

Query Management Facility  
Version 13 Release 1

*Using QMF*



**Note**

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

**2024-07-05 edition**

This edition applies to Version 13 Release 1 IBM® Db2 Query Management Facility (QMF) Classic Edition and Enterprise Edition, which are features of IBM Db2 13 for z/OS (5698-DB2). It also applies to Version 13 Release 1 of IBM Db2 QMF for z/OS (5698-QMF), which is a stand-alone IBM Db2 for z/OS tool. This information applies to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1982, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Rocket Software, Inc. 2007, 2022.**

---

# Contents

<b>About this information.....</b>	<b>xi</b>
Service updates and support information.....	xi
Highlighting conventions.....	xi
How to send your comments.....	xi
<b>Chapter 1. Getting acquainted with QMF.....</b>	<b>1</b>
Tables, columns, and rows.....	1
QMF sample tables.....	2
Accessing your data.....	3
QMF objects.....	3
Starting and ending QMF.....	4
Starting from the QMF home panel.....	4
Ending a QMF session.....	5
Issuing QMF commands.....	5
Entering a command on the command line.....	5
Entering commands using a function key.....	6
Entering commands on a command prompt panel.....	6
Retrieving a command.....	6
Setting up and changing your QMF user profile.....	7
Saving and retrieving objects in the database.....	9
Saving a QUERY, FORM, or PROC object.....	9
Saving a profile.....	10
Saving a DATA object as a table.....	11
Retrieving an object from the database.....	11
Using QMF help.....	12
Navigating in QMF help.....	12
Getting help about an error occurs.....	13
<b>Chapter 2. QMF in three quick lessons.....</b>	<b>15</b>
Lesson one: Finding the data you need.....	15
Lesson two: Choosing the specific data you want.....	17
Selecting the columns to display.....	17
Selecting the rows to display.....	18
Lesson three: Customizing a report.....	20
<b>Chapter 3. Displaying a list of database objects.....</b>	<b>23</b>
Displaying a list of database objects by using the List key.....	23
Displaying a list from a command prompt.....	24
Displaying a list of database objects by using the LIST command.....	25
Choosing specific objects using selection symbols.....	27
Entering commands on the list of database objects.....	28
Commands you can use on the list of database objects.....	28
Using a placeholder on the list of database objects.....	29
Correcting errors when you enter an incorrect command.....	30
Returning to the list from another QMF panel.....	30
<b>Chapter 4. Viewing the data in the database using Prompted Query.....</b>	<b>31</b>
Prompted Query panels.....	31
Rules for using Prompted Query.....	33
Starting Prompted Query and selecting tables.....	33

Column selection with Prompted Query.....	34
Selecting columns with Prompted Query.....	34
Creating a column by defining expressions.....	36
Creating a column that summarizes a group of rows.....	37
Row Selection with Prompted Query.....	37
Selecting rows.....	37
Narrowing row selection by using multiple row conditions.....	40
Sorting the rows in a query.....	41
Data display from multiple tables using Prompted Query.....	43
Displaying data from two tables.....	43
Eliminating duplicate rows from the report.....	45
Specifying a time period.....	47
Joining multiple tables.....	47
Joining multiple columns.....	50
Making your query reusable with substitution variables.....	50
Specifying a variable value as part of the RUN command.....	51
Specifying a value on the <b>RUN Command Prompt</b> panel.....	52
Specifying a value with a global variable.....	52
Running a query and displaying a report.....	52
Saving a new query.....	53
Canceling a running query.....	54
Changing a query.....	54
Erasing a saved query.....	56
Viewing the SQL equivalent of a prompted query.....	57
Converting a prompted query to an SQL query.....	57

## **Chapter 5. Viewing the data in the database using SQL statements..... 59**

Format of SQL queries.....	59
Starting an SQL query.....	60
Entering SQL statements and running the query.....	61
Including multiple SQL statements in one query.....	61
Spanning a long object name over multiple lines in a query.....	63
Selecting columns and tables.....	63
Creating a column by defining expressions.....	64
Row selection with SQL statements.....	65
Selecting rows.....	65
Selecting rows that have no data.....	65
Selecting rows by using specific character values.....	65
Selecting rows by using conditions.....	65
Selecting rows using opposite conditions.....	66
Selection symbols to narrow row selection.....	67
Narrowing row selection using multiple row conditions.....	67
Eliminating duplicate rows.....	69
Sorting the rows in a query.....	69
Adding or deleting lines in an SQL query.....	70
Data display from multiple tables using SQL statements.....	70
Displaying data from more than one table.....	70
Joining columns in two or more tables.....	70
Merging data from multiple tables into a single column.....	72
Creating a subquery to retrieve data from more than one table.....	74
Writing correlation names.....	76
SQL column functions.....	77
SQL scalar functions.....	78
Converting a value from one data type to another.....	79
Date and time formats.....	79
Allowing the database requester to control date and time format.....	81
Isolating the day, month, or year of a date.....	81

Isolating the hour, minutes, seconds, or fractional seconds of a time.....	83
Finding the length of a value.....	84
Displaying parts of a value.....	85
Replacing null values in the report with other values.....	85
Nested SQL functions.....	86
Adding and subtracting dates and times.....	87
Rules for date/time addition.....	87
Rules for date/time subtraction.....	87
The number of days between two dates.....	88
Accounting for months with different numbers of days.....	88
Using durations to represent date/time intervals.....	89
Incrementing and decrementing dates by durations.....	89
Subtracting dates.....	91
Making durations easier to read.....	92
Incrementing and decrementing times by durations.....	92
Subtracting times.....	93
Incrementing and decrementing timestamps by durations.....	93
The concatenation operator.....	94
Making your query reusable with substitution variables.....	95
Saving a new query.....	95
Inserting the results of a query into a table.....	96
Previewing reports for SELECT with change statements .....	96

**Chapter 6. Customizing your reports..... 101**

About the QMF form panels.....	101
The default report format.....	102
Report columns.....	103
Changing the columns on your report.....	103
Adding a column to a report.....	104
Changing the order in which columns are displayed.....	105
Changing column headings.....	106
Changing column widths and space between columns.....	107
Changing the alignment of the column heading and data.....	108
Specifying punctuation for the values in a column.....	109
Specifying the way QMF uses values in a column.....	112
When QMF displays column metadata instead of the actual data.....	113
Adding subtotals to a report.....	114
Breaking on a value to add subtotals.....	114
Specifying text for the subtotal line on a report.....	115
Adding page headings and footings.....	116
Specifying fixed columns on a report.....	117
Displaying a representative report before you select data.....	119
Refining page headings and footings.....	121
Using a global variable in a heading or footing.....	121
Adding the date, time, and page number to a heading or footing.....	122
Changing the alignment of page headings and footings.....	123
Adding break segments and text to your report.....	123
Adding break heading and footing text to a report.....	124
Refining the format of your report with detail blocks.....	127
Specifying text to appear at the end of your report.....	130
Calculating values to use in a report.....	132
Displaying a calculated value on a report.....	132
Displaying special conditions on your report.....	134
Mixing tabular data with reformatted text.....	139
Showing totals across rows in a report.....	140
Correcting errors on a form before displaying a report.....	142
Saving the report form.....	142

Resetting the values on a form panel.....	143
Printing your report.....	143
Limiting the number of rows returned from a query.....	144
<b>Chapter 7. Analyzing query results.....</b>	<b>147</b>
Charting QMF query results with QMF Analytics for TSO.....	153
Charting query results as a histogram.....	155
Charting query results as a map.....	161
Charting query results as a mixed chart.....	169
Charting query results as a pie chart.....	174
Charting query results as a plot chart.....	178
Charting query results as a tower chart.....	183
Analyzing query results with QMF Analytics for TSO Statistics.....	186
Running a basic statistical analysis on query results.....	188
Analyzing query results using bivariate curve fitting analysis.....	194
Analyzing query results using univariate curve fitting analysis.....	199
Running a discounted cash flow analysis of your query results.....	204
Running an F-Test analysis of your query results.....	209
Running a linear trend analysis of your query results.....	212
Running a Mann-Whitney U test of your query results.....	218
Running a Wilcoxon Signed-Rank test of your query results.....	222
Running your chart or statistical analysis from QMF for TSO.....	225
<b>Chapter 8. Displaying your report as a chart.....</b>	<b>227</b>
QMF chart formats.....	227
Where QMF report data appears on a chart.....	227
How QMF spaces data along the X-axis.....	228
Where data appears on pie charts.....	229
Chart data size limits.....	229
Displaying report data as a chart.....	230
Changing a chart with QMF forms.....	231
Working with charts using the ICU.....	235
Changing a chart format.....	235
Saving a chart format.....	235
Specifying a new default chart format.....	236
Fixing problems with charts.....	236
Printing your chart.....	237
<b>Chapter 9. Procedures to run QMF commands.....</b>	<b>239</b>
Linear procedures.....	239
An example linear procedure.....	239
Guidelines for writing linear procedures.....	240
Procedures with logic.....	241
An example procedure with logic.....	241
Guidelines for writing procedures with logic.....	242
Using REXX variables in procedures with logic.....	243
REXX error-handling instructions in procedures with logic.....	246
Calling programs from a procedure with logic.....	247
Running a procedure.....	249
Sharing a procedure with other QMF users.....	249
Creating reusable procedures with substitution variables.....	250
Specifying values for variables on the RUN command.....	250
Specifying values for variables using global variables.....	251
Specifying values on the RUN command prompt panel.....	251
Connecting to a remote location from a procedure.....	252
Writing a procedure that runs a generic query.....	254
Writing a generic SQL statement.....	254

Passing variables to the generic query.....	254
Writing a procedure that sets global variables for the generic query.....	255
Writing procedures that can run unattended.....	256
Inhibiting the display of panels.....	256
Using IMPORT or EXPORT commands.....	257
Using the EXIT command.....	257
Writing procedures for a National Language Feature.....	257
Error handling.....	257
Example of a procedure that requires no user interaction.....	258
<b>Chapter 10. Reusing QMF variables and input values.....</b>	<b>259</b>
The global variable list.....	259
Changing a global variable value.....	260
Adding a global variable.....	260
Deleting a global variable.....	261
Using the global variable list and the CASE option.....	261
Saving global variables across QMF sessions.....	262
Retaining panel input values.....	262
<b>Chapter 11. Creating tables.....</b>	<b>263</b>
Planning for your table.....	263
Creating a table.....	263
Saving and appending to a table.....	264
Creating a copy of a table.....	265
Creating a view of a table.....	265
Synonyms for tables or views.....	266
Aliases for tables or views.....	266
Deletion of tables, views, synonyms, and aliases.....	266
<b>Chapter 12. Maintaining the data in your tables.....</b>	<b>267</b>
Maintaining data with the Table Editor.....	267
Deciding when to save your data.....	267
New table rows using the Table Editor.....	267
Row changes using the Table Editor.....	271
Deleting rows from a table.....	273
Ending a Table Editor session.....	274
Using SQL statements to maintain data.....	274
New table rows using SQL statements.....	275
Row changes using SQL statements.....	276
Deleting rows from a table.....	278
Copying rows from one table into another.....	279
Adding a column to a table.....	279
Table access.....	279
Entering date and time values with QMF.....	281
<b>Chapter 13. Exporting and importing objects.....</b>	<b>283</b>
Exporting objects.....	283
Exporting objects from QMF for TSO.....	284
Exporting objects from QMF for CICS.....	284
Exporting QMF reports for use on the Internet.....	285
Exporting QMF report text.....	285
Scenario: Exporting data in CSV format.....	285
Importing objects.....	287
Importing objects from TSO data sets.....	287
Importing objects from CICS data queues.....	288
<b>Chapter 14. Remote data access.....</b>	<b>289</b>

Accessing data at a remote database.....	289
The QMF CONNECT command and access data at a remote database.....	289
Determining the database to which you are connected.....	290
Guidelines for the QMF Connect command.....	290
How to issue the CONNECT command.....	291
Reconnecting to a remote database.....	293
Three-part names in QMF commands to access data in a remote database.....	294
<b>Chapter 15. Introducing QMF Enhanced Editor.....</b>	<b>295</b>
Coloring object syntax.....	295
Copying and moving objects.....	296
Creating and replacing objects.....	297
Listing objects and Db2 tables.....	297
Updating object attributes.....	298
Running objects.....	299
Saving objects.....	299
Primary commands.....	300
Useful notes.....	301
Table display area.....	302
Previewing query results.....	303
Using Assist.....	303
Formatting queries.....	304
Syntax-checking queries.....	305
<b>Chapter 16. Multicultural support in QMF.....</b>	<b>307</b>
Bilingual command support.....	307
Exporting and importing bilingual forms.....	307
Working with double-byte character set (DBCS) data.....	308
How DBCS data looks when displayed.....	308
How DBCS data changes the length of names and fields.....	308
Data types you can use for DBCS data.....	309
DBCS data and QMF objects.....	310
Exporting and importing DBCS data.....	314
Printing DBCS reports.....	315
<b>Chapter 17. QMF with other products.....</b>	<b>317</b>
Edit objects external to QMF with ISPF.....	317
Editing an object by using the ISPF-PDF editor.....	317
Editing an object using a CLIST.....	317
Starting an edit session.....	318
Using ISPF from QMF.....	318
Inserting a QMF report in a document.....	318
Formatting the report.....	319
Specifying whether to insert a new or existing report.....	320
Information about your editor.....	321
Restrictions on the document interface.....	321
Using the QMF document interface.....	322
Accessing QMF from an editor.....	322
Accessing an editor from QMF.....	323
<b>Chapter 18. Db2 for z/OS stored procedures and QMF.....</b>	<b>325</b>
Running a CALL statement from the SQL QUERY panel.....	325
Examples of running a stored procedure.....	325
A stored procedure that returns no output parameters.....	325
A stored procedure that contains input and output parameters.....	326
A stored procedure that returns a result set.....	327
A stored procedure that returns multiple result sets.....	328



QMF forms with result sets returned from stored procedures.....	329
<b>Appendix A. Query-by-Example.....</b>	<b>331</b>
Starting a QBE query.....	331
Running and saving queries.....	331
Listing existing queries.....	331
Drawing example tables.....	332
Displaying all the columns of a table.....	333
Displaying certain columns of a table.....	334
Displaying certain rows of a table.....	335
Displaying rows that contain a certain value.....	335
Example elements.....	336
Writing expressions.....	336
Unnamed columns in an example table.....	338
Adding conditions to the example table.....	339
Adding a CONDITIONS box.....	339
Adding a target table.....	341
Eliminating duplicate rows.....	342
Displaying data from more than one table.....	342
Writing queries to be shared.....	344
QMF commands specific to QBE.....	346
CONVERT command.....	346
DELETE command.....	347
DRAW command.....	348
ENLARGE command.....	348
REDUCE command.....	350
QBE keyword reference.....	352
QBE exercises with solutions.....	372
<b>Appendix B. QMF sample tables.....</b>	<b>383</b>
Q.APPLICANT.....	383
Q.INTERVIEW.....	383
Q.ORG.....	384
Q.PARTS.....	385
Q.PRODUCTS.....	385
Q.PROJECT.....	386
Q.SALES.....	386
Q.STAFF.....	387
Q.SUPPLIER.....	388
Q.CASHFLOW.....	389
Q.CLIMATE_10YR.....	390
Q.CLIMATE_USA.....	390
Q.WORLDINFO.....	392
<b>Appendix C. QMF functions that require specific support.....</b>	<b>393</b>
Functions that vary according to database type.....	393
Functions not available in CICS.....	394
<b>Notices.....</b>	<b>397</b>
Trademarks.....	398
Terms and conditions for product documentation.....	398
Privacy policy considerations.....	398
<b>Glossary of terms and acronyms.....</b>	<b>401</b>
<b>Index.....</b>	<b>415</b>



## About this information

---

IBM Db2® Query Management Facility is a family of tightly integrated, powerful, and reliable tools that help you access relational or multidimensional data in the IBM Db2 product family and beyond.

These topics are designed to help both administrators and end users with:

- Understanding new features of products in the QMF family
- Understanding, at a high level, how each product in the QMF family works
- Understanding the features offered in each of the different QMF editions

This information introduces you to basic QMF tasks and includes examples that you can use and adapt for your own work. The following basic tasks are covered here:

- Retrieving data using the three different query methods available in QMF for TSO and CICS®
- Formatting data into reports or charts
- Creating QMF procedures as well as Db2 for z/OS® stored procedures that you can run from within QMF
- Editing data using SQL statements or the QMF Table Editor
- Accessing data at remote databases
- Working with double-byte character data

This information is for new and occasional QMF users and is designed to be used with , which contains detailed information about QMF commands and panels. More information about any topic is also available in the QMF online help.

## Service updates and support information

---

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

[IBM Software Support website](#)

## Highlighting conventions

---

This information uses the following highlighting conventions:

- **Boldface** type indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- Monospace type indicates examples of text that you enter exactly as shown.
- *Italic* indicates the titles of other publications or emphasis on significant terms. It is also used to indicate variables that you should replace with a value.

## How to send your comments

---

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other documentation, use either of the following options:

- Use the online reader comment form, which is located at:

<http://www.ibm.com/software/data/rcf>

- Send your comments by e-mail to [comments@us.ibm.com](mailto:comments@us.ibm.com). Be sure to include the name of the book, the part number of the book, the version of your product, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).



# Chapter 1. Getting acquainted with QMF

Query Management Facility (QMF) lets you work with data that is stored in relational databases, including many IBM databases.

- Db2 for z/OS
- Db2 for Linux®, UNIX, and Windows
- DB2® for iSeries
- DB2 Server for VSE and VM

## Tables, columns, and rows

In QMF, data is arranged in tables, which are sets of data arranged in rows and columns.

The following figure shows an example of a table.

	COLUMN						
ROW	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	10	SANDERS	20	MGR	7	18357.50	-
	20	PERNAL	20	SALES	8	18171.25	612.45
	30	MARENGHI	38	MGR	5	17506.75	-
	40	O'BRIEN	38	SALES	6	18006.75	846.55
	50	HANES	15	MGR	10	20659.80	-
	60	QUIGLEY	38	SALES	-	16808.30	650.25
	70	ROTHMAN	15	SALES	7	16502.83	1152.00
	80	JAMES	20	CLERK	-	13504.60	128.20
	90	KOONITZ	42	SALES	6	18001.75	1386.70
	100	PLOTZ	42	MGR	7	18352.80	-
	110	NGAN	15	CLERK	5	12508.20	206.60
	120	NAUGHTON	38	CLERK	-	12954.75	180.00

Figure 1. Data in tables is arranged in columns and rows.

Columns:

- Appear vertically on the screen.
- Contain data of the same kind.
- Are referred to by names, which appear at the top as headings.

Rows:

- Appear horizontally on the screen
- Contain different kinds of data about a single thing
- Have no names

In QMF, you normally refer to tables by using two-part names: a table identifier and an owner identifier, separated by a period. For example, the exercises in this information use a table named Q.STAFF, where STAFF identifies the table, and Q identifies the owner of that table. Normally, the person who creates a table is the owner of that table. QMF identifies the owner of the table by using the user ID of the individual who created it. The owner of a table can authorize others to access the information in the table. When

referring to your own tables, you can leave out the owner identifier. QMF assumes that you are referring to a table that you own.

If your site supports three-part names, also known as distributed unit of work, and you are using QMF on a Db2 for z/OS database, you can use a table from a remote location by including a location identifier. For example, NEW\_YORK.JOHN.ACCOUNTS refers to a table named ACCOUNTS, which is owned by user JOHN and is located in a remote Db2 database known as NEW\_YORK.

### Related concepts

Three-part names in QMF commands to access data in a remote database

You can use QMF to access tables and views that are stored in a remote database by using a three-part name to refer to the table or view. By using three-part names, you can access and manipulate the data that you need without first having to use the QMF CONNECT command to connect to the remote database.

## QMF sample tables

QMF provides some sample tables that you can use while learning QMF before you begin working with your own tables.

The sample tables are used throughout the examples in this information. They contain information about the J & H Supply Company, a fictitious electrical parts manufacturer.

The following table describes what each sample table contains.

<i>Table 1. QMF sample tables.</i> QMF provides sample tables to use while learning the program.	
<b>Sample table name:</b>	<b>Contains information about:</b>
Q.APPLICANT	People who have applied for jobs with J & H Supply Company
Q.INTERVIEW	Interview dates and times, in ISO format, of people who have applied for jobs with the company
Q.ORG	Organization of the company, showing both departments and divisions within the company
Q.PARTS	Materials supplied to J & H Supply Company
Q.PRODUCTS	Products produced by the company
Q.PROJECT	J & H Supply Company projects
Q.SALES	Sales made by salespeople at the company
Q.STAFF	The employees of the company
Q.SUPPLIER	Companies who supply materials to J & H Supply Company

Additionally, QMF Analytics for TSO provides sample tables that you can use to learn about QMF Analytics for TSO functions. The following table shows the type of data that is contained in each sample table.

<i>Table 2. QMF Analytics for TSO sample tables</i>	
<b>Table</b>	<b>Contains information about:</b>
Q.CASHFLOW	Data about cost and revenue that can be used with QMF Analytics for TSO discounted cash flow analytics
Q.CLIMATE_10YR	Data about climate over a 10-year period
Q.CLIMATE_USA	Data about climate in the United States, including data about rainfall and sunshine
Q.WORLDINFO	Data about geographic regions in which the J & H Supply Company conducts business

## Related concepts

[QMF sample tables](#)

## Accessing your data

---

To get information from a database, you write a query, which is a request to the database to retrieve a specific set of data.

You can use QMF to send your request for information to the database in three different ways. Each of the three ways has its own rules:

### Prompted Query

Prompted query is an easy-to-use query method that displays prompt panels to help you choose the specific information you are looking for. You do not need to know the syntax for your database request; Prompted Query does the work for you by converting your request into a language the database can understand. This method is especially suited for beginners and occasional QMF users.

### SQL (Structured Query Language)

SQL is a powerful query language that lets you define, retrieve, change, and authorize access to data. It has a specific syntax you must follow so the database can process your request. It is especially suited for users who must work with large amounts of data or those who work with QMF frequently.

### QBE (Query-by-Example)

QBE is a graphical query method that lets you retrieve and change data with just a few keystrokes.

## Related concepts

[Query-by-Example](#)

[Viewing the data in the database using Prompted Query](#)

You can use Prompted Query to select and view data in the database. If you are new to QMF, or if you only use QMF occasionally, Prompted Query is an easy way to retrieve your data.

[Viewing the data in the database using SQL statements](#)

You can select and view data in the database by using SQL statements.

## QMF objects

---

QMF stores information as QMF objects.

Some objects, such as queries, forms, procedures, and QMF Analytics for TSO objects, are stored in the database. Other objects, such as reports and form-based charts, exist only in temporary storage while you work on them.

The types of QMF objects are shown in the following table.

Object name	Contains
QUERY	Specifications for retrieving or manipulating data in the database
ANALYTIC	Specifications for charts and statistics created using QMF Analytics for TSO
DATA	Data you select using a query or data you import from outside QMF The DATA object can be formatted using QMF forms to produce a report. To see the retrieved data that is currently in temporary storage, issue the command SHOW REPORT or DISPLAY REPORT.
REPORT	The DATA object as formatted by the form currently in the FORM temporary storage area (or a form you specify when you run your query)
FORM	Report formatting specifications for the data you select
CHART	Graphic display of formatted report data

Object name	Contains
PROC	A series of QMF commands that you can issue with a single RUN command; PROC is the abbreviation for PROCEDURE in QMF
FOLDER	A container to group references to QMF queries, procedures, forms, and analytic objects.
PROFILE	Specifications for your QMF user session

Each QMF object in the database has an owner, who is usually the person who saved the object in the database. You cannot use an object unless you own it, the owner has authorized you to use it, or you are a QMF administrator. A QMF administrator can use any object that is owned by anyone.

## Starting and ending QMF

Your QMF administrator can tell you how to start a QMF session and give you a user ID.

People who perform setup and maintenance tasks for QMF are called "QMF administrators". Check with your administrator if you have any questions about getting started with QMF.

### Starting from the QMF home panel

After you start QMF, you see the QMF home panel. The home panel displays basic information about your session.

```

-----
-----
QMF HOME PANEL                               Query      Management  Facility
Version 13 Release 1

Authorization ID 1
KMRAMEY

Connected to
DB2P 2

*****  **  **  *****
**  **  ***  ***  **
**  **  ****  ****  *****
**  **  **  **  **  **  **
**  *  **  **  ****  **  **
*****  **  ***  **  **

**
http://www.ibm.com/qmf
Enter a command on the command line or press a function key.
For help, press the Help function key or enter the command HELP.

-----
1=Help      2=List      3=End      4=Show      5=Chart      6=Query 3
7=Retrieve  8=Edit Table  9=Form     10=Proc     11=Profile   12=Report
OK, END command executed successfully. 4
COMMAND ==> 5

```

Figure 2. The QMF home panel

On the home panel, you see:

**1 Authorization ID**

The user ID you used to log on to QMF.

You can access only objects (such as queries, forms, or procedures) created using this ID and objects to which this ID has been granted access. Any object you create during a session belongs to this user ID.

**2 Connected to**

The name of the database to which you are connected.



### **3 Function keys**

A function key performs a single operation.

The function of each key depends on the panel you are looking at. You can reduce keystrokes by pressing a single function key to start the operation you want to perform.

If your administrator customizes your function keys, they may not be the same as the examples presented in this information.

### **4 The message line**

On this line, QMF tells you what was accomplished by the last operation you started or what you can do next.

### **5 The command line**

If no function key is programmed to start the operation you want to do next, you can tell QMF what to do by entering a command on this line after the arrow. When you see "Enter a command" in this information, type the command on the command line and press Enter. Some QMF panels display SCROLL ==> PAGE on the right side of the command line. You can type another value over the PAGE value to change the scrolling behavior when you press the Forward or Backward function keys.

#### **Related concepts**

Entering commands You can issue QMF commands in several ways: on the command line, with a function key, on a prompt panel, or from a procedure or application.

## **Ending a QMF session**

You can end a QMF session from the home panel or you can bypass the home panel and end the session directly.

#### **Procedure**

To end a QMF session, choose one of the following options.

- To end a QMF session from the home panel:
  - a) From any QMF panel, press the End function key.  
The QMF home panel appears.
  - b) Press the End key again to end the QMF session.
- To bypass the home panel and end a QMF session, enter EXIT on the command line.

## **Issuing QMF commands**

---

You can issue QMF commands by entering a command on the command line, pressing a function key, or specifying a command on a prompt panel.

### **Entering a command on the command line**

You can enter a command on any QMF panel that has a command line. You can enter the full command or use the minimum unique abbreviation for any command that can be abbreviated. For example, to display a report that contains data from the sample table Q.STAFF, you can enter DISPLAY Q . STAFF or DI Q . STAFF.

#### **Related concepts**

Entering commands using a function key

QMF commands can be entered by using a function key. Each QMF panel has a default set of function keys; these defaults can be changed to your preferences. The keys you see on your QMF panels might be different than the ones in this information.

#### **Related tasks**

Entering commands on a command prompt panel

Command prompt panels ask you for the required information that you need to enter a QMF command.

Retrieving a command

You can retrieve and reissue commands that you already entered.

## Entering commands using a function key

QMF commands can be entered by using a function key. Each QMF panel has a default set of function keys; these defaults can be changed to your preferences. The keys you see on your QMF panels might be different than the ones in this information.

### Related concepts

[Entering a command on the command line](#)

You can enter a command on any QMF panel that has a command line. You can enter the full command or use the minimum unique abbreviation for any command that can be abbreviated. For example, to display a report that contains data from the sample table Q.STAFF, you can enter `DISPLAY Q . STAFF` or `DI Q . STAFF`.

### Related tasks

[Entering commands on a command prompt panel](#)

Command prompt panels ask you for the required information that you need to enter a QMF command.

[Retrieving a command](#)

You can retrieve and reissue commands that you already entered.

## Entering commands on a command prompt panel

Command prompt panels ask you for the required information that you need to enter a QMF command.

### Procedure

To enter commands on a command prompt panel, choose one of the following options:

- To see a command prompt panel, enter the command and follow it with a space and a question mark. For example, enter `RUN ?` to display the prompt panel for the `RUN` command.
- To see a command prompt panel when a QMF error message is displayed (for example, if you entered a command but received an error message), press Enter while the message is displayed. QMF displays the command prompt panel so you can enter the correct information.

### Related concepts

[Entering a command on the command line](#)

You can enter a command on any QMF panel that has a command line. You can enter the full command or use the minimum unique abbreviation for any command that can be abbreviated. For example, to display a report that contains data from the sample table Q.STAFF, you can enter `DISPLAY Q . STAFF` or `DI Q . STAFF`.

[Entering commands using a function key](#)

QMF commands can be entered by using a function key. Each QMF panel has a default set of function keys; these defaults can be changed to your preferences. The keys you see on your QMF panels might be different than the ones in this information.

### Related tasks

[Retrieving a command](#)

You can retrieve and reissue commands that you already entered.

## Retrieving a command

You can retrieve and reissue commands that you already entered.

### Procedure

- To retrieve a command entered on the command line, enter `RETRIEVE` or `?` to see the last command entered.  
You can use multiple question marks.

For example, entering ? retrieves the last command you entered; entering ?? retrieves the command before that, and so on.

### **Related concepts**

#### Entering a command on the command line

You can enter a command on any QMF panel that has a command line. You can enter the full command or use the minimum unique abbreviation for any command that can be abbreviated. For example, to display a report that contains data from the sample table Q.STAFF, you can enter DISPLAY Q . STAFF or DI Q . STAFF.

#### Entering commands using a function key

QMF commands can be entered by using a function key. Each QMF panel has a default set of function keys; these defaults can be changed to your preferences. The keys you see on your QMF panels might be different than the ones in this information.

### **Related tasks**

#### Entering commands on a command prompt panel

Command prompt panels ask you for the required information that you need to enter a QMF command.

## **Setting up and changing your QMF user profile**

---

Your QMF user profile tells QMF how you want to display information and process commands, and what printer to use when you print reports or charts.

### **About this task**

Your user profile is the same as the system default when you first begin to use QMF. You can change the information in your profile to match your specific needs at any time.

### **Procedure**

Choose one of the following options, depending on whether you want to display, change, or save your QMF user profile:

- To display your QMF user profile, do one of the following operations:
  - Press the Profile function key.
  - Enter SHOW PROFILE (or SH PROF) on the command line.

The **Profile panel** displays, as shown in the following figure.

```

PROFILE

General Operands:
CASE    ==> UPPER    Enter UPPER, STRING, or MIXED.
DECIMAL ==> PERIOD   Enter PERIOD, COMMA, or FRENCH.
CONFIRM ==> YES      Enter YES or NO.
LANGUAGE ==> PROMPTED Enter SQL, QBE, or PROMPTED.
MODEL   ==> REL      QMF Administration operand. May not be changed.

Defaults for printing:
WIDTH   ==> 132      Number of characters per line.
LENGTH  ==> 60       Number of lines per page.
PRINTER ==>          Printer to be used for output.

QMF Administration Operands: (Not usually changed)
SPACE   ==> "DSQDBDEF"."DSQTSDEF"
          Enter the name of Db2 DATABASE or TABLESPACE in which
          tables will be saved by the SAVE DATA command.
TRACE   ==> NONE
          Enter ALL, NONE or a character string of function-id,
          trace-level pairs.

1=Help      2=Save      3=End      4=Print      5=Chart      6=Query
7=          8=          9=Form    10=         11=         12=Report

COMMAND ==>

```

Figure 3. The QMF **Profile panel** controls aspects of your QMF session.

- To change your QMF user profile, position the cursor on any value in your QMF profile and type over it with the value you want.

You can change any of the profile values at any time. The following table shows some of the values of the options in your profile.

Table 4. Some settings in the QMF profile. You can set your QMF profile with your preferences.	
Profile value	Explanation
CASE ==> UPPER	QMF recognizes commands only in uppercase. If you want to enter information in mixed or lowercase, change your profile to show CASE ==> UPPER so that QMF converts your data entries to uppercase. However, if your site uses mixed-case passwords, set the CASE value to MIXED if you use the CONNECT command. Otherwise, QMF converts your password to uppercase upon entry and fails to connect to the target database. When CASE is MIXED, you must enter all QMF commands in uppercase for them to be recognized.
DECIMAL ==> PERIOD	Although other indicators (such as a comma) are available, the examples in the topics in this information use a period as a decimal point indicator.
CONFIRM ==> YES	When CONFIRM ==> YES is specified, QMF displays a confirmation panel before a command changes or replaces an object in the database. For the exercises in this information, be sure YES is specified.
LANGUAGE ==> PROMPTED	Choose LANGUAGE ==> PROMPTED when you want QMF to prompt you for the information you need to write a query.  Choose LANGUAGE ==> SQL when you want to write queries directly in Structured Query Language (SQL).  Choose LANGUAGE ==> QBE when you want to use the QMF Query-by-Example feature.

- To save your QMF user profile after you have all the values set the way that you want, press the Save function key or enter `SAVE` or `SAVE PROFILE` on the command line. QMF stores the changed profile in the database, and uses the changed profile the next time you begin a QMF session.

If you press the End function key before you save your profile, the changes you make to your profile remain in effect only until you end your QMF session.

### Related reference

[SET PROFILE](#)The `SET PROFILE` command changes values in your QMF profile. These values influence the behavior of your QMF session.

## Saving and retrieving objects in the database

---

When you display or work with any QMF object, QMF places a copy of that object in a temporary storage area. There is one temporary storage area for each type of object, so you can only have one of each object in temporary storage at a time.

QMF places `QUERY` objects in the `QUERY` temporary storage area, `DATA` objects (which are formatted by either the default QMF form or a form you specify when you run a query) in the `REPORT` temporary storage area, and so on.

Changes that you make to objects in temporary storage do not change the actual objects stored in the database unless you save those objects when you finish working with them. Thus, you will lose changes to an object in temporary storage if you end a QMF session without saving the object. You will also lose changes if you display another object of the same type before saving the one you are currently working with.

You cannot save a `REPORT` object. Instead, you save the query and form that produce the report. To save a `CHART`, you save the data and the chart format.

### Related concepts

[QMF objects](#)

QMF stores information as QMF objects.

[Displaying your report as a chart](#)

You can display your tabular data in a wide variety of charts. You can change QMF chart formats, or create new chart formats.

## Saving a `QUERY`, `FORM`, or `PROC` object

You can save a query, form, or procedure by entering the `SAVE` command on the QMF command line.

### Procedure

To save a query, form, or procedure, choose one of the following options:

- If you are on the `QUERY`, `FORM`, or `PROC` panel, and you want to save the currently displayed object, enter:

```
SAVE
```

If the object is an existing one that you loaded from the database, QMF saves it using its existing name.

If the object is new, QMF prompts you for a name for the object.

You can also enter the following command:

```
SAVE AS objectname
```

where *objectname* is the name you want to assign to the object.

If the object is an existing one that you loaded from the database, QMF saves it under the new name. The object stored in the database with the old name remains unchanged.

- If you are on any QMF panel, and you want to save a currently loaded object even though it is not currently displayed, enter:

```
SAVE object AS objectname
```

In this command, *object* is the type of object you want to save and *objectname* is the name you want to assign to the object. For example, if you are on the FORM panel, and you want to save the query that is currently loaded in temporary storage under the name MYQUERY, enter SAVE QUERY AS MYQUERY.

You can use this command syntax for new or existing objects. If the object is an existing one, QMF saves the currently loaded object with the new name. The object stored in the database with the old name remains unchanged.

- If you want to save an object and share it with other users, add the SHARE=YES parameter to the SAVE command:

```
SAVE (SHARE=YES
SAVE AS objectname (SHARE=YES
SAVE object (SHARE=YES
SAVE object AS objectname (SHARE=YES
```

If you issue a SET GLOBAL command with the value DSQEC\_SHARE=1 prior to issuing the SAVE command, you do not need to include the SHARE=YES parameter.

- If you want to save QMF queries, procedures, or forms into a folder for the purpose of organizing objects into meaningful groups, add the FOLDER parameter to the SAVE command:

```
SAVE object AS objectname (FOLDER=foldername
```

Alternately, instead of specifying the FOLDER parameter and folder name on the SAVE command, you can set the DSQEC\_CURR\_FOLDER global variable to specify a folder name that will be used when the SAVE command is executed. If you set the DSQEC\_CURR\_FOLDER global variable, and a QMF object is successfully saved, information about that QMF object will also be added to the folder that you specified in DSQEC\_CURR\_FOLDER. If the FOLDER keyword is specified with the SAVE command, that folder name overrides the folder name that is set in DSQEC\_CURR\_FOLDER.

#### Notes:

- A folder object entry does not contain any QMF object data; it contains only the QMF query, procedure, or form object name.
- You can save and add a QMF analytic object to a folder object by using QMF Analytics for TSO. The QMF global variable DSQEC\_CURR\_FOLDER is used when an analytic object is saved. You can override the global variable by updating the folder name in the **Folder** field for the analytics save command.

#### Related reference

**SAVE**The SAVE command saves in the database at the current location objects that are currently in QMF™ temporary storage.

**SET GLOBAL**The SET GLOBAL command assigns values to global variables from the QMF command line, from a procedure, or through the callable interface.

**QMF global variables**QMF provides many global variables that help you control aspects of your QMF session, QMF commands, and panel display. The global variables also help you control behavior of QMF functions in procedures and applications.

## Saving a profile

You can have only one PROFILE object in the database.

### Procedure

To save your profile, choose one of the following options:

- Enter either of the following commands on the QMF command line of the PROFILE panel:

```
SAVE
SAVE PROFILE
```

- Enter SAVE PROFILE on the command line of any QMF panel.

## Saving a DATA object as a table

If you want to save the data in a DATA object, you must save it as a table.

### About this task

A DATA object is not stored in the database, but is created for you to work with temporarily when you create and run a query. All data is stored in the database in tables.

### Procedure

To save a DATA object as a table, enter the following command on the QMF command line:

```
SAVE DATA AS tablename
```

where *tablename* is the name you want to assign to the new table.

### Related reference

[SAVE](#)The SAVE command saves in the database at the current location objects that are currently in QMF™ temporary storage.

## Retrieving an object from the database

You can retrieve any object from the database after you save it.

### Procedure

- To retrieve an object from the database, enter one of the following commands on the QMF command line:

- Enter the following command on the QMF command line:

```
DISPLAY objectname
```

where *objectname* is the name of the specific object.

For example, to retrieve an object that is named MYQUERY, enter DISPLAY MYQUERY.

- You can also enter the following command:

```
DISPLAY object objectname
```

where *object* is the type of object you want to retrieve, and *objectname* is the name of the specific object.

For example, to retrieve and display a procedure named MYPROC, enter DISPLAY PROC MYPROC.

### Related reference

[DISPLAY](#)The DISPLAY command displays an object from QMF temporary storage or an object from the database.

## Using QMF help

---

You can display online help to learn about QMF as well as to help you troubleshoot any errors or problems that occur. To see help information, press the Help function key from anywhere within QMF.

### Navigating in QMF help

The QMF main help menu for QMF for TSO and CICS provides a list of general topics, including topics related to QMF Analytics for TSO. From this menu, you can choose more specific topic panels.

QMF help varies slightly in how it allows you to access information, depending on what part of QMF you are using. When you press the Help function key:

- In most parts of QMF, you see a menu related to the QMF panel you are using. Select the topic or area of information that you want to see.
- In Prompted Query or the Table Editor (a feature that lets you add, change, or delete data in tables), you immediately see information specifically related to the part of the process you are working on.
- While you are creating an SQL query, you see a table of contents from which you can select the information you want to see.
- While you are working on form panels, you see information specifically related to the field on which you are working.
- If an error message is currently displayed above the QMF command line, additional information related to the message is shown.

If you installed and configured QMF Analytics for TSO, you can access a list of general help topics from the QMF Analytics for TSO home page by selecting the Help function key and then pressing the Menu option. QMF Analytics for TSO also provides context-sensitive help for the Parameter Selection panels and the fields on the Parameter Selection panels. The help that opens (field-level or panel-level) depends on the location of the cursor when you select the Help option.

The following functions are available on function keys to help you navigate through QMF help:

#### **Exit**

Immediately removes all help panels and activates the underlying QMF panel.

#### **More Help**

Shows you a menu of panels that are related to the help panel currently displayed (available for selected topics).

#### **Menu**

Displays either the most recently displayed menu, or the menu for the underlying QMF panel.

You can repeatedly press the Menu function key to return to a menu one level higher than the one you are currently viewing.

#### **Backward**

Scrolls backward through the panel.

#### **Forward**

Scrolls forward through the panel.

#### **Keys**

Lists the functions of the keys for the underlying QMF panel.

#### **Cancel**

Closes one help panel at a time.

You can return to the underlying QMF panel by repeatedly pressing the Cancel function key.

#### **Switch**

On help panels for some specific topics, activates the underlying QMF panel; you can enter commands on the command line of the QMF panel while the help panel is still displayed.



## Getting help about an error occurs

When you misspell a command or try to run a query that is worded incorrectly, QMF presents a brief explanation of the error on the message line of the panel. If you need more information about the error, press the Help function key.

You can also enter the HELP command on the command line. A panel displays a more detailed explanation of the error and a suggestion about how to correct the error.

For example, to see how error message help works, type the command SHOW PROFILE in uppercase characters on the command line of the home panel. You see the following error message:

You cannot show PROFILE.

To find out more, press the Help function key. A help panel like the one in the following figure displays.

```
Help: Message                                1 to 13 of 41
You cannot show PROFILE.
Explanation:
  The SHOW command must be followed by one of the following keywords:
o Home
o Query
o Form
o PROFILE
o PROC
o Report
o Globals
o CHART
```

*Figure 4. An example of help text for a QMF error message*

The help explains that the correct command is SHOW PROFILE. To scroll through the help panel, press the Forward function key.

For even more information, press the More Help function key if it is available. What you see depends on what you were doing when you first displayed the help. For example, You see an error message while you are editing a table in the Table Editor, so you press the Help function key. You then press the More Help function key. You see the list of help panels that are associated with different aspects of the Table Editor. This help is identical to the help that displays when you press the Help function key directly from the Table Editor. This help displays on top of the first help panel. Press the Cancel function key to close each panel.



## Chapter 2. QMF in three quick lessons

Working with QMF usually requires three basic tasks: finding data you need, selecting specific items from that data, and turning the data into a report. This topic gives you the following three quick lessons on how to do these tasks using the data in the QMF sample tables.

### Related concepts

[Getting acquainted with QMF](#)

Query Management Facility (QMF) lets you work with data that is stored in relational databases, including many IBM databases.

## Lesson one: Finding the data you need

In this lesson, you create a query to show the records for all the clerks in the Q.STAFF table. You use the Prompted Query method, which is the easiest method to start with.

### About this task

The Q.STAFF table is one of the sample tables that comes with QMF.

### Procedure

Select the Q.STAFF table by completing the following steps.

1. On the command line of the QMF home panel, enter:

```
RESET QUERY (LANG=PROMPTED
```

The **Prompted Query** panel displays the **Tables** panel, as shown in the following figure:

```
PROMPTED QUERY                                     LINE    1
Tables:
> ...
*** END ***
Type one or more table names.
1 to 5 of 15
(
(
(
(
(
F1=Help F4=List F5=Show Field F7=Backward
F8=Forward F12=Cancel
```

Figure 5. The **Tables** panel

Normally, you would type the name of the table you want to use (Q.STAFF) on the **Tables** panel and press Enter. However, for this lesson you choose the table from a list.

In addition, you limit the list to only the tables that belong to user Q and that begin with the letter S. To do that, you use selection criteria. In this case, you use the characters "q.s" followed by a "%" sign.

2. Type q.s% on the first line of the **Tables** panel.
3. Press the List function key.

The **Table List** panel displays with the names of all the tables that are owned by user Q and begin with S.

4. Move the cursor to STAFF and type x, as shown in the following figure.

```

PROMPTED QUERY
Tables:
> ...
*** END ***

```

Table List		1 to 5 of 5
Name	Owner	
SALES	Q	
SALG	Q	
SAMTALE	Q	
x STAFF	Q	
SUPPLIER	Q	

```

F1=Help F5=Describe F7=Backward F8=Forward
F10=Comments F11=Sort F12=Cancel
OK, LIST command executed successfully.

