TRIMqmr
Users Guide

May 30, 2017

Trifox Inc.

www.trifox.com

Trademarks

TRIMapp, TRImpl, TRIMqmr, TRIMreport, TRIMtools, GENESISsql, DesignVision, DVapp, DVreport, VORTEX, VORTEXcli, VORTEXc, VORTEXcobol, VORTEXperl, VORTEXjdbc, VORTEX++, VORTEXJava Edition, LIST Manager, VORTEXodbc, VORTEXnet, VORTEXclient/server, VORTEXaccelerator, VORTEXreplicator are all trademarks of Trifox, Inc.

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

Copyright

Printed in the U.S.A.

Contents

Preface

Purpose

This document explains how to use TRIMqmr, Trifox's query manager and report writer, to retrieve information and create reports from data in relational databases. It does not discuss managing the database. For instructions and information about these tasks, read your database vendor's documentation.

Audience

TRIMqmr is designed for end users who have some familiarity with computers. If you are reading this manual, you are probably an analyst, manager, or administrative assistant who knows how to use the computer at your desk.

While this document does not assume that you are a database expert, understanding the information you are working with is important.

Using the Exercises

The exercises in this guide provide a fast start by explaining basic concepts while illustrating screen elements. Because you are guided through the menus and commands to create true-to-life examples, you get the most benefit by actually working through the exercises, not just reading them.

The online help, used with the exercises, provides quick assistance and solutions to common problems. The exercises use sample tables that were created by Trifox, installed with the TRIMqmr application, and integrated with your database management system.

The database examples are based on a fictitious wholesale company, and include four tables:

- ***CUST*** --- Customers.
- ***ORDS*** --- Orders placed by customers.
- ***ORDITEM*** --- Details of items ordered.
- ***INVENT*** --- Company's inventory.

Background

Trifox Inc. has been serving the relational database market since 1984 through consulting and the development of software products. In 1987, Trifox created SQL*QMX for Oracle. This easy-to-use, powerful querying and report writing tool, which is based on IBM's QMF, continues to be used at thousands of sites. In 1989, Trifox created TRIMtools, a family of application and reportwriting tools now known as DesignVision. DesignVision was developed in response to the OLTP requirements of several large application vendors.

Database Access

VORTEX is an integrated family of products that allows nearly any production application to access SQL data:

- On any or all of the major relational databases.
- Across networks.
- Across platforms.
- With a dramatic increase in the number of concurrent users.
- Without any additional hardware.

In a client/server or multi-tier configuration, VORTEX makes it possible for your SQL applications to access data on different platforms over one or more network configurations. Currently it supports only TCP/IP.

Inherent in this approach are services that allow production applications originally written for one relational database (such as Oracle) can access the same data on another database (such as Informix), even if it is spread across different databases.

VORTEX Precompilers for C and COBOL, as well as a variety of program interfaces, allow existing SQL programs to take full advantage of VORTEX services such as performance enhancement, transaction monitoring, and flat-file database access.

With VORTEXaccelerator in your configuration, you dramatically increase the number of concurrent users who can log on to a specific SQL production application. Your users experience faster performance and you won't have to change any programs or add any hardware.

Application and Report Development

DesignVision DVapp lets you design, generate, and maintain forms-based applications. You can easily port the pop-up windows, customizable menus and submenus, and custom keyboard assignments, in fact the entire application, to Windows .NET, Unix, OpenVMS, or HTML5 with no extra effort.

The reportwriter, TRIMreport, lets you create simple reports quickly, or complex reports with absolute confidence in their power.

When you want to write stand-alone applications (including triggers) without a user interface, the TRIMpl 4GL language gives you the freedom you want. The procedural language has over 100 database-specific functions that help you write powerful applications in very little time.

Reaching Legacy Data

GENESISsql is a SQL processor that accesses low-level data sources such as ISAM, SDMS, ADABAS, RMS, and MicroFocus and makes the data accessible to VORTEX clients. You can add GENESIS data sources to a VORTEX system in a matter of days, simplifying what used to be an enormous task.

Conventions

Screen shots in this manual come from the Windows version of our software.

Trifox documentation uses the following conventions for communicating information:

Example	Describes
`CHOOSE REPORT > [F3] >`	Press [F3] on the CHOOSE REPORT menu and ...
Right-click	Clicking the right mouse button.
Left-click	Clicking the left mouse button.
`connect_string`	Replace italicized text with your own variable.
`vtxnetd`	Text in bold typewriter style represents strings that you type exactly as they appear in the manual.

Support

If you have a question about a TRIFOX product that is not answered in the documentation (paper or online), contact the Customer Support Services group at:

- support@trifox.com
- Trifox Customer Support Services
  2959 Winchester Boulevard
  Campbell, CA 95008
  U.S.A.
- 408-796-1590

Chapter 1 Introduction

TRIMqmr is an ***ad hoc*** querying and reporting tool. You can access database information in real time when you need it without running a long procedure that may perform many other functions. In addition, you can run pre-defined queries against a database either in real-time or in batch mode.

This chapter:

- Introduces TRIMqmr.
- Describes database concepts and terms used in database access.
- Describes how to start TRIMqmr and lists the options.
- Includes a short exercise that starts TRIMqmr and connects to a database.

What is a Database?

A database is an organized collection of data typically stored in electronic format. Databases can be "flat files," relational, or object databases.

A ***relational database management system***, or RDBMS, allows you to organize information and indicate ***relationships*** between different items. Thus, you can combine, calculate, and retrieve information based on a question, rather than on a directed path of access for flat files.

The Relational Approach

Relational databases keep the data in ***tables***, which helps you visualize, understand, and use the information.

You use ***queries*** to retrieve and edit information in tables.

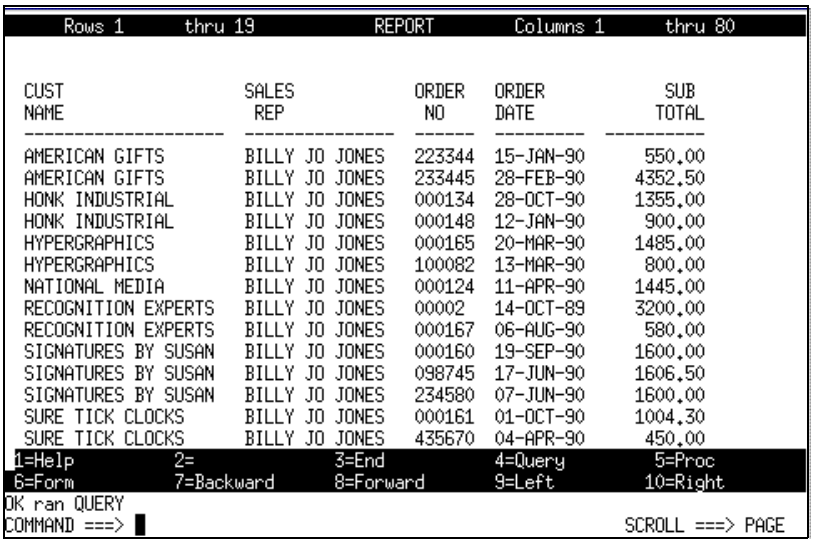
Tables in a relational database (hereafter simply called database) are made up of ***rows*** and ***columns***. Each column contains a type of information and each row contains a ***record*** of information.

Rows comprise ***fields***, one for each column, and each field holds one ***value*** or item of information.

For example, in one of the sample tables, called INVENT:

- The five columns are called PART_NO, DESCRIPTION, UNIT_PRICE, ON_HAND, and SOLD_YDT.
- The values in this table are both numeric and character.

This is a simple table.



Each column can contain a particular ***type*** of data (often called ***datatype***). The datatypes available depend on the database in which the information is stored. Typically, you can have character, number, or datetime data.

Data versus Information

Databases are critical to nearly every application we use today, from word processors' spell checkers to address books, to even today's most sophisticated World Wide Web sites. People store data, and retrieve it to save time and resources. Studying information in database reports can help people make better decisions about a variety of tasks.

This retrieval and examination is what databases are really all about.

Typically the elements in the fields, rows, and columns, are called the ***data***. ***Information*** is what you get, and create, when you are able to find the answer to a question you have, or answer other peoples' questions, using what's in the database.

You can define relationships between items in a table, or between items in different tables, whenever they seem helpful. The ability to relate one table to another is very important. It lets you organize information in separate, manageable units — so the inventory information for a product is stored separately from the details of the customers' addresses who order it.

You begin by indentifying a table name (you have already identified the database when you attached to it) and then requesting "presentation" of some stored data. Usually you refine your searches to limit the amount of data they return to you. This process of refinement is necessary if your query tool is not very efficient, if you have limited storage space, or if you must remain connected to the database.

Because TRIMqmr is fast, and does not need to remain connected to the database, however, you can retrieve the greatest number of records and refine your report "off-line."

Query By Example (QBE)

Query By Example — QBE — is a nonprocedural language you can use to select and retrieve data. You simply indicate which columns you want to display and how you want the data in each column presented.

You can select data based on:

- Columns from one or more tables.
- Conditions under which columns of data are to be retrieved:
  i. Logical operators.
  ii. Expressions involving one or more columns of data.
  iii. A range of values.
- Sorting data in ascending or descending order.
- Joining two or more tables.

QBE also lets you insert, update, and delete data.

Structured Query Language (SQL)

Structured Query Language (SQL), a widely used fourth-generation language (4GL) developed to access and manage database systems, has become the industry standard. TRIMqmr's SQL facilities conform to the standards proposed for relational database systems by the American National Standards Institute (ANSI).

Before You Begin

Before you use TRIMqmr it must be installed. In a multi-user system your ***database administrator***, or DBA, normally performs all the tasks of installing and configuring a database and copying demonstration files to the system.

If you have a single-user system, or you are the DBA, follow the installation instructions for your system that accompany the product release. Because the tool works on a wide variety of client platforms, installation instructions can vary from machine to machine.

With TRIMqmr you can issue a query against your database once, save the data to a file, and do your report refinement, processing, and design without issuing queries against a database, or being connected to a database. This flexibility is particularly useful for people working at sites that require remote connections to a database, or who are mobile and want to be productive with a laptop that cannot store the entire contents of a database.

Of course, you must always keep in mind that the data you are working with is essentially a snapshot and evaluate the quality of your work based on the necessity for the most up-to-the minute information.

Getting Started

All the exercises in this manual assume that you have access to a computer or terminal with TRIMqmr properly installed.

Sample Tables

The exercises use a database that is supplied with the software and available on the Trifox FTP site as `TUT_QMR` with an extension appropriate to your database operating system.

The exercises rely on the database being in its original state. If your results differ from those in the manual, changes may have been made by other users or your database or operating system may alter the appearance.

You can recreate the database by running `TUT_QMR.PRC`, according to the instructions in `README.NOW` that accompanies the files. This procedure creates the four tables used in the exercises, populates them, and saves the new tables in your database. The tables, and the datatypes they use are:

Table	CHAR	NUMBER	DATE
CUST	X	X	
INVENT	X	X	
ORDITEM	X	X	X
ORDS	X	X	X

To get data for the series of exercises, you must connect to a database. You can use the function key specified on the **Home** screen or type the command.

You can start TRIMqmr without connecting to a database, or you can specify parameters on the command line at start up.

Starting TRIMqmr

To start the program, type the operating system command (or click on the icon). To connect to a database, you must have a connect string, which specifies the database to which you want to connect. Contact your DBA if you have any questions about the connect string, your authority to read and write to the database, and the location of the sample tables at your site.

TRIMqmr's options at start up are:

- **-rp [*proc_name*]** — Runs a procedure named *proc_name.*
- **-ip [*filename*] —** Runs a procedure imported from *filename.*
- **-rq [*query_name*] —** Runs a query named *query_name.*
- **-iq [*filename*] —** Runs a query imported from *filename.*
- **-v [*n*] / [*m*] —** Specifies sizes of data and report buffers.

  n is the number of 4KB pages. The minimum number of pages is 4, the maximum is 2,147,483,647, and the default is 48.

  m is the number of *n* kept in main memory. The minimum number is 1, the maximum number 16, and the default value is 1.
- **-o [*filename*] —** Instructs that the key settings are output or saved to *filename.*
- **-k [*filename*] —** Instructs that they key settings are played back, or input from *filename.*

TRIMqmr Beginning Tutorial

For this exercise, you'll start TRIMqmr and connect to a database in the second step.

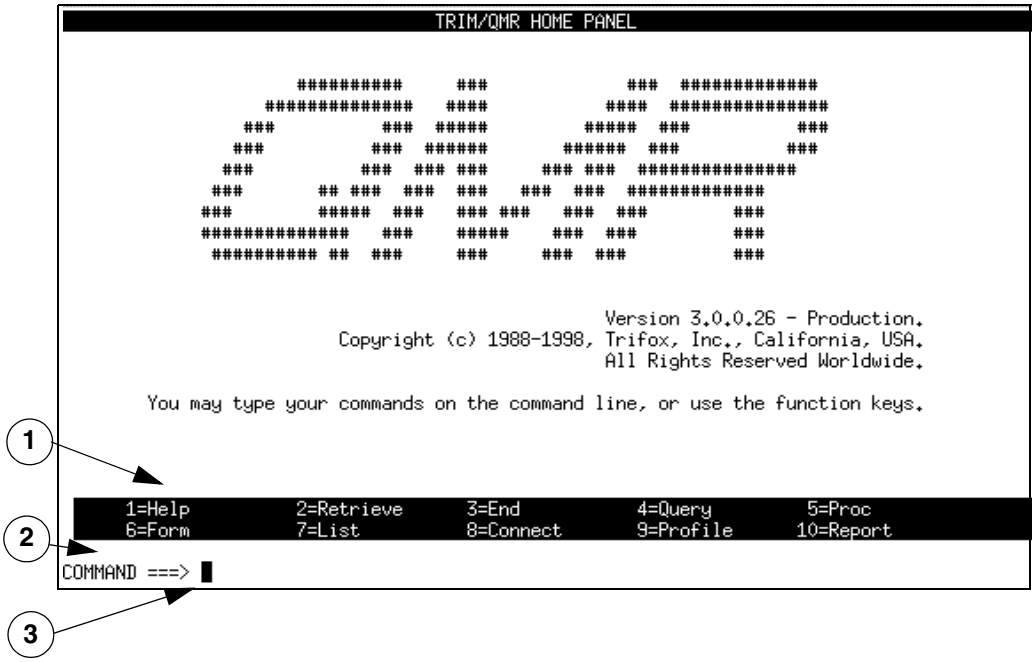
Exercises typically specify the function key, but you can type the command on the command line if you find it more convenient. Function keys and typed commands often are interchangeable in TRIMqmr. Using the function key means you don't have to move the cursor from the workspace to execute (type) a command; however, typing gives you access to commands not explicitly enumerated on a given screen.

Starting TRIMqmr

1. At the system prompt, type **`trimqmr`** > [Enter].

   TRIMqmr displays the **Home** screen, your point of entry to the query and reporting facilities.

   If you want to see complete descriptions for each of the function keys show, [F1] at any time or check in the command reference (Chapter 3) of this guide.



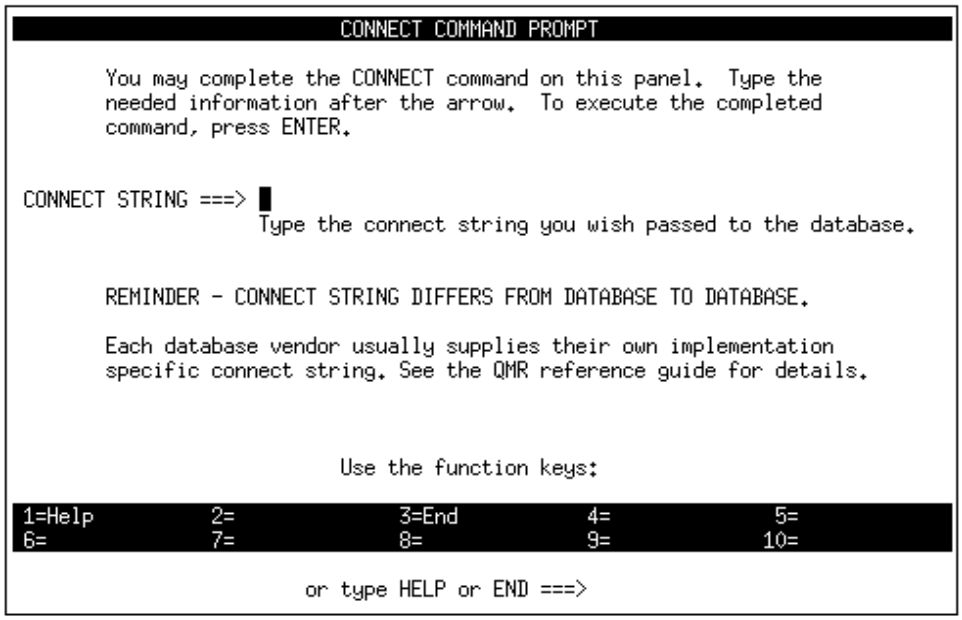
The Home screen has three parts:

1. ***Function keys*** — Provide single-key shortcuts to typing commands. Each screen has its own set of function keys, which appears at the bottom of the screen.
2. ***Message line*** — Shows instructions, confirmations, and current program status. A typical message might say `"OK DRAW performed."`
3. ***Command line*** — Receives your input (typed commands). All commands are available via typing.

Connecting to the database

If you know the syntax of the entire connect string, you can type it when you start TRIMqmr. This exercise demonstrates using the help screens.

1. If TRIMqmr is not yet running, type **`trimqmr`** on the command line.
2. [F8] > Connect screen.



3. Type your connect string in the field. (You may need to consult your DBA to acquire this information.)

   Once you type the string correctly, and [Enter], TRIMqmr responds with the message `"OK CONNECT performed"` on the status line.

Navigating

In addition to the function keys shown at the bottom of each screen, you can use the following keys to navigate through TRIMqmr.

Help is only an [F1] away, too. And if you forget the syntax of a command, you can execute the following action to view the help screen for that command.

- Type **`command-name ?`** [Enter]

In addition to the 10 function keys that may be active for each screen you can use other keys to navigate through TRIMqmr screens:

To …	Press …
Scroll	[PgUp] and [PgDown] or button commands on screen.
Move cursor or pointer	Arrow keys

To …	Press …
Move cursor from workspace to command line	[Home]
Move cursor to end of report	[End]
Move cursor to the next field	[Tab]
Move cursor backward through fields	[Shift][Tab]
End session with confirmation	[F3] or type `END`
End session immediately	Type `EXIT`

You can change the function keys' behaviors if there is another way you want them arranged. See "***Key Mapping***" on page 131 for information about customizing your environment.

If you are ready for a break, you can quit TRIMqmr ([F3]) and confirming the session end. If not, turn the page.

Chapter 2
Using Query By Example

This chapter introduces Query By Example (QBE), one of two query languages available in TRIMqmr. QBE provides a visual representation of database tables with a picture that shows columns with headings and spaces below each column head for command and data entry.

If you are familiar with QBE and simply want to review the commands available and their syntax, skip to Chapter 4, ***QBE Controls***.

Basics

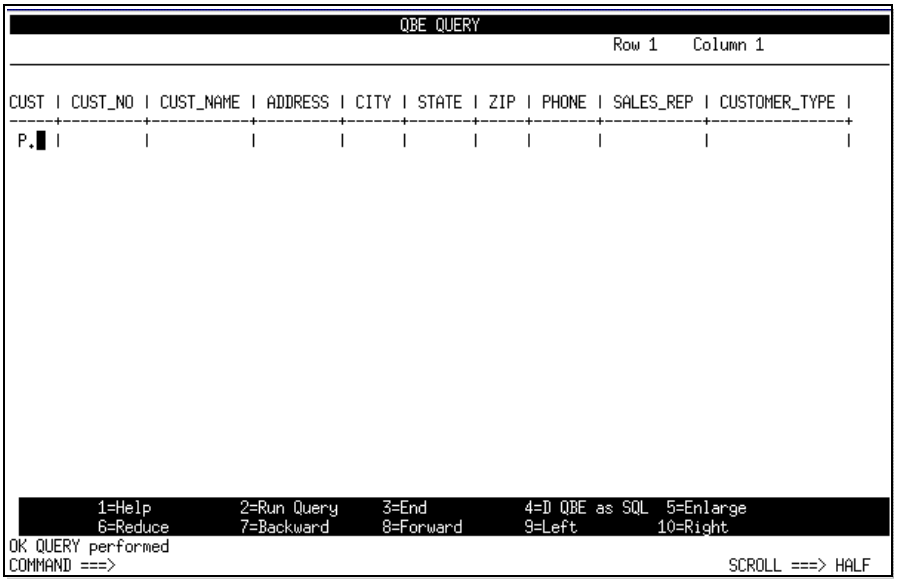
QBE allows you to define the query in a step-by-step process in which you execute the query to determine the results, save the query at any point, and continue to refine the data selection. The results of a QBE query are displayed on the Report screen.

You have several options for issuing TRIMqmr commands (including QBE commands). You execute QBE commands by typing them on the command line and pressing [Enter] or, in many cases, by pressing a single function key, such as [F2]. The exercises in this chapter do not give all options for each command, but focus on either function key or typing command options to familiarize you with two alternatives.

Be aware that the function keys may be customized for your site. If you press a function key and do not get the behavior you expect, you may want to consult your DBA, or other IT professional, to see if someone has modified the key file.

The Screen

The QBE screen, accessed from the Home screen by [F4], has the following function keys, unless they have been redefined at your site. The illustration shows the table outline that results from issuing the command `DRAW ORG`.



Function Keys

Command	Key	Description
Help	[F1]	Displays the Help screen for the QBE Query screen.
Run Query	[F2]	Displays the Report screen with the current QBE query results in the default or current report format. If there is an existing report format, the default format prompt is displayed. To display the query in its default format, respond YES to the prompt; to return to the QBE Query screen, respond NO to the prompt.
End	[F3]	Displays the Home screen.
D QBE as SQL	[F4]	Displays the current QBE query as a SQL query. Changing the SQL Query screen with this command does not change the Language setting on the Profile screen.
Enlarge	[F5]	Increases the width of QBE object, or adds a column or row.
Reduce	[F6]	Decreases the width of a QBE object, or deletes a column or row.
Backward	[F7]	Displays the portion of the QBE query above the portion currently displayed; if the work area displays the top of the query, a message is displayed on the status line.
Forward	[F8]	Displays the portion of the QBE query below the portion currently displayed; if the work area displays the bottom of the query, a message is displayed on the status line.
Left	[F9]	Displays the portion of the QBE query to the left of the portion currently displayed; if the work area displays the leftmost portion of the query, a message is displayed on the status line.
Right	[F10]	Displays the portion of the QBE query to the right of the portion currently displayed; if the work area displays the rightmost portion of the query, a message is displayed on the status line.

The Exercises

This exercise takes approximately an hour to complete.

Begin at the Home screen. If you do not remember how to start TRIMqmr, refer to "***Starting TRIMqmr***" on page 8.

You can quit the TRIMqmr application at any time with [F3] until you see the Home screen, and then a "Do you really want to exit?" prompt screen. (If you are used to a Windows environment, you may think of these screens as dialog boxes.)

To start from the beginning, simply type `RESET` *`obj_name`* [Enter]. This command clears any commands in process and returns you to a blank screen.

NOTE: *The instructions in this document may refer to a command by its name, and omit the "type COMMAND on the command line, and [Enter]" detail. Be assured that "entering" and "typing" commands are the same.*

Setting profile

You can set (or change) a variety of preferences in the Set Command prompt (or Profile) screen. For this exercise, it's only important that the `LANGUAGE` is QBE.

Before beginning the exercise, make sure that you are in QBE mode.

1. [F9].

   Your screen should look like this:

```
                          SET COMMAND PROMPT

    CASE     ===> STRING   (UPPER, MIXED, or STRING)
    DECIMAL  ===> PERIOD   (PERIOD, COMMA, or FRENCH)
    CONFIRM  ===> NO       (YES or NO)
    LANGUAGE ===> qbe      (SQL or QBE)

    WIDTH    ===> 132      Number of characters per line for printing.
    LENGTH   ===> 66       Number of lines per page for printing.

    PRINTER  ===>                              Printer used for output.

  WARNING - These options are not usually changed. You should modify them
            only if you know what the resulting effect will be.

    SPACE ===>
             Type the name of the dbspace in the database to save DATA in.

                          Use the function keys:

1=Help          2=              3=End           4=              5=
6=              7=              8=              9=              10=

                       or type HELP or END ===>
```

If you need to change the setting,

2. Move your cursor down through the fields and type `QBE` over `SQL`. [Enter].

   The Profile is saved ***for this session only***. If you want to change the Profile permanently, type `SAVE PROFILE` on the command line after you [Enter].

About Upper and Lower Case Characters

The setting for CASE specified in your profile determines whether lower case letters are distinguished from UPPERCASE. The setting should match your database's setting for case-sensitivity.

Case	Description
UPPER	All input is translated into uppercase. Lowercase letters are ***not*** distinguished from uppercase.
String	Input is translated into uppercase except for character strings enclosed in single or double quotation marks (They must agree with the case of the database string itself.); comments in SQL queries, QBE queries, or procedures; text fields in FORMs.
Mixed	No input is translated into uppercase. Lowercase letters are not the same as uppercase.

*NOTE: For consistency, all commands appear in uppercase in this chapter. If CASE is set to MIXED, you **must** type commands in uppercase for them to work. If your profile is set to UPPER or STRING, you can enter commands in any case you want.*

Formulating Queries

Simple queries let you to select all rows and columns, specific columns, or specific rows, from a table. The more simple the query, usually, the larger the number of rows you get back (also called ***result set***).

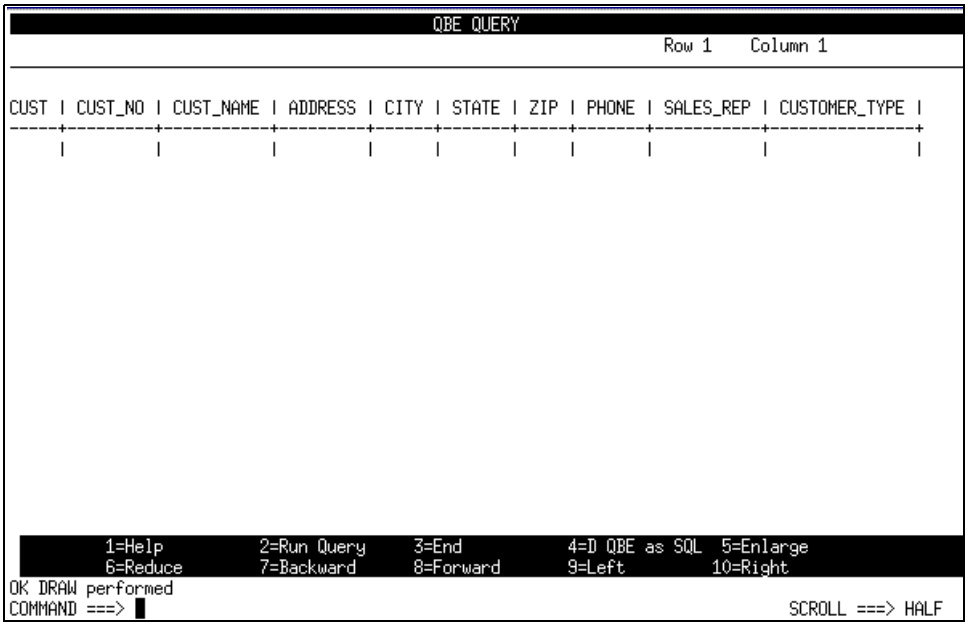
In general, simple queries do not require calculated values or produce summary data, sort records into groups, or join two tables. These queries are called "Complex Queries."

Even in simple queries, however, you can use single or multiple search conditions to define your query result to the appropriate level of detail. You can enter most search conditions under the column heading; however, a few conditions, as well as multiple conditions, require a ***conditions box***.

Creating and running a query

QBE queries must start on the QBE Query screen. To display the information in a table you must first see the column heads of that table.

1. [F4] to view the Query (QBE) screen.
2. Type **`DRAW CUST`** on the command line and [Enter].



Selecting the whole table

To construct the simplest query, select all the rows and columns of CUST table using the QBE command PRESENT.

When you get the results of your query, notice the new screen "Report." Query results from QBE and SQL both appear on a Report screen. This is also the screen that shows you the results of the formatting decisions you make on the Forms screens.

When you've completed the query, your screen shows the CUST table and the system message `"OK RUN Query"`. Only part of the table is visible. When a table display includes more columns than can fit on a screen, scrolling commands (and their function keys) let you see more columns or rows.

1. Put the cursor under the leftmost heading, `CUST` (if your cursor is in the command line, simply use the arrow keys to move it into the workspace.)
2. Type **P.** (the command for "present")

```
                                   QBE QUERY
                                                          Row 1    Column 1

CUST | CUST_NO | CUST_NAME | ADDRESS | CITY | STATE | ZIP | PHONE | SALES_REP | CUSTOMER_TYPE |
-----+---------+-----------+---------+------+-------+-----+-------+-----------+---------------+
 P.  |         |           |         |      |       |     |       |           |               |
```

3. [F2] to execute the query.

```
            Rows 1      thru 24          REPORT       Columns 1      thru 99


 CUST    CUST
  NO     NAME                 ADDRESS              CITY            ST     ZIP  PHONE
 ------  -------------------- -------------------- --------------- -- ------- ---------------
 LOSGAT  LOS GATOS HARDWARE   100 SARATOGA AVENUE  LOS GATOS       CA   95038  408 324 1538
 QUALIY  QUALITY PRODUCTS     E 227 3RD AVE, #5    SPOKANE         WA   99202  509 624 8041
 KINGPR  KING PRODUCTS        244 NO. GRANT        STOCKTON        CA   95202  209 464 4602
 GOLDEN  GOLDEN STATE EQUIP.  700 MAIN STREET      DUBLIN          CA   94566  415 345 9998
 GIFTSA  GIFTS AND STUFF      421 SOUTH HILL ST.   FRESNO          CA   93718  209 488 2111
 KEEPSA  KEEPSAKES GIFTS      CENTRAL MALL #146    SUNNYVALE       CA   94086  408 733 4545
 CARLOS  CARLOS STATIONERS    5717 GLENROSA        PHOENIX         AZ   85031  602 247 6196
 KANDSG  K AND S GIFTS        4884 EASTSIDE BLVD   REDDING         CA   96001  916 243 3546
 MARINA  MARINA PRODUCTS      123 OCEAN PARK AVE   SANTA MONICA    CA   91117  213 345 9851
 SANTAB  SANTA BARBARA        100 AERO CAMINO      GOLETA          CA   93116  805 968 3131
 LASVEG  LAS VEGAS PEN CO.    20 GREG STREET       SPARKS          NV   89431  702 355 1333
 MRPENA  MR PEN AND PENCIL    3617 PANAMA ROAD     BRIGHTON        MI   48116  313 229 6767
 NEWGEN  NEW GENERATION       4844 OLD PLANK ROAD  MILFORD         MI   48042  313 685 8228
 GRADY   GRADY GIFTS          111 CRYSTAL LAKE BL. CRYSTAL LAKE    IL   60014  815 455 5557
 HONKIN  HONK INDUSTRIAL      6335 RENOIR          BATON ROUGE     LA   70806  504 926 5000
 MAINSU  MAIN SUPPLIES        1681 IMPERIAL WAY    MT. PROSPECT    IL   60056  312 421 5123
 NATION  NATIONAL MEDIA       413 SUTHERLAND AVE   LAKE DALLAS     TX   75065  817 497 2225
 AMERIC  AMERICAN GIFTS       116 S. MILL STREET   PRYOR           OK   74361  918 825 4884
 TEXASG  TEXAS GRAPHICS       1919 PETERS ROAD     IRVING          TX   75061  800 223 8014
        1=Help        2=            3=End         4=Query       5=Proc
        6=Form        7=Backward    8=Forward     9=Left       10=Right

COMMAND ===>                                                        SCROLL ===> PAGE
```

Navigating

As you scroll, the Report screen header field identifies the portion of the table currently displayed with rows on the left and columns on the right. If you have scrolled to the row/column at the top, bottom, right, or left side of a table, the appropriate message is displayed on the status line.

