, or Next panel action is performed, except at the end of a panel set.

Each of the panel's table fields are defined as locally created variables in the variable pool. Each panel instance resets all of the panel's local variables when data is read from the database.

The **End Rules** from the Rules for Panels menu allows you to control how to end processing of the panel set. You can control or end panel processing with one of the following End Rule items:

Search Selecting this item places the user in the Search mode when the end of the current panel set is reached or when no rows are found as a result of a search and the empty Search Mode panel is displayed.

Extended Search Selecting this item displays the Search Conditions menu when the end of the current panel set is reached or when no rows are found as a result of a search. This item allows the user to specify the name of the query that will be used to determine the new panel set to be available for changes.

Exit the panel Selecting this item causes the panel execution to stop when the end of the current panel set is reached or when no rows are found as a result of a search. In addition, by selecting this item, you can also select whether or not to display a message explaining to the user that the last panel instance has been processed by a Change and Next, Delete and Next, or Next panel action, or that no rows were found as a result of a Search, Extended Search or Initial Search query.

Specifying a Panel

To design a panel, you can specify the tables to be used, panel fields, the connecting columns, panel actions for the Actions pull-down, Rules for Panels search and panel action control conditions, titles for the panel, and presentation of rows that need to be repeated on the panel. The following is a brief description of each item in the Specify pull-down.

Default definition

The panel that is generated for the default definition looks like the panel used when editing the data in the table. You can also use the default definition as a starting point for the customizing of single- or multiple-table panels. The table specified for the default definition becomes the root table for the panel. The default definition causes the following variables to be created when the panel is run: PANELTITLE, INSTRUCTIONS, and FIELDx, where x is a number.

Table selections

Use to specify the tables to be used for the panel. You must indicate whether a selected table is the root table, the subtable, or a lookup table. A maximum of 11 tables can be used within one panel. If the subtable is a view that is based on the root table, the results are unpredictable.

Table fields

Allows you to designate which columns in the tables are to be used for the panel. These panel field names are variable names. Data is read from the table and put into Query Manager variables when the panel is run. All fields defined do not have to be displayed on the panel; they could be defined in order to make them available as Query Manager variables (for example, within a computed field). As Query Manager variables, they are also available to panels, menus, and procedures run from the panel. A field can be used for adding, changing, computing, or presenting data. When you fill in the Table Fields panel, you can select Help for detailed information on each heading in the panel.

Connecting columns

Use to define the connections between the tables selected for the panel in the Table Selections panel. The Connecting Columns panel must be completed when multiple tables are used in a panel. Connecting columns must be defined from the root table to the subtable if a subtable is defined. There must also be connections to all lookup tables. A lookup table can be connected to the root table, subtable, or to another lookup table as long as there is a path back to the root table or subtable.

A **Connection Type** entry field can only be specified for the root table to the subtable connection. The connection type for the root table or subtable to a lookup table, or a lookup table to another lookup table, is always assumed to be One-to-One. Even if the connection to a lookup table is actually One-to-Many or something else, only the first row is selected from the database. You cannot connect to a table more than once; that is, you cannot specify the same table type more than once in the **To Table Type** entry field from the Connecting Columns panel.

Computed fields

Use to specify fields that have a value that is the result of a formula. Computed fields can be used to present data on the panel (if you mark them); they can never be used to enter data. A computed field can be used in the formula of another computed field. Computed fields are Query Manager variables. As Query Manager variables, they are also available to panels, menus, and procedures run from the panel. An example of a computed field is **Cost**, which contains the value from an input field for **Units** multiplied by the value of an output field for **Price**.

A computed field expression can include the summary functions SUM, AVG, COUNT, MIN, or MAX; these functions must have repeated fields as their arguments. Using these functions does not cause a computed field to become a repeated field. A computed field that includes a repeated field that is outside an argument of a summary function is itself a repeated field. A repeated computed field can only be marked in the repeated field area.

When you fill in the Computed Fields panel, you can select Help for detailed information on each heading in the panel.

Panel actions

Use to tailor the specific menu items you want in the Actions pull-down, which are available from the action bar when the panel is run. These items are to be selected by the user of the panel. The actions for both Add and Change mode are defined in this panel. Query Manager provides a default set of menu items for the Actions pull-down similar to those actions provided when editing the data in tables. If the items do not meet your needs, you can specify different text for each item, along with different actions to be performed when that item is selected by the panel user.

Rules for Panels

You can make selections in the Rules for Panels menu that allow automated executions of a procedure for each panel instance and control how to end Change mode processing.

The Begin Rule option provides a way for you to specify an initial panel search *query* for panels operated in Change mode.

The Instance Rules options allow you to specify *initial procedures* for a panel running in Add or Change mode. The specified procedure will be run automatically before displaying each panel instance and can modify variables for the panel instance.

You can specify how to stop processing of panels operated in Change mode when the end of the panel set is reached by selecting one of the items available under the End Rule option. You can specify whether to display the Search mode panel, the Search Conditions menu, or exit the panel.

Title line

Use to indicate the panel title, which is centered on the first line of the panel when it is run. A title is not required. You can create a title to correspond with each mode type. Titles can be specified for Add, Change, or Search modes. If the default definition is used, a variable is assigned to the title and is replaced depending on the mode. The title is the same as the title used when you select **Add data rows** or **Change data rows** from the Actions pull-down in the Tables and Views primary menu.

Repeated rows Use to determine how many repeated rows you want to add using one panel instance and how repeated rows should be listed on the printed output. For example, if the number of repeated rows you want to add is the same as the number of repeated rows to be listed on the printed output, fields below the repeated rows would always print in the same location. The values specified in this panel are independent of the actual number of repeated rows that are displayed on the screen at one time. The number of rows displayed is determined when you lay out the panel. If, when you run the panel, the number of rows displayed is less than what you specified, it is because fewer rows were retrieved from the database.

Examples of Specifying Panels and Procedures to Automate Panel Actions

The example panel illustrated in the steps under "To Specify a Panel" on page 13-36 is from the scenario in Chapter 11. The following examples illustrate how you could design a panel and procedures to run automated panel actions.

Suppose you want to add several data rows to the STAFF table, and you want each row to have the same entry for DEPT column. You would, therefore, want to create a panel and set default values for the DEPT field. To quickly specify a panel, you can select **Default definition** and then specify STAFF to automatically define a panel based on the STAFF table. Then, in the Rules for Panels panel, specify the name of the Procedure (Add) item as `INIT_DEPT`. Save the panel.

Create the INIT_DEPT procedure to set default values in the STAFF panel as follows:

```
/* initialize department number to 39 */
'SET CURRENT (FIELD3 = 39)'
```

Note: `FIELD3` is the panel field corresponding to the department number field in the STAFF table.

When the panel is run in Add mode, a blank panel is displayed except for the DEPT field, which displays the value 39. You can type values in the other fields in the panel. When **Add and Next** is selected, the current row is added to the STAFF table, and a blank panel is displayed again except for the DEPT field, which displays the value 39. The next row can be specified and added to the STAFF table. You can continue to repeat these steps until you have added all the data you want.

In this way, many fields on the panel can have default values set before being displayed. Also, note that the fields whose values are being set by the procedure specified in the Instance Rules panel do not have to be displayed on the panel. By doing this, you can set values that are not displayed and cannot be altered using a procedure.

To illustrate how you can specify a panel and use a procedure to automatically validate and control data input, consider the following example. Suppose you want to run a range check for the amount specified for the SALARY field in the STAFF table. Select **Default definition** to create a panel using the STAFF table. Next, in the Panel Actions menu, alter the line with **Change and Next** so that the Panel Operation Command column contains `run proc Change`. Save the panel. Create a procedure that specifies that the value entered for SALARY must be within a particular range as follows:

```
/* range check SALARY field before CHANGE and NEXT */
'GET CURRENT (salary =FIELD6)'
if (salary > 2500) & (salary < 30000) then
'Change'
else say 'Salary is out of range.  Re-enter.'
```

Note: `FIELD6` is the panel field corresponding to the salary field in the STAFF table.

When the panel is run in the Change mode, and you perform a search, the data cannot be committed to the database unless the conditions in the CHANGE procedure are met.

If you want to print all the rows in a table that met a particular criteria, you can specify a procedure in the Rules for Panels panel when you design a panel to perform this function automatically. The procedure you specify in the Rules for Panels panel should be created as follows:

```
/* Print all support */
'Print'
'Next'
```

Specify this procedure as your Procedure (Change) in the Rules for Panels panel for any panel (existing or currently being created). When the panel is run in Change mode, and you perform a search, the procedure shown will run before displaying the first row in the panel set. The procedure will issue the commands to print the current row and retrieve the next row in the panel set; the procedure will then run again until the last instance of the panel set is processed. All rows in the panel set will process this way. There is no screen display for any of the rows, and therefore, no user interaction is required until the end of the panel set is reached.

Concurrency Considerations

Panels that are intended to be run in a multi-user environment should be designed in such a way that they allow for maximum possible concurrency. Refer to Appendix K for information on designing panels within a multi-user environment.

To Specify a Panel

The steps that follow are meant to be followed sequentially; however, they are not all required to define a panel. Skip the steps that do not apply to your work.

1. In the Main Selection menu, select **Panels**.
2. Select **—NEW—** from the Panels primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.

4. In the Panel definition panel, select **Specify** from the action bar.

   Notes:

   a. You can select **Default definition** if your panel is based on a single table and you want to automatically define a panel using the columns in a table without having to specify the panel and lay it out. You can also select the default definition as a starting point for a customized panel.

   b. Using the Specify pull-down, you can specify what tables and fields are needed for the panel as well as the relationship between the tables if multiple tables are needed. Once you have specified the panel, you lay it out.

5. Select **Table selections** from the Specify pull-down.

6. Type the names of the tables on which the panel is based in the Table Selections panel and select Enter. A qualifier can be specified as a prefix to qualify the table name.

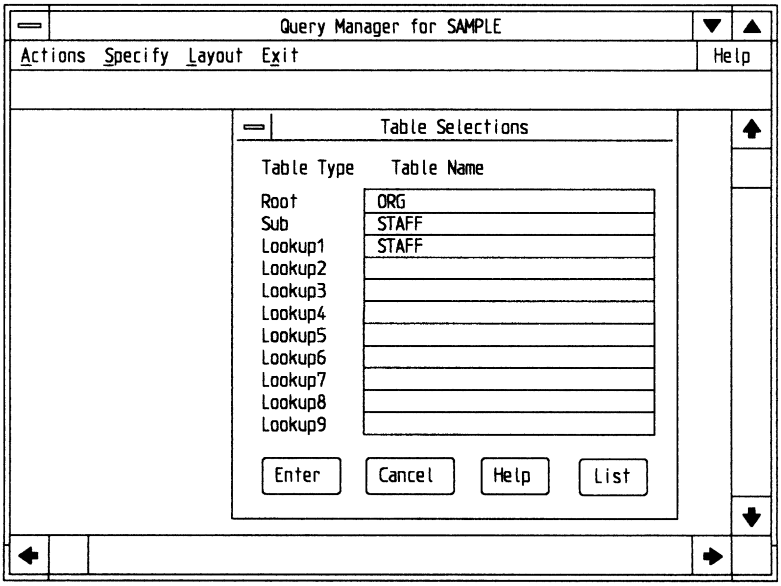
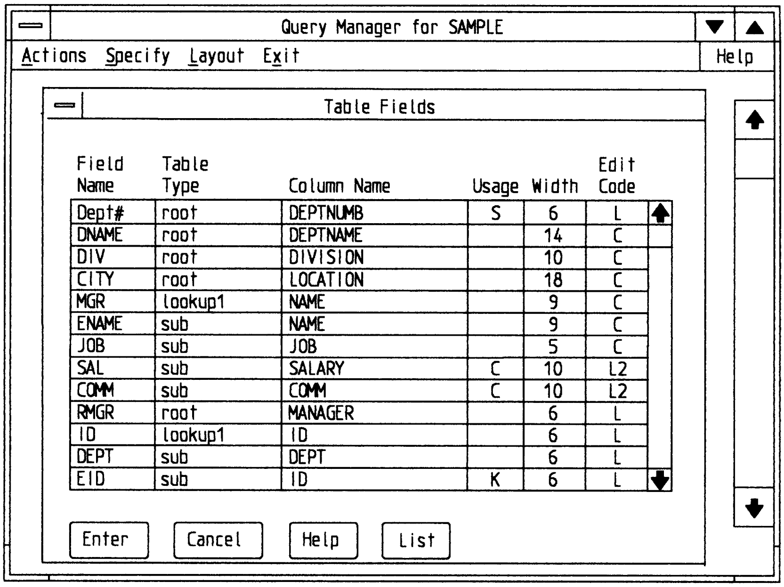


Table Type	Table Name
Root	ORG
Sub	STAFF
Lookup1	STAFF
Lookup2	
Lookup3	
Lookup4	
Lookup5	
Lookup6	
Lookup7	
Lookup8	
Lookup9	

7. Select **Specify** from the action bar and then select **Table fields** from the Specify pull-down.

8. Type the values for **Field Name**, **Table Type**, **Column Name**, **Usage**, **Width**, and **Edit Code** in the Table Fields menu and select Enter.



Field Name	Table Type	Column Name	Usage	Width	Edit Code
Dept#	root	DEPTNUMB	S	6	L
DNAME	root	DEPTNAME		14	C
DIV	root	DIVISION		10	C
CITY	root	LOCATION		18	C
MGR	lookup1	NAME		9	C
ENAME	sub	NAME		9	C
JOB	sub	JOB		5	C
SAL	sub	SALARY	C	10	L2
COMM	sub	COMM	C	10	L2
RMGR	root	MANAGER		6	L
ID	lookup1	ID		6	L
DEPT	sub	DEPT		6	L
EID	sub	ID	K	6	L

Notes:

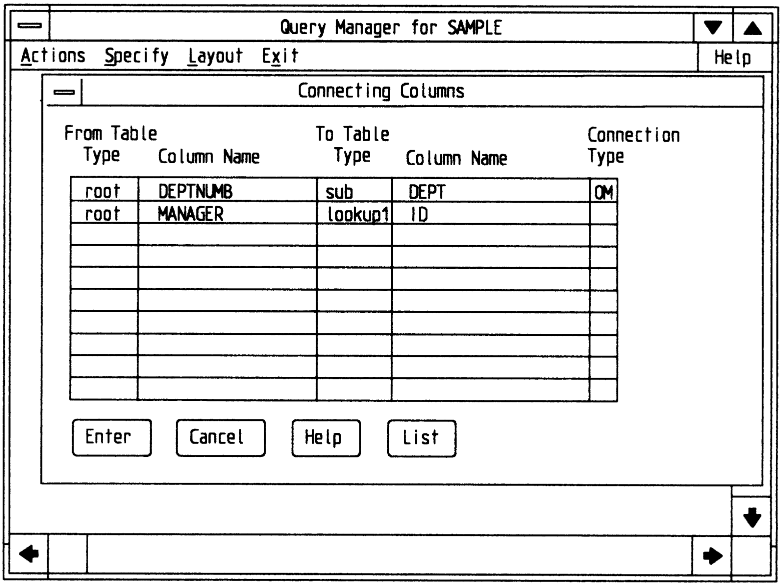
a. In the Table Fields menu, you define the fields and the field characteristics that can be displayed on the panel itself. Also, the fields you define for the panel must be mapped to columns in the tables you are using for the panel.

b. For information about edit codes, see Appendix I. Note that the TTCx edit code is *not* valid for time data, and that only the Ln and E edit codes are valid for decimal numeric data. For scientific notation data, use E, L, or Lx, where x is greater than or equal to 0 and less than or equal to 15. For decimal data, use L or Lx, where x is greater than or equal to 0 and less than or equal to 15. For small integer and large integer data, use L or L0.

c. The valid usages for table fields depend on the type of table (root, sub-, or lookup), and how the field is used in the panel. Usages can be combined in any order; the following combinations are valid: AC, SAC, KAC, KA, SA, SC, and KC. Lookup fields are always output fields; therefore, you do not specify a Usage value for them. See the *Hints* that follow these steps for information on values for **Usage**.

d. If you do not specify a width or edit code, Query Manager fills in a default for you after you save the panel (if the table exists).

9. Select **Specify** from the action bar and then select **Connecting columns** from the Specify pull-down (if you have multiple tables specified in the Table Selections panel).



From Table Type	Column Name	To Table Type	Column Name	Connection Type
root	DEPTNUMB	sub	DEPT	OM
root	MANAGER	lookup1	ID	

Notes:

a. The Connecting Columns panel allows you to define the connections between the tables selected in the Table Selections panel. The type of connection can be One-to-One, One-to-Many, Many-to-One, or Many-to-Many for the root-table-to-subtable connection only.

b. Do not specify the same table as the root table and subtable either directly or indirectly through a view.

10. Type the values for **From Table Type**, **Column Name**, **To Table Type**, **Column Name**, and **Connection Type** in the Connecting Columns panel and select Enter. Only specify Connection Type values for the root-table-to-subtable connection.

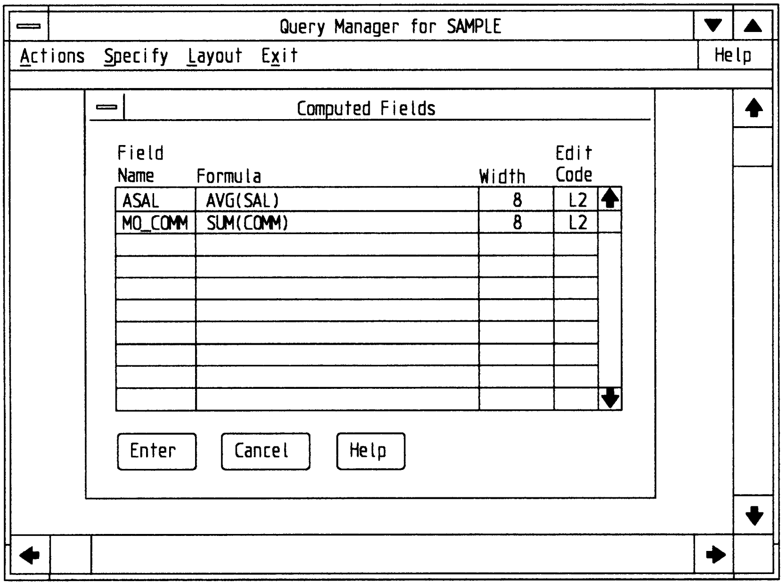
11. Select **Specify** from the action bar and then select **Computed fields** from the Specify pull-down.

Notes:

a. The Computed Fields panel allows you to specify the appearance and computation for the computed fields for the panel. Remember, each computed field contains a value that is the result of the formula you specify for that computed field.

b. You can type a table field name in the Formula column for a table column that is not actually marked on the panel. The value that field name represents can be used to compute a field that is on the panel, or the value can be used by a procedure, menu, or panel.

12. Type the values for **Field Name**, **Formula**, **Width**, and **Edit Code** in the Computed Fields panel and select Enter.



Field Name	Formula	Width	Edit Code
ASAL	AVG(SAL)	8	L2
MO_COMM	SUM(COMM)	8	L2

Notes:

a. The order of definition of computed fields is important and should be sequential; before a computed field can be used in the formula of another computed field, it must have been defined.

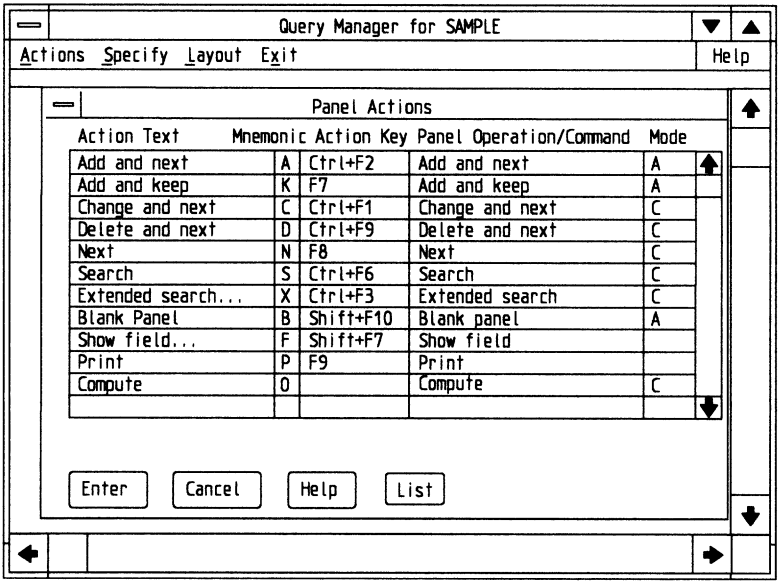
b. For information on edit codes, see Appendix I.

c. If you do not provide a width or edit code, Query Manager provides a default after you commit this panel.

13. Select **Specify** from the action bar and then select **Panel actions** from the Specify pull-down.

    Note: The Panel Actions panel allows you to specify how you want the Actions pull-down to be displayed, as well as what panel action or command you want each item from the Actions pull-down to perform when your panel is being run. **Actions** is selected from the action bar.

14. Type the values for **Action Text**, **Mnemonic**, **Action Key**, **Panel Operation/Command**, and **Mode** in the Panel Actions panel and select Enter.

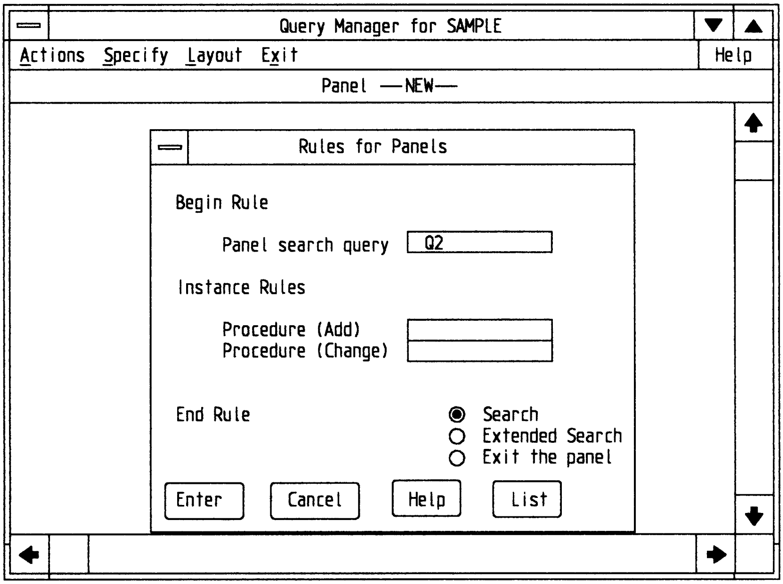


Action Text	Mnemonic	Action Key	Panel Operation/Command	Mode
Add and next	A	Ctrl+F2	Add and next	A
Add and keep	K	F7	Add and keep	A
Change and next	C	Ctrl+F1	Change and next	C
Delete and next	D	Ctrl+F9	Delete and next	C
Next	N	F8	Next	C
Search	S	Ctrl+F6	Search	C
Extended search...	X	Ctrl+F3	Extended search	C
Blank Panel	B	Shift+F10	Blank panel	A
Show field...	F	Shift+F7	Show field	
Print	P	F9	Print	
Compute	O		Compute	C

Note: If more than one Query Manager command is required to define a panel action, you should create a procedure that processes the commands and assign the RUN PROC command to the panel action. For information on Query Manager commands, see Chapter 10. For information on procedures, see Chapter 14.

15. Select **Specify** from the action bar and then select **Rules for panels** from the Specify pull-down. The Rules for Panels panel is displayed.

16. Where appropriate, type in the name of the Panel search query for the **Begin Rule**, the name of the procedure for **Procedure (Add)** and **Procedure (Change)** for the **Instance Rules**, and make a selection for the **End Rule** to control processing of the panel set.



Begin Rule

Panel search query Q2

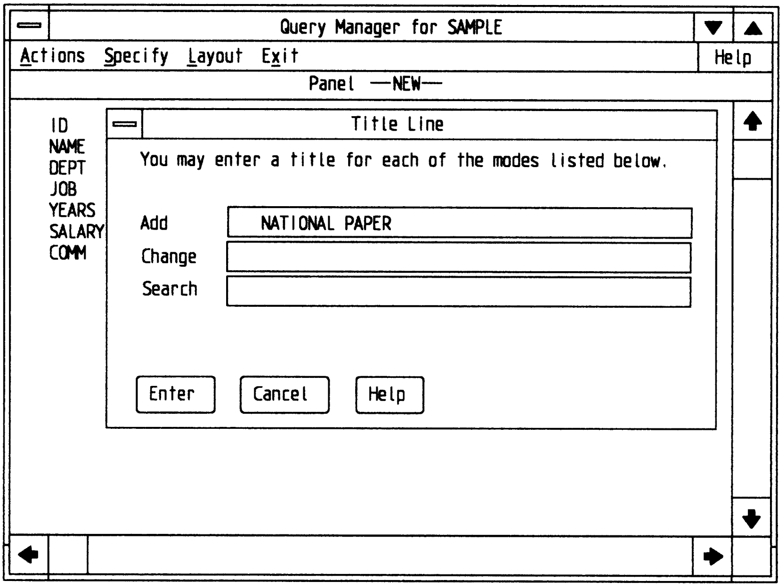
Instance Rules

Procedure (Add)
Procedure (Change)

End Rule
● Search
○ Extended Search
○ Exit the panel

Note: If you set **End Rule** to **Exit the panel**, the Exit pull-down is displayed at the end of the panel set. You can choose to display a message that results from reaching the end of a panel as a result of Next, Change and Next, and Delete and Next panel actions or when no rows are found by an extended search, search, or initial search query.

17. Select **Specify** from the action bar and then select **Title line** from the Specify pull-down.



You may enter a title for each of the modes listed below.

Add NATIONAL PAPER

Change

Search

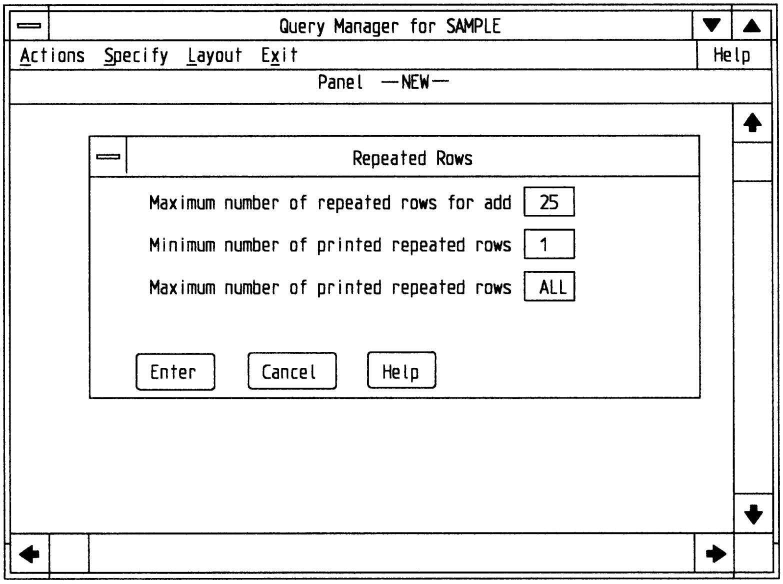
Note: The Title Line panel allows you to specify a title for each mode the panel will run in. If you are using the default definition, a variable is assigned to the title and is replaced depending on the mode (Add, Change, or Search).

18. Type the titles for the panel in the Title Line panel and select Enter.

19. Select **Specify** from the action bar and then select **Repeated rows** from the Specify pull-down.

    Note: The Repeated Rows panel allows you to specify the maximum number of rows that can be added using one instance of a panel (Add mode) and the maximum number of rows to print when the panel is printed.