```

Figure 6. The **Table List** panel

5. Press Enter.

The **Tables** panel displays again with Q.STAFF on the first line.

6. Press Enter to select the Q.STAFF table.

## Results

Q.STAFF is displayed under the Tables heading on the left side of the **Prompted Query** panel, in the echo area. The echo area shows you each part of the query as you create it.

The **Specify** panel, which is shown in the following figure, is also displayed. Now that you selected a table, use the **Specify** panel to create the rest of the query.

```

PROMPTED QUERY
Tables:
Q.STAFF
Columns:
ALL
*** END ***

```

MODIFIED	LINE	1
		Specify
		Select an item.
2	1.	Tables...
	2.	Columns...
	3.	Row Conditions...
	4.	Sort...
	5.	Duplicate Rows...
	6.	Time Period...
		F1=Help F12=Cancel

Figure 7. The **Specify** panel

## Related concepts

### [Creating tables](#)

You can create tables in the database by using SQL statements on the SQL Query panels.

### [Displaying a list of database objects](#)

To quickly view information about database objects, you can display a list of those objects.

### [Maintaining the data in your tables](#)

After you create your tables, you can add to or change the data in them.

### [Viewing the data in the database using Prompted Query](#)

You can use Prompted Query to select and view data in the database. If you are new to QMF, or if you only use QMF occasionally, Prompted Query is an easy way to retrieve your data.

### [Viewing the data in the database using SQL statements](#)

You can select and view data in the database by using SQL statements.

## Lesson two: Choosing the specific data you want

After you locate the table that you want to work with, you can choose specific data from the table. To do this, you select the columns and rows you want to display.

### Related concepts

Viewing the data in the database using Prompted Query

You can use Prompted Query to select and view data in the database. If you are new to QMF, or if you only use QMF occasionally, Prompted Query is an easy way to retrieve your data.

Viewing the data in the database using SQL statements

You can select and view data in the database by using SQL statements.

## Selecting the columns to display

Columns contain data of the same kind for each individual entry in the table. For example, the column called JOB contains data about the job title of each person in the Q.STAFF table. You want to see several columns from the table for this lesson.

### About this task

Choice 2, **Columns**, is already selected, because selecting columns is usually the next step in creating a query.

If you look in the echo area on the Specify panel, you can see that the word "ALL" appears directly under the **Columns** heading. If you do not select specific columns, QMF automatically selects all the columns in the table.

### Procedure

Complete the steps below to select specific columns.

1. Press Enter.

The Columns panel displays, which shows all the columns in the Q.STAFF table. You might need to scroll forward to see them all.

At the bottom of the panel, you will see selections for **Expression** and **Summary Functions**.

2. As shown in the following figure, type x beside each of the following columns: NAME, DEPT, JOB, SALARY, and COMM.

```
PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
> ...
*** END ***

+-----+
| Columns                                     |
| Select one or more columns. You can also  |
| select either an expression or function.  |
|                                           |
| Q.STAFF -- all                            |
| ID                                         |
| x NAME                                     |
| x DEPT                                    |
| x JOB                                     |
| YEARS                                     |
| x SALARY                                  |
| x COMM                                    |
|                                           |
| 1. Expression (A+B, etc.)...              |
| 2. Summary Functions (SUM, etc.)...       |
+-----+
| F1=Help F5=Describe F7=Backward          |
| F8=Forward F12=Cancel                    |
+-----+
```

Figure 8. The Columns panel

3. Press Enter.

## Results

The columns you selected are displayed in the echo area, and the Specify panel displays again.

### Related concepts

[Viewing the data in the database using Prompted Query](#)

You can use Prompted Query to select and view data in the database. If you are new to QMF, or if you only use QMF occasionally, Prompted Query is an easy way to retrieve your data.

## Selecting the rows to display

You can select the rows to display on your report. For this lesson, select only the rows for employees who are clerks.

### About this task

In QMF, selecting specific rows to display is called creating a *row condition*.

### Procedure

To create a row condition:

1. Press Enter to select choice 3, **Row Conditions**.

The Row Conditions panel displays, as shown in the following figure.

```
PROMPTED QUERY
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
Row Conditions:
> If...
*** END ***
```

MODIFIED	LINE	1
+-----+-----+-----+-----+-----+-----+		
Row Conditions		
Begin a condition by selecting one column, or by entering an expression or function.		
		1 to 8 of 8
* Q.STAFF		
2. ID		
3. NAME		
4. DEPT		
5. JOB		
6. YEARS		
7. SALARY		
8. COMM		
Expression (A+B, etc.)...		
+-----+-----+-----+-----+-----+-----+		
F1=Help F5=Describe F7=Backward		
F8=Forward F12=Cancel		
+-----+-----+-----+-----+-----+-----+		

Figure 9. The Row Conditions panel

To create a row condition, select a column on which to base your row condition. You can use any column in the table, even if you are not displaying it on the report.

2. In this example, you want to display only rows where the job is CLERK; select the JOB column by typing 5.
3. Press Enter.

As shown in the following figure, the Comparison Operators panel displays:

```

PROMPTED QUERY                                MODIFIED LINE 1
-----+-----+-----+
Tables:                                       | Row Conditions |
  Q.STAFF                                    +-----+
Columns:                                     | Comparison Operators |
  NAME                                       | Select a verb and a comparison. |
  DEPT                                       | Verb. . . . 1 1. Is |
  JOB                                        | 2. Is not |
  SALARY                                     | Comparison 1 1. Equal to... |
  COMM                                      | 2. Less than... |
Row Conditions:                             | 3. Less or equal to... |
> If JOB...                                 | 4. Greater than... |
*** END ***                                 | 5. Greater or equal to... |
                                             | 6. Between... |
                                             | 7. Starting with... |
-----+-----+-----+
| F1=Help F7=Backward F8=Forward |
| F12=Cancel |
+-----+-----+-----+

```

Figure 10. The Comparison Operators panel

On the **Comparison Operators** panel, create a row condition in which you see only rows where JOB is equal to CLERK. To create the condition, select **1. Is** and select **1. Equal to**. (These choices are already selected for you in this example.)

- 4. Press Enter.

The Equal To panel displays.

- 5. Type clerk on the first line of the Equal To panel, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
-----+-----+-----+
Tables:                                       | Row Conditions |
  Q.STAFF                                    +-----+
Columns:                                     | Comparison Operators |
  NAME                                       | Equal To |
  DEPT                                       | Type one or more values, column names, or expressions. |
  JOB                                        | . . ( CLERK ) |
  SALARY                                     | Or. . ( ) |
  COMM                                      | Or. . ( ) |
Row Condit |                                | Or. . ( ) |
> If JOB I |                                | Or. . ( ) |
*** END ** |                                | Or. . ( ) |
-----+-----+-----+
| F1=Help F4=List F5=Show Field F12=Cancel |
+-----+-----+-----+
| F1=Help F7=Backward F8=Forward |
| F12=Cancel |
+-----+-----+-----+

```

Figure 11. The Equal To panel

- 6. Press Enter.

The row condition you created is displayed in the echo area, and the Specify panel displays again. At this point, you can add more conditions to your query such as a sort order, but for this lesson, continue with only the row condition.

- 7. Press the Cancel function key to close the Specify panel.

The Prompted Query panel displays. Your query is displayed in the echo area, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED                                LINE 1

  Tables:
  _ Q.STAFF

  Columns:
  _ NAME
  _ DEPT
  _ JOB
  _ SALARY
  _ COMM

  Row Conditions:
  _ If JOB Is Equal To 'CLERK'

  *** END ***

1=Help      2=Run      3=End      4=Show SQL  5=Change    6=Specify
7=Backward  8=Forward  9=Form    10=Insert  11=Delete   12=Report
OK, CANCEL command executed successfully.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 12. QMF displays your query on the Prompted Query panel.

8. To run the query and display your data, press the Run function key.

The report shown in the following figure displays:

```

REPORT                                LINE 1      POS 1      79

  NAME      DEPT  JOB      SALARY      COMM
  -----
  JAMES      20   CLERK    13504.60    128.20
  NGAN       15   CLERK    12508.20    206.60
  NAUGHTON   38   CLERK    12954.75    180.00
  YAMAGUCHI  42   CLERK    10505.90    75.60
  KERMISCH  15   CLERK    12258.50    110.10
  ABRAHAMS   38   CLERK    12009.75    236.50
  SNEIDER    20   CLERK    14252.75    126.50
  SCOUTTEN   42   CLERK    11508.60    84.20
  LUNDQUIST  51   CLERK    13369.80    189.65
  WHEELER    51   CLERK    14460.00    513.30
  BURKE      66   CLERK    10988.00    55.50
  GAFNEY     84   CLERK    13030.50    188.00

1=Help      2=          3=End      4=Print     5=Chart     6=Query
7=Backward  8=Forward   9=Form    10=Left     11=Right    12=

```

Figure 13. QMF displays the data from your query.

## Lesson three: Customizing a report

A report is a display of data that is formatted to make it easy to read or view. In this lesson, you learn how to change column headings and column widths and add a page heading to a report.

### About this task

When you run a query, QMF uses a default report format to display the report. You can change the appearance of your report by changing the default report format.

### Procedure

1. From the report panel, press the Form function key to display the default report form.

The FORM.MAIN panel displays, as shown in the following figure:



```

FORM.MAIN

COLUMNS:          Total Width of Report Columns: 50
NUM COLUMN HEADING  USAGE  INDENT  WIDTH  EDIT  SEQ
-----
 1 NAME              2      9      C      1
 2 DEPT              2      6      L      2
 3 JOB               2      5      C      3
 4 SALARY            2     10     L2     4
 5 COMM              2     10     L2     5

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=Check    3=End      4=Show    5=Chart    6=Query
7=Backward 8=Forward  9=         10=Insert 11=Delete 12=Report
OK, FORM is displayed.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 14. The FORM.MAIN panel

2. Change some column headings.

When you use the default report form, QMF assigns a heading to each column on the report. The heading defaults to the database label for the column. You can change this default behavior by setting the DSQDC\_COL\_LABELS variable.

To change a column heading, position the cursor at the beginning of the column heading you want to change and type the new heading over the existing heading. To split the heading on two lines, type an underscore wherever you want to start a new line. For this example, change the following headings:

- Type EMPLOYEE\_NAME over the heading NAME.
- Type DEPT . \_NUMBER over DEPT.
- Type COMMISSIONS over COMM.

3. Change the column width of the COMMISSIONS column so the title fits. Move the cursor to the column width for COMMISSIONS and type 11.

4. Specify the text that you want to display at the top of each page of your report.

Move the cursor to **PAGE: HEADING ==>** and type:

```
TOTAL COMMISSIONS - CLERKS
```

The FORM.MAIN panel should look like the panel that is shown in the following figure:

```

FORM.MAIN                                MODIFIED
COLUMNS:                               Total Width of Report Columns: 51
NUM COLUMN HEADING                       USAGE      INDENT  WIDTH  EDIT  SEQ
-----
 1 EMPLOYEE_NAME                         2         9      C      1
 2 DEPT._NUMBER                          2         6      L      2
 3 JOB                                    2         5      C      3
 4 SALARY                                  2        10     L2     4
 5 COMMISSIONS                            2        11     L2     5

PAGE:   HEADING  ==> TOTAL COMMISSIONS - CLERKS
        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=Check    3=End        4=Show    5=Chart    6=Query
7=Backward 8=Forward  9=          10=Insert 11=Delete 12=Report
OK, cursor positioned.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 15. The FORM.MAIN panel shows your report formatting changes.

5. Press the Report function key to see the changed report, which is shown in the following figure:

```

REPORT                                LINE 1      POS 1      79
                                TOTAL COMMISSIONS - CLERKS

EMPLOYEE   DEPT.   JOB      SALARY    COMMISSION
NAME       NUMBER
-----
JAMES      20    CLERK    13504.60   128.20
NGAN       15    CLERK    12508.20   206.60
NAUGHTON   38    CLERK    12954.75   180.00
YAMAGUCHI  42    CLERK    10505.90    75.60
KERMISCH  15    CLERK    12258.50   110.10
ABRAHAMS   38    CLERK    12009.75   236.50
SNEIDER    20    CLERK    14252.75   126.50
SCOUTTEN   42    CLERK    11508.60    84.20
LUNDQUIST  51    CLERK    13369.80   189.65
WHEELER    51    CLERK    14460.00   513.30
BURKE      66    CLERK    10988.00    55.50
GAFNEY     84    CLERK    13030.50   188.00

1=Help    2=          3=End    4=Print    5=Chart    6=Query
7=Backward 8=Forward  9=Form   10=Left    11=Right   12=
OK, REPORT is displayed.

```

Figure 16. The report reflects the changes you made.

6. Press the End function key to return to the QMF home panel.

**Related concepts**

Customizing your reports

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

Displaying your report as a chart

You can display your tabular data in a wide variety of charts. You can change QMF chart formats, or create new chart formats.

**Related reference**

Global variables that control various displays DSQDC global variables control the display of certain kinds of information. All of these global variables can be modified by the SET GLOBAL command.

## Chapter 3. Displaying a list of database objects

To quickly view information about database objects, you can display a list of those objects.

You can filter the objects in the list in many different ways. For example, you can limit objects to a certain type, such as queries, or to objects with names that contain a certain sequence of characters (for example, all objects with names beginning with ST).

By default, QMF allows you to see only objects that are authorized for use by your user ID. Depending on the list functions that your administrator has installed, objects authorized to any secondary or group IDs that you use might also be displayed in the list. These objects can include objects you saved in the database and objects that other users share with you.

If your administrator has configured QMF to allow you to use three-part names for tables and views in QMF commands, you can also display a list of tables and views that are located at a remote location.

### Related concepts

[Three-part names in QMF commands to access data in a remote database](#)

You can use QMF to access tables and views that are stored in a remote database by using a three-part name to refer to the table or view. By using three-part names, you can access and manipulate the data that you need without first having to use the QMF CONNECT command to connect to the remote database.

## Displaying a list of database objects by using the List key

You can use the List key to display database objects. You cannot list REPORT or CHART objects because they are not saved in the database. Only the queries, or data and forms to produce them, are saved.

### Procedure

To display a list of database objects by using the List function key:

1. On the QMF home panel, type the object type (QUERIES, PROCS, FORMS, FOLDERS, ANALYTICS, QMF, TABLES, or ALL) on the QMF command line.
2. Press the List function key. The list for the object type you specified displays.

For example, to display a list of all the queries you own, type the following on the QMF command line and press the List function key:

```
QUERIES (OWNER=userid)
```

where *userid* is your QMF user ID. The default for the OWNER parameter of the LIST command (issued when you press the List key) is taken from the DSQEC\_LIST\_OWNER global variable. The variable has a blank value by default, which causes QMF to display objects that are owned by the ID of the current QMF user if the OWNER parameter is not specified with the command.

When QMF runs the list command, a list that is similar to the one shown in the following figure is displayed.

```
+-----+
|                                     Query List                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Action  Name           Owner      Modified   Last Used   |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          MYQUERY      USERID    1997-04-24 1997-04-24   |
|          MYQUERY2     USERID    1997-05-01 1997-05-01   |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| F1=Help  F4=Command  F5=Describe F6=Refresh  F7=Backward F8=Forward  |
| F9=Clear F10=Comments F11=Sort   F12=Cancel  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 to 2 of 2
```

Figure 17. Displaying a list of query objects

The name of the list depends on the type of object displayed. The list panel that is shown in the previous figure is titled "Query List" because it contains only queries. A list that contains more than one type of object is called an "Object List".

If the DSQEC\_CURR\_FOLDER global variable is set to a folder name when the List key is pressed, only the specified QMF objects that exist in that folder are listed. If DSQEC\_CURR\_FOLDER is not set, you can specify the FOLDER keyword to look for objects in a specific folder.

You can also use the FOLDER keyword to override the current DSQEC\_CURR\_FOLDER setting. For example, to display a list of all QMF objects in a folder named SALES, type the following command and press the List function key:

```
ALL (FOLDER=SALES
```

You can type QMF commands in the Action area. Press the Comments function key to see a descriptive comment line for each object on the list. Press the Describe function key if the comment for that object is too long to display on the screen, or if you need more detailed information about an object. Press the Cancel function key to remove the list.

### Related reference

Global variables that control how commands and procedures are executed DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## Displaying a list from a command prompt by using by using the List key

You can display a list of database objects from a command prompt panel for any field with a "+" sign next to it.

### Procedure

To display a list of database objects from a command prompt panel:

1. Move the cursor to the field with the "+" sign after it.

For example, the **Name** field on the RUN Command Prompt panel has a "+" sign next to it; you can display a list of object names for this command. To see the RUN Command Prompt panel, enter the following on the QMF command line:

```
RUN ?
```

The RUN Command Prompt panel displays, as shown in the following figure:

```
+-----+
|                                     RUN Command Prompt                                     |
|                                     1 to 13 of 13                                     |
| Type (                               )                                             |
| Name (                               ) +                                           |
| .... (                               ) +                                           |
| .... (                               ) +                                           |
| .... (                               ) +                                           |
| .... (                               ) +                                           |
| .... (                               ) +                                           |
| To run an object from temporary storage, enter its type:                          |
| QUERY or PROC.                                                                    |
|                                                                                       |
| To run an object from the database, enter its name (and                           |
| optionally its type). Type can be QUERY or PROC.                                 |
+-----+
| F1=Help F3=End F4=List F7=Backward F8=Forward                                     |
+-----+
```

Figure 18. You can list objects from a command prompt panel.

You can also type information in the entry fields to choose the type of objects listed or to choose an owner whose objects you want to list. For example, you can type QUERY in the Type field to see only query objects.

2. Press the List function key, which displays a Query List panel similar to the one shown in the following figure:

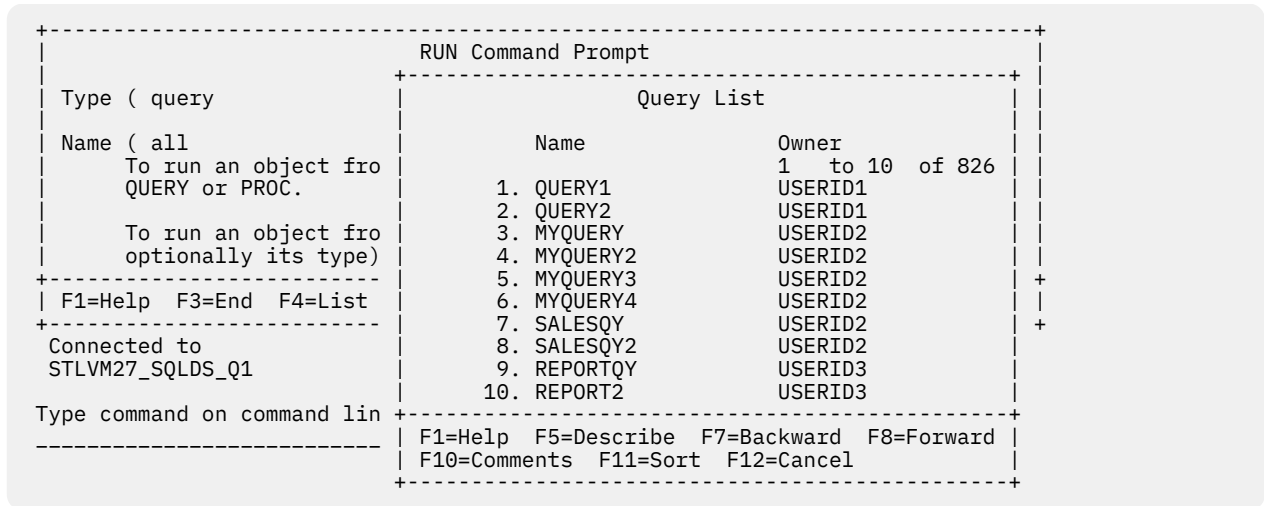


Figure 19. QMF displays a list of query objects.

The list displayed shows only the objects that are valid for the command you are issuing. For example, the valid objects for the RUN command are QUERY and PROC, so only those objects are displayed on any list you request.

3. To select an object from the list, enter its number.

## Results

The command prompt panel displays again. The object that you selected displays in the appropriate field.

## Displaying a list of database objects by using the LIST command

You can choose the objects to display with the LIST command.

The following table shows the LIST commands to display specific objects in your list.

Objects you want in the list	What you enter	Comments
Objects that you own	LIST ALL	Displays all the objects that you own: tables, views, aliases, history tables, queries, procedures, forms, folders, and analytics objects.
Objects of a specific object type that you own	LIST <i>objecttype</i>	For example, enter the following to display a list of all the queries you saved in the database:  LIST QUERIES

Table 5. Choosing specific objects to display with the LIST command. (continued)

Objects you want in the list	What you enter	Comments
Objects another user owns and shares with you	LIST <i>objecttype</i> (OWNER= <i>userid</i> )	<p>For example, enter the following to display a list of all the tables a person with user ID KRISTI shared with you:</p> <pre data-bbox="967 386 1471 432">LIST TABLES (OWNER=KRISTI)</pre> <p>The default for the OWNER parameter of the LIST command is taken from the DSQEC_LIST_OWNER global variable. The variable has a blank value by default, which causes QMF to display objects that are owned by the ID of the current QMF user.</p>
Objects at a remote location	LIST <i>objecttype</i> (LOCATION= <i>location</i> )	<p>If you are connected to a Db2 for z/OS database, you can display a list of tables and views at a remote location. For example, issue the following command to display a list of all tables in a database named NEWYORK:</p> <pre data-bbox="967 884 1471 930">LIST TABLES (LOCATION=NEWYORK)</pre> <p>If you are using three-part names in QMF commands to refer to tables or views in a remote database, your administrator should start QMF without the multirow fetch feature if the command is directed to any database other than Db2 for z/OS; otherwise the command fails. QMF commands with three-part names cannot be directed to DB2 for VSE and VM databases. By default, three-part names cannot be used to access remote tables that contain LOB data. However, you can set the DSQEC_LOB_RETRV global variable to 2 or 3 to access LOB metadata or data with a three-part name. Or, you can use the CONNECT command to connect to the database, and then run the query to access the remote table.</p> <p>If the database you are working with does not support three-part names, you can use the CONNECT command to connect to a remote database. Then, you can issue the LIST command to see tables and other QMF objects that are stored in the remote database.</p> <p>You cannot connect to or access data at a remote database if you started QMF as a stored procedure.</p>

Table 5. Choosing specific objects to display with the LIST command. (continued)

Objects you want in the list	What you enter	Comments
All procedures in a folder named SALES	LIST PROCS (FOLDER=SALES	Displays all procedures that you own in the SALES folder.  <b>Tip:</b> You can get the same LIST results by setting the DSQEC_CURR_FOLDER global variable to SALES and running the LIST command without the FOLDER keyword. If the DSQEC_CURR_FOLDER global variable is set to a different folder name, the folder name specified with the FOLDER keyword overrides that setting.

If you use the LIST command in a procedure and you need to continue an authorization ID or an object name on a subsequent line, you must use single quotation marks around the fully qualified name.

If the DSQEC\_CURR\_FOLDER global variable is set to a folder name when the LIST command is executed, only the QMF objects that meet the LIST criteria that exist in that folder are listed. If DSQEC\_CURR\_FOLDER is not set, all query objects are listed.

You can list query objects in a specific folder by specifying the FOLDER keyword. You can also use the FOLDER keyword to override the current DSQEC\_CURR\_FOLDER.

#### Related tasks

[Accessing data at a remote database](#)

When you start QMF, the system from which you start the program is known as the *local* system. The Db2 database that resides on this system is known as the *local* database. Databases that do not reside on this system are known as *remote* databases, or remote servers.

#### Related reference

[Global variables that control how commands and procedures are executed](#) DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## Choosing specific objects using selection symbols

You can select specific objects or values to display in your list by using wild cards or selection symbols.

You can use selection symbols in both the commands you enter on the QMF command line and on command prompt panels. The following table shows the two kinds of selection symbols that QMF recognizes.

Table 6. QMF's selection symbols		
Selection symbol	Represents	Examples
Underscore (_)	One character	To display a list of queries whose names start with REPORT but have an unknown character in the seventh position, enter:  LIST QUERIES (NAME=REPORT_

Selection symbol	Represents	Examples
Percent sign (%)	Zero or more characters	To display queries that are owned by users whose user IDs begin with MC, enter:  LIST QUERIES (OWNER=MC%)

## Entering commands on the list of database objects

You can enter QMF commands and parameters in the Action area of the List panel for any object in the list. The command performs an action for that specific object.

Be careful to keep the entries in the list aligned. For example, if you press the Backspace key to correct a typing error, the entire line may shift to the left. Enter another space to keep the entries aligned in columns, or you will get an error message when you issue the command. If the command you are entering is longer than the Action area, you can continue typing the command across the entire width of the list panel, as shown in the following figure.

```

+-----+
|                                     |
|                               Object List                               |
|                                     |
| Action   Name                     Owner   Type   Modified   Last Used   |
|-----|-----|-----|-----|-----|-----|
|          PN12345                   CLERK1  QUERY  1994-04-14  1994-04-14  |
|          INVENTORY                 CLERK1  QUERY  1994-04-14  1994-04-14  |
| EXPORT / TO MONTHLY SALESFIGMGR1234  QUERY  1994-03-21  1994-07-07  |
|          MGRS                      SMITH   QUERY  1994-03-21  1994-07-07  |
|          PANID                     SYSADMIN PROC   1994-03-14  1994-07-11  |
|-----|-----|-----|-----|-----|-----|
| F1=Help  F4=Command  F5=Describe  F6=Refresh  F7=Backward  F8=Forward  |
| F9=Clear  F10=Comments  F11=Sort  F12=Cancel  |
|-----|-----|-----|-----|-----|-----|

```

Figure 20. You can enter commands directly on the object list.

You can enter more than one command on the list. QMF runs commands by starting at the top of the list and moving to the bottom. Type an equal sign (=) to repeat a command for more than one object. Press the Clear function key to erase all the commands you typed on the list.

## Commands you can use on the list of database objects

You can issue specific commands in the Action area of a list. Type QMF before any command to make sure that the QMF command runs instead of a command synonym.

Your site might have defined a command with the same name as a QMF command. Depending on the command you use, there might be some restrictions or special considerations you need to know about.

### CONVERT

Converts a prompted, QBE, or SQL query into an equivalent SQL query; the comments in the original query do not appear in the converted query.

### DISPLAY

Retrieves an object from the database and displays it.

### EDIT

Edits a table in the database using the Table Editor.

The Table Editor operates on all tables except those that contain columns defined as BINARY, VARBINARY, BLOB, or XML. If a table contains CLOB or DBCLOB columns, you cannot edit those columns, but you can edit other columns in the table. To edit a table that contains DECFLOAT data, the processor on which QMF is running must support decimal floating point instructions.



If you want to edit a query or procedure, you must display it first.

**ERASE**

Deletes an object from the database.

**EXIT**

Ends your QMF session.

**EXPORT**

Exports QMF objects that are stored in the database directly from the database to a UNIX file, TSO data set, or CICS queue.

**IMPORT**

Imports QMF objects directly from a UNIX file, TSO data set, or CICS queue.

**LAYOUT**

Displays the format of a report that is produced from a given form without using any data.

You can use LAYOUT only with form objects, and only in an environment in which both REXX and ISPF are available.

**PRINT**

Prints a database object.

**RUN**

Runs a query or procedure that is stored in the database.

**SAVE**

Replaces the object in the database with the object currently in temporary storage. For example, if you enter the following next to a query on the database object list, QMF replaces the query in the database with the query currently in temporary storage:

```
SAVE QUERY AS objectname
```

**Related reference**

[QMF commands](#) Look up syntax, option descriptions, and usage information for commands used with QMF.

## Using a placeholder on the list of database objects

You can use a slash (/) as a placeholder to represent the object type, owner, and name in a QMF command.

For example, the first command shown below, when entered in the Action area of an object list, is equivalent to entering the second command shown below on the QMF command line:

```
EDIT / (MODE=ADD  
EDIT TABLE owner.tablename (MODE=ADD
```

In the second command shown above, *owner.tablename* is the owner and name of the table listed.

You can also use /T if you just want to specify the object type, or /N if you just want to specify the owner and name. The /T and /N placeholders are especially useful if you are issuing a command to run a user-written application that requires just the object type or just the object owner and name.

If you are displaying a list from a remote location, the placeholder symbols (/ and /N) include the location with the owner and name.

When you use a list placeholder, put a blank space before and after the placeholder token (' / ', ' /N ', ' /T ') so that QMF recognizes it as a placeholder; otherwise QMF considers it a path name.

You can also display the prompt panel for a command with the object type, object owner, and name automatically generated. To do this, type the command followed by the /placeholder and a question mark.

For example, to display the RUN Command Prompt panel for the DEPTQUERY object, enter RUN / ? in the Action area next to the object. The RUN QUERY Command Prompt panel displays. The first panel already has the object name and owner automatically generated. The next panel requests a form name to use in formatting the data from the query.

## Correcting errors when you enter an incorrect command

QMF runs the commands you issue on the list of database objects from top to bottom. If QMF finds an incorrect command, it stops, displays an error message, and highlights the line that contains the error.

### About this task

Commands that run successfully have an asterisk (\*), followed by up to seven letters of the command displayed in the Action area. If the objects on the list change as a result of running the commands, press the Refresh function key to display the changed list.

### Procedure

To correct an error:

1. Read the error message to see why the error occurred. If you need additional help, press the Help function key to get an explanation of the error message.
2. Press the spacebar to enter blanks over the incorrect command. If you want to delete all the commands on the list, press the Clear function key.
3. Type the correct command in the Action area, and press Enter to begin issuing commands again.

## Returning to the list from another QMF panel

When you issue commands from the list, another panel displays. For example, if you issue the RUN command for a query, QMF displays a report panel. If you issue the EDIT command for a table, QMF displays a Table Editor panel.

### Procedure

To return to the database object list from any panel, press the End function key on that panel.

# Chapter 4. Viewing the data in the database using Prompted Query

You can use Prompted Query to select and view data in the database. If you are new to QMF, or if you only use QMF occasionally, Prompted Query is an easy way to retrieve your data.

Prompted Query prompts you step-by-step through building a query; you do not need to know a query language.

### Related concepts

#### [QMF in three quick lessons](#)

Working with QMF usually requires three basic tasks: finding data you need, selecting specific items from that data, and turning the data into a report. This topic gives you the following three quick lessons on how to do these tasks using the data in the QMF sample tables.

## Prompted Query panels

The Prompted Query panel displays the information that you specify as you create the query.

The following figure shows the Prompted Query panel while a query is being built. Each area marked by a number in the figure is described in the list following the figure.

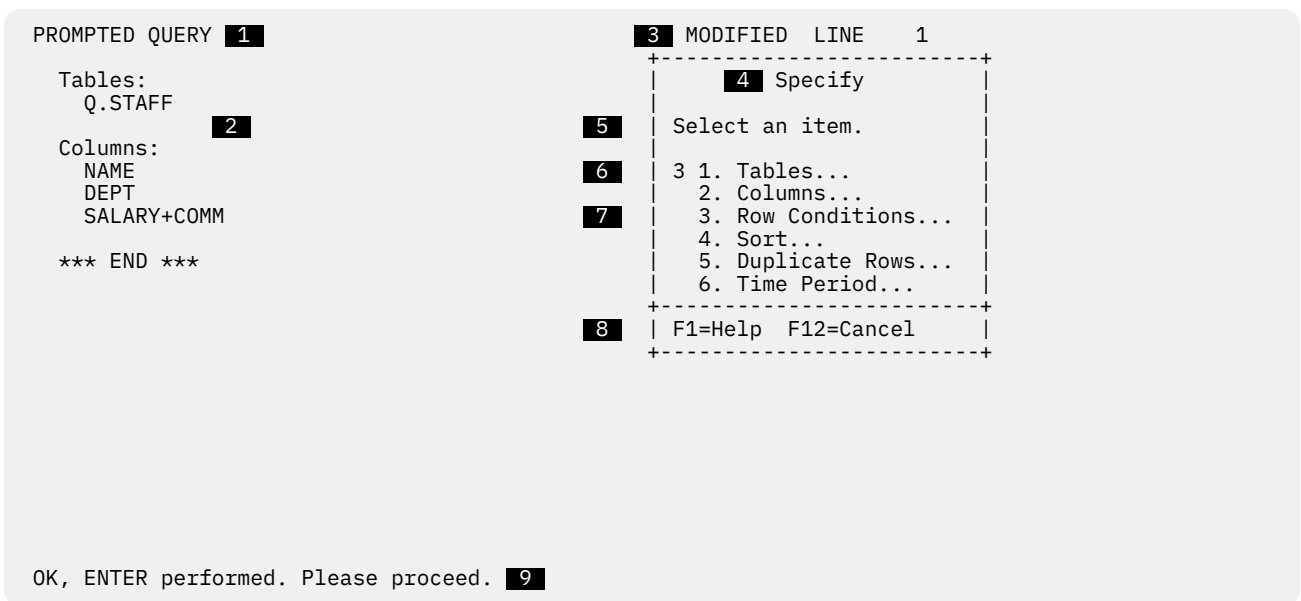


Figure 21. Building a prompted query

- 1** The name of the main panel. When you are using Prompted Query, this always says "PROMPTED QUERY".
- 2** The echo area. All of the information that you specify in dialog panels is shown here.
- 3** This area shows whether the screen you are looking at was changed in some way. If you have not made any changes, the word "MODIFIED" does not appear. The panel shows a line number that indicates the first line of the query that appears on the panel. When you scroll forward, the line number changes.
- 4** The Specify panel is a dialog panel. Use dialog panels to specify different kinds of information.

- 5** This field shows instructions that tell you what kind of information you can specify on this dialog panel, and how you can specify it.
- 6** This is the area where you specify your choice. Different types of dialog panels have different ways to specify items. This dialog panel shows that your next item is 3. Row Conditions.
- 7** This field shows the list of items you can choose from. You might have to scroll forward to see all of the items on a list.
- 8** This field shows the active function keys. Use the function keys to issue certain commands, such as the HELP command.
- 9** This is the message line. Whenever you issue a command in QMF, the message line indicates whether your command was successful.

The following figure shows the main Prompted Query panel after the query is built.

```

PROMPTED QUERY 1          QMFUSER.MYQUERY 2          MODIFIED LINE 1 3

  Tables:
  _ Q.STAFF

  Columns:
  _ NAME
  _ DEPT
  _ SALARY+COMM 4

  Row Conditions:
  _ If JOB Is Equal To 'CLERK'

  Sort:
  _ Ascending by DEPT

  *** END *** 5

1=Help      2=Run      3=End      4=Show SQL  5=        6=Specify 6
7=Backward  8=Forward   9=Form    10=Insert  11=Delete 12=Report
OK, CANCEL command executed successfully.
COMMAND ==> 7          SCROLL ==> PAGE 8

```

Figure 22. The built query

- 1** The name of the panel.
- 2** The name of a saved query.  
If you save a prompted query in the database, it has a two-part name: the user ID of the person who saved the query, and the name you gave the query when you saved it, separated by a period.
- 3** This shows that the panel was changed.
- 4** This is the echo area; it shows the completed query that was built using the dialog panels.
- 5** The word END means that there is no more information to look at; if END does not appear, you can scroll forward to look at the rest of your query.
- 6** This field shows the active function keys.

Commonly used commands are assigned to the function keys. Instead of typing a command, you can just press the numbered function key with that label. Your function keys might have different commands assigned to them than the ones that are shown in the figure. The functions of the keys depend on the assignments your administrator makes.

**7**

This is the command line; whenever you see `COMMAND ==>`, you can enter a QMF command.

**8**

This is the scroll indicator; it shows how many lines QMF scrolls when you enter a `FORWARD` or `BACKWARD` command.

## Rules for using Prompted Query

---

You must follow certain rules when you use QMF Prompted Query.

- Your QMF profile must have the language operand set to `PROMPTED`; otherwise, each time you start a query, you must enter `RESET QUERY (LANGUAGE=PROMPTED)`.
- You must type all data in upper case, unless you set the `CASE` operand in your QMF profile to `UPPER`. A `MIXED` value for the `CASE` parameter may have implications in TSO for the QMF `CONNECT` command.
- When entering data, if a character string contains a special character, you must enclose the character string in quotation marks.
- You must put quotes around all graphic data that you enter.

### Related reference

**Naming conventions** Ensure that names of your objects adhere to the naming conventions for QMF.

**SET PROFILE** The `SET PROFILE` command changes values in your QMF profile. These values influence the behavior of your QMF session.

## Starting Prompted Query and selecting tables

---

You use a `RESET QUERY` command to start Prompted Query. You use the Tables panel to select tables and columns.

### Procedure

To start Prompted Query and select tables, perform the following steps:

1. On the QMF command line, enter one of these commands:

- If the Language field in your QMF user profile is `PROMPTED`, enter

```
RESET QUERY
```

- If the Language field is not `PROMPTED` and you do not want to change your QMF profile, enter the following each time you want to start Prompted Query:

```
RESET QUERY (LANGUAGE=PROMPTED)
```

Prompted Query starts and displays the **Tables** panel for you to select tables.

2. Enter the name of your table on the Tables panel. You can specify up to 15 tables.
  - To find the name of a table, you can list available tables by pressing the **List** function key on the **Tables** panel.
  - You can specify search criteria to filter the list so it is smaller. For example, you can type `Q.S%` on the first line of the **Tables dialog** panel, and then press the **List** function key. QMF lists all the tables with names that start with "Q.S". The % sign represents a string of any length, containing any characters. The **Table List** panel displays, as shown in the following figure.

```

PROMPTED QUERY                                     LINE    1
Tables:
> ...
*** END ***
+-----+-----+-----+-----+
|                                     Tables                                     |
+-----+-----+-----+-----+
|                                     Table List                                     |
| Name                               Owner                                     |
|                                     1 to 5 of 5                                     |
| SALES                               Q                                       |
| SALG                                Q                                       |
| SAMTALE                             Q                                       |
| STAFF                               Q                                       |
| SUPPLIER                             Q                                       |
+-----+-----+-----+-----+
| F1=Help  F5=Describe  F7=Backward  F8=Forward |
| F10=Comments  F11=Sort  F12=Cancel |
+-----+-----+-----+-----+

```

OK, LIST command executed successfully.

Figure 23. The Table List panel

To see a descriptive comment for each object in the list, press the **Comments** function key. Press the **Describe** function key to see more detailed information about an object, such as the object type. Press the **Sort** function key to sort by name, type, or date. Press the **Cancel** function key to remove the list.

The name of the table is displayed in the echo area, and the **Specify** panel displays.

## What to do next

Select the columns to appear on your report.

### Related tasks

[Setting up and changing your QMF user profile](#)

Your QMF user profile tells QMF how you want to display information and process commands, and what printer to use when you print reports or charts.

## Column selection with Prompted Query

With Prompted Query, you can select columns to include in a report. You can also create columns that show the results of expressions or that calculate or count information in a group of rows.

### Selecting columns with Prompted Query

In Prompted Query, after you select a table, you choose the columns that you want to display in your report.

#### Procedure

To select columns from a table, perform the following steps:

1. Select a table, either by typing the name on the **Tables** panel or by selecting the table from the **Table List** panel.
2. Select the columns that you want to appear on your report from the **Specify** panel:
  - To include all columns, remove choice 2. Prompted Query selects all columns and the word ALL appears under the Columns heading in the echo area of the **Prompted Query** panel. You can skip the remaining steps.
  - To select columns, specify choice 2, **Columns** and press Enter.

The **Columns** panel displays, as shown in the following figure:

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
> ...
*** END ***

+-----+
|          Columns          |
| Select one or more columns. You can also |
| select either an expression or function. |
|                                     1 to 8 of 8 |
| Q.STAFF -- all              |
| ID                          |
| NAME                        |
| DEPT                        |
| JOB                          |
| "YEARS"                     |
| SALARY                      |
| COMM                        |
|                               |
| 1. Expression (A+B, etc.)... |
| 2. Summary Functions (SUM, etc.)... |
+-----+
| F1=Help  F5=Describe  F7=Backward |
| F8=Forward  F12=Cancel |
+-----+

```

Figure 24. The **Columns** panel

- 3. Choose the columns that you want to display in your report by typing any character in the space before the column name.

You can use the Tab key to move past columns you do not want. Press the **Forward** function key to display more columns.

To see information about the columns before you select them, tab to the column name and press the **Describe** function key. The **Column Description** panel displays. If you select columns that contain binary, XML, or large object (LOB) data types, QMF by default displays on the resulting report only the data type and length, or metadata, for those columns instead of the actual data. This is also the case if the table contains columns with decimal-floating point data and the processor on which QMF is running does not support decimal floating-point instructions.

Depending on the data type, you can change the default edit code from M to another edit code to display the actual data. The ability to change the edit code for LOB data is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default. You cannot change the edit code for columns with decimal floating-point data.

- 4. Press Enter.

The columns that you select are shown in the echo area. The **Specify** panel displays, with choice 3, **Row Conditions**, selected, as shown in the following figure:

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
*** END ***

+-----+
|          Specify          |
| Select an item.          |
| 3 1. Tables...           |
| 2. Columns...           |
| 3. Row Conditions...    |
| 4. Sort...              |
| 5. Duplicate Rows...    |
| 6. Time Period...       |
+-----+
| F1=Help  F12=Cancel |
+-----+

```

Figure 25. QMF lists the columns you selected.

**Related concepts**

Edit codesAn edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

## Creating a column by defining expressions with Prompted Query

You can create a column in your report by defining an expression. An expression adds, subtracts, multiplies, or divides the values in two or more columns and places the results in a new column.

### About this task

In this example, you create a total earnings column that is the sum of each employee's salary and commission.

### Procedure

To create a column by using expressions:

1. Select the table, either by typing the name on the **Tables** panel or by selecting the table from the Table List.
2. Select the columns from the **Columns** panel by typing any character beside each one.
3. At the bottom of the **Columns** panel, select choice 1, **Expression (A+B, etc.)**.
4. Press Enter.

The **Expression** panel displays, which is shown in the following figure.

```
PROMPTED QUERY                                MODIFIED LINE 1
+-----+-----+-----+-----+
Tables: |                                         | Columns
Q.STAF  +-----+-----+-----+-----+
Columns: |                                         | Expression
NAME    | Type an expression. You can use the following arithmetic |
DEPT    | operators: add(+), subtract(-), multiply(*) and divide(/). |
SALARY  | ( salary + comm                                         ) |
COMM    | (                                                         ) |
*** END | (                                                         ) |
        | (                                                         ) |
        | (                                                         ) |
+-----+-----+-----+-----+
| F1=Help F4=List F12=Cancel
+-----+-----+-----+-----+
| 2. Summary Functions (SUM, etc.)...
+-----+-----+-----+-----+
| F1=Help F5=Describe F7=Backward
| F8=Forward F12=Cancel
+-----+-----+-----+-----+

OK, ENTER performed. Please proceed.
```

Figure 26. Enter the expression you want to use.

5. Enter your expression.  
For example, enter:

```
SALARY + COMM
```

You can also create more complex expressions. For more information, press the Help function key on the **Expressions** panel. You can also press the List function key to see columns that you can use in your expression.

The names of the tables and columns you selected are displayed in the echo area. The new column that you created is also displayed under the heading **Columns**. The **Specify** panel displays again with choice 3, **Row Conditions**, selected, as shown in the following figure:



```

PROMPTED QUERY
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
  SALARY+COMM
*** END ***

```

```

MODIFIED LINE 1
+-----+
| Specify |
+-----+
| Select an item. |
+-----+
| 3 1. Tables... |
| 2. Columns... |
| 3. Row Conditions... |
| 4. Sort... |
| 5. Duplicate Rows... |
| 6. Time Period... |
+-----+
| F1=Help F12=Cancel |
+-----+

```

Figure 27. The new column, SALARY+COMM, appears in the echo area.

### Related reference

**Calculated results** You can use calculated values as part of a search condition. You can also display them for selected rows just as you display column values.

## Creating a column that summarizes a group of rows

You can create columns that summarize a group of rows. With the summary functions in QMF, you can calculate sums, averages, minimums, and maximums. You can also display a count of rows that meet certain criteria.

### Procedure

To specify a summary function:

1. On the **Columns** panel, select choice 2, **Summary Functions (SUM, etc.)**.
2. Press Enter.

The **Summary Functions** panel displays.

3. Select the functions that you want to perform by typing a character in the space before the items.

The **Summary Function Items** panel displays for any item you select except COUNT. The COUNT function does not need a column on which to act. In the echo area, COUNT is automatically assigned an asterisk, which indicates that it returns a count of the number of rows for each group.

4. Select the column on which you want the summary function to act.
5. Press Enter.

QMF displays an error message if your expression is over 255 characters after the database processes it. If you see this message, return to the panel and specify your summary function so that the expression is less than or equal to 255 characters.

The columns that you selected are shown in the echo area, and the **Specify** panel displays again.

## Row Selection with Prompted Query

You can use Prompted Query to select rows from the database to display on your report.

### Selecting rows with Prompted Query

Selecting rows limits, or creates a subset of, the data in a table. You select rows by creating a row condition.

### Procedure

To create a row condition:

1. On the **Specify** panel, select choice 3, **Row Conditions**.

The **Row Conditions** panel displays, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
Row Conditions:
> If...
*** END ***

```

Row Conditions

Begin a condition by selecting one column,  
or by entering an expression or function.