Type…	OR Press…	To Go to …
Forward	[F8]	the end of the screen by the scroll rate.
Forward*n*		the end of the screen contents by the number of lines specified.
Forward MAX		directly to the end of the screen content.
BOTtom		the end of the screen content (same as F MAX).
Backward	[F7]	backwards to the beginning of the screen content by scroll rate.
Backward*n*		backwards to the beginning of the screen content by specified number of lines.
TOP		the beginning of screen contents.
B MAX		directly to the beginning of screen contents (same as BOT MAX).
RIGHT	[F10]	to the right, in increments of the scroll rate.
LEFT	[F9]	to the left, in increments of the scroll rate.

NOTE: *You can also type the commands FORWARD, BACKWARD, and so on, instead of using the function keys.*

To change the setting for Scroll Rate (the default is PAGE), move the cursor to the Scroll field and enter one of the following:

Scroll Indicator	Description
PAGE	Reverse or advance an entire page.
HALF	Reverse or advance half a page.
CSR	Move back or forward to the cursor position.
MAX	Move to the end or bottom of the screen contents.
TOP	Move to the beginning or top of the screen contents.
Number	Move back or forward to the specified number of lines.

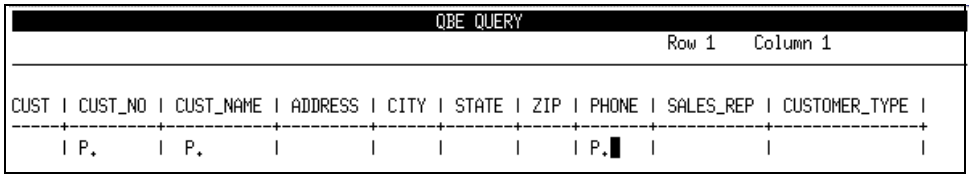
NOTE: *You can specify scroll values in the SCROLL= field or specify the value explicitly with each movement command.}*

Selecting specified columns

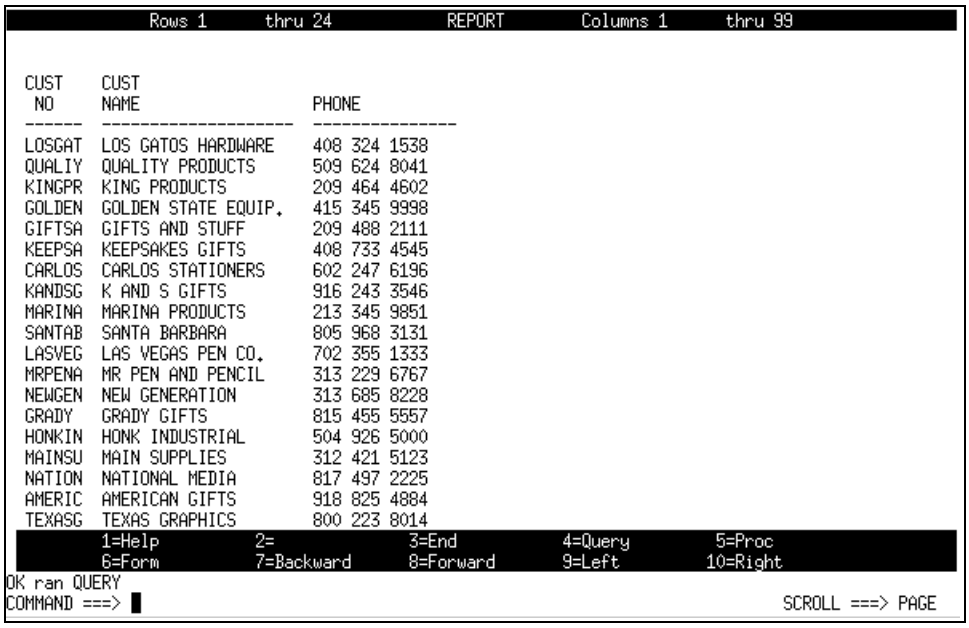
You just selected all the data in the CUST table. In a very large table scrolling through many columns or rows of unneeded data can be inconvenient and expensive. With QBE, you can use P. under the column heads of a table to display only the columns you need.

If you haven't already done so, [F4] to return to the QBE Query screen. You see the current query is maintained.

1. Move the cursor to the CUST (table name) field and [Spacebar] to clear the field (or else the entire table will appear again).
2. Move the cursor to the first column field, CUST_NO, and type **P.**
3. [Tab] to the CUST_NAME and PHONE fields and, in each, type **P.**



4. [F2] to run the query.



Ordering rows

You often want to sort the data to rank values or show a time increase. You can order your results using either ascending order (AO.) or descending order (DO.) For example, it seems logical to want to see the customer information in alphabetical order, according to customer name.

1. Ensure that you are at the QBE Query screen ([F4] if you are not.)
2. Clear the **P.** commands from CUST_NO, CUST_NAME, and PHONE fields, using the [Spacebar].

3. Move the cursor to the CUST (table name) field and type **`P.`** to display the entire CUST table.
4. To arrange the data in ascending order by customer name, move the cursor to the CUST_NAME field and type **`AO.`**

```
                                   QBE QUERY
                                                 Row 1    Column 1

CUST | CUST_NO | CUST_NAME | ADDRESS | CITY | STATE | ZIP | PHONE | SALES_REP | CUSTOMER_TYPE |
-----+---------+-----------+---------+------+-------+-----+-------+-----------+---------------+
P.   |         | AO.       |         |      |       |     |       |           |               |
```

5. [F2] (Run Query).

```
              Rows 1      thru 24          REPORT        Columns 1      thru 99


 CUST   CUST
  NO    NAME                  ADDRESS               CITY             ST      ZIP  PHONE
 ------ --------------------  --------------------  ---------------  --  -------  ---------------
 AAAGIF AAA GIFT COMPANY      4203 LAS CRUCES       SEATTLE          WA    98125  206 364 1220
 AMERIC AMERICAN GIFTS        116 S. MILL STREET    PRYOR            OK    74361  918 825 4884
 APEX   APEX                  3707 HWY 99           VANACOUVER       WA    98665  206 574 1989
 BIGJOH BIG JOHNS             5550 PACIFIC AVE      TACOMA           WA    29840  206 473 5852
 BOBAND BOB AND CINDYS PLACE  2020 MAIN STREET      FERNDALE         WA    98248  206 384 9085
 BUDGET BUDGET STATIONERS     232 SW 153RD          BURIEN           WA    98166  206 246 7153
 CARLOS CARLOS STATIONERS     5717 GLENROSA         PHOENIX          AZ    85031  602 247 6196
 CELEBR CELEBRITY             783 FOREST AVE        PLYMOUTH         MI    41870  313 455 8923
 EXCELS EXCEL SPECIALITIES    1118 SW 310TH ST      FEDERAL WAY      WA    98023  206 661 8491
 GIFTSA GIFTS AND STUFF       421 SOUTH HILL ST.    FRESNO           CA    93718  209 488 2111
 GOLDEN GOLDEN STATE EQUIP.   700 MAIN STREET       DUBLIN           CA    94566  415 345 9998
 GRADY  GRADY GIFTS           111 CRYSTAL LAKE BL.  CRYSTAL LAKE     IL    60014  815 455 5557
 HONKIN HONK INDUSTRIAL       6335 RENOIR           BATON ROUGE      LA    70806  504 926 5000
 HYPER  HYPERGRAPHICS         6453 LANE AVE         TUCUMACRI        NM    84555  704 335 5553
 KANDSG K AND S GIFTS         4884 EASTSIDE BLVD    REDDING          CA    96001  916 243 3546
 KBARB  K BAR B               3745 SE DIVISION ST.  PORTLAND         OR    97026  503 236 7323
 KEEPSA KEEPSAKES GIFTS       CENTRAL MALL #146     SUNNYVALE        CA    94086  408 733 4545
 KINGPR KING PRODUCTS         244 NO. GRANT         STOCKTON         CA    95202  209 464 4602
 LASVEG LAS VEGAS PEN CO.     20 GREG STREET        SPARKS           NV    89431  702 355 1333
        1=Help          2=              3=End           4=Query         5=Proc
        6=Form          7=Backward      8=Forward       9=Left         10=Right
OK ran QUERY
COMMAND ===>                                                                 SCROLL ===> PAGE
```

Sort columns on more than one criteria by specifying different priorities. The command code “AO(2).” under the ZIP column head indicates that the ZIP code is the second priority.

6. Replicate the following query and see what happens.

   Notice that since the entire table (CUST) is not being displayed, each column you want to see must be have a “P.” command.

7. Because the ZIP field does not have enough room for the entire “P.AO(2).” command, you must enlarge it. Put the cursor in the field and [F5] 5 times to add 5 character spaces to the column display.

```
                                   QBE QUERY
                                                 Row 1    Column 1

CUST | CUST_NO | CUST_NAME | ADDRESS | CITY | STATE | ZIP      | PHONE | SALES_REP | CUSTOMER_TYPE
-----+---------+-----------+---------+------+-------+----------+-------+-----------+---------------
     |         | P.        | P.      | P.   | P.AO. | P.AO(2). |       |           |
```

8. [F2] to run the query and see results.

☞ *Sorting rows, descending order*

If your company must reduce the number of products it manufactures, you might want to examine the sales data to determine which offerings are the slowest movers, and thus, good candidates for removal from inventory.

You can use QBE to arrange inventory by unit sales, with the best-selling items listed first.

1. Remove the CUST table from the workspace by typing **`DELete`**, on the command line, moving the cursor into the table, and [Enter].

   Be sure to remove the DEL commands from the command line when you have completed the operation. Otherwise, you could inadvertently delete objects from the database.

2. Type **`DRAW INVENT`** and [Enter].

3. Specify the tables to display, and the criteria by which to order them. Put P. in the INVENT column (to show the entire table) and DO. in the SOLD_YTD column.

```
                                   QBE QUERY
                                                            Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+
 P.    |         |             |            |         | DO.      |
```

   (Remember, [F2] to see the query results.)

You can see that the company has sold more deluxe gift boxes than any other item. Also, each of the two types of pens has sold nearly as many units as the deluxe gift box, and, incidentally, at a higher price.

```
              Rows 1      thru 24          REPORT        Columns 1      thru 99


  PART                                UNIT       ON      SOLD
   NO        DESCRIPTION              PRICE     HAND      YTD
  ----------  --------------------  ----------  --------  --------
  BOX-DLX     DELUXE GIFT BOX           2.50     12350     25987
  R510B4-A    PEN GOLD SILVER SWIR      5.00     15450     25900
  R510B2-A    PEN BLACK GOLD SWIRL      4.50     15000     25000
  TK1015      KEY CHAIN RECT. LGE.      2.30      8294     10528
  R310G       PEN RHINESTONE GOLD       5.50      8500     10500
  R510B1-B    PENCIL GRAY GOLD SWR      6.00      4300      8525
  BOX-PLS     PLASTIC GIFT BOX          0.75     15000      8000
  R212C1      PEN GRAY GOLD             4.50      3500      7900
  TK1001      KEY CHAIN RECTANGLE       1.85      4753      7582
  R510B3-A    PEN RED GOLD SWIRL        4.50      8400      7550
  R510B2-B    PENCIL BLACK GLD SWR      6.00      2100      7500
  R212E1      PEN BLK GLD DIAMOND       4.50      4500      7500
  R510B3-B    PENCIL RED GOLD SWRL      6.00      3300      6600
  R310S       PEN RHINESTONE SILVR      5.50      2585      5500
  TK1013      KEY CHAIN RECT. LG.       2.15      4521      4892
  LT014       LUGGAGE TAG W/STRAP       3.20      2500      4590
  TK1011      KEY CHAIN TAG POINT       2.15      8975      4351
  R212E2      PEN BLK GLD SQUARE        4.50      3200      4238
  R510B1-A    PEN GRAY GOLD SWIRL       4.50     10500      3500
          1=Help          2=              3=End           4=Query         5=Proc
          6=Form          7=Backward      8=Forward       9=Left         10=Right
OK ran QUERY
COMMAND ===>                                                          SCROLL ===> PAGE
```

Using a single condition for specified rows

When you only need specific rows in a table for your results, you can eliminate unnecessary retrieval and processing time. Using single search conditions, multiple search conditions, or a Conditions Box, you can specify exactly the data you want to receive.

You refine your query by establishing criteria for data in the rows. The criteria are defined in QBE using literal values and logical conditions.

To query using a single search condition, you enter the condition under the column head that defines the result. For example, to see a list of items that have more than 10,000 on hand, you modify the previous query using the logical condition "greater than":

[F4] to return to the QBE Query screen, if it isn't currently displayed on your screen.

1. Type **>10000** in the column ON_HAND.

```
                                    QBE QUERY
                                                              Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+
 P.    |         |             |            | >10000  | DO.      |
```

2. [F2] to see the results.

```
            Rows 1      thru 24          REPORT       Columns 1      thru 99


 PART                                      UNIT       ON       SOLD
  NO        DESCRIPTION                    PRICE      HAND     YTD
 ----------  --------------------  ---------  --------  --------
 BOX-DLX     DELUXE GIFT BOX            2.50     12350     25987
 R510B4-A    PEN GOLD SILVER SWIR       5.00     15450     25900
 R510B2-A    PEN BLACK GOLD SWIRL       4.50     15000     25000
 BOX-PLS     PLASTIC GIFT BOX           0.75     15000      8000
 R510B1-A    PEN GRAY GOLD SWIRL        4.50     10500      3500
 TK1005      KEY CHAIN OVAL             2.15     27690        35

        1=Help          2=              3=End          4=Query        5=Proc
        6=Form          7=Backward      8=Forward      9=Left         10=Right
OK ran QUERY
COMMAND ===>                                                    SCROLL ===> PAGE
```

Using multiple conditions

Usually, however, questions are more complicated than can be answered by single-condition queries. You may have some concerns about the variety of pens your company offers. Now that you have a list of items that are well-stocked, refine it to show only those items that have the word Pen in their DESCRIPTION.

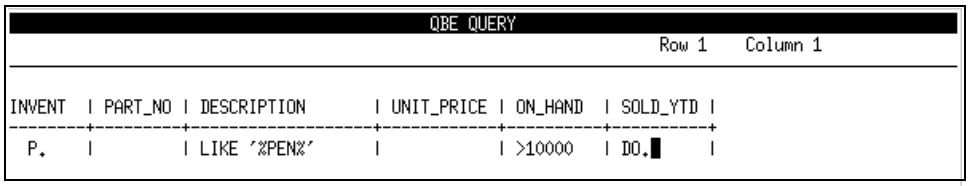
Literal values must be enclosed with single quotes when they serve as query conditions. If you are not sure about the entire string for the literal search, or you cannot define the entire string for a column, use the "wildcard" character, '%' with the word "like." (If you know the entire string, simply use '*string*' where *string* represents the value of the literal value.)

Make sure you are at the Query screen before beginning this procedure.

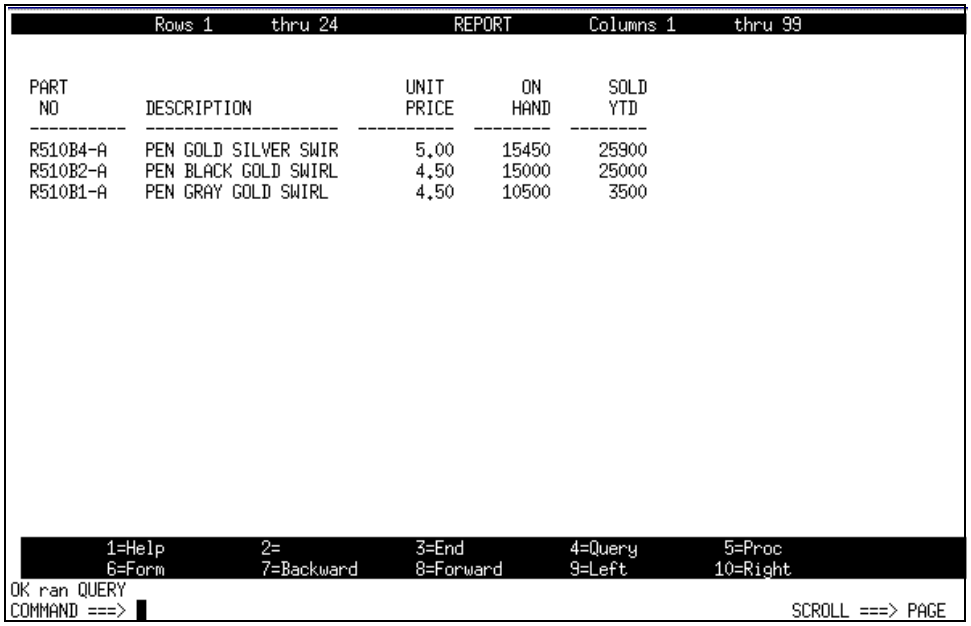
1. Type **`like '%PEN%'`** in the DESCRIPTION column.

   You may need to increase the width of the column when you use wildcards. [F5] adds a character space to the column width.

   Some databases are not case-sensitive; however, most are. Ensure that the wildcard's case matches that of the string in the database; that is, if the text appears in uppercase letters, type it that way in the table outline.



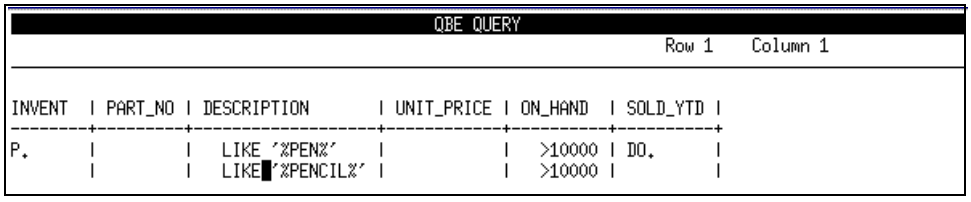
2. [F2] to see the results.



Return to the Query screen, and check the inventory for both Pens and Pencils. Because you want to see records for either item, you create an OR query.

3. [F4] to return to the QBE window.
4. Move the cursor to the INVENT column (the column head that defines the entire table) and [F5] to create an additional field for the OR query condition.
5. In the second line of the DESCRIPTION field, enter **`like 'PENCIL'`**

6. In the second line of the ON_HAND column, then, enter **> 10000** to indicate that you only want to see Pencils that have more than 10,000 items in stock.



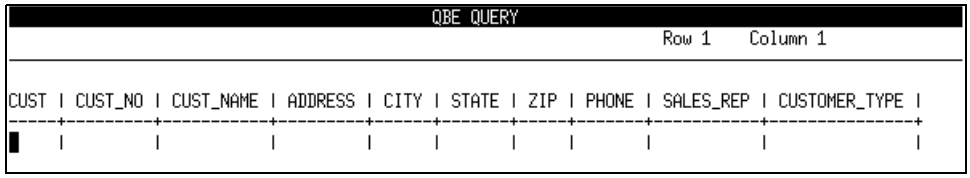
7. [F2] to see the results of your query.

Using a conditions box

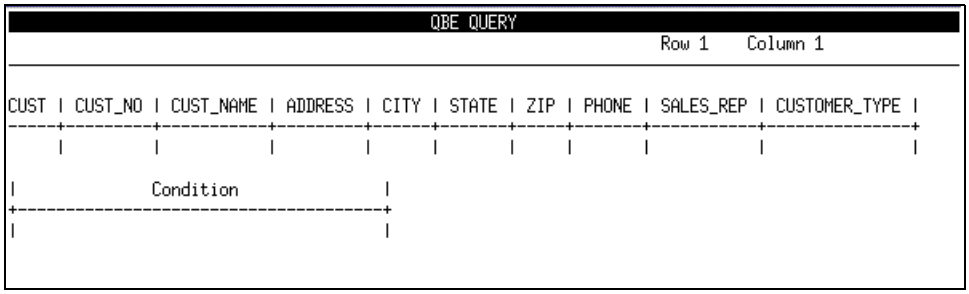
Conditions boxes let you specify more complicated queries than those allowed in column fields. You can perform AND and OR queries to search large databases for detailed information.

1. Move your cursor to the command line (at the bottom of the screen) and type **RESET QUERY** and [Enter].
2. On the command line, type **DRAW CUST** [Enter].

   The CUST table outline appears on the QBE Query screen.



3. Type **DRAW COND** and [Enter] to create an empty conditions box.

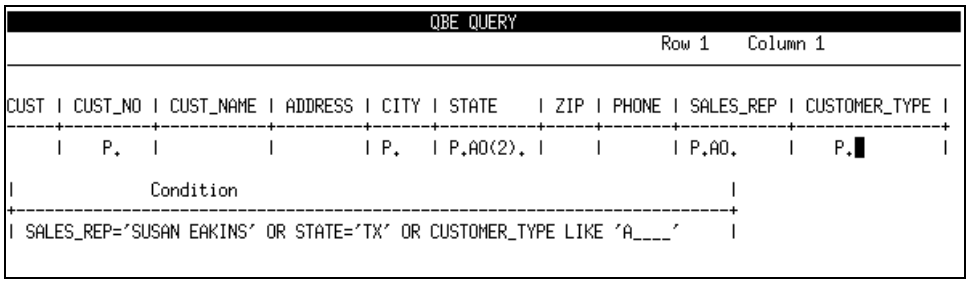


4. Move the cursor to the conditions box using the [Tab] and Arrow keys.
5. [F5] to enlarge the conditions box (just as you enlarged the column fields in previous exercises) and type the following conditions in the box:

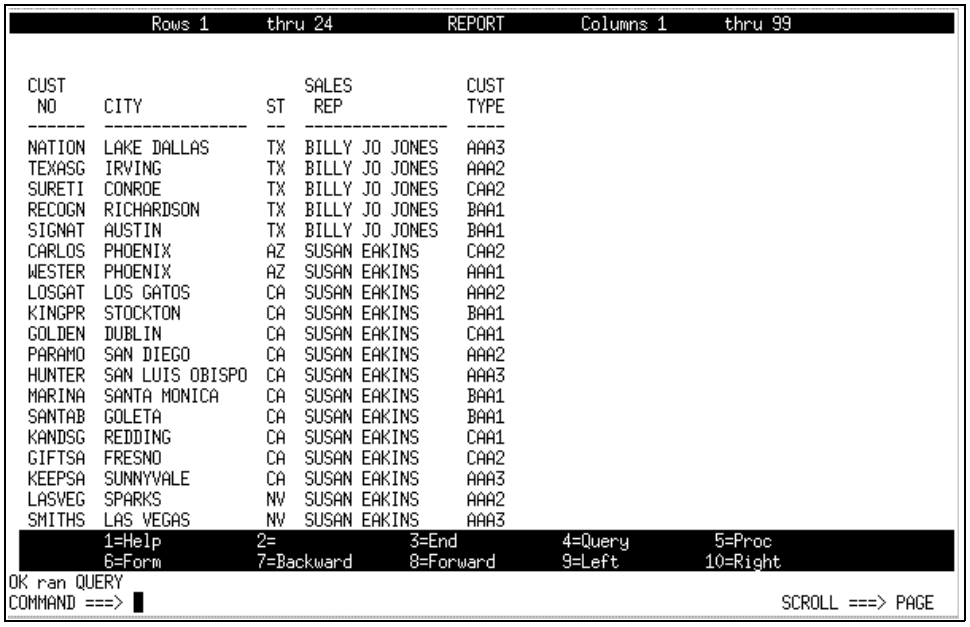
```
SALES_REP='SUSAN EAKINS'
OR STATE = 'TX'
OR CUSTOMER_TYPE LIKE 'A_____'
```

6. Type **P.** in the CUST_NO, CITY, and TYPE (not shown) columns to select them for the report.

7. Type **P.AO.** under SALES_REP and **P.AO(2).** under STATE to order the results.



8. [F2] to run the query.



Calculated Values

All the queries you have created are considered "simple queries." To obtain information you may need to select rows on the basis of calculated values, produce summary data, sort records into groups, or join two or more tables. These actions are all part of complex querying.

You have been asked to create a report that sorts products sold by their value. Return to the QBE Query screen and the INVENT table outline.

([F4], type **DELETE**, move the cursor into the table, [Enter] and remove **DELETE** from the command line.)

Creating calculated values requires adding a new column to the existing table and defining its value based on the results of combining data from two or more other columns. To define synonyms to operate on columns. The synonyms can be arbitrary letters, but ***must*** start with an underscore(_).

To extend the power of QBE in calculating values, you use arithmetic operators:

- ***Add*** --- ++

- *Subtract* --- -
- *Multiply* --- *
- *Divide* --- /

☞ *Selecting on calculated values*

1. Type **RESET QUERY** on the command line and [Enter].
2. Type **DRAW INVENT** and [Enter].
3. Move the cursor to the SOLD_YTD column ***heading*** and [F5] to create a new column to the right. (To delete a column, [F6].)
4. In the INVENT field, type **P.** to display all data in all columns. In the UNIT_PRICE field, type **_U** and **_S** in the SOLD_YTD as column identifiers.
5. Then, type the expression DO._U*_S in the new column field. This expression is shorthand for “Present in Descending Order the value of UNIT_PRICE multiplied by SOLD_YTD.”

```
                                        QBE QUERY
                                                              Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |           |
-------+---------+-------------+------------+---------+----------+-----------+
 P.    |         |             | _U         |         |  _S      | DO._U*_S  |
```

6. [F2] to display the report.

```
               Rows 1      thru 24           REPORT        Columns 1      thru 99


                                                                       A.UNIT
  PART                                   UNIT        ON       SOLD   PRICE*A.SO
   NO        DESCRIPTION                 PRICE      HAND       YTD      YTD
  ----------  --------------------   ----------  --------  --------  ----------
  R510B4-A    PEN GOLD SILVER SWIR         5.00     15450     25900   129500.00
  R510B2-A    PEN BLACK GOLD SWIRL         4.50     15000     25000   112500.00
  BOX-DLX     DELUXE GIFT BOX              2.50     12350     25987    64967.50
  R310G       PEN RHINESTONE GOLD          5.50      8500     10500    57750.00
  R510B1-B    PENCIL GRAY GOLD SWR         6.00      4300      8525    51150.00
  R510B2-B    PENCIL BLACK GLD SWR         6.00      2100      7500    45000.00
  R510B3-B    PENCIL RED GOLD SWRL         6.00      3300      6600    39600.00
  R212C1      PEN GRAY GOLD                4.50      3500      7900    35550.00
  R510B3-A    PEN RED GOLD SWIRL           4.50      8400      7550    33975.00
  R212E1      PEN BLK GLD DIAMOND          4.50      4500      7500    33750.00
  R310S       PEN RHINESTONE SILVR         5.50      2585      5500    30250.00
  TK1015      KEY CHAIN RECT. LGE.         2.30      8294     10528    24214.40
  R212E2      PEN BLK GLD SQUARE           4.50      3200      4238    19071.00
  R510B1-A    PEN GRAY GOLD SWIRL          4.50     10500      3500    15750.00
  LT014       LUGGAGE TAG W/STRAP          3.20      2500      4590    14688.00
  TK1001      KEY CHAIN RECTANGLE          1.85      4753      7582    14026.70
  SBLP1206    LICENSE PLATE FRAME         12.00       842      1049    12588.00
  R212C3      PEN RED GOLD                 4.50      1400      2500    11250.00
          1=Help          2=            3=End          4=Query         5=Proc
          6=Form          7=Backward    8=Forward      9=Left          10=Right
OK ran QUERY
COMMAND ===>                                                          SCROLL ===> PAGE
```

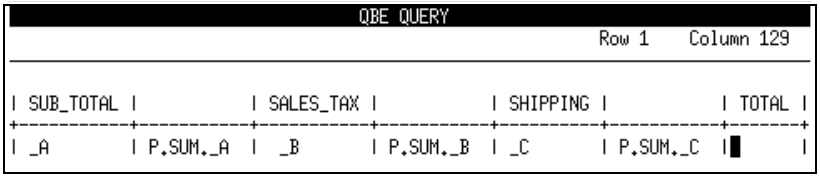
Summary Data

In addition to using calculated values, you can also produce summary data from your database tables. For example, if you want to know the total SUB_TOTAL, SALES_TAX, and SHIPPING_CHG for all orders to date, you create three new columns, define them as the average of each of the specified columns, and run the query.

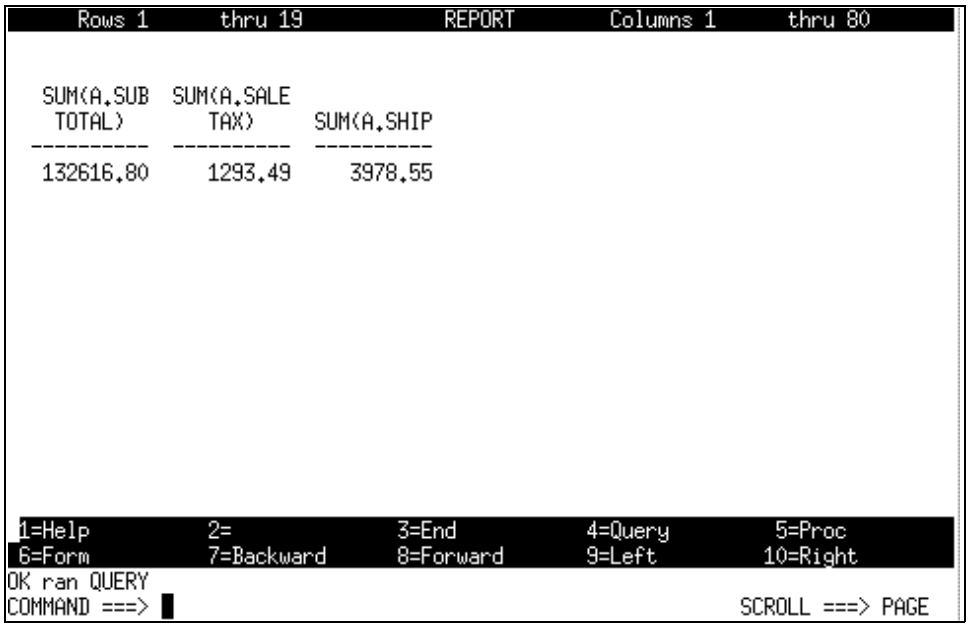
Producing a summary report

Begin at the QBE screen.

1. Type **RESET QUERY** at the command line and [Enter].
2. Type **DRAW ORDS** and [Enter.
3. Use [F5] to create three new columns, each to the right of SUB_TOTAL, SALES_TAX, and SHIPPING_CHG.
4. Create synonyms for each of the columns that will be summarized.
   - SUB_TOTAL --- **_A**
   - SALES_TAX --- **_B**
   - SHIPPING_CHG --- **_C**
5. Create the calculating expression for each unnamed column to the right of the summarized columns.
   - SUB_TOTAL --- **P.SUM._**
   - SALES_TAX --- **P.SUM._B**
   - SHIPPING --- **P.SUM._C**



6. [F2] to run the query and view the averages for each of the columns specified.



You can use any of the following summarizing functions:

Function	Description
SUM	Total of column
MIN	Lowest value in column.
MAX	Highest value in column.
COUNT	Number of different values in a column (not the number of rows in the table.)