20. Type the values you want to use in the Repeated Rows panel and select Enter.



Maximum number of repeated rows for add 25

Minimum number of printed repeated rows 1

Maximum number of printed repeated rows ALL

21. Continue with the steps in "Laying Out a Panel" on page 13-45.

Hints:

- Repeated fields are not available as Query Manager variables to another panel, a procedure, or a menu.
- For the **Usage** entry field in the Table Fields panel, the following values are valid (note that blank ( ) is also valid and implies that the field is for output only on the panel):

Usage Value	**Definition**
A for Add	Indicates that the data value input into the field is part of the new row when running in Add mode. *This is valid for fields from either a root table or a subtable.*
C for Change	Indicates that the column in an existing row can be changed when running in Change mode. *This is valid for fields in either a root table or subtable*, except the root table and subtable connecting columns indicated in the root-table-to-subtable connection.

Usage Value	Definition
S for Search	Indicates that a value can be typed in this field and used with the **Perform Search** panel action when running in Change mode. *Search fields can only be from the root table.*
K for Key	Indicates that this field is used to uniquely identify a single row in the subtable. One or more fields from the subtable must be defined as Key fields although Key fields do not have to be marked on the panel. If you have a One-to-Many or Many-to-Many connection type, the subtable connecting column in the root-table-to-subtable connection cannot be the only Key field since the *Many* type defines this field as not being unique. If the user selects an **Add and Next** or **Change and Next** action, and the result of the add or change causes the column key in the table to no longer be unique, an error message is returned. If this occurs, you may need to design a different panel to add data to the subtable. For subtable key columns, you should define a unique index for that column as part of the table definition. This ensures that rows will not be added that violate the uniqueness requirements of panels based on the table.

Laying Out a Panel

Before you lay out the fields for the panel, you should determine where you want each field to be displayed on the panel. You can also place instructions or informational text in the panel. It is suggested that you consider a logical order and placement of fields on the panel to aid the user in changing, adding, or displaying data.

Moving and Sizing Text and Fields

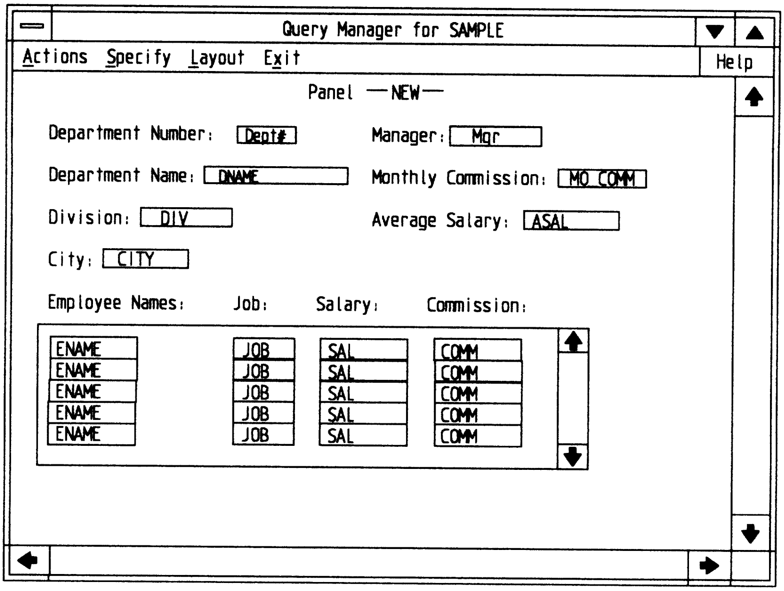
Laying out a panel involves moving and sizing text, fields, and repeated fields on the Panel definition panel. When you select **Mark text** and select a field from the **Mark Field** pull-down, a *tracking* *rectangle* is displayed on the Panel definition panel. You can use the mouse pointer or cursor movement keys to move the tracking rectangle to select a position in which you want to place the text area or field. The *selected* marked text or field is displayed within a highlighted *size border*. When the marked text or field is selected, you can use the mouse pointer or keyboard cursor keys to size the marked item, making it smaller or larger.

You can position an area of text anywhere on the Panel definition panel. When you mark fields, keep in mind that you are marking the fields you previously specified in both the Table Fields panel and the Computed Fields panel. Some of these fields may be defined as *repeated* fields. You can mark a *non-repeated* field anywhere on the Panel definition panel. You can mark a non-repeated field multiple times on the panel. You cannot, however, mark a *repeated* field more than once.

The *first* time you mark a *repeated* field, you will also automatically be laying out on the panel the repeated field *area*. Repeated fields must be marked within the *repeated field area* to form a set of repeated rows.

You can horizontally size the repeated field area to accommodate multiple repeated fields. You can vertically size the repeated field area to create additional rows of repeated fields. When running the panel, if more rows are to be changed or added than you have laid out, you can scroll the repeated field area to display additional rows.

You can lay out text and fields on a panel in an order you are comfortable with. When laying out a panel, **Adjust marks** can serve as an aid in aligning the marked text and fields. Adjust marks divides the panel area with an invisible grid and then allows you to move the selected text and fields in evenly blocked spaces. A check displayed beside Adjust marks in the Layout pull-down indicates that the option is turned on. Text and fields marked on the panel after this option is turned off are not adjusted. The following is an example of a laid-out panel.



Panel —NEW—

Department Number: Dept#
Manager: Mgr
Department Name: DNAME
Monthly Commission: MO_COMM
Division: DIV
Average Salary: ASAL
City: CITY

Employee Names:	Job:	Salary:	Commission:
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM

Once the panel is specified and laid out, you can select **Check final panel** from the Layout pull-down to ensure that the panel has been correctly defined.

To Lay Out a Panel

1. In the Panel definition panel, select **Layout** from the action bar and then select **Mark text** from the Layout pull-down, or press the Mark Text (Ctrl+F7) key. The tracking rectangle is displayed.
2. In the Panel definition panel, use the mouse pointer or cursor movement keys to move the tracking rectangle to select a position, for the text area and press button 1 or press the Enter key. The Mark Text panel is displayed.
3. In the Mark Text panel, type the text and specify alignment and styles to display text and press the Enter key. The specified text is then displayed in the selected marked text in the Panel definition panel.
   a. Set **Horizontal** to **Left**, **Center**, or **Right** to specify the horizontal alignment of the text within the boundary of the text area as displayed by the selected marked text.

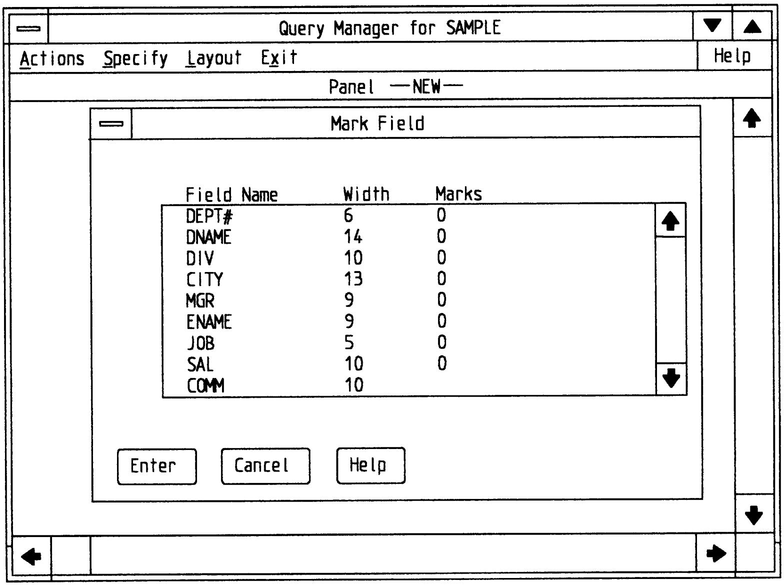
b. Set **Vertical** to **Top**, **Center**, or **Bottom** to specify the vertical alignment of the text within the boundary of the text area as displayed by the selected marked text.

c. Specify **Styles** for text. Select **Word break** to wrap words within the boundary of the text area as displayed by the selected marked text. Select **Halftone** to display *greyed* text.

Note: You can display the Mark Text panel again to change the selected marked text and display style. Select the text by positioning the mouse pointer on the text and press button 1 twice. You can also select **Edit** from the Layout pull-down, or press the Edit (Ctrl+F1) key.

4. Size or move the selected marked text as desired.

5. To mark fields, select **Layout** from the action bar and then select **Mark field** from the Layout pull-down, or press the Mark Field (Ctrl+F4) key. The Mark Field menu is displayed.



Field Name	Width	Marks
DEPT#	6	0
DNAME	14	0
DIV	10	0
CITY	13	0
MGR	9	0
ENAME	9	0
JOB	5	0
SAL	10	0
COMM	10	

To mark a *non-repeated* field, continue with step 6. To mark a *repeated* field, continue with step 9.

A repeated field comes from a subtable or lookup table in a One-to-Many or a Many-to-Many connection or from a computed field using a repeated field.

6. Select the name of the field you want from the Mark Field menu and select Enter. The tracking rectangle is displayed in the Panel definition panel.

   Note: The list of field names corresponds to the **Field Name** entries you specified in both the Table Fields panel and the Computed Fields panel. Notice that table fields with a data type of special data (as defined in the definition of each table) are not listed and cannot be marked on a panel.

7. In the Panel definition panel, use the mouse pointer or cursor movement keys and move the tracking rectangle to locate a position for the marked field. Press button 1 or select Enter to select a position. The selected marked field is displayed as indicated by the size border.

8. Size and move the selected marked field as desired.

9. To mark *repeated fields*, select the name of the repeated field from the Mark Field menu and select Enter. The tracking rectangle is displayed in the Panel definition panel.

10. In the Panel definition panel, use the mouse pointer or the cursor movement keys to move the tracking rectangle to locate a position for the selected marked repeated field. Press button 1 or select Enter to select a position.

    The repeated field area (as indicated by the solid border containing a scroll bar) is positioned on the panel. The tracking rectangle for the repeated field selected from the Mark Field menu is displayed inside the repeated field area.

11. Move the tracking rectangle within the repeated field area to locate a position for the repeated field and press button 1 or select Enter.

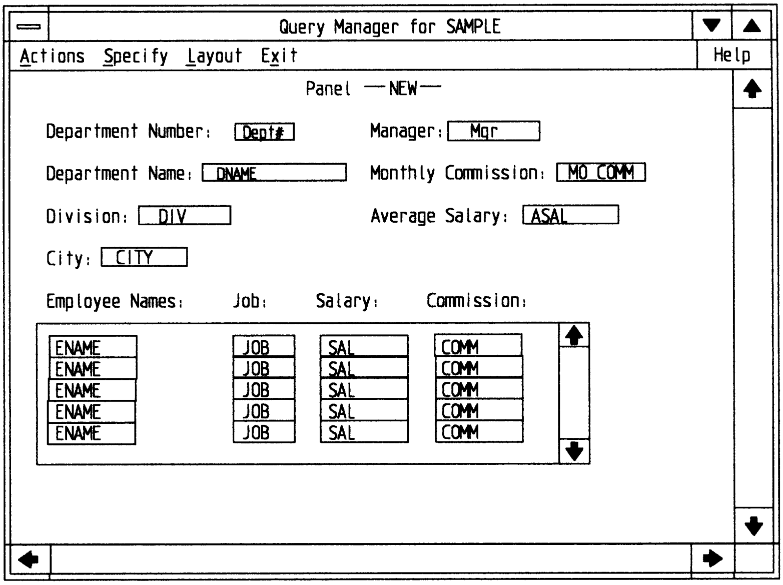
    Note: Vertically size the repeated field area to create additional repeated field *rows*.

12. Size the marked repeated field if desired.

    Repeat step 9 to select the next repeated field. Then, repeat steps 10 and 11 to mark the next repeated field.

    Note: Size the repeated field area as needed to accommodate additional repeated fields.

13. Repeat the steps to mark fields and text until you complete your panel design.



Panel —NEW—

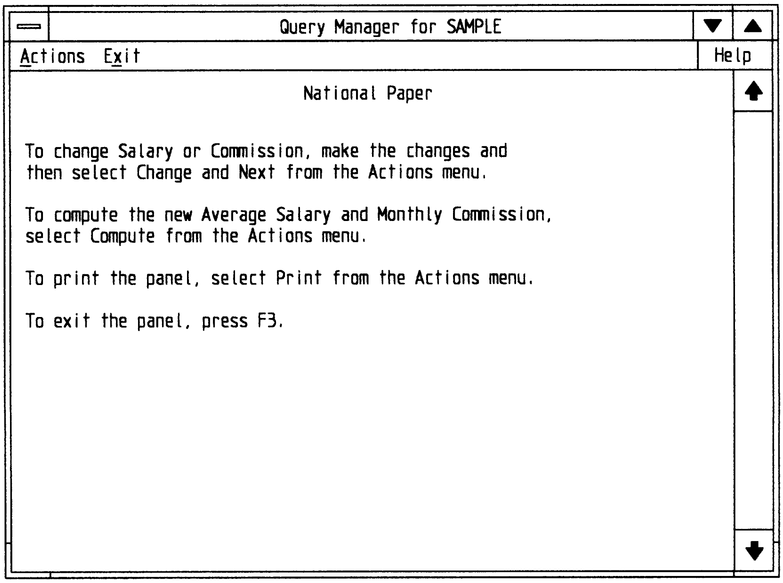
Department Number: Dept#   Manager: Mgr

Department Name: DNAME   Monthly Commission: MO COMM

Division: DIV   Average Salary: ASAL

City: CITY

Employee Names:	Job:	Salary:	Commission:
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM
ENAME	JOB	SAL	COMM



National Paper

To change Salary or Commission, make the changes and then select Change and Next from the Actions menu.

To compute the new Average Salary and Monthly Commission, select Compute from the Actions menu.

To print the panel, select Print from the Actions menu.

To exit the panel, press F3.

Notes:

a. To display Change, Add, or Search mode instructions when the panel is running, select **Instructions** from the Layout pull-down. A check mark is displayed beside this item in the Layout pull-down to indicate that mode instructions are displayed when the panel is running.

b. If you want to remove marked text or marked fields, select the text or field and select **Delete mark** from the Layout pull-down, or press the Delete Mark (Ctrl+F10) key.

c. If you need more information on the available editing keys in panels, select Help and then select Keys.

14. Select **Exit** from the action bar and then select **Exit Panel** from the Exit pull-down, or press the Exit Panel (F3) key. The confirmation message is displayed.

15. Select **Save and exit** from the confirmation message to save the panel definition and select Enter. The Save panel is displayed.

Notes:

a. If you select **Exit without saving**, the panel definition is not saved and the Panels primary menu is displayed again.

b. If you select **Resume**, the Panel definition panel is displayed again and you can continue editing the panel definition.

16. Type the panel name and an optional comment in the Save panel, set **Share** to **Yes** or **No**, and then select Enter.

Notes:

a. The panel is checked by Query Manager. If an error is encountered, the verification is ended. You can correct the error and repeat this step. You can save the panel without correcting the errors, but you cannot run it.

b. For information on the naming rules for panel names, see Appendix A.

Hints:

- To print the panel definition from the Panel definition panel, select **Print** from the Actions pull-down, or press the Print (F9) key.
- If you want to create an SQL query for the Panel search query at this time, return to the Main Selection menu, select **Commands** from the action bar, and then select **SQL Query** from the Commands pull-down, or press the SQL Query (Shift+F4) key.
- To edit an existing panel, select the panel in the Panels primary menu, select **Specify** and **Layout** from the actions bar in the Panel definition panel to make the changes, or press the Open (F6) key.
- All table fields and computed fields do not need to be marked.

Running a Panel

When a panel is run, you can select the **Actions**, **Exit**, or **Help** from the action bar. The Actions pull-down contains the panel actions for Add mode or Change mode, depending on which mode is active. **Exit** allows you to end the task and selecting Help provides information on using panels.

When you access a panel created under a previous version of Query Manager, the panel is automatically migrated to the Query Manager Version 1.2 format. See Appendix G for information on how the panel may change as a result of migration.

To Run a Panel

1. In the Main Selection menu, select **Panels**.
2. Select the name of the panel you want to run from the Panels primary menu.
3. Select **Actions** from the action bar and then select **Run** from the Actions pull-down, or press the Run (Shift+F1) key. The Run menu is displayed.
4. Select your choice for adding or changing data rows in the Run menu.
5. Type the information in the panel as required to do your work. Use the Actions pull-down as required.

6. In the Panels panel, select **Exit** from the action bar and select **Exit Panel** from the Exit pull-down, or press the Exit Panel (F3) key. The Panels primary menu is displayed.

   Note: If you select **Resume** from the Exit pull-down, the panel you were working from is displayed again and you can resume the task of adding or changing the data.

Hints:

- When the panel is running, you can scroll repeated rows by positioning the cursor on a repeated field within the repeated field area and pressing the cursor movement keys to move up and down the displayed rows of repeated fields.
- In order to display data in a field that has been defined as character (variable), which has a length greater than 254 bytes, position the cursor in the field and select **Show field** from the Actions pull-down (provided you have included **Show field** as a panel action), or press the Show Field (Shift+F7) key.

  As you type the data in the Show Field panel, keep in mind that the data is stored as one continuous character string in the table. The data prints based on the width of the field and edit code are specified in the panel.

  When you are revising data in a field that is longer than 254 characters, it may be necessary to retype the remaining text of the field from the revision point forward to ensure that the resulting data is what you want.
- To run a panel from the Command Line panel and have Query Manager prompt for the type of mode (Add or Change), select **Commands** from the action bar and then select **Command Line** from the Commands pull-down, or press the Command Line (Shift+F9) key. Type `RUN PANEL` and the panel name in the Command Line panel. You can also indicate whether to run the panel in Add or Change mode by typing the following, where *objectname* is the name of the panel:

  ```
  RUN PANEL objectname (MODE=ADD)
  ```

  or

  ```
  RUN PANEL objectname (MODE=CHANGE)
  ```
- You can also run the panel you are designing from the Panel definition panel by selecting **Actions** from the action bar and

selecting **Run** from the Actions pull-down,or pressing the Run (Shift+F1) key.

- A panel can be run from another menu, panel, or procedure.
- For Add mode, the subtable connecting column in the root-table-to-subtable connection is displayed as an output field even though it must have an Add usage. If it is a repeated field, all occurrences of the field in the repeated rows are output only. The value specified for the root table connecting column in the root-table-to-subtable connection is used for the subtable connecting column when the data is to be added to the table.
- The following table describes the results for summary functions and operations that can be used in computed field formulas in panels when the panel is run:

Function or Operation	**Results**
AVG	Null values are not included in the averaging process. If all values are null, then the result is null.
COUNT	Null values are not included in the count. If all values are null, then the result is null.
MIN	Null values are not included in determining the result. If all values are null, then the result is null. For example, if the three values are 7, 3, and null, the result of a MIN function would be 3.
MAX	Null values are not included in determining the result. If all values are null, then the result is null. For example, if the three values are 7, 3, and null, the result of a MAX function would be 7.
SUM	Null values are not included in the sum. If all values are null, the result is null.
+, -, *, /	If any of the values are null, the result is null.

Chapter 14. Defining Procedures

This chapter provides information and steps for writing procedures used with a customized interface.

Use this chapter in conjunction with the following chapters:

- Chapter 15
- Chapter 16.

Before You Begin

Writing procedures is one step in designing a customized interface. As the designer of a customized interface, you should be a user with SYSADM (system administrator) authority or a database administrator for the selected database. With either of these administrative authority levels, you have authorization to access and control the tables, views, and other Query Manager objects for the selected database.

Writing effective procedures calls for a full understanding of your database design as well as knowledge of Query Manager commands and panel operations. It is suggested that you perform a backup to your database before you run your procedure for the first time. Testing the procedure on sample data could prove helpful throughout the process of designing the customized interface.

About Procedures

A procedure can simply be a list of Query Manager commands or it can also contain procedure language statements. Every procedure must begin with a comment. Procedures allow you to invoke a sequence of operations using a single Query Manager command. Query Manager commands allow you to use and run previously defined Query Manager objects. For information on Query Manager commands, see the *OS/2 Command Reference.*

You can also define procedures to run panel operations. You can combine and automate several panel operations by defining procedures as the initial panel operation or to run as the *result* of another panel operation. The following panel operations can be run from a procedure:

- ADD and NEXT
- ADD and KEEP
- DELETE and NEXT
- CHANGE and NEXT
- NEXT
- PRINT
- COMPUTE
- SEARCH
- EXTENDED SEARCH
- QUIT PANEL.

See the *OS/2 Command Reference* for the syntax, example, and parameter descriptions for each of the panel operations previously listed. A simple procedure could run a query then print a report using a certain form as follows:

```
/* Example comment */
'run query qclerk (report=no'
'print report (form=salform)'
```

The comment must be enclosed between the /* and the */ symbols. The comment can be any text of your choosing. You can use the comment line, for example, to date the procedure or to display informational text as to the purpose of the procedure. When typing in the comment, be sure that you do not place a space between the / and the * symbols.

Some procedure statements can change the flow of processing, causing statements to be processed multiple times or to be bypassed. In the scenario described in Chapter 11, procedures are used to:

- Run the NATIONAL PAPER menu.
- Test a variable value and, based on that value, run a panel in Add or Change mode (STAFFCHG).
- Run a query and print a report (PRINTREPT).

The following terms are defined for your use as you read this chapter:

Procedure language

The procedure language is a structured programming language. A procedure uses both procedure language statements and Query Manager commands. Variables can be used in procedures also. The following are types of procedure language statements:

- DO WHILE
- DO UNTIL
- IF THEN ELSE.

Instructions

An instruction is a unit within a procedure that can be executed. An instruction can be a Query Manager command, panel operation, or a procedure language statement.

Instruction list

An instruction list is a sequence of one or more instructions, each of which is ended with a semicolon or by the end of the line on the screen.

Comments

Comments can be added to any line of a procedure. A comment begins with /* and ends with */. All characters between the beginning of the comment and the end of the comment are considered to be part of the comment. Comments can be continued across multiple lines. Every procedure must begin with a comment.

Variables

A variable name can consist of a string from 1 to 18 characters. Variable names must begin with an alphabetic character, a dollar sign ($), a pound sign (#), or an at sign (@). The rest of the variable names can be alphabetic or numeric (0 through 9), including the symbols dollar sign ($), a pound sign (#), an at sign (@), or an underscore (_). Variable names can be in mixed case. For information on the naming rules for variables, see Appendix A.

The value assigned to a variable can be a character string or a number. The value of a character string is case-sensitive; for example, YES is not equal to yes.

Procedure variables

Variables can be used within procedures to control the flow of the procedure or to allow for the substitution of variable values into instructions. You do not need to declare procedure variables; they are created automatically when needed and do not persist after the procedure ends. For more information on using variables within procedures, see Chapter 15.

Query Manager variables

Query Manager variables exist outside of a procedure. For more information on Query Manager variables, see Chapter 15. Also, see the GET and SET Query Manager commands in the *OS/2 Command Reference* for additional information on accessing Query Manager variables from a procedure.

Constants

Constants are character strings or numbers that do not change.

Character strings

Character strings must be enclosed in either single or double quotes; the quotes are not considered part of the character string. A character string can contain from 0 to 64 characters, with the exceptions of fields in panels, which can contain character strings longer than 64 characters. There are three methods for specifying character strings:

- If double quotes enclose a character string, then single quotes within the string are treated as part of the string, as follows:

  "John O'Hara" would be displayed as John O'Hara

- If single quotes enclose a character string, then double quotes within the string are treated as part of the string, as follows:

  `'"Department 10"'` would be displayed as `"Department 10"`

- If there are two consecutive occurrences of the enclosing quote within the string, they are treated as a single occurrence of the quote and are considered to be part of the string, as follows:

  `'John O''Hara'` would be displayed as `John O'Hara`
  `"""Department 10"""` would be displayed as `"Department 10"`

Character strings can also be constructed by concatenating a constant and a variable, as follows:

```
SAY "The value of A is now" ¦¦ A
```

Numbers

Numbers must not be enclosed in quotes and can be decimal numbers with no thousands separators. The decimal separator is assumed to be a period (.), the leading positive indicator an addition symbol (+), and the leading negative indicator a subtraction symbol (–). There is no trailing negative indicator. For numbers specified in scientific notation, both `E` and e are allowed. Examples of valid numbers are:

```
15
+6.9
-12345.6789
.0123
-1.234E+03
```

Query Manager commands

Within a procedure, any character string that is not part of a procedure language statement or part of a panel operation is considered to be a Query Manager command. The character string can be constructed from constants or through variable substitution. For more information on Query Manager commands, see Chapter 10.

Expressions

Expressions can be a single variable or constant or can be a formula for combining multiple variables or constants. Expressions can contain arithmetic, logical, string, or comparison operators. Arithmetic operators are add (+), subtract (-), multiply (*), and divide (/). Logical operators are And (&), Or (¦), Exclusive Or (&&), and Not (\). String expressions can be concatenated by using the concatenation string operator (¦¦). Comparison operators are equal to (=), less than (<), greater than (>), less than or equal to (<=), greater than or equal to (>=), and not equal to (<> or ><).

An expression can evaluate to a number, a character string, or a logical value of 0 (false) or 1 (true).

Expressions are not allowed within Query Manager commands or panel operations. If any of the operators previously noted are displayed in a Query Manager command or panel operation (other than within a quoted string), an error condition will result.

Note: The ¦ symbol, which is used for Or, and the ¦¦ symbol, which is used for concatenation, is the character produced by pressing the key engraved with the ¦ symbol. Depending on options in your CONFIG.SYS file, the symbol used may actually display as a solid vertical bar (| and ||).

Predicates A predicate is a test comparing two expressions to determine a true or false condition. The comparison operators are shown under Expressions. Note that if character strings are being compared, their case must match. For example, YES is not equal to yes.

Authorization

When you write a procedure, you are considered the *owner* of the procedure. When you are finished writing a procedure you can select to exit and save the procedure. A Save panel that prompts you to type a name and comment and make a **Share** selection is then displayed. As owner of the procedure, you can choose to grant or revoke other users the authority to access and use the procedure by setting **Share** to **Yes** or **No** in the Save panel.

If the procedure is saved with **Share** set to **Yes**, any user who can access the database can *select* the procedure. When the procedure is saved with **Share** set to **No**, then only the owner, a user with SYSADM (system administrator) authority, or a database administrator for the selected database can select the procedure.

Writing a Procedure

You must type the procedure language statements and Query Manager commands using the correct syntax. Procedure language statement syntax and Query Manager command syntax are described in the *OS/2 Command Reference*.

Once a procedure is defined, you can select **Actions** from the action bar to run, save, get, reset, or print a procedure.

As you specify a procedure, keep the following considerations in mind:

- Procedure language statements and Query Manager commands can be typed in mixed case. Character strings should be typed using the case (including mixed case) that you want to use to display, print, or process the string. For example, the character string YES is not equal to the character string yes.

- Use single quotes (' ') or double quotes (" ") to surround Query Manager commands or character strings. The quotes are removed from the command before it is executed and from the character string before it is assigned as a constant or passed on to be executed as commands or panel operations. For example:

```
'SET GLOBAL (ABC = "Hi")'
'print report (form=commform)'
SAY "Department Commission Report was printed successfully"
say 'The report "Department Commission" was printed successfully'
```

- If you are using a variable within the procedure or panel operation to be passed from a procedure, you should leave the variable outside of any quotes, so the variable is properly substituted. For example, if you want to print 10 reports using forms FORM1 through FORM10, the procedure would include the following:

```
DO I=1 to 10
  'print report (form=form'¦¦I
END
```

- There are two separate variable pools to consider when writing a procedure: the procedure variable pool and the Query Manager variable pool. The procedure variable pool is only known within the procedure itself. The Query Manager variable pool is separate from the procedure pool and values must be retrieved from it into the procedure variable pool in order to use the values in the procedure. For example, in the STAFFCHG procedure that is used in the scenario when **Change Staff** is selected from the NATIONAL PAPER menu, the user types a value for the Query Manager variable *n*, and that value is put into the procedure variable NAME, as follows:

```
'GET CURRENT (NAME=n'  /*Put the value just typed into NAME  */
IF NAME = 'none'       /*If the prompt value none was not typed
                         over, you want to add data to STAFF */
  THEN 'RUN PANEL ASTAFF (MODE=ADD)'
  ELSE 'RUN PANEL ASTAFF (MODE=CHANGE)' /*If a NAME is typed, you
                                          want to change data
                                          based on the NAME  */
```

For more information on variables, see Chapter 15.