1 to 8 of 8

\* Q.STAFF

2. ID

3. NAME

4. DEPT

5. JOB

6. YEARS

7. SALARY

8. COMM

Expression (A+B, etc.)...

F1=Help F5=Describe F7=Backward  
F8=Forward F12=Cancel

Figure 28. The **Row Conditions** panel

To create a row condition, select a column that contains data from which you want to choose a limited selection. For example, you might want to see only the employees who are clerks. Select the rows where the value in the JOB column is CLERK. You can use any column in the table, even if you do not display it in the final report.

2. Enter the number of the column you want. The **Comparison Operators** panel displays, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
Row Conditions:
> If JOB...
*** END ***

```

Comparison Operators

Select a verb and a comparison.

Verb. . . . 1 1. Is

2. Is not

1 to 7 of 10

Comparison 1 1. Equal to...

2. Less than...

3. Less or equal to...

4. Greater than...

5. Greater or equal to...

6. Between...

7. Starting with...

F1=Help F7=Backward F8=Forward  
F12=Cancel

Figure 29. The **Comparison Operators** panel

3. Enter the number of the verb or comparison operator you want. You can scroll forward through the list of comparisons to find the one you want.

On the **Comparison Operators** panel, you complete the row condition by selecting the specific values that you want to see from the column you previously selected. In this example, you want to select only rows where JOB is equal to clerk.

Select the verb and comparison operator that define the relationship between the column and the values you want to see on your report. (QMF places an asterisk by comparison-operator choices that are not valid for the data type of the column you selected for the row condition.) For this example, you need the verb "Is" and the comparison operator "Equal to", which are already selected.

4. If the comparison operator you choose requires more information, a panel displays for you to enter a value to use to select the rows.

In this example, the **Equal To** panel displays, as shown in the following figure. You can specify more than one value on this panel. Enter the values on separate lines. If a character string contains a special character, such as a hyphen (-), enclose the character string in quotation marks.

For example, if you are looking for an employee with a hyphenated name, such as Smith-Wiggins, type the name on the **Equal To** panel with single quotation marks around it:

```
'Smith-Wiggins'
```

Entering the name in this way ensures that the database does not interpret the hyphen as a minus sign.

If your expression references a Db2 global variable, it must specify the Db2 global variable as a substitution variable.

```

PROMPTED QUERY                                MODIFIED LINE 1
-----+-----+
Tables:                                       |
Q.STAFF                                     |-----+
Row Conditions                               |-----+
Comparison Operators                         |-----+
Columns:                                     |-----+
NAME                                         |-----+
DEPT                                         |-----+
JOB                                           |-----+
SALARY                                       |-----+
COMM                                         |-----+
Equal To                                     |-----+
Type one or more values, column names, or expressions.
. . ( CLERK                                 )
Or. . (                                     )
Row Condit Or. . (                           )
> If JOB I Or. . (                           )
*** END ** Or. . (                           )
-----+-----+
F1=Help F4=List F5=Show Field F12=Cancel    |-----+
-----+-----+
F1=Help F7=Backward F8=Forward             |-----+
F12=Cancel                                  |-----+
-----+-----+

```

Figure 30. Enter the value to use in selecting rows.

The row condition you created is displayed in the echo area (as shown in the following figure), and the **Specify** panel displays. Notice that in the echo area, Prompted Query has put single quotations around CLERK because it consists of character data.

```

PROMPTED QUERY                                MODIFIED LINE 1
-----+-----+
Tables:                                       |
Q.STAFF                                     |-----+
Columns:                                     |-----+
NAME                                         |-----+
DEPT                                         |-----+
JOB                                           |-----+
SALARY                                       |-----+
COMM                                         |-----+
Row Conditions:                             |-----+
If JOB Is Equal To 'CLERK'                  |-----+
*** END ***                                  |-----+
-----+-----+
Specify                                      |-----+
Select an item.                              |-----+
1. Tables...                                |-----+
2. Columns...                               |-----+
3. Row Conditions...                        |-----+
4. Sort...                                  |-----+
5. Duplicate Rows...                       |-----+
6. Time Period...                          |-----+
-----+-----+
F1=Help F12=Cancel                          |-----+
-----+-----+

```

Figure 31. QMF displays the row condition you created.

### Related reference

Naming conventions Ensure that names of your objects adhere to the naming conventions for QMF.

## Narrowing row selection by using multiple row conditions

You can create more than one row condition to further filter the data you want to display on the report.

### About this task

The following example shows how to select only rows for employees who earn a commission and whose total earnings are greater than \$17,000.00. To select those rows, you need to create two row conditions: one to select only employees who earn a commission and another to select employees whose total earnings are greater than \$17,000.00. Only the rows that meet both conditions appear on the report.

### Procedure

To create multiple row conditions:

1. Select the table and the columns for your report.
2. Press Enter to select choice 3, **Row Conditions**, on the **Specify** panel.
3. On the **Row Conditions** panel, enter the column on which you want to base your first row condition. For this example, select the **COMM** column.
4. On the **Comparison Operators** panel, enter the verb and the comparison operator for the row condition. For this example, select the verb **Is not** and the comparison operator **NULL**.

This combination selects employees who earn any commission.

You created the first row condition. The row condition is displayed in the echo area, and the **Specify** panel displays with no choice selected.

5. Now create the second row condition. On the **Specify** panel, select choice 3, **Row Conditions**. The **Condition Connectors** panel, which is shown in the following figure, displays.
6. Enter 1 to select rows that meet either of the conditions, or enter 2 to select rows that meet both of the conditions. For this example, enter 2 to select rows that meet both conditions.

```
PROMPTED QUERY

Tables:
  Q.STAFF

Columns:
  NAME
  DEPT
  SALARY
  COMM
  SALARY+COMM

Row Conditions:
  If COMM is not NULL
  > ...

*** END ***

                               MODIFIED LINE 1
+-----+-----+-----+
|                               |
|                               |
|          Condition Connectors |
|                               |
|          Select a connector.  |
|                               |
|          1. Or (Either condition is true) |
|          2. And (Both conditions are true) |
|-----+-----+-----+
| F1=Help  F12=Cancel          |
+-----+-----+-----+
```

Figure 32. The **Condition Connectors** panel

The **Row Conditions** panel displays.

7. On the **Row Conditions** panel, enter the column on which you want to base your second row condition. For this example, select the choice **Expression (A+B, etc.)** at the bottom of the **Row Conditions** panel to create a **SALARY+COMM** column. You created this column in [“Creating a column by defining expressions with Prompted Query”](#) on page 36.
8. On the **Comparison Operators** panel, enter the verb and the comparison operator for the row condition. For this example, select **Is** and **Greater than**.

9. If a panel displays for the comparison operator you selected, enter the value you want to use to select the rows.  
For this example, enter 17000 without commas or quotes.

You created the second row condition. Both row conditions are displayed in the echo area, and the **Specify** panel (shown in the following figure) displays, with no choice selected.

```
PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  JOB
  SALARY
  COMM
Row conditions:
  If COMM Is Not NULL
  And SALARY+COMM Is Greater Than 17000
*** END ***
```

Specify

Select an item.

- 1. Tables...
- 2. Columns...
- 3. Row Conditions...
- 4. Sort...
- 5. Duplicate Rows...
- 6. Time Period...

F1=Help F12=Cancel

Figure 33. The query shows both row conditions you created.

## What to do next

Repeat these steps to create more row conditions. You can create as many row conditions as necessary to select the data that you want.

## Sorting the rows in a query

After you select the rows to display on your report, you can specify the way that you want to sort the rows on the report.

### About this task

In this example, you sort the rows in ascending order by department number.

### Procedure

1. On the **Specify** panel, select choice 4, **Sort**.

The **Sort** panel displays (as shown in the following figure) with choice 1, **Ascending** already selected for you.

```

PROMPTED QUERY      USERID.MYQUERY      MODIFIED  LINE
-----
Tables:
  Q.STAFF
Columns:
  NAME
  DEPT
  SALARY
  COMM
  SALARY+COMM
Row Conditions:
  If COMM Is Not NULL
  And SALARY+COMM Is Greater Than
Sort:
> ...
-----
Sort
Select the order for sorting and the
column you want to sort.
Order...
1 1. Ascending (A-Z, 0-9)
  2. Descending (9-0, Z-A)
                                     1 to 5 of 5
Columns...
1. SALARY+COMM
2. NAME
3. DEPT
4. SALARY
5. COMM
-----
F1=Help  F7=Backward  F8=Forward
-----

```

Figure 34. The **Sort** panel

2. Leave 1 selected if you want to sort the rows in ascending order, or type 2 to sort the rows in descending order.
3. Type the number of the column that contains the data you want to sort.
4. Press Enter.

The sort order that you specified is displayed in the echo area, and the **Specify** panel displays.

## What to do next

Repeat these steps if you want to sort on other columns. For example, in this query, in addition to sorting by department number, you might also want to sort by name within department.

QMF displays the sort order that you selected, as shown in the following figure.

```

PROMPTED QUERY      MODIFIED  LINE  1
-----
Tables:
  _  Q.STAFF
Columns:
  _  NAME
  _  DEPT
  _  SALARY
  _  COMM
  _  SALARY+COMM
Row Conditions:
  _  If COMM Is Not NULL
  _  And SALARY+COMM Is Greater Than 17000
Sort:
  _  Ascending by DEPT
  _  Ascending by NAME
1=Help      2=Run      3=End      4=Show SQL  5=Change      6=Specify
7=Backward  8=Forward  9=Form     10=Insert   11=Delete     12=Report
OK, CANCEL command executed successfully.
COMMAND ==>          SCROLL ==>> PAGE

```

Figure 35. QMF displays the order in which rows are sorted.

QMF displays the report in the following figure when you run this query. The rows are first sorted by department number, and then by employee name within each department.

NAME	DEPT	SALARY	COMM	COL1
ROTHMAN	15	16502.83	1152.00	17654.83
PERNAL	20	18171.25	612.45	18783.70
O'BRIEN	38	18006.00	846.55	18852.55
QUIGLEY	38	16808.30	650.25	17458.55
KOONITZ	42	18001.75	1386.70	19388.45
SMITH	51	17654.50	992.80	18647.30
WILLIAMS	51	19456.50	637.65	20094.15
GONZALES	66	16858.20	844.00	17702.20
GRAHAM	66	21000.00	200.30	21200.30
WILSON	66	18674.50	811.50	19486.00
EDWARDS	84	17844.00	1285.00	19129.00

Figure 36. The report is sorted by department number and employee name.

## Data display from multiple tables using Prompted Query

With QMF, you can display data from more than one table, up to 15 tables. Joining tables means linking them by common columns. When you join tables you might need to hide duplicate information.

### Displaying data from two tables

In this example, you join two tables by two common columns to get the data that you need.

#### About this task

Suppose that you need salary data for each department within each corporate division. This data is not in one table. For example, in the sample tables, department numbers are found in both the Q.ORG and Q.STAFF tables, division and department names are found only in the Q.ORG table, and salary data is found only in the Q.STAFF table.

In this example, the DEPT column in the Q.STAFF table and the DEPTNUMB column in the Q.ORG table both contain department numbers. You can join the tables by these two columns.

When you join the Q.STAFF and Q.ORG tables by department number, QMF joins every row in Q.STAFF to every row in Q.ORG that has the same department number. By joining the tables, you produce a report that shows the names of all departments.

#### Procedure

To display data from more than one table:

1. On separate lines on the **Tables** panel, enter the names of the tables from which you want to display data. For this example, type Q.STAFF on the first line and Q.ORG on the second line.

If the table name is too long to display in the **Tables** panel, it is split into two lines. If you see a ">" sign, use the Describe key to display the full table name. Examples of tables with long names are shown in the following figure.

```

Join Tables

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

Joined Tables          Tables
-----
1 1. DALLAS.MIRIAM_SMITH_AT_OFFFC1234.          * DALLAS.MIRIAM_SMITH_AT_OFFFC1234.
   OVERSEAS_PARTS_LIST_ASIALOC5678>            OVERSEAS_PARTS_LIST_ASIALOC5678>
2. DALLAS_SYSTEMS.JOHNJONES.
   DOMESTICPARTS
3. DOMESTIC_CUSTOMER_DATA>

F1=Help  F5=Describe  F7=Backward  F8=Forward  F12=Cancel

```

Figure 37. Examples of long table names

When you finish typing the names of the tables, press Enter. In the **Join Columns** panel, select the column or columns on which to join the tables. The columns from each table appear under separate headings, as shown in the following figure:

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:                                         +-----+
Q.ST +-----+ Tables +-----+
Q.OR |                                         |
      |                                         |
Join T | Select a column from each table. Rows that have equal
> ... | values in those columns will be joined.
      |
Column | Q.STAFF                                Q.ORG
ALL    |                                         |
      |                                         |
*** EN | 3 1. ID                                1 1. DEPTNUMB
      |     2. NAME                            2. DEPTNAME
      |     3. DEPT                            3. MANAGER
      |     4. JOB                             4. DIVISION
      |     5. YEARS                           5. LOCATION
      |     6. SALARY
      |     7. COMM
      |                                         |
      |-----+-----+
      | F1=Help F5=Describe F7=Backward F8=Forward F12=Cancel
      |-----+-----+

```

Figure 38. The **Join Columns** panel

2. Select the column that you want from the first table. For this example, type 3 to select the DEPT column from the Q.STAFF table.
3. Select the column that you want from the second table. For this example, type 1 to select the DEPTNUMB column from the Q.ORG table.
4. Press Enter.

The columns that you used to join the tables are displayed in the echo area. The **Specify** panel displays, with choice 2, **Columns**, selected for you, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables:                                         +-----+
Q.STAFF(A)                                     | Specify |
Q.ORG(B)                                       |         |
      |                                         |
Join Tables:                                  | Select an item. |
A.DEPT And B.DEPTNUMB                        |         |
      |                                         |
Columns:                                       | 2 1. Tables... |
ALL                                            | 2. Columns...  |
      |                                         | 3. Row Conditions... |
      |                                         | 4. Sort...     |
      |                                         | 5. Duplicate Rows... |
      |                                         | 6. Time Period... |
      |                                         |         |
*** END ***                                   | F1=Help F12=Cancel |
      |                                         |-----+-----+

```

Figure 39. QMF displays the names of the tables you selected to join.

5. Press Enter to begin selecting the columns that you want to display in your report. The **Columns** panel displays with the names of the columns from all the tables you selected.
6. Enter x beside each column you want to display on the report. For this example, select the **DEPTNAME** column and the **DIVISION** column from the Q.ORG table. The columns that you select to display on the report appear in the echo area. The **Specify** panel displays, with choice 3, **Row Conditions**, selected.
7. To get the total salary for each department, use the SUM function. On the **Specify** panel, select choice 2, **Columns**. The **Columns** panel displays again.
8. Select choice 2, **Summary Functions (SUM, etc.)** at the bottom of the panel. The **Summary Functions** panel displays.



9. Select the summary function. For this example, select **Sum of**. The **Summary Function Items** panel displays.
10. Select the column that you want the summary function to act on. For this example, select the **SALARY** column.
11. To finish the query, sort the rows by department name and division. On the **Specify** panel, select choice 4, **Sort**. The **Sort** panel displays.
12. For this example, select **Ascending** and the DIVISION column.

Repeat steps “11” on page 45 and “12” on page 45 to sort the DEPTNAME column in ascending order, then press the Cancel key to remove the **Specify** panel. The finished query is shown in the following figure:

```
PROMPTED QUERY                                MODIFIED LINE    1

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

Join Tables:
  A.DEPT And B.DEPTNUMB

Columns:
  DEPTNAME
  DIVISION
  SUM (SALARY)

Sort:
  Ascending by DIVISION
  Ascending by DEPTNAME

*** END ***
```

Figure 40. QMF displays the finished query.

13. Press the Run function key to see the final report that is shown in the following figure, which reflects the total salary data for each department within each division.

DEPTNAME	DIVISION	COL1
HEAD OFFICE	CORPORATE	83463.45
MID ATLANTIC	EASTERN	64286.10
NEW ENGLAND	EASTERN	61929.33
SOUTH ATLANTIC	EASTERN	77285.55
GREAT LAKES	MIDWEST	58369.05
PLAINS	MIDWEST	86090.80
MOUNTAIN	WESTERN	66147.00
PACIFIC	WESTERN	86076.20

Figure 41. The report shows salary data for departments within divisions.

**Related concepts**

[QMF sample tables](#)

## Eliminating duplicate rows from the report

In Prompted Query, you can request that rows with duplicate information not appear in your report.

### About this task

If you create a query that is based on the sample tables that shows all departments that have salespeople, the report will show duplicate rows for each department with more than one salesperson. Such a report is shown in the following figure:

```

DEPT  DIVISION
-----
15    EASTERN
20    EASTERN
38    EASTERN
38    EASTERN
42    MIDWEST
51    MIDWEST
51    MIDWEST
66    WESTERN
66    WESTERN
66    WESTERN
84    WESTERN
84    WESTERN

```

Figure 42. The report shows duplicate rows for several departments.

## Procedure

To eliminate duplicate rows:

1. On the **Specify** panel, select **Duplicate Rows**.

The **Duplicate Rows** panel displays, as shown in the following figure.

```

PROMPTED QUERY
Tables:
  Q.STAFF(A)
  Q.ORG(B)
Join Tables:
  A.DEPT And B.DEPTNUMB
Columns:
  DEPT
  DIVISION
Row Conditions:
  If JOB Is Equal To 'SALES'
Duplicate Rows:
> ...
*** END ***

```

MODIFIED LINE 1

Duplicate Rows

Select one of the following.

Keep. . 2 1. Duplicate rows.  
          2. Single copy of each row.

F1=Help F12=Cancel

Figure 43. The **Duplicate Rows** panel

2. Select **Single copy of each row**.

The **Prompted Query** panel displays with your query in the echo area. The **Specify** panel does not display because eliminating duplicate rows is the last step in creating a prompted query. Press the Specify key to display the **Specify** panel if you want to work on your query again.

When you run the query, QMF displays a report that shows one row for each department, as shown in the following figure.

```

DEPT  DIVISION
-----
15    EASTERN
20    EASTERN
38    EASTERN
42    MIDWEST
51    MIDWEST
66    WESTERN
84    WESTERN

```

Figure 44. The report shows only one row for each department.

## Specifying a time period

In Prompted Query, you can specify a period of time from which data is retrieved for your report. You can specify the time period as an application period (BUSINESS\_TIME) or a system period (SYSTEM\_TIME).

### About this task

You can specify a time period for a temporal table or for a view. However, if you specify a time period for a view that does not reference a temporal table, the time period specification is ignored.

### Procedure

To specify a time period:

1. On the Specify panel, select **Time Period**.
2. From the list of eligible tables and views, select a table or view for which you will specify a time period.
3. Enter the time period specification in one of the following formats, and then press Enter:
  - FOR BUSINESS\_TIME AS OF 'yyyy-mo-dd'
  - FOR BUSINESS\_TIME FROM 'yyyy-mo-dd' TO 'yyyy-mo-dd'
  - FOR BUSINESS\_TIME BETWEEN 'yyyy-mo-dd' AND 'yyyy-mo-dd'
  - FOR SYSTEM\_TIME AS OF 'yyyy-mo-dd-hh.mm.ss.nn'
  - FOR SYSTEM\_TIME FROM 'yyyy-mo-dd-hh.mm.ss.nn' TO 'yyyy-mo-dd-hh.mm.ss.nn'
  - FOR SYSTEM\_TIME BETWEEN 'yyyy-mo-dd-hh.mm.ss.nn' AND 'yyyy-mo-dd-hh.mm.ss.nn'

where:

- *yyyy* is a four-digit value that represents the year.
- *mo* is a two-digit value that represents the month.
- *dd* is a two-digit value that represents the day.
- *hh* is a two-digit value that represents the hour.
- *mm* is a two-digit value that represents the minutes.
- *ss* is a two-digit value that represents the seconds.
- *nn* is a 0- through 12-digit value that represents the number of fractional seconds.

## Joining multiple tables

In this example, you join three tables by columns that are common between two tables.

### About this task

This example uses the Q.SUPPLIER, Q.PARTS, and Q.PROJECT tables. You create a query that shows the supplier name, the part name, the project number, and the start date for all the parts that are used by each project in the Q.PROJECT table.

In the sample tables, the supplier name is in Q.SUPPLIER as COMPANY, the part name is in Q.PARTS as PARTNAME, and the project number is in Q.PROJECT as PROJNO. To get all the information for the report, you need to join these three tables.

The ACCTNO column in Q.SUPPLIER and the SUPPNO column in Q.PARTS contain the same information, so you can join those two tables. Likewise, the PRODNUM column in Q.PARTS and the PRODNO column in Q.PROJECT contain the same information, so you can also join those tables.

### Procedure

To join multiple tables:

1. On separate lines on the **Tables** panel, enter the names of the tables from which you want to display data. For this example, enter Q.SUPPLIER, Q.PARTS, and Q.PROJECT.

The **Join Tables** panel displays, as shown in the following figure. The first table that you selected appears under the Joined Tables heading. Select the second table that you want to join from the list under the Tables heading.

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables: |-----+-----+-----+-----+
|
|-----+-----+-----+-----+
|                               Join Tables
|
| Select two tables that you want to join; one from each group.
|
|   Joined Tables                Tables
|   -----                    -----
| 1 1. Q.SUPPLIER                2 * Q.SUPPLIER
|                               2. Q.PARTS
|                               3. Q.PROJECT
|
|-----+-----+-----+-----+
| F1=Help F5=Describe F7=Backward F8=Forward F12=Cancel
|-----+-----+-----+-----+

```

Figure 45. The **Join Tables** panel

2. From the Joined Tables heading, select the first table that you want to join. For this example, select the **Q.SUPPLIER** table.
3. From the Tables heading, select the second table that you want to join. For this example, select the **Q.PARTS** table.
4. Press Enter.

The **Join Columns** panel displays, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
Tables: |-----+-----+-----+-----+
|
|-----+-----+-----+-----+
|                               Join Tables
|
| Select t |-----+-----+-----+-----+
|                               Join Columns
|   Joined | Select a column from each table. Rows that have equal
|   -----| values in those columns will be joined.
| 1 1. Q.S | Q.SUPPLIER                Q.PARTS
|          |                               1 to 6 of 7
|          | 1. ACCTNO                    1. SUPPNO
|          | 2. COMPANY                    2. PARTNAME
| F1=Help | 3. STREET                          3. PRODUCT
|          | 4. CITY                        4. PRODNO
|          | 5. STATE                      5. PROJNO
|          | 6. ZIP
|
|-----+-----+-----+-----+
| F1=Help F5=Describe F7=Backward F8=Forward F12=Cancel
|-----+-----+-----+-----+

```

Figure 46. The **Join Columns** panel

5. From the columns that are listed for the first table, select the column on which you want to join the tables. For this example, select **ACCTNO**.
6. From the columns that are listed for the second table, select the column that contains the same kind of data. For this example, select **SUPPNO**.
7. Press Enter.

You joined the first two tables. The **Join Tables** panel displays again. This time, both tables you joined appear under the Joined Tables heading.

Now you can select the last table to join with the other two from the list under the Tables heading. For this example, select **Q.PARTS** from the list under Joined Tables; select **Q.PROJECT** from the list under Tables, as shown in the following figure:

```

PROMPTED QUERY                                MODIFIED LINE 1
-----
Tables:                                        Tables
-----+-----
Join Tables
Select two tables that you want to join; one from each group.
Joined Tables          Tables
-----              -----
2 1. Q.SUPPLIER       3 1. Q.SUPPLIER          1 to 3 of 3
 2. Q.PARTS           2. Q.PARTS
                    3. Q.PROJECT
-----+-----
F1=Help F5=Describe F7=Backward F8=Forward F12=Cancel
-----
  
```

Figure 47. The **Join Tables** panel displays the tables you already joined.

8. The **Join Columns** panel displays again. Select the columns that you want from each table. For this example, select **PRODNO** from the Q.PARTS table; select **PRODNUM** from the Q.PROJECT table, as shown in the following figure.

```

PROMPTED QUERY                                MODIFIED LINE 1
-----
Tables:                                        Tables
-----+-----
Join Tables
Join Columns
Select a column from each table. Rows that have equal values in those columns will be joined.
Joined Tables          Tables
-----              -----
1 1. Q.SUPPLIER       2 1. PROJNO
 2. Q.PARTS           2. PRODNUM
                    3. DEPT
                    4. STARTD
                    5. ENDD
                    6. TIMESTAMP
-----+-----
F1=Help
-----
F1=Help F5=Describe F7=Backward F8=Forward F12=Cancel
-----
  
```

Figure 48. QMF displays columns from the second set of tables.

9. Press Enter.

You joined the second and third tables.

If you have more tables to join, the **Join Tables** panel displays. If you joined all the tables, the **Specify** panel displays so you can select the columns to display on the report.

### Related concepts

[QMF sample tables](#)

## Joining multiple columns

In this example, you join two tables on two columns because joining on only one column might produce incorrect information.

### About this task

If you join the QMF sample tables Q.PARTS and Q.PROJECT on their PRODNO and PRODNUM columns alone, you might get a report that shows incorrect project numbers for the products. The sample tables have more than one project number for some products. Join the Q.PARTS and Q.PROJECT tables at both of the columns that share the same type of information. You must join the tables on the PROJNO columns as well as the PRODNO and PRODNUM columns.

### Procedure

To join multiple columns:

1. Create or display a query that joins two tables at a single column.  
For this example, display a query that joins the Q.PARTS and Q.PROJECT tables.
2. Press the Cancel function key to remove the **Specify** panel from the **Prompted Query** panel.
3. Move the cursor to the last pair of joined columns under the **Join Tables** heading.
4. Press the Insert function key. The **Join Tables** panel displays.
5. Select the tables that you want to join on an additional column.  
For this example, select the **Q.PARTS** and the **Q.PROJECT** tables again. The **Join Columns** panel displays.
6. Select the columns that you want to join.  
For this example, select the **PROJNO** columns from both tables.

The query displays with the additional columns you joined, as shown in the following figure.

```
PROMPTED QUERY                                MODIFIED    LINE    1
Tables:
-   Q.PARTS(B)
-   Q.PROJECT(C)
Join Tables:
-   A.ACCTNO And B.SUPPNO
-   And B.PRODNO And C.PRODNUM
-   And B.PROJNO And C.PROJNO
Columns:
-   PARTNAME
-   C.PROJNO
-   STARTD
*** END ***
1=Help      2=Run      3=End      4=Show SQL 5=Change   6=Specify
7=Backward 8=Forward  9=Form     10=Insert  11=Delete  12=Report
OK, ENTER performed. Please proceed.
COMMAND ==> SCROLL ==> PAGE
```

Figure 49. QMF shows that the two tables are joined at a second column.

## Making your prompted query reusable with substitution variables

When you specify substitution variables in a prompted query, you can use the same query to retrieve different information. You supply a new value for the variable each time you run the query.

The prompted query in the following figure selects department data. By using a substitution variable (&DEPARTMENT) for the department number in the row condition, you can specify a different department number each time you run the query.

```
PROMPTED QUERY                                MODIFIED LINE 1
Tables:
  Q.STAFF
Columns:
  ID
  NAME
  JOB
  SALARY
Row Conditions:
  If DEPT Is Equal To &DEPARTMENT;
```

Figure 50. This query uses a substitution variable for the department number.

You can enter substitution variables on any **Prompted Query** panel where you can enter expressions. Expressions that reference Db2 global variables must specify the Db2 global variables as substitution variables.

You can specify values for substitution variables in any of the following ways:

- As part of the RUN command
- From the RUN command prompt panel
- By setting a global variable

## Specifying a variable value as part of the RUN command

You can specify values for substitution variables when you issue the RUN QUERY command in a prompted query.

### Procedure

1. Type the value following the variable name and the "=" sign in the RUN QUERY command as in the following example:

```
RUN QUERY (&DEPARTMENT = 38
```

2. Enclose all character data, as well as the following special characters, in single quotes:

- Blank
- Comma
- Left or right parenthesis
- Single or double quotation mark
- Equal sign

### Example

For example, the following query has two variables. For the first variable, you specify one or more column names as the value; for the second, you specify text that contains a quotation mark.

```
SELECT &X
FROM Q.STAFF
WHERE NAME=&Y
```

If the text itself contains quotation marks, add another set of quotation marks for each quotation mark:

```
RUN QUERY (&X=SALARY, &Y='O' 'BRIEN'
```

If you want to select more than one column, you can provide multiple column names for a single variable value, as in the following example:

```
RUN QUERY (&X=(DEPT,NAME,SALARY)
```

## Specifying a value on the RUN Command Prompt panel

If your query contains a variable, and you do not specify a value for the variable when you type the RUN command, the **RUN Command Prompt** panel displays. The variables that need values appear on the panel.

### Procedure

Type the values for the variables, as shown in the following figure.

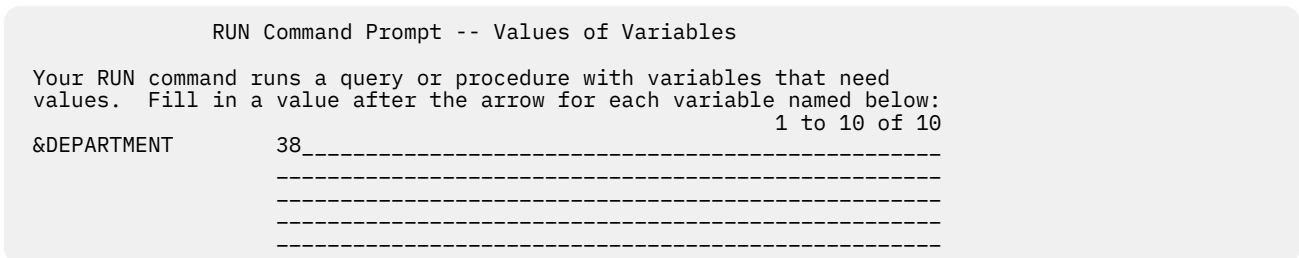


Figure 51. Specifying a value for a variable on the **Run Command Prompt** panel

## Specifying a value with a global variable

You can define global variables with the SET GLOBAL command.

For example, to set a global variable value for the &DEPARTMENT variable, on the QMF command line enter:

```
SET GLOBAL (DEPARTMENT=38
```

You can specify up to 10 variable/value pairs with a single SET GLOBAL command. Separate the values with either commas or blanks. Variable values that are used in queries cannot begin with dashes because the database misinterprets them.

By default, values for global variables persist for the duration of the QMF session or until you reset them. However, the DSQEC\_USERGLV\_SAV global variable can be set to save global variable values from one session to another.

### Related reference

**SET GLOBAL**The SET GLOBAL command assigns values to global variables from the QMF command line, from a procedure, or through the callable interface.

## Running a query and displaying a report

When you run a query, QMF displays the retrieved data as a report. The appearance of the report is controlled by choices you make on QMF form panels.

### About this task

You can run a query with only one table selected, and some or all columns selected.

Edit codes that are used on forms control how different data types are formatted. Because QMF applies different default edit codes to different data types, some of the columns on your report might display only the data type and length rather than the actual data in the column.

### Procedure

To run a query:

1. From the **Specify** panel, press the Cancel function key. The **Prompted Query** panel displays your query.
2. Press the Run function key.



You can also enter RUN QUERY on the QMF command line.

When QMF finishes running your query, it displays a report that shows all of the data that you selected, as shown in the following figure.

If your query selects many rows, you might need to scroll forward to see all the data.

NAME	DEPT	JOB	SALARY	COMM
JAMES	20	CLERK	13504.60	128.20
NGAN	15	CLERK	12508.20	206.60
NAUGHTON	38	CLERK	12954.75	180.00
YAMAGUCHI	42	CLERK	10505.90	75.60
KERMISCH	15	CLERK	12258.50	110.10
ABRAHAMS	38	CLERK	12009.75	236.50
SNEIDER	20	CLERK	14252.75	126.50
SCOUTTEN	42	CLERK	11508.60	84.20
LUNDQUIST	51	CLERK	13369.80	189.65
WHEELER	51	CLERK	14460.00	513.30
BURKE	66	CLERK	10988.00	55.50
GAFNEY	84	CLERK	13030.50	188.00

Figure 52. QMF displays your data as a report.

3. If you want to change the query, press the Query function key to return to the **Prompted Query** panel.

### Related concepts

#### Customizing your reports

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

**Edit codes**An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

## Saving a new query

You can save your query in the database after you create it. You can run a saved query and display the report again. You can also add, delete, or change the information in a saved query.

You save a prompted query like you save any other QUERY, FORM, or PROC object.

In some cases, a long report might not be complete when you attempt to save your query. When this happens, QMF cannot save the query until the report is complete. This could create performance problems. The global variable DSQEC\_RESET\_RPT allows you to predefine how you want QMF to handle this situation.

To avoid performance issues, you can use multiple threads by setting the DSQAO\_DSQSMTHD global variable.

After you save the query, the Prompted Query panel displays with the name you gave the query.

### Related tasks

#### Saving a QUERY, FORM, or PROC object

You can save a query, form, or procedure by entering the SAVE command on the QMF command line.

### Related reference

**Global variables that control how commands and procedures are executed**DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## Canceling a running query

You can cancel a query by interrupting the RUN QUERY command. Queries might also be canceled by the QMF governor.

### About this task

In addition to your canceling a query yourself, your site might have a governor interrupt routine. The interrupt routine can attempt to cancel a query that takes a long time to run or tries to retrieve too many rows.

If a **QMF governor prompt** panel similar to the one in the following figure displays while you are trying to run a query, follow the instructions on the panel to cancel the query or continue. If you continue, the QMF governor can still cancel your command.

```
DSQnnnnn QMF governor prompt:
          Command has executed for xxxxxxxx minutes
          and fetched yyyyyyy rows of data.

==> To continue QMF command
          press the "ENTER" key.
==> To cancel QMF command
          type "CANCEL" and then press the ENTER key.
==> To turn off prompting
          type "NONPROMPT" then press the ENTER key.
```

Figure 53. The **QMF governor prompt** panel

### Procedure

To cancel a query by using the QMF command interrupt, follow these steps:

1. While the **Database Status** panel displays, press the PA1 key if you are using CICS and the ATTN key if you are using TSO. (If you are using a terminal emulator to simulate the 3270 environment, right-click on the session you want to cancel, which displays a menu of key names.)

The following message displays:

```
DSQ50465 QMF command interrupted! Clear screen and press ENTER.
```

2. Press the Clear function key.
3. Press Enter.

The **QMF Command Interrupt** panel displays, as shown in the following figure.

```
DSQ50547 QMF command interrupted! Do one of the following:
==> To continue QMF command,      type CONT
==> To cancel QMF command,       type CANCEL
==> To enter QMF debug,          type DEBUG
```

Figure 54. The **QMF Command Interrupt** panel

4. On the QMF command line, enter CANCEL.  
QMF cancels the query.

## Changing a query

You can change a prompted query at any time. If you saved the query, you must first retrieve the it from the database.

### Retrieving a query from the database

To retrieve a query from the database, enter the following on the QMF command line:

```
DISPLAY QUERY queryname
```

The **Prompted Query** panel displays with the query you requested from the database.

## Correcting a query that does not successfully run

If you cannot successfully run a query, one or more of the database objects that are specified in the query probably was updated. For example, perhaps a table name was changed or a column was deleted from a table since the last time you ran the prompted query. You cannot change the query from within QMF in these cases. You must convert the query to SQL or export the query to change it.

## Adding information to a query

You can add information to a query before or after you save it. You can add specifications and you can change existing specifications.

To add a specification to a query:

1. On the **Prompted Query** panel, press the Specify function key.
2. On the **Specify** panel, enter the number of the specification you want to add.

To add further detail to an existing specification:

1. In the echo area, position the cursor on the underscore character that appears to the left of the specification to which you want to add further detail. If a specification is longer than one line, place the cursor on the first line of the specification.
2. Press the Insert key.
3. Enter the information that you want to add to the query. After you press Enter on the last panel, the **Prompted Query** panel displays with the information you added.

Alternatively, you can:

1. Type INSERT on the command line of the **Prompted Query** panel. Do not press Enter.
2. In the echo area, position the cursor on the underscore character that appears to the left of the specification to which you want to add further detail. If a specification is longer than one line, place the cursor on the first line of the specification.
3. Press Enter.

## Changing information in a query

You can change column names, row conditions, sort information, and time period specifications in a query. You cannot change table names, but you can delete a table and specify a new one.

To change information in a query:

1. In the echo area, position the cursor on the underscore character that appears to the left of the specification that you want to change. If the specification is longer than one line, place the cursor on the first line of the specification. For example, to change a time period specification that is longer than one line, place the cursor on the line that contains the table or view name.
2. Press the Change function key.

The appropriate change panel, like the one in the following figure, displays.

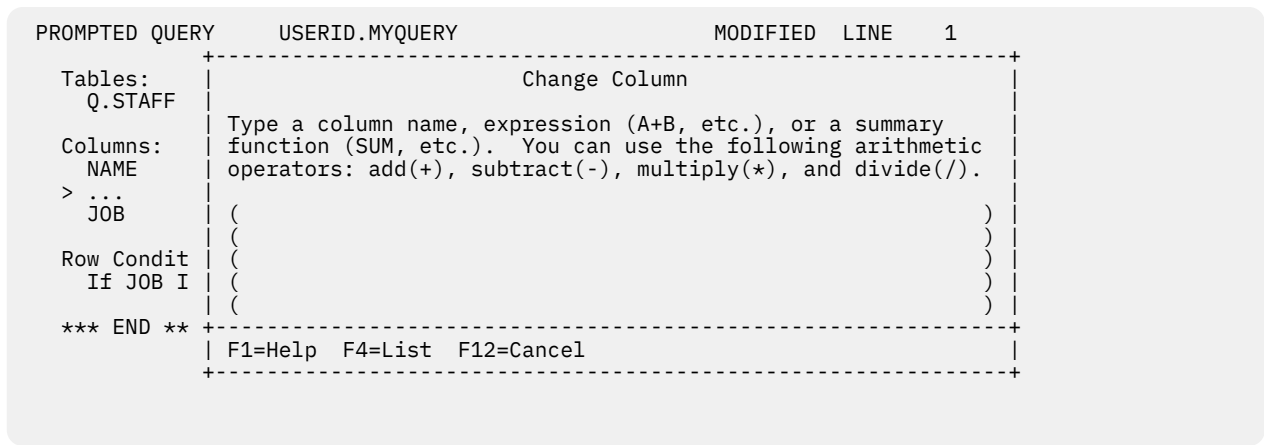


Figure 55. QMF displays a panel where you specify changes to a query.

3. Enter the changes to the information.

The **Prompted Query** panel displays with the changed information in the echo area.

## Deleting information from a query

You can delete any information from a query, including table names.

To delete information from a query:

1. Position the cursor on the underscore character that appears to the left of the specification that you want to delete. To delete a specification that is longer than one line, place the cursor on the first line of the specification.
2. Press the Delete function key.

Remember the following when you are deleting tables or table joins from a query:

- When you delete a table from a query, QMF also deletes any table joins you created with that table.
- When you delete a table or view from a query, any time period specifications that are based on that table or view are deleted also.
- When you change a query so that two or more of the tables in the query are no longer joined, the **Join Tables** panel displays. You can link the tables by another common column.

### Related tasks

#### Converting a prompted query to an SQL query

You can convert a prompted query to an SQL query. Converting a query is useful if you would like to expand a basic prompted query into a more complex query by using the SQL language.

### Related reference

**EXPORT in CICS** Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

**EXPORT in TSO** The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

## Erasing a saved query

You can erase any query you have saved in the database. Only a QMF administrator can erase queries that are saved by others.

### Procedure

To erase a query in the database, enter the following on the QMF command line:

```
ERASE QUERY queryname
```

## Viewing the SQL equivalent of a prompted query

You might want to see the SQL statements that make up a query you create with Prompted Query.

### About this task

When you display the SQL equivalent of a prompted query, you cannot edit, run, or save the query that you display. (You can, however, convert the prompted query to an SQL query.)

### Procedure

To display the SQL equivalent of a prompted query:

1. Display the prompted query on the **Prompted Query** panel.
2. Press the Show SQL function key.

You can also enter `SHOW SQL` on the command line.

The SQL equivalent of the prompted query displays. An example is shown in the following figure.

```
PROMPTED QUERY      USERID.MYQUERY      LINE      1
-----
Tables:
  Q.STAFF(A)
  Q.ORG(B)
Join Tables:
  A.DEPT And B
Columns:
  DEPT
  SALARY
  LOCATION
Row Conditions:
  If DIVISION Is Equal To 'EASTERN'
  Or DEPT Is Equal To 84
*** END ***

+-----+
|          SQL          |
+-----+
| The following SQL statement is equivalent to your query. |
|                               1 to 5 of 5 |
+-----+
| SELECT A.DEPT, A.SALARY, B.LOCATION |
| FROM Q.STAFF A, Q.ORG B |
| WHERE ((B.DIVISION = 'EASTERN') |
|        OR (A.DEPT = 84)) |
|        AND (A.DEPT = B.DEPTNUMB) |
+-----+
| F1=Help F7=Backward F8=Forward F12=Cancel |
+-----+
```

Figure 56. QMF can display the SQL equivalent of a prompted query.

### Related concepts

[Viewing the data in the database using SQL statements](#)

You can select and view data in the database by using SQL statements.

### Related tasks

[Converting a prompted query to an SQL query](#)

You can convert a prompted query to an SQL query. Converting a query is useful if you would like to expand a basic prompted query into a more complex query by using the SQL language.

## Converting a prompted query to an SQL query

You can convert a prompted query to an SQL query. Converting a query is useful if you would like to expand a basic prompted query into a more complex query by using the SQL language.

### About this task

After you convert a prompted query into an SQL query, you cannot convert it back into a prompted query. If you want to keep a copy of your original prompted query, be sure to save it in the database before you convert it to SQL.

## Procedure

To convert a prompted query to an SQL query:

1. If the prompted query is saved in the database, display it by entering the following on the QMF command line:

```
DISPLAY QUERY queryname
```

2. On the QMF command line, enter CONVERT QUERY

The **CONVERT Confirmation** panel displays (if you specified **Yes** for the Confirm option in your QMF user profile), as shown in the following figure.

```
SQL QUERY          USERID.QUERY1          LINE    1
+-----+-----+-----+
S |                                     CONVERT Confirmation
|
| WARNING:
| Your CONVERT command will convert your current query and place
| the SQL translation on the SQL query panel. The original query
| cannot be redisplayed unless it has been saved or exported.
|
| Do you want to convert this query?
| 1 1. YES - Convert the query to an SQL query.
|   2. NO  - Do not convert the query to an SQL query; do not execute the
|           CONVERT command.
+-----+-----+-----+
| F1=Help  F12=Cancel
+-----+-----+-----+
```

Figure 57. The **CONVERT Confirmation** panel

3. Press Enter to accept choice 1, **Yes**.

The SQL query displays, as shown in the following figure.