Grouping Data

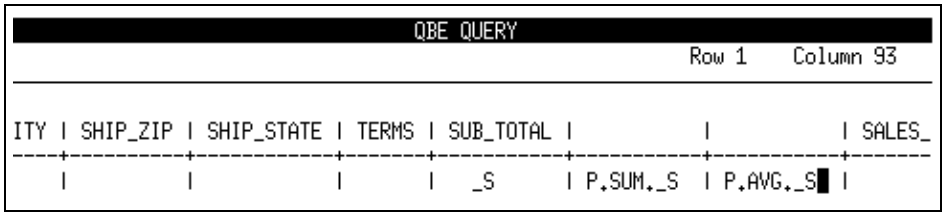
Grouping enables you to create sets of rows on which calculations and summaries can be performed. For example, you can use group functions to calculate:

- Average value of COD orders.
- Customer orders shipped on NET10 or NET30 terms.
- Logical and arithmetic functions.
- Complex queries using both AND and OR logic combinations.

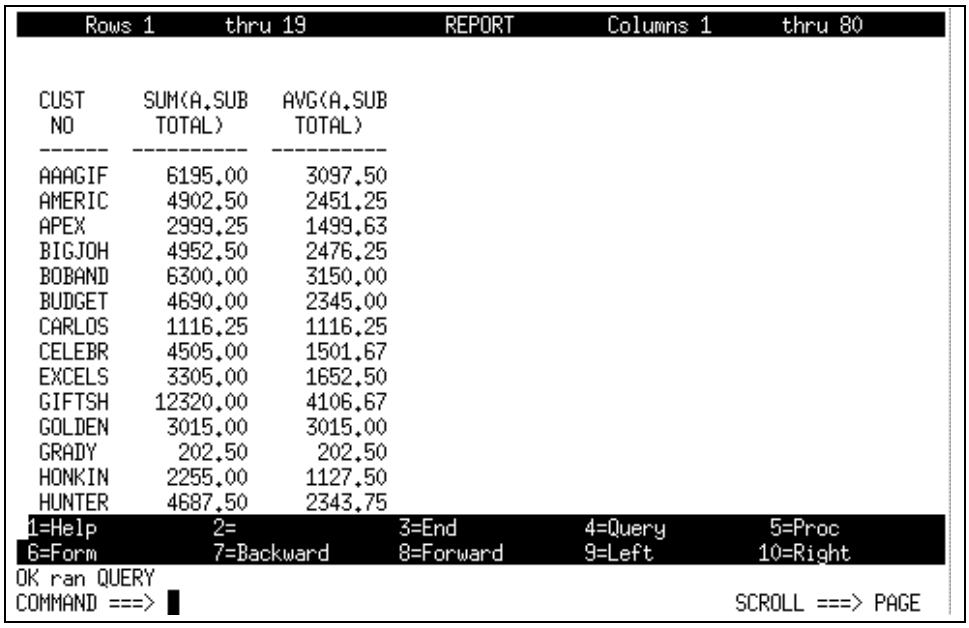
The Group command is **`G.`** Generally, `G.` produces one group for each set of identical values in a column. You can group more than one column; if more than one column contains a `G.`, the rows are grouped by the combined G. values. To display a column that uses G., you must also use P. to "present."

Creating a report with total and average data

1. On the QBE Query screen, reset the query and draw the ORDS table outline.
2. Type **G.P.** in the CUST_NO field to group the values and display the result.
3. [F5] to create two new columns adjacent to SUB_TOTAL.
4. Type **_S** in the SUB_TOTAL field as an identifier.
5. In the first of the new columns, type **P.SUM._S** and, in the other new column, type **P.AVG._S** to calculate total and average on the columns respectively.



6. [F2].



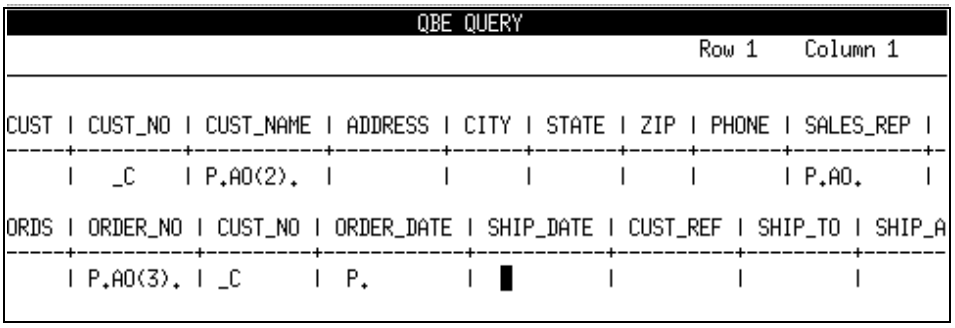
Creating a report from two tables

Relational database design typically supports the idea of storing data in tables that logically represent some entity or action. These tables are related to each other by means of corresponding columns, or keys. QBE allows you to perform “joins” on tables, that is ask for information that requires two or more tables to be combined in a single query result, in a visual manner.

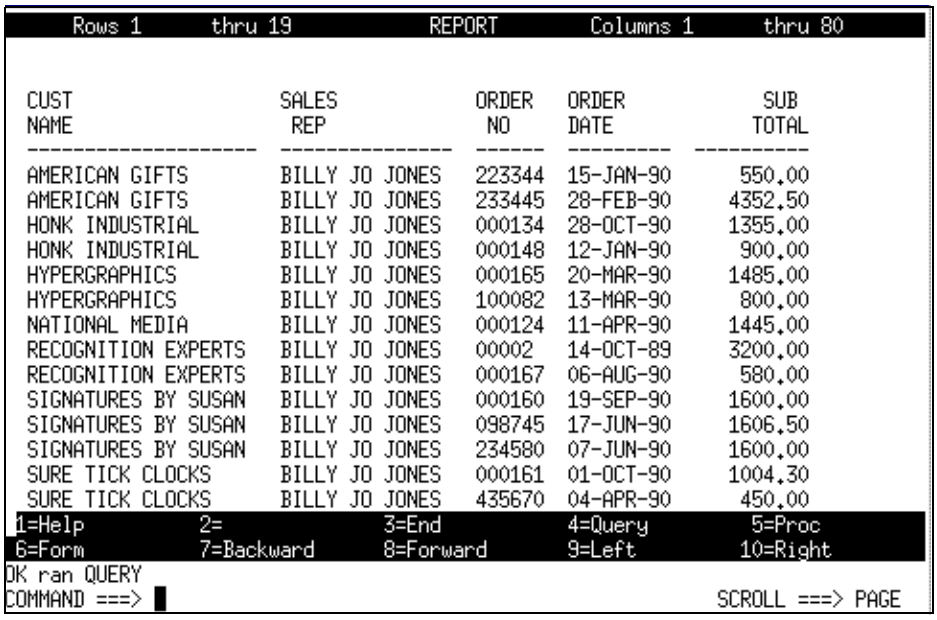
You join the tables by entering a synonym in the columns that have corresponding column heads.

In this exercise, you create a report listing all orders by sales representative, customer, and order number, alphabetized by customer name with orders for each customer in sequence.

1. [F4] to return to the QBE Query screen.
2. Type **`RESET QUERY`** on the command line, and [Enter].
3. Type **`DRAW CUST`** and [Enter].
4. Type **`DRAW ORDS`** and [Enter].
5. Type **`_C`** in the CUST_NO field of ***both*** tables to indicate the common column on which to perform the join.
6. In the SALES_REP field type **`P.AO.`**
7. In the CUST_NAME field type **`P.AO(2).`**
8. In the ORDER_NO field, type **`P.AO(3).`**
9. Type **`P.`** in the ORDER_DATE and SUB_TOTAL fields to include order date and amount.



10. [F2].

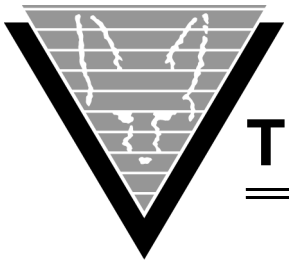


The report appears with the CUST_NAME and SALES_REP columns from the CUST table, and the ORDER_NO, ORDER_DATE, and SUB_TOTAL columns from the ORDS table.

Summary

If you've completed the exercises in this chapter, you now know how to perform all the important operations using QBE in TRIMqmr. The following chapter lists all the commands you can use, as well as their parameters and values. It does not explain how to use the commands in detail, so you may want to go through this tutorial a few more times to completely familiarize yourself with TRIMqmr operations if you are new to database queries.

To quit TRIMqmr, simply [F3] until you see a prompt that asks you if you really want to end the application, and respond "YES." The next exercises, "***Formatting Reports***" on page 88 teach you how to format reports.



Chapter 3
TRIMqmr Commands

TRIMqmr uses and includes a variety of commands. The application has its own commands, which you can issue from the command line, and often with single function keystrokes, depending on the context of the command.

This chapter enumerates and describes the variety of command available, without specific reference to their scope, including:

- TRIMqmr
- Editing
- Cursor control (scrolling)

The capital letters in each command indicate the minimum characters that you must type for each command. For example, you can issue a DRAW by typing **DR**.

Most commands invoke a dialog screen if you do not specify options. For example, typing **ER** without any options displays the Erase Command prompt.

Command	Description
Backward	Scrolls toward the beginning of the information displayed.
BOttom	Scrolls to the end of the information displayed.
CONNect	Connect to a database.
DELete	Remove a query object (column, conditions box, or comment box) from a QBE Query screen.
Display	Display a database object (procedure, query, profile form, report, or table).
DRaw	QBE: Display a query object SQL: display a SELECT, INSERT INTO, or UPDATE.
End	Show the Home screen. If issued from the Home screen, show the “End QMR Confirmation prompt.
ENLarge	Increase the width of a query object. Add an empty row to a query object.
ERase	Remove a database object from the database.
EXit	End a TRIMqmr session immediately (no confirmation prompts).
EXPort	Copy a database object or report to a file external to the database.

Command	Description
FORM.*name*	Show the named form. May be BREAK*n*, COLUMNS, FINAL, OPTIONS, PAGE or MAIN.
Forward	Scroll toward the end of the information displayed.
HElp	Invoke the context-sensitive help system.
IMPort	Copy the contents of a file external to the database into a form, procedure, query, profile, or report.
Left	Scroll toward the left of the information displayed.
List	Show a list of items that are available in the database.
PRint	Print a copy of a database object or report to a specified output. Output can be a printer, file, or computer screen.
PROCedure	Show the Procedure screen.
PROFile	Show the Profile screen.
Query	Show the Query screen as specified in the current profile (either QBE or SQL), or specify the language on the command line.
REDuce	Reduce the width of a query object. Delete a query object. Delete a row from a query object on the QBE screen.
RESet	Restore a database object (query, procedure, profile, form, or table data) to its previously saved state.
RETrieve or ?	Show previously entered commands.
Right	Scroll towards the right of the information displayed.
Run	Execute a query or procedure.
SAve	Store the current contents of a named database object in a database.
SEt	Change a specific Profile attribute.
SYStem	Execute an operating system command without ending the current TRIMqmr session.
Top	Scroll to the beginning of the information displayed.

Entering Commands

You can enter a QMR command in any of the following ways.

On the Command Line

You can enter any QMR command by typing it in full after the arrow on the command line of any panel. For example,

```
COMMAND ===> RUN THISQUERY (FORM=THISFORM
```

could also be abbreviated (according to the minimum character rules) as:

```
COMMAND ===> RU THISQUERY (F THISFORM
```

To execute either command, simply [Enter].

On the Prompt Screen

If you can't remember how to write the name, or the possible options, for a command, you can use a prompt screen. Simply type the command and [Enter] to see the prompt and complete the required information.

[Enter] again to run the command.

Function Keys

TRIMqmr has a default set of keys defined for each screen. These definitions can be modified by editing the keymap file, described in "***Key Mapping***" on page 131.

Procedures

You can include almost any QMR command as a line in a procedure, including `RUN` to run the same or another procedure. Running the procedure executes the commands on each line in turn.

When you do put commands into a procedure, you should use the full, rather than abbreviated, command names, options, and values.

In the List Screen

You can execute a number of QMR commands from the List panel. See ***"LIST"*** on page 50.

BACKWARD

Scrolls the contents of the workspace backward.

Syntax

```
Backward [ amount ]
```

Description

Backward lets you move backward by *amount* through a report, form, query, or procedure.

amount can be:

Csr	Moves the contents until the boundary is next to the cursor position. For example, `BACKWARD CSR` moves the contents up until the bottom boundary is just below the line on which the cursor rests. If the cursor is not in the screen workspace, or is already on the boundary of the screen, then CSR has the same effect as PAGE.
Half	Moves the contents half the depth of the workspace.
Max	Moves the contents until the boundary is at the edge of the item being viewed. For example, `BACKWARD MAX` moves the contents as far to the top of the screen as it can go.
Page	Moves the contents to the depth of the screen.
n	Moves the contents *n* character spaces where 1 > *n* > 9999.

If you don't specify an *amount*, TRIMqmr uses the setting shown on the `SCROLL ===>` area of the status line.

[F7] is available to scroll backward on the Form, Query, Report, and Procedure screens.

BOTTOM

Scrolls the workspace to view the last line of the contents.

Syntax

```
BOttom
```

Description

Equivalent to `FORWARD MAX`, this command allows you to see the end of the contents with a single command.

You can use this with any scrollable screen, including forms, reports, queries, and procedures.

CONNECT

Connects to a database.

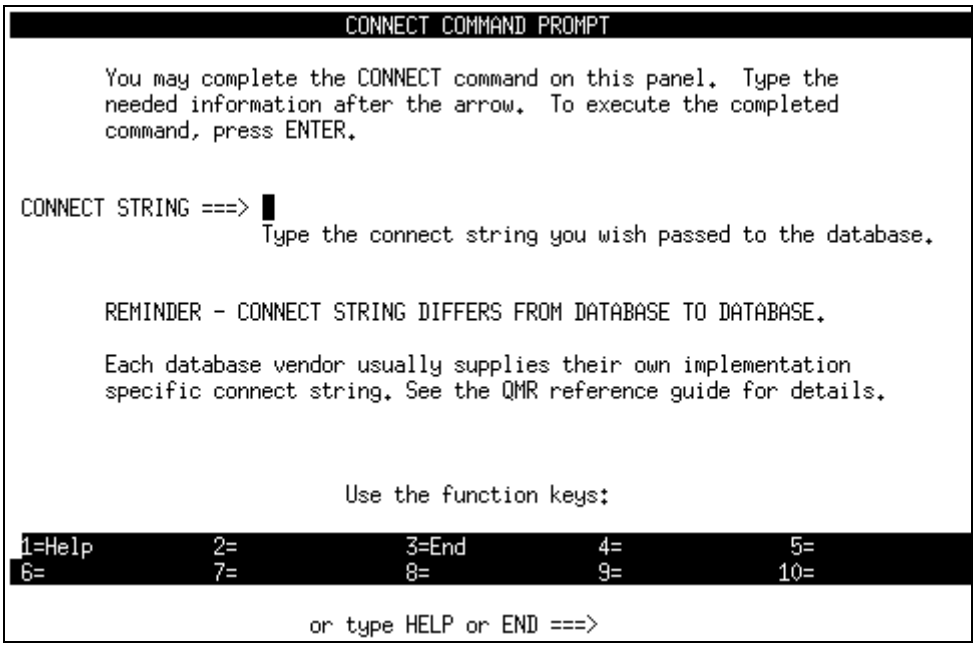
Syntax

`CONNect` *`connect_string`*

Screen

Type `CONNECT` in TRIMqmr with no options, and see this Prompt.

Description



`connect_string` is the connect string specific to your database.

You must connect to the database before you are allowed to execute commands against stored data. You can only connect to one database at a time.

You can CONNECT implicitly when you start TRIMqmr, or explicitly from the Home screen.

Example

```
CONN net:boss/goodguy/ORG
```

connects the user named boss whose password is goodguy to the ORG database on a VORTEX network.

DELETE

Removes an object from the QBE Query screen.

Syntax

```
DELete +  [Enter].
```

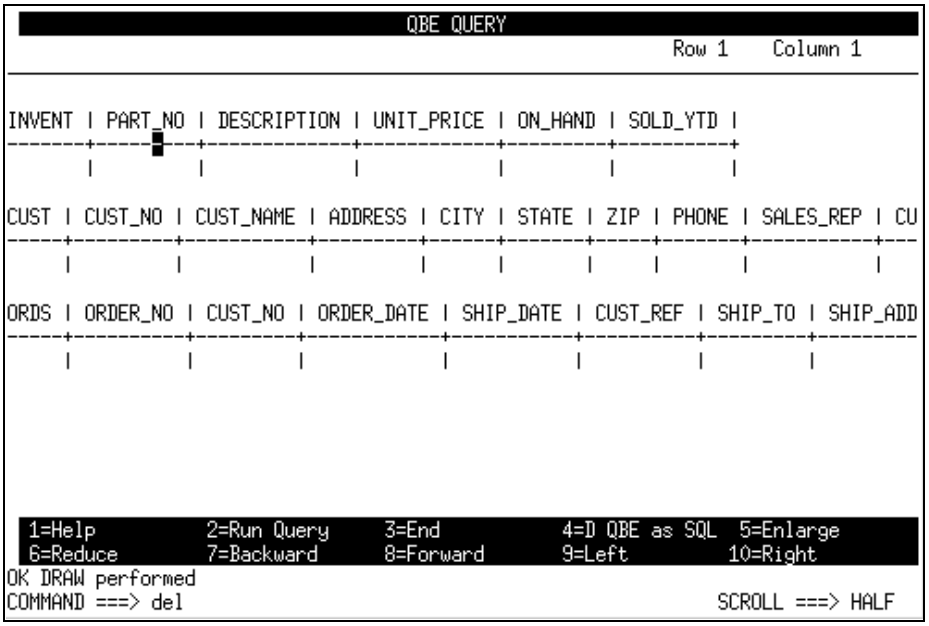
Description

You can delete a table, conditions box, or comments box from a QBE Query screen.

NOTE: *Be sure to remove this command from the command line when you have completed your delete. Otherwise, you may inadvertently delete another object when you press [Enter].*

Example

In the following screen, pressing [Enter] deletes, or removes, the INVENT table from the QBE Query screen.



DISPLAY

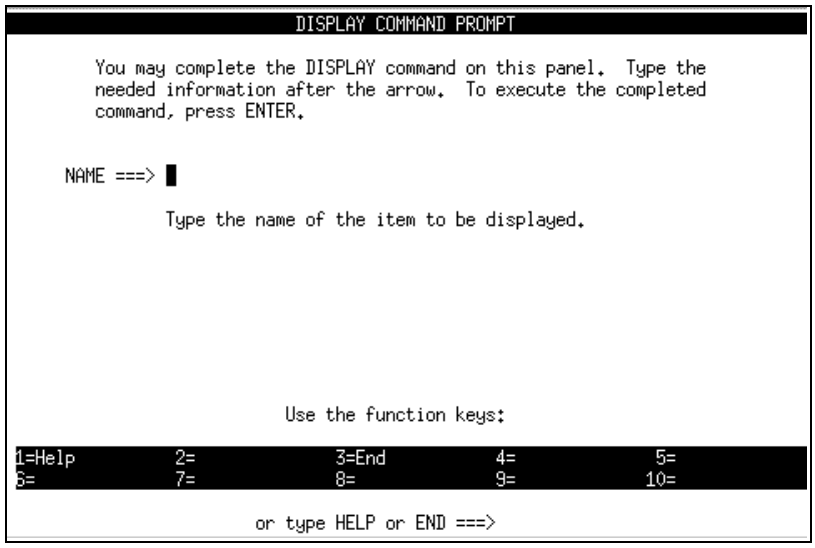
Displays database objects.

Syntax

DIsplay *name* | FORM | PROC | PROFILE | DBE | QUERY]

Screen

Type `DISPLAY` without options to see the Prompt screen.



Description

You can display an item by specifying its name, or you can display a screen showing a current form, procedure, profile, DBError or query. If none is current, the screen is empty.

Displaying DBError shows the last database error recorded, even if subsequent commands have execute properly.

Displaying a named table is equivalent to the SQL command “SELECT * from”

Example

```
DI PROFILE
```

displays the current profile (Set Command prompt) screen.

DRAW

Draws a specified object in the Query screen.

Syntax

```
DRaw [COND | COMM |[table_name [([TYPE[=]] [SELECT |INSERT | UPDATE])]]
```

where TYPEs are SQL Query specifications.

Screen

None. DRAW requires an object name.

Description

You can DRAW anywhere in TRIMqmr and the status line communicates the result of the command (for example, `"OK DRAW performed"`), but you must display a Query screen to see the query.

If you are on the Home screen when you DRAW, for example, you must issue a QUERY to display the results of the DRAW command on the Query screen.

Typing DRAW with no options on the QBE Query screen displays a table with blank column headings and fields.

You can DRAW the following:

- DR *table_nam*e where *table_name* is existing table shows the table outline on the QBE Query screen.
- DR COND creates a conditions box on the QBE Query screen.
- DR COMM creates a comments box on the QBE Query screen.

NOTE: *Remember, although you can DRAW anywhere, you cannot see the results of the action unless you display a Query screen.*

DRAW in SQL Query

The DRAW command does not have a command prompt and it behaves differently in QBE and SQL: on a SQL Query screen you must use a TYPE option and table name with the DRAW command.

Once you have a valid QBE query, you can convert it to SQL by issuing the command `D QBE AS SQL`.

For more information about using DRAW in SQL, refer to See ***"Using SQL"*** on page 82.

Using SQL, DRAW table_name has three options.

(TYPE=SELECT)

This type is the default. It creates a basic SQL SELECT that retrieves all rows of data in all columns of the named table. You can add other tables to the SELECT command with additional DRAWs.

```
DRaw table_name [(TYPE[=]SELECT)]
```

To add clauses and edit the SELECT command, move the cursor to the workspace and use [Insert] and [Del] to add and delete text.

(TYPE=INSERT)

Creates a SQL INSERT INTO that allows you to enter new rows of data. Information on name, datatype, length, and whether nulls are accepted appears for each column.

```
DRaw table_name [(TYPE[=]]INSERT)
```

To insert a row into the table, type values into the left column (identified as "ENTER VALUES BELOW" on the screen). Before processing the query, be sure that:

- Every column in the INSERT has value.

  If a specific column or columns have no data, you must delete the column. To delete a column, place the cursor in the row containing the column name and press [F4].

- Commas always follow a column name.

  If you delete the first column name, also delete the comma preceding the next column name.

- CHAR datatype column values, which are "literal values," are enclosed in apostrophes (single quotes). For example, 'OMAHA.'

Issue a RUN QUERY to actually add the new data to the database.

(TYPE=UPDATE)

This option creates a SQL UPDATE *table_name* SET … that allows you to enter data in selected columns as specified in a WHERE condition.

```
DRaw table_name [(TYPE[=]]UPDATE)
```

To update table data, you type values into the second column (identified by the "ENTER VALUES BELOW" heading on your screen). To update an entire row, type the new values and complete the WHERE clause.

Before processing the query, be sure that:

- Every column in the INSERT has value.

  If a specific column or columns have no data, you must delete the column. To delete a column, place the cursor in the row containing the column name and press [F4].

- Commas always follow a column name.

  If you delete the first column name, also delete the comma preceding the next column name.

- CHAR datatype column values, which are "literal values," are enclosed in apostrophes (single quotes). For example, 'OMAHA.'

RUN QUERY to actually update the database itself.

END

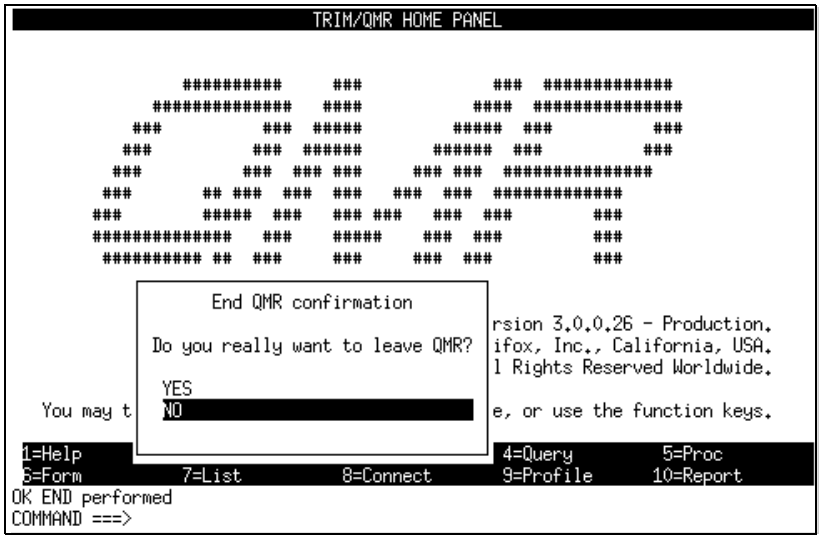
Ends the TRIMqmr session or exits the Help system or Prompt screen.

Syntax

```
End
```

Screen

Ending to exit TRIMqmr asks for confirmation.



Description

If you are on the Home screen, you are asked to confirm that you want to end the session. If you are on a prompt screen, End, or its function key equivalent, returns you to the screen that was previously displayed; on any other screen, it invokes the Home screen.

If you respond YES to the END QMR Confirmation prompt, your TRIMqmr session ends and you return to the operating system command line. If you enter NO, you return to the Home screen.

[F3] is available from most screens.

ENLARGE

Enlarges an object by adding character spaces, columns, or rows in the QBE screen.

Syntax

```
ENLarge (cursor in object) [Enter]
```

Description

ENLARGE is only valid on the QBE Query screen.

NOTE: *After execution, the command remains on the command line until it is deleted, replaced, or a different screen is displayed.*

[F5] is available on the QBE Query screen.

Examples

Increasing the width of an object by characters

1. On the command line, type `ENLarge`

   Do *not* press [Enter].

2. Position the cursor and [Enter] for each additional space.

   The cursor should be in the column or field o be enlarged.

To reduce the width of a column, place the cursor in the column field. To reduce a table name, put the cursor in the table name. To reduce the width of a conditions or comments box, put the cursor in the box field.

Adding or creating an empty, unnamed column in a table

1. On the command line, type `ENLarge`

   Do *not* press [Enter].

2. Position the cursor and [Enter].

To add a column to the right of a column, put the cursor in the column head to the left of the new column. To add a column to the right of the table name, place the cursor on the vertical line between the table head and the first column heading.

Adding an empty row to an object

1. On the command line, type `ENLarge`

   Do *not* press [Enter].

2. Position the cursor and [Enter].

To add a row to the bottom of a table, put the cursor in the last row of the table name column. To add a row below a specific row, place the cursor in the row above the new now in the table name column.

ERASE

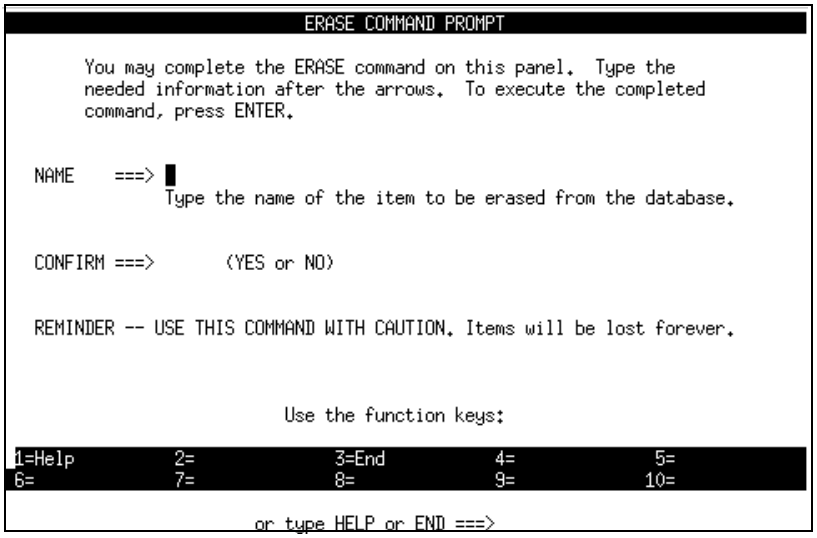
Removes an object from the database.

Syntax

```
ERase [name [(Confirm[=]Yes | No)]]
```

Screen

Type `Erase` with no options to see the Prompt screen.



Description

This command removes a form, procedure, query, table, or view from a database.

NOTE: *This command affects the database. Use with extreme care.*

---

name is the database object that you want to remove. When you specify form, all parts of the form are erased at once.

Confirm=Yes gives you a chance to confirm or reject the action before it removes the object.

Confirm=No does not display the panel.

If you don't specify the Confirm option, TRIMqmr uses the CONFIRM value set in the current profile.

(See ***"PROFILE"*** on page 54.)

EXIT

Immediately ends session, exits application.

Syntax

```
EXit
```

Screen

None. Immediate application exit.

Description

You can EXIT from a procedure screen, but not from a command prompt, confirmation, or Help screen.

EXIT erases all work that has not been saved— without displaying a confirmation prompt, no matter what setting you have in your profile.

EXPORT

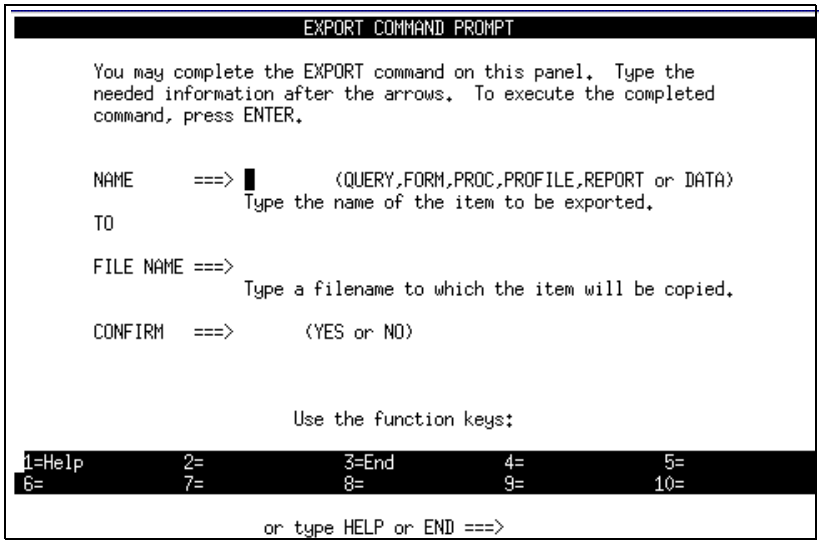
Copies an object to a file outside the database.

Syntax

```
EXPort [obj_name to '[file_name]' [(Confirm[=] Yes | No)]
```

Screen

Type *EXPORT* without options to see the Prompt screen.



Description

obj_name is an existing query, form, procedure, profile, or set of data.

Export is useful to transfer data from one database to another, especially if you only want certain columns, or rows, rather than the entire database.

The Confirm clause, especially useful when export is part of a procedure, allows you to include a prompt before the item is exported.

FORM

Displays Form screens.

Syntax

FORM[.*name*]

Screen

The FORM command has no prompt screen. The command defaults to the Main form screen, or if another has been displayed, the last-requested form screen.

Description

name is one of the following forms that you can use to format your TRIMqmr report:

- MAIN
- FINAL
- BREAK*n* (where *n* specifies the break level)
- PAGE
- COLUMNS
- OPTIONS

If the session has a report, the form screens show settings for that report. If you have executed a query that does not have a formatted report, the forms show default settings. Form screens appear empty if there is neither report nor query issued.