- When you write a procedure, use the following rules to split the procedure across lines:
  - Comments can be split across lines.
  - Procedure instructions can only be split where the instruction ends or where the instruction syntax allows splitting.
  - Query Manager commands and panel operations must be contained on one line of the procedure.
  - Any character string must be contained on one line.

  For more information, see the *OS/2 Command Reference*.

- You can place multiple Query Manager commands and procedure language statements on one line by using a semicolon (;) to separate the commands or statements. The last statement, panel operation, or command on the line does not need to be followed by a semicolon; for example:

  ```
  DO MONTH = 1 to 12; 'RUN PROC SUMMONTH'; END
  ```

- Procedures must begin with a comment and comments can be used to document a procedure. Comments must be enclosed by the /* and */ symbols. When typing in the comment, be sure that you do not place a space between the / and the * symbols; for example:

  ```
  /* This is the beginning comment */
  'get global (tl = time'        /* Access the global variable
                                    to get the system time */
  ```

- Procedures can test the value of a return code received from a Query Manager command and panel operation. Every Query Manager command sets the variable RC (for return code) to a numeric value when processing of the command completes; a zero represents a successful completion. A non-zero return code means that processing was not successful. A procedure can test return codes to determine whether to continue processing the rest of the procedure, exit the procedure, or take a special action to correct a problem.

- Query Manager commands and panel operations must be contained on one line of the procedure.

- When you exit the procedure, the EXIT statement can be used to set RC to a value. If no value is specified, or if there is not an EXIT statement specified, RC is set to 0.

- For information on using variables with procedures, see Chapter 15.

To Write a Procedure

1. In the Main Selection menu, select **Procedures.**
2. Select **—NEW—** from the Procedures primary menu.
3. Select **Actions** from the action bar and then select **Open** from the Actions pull-down, or press the Open (F6) key.
4. In the Procedure panel, type the procedure.

   Notes:

   a. Remember, a procedure must begin with a comment (enclosed in /* and */).

   b. Press the Enter key to go to the next line and continue typing. The Enter key works as a line-advance key.

   c. If you need more information on the available editing keys in procedures, select Help and then select Keys.
5. Select **Exit** from the action bar and then select **Exit Procedure** from the Exit pull-down, or press the Exit Procedure (F3) key. The confirmation message is displayed.
6. Select **Save and exit** from the confirmation message. The Save panel is displayed.

   Notes:

   a. If you select **Exit without saving**, the procedure will not be saved and the Procedures primary menu is displayed again.

   b. If you select **Resume**, the procedure definition menu is displayed again and you can continue editing the procedure.
7. Type the procedure name and an optional comment in the Save panel, set **Share** to either **Yes** or **No**, and then select Enter.

   Note: When you select **Save and exit** from the confirmation message, or when you select **Actions** and select **Save** from the Actions pull-down, the procedure is verified by Query Manager. You may receive an error message that indicates an error within the procedure syntax. Correct the error from within the Procedure panel and then select one of these items again. You can save a procedure that contains errors, but you cannot run it until all the errors are removed.

Hints:

- To print the procedure definition from the Procedure panel, select **Print** from the Actions pull-down, or press the Print (F9) key.
- To clear the Procedure panel, select **Refresh** item from the Actions pull-down, or press the Refresh (F5) key.
- To retrieve a copy of an existing procedure into the Procedure panel, select **Get** from the Actions pull-down, or press the Get (F2) key.

Running a Procedure

The user can run procedures from within Query Manager, from the Group – Main window, from the OS/2 command prompt, or through the Query Manager Callable Interface. Procedures can be run during either interactive or customized processing.

Interactive processing is used when the procedure is run from within Query Manager, either from the Command Line panel, as a menu action, or as a panel action from within another procedure, or from the **Run** action from the Procedures primary menu or Procedure panel. The user can interact with Query Manager panels and menus, as well as any customized panels and menus the procedure calls while running, using interactive processing.

Customized processing is used when the procedure is specified as a parameter in the command to start Query Manager. This allows the user to bypass the Query Manager interface completely and only see the menus and panels that you have defined. You can personalize your Query Manager start-up and indicate a procedure to run using the Properties panel or you can specify the procedure as a parameter in the command to start Query Manager that the user types from the OS/2 command prompt. For more information on these two ways to start Query Manager, see Appendix B. You should specify a database name as a parameter, or be sure one is specified in the profile.

When the user runs a procedure, the following can occur:

- Database Manager will check that the user ID running the procedure has authority to access or perform queries and panel operations on any of the specified objects, such as tables and views.

- When messages are received during either interactive processing or customized processing, they are displayed in a panel. If the message requires user action, the next statement in the procedure is not processed until the message is acknowledged by the user.

- If a Query Manager variable is encountered with no value assigned, a panel is displayed prompting for a value. The variable name is the prompt text.

- If the procedure processes a RUN QUERY and the REPORT keyword is set to YES (the default), the formatted report is displayed. The user can scroll forward through the report, then when the user exits the report, the procedure processing continues. If REPORT is set to NO, the query is run but the report is not displayed; however, the report is available for printing or saving. For more information on the Query Manager RUN command, see the *OS/2 Command Reference*.

- If the procedure processes a RUN MENU or a RUN PANEL command, the menus or panels defined for the customized application are presented. Once the user completes the menus or panels, the procedure continues.

To Run a Procedure Interactively

1. In the Main Selection menu, select **Procedures**.

2. Select the name of the procedure you want to run from the Procedures primary menu.

3. Select **Actions** from the action bar and then select **Run** from the Actions pull-down, or press the Run (Shift+F1) key.

4. Respond to any prompts, menus, panels, or messages that the procedure processes.

5. In the Procedures menu, select **Exit** from the action bar and then select **Exit Procedures** from the Exit pull-down, or press the Exit Procedures (F3) key.

Hints:

- To run a procedure from the Command Line panel, select **Command Line** from the Commands pull-down from the Main Selection menu or the primary menus for objects, or press the Command Line (Shift+F9) key. Type `RUN PROC` and the procedure name in the Command Line panel.
- You can also run a procedure from the Procedure panel by selecting **Actions** from the action bar and then selecting **Run** from the Actions pull-down, or by pressing the Run (Shift+F1) key.

If you want to run a procedure using the Group – Main window or from the OS/2 command prompt, see Appendix B for more information.

Hints:

- You can also create a .CMD file to run a procedure. For information on .CMD files, see the *OS/2 Command Reference*.
- Since procedure names are prefixed with a qualifier, be sure the qualifier of the procedure you want to run matches the qualifier you are currently using or have specified.

Building the Customized Interface

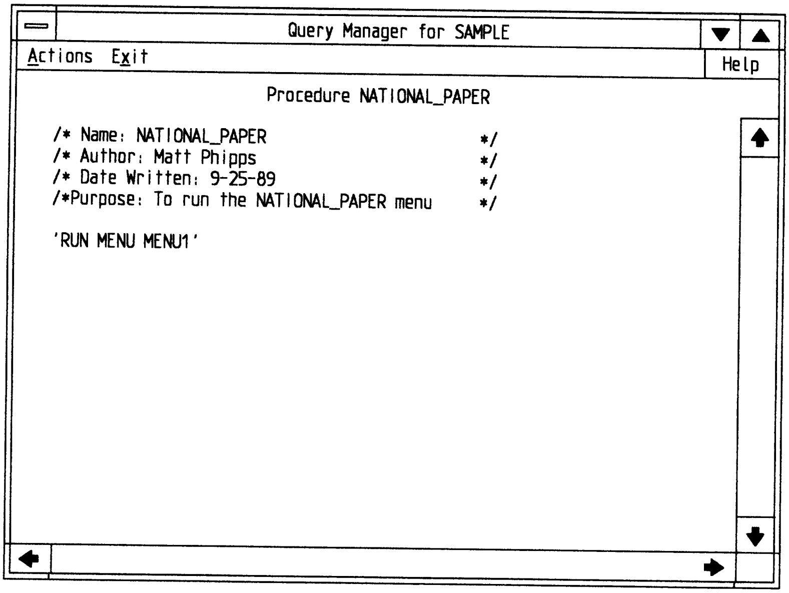
This section shows the Procedure panel as it is displayed after each procedure for the scenario is defined. These procedures are named NATIONAL_PAPER, STAFFCHG, and PRINTREPT.

To run this customized scenario, type the following command from the OS/2 command prompt:

```
QUERYMGR /RUNPROC:national_paper /DAT:sample
```

Procedure Panel, After NATIONAL_PAPER Is Saved

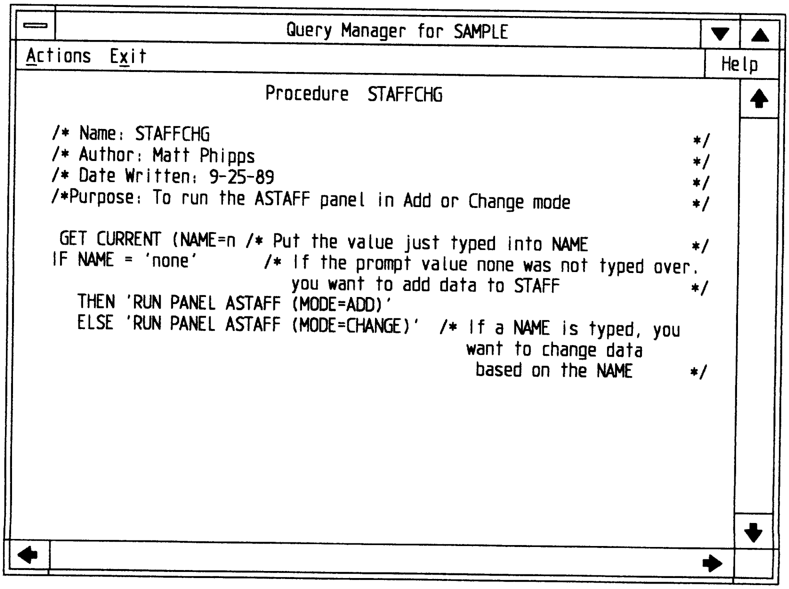
The following illustration shows the Procedure panel as it would be displayed after you have saved the procedure NATIONAL_PAPER.



```
/* Name: NATIONAL_PAPER                          */
/* Author: Matt Phipps                           */
/* Date Written: 9-25-89                         */
/*Purpose: To run the NATIONAL_PAPER menu        */

'RUN MENU MENU1'
```

Procedure Panel, After STAFFCHG Is Saved

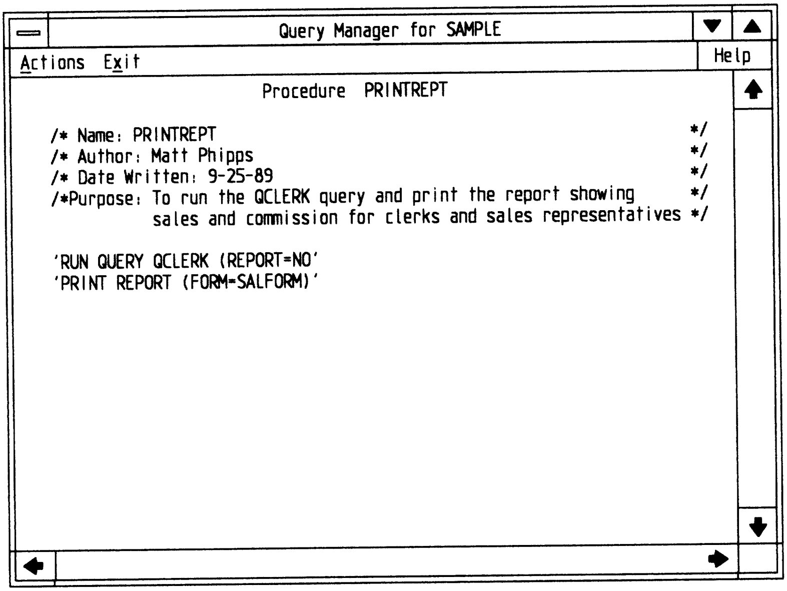
The following shows the Procedure panel as it would be displayed after you have saved the procedure STAFFCHG.



```
/* Name: STAFFCHG                                                  */
/* Author: Matt Phipps                                             */
/* Date Written: 9-25-89                                           */
/*Purpose: To run the ASTAFF panel in Add or Change mode           */

 GET CURRENT (NAME=n /* Put the value just typed into NAME         */
IF NAME = 'none'       /* If the prompt value none was not typed over,
                          you want to add data to STAFF            */
   THEN 'RUN PANEL ASTAFF (MODE=ADD)'
   ELSE 'RUN PANEL ASTAFF (MODE=CHANGE)'  /* If a NAME is typed, you
                                             want to change data
                                              based on the NAME    */
```

Procedure Panel, After PRINTREPT Is Saved

The following shows the Procedure panel as it would be displayed after you have saved the procedure PRINTREPT.



```
/* Name: PRINTREPT                                                  */
/* Author: Matt Phipps                                              */
/* Date Written: 9-25-89                                            */
/*Purpose: To run the QCLERK query and print the report showing     */
           sales and commission for clerks and sales representatives */

'RUN QUERY QCLERK (REPORT=NO'
'PRINT REPORT (FORM=SALFORM)'
```

Chapter 15. Using Query Manager Variables in Customization Tasks

This chapter provides information about using variables with the menus, panels, and procedures of a customized interface.

Use this chapter in conjunction with the following chapter:

Chapter 16.

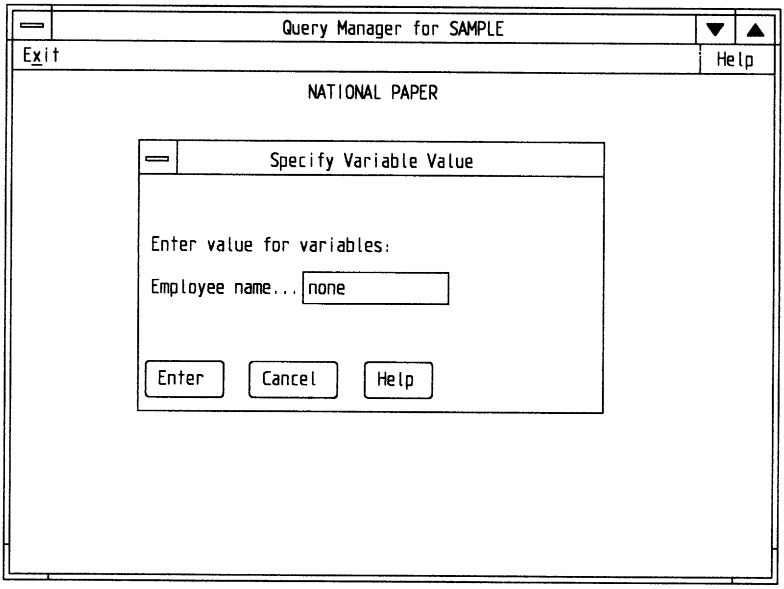
About Variables in Customization Tasks

You can use variables within Query Manager objects. A variable is a named entity that can be assigned a value. Remember, Query Manager objects include panels, procedures, menus, queries, and forms. Procedure variables can be used within procedures to control the flow of the procedure or to allow for the substitution of variable values into instructions. You do not need to declare procedure variables; they are created automatically when needed and do not exist after the procedure ends. When procedure variables are created, they are assigned a value, if specified, or set to a character string (equal to the variable name) if a value is not specified.

Query Manager variables exist outside of a procedure. Query Manager variables can also be used within procedures to control the flow of the procedure or to allow for the substitution of variable values into instructions by using the Query Manager GET command.

In the scenario in Chapter 11, a variable was defined for the menu item **Change Staff**: n for NAME. This variable was defined so that the user could select the row or rows in the STAFF table to change based on a value for NAME. If users type a name, they are identifying the employee record they want to change; if users accept the default prompt value of none, they want to add data to the table.

When users select **Change Staff**, they are prompted for a value for name as follows:



Enter value for variables:

Employee name... none

The variable *n* is part of the definition of the query used for the panel search query when the panel is run. The variable *n* is used in the definition of the NATIONAL PAPER menu, as well as in the query Q1. When a variable is used in the definition of an object, the variable serves as a placeholder identifying the point in the definition where a value is to be substituted when the object is processed.

Variables have a *life-span*. There are two basic types of life-spans for variables:

- *Local* variables exist until the object that created them ends. During this time, the local variable is available to any other Query Manager object that is run.
- *Global* variables exist from the time they are initially created until Query Manager ends.

Creating Variables

Variables are created when they are first referenced or when they are set to a value; they are not explicitly declared. Use the Query Manager SET GLOBAL command to create global variables. (For information on Query Manager commands, see Chapter 10.)

Local variables are created in one of the following ways:

- Use the SET LOCAL or SET CURRENT commands from within a procedure, the Query Manager Command Line, or as an action from a customized menu or panel to create a variable if one does not already exist.
- Run a customized panel. All table fields defined for a panel are local variables.
- Run a customized panel that has a computed field. Computed fields are local variables.
- Run a customized menu that has a variable defined as the response to the selection of an item in the menu. Type a value for the variable when prompted.
- If the user runs a customized menu, panel, query, or form that contains a reference to a variable that has not been defined, the user is prompted for a value.

Referencing Variables

When a variable name is in a character string (as in a title), the variable name must be preceded by an ampersand (&) so that Query Manager recognizes it as a variable name. An ampersand must precede the variable name in the following cases:

- In panels:
  - In a Query Manager command that is a panel action
  - In the title field.
- In a Query Manager command that is a menu action
- In menus in a variable prompt value
- In a report form specification for heading, footing, break, or final text (usually the Date, Time, and Timestamp variables)
- In SQL query. (For information on SQL query, see Chapter 9.)

An ampersand (&) is *not* required to reference a variable in the following cases:

- In panels, in an expression in a computed field
- In procedures using the Query Manager GET command
- In the Command Line panel using the GET command.

Except for the Query Manager GET LOCAL and GET GLOBAL commands, when a variable reference is encountered it is searched when using the GET CURRENT command; the most recent setting of the variable is used. If a local and a global variable by the same name exist, the value from the local variable is used.

Naming and Defining Variables

Variables have a variety of characteristics, as follows:

Variable name The variable name must begin with an alphabetic character, pound sign (#), at sign (@), or dollar sign ($). Subsequent characters can be alphabetic or numeric (0 through 9). Special characters allowed are the pound sign (#), at sign (@), dollar sign ($), or underscore (_). Variable names can be typed in mixed case, but the variable name is always displayed in uppercase. For more information on naming rules for variables, see Appendix A.

Variable values The value assigned to a variable can be a number or a character string. Character strings can contain from 0 to 64 characters, with the exception of fields in panels, which can contain character strings longer that 64 characters. A character string is identified by enclosing it in either single or double quotes. A number is not enclosed in quotes, but can be a decimal number with no thousands separators, or can be in scientific notation. The decimal separator is assumed to be a period (.), the leading positive indicator is a plus (+), the leading negative indicator is a minus (−), and there is no trailing negative indicator.

The following are examples of variables:

Variable Specified	**Is Displayed As**
Name="John O'Hara"	John O'Hara
NAME='"Department 10"'	"Department 10"
NaMe='John O''Hara'	John O'Hara
nAmE="'"Department 10"'"	"Department 10"
Number=15	15
NUMBER=+6.9	+6.9
NUmBEr=−12345.6789	−12345.6789
nUMbER=.0123	.0123

Typing Variable Values

When you are prompted to type a variable value, while running either a menu or panel, the following rules apply:

- For character values, quotes are not required. Any quotes that are typed are considered to be part of the value.
- For numeric values, type the value using − and + as the negative and positive indicators, and . as the decimal point indicator. This is based on the Ln edit code as described in "Column Edit Rules" on page 4-16. The n is assumed to be 15 digits, so you can type up to 15 digits after the decimal separator. The decimal separator and leading and trailing negative indicators used are determined from the profile. For information on profiles, see "Using Profiles" on page 8-35.

Using System Variables

Query Manager has nine predefined variables that can be used at any time. For example, you can use the DATE system variable as a variable in a report form or in an SQL statement. You cannot update these system variables using the Query Manager SET GLOBAL command, but you can use the GET GLOBAL or GET CURRENT command to get them. The predefined system variables are as follows:

PROFILE Contains the profile name currently active.

DATABASE Contains the name of the database to which you are connected.

DATE Contains the current date formatted according to the TDY- edit code; for example, 1988-08-29.

TIME Contains the current time formatted according to the TTS. edit code; for example, 16.42.35.

TIMESTAMP Contains the current time stamp formatted according to the TSI edit code; for example, 1988-08-29-16.42.19.123456.

SYSLEVEL Contains a release identification to enable programs to test the environment in which they are running.

SQLUSER Contains your own user ID used to log on to your workstation.

SQLLIST Contains the current active qualifier for lists. This value is set one of three ways: from the OS/2 command prompt or the Group-Main window, using the QUERYMGR EXE file, from the Query Manager profile, whenever you start Query Manager, or from the user ID used to log on to your workstation. It can be changed by selecting the **Change qualifier for lists** item. The value for SQLLIST will reset every time the database is opened.

RC Contains the return code from the last object processed.

Note: For information on DATE, TIME, and TIMESTAMP, see the information on edit codes for these data types in Appendix I.

Relating the Procedure Variable Pool to the Query Manager Variable Pool

When you use variables within objects, there are two variable pools to consider: the Query Manager variable pool and the procedure variable pool. The procedure variable pool is only known within the procedure itself and is not shared with other procedures. The Query Manager variable pool is separate from a procedure variable pool and values must be retrieved from it into a procedure variable pool in order to use the values for substitution within a procedure.

Each time a Query Manager object is run, a new level in the Query Manager variable pool is created to contain Query Manager variables created by the object. These variables exist until the object completes running and then the variables are discarded. When an object is run, and another object is nested within that object (for example, a procedure is run as the action from a panel), the nested object has access to all of the Query Manager variables known by the calling object. When a variable is referenced with a GET CURRENT command, Query Manager searches for the named variable beginning with the variables allocated by the object, then the variables allocated by the object that called this object, then the variables allocated by the object that called that object, and so on. If no local variable is found by this search, the global variables are searched for the named variable. If the variable is still not found, the user is prompted for a value.

When an attempt is made to update a variable with a SET CURRENT command, the same search for the named variable is made as previously described, except the Global variables are not searched since they can only be updated using the SET GLOBAL command. If the variable is not found, a new local variable is allocated belonging to the object being run.

In order to provide an interface from a procedure to the Query Manager variable pool, the Query Manager SET and GET commands are provided. These commands are described in the *OS/2 Command Reference*.

The following two procedures and the accompanying diagram illustrate the relationship of the procedure variable pool to the Query Manager variable pool. Notice that the procedure variable pool is clear when P1 is started.

P1 consists of the following:

```
/*    */
'SET GLOBAL (EVAR=20)'   /* Sets the QM global variable EVAR to 20  */
V1=5                     /* Sets the proc variable V1 to 5          */
'SET LOCAL (EVAR=10)'    /* Sets the QM local variable EVAR to 10   */
'RUN PROC P2'            /* Runs the procedure P2                   */
P2RC=RC                  /* Sets the proc variable P2RC to 30       */
SAY 'P1 VAR V1=' ¦¦ V1
```

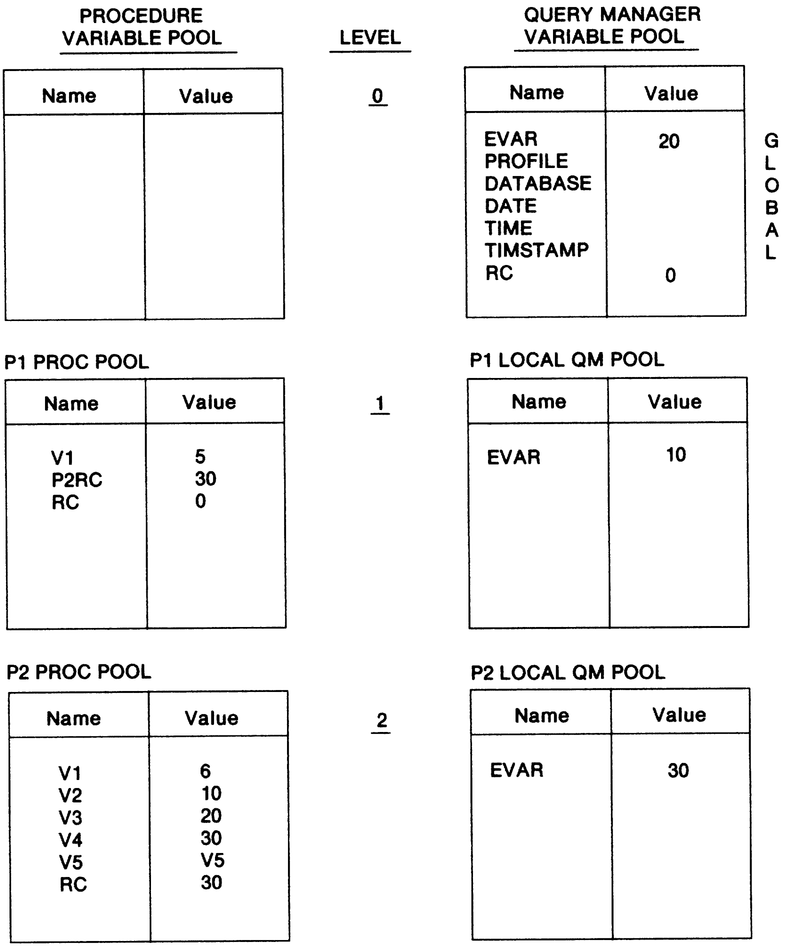
P2 consists of the following:

```
/*    */
V1=6                     /* Sets the proc variable V1 to 6          */
SAY 'P2 VAR V1=' ¦¦ V1
'GET CURRENT (V2=EVAR)'  /* Places the current value (10) of the QM
                            local variable EVAR in procedure
                            variable V2                             */
SAY 'P2 VAR V2=' ¦¦ V2
'GET GLOBAL (V3=EVAR)'   /* Places the current value (20) of the QM
                            global variable EVAR in procedure
                            variable V3                             */
SAY 'P2 VAR V3=' ¦¦ V3
'SET LOCAL (EVAR=30)'    /* Sets the QM local variable EVAR to 30   */
'GET CURRENT (V4=EVAR)'  /* Places the value (30) of the QM local
                            variable EVAR in procedure variable V4  */
SAY 'P2 VAR V4=' ¦¦ V4
SAY 'P2 VAR V5=' ¦¦ V5
EXIT (V4)                /* Sets the QM variable RC to 30           */
```

When P1 is run (and consequently P2, since it is nested within P1), the following data is presented:

```
P2 VAR V1=6
P2 VAR V2=10
P2 VAR V3=20
P2 VAR V4=30
P2 VAR V5=V5
P1 VAR V1=5
```

The following diagram shows the values that are present in the Query Manager global variable pool, the P1 procedure pool, the Query Manager local variable pool, the P2 procedure pool, and the Query Manager local variable pool:



PROCEDURE VARIABLE POOL

LEVEL

QUERY MANAGER VARIABLE POOL

0

Name	Value

Name	Value
EVAR	20
PROFILE	
DATABASE	
DATE	
TIME	
TIMSTAMP	
RC	0

P1 PROC POOL

Name	Value
V1	5
P2RC	30
RC	0

1

P1 LOCAL QM POOL

Name	Value
EVAR	10

P2 PROC POOL

Name	Value
V1	6
V2	10
V3	20
V4	30
V5	V5
RC	30

2

P2 LOCAL QM POOL

Name	Value
EVAR	30

Customizing Variables

The following two scenarios are examples of how variables can be used within a customized interface. Use these scenarios to aid you in designing your own customized interfaces.