```
SQL QUERY          LINE    1
SELECT A.DEPT, B.LOCATION, AVG (A.SALARY)
FROM Q.STAFF A, Q.ORG B
WHERE ((B.DIVISION = 'EASTERN')
       OR (A.DEPT = 84))
       AND (A.DEPT = B.DEPTNUMB)
GROUP BY A.DEPT, B.LOCATION
*** END ***
```

Figure 58. QMF displays the SQL query.

## What to do next

You can change the converted query that is shown on the SQL query panel by modifying the existing statements or adding or deleting SQL statements.

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

### Related reference

[CONVERT](#)The CONVERT command converts a Prompted, SQL, or QBE query into a query with standard SQL syntax.

---

# Chapter 5. Viewing the data in the database using SQL statements

You can select and view data in the database by using SQL statements.

When you use SQL statements to select and view data, QMF does not prompt you for information as it does with Prompted Query. However, after you learn the basic rules for writing SQL queries, you might find it faster and simpler.

The examples in this section show reports resulting from sample queries. Some of these queries use expressions to generate columns from existing columns in the database. You can change the default column headings that you see on the resulting reports by using QMF forms.

## Related concepts

### [Customizing your reports](#)

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

## Related information

The IBM Publications Center To find information about the Structured Query Language, see the SQL reference information for the database you are using.

---

## Format of SQL queries

A simple SQL query includes just a few statements.

When you enter SQL statements on the SQL Query panel, you need to know the following items:

- The name of the table from which you want data
- The column names in the table
- The row conditions you want to specify
- The sequence in which you want the data to appear

Many simple SQL queries use the following basic SQL statement:

```
SELECT columnname
FROM tablename
WHERE condition
ORDER BY columnname
```

The following figure shows a basic SQL query. This query displays the employee names, years of service, and salaries from the Q.STAFF table.

```

SQL QUERY                                MODIFIED    LINE 1
SELECT NAME, YEARS, SALARY
FROM Q.STAFF
ORDER BY NAME

*** END ***

1=Help      2=Run      3=End      4=Print    5=Chart    6=Draw
7=Backward  8=Forward  9=Form     10=Insert  11=Delete  12=Report
OK, cursor positioned.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 59. A basic SQL query

## Starting an SQL query

You can use a RESET QUERY statement to start an SQL query panel.

### Procedure

To start an SQL query:

1. Make sure that the value of the Language field in your QMF user profile is SQL.
2. On the QMF command line, enter:

```
RESET QUERY
```

If you do not want to change your QMF profile, you can enter the following command each time you use the RESET command:

```
RESET QUERY (LANG=SQL
```

The SQL query panel displays, as shown in the following figure.

```

SQL QUERY                                MODIFIED    LINE 1

*** END ***

1=Help      2=Run      3=End      4=Print    5=Chart    6=Draw
7=Backward  8=Forward  9=Form     10=Insert  11=Delete  12=Report
OK, cursor positioned.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 60. The SQL Query panel



### Related tasks

[Setting up and changing your QMF user profile](#)

Your QMF user profile tells QMF how you want to display information and process commands, and what printer to use when you print reports or charts.

## Entering SQL statements and running the query

---

You enter SQL statements in the **SQL Query** panel.

### Before you begin

Issue the RESET QUERY command to [start the SQL query](#).

### Procedure

To create and run a SQL query:

1. On the **SQL Query** panel, type the SQL statements that you want in the query.
2. Press the Run function key to run the query.

You can also enter RUN QUERY on the command line.

### Results

When the query completes successfully, a report displays, showing all the data that you selected.

### Related concepts

[Basic SQL statements and functions used in QMF queries](#) You can issue SQL statements directly to the database from the QMF SQL Query panel.

### Related tasks

[Canceling a running query](#)

You can cancel a query by interrupting the RUN QUERY command. Queries might also be canceled by the QMF governor.

[Running a query and displaying a report](#)

When you run a query, QMF displays the retrieved data as a report. The appearance of the report is controlled by choices you make on QMF form panels.

### Related reference

[RUN](#) The RUN command runs queries or procedures from QMF temporary storage or from the database at the current location.

## Including multiple SQL statements in one query

---

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

### About this task

If confirmation prompts are enabled, a confirmation prompt is displayed after every COMMIT statement and after the last statement in the query. However, some SQL statements, such as SET, apply to the QMF session or environment and do not cause a confirmation prompt to be displayed. If the query does not contain a COMMIT statement, a single confirmation prompt is displayed. Your response to the prompt applies to all changes that are made by the query.

If a query contains multiple statements that insert, update, or delete rows in the database object, your response to the confirmation prompt applies to all of the database updates. The information in the confirmation prompt contains the total number of rows that are updated by all of the statements in the query. If the query does not contain statements that insert, update, or delete rows and does not delete objects from the database, a confirmation prompt is not displayed. If the query does not

contain statements that insert, update, or delete rows but deletes an object from the database, a DROP confirmation prompt is displayed. Your response to the prompt in this case applies to both the DROP statement and the other statements in the query.

Any variable value that you supply for the query applies to all SQL statements in the query that include a variable with that name. For example, if you supply a value of "Q.STAFF" for the variable *&TABLE* in the following query, both the UPDATE and INSERT statement change the Q.STAFF table:

```
UPDATE &TABLE
SET ID=53
WHERE NAME='HANES';
INSERT INTO &TABLE
VALUES (42, 'GRIMLEY', 15, 'SALES', 10, 19264.50, 656.34)
```

If the query contains multiple statements and one of the statements fails, processing stops and no subsequent statements are run. If statements before the failing statement changed the database, these changes are rolled back (not applied to the database) unless the query contains a COMMIT statement. If the query contains one or more COMMIT statements, all database changes that occurred before the SQL error and after the last successful COMMIT statement are rolled back. Some statements, such as SET, apply to the QMF session or environment and therefore are not rolled back in error situations. The DSQEC\_RUN\_MQ global variable controls whether queries with multiple SQL statements are allowed. To run a query with multiple statements, ensure that each statement is separated by a semicolon; then set the DSQEC\_RUN\_MQ global variable to 1 and run the query. When the variable is set to zero, all statements after the first semicolon are ignored.

The maximum allowed length of an SQL query that can be run by a RUN QUERY command is determined by the database to which you are connected when you issue the command:

- In Db2 for z/OS, SQL queries can be up to 2 MB long when the DSQEC\_SQLQRYSZ\_2M global variable is set to 1. When the variable is set to 0, the maximum size is 32 KB.
- In DB2 for iSeries and Db2 for Linux, UNIX, and Windows, SQL queries can be up to 65 KB when the DSQEC\_SQLQRYSZ\_2M global variable is set to 1. When the variable is set to 0, the maximum size is 32 KB.
- DB2 for VSE and VM, SQL queries are limited to 8 KB regardless of how the DSQEC\_SQLQRYSZ\_2M global variable is set.

## Procedure

To include multiple statements in a SQL query:

1. Set the DSQEC\_RUN\_MQ global variable to 1:

```
SET GLOBAL (DSQEC_RUN_MQ=1
```

When the variable is set to 0, all statements after the first statement in the query are ignored when you issue the RUN QUERY command.

2. Place a semicolon at the end of each SQL statement on the panel.

No semicolon is required after the last SQL statement.

For example, the following query issues an UPDATE statement that updates the product number that is associated with project number 1401 to 20 in the Q.PROJECT sample table. The UPDATE statement is followed by an INSERT statement that adds a row for a new employee named GRIMLEY in the Q.STAFF sample table:

```
UPDATE Q.PROJECT
SET PRODNUM=20
WHERE PROJNO='1401';
INSERT INTO Q.STAFF
VALUES (42, 'GRIMLEY', 15, 'SALES', 10, 19264.50, 656.34)
```

## Related reference

Global variables that control how commands and procedures are executed  
DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## Spanning a long object name over multiple lines in a query

To continue a statement that contains a long object name on the next line, use double quotation marks.

A single SQL query line is limited to 79 bytes. When any part of an object name (the location, authorization ID, or the object name itself) is continued on a new line, that part of the name must be delimited by double quotation marks. The following figure shows an example of a long object name that spans two lines. The name is qualified with an authorization ID that also spans two lines.

```
SQL QUERY                                MODIFIED LINE    1
SELECT * FROM "VERY_LONG_AUTHID_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXX" ."VERY_LONG_TABLE_NAME_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXX"
```

Figure 61. Delimiting qualified object names that span multiple lines.

## Selecting columns and tables

The format of SQL queries requires you to select columns before you select tables.

### Procedure

To select columns, choose one of the following options:

- Type SELECT, followed by the names of the columns in the order that you want them to appear on the report. Use commas to separate the column names.

For example, to select the **DEPTNAME** and **DEPTNUMB** columns, type SELECT DEPTNAME, DEPTNUMB

To select all columns in the table, type SELECT \*

- If you know the table from which you want to select data, but do not know all the column names, you can use the Draw function key on the **SQL Query** panel to display the column names.

- a) On the QMF command line, type the name of the table whose columns you want to see.

For example, to find the names of the columns in the Q.ORG table, type Q.ORG. If you need to see a list of tables, use the LIST TABLES command.

- b) Press the Draw function key.

QMF displays a query that selects all the columns for the table you specified:

```
SELECT DEPTNUMB, DEPTNAME, MANAGER, DIVISION
      , LOCATION
FROM Q.ORG
```

- c) Leave the query as is, or change it to select a subset of data.  
For example, you can delete certain columns or specify criteria for selecting specific rows).

### Related concepts

[Creating a column by defining expressions in the SQL Query panel](#)

You can create a column for your report by adding, subtracting, multiplying, or dividing the values in two or more columns. Then, you can include the resulting value as a new column with the WHERE keyword.

#### Row selection with SQL statements

You can use SQL statements to select rows from the database to display on your report.

#### Displaying a list of database objects by using the LIST command

You can choose the objects to display with the LIST command.

#### **Related reference**

DRAWThe DRAW command helps you compose a basic SQL query or QBE query.

## Creating a column by defining expressions in the SQL Query panel

You can create a column for your report by adding, subtracting, multiplying, or dividing the values in two or more columns. Then, you can include the resulting value as a new column with the WHERE keyword.

For example, this statement creates a new column that is the total of each employee's salary and commission:

```
SELECT NAME, SALARY + COMM
FROM Q.STAFF
WHERE SALARY + COMM > 20000
```

QMF displays the report shown in the following figure when you run the query:

NAME	COL1
WILLIAMS	20094.15
GRAHAM	21200.30

Figure 62. Report with a column (COL1) resulting from the expression SALARY+COMM

QMF names the calculated column as follows:

- EXPRESSION 1 if you are accessing data in a DB2 for VSE and VM database
- COL1 if you are accessing data in a Db2 for z/OS database
- 1 if you are accessing data in a Db2 for Linux, UNIX, and Windows database
- 00001 if you are accessing data in a DB2 for iSeries database

QMF names additional calculated columns sequentially (for example EXPRESSION 2, COL2, 2, and 00002).

After you define the new column, you can use it just as you would a column you select from a table.

#### **Related tasks**

##### Changing column headings

When you display a report with the default report format, QMF assigns each column a heading. The DSQDC\_COL\_LABELS global variable controls whether the column heading shown in QMF forms defaults to the database label assigned to the column or the name of the column in the table from which it was selected.

#### **Related reference**

Calculated resultsYou can use calculated values as part of a search condition. You can also display them for selected rows just as you display column values.

## Row selection with SQL statements

You can use SQL statements to select rows from the database to display on your report.

### Selecting rows with SQL statements

Selecting rows limits, or creates a subset of, the data in a table. You select rows by creating a row condition.

#### Procedure

To select specific rows to view, use the WHERE keyword, followed by a condition.

If you do not use the WHERE keyword, all the rows in the table are displayed.

For example, to select only the rows for employees who work in Department 20, type:

```
SELECT DEPT, NAME, JOB, COMM
FROM Q.STAFF
WHERE DEPT = 20
```

QMF displays the report shown in the following figure when you run the query:

DEPT	NAME	JOB	COMM
20	SANDERS	MGR	-
20	PERNAL	SALES	612.45
20	JAMES	CLERK	128.20
20	SNEIDER	CLERK	126.50

Figure 63. Report resulting from using a WHERE clause in a query

### Selecting rows that have no data

You can select rows that have no data in a column. The columns that contain no data are considered to be null, meaning their values are unknown.

#### Procedure

To select only rows that have no data, enter:

```
WHERE columnname IS NULL
```

For example, to select employees with no commission, type:

```
WHERE COMM IS NULL
```

### Selecting rows by using specific character values

You can use character values to select the rows you want to view. Make sure that you enclose the data with single quotation marks.

For example:

```
SELECT NAME, JOB
FROM Q.STAFF
WHERE NAME = 'SANDERS'
```

### Selecting rows by using conditions

You can specify a condition when you select rows.

=

Equal to

- >  
Greater than
- >=  
Greater than or equal to
- <  
Less than
- <=  
Less than or equal to
- ≠  
Not equal to
- <>  
Not equal to

The following query selects employees who earn a commission greater than or equal to \$1,000.00.

```
SELECT ID, COMM
FROM Q.STAFF
WHERE COMM >= 1000
```

The following query selects employees who earn a commission of at least \$170.00, but not more than \$220.00.

```
SELECT ID, COMM
FROM Q.STAFF
WHERE COMM BETWEEN 170 AND 220
```

### Related information

The IBM Publications Center To find information about the BETWEEN keyword, see the SQL reference information for the database you are using.

## Selecting rows using opposite conditions

You specify the opposite of any condition by typing NOT before the entire condition.

For example, the following selects all rows except those rows where the employee has 10 years of service.

```
SELECT * FROM Q.STAFF
WHERE NOT YEARS = 10
```

To specify the opposite of a NULL, LIKE, IN, or BETWEEN condition, type NOT right before the condition keyword.

For example, the following selects all rows where the employee has any number of years of service listed in the YEARS column:

```
SELECT * FROM Q.STAFF
WHERE YEARS IS NOT NULL
```

The following query selects employees whose salary is under \$16,000.00 or over \$22,000.00:

```
SELECT ID, NAME, SALARY
FROM Q.STAFF
WHERE SALARY NOT BETWEEN 16000 AND 22000
```

The following query selects employees whose salary is under \$16,000.00 and who earn less than \$500.00 in commissions:

```
SELECT ID, NAME, SALARY, COMM
FROM Q.STAFF
WHERE NOT SALARY > 16000 AND NOT COMM > 500
```

## Selection symbols to narrow row selection

To select rows using selection symbols for character or graphic data, use the LIKE keyword in a WHERE clause, and the underscore and percent sign as selection symbols.

An underscore (\_) signifies any one character. The percent sign (%) signifies zero or more characters. Any other character represents a single occurrence of itself.

For example, this query selects the rows for employees whose names end in SON.

```
SELECT NAME
FROM Q.STAFF
WHERE NAME LIKE '%SON'
```

This query selects the rows for employees whose names are five characters long and end in ES.

```
SELECT NAME
FROM Q.STAFF
WHERE NAME LIKE '___ES'
```

(The line ' \_\_\_ES ' includes three underscores.)

The query produces the report shown in the following figure.

```
NAME
-----
HANES
JAMES
JONES
```

Figure 64. Report resulting from a query that uses the LIKE operator

You can use % more than once in an expression. For example, the following WHERE clause selects the rows for employees whose names contain an M and an N:

```
WHERE NAME LIKE '15N%'
```

The query returns rows for employees MARENGHI, ROTHMAN, and MOLINARE.

You can use the % and \_ selection symbols in the same WHERE clause. For example, the following query selects the rows for employees whose names have R as the second letter. From the Q.STAFF sample table, this query selects rows for employees FRAYE and GRAHAM:

```
SELECT * FROM Q.STAFF
WHERE NAME LIKE '_R%'
```

You can use the NOT keyword with a LIKE clause to specify rows you do not want to select. For example, the following query selects the rows for employees whose names do not begin with G.

```
WHERE NAME NOT LIKE 'G%'
```

## Narrowing row selection using multiple row conditions

You can create multiple row conditions, and use the AND, OR, or IN keywords to connect the conditions.

### Selecting rows where both conditions are true

If you want to select rows that meet both conditions, use the AND keyword to connect the conditions.

The following query displays the ID, NAME, YEARS, and SALARY of employees in the Q.STAFF table that have at least 10 years of service and earn more than \$20,000.

```
SELECT ID, NAME, YEARS, SALARY
FROM Q.STAFF
```

```
WHERE YEARS >= 10
AND SALARY > 20000
```

## Selecting rows where either condition is true

If you want to select rows that meet either condition, use the OR keyword to connect the conditions.

The following query displays the same columns in the Q.STAFF table, but selects employees who either have at least 10 years of service or earn more than \$20,000.00.

```
SELECT ID, NAME, YEARS, SALARY
FROM Q.STAFF
WHERE YEARS >= 10
OR SALARY > 20000
```

## Grouping row conditions

You can use AND and OR statements together to connect conditions. Use parentheses to indicate which conditions you want QMF to check first. The conditions inside the parentheses are checked first, and then the conditions outside the parentheses.

If you do not use parentheses, NOT is applied before AND, and AND is applied before OR.

For example, when you run this query:

```
SELECT NAME, ID, DEPT
FROM Q.STAFF
WHERE (JOB='SALES' AND COMM > 1200) OR YEARS > 10
```

QMF displays the report shown in the following figure.

NAME	ID	DEPT
KOONITZ	90	42
JONES	260	10
GRAHAM	310	66
EDWARDS	340	84

Figure 65. Report resulting from a query using AND and OR

Now run the same query with the parentheses moved:

```
SELECT NAME, ID, DEPT
FROM Q.STAFF
WHERE JOB='SALES' AND (COMM > 1200 OR YEARS > 10)
```

QMF displays the report shown in the following figure:

NAME	ID	DEPT
KOONITZ	90	42
GRAHAM	310	66
EDWARDS	340	84

Figure 66. The same query with parentheses in different places produces a different report.

## Selecting rows using the IN predicate

You can use one IN statement to replace multiple OR statements.

Both of the following statements select the same rows to view on the report:

```
SELECT ID, NAME, YEARS, SALARY
FROM Q.STAFF
WHERE DEPT = 38 OR DEPT = 20 OR DEPT = 42
```



```
SELECT ID, NAME, YEARS, SALARY
FROM Q.STAFF
WHERE DEPT IN (38, 20, 42)
```

Use NOT with the IN statement to specify rows you do not want to select. For example, the following clause in a query selects all rows except those where DEPT is 15, 20, or 38:

```
WHERE DEPT NOT IN (15, 20, 38)
```

## Eliminating duplicate rows

Use the DISTINCT keyword to eliminate duplicate rows from a report.

The following query displays each department in which some employee is a salesperson. Even if a department has more than one salesperson, QMF displays the department number only once on the report.

```
SELECT DISTINCT DEPT
FROM Q.STAFF
WHERE JOB = 'SALES'
ORDER BY DEPT
```

## Sorting the rows in a query

If you sort rows by more than one column, the first column is ordered first, the second column is ordered within the order of the first column, and so on.

### Procedure

To specify the way you want to sort the rows, use the ORDER BY keyword. Follow ORDER BY with the name of the column (or columns) on which you want to sort the rows.

QMF sorts the rows in ascending order unless you specify descending order.

### Example

For example, the following query displays the rows in ascending order by job:

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE DEPT = 84
ORDER BY JOB
```

This query displays the rows in ascending order by job, and orders the years within job in descending order:

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE DEPT=84
ORDER BY JOB, YEARS DESC
```

The resulting report appears as shown in the following figure:

NAME	JOB	YEARS
GAFNEY	CLERK	5
QUILL	MGR	10
EDWARDS	SALES	7
DAVIS	SALES	5

Figure 67. Sorting rows by years within job

## Adding or deleting lines in an SQL query

To add or delete lines in an SQL statement, use the Insert or Delete function keys.

### Procedure

Choose one of the following options, depending on whether you want to add or delete lines from an SQL query:

- To add new lines to an SQL query:
  - a) If you did not save the query in the database, display the query again by entering `SHOW QUERY` on the QMF command line. You can also display the query by pressing the Query function key. If you saved the query in the database, enter `DISPLAY QUERY queryname`.
  - b) Move the cursor above the place where you want to add information.
  - c) Press the Insert function key. QMF displays a blank line.
  - d) Type the information on the blank line.

For this example, add a condition to select only employees from Department 38:

```
SELECT NAME, YEARS, SALARY
FROM Q.STAFF
WHERE DEPT=38
ORDER BY NAME
```

- To delete a line, move the cursor to the line you want to delete, and press the Delete function key.

## Data display from multiple tables using SQL statements

With QMF, you can display data from more than one table, eliminate information from duplicate rows, and join multiple tables.

### Displaying data from more than one table

There are many ways to display data from more than one table. You can join tables or views by a common column. You can also merge data from two or more tables or views into a single column or create a subquery to retrieve data from several tables.

### Joining columns in two or more tables

You can use a `SELECT` statement to join columns in two or more tables.

The SQL statement in the following figure uses data from the `Q.STAFF` and `Q.ORG` tables to select all the clerks in the Eastern division.

If you check the sample tables, you see that the department numbers are found in both tables, the division name is in the `Q.ORG` table, and the job title is in the `Q.STAFF` table. In `Q.ORG`, the department number is in the `DEPTNUMB` column, and in `Q.STAFF`, the department number is in the `DEPT` column. You will join the tables by these two columns.

Specify all the columns you want to display on the report in the `SELECT` clause. Use the `FROM` clause to specify the tables you want to join. Specify the columns whose values are equal in the `WHERE` clause, separated by an equal (=) sign.

```
SELECT DIVISION, ID, LOCATION, NAME
FROM Q.STAFF, Q.ORG
WHERE DIVISION = 'EASTERN'
AND JOB='CLERK'
AND DEPTNUMB = DEPT
ORDER BY ID
```

Figure 68. A SQL query that joins the `Q.STAFF` and `Q.ORG` tables

The report in the following figure displays when you run the query. If you do not specify a common column when you join two tables, each row in the first table is joined to each row in the second table. The resulting report might contain duplicate data and might be very large.

DIVISION	ID	LOCATION	NAME
EASTERN	80	WASHINGTON	JAMES
EASTERN	110	BOSTON	NGAN
EASTERN	120	ATLANTA	NAUGHTON
EASTERN	170	BOSTON	KERMISCH
EASTERN	180	ATLANTA	ABRAHAMS
EASTERN	190	WASHINGTON	SNEIDER

Figure 69. The report shows the data from both tables.

The columns in the tables you are joining might have the same name. Use one of the following methods to distinguish between columns with the same name:

- Add a qualifier to the column name
- Specify a correlation name to identify a column with a particular table

### Using qualifiers to distinguish between column names

You can add a qualifier to identical column names to identify the table from which you selected the column.

For example, to distinguish between the PRODNUM column in the Q.PRODUCTS table, and the PRODNUM column in the Q.PROJECT table, add the following qualifiers to the column names:

- Add Q.PRODUCTS to the PRODNUM column from the PRODUCTS table
- Add Q.PROJECT to the PRODNUM column from the PROJECT table

The SQL statement in the following figure selects all the product numbers in both the Q.PRODUCTS and Q.PROJECT tables, as well as the project numbers, departments, and product prices.

You only need to specify one of the duplicate column names when you select columns because the two columns are combined in the report. Use a qualifier for duplicate column names everywhere you refer to them in the query.

```
SELECT PROJNO, Q.PRODUCTS.PRODNUM, DEPT, PRODPRICE
FROM Q.PROJECT, Q.PRODUCTS
WHERE Q.PRODUCTS.PRODNUM < 100 AND
Q.PRODUCTS.PRODNUM = Q.PROJECT.PRODNUM
```

Figure 70. This SQL query selects data from two columns with the same name.

### Using correlation names to distinguish between column names

Correlation names are used to identify the tables or views from which you selected columns when more than one column has the same name.

For example, to distinguish between the PRODNUM column in the Q.PRODUCTS table and the PRODNUM column in the Q.PROJECT table, you could specify a correlation name of P for Q.PROJECT and a correlation name of S for Q.PRODUCTS.

Use the correlation name as a prefix to the column name wherever you refer to that column. The query in the following figure shows examples of using correlation names:

```
SELECT PROJNO, S.PRODNUM, DEPT, PRODPRICE
FROM Q.PROJECT P, Q.PRODUCTS S
WHERE S.PRODNUM < 100 AND
S.PRODNUM = P.PRODNUM
```

Figure 71. Using correlation names in a query

## Related concepts

[Writing correlation names](#)

Names that are used for correlations in queries are arbitrary. Choose any name up to 128 characters long. It must not duplicate other words in the query or any SQL reserved words.

[QMF sample tables](#)

## Merging data from multiple tables into a single column

You can merge data from two or more tables into a single column on a report by using the keyword UNION.

### Procedure

- Create two or more queries to select the data you want to merge, then specify the keyword UNION between the queries.

In the following figure, the first query selects the department name and number from the Q.ORG table and creates a column that displays the words WAITING FOR WORK. The second query selects the department name and number from the Q.PROJECT and Q.ORG tables, and creates a column that displays the words HAS WORK. The database determines the name of the new column, unless you change it using QMF forms.

Select the same number of columns for each query. Corresponding columns must be the same general data type. Corresponding columns must all either allow null values or not allow null values. If you want to order the columns, specify a column number because the names of the columns you are merging are probably different. If you want to display duplicate rows on the report, specify UNION ALL instead of UNION.

```
SELECT DEPTNUMB, DEPTNAME, 'WAITING FOR WORK'  
  FROM Q.ORG  
 WHERE DEPTNUMB NOT IN (SELECT DEPT FROM Q.PROJECT)  
UNION  
SELECT O.DEPTNUMB, O.DEPTNAME, 'HAS WORK'  
  FROM Q.PROJECT P, Q.ORG O  
 WHERE P.DEPT = O.DEPTNUMB  
ORDER BY 1
```

Figure 72. This SQL query merges data from two columns into one.

QMF displays the report that is shown in the following figure when you run the query, showing the department names and numbers and their status information on the same report.

DEPTNUMB	DEPTNAME	COL1
10	HEAD OFFICE	HAS WORK
15	NEW ENGLAND	HAS WORK
20	MID ATLANTIC	HAS WORK
38	SOUTH ATLANTIC	HAS WORK
42	GREAT LAKES	HAS WORK
51	PLAINS	HAS WORK
66	PACIFIC	HAS WORK
84	MOUNTAIN	WAITING FOR WORK

Figure 73. The report shows the two new columns merged into one.

You can specify the order in which you want to merge the columns from multiple tables. Specifying order is important when you use the UNION and UNION ALL keywords. Use parentheses to indicate the columns or which tables you want merged first. The conditions inside the parentheses are checked before the conditions outside the parentheses.

## Example

For example, this query produces Report A in the following figure:

```
(SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE SALARY>12000
 UNION ALL
 SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE DEPT=38)
 UNION
 SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE JOB='SALES'
```

If you move the parentheses, the same query produces Report B in the following figure:

```
SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE SALARY>12000
 UNION ALL
 (SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE DEPT=38)
 UNION
 SELECT ID, NAME, SALARY
 FROM Q.STAFF
 WHERE JOB='SALES')
```

REPORT A			REPORT B		
ID	NAME	SALARY	ID	NAME	SALARY
10	SANDERS	18357.50	20	PERNAL	18171.25
20	PERNAL	18171.25	30	MARENGHI	17506.75
30	MARENGHI	17506.75	40	O'BRIEN	18006.00
40	O'BRIEN	18006.00	60	QUIGLEY	16808.30
50	HANES	20659.80	70	ROTHMAN	16502.83
60	QUIGLEY	16808.30	90	KOONITZ	18001.75
70	ROTHMAN	16502.83	120	NAUGHTON	12954.75
80	JAMES	13504.60	150	WILLIAMS	19456.50
90	KOONITZ	18001.75	180	ABRAHAMS	12009.75
100	PLOTZ	18352.80	220	SMITH	17654.50
110	NGAN	12508.20	280	WILSON	18674.50
120	NAUGHTON	12954.75	300	DAVIS	15454.50
140	FRAYE	21150.00	310	GRAHAM	21000.00
150	WILLIAMS	19456.50	320	GONZALES	16858.20
160	MOLINARE	22959.20	340	EDWARDS	17844.00
170	KERMISCH	12258.50	10	SANDERS	18357.50
180	ABRAHAMS	12009.75	20	PERNAL	18171.25
190	SNEIDER	14252.75	30	MARENGHI	17506.75
210	LU	20010.00	40	O'BRIEN	18006.00
220	SMITH	17654.50	50	HANES	20659.80
230	LUNDQUIST	13369.80	60	QUIGLEY	16808.30
240	DANIELS	19260.25	70	ROTHMAN	16502.83
250	WHEELER	14460.00	80	JAMES	13504.60
260	JONES	21234.00	90	KOONITZ	18001.75
270	LEA	18555.50	100	PLOTZ	18352.80
280	WILSON	18674.50	110	NGAN	12508.20
290	QUILL	19818.00	120	NAUGHTON	12954.75
300	DAVIS	15454.50	140	FRAYE	21150.00
310	GRAHAM	21000.00	150	WILLIAMS	19456.50
320	GONZALES	16858.20	160	MOLINARE	22959.20
340	EDWARDS	17844.00	170	KERMISCH	12258.50
350	GAFNEY	13030.50	180	ABRAHAMS	12009.75

Figure 74. The two reports show the differences in merging order.

The first query selects employees whose salaries are greater than \$12,000.00 and all employees from Department 38. Then, it eliminates any duplicate entries by selecting only employees who work in Sales and are not in Department 38 or making more than \$12,000.00 a year.

The second query allows duplicate entries because it first selects employees from Department 38 and employees from outside Department 38 who work in Sales. Then, it adds employees whose salaries are more than \$12,000.00.

## Creating a subquery to retrieve data from more than one table

When you need to include in your WHERE clause selection criteria that only exists in another table, you can add subqueries to a SQL statement to retrieve the values that satisfy the condition. A subquery is a complete query that appears in the WHERE or HAVING clause of an SQL statement.

You can specify up to 16 subqueries within a single SQL statement, and you can specify subqueries within a subquery. Subqueries run from last to first within the main SQL statement in which they appear.

### Rules for creating a subquery

- Enclose the subquery in parentheses.
- Specify only one column or expression in a subquery unless you are using IN, ANY, ALL, or EXISTS.
- A subquery cannot contain a BETWEEN or LIKE clause.
- A subquery cannot contain an ORDER BY clause.
- A subquery in an UPDATE statement cannot retrieve data from the same table in which data is to be updated.
- A subquery in a DELETE statement cannot retrieve data from the same table in which data is to be deleted.

The following query displays the names and IDs of employees who work in Boston. The subquery (in parentheses) finds the department number for the location of BOSTON in the Q.ORG table. Then the main query selects the names of the employees in that department from the Q.STAFF table.

```
SELECT NAME, ID
FROM Q.STAFF
WHERE DEPT=(SELECT DEPTNUMB
            FROM Q.ORG
            WHERE LOCATION='BOSTON')
```

In the next example, the subquery and main query retrieve data from the same table. The subquery calculates the average salary for all the employees in the Q.STAFF table. Then, the main query selects the salespeople whose salaries are equal to or greater than the average salary.

```
SELECT ID, NAME, SALARY
FROM Q.STAFF
WHERE JOB = 'SALES' AND
      SALARY >= (SELECT AVG(SALARY)
                FROM Q.STAFF)
```

### Retrieving more than one value with a subquery

Usually a subquery selects only one column and returns only one value to the query. However, you can create a subquery that uses the ANY or ALL keywords with comparison operators (=, <=>, >, >=>, <, or <=>) to return a particular set of values. In addition, just as you use the IN keyword in place of multiple OR statements in a query, you can also use IN in place of the ANY keyword in a subquery.

The query in the following figure selects any employee who works in the Eastern division. The subquery finds the department numbers in the Eastern division, and then the main query selects the employees who work in any of these departments.

Use the ANY keyword for this query, because it is likely that the subquery will find more than one department in the Eastern division. If you use the ALL keyword instead of the ANY keyword, no data is selected because no employee works in all departments of the Eastern division.

```

SELECT NAME, ID
FROM Q.STAFF
WHERE DEPT = ANY
      (SELECT DEPTNUMB
       FROM Q.ORG
       WHERE DIVISION='EASTERN')

```

Figure 75. This SQL query contains a subquery that uses the ANY keyword.

The query in the following figure selects the department with the highest average salary. The subquery finds the average salary for each department, and then the main query selects the department with the highest average salary.

Use the ALL keyword for this subquery. The department selected by the query must have an average salary greater than or equal to all the average salaries of the other departments.

```

SELECT DEPT, AVG(SALARY)
FROM Q.STAFF
GROUP BY DEPT
HAVING AVG(SALARY) >= ALL
      (SELECT AVG(SALARY)
       FROM Q.STAFF
       GROUP BY DEPT)

```

Figure 76. This SQL query contains a subquery using the ALL keyword.

The query in following figure selects all salespeople (and their salaries) who work for managers earning more than \$20,000 a year. The subquery finds the managers who earn more than \$20,000 a year, and the main query selects the salespeople who work for those managers.

Use the IN keyword for this subquery because you need to find values from more than one department.

```

SELECT ID, NAME, SALARY
FROM Q.STAFF
WHERE JOB = 'SALES'
AND DEPT IN
      (SELECT DISTINCT DEPT
       FROM Q.STAFF
       WHERE JOB = 'MGR'
       AND SALARY > 20000)

```

Figure 77. This SQL query contains a subquery using the IN keyword.

You can also specify NOT IN in a subquery to select information from one table when corresponding information does not exist in the other table.

## Checking for rows that satisfy a condition

Not only can you use a subquery to return a value to the query, you can also use a WHERE EXISTS clause in a subquery to check for rows that satisfy a certain row condition.

The query in the following figure selects employees from the Q.STAFF table who have a salary of less than \$14,000 and who work in a department where at least one other employee with the same job earns a salary greater than \$14,000. (The subquery checks for other employees in the department with the same job, but who earn a salary greater than \$14,000.)

```

SELECT NAME, DEPT, JOB, SALARY
FROM Q.STAFF S
WHERE S.SALARY < 14000 AND
      EXISTS (SELECT * FROM Q.STAFF
             WHERE S.DEPT=DEPT AND SALARY >14000
             AND S.JOB=JOB)
ORDER BY S.DEPT

```

Figure 78. This subquery checks for rows that satisfy a condition.

## Specifying a correlation name in a subquery

You can specify a correlation name in a subquery to use a condition to evaluate every row retrieved by the main query.

The query in the following figure selects the department, name, and salary of the employees who have the highest salary in their departments. The subquery calculates the maximum salary for each department that is selected by the main query. The correlation name, Y, compares each row that is selected by the query to the maximum salary that is calculated for the department in the subquery.

```
SELECT DEPT, NAME, SALARY
FROM Q.STAFF Y
WHERE SALARY = (SELECT MAX (SALARY)
                FROM Q.STAFF
                WHERE DEPT = Y.DEPT)
```

*Figure 79. This subquery specifies a correlation name.*

## Writing correlation names

Names that are used for correlations in queries are arbitrary. Choose any name up to 128 characters long. It must not duplicate other words in the query or any SQL reserved words.

A correlation name is an identifier that you specify for a table or view. Use correlation names to avoid ambiguity, to establish correlated references, or to use shorter names for tables or views. The correlation name must be unique within the query and must appear in the FROM clause of the main query and the WHERE clause of the subquery.

If you use correlation names and several table names, separate the items in the list with commas. For example:

```
FROM Q.ORG XXX, Q.APPLICANT, Q.STAFF YYY
```

### Example 1

You might need correlation names even without a subquery.

This query lists employees whose salaries are greater than their managers' salaries, and selects twice from Q.STAFF. Selecting twice from Q.STAFF is necessary so that the DEPT of each person can be matched with every other DEPT in the table to discover which employees work for which managers.

```
SELECT X.ID, X.NAME, X.SALARY, Y.SALARY
FROM Q.STAFF X, Q.STAFF Y
WHERE X.DEPT = Y.DEPT
      AND Y.JOB = 'MGR'
      AND X.SALARY > Y.SALARY
```

The WHERE condition selects employees from both tables who are in the same department, and uses the "Y" correlation name to select employees who are managers. Then, it selects employees whose salaries are greater than their managers' salaries.

### Example 2

This query lists employees who earn the largest commission in each location.

```
SELECT LOCATION, ID, NAME, COMM
FROM Q.STAFF, Q.ORG ZZZ
WHERE DEPT=DEPTNUMB
      AND COMM = (SELECT MAX(COMM)
                  FROM Q.STAFF, Q.ORG
                  WHERE DEPT=DEPTNUMB
                  AND LOCATION = ZZZ.LOCATION)
```



In this query, the subquery first finds the largest commission within a location. The main part of the query finds who within that location earned that commission. Because the query names two tables, it includes a correlation name that indicates which table contains the LOCATION column.

### Example 3

This query lists employees whose salaries are greater than their managers' salaries. Another version of this query, which is shown in Example 1, selects twice from Q.STAFF without using a subquery. A query that joins two or more tables probably runs faster than a similar query that uses a subquery.

```
SELECT ID, NAME, SALARY
FROM Q.STAFF EMP
WHERE SALARY > (SELECT SALARY
FROM Q.STAFF
WHERE ID = (SELECT MANAGER
FROM Q.ORG
WHERE DEPTNUMB = EMP.DEPT))
```

#### Related information

The IBM Publications Center To find information about SQL reserved words, see the SQL reference information for the database you are using.

## SQL column functions

A column function produces a single value for a group of rows.

For example, if an SQL SELECT clause asks for the value SUM(SALARY), QMF returns only one value, the sum.

These column functions are available in QMF.

#### AVG

Finds the average of the values in a particular column or a set of values that are derived from one or more columns. The column or expression that is summarized must contain numeric data.

#### MAX

Finds the maximum value in a particular column or a set of values that are derived from one or more columns. MAX applies to all data types.

#### MIN

Finds the minimum value in a particular column or a set of values that are derived from one or more columns. MIN applies to all data types.

#### SUM

Finds the sum of the values in a particular column or a set of values that are derived from one or more columns. The column or expression that is added must contain numeric data.

#### COUNT

Finds the number of rows that satisfy the search condition or finds the number of distinct values in a particular column.

The following example query shows the use of the column function SUM:

```
SELECT SUM(SALARY)
FROM Q.STAFF
WHERE DEPT = 38
```

QMF returns the report that is shown in the following figure, which is the sum of all salaries for employees in Department 38:

```
----- COL1
-----
77285.55
```

Figure 80. Report produced by using the SUM column function

The SELECT clause of the SQL statement in this figure uses all five column functions.