To cancel changes to a current form, type `RESET FORM`. The form reverts to the previously saved form or, if no SAVE FORM command has been executed, to the default form.

[F6] is available from the Report screen. From the Main form screen, you can access all other form screens with function keys.

For more detailed information on the specific form screens, review See ***"Forms Reference"*** on page 103.

FORWARD

Scrolls the workspace forward through the screen contents.

Syntax

Forward [*amount*]

Description

Forward lets you move backward by *amount* through a report, form, query, or procedure.

amount can be:

Csr	Moves the contents until the boundary is next to the cursor position. For example, `FORWARD CSR` moves the contents down until the top boundary is just above the line on which the cursor rests. If the cursor is not in the screen workspace, or is already on the boundary of the screen, then CSR has the same effect as PAGE.
Half	Moves the contents half the depth of the workspace.
Max	Moves the contents until the boundary is at the edge of the item being viewed. For example, `FORWARD MAX` moves the contents as far to the top of the screen as it can go.
Page	Moves the contents to the depth of the screen.
n	Moves the contents *n* character spaces where 1 > *n* > 9999.

If you don't specify an *amount*, TRIMqmr uses the setting shown on the `SCROLL ===>` area of the status line.

[F8] is available to scroll backward on the Form, Query, Report, and Procedure screens.

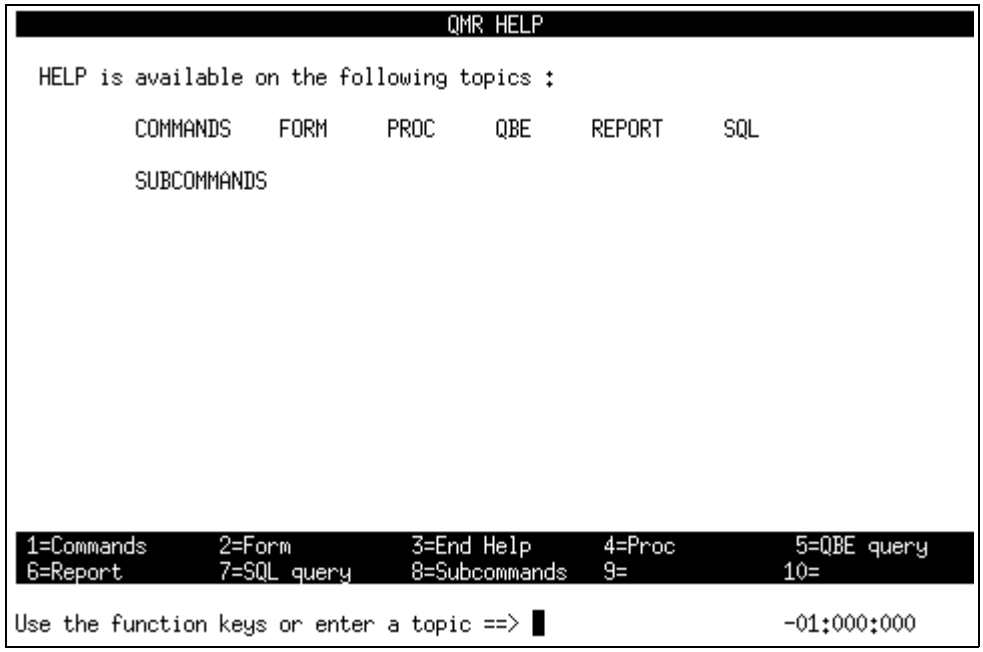
HELP

Invokes the TRIMqmr Help system, which contains information about commands and procedures as well as explanations of error messages.

Syntax

HElp [*name*]

Screen



Description

name is:

CAse	ERase	LENgth	REPort
Confirm	EXIt	LIst	RESet
CONNect	EXPort	PRint	RETrieve
DELete	FORM	PRINTER	Run
DIsplay	HElp	PROCedure	SAve
DRaw	IMPort	PROfile	SEt
DUplicate	Insert	Query	SYStem
ENLarge	LAnguage	REDuce	Width

The following commands allow you to navigate within help:

BAckward	BOttom	LEft
FOrward	TOP	RIght

The TRIMqmr Help screens contain information on commands and procedures as well as explanations of error messages. TRIMqmr attempts to select the Help screen that is most appropriate to the current situation. If you don't specify a command name, the Help screen displays information on the last screen or procedure shown. If you ask for HELP immediately following an error, TRIMqmr displays an explanation of the problem, usually with some suggestions for a solution.

Some Help topics give direct access to related topics or subtopics. For example, the Help screen for the SET command (which makes changes on the Profile screen) describes each of the options that can be used with that command.

In the Help system, you can select additional topics by typing the item or command on the command line or by using the function keys.

The function keys available to navigate through Help screens are:

- [F3] (End)— Returns to the screen from which you accessed Help.
- [F1] (More)— Moves to the next Help screen in a sequence.
- [F9] (Previous) — Returns to the last Help screen that was displayed.

IMPORT

Copies the contents of a file outside the database into a session screen.

Syntax

`IMPort [obj_name FROM 'file_name']`

Screen

Type `IMPORT` without any options to see the Prompt screen.

```
                        IMPORT COMMAND PROMPT

      You may complete the IMPORT command on this panel.  Type the
      needed information after the arrows.  To execute the completed
      command, press ENTER.


      NAME      ===> _          (QUERY, FORM, PROC, PROFILE, or DATA)
                     Type the name of the item to be imported.
      FROM

      FILE NAME ===>
                     Type a filename from which the item will be copied.




                           Use the function keys:

1=Help          2=              3=End           4=              5=
6=              7=              8=              9=              10=

                       or type HELP or END ===>
```

Description

obj_name is a form, procedure, query, profile, or set of data.

and

file_name is either a new name, or the same object as the one you are Importing.

The imported item replaces the current item; for example, an imported query replaces the current query; the Form settings are replaced by default settings for the imported query.

SQL queries and procedures can be created or edited outside TRIMqmr and the files imported. However, QBE queries cannot be edited outside TRIMqmr.

A file created outside TRIMqmr, or within and edited outside, must conform to the following rules:

- Lines must be exactly 79 characters long. Imported lines longer than 79 characters are truncated; shorter lines are padded with blanks.
- All characters must be valid in TRIMqmr. IMPORT transfers all characters, but invalid characters do not display.
- The file cannot be more than 256 lines long. For all platforms except DOS and MVS/VMS, IMPORT truncates files after the 256th line. DOS limit is 64 lines; MVS/VMS limit is 416.)

LEFT

Scrolls the contents of the workspace left.

Syntax

LEft [*amount*]

Description

Left lets you move sideways by *amount* through a report, form, query, or procedure.

amount can be:

Csr	Moves the contents until the boundary is next to the cursor position. For example, `LEFT CSR` moves the contents up until the boundary is just left of the line on which the cursor rests. If the cursor is not in the screen workspace, or is already on the boundary of the screen, then CSR has the same effect as PAGE.
Half	Moves the contents half the depth of the workspace.
Max	Moves the contents until the boundary is at the edge of the item being viewed. For example, `LEFT MAX` moves the contents as far to the left of the screen as it can go.
Page	Moves the contents to the depth of the screen.
n	Moves the contents *n* character spaces where 1 > *n* > 9999.

If you don't specify an *amount*, TRIMqmr uses the setting shown on the `SCROLL ===>` area of the status line.

[F9] is available to scroll backward on the Form, Query, Report, and Procedure screens.

LIST

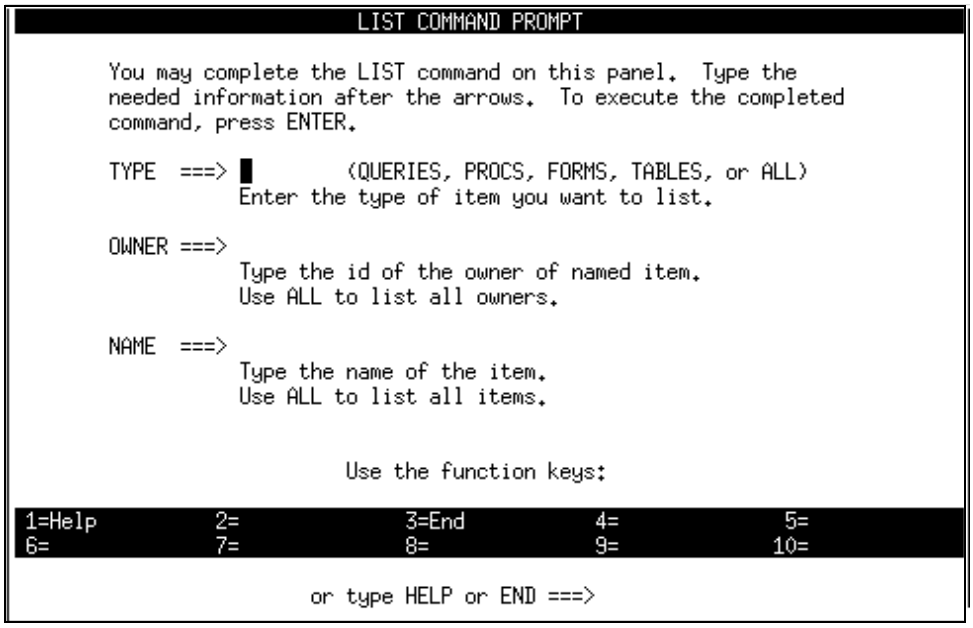
Displays a list of database objects.

Syntax

LIst [*type*] [Owner[=] *ownerid*] [NAme[=]*name*]

Screen

Type `LIST` to see the Prompt screen.



Description

LIST shows a list of QMR objects in the database: tables, queries, procedures, and forms. It shows only the items that you are authorized to see on the Item List.

- ***type*** — can be queries, procedures, forms, tables, or all.

  You must specify the type the first time you issue a LIST in a session. Once QMR generates an object list, LISTing again without specifying a type displays the previous list, enabling you to return to the list without regenerating it every time. Pressing [F7] produces the same result. The most-recently generated list remains in temporary storage until you execute LIST with a type or end the session.

- ***ownerid*** — defaults to your own user ID. Selecting an ownerid displays only items belonging to the specified owner.

  The ownerid can, but is not required to, be enclosed in single quotation marks. If the owner name contains embedded banks or quotation marks, then the name ***must*** be enclosed in single quotes.

  The ownerid can contain selection symbols, "%'' or "_'' to specify a list of owners of like values. The percent (%) specifies owners that contain any of the characters before or after the symbol.

The underscore (_) specifies a list of objects that contains a specific set of characters in a particular position.

If you specify ownerid=all, you see all qualified QMR items without regard to ownerid.

- ***name*** --- limits the list to objects that contain the specified character string.

  The name can, but is not required to, be enclosed in single quotation marks. If the name contains embedded banks or quotation marks, then the name ***must*** be enclosed in single quotes.

  The name can contain selection symbols, "%'' or "_'' to specify a list of names of like values. The percent (%)specifies names that contain any of the characters before or after the symbol.

  The underscore (_) specifies a list of objects that contains a specific set of characters in a particular position.

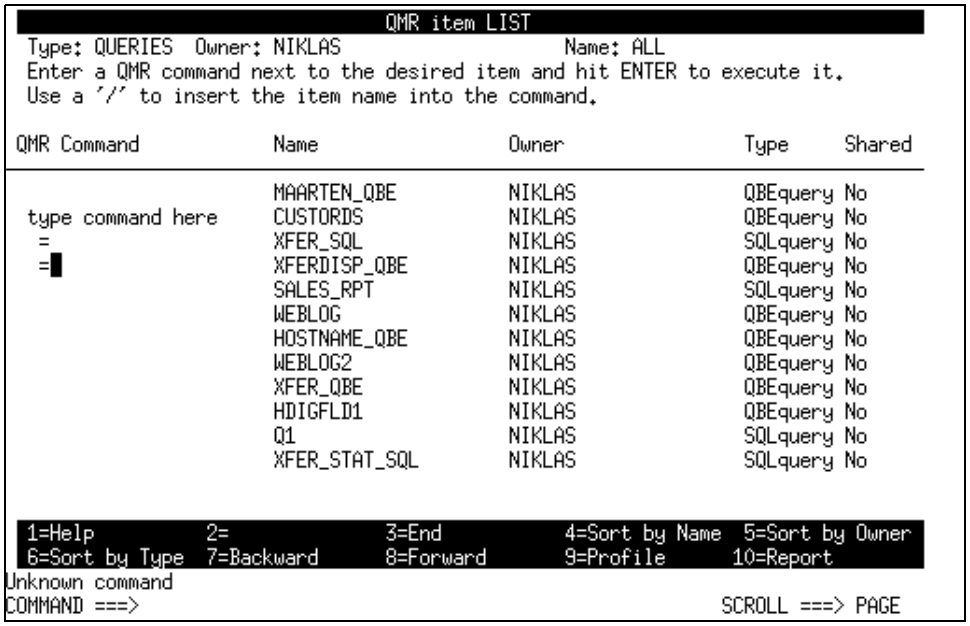
  Name=all indicates that no restrictions are placed on the list; this is the default value.

Issuing QMR Commands from LIST

You can use the List Panel to issue the following commands by typing them in the QMR command area (the first column).

On the LIST command screen, you can enter a command in the left column on the same line as the desired item. If you enter more than the command, such as a RUN command option, use a slash (/) to represent the item name in your command. For example, to place a form name in a RUN command, you must insert a slash to represent the query name:

```
RUN / (FORM=var01
```



You should not try to use IMPORT or EXPORT from the database object list.

PRINT

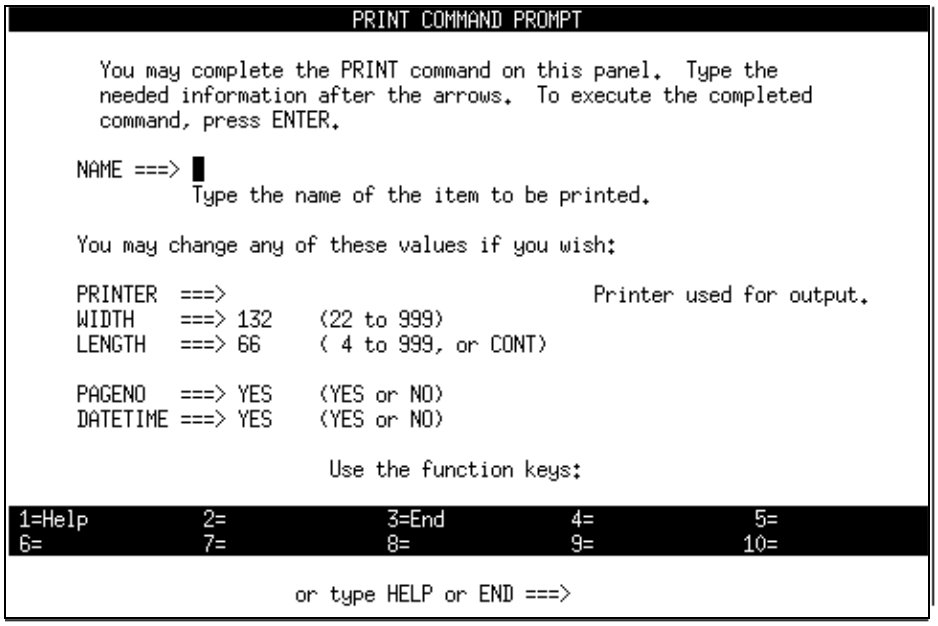
Prints a copy of an object.

Syntax

```
PRINT [object_name [([PRinter[=]printer]
[Width[=]n1] [Length[=]n2
[PAgeno[=]No | Yes] [Datetime[=]No | Yes])]]
```

Screen

Type `Print` without options to see the Prompt screen.



Description

PRINT prints a copy of a query, form, procedure, profile, or report.

When *object_name* is a report, the data is formatted as specified in the Form. If *object_name* is a procedure, query, or form, it is printed, and the contents are replaced. If name is a table in the database, the table is printed using a default form and the contents of Form are not changed.

If you do not specify a printer, QMR uses the setting on the profile screen.

If printer is set to DISPLAY, the current report (if any) is displayed on the terminal screen just as it is printed with headers/footers, data breaks, and so on.

Width can be from 22 to 999 characters (default is 132). When you print a report, lines longer than WIDTH are formatted on following pages, unless you have specified line wrapping on the Options screen. When you print other objects, such as forms or queries, lines longer than WIDTH are cut off at the right and the excess is lost.

Length can be from 4 to 999 lines (default is 66) or CONT (for continuous). If this option is not specified, TRIMqmr uses the setting from the Profile. CONT is not valid if you specify a printer name when printing a form.

Datetime determines when you print a report, whether the date and time are printed in the page footing.

Unless use specify datetime=no or set the &DATE and &TIME appropriately, they appear.

PAgeno controls whether page numbers are printed or not. The default is YES.

PROCEDURE

Displays the Procedure screen.

Syntax

```
PROCedure
```

Description

If there is a current procedure it appears on the screen; if there is no current procedure, a blank Procedure screen is displayed.

PROFILE

Displays the Profile screen.

Syntax

```
PROFile
```

Description

It is identical to the SET Command prompt.

(See ***"SET"*** on page 63.)

QUERY

Displays either the QBE or SQL Query screen, depending on the language specified.

Syntax

```
Query [QBE | SQL]
```

Description

You can specify either QBE or SQL with the QUERY command to override the current setting.

REDUCE

Reduces or removes (deletes) character spaces, columns, or rows in the QBE screen.

Syntax

```
REDuce  (cursor in object) [Enter]
```

Description

REDUCE is only valid on the QBE Query screen.

NOTE: *After execution, the command remains on the command line until it is deleted, replaced, or a different screen is displayed.*

The location of the cursor determines the action that is performed when you press [Enter] (or [F6]) with REDUCE on the command line.

Examples

Reducing the width of an object by characters

1. On the command line, type `REDuce`

   Do ***not*** press [Enter].

2. Position the cursor. To reduce the width of a column, place the cursor in the column field. To reduce a table name, put the cursor in the table name. To reduce the width of a conditions or comments box, put the cursor in the box field.

3. Press [Enter].

Deleting a column from an object

1. On the command line, type `REDuce`

   Do ***not*** press [Enter].

2. Move the cursor to within the column heading.

3. Press [Enter].

Deleting a row from an object

1. On the command line, type `REDuce`

   Do ***not*** press [Enter].

2. Move the cursor to the table name column in the row to be deleted.

3. Press [Enter].

REPORT

Displays the Report screen.

Syntax

```
REPort
```

Description

You must have a current, valid query on the Query screen. Otherwise, TRIMqmr returns a message on the status line and the current screen remains displayed.

RESET

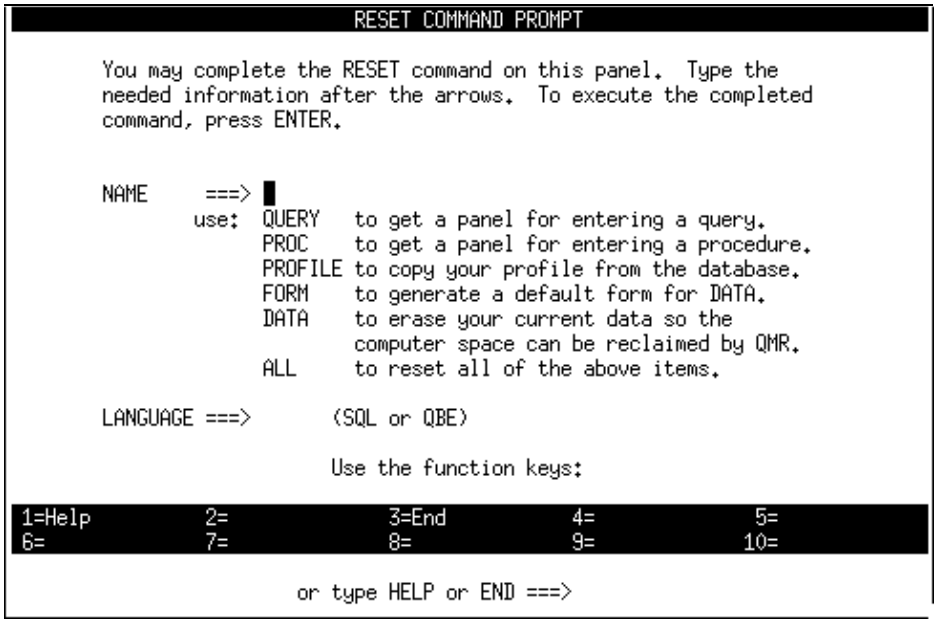
Restores an object to its previous state or to clear data.

Syntax

```
RESet [obj_name] [(Language[=]QBE | SQL)]
```

Screen

Type `RESET` to see the Prompt screen.



Description

RESET restores any of the following:

- ***Query*** — Displays a blank QBE or SQL Query screen.
- ***Procedure*** — Displays a blank Procedure screen.
- ***Profile*** — Cancels any changes made to the Profile since the last SAVE PROFILE.
- ***Form.x*** — Sets the specified Form screen to its default based on the data. If no Form is specified, generates a default form.
- ***Data*** — Clears the current data.

RETRIEVE

Displays commands that were entered on the command line. Also invoked by a question mark (?).

Syntax

```
RETrieve
```

Description

Commands are returned to the command line in the reverse order in which they were entered. Each time you enter a command, it is added to the stack of commands.

If the command contains an error, it is not executed but it is added to the stack of commands. The stack of commands remains the same until you press [Enter] with a command on the command line.

You can edit a retrieved command. Use [Back Space] and [Del] to erase characters. If INSERT is active; you can insert characters into a retrieved command.

RETRIEVE or a question mark (?) redisplays text that was previously entered on the command line. The text must have been entered interactively; thus, commands issued in batch mode, via command interface, or through a procedure cannot be RETRIEVEd.

You can RETRIEVE multiple times in succession (or use the ?) to see text that was entered several commands ago. Type a ? for each interaction in between the current and the one you want to retrieve (type ?s without spaces in between).

The confirmation message you receive after entering RETRIEVE indicates how far back the retrieved command was entered relative to the command that was most recently entered. When the oldest command has been retrieved, and you type RETRIEVE again, command retrieval wraps around, and the most recently entered command is again displayed.

After the text appears, you can [Enter] to reissue the command just retrieved. Characters in receiving text are converted (or not converted) into uppercase according to the CASE option specified in your profile.

When RETRIEVing a previously entered command, following these rules:

- Type question marks with no spaces between them or between the text and the adjacent question mark.
- On the command line type at least one space between the command and other text.
- Do not try to RETRIEVE from a procedure or from the database item list.

RIGHT

Scrolls the contents of the workspace right.

Syntax

RIght [*amount*]

Description

Right lets you move sideways by *amount* through a report, form, query, or procedure.

amount can be:

Csr	Moves the contents until the boundary is next to the cursor position. For example, `RIGHT CSR` moves the contents up until the boundary is just right of the line on which the cursor rests. If the cursor is not in the screen workspace, or is already on the boundary of the screen, then CSR has the same effect as PAGE.
Half	Moves the contents half the width of the workspace.
Max	Moves the contents until the boundary is at the edge of the item being viewed. For example, `RIGHT MAX` moves the contents as far to the right of the screen as it can go.
Page	Moves the contents to the depth of the screen.
n	Moves the contents *n* character spaces where 1 > *n* > 9999.

If you don't specify an *amount*, TRIMqmr uses the setting shown on the `SCROLL ===>` area of the status line.

[F10] is available to scroll backward on the Form, Query, Report, and Procedure screens.

RUN

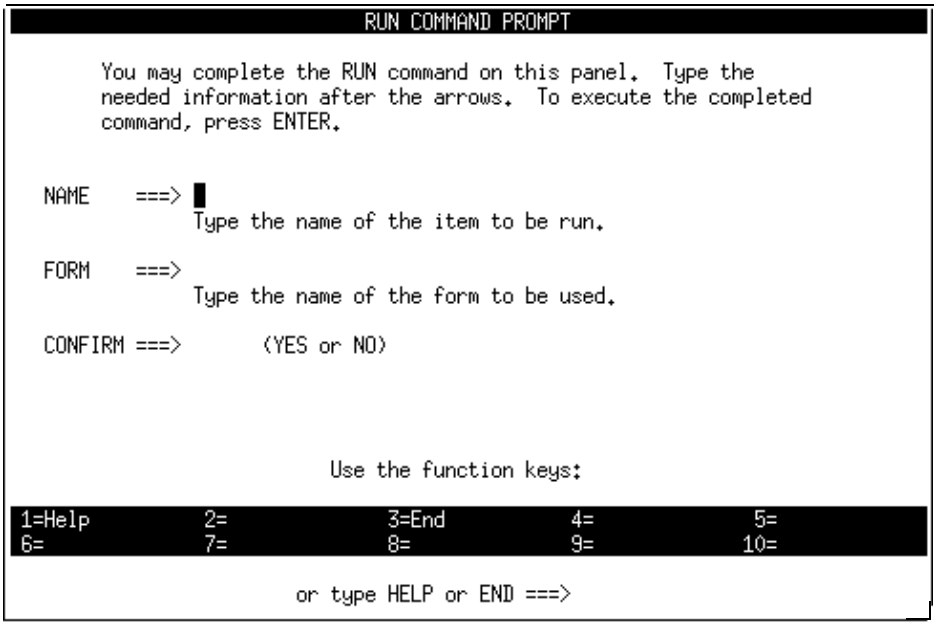
Executes a query or a procedure and puts the new results on the Procedure or Query screen.

Syntax

`RUN [name] [(Form[=]form_name] [Confirm[=]Yes|No][&variable=value]`

Screen

Type `RUN` to see the prompt screen.



Description

name specifies the name of a procedure or query to execute. If the object from the database, it also replaces the contents of the procedure or query temporary storage area in TRIMqmr.

If you RUN a query that selects data, the temporary form may also be replaced.

If the query or procedure belongs to another user, you must prefix ***name*** with the appropriate ***userid***:

```
RUN userid.query_name
```

FORM=*form_name* takes effect only if you RUN a query that selects data. This option isn't valid for a procedure. *form_name* can be FORM, or the name of a form in the database. If you don't specify a form for returned data, TRIMqmr uses the default form.

CONFIRM is only valid if you are RUNning a query that changes a table or view in the database. This option is not valid for procedures. `Yes` allows you to confirm or reject your action before the change is committed to the database. If you specify `No`, the change is committed without confirmation. If you don't specify a CONFIRM option, TRIMqmr uses the value in your profile.

&Variables

You can use up to 10 variables to specify values for the procedure or query at runtime.

```
&variable=[value]
```

When you execute a procedure that includes a query, you can assign a value to a variable in the query by using the form &&variable.

If you omit the variable and are running a query or procedure that uses variables, a prompt displays to allow you to enter values for the variables.

To assign a value to a variable in a query or a procedure within a procedure, use the &&variable form:

```
RUN proc-monrpt &&month = "JANUARY 1997"
```

The example assigns a value to the &&month variable in a query or a procedure within a procedure.

If the variable is in the procedure itself, use the &variable form.

The following restrictions apply to variables:

- Names may not be more than 17 characters plus the ampersand.
- Names are limited to the following characters:
  i. Letters of the English alphabet.
  ii. Special characters: @ # $: % ? ~ ' } |
  iii. Digits

(For more examples, see "*&Variables in SQL*" on page 85.)

SAVE

Stores the current contents of a named item, such as a query, in the database.

Syntax

```
SAve [name_1 AS name_2] [(Confirm[=]Yes | No] [COmment[=]'text']
[Share[=]Yes | No])]
```

Screen

Type SAVE to see the prompt screen.

```
                        SAVE COMMAND PROMPT

  You may complete the SAVE command on this panel. Type the needed information
  after the arrows. To execute the completed command, press ENTER.

  NAME    ===> _          (QUERY, FORM, PROFILE, PROC, or DATA)
               Type the name of the item to be saved.

  AS NAME ===>
               Type a name that the item will have in the database.
               If saving PROFILE do not type in this field.

  COMMENT ===>
               Type any text, within single quotes, to be saved with the item.

  CONFIRM ===>       (YES or NO)
  SHARE   ===>       (YES or NO)

                          Use the function keys:

1=Help          2=              3=End           4=              5=
6=              7=              8=              9=              10=

                      or type HELP or END ===>
```

Description

name_1 can be procedure, form, profile, or query and *name_2*, if it exists, must be the same type with the same characteristics.

For example, if they are tables, they must have the same number of columns and the columns must be of the same datatypes.

Confirm lets you confirm or reject your action before it is committed to the database. Confirm is not a valid option when saving a profile; if you do not specify confirmation, TRIMqmr uses the value in your profile.

Comment lets you store a comment with any saved object, except profile. The comment can be up to 57 characters long. Share allows you to specify whether the item is accessible to other users. It displays along with *name_2* in the LIST to all users who have authority. You cannot use quotes.

Note that the comments saved with the comment option are different than the ones you can enter on a query panel (which are only displayed when the saved query is displayed).

The Share option lets you determine who else can have access to a form, procedure, or query. This option is not valid for profile or data. For an object being updated (or replaced with the same name), the current value of share is left unchanged.

Examples

If the saved item is available to other users, any user can include the item in a command by prefixing the owner's id to the item name. For example:

```
RUN userid.query_name
```

SET

Changes specific attributes on the Profile screen from the command line.

Syntax

```
SEt    [name] [([CAse[=]Upper | Mixed | String]
       [Decimal[=]Period | Comma | French]
       [COnfirm[=]Yes | No]
       [LAnguage[=]SQL | QBE]
       [LEngth[=] 20 n 999; CONT]
       [Printer[=]printer, filename, or DISPLAY]
       [Space[=]space-name])]
```

Screen

Type `SET` to see the prompt screen.

```
                          SET COMMAND PROMPT

    CASE     ===> STRING   (UPPER, MIXED, or STRING)
    DECIMAL  ===> PERIOD   (PERIOD, COMMA, or FRENCH)
    CONFIRM  ===> NO       (YES or NO)
    LANGUAGE ===> QBE      (SQL or QBE)

    WIDTH    ===> 132      Number of characters per line for printing.
    LENGTH   ===> 66       Number of lines per page for printing.

    PRINTER  ===>                              Printer used for output.

  WARNING - These options are not usually changed. You should modify them
            only if you know what the resulting effect will be.

    SPACE ===>
              Type the name of the dbspace in the database to save DATA in.

                          Use the function keys:

1=Help          2=              3=End           4=              5=
6=              7=              8=              9=              10=

                       or type HELP or END ===>
```

Description

SET changes values in your profile for the current session, unless you type `SAVE PROFILE`, which makes the change permanent for all subsequent sessions.

- ***CAse*** — Specifies whether input from the keyboard is converted to upper case.

  `Upper` converts all input to uppercase.

  `String` converts input to uppercase except for characters enclosed in quotation marks (single or double); comments in queries and procedures; text fields on the forms.

  `Mixed` does not convert input. You must enter all operators and command names in uppercase and all column names in QBE queries must match the case of the names in the database.

- ***Decimal*** — Tells how large numbers are punctuated for display.

`Period` displays the number as 1,234,567.89; common in the United States.

`Comma` displays it as 1.234.567,89; common in much of Europe.

`French` displays it as 1 234 567,89; common in France.

- ***COnfirm*** — Specifies whether you want the opportunity to confirm or reject an action that changes a database object, external data set, or file. If you not specify this option, TRIMqmr uses the value in your profile.
- ***Width*** — Specifies the maximum number of characters to print on any line when using the PRINT command. Values can be from 22 to 999.
- ***LEngth*** — Specifies the maximum number of lines to print on any page when using the PRINT command. Values may range from 1 to 999 or be set to CONT (continuous printing). You cannot use CONT if you specify a printer name, or when printing a FORM.