In the first scenario, you want to be able to retrieve the information on sales for a specific department and generate the appropriate report. First, you must create a query to access the data. Since the same query can be used for each department, you do not place a specific department number in the query; instead, you define a variable, DEPARTMENT, in the position where the department number belongs in the query. The SQL SELECT statement would be:

```
SELECT NAME, JOB, SALARY
FROM STAFF
WHERE DEPT=&DEPARTMENT
```

The variable DEPARTMENT is not created when you define the query; the variable is part of the query definition. When you run the query, &DEPARTMENT is replaced by a value. There are several ways to replace &DEPARTMENT with a value, depending on your needs. A few of the choices are:

- Run the query directly and Query Manager would prompt you to type a value for DEPARTMENT.
- Build a procedure that sets DEPARTMENT and then runs the query. DEPARTMENT would be set using some form of the Query Manager SET command.
- Use the SET command from the Command Line panel to create a variable named DEPARTMENT and then run the query.
- Create a menu with an action that causes a prompt for DEPARTMENT to be displayed before running the query.

In the second scenario, you want to update rows in a table based on inventory changes. You decide to create a panel, but you need to determine the search condition to use to access the rows from the table. You name a query for the initial panel search and have the variable &ITEM in the position where the item number should be displayed in the query. As in the previous example, the variable is not created, but is part of the query definition.

The SQL SELECT statement for the query follows:

```
SELECT *
FROM INVENTORY
WHERE PARTNO=&ITEM
```

To assign a value to &ITEM, you have many different options:

- Run the panel directly and Query Manager would prompt you to type a value for ITEM.
- Define a procedure that sets ITEM and then runs the panel. Again, ITEM should be set using the SET command.
- Use the SET command from the Command Line panel to create a variable named ITEM and then run the panel.
- Create a menu with an action that causes a prompt for ITEM to be displayed before running the query.

By understanding how variables can be used to provide a connection between the various objects in Query Manager, you can customize your own interface for working with Database Manager databases.

Chapter 16. Running the Customized Interface

This chapter illustrates the flow of the customized interface for the scenario described in Chapter 11.

About the Finished Customized Interface

Rules and examples for customizing an interface using panels, procedures, menus, queries, and forms were discussed in Chapter 11 through this chapter. All these objects can be used together to create a powerful customized interface for performing your work. The scenario for the National Paper company described in Chapter 11 was one example of how these objects can be used together. The following diagrams illustrate the flow of the customized interface for the National Paper company.

Running the NATIONAL PAPER Procedure

The user runs the NATIONAL PAPER procedure from either the OS/2 command prompt or from the Group – Main window. For example, to run the NATIONAL PAPER procedure from the OS/2 command prompt, the user would type the following:

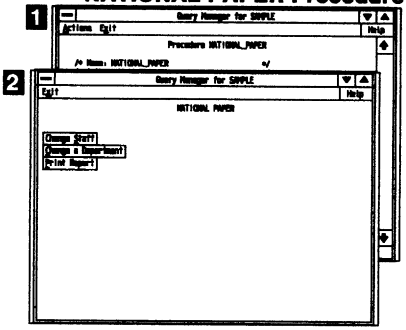
```
QUERYMGR /RUNPROC:national_paper /DAT:sample
```

Running the NATIONAL PAPER Menu

The NATIONAL PAPER procedure:

- Contains the Query Manager RUN command to run the NATIONAL PAPER menu
- Presents the NATIONAL PAPER menu to the user.

NATIONAL PAPER Procedure



```
/* Name: NATIONAL_PAPER                    */
```

NATIONAL PAPER
Menu

Selecting Change Staff from the NATIONAL PAPER Menu

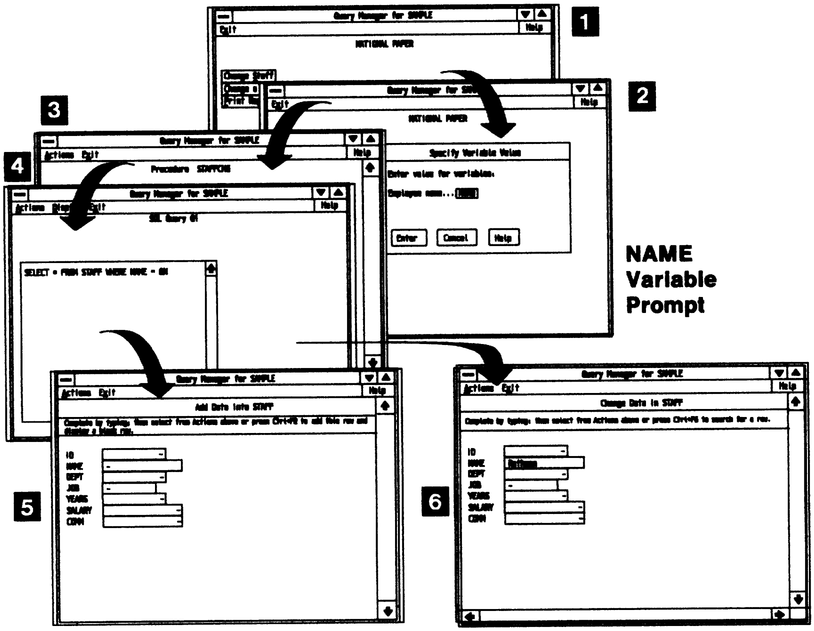
The **Change Staff** menu item:

- Is the first item on the NATIONAL PAPER menu.
- Presents the Specify Variable Value panel so users can indicate the employee for which they want to change data, or so users can add employees.
- Causes the STAFFCHG procedure to be run. This procedure takes the data from the Specify Variable Value panel and runs the ASTAFF panel in either Add or Change mode. If the user types an employee name, the mode is Change. If the user does not type a name (selects Enter with none in the entry field), the mode is Add.

  If the user types a name in the Specify Variable Value panel, the name typed is passed to the query Q1, which is the initial panel search for the panel. The query retrieves the row for that employee name from the STAFF table.

- Presents the ASTAFF panel for adding or changing data in the STAFF table. This panel is the default definition panel for the STAFF table.

STAFFCHG Procedure



Enter value for variables.

Employee name...

SELECT * FROM STAFF WHERE NAME = &N

Complete by typing; then select from Actions above or press Ctrl+F6 to add this row and display a blank row.

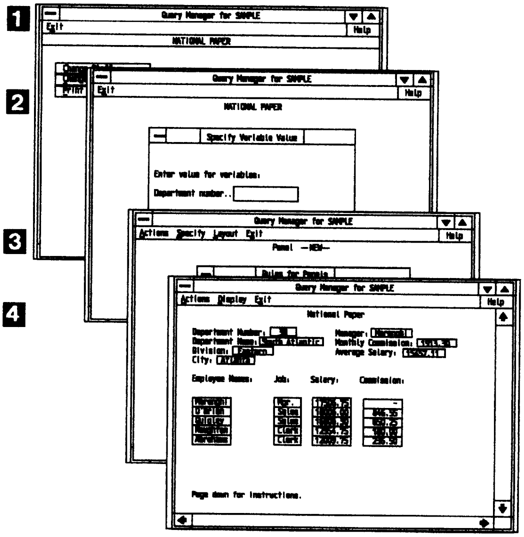
Complete by typing; then select from Actions above or press Ctrl+F6 to search for a row.

ASTAFF Panel

Selecting Change a Department from the NATIONAL PAPER Menu

The **Change a department** menu item:

- Is the second item on the NATIONAL PAPER menu.
- Presents the Specify Variable Value panel so users can indicate the department number in which they want to browse or change data.
- Passes the value indicated in the Specify Variable Value panel to the query Q2 before the panel is displayed so the user can change data. Query Q2 is the initial panel search for the panel. Rows for that department number from both the STAFF and ORG tables are retrieved.
- Presents the DEPT panel for changing or reviewing data from the STAFF and ORG tables. The only data that can be changed is the salary and commission.



Enter value for variables:

Department number..

Department Number:
Department Name:
Division:
City:
Manager:
Monthly Commission:
Average Salary:

Employee Names: Job: Salary: Commission:

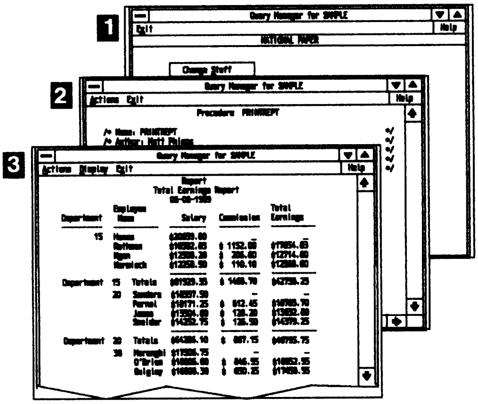
Page down for instructions.

DEPT Panel

Selecting Print Report from the NATIONAL PAPER Menu

The **Print report** menu item:

- Is the third item on the NATIONAL PAPER menu.
- Runs the PRINTREPT procedure, which in turn runs the QCLERK query and generates a report using the SALFORM form.
- Prints the EARNINGS FOR 1989 report, which shows the name, department, and salary plus commission for all employees of the National Paper company.



```
/* Name: PRINTREPT                */
/* Author: Matt Phelps            */
```

```
                          Report
                  Total Earnings Report
                        06-08-1989

             Employee                             Total
Department   Name        Salary     Commission    Earnings
----------   --------   ---------   ----------   ---------
        15   Hanes      $20659.80        -           -
             Rothman    $16502.83   $ 1152.00    $17654.83
             Ngan       $12508.20   $  206.60    $12714.80
             Kermisch   $12258.50   $  110.10    $12368.60
                        ---------   ----------   ---------
Department   15  Totals $61929.33   $ 1468.70    $42738.23
             20  Sanders $18357.50       -           -
                 Pernal  $18171.25  $  612.45    $18783.70
                 James   $13504.60  $  128.20    $13632.80
                 Sneider $14252.75  $  126.50    $14379.25
                        ---------   ----------   ---------
Department   20  Totals $64286.10   $  867.15    $46795.75
             38  Marenghi $17506.75      -           -
                 O'Brien  $18006.00  $  846.55   $18852.55
                 Quigley  $16808.30  $  650.25   $17458.55
```

Running the Customized Interface **16-5**

Appendix A. Database Manager Naming Rules

The naming rules described in this appendix should be used when you provide a name for a database, an object, or a qualifier. SQL reserved words cannot be used as the name of a table, view, column, index, or qualifier. A list of SQL reserved words is also provided.

Note: Database Manager does not support the long file names used by High Performance File System (HPFS).

Naming Rules for Databases, Objects, and Qualifiers

Every time a new database is created, Database Services creates a separate OS/2 directory to store all the database objects associated with that particular database. The naming scheme used by Database Services is of the form –SQL00001 through –SQLnnnnn, where –SQL00001 contains objects associated with the first database created and –SQLnnnnn contains objects for the last database created. Database Services maintains these directories automatically. In the case where databases have been created and dropped, Database Services can reuse these directories. Therefore, in order to avoid potential problems, you should *not*:

- Create other OS/2 directories that use the same naming scheme
- Manipulate directories that have already been created by Database Services.

Database Names

For database names, the name you specify:

- Can contain 1 to 8 characters
- Must begin with a letter (A to Z or a to z), *at* symbol (@), number symbol (#), or dollar symbol ($)

- After the first character, can include the following as part of the last 7 characters:
  - Letters (A to Z or a to z)
  - Digits (0 to 9)
  - *At* symbol (@), number symbol (#), dollar symbol ($), or underscore (_).
- Although typed in lowercase, it is displayed in uppercase.

Hints:

- To avoid potential problems, you should avoid using the characters @, #, and $ in a database name if you intend to use the database in a communications environment. Also, since these characters are not common to all keyboards, they should be avoided if you plan to use the database in another country.
- Database names are not case-sensitive (for example, `CINDY` = `cindy` = `Cindy` = `ciNdy`), but the names are converted to uppercase when typed.

Object and Variable Names

Objects include the following:

- Database Manager (Database Services) objects (also used by Query Manager):
  - Tables
  - Views
  - Columns
  - Indexes.
- Query Manager objects:
  - Queries
  - Forms
  - Menus
  - Panels
  - Procedures.

For object and variable names, the name you specify:

- Can contain 1 to 18 characters.

- Must begin with a letter (A to Z, a to z, or any valid accented letter), *at* symbol (@), number symbol (#), or dollar symbol ($).
- After the first character, can include the following as part of the last 17 characters:
  - Letters (A to Z, a to z, or a valid accented letter such as ö)
  - Digits (0 to 9)
  - *At* symbol (@), number symbol (#), dollar symbol ($), or underscore (_).
- For Database Manager objects (tables, views, columns, and indexes), cannot be any of the SQL reserved words (such as SELECT, FROM, and so on). The SQL reserved words are listed in "SQL Reserved Words."
- Although typed in lowercase, is displayed in uppercase.

Hints:

- Object names are not case-sensitive (for example, `FAYMILLS` = `faymills` = `FayMills`), but the names are converted to uppercase when typed.
- Variable names can also be required to have a leading ampersand (&). See Chapter 15 for more information on naming and referencing variables.

User ID

For user IDs, the name you specify:

- Can contain 1 to 8 characters
- Can include the following:
  - Letters A to Z or a to z (lowercase are folded to uppercase)
  - Digits (0 to 9)
  - @ or # or $.
- Cannot start with any of the following (in uppercase, lowercase, or mixed case):
  - SYS
  - IBM
  - SQL.

- Cannot be the character string PUBLIC, USERS, ADMINS, or GUESTS (in uppercase, lowercase, or mixed case)
- Cannot end with $.

Hint:

If you are not logged on, Database Manager will prompt you to log on.

Passwords

For passwords, the word or name you specify:

- Can contain 4 to 8 characters
- Can include the following:
  - Letters A to Z or a to z (lowercase are folded to uppercase)
  - Digits (0 to 9)
  - @ or # or $.

Qualifiers for Lists

For qualifiers for lists, the name you specify:

- Can contain 1 to 8 characters.
- Must begin with a letter (A to Z, a to z, or any valid accented character such as ö), *at* symbol (@), number symbol (#), or dollar symbol ($).
- After the first character, can include the following as part of the last 7 characters:
  - Letters (A to Z, a to z, or any valid accented letter)
  - Digits (0 to 9)
  - *At* symbol (@), number symbol (#), dollar symbol ($), or underscore (_).
- Cannot be the character string SYSIBM (in uppercase, lowercase, or mixed case) if you intend to create or update a table, index, or view.
- Should not be the character string PUBLIC (in uppercase, lowercase, or mixed case).

- Should not be any of the SQL reserved words (such as SELECT, FROM, and so on). The SQL reserved words are listed in "SQL Reserved Words" on page A-5.

Hints:

- Qualifiers for lists are not case-sensitive (for example, AUBREY = aubrey = Aubrey = AuBrey), but the IDs are converted to uppercase when typed.
- Qualifiers are used to name and display groups of objects under a particular category or user ID.

SQL Reserved Words

In Database Manager, SQL reserved words cannot be used as names of tables, views, columns, indexes, statements, cursors, or authorization IDs. On the following pages are two lists:

- The first list identifies the current list of SQL reserved words. When you are creating a table, Database Manager verifies that these words are reserved and does not allow you to create a table or column with these names.
- The second list shows words that are not reserved words in Database Manager now, but may possibly become reserved words in future releases of Database Manager or are reserved words from other IBM products that follow the SAA* specification.

  It is recommended that you do not use any of the SQL reserved words as ordinary identifiers.

Collectively, these lists represent the reserved list as documented in the *IBM Systems Application Architecture Common Programming Interface Database Reference.*

SQL Reserved Words

ADD
ALL
ALTER
AND
ANY
AS
ASC
AVG

BETWEEN
BIND
BINDADD
BIT
BY

CASCADE
CHAR
CHARACTER
COLUMN
COMMENT
COMMIT
CONNECT
CONSTRAINT
CONTROL
COUNT
CREATE
CREATETAB
CURRENT
CURSOR

DATA
DATABASE
DATE
DBADM
DEC
DECIMAL
DELETE
DESC
DISTINCT
DOUBLE
DROP

EXCEPT
EXCLUSIVE
EXECUTE
EXISTS

FETCH
FLOAT
FOR
FOREIGN
FROM

GRANT
GRAPHIC
GROUP

HAVING

IN
INDEX
INSERT
INT
INTEGER
INTERSECT
INTO
IS

KEY

LIKE
LOCK
LONG

MAX
MIN
MIXED
MODE

NOT
NULL
NUMERIC

OF
ON
ONLY
OPTION
OR
ORDER

PACKAGE
PRECISION
PRIMARY
PRIVILEGES
PROGRAM
PUBLIC

REAL
REFERENCES
RESTRICT
REVOKE
ROLLBACK

SBCS
SELECT
SET
SHARE
SMALLINT
SOME
SUM
SYNONYM

TABLE
TIME
TIMESTAMP
TO

UNION
UNIQUE
UPDATE
USER

VALUES
VARCHAR
VARGRAPHIC
VIEW

WHERE
WITH
WORK

Future SQL Reserved Word Candidates

ABSOLUTE
ACQUIRE
ADA
AFTER
ASSERTION
AUDIT
AUTHORIZATION
AVERAGE

BEFORE
BEGIN
BUFFERPOOL

C
CASE
CAST
CHECK
CLOSE
CLUSTER
COALESCE
COBOL
COMPLETION
CONSISTENCY
CONSTRAINTS
CONTINUE
CORRESPONDING

DATETIME
DAY
DAYS
DBA
DBCS
DBSPACE
DECLARE
DEFAULT
DESCRIBE
DESCRIPTOR
DISPLACEMENT
DOMAIN
DURATION

EACH
EDITPROC
ELSE
END
ERASE
ESCAPE
EXEC
EXPLAIN
EXTEND
EXTENDED

FIELDPROC
FIRST
FORTRAN
FOUND
FRACTION

GLOBAL
GO
GOTO

HOLD
HOUR
HOURS

IDENTIFIED
IMMEDIATE
INDICATOR
INPUT
INTERVAL
ISOLATION

LABEL
LANGUAGE
LAST
LENGTH
LEVEL
LOCAL
LOCKSIZE

MATCH
MICROSECOND
MICROSECONDS
MINUTE
MINUTES
MODIFY
MODULE
MONTH
MONTHS

N
NAMED
NATIONAL
NATURAL
NCHAR
NEW
NEXT
NHEADER
NONE
NULLIF
NUMPARTS

OFF
OLD
OPEN
OUTER

PAGE
PAGES
PART
PASCAL
PCTFREE
PCTINDEX
PENDANT
PLAN
PLI
PREPARE
PRESERVE
PRIOR
PRIQTY
PRIVATE
PROCEDURE

READ
REFERENCE
REFERENCING
RELATIVE
RESOURCE
ROW
ROWS
RUN

SCHEDULE
SCHEMA
SCROLL
SECOND
SECONDS
SECQTY
SECTION
SESSION
SQL
SQLCODE
SQLDESCRIPTOR
SQLERROR
SQLSTATE
STATISTICS
STOGROUP
STORPOOL
SUBSTR

SUBSTRING
SYSTEM

TABLEGROUP
TABLESPACE
TEMPORARY
TEST
TEXT
THEN
TRANSACTION
TRIGGER

UNITS
UNLOCK
USING

VALIDPROC
VALUE
VARIABLE
VARIABLES
VARYING
VCAT
VOLUMES

WAIT
WHEN
WHENEVER
WRITE

YEAR
YEARS

Appendix B. Query Manager – Personalizing Your Start-Up

This appendix provides information and steps to changing your Query Manager program information and starting Query Manager from the OS/2 command prompt.

Personalizing Your Program Information

You can personalize your Query Manager start-up to specify a database, profile, qualifier for lists, procedure, or row buffer size.

To Change Program Information for Query Manager

1. In the Group – Main window, select **Query Manager**.
2. Select **Program** from the action bar and select Enter.
3. Select **Properties** from the Program pull-down and select Enter.
4. In the Properties panel, type the following information in the Parameters field:

   ```
   /DAT:databasename /PRO:profilename
   /DQUAL:qualifier /RUN:name /ROWBUFFER:number
   ```

   See "Starting Query Manager from the OS/2 Command Prompt" on page B-3 for a detailed description of each parameter.

   Make sure **Program Type** is set to **Presentation Manager**.
5. Select **Change** from the Properties pull-down and select Enter.

   The information about the selected program is changed and the Group – Main window is displayed again.

Hint:

For more detailed information about using the Properties panel, see *IBM Operating System/2 Extended Edition Version 1.3 User's Guide, Volume 1: Base Operating System.*

To Recreate a Program Title for Query Manager

1. In the Group – Main window, select **Program** from the action bar and select Enter.
2. Select **New** from the Program pull-down and select Enter. The New Programs Menu panel is displayed.
3. In the New Programs Menu panel, type the following:
   a. For the Program title entry field, type:
   ```
   Query Manager
   ```
   b. For the Path and filename entry field, type:
   ```
   C:\SQLLIB\QUERYMGR.EXE
   ```
   Note: If `C` is not the drive where the code is installed, then type the correct drive instead of `C`.

   c. Do not type a parameter in the optional Parameters entry field.

   d. For the Working directory entry field, type:
   ```
   C:\SQLLIB
   ```
4. Select **Add** from the New Program pull-down to add the Query Manager title to the Group – Main window.

   See "Starting Query Manager from the OS/2 Command Prompt" on page B-3 for a detailed description of each parameter.

Hint:

For more detailed information about using the New Program panel, see the *User's Guide, Volume 1: Base Operating System.*

Starting Query Manager from the OS/2 Command Prompt

You can start Query Manager from the OS/2 command prompt by typing:

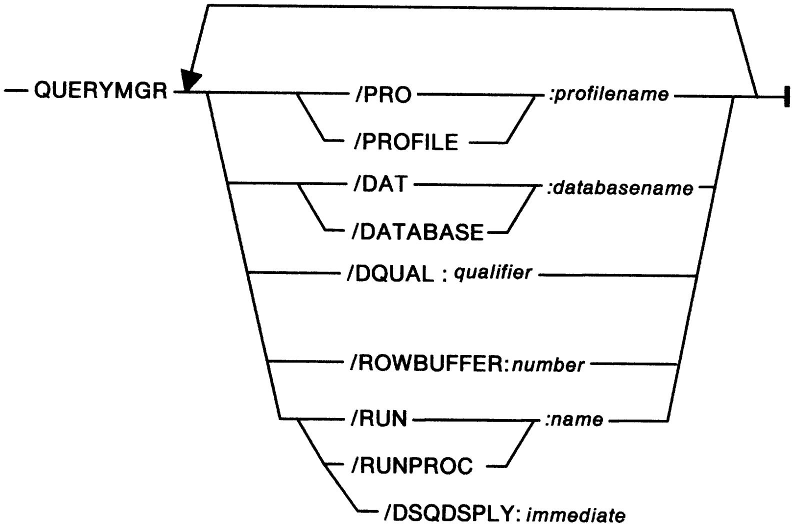
```
QUERYMGR.
```

Any of the parameters mentioned in the procedures "To Change Program Information for Query Manager" on page B-1 may be used with the QUERYMGR command. With the parameters, you can specify to use a profile. The profile must be in the same directory from where Query Manager started. All of the parameters can be specified in the profile except for the RUNPROC parameter. Any parameters typed from the OS/2 command prompt will override the corresponding parameter in the profile. For more information about reading the following syntax diagram, see the *OS/2 Command Reference*.

Purpose

You can use this command to start Query Manager from the OS/2 command prompt and specify parameters for running Query Manager.

Syntax


QUERYMGR
/PRO
/PROFILE
:profilename
/DAT
/DATABASE
:databasename
/DQUAL : qualifier
/ROWBUFFER:number
/RUN
/RUNPROC
:name
/DSQDSPLY: immediate

Example

```
querymgr /dat:sample /dqual:hise
/profile:profile1
querymgr /dat:sample /runproc:national_paper
```

Parameters

PRO or PROFILE
: The name of a Query Manager profile. This profile can contain other parameters used to start Query Manager. If there is a conflict between the parameters in the profile and those specified in this command, the parameters specified in the command are used.

DAT or DATABASE
: The name of the database you want to use. If this parameter is not specified, the profile searches for a database name. If a database name is not found in the profile, the Databases primary menu is displayed so you can select a database.

DQUAL
: The DQUAL is used as the default for lists. This is used to indicate which objects to list in both primary menus for objects and when you select List from within Query Manager. DQUAL also prefixes any objects you create. To fully qualify a procedure name, be sure to specify DQUAL.

: **Note:** In previous versions of Database Manager, you started Query Manager from the OS/2 command prompt by specifying the AUT or the AUTHID parameters. While these parameters are still valid, it is recommended in Database Manager Version 1.3 that you use the DQUAL parameter.

ROWBUFFER
: The random access memory (RAM) size to use for the Query Manager row pool buffer size allocated when Query Manager is started. If this parameter is not specified, the profile is searched for a value for **Buffer size for rows (# of K bytes)**. If no value is found in the profile, 16KB is used. For more information, see "Using Profiles" on page 8-35.

RUN or RUNPROC

This is used to indicate that you want to run a Query Manager procedure in customized mode. The name typed is the name of a saved Query Manager procedure in the database being used. The procedure name is prefixed with a qualifier, just as any other object in Query Manager is. Be sure that the qualifier for the procedure you are using matches the qualifier you are currently using or have specified. When you specify a procedure to be run, the database parameter must be specified or it must be in the profile.

DSQDSPLY

This parameter is used to indicate that you want the screens to appear immediately when you are running a Query Manager Callable Interface application. The only valid value for this parameter is `immediate`.

To Run a Procedure from the OS/2 Command Prompt

1. At the OS/2 command prompt, type:

   ```
   QUERYMGR /RUNPROC:procedure-name /DAT:databasename
   ```

   The *procedure-name* is the name of the procedure you want to run and *databasename* is the name of the database you want to use. Press the Enter key.

   Note: If you are using the QUERYMGR command to start Query Manager from the OS/2 command prompt and you specify a profile with the `/PROFILE` or the `/PRO` parameter, the profile must be in the same directory from where Query Manager started.

   If the database name you want to use is in the profile, you do not need to type the `/DAT` parameter; however, you must type the `/PRO` parameter to name the profile to be used.

2. If necessary, respond to any prompts, menus, or panels that the procedure processes.

   The OS/2 command prompt is displayed when the procedure is completed.

Hint:

For information on specifying profiles, see “Using Profiles” on page 8-35.

Appendix C. Database Manager—Designing Your Database

This appendix consists of a brief description of what to consider when designing a database. You might also want to consult one of the many textbooks available that deal with relational databases and their design.

There is no universally accepted terminology for database design. Hence, the terms used here may be used differently in other books. This section defines the basic terms for database design. Other terms are defined as they occur.

An *entity* is something about which information is stored in a database. In the sample tables that follow, some of the entities are employees, jobs, and salaries. In the "Employee Table (TEMPL)" on page C-17, those entities are represented by employee numbers, job codes, and amounts, which appear in the columns EMPNO, JOBCODE, and SALARY. Each value in the EMPNO column is an *occurrence* of the entity *employee number*. The entities *birthday* and *salary* are *properties* of the entity *employee* and can contain duplicate values.

In Database Manager tables, entities are represented as columns and occurrences as values in the columns, as in Table C-1 that follows:

Table C-1. Occurrences of Entities and Their Properties

Entity (employee)	**Properties**			
	EMPNO	**JOBCODE**	**BIRTHDAY**	**SALARY**
Sally Kwan	000030	60	410511	38250
John Geyer	000050	58	250915	40175

Deciding What Data to Record in the Database

Each column of a row is related in some way to all the other columns for that row. Some of the relationships expressed in the sample tables are:

- Employees are assigned to departments.
  - Dolores Quintana is assigned to Department C01.
  - Heather Nicholls is assigned to Department C01.
- Employees earn money.
  - Dolores earns $23,800 per year.
  - Heather earns $28,420 per year.
- Departments report to other departments.
  - Department C01 reports to Department A00.
  - Department D01 reports to Department A00.
- Employees work on projects.
  - Dolores and Heather work on project IF1000.
- Employees manage departments.
  - Sally Kwan manages Department C01.