```
SELECT SUM(SALARY), MIN(SALARY), MAX(SALARY),  
       AVG(SALARY), COUNT(*)  
FROM Q.STAFF  
WHERE DEPT = 38
```

Figure 81. This SQL query uses the QMF column functions.

The SQL statement produces this report:

COL1	COL2	COL3	COL4	COL5
77285.55	12009.75	18006.00	15457.1100000000	5

Figure 82. The report demonstrates the results of QMF column functions.

You can use the FORM.MAIN or FORM.COLUMNS panel to make the column headings more descriptive.

If you use column functions in an SQL statement where there is no GROUP BY clause, every occurrence of a column name must have a column function so the query can return a single row.

### Related concepts

#### [Customizing your reports](#)

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

[Basic SQL statements and functions used in QMF queries](#) You can issue SQL statements directly to the database from the QMF SQL Query panel.

## SQL scalar functions

You can use scalar functions to convert a value from one data type to another and process date/time values. You can also use them to manipulate parts of character or graphic strings and avoid null values.

Scalar functions produce a single scalar value for each row that satisfies the search condition in your query. For example, if you replace SUM in the SELECT clause of the SQL statement in [Figure 81 on page 78](#) with the scalar function HEX, five rows are returned. One hexadecimal value is returned for each row that satisfies the search condition. The new query is as follows:

```
SELECT HEX(SALARY)  
FROM Q.STAFF  
WHERE DEPT = 38
```

This query produces the report that is shown in the following figure:

COL1
1750675C
1800600C
1680830C
1295475C
1200975C

Figure 83. Report produced by using the HEX function in the query

Conversion functions, date/time functions, and string functions are subsets within the set of scalar functions.

### Related reference

[SQL scalar functions](#) Three types of SQL scalar functions are date/time functions, conversion functions, and string functions.

## Converting a value from one data type to another

The scalar functions DECIMAL, DIGITS, FLOAT, DECFLOAT, HEX, INTEGER, VARGRAPHIC, BIGINT, BINARY, and VARBINARY allow you to convert a value from one data type to another.

The DECIMAL function returns a decimal representation of a number. The DECIMAL function takes the following as input:

- A numeric expression is an expression that returns a value of any numeric data type.
- A precision integer is an integer constant with a value in the range of 1 to 31.
- A scale integer is an integer constant in the range of 0 to the precision-integer value.

The DIGITS function returns values without a decimal point.

The FLOAT function returns a floating-point representation of a number.

The DECFLOAT function returns a decimal floating-point representation of a number. The processor on which QMF is running must support decimal floating-point instructions.

The HEX function uses the hexadecimal numbering system.

The VARGRAPHIC function converts a mixed single-byte and double-byte character string to a pure double-byte character string. VARGRAPHIC returns a varying-length graphic string (data type VARGRAPHIC) result.

The first or only argument of each of these functions is a number or an expression that gives the value to convert.

For example, when you run this SQL statement:

```
SELECT SALARY,          --SALARY
       DECIMAL(SALARY,9,3), --COL1
       DIGITS(SALARY),   --COL2
       FLOAT(SALARY),    --COL3
       HEX(NAME),        --COL4
       FLOAT(YEARS)      --COL5
FROM Q.STAFF
WHERE DEPT = 10
```

QMF produces the report shown in the following figure:

SALARY	COL1	COL2	COL3	COL4	COL5
22959.20	22959.200	2295920	2.296E+04	D4D6D3C9D5C1D9C5	7.000E+00
20010.00	20010.000	2001000	2.001E+04	D3E4	1.000E+01
19260.25	19260.250	1926025	1.926E+04	C4C1D5C9C5D3E2	5.000E+00
21234.00	21234.000	2123400	2.123E+04	D1D6D5C5E2	1.200E+01

Figure 84. Report produced using a number of scalar functions at once

### Related information

The IBM Publications Center To find descriptions of scalar functions, see the SQL reference information for the database you are using.

## Date and time formats

The date, time, and timestamp scalar functions change the data types of their arguments to the data type specified by the function.

The DATE function returns a date from a value. The argument must be a timestamp, a date, or a string representation of a date.

In the following SQL statement, the argument for DATE is a timestamp:

```
SELECT PROJNO, DATE(TIMESTAMP)
FROM Q.PROJECT
WHERE PROJNO = '1401'
```

The query produces the report that is shown in the following figure:

```
PROJNO  COL1
-----  -----
1401    1994-12-18
```

Figure 85. Report produced by using the DATE function

The TIME function returns a time from a value. The argument must be a time, a timestamp, or a string representation of a time. For example, consider the following SQL statement:

```
SELECT PRODNUM, TIME(TIMESTAMP)
FROM Q.PROJECT
WHERE YEAR(STARTD) = 1996
```

When you run this query, QMF produces the report that is shown in the following figure, where TIME shows the time portions of three timestamps in the Q.PROJECT table:

```
PRODNUM  COL1
-----  -----
      10  10.14.44
      50  10.15.01
     150  10.22.23
```

Figure 86. Report produced by using the TIME function

The TIMESTAMP function returns a timestamp from a value or a pair of values. If only one argument is specified, it must be a timestamp, a string representation of a timestamp, a character string of length 8, or a character string of length 24. If the value is a character string of length 24, it must be in the form *yyyymmddhhmmssssssssssss*, where *yyyy* is the year, *mo* is the month, *dd* is the day, *hh* is the hour, *mm* is the minute, and *ssssssssss* specifies the fractional seconds.

If a second optional argument is specified, it must be a time or a string representation of a time, and the first argument must be a date or a string representation of a date. For example, for the following statement, QMF produces a timestamp that represents 10 a.m. today:

```
TIMESTAMP (CURRENT DATE, '10.00.00')
```

The CHAR function returns a string representation of a date/time value. CHAR changes the value of its argument (a date or time value) to the CHAR data type. The result of CHAR is a fixed-length character string representation of a date/time value in the format that is specified by its optional second argument. If the first argument is a date or a time, the second argument must be USA, ISO, JIS, EUR, or LOCAL. LOCAL refers to a site default format. If you omit the second argument and the DSQSFISO program parameter is set to YES, the date or time format is ISO. If you omit the second argument and the DSQSFISO program parameter is set to NO, the date or time is returned in the format that is specified in the DATE FORMAT and TIME FORMAT fields on Db2 installation panel DSNTIP4. To determine the DATE FORMAT and TIME FORMAT values defined in the current Db2 subsystem, reference the DSQAO\_DATE\_FORMAT and DSQAO\_TIME\_FORMAT global variables.

The CHAR format behavior may be changed within a QMF session by setting the DSQEC\_DSQSFISO global variable. For example, if the DATE FORMAT on my Db2 subsystem is USA and global variable DSQEC\_DSQSFISO is set to a value of '0', running the following query:

```
SELECT CHAR(CURRENT DATE) FROM SYSIBM.SYSDUMMY1
```

will produce a report with the current date formatted in USA format:

```
COL1
-----
06/20/2018
```

Figure 87. Report produced with the current date formatted in USA format

The following tables show examples of DATE and TIME formats in USA, ISO, JIS, and EUR. In these tables, USA refers to United States of America format, ISO refers to International Standards Organization format, JIS refers to Japanese Industrial Standard format, and EUR refers to European format.

Table 7. DATE formats		
Date format	Edit code	Example
USA	TDMx	12/15/2006
ISO, JIS	TDYx	2006-12-15
EUR	TDDx	15.12.2006

Table 8. TIME formats		
Time format	Edit code	Example
USA	TTUx	01:25 PM
ISO, EUR	TTSx	13.25.10
JIS	TTSx	13:25:10

Consider the following SQL query with a second argument of USA for CHAR:

```
SELECT TEMPID, CHAR(INTDATE, USA)
FROM Q.INTERVIEW
WHERE MANAGER = 140
```

When you run this query, QMF produces the report that is shown in the following figure:

```
TEMPID  COL1
-----  -----
      420  04/07/1996
      490  09/30/1996
```

Figure 88. A report produced by using the CHAR function with two arguments

### Related reference

[Edit codes for date data](#) By default, the TD edit code displays dates in the format specified at the database requester.

[Edit codes for time data](#) You can use several edit codes to format time data.

## Allowing the database requester to control date and time format

The default edit codes for formatting date and time allow the database requester to control how these values are displayed. These default edit codes are set by the DSQDC\_EC\_DATE and DSQDC\_EC\_TIME global variables.

Whatever format is in effect at the database requester is used by QMF. For example, if you are using the TD edit code in QMF and the database requester specifies USA format for the DATE and TIME fields, the date is displayed as MM/DD/YYYY (TDMx). If the database requester specifies Japanese Industrial Standard format for the DATE and TIME fields, the date is displayed as YYYY-MM-DD (TDYx).

## Isolating the day, month, or year of a date

The DAY function returns the day part of a value. The argument must be a date, timestamp, or a decimal number that is interpreted as a duration of years, months, or days.

The following SQL statement produces a report showing on which day of the month an interview occurs:

```
SELECT TEMPID, DAY(INTDATE)
FROM Q.INTERVIEW
WHERE MANAGER = 270
```

The report shown in the following figure shows that the interviews take place on the fifth day of the month:

TEMPID	COL1
400	5
470	5

Figure 89. A report produced using the DAY function

The MONTH function returns the month part of a value. The argument must be a date, timestamp, or a decimal number that is interpreted as a duration of years, months, or days. The following SQL statement produces a report showing on which month of the year an interview occurred:

```
SELECT MANAGER, DISP, MONTH(INTDATE)
FROM Q.INTERVIEW
```

The report in the following figure shows the months in which certain managers interviewed prospective employees and the disposition of each interview.

MANAGER	DISP	COL1
270	NOHIRE	2
10	HIRE	2
140	HIRE	4
290	NOHIRE	4
160	HIRE	3
50	HIRE	9
100	HIRE	10
270	HIRE	2
160	NOHIRE	3
140	NOHIRE	9

Figure 90. A report produced using the MONTH function

The YEAR function returns the year part of a value. YEAR works like DAY and MONTH. The argument must be a date, timestamp, or a decimal number that is interpreted as a duration of years, months, or days. Consider the following SQL query:

```
SELECT PROJNO, YEAR(ENDD)
FROM Q.PROJECT
WHERE PRODNUM = 190
```

When you run this SQL query, QMF produces the report shown in the following figure:

PROJNO	COL1
1404	1999
1410	2000

Figure 91. A report produced using the YEAR function

This report shows the year portion of project completion dates for a given product. It disregards the day and month.

### Related concepts

[Using durations to represent date/time intervals](#)

A duration is a number that represents an interval of time. The number can be a constant, a column name, a function, or an expression.

## Isolating the hour, minutes, seconds, or fractional seconds of a time

The HOUR function returns the hour part of a value. The argument must be a time, timestamp, or a decimal number that is interpreted as a time.

For example, if you want to query the data in the Q.INTERVIEW table to determine the time that a particular interview ended, you could run a query like the following:

```
SELECT TEMPID, ENDTIME
FROM Q.INTERVIEW
WHERE TEMPID = 400
```

QMF shows the ENDTIME column and the time displays the hours, minutes, and seconds, as shown in the following figure.

TEMPID	ENDTIME
400	15.12.00

Figure 92. A report that shows a specific time an interview ended

To show only the hour that the interview ended, you could run a query like the following:

```
SELECT TEMPID, HOUR(ENDTIME)
FROM Q.INTERVIEW
WHERE TEMPID = 400
```

QMF produces the report shown in the following figure, which shows only the hour portion of ENDTIME:

TEMPID	COL1
400	15

Figure 93. A report showing the hour portion of a timestamp

The MINUTE function returns the minute part of a value. The argument must be a time, timestamp, or a decimal number that is interpreted as a duration of hours, minutes, or seconds.

For example, consider the following SQL statement:

```
SELECT TEMPID, MINUTE(ENDTIME)
FROM Q.INTERVIEW
WHERE TEMPID = 400
```

When you run this query, QMF produces the report shown in the following figure:

TEMPID	COL1
400	12

Figure 94. A report showing the minute portion of a timestamp

The SECOND function returns the seconds part of a value. The argument must be a time, timestamp, or a decimal number that is interpreted as a duration of hours, minutes, or seconds. For example, consider the following SQL statement:

```
SELECT TEMPID, SECOND(ENDTIME)
FROM Q.INTERVIEW
WHERE TEMPID = 400
```

When you run this query, QMF produces the report shown in the following figure:

TEMPID	COL1
400	0

Figure 95. A report showing the seconds portion of a timestamp

The MICROSECOND function returns the part of a value that represents fractional seconds. The argument can only be a column that contains timestamp data. For example, consider the following SQL statement, where the argument for the MICROSECOND function is a column named TIMESTAMP in the Q.PROJECT sample table:

```
SELECT PROJNO, MICROSECOND(TIMESTAMP)
FROM Q.PROJECT
WHERE PROJNO = '1409'
```

When you run this query, QMF produces the report shown in the following figure:

PROJNO	COL1
1409	149572

Figure 96. A report showing the portion of a timestamp that represents fractional seconds

The portion of a timestamp that represents fractional seconds can be from zero to 12 characters. The TIMESTAMP column of the Q.PROJECT sample table has a timestamp precision of six characters.

## Finding the length of a value

The LENGTH function returns the length of a value.

The length of a value depends on its data type, as follows:

- The length of a graphic string is the number of double-byte characters.
- The length of all other values is the number of bytes used to represent the value:
  - 2 for SMALLINT
  - 4 for INTEGER
  - 8 for BIGINT
  - For decimal numbers with a precision of P, the length is the integer part of the following equation:  $(P/2)+1$
  - 16 for DECFLOAT(34)
  - 8 for DECFLOAT(16)
  - 4 for FLOAT(21)
  - 8 for FLOAT(53)
  - 4 for DATE
  - 3 for TIME
  - $7 + ((p+1)/2)$  for TIMESTAMP(p), resulting in lengths from 7 to 13 bytes
  - $9 + ((p+1)/2)$  for TIMESTAMP(p) WITH TIME ZONE, resulting in lengths from 9 to 15 bytes
  - The length of the string (for string values)
  - The length of the row ID (for ROWID values)

The following SQL statement shows the length of the project number (PROJNO) column in the Q.PROJECT sample table.

```
SELECT PROJNO, LENGTH(PROJNO)
FROM Q.PROJECT
WHERE DEPT = 51
```



When you run this statement, QMF produces the report shown in the following figure:

PROJNO	COL1
1403	4
1501	4

Figure 97. The result of a query that finds the length of a value

### Related information

The IBM Publications CenterSearch for information about data types by referring to the SQL reference information provided for your database.

## Displaying parts of a value

The SUBSTR function returns a part of a string.

The format of SUBSTR is:

```
SUBSTR(M,N,L)
```

In this syntax:

- M represents a character, graphic, or binary string to manipulate
- N represents the position of the first character of the substring you want
- L represents the length of the substring to select

The following SQL statement selects a column that contains a last name and a column that contains the first initial of the first name:

```
SELECT LASTNAME, SUBSTR(FIRSTNAME,1,1)
FROM Q.INTERVIEW
WHERE MANAGER = 140
```

When you run this statement, QMF produces the report that is shown in the following figure, where COL1 shows the first initial of each first name:

LASTNAME	COL1
MONTEZ	R
GASPARD	P

Figure 98. Result of a query that uses the SUBSTR function

## Replacing null values in the report with other values

The VALUE function substitutes a non-null value (specified in the second argument) for each null value found in the column designated by the first argument. You must specify two or more arguments, and the data types of the arguments must be comparable.

For example, consider the following SQL statement, which selects a column that contains null values:

```
SELECT COMM
FROM Q.STAFF
WHERE YEARS = 5
```

When you run the above query, QMF produces the report shown in the following figure:

```

      COMM
-----
      -
    206.60
      -
    806.10
    188.00

```

Figure 99. A report that displays the COMM column from Q.STAFF for anyone with 5 years of service

When you use the VALUE function with a second argument of 0 in the SELECT clause, the null values are replaced with 0.00 because the data type is DECIMAL. For example, consider the following query:

```

SELECT VALUE(COMM, 0)
FROM Q.STAFF
WHERE YEARS = 5

```

When you run this query, QMF produces the report shown in the following figure:

```

      COL1
-----
      0.00
    206.60
      0.00
    806.10
    188.00

```

Figure 100. Result of a query that uses the VALUE function

## Nested SQL functions

You can nest built in column and scalar functions within other functions.

You can nest scalar functions within other scalar functions and within column functions. You can nest column functions within scalar functions. You cannot nest column functions within other column functions.

### Nesting scalar functions within scalar functions

Suppose you want to know the month and day of interview for all applicants that are interviewed by manager 140, and you want the result in USA format. The following query shows how you can nest scalar functions to get the information you want:

```

SELECT SUBSTR((CHAR(INTDATE, USA)), 1, 5)
FROM Q.INTERVIEW
WHERE MANAGER = 140

```

When you run this query, QMF produces the report shown in the following figure:

```

COL1
----
04/07
09/30

```

Figure 101. Result of a query that nests scalar functions within other scalar functions

### Nesting scalar functions within column functions

If a column function's argument is a scalar function, the scalar function must include a reference to a column. For example, if you want to know the last year that any project begins and the last year that any project completes, you can run this query:

```

SELECT MAX(YEAR(STARTD)), MAX(YEAR(ENDD))
FROM Q.PROJECT

```

When you run this query, QMF produces the report that is shown in the following figure:

```

COL1          COL2
-----
1999          2000

```

Figure 102. Result of a query that nests scalar functions within column functions

## Nesting column functions within scalar functions

Suppose that you want to know the year in which the last project in Department 20 will be initiated. Consider the following query:

```

SELECT YEAR(MAX(STARTD))
FROM Q.PROJECT
WHERE DEPT = 20

```

When you run this query, QMF produces the report that is shown in the following figure:

```

COL1
-----
1997

```

Figure 103. Result of a query that nests column functions within scalar functions

## Adding and subtracting dates and times

Addition and subtraction are the only operations that work with date/time values.

You can increment or decrement a date, time, or timestamp by a duration. You can subtract a date from a date, or a time from a time. You cannot subtract a timestamp from a timestamp.

### Rules for date/time addition

If a date/time value is the operand of an addition, the other operand must be a duration.

The operands of date/time addition must be as shown in the following table:

If one operand is a:	The other operand must be:
Date	A duration of years, months, or days
Time	A duration of hours, minutes, or seconds
Timestamp	Any valid duration

### Rules for date/time subtraction

Subtracting two date/time values is different from subtracting a duration from a date/time value.

The operands of date/time subtraction follow the rules shown in the following table:

If the first operand is a:	The second operand must be:
Date	A date, string representation of a date, or a duration of years, months, or days
Time	A time, string representation of a time, or a duration of hours, minutes, or seconds

<i>Table 10. Rules for date/time subtraction (continued)</i>	
<b>If the first operand is a:</b>	<b>The second operand must be:</b>
Timestamp	A duration A timestamp can only be the first operand of subtraction.

The following table shows what the first operand must be depending on the data type of the second operand.

<i>Table 11. Rules governed by the second operand in date/time subtraction</i>	
<b>If the second operand is a:</b>	<b>The first operand must be:</b>
Date	A date or string representation of a date
Time	A time or string representation of a time

Because character strings cannot be subtracted, a string representation of a date or time value cannot be subtracted from another string representation of a date or time value. For example, the following expression is not valid:

```
'2006-01-01' - '2005-01-01'
```

However, if you convert one of the strings to a date or time, the expression is valid. For example, the following expression is valid:

```
DATE('2006-01-01') - '2005-01-01'
```

## The number of days between two dates

The DAYS function calculates the number of days between one date and another.

The DAYS function returns an integer representation of a date. The result of DAYS is the number of days since January 1, 0001. The argument can be a date, a timestamp, or a string representation of a date.

You can calculate the number of days between two dates with an equation. In the following equation, &DATE supplies the current date.

```
DAYS (future date) - DAYS (&DATE)
```

The DAYS function returns an integer representation of a date. The result of DAYS is the number of days since January 1, 0001. The argument can be a date, a timestamp, or a string representation of a date. For example, the following statement results in a value of 368 days:

```
DAYS('0002-01-03')
```

## Accounting for months with different numbers of days

Because of inconsistencies in the number of days in the months of the year, adding a month to a date does not always result in the same day of the next month.

The result of adding one month to January 31 cannot be February 31. Adding a month to a date results in the same day of the next month when such a day exists. If it does not exist, adding a month to a date results in the last day of the next month. For example, if you add one month to January 31, the result is February 28 (or February 29 in a leap year).

Use the DAYS function to avoid inconsistencies in date arithmetic that is caused by inconsistencies in the number of days in each month. For example, to increment a date by the difference between two dates, you can use the following SQL query:

```
SELECT DATE(DAYS('1988-01-05') + DAYS(ENDD) - DAYS(STARTD))
FROM Q.PROJECT
WHERE PROJNO = '1408'
```

When you run this query, QMF produces the report that is shown in the following figure:

```
COL1
-----
1989-07-25
```

Figure 104. Result of using the DAYS function to account for months with different days

## Using durations to represent date/time intervals

A duration is a number that represents an interval of time. The number can be a constant, a column name, a function, or an expression.

A duration represents any number of years, months, days, hours, minutes, seconds, or fractional seconds. The unit is expressed by a keyword that follows the number. In the expression STARTD+25 YEARS, the duration is 25 YEARS.

You can use a duration only in an expression that involves a date or time value. For example, STARTD+25 YEARS+1 MONTH is a valid expression. (STARTD is a column in Q.PROJECT that gives the start date of a project.) STARTD+(25 YEARS+1 MONTH) is not a valid expression, because (25 YEARS+1 MONTH) does not include a date or time value within the parentheses.

YEAR(ENDD - STARTD) < 3 YEARS is not valid because you cannot use the duration 3 YEARS as an operand of comparison. A valid way of coding this is YEAR(ENDD - STARTD) < 3.

Subtracting one date from another date results in a duration that is expressed by the number of years, months, and days. Subtracting one time from another time results in a duration that is expressed in the number of hours, minutes, and seconds.

### Related concepts

#### Subtracting dates

The result of date subtraction is a duration that is expressed in years, months, and days between the two dates. A negative result is possible with date subtraction.

#### Subtracting times

If you subtract two times, the result is a duration that represents the number of hours, minutes, and seconds between the two times. A negative result is possible when subtracting two times.

## Incrementing and decrementing dates by durations

In these examples, you use a duration to find out how dates change.

Suppose that you want to know what the start date for project 1404 would be if you delayed it one year. You would increment the current start date (1997-01-04) by using a duration of 1 year, as shown in the following query:

```
SELECT STARTD + 1 YEAR
FROM Q.PROJECT
WHERE PROJNO = '1404'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1998-01-04
```

*Figure 105. Result of incrementing the project start date by 1 year*

The month of the result is the same as the month of the date you are incrementing. The day of the result is the same as that of the date incremented, unless the result is February 29 of a year that is not a leap year. In that case, the day is February 28.

If you want to know what the end date of project 1404 (currently slated for 1999-06-30) would be if you finish the project two months ahead of schedule, run this SQL statement using a duration of 2 months:

```
SELECT ENDD - 2 MONTHS
FROM Q.PROJECT
WHERE PROJNO = '1404'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1999-04-30
```

*Figure 106. Result of decrementing the project end date by a duration of 2 months*

QMF counts only months (calendar pages) and years (if necessary). The day of the result is the same as the day of the date you are decrementing, unless the result would be a date that is not valid. In that case, the day part of the result is the last day of the month.

To find out what the start date of project 1407 would be if the project is started 30 days early, run the following SQL statement using a duration of 30 days:

```
SELECT STARTD - 30 DAYS
FROM Q.PROJECT
WHERE PROJNO = '1407'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1997-11-12
```

*Figure 107. Result of decrementing the project start date by 30 days*

If you want to know what the end date for project 1407 would be if you delay the project by 2 years and 11 months, you could run the following SQL statement:

```
SELECT ENDD + 2 YEARS + 11 MONTHS
FROM Q.PROJECT
WHERE PROJNO = '1407'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
2003-05-15
```

*Figure 108. Result of incrementing the project end date by 2 years and 11 months*

Suppose that, rather than increment by the 2 years and 11 months' duration used in the preceding example, you want to decrement by the same duration. You want to know the project end date if the project were to finish 2 years and 11 months ahead of schedule. You can run the following SQL statement:

```
SELECT ENDD - 2 YEARS - 11 MONTHS
FROM Q.PROJECT
WHERE PROJNO = '1407'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1997-07-15
```

Figure 109. Result of decrementing the project end date by 2 years and 11 months

## Subtracting dates

The result of date subtraction is a duration that is expressed in years, months, and days between the two dates. A negative result is possible with date subtraction.

Suppose that you want to know how many years, months, and days project 1407 will take to complete. You could run the following SQL statement:

```
SELECT ENDD - STARTD
FROM Q.PROJECT
WHERE PROJNO = '1407'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
20603
```

Figure 110. Result of subtracting the project start date from the end date to get a total project duration

The result is a numeric representation of the duration: 2 years, 6 months, and 3 days. You can treat this result like any other numeric value. The duration format for dates is *yyyymmdd*, where *yyyy* represents years, *mm* represents months, and *dd* represents days. Leading zeros are always truncated in the results.

To find out the number of weeks project 1405 would take to complete, run the following SQL statement:

```
SELECT (DAYS(ENDD) - DAYS(STARTD))/7
FROM Q.PROJECT
WHERE PROJNO = '1405'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
130
```

Figure 111. Result of a query to determine the number of weeks the project takes

The result is a duration of 130 weeks.

If you would like to know, in terms of years, how many years it takes to complete project 1403, you can run the following SQL statement:

```
SELECT (DAYS(ENDD) - DAYS(STARTD))/365.24
FROM Q.PROJECT
WHERE PROJNO='1403'
```

When you run this statement, QMF produces the report shown in the following figure:

```

          COL1
-----
          3.31

```

Figure 112. Result of a query to determine the number of years the project takes

## Making durations easier to read

In this example, you separate a date duration into year, month, and day so that you can read it more easily.

Suppose that you run the following SQL statement:

```

SELECT ENDD-STARTD
FROM Q.PROJECT
WHERE PROJNO='1403'

```

When you run this statement, QMF produces the report shown in the following figure:

```

          COL1
-----
          30327

```

Figure 113. Durations can be difficult to read if the years, months, and days are not separated.

The result of this date subtraction is a duration of 3 years, 3 months, and 27 days.

To get results that are easier to read in a report, run the following SQL statement:

```

SELECT YEAR(ENDD - STARTD), MONTH(ENDD - STARTD), DAY(ENDD - STARTD)
FROM Q.PROJECT
WHERE PROJNO='1403'

```

When you run this statement, QMF produces a report like the one in the following figure:

```

          COL1          COL2          COL3
-----
          3            3            27

```

Figure 114. Result showing an easier-to-read duration

## Incrementing and decrementing times by durations

Adding a duration to a time or subtracting a duration from a time results in a time.

The next example increments a time by a duration. It uses the following SQL query to find the start time for an interview if the interview starts 2 hours, 30 minutes, and 45 seconds late:

```

SELECT STARTTIME + 2 HOURS + 30 MINUTES + 45 SECONDS
FROM Q.INTERVIEW
WHERE TEMPID = 400

```

When you run this query, QMF produces the report shown in the following figure:

```

          COL1
-----
          15.30.45

```

Figure 115. Result of a query that increments the interview start time by a duration

The next example decrements a time by a duration. To find out what time an interview would end if it ended 1 hour, 20 minutes, and 20 seconds early, use the following SQL statement:



```
SELECT ENDTIME - 1 HOUR - 20 MINUTES - 20 SECONDS
FROM Q.INTERVIEW
WHERE TEMPID = 410
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
14.57.40
```

Figure 116. Result of a query that decrements the interview end time by a duration

## Subtracting times

If you subtract two times, the result is a duration that represents the number of hours, minutes, and seconds between the two times. A negative result is possible when subtracting two times.

For example, if you want to know how much time an interview for a person with temporary ID 410 took, you can use this SQL statement:

```
SELECT ENDTIME - STARTTIME
FROM Q.INTERVIEW
WHERE TEMPID = 410
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
11800
```

Figure 117. Result of a query that subtracts an interview start time from the interview end time

The result is a numeric representation of the duration: 1 hour, 18 minutes, and 0 seconds. You can treat this result like any other numeric value. The duration format for time is *hhmmss*, where *hh* represents hours, *mm* represents minutes, and *ss* represents seconds. QMF always removes leading zeros from the result.

## Incrementing and decrementing timestamps by durations

The result of adding a duration to a timestamp or subtracting a duration from a timestamp is a timestamp.

The following example increments a timestamp by a duration of 30 fractional seconds:

```
SELECT TIMESTAMP + 30 MICROSECONDS
FROM Q.PROJECT
WHERE PROJNO = '1409'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1996-03-13-09.12.57.149602
```

Figure 118. Result of a query that adds a duration (in fractional seconds) to a timestamp

Suppose that you want to know what the timestamp will be for project 1409 if you add a duration of 2 years, 1 month, and 2 hours to the project's existing timestamp. Use the following SQL statement:

```
SELECT TIMESTAMP + 2 YEARS + 1 MONTH + 2 HOURS
FROM Q.PROJECT
WHERE PROJNO = '1409'
```

When you run this statement, QMF produces the report shown in the following figure:

```
COL1
-----
1998-04-13-11.12.57.149572
```

Figure 119. Result of a query that adds a duration (in years, months, and hours) to a timestamp

The fractional seconds portion of a timestamp can be from zero to 12 characters. The values in the `TIMESTAMP` column of the `Q.PROJECT` sample table have a precision of six characters.

## The concatenation operator

Use the concatenation operator (`||`) to join two values of an expression into a single string. In some non-English, single-byte character sets, the `||` can display as `!!` (exclamation marks) or other special characters.

### Rules for concatenation

The following rules apply to using the concatenation operator:

- The operands that you concatenate must be either all character strings or all graphic strings.
- The length of the result is the sum of the lengths of the operands.
- The data type of the result is:
  - `VARCHAR` when all operands are `CHAR` and their combined length is greater than 255, or when one or more operands is `VARCHAR`.
  - `VARGRAPHIC` when all operands are `GRAPHIC` and their combined length is greater than 255, or when one or more operands is `VARGRAPHIC`.
- If either operand is null, the result is a null value. (To avoid null values, use the `VALUE` scalar function.)
- You cannot use concatenation in a `LIKE` clause.
- You cannot use concatenation in the `SET` clause of an `UPDATE` statement.

### Examples that use concatenation

In the `SELECT` clause of the following SQL statement, the concatenation operator is used with the `SUBSTR` scalar function to join the first character of `FIRSTNAME` with `LASTNAME`.

```
SELECT LASTNAME||SUBSTR(FIRSTNAME,1,1)
FROM Q.INTERVIEW
WHERE MANAGER = 140
```

When you run this statement, QMF produces the report that is shown in the following figure:

```
COL1
-----
MONTEZR
GASPARDP
```

Figure 120. Result of a query that uses the concatenation operator to show the last name and first initial of employees

There is no space between the last name and the initial because the data types for the columns `FIRSTNAME` and `LASTNAME` are defined as `VARCHAR` rather than `CHAR`.

The next example concatenates a substring of the first name with a period and a space, and then with the last name:

```
SELECT SUBSTR(FIRSTNAME,1,1)||'. '||LASTNAME
FROM Q.INTERVIEW
WHERE TEMPID = 400
```

When you run this statement, QMF produces the report that is shown in the following figure:

```
COL1  
-----  
R. FROMMHERZ
```

*Figure 121. Result of a query that uses the concatenation operator to show the first initial and last name in a more readable format*

### Related reference

[SQL scalar functions](#) Three types of SQL scalar functions are date/time functions, conversion functions, and string functions.

## Making your SQL query reusable with substitution variables

When you specify substitution variables in an SQL query, you can use the same query to retrieve different information. You supply a new value for the variable each time you run the query.

You use substitution variables in an SQL query in the same way you use them in a prompted query. Though that topic shows prompted query examples, the concepts of using substitution variables are the same.

The following query selects department data. By using a substitution variable (&DEPARTMENT) for the department number in the row condition, you can specify a different department number each time you run the query.

```
SELECT ID, NAME, JOB, SALARY  
FROM Q.STAFF  
WHERE DEPT=&DEPARTMENT
```

For example, you can enter the name of the variable and its value as a parameter on the RUN QUERY command, as shown here:

```
RUN QUERY (&DEPARTMENT = 38
```

If the query includes multiple SQL statements, any variable value that you supply for the query applies to all SQL statements in the query that include a variable with that name.

### Related concepts

[Making your prompted query reusable with substitution variables](#)

When you specify substitution variables in a prompted query, you can use the same query to retrieve different information. You supply a new value for the variable each time you run the query.

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

## Saving a new query

You can save your query in the database after you create it. You can run a saved query and display the report again. You can also add, delete, or change the information in a saved query.

### Related tasks

[Saving a QUERY, FORM, or PROC object](#)

You can save a query, form, or procedure by entering the SAVE command on the QMF command line.

## Inserting the results of a query into a table

When you run a query, you can insert the results directly into a table rather than having them returned to QMF.

This option can help with storage and performance issues for queries that have large results.

To insert query results into a new or existing table, specify the TABLE option on your RUN QUERY command. You can also optionally specify the ACTION and COMMENT keywords. For example, the following command replaces any existing data in the MYTABLE with the query results and comment:

```
RUN QUERY (TABLE=MYTABLE ACTION=REPLACE COMMENT=(Staff report for departments))
```

Additionally the RUN QUERY command using the TABLE keyword can optionally be configured to use the Db2 cross-loader function.

Note that the Db2 cross-loader function is an extra cost feature available through the LOAD utility found in the Db2 for z/OS Utilities Suite. When using the optional cross-loader functionality of the RUN QUERY command, QMF makes use of the Db2 supplied stored procedure SYSPROC.DSNUTILU. The stored procedure must be installed and configured to use the cross-loader function.

To use the optional cross-loader support for RUN QUERY, reference the global variables DSQEC\_SAV\_LOADER, DSQEC\_SAV\_LOGCTL and DSQEC\_SAV\_LOGTABLE. Be aware the cross-loader does not support using an ACCELERATOR. If you have existing AOT tables, you will get a return code 8 and a result set with error information from the ACCELERATOR. If you specify the DSQEC\_SAV\_ALLOWED global variable options 2,3,or 4 while the DSQEC\_SAV\_LOADER global variable is set to 1, you will get an error message from QMF.

## Previewing reports for SELECT with change statements

By setting the DSQDC\_DISPLAY\_RPT global variable to a value of 1, you can use a QMF mini session which allows you to preview report output from SELECT with change statements.

### About this task

A QMF mini session gives you control for any SQL COMMIT or SQL ROLLBACK statements. You can preview the outcome of your SELECT with change statement and choose whether to save or cancel the report.

The following example is a mini session during which you can choose to accept or COMMIT the report:

From the QMF command line, set the global variable:

```
1=Help      2=Run      3=End      4=Print    5=Chart    6=Draw
7=Backward  8=Forward  9=Form    10=Insert  11=Delete  12=Report
OK, QUERY is displayed.
COMMAND ==>> SET GLOBAL (DSQDC_DISPLAY_RPT = 1) SCROLL ==>> PAGE
```

Run a SELECT with change statement:

```

SQL QUERY                                     LINE 1

SELECT NAME FROM OLD TABLE ( DELETE FROM MYSTAFF
WHERE NAME = 'JONES')

*** END
***

1=Help      2=Run      3=End      4=Print      5=Chart      6=Draw
7=Backward  8=Forward  9=Form     10=Insert    11=Delete    12=Report
OK, SET performed. Please proceed.
COMMAND ==>                                     SCROLL ==> PAGE

```

Run the query to enter the QMF mini session and view the QMF report preview:

```

REPORT                                     LINE 1      POS 1      79

NAME
-----
JONES

*** END ***

1=Help      2=      3=End      4=Print      5=Chart      6=Query
7=Backward  8=Forward  9=Form     10=Left     11=Right     12=
OK, this is the REPORT from your RUN command.
COMMAND ==>                                     SCROLL ==> PAGE

```

When you press PF3 (END), the following confirmation panel is displayed. Note that **1. YES** is specified by default:

```

REPORT                                     LINE 1      POS 1      79

+-----+
|                                     RUN Confirmation                                     |
|                                                                                       |
| WARNING:                                                                              |
| Your RUN command modified this number of rows in the                               |
| database:          1                                                                 |
|                                                                                       |
| Do you want to make this change?                                                    |
|  1. YES - Make the changes permanent in the database.                             |
|  2. NO - Roll back the changes to the last COMMIT operation                       |
|           or to the beginning of the query. The database that you are using       |
|           with QMF determines the rollback rules.                                  |
+-----+
| F1=Help  F12=Cancel                                                                |
+-----+

```

Please follow the directions on the confirmation prompt panel.

Press Enter to accept the report; this processes the COMMIT and finalizes the SELECT with change SQL statement. The row with the column NAME containing the value of JONES will be deleted from the table. The following message displays above the QMF command line so that you can confirm your changes:

```

REPORT                                LINE 1    POS 1    79
NAME
-----
JONES

*** END ***

1=Help      2=      3=End      4=Print      5=Chart      6=Query
7=Backward  8=Forward 9=Form     10=Left     11=Right     12=
OK, number of rows modified in the database: 1.
COMMAND ==> SCROLL ==> PAGE

```

The following example is a SELECT with change statement in which you decide not to save changes and instead issue a ROLLBACK.

Set the global variable:

```

SQL QUERY                                LINE 1

*** END ***

1=Help      2=Run      3=End      4=Print      5=Chart      6=Draw
7=Backward  8=Forward  9=Form     10=Insert    11=Delete    12=Report
OK, cursor positioned.
COMMAND ==> SET GLOBAL (DSQDC_DISPLAY_RPT = 1 SCROLL ==> PAGE

```

Run a query to insert a row into table MYSTAFF:

```

SQL QUERY                                TS5519.SELECT_MERGE_Q    LINE 1

SELECT NAME FROM FINAL TABLE (
MERGE INTO MYSTAFF AR
USING (VALUES
(4, 'KRAMER',20, 'CLERK',8,18171.25,612.45))
AS
AC(ID, "NAME", DEPT, JOB, "YEARS", SALARY, COMM)
ON AR.ID = AC.ID
WHEN MATCHED THEN
UPDATE SET AR.JOB = AC.JOB
WHEN NOT MATCHED THEN
INSERT (ID, "NAME", DEPT, JOB, "YEARS", SALARY, COMM)
VALUES (AC.ID,AC."NAME",AC.DEPT,AC.JOB,AC."YEARS",AC.SALARY,AC.COMM)
NOT ATOMIC CONTINUE ON SQLEXCEPTION)
*** END ***

1=Help      2=Run      3=End      4=Print      5=Chart      6=Draw
7=Backward  8=Forward  9=Form     10=Insert    11=Delete    12=Report
OK, cursor positioned.
COMMAND ==> SCROLL ==> PAGE

```

Run the query to see a preview of the report that would be generated:

```

REPORT                                     LINE 1      POS 1      79
      NAME
      -----
      KRAMER
*** END ***

1=Help      2=          3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward   9=Form    10=Left   11=Right   12=
OK, this is the REPORT from your RUN command.
COMMAND ==>                                SCROLL ==> PAGE

```

After reviewing the report, if you do not want to keep this change, specify option **2. NO** (ROLLBACK) on the confirmation prompt when you end the QMF report mini session (PF3 END).

```

REPORT                                     LINE 1      POS 1      79
      +-----+
      |          RUN Confirmation          |
      | WARNING:                          |
      | Your RUN command modified this number of rows in the |
      | database:          1              |
      |                               |
      | Do you want to make this change? |
      | 2 1. YES - Make the changes permanent in the database. |
      |   2. NO - Roll back the changes to the last COMMIT operation |
      |               or to the beginning of the query. The database that you are using |
      |               with QMF determines the rollback rules.     |
      |-----+
      | F1=Help  F12=Cancel                |
      +-----+

```

Please follow the directions on the confirmation prompt panel.

The informational message above the command line confirms that your change was canceled:

```

REPORT                                     LINE 1      POS 1      79
      NAME
      -----
      KRAMER
*** END ***

1=Help      2=          3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward   9=Form    10=Left   11=Right   12=
The END command was executed and changes discarded.
COMMAND ==>                                SCROLL ==> PAGE

```

**Tip:** If you want to return to the mini session without committing or rolling back your changes, press F12 instead of selecting option **1. YES** or **2. NO**.





## Chapter 6. Customizing your reports

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

### About the QMF form panels

You can change the appearance of your reports by changing formatting specifications on the QMF form panels. You specify a different aspect of the formatting of your report on each panel.

The following figure shows the QMF form panels and their purposes.

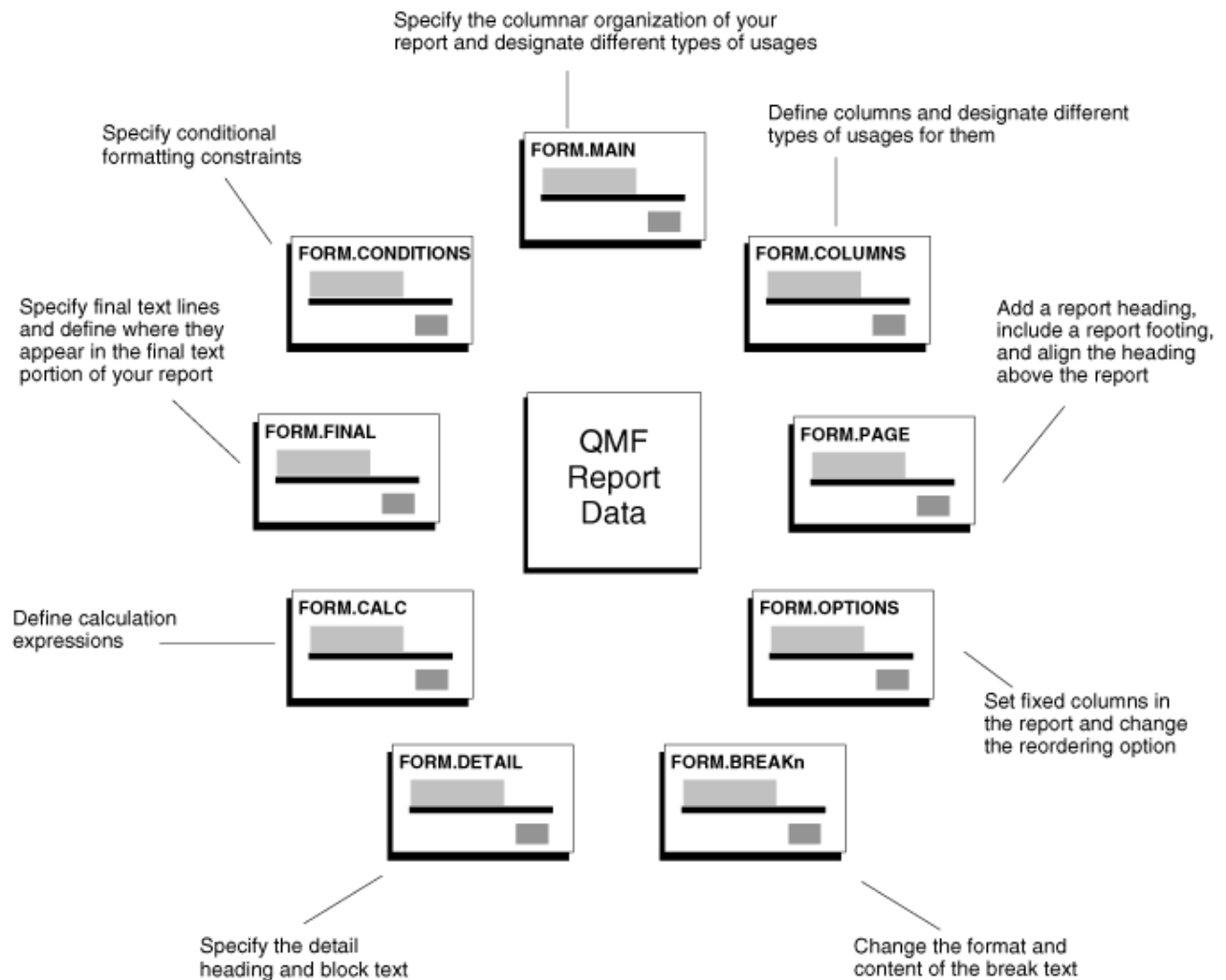


Figure 122. QMF creates reports from forms where you specify how to format the data.

Display form panels by using either the SHOW command or the DISPLAY command. For example, to display FORM.MAIN, enter one of the following commands:

```
SHOW FORM.MAIN      o r   SH F.M
DISPLAY FORM.MAIN   o r   DI FORM.MAIN
```

You can also enter the following:

```
FORM.MAIN  o r   F.M
```

Then, press the Show function key. You can use the abbreviated form panel name when using the SHOW command. For a complete list of the shortened form panel names, enter:

```
SHOW ?
```

## The default report format

QMF includes a default report format. You can use the QMF form panels to show the same data in different ways.

When you display a report by using the default report format, the report looks like the one in the following figure.

NAME	DEPT	JOB	SALARY	COMM
KERMISCH	15	CLERK	12258.50	110.10
NGAN	15	CLERK	12508.20	206.60
ROTHMAN	15	SALES	16502.83	1152.00
JAMES	20	CLERK	13504.60	128.20
PERNAL	20	SALES	18171.25	612.45
SNEIDER	20	CLERK	14252.75	126.50
ABRAHAMS	38	CLERK	12009.75	236.50
NAUGHTON	38	CLERK	12954.75	180.00
O'BRIEN	38	SALES	18006.00	846.55
QUIGLEY	38	SALES	16808.30	650.25

Figure 123. QMF uses a default report format like this one.

By tailoring the report format, you can use the same data to produce the report in the following figure.

DIVISION EARNINGS REPORT				
DEPT. NUMBER	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
	NGAN	\$12,508.20	\$206.60	\$12,714.80
	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
DEPT. 15	TOTALS	\$41,269.53	\$1,468.70	\$42,738.23
20	JAMES	\$13,504.60	\$128.20	\$13,632.80
	PERNAL	\$18,171.25	\$612.45	\$18,783.70
	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
DEPT. 20	TOTALS	\$45,928.60	\$867.15	\$46,795.75
38	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55
DEPT. 38	TOTALS	\$59,778.80	\$1,913.30	\$61,692.10
		=====	=====	=====
		\$146,976.93	\$4,249.15	\$151,226.08
COMPANY NAME				

Figure 124. A customized report can show the same data in different ways.

## Report columns

You can change aspects of columns that are displayed in your reports.

### Changing the columns on your report

An example that shows how to display the default FORM.COLUMNS panel. You can use this panel to change the columns of your report.

#### About this task

The first step in producing the report that is shown in [Figure 124 on page 102](#) is to change the appearance of the columns.

#### Procedure

To display the default FORM.COLUMNS panel, follow these steps:

1. Retrieve the data to display on the report by using either a prompted query or an SQL query. For example, the following figure shows the prompted query for this procedure.

```
PROMPTED QUERY                                MODIFIED LINE 1

Tables:
Q.STAFF 1

Columns: 2
NAME
DEPT
JOB
SALARY
COMM

Row Conditions: 3
If DEPT Is Equal To 15, 20 or 38
And JOB Is Not Equal To 'MGR'

Sort: 4
Ascending by DEPT
Ascending by NAME
```

Figure 125. This query produces the data for the report.

- 1 QMF retrieves the data from the Q.STAFF table.
- 2 QMF displays these columns on the report.
- 3 The employees are non-managers from Departments 15, 20, and 38.
- 4 QMF orders the rows by department number and then by name.

You can save this query to help you with the examples. For example, on the QMF command line, enter:

```
SAVE QUERY AS NEWQUERY
```

2. Run the query to display the report.
3. On the QMF command line, enter `SHOW FORM.MAIN`.

The FORM.MAIN panel displays (as shown in the following figure) with the default report format for this report:

```

FORM.MAIN

COLUMNS:          Total Width of Report Columns: 50
NUM COLUMN HEADING          USAGE  INDENT  WIDTH  EDIT  SEQ
-----
 1 NAME                    2      9     C     1
 2 DEPT                    2      6     L     2
 3 JOB                      2      5     C     3
 4 SALARY                   2     10    L2    4
 5 COMM                     2     10    L2    5

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=Check    3=End        4=Show    5=Chart    6=Query
7=Backward 8=Forward  9=          10=Insert 11=Delete 12=Report
OK, FORM.MAIN is shown.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 126. You can change column appearance on the QMF FORM.MAIN panel.

You can change the columns on the FORM.MAIN panel. However, in this example, you use the FORM.COLUMNS panel to make all the changes to the columns on your report.