  If you don't specify a length, TRIMqmr uses the value in your profile.
- ***Printer*** — Specifies the nickname of the printer to use. Defaults to the value set in the profile.
- ***LAnguage*** — Specifies the query language.

SYSTEM

Executes an operating system command without terminating your TRIMqmr session.

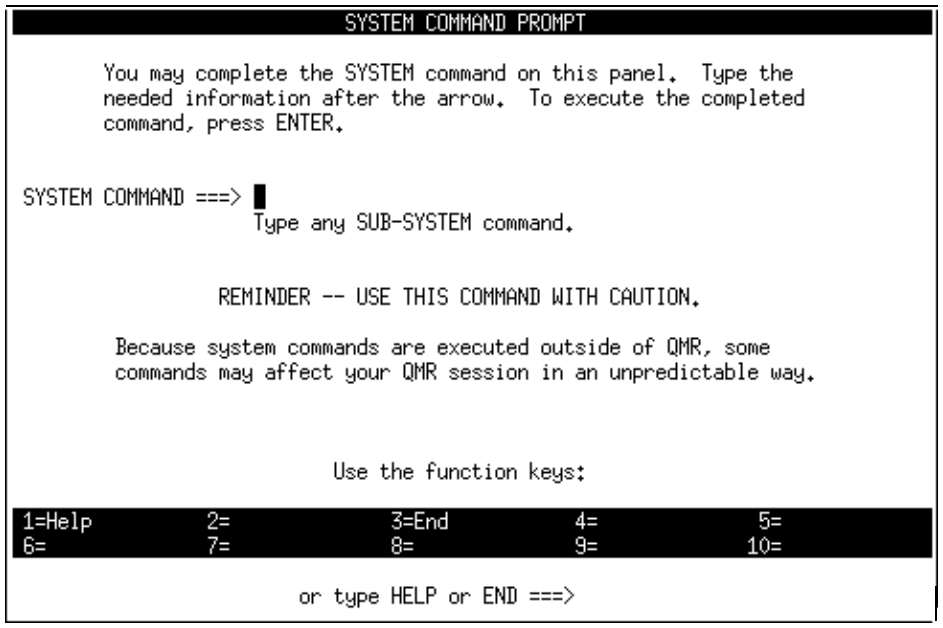
Syntax

`SYStem [command]`

command can be any operating system command.

Screen

Type `SYSTEM` to see the prompt screen.



Description

If you do not specify command, a system command screen is displayed and you are prompted for the command. If the command executes successfully, you are returned to the TRIMqmr screen from which you entered the SYSTEM command.

Chapter 4
QBE Controls

QBE Commands

Command	Brief Description
ALL.	Presents duplicate rows.
AND	Presents on two conditions.
AO[*n*].	Orders a query result in ascending order.
AVG.	Calculates the average value.
BETWEEN x AND y	Presents values within a range.
COUNT.	Counts the number of values in a column.
D.	Deletes a row from a table.
DO[*n*].	Orders a query result in descending order.
G.	Summarizes records into groups.
I.	Enters a row or rows into the database.
IN (x, y, z)	Presents specified values in a list.
LIKE '%abc%'	Presents a specified string of characters.
LIKE '_a_'	Ignores specified characters.
MAX.	Calculates the maximum value.
MIN.	Calculates the minimum value.
NOT	Presents the opposite condition.
NULL	Presents rows with no entries.
OR	Presents either of two conditions.
P.	Displays one or more columns of a table.
SUM.	Calculates the sum
U.	Updates or modifies a row or rows in the database.
UNQ.	Eliminates duplicate rows.

Note that a period follows each command.

Present — P.

You can present (P.) all the columns in a table, or just some of the columns.

Presenting All Columns

To see all columns of a table outline, put **P.** under the table name (the first column). Only named columns in the table are presented, but all named columns are presented.

This query:

```
                                QBE QUERY
                                                          Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+
  P.   |         |             |            |         |          |
```

Returns this data:

```
     Rows 1      thru 20          REPORT       Columns 1      thru 82


 PART                                    UNIT       ON       SOLD
  NO         DESCRIPTION                 PRICE     HAND      YTD
 ----------  --------------------  ----------  --------  --------
 TK1001      KEY CHAIN RECTANGLE         1.85      4753      7582
 TK1003      KEY CHAIN SLIM OVAL         1.85      2778      1042
 TK1005      KEY CHAIN OVAL              2.15     27690        35
 TK1007      KEY CHAIN ROUND 2"          2.15      2349      1294
 TK1011      KEY CHAIN TAG POINT         2.15      8975      4351
 TK1013      KEY CHAIN RECT. LG.         2.15      4521      4892
 TK1015      KEY CHAIN RECT. LGE.        2.30      8294     10528
 LO20        LETTER OPENER W/BOX         4.10      2150      1500
 LT014       LUGGAGE TAG W/STRAP         3.20      2500      4590
 SBLP1206    LICENSE PLATE FRAME        12.00       842      1049
 R510B1-A    PEN GRAY GOLD SWIRL         4.50     10500      3500
 R510B2-A    PEN BLACK GOLD SWIRL        4.50     15000     25000
 R510B3-A    PEN RED GOLD SWIRL          4.50      8400      7550
 R510B4-A    PEN GOLD SILVER SWIR        5.00     15450     25900
 R510B1-B    PENCIL GRAY GOLD SWR        6.00      4300      8525
  1=Help          2=              3=End          4=Query         5=Proc
  6=Form          7=Backward      8=Forward      9=Left          10=Right
OK ran QUERY
COMMAND ===>                                                SCROLL ===> PAGE
```

Presenting Selected Columns

To see only specific columns from a table outline, put **P.** under the names of the columns you want to see. If you have criteria for display, they can follow or precede the Present command.

This query:

```
                                   QBE QUERY
                                                          Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+
       | P.AO.   |             |            | P.      |          |
```

Returns this data:

```
      Rows 1      thru 20          REPORT       Columns 1      thru 82


 PART            ON
  NO            HAND
 ----------  --------
 BOX-DLX        12350
 BOX-PLS        15000
 LO20            2150
 LT014           2500
 R212C1          3500
 R212C2          8500
 R212C3          1400
 R212E1          4500
 R212E2          3200
 R212E3          1000
 R310G           8500
 R310S           2585
 R510B1-A       10500
 R510B1-B        4300
 R510B2-A       15000
 1=Help         2=              3=End          4=Query        5=Proc
 6=Form         7=Backward      8=Forward      9=Left         10=Right
OK ran QUERY
COMMAND ===>                                                SCROLL ===> PAGE
```

Columns appear in the same order as they are in the sample table.

To change the order of output in the report, change the order in the table outline by overwriting the column names. Make sure that you retype the names in their new positions as they appeared in their old ones. In fact, you may want to create a new column and name it, before deleting an old one, to ensure you type the name correctly.

You may need to Enlarge some columns to fit new names in.

Rearranging columns (the safe way)

1. With a table outline displayed on the query screen, move the cursor to the column name immediately to the left of the *new* column position.
2. [F5] (or type `ENLARGE` with the cursor positioned) to create the unnamed column.
3. Type the column name in the heading of the unnamed column.

4. Move the cursor to the column name in the old location and [F6] (or type `REDUCE`) to delete the column.

Adding Descriptive Columns

You can add columns for descriptive information to your report by adding a constant in the table outline in an unnamed column. The constant must be in single quotes:

This query:

```
                                   QBE QUERY
                                                          Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+----------------
       | P.AO.   |             |            | P.>5000 | P.<10000 | P.'SLOW MOVER'
```

Returns this report:

```
     Rows 1      thru 20          REPORT       Columns 1      thru 82


 PART            ON       SOLD
  NO            HAND      YTD   'SLOWMOVER
 ----------  --------  --------  ----------
 BOX-PLS        15000      8000  SLOW MOVER
 R212C2          8500      1500  SLOW MOVER
 R510B1-A       10500      3500  SLOW MOVER
 R510B3-A        8400      7550  SLOW MOVER
 TK1005         27690        35  SLOW MOVER
 TK1011          8975      4351  SLOW MOVER

 1=Help          2=              3=End          4=Query         5=Proc
 6=Form          7=Backward      8=Forward      9=Left          10=Right
OK ran QUERY
COMMAND ===>                                                SCROLL ===> PAGE
```

Arranging Data with AO., DO.

After you have selected the information you want, you can choose a priority order and whether or not you want duplicate rows presented.

To put rows in a report in ascending order by the value in a specific column, put **`AO.`** in that column. For descending order, use **`DO.`**

Be sure to use the letter “O” not the digit “0” in the command.

The sorting sequence for character data in ascending order is:

- Special characters, including blank.
- Lowercase letters, in alphabetic order.

- Uppercase letters, in alphabetic order.
- Numbers, in ascending order.
- NULL.

Numbers are sorted in ascending order.

To order by more than one column, simply put a (2) under the table heading of the column you want as second priority (either to sort ascending or descending order), and so on. This is called a sort priority; the sequence need not be complete, but no two columns can have the same priority. Thus, you can use 1, 2, and 5 but not two 2s.

Eliminating Duplicate rows with ALL., UNQ.

You can control the display of duplicate rows in a report by specifying **`UNQ.`** to eliminate duplicate rows or **`ALL.`** to specify that ALL rows be displayed. The operators go in the same row as the **`P.`** operator.

If your table outline contains two or more P. designations, UNQ. is the default operator and you do not have to specify it. However, if your table has only one P. row, and you want to prevent duplicate rows, use UNQ. under the table name in the row with P.

Refining Your Query

The following operators and commands help you restrict the data, in accordance with the items that interest you.

Presenting on Conditions

To display only rows that have a certain value in some column, you put the value under the column in the table outline. That value is what is called a ***condition***. The query selects only those rows of a table (or tables) that are indicated by the condition.

In the following example, the condition states that the rows returned should be for items that have more than 10,000 units on hand, eliminating items that have smaller current inventory.

```
                                   QBE QUERY
                                                          Row 1    Column 1

INVENT | PART_NO | DESCRIPTION | UNIT_PRICE | ON_HAND | SOLD_YTD |
-------+---------+-------------+------------+---------+----------+
 P.    |         |             |            | >10000  | DO.      |
```

You can write a condition using any of the following “logical” conditions, including equality and inequality. You can also write expressions that include:

Condition	Operator
Equal	=
Less than	<
Less than or equal	<=
Greater than	>

Condition	Operator
Greater than or equal	>=
Not equal (unequal)	< >
Values within a range	BETWEEN
Values from a list	IN (a,b,c)
Partial strings	LIKE '%abc%' or LIKE '_a_'
Negative conditions	NOT

BETWEEN, IN

Use the BETWEEN operator to select all rows that have a value between two limits. The limits are inclusive. The operator precedes the limits, and limits are separated by the keyword AND. For example:

```
BETWEEN value_1 AND value_2
```

or

```
BT value_1 AND value_2
```

Use the IN operator to select all rows that contain any of a list or set of values. When you use IN:

- Do not use NULL in a set of values.
- Separate values with a comma; blank spaces are permitted.
- List values in any order.

  With the cursor in the column field for which you're creating the condition, type `IN value_1,value_2,value_3,…`

  If you are not Presenting the entire table, don't forget to add a `P.` in the column field as well. Also, you may have to enlarge column fields to complete the conditions statement. Consider using a conditions box.

Presenting on Part of a Value

Use LIKE to select rows that contain a string of characters that can be part of a word or number you know to exist in the data. Use percent (%) at either or both ends of the string to specify substitution of any number of characters (this is called a ***wildcard***.

The underscore (_) at either or both ends of the string indicate a specific number of any character. You can use both percents and underscores in the same condition.

Use an apostrophe at the beginning and end of the string, including the special characters. For example, `'%string%'`.

When you use LIKE to search for data with a specific ending, you need to know the column's data type. If it has a specific width, and the data in the column varies in width, you have to add blanks to the character string (probably with a%) to match the total number of characters allowed in the column.

Also, note that LIKE only works with character data.

Presenting Calculated Values: Expressions

You can use QBE not only to select data that is already in a table, but also results that are calculated by using that data by using an ***expression***.

An expression is an arithmetic operation that involves data from one or more columns. You can use the result of an expression as the data for another column, or in a query search condition. Only numeric data types work in expressions; if you try to use a NULL, the result is a NULL.

Expressions can include column names, or synonyms, constants, and built-in functions with the following operators:

Operation	Operator
Addition	+
Subtraction	-
Multiplication	*
Division	/

Expression elements are evaluated in the following order:

- Built-in functions.
- Plus or minus sign before a single value
- Multiplication and division of two values
- Addition and subtraction of two values
- All other operations, left to right

Variables

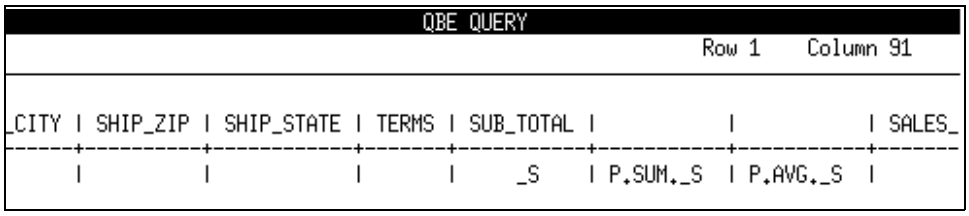
You can use variables in QBE queries, just as in SQL queries. Variables are designated by a preceding ampersand (&) and represent a value that can be chanced each time the query is run. Each time you run a query with variables, the Run Command Prompt appears, allowing you to enter values for all specified variables.

Unnamed Columns and Target Tables

In many cases, columns adequately represent the report you need to create. For some complex queries, however, unnamed columns and target tables let you:

- Combine data from two or more columns into one.
- Include constant values in a report.
- Apply built-in functions to grouped data.
- Combine columns from two or more tables into one report.
- Combine rows from two or more tables.

You can produce reports that contain columns that show the values of expressions: simply put the expression in an unnamed column. (The column name for the calculated value depends on settings in your database.)



Adding an empty column

1. Put the cursor in the column name to the left of the new column's position.
2. Press [F4] or type `ENLARGE` and [Enter].

You can also create a target table, an empty table outline that uses synonyms to refer to other outline tables. Anything you can put in an unnamed column can go in a target table.

Adding a target table outline

1. On the QBE screen, type **`DRAW`** without specifying a table name.
2. Use synonyms from table outlines you want to combine, with any expressions, in the column fields.
3. Format the report as necessary.
4. SAVE your work.

Unnamed columns and target tables have some restrictions.

They cannot be used to:

- Name the column for your report.

  QMR names columns created by expressions. To change the column name, you must use a form.
- Write a condition. Use a conditions box, instead.
- Define a synonym; they must be defined in a named column of the table outline.

Presenting Rows with No Values: NULL

If a table is only partially filled with data, the locations in which nothing is entered contain a code word called NULL, or "value unknown." NULL is ***not***:

- A numeric value of zero.
- A character string of all blanks.
- A character string of length zero.
- The character string N U L L (of length 4)

Each of these is a legitimate value that can exist in some row and column of a table. NULL occurs where no value has been entered or where the value has specifically been set to NULL. It prints and displays as a single hyphen (-).

To select rows that have no entry in a column, put the word NULL in that column.

To select rows that are "not NULL" use the NOT operator in the condition.

Presenting on Negative Conditions: NOT

Negative conditions produce query results that are ***not equal*** to a specified value or range of values. They can be useful when you want to produce a query result that excludes a given range of rows. You create a negative condition by adding the keyword NOT before the condition.

- Conditions preceded by NOT are negated before they are connected by AND or OR.
- Use parentheses in complex conditions to make the NOT condition clear.
- Do not try to use NOT with the symbols <, >, <=, or >=, in a conditions box.
- You can use NOT as in NOT NULL, NOT LIKE, NOT IN, and NOT BETWEEN.

Multiple Conditions: AND, OR

Multiple conditions are two or more basic conditions that are linked by an operator: AND or OR.

In a conditions box, two conditions connected by an AND retrieve rows that satisfy both conditions. In a table outline, the AND is implied by placing the conditions on the same row.

Or connects two conditions that retrieve rows that satisfy either or both conditions. In the table outline, putting two more conditions in different rows is equivalent to connecting them with OR.

If you use both AND and OR in a conditions box, you need to use parentheses to establish precedence according to the following rules:

- You can use more than one level of parentheses. Multiple parentheses are evaluated from the most nested outward.
- Without parentheses, all conditions connected by AND are evaluated from left to right, and then all the conditions connected by OR, from left to right.

Conditions Boxes

You can use a conditions box for any condition, but it is necessary when you want to:

- Refer to two or more columns in a condition.
- Use a built-in function, like AVG.
- Avoid widening a column of the table outline more than once for a long condition.
- Use the `AND` or `OR` operator in a condition that requires example elements.

- Use parentheses in a complex condition to change the order of precedence.

To add a conditions box to your query, type `DRAW COND` and [Enter]. If you type `DRAW CONDITION` or `DRAW CONDITIONS` you'll see a table outline of that name rather than a conditions box.

You can use more than one conditions box, or more than one condition in any box; however, each condition must fit on a single line in a box.

Two Conditions Equivalent to AND

Putting more than one condition in one or more conditions boxes is equivalent to connecting them all by `AND`, since the rows selected in such a query must satisfy all conditions in both boxes.

If, however, you use more than one row in the table outline and the conditions in the boxes refer to different rows, you create an `OR` condition.

Using Synonyms

In this example, the conditions are linked by use of a common symbol, or synonym (also called example element, or column identifier).

```
                                  QBE QUERY
                                                          Row 1    Column 91

_CITY | SHIP_ZIP | SHIP_STATE | TERMS | SUB_TOTAL |           |           | SALES_
------+----------+------------+-------+-----------+-----------+-----------+-------
      |          |            |       |     _S    | P.SUM._S  | P.AVG._S  |
```

Synonyms:

- Must begin with an underscore (_).
- Can have any combination of up to 17 letters or numbers.
- Can be used alone, or with one or more QBE commands.
- Must appear in a query at least twice: once to define it in the table outline, and one (or more times) in a condition or (as above) calculation.

  Used in a condition or calculation without begin defined causes an error message. If you define an example element without using it, TRIMqmr sends a message saying that the example element was ignored.

  If you define a synonym more than once, you are writing an implicit condition that says that the several defined values must be equal to each other.

  Used in a join, the columns with the same identifier are assumed to be equal to each other.

Joining Tables

You can use QBE to join two or more tables simply by identifying the "common" column in each table with a synonym.

Grouping Data: G.

You often want to group retrieved data to show information in a report. Grouping by a specific column means dividing the table into groups of data, each group applying to that one column. Grouping data with the `G.` command allows conditional selection of groups.

`G.` accumulates the results by group, but does not necessarily order the groups, or perform any other actions. Generally, `G.` produces one group for each set of identical values in a column. If NULLs exist, each one is treated as a separate group.

`G.` can appear in more than one column. In that case, rows are grouped by the combined `G.; G.` and `AO.` can be in any order when used together. A grouped row can also use the `P.` command if the results are meaningful.

A row that includes `G.` cannot use `I.`, `U.`, or `D.`, operators that change the contents of the tables in a database.

Changing the Database

QBE has three operators that let you add, modify, and delete rows in any database table according to your access privileges.

Whenever you modify a database object, TRIMqmr displays the Run Confirmation prompt. The prompt gives you the opportunity to respond YES and make the change, or respond NO and return to the QBE Query screen without the modification having been made.

Inserting Data

To insert a row of data, place the INSERT command code "I." in the table name field and enter the data for the new row in the column fields. If you leave a blank under a column, or omit a column from the table outline, a NULL is inserted in that column in the database. You must specify values for all columns that are defined as NOT NULL.

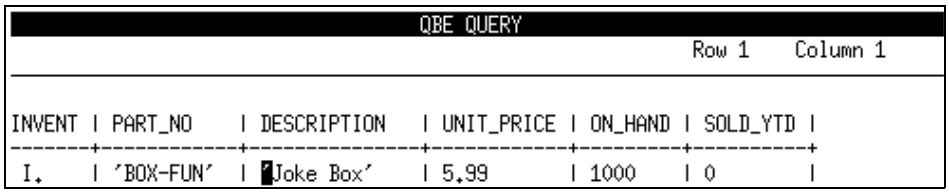
You can insert multiple rows by adding blank rows to the table outline by pressing [F5].

Inserting a row of data

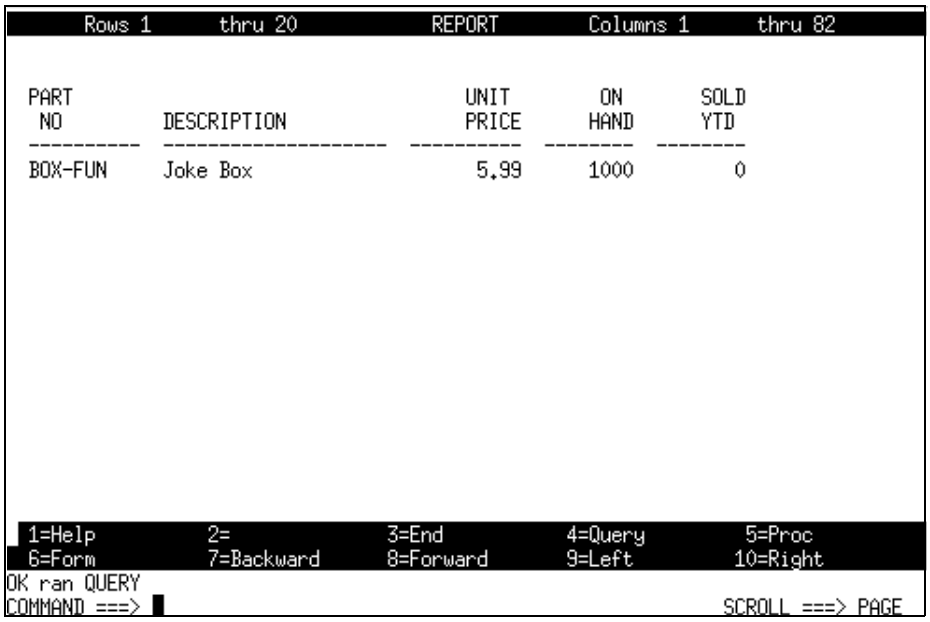
1. In the table name field enter **`I.`**
2. Type new data in each appropriate field; enclose alphanumeric datatype values in single quotes (‘’).
3. With the cursor in the table name field, press [F5] for ***each new row***, and add the data. Repeat for as many new rows of data as you have.
4. Press [F2] to run the query and add the row to the table.

The query below contains three new items to add to the INVENT table. The additional two lines were placed in the table outline by placing the cursor in the table name field and pressing [F5] twice.

This query:



Adds this line to the database:



Updating Table Data

To update a row of data, identify the row that is to be modified by placing a unique value in the appropriate column. Place the UPDATE command code "U." in each of the columns to be updated and include the new values for each column.

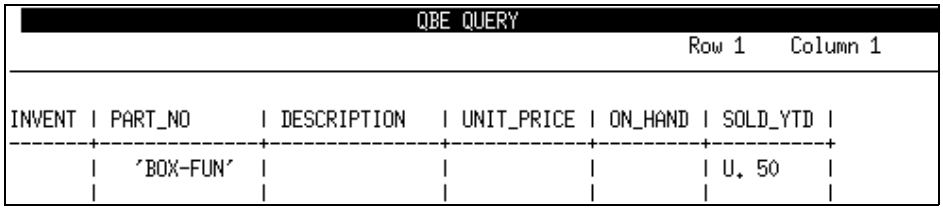
You can update all rows in the table by placing the UPDATE command code "U." in the table name field. Enter the data or formula for the change in the appropriate column fields.

Updating a row

1. Move the cursor to the column field that is to be used for the unique value, in this case PART_NO.
2. Type the unique values in the field to identify the row.

   You can update multiple rows by adding blank rows to the table outline. Add a blank line for each IND row to be updated. An individual row that is being updated must be identified by a unique value.

3. Move the cursor to each of the column fields that is to be updated and enter the new data following the UPDATE command code:



4. When the new row of data is complete, press [F2] to add the information to the table in the database.

☞ *Deleting rows of data*

To delete a row of data, place the DELETE command code "D." and identify the row or rows to be deleted by placing a value in an appropriate column, or specify conditions that select a specific row or rows of data.

NOTE: *If you don't specify rows, DELETE removes ALL data from the table.*

☞ *Deleting rows of data from a table*

1. Move the cursor to the table name field and type **`D.`**

NOTE: *One D. removes all the listed rows.*

2. In the appropriate column fields or conditions box, enter the values and conditions necessary to identify only the rows to be deleted.
3. Press [F2] to delete the row or rows from the database table.

To delete all data from the table

Move the cursor to the table name field, type **`D.`**, and press [F2}. ALL rows of data from the database table are removed, but the table remains in the database.

Query Summaries and Totals

QBE's built-in functions act on a group or set of rows and require that you use a new, unnamed column to display their results. You can use these functions on numbers, and in some cases, character and datetime data.

Function	Description
AVG.	Average value of column
COUNT.	Number of different values.
MAX.	Highest value in column

Function	Description
MIN.	Lowest value in column
SUM.	Total of column

Presenting an Average Value --- AVG.

The AVG. column function calculates the average values for all selected rows in a column. It applies to a group of numbers and returns a single value for each group of numbers to which it is applied. You can use UNQ. operator with AVG. to request that only unique values be used when calculating the average.

Rules for AVG.

- AVG. can only be used on columns of numeric data type.
- AVG. must be specified in an unnamed column along with the column heading (or synonym) that identifies the column to be averaged.
- The item that follows AVG. can be a synonym, an arithmetic expression containing at least one synonym, or the UNQ. operator, followed by a synonym. If the column to be averaged is defined by an arithmetic expression, the expression must be enclosed in parentheses.
- When you apply AVG. to one column named in a table outline, you must apply a column function (AVG., MIN., MAX., COUNT., OR SUM.) or the G. (group) operator to every other column to be presented.

Presenting a Total Value — SUM

SUM. column function calculates the total value for all selected rows in a column. It applies to a group of numbers and returns only one value for each group of numbers to which it is applied. You can use the UNQ. operator with SUM. to request that only unique values be used when calculating the sum. Nulls are ignored. If all values in the specified column are null, the total is null.

Rules for SUM.

- SUM. can only be used on columns of numeric data types.
- It must be specified in an unnamed column along with the column heading (or synonym) that identifies the column to be totaled.
- When you apply SUM. to a column named in a table outline, you must apply a column function (AVG., SUM., MIN., MAX., COUNT.), or the G. (group) operator to every other column to be presented
- The item that follows SUM. can be an synonym, an arithmetic expression containing at least one synonym, or the UNQ. operator, followed by a synonym. If the column to be totaled is defined by an arithmetic expression, the arithmetic expression must be enclosed in parentheses.

Presenting a Minimum Value — MIN.

The MIN. column function returns the smallest value in the group of numbers or characters in a specified column. It can be applied to columns of any type.

If MIN. is applied to a character data column, use the alphameric ordering.

- The number 9 is greater than 8, and so on to 0 (zero).
- Zero is greater than the uppercase *Z*, which is greater than *Y*, and so on to *A*.
- *A* is greater than the lowercase *z*, which is greater than *y*, and so on to *a*.
- Lowercase a is greater than the special characters, which are ordered last.

TRIMqmr ignores null values when searching for the minimum. If all values specified in a column are null, no value is returned.

Rules for MIN.

It must be specified in an unnamed column along with the synonym that identifies the column from which the smallest value is to be retrieved. This synonym also appears in the column containing the value to be retrieved.

The item that follows MIN. can be a synonym or an arithmetic expression containing at least one synonym.

When you apply MIN. to a column named in a table outline, you must apply a column function (AVG., SUM., MIN., MAX., COUNT.), or the G. (group) operator to every other column to be presented.

Presenting a MAXIMUM Value — MAX.

The MAX. column function returns the greatest value in the group of numbers or characters in a specified column. It can be applied to columns of any type.

If MAX. is applied to a character column, use the alphameric ordering.

The number 9 is greater than 8, and so on to 0 (zero).

Zero is greater than the uppercase *Z*, which is greater than *X*,

and so on to *A*. *A* is greater than the lowercase *z*, which is greater than *x*, and so on to *a*. Lowercase *a* is followed by the special characters.

QMR ignores null values when searching for the maximum. If all values specified in a column are null, no value is returned.

Rules for MAX.

It must be specified in an unnamed column along with the synonym that identifies the column from which the largest value is to be retrieved.

This synonym also appears in the column containing the value to be retrieved.

The item that follows MAX. can be a synonym or an arithmetic expression containing at least one synonym.

When you apply MAX. to a column named in a table outline, you must apply a column function (AVG., SUM., MIN., MAX., COUNT.), or the G. (group) operator to every other column to be presented.

Counting a Number of Values — COUNT.

The COUNT. column function finds the number of unique values in a column. COUNT. can be abbreviated to CNT. COUNT. can be followed by UNQ and then a synonym, or by only a synonym. In either case, COUNT is unique.

Use COUNT. to count values in columns of any data type. For example, you could determine the number of employees with a salary in a given range, or the number of employees at a given location by adding a search condition.

Rules for COUNT.

- COUNT. can be used to count values in either numeric or alphabetic columns.
- COUNT. counts only unique values.
- A synonym must follow COUNT. or CNT.
- COUNT. cannot be followed by an expression or a synonym within an expression.
- COUNT. must be applied to a group of data. Attempting to present all columns of a table, with COUNT. specified in one of the columns, produces an error message.
- COUNT. cannot be specified for more than one column per query.

Chapter 5
Using SQL

This chapter briefly discusses SQL, focusing on using QBE to generate SQL Queries. TRIMqmr allows you full functionality of your database's SQL language.

Introduction

SQL is the standard language for relational database management systems. SQL uses English-like commands to communicate with a database management system and let you retrieve data as well as manage your database. With very little data processing experience, you can learn SQL's basic features and use them effectively. If you have extensive data processing experience, you are probably already familiar with the power and flexibility of SQL.

Like QBE, SQL allows you to:

- Perform queries on complex conditions.
- Calculate values and select data on these values.
- Modify the database.
- Combine data from two or more tables.

In addition, you can:

- Perform outer joins on database tables.
- Execute subqueries.

SQL Query Screen

The SQL Query screen, accessed by pressing [F4] from Home when the language is set to SQL, or by pressing [F4] from the QBE Query screen, provides the workspace for constructing your SQL queries.

Command	Key	Description
Help	[F1]	Displays the Help screen for QBE Query.
Run Query	[F2]	Displays the Report screen with the current QBE query results in the default or current report format. If there is an existing report format, the default format prompt is displayed. To display the query in its default format, respond YES to the prompt; to return to the QBE Query screen, respond NO to the prompt.
End	[F3]	Displays the Home screen.
Delete	[F5]	Deletes the entire SQL query line in which the cursor is located.
Duplicate	[F5]	Duplicates the SQL query line in which the cursor is located.
Proc	[F6]	Displays the Procedure screen.
Backward	[F7]	Displays the portion of the SQL query above the portion currently displayed; if the work area displays the top of the query, a message is displayed on the status line. The number of lines moved is determined by the Scroll setting in the bottom right corner of the SQL Query screen.
Forward	[F8]	Displays the SQL query below the portion currently displayed; if the last line of the query is in the work area, a message is displayed on the status line. The number of lines moved is determined by the Scroll setting in the bottom right corner of the SQL Query screen.

Command	Key	Description
Insert	[F9]	Inserts a blank line immediately above the line on which the cursor is located.
Report	[F10]	Displays the results of the current query in the default report format. If there is an existing report, the default form prompt is displayed. Respond YES to delete the current report format and display the results of the current query in the default format; respond NO to return to the SQL Query screen.