Employee and *department* are entities; Sally Kwan and C01 are occurrences of the entities. Entities and their relationships can be represented as in Table C-2 that follows.

Table C-2. Some Relationships in the Sample Tables

ENTITY	RELATIONSHIP	ENTITY
Employees	are assigned to	departments
Employees	earn	money
Departments	report to	departments
Employees	work on	projects
Employees	manage	departments

Relationships are repetitive, corresponding to the multiple rows of a table. For example, one row of a table expresses the relationship that

Sally Kwan manages Department C01; another expresses the relationship that John Geyer manages Department E01.

Defining Tables for Each Type of Relationship

In a relational database, you can express several types of entity relationships. Consider the possible relationships between employees and departments. A given employee can work in only one department, thus we have a *single-valued* relationship for employees. On the other hand, one department can have many employees so we have a *multivalued* relationship for departments.

As you can see, this relationship is single-valued in one direction and multivalued in the other, because the assignment of employees to departments is a single-valued relationship for employees but is a multivalued relationship for departments.

Relationships can be:

- One-to-Many
- Many-to-One
- One-to-One
- Many-to-Many.

The relationship type of a given entity relationship can vary, depending on the specific environment. If employees of your company can earn multiple salary amounts, the relationship between employees and salaries is Many-to-Many. If, however, an employee can earn only one salary, the relationship is Many-to-One for employees (several employees may earn the same salary) and One-to-Many for salaries.

You should define separate tables for different types of entity relationships.

One-To-Many and Many-To-One Relationships

A relationship that is multivalued in one direction and single-valued in the other is a One-to-Many or Many-to-One relationship. The relationship between employees and the departments to which they are assigned is multivalued for departments and single-valued for employees.

The assignment to C01 is a single-valued relationship for Dolores, as shown in Table C-3 that follows:

Table C-3. A Single-Valued Relationship

EMPLOYEE	DEPARTMENT
Dolores Quintana	C01

If you ask the question, "What employees are assigned to Department C01?," your answer would be a list of all the employees in the table who work in Department C01. The assignments of Dolores Quintana, Sally Kwan, and Heather Nicholls to Department C01 express a multivalued relationship for Department C01, as shown in Table C-4 that follows:

Table C-4. A Multivalued Relationship

DEPARTMENT	EMPLOYEES
C01	Dolores Quintana
	Heather Nicholls
	Sally Kwan

To define tables for each One-to-Many and Many-to-One relationship, you must:

- Group all the relationships for which the *many* side of the relationship is the same entity.
- Define a separate table for each group (*employees* and *departments*).

In Table C-5 on page C-5, the *many* side of the first and second relationships is *Employees*, so you can define an employee table, TEMPL. In the third relationship, *Departments* is the many side, so you can define a department table, TDEPT. Figure C-1 on page C-5 illustrates the process.

Table C-5. Some Many-To-One Relationships

ENTITY	RELATIONSHIP	ENTITY
1. Employees	are assigned to	departments
2. Employees	earn	money
3. Departments	report to	(administrative) departments

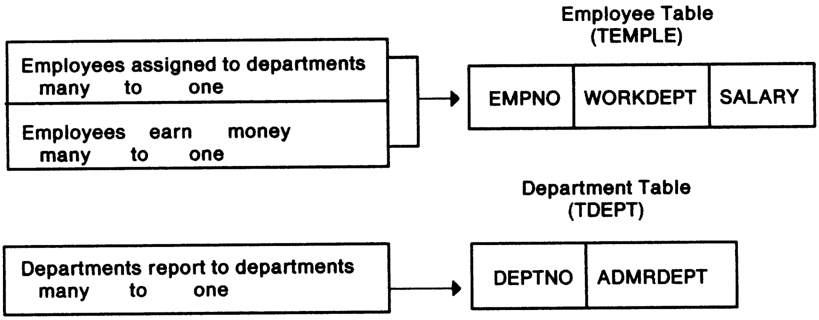


Figure C-1. Assigning Many-To-One Facts to Tables

Many-To-Many Relationships

A relationship that is multivalued in both directions is a Many-to-Many relationship.

An employee can work on more than one project, and a project can have more than one employee assigned to it. The questions, "What does Dolores Quintana work on?" and "Who works on project IF1000?" both yield multiple answers. A Many-to-Many table includes a column for each entity (employees and projects), as shown in Figure C-2 on page C-6.

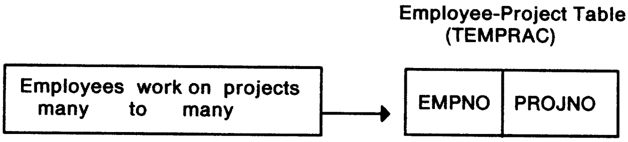


Figure C-2. Assigning Many-To-Many Facts to a Table

One-To-One Relationships

One-to-One relationships are single-valued in both directions. A manager manages one department; a department has only one manager. The questions, "Who is the manager of Department C01?" and "What department does Sally Kwan manage?" both have single answers. The relationship could be assigned to either the department table or the employee table. Because all departments have managers, but not all employees are managers, it would be most logical to add the manager to a department table as shown in Figure C-3.

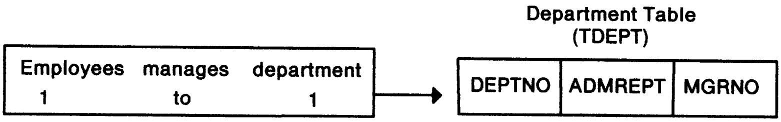


Figure C-3. Assigning One-To-One Facts to a Table

Providing Column Definitions for All Tables

Defining a column in a Database Manager table consists of:

- Choosing a name for the column. Each column in a table must have a name that is unique within the table.
- Assigning valid data to each column. The data type of a column indicates the length of the values in the column and the kind of data that is valid for the column.

- Determining which columns may need filler values. Some columns cannot have meaningful values in all rows because:
  - A value of the column is not applicable to the row.

    For example, a column containing an employee's middle initial is not applicable to an employee who has no middle initial.
  - A value is applicable, but the value is not known at this time.

    As an example, the MGRNO column might not contain a valid manager number because the previous manager of the department has been transferred and a new manager has not been appointed yet.

In both situations, you can allow a null value, which is a special value indicating that the column value is unknown or not applicable.

Identifying One or More Columns as a Primary Key

If every row in a table represents relationships for a unique entity, the table should have one column or a set of columns that provides a unique identifier for each record contained in the table. This column (or set of columns) is called the *primary key* of the table. To ensure that the primary key does not contain duplicate values, create a unique index on the column or columns that constitute the primary key (in Query Manager, this is done by setting **Duplicates allowed** to **No** in the Save Index panel).

Primary keys for some of the sample tables are:

Table	**Key Column**
Employee table (TEMPL)	EMPNO
Department table (TDEPT)	DEPTNO
Project table (TPROJ)	PROJNO

Figure C-4 shows part of the project table with the unique key column indicated.

UNIQUE KEY COLUMN

Project Table (TPROJ)

PROJNO	PROJNAME	DEPTINO
MA2100	Weld Line Automation	D01
MA2110	W L Programming	D011

Figure C-4. A Unique Key on a Table

Figure C-5 shows a unique key consisting of more than one column. It is called a *composite key.*

UNIQUE KEY COLUMN

Project Activity Table (TPROJAC)

PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
MA2100	10	.5	820101	821101
MA2100	20	1.0	820101	820301
MA2110	10	1.0	820101	830201

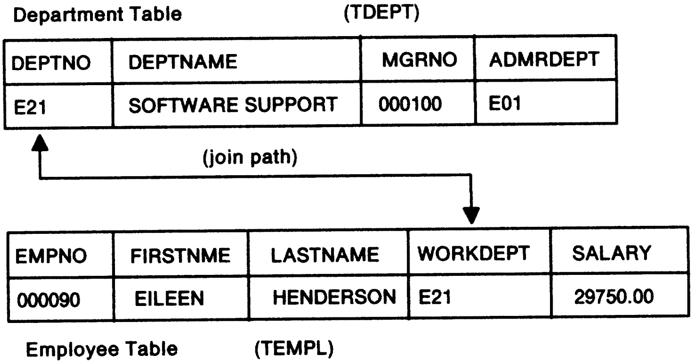
Figure C-5. A Composite Unique Key

All three columns (PROJNO, ACTNO, and ACSTDATE) are required to compose a unique key.

Ensuring That Equal Values Represent the Same Entity

Whenever the values of two columns are intended to represent relationships between entities, be sure that all equal values uniquely identify the same entity. If there are two employees named Sally Kwan, a join on employee names may not match the correct rows. Similarly, if one person has several social security numbers, a join on social security numbers may not produce the correct match.

Figure C-6 shows a join path between the department and employee tables.



Department Table (TDEPT)

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
E21	SOFTWARE SUPPORT	000100	E01

EMPNO	FIRSTNME	LASTNAME	WORKDEPT	SALARY
000090	EILEEN	HENDERSON	E21	29750.00

Employee Table (TEMPL)

Figure C-6. A Join Path between Two Tables

The connecting columns can have different names (like WRKDEPT and DEPTNO in Figure C-6), or they can have the same name.

Considering Normalizing Your Tables

The topic of normalizing tables draws much attention in database design. This section briefly reviews the rules for first, second, third, and fourth normal forms of tables, and describes some reasons why these rules should, or perhaps should not, be followed.

The fifth normal form of a table, which is covered in many books on database design, is not discussed here.

First Normal Form

Any relational table satisfies the requirement of first normal form. In this form, at each row-and-column position in the table, there exists only one value, never a set of values.

Second Normal Form

A table is in second normal form if each column not in the key provides a fact that depends on the entire key.

Second normal form is violated when a nonkey column is a fact about a subset of a composite key, as in the following example. An inventory table records quantities of specific parts stored at particular warehouses; its columns are shown in Figure C-7.

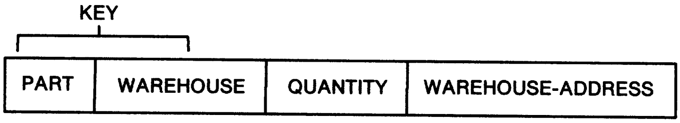


Figure C-7. Key That Violates Second Normal Form

The key in the preceding figure consists of the PART and the WAREHOUSE columns together. Because the column WAREHOUSE-ADDRESS depends only on the value of WAREHOUSE, the table violates the rule for second normal form. The basic problems with this design are:

- The warehouse address is repeated in every record for a part stored in that warehouse.
- If the address of the warehouse changes, every row referring to a part stored in that warehouse must be updated.
- Because of the redundancy, the data might become inconsistent, with different records showing different addresses for the same warehouse.

- If at some time there are no parts stored in the warehouse, there may be no row in which to record the warehouse address.

To satisfy second normal form, the preceding information should be in two tables, as in Figure C-8.

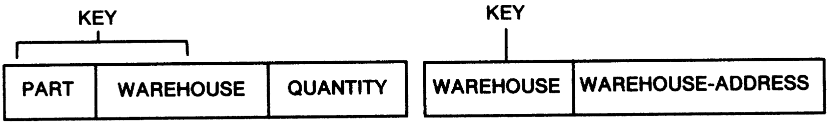


Figure C-8. Two Tables That Satisfy Second Normal Form

However, there is a performance disadvantage in having the two tables in second normal form. Queries that produce reports on the location of parts have to join both tables to retrieve the relevant information.

Third Normal Form

A table is in third normal form if each nonkey column provides a fact that depends only on the key.

Third normal form is violated when a nonkey column is a fact about another nonkey column. For example, the TEMPL table contains the EMPNO column (employee number) and the WRKDEPT column (department number). Suppose a column is added to give the department name: the new DEPTNAME column depends on WRKDEPT, whereas the primary key is the column EMPNO. The table now violates third normal form. Changing the DEPTNAME column for a single employee, John Parker, does not change the department name for other employees in that department. The inconsistency that results is shown in the updated version of the table in Figure C-9 on page C-12.

Employee-Department Table (EMPDEPT) Before Update

EMPNO	FIRTSNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Operations
000320	Ramlal	Mehta	E21	Software Services
000310	Maude	Setright	E11	Operations

Employee-Department Table (EMPDEPT) After Update

EMPNO	FIRTSNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Installation Mgmt
000320	Ramlal	Mehta	E21	Software Services
000310	Maude	Setright	E11	Operations

Figure C-9. Update of an Unnormalized Table

The information in the previous table has become inconsistent. The table can be normalized by providing a new table with columns for WRKDEPT and DEPTNAME. In that case, an update such as changing a department name is much easier; the update need only be made to the new table. A query that shows the department name along with the employee name is more complex to write. Such a query requires joining the two tables and can take longer to execute than the query of a single table. Also, the arrangement takes more storage space because the WRKDEPT column must be listed in both tables.

Fourth Normal Form

A table is in fourth normal form if no row contains two or more independent, multivalued facts about an entity.

Consider facts about employees, skills, and languages, where an employee may have several skills and know several languages. There are two relationships: one between employees and skills and one between employees and languages.

A table is not in fourth normal form if it represents both relationships, as in the example of Figure C-10 on page C-13.

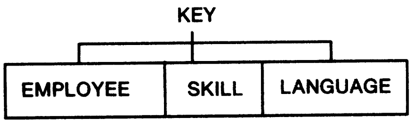


Figure C-10. A Table That Violates Fourth Normal Form

Instead, the relationships should be represented in two tables, as in Figure C-11.

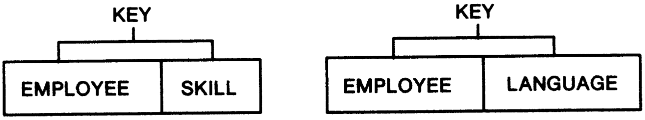


Figure C-11. Tables in Fourth Normal Form

If, however, the facts are interdependent (that is, the employee applies certain languages only to certain skills), the table should *not* be split.

Any data can be put into fourth normal form. A good rule when designing a database is to arrange all data in tables in fourth normal form and then decide whether the result gives you an acceptable level of performance. If it does not, you can un-normalize your design.

Planning for Referential Integrity

Referential integrity is a condition of a set of database tables in which all references from one table to another are valid and accurate. You can build a referential structure to ensure referential integrity. When planning for referential integrity, you should determine the relationships that are to be established between database tables. You can build a referential structure with SQL statements from the OS/2 command prompt or from the Query Manager prompted interface.

A *foreign key* is one or more columns in a table that contains values that must match the values of the primary key of a different table. A foreign key is used to establish a relationship with a primary key for the purpose of enforcing referential integrity. Primary and foreign keys that have more than one column are *composite* keys.

A *parent table* is the table containing the primary key that defines the relationship with a foreign key in another table. A table can be a parent table in several relationships.

A *parent row* is a row of a parent table whose primary key value matches a foreign key value in a dependent table. A row in a parent table is not necessarily a parent row.

A *dependent table* is a table containing one or more foreign keys. A dependent table can also be a parent table. A table can be a dependent in several relationships.

A *dependent row* is a row of a dependent table that has a non-null foreign key value that matches a primary key value of its parent table. The foreign key value represents a reference from the dependent row to the parent row. Since foreign keys can accept null values, a row in a dependent table is not necessarily a dependent row.

A table is a *descendent* of a table if it is dependent or if it is a descendent of a dependent table. A descendent table contains a foreign key that can be traced back to the primary key of some table.

A *cycle* is a path that connects a table to itself. When a table is connected to itself, it is a *self-referencing* table.

The following tables illustrate some of the relationships discussed so far:

Table C-6. Department Table

DEPTNO (PRIMARY KEY)	DEPTNAME
A00	Spiffy Computer Service Division
B01	Planning
C01	Information Center
D11	Development Center

Table C-7. Employee Table

EMPNO (PRIMARY)	FIRSTNME	LASTNAME	WORKDEPT (FOREIGN)	PHONE
000120	Sean	O'Connell	A00	2167
000020	Michael	Thompson	B01	3476
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

The primary key (DEPTNO) in the department table and the foreign key (WORKDEPT) in the employee table establish a referential structure. The department table is the parent table of the illustrated relationship, and the employee table is the dependent table.

The primary key (EMPNO) in the employee table indicates that the employee table is also the parent table in a different relationship.

Referential Integrity and Data Integrity

Database Manager supports referential integrity and data integrity. Both are enforced with *constraints.* A constraint places restrictions on columns and tables. The purpose of the constraint is to guarantee that database relationships are maintained and data entry rules are followed. Constraints can be defined when a table is created or altered. Referential constraints enforce referential integrity among the tables of a database, whereas column constraints enforce data integrity within the columns of a table.

You can define tables with referential integrity when you specify constraints on columns and tables. A referential constraint defines the rules for a relationship between a primary key of a parent table and a foreign key of a dependent table. The referential constraint requires that for each row in a dependent table, the value of the foreign key must match the primary key of a row in the specified parent table.

When defining primary and foreign keys, you are enforcing constraints on the values within the rows of a table or between the rows of two tables. Database Manager checks the constraints that are specified in a table definition and maintains the relationship accordingly.

Building a Referential Structure

Referential structures can be built while creating database tables. A *referential structure* is a set of tables in which each table in the set is a parent or a dependent of itself or of some other table in the set. A referential structure identifies the relationships within tables and among tables to ensure that database integrity is maintained. Building a referential structure requires that these relationships have been identified.

To construct relationships with referential constraints, define a parent table with a primary key and one or more dependent tables with foreign keys and referential constraints.

You can define referential constraints for a table when you define a new table or modify an existing table by selecting **Constraints** from the action bar in the Table primary menu. Follow the steps in "To Define a Table" on page 5-15 to define constraints for parent and dependent tables.

Note: When planning for referential integrity, keep in mind that it is quicker to define constraints when initially creating a table than it is to add constraints to an existing table. This is because the value of each foreign key must be checked against the primary key value in the parent table.

Sample Tables

Use these sample tables to aid you in understanding the examples of normalization.

Employee Table (TEMPL)

The employee table is used to describe employees in terms that are pertinent to a business enterprise. The table is intended for use by managers and by the personnel department.

Table C-8. TEMPL: Employee Table (Left Half)

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WRKDEPT	PHONENO	
000010	Christine	I	Haas	A00	3978	==>
000020	Michael	L	Thompson	B01	3476	==>
000030	Sally	A	Kwan	C01	4738	==>
000050	John	B	Geyer	E01	6789	==>
000060	Irving	F	Stern	D11	6423	==>
000070	Eva	D	Pulaski	D21	7831	==>
000090	Eileen	W	Henderson	E11	5498	==>
000100	Theodore	Q	Spenser	E21	0972	==>
000110	Vincenzo	G	Lucchessi	A00	3490	==>
000120	Sean		O'Connell	A00	2167	==>
000130	Dolores	M	Quintana	C01	4578	==>
000140	Heather	A	Nicholls	C01	1793	==>
000150	Bruce		Adamson	D11	4510	==>
000160	Elizabeth	R	Pianka	D11	3782	==>
000170	Masatoshi	J	Yoshimura	D11	2890	==>
000180	Marilyn	S	Scoutten	D11	1682	==>
000190	James	H	Walker	D11	2986	==>
000200	David		Brown	D11	4501	==>
000210	William	T	Jones	D11	0942	==>
000220	Jennifer	K	Lutz	D11	0672	==>
000230	James	J	Jefferson	D21	2094	==>
000240	Salvatore	M	Marino	D21	3780	==>
000250	Daniel	S	Smith	D21	0961	==>
000260	Sybil	P	Johnson	D21	8953	==>
000270	Maria	L	Perez	D21	9001	==>
000280	Ethel	R	Schneider	E11	8997	==>
000290	John	R	Parker	E11	4502	==>
000300	Philip	X	Smith	E11	2095	==>
000310	Maude	F	Setright	E11	3332	==>
000320	Ramlal	V	Mehta	E21	9990	==>
000330	Wing		Lee	E21	2103	==>
000340	Jason	R	Gounot	E21	5698	==>

Table C-9. TEMPL: Employee Table (Right Half)

EMPNO	...	HREDATE	JOBCODE	EDUCLVL	SEX	BRTHDATE	SALARY
000010	...	650101	66	18	F	330814	52750
000020	...	731010	61	18	M	480202	41250
000030	...	750405	60	20	F	410511	38250
000050	...	490817	58	16	M	250915	40175
000060	...	730914	55	16	M	450707	32250
000070	...	800930	56	16	F	530526	36170
000090	...	700815	55	16	F	410515	29750
000100	...	800619	54	14	M	561218	26150
000110	...	580516	58	19	M	291105	46500
000120	...	631205	58	14	M	421018	29250
000130	...	710728	55	16	F	250915	23800
000140	...	761215	56	18	F	460119	28420
000150	...	720212	55	16	M	470517	25280
000160	...	771011	54	17	F	550412	22250
000170	...	780915	54	16	M	510105	24680
000180	...	730707	53	17	F	490221	21340
000190	...	740726	53	16	M	520625	20450
000200	...	660303	55	16	M	410529	27740
000210	...	790411	52	17	M	530223	18270
000220	...	680829	55	18	F	480319	29840
000230	...	661121	53	14	M	350530	22180
000240	...	791205	55	17	M	540331	28760
000250	...	691030	52	15	M	391112	19180
000260	...	750911	52	16	F	361005	17250
000270	...	800930	55	15	F	530526	27380
000280	...	670324	54	17	F	360328	26250
000290	...	800530	42	12	M	460709	15340
000300	...	720619	48	14	M	361027	17750
000310	...	640912	43	12	F	310421	15900
000320	...	650707	52	16	M	320811	19950
000330	...	760223	55	14	M	410718	25370
000340	...	470505	54	16	M	260517	23840

Department Table (TDEPT)

The department table describes each department in the business enterprise and specifies the department's manager and administrative (or next-higher-level) department.

Table C-10. TDEPT: Department Table

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
A00	Spiffy Computer Service Div.	000010	
B01	Planning	000020	A00
C01	Information Center	000030	A00
D01	Development Center		A00
E01	Support Services	000050	A00
D11	Manufacturing Systems	000060	D01
D21	Administration Systems	000070	D01
D31	Order Processing Systems		D01
E11	Operations	000090	E01
E21	Software Support	000100	E01

Project Table (TPROJ)

The project table describes each project that the business enterprise is currently involved with. Some projects are parts of other projects. Data contained in each row includes the project's number, name, person responsible, and schedule dates.

Table C-11 (Page 1 of 2). TPROJ: Project Table (Left Half)

PROJNO	PROJNAME	DEPTNO	RESPEMP	
AD3100	Admin Services	D01	000010	==>
AD3110	General Ad Systems	D21	000070	==>
AD3111	Payroll Programming	D21	000230	==>
AD3112	Personnel Programming	D21	000250	==>
AD3113	Account Programming	D21	000270	==>
IF1000	Query Services	C01	000030	==>
IF2000	User Education	C01	000030	==>
MA2100	Weld Line Automation	D01	000010	==>
MA2110	W L Programming	D11	000060	==>
MA2111	W L Program Design	D11	000220	==>
MA2112	W L Robot Design	D11	000150	==>
MA2113	W L Prod Cont Progs	D11	000160	==>

Table C-11 (Page 2 of 2). TPROJ: Project Table (Left Half)

PROJNO	PROJNAME	DEPTNO	RESPEMP	
OP1000	Operation Support	E01	000050	==>
OP1010	Operation	E11	000090	==>
OP2000	Gen Systems Services	E01	000050	==>
OP2010	Systems Support	E21	000100	==>
OP2011	SCP Systems Support	E21	000320	==>
OP2012	Applications Support	E21	000330	==>
OP2013	DB/DC Support	E21	000340	==>
PL2100	Weld Line Planning	B01	000020	==>

Table C-12. TPROJ: Project Table (Right Half)

PROJNO		PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
AD3100	...	6.5	820101	830201	
AD3110	...	6	820101	830201	AD3100
AD3111	...	2	820101	830201	AD3110
AD3112	...	1	820101	830201	AD3110
AD3113	...	2	820101	830201	AD3110
IF1000	...	2	820101	830201	
IF2000	...	1	820101	830201	
MA2100	...	12	820101	830201	
MA2110	...	9	820101	830201	MA2100
MA2111	...	2	820101	821201	MA2110
MA2112	...	3	820101	821201	MA2110
MA2113	...	3	820215	821201	MA2110
OP1000	...	6	820101	830201	
OP1010	...	5	820101	830201	OP1000
OP2000	...	5	820101	830201	
OP2010	...	4	820101	830201	OP2000
OP2011	...	1	820101	830201	OP2010
OP2012	...	1	820101	830201	OP2010
OP2013	...	1	820101	830201	OP2010
PL2100	...	1	820101	820915	MA2100

Project Activity Table (TPROJAC)

The project activity table describes each project in terms of the activities performed for the project. The table also includes starting and ending dates for each activity.

Table C-13 (Page 1 of 2). TPROJAC: Project Activity Table

PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
MA2100	10	.5	820101	821101
MA2100	20	1.0	820101	820301
MA2110	10	1.0	820101	830201
MA2111	40	1.0	820101	830201
MA2111	50	1.0	820101	820601
MA2111	60	1.0	820601	830201
MA2112	60	2.0	820101	820701
MA2112	180	1.0	820701	830201
MA2112	70	1.5	820215	830201
MA2113	80	1.5	820901	830201
MA2113	60	1.0	820215	820901
MA2113	70	2.0	820401	831215
MA2113	180	.5	821001	830101
PL2100	30	1.0	820201	820901
IF1000	90	1.0	820101	830101
IF1000	100	.5	821001	830101
IF1000	10	.5	820101	830101
IF2000	10	.5	820101	830101
IF2000	100	.75	820101	820701
IF2000	110	.5	820301	820701
IF2000	110	.5	821001	830101
AD3100	10	.5	820101	820701
AD3110	10	1.0	820101	830101
AD3111	60	.80	820101	820415
AD3111	70	1.5	820215	821015
AD3111	180	1.0	821015	830115
AD3111	80	1.25	820415	830115
AD3112	60	.75	820101	820315
AD3112	60	.75	821201	830101
AD3112	70	.75	820101	821015
AD3112	80	.35	820815	821201
AD3112	180	.5	820815	830101

Table C-13 (Page 2 of 2). TPROJAC: Project Activity Table

PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
AD3113	60	.75	820301	821015
AD3113	70	1.25	820601	821215
AD3113	80	1.75	820101	820415
AD3113	180	.75	820301	820701
OR4100	10	1.0	820501	830201
OR4100	20	.5	820201	820901
OR4110	30	1.0	820101	820515
PL4100	30	1.0	820201	820901
OP1000	10	.25	820101	830201
OP2000	50	.75	820101	830201
OP1010	10	1.0	820101	830201
OP1010	130	4.0	820101	830201
OP2010	10	1.0	820101	830201
OP2011	140	.75	820101	830201
OP2011	150	.25	820101	830201
OP2012	140	.25	820101	830201
OP2012	160	.75	820101	830201
OP2013	140	.5	820101	830201
OP2013	170	.5	820101	830201

Employee-To-Project Activity Assignment Table (TEMPRAC)

The employee-to-project activity assignment table relates a project with the employee who performs each activity involved with the project. The employee's level of involvement (that is, full-time or part-time) and schedule for the activity are also in the table.