#### 4. Enter SHOW FORM.COLUMNS.

The default FORM.COLUMNS panel for Q.STAFF is shown in the following figure:

```

FORM.COLUMNS

COLUMNS:          Total Width of Report Columns: 50
NUM COLUMN HEADING          USAGE  INDENT  WIDTH  EDIT  SEQ
-----
 1 NAME                    2      9     C     1
 2 DEPT                    2      6     L     2
 3 JOB                      2      5     C     3
 4 SALARY                   2     10    L2    4
 5 COMM                     2     10    L2    5
  *** END ***

1=Help    2=Check    3=End        4=Show    5=Chart    6=Query
7=Backward 8=Forward  9=Specify   10=Insert 11=Delete 12=Report
OK, CANCEL command executed successfully.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 127. The FORM.COLUMNS panel

## Adding a column to a report

You can define a new column on the FORM.COLUMNS panel similar to the way you define a new column in a query. In this example, you add a total earnings (SALARY+COMM) column for each employee.

### Before you begin

Display the FORM panels as described in [“Changing the columns on your report”](#) on page 103.

### Procedure

1. On the **FORM.COLUMNS** panel, move the cursor to the column heading after which you want to add the new column. In this example, move the cursor to the line for the COMM column.

2. Press the Insert function key. A new column number displays.
3. Type the column heading for the new column. For this example, type TOTAL\_EARNINGS. Use an underscore to split the heading between two lines when it appears on the report.
4. Press the Specify function key. The **Specify** panel displays.
5. Select **Definition**. The **Definition** panel for the column displays, as shown in the following figure.
6. Press Enter.

```

FORM.COLUMNS                                MODIFIED
+-----+-----+
|                                     Definition                                     |
| Column Number :      6                                                         |
| Column Heading:  TOTAL_EARNINGS                                                |
| Type an expression to define this column.                                     |
| Expression ( &4_+_&5_____ )                                                 |
| Pass Nulls? ( _NO_____ )                                                    |
+-----+-----+
| F1=Help  F5=Previous Column  F6=Next Column                                   |
| F10=Previous Definition  F11=Next Definition  F12=Cancel                       |
+-----+-----+

```

Figure 128. The **Definition** panel

7. Type the expression that you want to use to define this column. For this example, type &4+&5, which specifies that the value in this column is the sum of the value in column 4 and the value in column 5 (SALARY+COMM).
8. Leave NO in the **Pass Nulls** field to process null values for the expression in this example.  
You can process null values if you are defining a column by using REXX execs.
9. Press Enter.  
You finished defining the new column.
10. Press the Cancel function key to close the **Specify** panel.
11. Press the Report function key to display the changed report, which is shown in the following figure.

NAME	DEPT	JOB	SALARY	COMM	TOTAL EARNINGS
KERMISCH	15	CLERK	12258.50	110.10	12368.60
NGAN	15	CLERK	12508.20	206.60	12714.80
ROTHMAN	15	SALES	16502.83	1152.00	17654.83
JAMES	20	CLERK	13504.60	128.20	13632.80
PERNAL	20	SALES	18171.25	612.45	18783.70
SNEIDER	20	CLERK	14252.75	126.50	14379.25
ABRAHAMS	38	CLERK	12009.75	236.50	12246.25
NAUGHTON	38	CLERK	12954.75	180.00	13134.75
O'BRIEN	38	SALES	18006.00	846.55	18852.55
QUIGLEY	38	SALES	16808.30	650.25	17458.55

Figure 129. The changed report shows the new column.

### Related concepts

Using REXX with QMF forms Expressions used in FORM.CALC, FORM.CONDITIONS, and FORM.COLUMNS (Column Definition) can consist of terms (strings, symbols, and functions) interspersed with operators and parentheses.

## Changing the order in which columns are displayed

Change the order in which the columns are displayed in your report by changing the sequence, or **SEQ**, field for that column.

### Before you begin

Display the FORM panels as described in [“Changing the columns on your report” on page 103](#).

## Procedure

For this example, you change the order of the columns to DEPT, JOB, NAME, SALARY, COMM, and TOTAL EARNINGS:

1. Type the new sequence number over the existing number for each column.

In this example, type the following:

- 3 for the NAME sequence number
- 1 for the DEPT sequence number
- 2 for the JOB sequence number

The following figure shows the SEQ column with the new order.

FORM.COLUMNS	MODIFIED						
NUM	COLUMN	HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
Total Width of Report Columns: 62							
1	NAME			2	9	C	3
2	DEPT			2	6	L	1
3	JOB			2	5	C	2
4	SALARY			2	10	L2	4
5	COMM			2	10	L2	5
6	TOTAL_EARNINGS			2	10	C	6
*** END ***							

Figure 130. Change the order of columns by changing the SEQ field.

2. Press the Report function key to see the changed report, which is shown in the following figure.

DEPT	JOB	NAME	SALARY	COMM	TOTAL EARNINGS
15	CLERK	KERMISCH	12258.50	110.10	12368.60
15	CLERK	NGAN	12508.20	206.60	12714.80
15	SALES	ROTHMAN	16502.83	1152.00	17654.83
20	CLERK	JAMES	13504.60	128.20	13632.80
20	SALES	PERNAL	18171.25	612.45	18783.70
20	CLERK	SNEIDER	14252.75	126.50	14379.25
38	CLERK	ABRAHAMS	12009.75	236.50	12246.25
38	CLERK	NAUGHTON	12954.75	180.00	13134.75
38	SALES	O'BRIEN	18006.00	846.55	18852.55
38	SALES	QUIGLEY	16808.30	650.25	17458.55

Figure 131. The changed report shows the columns in their new order.

## Changing column headings

When you display a report with the default report format, QMF assigns each column a heading. The DSQDC\_COL\_LABELS global variable controls whether the column heading shown in QMF forms defaults to the database label assigned to the column or the name of the column in the table from which it was selected.

### About this task

QMF also gives default names to columns that you define in a query (for example, if you define a new column by specifying an expression or an SQL function that operates on existing columns). In Db2 for z/OS, this default name begins with COL and is followed by a number so that each column name is unique. You can change these default names on the FORM.COLUMNS panel.

## Procedure

In this example, you will change the headings for the NAME, DEPT, and COMM columns:

1. Type the new heading over the existing heading. Use an underscore to split the heading between two lines.

For this example, type:

- EMPLOYEE\_NAME over NAME
- DEPT .\_NUMBER over DEPT
- COMMISSIONS over COMM

These changes are shown in the following figure:

FORM.COLUMNS		MODIFIED				
NUM	COLUMN HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
Total Width of Report Columns: 62						
1	EMPLOYEE_NAME		2	9	C	3
2	DEPT ._NUMBER		2	6	L	1
3	JOB		2	5	C	2
4	SALARY		2	10	L2	4
5	COMMISSIONS		2	10	L2	5
6	TOTAL_EARNINGS		2	10	C	6
*** END ***						

Figure 132. Change the column headings on FORM.COLUMNS.

2. Press the Report function key to see the changed report, as shown in the following figure.

DEPT. NUMBER	JOB	EMPLOYEE NAME	SALARY	COMMISSION	TOTAL EARNINGS
15	CLERK	KERMISCH	12258.50	110.10	12368.60
15	CLERK	NGAN	12508.20	206.60	12714.80
15	SALES	ROTHMAN	16502.83	1152.00	17654.83
20	CLERK	JAMES	13504.60	128.20	13632.80
20	SALES	PERNAL	18171.25	612.45	18783.70
20	CLERK	SNEIDER	14252.75	126.50	14379.25
38	CLERK	ABRAHAMS	12009.75	236.50	12246.25
38	CLERK	NAUGHTON	12954.75	180.00	13134.75
38	SALES	O'BRIEN	18006.00	846.55	18852.55
38	SALES	QUIGLEY	16808.30	650.25	17458.55

Figure 133. The changed report shows the new column headings.

### Related tasks

#### Creating a column by defining expressions with Prompted Query

You can create a column in your report by defining an expression. An expression adds, subtracts, multiplies, or divides the values in two or more columns and places the results in a new column.

### Related reference

Global variables that control various displays DSQDC global variables control the display of certain kinds of information. All of these global variables can be modified by the SET GLOBAL command.

## Changing column widths and space between columns

To change the column width and the spacing between columns on the FORM.COLUMNS panel, move the cursor to the column whose width you want to change and type the new width under the **WIDTH** field. Be sure to include room for punctuation, such as dollar signs, commas, and decimal points.

In the report shown in [Figure 133 on page 107](#), the last letter of the COMMISSIONS column heading does not appear because the column is not wide enough.

For this example, use the following widths:

- 12 for the column width of the SALARY column
- 11 for the column width of the COMMISSIONS column
- 12 for the column width of the TOTAL\_EARNINGS column

The maximum width of a column in a QMF report is 32,767 characters. To display data from columns wider than this value, you can set the WIDTH field to 32767 and assign an edit code that wraps the data on subsequent lines of the report instead of truncating it.

To change the column spacing, move the cursor to the column you want to shift to the right. Then type the new spacing under the **INDENT** field.

For this example, type 4 for the spacing of the EMPLOYEE\_NAME, JOB, and TOTAL\_EARNINGS columns, as shown in the following figure.

FORM.COLUMNS		MODIFIED					
NUM	COLUMN HEADING	Total Width of Report Columns: 73	USAGE	INDENT	WIDTH	EDIT	SEQ
1	EMPLOYEE_NAME		4	9	C	3	
2	DEPT._NUMBER		2	6	L	1	
3	JOB		4	5	C	2	
4	SALARY		2	12	L2	4	
5	COMMISSIONS		2	11	L2	5	
6	TOTAL_EARNINGS		4	12	C	6	
	*** END ***						

Figure 134. Changing the width and spacing of columns on FORM.COLUMNS

Press the Report function key to see the changed report, which is shown in the following figure.

DEPT. NUMBER	JOB	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	CLERK	KERMISCH	12258.50	110.10	12368.60
15	CLERK	NGAN	12508.20	206.60	12714.80
15	SALES	ROTHMAN	16502.83	1152.00	17654.83
20	CLERK	JAMES	13504.60	128.20	13632.80
20	SALES	PERNAL	18171.25	612.45	18783.70
20	CLERK	SNEIDER	14252.75	126.50	14379.25
38	CLERK	ABRAHAMS	12009.75	236.50	12246.25
38	CLERK	NAUGHTON	12954.75	180.00	13134.75
38	SALES	O'BRIEN	18006.00	846.55	18852.55
38	SALES	QUIGLEY	16808.30	650.25	17458.55

Figure 135. The changed report shows the new widths and spacing of columns.

### Related concepts

#### [Specifying punctuation for the values in a column](#)

You can use edit codes in QMF to specify how to punctuate the values in a column. An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report. You specify edit codes on the FORM.COLUMNS panel.

### Related reference

[FORM.COLUMNS](#) Use FORM.COLUMNS to make choices about the uses of the columns. What you specify on FORM.COLUMNS is reflected on FORM.MAIN.

## Changing the alignment of the column heading and data

You change the alignment of the column heading and the data in much the same way you define a new column. In this example, you change the data alignment to CENTER for the DEPT.\_NUMBER column.

### Before you begin

Display the FORM panels as described in [“Changing the columns on your report”](#) on page 103.



## Procedure

1. On the FORM.COLUMNS panel, move the cursor to the column whose alignment you want to change. For this example, move the cursor to the line for the DEPT.\_NUMBER column.
2. Press the Specify function key.  
The **Specify** panel displays.
3. Select **Alignment**. The **Alignment** panel for the column displays.
4. Type the alignment for the data in the column (LEFT, CENTER, RIGHT, or DEFAULT). For this example, type CENTER, as shown in the following figure.

```
+-----+
|                                     |
|                               Alignment |
| Column Number :      2              |
| Column Heading:   DEPT._NUMBER     |
| Heading Alignment ( DEFAULT )      |
| Data Alignment   ( CENTER )        |
+-----+
| F1=Help  F5=Previous Column  F6=Next Column  F12=Cancel |
+-----+
```

Figure 136. The **Alignment** panel

5. Press Enter.  
You finished changing the alignment of the column.
6. Press the Cancel function key to close the **Specify** panel.
7. Press the Report function key to display the changed report, which is shown in the following figure.

REPORT		LINE 1	POS 1	79	
DEPT. NUMBER	JOB	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	CLERK	KERMISCH	12258.50	110.10	12368.60
15	CLERK	NGAN	12508.20	206.60	12714.80
15	SALES	ROTHMAN	16502.83	1152.00	17654.83
20	CLERK	JAMES	13504.60	128.20	13632.80
20	SALES	PERNAL	18171.25	612.45	18783.70
20	CLERK	SNEIDER	14252.75	126.50	14379.25
38	CLERK	ABRAHAMS	12009.75	236.50	12246.25
38	CLERK	NAUGHTON	12954.75	180.00	13134.75
38	SALES	O'BRIEN	18006.00	846.55	18852.55
38	SALES	QUIGLEY	16808.30	650.25	17458.55

Figure 137. The changed report shows the centered data in the DEPT. NUMBER column.

## Specifying punctuation for the values in a column

You can use edit codes in QMF to specify how to punctuate the values in a column. An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report. You specify edit codes on the FORM.COLUMNS panel.

QMF assigns a default edit code to each column in the report that is based on the data type. You can change the default. For example, you can use an edit code to specify a currency symbol, a decimal point, and commas for numeric data.

Default edit codes may also be overridden by setting the following global variables:

1. DSQDC\_EC\_DATE
2. DSQDC\_EC\_TIME
3. DSQDC\_EC\_CHAR
4. DSQDC\_EC\_NUM

## 5. DSQDC\_EC\_DEC

See [QMF global variables](#) for more information on these global variable settings.

You can follow a numeric data edit code (such as L, D, P, or K) with a number that indicates the number of decimal places to use for that data. This number can range from 0 to 99. For example, L2 means to display a numeric value by using the L edit code, and allow two digits after a decimal.

Here are some common edit codes:

### **C**

Character data – specifies no punctuation

### **L**

Numeric data – specifies a decimal point and negative sign, if they occur

### **D**

Numeric data – specifies a currency symbol and a separator for groups of three digits, as well as a decimal point and negative sign, if they occur

### **P**

Numeric data – specifies numeric data as a percentage by using the % symbol, as well as a decimal point and negative sign, if they occur

### **K**

Numeric data – supplies a minus sign for negative values, a separator for groups of three digits, and decimal placement

### **M**

Metadata – specifies that the data type and length of the column is shown rather than actual column data. This code is the default edit code for columns that contain XML, BINARY, VARBINARY, BLOB, CLOB, or DBCLOB data. The ability to change the edit code for LOB data is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default.

M is also the default for DECFLOAT data if the processor on which QMF is running does not support decimal floating-point instructions; in this case, the edit code cannot be changed.

## **Suppressing zero values**

With numeric data edit codes, you can also choose to use a Z edit code in the second position to suppress zero values in a report. For example, DZ indicates numeric data, zero suppression, with a currency symbol, a separator for groups of three digits, and a decimal point and negative sign, if they occur.

## **Specifying a currency symbol**

With the D edit code, you can also choose to use a C edit code in the second or third position. The C edit code causes QMF to use the currency symbol that you specify with the DSQDC\_CURRENCY global variable.

For example, DC indicates numeric data, with the currency symbol that is specified with the DSQDC\_CURRENCY global variable, a separator for groups of three digits, and a decimal point and negative sign, if they occur.

If you use both Z and C with the D edit code, C must follow Z.

## **Changing edit codes**

In this example, you change the edit codes for the SALARY, COMMISSIONS, and TOTAL\_EARNINGS columns to display the values as dollar amounts.

1. On the FORM.COLUMNS panel, move the cursor under the **EDIT** field for the column you want to change.
2. Type the new edit code.

In this example, type D2 in the SALARY, COMMISSIONS, and TOTAL\_EARNINGS columns, as shown in the following figure. The D2 edit code tells QMF to punctuate the values in these columns with a currency symbol, and to allow two digits after the decimal.

FORM.COLUMNS		MODIFIED				
NUM	COLUMN HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
Total Width of Report Columns: 73						
1	EMPLOYEE_NAME	4	9		C	3
2	DEPT._NUMBER	2	6		L	1
3	JOB	4	5		C	2
4	SALARY	2	12		D2	4
5	COMMISSIONS	2	11		D2	5
6	TOTAL_EARNINGS	4	12		D2	6
*** END ***						

Figure 138. Change the way columns are punctuated on FORM.COLUMNS.

3. Press the Report function key to see the changed report, which is shown in the following figure.

DEPT. NUMBER	JOB	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	CLERK	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
15	CLERK	NGAN	\$12,508.20	\$206.60	\$12,714.80
15	SALES	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
20	CLERK	JAMES	\$13,504.60	\$128.20	\$13,632.80
20	SALES	PERNAL	\$18,171.25	\$612.45	\$18,783.70
20	CLERK	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
38	CLERK	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
38	CLERK	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
38	SALES	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
38	SALES	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55

Figure 139. The changed report shows the dollar sign punctuation.

The default currency symbol displays. You can specify a different currency symbol to use in the report by using the currency symbol edit code as explained in the following procedure. Check with your administrator to be sure that both GDDM and the display device (or terminal emulation software) that you are using is configured to support the currency symbol that you need to use.

To change the default currency symbol to the euro in this example, complete these steps:

1. On the command line, issue the SET GLOBAL command to set the DSQDC\_CURRENCY global variable to the symbol that you want to display. If you require a currency symbol that is not represented on the keyboard, you can specify the symbol by setting the DSQDC\_CURRENCY variable in a procedure with logic to the hexadecimal value that is equivalent to the correct symbol. For example, the following procedure sets the currency symbol to HEX '9F', which specifies the euro currency symbol in English QMF:

```
/* */
"SET GLOBAL (DSQDC_CURRENCY =" '9F'X
```

The currency symbol can be a string with a length from 1 to 18 bytes.

2. On the FORM.COLUMNS panel, change the column width for the TOTAL\_EARNINGS column to 13.
3. Change the edit code for TOTAL\_EARNINGS to DC2. The C in the edit code causes QMF to display the currency symbol you defined with the SET GLOBAL command.
4. Press the Report function key to see the changed report, which is shown in the following figure.

DEPT. NUMBER	JOB	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	CLERK	KERMISCH	€12,258.50	€110.10	€12,368.60
15	CLERK	NGAN	€12,508.20	€206.60	€12,714.80
15	SALES	ROTHMAN	€16,502.83	€1,152.00	€17,654.83
20	CLERK	JAMES	€13,504.60	€128.20	€13,632.80
20	SALES	PERNAL	€18,171.25	€612.45	€18,783.70
20	CLERK	SNEIDER	€14,252.75	€126.50	€14,379.25
38	CLERK	ABRAHAMS	€12,009.75	€236.50	€12,246.25
38	CLERK	NAUGHTON	€12,954.75	€180.00	€13,134.75
38	SALES	O'BRIEN	€18,006.00	€846.55	€18,852.55
38	SALES	QUIGLEY	€16,808.30	€650.25	€17,458.55

Figure 140. The changed report shows the euro punctuation.

If you want a space between the currency symbol and the currency values, reissue the SET GLOBAL command so that the value of the variable includes the space.

To change the currency symbol to a dollar sign, issue this command:

```
SET GLOBAL (DSQDC_CURRENCY = $
```

## Usage Notes

You can override the default formatting by setting the following global variables:

- DSQDC\_EC\_DATE
- DSQDC\_EC\_TIME
- DSQDC\_EC\_CHAR
- DSQDC\_EC\_NUM
- DSQDC\_EC\_DEC

### Related concepts

**Edit codes** An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

### Related tasks

Using **GDDM services to handle printing** You can use GDDM (rather than QMF) services to handle printing in native z/OS batch, TSO, ISPF, and CICS.

## Specifying the way QMF uses values in a column

On the FORM.COLUMNS panel, you can specify usage codes to display the values in a column in a way that is meaningful to you. A usage code is a set of characters that tells QMF what to do with the values in a column when it displays them in the report.

### Before you begin

Display the FORM panels as described in [“Changing the columns on your report” on page 103](#).

### About this task

For example, you can sum a column of numbers and display the total, or break the report at certain values to calculate subtotals.

In this example, you sum a column of numbers, omit a column from the report, and calculate subtotals.

### Procedure

1. Move the cursor to the **USAGE** field for that column.
2. Type the usage code.

For this example, type:

- SUM in the **USAGE** field for the SALARY, COMMISSIONS, and TOTAL\_EARNINGS columns
- OMIT for the JOB column

These changes are shown in the following figure:

FORM.COLUMNS		MODIFIED						
NUM	COLUMN HEADING	Total Width of Report	Columns: 64	USAGE	INDENT	WIDTH	EDIT	SEQ
1	EMPLOYEE_NAME				4	9	C	3
2	DEPT._NUMBER				2	6	L	1
3	JOB			OMIT	4	5	C	2
4	SALARY			SUM	2	12	D2	4
5	COMMISSIONS			SUM	2	11	D2	5
6	TOTAL_EARNINGS			SUM	4	12	D2	6
*** END ***								

Figure 141. Specifying usage codes to change the way QMF displays columns

3. Press the Report function key to display the changed report, which is shown in the following figure.

DEPT. NUMBER	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
15	NGAN	\$12,508.20	\$206.60	\$12,714.80
15	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
20	JAMES	\$13,504.60	\$128.20	\$13,632.80
20	PERNAL	\$18,171.25	\$612.45	\$18,783.70
20	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
38	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
38	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
38	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
38	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55
		=====	=====	=====
		\$146,976.93	\$4,249.15	\$151,226.08

Figure 142. The changed report sums the columns and omits the JOB column.

### Related concepts

**Usage codes** QMF usage codes can be entered in the USAGE field on QMF FORM.MAIN or FORM.COLUMNS to define how to use column data to produce reports and charts.

## When QMF displays column metadata instead of the actual data

The default edit code for BLOB, CLOB, DBCLOB, XML, BINARY, and VARBINARY data is M. This is also the default edit code for DECFLOAT data when QMF is running on a processor that does not support decimal floating-point instructions.

The M edit code displays the column metadata (the data type and length) instead of the actual data. This edit code allows you to use these data types in a report without having to recast them to other data types. With the exception of DECFLOAT data, you can view the actual column data by modifying FORM.MAIN or FORM.COLUMNS and changing the edit code for the column. The ability to change the edit code for LOB data in your environment is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default. If the data type data in the column is greater than 32 KB, use a wrapping edit code such as CW so that QMF wraps data that does not fit the display width instead of truncating it.

If you are working with XML or LOB data and you receive out-of-storage errors while using an edit code other than M, you can change the edit code to M to clear the error and display the report.

### Related concepts

**Edit codes** An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

## Adding subtotals to a report

Adding subtotals to a report can help make the report easier to read and understand.

To add subtotals to a report, you need to:

- Specify where you want to break the report to show the subtotals.
- Specify text for the subtotal lines that will appear at the breaks.

### Breaking on a value to add subtotals

To include subtotals in a report, specify a BREAK usage code that tells QMF where to make a break in the report and calculate a subtotal.

#### Before you begin

Display the FORM panels as described in [“Changing the columns on your report” on page 103](#).

#### About this task

The BREAKn usage code divides the report whenever the value of the associated column changes. Control breaks are the points at which a report breaks. Control columns are the columns that govern control breaks. You can have up to six breaks in a report. To indicate a break, type the word BREAK and follow it with a number from 1 to 6.

In this example, you add a break to one column to divide the report every time the department number changes. Because you want to show subtotals by department (that is, calculate a subtotal whenever the department number changes), specify the break usage code in the DEPT.\_NUMBER column. Therefore, the DEPT.\_NUMBER column is the control column.

#### Procedure

1. On the **FORM.COLUMNS** panel, move the cursor to the **USAGE** field for the column you want to use as the control column.
2. For this example, type BREAK1 for the DEPT.\_NUMBER column, as shown in the following figure.

FORM.COLUMNS		MODIFIED				
NUM	COLUMN HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
-----						
1	EMPLOYEE_NAME		4	9	C	3
2	DEPT._NUMBER	BREAK1	2	6	L	1
3	JOB	OMIT	4	5	C	2
4	SALARY	SUM	2	12	D2	4
5	COMMISSIONS	SUM	2	11	D2	5
6	TOTAL_EARNINGS	SUM	4	12	D2	6
*** END ***						

Figure 143. Create breaks for subtotals in your report on FORM.COLUMNS.

3. Press the Report function key to display the changed report, which is shown in the following figure.

DEPT. NUMBER	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
	NGAN	\$12,508.20	\$206.60	\$12,714.80
	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
	*	\$41,269.53	\$1,468.70	\$42,738.23
20	JAMES	\$13,504.60	\$128.20	\$13,632.80
	PERNAL	\$18,171.25	\$612.45	\$18,783.70
	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
	*	\$45,928.60	\$867.15	\$46,795.75
38	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55
	*	\$59,778.80	\$1,913.30	\$61,692.10
		=====	=====	=====
		\$146,976.93	\$4,249.15	\$151,226.08

Figure 144. The changed report shows a subtotal after each department.

### Related reference

[BREAK usage codes](#)The BREAK usage codes provide six levels of breaks (or groupings) in a report.

## Specifying text for the subtotal line on a report

If you do not specify the text that you want to display, asterisks display at each subtotal line.

### Procedure

To specify the text you want to display at each subtotal line in your report:

1. Enter `SHOW FORM.BREAKn`, where *n* is the number you specified on the **FORM.COLUMNS** panel for this break.

In “[Breaking on a value to add subtotals](#)” on page 114 you specified `BREAK1` for the `DEPT._NUMBER` column, so enter the following on the QMF command line:

```
SHOW FORM.BREAK1
```

The **FORM.BREAK1** panel displays.

2. Type the text you want to display at each subtotal line in the **BREAK 1 FOOTING TEXT** field.

For this example, type `DEPT. &2 TOTALS`, as shown in the following figure.

```
FORM.BREAK1

New Page for Break?      ==> NO      Repeat Detail Heading?  ==> NO
Blank Lines Before Heading ==> 0      Blank Lines After Heading ==> 0
LINE  ALIGN  BREAK 1 HEADING TEXT
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+
1     LEFT
2     LEFT
3     LEFT

*** END ***

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

*** END ***
```

Figure 145. Enter the subtotal text in the **BREAK 1 FOOTING TEXT** field.

The &2 is a form variable that tells QMF to display, for each subtotal line, the current value of column 2. The columns are numbered in the same order in which they were selected in the query; each column's number appears in the NUM column of the **FORM.COLUMNS** panel. In this example, column 2 is the DEPT.\_NUMBER column, so the current department number displays as part of the text for each subtotal line.

3. Press the Report function key to see the changed report, which is shown in the following figure.

DEPT. NUMBER	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
	NGAN	\$12,508.20	\$206.60	\$12,714.80
	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
DEPT. 15 TOTALS		\$41,269.53	\$1,468.70	\$42,738.23
20	JAMES	\$13,504.60	\$128.20	\$13,632.80
	PERNAL	\$18,171.25	\$612.45	\$18,783.70
	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
DEPT. 20 TOTALS		\$45,928.60	\$867.15	\$46,795.75
38	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55
DEPT. 38 TOTALS		\$59,778.80	\$1,913.30	\$61,692.10
		=====	=====	=====
		\$146,976.93	\$4,249.15	\$151,226.08

Figure 146. The changed report shows subtotal text after each department.

#### Related reference

**FORM.BREAKn** Use the **FORM.BREAKn** panels (where n is a number from 1 through 6) to make choices about the text and its placement for up to six breaks in a report. QMF places the text you specify on each break panel after its associated break in the report.

## Adding page headings and footings

You can use the **FORM.PAGE** panel to display headings and footings at the top and bottom of your report.

### Procedure

1. On the QMF command line, enter:

```
SHOW FORM.PAGE
```

The **FORM.PAGE** panel displays.

2. Move the cursor to the **PAGE HEADING TEXT** field.
3. Type the text you want to display at the top of each page in the report.

For this example, type `DIVISION EARNINGS REPORT`.

You can specify either left or right justification of the text, or align the text at a specific column.

If all the heading text for one line does not fit in the space that is allowed for that line, type `APPEND` in the `ALIGN` column of the next line. Then, change its line number to the same number as the preceding line, and continue typing the text. Be sure to indent the appended text at least one space.

If you need another line, move the cursor to the line above the place where you want the new line and press the Insert function key. You can specify up to 999 lines of text.

4. Type the text you want to display at the bottom of each page for the report in the **PAGE FOOTING TEXT** field.

For this example, type `COMPANY NAME`.



The following figure shows an example of the finished **FORM.PAGE** panel.

```

FORM.PAGE

Blank Lines Before Heading ==> 0      Blank Lines After Heading ==> 2
LINE  ALIGN  PAGE HEADING TEXT
----  -
1     CENTER  DIVISION EARNINGS REPORT
2     CENTER
3     CENTER
4     CENTER
          *** END ***

Blank Lines Before Footing ==> 2      Blank Lines After Footing ==> 0
LINE  ALIGN  PAGE FOOTING TEXT
----  -
1     CENTER  COMPANY NAME
2     CENTER
3     CENTER
4     CENTER
          *** END ***

```

Figure 147. Add page headings and footings on the **FORM.PAGE** panel.

5. Press the Report function key to see the changed report, which is shown in the following figure.

DIVISION EARNINGS REPORT				
DEPT. NUMBER	EMPLOYEE NAME	SALARY	COMMISSIONS	TOTAL EARNINGS
15	KERMISCH	\$12,258.50	\$110.10	\$12,368.60
	NGAN	\$12,508.20	\$206.60	\$12,714.80
	ROTHMAN	\$16,502.83	\$1,152.00	\$17,654.83
DEPT. 15 TOTALS		\$41,269.53	\$1,468.70	\$42,738.23
20	JAMES	\$13,504.60	\$128.20	\$13,632.80
	PERNAL	\$18,171.25	\$612.45	\$18,783.70
	SNEIDER	\$14,252.75	\$126.50	\$14,379.25
DEPT. 20 TOTALS		\$45,928.60	\$867.15	\$46,795.75
38	ABRAHAMS	\$12,009.75	\$236.50	\$12,246.25
	NAUGHTON	\$12,954.75	\$180.00	\$13,134.75
	O'BRIEN	\$18,006.00	\$846.55	\$18,852.55
	QUIGLEY	\$16,808.30	\$650.25	\$17,458.55
DEPT. 38 TOTALS		\$59,778.80	\$1,913.30	\$61,692.10
		=====	=====	=====
		\$146,976.93	\$4,249.15	\$151,226.08
COMPANY NAME				

Figure 148. The changed report displays the page heading and footing.

## Specifying fixed columns on a report

Using QMF forms, you can specify fixed columns on a report.

### About this task

In a displayed report, the fixed columns remain in place on the left side of the screen when you press the Left or Right function keys. A vertical line (|) separates the fixed area from the scrollable portion of the report.

On a printed report, QMF repeats the fixed columns on the left side of each page.

For this example, use the query in the following figure.

PROMPTED QUERY                    MODIFIED LINE 1

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

Join Tables:  
A.DEPT And B.DEPTNUMB

Columns:  
ID  
NAME  
JOB  
YEARS  
SALARY  
COMM  
DEPTNUMB  
DEPTNAME  
MANAGER  
DIVISION  
LOCATION

Sort:  
Ascending by ID

Figure 149. Use this query to see how fixed columns affect a report.

The query creates the report that is shown in the following figure:

ID	NAME	JOB	YEARS	SALARY	COMM	DEPTNUMB	DEPTNAME
10	SANDERS	MGR	7	18357.50	-	20	MID ATLAN
20	PERNAL	SALES	8	18171.25	612.45	20	MID ATLAN
30	MARENGHI	MGR	5	17506.75	-	38	SOUTH ATL
40	O'BRIEN	SALES	6	18006.00	846.55	38	SOUTH ATL
50	HANES	MGR	10	20659.80	-	15	NEW ENGLA
60	QUIGLEY	SALES	-	16808.30	650.25	38	SOUTH ATL
70	ROTHMAN	SALES	7	16502.83	1152.00	15	NEW ENGLA
80	JAMES	CLERK	-	13504.60	128.20	20	MID ATLAN
90	KOONITZ	SALES	6	18001.75	1386.70	42	GREAT LAK
100	PLOTZ	MGR	7	18352.80	-	42	GREAT LAK
110	NGAN	CLERK	5	12508.20	206.60	15	NEW ENGLA
120	NAUGHTON	CLERK	-	12954.75	180.00	38	SOUTH ATL
130	YAMAGUCHI	CLERK	6	10505.90	75.60	42	GREAT LAK
140	FRAYE	MGR	6	21150.00	-	51	PLAINS

Figure 150. The right side of the report is not visible.

One problem with this report is that, when you press the Right function key to see the rest of the information, you can no longer see the ID and NAME fields. You cannot tell which information belongs to which employee.

## Procedure

To specify fixed columns on a report:

1. On the QMF command line, enter:

```
SHOW FORM.OPTIONS
```

The **FORM.OPTIONS** panel displays, as shown in the following figure.

```

FORM.OPTIONS                                MODIFIED

What do you want for
Detail spacing?                             ===> 1
Line wrapping width?                       ===> NONE
Report text line width?                    ===> DEFAULT
Number of fixed columns in report?         ===> 2
Do you want
Outlining for break columns?               ===> YES
Default break text (*)?                   ===> YES
Function name in column heading when grouping? ===> YES
Column wrapped lines kept on a page?      ===> YES
Across summary column?                    ===> YES
Automatic reordering of report columns?   ===> NO
Page renumbering at the highest break level? ===> NO

Do you want separators for
Column heading?                            ===> YES      Break summary?          ===> YES
Across heading?                            ===> YES      Final summary?         ===> YES

```

Figure 151. Specify the number of columns you want to remain fixed.

2. Move the cursor to the **Number of fixed columns in report?** field.
3. Type the number of columns you want to remain fixed.  
For this example, you want the ID and NAME columns to display always.
  - a) Type 2 in the **Number of fixed columns in report?** field.
  - b) Press the Report function key to see the changed report.
  - c) Press the Right function key to display the rest of the information.

The columns that you specified as fixed remain displayed on the screen, as shown in the following figure.

ID	NAME	EPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	SANDERS	20	MID ATLANTIC	10	EASTERN	WASHINGTON
20	PERNAL	20	MID ATLANTIC	10	EASTERN	WASHINGTON
30	MARENGHI	38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
40	O'BRIEN	38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
50	HANES	15	NEW ENGLAND	50	EASTERN	BOSTON
60	QUIGLEY	38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
70	ROTHMAN	15	NEW ENGLAND	50	EASTERN	BOSTON
80	JAMES	20	MID ATLANTIC	10	EASTERN	WASHINGTON
90	KOONITZ	42	GREAT LAKES	100	MIDWEST	CHICAGO
100	PLOTZ	42	GREAT LAKES	100	MIDWEST	CHICAGO
110	NGAN	15	NEW ENGLAND	50	EASTERN	BOSTON
120	NAUGHTON	38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
130	YAMAGUCHI	42	GREAT LAKES	100	MIDWEST	CHICAGO
140	FRAYE	51	PLAINS	140	MIDWEST	DALLAS

Figure 152. The first two columns remain fixed when you press the Right function key.

## Displaying a representative report before you select data

Before you select any data, you can use the LAYOUT command to display a representation of the report a form will produce.

### About this task

Displaying a representation is helpful if you want to test or change a form without running a query. Also, you can display a representation to help you remember the report produced by a particular form.

Your site might not support use of the LAYOUT command for the following reasons:

- ISPF is not available when you run QMF.
- Your QMF administrator has not made the default command synonyms available.

Check with your administrator before you try to use this command.

On a representative report produced by the LAYOUT command, strings of letters represent character data and strings of numbers represent numeric data.

## Procedure

To display a representative report before you select data, choose one of the following options.

- To display a representative report formatted with a saved FORM object, enter the following command on the QMF command line:

```
LAYOUT FORM formname
```

For example, for a saved form that is named FORM5, enter:

```
LAYOUT FORM FORM5
```

- To display a representative report for a form in temporary storage, enter the following command:

```
LAYOUT FORM
```

## Example

The following figure shows an example of a representative report.

```
Employee Data for the XXXXXXXXXX Division
Date: 11/27/1991

Department number 1, Department name XXXXXXXXXXXXXXXX
Manager: 0

*****
** Personnel Status Report **
*****
Position: AAAAA

    Employee: XXXXXXXXX
           ID: 0
Years of Service: 0
           Salary: 0.00
           Commission: 0.00
Total Earnings: 0.00

====> Number in AAAAA position in Department 1: 1

Position: BBBBB

    Employee: XXXXXXXXX
           ID: 0
Years of Service: 0
           Salary: 0.00
           Commission: 0.00
Total Earnings: 0.00

====> Number in BBBBB position in Department 1: 1

Department number 2, Department name XXXXXXXXXXXXXXXX
Manager: 0
.
.
.
```

Figure 153. A representative report shows how data will be displayed.

The first control break, on the DEPT column (a numeric column), is represented by a 1 for the first department and a 2 for the second department. The second control break, on the JOB column (a character column), is represented by AAAAA for the first job title and BBBBB for the second job title.

## Refining page headings and footings

Use the FORM.PAGE panel to refine heading and footing text.

You can do the following with the heading and footing text in a report:

- Include form variables, such as &DATE and &TIME, and global variables
- Control the placement of the text
- Indicate the number of blank lines to appear before and after the text

## Using a global variable in a heading or footing

Global variables allow you to save a QMF object and use it multiple times for different purposes without having to change it.

### About this task

The SQL query in the following figure selects and joins columns from the Q.STAFF and Q.ORG tables. It also uses a variable to specify the division so that, each time the query is run, a different division can be specified. Multiple queries for each of the different divisions are not necessary.

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY,  
       COMM, DEPTNUMB, DEPTNAME, MANAGER,  
       DIVISION, LOCATION  
FROM Q.STAFF, Q.ORG  
WHERE DEPT=DEPTNUMB  
      AND DIVISION = &DIVISION  
ORDER BY DEPT, JOB, LOCATION
```

Figure 154. A query that joins columns from the Q.STAFF and Q.ORG tables

You can provide the value of the variable when you run the query or you can set a global variable with the value before you run the query.

### Procedure

To set a global variable:

1. On the QMF command line, enter:

```
SET GLOBAL (varname=value
```

For this example, enter:

```
SET GLOBAL (DIVISION = ''WESTERN''
```

If you do not set global variables before you run your query, QMF displays a panel that prompts you to enter values for the variables. By default, values for global variables persist for the duration of the QMF session or until you reset them. However, the DSQEC\_USERGLV\_SAV global variable can be set to save global variable values from one session to another.

2. Run the query to display the default report. The following figure shows the report for this example. Press the Right key to see the part of the report not visible on the screen.

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	DEPTNUMB	D
330	BURKE	66	CLERK	1	10988.00	55.50	66	P
270	LEA	66	MGR	9	18555.50	-	66	P
320	GONZALES	66	SALES	4	16858.20	844.00	66	P
310	GRAHAM	66	SALES	13	21000.00	200.30	66	P
280	WILSON	66	SALES	9	18674.50	811.50	66	P
350	GAFNEY	84	CLERK	5	13030.50	188.00	84	M
290	QUILL	84	MGR	10	19818.00	-	84	M
300	DAVIS	84	SALES	5	15454.50	806.10	84	M
340	EDWARDS	84	SALES	7	17844.00	1285.00	84	M

Figure 155. The default report has no page headings or footings.

- On the QMF command line, enter:

```
SHOW FORM.PAGE
```

The FORM.PAGE panel displays.

- In line 1 of the **PAGE HEADING TEXT** field, change the alignment to LEFT and type the text you want to display as the page heading.

Because the DIVISION column is the 11th column in the SELECT statement of the query, type the following for this example:

```
EMPLOYEE DATA FOR THE &11 DIVISION
```

- Press the Report function key to see the changed report.

### Related concepts

[Making your prompted query reusable with substitution variables](#)

When you specify substitution variables in a prompted query, you can use the same query to retrieve different information. You supply a new value for the variable each time you run the query.

### Related reference

[SET GLOBAL](#)The SET GLOBAL command assigns values to global variables from the QMF command line, from a procedure, or through the callable interface.

## Adding the date, time, and page number to a heading or footing

You can display the date, time, or page number on a page heading or footing by using default form variables.

### &DATE

Adds the current date to a heading or footing when you run the report.

### &TIME

Adds the current time to a heading or footing when you run the report.

### &PAGE

Adds the current page number to a heading or footing when you run the report.

Specify the variable in the appropriate place on FORM.PAGE. You can also specify text before or after the form variable. For this example, add the date to the second line of the report heading by changing the alignment to LEFT and typing Date: &DATE in the **PAGE HEADING TEXT** field on the second line of the page heading.

### Related concepts

[Variables used in forms](#)You can use global variables and form variables in QMF forms. A variable can replace a string of text or a numeric value. You can assign different values to the variable to produce different reports without changing the form.

## Changing the alignment of page headings and footings

The default alignment for page headings and footings is centered (CENTER), but you can change the alignment on the FORM.PAGE panel.

### About this task

In this example, you will change the alignment of the page footing to the left margin of the report.

### Procedure

1. Move the cursor to the **ALIGN** field for the line you want to change.
2. Type the new alignment value. For this example, change the alignment for line 1 of the page footing to LEFT, and type **\*\* Company Name \*\*** as the footing text, as shown in the following figure:

```
Blank Lines Before Footing ==> 2      Blank Lines After Footing ==> 0
LINE  ALIGN  PAGE FOOTING TEXT
-----1-----2-----3-----4-----5-----+
1     LEFT   ** Company Name **
2     CENTER
3     CENTER
4     CENTER
*** END ***
```

Figure 156. Change the alignment of page headings and footings on FORM.PAGE.

3. Press the Report function key to see the changed report, shown in the following figure. Press the Right key to see the part of the report not visible on the screen.

```
Employee Data for the WESTERN Division
Date: 1998-02-17

   ID  NAME          DEPT  JOB      YEARS   SALARY    COMM  DEPTNUMB  D
-----1-----2-----3-----4-----5-----6-----7-----8-----9-----
 330  BURKE           66    CLERK    1      10988.00  55.50   66        P
 270  LEA              66    MGR      9      18555.50  -        66        P
 320  GONZALES        66    SALES    4      16858.20  844.00   66        P
 310  GRAHAM          66    SALES   13      21000.00  200.30   66        P
 280  WILSON          66    SALES    9      18674.50  811.50   66        P
 350  GAFNEY          84    CLERK    5      13030.50  188.00   84        M
 290  QUILL           84    MGR     10      19818.00  -        84        M
 300  DAVIS           84    SALES    5      15454.50  806.10   84        M
 340  EDWARDS         84    SALES    7      17844.00  1285.00  84        M

** Company Name **
```

Figure 157. The changed report shows the left-aligned headings and footings.

## Adding break segments and text to your report

You can add break segments to your report by specifying BREAKn usage codes for columns.

In addition, you can use the six FORM.BREAKn panels to do the following:

- Specify the break heading text lines and footing text lines for your report.
- Control the placement of the break heading and footing text.
- Indicate the number of blank lines that appear before and after the break heading and footing.
- Specify whether you want a new page at each break heading or footing.
- Specify whether you want to repeat the detail heading after the break heading.
- Use form or global variables to place additional information at breaks.
- Place the break summary at a specific line.

- Place the results of calculation expressions that are specified on FORM.CALC in the BREAK footings of your report.

**Related concepts**

Calculating values to use in a report

You can specify calculations in a form that are similar to the calculations that are performed in a query.

## Adding break heading and footing text to a report

You can add heading and footing text for each break segment in your report by using the FORM.BREAK panels.

**About this task**

In the following example, you add break heading text for BREAK1 and break heading and footing text for BREAK2.

**Procedure**

1. On the QMF command line, enter:

```
SHOW FORM.COLUMNS
```

The FORM.COLUMNS panel displays.

2. In the **USAGE** field, you can specify up to six breaks for the columns in the report. For this example, type BREAK1 for DEPT and BREAK2 for JOB (as shown in the following figure) so that the report is broken down by department and then further by type of job within each department.