QBE as SQL

While QBE provides a graphical representation of the data and the queries you issue against it, SQL can add power and flexibility to your information assessment tasks.

With TRIMqmr you can create your query graphically, and review it in SQL with a single key command, [F4], called ***D QBE as SQL*** or you can create the query and edit it directly in SQL.

Creating a QBE query and converting to a SQL query is useful if the query includes a number of columns, all columns in a table, or columns from two or more tables. In most cases, the decision to use QBE or SQL for database queries is a personal choice.

The Profile screen and the TRIMqmr SET command determine whether the QBE or SQL Query screen appears as default. In a previous exercise you checked to be sure that QBE was set as your default.

Remember that unless you explicitly make changes permanent, they remain in effect only through your current TRIMqmr session. If you end the session without saving the new profile, the settings revert to the previous defaults when you begin a new session. To make the changes permanent, you type `SAVE PROFILE` on the command line.

Creating a SQL Query in QBE

1. On the QBE screen type **`DRAW CUST`**.
2. Type **`DRAW ORDS`**.
3. Type **`P.`** in the column fields for CUST_NAME, SALES_REP in the CUST table.
4. Type **`P.`** in the column fields for ORDER_NO, ORDER_DATE, and TOTAL from the ORDS table.
5. Type `_CN` in the CUST_NO column of ***each*** table to join them.
6. [F2] to validate the query and display the retrieved data.
7. Now, [F4] to return to the QBE Query screen.
8. [F4] to display the query on the SQL Query screen.

You can edit the SELECT statement in SQL, or save it, and run it from the SQL Query screen as a SQL query.

9. On the SQL Query screen, rearrange the columns so they look like this:

```
SELECT  A.SALES_REP,A.CUST_NAME,B.ORDER_NO,
```

```
        B.ORDER_DATE, B.TOTAL
FROM CUST A,ORDS B
WHERE(A.CUST_NO=B.CUST_NO)
ORDER BY 1 DESC,5
```

10. [F2] to display the data in the edited format.

NOTE: *To return to the QBE screen from SQL, you must type* **QUERY**. *Changes made to the query on the SQL Query screen do* ***not*** *transfer to the QBE Query screen.*

You can see from the exercise that the TRIMqmr DRAW command can be a convenient way to create a basic SQL SELECT for future editing before you run a query.

&Variables in SQL

You may want to create a general-use query for multiple uses. This query might always access a specific table, but search on different criteria that you want to set at runtime. The criteria that are set at runtime are called variables. You identify the variable in the SQL statement by prefixing the variable name with an ampersand (&). When you run the query, you include a value for the variable. For example, you can execute a stored query from the List screen by typing RUN in the TRIMqmr Command column of the query line, and typing a value for the variable before [Enter].

If you try to execute a query that contains a variable without supplying a value, the Run Prompt screen appears with a prompt field in which you enter the missing value.

Creating a query with a variable

1. On the SQL Query screen command line, type **RESET QUERY** and [Enter].
2. In the workspace area, complete the following:

```
SELECT part_no, quantity, order_date
FROM orditem
WHERE order_date > &begindate
AND order_date < &enddate
```

3. [F2].
4. On the Run Prompt screen, for *&begindate* type **'01-MAR-1990'**
5. For *&enddate* type '**31-MAY-1990'**
6. [Enter] to see the result set for the three-month period.

Summary

If you are familiar and comfortable with your database's SQL language, this screen lets you access the full power of SQL and retain the formatting benefits of TRIMqmr.

Chapter 6 Creating & Formatting Reports

Whenever you run a query, TRIMqmr uses a ***form*** to produce a report, or display, for the information you request. If the default form does not meet your needs, you can tailor, or modify, reports to your exact specifications. This chapter covers report terminology, how to reach and use the screens, as well as how to change, save, reset and print reports using various forms.

Types of Reports

Three basic report types are available:

- ***Basic Reports*** — the default report that resembles the tables in a QBE Query screen.
- ***Break Reports*** — generates intermediate and summary results. The results can be totals, summaries, statistics, messages, or blank lines to make the report more readable.
- ***Across Reports*** — provide a convenient way of presenting information in a matrix format. Spreadsheets are a typical example of across reports.

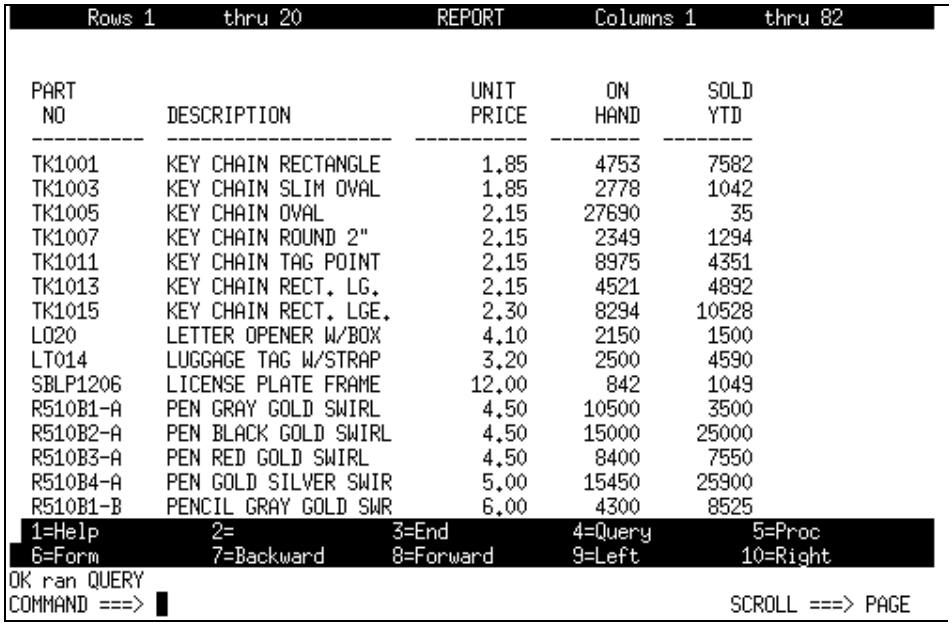
Seeing Reports On Screen

The report displayed on your monitor is composed of a single page. Thus, page breaks, even if they are specified explicitly, do not appear online. If you want to see the printed version of the report online, you must specify DISPLAY as the printer when issuing the PRINT REPORT command.

Unless you specify a form name on the command line, the report appears with default settings. The Report screen shows the portion of your report that is currently displayed in the window. The row numbers appear in the upper left of the screen and the column numbers in the upper right. If more of the report exists above/below or left/right of the current portion, you can view the areas by pressing a function key or typing a scroll command.

However, to properly print the report you must define the page's attributes — number of lines per page, headers and footers, column names and order, page numbering, and other display elements.

The Report Screen



Function Keys

Command	Key	Description
Help	[F1]	Displays Help for the QBE Query screen.
End	[F2]	Displays the Home screen.
Query	[F4]	Displays the QBE or SQL Query screen, depending on which is currently specified on your Profile screen.
Proc	[F5]	Displays the Procedure screen. If there is a current procedure, the text is displayed; if there is no current procedure, the empty screen is displayed.
Form	[F6]	Displays the Form Main screen. If there is a current, valid query, the default form settings for the query are displayed. If there is no valid query, there are no settings on the Form Main screen.
Backward	[F7]	Displays the portion of the report above the current portion. The number of lines by which the window scrolls is determined by the Scroll setting at the bottom right corner of the Report screen.
Forward	[F8]	Displays the portion of the report below the current portion. The number of lines by which the window scrolls is determined by the Scroll setting at the bottom right corner of the Report screen.

Command	Key	Description
Backward	[F7]	Displays the portion of the report to the left of the current portion. The number of lines by which the window scrolls is determined by the Scroll setting at the bottom right corner of the Report screen.
Forward	[F8]	Displays the portion of the report to the left of the current portion. The number of lines by which the window scrolls is determined by the Scroll setting at the bottom right corner of the Report screen.

Report Forms

The report's format is determined by settings on screens in TRIMqmr that are called ***Forms***. To make changes to the report format, you make changes to Form screen settings.

Report forms help format your query results into readable and understandable reports. They include a number of data fields that allow you to add or change information to modify the appearance of a report.

Report forms let you:

- Produce tabular or cross-tabular reports.
- Make column heads more descriptive.
- Modify column widths.
- Adjust the spacing between columns.
- Enter dollar signs and decimals.
- Print totals and subtotals.
- Make page headings and footings more descriptive.

You can make most of your formatting decisions on the main report form (or screen), which is called FORM.MAIN. This screen allows you to set columns, headings and footings, final text, and control breaks.

More extensive formatting may require that you use one of the other forms including forms for detailed modifications to Columns, the Page, Final, breaks, and various Options.

You can move between each of these form screens with a single key press, or you can display forms by typing the names on the command line and pressing [Enter].

Formatting Reports

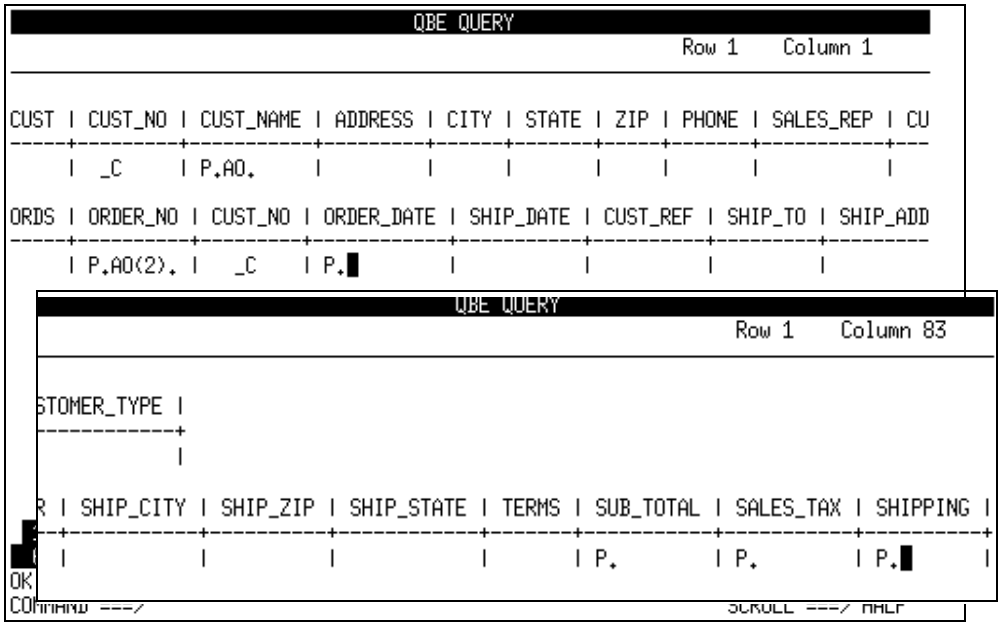
In the exercises in this chapter, you make simple and complex changes to the reports you generated in earlier chapters.

Simple changes include modifying column headings, indentation, report width, and basic format.

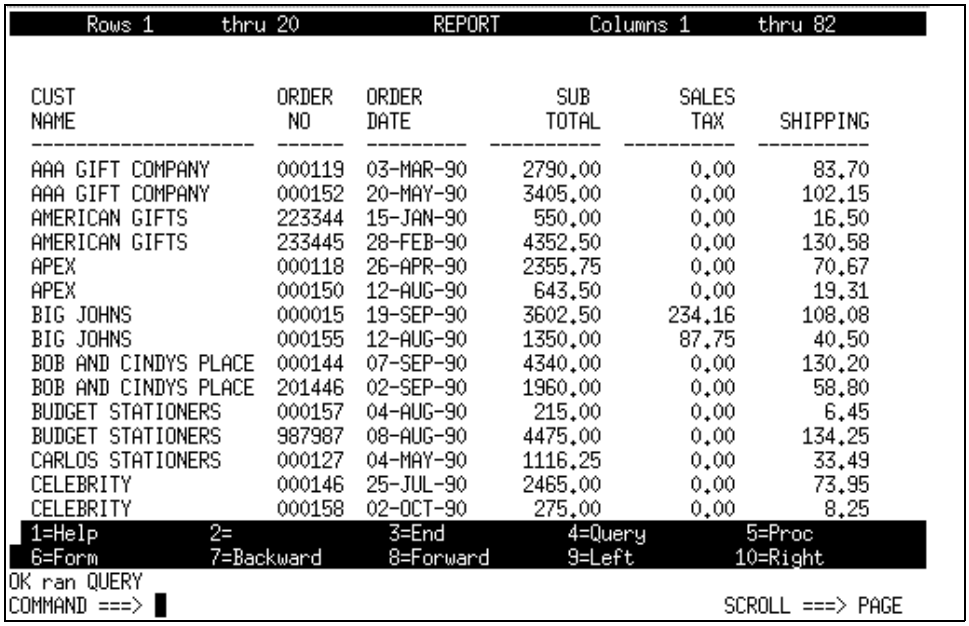
Creating a query

In the last set of exercises, you worked with a variety of queries to get information for reports. Before you begin reformatting the reports, create and save a multi-table query that uses the CUST and ORDS tables. Name it "CUSTORDS."

1. Ensure that you are using QBE as the language. If you don't remember how to check and set it, [F9] and use the help for additional support.
2. On the command line of the QBE Query screen ([F4] from the Home screen), type **`DRAW CUST`** and [Enter].
3. DRAW the ORDS table following the same procedure.
4. Identify the corresponding columns in each table by typing **`_C`** in the CUST_NO field of both tables.
5. Type **`P.AO.`** in the CUST_NAME field to display the results in alphabetical order by customer name.
6. Type **`P.AO(2)`**. in the ORDER_NO field to arrange the orders by number for each customer.
7. Type **`P.`** in the ORDER_DATE, SUB_TOTAL, SALES_TAX, and SHIPPING fields to display the data in those columns.



8. [F2] to see the results of the query that shows all customer orders.



```
     Rows 1      thru 20          REPORT       Columns 1      thru 82


 CUST                 ORDER   ORDER          SUB        SALES
 NAME                  NO     DATE          TOTAL       TAX      SHIPPING
 -------------------  ------  ---------  ----------  ----------  ----------
 AAA GIFT COMPANY     000119  03-MAR-90      2790.00        0.00       83.70
 AAA GIFT COMPANY     000152  20-MAY-90      3405.00        0.00      102.15
 AMERICAN GIFTS       223344  15-JAN-90       550.00        0.00       16.50
 AMERICAN GIFTS       233445  28-FEB-90      4352.50        0.00      130.58
 APEX                 000118  26-APR-90      2355.75        0.00       70.67
 APEX                 000150  12-AUG-90       643.50        0.00       19.31
 BIG JOHNS            000015  19-SEP-90      3602.50      234.16      108.08
 BIG JOHNS            000155  12-AUG-90      1350.00       87.75       40.50
 BOB AND CINDYS PLACE 000144  07-SEP-90      4340.00        0.00      130.20
 BOB AND CINDYS PLACE 201446  02-SEP-90      1960.00        0.00       58.80
 BUDGET STATIONERS    000157  04-AUG-90       215.00        0.00        6.45
 BUDGET STATIONERS    987987  08-AUG-90      4475.00        0.00      134.25
 CARLOS STATIONERS    000127  04-MAY-90      1116.25        0.00       33.49
 CELEBRITY            000146  25-JUL-90      2465.00        0.00       73.95
 CELEBRITY            000158  02-OCT-90       275.00        0.00        8.25
  1=Help        2=             3=End         4=Query       5=Proc
  6=Form        7=Backward     8=Forward     9=Left        10=Right
OK ran QUERY
COMMAND ===>                                              SCROLL ===> PAGE
```

Saving a query

1. On the command line, type **SAVE QUERY AS CUSTORDS** and [Enter] to save the query.
2. Type **RESET QUERY** and [Enter] to clear the query.
3. Test the saved query by running it: Type **RUN CUSTORDS** and [Enter].

The same report appears in the default format.

Getting to and Using FORM Screens

You can navigate to a TRIMqmr FORM screen by using the command line interface, or by pressing a [function key], when one is mapped.

Use the [Tab] or arrow keys to move between field on a FORM.

Typographical errors are interpreted as invalid entries, or errors, in FORM screens. You have to correct them before you can use a FORM for report display.

To test your work, be sure to [Enter] frequently. TRIMqmr rolls back all entries to the first error.

To display the FORM.MAIN, type **FORM** or **DISPLAY FORM** at the command line. Thereafter, typing **FORM** displays the last form that you saw. If a function key is mapped to FORM, the first time you press the key, TRIMqmr displays FORM.MAIN. Subsequent requests allow you to see the last form displayed.

To display a specific form, type the name on the command line, for example:

```
DISPLAY FORM.MAIN
```

to see the Main Form.

The report definitions for the CUSTORDS query should still be displayed on the Main Form. The top half of the screen shows details of the column formatting. Below that, you see the fields in which you specify page headers (headings) and footers (footings). You see the area "final text" for the end of report items, and the bottom of the screen provides areas for breaks, options, and other report details.

In the following exercise you will change column headings, and the report indent and width formatting.

Making simple changes

1. Type **`DISPLAY FORM`** if you are not already on that screen.
2. Change the appearance of column headings as shown below

```
                                FORM.MAIN

   Column Descriptions:                Report width is now: 77
 NUM  COLUMN HEADING                          USAGE    INDENT  WIDTH  EDIT
 ---  ------------------------------------  -------  ------  -----  -----
   1  Customer Name                                        2     20  C
   2  Order_Number                                         2      6  C
   3  Order_Date                                           2      9  TDOA-
   4  Merchandise_Total                                    2     10  L2
   5  Sales_Tax                                            2     10  L2
   6  Shipping_Charges                                     2     10  L2

PAGE:  Heading ==>
       Footing ==>
FINAL :Text    ==>
BREAK1:New page for BREAK? ==> NO
       Footing ==>
BREAK2:New page for BREAK? ==> NO
       Footing ==>
OPTIONS:Outline? ==> YES     Default BREAK text? ==> YES

 1=Help          2=Report        3=End           4=Form.Break1   5=Form.Options
 6=Query         7=Backward      8=Forward       9=Form.Page    10=Form.Columns
OK FORM performed
COMMAND ===>                                                  SCROLL ===> PAGE
```

3. [F2] to see the report with the new headings.

```
       Rows 1      thru 20            REPORT        Columns 1      thru 82


  Customer Name         Order   Order      Merchandis       Sales   Shipping
                        Number  Date         Total            Tax   Charges
  --------------------  ------  ---------  ----------  ----------  ----------
  AAA GIFT COMPANY      000119  03-MAR-90     2790.00        0.00       83.70
  AAA GIFT COMPANY      000152  20-MAY-90     3405.00        0.00      102.15
  AMERICAN GIFTS        223344  15-JAN-90      550.00        0.00       16.50
  AMERICAN GIFTS        233445  28-FEB-90     4352.50        0.00      130.58
  APEX                  000118  26-APR-90     2355.75        0.00       70.67
  APEX                  000150  12-AUG-90      643.50        0.00       19.31
  BIG JOHNS             000015  19-SEP-90     3602.50      234.16      108.08
  BIG JOHNS             000155  12-AUG-90     1350.00       87.75       40.50
  BOB AND CINDYS PLACE  000144  07-SEP-90     4340.00        0.00      130.20
  BOB AND CINDYS PLACE  201446  02-SEP-90     1960.00        0.00       58.80
  BUDGET STATIONERS     000157  04-AUG-90      215.00        0.00        6.45
  BUDGET STATIONERS     987987  08-AUG-90     4475.00        0.00      134.25
  CARLOS STATIONERS     000127  04-MAY-90     1116.25        0.00       33.49
  CELEBRITY             000146  25-JUL-90     2465.00        0.00       73.95
  CELEBRITY             000158  02-OCT-90      275.00        0.00        8.25
  1=Help          2=              3=End           4=Query         5=Proc
  6=Form          7=Backward      8=Forward       9=Left          10=Right
OK DISPLAY performed
COMMAND ===>                                                  SCROLL ===> PAGE
```

Notice that:

- "Customer Name" remained on a single line (because the words are separated by a space, rather than an underscore, which allows line breaking).
- The final "e" in "Merchandise" is missing.
- Less space is needed between Sales Tax and adjacent columns.
- The presentation of dollar can be improved.
- Press [F6] to return to the Main form.
- The numbers in the INDENT column refer to the number of spaces to the left of the column, from the left margin for the first column and from the column to the left for each subsequent column.

4. Use the arrow keys and/or the [Tab] to move the cursor to the INDENT fields and enter a **1** in each field for column headings 4, 5, and 6 (Merchadise_Total, Sales_Tax, and Shipping_Charges).

5. Move the cursor to the WIDTH column and for each column, make the following changes, increasing the first column width, and decreasing the next two:

   - Type **13** for Merchandise_Total to include the "e".
   - Type **13** for Sales_Tax.
   - Type **13** for Shipping_Charges.

6. Check your work. [F2] to see the results of the changes.
7. [F6] to return to the Main Form and continue formatting by adding a dollar sign and two decimal places to columns of numeric (dollar) value:
8. Move the cursor to the last column, EDIT, and type **D2** to specify a money format of dollars with two decimal places for each of Merchandise_Total, Sales_Tax, and Shipping_Charges.
9. [F2] to display the reformatted report.

☞ *Save the new report format*

1. [F6] (Form) to return to the Form main screen.
2. On the command line, type **SAVE FORM AS CUSO_FRM** and press [Enter].
3. Respond YES to the confirmation prompt (if displayed).

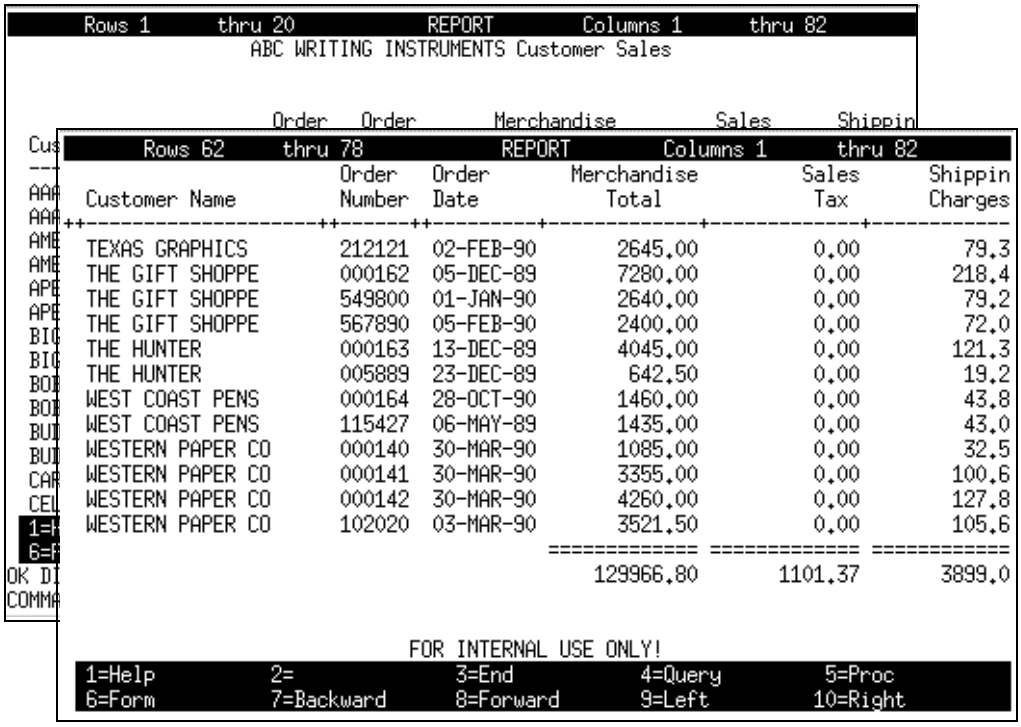
Making Complex Changes

Complex changes to formatting requires that you use Usage Codes, which give specific instructions on how columns of data are to be displayed. USAGE codes can apply to detail lines, report breaks, and summaries depending on the particular code and the way it is used.

See “Usage” on page 106 for a complete list of the available codes.

Edit the USAGE column and add headers/footers

1. On the Form main screen (FORM.MAIN), move the cursor to the USAGE column and enter **SUM** for Merchandise_Total, Sales_Tax, and Shipping_Charges.
2. In the Page Heading field type **ABC WRITING INSTRUMENTS Customer Sales**.
3. In the Page Footing field, type **FOR INTERNAL USE ONLY!**
4. [F2] to display the report with your final formatting additions. Type **BOT** and [Enter].



The Page Heading and Page Footing text appears at the top and bottom of each page of the report. Final Text is printed only at the end of the report.

Break Reports

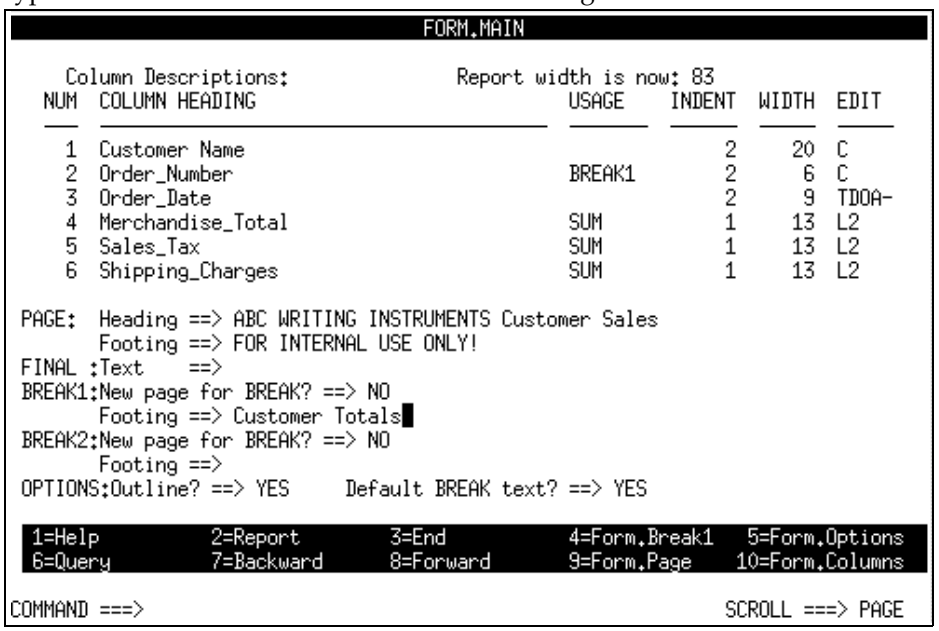
A break report generates intermediate and summary results. The results can be totals, summaries, statistics, messages, or blank lines to make the report more readable.

Creating a break report

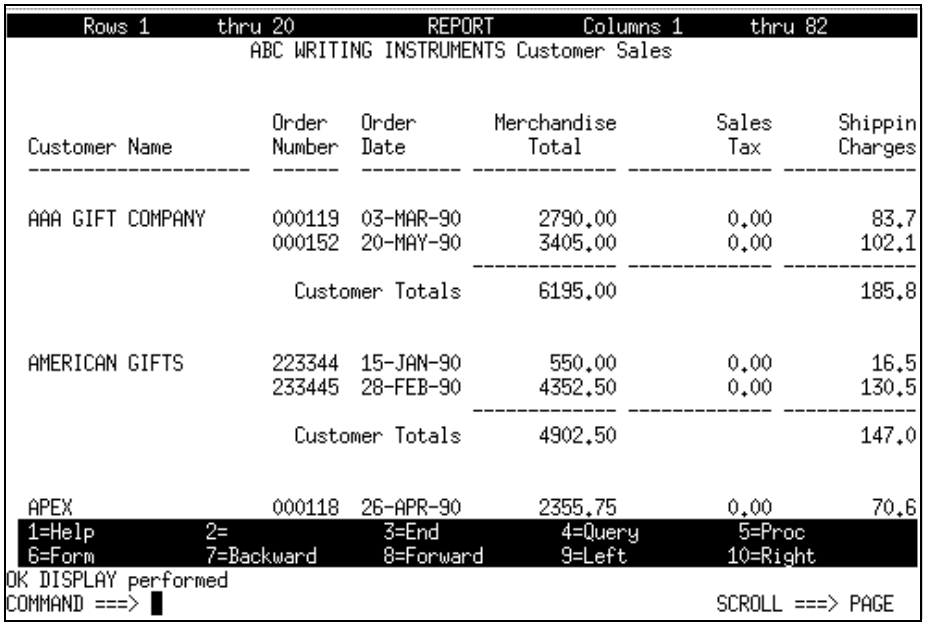
1. Return to the Form main screen (FORM.MAIN).
2. Move the cursor to the USAGE column for Customer Name and type **BREAK1** or **B1**.

   The BREAK1 usage command creates a total for each customer. Customer Name only appears for the first row of data that is displayed for each customer.

3. Type **`Customer Totals`** in the BREAK1 Footing field.



4. [F2] to see the new customer orders report.



5. Return to the Main Form screen ([F6]) and [F4].

Edit the Page Header

Using the Form Break1 screen, you can add page header and footer text. You can specify blank lines or page changes for both headers and footers. You can repeat column headings along with page header information for each break. The Form Break1 screen displays current defaults and Customer Totals as the values for the break. You can make additional report modifications on this screen.

1. Type **1** in the Blank Lines Before Heading field.
2. Move the cursor to the BREAK FOOTING TEXT field for line 1 and replace the text "Customer Totals" with the text **&1 TOTALS.**

   Instead of placing a constant on the break line, Form inserts the current value from the first column — the customer's name.

```
                                FORM.BREAK1

 New Page for Break?        ===> NO      Repeat Column Headings?   ===> NO
 Blank Lines Before Heading ===> 1       Blank Lines After Heading ===> 0
 LINE  ALIGN   BREAK HEADING TEXT
 ----  ------  ------------------------------------------------------------
    1  LEFT
    2  LEFT
    3  LEFT

 New Page for Footing?      ===> NO      Put Break Summary at Line ===> 1
 Blank Lines Before Footing ===> 0       Blank Lines After Footing ===> 1
 LINE  ALIGN   BREAK FOOTING TEXT
 ----  ------  ------------------------------------------------------------
    1  RIGHT   & TOTALS
    2  RIGHT
    3  RIGHT
    4  RIGHT
    5  RIGHT
```

   You can use the &*n* variable on any of the Form screens to display and print the current value of *n*, which is the number of a column as listed in the NUM column of the Main and Columns Forms.