Table C-14 (Page 1 of 3). TEMPRAC: Employee-To-Project Activity Assignment Table

EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
000130	IF1000	90	1.0	820101	821001
000130	IF1000	100	.5	821001	830101
000140	IF1000	90	.5	821001	830101
000030	IF1000	10	.5	820601	830101
000030	IF2000	10	.5	820101	830101
000140	IF2000	100	1.0	820101	820301
000140	IF2000	100	.5	820301	820701
000140	IF2000	110	.5	820301	820701

Table C-14 (Page 2 of 3). TEMPRAC: Employee-To-Project Activity Assignment Table

EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
000140	IF2000	110	.5	821001	830101
000010	MA2100	10	.5	820101	821101
000110	MA2100	20	1.0	820101	820301
000020	PL2100	30	1.0	820101	820915
000010	MA2110	10	1.0	820101	830201
000220	MA2111	40	1.0	820101	830201
000200	MA2111	50	1.0	820101	820615
000200	MA2111	60	1.0	820615	830201
000150	MA2112	60	1.0	820101	820715
000150	MA2112	180	1.0	820715	830201
000170	MA2112	60	1.0	820101	830601
000170	MA2112	70	1.0	820601	830201
000190	MA2112	70	1.0	820201	821001
000190	MA2112	80	1.0	821001	831001
000160	MA2113	60	1.0	820715	830201
000170	MA2113	80	1.0	820101	830201
000180	MA2113	70	1.0	820401	821215
000210	MA2113	80	.5	821001	830201
000210	MA2113	180	.5	821001	830201
000010	AD3100	10	.5	820101	820701
000070	AD3110	10	1.0	820101	830201
000230	AD3111	60	1.0	820101	820315
000230	AD3111	60	.5	820315	820415
000230	AD3111	70	.5	820315	821015
000230	AD3111	80	.5	820415	821015
000230	AD3111	180	1.0	821015	830101
000240	AD3111	70	1.0	820215	820915
000240	AD3111	80	1.0	820915	830101
000250	AD3112	60	1.0	820101	820201
000250	AD3112	60	.5	820201	820315
000250	AD3112	60	.5	821201	830101
000250	AD3112	60	1.0	830101	830201
000250	AD3112	70	.5	820201	820315
000250	AD3112	70	1.0	820315	820815
000250	AD3112	70	.25	820815	821015
000250	AD3112	80	.25	820815	821015
000250	AD3112	80	.5	821015	821201
000250	AD3112	180	.5	820815	830101
000270	AD3113	60	.5	820301	820401
000270	AD3113	60	1.0	820401	820901
000270	AD3113	60	.25	820901	821015
000270	AD3113	70	.75	820901	821015
000270	AD3113	70	1.0	821015	830201
000270	AD3113	80	1.0	820101	820301
000270	AD3113	80	.5	820301	820401

Table C-14 (Page 3 of 3). TEMPRAC: Employee-To-Project Activity Assignment Table

EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
000260	AD3113	70	.5	820615	820701
000260	AD3113	70	1.0	820701	830201
000260	AD3113	80	1.0	820101	820301
000260	AD3113	80	.5	820301	820415
000260	AD3113	180	.5	820301	820415
000260	AD3113	180	1.0	820415	820601
000260	AD3113	180	.5	820601	820701
000050	OP1000	10	.25	820101	830201
000090	OP1010	10	1.0	820101	830201
000280	OP1010	130	1.0	820101	830201
000290	OP1010	130	1.0	820101	830201
000300	OP1010	130	1.0	820101	830201
000310	OP1010	130	1.0	820101	830201
000050	OP2010	10	.75	820101	830201
000100	OP2010	10	1.0	820101	830201
000320	OP2011	140	.75	820101	830201
000320	OP2011	150	.25	820101	830201
000330	OP2012	140	.25	820101	830201
000330	OP2012	160	.75	820101	830201
000340	OP2013	140	.5	820101	830201
000340	OP2013	170	.5	820101	830201
	OR4100	10	1.0	820501	830201
	OR4100	20	.5	820201	820901
000120	OR4110	30	1.0	820101	820515
000110	OR4120	40	1.0	820301	830201
	PL4100	30	1.0	820201	820915

Appendix D. Database Manager Import and Export Considerations

Import can be used to insert data from an OS/2 file into a Database Manager table or view. Export can be used to export data from a Database Manager table or view to an OS/2 file.

Four types of files can be imported to and three types can be exported from a database. The type indicates the format of the data within the OS/2 file. The OS/2 file formats supported by Database Manager are:

DEL Delimited ASCII, for exchange with dBASE II or dBASE III, BASIC programs, IBM Personal Decision Series (PDS), and DB2 SQL/DS (import only). For more information on importing DB2 and SQL/DS, see Appendix E.

ASC Non-delimited ASCII for importing data from other applications; for example, from the IBM DisplayWrite Series and IBM Personal Editor.

WSF Work-Sheet formats, for exchange with Lotus 1-2-3 versions 1.0, 1A, and 2.0, and Symphony versions 1.0 and 1.1.

IXF PC/IXF version, the preferred method for exchange within Database Manager. Use PC/IXF to export data from a table so that it can be imported later into the same table or into another table.

Delimited ASCII (DEL) File Format

The following information describes the format of Delimited ASCII (DEL) files, which can be imported into Database Manager or can be generated as the result of exporting a Database Manager table or view. A DEL file is sequential ASCII with row and column delimiters. It can be used to exchange data with dBASE II and dBASE III as a DELIMITED WITH " file, with BASIC programs using READ and WRITE statements, and with PDS as a Basic Sequential file.

Each DEL file is a stream of ASCII characters consisting of cell values ordered by row and then by column. Rows in the data stream are separated by row delimiters. Within a row, the individual cell values are separated by column delimiters. During import of a DEL file, spaces that precede the first character of a cell value or that follow the last character of a cell value are discarded.

Using the Query Manager Profiles task, you can override the default character for column delimiter (,), character string delimiter ("), and decimal point character (.).

The following is an example of a DEL file:

```
"Smith, Bob,"4973,15.46
"Jones, Bill,"12345,16.34
"Williams, Sam,"452,193.78
```

Each line ends with a carriage return (/) line feed sequence (this is the row delimiter). A row is `"Smith, Bob,"4973,15.46` . A column contains `"Smith, Bob"` `"Jones, Bill"` `"Williams, Sam"`.

If you change the column delimiter to a semicolon (;), the character string delimiter to a single quote ('), and the decimal point character to a comma (,), the same file would appear as follows:

```
'Smith, Bob';4973;15,46
'Jones, Bill';12345;16,34
'Williams, Sam';452;193,78
```

When you are importing or exporting DEL files, keep in mind the following:

- For the profile default characters for character string delimiter and column delimiter:
  - A space (x'20') is never a valid character string delimiter nor a valid column delimiter.
  - The period (.) is not a valid character string delimiter due to conflicts with periods in time and time-stamp values.
  - The character string delimiter (default double quote) must not be typed within character strings. An attempt to export character data containing a character string delimiter causes a warning message. Attempts to import a file with a character string delimiter within character data produces erroneous results.
- Import of character strings that are not enclosed in character string delimiters is allowed. The end of a *nondelimited* character string is

determined by the first occurrence of a space, character string delimiter or, row delimiter.

- A null value is indicated by the absence of a cell value where one would normally occur or by a string of spaces. There are only spaces or nothing at all instead of an ASCII representation of a numeric value or a character string.
- Since dBASE restricts character fields to 254 characters and BASIC restricts them to 255 characters, Export outputs a warning message whenever a character column of maximum length greater than 254 characters is selected for export. Import accommodates fields as long as the longest possible length of a special data column.
- The first occurrence of an end-of-file character (Hex '1A') that is not within character string delimiters indicates the end-of-file. No data following the end-of-file character is imported.
- Integer, decimal, and scientific notation constants can be imported into numeric database columns that are within the proper range.
- The acceptable forms for importing date and time data are based on the OS/2 country code of the target database. For more information on these data types, see Chapter 5.

Nondelimited ASCII (ASC) File Format

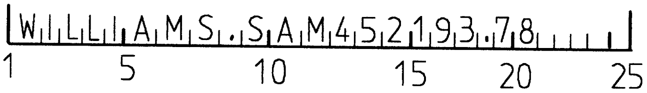
This information describes the format of nondelimited ASCII (ASC) files, which can be imported into Database Manager. An ASC file is sequential ASCII with row delimiters. It can be used to exchange data with any ASCII product that can create data in a columnar format, including word processors.

Each ASC file is a stream of ASCII characters consisting of data values organized by row and column. Rows in the data stream are separated by ASCII carriage return/line feed sequences. Each column within a row is defined in accordance with a beginning-ending location pair (supplied in a file specification). Each pair represents locations within a row specified as byte positions. (The first position within a row is byte position 1). The first element of each location pair is the byte within the row where the column begins. The second element of each location pair is the byte where the column ends. The columns may overlap. For left-to-right languages, the content of a column is assumed to read from left to right. Within one ASCII file, every row has the same column definitions.

No special processing is done for column names. Each row is considered to be data, which means ASC files are assumed to have no row of column names.

The following is an example of an ASC file. `Williams, Sam` begins in byte position 1 and ends in byte position 12; `452` begins in byte position 13 and ends in byte position 15, `193.78` begins in byte position 16 and ends in byte position 21.

```
Smith, Bob        4973           15.46
Jones, Bill      12345           16.34
Williams, Sam      452          193.78
```



Each line ends with a carriage return/line feed sequence. A sample row is:

```
Smith,Bob          4973           15.46
```

A sample column is:

```
Smith, Bob
Jones, Bill
Williams, Sam
```

WSF File Format

The format and description of WSF files that can be imported into Database Manager or can be generated as the result of a Database Manager Export are discussed in the following information. The various Lotus 1-2-3 and Symphony products use the same basic format with additional functions added at each new release. Database Manager supports a subset of the worksheet records that is the same across all the Lotus products. Thus, Database Manager supports any Lotus 1-2-3 or Symphony file produced by the releases supported by Database Manager (as indicated at the beginning of this appendix) no matter what OS/2 extension (WKS, WK1, WRK, WR1, and so on) is in the file name.

Each WSF file represents one worksheet. The following conventions are used by Database Manager to successfully interpret worksheets and to provide consistency in worksheets generated by Database Manager Export operations:

- Cells in the first row (ROW value 0) are reserved for descriptive information about the entire worksheet. All data within this row is optional. It is ignored by Database Manager when importing from a WSF file.
- Cells in the second row (ROW value 1) are used for column labels.
- The remaining rows are data rows (records, or rows of data from the table).
- Cell values under any column heading are values for that particular column or field.
- A null value is indicated by the absence of a real cell content record (for example, no integer, number, label, or formula record) for a particular column within a row of cell content records.

  Note: Database Manager does not support importing a record of all nulls.

PC/IXF File Format

The personal computer (PC) version of the IXF format is a specific format used by Database Manager. Import accepts only PC/IXF files, *not* Host IXF files. PC/IXF is a structured description of a database table that contains an external representation of the internal table. Data exported in PC/IXF format can be imported by a different Database Manager system. The code page value for the PC/IXF file must match the code page value of the database, unless the code page value is overridden (**Compare code page on import** set to **No** in the profile).

The following list contains rules to keep in mind when importing PC/IXF files into Database Manager tables and views:

- A non-nullable PC/IXF column can be imported into a nullable column (a column with **Data required** set to **No**).
- A nullable PC/IXF column can be imported into a non-nullable column, although some rows may be rejected.

- Numeric columns accept columns of any numeric type, although some data may be rejected because it is out of range.
- Fixed-length string columns in the PC/IXF file that are too long for the target column are not compatible and are not imported. Variable-length string columns with actual lengths that are not compatible with the target column are processed according to the compatibility rules used when adding data to a table or view. The data is padded on the right with spaces if necessary.
- Date, time, and time stamp columns accept data from PC/IXF columns with matching types and from character PC/IXF columns. Data values from character PC/IXF columns must be valid input values for dates, times, or time stamps for successful insertion into each of the corresponding type columns.

The following restrictions apply when importing PC/IXF files into Database Manager tables and views:

- Import does not process columns containing 4-byte floating-point data. This is considered to be an incompatible type.
- Import rejects a PC/IXF file with more than 1024 columns.

Appendix E. Importing Data from DB2 and SQL/DS Databases

The SQLQMF facility allows you to import table data stored in IBM Database 2 (DB2) or Structured Query Language/Data System (SQL/DS) host relational databases into a Database Manager table.

This facility takes data exported by the IBM host Query Management Facility (QMF), stored in QMF format, downloads the data, and converts it into a delimited ASCII file. The delimited ASCII file can then be imported into a Database Manager table. In Database Manager Version 1.3, the SQLQMF facility is enhanced to support a third command: SQLQMF. The commands SQLQMFV (for the Virtual Storage (VM) environment) and SQLSMFM (for the Multiple Virtual Storage (MVS) environment) continue to be supported.

The new SQLQMF command allows you to specify automatic table creation (if a table does not already exist) as well as allowing you to specify that you want the data imported directly when you issue the SQLQMF command. Using the SQLQMFV and SQLQMFM commands, you will have to create the table before the import process can begin and perform the import as a separate step after the data has been downloaded.

Additional parameters available using the new SQLQMF command are also discussed in this appendix. If you are a first-time user of this facility, it is recommended that you use the SQLQMF command because of the time-saving steps it provides.

Using the SQLQMF Facility

To use the SQLQMF facility, you must have the following:

- The DB2 relational database program product installed on your IBM MVS system or the SQL/DS relational database program product installed on your IBM VM system.
- QMF installed on your IBM VM or MVS system.
- Communications Manager and Database Services installed on your workstation. Query Manager may optionally be installed.

- A database created with Database Manager.

Familiarity with QMF, Communications Manager, and Database Manager is required in order to use this facility.

To use the SQLQMF facility, you must be logged on to the appropriate host system through Communications Manager.

To import data from the host using the new SQLQMF command, use the following procedures:

- Export the data from QMF
- Run the SQLQMF command.

To import data from the host using the SQLQMFV or SQLQMFM command, you must also do the following:

- Print the column definition file and create the table if one does not already exist
- Import the data into a Database Manager table.

Each of these procedures is described on the following pages. The purpose, syntax, and parameters of the SQLQMF, SQLQMFV, and SQLQMFM commands are also described in the *OS/2 Command Reference.*

The SQLQMF facility downloads the exported QMF data and performs the necessary data format conversion in a single step. During this process, up to four files are automatically created:

- Column definition file

  The column definition file contains a description of the exported QMF table. You can use this information to define the Query Manager table into which the data will eventually be imported if you have opted not to used the automatic table creations option of the SQLQMF command. The column definition file indicates for each column the data type, name, length, and whether the column should be defined as *data required.*

  The file created will have a file extension of `.COL` to indicate it is the column definition file. If the SQLQMF command is used and data is imported with no error or warnings, this file is deleted.

- Import file

  The file exported from QMF and downloaded from the host is converted to a delimited ASCII file. This file contains the data that will be imported into a Database Manager table. This file is set if you have chosen not to use the automatic import option of the SQLQMF command.

  The file created will have a file extension of .DEL to indicate it is the delimited file that contains data. If the SQLQMF command is used and data is imported with no errors or warnings, this file is deleted.

- CREATE TABLE statement file

  An SQL CREATE TABLE statement containing the same information as the column file. This file will have a file extension of .CRE and will exist after use of the SQLQMF command where import had an error or warnings.

- Import message file

  A file of messages generated by import.

  This file will have a file extension of .IML and will exist after use of the SQLQMF command if import had an error or warning messages.

The SQLQMF command allows you to place the downloaded files in any drive and directory you specify. The SQLQMFV and SQLQMFM commands store these files in the SQLLIB\QMF subdirectory.

Exporting the Data from QMF

The QMF EXPORT command provides the capability to send data to a CMS file in the VM environment or to a TSO data set in the MVS environment.

To export data, you first select the data to be exported and then issue an EXPORT DATA TO *filename* command within QMF.

There are several ways that you can select data to be exported, including first running a query, which brings data into the workspace, or importing data into the QMF workspace, for eventual export, or by both, running a query and then importing data. Refer to the reference documentation for QMF for more information on how to export data.

Two points to remember while using QMF are:

- The export data format chosen should be QMF format, *not* IXF format. QMF is the default.

  The following is an example of the QMF export command:

  ```
  EXPORT DATA TO filename
  ```

  The file name specified can contain from 1 to 8 characters and must adhere to the file name restrictions of both the host system and the OS/2 program.

  A QMF menu interface is also provided for exporting data. You can type `EXPORT ?` on the QMF command line, and then follow the menu sequences to prepare the table data for exporting.

  As the data is being exported out of the host database, QMF will create a CMS file (in the VM environment) or a TSO data set (in the MVS environment). Some examples of QMF exported files are shown in the following text.

  If you are exporting data from QMF and intend to use the SQLQMFV or SQLQMFM commands, the file names *must* be specified as follows:

  For VM: *file name* DATA A

  An example is: ORGTABLE DATA A

  For MVS: *userid.filename*.DATA

  An example is: SMITH.ORGTABLE.DATA

  If you are exporting data from QMF and intend to use the SQLQMF command, the parameters after the file name can be something other than DATA A (for VM) or DATA (for MVS). For example:

  For VM: ORGTABLE.HOST B

  For MVS: SMITH.ORGTABLE.HOST

- Once the desired data has been exported from QMF, type `EXIT` and return to the host-ready prompt.

  In order to successfully invoke the SQLQMF facility from the OS/2 command prompt, the VM `R;` prompt or the MVS `Ready` prompt must be displayed on the host session screen.

Running the SQLQMF Facility

Before you run the SQLQMF facility, be sure that you have already exported the data to a VM CMS file or MVS data set and that you are logged on to the appropriate host from which you exported the QMF data. You should also read "Communications Session ID Considerations" on page E-10 and "Data Considerations" on page E-11.

Steps for using the SQLQMF, SQLQMFV and the SQLQMFM commands follow. Choose the command that is most appropriate for your environment.

Running the SQLQMF Command

The new SQLQMF command, used by Database Manager Version 1.3, allows you to specify automatic table creation and automatic import of data, along with other options, as the SQLQMF command is specified. Each of these options is described in this appendix. The new command can be run in either the MVS or VM environment. MVS is the default. The command syntax is as follows:

```
SQLQMF hostfilename pcfilename options
```

The following is an example of the SQLQMF command:

```
SQLQMF PAYROLL HOST B2 PAYROLL /V /D:PERSONNEL
/T:USERID.PAYROLL/I:C /S:B
```

The following is a list and description of SQLQMF parameters.

- **host filename (required)**

  The name of the file as it is stored on the host after being exported from QMF.

- **pc filename (required)**

  Indentifies the PC file name that is used to name various datasets used for downloaded and converted data and other information. The file name can be up to 8 characters and may be optionally prefixed with a drive letter and path information to cause all files to be created in a different OS/2 subdirectory than the currently active directory.

- **V (optional keyword parameter)**

  Host system is a VM system. There is no operand following this parameter. If not specified, an MVS system will be assumed.

- **M (optional keyword parameter)**

  Host system is an MVS system. There is no operand following this parameter. This is the assumed default unless the host system is specified as a VM system with the **V** parameter.

- **D (optional keyword parameter)**

  Name of the database into which the table is to be imported. All characters following the colon until a (/) delimiter or end of line is encountered are assumed to be the database name. Remember, the Database Manager must be started in order to create the table and import the data into a database. You can do this by issuing a STARTDBM command from the OS/2 command line if necessary.

  If omitted, the host file will only be downloaded and converted into an ASCII delimited file. If the /I (for import) parameter is specified, this parameter must also be specified.

- **T (optional keyword parameter)**

  The name to be put in the CREATE TABLE statement. If omitted, the table name will be the same as the PC file name (not including the drive letter and path names).

  The name, if supplied, can be qualified with a user ID prefixing it; for example, BROWN.PAYROLL. All characters found following the colon until a (/) delimeter or end of line is encountered are assumed to be the table name.

- **I (optional keyword parameter)**

  Option to be used for the import operation. If specified, the *D* (database) option must also be specified. If this parameter is omitted, and D is specified, */I:C* is assumed. The valid options are as follows:

- **R**

  Replaces all rows of an existing table.

- **C**

  Creates the table and import the data. The table must not previously exist.

  If this option is chosen, a COMMIT command is run after the table is created. If a system failure occurs at this point, the table will contain no data.

- **A**

  Appends the data to rows of an existing table. The table must exist.

- **O**

  Overlays any existing column definition and its data. The table may or may not exist in the database.

  If this option is chosen and the table exists, the table is erased and redefined. If a system failure occurs at this point, the table may contain no data.

- **S (optional keyword parameter)**

  Identifies the 3270 communication's short session ID to be used to transfer the host file to the PC. If omitted, session *A* will be assumed. If provided, the operand must be a single alphabetic character.

Keyword parameters can appear in any order and must start with a slash.

During the SQLQMF facility process, the number of bytes downloaded is displayed on the screen. If the SQLQMF facility was successful, you will also see a completion message including the names of the output files created and the number of rows and columns processed.

Using the SQLQMF command, you can also choose to only download the data and not do the automatic table creation or automatic import until later. Steps for printing the Column Definition File, creating the table, and importing the data are described later in this appendix.

Running the SQLQMFV and the SQLQMFM Commands

The SQLQMFV and the SQLQMFM commands continue to be supported in the OS/2 program Version 1.3. A description of how to issue these commands follows:

- In the VM environment, use the SQLQMFV command, which has the syntax:

  SQLQMFV *filename* [*session id*]

  If you are operating in an MVS environment, use the SQLQMFM command, which has the following syntax:

  SQLQMFM *filename* [*session id*]

- The file name specified should be the file name typed when exporting the data from QMF.
- You may need to specify the optional session ID parameter to override the default host communications session ID (A).

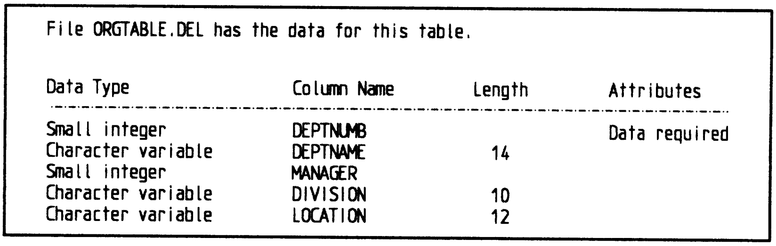
Printing the Column Definition File and Creating the Table

The SQLQMF command used by Database Manager Version 1.3 automatically imports and creates a Database Manager table. The following instructions apply if you are using the SQLQMFV and the SQLQMFM commands, or the SQLQMF command with no import parameter specified.

The column definition file provides a description of the exported QMF table. This description can aid you in defining the Database Manager table into which the data will be imported.

If you have not already defined a Database Manager table, you should print the column definition file and review its contents. If you have already defined the table, you can skip this procedure. The column definition file is located in the \SQLLIB\QMF directory for the specified drive. (It may be in a different location if the SQLQMF command was used and you specified a different library.)

The following is an example of the information the column definition file provides:



```
File ORGTABLE.DEL has the data for this table.

Data Type             Column Name     Length     Attributes
-------------------------------------------------------------
Small integer         DEPTNUMB                   Data required
Character variable    DEPTNAME         14
Small integer         MANAGER
Character variable    DIVISION         10
Character variable    LOCATION         12
```

To Print the Column Definition File

1. At the OS/2 command prompt, specify the directory and type: `\SQLLIB\QMF`.

   Type `DIR`, if necessary, to view the list of column definition files and imported tables.

2. Type `PRINT` *`filename`*`.COL`, where *`filename`* is the name of the column definition file you want to print.

To Create the Database Manager Table

1. Start Query Manager, open the desired database, and create a new table using the information provided in the column definition file.

2. Follow the steps in Chapter 5 to create a table as required.

Importing the Data into a Database Manager Table

If you use the SQLQMF command and specified an import option, the following steps are not required.

To Import Data Into a Database Manager Table

1. Start Query Manager, using the correct database profile.

2. Open the Database Manager database into which you want to import the data from QMF.

3. In the Tables and Views primary menu, select the appropriate table.

4. Select **Tools** from the action bar.

5. Select **Import** from the Tools pull-down.

6. Type the name of the OS/2 file (QMF file name) where the data is stored.
7. Set **OS/2 data format** to **DEL**.
8. Set **Data** to **Replace existing data** or **Append to existing data** and then select Enter.

Notes:

1. To view any error messages that occur during the import process, view or print the QRWIMPRT.LOG file. The error message states the directory where the file is located. It is the current directory, which may be SQLLIB unless the working directory parameter was changed.
2. If you have a lot of data to be imported, you can get some log full messages during the import process. These messages include message numbers SQL3186W and SQL0964. Import usually recovers from log full conditions if no other process is using the same database.

Communications Session ID Considerations

Any host 3270 or ASCII terminal emulation session can be used to transfer QMF-extracted data.

Using Communications Manager, it is possible to be logged on to more than one host at a time. Each host session is assigned an ID. The SQLQMF facility needs to know which session ID to address during its download process.

The user can identify a short communications session ID to be used with the SQLQMF command as a single letter (A, B, C, D, and so on). If you do not specify a communications session ID, the ID will default to session A.

Data Considerations

The following considerations should be taken into account when using the SQLQMF facility to download data from a host SQL database to a Database Manager table:

- Data type support

  The SQLQMF facility supports the following data types:

  - SMALLINT
  - INTEGER
  - DECIMAL (a maximum of 31 digits are supported)
  - FLOAT (scientific notation)
  - CHAR (character, fixed-length, 254 bytes maximum)
  - VARCHAR (character, variable-length, 4000 bytes maximum)
  - DATE
  - TIME
  - TIMESTAMP (system date and time).

  The SQLQMF facility does *not* support the following data types:

  - LONG VARCHAR (special data)
  - GRAPHIC
  - VARGRAPHIC
  - LONG VARGRAPHIC.

- Floating-point (scientific notation) data

  Floating-point data has a different internal representation in the IBM 370 machine format as compared to the IEEE standard for the PC. During the host-to-PC conversion, some loss of precision of the mantissa takes place. As a result of the conversion, the mantissa will have at most 8 digits of precision.

- Reserved Words

  Query Manager does not allow reserved words to be used as column names. See Appendix A for a complete listing of reserved words. Some of these reserved words are not reserved in the host database environment. As a result, even though your column definition file may list these words for the column names in the host-generated table, you should use a different name when defining the Database Manager table. For example, if the column definition file contains a column heading of TIMESTAMP or DATE, you could name the same column in your Database

Manager table with TIMESTAMPX or DATEX to differentiate them from the reserved words TIMESTAMP and DATE.

- Nontext (binary) data

  If the host database table contains nontext (binary) data stored in a character column, the non-text data will be subjected to regular EBCDIC-to-ASCII conversion, producing undesired results. The SQLQMF facility cannot recognize this situation.

- Double-byte character support (DBCS)

  The SQLQMF facility does not support DBCS graphic data types.

- Code page support

  The code pages supported during the downloading of the tables using Communications Manager depends on the type of terminal emulator in use for the host 3270 connection. Translation services are provided by Communications Manager and are limited to the following:

  For ASCII terminal connections, the host code page 037 – 1 is assumed and can be used with any of the PC code pages listed in the following table.

  For a 3270 connection, any of the host code pages listed can be used with any of PC code pages listed in the following table.

Host Code Page	**PC Code Page**
037 – 1	437
273 – 1	850
277 – 1	860
278 – 1	863
280 – 1	865
284 – 1	
285 – 1	
297 – 1	
500 – 1	

- Delimiter support

  If your DB2 or SQL/DS table contains character column data that contains double quotes ("), there will be some loss of data during the import process. Typically, the column data following the quotes will be ignored while importing to the Database Manager table. You will see a warning message notifying you that data was truncated during import. Revise this data as required.

- File delimiters

  The SQLQMF facility uses the following file delimiters:

  - Column delimiter is a comma (,).
  - Character string delimiter is double quotes (").
  - Decimal point character is a period (.).

  Be sure that when you start Query Manager to import data into a Database Manager table, you use a profile that has the file delimiters specified for DEL format, which are the same as those listed in the preceding text. Refer to Chapter 8 for more information.