FORM.COLUMNS			MODIFIED				
NUM	COLUMN	HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
Total Width of Report Columns: 128							
1	ID			2	6	L	1
2	NAME			2	9	C	2
3	DEPT		BREAK1	2	6	L	3
4	JOB		BREAK2	2	5	C	4
5	YEARS			2	6	L	5
	.						
	.						
	.						
	***	END	***				

*Figure 158. Specify breaks at the DEPT and JOB columns.*

3. Press the Report function key to see the changed report, which is shown in the following figure.



ID	NAME	DEPT	JOB	YEARS	MANAGER	DIVISION	LOCATION
330	BURKE	<b>1</b> 66	CLERK	1	270	WESTERN	SAN FRANCISCO <b>2</b> *
270	LEA	<b>2</b>	MGR	9	270	WESTERN	SAN FRANCISCO <b>2</b> *
280	WILSON			9	270	WESTERN	SAN FRANCISCO
310	GRAHAM			13	270	WESTERN	SAN FRANCISCO
320	GONZALES	<b>2</b>	SALES	4	270	WESTERN	SAN FRANCISCO <b>1</b> **
350	GAFNEY	84	CLERK	5	290	WESTERN	DENVER *
290	QUILL		MGR	10	290	WESTERN	DENVER *
340	EDWARDS		SALES	7	290	WESTERN	DENVER
300	DAVIS			5	290	WESTERN	DENVER

Figure 159. The changed report shows breaks after each department and job.

**1**

Shows a first-level break (by department)

**2**

Shows a second-level break (by job)

4. On the QMF command line, enter `SHOW FORM.BREAKn`.

In this command, *n* is the break level. For this example, enter:

```
SHOW FORM.BREAK1
```

The FORM.BREAK panel for the break level you specified displays. You specify break heading and footing text on this panel.

5. Type 2 for **Blank Lines After Heading**.

For your own reports, you can leave the defaults or type new values for the **New Page for Break**, **Blank Lines Before Heading**, **Repeat Detail Heading?**, and **Blank Lines After Heading** fields.

6. Type the break text for each line under the BREAK 1 HEADING TEXT heading.

You can use variables in the text that represent the order of a particular column in the SELECT statement of the query.

In the example query on which the report is based (shown in [“Using a global variable in a heading or footing”](#) on page 121), DEPTNUMB is the third column in the SELECT statement, DEPTNAME is the ninth column, and MANAGER is the tenth column.

- For this example, type `Department number &3`, `Department name &9` for the first line of text, and `Manager: &10` for the second line of text.
- You can specify either left or right justification of the text or align the text at a specific column. For this example, leave the default alignment at LEFT.

If all the break text for one line does not fit in the space that is allowed on that line, type APPEND in the ALIGN column of the next line. Then, change its line number to the same number as the preceding line, and continue typing the text. Be sure to indent the appended text at least one space.

If you need another line, move the cursor to the line above the place where you want the new line and press the Insert function key. You can specify up to 999 lines of text.

The following figure shows the alignment values and the heading text that you entered so far.

```

LINE  ALIGN  BREAK 1 HEADING TEXT
----  -
1     LEFT   Department number &3, Department name &9
2     LEFT   Manager: &10

```

Figure 160. Specify headings and footings for report breaks on a FORM.BREAK panel.

7. Specify text for the break footing in the same way you specified text for the break heading.
  - a) For this example, do not specify footing text for the level-one break.
  - b) For the second break, on the JOB column, enter both heading and footing text.
8. On the QMF command line, enter:

```
SHOW FORM.BREAK2
```

9. Type Position: &4 for the first line of heading text because JOB is the fourth column that is named in the SELECT statement of the query the report is based on.
10. Change the alignment to LEFT and type ==> Number in &4 position in Department &3: for the first line of footing text.
11. Change the line number in the **LINE** field for the second line of footing text from **2** to **1**.
12. Type APPEND in the **ALIGN** field for the second line of footing text.
13. Type &COUNT2 in the **BREAK 2 FOOTING TEXT** field. Be sure to leave a space at the beginning of the second line of text. These two lines appear as one line on the report. The variable &COUNT2 is an aggregation variable that counts the number of values for column 2, NAME.

The following figure shows the completed FORM.BREAK2 panel for this example.

```

FORM.BREAK2
New Page for Break?      ==> NO      Repeat Detail Heading?  == > NO
Blank Lines Before Heading ==> 0      Blank Lines After Heading == > 1
LINE  ALIGN  BREAK 2 HEADING TEXT
----  -
1     LEFT   Position: &4
2     LEFT
3     LEFT
          *** END ***

New Page for Footing?    ==> NO      Put Break Summary at Line == > 1
Blank Lines Before Footing ==> 0      Blank Lines After Footing == > 1
LINE  ALIGN  BREAK 2 FOOTING TEXT
----  -
1     LEFT   ==> Number in &4 position in Department &3:
1     APPEND   &COUNT2
3     RIGHT
          *** END ***

```

Figure 161. Specify text for the second break on FORM.BREAK2.

14. Press the Report function key to see the changed report.

The following figure shows how the changed report looks.

EMPLOYEE DATA FOR THE WESTERN DIVISION  
DATE: 1998-03-17

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	DEPTNUMB
DEPARTMENT NUMBER 66, DEPARTMENT NAME PACIFIC							
MANAGER: 270							
POSITION: CLERK							
330	BURKE	66	CLERK	1	10988.00	55.50	66
==> NUMBER IN CLERK POSITION IN DEPARTMENT 66: 1							
POSITION: MGR							
270	LEA	66	MGR	9	18555.50	-	66
==> NUMBER IN MGR POSITION IN DEPARTMENT 66: 1							
POSITION: SALES							
320	GONZALES		SALES	4	16858.20	844.00	66
310	GRAHAM			13	21000.00	200.30	66
280	WILSON			9	18674.50	811.50	66
==> NUMBER IN SALES POSITION IN DEPARTMENT 66: 3							
DEPARTMENT 84, DEPARTMENT NAME MOUNTAIN							
MANAGER: 290							
POSITION: CLERK							
350	GAFNEY	84	CLERK	5	13030.50	188.00	84
==> NUMBER IN CLERK POSITION IN DEPARTMENT 84: 1							
POSITION: MGR							
290	QUILL	84	MGR	10	19818.00	-	84
==> NUMBER IN MGR POSITION IN DEPARTMENT 84: 1							
POSITION: SALES							
340	EDWARDS		SALES	7	17844.00	1285.00	84
300	DAVIS			5	15454.50	806.10	84
==> NUMBER IN SALES POSITION IN DEPARTMENT 84: 2							
** COMPANY NAME **							

Figure 162. The changed report shows first- and second-level break text.

### Related reference

[BREAK usage codes](#)The BREAK usage codes provide six levels of breaks (or groupings) in a report.

## Refining the format of your report with detail blocks

You can reformat and add text to your report with detail blocks. A detail block is a set of specifications that tells QMF to put in whatever special formatting you want for one row of the data that is retrieved by your query.

### About this task

Specify detail blocks on the **FORM.DETAIL** panel. You can use the panel to do the following:

- Format the detail heading and block text in your report.

- Enter your own text for the detail heading to either replace the column headings or combine the detail heading with the column headings.
- Enter your own block of detail text and specify its location anywhere within the detail block of the report.
- Use data from the columns that are selected in your query by using form column variable names or aggregation variables. Place that data at any location within the detail block.
- Specify the placement of tabular data.
- Mix tabular (column) data with free-flowing text (detail block text).
- Omit tabular data from your report.
- Place the results of calculation expressions, which are specified on FORM.CALC, in the detail block text of your report.

In this example, you use the **FORM.DETAIL** panel to reformat the following report.

```

EMPLOYEE DATA FOR THE WESTERN DIVISION
DATE: 1998-03-17

      ID  NAME      DEPT  JOB    YEARS    SALARY    COMM  DEPTNUMB
-----
DEPARTMENT NUMBER 66, DEPARTMENT NAME PACIFIC
MANAGER: 270

POSITION: CLERK
      330  BURKE      66  CLERK     1    10988.00    55.50    66
====> NUMBER IN CLERK POSITION IN DEPARTMENT 66: 1

POSITION: MGR
      270  LEA        66  MGR       9    18555.50     -        66
====> NUMBER IN MGR POSITION IN DEPARTMENT 66: 1

POSITION: SALES
      320  GONZALES      SALES     4    16858.20    844.00    66
      310  GRAHAM        13    21000.00    200.30    66
      280  WILSON        9    18674.50    811.50    66
====> NUMBER IN SALES POSITION IN DEPARTMENT 66: 3

DEPARTMENT 84, DEPARTMENT NAME MOUNTAIN
MANAGER: 290

POSITION: CLERK
      350  GAFNEY      84  CLERK     5    13030.50    188.00    84
====> NUMBER IN CLERK POSITION IN DEPARTMENT 84: 1

POSITION: MGR
      290  QUILL       84  MGR      10    19818.00     -        84
====> NUMBER IN MGR POSITION IN DEPARTMENT 84: 1

POSITION: SALES
      340  EDWARDS      SALES     7    17844.00    1285.00    84
      300  DAVIS        5    15454.50     806.10    84
====> NUMBER IN SALES POSITION IN DEPARTMENT 84: 2

** COMPANY NAME **

```

Figure 163. The changed report shows first- and second-level break text.

You also remove the column headings from the report, omit tabular data (columns), and add headings for each subsection.

## Procedure

1. On the QMF command line, enter:

```
DISPLAY FORM.DETAIL
```

The **FORM.DETAIL** panel displays, as shown in the following figure.

```
FORM.DETAIL                                MODIFIED      Var 1  of 1
Include Column Headings with Detail Heading? ==> NO
LINE  ALIGN  DETAIL HEADING TEXT
----  -1-2-3-4-5-
1     LEFT   *****
2     LEFT   ** PERSONNEL STATUS REPORT **
3     LEFT   *****

New Page for Detail Block? ==> NO      Repeat Detail Heading? ==> NO
Keep Block on Page? ==> NO      Blank Lines after Block ==> 0
Put Tabular Data at Line (Enter 1-999 or NONE) ==> 1
LINE  ALIGN  DETAIL BLOCK TEXT
----  -1-2-3-4-5-
1     LEFT
2     LEFT
      *** END ***

Select Panel Variation? ==> YES
```

Figure 164. Specify text for the detail heading on FORM.DETAIL.

2. Because the final report is not in tabular format (columns and rows), you do not want to display the column headings. Type **NO** in the **Include Column Headings with Detail Heading?** field for this example.

If you leave YES, the column headings display immediately after any detail heading text in the report.

3. Type the detail heading text for each line in the **DETAIL HEADING TEXT** field. You can use form variables in the text.

For this example, type the text as it is shown on the **FORM.DETAIL** panel in the preceding figure. Insert an extra line and change its line number.

- a) Rearrange the data from each column into a vertical list, rather than a tabular arrangement of columns and rows.

Give each column value a new label to replace the column headings. Specify the format of the detail block on the bottom half of the **FORM.DETAIL** panel.

4. Leave the defaults, or type new values for the **New Page for Detail Block?**, **Repeat Detail Heading?**, **Keep Block on Page?**, and **Blank Lines After Block** fields.

For this example, type 1 for **Blank Lines After Block**.

5. Type a line number if you want to include tabular data in the report, or type NONE if you want to remove all the tabular data.

For this example, type NONE to remove all the tabular data from the report.

6. Type the detail block text for each line in the **DETAIL BLOCK TEXT** field. Use form variables and text to provide values for the column headings and column data on the report.

For this example, type the text as it is shown on the **FORM.DETAIL** panel in the following figure.

LINE	ALIGN	DETAIL	BLOCK	TEXT
----	----	-----1-----	2-----3-----4-----5-----	+
1	LEFT		Employee:	&2
2	LEFT		ID:	&1
3	LEFT	Years of Service:		&5
4	LEFT		Salary:	&6
5	LEFT		Commission:	&7

Figure 165. Specify replacement column headings with detail block text.

You can create variations of the detail blocks to use with different conditions in a report format with panel variations.

7. Press the Report function key to see the changed report, which is shown in the following figure.

```

Employee Data for the WESTERN Division
Date: 1998-03-17

*****
** Personnel Status Report **
*****
Department number 66, Department name PACIFIC
Manager: 270

Position: CLERK
Employee: BURKE
ID: 330
Years of Service: 1
Salary: 10988.00
Commission: 55.50

==> Number in CLERK position in Department 66: 1

Position: MGR
EMPLOYEE: LEA
ID: 270
Years of Service: 9
Salary: 18555.50
Commission: -

==> Number in MGR position in Department 66: 1

```

Figure 166. The changed report shows the results of reformatting.

### Related concepts

[Calculating values to use in a report](#)

You can specify calculations in a form that are similar to the calculations that are performed in a query.

### Related reference

[FORM.DETAILFORM.DETAIL](#) consists of detail variations that you define. You can create up to 99 variations, and each variation can correspond to conditions entered on [FORM.CONDITIONS](#).

## Specifying text to appear at the end of your report

Use [FORM.FINAL](#) to place text at the end of your report. For example, you might want to explain items in the report or include information that summarizes the data, such as totals or averages.

### About this task

The **FORM.FINAL** panel controls the following aspects of the final text in the report:

- Position of the text
- Whether final text begins on a new page
- Number of blank lines that appear before the text
- The line number at which the final text begins

The final text can also include the results of calculation expressions, which you specify on FORM.CALC. For this example, you add final text to the report that shows the total number of employees for the Western Division and the average of their salaries.

## Procedure

1. On the QMF command line, enter:

```
SHOW FORM.FINAL
```

The **FORM.FINAL** panel displays, as shown in the following figure:

```
FORM.FINAL                                MODIFIED
New Page for Final Text?  ===> NO          Put Final Summary at Line ===> 1
Blank Lines Before Text  ===> 0
LINE  ALIGN  FINAL TEXT
----  -
1     LEFT   Total Number of Employees for the &11 Division is
1     APPEND &COUNT1.
2     LEFT   Average Salary for the &11 Division is &AVG6.
```

Figure 167. Specify text for the end of a report on FORM.FINAL.

2. Leave the defaults, or type new values for the **New Page for Final Text?**, **Put Final Summary at Line**, and **Blank Lines Before Text** fields.

For this example, leave the defaults for these fields.

3. Type the final text for each line under the **FINAL TEXT** field. You can use form variables in the text.

For this example, change the alignment to LEFT. Because DIVISION is the eleventh column in the SELECT statement of the query on which the report is based, type the following for the first line:

```
Total Number of Employees for the &11 Division is
```

Specify APPEND for alignment for the next line, change the line number to 1, and type &COUNT1 because you need a total, or a count, of all employees in the division; ID (employee identification numbers) is the first column in the SELECT statement of the query, and &COUNT1 counts all unique values in column 1. Be sure to leave a space before &COUNT1.

For the next line of final text, change the alignment to LEFT and type Average Salary for the &11 Division is &AVG6 because you need to average all the values in column 6, which is the SALARY column.

4. Press the Report function key to display the changed report, which is shown in the following figure:

```
Employee:  DAVIS
           ID:  300
Years of Service:  5
           Salary: 15454.50
           Commission: 806.10

===> Number in SALES position in Department 84:  2

Total Number of Employees for the WESTERN Division is 9.
Average Salary for the WESTERN Division is 16913.69.

** Company Name **
*** END ***
```

Figure 168. The final text displays at the end of your report.

## Related tasks

[Using a global variable in a heading or footing](#)

Global variables allow you to save a QMF object and use it multiple times for different purposes without having to change it.

## Calculating values to use in a report

You can specify calculations in a form that are similar to the calculations that are performed in a query.

QMF evaluates calculations in a form by using the REXX programming language. Calculations can take advantage of all the built-in REXX functions as well as REXX execs that you write. However, calculations in a form can affect performance.

The values in your reports can come from the following:

- Data that you import or have stored in the database
- Calculations that are performed in a query
- Calculations that are performed within a QMF form

You can create calculations to use in a report in one of the following ways:

- Define an expression that calculates a value.
- Create a REXX program to return a value.

Your site might not support the use of calculations and REXX functions. Check with your administrator before you try to use calculations in a report.

**Restriction:** You cannot calculate values to use in a report in CICS.

### Related concepts

Using [REXX with QMF forms](#) Expressions used in FORM.CALC, FORM.CONDITIONS, and FORM.COLUMNS (Column Definition) can consist of terms (strings, symbols, and functions) interspersed with operators and parentheses.

### Related information

The [IBM Publications Center](#) Search for information about REXX.

## Displaying a calculated value on a report

You can display a calculated value in detail block text, break footing text, and final text on a report.

### About this task

In this example, you will define an expression that adds an employee's salary and commission. Then, you will display the result in the detail block text on the sample Personnel Status Report.

### Procedure

1. On the QMF command line, enter:

```
SHOW FORM.CALC
```

The **FORM.CALC** panel displays, as shown in the following figure.

FORM.CALC		MODIFIED		
ID	CALCULATION EXPRESSION	Pass Nulls?	For WIDTH	&CALCid EDIT
1	&6 + NULL(&7) *** END ***	YES	12	D2

Figure 169. Specify an expression to calculate a value on FORM.CALC.

2. Type an ID number for the expression. You can use any number from 1 through 999.



In this example, type 1 for ID.

3. Type the expression, using form variables to specify the columns, in the **CALCULATION EXPRESSION** field.

In this example, type `&6 + NULL (&7)`, which means to add the values in columns 6 (SALARY) and 7 (COMM).

Because some of the commission values in the sample tables are null, they appear as a hyphen in the report. REXX cannot perform an arithmetic operation on columns that contain both numeric values and null values. The REXX program called NULL looks for the nulls in the data and replaces them with a specified value. In this case, it replaces nulls with zeros.

When you write a REXX program, make sure that you make it available to QMF by specifying the correct data set. See your administrator for more information. Here is the NULL program for this example:

```
/* REXX program to substitute 0 in place of nulls */
parse arg in1
if in1 = "DSQNULL" then
  value = 0
else value = in1
return value
```

4. In the **PASS NULLS** field, type YES to process nulls for this example.
5. In the **WIDTH** field, type 12 to accommodate the number of characters expected in the result of the calculation.
6. In the **EDIT** field, type the edit code for the result of this calculation. Because you want to display total earnings as a dollar value, type D2.

After you define the expression, you can use the **FORM.DETAIL** panel to define how you want to display the result of the calculation on the report.

7. On the QMF command line, enter:

```
SHOW FORM.DETAIL
```

The **FORM.DETAIL** panel displays.

8. Type the new line of text in the **DETAIL BLOCK TEXT** field.

For this example, type `Total Earnings: &CALC1`. The variable `&CALC1` corresponds to the calculation expression you created on the **FORM.CALC** panel.

Change the line number and alignment for the new line of text. For this example, change the line number to 6 and change the alignment to 3. This means that you want this line of detail block text to begin in column 3.

The following figure shows the **FORM.DETAIL** panel with the calculation expression for Total Earnings and the detail block text.

LINE	ALIGN	DETAIL BLOCK TEXT
1	LEFT	Employee: &2
2	LEFT	ID: &1
3	LEFT	Years of Service: &5
4	LEFT	Salary: &6
5	LEFT	Commission: &7
6	3	Total Earnings: &CALC1

*Figure 170. Specify where a calculated value appears by entering calculation expression IDs in detail block text*

9. Press the Report function key to see the changed report, which is shown in the following figure.

```

Employee Data for the WESTERN Division
Date: 1998-03-17

*****
** Personnel Status Report **
*****
Department number 66, Department name PACIFIC
Manager: 270

Position: CLERK
      Employee: BURKE
              ID: 330
Years of Service: 1
      Salary: 10988.00
      Commission: 55.50
      Total Earnings: $11,043.50

```

Figure 171. The calculated value appears next to Total Earnings in the report.

In this example, the value for Total Earnings comes from the &CALC1 variable.

### Related concepts

#### Specifying punctuation for the values in a column

You can use edit codes in QMF to specify how to punctuate the values in a column. An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report. You specify edit codes on the FORM.COLUMNS panel.

Edit codesAn edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

### Related tasks

#### Creating a column by defining expressions with Prompted Query

You can create a column in your report by defining an expression. An expression adds, subtracts, multiplies, or divides the values in two or more columns and places the results in a new column.

## Displaying special conditions on your report

You can define a calculation that identifies a special condition on your report by using either an expression or a REXX program.

In these examples, you use calculated values to identify two special conditions on a report – one condition to identify employees who deserve a commission bonus, and one condition to identify employees who need a raise.

Your site might not support the use of calculations and REXX functions. Check with your administrator before you try to use calculations in a report.

### Identifying a special condition by using a REXX program

In this example, you create a condition that prints text on a report for all employees with a commission greater than or equal to \$800.00.

### Procedure

1. On the QMF command line, enter:

```
SHOW FORM.CALC
```

The **FORM.CALC** panel displays.

2. In the **ID** field, type an ID number for the expression. You can use any number from 1 through 999. If you created an expression in “[Displaying a calculated value on a report](#)” on page 132, you already used a value of 1 for the expression ID, so type 2.
3. In the **CALCULATION EXPRESSION** field, type the expression, and use form variables to specify the columns.

For this example, type BONUS (&7 800). BONUS is a REXX program that you can write that checks the value in the COMMISSION column (&7) to see whether it is greater than or equal to the amount you specify for the commission bonus (800).

Here is the BONUS program for this example:

```
/* REXX BONUS */
/* program to flag employees whose commission levels warrant a bonus */

parse arg commission commission_level
retvalue = ' '
if (commission ^= "DSQNULL") & (commission >= commission_level) then
retvalue = '*** Commission Bonus ***'
return retvalue
```

- In the **WIDTH** field, type 24 to accommodate the number of characters in the text string \*\*\* Commission Bonus \*\*\*.
- In the **EDIT** field, type the edit code C to treat the text string as character data, as shown in the following figure.

FORM.CALC		MODIFIED		
ID	CALCULATION EXPRESSION	Pass Nulls?	For &CALCid WIDTH	EDIT
1	&6 + NULL(&7)	YES	12	D2
2	BONUS(&7 800)	YES	25	C
	*** END ***			

Figure 172. Use FORM.CALC to specify an expression that uses a REXX program.

Now that you specified the calculation expression, use detail block text to specify the placement of the text string.

- On the QMF command line, enter:

```
SHOW FORM.DETAIL
```

The **FORM.DETAIL** panel displays.

- Type the information for the detail block text. For this example, you want to display the result of the calculation (&CALC2) on the same line as the total earnings value (line 6), in column 40.

If the text you want to display on the report exceeds the report text line width, you can increase the report text line width on the **FORM.OPTIONS** panel.

The following figure shows the changes on the form that reflect the necessary calculations:

```
Put Tabular Data at Line (Enter 1-999 or NONE) ==> NONE
LINE  ALIGN  DETAIL BLOCK TEXT
-----1-----2-----3-----4-----5-----+
6      3      Total Earnings:    &CALC1
6      40      &CALC2
```

Figure 173. Calculations in a form

- Press the Report function key to see the changed report, which is shown in the following figure:

```

Employee Data for the WESTERN Division
Date: 1998-03-17

*****
** Personnel Status Report **
*****
Department number 66, Department name PACIFIC
Manager: 270

Position: CLERK
    Employee: BURKE
           ID: 330
Years of Service: 1
    Salary: 10988.00
    Commission: 55.50
    Total Earnings: 11043.50

==> Number in CLERK position in Department 66: 1

Position: MGR
    Employee: LEA
           ID: 270
Years of Service: 9
    Salary: 18555.50
    Commission: 0.00
    Total Earnings: 18555.50

==> Number in MGR position in Department 66: 1

Position: SALES
    Employee: WILSON
           ID: 280
Years of Service: 9
    Salary: 18674.50
    Commission: 811.50
    Total Earnings: 19486.00          *** Commission Bonus ***
.
.
.

Total Number of Employees for the WESTERN Division is 9.
Average Salary for the WESTERN Division is 16913.69.

** Company Name **
*** END ***

```

Figure 174. The changed report shows the conditional text for bonuses.

### Related reference

[FORM.OPTIONS](#) Use FORM.OPTIONS to adjust the appearance of your report.

### Related information

[The IBM Publications Center](#) Search for information about REXX.

## Identifying a special condition using an expression

In this example, you create a report that displays specific text whenever the data meets certain conditions by using the FORM.CONDITIONS panel with the FORM.DETAIL panel.

### About this task

You can format your report in different ways with this method, depending on conditions you want QMF to test for. You specify an expression for any tests or conditions on the FORM.CONDITIONS panel. Then you associate a FORM.DETAIL panel with each test. When the test evaluates to true for the data in a particular row, QMF formats the report the way you specify on the associated FORM.DETAIL panel. When the test does not evaluate to true, you can specify a different format on another FORM.DETAIL panel.

In this example, you create a condition that prints the text `*** Needs Raise ***` in the report for all employees with salary plus commissions less than \$17,000.00.

## Procedure

1. Run the query and display the report.

For this example, use this query:

```
SELECT ID, NAME, JOB, DEPT, SALARY, COMM
FROM Q.STAFF
```

2. On the QMF command line, enter:

```
SHOW FORM.CONDITIONS
```

The FORM.CONDITIONS panel displays.

3. Type an ID number for the expression. You can use any number from 1 through 999.

For this example, you need to create two conditions: one for employees who earn greater than or equal to \$17,000.00, and one for employees who earn less than \$17,000.00.

Type 1 for the first expression and 2 for the second expression.

4. In the **CONDITIONAL EXPRESSION** field, type the first expression, using form variables to specify the columns according to their position numbers in the SELECT statement of the query.

For this example, type the following:

```
&5 + NULL(&6) >= 17000.00
```

This is the first condition, which finds each row for which the employee's total earnings (SALARY+COMM) are greater than or equal to \$17,000.00.

For the second condition, type the following, which finds each row for which the employee's total earnings are less than \$17,000.00:

```
&5 + NULL(&6) < 17000.00
```

5. Type YES in the **PASS NULLS** field to process nulls for this example.

If you want to process nulls, you can create a REXX program like NULL to substitute a 0 (or whatever value is appropriate) for the null value. Then you can use that value in the calculation. A sample NULL program is shown in “Displaying a calculated value on a report” on page 132. The following figure shows how to use the NULL program in the CONDITIONAL EXPRESSION field of the FORM.CONDITIONS panel.

FORM.CONDITIONS

ID	CONDITIONAL EXPRESSION	Pass Nulls?
1	&5 + NULL(&6) >= 17000.00	YES
2	&5 + NULL(&6) < 17000.00	YES
	*** END ***	

Figure 175. Specifying conditional expressions for employees' raises

Now that you have defined the conditions you want to display on the report, use the FORM.DETAIL panel to create a report variation for each condition. This type of report formatting is called conditional formatting.

You can select a report variation without associating it with a specific condition, or associate it with a condition such as those you entered on the FORM.CONDITIONS panel.

6. On the QMF command line, enter:

```
SHOW FORM.DETAIL
```

The FORM.DETAIL panel displays.

- In the **Select Panel Variation** field, type C1 to associate this report format variation with the first condition on the FORM.CONDITIONS panel, where the employee makes greater than or equal to \$17,000.00. For this variation, do not enter any detail text.
- Create a report variation for the second condition. For this example, move the cursor to **Var 1 of 1**.
- Type 2 over the 1 and press Enter. (You can also enter NEXT on the QMF command line.)

A second FORM.DETAIL panel displays.

- In the **Select Panel Variation** field, type C2 to associate this report format variation with the second condition on the FORM.CONDITIONS panel, where the employee makes less than \$17,000.00.
- For this example, type **\*\*\* Needs Raise \*\*\*** for the first line of detail block text, as shown in the following figure. Change the alignment to column 60. QMF displays the text on the report when this condition is true.

```
FORM.DETAIL                                     Var 1 of 1

Include Column Headings with Detail Heading? ==> YES
LINE  ALIGN  DETAIL HEADING TEXT
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+
1     LEFT
2     LEFT
      *** END ***

New Page for Detail Block? ==> NO      Repeat Detail Heading? ==> NO
Keep Block on Page? ==> NO      Blank Lines after Block ==> 0
Put Tabular Data at Line (Enter 1-999 or NONE) ==> 1
LINE  ALIGN  DETAIL BLOCK TEXT
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+
1     60     *** Needs Raise ***
2     LEFT
      *** END ***

Select Panel Variation? ==> C2
```

Figure 176. Specify detail block text to appear when the condition is true.

- Change the width of the report so that the detail block text appears. On the QMF command line, enter:

```
SHOW FORM.OPTIONS
```

- In the **Report text line width** field, type 80 to change the report width to 80 columns.
- Press the Report function key to see the changed report, which is shown in the following figure.

ID	NAME	JOB	DEPT	SALARY	COMM	
10	SANDERS	MGR	20	18357.50	-	
20	PERNAL	SALES	20	18171.25	612.45	
30	MARENGHI	MGR	38	17506.75	-	
40	O'BRIEN	SALES	38	18006.00	846.55	
50	HANES	MGR	15	20659.80	-	
60	QUIGLEY	SALES	38	16808.30	650.25	
70	ROTHMAN	SALES	15	16502.83	1152.00	
80	JAMES	CLERK	20	13504.60	128.20	*** Needs Raise ***
90	KOONITZ	SALES	42	18001.75	1386.70	
100	PLOTZ	MGR	42	18352.80	-	
110	NGAN	CLERK	15	12508.20	206.60	*** Needs Raise ***
120	NAUGHTON	CLERK	38	12954.75	180.00	*** Needs Raise ***
130	YAMAGUCHI	CLERK	42	10505.90	75.60	*** Needs Raise ***
140	FRAYE	MGR	51	21150.00	-	
.						
.						
.						

Figure 177. This report uses conditional formatting to show employees who need a raise.

## Mixing tabular data with reformatted text

---

Complete the following tasks to mix tabular data and reformatted text within blocks of text.

### Procedure

1. Run the query and display the report.

For this example, run the following SQL query:

```
SELECT ACCTNO, COMPANY, NOTES  
FROM Q.SUPPLIER
```

2. On the QMF command line, enter:

```
SHOW FORM.COLUMNS
```

For this example, make the following changes on the FORM.COLUMNS panel:

- a. Specify an OMIT usage code for all columns except the NOTES column, because the NOTES column is the only column you will be displaying as tabular data.
  - b. Change the edit code for the NOTES column to CT, to allow text wrapping within the column, and change the width to 40.
3. On the QMF command line, enter:

```
SHOW FORM.DETAIL
```

4. Make the following changes on the FORM.DETAIL panel:

- a. In the **Include Column Headings with Detail Headings?** field, type NO so that the column headings will not appear after the detail heading text.
- b. In the **Blank Lines after Block** field, type 6.
- c. Specify the line number on which you want to display the tabular data in the **Put Tabular Data at Line** field. Specify different line numbers for the detail block text and the tabular data, or one will overlay the other when you display the report. For this example, type 4.
- d. Type the information for the detail block text. For this example, type Company : &2 for the first line because COMPANY is the second column that appears in the SELECT statement of the query. Type Account Number : &1 for the second line, because ACCTNO is the first column in the SELECT statement. Type Notes : for the third line.

The following figure shows these changes. You do not have to specify a form variable for the tabular data; the data will follow the last line of detail block text.

For more information on specifying detail block text, see the following figure.

```

FORM.DETAIL
  MODIFIED      Var 1  of 1

Include Column Headings with Detail Heading? ==> NO
LINE  ALIGN    DETAIL HEADING TEXT
----  -
1     LEFT
2     LEFT

      *** END ***

New Page for Detail Block? ==> NO      Repeat Detail Heading?
==> NO
Keep Block on Page?          ==> NO      Blank Lines after Block
==> 6
Put Tabular Data at Line (Enter 1-999 or NONE) ==> 4
LINE  ALIGN    DETAIL BLOCK TEXT
----  -
1     LEFT     COMPANY: &2;
2     LEFT     ACCOUNT NUMBER: &1;
3     LEFT     NOTES:

Select Panel Variation?      ==> NO

```

Figure 178. Specifying placement of tabular data with detail block text

5. Press the Report function key to see the changed report, which is shown in the following figure.

```

COMPANY: WESTCO, INC.
ACCOUNT NUMBER: 1100P
NOTES:
  THIS COMPANY HAS A STRONG HISTORY OF
  ON-TIME DELIVERY. WESTCO IS GROWING
  QUICKLY.

```

Figure 179. The changed report shows tabular data mixed with a detail block.

## Results

When you omit some columns from the report, as in this example, you reduce the automatic total width of the report. Make sure that your report is wide enough to include all of the detail block text. Change the width of your report by changing the **Report text line width** field on the FORM.OPTIONS panel.

### Related tasks

[Refining the format of your report with detail blocks](#)

You can reformat and add text to your report with detail blocks. A detail block is a set of specifications that tells QMF to put in whatever special formatting you want for one row of the data that is retrieved by your query.

## Showing totals across rows in a report

You can display a report that gives a total or average across the rows in a report by using column usage codes.

### About this task

Totals and averages are examples of aggregation functions, which are any functions that summarize the data in a column. You can also specify other aggregation usages, such as standard deviation, percent, or cumulative totals.

In this example, you use the ACROSS, GROUP, SUM, and OMIT usage codes to create a report that summarizes the salary, commission, and total earnings for each job description within every department.

### Procedure

To summarize report data:

1. Run the query to display the report:



For this example, run the following SQL query:

```
SELECT NAME, DEPT, JOB, SALARY, COMM, SALARY + COMM
FROM Q.STAFF
WHERE DEPT IN (15, 20, 38) AND JOB <> 'MGR'
ORDER BY DEPT, JOB
```

2. On the QMF command line, enter:

```
SHOW FORM.COLUMNS
```

The FORM.COLUMNS panel displays.

3. Type any changes to the column headings in the **COLUMN HEADING** field.

For this example, change the column heading for the column that is created by the query to **TOTAL\_EARNINGS**.

4. Type the usage codes and other changes for the columns. For this example, make the following changes:

- a. Specify **GROUP** for the **DEPT** column to group your data by department. To get totals in your report, you must group your data by at least one column. Be sure to order your data by that column.
- b. Specify **ACROSS** for the **JOB** column to summarize salary, commission, and total earnings by job.
- c. Specify **SUM** for the **SALARY**, **COMM**, and **TOTAL\_EARNINGS** columns.
- d. Specify **OMIT** for the **NAME** column, because you do not want to display it on the report. When you use the **GROUP** usage code, if you leave a column usage blank, that column does not appear on the report.
- e. Specify **D2** in the **EDIT** field for the **SALARY**, **COMM**, and **TOTAL\_EARNINGS** columns.
- f. Change the width of the **SALARY** column to 11.

The following figure shows these changes.

FORM.COLUMNS		MODIFIED				
NUM	COLUMN HEADING	USAGE	INDENT	WIDTH	EDIT	SEQ
1	NAME	OMIT	2	9	C	1
2	DEPT	GROUP	2	6	L	2
3	JOB	ACROSS	2	5	C	3
4	SALARY	SUM	2	11	D2	4
5	COMM	SUM	2	10	D2	5
6	TOTAL_EARNINGS	SUM	2	12	D2	6
	*** END ***					

Figure 180. Use **GROUP** and **ACROSS** usage codes to summarize data in a report.

5. On the QMF command line, enter:

```
SHOW FORM.OPTIONS
```

The FORM.OPTIONS panel displays.

6. In the **Automatic reordering of report columns?** field, type **YES**. If you do not make this change, your report displays with a warning at the top.
7. Press the Report function key to see the changed report, shown in the following figure.
8. For this example, press the Right function key to see the summary column.

DEPT	SUM SALARY	SUM COMM	SUM TOTAL EARNINGS	SUM SALARY	SUM COMM
15	\$24,766.70	\$316.70	\$25,083.40	\$16,502.83	\$1,152.00
20	\$27,757.35	\$254.70	\$28,012.05	\$18,171.25	\$612.45
38	\$24,964.50	\$416.50	\$25,381.00	\$34,814.30	\$1,496.80
	=====	=====	=====	=====	=====
	\$77,488.55	\$987.90	\$78,476.45	\$69,488.38	\$3,261.25

Figure 181. Pressing the Right function key displays the rest of the summarized data.

### Related concepts

**Usage codes** QMF usage codes can be entered in the USAGE field on QMF FORM.MAIN or FORM.COLUMNS to define how to use column data to produce reports and charts.

### Related reference

**Aggregation usage codes** You use aggregation usage codes to summarize data in a column or replace data with a calculation.

## Correcting errors on a form before displaying a report

You can use the CHECK command to check for errors on a form panel before you run the report.

### Procedure

To check a form panel:

1. Display the form panel you want to check for errors. You can display any form panel for a particular form. QMF checks for errors on that panel and all the other panels for that form as well.

QMF checks for two types of errors:

- Errors, which you must correct before you display the report
- Warnings, which you do not have to correct (however, uncorrected warning conditions can cause unexpected results when you display the report)

2. On a form panel, press the Check function key.

You can also enter CHECK on the QMF command line.

If QMF finds an error on a panel, it displays that panel with the field that caused the error highlighted. The message on the message line describes the error.

3. Correct the error.

Press the Help function key to see more information about the error and instructions to correct it.

4. Press the Check function key or enter CHECK to see any remaining errors.

### What to do next

After you correct any errors, QMF displays any warning conditions when you issue the CHECK command. The steps for correcting warning conditions are the same as for correcting errors.

## Saving the report form

If you want to display a report in the same format again, you can save your form in the database.

To use the report form again when you run a query, enter:

```
RUN QUERY queryname (FORM=formname
```

The data in a query must fit the form you use, or the report will not display.

## Related tasks

### Saving a QUERY, FORM, or PROC object

You can save a query, form, or procedure by entering the SAVE command on the QMF command line.

## Resetting the values on a form panel

---

You can reset the values on a form panel to the default values. This ability is useful if you entered values that do not produce the results you want; you can start over again using the default values.

### About this task

Remember that you can only reset the values on the FORM panels before you save the form in the database.

### Procedure

To reset the values on a form panel, choose one of the following options:

- To reset values on all form panels, on the QMF command line, enter:

```
RESET FORM
```

When you enter this command from the FORM.MAIN panel, you do not need to specify the object type as FORM. The object type defaults to FORM when the RESET command is entered from a form panel.

- To reset the values on a specific form panel, enter:

```
RESET FORM.panelname
```

For example, to reset the values on the FORM.COLUMNS panel, enter:

```
RESET FORM.COLUMNS
```

When you enter this command from any specific form panel, you do not need to specify the object type as FORM.*panelname*. The object type defaults to the type of panel you are on when the RESET command is entered from a specific form panel.

### Related reference

[RESET object](#)The RESET command restores an object in temporary storage to its initial state. This command does not apply to ANALYTIC objects.

## Printing your report

---

In addition to displaying your report online, you can print your report.

### About this task

The report must be in temporary storage. The rules for printing QMF reports vary depending on your operating environment and how you set up your printer. See your administrator if you need help printing reports.

With the CC (carriage control) keyword, you can suppress the carriage control characters in the report output when the name of the print device is not supplied. The CC keyword has these constraints:

- When the PRINTER keyword option is set to a printer name, the CC option is ignored.
- When the PRINTER keyword option is set to blank, the CC option can be set from the command line or from the **PRINT** panel prompt.

If you do not specify the CC keyword, carriage control is set by using the value of global variable DSQEC\_CC. You can set DSQEC\_CC to the value you need to use most of the time and then use the CC keyword to override that value when necessary.

## Procedure

1. To print a report, enter the following command on the QMF command line:

```
PRINT REPORT (PRINTER=printer
```

Where *printer* is the printer nickname that your administrator set up for you.

When you initiate the PRINT command from a report panel, you do not need to specify the object type in the command. From a report panel, the object type for the PRINT command defaults to REPORT.

2. From the command line, type the following to specify the CC keyword:

```
PRINT REPORT (CC = x
```

In this statement, *x* can be Y or N, as follows:

- Y specifies to leave the carriage control characters in place
- N specifies to remove the carriage control characters and shifts the report to the left by one space

CC can also be set from the **PRINT** panel prompt, which is shown in the following figure:

```
PRINT REPORT Command Prompt                                1 to 15 of 15
Printer (          ) GDDM nickname for the printer. (PRINTER must be specified to print
                    forms or prompted queries.)
Width   ( 132      ) Number of characters per printed line (22 to 999).
Length  ( 60       ) Number of lines per printed page (1 to 999, or CONT, to print without
                    page breaks. CONT cannot be used to print forms.
Datetime ( YES     ) Print the date and time? YES or NO.
Pageno  ( YES     ) Print the page numbers? YES or NO.
Form    (          )
                    Enter the name of a specific form to be used to print the report. To use the current form,
                    enter FORM or leave blank.
CC      (          ) Carriage Control? YES or NO or leave blank to use default.
-----
F1=Help  F3=End  F7=Backward  F8=Forward
```

Figure 182. Setting CC from the **PRINT** panel prompt

### Related reference

**PRINT in CICS**The PRINT command in CICS prints a copy of an object in the QMF temporary storage area or an object that is stored in the database.

**PRINT in TSO**The PRINT command prints a copy of an object in the QMF temporary storage area or an object that is stored in the database.

## Limiting the number of rows returned from a query

You can create a report that contains a limited number of ordered entries.

### Procedure

1. Construct an SQL query that selects and orders data as appropriate to your objective.
2. Run the query and specify the row limit value that gives the result you want.

The key elements are the ORDER BY clause of the SQL statement and the ROWLIMIT parameter of the RUN QUERY command.

### Example

For example, to create a report that contains the five managers with the most years of service, you could use the following query and QMF command.

SQL query:

```
SELECT NAME, YEARS
FROM Q.STAFF
WHERE JOB='MGR'
ORDER BY YEARS DESC
```

QMF command:

```
RUN QUERY (ROWLIMIT=5
```

Resulting report:

NAME	YEARS
JONES	12
QUILL	10
HANES	10
LU	10
LEA	9



---

## Chapter 7. Analyzing query results

Using QMF Analytics for TSO to analyze query results involves a series of user tasks.

### Task flow for using QMF Analytics for TSO

The following diagram illustrates the tasks associated with using QMF Analytics for TSO. Some of these tasks should be performed before using the application.

The actions associated with each task are explained in the sections that follow the diagram.

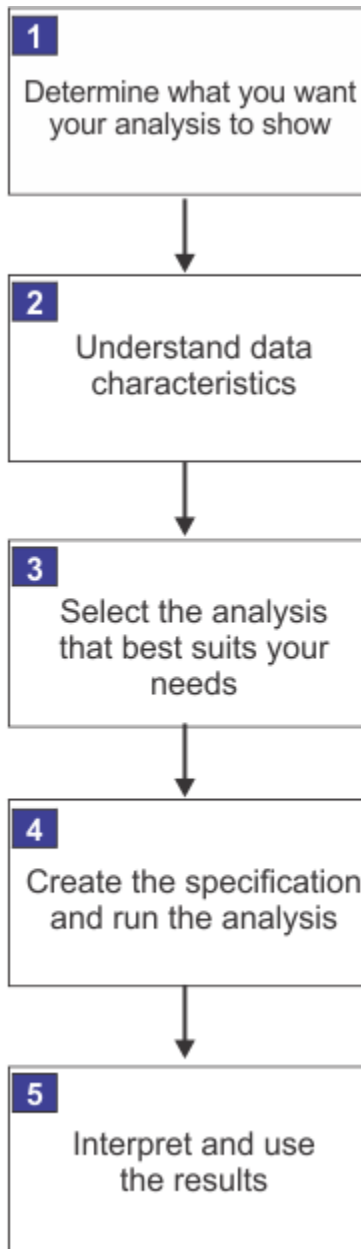


Figure 183. QMF Analytics for TSO - User task flow

### Determining the purpose of your analysis

Before using QMF Analytics for TSO you need determine the purpose of your analysis. If you do not have a clear understanding the purpose, the results returned by the analysis might very well be accurate and useless all at once.

Statistical analyses in-and-of themselves do not have purpose. Purpose is borne out of need, and needs vary from one business to another (and can even vary from one department to another). For example, sales departments gather and track data specific to revenue and performance, while human resource departments gather and track data specific to employee service, hire dates and employee benefits. There is no one-size fits all when it comes to analyzing data. The effectiveness of your analysis depends on you having a clear understanding of the purpose of your analysis.

When you have a clear understanding of the purpose of your analysis, you can use QMF Analytics for TSO to identify patterns and trends and then use that information to help you make better informed business decisions.