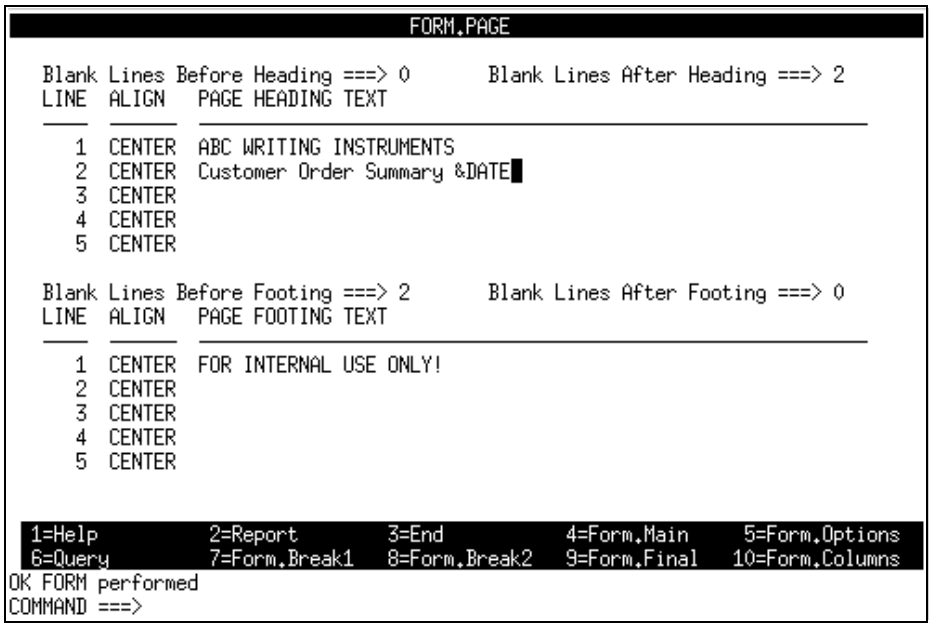
3. [F2] to see the result.

```
     Rows 1      thru 20          REPORT       Columns 1      thru 82
                  ABC WRITING INSTRUMENTS Customer Sales


                        Order   Order       Merchandise         Sales      Shippin
 Customer Name          Number  Date            Total           Tax       Charges
 --------------------   ------  --------- ------------- ------------- ------------

 AAA GIFT COMPANY       000119  03-MAR-90       2790.00          0.00         83.7
                        000152  20-MAY-90       3405.00          0.00        102.1
                                                ------------- ------------- ------------
                 AAA GIFT COMPANY TOTALS        6195.00                      185.8


 AMERICAN GIFTS         223344  15-JAN-90        550.00          0.00         16.5
                        233445  28-FEB-90       4352.50          0.00        130.5
                                                ------------- ------------- ------------
                   AMERICAN GIFTS TOTALS        4902.50                      147.0


 APEX                   000118  26-APR-90       2355.75          0.00         70.6
 1=Help          2=              3=End           4=Query         5=Proc
 6=Form          7=Backward      8=Forward       9=Left          10=Right
OK DISPLAY performed
COMMAND ===>                                                  SCROLL ===> PAGE
```

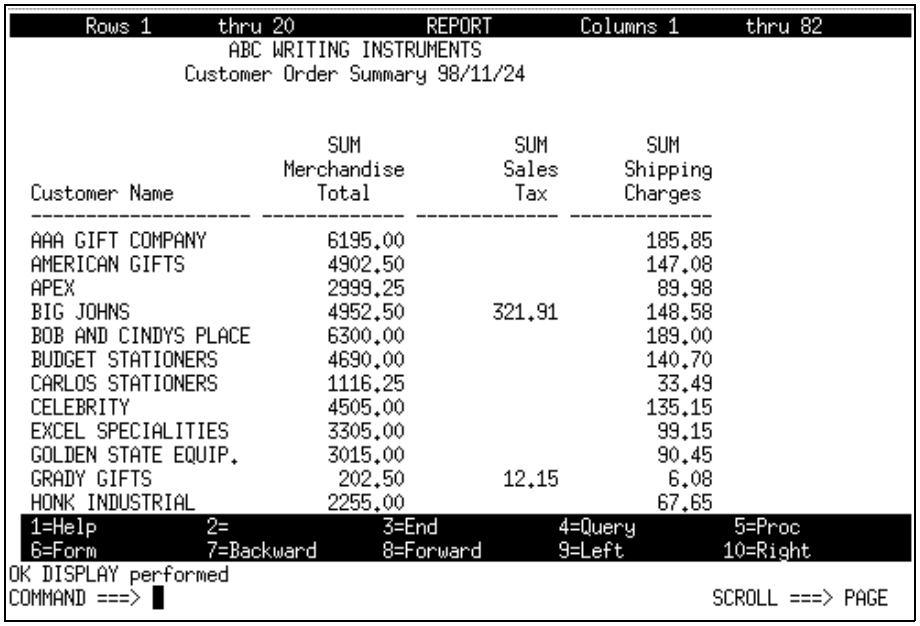
Create a summary break

You can also create a break that produces summaries only, without a data row. You will modify the page header to indicate the type of summary contained in this report. In addition, you use the DATE variable to display and print the current date.

1. [F6] > [F4] to return to the Main Form screen.
2. Move the cursor to the USAGE field for Customer Name and type **GROUP**.
3. [F9] to display the Page Form screen with the current page header and footer definitions.
4. For page heading text line 1, type **ABC WRITING INSTRUMENTS.**
5. For page heading text line 2, type **Customer Order Summary &DATE.**



6. [F2] to display the results.



Across Reports

Across reports provide a convenient way of presenting information in a matrix format. Spreadsheets are examples of typical across reports.

In an across report:

- Only one column can be given the USAGE of ACROSS.
- At least one other column must have the USAGE of GROUP.
- All columns must have a USAGE. (OMIT is the default and is inserted if you leave the USAGE blank.)

ACROSS identifies the column whose values become, along with the column name, the column headings; GROUP identifies each row with a different value from the specified column.

☞ *Creating a default report for across formatting*

In this exercise, you will create an across report that shows sales of each item by customer; totals are included for each item in the row at the bottom of the report and for each customer in the rightmost column.

The example joins three tables: CUST (customers), ORDITEM (order items), and INVENT (inventory).

1. [F4] to return to the QBE Query screen.
2. On the command line, type **`RESET QUERY`** and [Enter].
3. Type **`DRAW CUST`** and [Enter].
4. Type **`DRAW ORDITEM`** and [Enter].

5. Finally DRAW the INVENT table the same way you created the previous two.

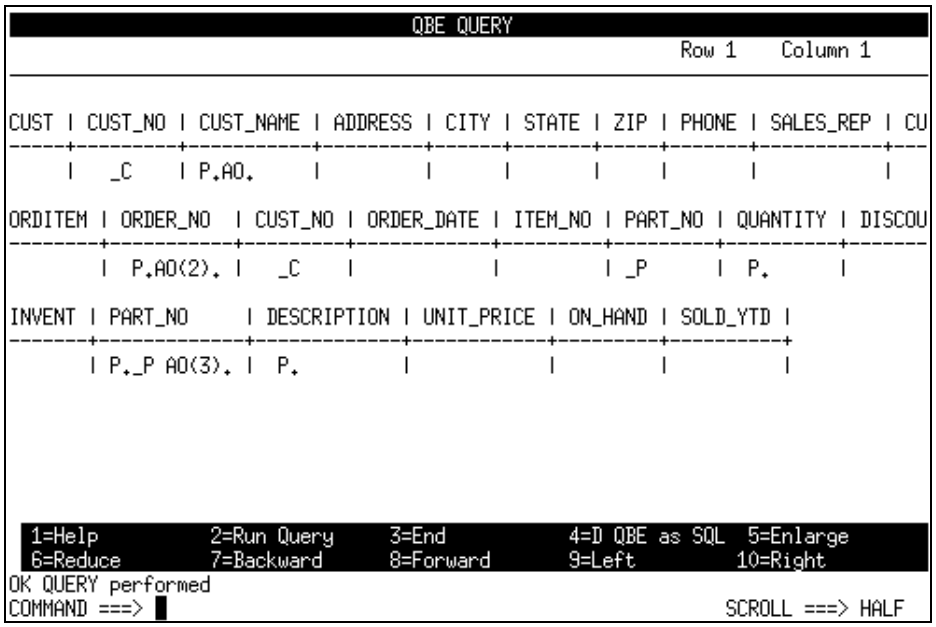
   All three table outlines are displayed on the QBE Query screen.

6. Now create the join relationships:
   - Type **_C** in the CUST_NO fields of CUST and ORDITEM tables
   - Type **_P** in the PART_NO fields of ORDITEM and INVENT

7. Indicate the columns you want to display by typing **P.** in:
   - CUST_NAME of CUST
   - ORDER_NO, QUANTITY, AND ITEM_TOTAL of ORDITEM
   - PART_NO and DESCRIPTION of INVENT

8. Finally, add the ordering commands:
   - Add **AO.** to the CUST_NAME field of CUST.
   - Type **AO(2).** in the ORDER_NO column field of ORDITEM.
   - Type **AO(3).** to the PART_NO column field of INVENT.

   Your QBE Query screen now displays the three table outlines with the entries as shown below. (The ORDITEM table column ITEM_TOTAL is not shown; it is to the right of the screen.)



9. [F2]. If your results differ from the illustration, check that you used the letter "O" rather than the number "0" in the QBE Query table outline.

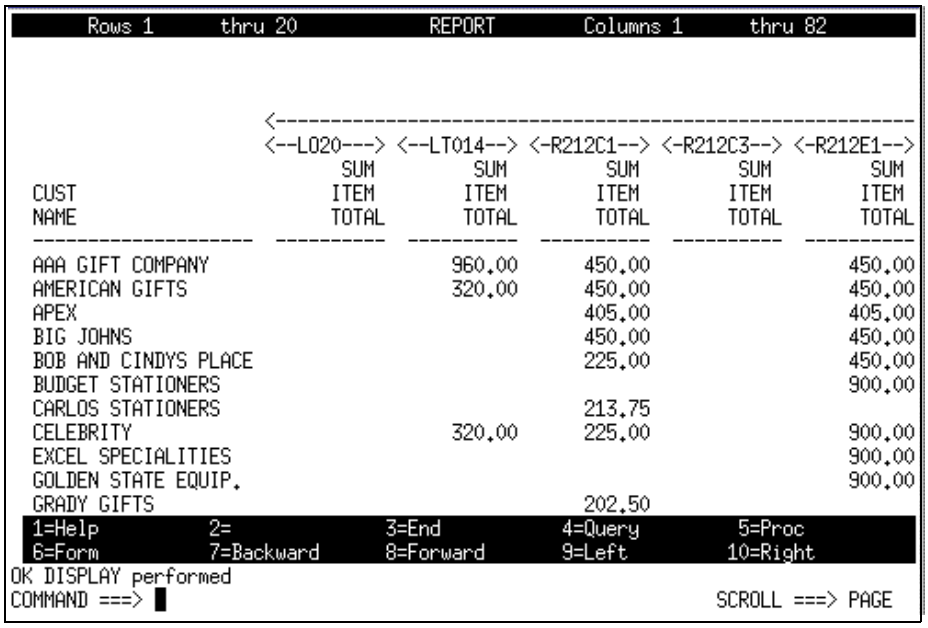
```
     Rows 1      thru 20          REPORT       Columns 1      thru 82


 CUST                ORDER                 ITEM   PART
 NAME                 NO     QUANTI        TOTAL   NO          DESCRIPTION
 ------------------  ------  ------  ----------  ----------  ----------------
 AAA GIFT COMPANY    000119     100      320.00  LT014       LUGGAGE TAG W/STRA
 AAA GIFT COMPANY    000119     100      450.00  R212C1      PEN GRAY GOLD
 AAA GIFT COMPANY    000119     100      450.00  R212E1      PEN BLK GLD DIAMON
 AAA GIFT COMPANY    000119      10       45.00  R212E2      PEN BLK GLD SQUARE
 AAA GIFT COMPANY    000119      50      275.00  R310G       PEN RHINESTONE GOL
 AAA GIFT COMPANY    000119     100      450.00  R510B3-A    PEN RED GOLD SWIRL
 AAA GIFT COMPANY    000119     200      370.00  TK1001      KEY CHAIN RECTANGL
 AAA GIFT COMPANY    000119     100      215.00  TK1005      KEY CHAIN OVAL
 AAA GIFT COMPANY    000119     100      215.00  TK1007      KEY CHAIN ROUND 2"
 AAA GIFT COMPANY    000152     100      320.00  LT014       LUGGAGE TAG W/STRA
 AAA GIFT COMPANY    000152     200      640.00  LT014       LUGGAGE TAG W/STRA
 AAA GIFT COMPANY    000152     100      450.00  R212E1      PEN BLK GLD DIAMON
 AAA GIFT COMPANY    000152      50      225.00  R212E2      PEN BLK GLD SQUARE
 AAA GIFT COMPANY    000152     100      450.00  R510B3-A    PEN RED GOLD SWIRL
 AAA GIFT COMPANY    000152     100      500.00  R510B4-A    PEN GOLD SILVER SW
  1=Help          2=              3=End           4=Query         5=Proc
  6=Form          7=Backward      8=Forward       9=Left         10=Right
OK ran QUERY
COMMAND ===>                                                  SCROLL ===> PAGE
```

☞ *Formatting across elements*

1. [F6] to return to the Form main screen to begin formatting the default report as an Across Report.
2. To create the across report, change the definitions as illustrated in the form:

```
                                  FORM.MAIN

   Column Descriptions:                  Report width is now: 84
 NUM  COLUMN HEADING                              USAGE    INDENT  WIDTH  EDIT
 ---  ------------------------------------------  -------  ------  -----  ----
   1  CUST_NAME                                   GROUP         2     20  C
   2  ORDER_NO                                    OMIT          2      6  C
   3  QUANTITY                                    OMIT          2      6  LO
   4  ITEM_TOTAL                                  SUM           2     10  L2
   5  PART_NO                                     ACROSS        2     10  C
   6  DESCRIPTION                                 OMIT          2     20  C

PAGE:  Heading ==> SALES MATRIX PART NUMBERS BY CUSTOMER
       Footing ==>
FINAL :Text    ==>
BREAK1:New page for BREAK? ==> NO
       Footing ==>
BREAK2:New page for BREAK? ==> NO
       Footing ==>
OPTIONS:Outline? ==> YES     Default BREAK text? ==> YES

 1=Help          2=Report        3=End           4=Form.Break1   5=Form.Options
 6=Query         7=Backward      8=Forward       9=Form.Page    10=Form.Columns
OK FORM performed
COMMAND ===>                                                  SCROLL ===> PAGE
```

3. [F2] to display the report.



4. [F10] to see the rightmost part of the report. Notice that the last column shows total sales for each customer.

5. [F8] to view the bottom right portion of the report. Here sales are totaled by Part Number and the overall sales total is given in the bottom right corner of the report (shown as a row of asterisks indicating that the value is too large for the column width).

6. [F6] to return to the Form main screen.

   Notice that the system has inserted OMIT in the USAGE column for the columns not displayed in the report.

☞ *Saving the formatted report*

You can save the report format for use later to format the report with current data, just as you saved the query.

On the command line, type **`SAVE FORM AS SALESMATRIX`** and [Enter].

You (or anyone with permission) can now regenerate this report with current data by using the FORM option with the RUN command:

```
RUN MATRIX (FORM = SALESMATRIX
```

To edit a saved query, form, or procedure, use the DISPLAY command:

- On the List screen, enter the DISPLAY command on the same line as the file name.
- On any command line, enter the DISPLAY command with the file name.

Summary

You should be familiar, after completing this chapter, with the basic formatting and design options of TRIMqmr. The following chapter lists, screen by screen, all the options you can adjust to create attractive and easy-to-understand reports.

Chapter 7 Forms Reference

Forms Quick Reference

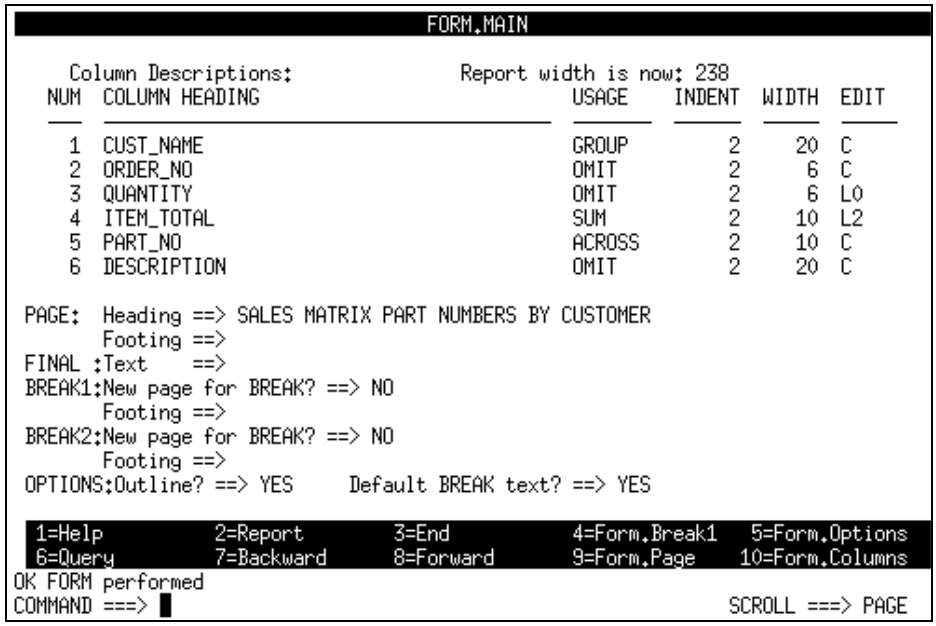
The following table shows the names of options you may want to customize and the form(s) on which that option is located.

Attribute	Field	Form(s) Located
Column heading (or title)	COLUMN HEADING	MAIN, COLUMNS
Column processing	USAGE	MAIN, COLUMNS
Spaces between columns	WIDTH	MAIN, COLUMNS
Width of columns	WIDTH	MAIN, COLUMNS
Column format (includes currency, decimal places, etc.	EDIT	MAIN, COLUMNS
Page heading text	PAGE HEADING TEXT	PAGE
Page footing text	PAGE FOOTING TEXT	PAGE
Final text	FINAL TEXT	FINAL
Break footing text	BREAK FOOTING TEXT	BREAK*n*
Placement of break text on page	PLACEMENT ON PAGE	BREAK*n*
Default break text	DEFAULT BREAK TEXT	OPTIONS

Form Main Screen

The Form Main (FORM.MAIN) screen includes the settings for most elements of the report format. The other Form screens allow for more detailed format settings for various aspects of the Report, as their names indicate:

- FORM.COLUMNS
- FORM.PAGE
- FORM.FINAL
- FORM.BREAK*n*
- FORM.OPTIONS



Fields

Report Width — See“***Report Width***” on page 105.

Num — “***Num***” on page 105.

Column Heading — “***Column Heading***” on page 106.

Usage — “***Usage***” on page 106.

Indent — “***Indent***” on page 108.

Edit — “***Edit***” on page 109.

Width — “***Width***” on page 108.

Page

The Page Heading and Footing fields allow you to enter a single line of text at the top and/or bottom of each page of the report.

The Page Heading and Page Footing fields also appear on the Form Page screen. You can enter multiple line headings and footings on the Form Page Screen.

Final Text

The Final Text field allows you to enter a line of text to appear at the end of the report. The Final Text field also appears on the Form Final screen. You can enter multiple lines of final text on the Form Final screen.

Break1 and Break2

The Break*n* fields allow you to enter text and a data break at two levels in the report, and to specify, if desired, a new page whenever a data break occurs.

The information in the Break*n* fields also appears on the Form Break*n* screens. You can enter multiple lines of text for each break as well as other changes on the Form Break*n* screens.

Options

The Options field on the Main screen allows you to customize:

- The value of the break column with each row of data, or only when the value changes.
- A row of asterisks (***) instead of break footing text. The number of asterisks corresponds to the Break level.

The information in the Options field also appears on the Form Options screen along with other settings that affect the appearance of your report.

Indicators

Indicator fields show values that cannot be set or changed on the screen.

Report Width

Shows the width of the total report in characters. You can't change this field; it changes automatically on all screens when you make any change to any field that affects the overall width of the report.

The value in this area is not always the total width of the report. In addition to the width of the columns, changing Usage, Indent, or Width for one or more columns can influence the actual report width.

When you use Usage code Across, the width is calculated when the report is sent to output.

(The width figure does not change until you [Enter] after making a change.)

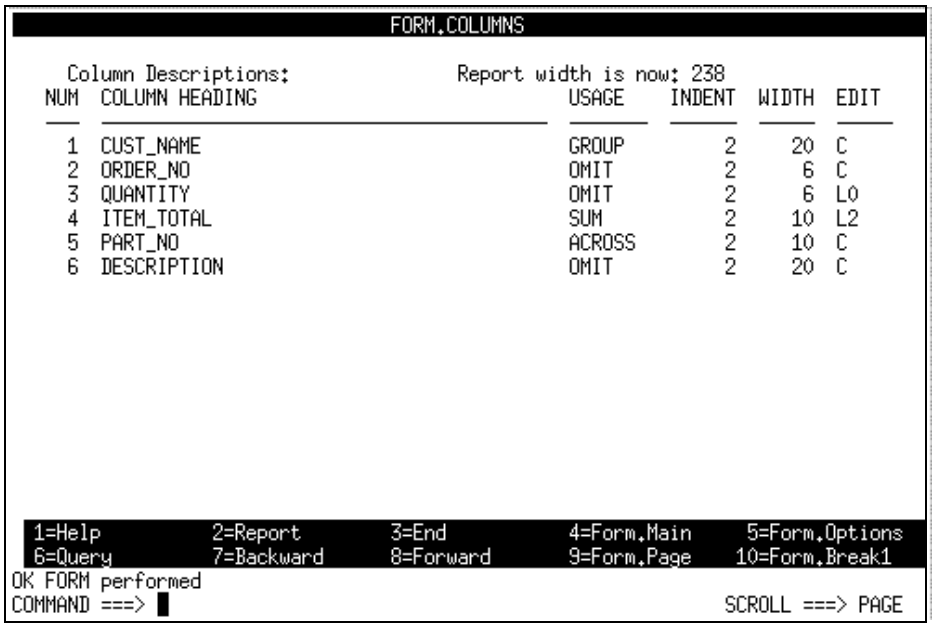
Num

Shows the column order as determined by the query. Usually columns appear on a report from left to right in order by their column numbers. You can have the columns automatically ordered when you use a Break or aggregating usage in the Form Options screen.

When you specify a column to be used in a data break or grouping, that column is automatically moved to the left of the other columns in the report. When you specify an aggregate function for a column, the column on which the calculation is made is moved to the right of the other columns in the report.

Form Columns Screen

Customize the following attributes of a report on this screen:



Column Heading

The Column Heading column is found on the Form Main screen and the Form Column screen. Column Heading displays:

- Name of each column as it is found in the database.
- Label assigned to the column.
- Heading created for columns of numeric values.

To change a column heading, type the new name over the current heading. The name can contain up to 40 characters including special characters such as blanks.

All upper case or upper and lower case text can be used in the report depending on the Case setting on the Profile screen. If Case is set to Mixed or String, upper and lower case can be used; otherwise, the entire report is printed in all uppercase.

To indicate a break in the column heading, insert an underscore (_).

Each underscore indicates a new line; consecutive underscores (_ _) indicate blank lines in the heading.

In general, the position of data within a column is determined by the column data type: Character data is justified left; numeric data is justified right.

Usage

The Usage field is found on the Form Main screen and the Form Columns screen. The code word placed in the Usage field specifies how the data in the specified column should be processed for the report. Usage codes can apply to rows of data, report breaks, and summaries depending on the code and the way it is used.

The following Usage codes, except OMIT, summarize the data in a column. They can be used with control breaks to display the summary data within a report as a subtotal. A null value is not included in the calculation.

Usage	Description
ACross	Each row in the column is used as a column heading; related data is displayed in matrix format.
AVerage	For numeric columns, calculates the average (mean) value.
BREAKn	Causes a summary break when a column value changes.
COunt	Counts and reports the number of different values in a column.
FIrst	Reports the first value in a column.
GRoup	Indicates the column that controls grouping for summary data.
Last	Reports the last value in a column.
MAximum	Reports the largest value in a column.
MInimum	Reports the smallest value in a column.
Omit	Does not include the column in the report.
Stdev	Reports the standard deviation of the values in a column.
SUm	Reports the total of the values in a column.

A row of asterisks (***) in the report indicates that the calculated value was greater than can be displayed in the column. (Change Width on the Form Main or Form Columns screen to accommodate the calculation.)

A row of greater-than symbols (>>>) in the report indicates an arithmetic overflow. (Change the data type for the column in the query.)

The following Usage codes denote an aggregate usage that replaces the value in a column of numeric data with a calculation. The final result of the usage is displayed at the end of the report as a subtotal.

Usage	Description
CPct	Cumulative percentage of the values in the column.
CSum	Cumulative total of the values in the column.
Pct	Percentage of the total represented by each value.
TCpct	Total cumulative percentage for each value in the column.
TPct	Percentage of the column total for each value.

The following Usage codes indicate control breaks in a column.

Usage	Description
BREAK1	Level-1 control break for the column. Column is placed on the left side of the report; any change in column value results in a control break--break text is displayed; sub-totals for aggregate columns are displayed.
BREAK2	Level-2 control break for the column. The column is placed to the right of the level-1 control break column.
BREAK3	
BREAK4	
BREAK5	Specifies successive levels of control breaks for columns.
BREAK6	
BREAK*n*	Control breaks specified are not included in the report.
BREAK*n*X	Specifies that the control column label should print for each line.

The Usage codes that follow have special implications for report data.

Usage	Description
GROUP	Displays one line of summary data for each group of values in a column; summary data can be the same for each value in the group or aggregate — a set of column summaries.
ACROSS	Displays a set of summary data for group of values in the column. The Group column or columns must be ordered before the Across column; the Across column must be ordered. At least one other column must have a Usage code of Group.

Indent

The Indent column is found on the Form Main screen and the Form Columns screen.

The Indent column specifies the number of blank spaces to be inserted to the left of a column to separate the column data from the column data to the left or the left margin.

Width

The Width column is found on the Form Main screen and the Form Columns screen. It determines the number of character positions used to display the column data in the report.

A width for numeric data must include space for the following characters as well as the digits:

Character	Used with Edit Code
Possible minus	E, D, I, K, L, P

Character	Used with Edit Code
Decimal point	E, D, I, J, K, L, P
Separators for thousands	D, K, P
Currency symbol	D
Percent sign	P

If the data is wider than the specified Width, the value appears as a row of asterisks (*****). You can place a code "CW" in the Edit column (see below) to continue data on consecutive lines within the column.

Edit

The Edit column is found on the Form Main screen and the Form Column screen. It specifies how character, numeric, and date data is displayed.

Character Data — the codes available for specifying character datatypes.

Code	Description
C	No change to the value display.
CW	No change to the value display. If the value is wider than indicated in the Width column, the text continues to the next line.
CT	No change to the value display. If the value is wider than indicated in the Width column, the text continues to the next line according to the text in the column. Form fits as much data as possible on a line, then continues to the next line as a blank space.
CD*x*	

No change to the value display. If the value is wider than indicated in the Width column, the text continues to the next line according to the delimiter *x* (***x*** may be any character including a blank).

Numeric Data — available codes for specifying numeric data, including date and time.

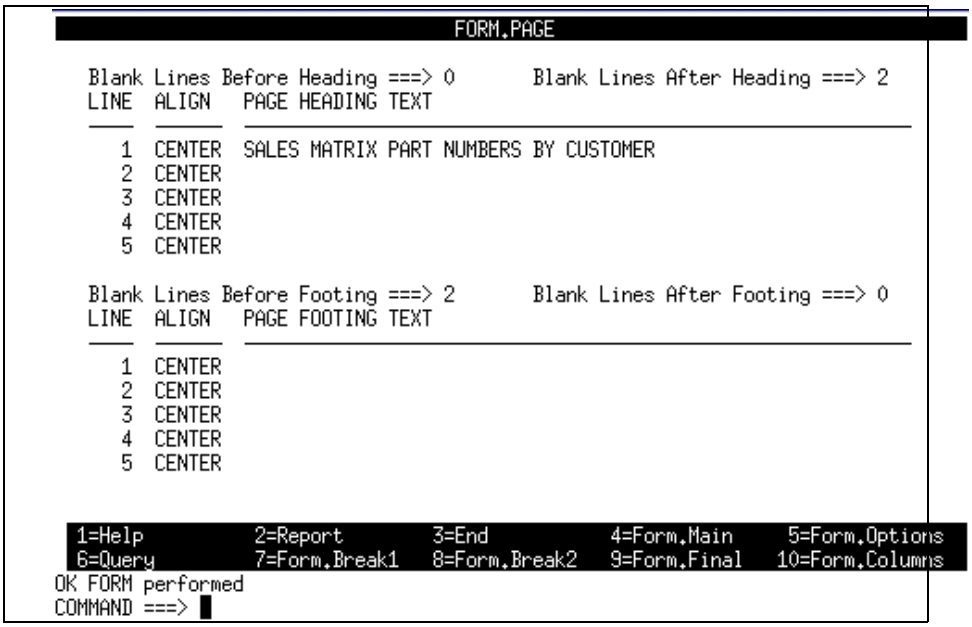
Code	Display of	Description
E	-9.87654321E+06	Exponent format (scientific notation).
D2	-$9,876,543.21	Dollar sign (or national currency symbol) with comma separators.
K2	-9,876,543.21	Comma separated, leading zeros suppressed.
L2	9876543.21	Leading zeros suppressed.
P2	-9,876,543.21%	Comma separated, leading zeros suppressed, followed by percent sign.

Date Data — codes available and possible formats for date data. (*x* indicates the symbol separating the units of time.)

Code	Format	Description	Example	Display of Feb 6, 1988; 2:30pm
TDYx	YYYYxMMxDD	Year first		
TDYAx	YYxMMxDD	Year first, year abbreviated	TDYA/	88/02/06
TDMx	MMxDDxYYYY	Month first		
TDMAx	MMxDDxYY	Month first, year abbreviated	DTMA	02 06 88
TDDx	DDxMMxYYYY	Day first	TDD.	06.02.1988
TDDAx	DDxMMxYY	Day first, year abbreviated		
TDOx	DDxMMMxYYYY	Oracle format	TDO	06 FEBRUARY 1988
TDOAx	DDxMMMxYY	Oracle format, year abbreviated	TDOA-	06-FEB-88
TTS	HHxMMxSS	24-hour clock	TTS:	14:30:00
TTCx	HHxMMxSS	12-hour clock	TTC.	02.30.00
TTAx	HHxMM	12-hour clock		
TTAN	HHMM	12-hour clock, no delimeter	TTAN	1430
TTUx	HHxMM AM/PM	12-hour clock, U.S. format		
TSI	YYYY-MM-DD-HH.MM.SS	Timestamp	TSI	1988-02-06-14:30:00

Form Page Screen

The Form Page screen allows you to specify details for the placement and content of the page headings and footings.



Blank Lines Before Heading

This field specifies the number of blank lines between the top of the page and the first line of the page heading.

The default is for no blank lines; you can replace the default with a number from 0 to 999.

Blank Lines After Heading

This field specifies the number of blank lines between the last line of the page header and the body of the report.

The default is for two blank lines; you can replace the default with a number from 0 to 999.

Line (Heading)

The Line field specifies the position number of the heading line on which the text is to appear. The default numbers are 1 through 5; valid line numbers are 1 to 999 and blanks (although the maximum number of lines that can be displayed is 5).

A blank in the Line field causes the text entered for that line to be ignored; use a blank to write comments in the Page Heading Text field or to temporarily eliminate the line from the report. Since the number identifies the location of the text line, you can enter the lines in any order.

You can use the same line number for more than one line of text. Use the Align field to position the text to avoid having the text overlap.

For example, on line 2, you can reposition Customer Order Summary to the left side of the report by placing LEFT in the Align field and deleting &DATE from the Page

Heading Text field. Renumber line 3 as line 2, change the Align field to RIGHT, and type &DATE in the Page Heading Text field. The two heading lines display/print similar to:

```
            ABC WRITING INSTRUMENTS
  Customer Order Summary          August 2, 1992
```

Align (Heading)

The Align field indicates the horizontal placement of each line of page heading text.

Align	Horizontal Placement of Page Heading Text
LEFT	Justified to the left side of the report.
RIGHT	Justified to the right side of the report.
CENTER	Positioned in the horizontal center of the report.
n	Begins text in the *n* character position from the left margin; *n* can be a number from 1 to 999,999.
APPEND	Positions the line at the end of a line with the same line number.