Appendix F. Resolving Conflict with Reserved Words

This appendix describes the procedure for changing the name of an object with a reserved word as its name. After this change, you can use the new object in the Database Manager component of OS/2 Version 1.3.

Reserved words are words that cannot be used to define a table, view, or column. See Appendix A for a complete list of all the reserved words included in the Database Manager component of OS/2 Version 1.3.

If you were a user of the Database Manager component of OS/2 Version 1.0 or 1.1, you may have previously created a table, view, or column name using a reserved word. Because certain words are reserved in OS/2 Version 1.3, Database Manager returns an error message when an object containing one of these words is encountered.

To Replace Object Names That Are Reserved Words

1. Secure your database by backing up all objects you intend to change.
2. Start Query Manager. See "Starting and Stopping Query Manager" on page 2-7 for instructions on starting Query Manager.
3. In the Main Selection menu, select **Tables and Views.**
4. In the Tables and Views primary menu, select the name of the table or view you were trying to access when the error occurred.
5. Select **Actions** from the action bar and then select **Open definition** from the Actions pull-down, or press the Open Definition (F6) key.
6. Select **Actions** from the action bar and then select **Print** from the Actions pull-down to print the table definition, or press the Print (F9) key. The printed table definition includes the name and definition of the table and the names and definitions of its columns and indexes.

Note: This printed definition is the record of definitions that you need in this procedure. Be sure to print the table definition.

7. Select **Exit** from the action bar and then select **Exit Table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

8. Select **Exit without saving** from the confirmation message.

9. If the reserved word is used as a view name, go to step 10.

   If the reserved word is used as a table name or column name, go to step 11.

10. To redefine a view, perform the following steps:

    a. Select —NEW— from the Tables and Views primary menu.

    b. Select **Actions** from the action bar and then select **Open definition** from the Actions pull-down, or press the Open Definition (F6) key.

    c. Select **View** from the Open menu.

    d. To define the tables for the view, follow the steps in "To Specify Tables for a Query" on page 3-10. Use the printed definition you made in step 6.

    e. To define the columns for the view, follow the steps in "To Specify Columns for a Query" on page 3-13.

       Note: After you leave the Columns menu, the New Column Names panel is displayed so you can rename the selected columns for the view.

    f. Type the name you want to use in the New Column Names panel and select Enter. The columns listed under **Column Name** are the columns you defined in step 10e.

       Notes:

       1) A new column name is only required for columns that are defined as summary functions or expressions, or when joining tables that have duplicate column names. If you do not type a new column name, the column name used is the one under **Column Name**.

       2) See Appendix A for the naming rules for column names and a list of SQL reserved words that should not be used when naming columns.

g. To define the row conditions for the view, follow the steps in "To Specify Row Conditions for a Query" on page 3-17.

h. To specify whether duplicate rows are allowed for the view, see "To Discard Duplicate Rows in a Query" on page 3-24.

i. Select **Exit** from the action bar and then select **Exit view** from the Exit pull-down, or press the Exit View (F3) key. The confirmation message is displayed.

   Notes:

   1) If you select **Resume**, the View panel is displayed again and you can continue editing the view definition.

   2) Be sure the view is defined as you want it since a view definition cannot be changed once it is saved. After a view is saved, it must be erased and created again if you need to make changes to the view.

j. Select **Save and exit** from the confirmation message to save the view definition and select Enter. The Save panel is displayed.

   Notes:

   1) If you select **Exit without saving**, the view definition is not saved and the Tables and Views primary menu is displayed again.

   2) If you select **Resume**, the View panel is displayed again and you can continue editing the view definition.

k. Type the view name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

   Notes:

   1) For information on the naming rules for view names, see Appendix A.

   2) If any procedures, queries, panels, or menus refer to the erased view, you must modify them so that they refer to the new view.

l. Select the name of the old view from the Tables and Views primary window.

m. Select **Actions** from the action bar and then select **Erase** from the Actions pull-down, or press the Erase (Shift + F8) key.

n. Select **Yes** when prompted for confirmation and select Enter. The View primary menu is displayed.

   The new view has replaced the old view, and the conflict with a reserved word has been resolved. If you need to replace other object names, return to step 3 of this procedure. Otherwise, if this is the only conflict with a reserved word, select **Exit** from the action bar and then select **Exit Query Manager** from the Exit pull-down, or press the Exit Query Manager (Shift+F3) key.

11. To redefine a table and the names of its columns, perform the following steps:

    a. Select **—NEW—** from the Tables and Views menu.

    b. Select **Actions** from the action bar and then select **Open definition** from the Actions pull-down, or press the Open Definition (F6) key.

    c. Select **Table** from the Open menu to display the Table panel.

    d. Select **Get template** from the actions pull-down, and then type the name of the table with a reserved word in its definition and select Enter.

    e. If a column name is a reserved word, perform the following steps:

       1) Select **Actions** from the action bar and then select **Change a column** from the Actions pull-down, or press the Change a Column (Ctrl+F1) key.

       2) Select the name of the column that is the reserved word in the Change Column menu.

       3) Change the name of the column in the Change Column menu and select Enter.

       4) Repeat steps 11e2 and 11e3 to change other column names in the table that have reserved words for names.

    f. Select **Exit** from the action bar and then select **Exit table** from the Exit pull-down, or press the Exit Table (F3) key. The confirmation message is displayed.

       Note: If you select **Resume**, the Table panel is displayed again and you can continue editing the table definition.

g. Select **Save and exit** from the confirmation message to save the table definition and select Enter. The Save panel is displayed.

   Notes:

   1) If you select **Exit without saving**, the table definition is not saved and the Tables and Views primary menu is displayed again.

   2) If you select **Resume**, the Table panel is displayed again and you can continue editing the table definition.

h. Type a valid table name and an optional comment in the Save panel and select Enter. The Tables and Views primary menu is displayed.

   Note: For information on the naming rules for table names and also for a list of reserved words, see Appendix A.

i. If the name of an index is a reserved word, perform the following steps:

   1) Select **Erase index** from the Actions pull-down, and then select the name of the first index that is a reserved word and select Enter.

   2) Select **Yes** when prompted for confirmation and select Enter. Repeat these steps until you have erased all the indexes that are reserved words to the table. The Tables and Views primary menu is displayed.

j. In the Tables and Views menu, select the table with the reserved word conflict.

k. Select **Tools** from the action bar and then select **Export** from the Tools pull-down.

l. Follow the prompts to type the appropriate information for the Export Data panel.

   Notes:

   1) In the Export Data panel, type the OS/2 file name into which the data is to be exported.

   2) In the Export Data panel, be sure that the **OS/2** data format is set to **IXF**. When you have responded to all the prompts, select Enter.

The Tables and Views primary menu is displayed.

m. Select **Tools** from the action bar and then select **Import** from the Tools pull-down.

n. Follow the prompts to type the appropriate information for the Import utility.

   Notes:

   1) In the Import panel, type the new table name you specified in step 11h.

   2) In the Import panel, type the same OS/2 file name you specified in step l for the Export Data panel.

   When you have responded to all the prompts, select Enter. The Tables and Views primary menu is displayed.

o. If you erased the indexes in step 11i, perform the following steps. Otherwise, continue with step 11p.

   1) Select **Add index** from the Actions pull-down.

   2) From the table definition you printed in step 6, type the column name or names you want in the index and the order for sorting in the Add Index panel.

      Note: Select List to display the columns defined in the selected table or to display the selections for order.

   3) When you have completed typing all the column names you want and their sorting order, select Enter.

   4) Type the index name and set **Duplicates allowed** to your choice from the Save Index panel and select Enter.

      Note: For information on the naming rules for index names, see Appendix A on page A-1.

p. Use the Run Statistics utility to update the information about accessing the data in the new table. See "To Reorganize a Table" on page 8-28 for instructions on using the Run Statistics utility.

   Note: When you erase a table, the table definition and the data in the table are removed. All views and indexes dependent on the erased table are erased also. If any procedures, queries, panels, or menus reference the erased view, you must modify them to reference the new table.

q. Select the name of the old table from the Tables and Views menu.

r. Select **Actions** from the action bar and then select **Erase** from the Actions pull-down, or press the Erase (Shift+F8) key.

s. Select **Yes** when prompted for confirmation and select Enter. The old table is erased and its name is deleted from the Tables and Views menu.

The new table has replaced the old table, and the conflict with a reserved word has been resolved. If you need to replace other object names, return to step 3 of this procedure. Otherwise, if this is the only conflict with a reserved word, select **Exit** from the action bar and then select **Exit Query Manager** from the Exit pull-down, or press the Exit Query Manager (Shift+F3) key.

Appendix G. Migration

This appendix provides information concerning migration of previous versions of Database Manager databases and migrating menus, panels, and forms created under previous versions of Query Manager. A database created with Version 1.2 does not require migration.

Migrating Database Manager Databases

Every database created with Database Manager Version 1.0 or Version 1.1 must be migrated before use with Database Manager Version 1.3. You can automatically perform a migrate from a prompted interface or you can also use the MIGRATE1 command at the OS/2 command prompt.

A confirmation message is displayed prompting you to migrate the selected database the first time you select:

- A database created under a previous version of Database Manager from the Databases primary menu
- **Authorizations**
- **Object table maintenance**.

As a user with an administrative authority level, you can select **Restore local database** to return a database to a prior backed up state. The first time you attempt to restore a database created with a previous version of Database Manager, Database Services will automatically run the migration utility.

Migration converts data created with Version 1.0 or Version 1.1 so that this data can be used with Database Manager Version 1.3. Databases should be migrated immediately after Database Manager Version 1.3 is installed. You can migrate only a single database at a time. When you migrate a database, you are also performing the following operations:

- Changing the database configuration file to reflect the new release parameter requirements

- Changing the recovery log files to the Database Manager Version 1.3 format
- Erasing indexes and recreate indexes as required
- Creating default Database Manager authorizations.

Migrating a database increases the size of storage requirements due to the additional files required by Database Manager Version 1.3. The size of the databases migrated will increase by at least 120KB. The space required by these files must be factored into the total database storage requirements.

The time required to migrate a database does not significantly depend on the amount of data within the database. Rather, the number of tables, indexes, and access plans determines the amount of time needed to migrate a database. The larger the number of tables and plans, the longer it takes the migration process to complete. The first time you select a table, the index for the table is recreated, which can result in a delay of processing. The index for a migrated table is recreated only once.

Migrating Query Manager Objects

When you access menus, panels, or forms created under a previous version of Query Manager, a migration on each of these objects is automatically performed. The structure of the other Query Manager objects (queries and procedures) has not changed, and the objects are not, therefore, migrated. Migration converts the previous version of the menu, panel, or forms to the Query Manager Version 1.3 format.

Panels

When you select a Version 1.0 or Version 1.1 panel, the following conversions are performed:

- The **Add, Change**, and **Search** titles in the Title Line panel for the Version 1.3 panel are set to the value displayed in the single title in the Title Line panel in the Version 1.0 or Version 1.1 panel.
- The **Panel search query** in the Rules for Panels panel for the Version 1.3 panel is set to the value displayed for the **Query name**

in the Search Conditions panel in the Version 1.0 or Version 1.1 panel.

- The **Procedure (Add)** and **Procedure (Change)** in the Rules for Panels panel for the Version 1.3 panel are not specified with a procedure name.
- The **End Rule** in the Rules for Panels panel for the Version 1.3 is set to **Search**.
- Any Alter and function keys (for example, Alt+F7) associated with panel actions specified for the Version 1.0 or Version 1.1 panel are removed and are no longer valid.
- Some text displayed in the Version 1.3 panel may be clipped (not truncated) due to the conversion from the fixed font used in Versions 1.0 and 1.1 to the proportional font used in Version 1.3. You can select **Edit** from the Layout pull-down to edit and align the text displayed in the Version 1.3 panel.

Menus

When you select a Version 1.0 or 1.1 menu, the following conversion is performed:

> Some text displayed in the Version 1.3 menu may be clipped (not truncated) due to the conversion from a fixed font used in Versions 1.0 and 1.1 to the proportional font used in Version 1.3. You can select **Edit** from the Layout pull-down to edit and align text displayed in the Version 1.3 menu.

Forms

The conversion of a Version 1.0 or 1.1 form to a Database Manager Version 1.3 on format does not result in any changes to the object.

You should save menus, panels, and forms after their first use. Otherwise, the objects will be migrated again the next time they are accessed. Saving the object reduces the time required to access them in the future.

Authorization

When Query Manager prompts you to perform the migrate utility, you must provide the database password created for the Database Manager Version 1.0 or 1.1 database. Once the database is migrated, a password is no longer used to control access to the database. Access to Database Manager Version 1.3 is controlled through User Profile Management. Knowledge of the database password used in previous versions of Database Manager is the only authorization required to run the migrate utility.

When a database is migrated, Database Manager automatically sets authorizations over the database, tables, and views. These authorizations can be modified from previous versions of Database Manager. The *Database Manager Administrator's Guide* discusses how Database Manager creates the new authorization defaults. Database Manager Version 1.3 authorizations are described in Chapter 2 and throughout this volume.

If you are a user with SYSADM (system administrator) authority, you automatically become a database administrator for the database you migrate. As a database administrator, you can make changes to the default authorizations Query Manager sets when migrating a Database Manager Version 1.0 or 1.1 database. For information and steps on how to grant and revoke database and table and view authorizations, see Chapter 8.

When you save a menu or panel (or any Query Manager object), you can also control access to the object by setting **Share** to **Yes** or **No** in the Save panel. The Share option for saving a menu is discussed in Chapter 12 and the Share option for saving a panel is discussed in Chapter 13.

To Migrate a Database

All existing databases should be backed up prior to installation of Database Manager Version 1.3. If you plan to use the High Performance File System (HPFS), you must back up your databases because your fixed disk will be reformatted before the HPFS is installed. After you install Database Manager, migrate Version 1.0 and Version 1.1 databases so that they can be used by Database Manager Version 1.3. Installation installs the Database Manager configuration file again, which does not require migration. However, any changes to this file to customize the system operation need to be repeated. You can migrate a database as follows:

- In the Databases primary menu, select the database name. The confirmation message is displayed. Type the database password and select Enter.

  This procedure works only for Database Manager Versions 1.0 and 1.1 databases that are on your fixed disk *after* Database Manager Version 1.3 has been installed. If you have existing Versions 1.0 and 1.1 databases on a drive and you do not format this drive before, during, or after install, you can use Query Manager Version 1.3 to perform this procedure.

- Issue the MIGRATE1 command from the OS/2 command prompt by typing:

  MIGRATE1 *database password*

  The *database* parameter is the name of the database to be migrated.

  The *password* parameter is the password for the Database Manager database denoted by the *database* parameter. MIGRATE1 works only for Version 1.0 and Version 1.1 databases that remain on your fixed disk *after* Database Manager Version 1.3 has been installed. If you format your C drive when you install Database Manager Version 1.3, any databases on your C drive are erased, and you cannot use the MIGRATE1 command. If you have Version 1.0 or Version 1.1 databases existing on a drive and you do not format this drive before, during, or after install, you can use the MIGRATE1 command after you install Database Manager Version 1.3.

- Use the Restore utility in Database Manager Version 1.3 to restore a database that was backed up using OS/2 Extended Edition Version 1.0 or Version 1.1. When the database is restored it is automatically migrated to a Database Manager Version 1.3 format.

Appendix H. Database Manager Messages

This appendix contains Database Manager messages that *do not* have online help information available. Each message has a message identifier that consists of a prefix and a message number. The following table lists each prefix and the section of the messages appendix in which these messages can be found. Within each section, messages are listed in numeric sequence based upon the message number. Each message is followed by the appropriate cause and action information.

Prefix	Description of Associated Messages	Refer to:
SQL	Messages that can be generated while the Sample Database program is being installed.	"Sample Database Installation Messages" on page H-3.
SQLQMF	Messages that can be generated while running the SQLQMF facility.	"SQLQMF Facility Messages" on page H-10

Note: If a message or message help directs you to contact your service coordinator, refer to "Messages That Direct You to Contact Your Service Coordinator" on page H-2.

Messages That Direct You to Contact Your Service Coordinator

Before you contact your service coordinator for problems that persist, you should use the following procedure. This procedure is intended to assist you in gathering diagnostic information that is needed to resolve a problem.

To Prepare to Contact Your Service Coordinator

1. Record any message information, including error information or codes imbedded in the messages, that is displayed on the screen.

   Note: Press the Print Screen (PrtSc) key to print displayed messages, if the keyboard is not locked.

2. Determine whether your installed version the of the OS/2 program has been changed since your application last ran successfully. If it was changed, examine the changes. If you can associate the error with any of the changes, record the change as part of the problem description.

3. If possible, correct all problems indicated by messages and ensure that any messages previously generated have nothing to do with the current problem.

4. Record the sequence of events that led to the error condition. This information will be used in the problem resolution process.

5. Reduce the size of the failing application or scenario as much as possible to eliminate extraneous symptoms and to demonstrate the failure more clearly. Record this information as part of the problem description.

6. If the failure is associated with a specific statement or operation, record the parameters or variables for that statement or operation and include them as part of the problem description.

7. If the problem persists, contact your service coordinator for assistance in diagnosing and reporting the problem.

Sample Database Installation Messages

The following is a list of warning and error messages that you can receive while installing the sample database. This list may not include all the error messages you may receive. For additional information about messages, refer to the *Database Manager Programming Guide and Reference*.

Message identifiers ending with a **W** indicate warning or informational messages. Those ending with an **N** indicate errors that cause the sample database installation program to end. Message identifiers ending with a **C** indicate critical system errors.

SQL0001N Binding did not complete successfully.

Cause: The bind request for the sample database was unsuccessful.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0002N The bind file name is not valid.

Cause: The bind file name for the sample database cannot be used as specified.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0031N File "filename" could not be opened.

Cause: The file *filename* is required but could not be opened.

Action: Erase the sample database using Query Manager. Install the sample database again. If the error persists, erase the sample database using Query Manager. Install Database Services again. Then install the sample database again.

SQL0083C Memory allocation error.

Cause: During processing, there was insufficient memory to continue processing.

Action: Possible solutions include:

- Verify that your workstation has the required memory.
- Change the MEMMAN "NO SWAP,NO MOVE" option in the CONFIG.SYS file to "SWAP,MOVE."
- Remove background processes.

SQL0094N Precompile was ended due to user interrupt request.

Cause: An interrupt was received during processing. Perhaps you pressed the System Interrupt (Ctrl+Break) key.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0818N A time stamp conflict occurred.

Cause: A program error occurred.

Action: Record the number and the text of this message. Then contact your service coordinator.

SQL0902C A system error (reason code = "reason-code") occurred. Subsequent SQL statements cannot be processed.

Cause: A system error occurred.

Action: Record the number and the text of this message. Then contact your service coordinator.

SQL0952N Processing was cancelled because the Break (Ctrl+Scroll Lock) key was pressed.

Cause: An interrupt was received during processing. Perhaps you pressed the System Interrupt (Break) (Ctrl+Scroll Lock) key.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0954C There is not enough storage available in the application heap to process the statement.

Cause: All available memory has been used.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0956C There is not enough storage available in the database heap to process the statement.

Cause: All available memory for the database has been used.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0958C The maximum number of open files has been reached.

Cause: The maximum number of OS/2 file handles available to the database has been reached.

Action: Try again. If the error persists, stop all other applications, and then try to install the sample database again.

SQL0960C The maximum number of files has been reached in the database.

Cause: The maximum number of database files has been reached.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0964C The transaction log for the database is full.

Cause: The file space for the transaction log has been used.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0968C The file system is full.

Cause: The file system (or drive) containing the database is full.

Action: Clear some space in the file system by erasing the sample database, erasing some files, or erasing (dropping) some indexes and/or tables. Then install the sample database again.

SQL0974N The drive containing the database is locked.

Cause: The OS/2 program has reported that the drive containing the database is locked.

Action: Ensure no other processing is occurring on the system (for example, in another session) that can lock the drive (for example, CHKDSK running in another session). Erase the sample database using Query Manager. Install the sample database again.

SQL0980C A disk error occurred. Subsequent SQL statements cannot be processed.

Cause: A disk error has occurred.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0984C COMMIT or ROLLBACK was not successful. Subsequent SQL statements cannot be processed.

Cause: A COMMIT or ROLLBACK operation could not be processed successfully because of a system error.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0985C A file error occurred while processing the database catalogs. The database is not usable.

Cause: The system is unable to recover from an I/O error on a catalog file.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL0986N A file error occurred while processing a user table. The table is not usable.

Cause: A file error occurred.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL1022C There is not enough memory available to process the command.

Cause: There is not enough memory available to install the sample database program.

Action: Possible solutions include:

- Verify that your workstation has the required memory.
- Change the MEMMAN "NO SWAP, NO MOVE" option in the CONFIG.SYS file to "SWAP,MOVE."
- Remove background processes.

SQL1034C The database is damaged. Issue a STOP USING DATABASE command and stop processing.

Cause: Damage has occurred to the database.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL1035N The database is already in use.

Cause: You are using the sample database in another session.

Action: Switch to the other session and stop using the sample database. Erase the sample database using Query Manager. Install the sample database again.

SQL1036C An I/O error occurred while accessing the database.

Cause: One or more of the database files has an I/O error.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL1039C An I/O error occurred while accessing the database directory.

Cause: The database directory cannot be accessed.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL1042C An unexpected system error occurred.

Cause: A system error occurred. One possible reason for this error is that Database Manager is not installed correctly.

Action: Erase the sample database using Query Manager. Install the sample database again.

If the error persists, record the number and the text of this message. Then contact your service coordinator.

SQL1043C Database services could not initialize the system catalogs. Error "<errnum>" was returned.

Cause: The CREATE DATABASE command failed while initializing the system catalogs.

Action: Record the number and the text of this message. Then contact your service coordinator.

SQL1044N Processing was interrupted.

Cause: An interrupt was received during processing. Perhaps you pressed the System Interrupt (Break) (Ctrl+Scroll Lock) key.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL1060N "<authid>" does not have the CONNECT privilege.

Cause: Your user ID is not granted database authorization to select the database.

Action: Contact a user with SYSADM (system administrator) or a database administrator for the selected database to grant your user ID *Open Database* authorization.

SQL1092N "<authid>" does not have authority to perform the requested command.

Cause: You attempted to create or erase an existing database, catalog, or uncatalog a workstation without having SYSADM (system administrator) authority.

Action: Contact a user with SYSADM (system administrator) authority for assistance.

SQL1093N The user is not logged on.

Cause: You are not logged on to User Profile Management with an authorized user ID.

Action: Contact a user with administrator level authority within User Profile Management to create you a user ID and grant you access authorization.

SQL5005C A system error occurred.

Cause: A system error occurred.

Action: Erase the sample database using Query Manager. Install the sample database again. If the error persists, install Database Services again.

SQL5025C The Database Manager configuration file is not current.

Cause: The system (Database Manager) configuration file has been updated (during another session) since you started installing the sample database.

Action:

1. Ensure no other sessions are using Database Manager or Query Manager.
2. Erase the sample database using Query Manager.
3. Install the sample database again.

SQL5030C The release number is not valid.

Cause: The release number in the system (Database Manager) configuration file or the database configuration file is not valid.

Action: If the sample database exists, erase it using Query Manager. Install the sample database again. If the error persists, erase the sample database using Query Manager and install Database Services again. Then install the sample database again.

SQL5045N The Database Manager configuration file is not compatible with the database configuration file.

Cause: The number of segments required by the entries in the database configuration file is more than the number of segments allowed for Database Services in the Database Manager configuration file. If there is more than one database allowed to run concurrently, there are not enough segments allowed to run all the allowed databases.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQL5055N The content of the database configuration file is not valid.

Cause: The database configuration file for the sample database is not valid. The file may have been altered by a text editor or a program other than Database Manager.

Action: Erase the sample database using Query Manager. Install the sample database again.

SQLQMF Facility Messages

The following is a list of error messages that you can receive while running the SQLQMF facility. This list may not include all the error messages you may receive. For additional information about messages, refer to the *Database Manager Programming Guide and Reference.*

SQL6001N The prefix for the file name was not specified.

Cause: The user has executed a module of the SQLQMF facility directly rather than using the SQLQMFV and SQLQMFM commands.

Action: Use the SQLQMFV or SQLQMFM command.

SQL6002N System administrator encountered a download error.

Cause: Communications Manager encountered an error during the download of the host file.

Action: Review the Communications Manager Message Log.

SQL6003N The file exported from QMF contains rows that are too long. The row length is <nnnn>.

Cause: The computed row size (the sum of the computed column sizes) exceeds the maximum of 7000 bytes.

Action: Return to your QMF host session and run a query with fewer data columns selected. Export the data again and then rerun the SQLQMFV or SQLQMFM command.

SQL6004N <function name> returned the unexpected return code <code>.

Cause: An unexpected error occurred during processing. Communications Manager or Database Manager may not be installed or configured correctly.

Action: Ensure that Communications Manager has been installed and the appropriate host communications session is active. Try the command again after checking for other errors. If the problem persists, contact your Communications Manager system administrator.

SQL6005N An error occurred while reading the downloaded QMF file.

Cause: One of the following conditions was encountered:

- The file could not be opened.
- The end of file was encountered prematurely.
- There was an input/output (I/O) error while reading the file.

Action: Review the system administrator Message Log. Try the command again. If the error persists, contact your Communications Manager system administrator.

SQL6006N An error occurred while writing to an output file.

Cause: One of the following conditions was encountered:

- Not enough space exists on drive C to write the data.
- An output file could not be opened.
- An input/output (I/O) error occurred while writing the file.
- An I/O error occurred while closing the file.
- The file is in use by another OS/2 process.

Action: Check to see if there is sufficient disk space on drive C. Try the command again.

SQL6007N The decimal number in row <row num>, column <name> cannot be converted to ASCII.

Cause: The indicated decimal field could not be converted.

Action: Verify that the host column data type is decimal. Retry the QMF EXPORT command again; then retry the SQLQMFV or SQLQMFM command. If the error persists, rerun the QMF query without the named column.

SQL6008N The file specified in the command is not in QMF data format.

Cause: The file specified by the *filename* parameter is not in the expected QMF data format.

Action: Verify that you have typed the name correctly. If correct, return to your QMF host session and type the command `EXPORT DATA TO` *`filename`* again. Be sure to export using the QMF data format.

SQL6009N The file exported from QMF contains Column <name> with a width <width> which is too long. The maximum column length is 4000 bytes.

Cause: The downloaded QMF file has a column with a width greater than 4000 bytes.

Action: Return to your QMF host session, rerun the QMF query without the named column, and export the data again. Then return the SQLQMFV or SQLQMFM command.

SQL6010N The downloaded QMF file has too many data columns.

Cause: The file being processed has more than 255 data columns.

Action: Return to your QMF host session and rerun a query, selecting 255 or fewer data columns. Export the data again, and then rerun the SQLQMFV or SQLQMFM command.

SQL6011N The data type <number> (<type-text>) of column <name> (positioned at column <number>) cannot be processed.

Cause: The QMF file contains a column with a data type that is not supported. SQLQMF does *not* support the following data types:

- LONG VARCHAR
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC.

Action: Return to your QMF host session and rerun the query without selecting the named column indicated. Then rerun the SQLQMFV or SQLQMFM command.

Appendix I. Edit Codes

The following is a series of tables defining each of the codes you can use to specify report forms and to specify panels. Edit codes are used to format data for displaying and printing. Every field must have an edit code. A blank is not valid for the edit code field.

Edit Codes for Character Data

The following table defines the edit codes for character data:

Edit Code	Definition
C	Makes no change to the display of a value. C is the default for character data. If the value cannot fit on one line in the column, the text is truncated according to the width of the column. If the value begins with blanks, the blanks are ignored.

CW	Makes no change to the display of a value, but if the value cannot fit on one line in the column, the text is wrapped according to the width of the column. For example, the following is displayed if the column width for LOCAT is 5:
	`DEPTNAME    LOCAT`
	`D01         New Y`
	`            ork`
	`D02         San F`
	`            ranci`
	`            sco`
	If the value begins with blanks, the blanks are ignored.

Edit Code	Definition
CT	Makes no change to the display of a value, but if the value cannot fit on one line in the column, the text is wrapped according to spaces (blanks) or full words, if possible. For example, the following is displayed if the column width for LOCAT is 5: DEPTNAME LOCAT D01 New York D02 San Franc isco If the value begins with blanks, the blanks are ignored.

Edit Codes for Numeric Data

The following charts define the edit codes for numeric data.

Note: The edit codes L, D, P, I, J, and K can be followed by a number (from 0 to 15) that indicates how many places to allow after the decimal point; if no number is specified, 0 places are assumed. Numbers with more places than what was specified are rounded; numbers with less spaces are padded with zeros.

Edit Codes for Numbers

Edit Code	Definition
E	Displays the numbers in scientific notation. For example, the number −1234.56789 is displayed as −1.234E+03, if the width of the column in the report form is 10. The E represents the exponent for the power of 10. E is the default for the data type scientific notation. If a leading plus + is entered, it is not displayed. A maximum of 15 digits are displayed in the report. One space is always reserved for a leading sign, although it does not display for positive numbers. There is always a sign and at least two digits after the E; if needed, there can be three digits. The number can have 15 significant digits and a value: • Of 0 (typed as 0E0 or 0.0E0) • In the positive range of +2.225E−307 to +1.79767E+308 • In the negative range of −1.79767E+308 to −2.225E−307. The positive and negative signs used within the number are always + and −, regardless of what is in your profile.

Edit Code	Definition
L	Displays the numbers in decimal notation with negative signs, if they occur (for example, L1 would specify showing one digit after the decimal point). The default Ln is used for all columns with the numeric data type of decimal. The value of n, for the default number of decimal places, is the same as in the column definition. The default L is used for all columns with the numeric data type of small integer or large integer.
D	Displays the numbers with currency symbols and thousands separators for groups of three digits, as well as decimal points and negative signs that occur (for example, D2 means two decimal places). The currency symbol and thousands separator are taken from the active profile.
P	Displays the numbers with percent signs and thousands separators for groups of three digits, as well as decimal points and negative signs that occur. A digit following the P, for example P1, indicates the number of decimal places.
I	Displays the number with leading zeros and negative signs, if they occur. A digit following the I, for example I3, indicates the number of decimal places.
J	Displays the number with leading zeros, if they occur. A digit following the J, for example J2, indicates the number of decimal places.
K	Displays the number with a negative sign and thousands separators, if they occur. A digit following the K, for example K3, indicates the number of decimal places.

The table that follows shows how the edit codes for numbers would format the number −1234567.885. For this example, the following is assumed:

- The column **Width** is set to **15** in the Form panel.

- The following values are set in the Profile panel:

  Decimal character is **Period** (.)
  Thousands separator is **Comma** (,)
  Rounding rule is **1-4 Down, 5-9 Up**
  Left currency symbol is **$**
  Left negative sign is **–**.

Examples of Edit Codes for Numeric Data

Edit Code	Lead Zeros	Negative Sign	Thousands Separators	Currency Symbol	Percent Sign	Display of -1234567.885
E	No	Yes	No	No	No	-1.23456788E+06
D2	No	Yes	Yes	Yes	No	-$1,234,567.89
I3	Yes	Yes	No	No	No	-0001234567.885
J2	Yes	No	No	No	No	000001234567.89
K3	No	Yes	Yes	No	No	-1,234,567.885
L2	No	Yes	No	No	No	-1234567.89
P2	No	Yes	Yes	No	Yes	-1,234,567.89%

Edit Codes for Date

The table that follows shows the edit codes for formatting date values. The x represents a separator used when printing or displaying the date, such as a slash (/), hyphen (-), period (.), or blank ( ); the separator cannot be an alphabetic or numeric character. The default edit code for date is based on the value specified for the Default Date Edit Code specified in the Profile panel for the active profile. For the U.S. country code 001, the default format is TDM–.

Edit Codes for Date

Edit Code	Format	Example
TDMx	MMxDDxYYYY	TDM- ==> 08-29-1988
TDYx	YYYYxMMxDD	TDY/ ==> 1988/08/29
TDDx	DDxMMxYYYY	TDD ==> 29 08 1988
TDYAx	YYxMMxDD	TDYA/ ==> 88/08/29
TDMAx	MMxDDxYY	TDMA- ==> 08-29-88
TDDAx	DDxMMxYY	TDDA. ==> 29.08.88

Edit Codes for Time

The table that follows shows the edit codes for formatting time values. The x represents a separator used when printing or displaying the time, such as a colon (:), period (.), or comma (,); the separator cannot be an alphabetic or numeric character. The default edit code for time is based on the value specified for Default Time Edit Code specified in the Profile panel for the active profile. For the U.S. country code 001, the default format is TTS.

Edit Codes for Time

Edit Code	Format	Note	Example
TTSx	HHxMMxSS	Includes seconds	TTS. => 16.42.35
TTCx	HHxMMxSS	Includes seconds, 12 hr clock	TTC: => 04:42:35
TTAx	HHxMM	Abbreviated (no seconds)	TTA, => 16,42
TTAN	HHMM	Abbreviated, no delimiter	TTAN => 1642
TTUx	HHxMM AM HHxMM PM	USA style	TTU: => 04:42PM

Edit Code for System Date and Time

The table that follows shows the edit code for formatting system date and time values. yyyy is year, the first mm is month, dd is day, hh is hour, the second mm is minutes, ss is seconds, and nnnnnn is microseconds. The column **Width** in the Form panel must be at least 19 so that the entire format, except for microseconds, can be displayed. If the **Width** is less than 26, the trailing digits of the microsecond part are truncated.

Edit Code for System Date and Time

Edit Code	Format	Example
TSI	yyyy-mm-dd-hh.mm.ss.nnnnnn	1988-08-29-16.42.19.123456

Appendix J. Writing Procedures Language 2/REXX Procedures

The Procedures Language 2/REXX interface has two functions that are callable from the Procedures Language 2/REXX: (referred to as Procedures Language 2/REXX in the remainder of this appendix).

- SQLEXEC provides the SQL interface into the relational database
- SQLDBS provides the interface into the Database Services environment, utility, and configuration commands.

This appendix explains these functions. For more information, refer to the *Database Manager Programming Guide and Reference*.

Connecting to a Database Manager Database

The following table shows the Procedures Language 2/REXX commands that are used to connect to a database or disconnect from a database.

The three figures on pages J-4, J-7, and J-10 illustrate a sample program. To use this example, you should enter and save the program into a single file. Each of the lines illustrated in the figures of this appendix are numbered. The lines are numbered to provide a reference point within the figure and should not be considered a part of the example file. You can run the program from the OS/2 command line by entering, for example: `SAMPDB.CMD`

Table J-1. REXX commands used to connect or disconnect from a database.

REXX Command	Description
SQLDBS START DATABASE MANAGER	This command starts the Database Manager background process and allocates resources. This command must be issued prior to attempting to connect to a Database Services database.
SQLDBS START USING DATABASE dbname	This command is used to connect an application to a database.
SQLDBS RESTART DATABASE dbname	This command is used to restart a database after the CONNECT command failed because the database must be restarted.
SQLDBS STOP DATABASE MANAGER	This command ends the Database Manager background process and releases resources.
SQLDBS STOP USING DATABASE	This command is used to disconnect an application from a database.

In order to call a Database Manager function from a REXX procedure, you must register the Database Manager functions. This is accomplished by using the REXX statement RXFUNCADD as shown on lines 6 and 8 in Figure J-1 on page J-4.

The START DATABASE MANAGER command is used to start the Database Manager background process and allocates resources for subsequent Database Manager requests. SQLCA is a global complex variable that contains return information from Database Services. SQLCA.SQLCODE contains the error code.

Lines 13 through 30 in Figure J-1 on page J-4 show a method for connecting to a database.

The START USING DATABASE command is used to connect the procedure to a database. The SAMPLE database used in the command is the database that can be created by issuing SQLSAMPL.EXE, which is shipped with Database Manager. Database Services activates the database and connects the application to it. If the database needs to be restarted, the SQLCODE −1015 is returned to the application. This allows the procedure to notify the user that restarting will take place. The RESTART DATABASE command is used to accomplish the restart.

Line 19 in Figure J-1 on page J-4 shows the usage of a global REXX variable that is set by Database Manager. The SQLMSG file contains the text of a message explaining the error.

Once the database is started, the example calls a REXX procedure, dotable(). Upon return from the procedure, the STOP USING command on line 39 is used to disconnect from the database. The STOP DATABASE MANAGER command on line 41 ends Database Manager and is included in the following example to show the command. Normally, a procedure would not issue the STOP DATABASE MANAGER command.

Since the .DLL modules associated with the SQLEXEC and SQLDBS functions require system memory, you may want to free this memory when not in use by issuing the REXX statement RXFUNCDROP.

Note: Exiting the REXX procedure does not automatically release the .DLL memory used by the SQLEXEC and SQLDBS functions.

```
/* ------------------------------------ */
/* Sample Database Manager procedure for */
/* connecting to a database              */
/* ------------------------------------ */
if RxFuncQuery('SQLDBS') <> 0 then
  rcy = RxFuncAdd('SQLDBS', 'SQLAR', 'SQLDBS')
if RxFuncQuery('SQLEXEC') <> 0 then
  rcy = RxFuncAdd('SQLEXEC', 'SQLAR', 'SQLEXEC')
sampl_rc = 0
call SQLDBS 'START DATABASE MANAGER'
if SQLCA.SQLCODE <> -1026 & SQLCA.SQLCODE <> 0
then signal ERROR
call SQLDBS 'START USING DATABASE sample'
if SQLCA.SQLCODE <> 0 then do
  if SQLCA.SQLCODE = -1013 then
    say 'Please create the SAMPLE database - SQLSAMPL'
  else do
    if SQLCA.SQLCODE = -1015 then do
      say SQLMSG
      say 'Restarting the database'
      call SQLDBS 'RESTART DATABASE sample'
      if SQLCA.SQLCODE <> 0 then
        signal ERROR
      else do
        say 'Starting the database'
        call SQLDBS 'START USING DATABASE sample'
      end
    end
  end
end

sampl_rc = dotable()

ERROR:
  sampl_rc = SQLCA.SQLCODE
  say SQLMSG
  signal FINISH

FINISH: call SQLDBS 'STOP USING DATABASE'
        say 'SAMPLJOB finished with rc = 'sampl_rc
        call SQLDBS 'STOP DATABASE MANAGER'
        rcy = RxFuncdrop('SQLEXEC')
        rcy = RxFuncdrop('SQLDBS')
        exit sampl_rc
```

Figure J-1. Example program using REXX commands.

Using SQLEXEC to Access a Database Manager Database

The following table contains the REXX commands that provide the SQL interface for accessing data in a database. The example in Figure J-2 on page J-7 shows how these commands are used.

Table J-2 (Page 1 of 2). REXX commands that provide SQL interface for accessing a database.

REXX Command	Description
SQLEXEC EXECUTE IMMEDIATE stmt	This command immediately submits the SQL statement to Database Manager.
SQLEXEC EXECUTE sname	This command runs an SQL statement processed through a previous PREPARE operation.
SQLEXEC DECLARE cursorname CURSOR FOR sname	This command reserves a predefined cursor name and statement name pair for a dynamic SELECT statement. There are 100 such pairs: C1 to C100 and S1 to S100.
SQLEXEC PREPARE sname FROM stmt	This command readies an SQL statement for a subsequent EXECUTE or OPEN operation.
SQLEXEC OPEN cursorname	This command positions a cursor at the first row of a result table defined by a PREPARE request. The table is retrieved through subsequent FETCH operations.
SQLEXEC FETCH cursorname INTO host-variable-list	This command obtains the next row from the result table of the SELECT statement that has been processed by an OPEN operation.
SQLEXEC CLOSE cursorname	This command closes a cursor that has been opened through a previous OPEN operation.

Table J-2 (Page 2 of 2). REXX commands that provide SQL interface for accessing a database.

REXX Command	Description
SQLEXEC COMMIT	This command makes permanent all changes made to the database since the transaction began or since the last COMMIT or ROLLBACK operation.
SQLEXEC ROLLBACK	This command backs out all changes made to the database since the transaction began or since the last COMMIT or ROLLBACK operation.

```
45. /* -------------------------- */
46. /* dotable internal procedure */
47. /* -------------------------- */
48. dotable: procedure
49.   creat_jobs = 'CREATE TABLE NULLID.JOBS(',
50.                  'JOB        CHAR(5) NOT NULL,',
51.                  'BASE_SAL   DECIMAL(7,2),',
52.                  'MAX_SAL    DECIMAL(7,2),',
53.                  'MIN_RAISE  DECIMAL(7,2),',
54.                  'MAX_RAISE  DECIMAL(7,2))'
55.   call SQLEXEC 'EXECUTE IMMEDIATE :creat_jobs'
56.   if SQLCA.SQLCODE <> 0 then signal ERROR
57.   call SQLEXEC 'COMMIT'
58.   if SQLCA.SQLCODE <> 0 then signal ERROR
59.
60.   sampl_rc = doimport()
61.   if sampl_rc <> 0 then signal ERROR
62.
63.   call SQLEXEC 'DECLARE c1 CURSOR for s1'
64.   if SQLCA.SQLCODE <> 0 then signal ERROR
65.
66.   sel_jobs = 'SELECT job,base_sal,max_sal,',
67.                      'min_raise,max_raise',
68.              'FROM NULLID.JOBS'
69.   call SQLEXEC 'PREPARE s1 FROM :sel_jobs'
70.   if SQLCA.SQLCODE <> 0 then signal ERROR
71.   call SQLEXEC 'OPEN c1'
72.   if SQLCA.SQLCODE <> 0 then signal ERROR
73.
74.   do while SQLCA.SQLCODE = 0
75.     call SQLEXEC 'FETCH c1 INTO :job, :base_sal, ',
76.                  ':max_sal, :min_raise, :max_raise'
77.     if SQLCA.SQLCODE = 0 then do
78.       line= job' 'base_sal' 'max_sal' 'min_raise' 'max_raise
79.       say line
80.     end
81.   end
82.   if SQLCA.SQLCODE <> 100 then signal ERROR
83.
84.   call SQLEXEC 'CLOSE c1'
85.   sampl_rc = 0
86.   return sampl_rc
```

Figure J-2. Example program using REXX commands.

This example creates a table in the SAMPLE database. The SAMPLE database is already connected as described in "Connecting to a Database Manager Database" on page J-1. The example then calls another REXX procedure to import rows into the table. Next, the example procedure retrieves the rows and displays them to the user.

The statement beginning at line 49 stores an SQL statement into the REXX variable CREAT_JOBS. The variable is used in the the EXECUTE IMMEDIATE command at line 55 resulting in the table called NULLID.JOBS to be created. The COMMIT command makes the table permanent.

The procedure, doimport(), is called to insert rows into the empty table.

Lines 63 through 72 in Figure J-2 on page J-7 declare and open a cursor on the table. The DECLARE statement reserves a cursor name (c1), and statement name (s1) pair. The statement name, s1, is used on the PREPARE request and the cursor name, c1, is used on the OPEN and FETCH requests.

The actual SELECT statement required to indicate that rows and columns are to be returned, is stored in the variable SEL_JOBS on lines 66 through 68. The variable SEL_JOBS is passed to SQLEXEC on the PREPARE statement. The OPEN command positions the cursor on the first row.

Lines 74 through 81 in Figure J-2 on page J-7 perform the FETCH loop to obtain and display the rows.

A SQLCA.SQLCODE of 100 is returned if there is no row to return. This may happen after the last row is returned or when there are no rows available to satisfy the request.

The example issues a CLOSE command to close the cursor and return to its caller on lines 84 through 86.

Importing and Exporting Data from a Database Manager Database

The following table contains the REXX commands that can be used to export data from database tables into OS/2 files or to import data from OS/2 files into database tables. The example in Figure J-3 on page J-10 shows how procedures can be written to import data into a table and to export data from a table into an OS/2 file.

The import procedure, doimport (), starting at line 87 in the example in Figure J-3 on page J-10 is called by the procedure dotable() on page J-7. The procedure assumes that a file named JOBS.DEL exists on the current directory and contains the following rows of data:

```
"Mgr",17000,35000,.05,.10
"Sales",15000,75000,.00,.15
"Clerk",9000,12000,.03,.07
"Temp",7000,12000,.00,.00
```

The file type for this file is DEL. The OS/2 file formats supported by Database Manager for import and export are:

DEL Delimited ASCII, for exchange with one of the following:

- dBase II or dBASE III
- BASIC programs
- IBM Personal Decision Series
- DB2 and SQL/DS (import only).

ASC Nondelimited ASCII (import only) for importing data from other applications; for example, from the IBM DisplayWrite Series and IBM Personal Editor.

WSF Work-Sheet Formats, a small subset of the formats supported in common by the following Lotus products:

- Lotus 1-2-3, versions 1.0, 1A, and 2.0
- Symphony, versions 1.0 and 1.1.

IXF Personal Computer version, the preferred method for exporting a table so that it can be imported later into the same table or into another Database Manager table.

Lines 91 through 93 in Figure J-3 on page J-10 contain the call to import the data.

CON is used to direct the import messages. This causes the messages to be displayed on the screen.

Table J-3. REXX command used to import data.

REXX Command	Description
SQLDBS IMPORT TO dbname FROM datafile OF filetype {INSERT | REPLACE | CREATE | REPLACE_CREATE} INTO tname MESSAGES msgfile	This command inserts data from an OS/2 file into a Database Manager table or view. For import of formats other than PC/IXF, the Database Manager table or view must already exist at the time the utility is started. The data types in the OS/2 file fields are converted into the type of the corresponding field in the Database Manager fields. The message file can be an OS/2 file name or a device name like CON or LPT1.

```
/* -------------------------- */
/* doimport internal procedure */
/* -------------------------- */
 doimport: procedure
    call SQLDBS 'IMPORT TO SAMPLE FROM',
                 'jobs.del OF DEL INSERT',
                'INTO NULLID.JOBS MESSAGES CON'
    sampl_rc = SQLCA.SQLCODE
    if SQLCA.SQLCODE <> 0 then say SQLMSG
    return sampl_rc
```

Figure J-3.

Managing the Database Manager Database Environment

Table J-4 on page J-12 contains some of the Database Manager REXX commands that can be used to manage the database environment.

The CREATE DATABASE command can be written as follows:

```
/**/
   call SQLDBS 'CREATE DATABASE depta ON D WITH "Accounting database"'
   if SQLCA.SQLCODE <> 0 then say SQLMSG
   return SQLCA.SQLCODE
```

This will create a database called *DEPTA* on drive D:. The information following WITH will be stored in the Database Manager system database directory as the comment for the database.

The MIGRATE DATABASE command can be written as follows:

```
/**/
   call SQLDBS 'MIGRATE DATABASE sample PASSWORD ironman'
   if SQLCA.SQLCODE <> 0 then say SQLMSG
   return SQLCA.SQLCODE
```

If the user has applications that access databases on another workstation, the CATALOG DATABASE and CATALOG NODE commands can be used to set up the catalog and node directories as follows:

```
/**/
   call SQLDBS 'CATALOG DATABASE depta AS account AT NODE deptsvr',
        'WITH "Accounting Database"'
   if SQLCA.SQLCODE < 0 then say SQLMSG
   else do
     if SQLCA.SQLCODE = 1100
     then do
       call SQLDBS 'CATALOG NODE deptsvr'
       if SQLCA.SQLCODE <> 0 then say SQLMSG
     end
   end
   return SQLCA.SQLCODE
```

The DEPTA database is located on node DEPTSVR. The example assumes that a set of Communications Manager Advanced Program-to-Program Communications Interface (APPC) profiles exist where a local APPC logical unit profile is set up with the default logical unit (LU) set to YES, an APPC partner LU profile has a partner LU alias set to DEPTSVR, and the partner LU use an APPC transmission service mode profile with a mode name of OS2DBMRS. These are the defaults used by Database Manager when only the NODE parameter, DEPTSVR, is used with the CATALOG NODE command.

Table J-4. REXX commands used to manage the database environment.

REXX Command	Description
SQLDBS CREATE DATABASE dbname [ON drive] WITH comment]	This command is used to start a new database.
SQLDBS DROP DATABASE dbname	This command is used to delete a database.
SQLDBS MIGRATE DATABASE dbname [PASSWORD password]	This command is used to convert Database Manager Version 1.0 or Version 1 .1 databases to Version 1.3 databases.
SQLDBS CATALOG DATABASE dbname [AS alias] {ON drive | AT NODE nodename} [WITH comment]	This command is used to store information about other databases to which the user may want to connect. These are databases that were created on a Database Manager workstation. This command is used to catalog the database in the database directory.
SQLDBS CATALOG NODE nodename [LOCAL local_lu] [REMOTE partner_lu] [MODE mode] [WITH comment]	This command is used to create a node name entry in the node directory file.
SQLDBS UNCATALOG DATABASE dbname	This command is used to delete a database entry from the database directory file.
SQLDBS UNCATALOG NODE nodename	This command is used to delete a node name from the node directory file on a file system.

Return Codes

REXX return codes fall into the range 7001 through 7999. The following return codes may be returned by REXX in SQLCA.SQLCODE, as well as the Database Manager return codes described in the *Database Manager Programming Guide and Reference*.

−7001SQL7001N Unknown command "<command>" was requested.

Explanation: The command submitted to REXX could not be recognized.

System: The command cannot be processed.

Programmer: Verify that the command is a valid SQL command and run the procedure again.

Note: All commands must be in uppercase.

−7002 SQL7002N The cursorname is invalid.

Explanation: The user has specified an incorrect cursor name.

System: The command cannot be processed.

Programmer: Verify that the cursor name is of the form c1 through c255.

−7003 SQL7003N The statement name is invalid.

Explanation: The user has specified an incorrect statement name.

System: The command cannot be processed.

Programmer: Verify that the statement name is of the form s1 through s255.

−7004 SQL7004N The syntax of the request is invalid.

Explanation: REXX was unable to parse the command string submitted.

System: The command cannot be processed.

Programmer: Verify that the proper command syntax was used.

7005 SQL7005W Stmt is an OPEN, and a DECLARE was not done.

Explanation: An OPEN statement was attempted without a cursor having been declared.

System: The command is processed.

Programmer: Insert a DECLARE statement prior to the OPEN statement, and run the procedure again.

−7006 SQL7006N Invalid keyword "<keyword>" given on "<request>".

Explanation: The statement contains a keyword that is not valid.

System: The command cannot be processed.

Programmer: Specify the correct format of the keyword.

−7007 SQL7007N REXX variable "<variable>" does not exist.

Explanation: A REXX variable was passed that did not exist in the REXX variable pool.

System: The command cannot be processed.

Programmer: Verify that all variable names in the host-variable list are assigned prior to the command that failed. Then run the procedure again.

–7008 SQL7008N Invalid REXX variable "<variable>."

Explanation: A variable was passed to REXX that contained inconsistent data.

System: The command cannot be processed.

Programmer: Verify that the data and lengths contained in the SQLDA command have been assigned correctly and run the procedure again.

–7009 SQL7009N Truncated REXX variable "<variable>."

Explanation: The variable that was passed to REXX contained inconsistent data. The data string from the variable was truncated.

System: The command cannot be processed.

Programmer: Verify that the data length is consistent with the length given in the input SQLDA command and run the procedure again.

–7010 SQL7010N Invalid scan ID "<variable>."

Explanation: The scan ID was passed to REXX, but it did not exist or, contained inconsistent or missing data.

System: The command cannot be processed.

Programmer: Verify that the data contained in the scan ID has been assigned correctly, and run the procedure again.

–7011 SQL7011N Required parameter "<parameter>" not given.

Explanation: The parameter is required by the REXX command syntax but was not specified.

System: The command cannot be processed.

Programmer: Specify the required parameter value and run the procedure again.

–7012 SQL7012W Attempted ISL change while connected to a database

Explanation: The isolation level cannot be changed while connected to a database.

System: The command is ignored.

Programmer: If a change in the isolation level is necessary, issue a DISCONNECT command for the current database; then set the isolation level and connect to the database again.

–7013 SQL7013N Mismatched cursor or statement name or hold attribute.

Explanation: In REXX, the cursor name and statement name must be of the form C*NN* and S*NN*, where *NN* is a number from 1 to 255, and these numbers must be the same for a cursor or statement pair. Also, C1 through C127 are declared without hold, C128 through C255 with hold.

System: The command is ignored.

Programmer: Verify that the cursor and statement numbers match; then run the procedure again.

–7014 SQL7014N Incorrect number of host variable components.

Explanation: In REXX, the first component of a compound host variable lists a number that is not equal to the actual number of components defined.

System: The command is ignored.

Programmer: Verify that the number in the first component matches the number of components actually defined and then run the procedure again.

–7099 SQL7099N Internal error "<error>" has occurred.

Explanation: A REXX internal error has occurred.

System: The command cannot be processed.

Programmer: Verify that REXX is properly installed; then run the procedure again. If the error persists, record the return code, record all error information from the SQLCA, and contact your service coordinator.

Appendix K. Concurrency Considerations in Designing Panels

This appendix discusses concurrency considerations when designing panels for a multi-user environment. As the designer, you should take into account these concurrency considerations when designing your database and customized applications.

Controlling Concurrency

Database Services allows access to the same data at essentially the same time. Concurrent multi-user access to Database Manager is supported by three *isolation levels* that ensure data integrity through locking. The three isolation levels are: *cursor stability, repeatable-read,* and *uncommitted read.* The isolation level controls record level locking and maintains the consistency of data. See the *Database Manager Administrator's Guide* for a discussion of each of these isolation levels.

Record level locking guarantees that a transaction maintains control over a database record until the transaction is completed. Record level locking prevents another application process from making changes to a record before the prior change is committed. Committing a change releases locks on a record and allows other application processes to access the previously locked record.

It is recommended that you consider the indexes specified on the tables you select when designing a panel. Indexes on tables greatly affect the achievable concurrency when running customized applications in a multi-user environment. Query Manager will normally commit the changes on each panel instance as a unit when the instance is ended by Next, Change and Next, or Add and Next, or by Search, Extended Search, or Quit Panel operations. This is called the *Auto-commit* mode of panel operation.

In Add mode, locks are held during the Add and Next and Add and keep operations. In Change mode, read locks are held on the rows out of the root and subtables while the instance is being displayed. These locks may be escalated to change locks during the Change and Next or the Delete and Next operations.

There are three elements that add complexity to the locking picture:

- You can introduce a BEGIN WORK command, which disables the auto-commit mode. Any read or change locks that are acquired will be held until the scope of the BEGIN WORK transaction is ended. (Refer to the *OS/2 Command Reference* for a discussion of this command.)
- To ensure data integrity, auto-commit is suspended while a Change mode panel is displayed; in other words, a panel being run from a Change mode will not perform the COMMIT command. For this reason, the END WORK command is not allowed, either directly or indirectly, as the result of an action on a Change Mode panel. The scope of time when locks are held can be much greater if the you allow a Change Mode panel to be stacked.
- The use of indexes can greatly effect the locks. When running panels in the Change mode, consider having indexes for:
  - Columns used to search on the root row
  - The connecting column in the subtable
  - Any connecting column in the lookup table
  - The subtable key column or columns.
- When running panels in the Add mode, consider having indexes for:
  - The root table connecting column if the relationship is defined as One-to-One or One-to-Many
  - The subtable key column or columns
  - Any connecting columns in the lookup table.

It is strongly recommended that the ERASE command not be used when auto-commit is suspended. Running the ERASE command on a procedure, query, form, panel, or menu will cause all other users to be unable to obtain lists of Query Manager objects.

Unnecessary nesting of panels or use of BEGIN WORK commands, or the absence of needed indexes, can result in a severe loss of concurrency.

Index