Page Heading Text

The Page Heading Text field contains a line of page heading text to be displayed at the top of the report. You can use up to five lines of 55 characters; if a longer line is desired, use the Append option of the Align field to concatenate multiple text lines.

Page heading text (and other Form screen text) can include one or more of the four predefined variables:

Variable	Description
&*n*	*n* represents the value in column *n* selected by the query (not necessarily the *n*th column in the report).
&DATE	Current date (year/month/day).
&TIME	Current time (hour:minute).
&PAGE	Current page number of the report.

Blank Lines Before Footing

Specifies the number of blank lines to insert between the bottom line of the report and the first line of the page footing. The two blank line-default can be replaced with a number from 0 to 999.

Blank Lines After Footing

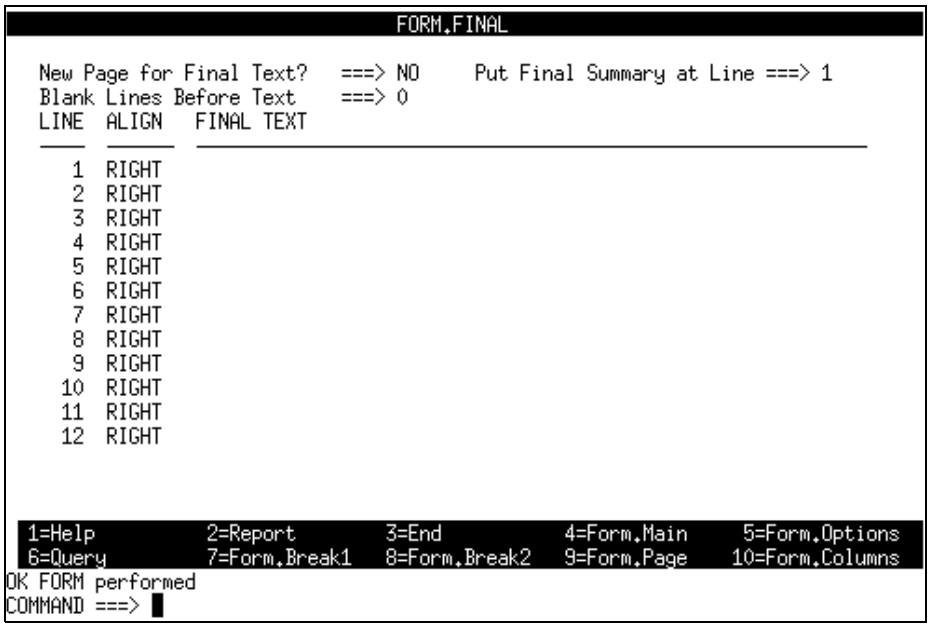
This field specifies the number of blank lines to be inserted between the last line of the page footing and the bottom of the page. The default is for no blank lines; you can replace the default with a number from 0 to 999.

Page Footing Text

Describes the page footing text displayed at the bottom of each page. The four types of variables possible in footing text are: &n, &DATE, &TIME, and &PAGE.

Form Final Screen

Use the Form Final screen to specify details of the placement and content of the final text of the report.



New Page for Final Text

This field specifies whether the final text is to appear on a page separate from the body of the report.

Blank Lines Before Text

This field specifies the number of blank lines to be inserted between the last line of the report and the first line of the final text.

The default is for no blank lines; you can replace the default with a number from 0 to 999, or the word BOTTOM or B.

Put Final Summary at Line

This field specifies whether the final summary of the report is to be generated and, if so, the placement in relation to the final text.

The default for this field is 1; you can replace the default with a number from 0 to 999, or the word NONE or N. If you use NONE, the final summary is not included with the report.

Line

See“***Line***” on page 113.

Align

See “***Align***” on page 113.

Final Text

The Text field contains the line of text to be displayed at the end of the report. You can use as many as 12 lines of 55 characters; if you want a longer line, use the APPEND option in the Align field to concatenate multiple text lines.

Form Final text can include six types of variables:

Variable	Value
&*n*	*n* is the first value in column *n* selected by the query (not necessarily the *n*th column shown in the report).
&DATE	Current date (year/month/day)
&TIME	Current time (hour:minute)
&PAGE	Current page number of the report.
&*an*	Aggregate function on the *n*th column selected by the query.

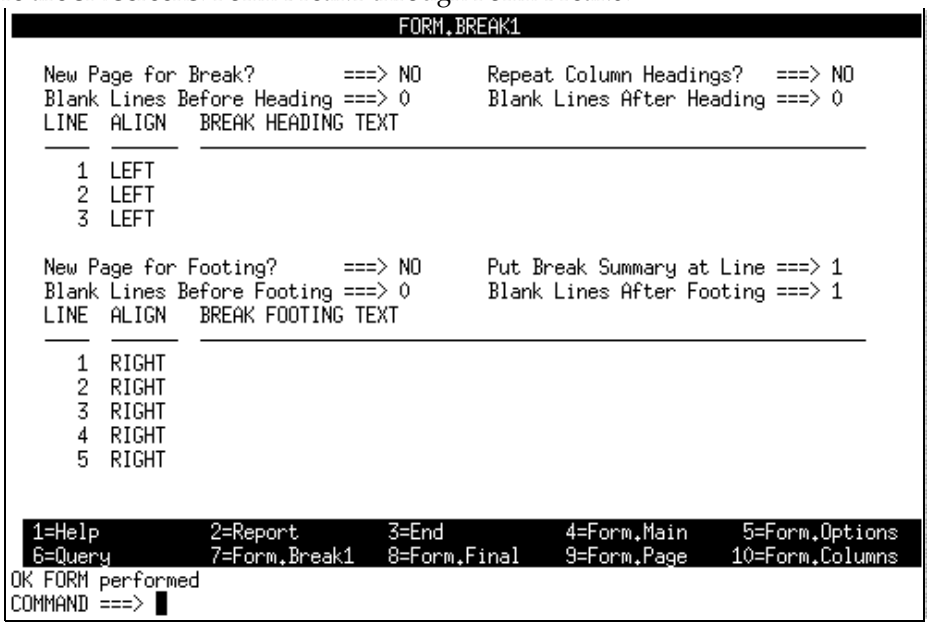
In the &*an* variable, *a* can be equal to any of the aggregate functions in the table below.

Aggregate Function	Value
AVG	Average value of column data (numeric).
CPCT	Cumulative percentage of column data (numeric).
CSUM	Cumulative total of the column data (numeric).
FIRST	First value in the column (numeric or character).
LAST	Last value in the column (numeric or character).
MAX	Maximum value in the column (numeric or character).
MIN	Minimum value in the column (numeric or character).
PCT	Percentage of the total represented by each value (numeric).
STDEV	Standard deviation of the column values (numeric).
SUM	Total of the column values (numeric).
TCPCT	Total cumulative percentage for each value in the column.
TPCT	Percentage of the column total for each value (numeric).

Form Break*n* Screen

Use the Form Break*n* screen to specify report breaks on column data. Details of the placement and content of each of the six possible levels of control breaks are specified as the usage of a column; the usage is indicated on the Form Main or Form Columns screen.

There are six screens: Form Break1 through Form Break6.



New Page for Break

This field specifies whether a new page is to be generated each time the value in a control column changes.

Repeat Column Heading

This field indicates whether the column headings are reprinted at the beginning of each break. When a data break occurs at the top of a page, column headings are repeated. If you select YES and the break heading text begins a new page, column headings are displayed only once.

Blank Lines Before Heading

See "***Blank Lines Before Heading***" on page 115.

Blank Lines After Heading

See "***Blank Lines Before Heading***" on page 111.

Line (Heading)

See "***Line***" on page 113.

Align (Heading)

See "***Align***" on page 113.

Breakn Heading Text

The Break*n* Heading Text field specifies the break heading text. The text can appear at each control break in the report. You can use up to three lines of 55 characters each. If a longer line is desired, use the Append option of the Align field to concatenate multiple text lines.

Break heading text can contain one type of variable, &*n*. *n* represents the first value in column *n* selected by the query (not necessarily the *n*th column shown in the report). The numbers can be found in the Num column of the Form Main screen or Form Columns screen.

New Page for Footing?

This field specifies whether a new page is be generated for break footing text.

Blank Lines Before Footing (Put Break Summary at Line)

See page "***Blank Lines Before Footing***" on page 112.

Blank Lines After Footing

See page "***Blank Lines After Footing***" on page 112.

Line (Footing)

The Line field contains the position number of the line within the page footing area.

The default numbers are 1 through 5; valid line numbers are 1 to 999 and blanks (although the maximum number of lines that can be displayed or printed is 5). See "***Line***" on page 113.

Align (Footing)

The Align field indicates the horizontal position of the break footing text: left, right, or center.

Breakn Footing Text

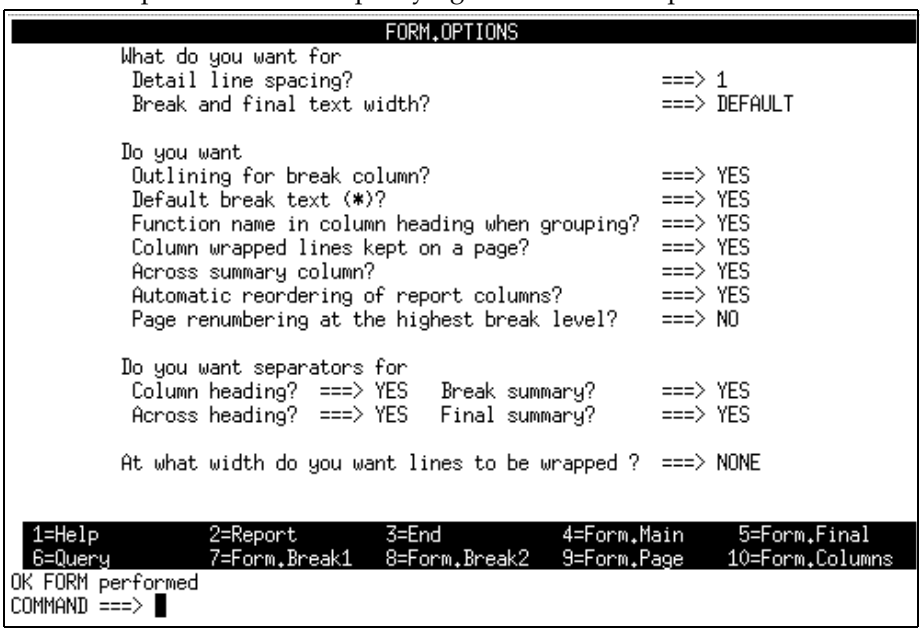
The Text field specifies the break heading or footing text. The text can appear at each control break in the report. You can use up to five lines of 55 characters each; if a longer line is desired, use the APPEND option of the Align field to concatenate multiple text lines.

Break footing text can contain three types of variable:

Variable	Description
&*n*	*n* represents the first value in column *n* selected by the query (not necessarily the *n*th column shown in the report). These numbers can be found in the Num column of the Form Main screen and the Form Columns screen.
&COUNT	Displays the total number of data rows in the report.
&*an*	Displays an aggregate function, a, of the *n*th column selected by the query.

Form Options Screen

Use the Form Options screen for specifying miscellaneous report details.



Detail Line Spacing

This field specifies the amount of spacing between rows of data.

The default is 1 (single spacing); valid numbers are 1 through 4. Placing a 2 in the field indicates double spacing (one blank line between detail lines); 3 indicates two blank lines; 4 indicates three blank lines.

Break and Final Text Width

The width of break and final text in the report. Valid entries are integers between 0 and 999,999, DEFAULT and COLUMNS.

Entry	Effect
0	No break of final text.
DEFAULT	Specifies the use of the full column width in formatting break heading text. Break footing and final text appears between the left margin and either the beginning of the summary column or the right margin. Text wider than the area provided is truncated. If the first column contains summary data, text is suppressed.
COLUMNS	Specifies that the break and final text extend to the width of the columns, which is given on Form Main and Form Column screens. Values from 1 to 999,999 characters.

Outlining for Break Columns

This field displays the break column value only when it changes (YES) or on every detail line of the report (NO).

Default Break Text

When no break footing text is specified for a data break, this field generates default break footing text to indicate the generated break lines. If you select YES, one or more asterisks is displayed, one for each break level in reverse order. For example, if there are two break levels, the first is indicated by two asterisks (**) the second by one asterisk (*).

Function Name in Column Heading When Grouping

This field determines whether a default heading is to be added to a column heading in which the data is grouped; for example, if you used the aggregate function AVG to find the average price of grouped items in the sample database INVENT table, placing a YES in this field adds AVERAGE to the UNIT_PRICE field:

```
AVERAGE
UNIT
PRICE
```

Column Wrapped Lines Kept on a Page

When the column reaches the end of the page, this field determines whether data can be split between two pages.

Across Summary Column

This field determines whether to display the summary column that is automatically generated for across reports. If YES is specified, the summary column is displayed in the rightmost column of the report.

If at least one column in your report has a Usage code of PCT (percentage), CPCT (cumulative percentage), or CSUM (cumulative total), and you have specified a final summary, the across summary column displays two data lines for each summary. The first line is the summary value for the specified group; the second line contains the summary value for those columns designated by the Usage codes above.

Automatic Reordering of Report Columns

This field indicates whether columns in the report are to be reordered when you specify a Usage of Break*n*, Group, or an aggregate function.

The columns are reordered:

Usage	Reordering
Break*n*	Break columns are moved to the left of all other columns.
Group	Group columns are moved to the left of all other columns except any Break columns.
Aggregate Functions	Aggregate function columns are moved to the far right of all other columns.

Placing a NO in this field keeps the columns in the order in which they appear on the Form Main and Form Columns screens.

For an across report, this field is ignored.

Page Numbering at the Highest Break Level

This field indicates whether the report is to begin new page numbering (beginning with 1) whenever the value changes in the control column with the highest break level (highest number of Break*n*).

Separators for Column Heading

This field indicates whether the broken horizontal lines separating column headings from rows of data are to be displayed.

Separators for Across Heading

This field indicates whether the broken horizontal lines and arrows (that define columns) are to be displayed.

Separators for Break Summary

This field indicates whether the broken double lines (equal signs) that separate the report from the final summary are to be displayed.

Default is YES.

Separators for Final Summary

This field indicates whether the broken double lines (equal signs) that separate the body from the final summary are to be displayed.

Default is YES.

At What Width Do You Want Lines to Be Wrapped?

This field indicates whether detail lines in the report are to be automatically wrapped and, if so, the width at which a new line is to be started.

Valid entries for this field are numbers 1 to 999 and NONE. The default, NONE, specifies that no line wrapping is allowed; if the report does not fit within the print width, the right side of the report is eliminated.

If you specify line wrapping at a width in which the report does not fit, the right side of the report is eliminated. If the report does fit within the line-wrapping width, each line of data is continued on one or more consecutive lines in the report.

Chapter 8 Procedures

This chapter describes procedures (PROCs) which are sets of commands executed together in a single RUN that produce only the final results.

Procedures allow you to combine TRIMqmr commands to retrieve and format data, perform data calculations and functions, format the data, and print a report. Within a single procedure you can include variables that prompt for values, as well as require confirmation screens before changes are made to a database. They are useful, for example, for generating periodic reports with current data such as sales or inventory reports.

Queries, procedures, and forms used within a procedure must be named and saved; current objects (those displayed on screen) cannot be included until they have been saved. You can include objects created by other users. However, each object name must be prefixed with the owner's ID.

Procedure Screen

This screen contains the current TRIMqmr procedure (if any), which can be used to enter a new procedure, modify a procedure in an existing file, or display a procedure.

```
                                    PROC
  Top line # 1                                          # of lines 0

run test_qbe (form = test_form

 1=Help          2=Run Proc      3=End           4=Delete        5=Duplicate
 6=Query         7=Backward      8=Forward       9=Insert       10=Report
OK PROC performed
COMMAND ===>                                                SCROLL ===> HALF
```

Continuation Lines

Commands in a procedure can be continued on consecutive lines by using the continuation character "+'' in the first column of subsequent lines. For example:

```
RUN query_name
   + (FORM = form_name)
```

If there is an apostrophe in the command line, the text that appears on the continued line is concatenated with the text in the previous line. For example:

```
RUN query_name (COMMENT='Sales total includes +
discount')
```

In the example, the words following the first apostrophe are concatenated: Sales total includes discount. To add a space when concatenating, include a space following the continuation symbol.

Command key words, parameters, and substitution variables cannot span lines. Comment and blank lines are allowed between continued lines.

Comment Lines

A procedure can include comment lines to describe individual command functions, explain procedure sections, overview of the procedure, etc.

A procedure can include any number of comment lines. A comment line can be placed on the same line as a command or on a line by itself. In either case, the comment must be preceded by a double hyphen (--); if a comment line wraps to the next line, insert a double hyphen to continue the comment text.

Variables

A procedure can include variables for which the user must provide values before the procedure can be executed. The list of variables is displayed on the RUN Command prompt and the values are entered on the screen.

Confirmation

A procedure can also require confirmation as allowed by commands within the procedure. For example, if you want to save the results of a query into an existing database table, a confirmation prompt can be requested with the SAVE command before the new data overwrites the old.

When you create a new or edit an existing procedure, your changes are not saved until you issue a SAVE PROCEDURE command. Otherwise, any changes are deleted when you end the current TRIMqmr session.

Creating a Procedure

You create a procedure by typing the commands on a blank Procedure screen's workspace and saving it with a unique name. You can select an existing procedure, and edit the commands and save it under a new name without altering the original procedure, as well.

A procedure can:

- Contain only TRIMqmr commands (except END), blank lines, and comment lines.
- Run other procedures.
- Include variables (&variables), and procedures and queries that contain variables (&&variables).
- Continue commands on consecutive lines.

Variables

Use variables in a procedure for values that are to be supplied at the time the procedure is run. For example, you can use the same query to generate the same report for different time periods. Place a variable in the query condition clause:

```
SELECT part_no, quantity, order_date FROM orditem
      WHERE order_date > &begindate
      AND order_date < &enddate
```

When the procedure containing the example query is run, you are prompted for a value for each date you've identified.

Enter a value for each of the variables and [Enter] to substitute them for the variables in the procedure.

Alternative, you can define the variable on the command line when you issue the RUN command.

Define in Procedure

You can also enter the variable values in the procedure itself. When the procedure is run, the values are automatically substituted. For example,

```
RUN itemsales (&&begindate = '01-JAN-92
   &&enddate = '12-DEC-92')
```

When you define a variable's value in a query run by a procedure, prefix the variable name with a ***double*** ampersand (&&) to identify it as an existing variable to which associated values are passed.

You can include other variables, such as a form within a RUN:

```
RUN ordersmon (FORM=&name &&date = '>02/28/92')
```

In this example, the variable &name prompts for a form name but the value "02/28/92" is passed to &date.

Saving and Sharing a Procedure

The SAVE PROC AS (proc_name) command saves the procedure for running at a later time.

Sharing procedures is just like sharing queries. When you save your procedure, be sure SHARE=YES is in your profile, or append (SHARE=YES to your SAVE statement. Setting SHARE=YES means that every TRIMqmr user can use your procedure. If you want to share a procedure, you must make sure to set SHARE=YES on each of the individual queries, forms, and procedures within the procedure, as well.

When creating a procedure to be shared, you must place your userid before your query and form names within the procedure, as in the following example. Users with a different userid can then invoke the procedure as `userid.procname`.

Executing a Procedure

RUN (procname) runs all the commands in the procedure as if they were executed from the command line, that is in "batch-mode." Contents of FORM, DATA, and QUERY are altered according to the procedure instructions. TRIMqmr only outputs the final results, not the intermediate steps.

Errors in any of the procedure's commands terminate the procedure and the procedure screen displays. The command containing the errors appends at the top of the screen; the error message is shown on the message line.

Batch-Mode Considerations

When you design a procedure to run in batch mode, remember that no screens appear. To ensure proper execution:

1. SET PROFILE (CONFIRM=NO

   Raising a prompt condition in batch mode is regarded as an error and will halt your procedure. To ensure no user confirmation is requested, set the first line of your procedure to turn the option off.

2. EXIT

   The last line of your procedure should be EXIT, so TRIMqmr ends the session cleanly.

You can also run queries and add forms in batch mode.

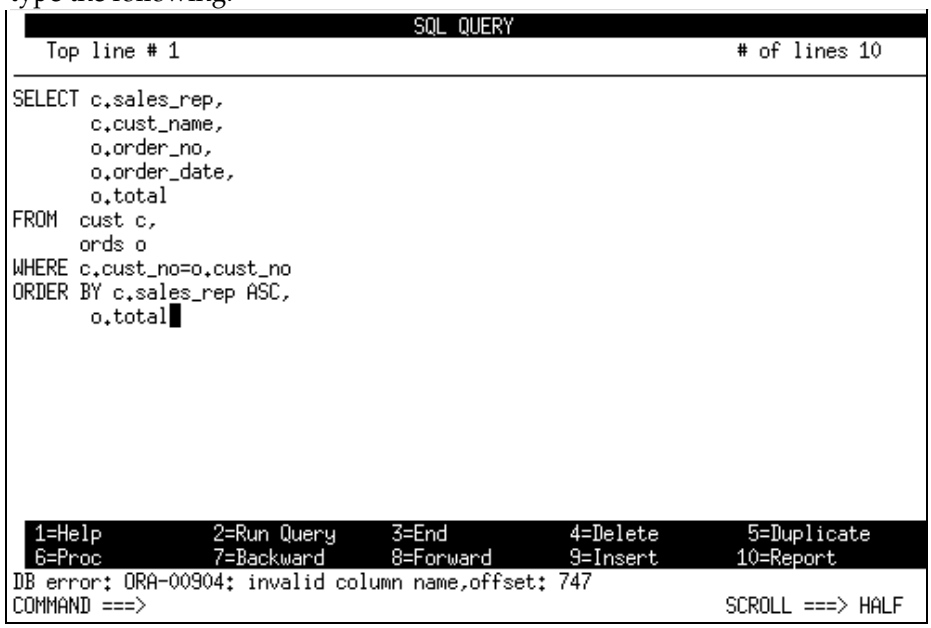
Short Exercise

Creating a query

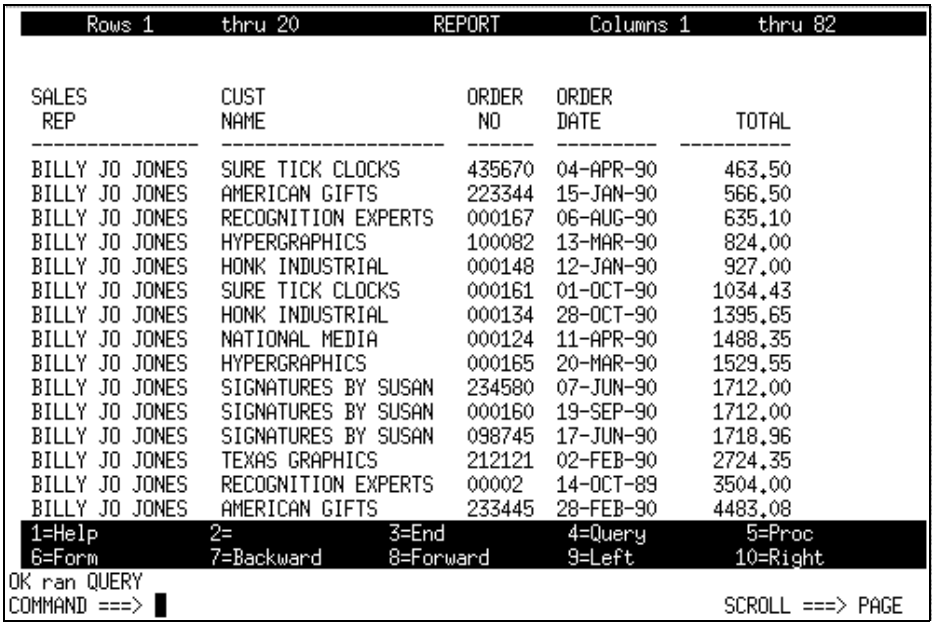
In this exercise, you will create a query, modify your profile, and use both in a query to print a report.

1. Set the language to SQL by typing **`SET LANGUAGE SQL`** and [Enter] on the Home screen command line.

2. Move the cursor to the SQL Query screen work area ([F4] a number of times) and type the following:



3. [F2] to validate the query.



4. [F4] to return to the SQL Query screen.

5. Type **`SAVE QUERY AS sales_rpt`** and [Enter].

☞ *Formatting the report*

Display the Main Form screen by typing **FORM** on the command line and [Enter].

1. Complete the form, following this example:

```
                                    FORM.MAIN

   Column Descriptions:                    Report width is now: 70
 NUM  COLUMN HEADING                              USAGE    INDENT  WIDTH  EDIT
 ---  ------------------------------------------  -------  ------  -----  -----
   1  Sales_Representative                                      2     15  C
   2  Customer_Name                                             2     20  C
   3  Order_Number                                              2      6  C
   4  Order_Date                                                2      9  TDOA-
   5  Order_Total                                               2     10  L2


PAGE:  Heading ==> Quarterly Sales Report
       Footing ==>
FINAL :Text    ==>
BREAK1:New page for BREAK? ==> NO
       Footing ==>
BREAK2:New page for BREAK? ==> NO
       Footing ==>
OPTIONS:Outline? ==> YES     Default BREAK text? ==> YES

 1=Help          2=Report        3=End           4=Form.Break1   5=Form.Options
 6=Query         7=Backward      8=Forward       9=Form.Page     10=Form.Columns
OK FORM performed
COMMAND ===>                                                    SCROLL ===> PAGE
```

2. Type **SAVE FORM AS qrtly_sales_rpt** and [Enter].
3. [F2] to see the report.

☞ *Creating a procedure*

1. [F5] to display the Procedure screen.
2. Type the following:

```
                                      PROC
  Top line # 1                                                  # of lines 1

-- Quarterly Sales Report: Q_Sales procedure
-- SALES_RPT Query
RUN sales_rpt (FORM=Qrtly_Sales_Rpt
PRINT REPORT
```

☞ *Saving and running a procedure*

1. Type **SAVE PROC AS q_sales** on the command line and [Enter].
2. Type **RESET PROC** and [Enter].
3. Type **RUN q_sales** and[Enter].

Summary

Procedures provide a way to automate the process of running regular queries and reports. In addition, you can create complex operations that are easy to execute on an ***ad hoc*** basis.

Appendix A
Initialization Files

Most of the Trifox tools and sub-systems read configuration and initialization data from special `.ini` files. These files typically have the same format:

```
option                value
```

The *option* is the name of the initialization option, setting name, or parameter. Lines with un-recognized options are ignored.

Value is the value of the option. Depending *onoption* the *value* can be a number, a yes/no, or a text string. The value can also represent one or more environment variables expressed as:

```
$(name)
```

The environment variable(s) are expanded before the value is evaluated.

The files support text strings as values, but they must be enclosed in double quotes ("), SQL-style, if blanks or quotes are part of the string. If no ending quote mark is provided, the string is terminated with a `\n`}.

If an option is not found in the file, then the default value is used.

The various relevant `.ini` files are described in detail in the following section(s).

Edit them using any ascii-based text editor. If you are reinstalling a product, we recommend you edit a ''clean'' copy of each `.ini` file, rather than modifying an existing one from your environment.

`qmr.ini` is used exclusively by TRIMqmr.

db_message_size

Type number

Default 512

Description Length (in characters) of database error message buffer.

Example The following specifies that the buffer hold no more than 400 characters:

```
db_message_size    400
```

fetch_buffer_size

Type number

Default 4096

Description Database fetch buffer size (in bytes).

Example The following specifies that the fetch buffer should be 4096 bytes:

```
fetch_buffer_size    4096
```

include_path

Type text

Default none

Description Where to look for #include files.

Example The following specifies that include files are located in c:\trifox\includes:

```
include_path     c:\trifox\includes
```

max_edit_lines

Type number

Default 500

Description Maximum number of lines in editor.

Example The following specifies that the editor contain 250 lines:

```
max_edit_lines     250
```

run_path

Type text

Default none

Description Where to look for run files.

Example The following specifies that run files are located in `c:\trifox\bin`:

```
run_path     c:\trifox\bin
```

oracle_number

Type text

Default data

Description Datatype to use when SAVing numeric data in Oracle.

Example The following specifies that numeric data be saved as Oracle data:

```
oracle_number    data
```

qmr_connect_cmd

Type yes/no

Default yes

Description Enable the CONNECT command. This is a security feature.

Example The following specifies that connect is disabled:

```
qmr_connect_cmd   no
```

share

Type yes/no

Default no

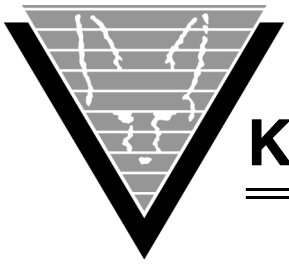
Description Default SHARE option.

Example The following specifies that sharing is allowed:

```
share    yes
```

Example

```
rem -------------- TRIMqmr configuration variables
db_message_size    300          -- max DB message length
fetch_buffer_size  4096         -- fetch buffer size (in bytes)
max_edit_lines     500          -- max # of edit buffer lines
oracle_number      number       -- default data type for ORACLE
save data
qmr_connect_cmd    yes          -- enable the CONNECT command
share              no           -- default SHARE option
rem
```



Appendix A
Key Mapping

The TRIM_HOME_TERM subdirectory contains the key mapping files, which are identified by the .key file_name extention. These files are used to map a keystroke or sequence of keystrokes to a function within TRIMqmr or a TRIMtools application.

The .key files can be used to remap any key to a function. For example, **Ctr+A** can be mapped to the function formerly performed by [F1].

There is a one-to-one correspondence between keys and functions.

The .key files also contains:

- Codes or characters for graphics and line drawings.
- The escape sequences to initialize and reset the terminal.
- The fixed, two-character function key labels used by TRIMqmr.

When a TRIMtools or TRIMqmr session runs it looks for a .key file. First, it searches the current directory for *user_name.key*.

For example:

```
LOGIN:  USER
PASSWORD: XXXX
$define TRIM_HOME DKA0: [QMR/TRIMTOOLS]
$trimrun MYAPPLICATION
$show deafult
DKA0: [USER_DIR]
```

The applicaiton looks for a DKA0:[USER_DIR]USER.KEY file. If it can't find that local key file, it reads the TERM environment variable to find that USER is running on a VT220 terminal.

Next, it tries to open the specific *termtype.key* file, DKA0:[QMR.TERM]VT220.KEY, and if that fails, returns an error message indicating that the file could not be found, or if it was found, was out of date.

You should run the GETKEY utility to solve the problem.

.KEY File

A .KEY file for a specific terminal type has the name of the terminal type with the .KEY file_name extention; for example, VT220.KEY.

The file is created and appropriately named by the GETKEY utility. However, GETKEY creates the file in a local directory. You must move it to TRIM_HOME/TERM or rename it to your *user_nam*e.KEY in order to use it with TRIMqmr.

Once a .KEY file exists, it can be directly used, or modified to take advantage of a terminal's graphics capability. The following example is a typical terminal .KEY for a VT220:

GETKEY Utility

The GETKEY utility allows users to redefine the keyboard mapping for use with TRIMqmr. To run GETKEY from the TRIMqmr root directory, type **getkey**.

The opening screen provides the necessary information. You first enter a key or keystroke sequence by which each sequence is terminated. Then, select a key or keystroke for each procedure. You are prompted for each key, simply enter it and end with the termination sequence.

GETKEY creates a new .KEY file in the current directory. To use the new file, you must move it to the TERM subdirectory. If one exists with the same name as the new file, and you want to save it, don't forget to rename the old file before moving the new one into the TERM subdirectory.

Symbols

A

B

C

D

E

V

W