Stating your business problem as a question that can be answered with a statistical analysis is a good way of determining how to use QMF Analytics for TSO. You might want to test a hypothesis, search for possible trends, gather preliminary information, or refine a research question. Whatever your purpose, you must have a clear idea of what you want to find out before you select an analysis type. For example, suppose you sell car insurance, and you want to decide whether to continue a good driver discount for people who have taken an advanced driving course. You can begin your investigation by asking the question: *Does the cost of claims from drivers who have taken an advanced driving course differ significantly from the cost of claims from drivers who have not?*

When you have stated your business question, you are ready to choose a statistical analysis that can help answer your question. But before doing so, you need to know something about the assumptions that each analysis makes about your data.

To generate meaningful, data-driven conclusions about real statistical populations, the analysis must make some background assumptions. The validity of conclusions drawn from a statistical analysis depends on the validity of any assumptions made.

State your business question, and decide whether a statistical analysis can help answer this question.

## **Understanding data characteristics**

All data that can be analyzed includes characteristics related to:

- Number of observations
- Number of variables
- Scale of measurement

### **Number of observations**

In statistics, an observation is the value, at a particular period, of a particular variable. An observation can be a:

- physical quantity (height or weight)
- answer to a question (yes or no)
- sorting (ordered or disordered), and so on.

For example, an observation can be an individual price, or one of a number of individual prices, collected for an item at an outlet store.

To perform calculations, all statistical analyses require that the data being analyzed have a minimum number of observations. When determining which type of analysis to run against your data, take into consideration the number of observations.

### **Number of variables**

In statistics, a variable is any characteristic, number, or quantity that can be measured or counted.



Data is variable if the value associated with it varies between data units in a statistical population, and can change in value over time. Sales commission data can vary between data units in a statistical population (for example, the sales representatives in the statistical population being studied might not have the same commissions) and can also vary over time for each data unit (for example, sales commissions can increase or decrease from year to year or quarter to quarter).

### **Scale of measurement**

Measurement scales quantify or measure the variables in your data.

Each scale of measurement satisfies one or more of the following properties of measurement:

#### **Identity**

Each value on the measurement scale has a unique meaning.

#### **Magnitude**

Values on the measurement scale have an ordered relationship to one another. That is, some values are larger and some are smaller.

#### **Equal intervals**

Scale units along the scale are equal to one another. This means, for example, that the difference between 1 and 2 would be equal to the difference between 19 and 20.

#### **Absolute zero**

The scale has a true zero point, below which no values exist.

Different statistical analysis types require different scales of measurement with which to quantify data.

When you know the scale of measurement for your data, you can narrow down the analyses that you can use with it. You can use those statistical analyses developed for a lower scale of measurement with data on a higher scale of measurement. For example, the median assumes an ordinal scale measurement, but you can use it with interval and ratio scales as well. However, you cannot apply it to values that you have measured with a nominal scale.

The scales of measurement are:

#### **Nominal**

The word nominal means being such in name only. When applied to the field of statistics, a nominal measurement is one that satisfies the identity property of measurement only.

Nominal measurements make no assumptions about the values assigned to the data. Each value serves simply as a label or name for that distinct category and does not imply anything about the order, magnitude, or distance between numbers.

For example, you might assign a numerical code to designate cities in your data table: Paris is 1, New York is 2, Bonn is 3, and so forth. The numbers simply represent the actual names of the cities and have no inherent numerical value with respect to magnitude. Because these numbers are arbitrary, you cannot perform any arithmetic calculations, such as adding or dividing, and get a meaningful result. For example, an average of your codes, 2.6, tells you nothing meaningful about the cities. You can, however, find the mode, which is the city that occurs most often in the data.

With nominal measurements in your data, you cannot use a statistical analysis type that assumes ordered or meaningful numerical distances between the categories.

#### **Ordinal**

The word ordinal means “of or pertaining to order, rank, or position in a series.” When applied to the field of statistics, an ordinal measurement is one that satisfies the identity and magnitude properties of measurement.

Ordinal measurements assume that the numbers assigned have significance in terms of rank or importance. For example, a consumer survey might rank responses so that the item most preferred is 1, the next item is 2, and the item least preferred is 3.

Each category and its numerical value have a unique position relative to the other categories and their numerical values. This scale tells you that 1 is greater than 2 and that 2 is greater than 3 in terms of

your scale of consumer preferences. However, you do not know by how much 1 is greater than 2; all you know is that it is greater.

Because ordering is the sole mathematical property of this scale, you cannot assume anything about the distance between the values. For instance, you could not say that the degree of preference between 1 and 2 was the same as the degree of preference between 2 and 3. Therefore, you cannot use statistical analyses that assume meaningful numerical distances between the ranking. You can, however, find the median. This is the central value when the data is ordered.

### **Interval**

The word interval means “the totality of points on a line between two designated points or endpoints that may or may not be included.” When applied to the field of statistics, an interval measurement is one that satisfies identity, magnitude, and equal interval properties of measurement.

This scale of measurement contains units that are in a particular order and are of equal distance from each other. For example, a thermometer uses a scale that measures temperature by degrees. This scale is made up of equal temperature units, so the difference between 30 and 40 degrees is the same as the difference between 80 and 90 degrees. Because of the property of equal distances, you can perform arithmetic calculations, such as adding and dividing, to get meaningful results.

Interval scales lack a true zero point. For the thermometer of our example, the zero point is arbitrarily set for the Celsius and Fahrenheit scales.

The lack of a true zero point means that you cannot make statements about proportional relationships on the scale. For example, someone with an IQ of 100 is not twice as smart as someone with an IQ of 50. Because it makes no sense to say that someone has zero intelligence, IQ scales lack a true zero point.

### **Ratio**

The word ratio means “the relation between two similar magnitudes with respect to the number of times the first contains the second.” When applied to the field of statistics, a ratio measurement is one that satisfies all four of the properties of measurement: identity, magnitude, equal intervals, and a zero point.

You can make ratio judgments based on this scale. For example, because time and distance are ratio measures, you can say that a car traveling at 50 miles per hour is traveling twice as fast as a car traveling at 25 miles per hour.

Ratio-scale measurements satisfy all the conditions of a real number system, and you can use them with any mathematical operation. While this scale of measurement is common in business, very few business statistics require all of its properties.

## **Selecting the analysis that best suits your needs**

You need to consider several attributes in your data as determinants for selecting an appropriate statistical analysis.

These attributes include:

- Number of observations
- Number of variables
- Scale of measurement

After you understand the aspects of scale of measurement, number of variables, and number of observations in the data to be analyzed, you can narrow down your choices of which analyses to use.

Use the following table to help you choose a statistical analysis type that is appropriate for the task you are performing.

Table 12. Tasks and data characteristics

Nature of task	Scale of measurement	Variables	Observations	Analysis type
Summarizing and exploring data	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	1 +	2	Basic Statistics
Measuring the degree of association between variables	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	2+	3	Linear Trend
Summarizing the relationships among two or more variables	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	1	(3 years for seasonal)	Univariate
	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	2	3	Bivariate
	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	1	(3 years for seasonal)	Linear
Exploring cash flows	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	1 or more	1	Discounted cash flow
Describing relationships among variables	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	2+	2 (3 if you specify origin)	Linear Statistics
	<ul style="list-style-type: none"> <li>Interval</li> <li>Ratio</li> </ul>	3+	Number of independents +2 (+3 if you specify origin)	Bivariate

## Creating the specification and running the analysis

The specification for your analysis is a set of instructions that you pass to QMF Analytics for TSO. The specification contains all of the information that QMF Analytics for TSO needs to analyze data and generate output.

You create the specification for your analysis by selecting the analysis type and setting its parameters.

The parameter selection process in QMF Analytics for TSO integrates with the querying capabilities of QMF for TSO by providing access to the column names in the query results. See [Running an analysis using QMF for TSO and QMF Analytics for TSO](#) for an illustration and explanation of the process.

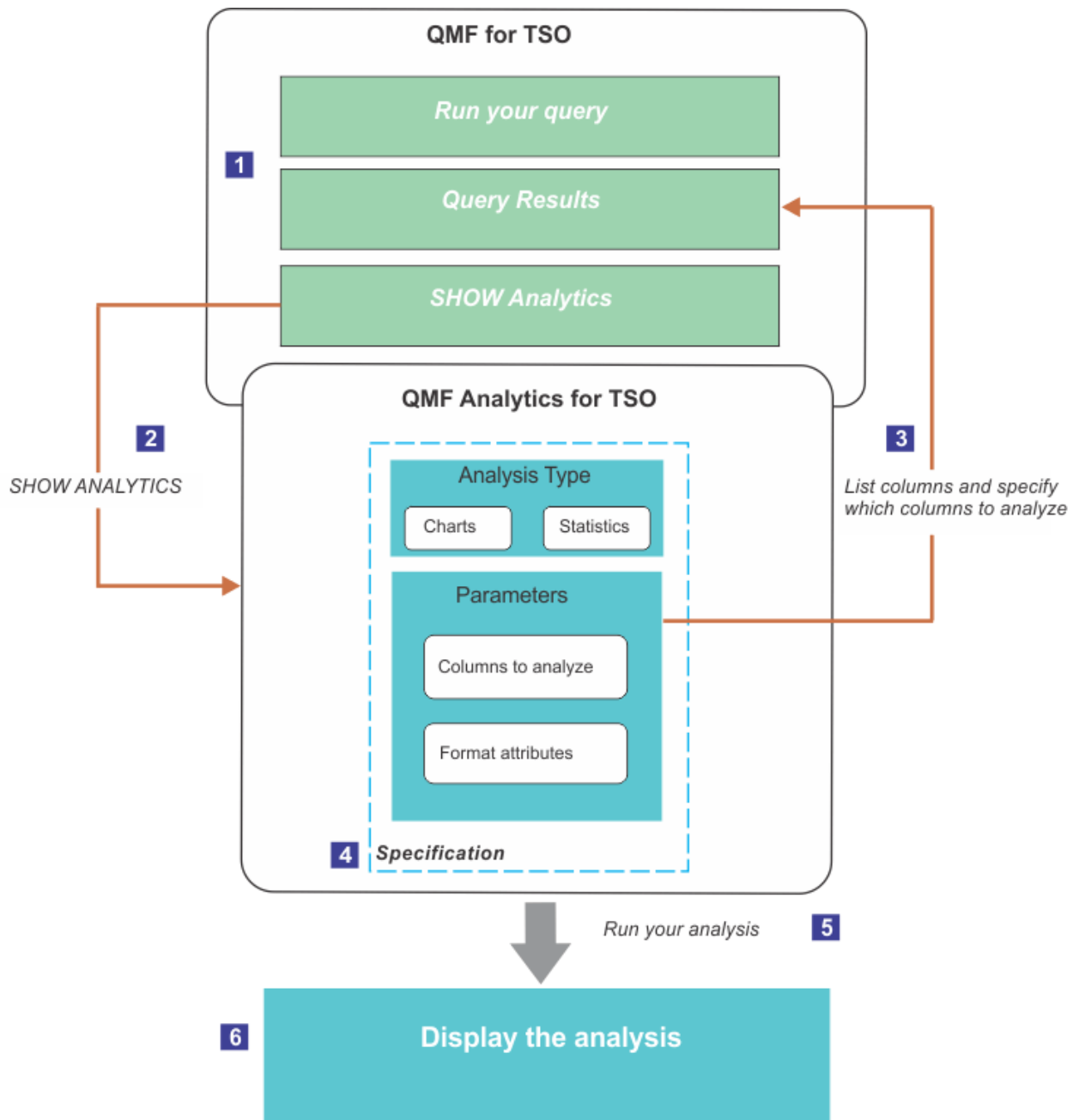


Figure 184. Running an analysis using QMF for TSO and QMF Analytics for TSO

Running an analysis using QMF for TSO and QMF Analytics for TSO shows the task flow for using QMF for TSO and QMF Analytics for TSO to analyze query results. Each step in the process is described in the following list:

1. Using QMF for TSO, you run a query and QMF returns the query result in a multicolumn report.
2. From the result page, you enter **SHOW ANALYTICS** on the QMF command line to show the **QMF Analytics for TSO Home** panel.

The Home panel has two sections that list the analysis types. One section lists the types of charts you can create from your query results and the other lists the types of statistics you can run against your query results.

3. Once you have selected which type of analysis to run, you set the parameters for the analysis.

For any column field on the **Parameter Selection** panel, you can press the List function key (F4) for list of column names from the query result. QMF Analytics for TSO displays a list of the columns and a descriptor of its data type in a Column Selection window.

Query result column data can be of the following types:

- **D** for decimal numeric data
- **I** for integer numeric data
- **C** for character data

You also use the Parameter Selection panel to set formatting attributes for the chart or statistical analysis.

**Note:** An asterisk (\*) denotes columns with unsupported data types.

4. A completed **Parameter Selection** panel represents the specification for the analysis, which:

- Defines the type and format of the analysis
- Lists the columns from the query result to be analyzed
- Contains format-related settings that affect the visual characteristics of the output.

If you know that you will run the analysis frequently or multiple times, you can use the Save function to save the specification as an ANALYTIC object in the QMF database (all Parameter Selection panels have a Save function).

Saving the specification to the QMF database allows you to run your analysis directly from the QMF command line, without having to navigate to the QMF Analytics for TSO Parameter Selection panel and reenter the parameters.

Each **Parameter Selection** panel has a Run function key.

Selecting the Run function passes the specification to QMF Analytics for TSO. QMF Analytics for TSO interprets the specification, analyzes the column data from the query results, performs the calculations specific to the analysis type.

5. QMF for TSO displays the analysis in the form of a chart or statistical graph, depending on the analysis type selected.

## Interpreting and using the results

After running an analysis, your main task is to interpret and understand the results of the analysis.

Some analysis types can help you see and understand relationships in the data that you might not have realized prior to running the analysis.

Interpreting results involves examining the results and using functions in QMF Analytics for TSO toggle to various views and to work with results dynamically.

## Charting QMF query results with QMF Analytics for TSO

---

Use the information in this section to learn how to use QMF Analytics for TSO to chart query results.

### Before you begin

To use an example topic as a hands-on exercise, you must have QMF for TSO installed and QMF Analytics for TSO configured for use. The examples make use of sample data tables that are included with QMF for TSO.

## About this task

Table 13 on page 154 lists the chart types supported by QMF Analytics for TSO. The table describes the each chart type and indicates when it would be appropriate to use the chart. The table also includes links to examples that show how to create the chart.

Each topic linked to from the table demonstrates:

- How to run a query from QMF for TSO
- How to start QMF Analytics for TSO and select a chart type
- How to create the specification for your chart by using the Parameter Selection panel
- How to run the specification to generate the chart

QMF Analytics for TSO has its own online help system and context-sensitive help for panels and fields. If you want to learn more about a field or panel as you go through the examples, use the Help and More Help functions.

<b>Chart</b>	<b>When would I use this type of chart?</b>	<b>For an example, see</b>
Histogram	<p>Use histograms when showing the frequency of occurrence is your main objective.</p> <p>For example, if you wanted to show frequency of accidents based on the time of day that they occurred. A histogram would have intervals of time.</p>	<p><a href="#">“Charting query results as a histogram” on page 155</a></p>
Maps	<p>Maps are commonly used in statistics to highlight data comparisons across different regions within a defined geography.</p> <p>For example, you might want to see how sales revenue in the United States breaks down on a state-by-state basis. Presenting sales revenue on a map is an effective way to communicate information to your sales representatives and executives.</p>	<p><a href="#">“Charting query results as a map” on page 161</a></p>
Mixed	<p>Mixed charts show multiple values against a single axis range, show two chart types together, or compare two continuous summary values.</p> <p>By combining different types of elements on a single chart, mixed charts can help users interpret two or three different measures on a single chart.</p>	<p><a href="#">“Charting query results as a mixed chart” on page 169</a></p>
Pie	<p>Pie charts are used to show the size relationship between the parts of a whole.</p> <p>If a company has five divisions, and the pie chart shows profits per division, the sum of all the slices/divisions is the total profits of the company.</p>	<p><a href="#">“Charting query results as a pie chart” on page 174</a></p>

Table 13. Types of charts you can create using QMF Analytics for TSO (continued)

Chart	When would I use this type of chart?	For an example, see
Plot	<p>Plot charts show specific points in your data and connect the points with lines. By connecting data points with lines you can see the rate of change (in the form of a line sloping upwards or downwards) from point-to-point.</p> <p>For example, you could use a plot chart to plot a river's discharge rate (volume of water that flows through it in a given time). The chart would show the flow rate increase or decrease by month, which could help you develop environmental studies or understand the impact of discharge rate on the environment.</p> <p>Use this chart type to plot more than one set of data on the same chart by specifying multiple Y variables.</p> <p>Each Y variable may be plotted as a narrow or wide line. In this way you can draw attention to specific variables.</p>	<p><a href="#">“Charting query results as a plot chart” on page 178</a></p>
Tower	<p>Tower charts show the frequency of data items in successive numerical intervals of equal size, or show a comparison of categorical values in a query result.</p> <p>For example, you might want to chart this type of data to understand which departments in a company have the highest cost.</p>	<p><a href="#">“Charting query results as a tower chart” on page 183</a></p>

## Charting query results as a histogram

You can chart your query results as a histogram by using QMF Analytics for TSO.

### About this task

Histograms are commonly used for showing *frequency distributions* and for presenting data categorically in the form of a vertical bar chart.

The type of histogram (bar versus frequency) that QMF Analytics for TSO generates depends on the type of data in the independent variable (horizontal x-axis) and other factors related to the data.

In this example you will query product sales data and chart how many units your sales representatives are selling.

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.SALES** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

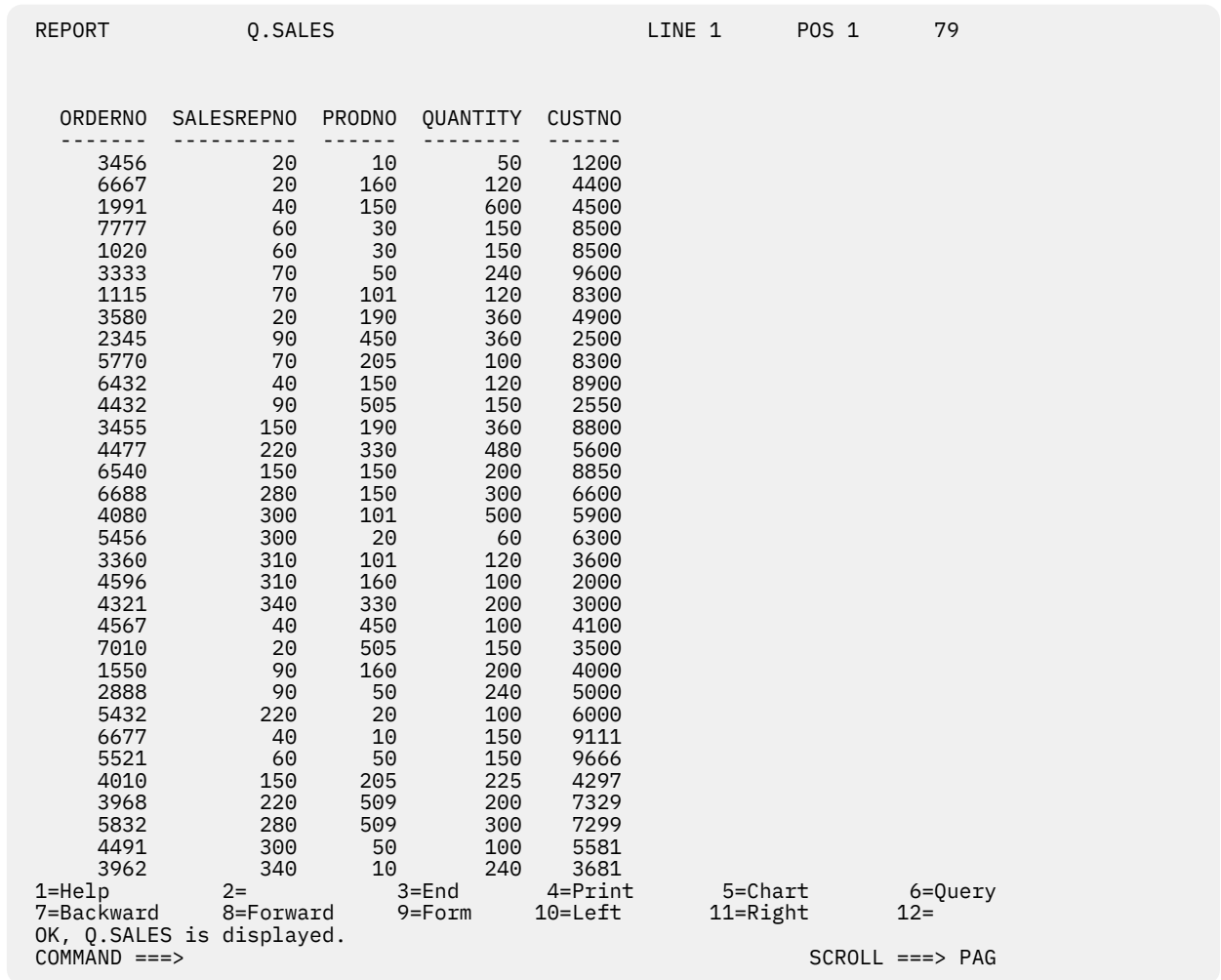


Figure 185. Query Results

- Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
- Tab to **Histogram** in the Charts section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

- Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Table 14. Parameter Settings for your Histogram	
Parameter	Setting
Specify the column to form the vertical coordinates (y-axis)	QUANTITY
Specify the column to form the horizontal coordinates (x-axis)	SALESREPNO



Parameter	Setting
Specify the column to control grouping of your data	Leave this field blank
Tabulation?	No
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

The following illustration shows the histogram parameters and the corresponding columns from the query result.

The screenshot shows two windows. The top window, titled 'ANALYTICS -- HISTOGRAM', displays 'Parameter Selection' with three prompts: 'Specify the column to form the vertical coordinates (y-axis)' with '( SALARY )' entered, 'Specify the column to form the horizontal coordinates (x-axis)' with '( DEPT )' entered, and 'Specify the column to control grouping of your d' with '( )' entered. The bottom window, titled 'REPORT', shows a table with columns: ID, NAME, DEPT, JOB, YEARS, SALARY, and COMM. Red arrows point from the 'SALARY' parameter in the top window to the 'SALARY' column in the report, and from the 'DEPT' parameter to the 'DEPT' column.

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	SANDERS	20	MGR	7	18357.50	-
20	PERNAL	20	SALES	8	18171.25	612.45
30	MARENGHI	38	MGR	5	17506.75	-
40	O'BRIEN	38	SALES	6	18006.00	846.55
50	HANES	15	MGR	10	20659.80	-
60	QUIGLEY	38	SALES	-	16808.30	650.25
70	ROTHMAN	15	SALES	7	16502.83	1152.00
80	JAMES	20	CLERK	-	13504.60	128.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
100	PLOTZ	42	MGR	7	18352.80	-
110	NGAN	15	CLERK	5	12508.20	206.60
120	NAUGHTON	38	CLERK	-	12954.75	180.00
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
140	FRAYE	51	MGR	6	21150.00	-
150	WILLIAMS	51	SALES	6	19456.50	637.65

Figure 186. Query Results and Parameter Selection

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable `DSQEC_SESSGLV_SAV` in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

7. Press the Run function key (F2).

QMF Analytics for TSO generates the histogram based on the parameters you selected.

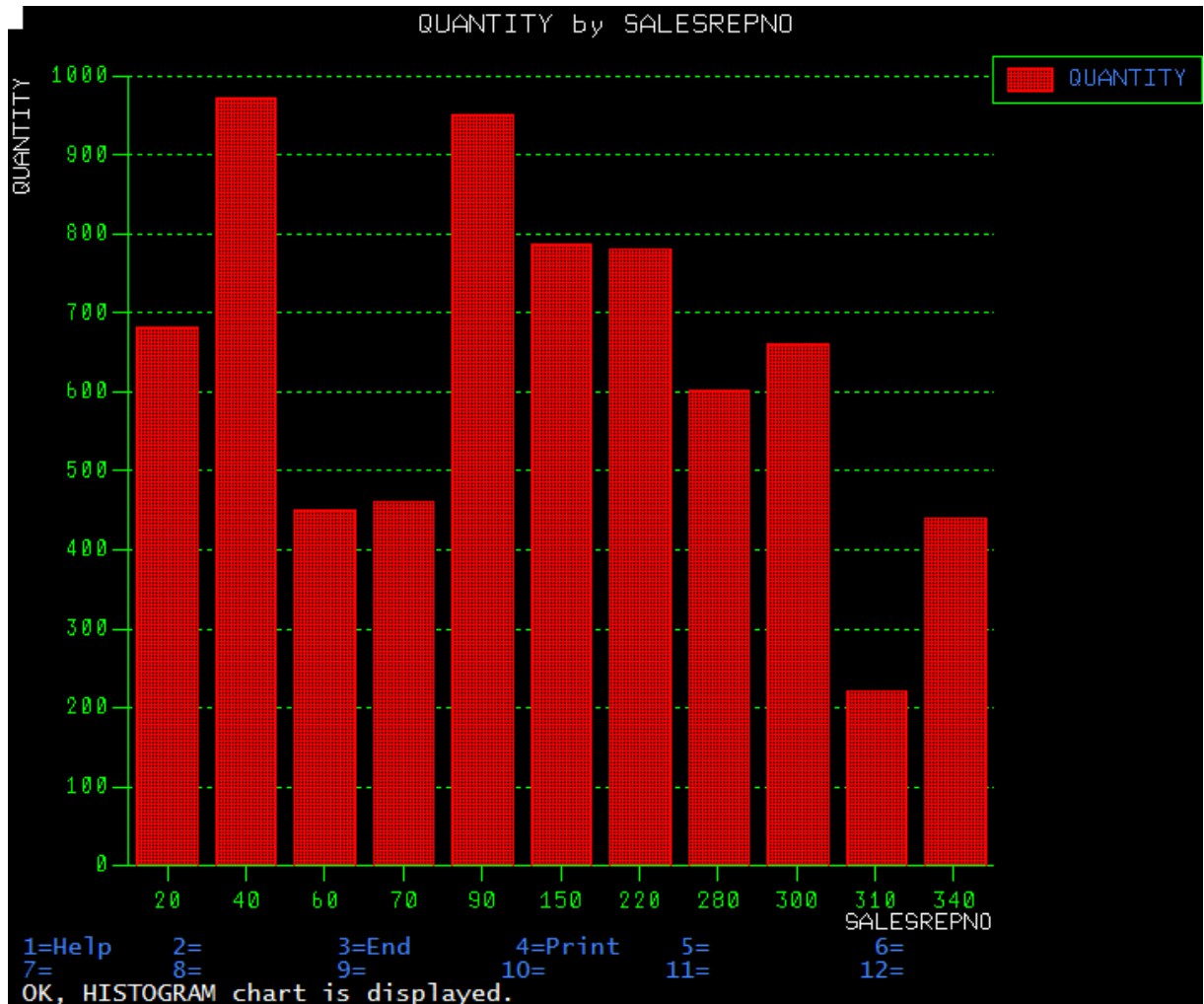


Figure 187. Histogram of query results

The quantity of units sold is plotted on the vertical axis (the Y axis). The sales representative ID number is plotted on the horizontal axis (the x axis). The chart illustrates that sales representative 40 has sold the most units, while sales representative 310 has sold the least.

8. Optional: If you have configured print capabilities in QMF, you can print the chart.
9. Press the End function key (F3) to return to the Parameter Selection panel.
10. Tab to the **Tabulation?** field and change the setting from No to Yes.
11. Press the Run function key (F2) to generate the histogram with tabulation.

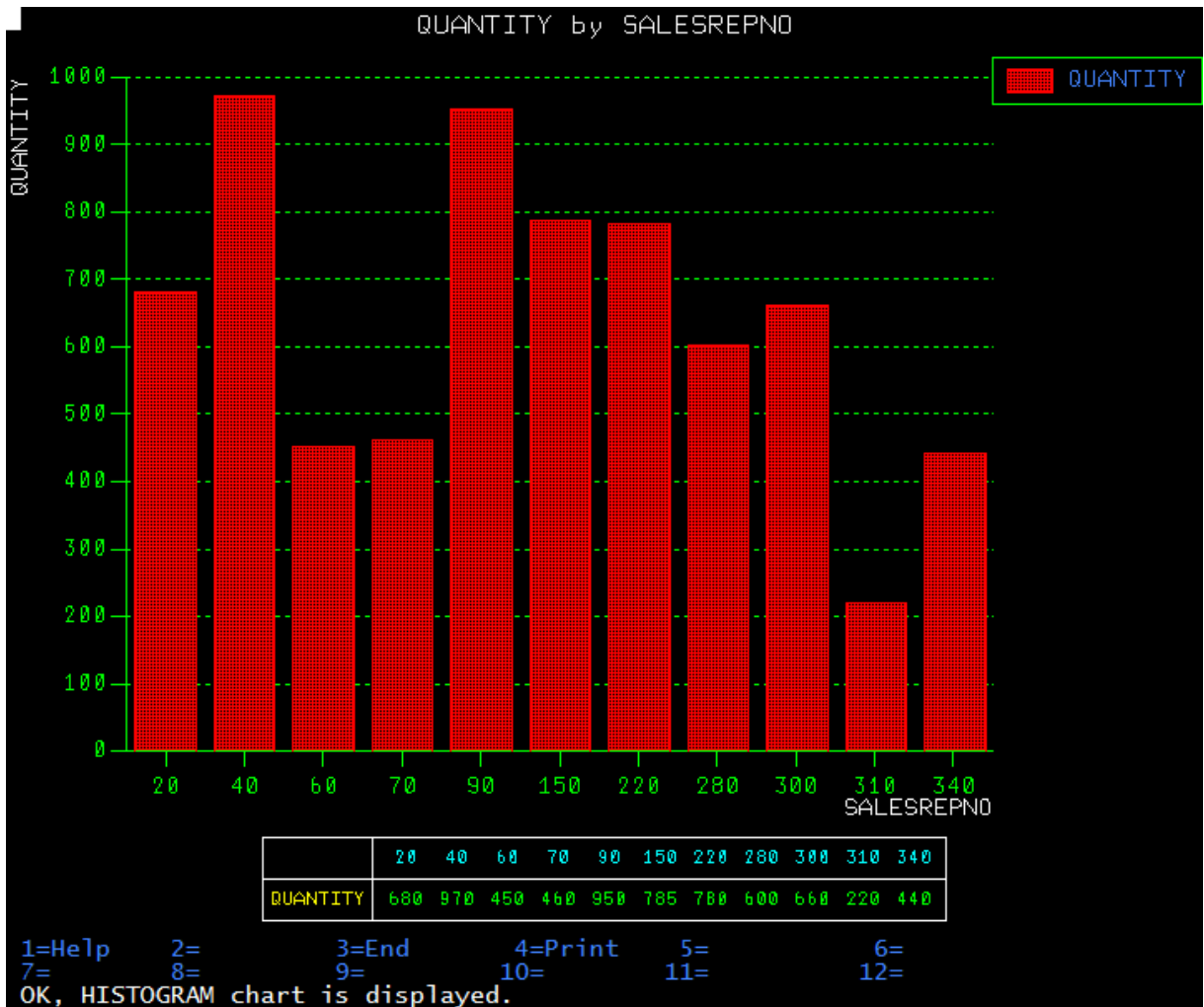


Figure 188. Histogram of query results with tabulation

QMF Analytics for TSO generates the histogram and includes the tabulated results under the horizontal axis.

12. Press the End function key (F3) to return to the Parameter Selection panel.

13. To demonstrate the Grouping function: and select MONTH.

a) Tab to the **Specify the column to control grouping of your data** field select the List function key (F4).

b) Tab to **PRODNO** and press Enter.

By adding a column to the **Specify the column to control grouping of your data** you are telling QMF Analytics for TSO to add an additional level of grouping to the analysis.

c) Tab to **Tabulation?** and select No.

Your **Parameter Selection** panel should look like this:

```

ANALYTICS -- HISTOGRAM

                                Parameter Selection

Specify the column to form the vertical coordinates (y-axis)
      ( QUANTITY                               )

Specify the column to form the horizontal coordinates (x-axis)
      ( SALESREPNO                             )

Specify the column to control grouping of your data
      ( PRODNO                                 )

Tabulation?
  Yes / No

Chart Title ==> *

1=Help      2=Run      3=End      4=List      5=Save      6=
7=          8=          9=          10=         11=         12=

```

Figure 189. Parameter Selection showing Grouping Column

d) Press the Run function key (F2).

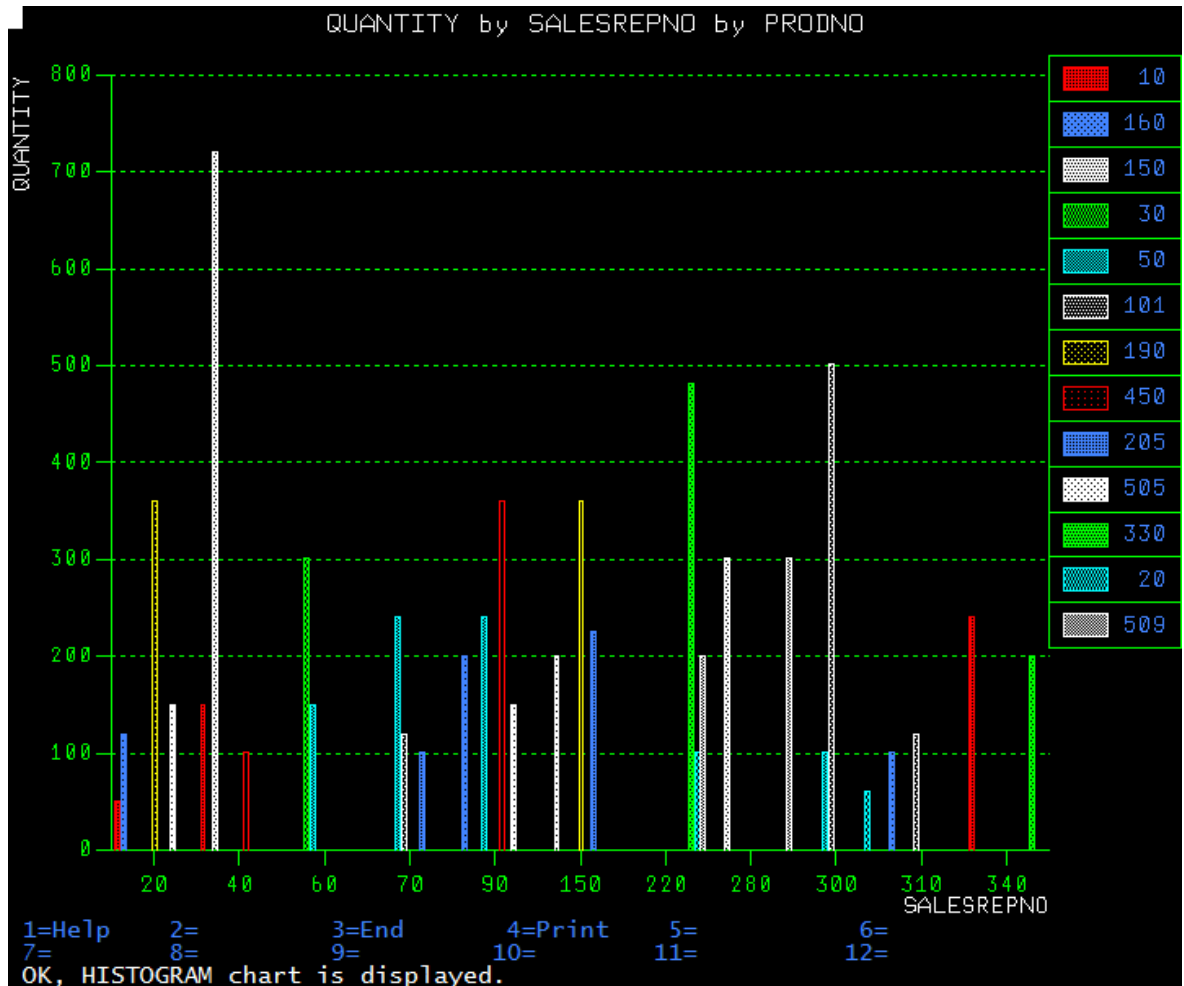


Figure 190. Histogram of query results charting rainfall by year and by month

Quantity is still plotted on the vertical axis. But data along the horizontal axis now includes an additional layer of information. The horizontal axis “groups” the sales representative ID with the product identifiers (PRODNO) .

14. Optional: If you have configured print capabilities in QMF, you can print the chart.
15. Press the End function key (F3) to return to the Parameter Selection panel.
16. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Charting query results as a map

You can chart query results as a map by using QMF Analytics for TSO.

### About this task

Maps render query results to a defined area.

In this example you want to query climate information for the United States and determine which states receive the most rain. You might want this type of information to help you determine in which region to increase your inventory for foul weather gear sales.

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_USA** on the command line and press Enter.  
QMF for TSO displays the query result as a multi-columned report.

MONTH	STATE	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
1	AK	9	25	1	1
1	AL	23	61	11	34
1	AR	19	50	12	1
1	AZ	12	68	7	80
1	CA	19	72	18	41
1	CO	-13	50	1	51
1	CT	18	39	3	1
1	DE	25	37	3	11
1	FL	34	70	12	83
1	GA	25	55	9	45
1	HI	63	82	7	133
1	IA	-6	27	4	1
1	ID	12	50	4	29
1	IL	9	36	2	10
1	IN	9	32	4	1
1	KS	9	54	3	50
1	KY	18	36	6	1
1	LA	27	63	10	51
1	MA	12	39	5	1
1	MD	19	39	9	1
1	ME	5	28	4	1
1	MI	9	34	3	1
1	MN	-2	19	3	1
1	MO	9	37	10	1
1	MS	28	57	6	44
1	MT	1	39	5	1
1	NC	14	50	8	23
1	ND	-6	23	3	1
1	NE	3	41	3	10
1	NH	10	32	1	1
1	NJ	21	41	2	11
1	NM	5	54	4	48
1	NV	10	57	2	58

1=Help 2= 3=End 4=Print 5=Chart 6=Query  
7=Backward 8=Forward 9=Form 10=Left 11=Right 12=  
OK, Q.CLIMATE\_USA is displayed.  
COMMAND ==> SCROLL ==> PAGE

Figure 191. Query Results

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Map** in the Charts section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Specify the column which contains the data to be mapped	RAINFALL
Specify the column which links the data to the map coordinates	STATE ID
Specify the location to be mapped	United States

Table 15. Parameter Settings for Map (continued)	
Parameter	Setting
Chart Title	<p>Enter a title or accept the default asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2) to generate the map.

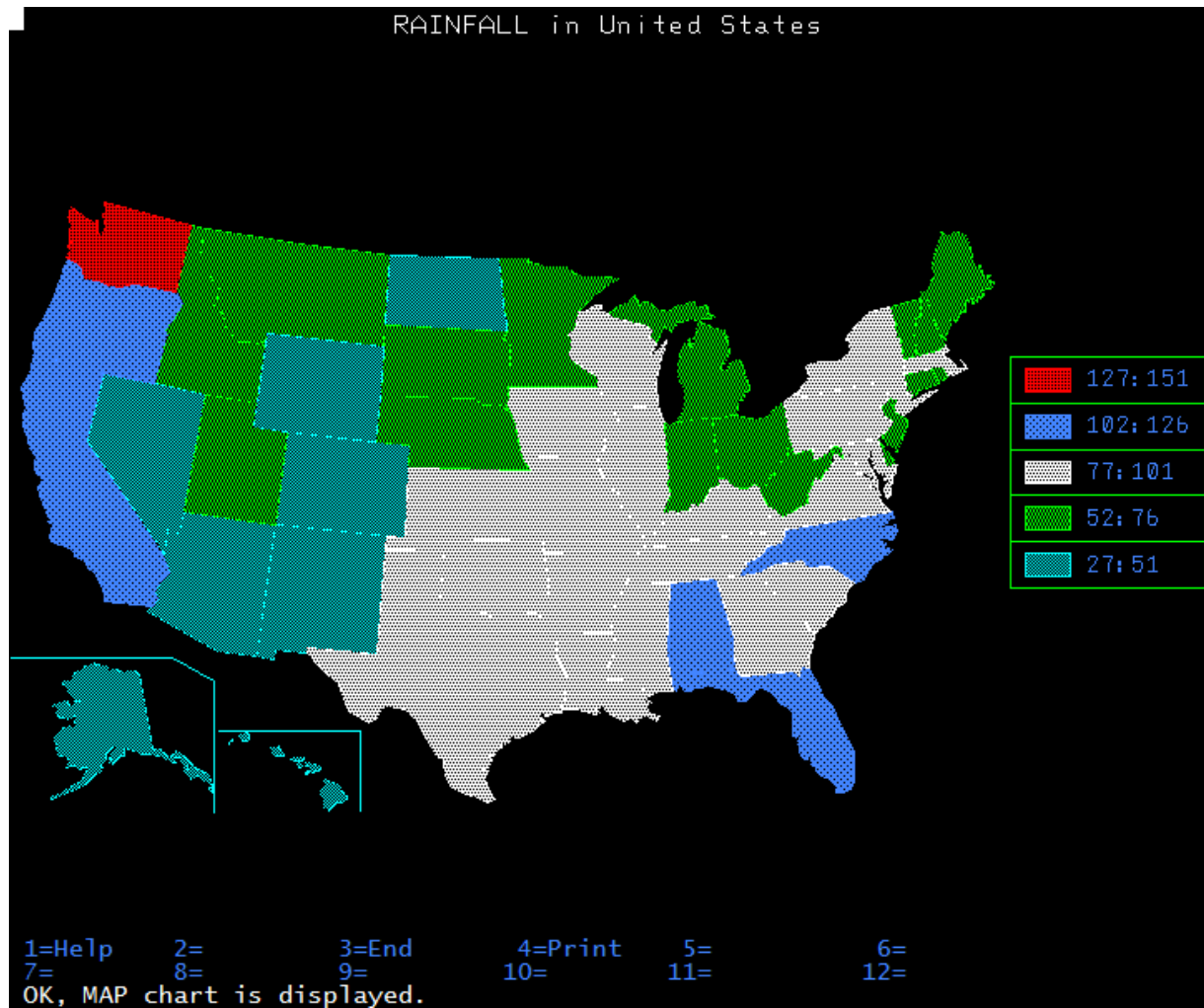


Figure 192. Map of Rainfall in the United States

The map generated is color-coded and includes a key for associating colors to rainfall amount. In this example you can see that Washington state receives the highest amount of rainfall of all the states

8. Press the End function (F3) key to return to the **Parameter Selection** panel.
9. Press the End function key again to return to the QMF Analytics for TSO Home panel.
10. Press the End function key again to return to the query results.
11. Enter DISPLAY Q.WORLDINFO on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

REPORT	Q.WORLDINFO			LINE 1	POS 1	79
COUNTRYID	ALPHA3	ALPHA2	COUNTRYNAME			
4	AFG	AF	Afghanistan			
8	ALB	AL	Albania			
12	DZA	DZ	Algeria			
20	AND	AD	Andorra			
24	AGO	AO	Angola			
28	ATG	AG	Antigua and Barbuda			
31	AZE	AZ	Azerbaijan			
32	ARG	AR	Argentina			
36	AUS	AU	Australia			
40	AUT	AT	Austria			
44	BHS	BS	Bahamas			
48	BHR	BH	Bahrain			
50	BGD	BD	Bangladesh			
51	ARM	AM	Armenia			
52	BRB	BB	Barbados			
56	BEL	BE	Belgium			
60	BMU	BM	Bermuda			
64	BTN	BT	Bhutan			
68	BOL	BO	Bolivia			
70	BIH	BA	Bosnia and Herzegovina			
72	BWA	BW	Botswana			
76	BRA	BR	Brazil			
84	BLZ	BZ	Belize			
90	SLB	SB	Solomon Islands			
92	VGB	VG	Virgin Islands, British			
96	BRN	BN	Brunei			
100	BGR	BG	Bulgaria			
104	MMR	MM	Myanmar			
108	BDI	BI	Burundi			
112	BLR	BY	Belarus			
116	KHM	KH	Cambodia			
120	CMR	CM	Cameroon			
124	CAN	CA	Canada			

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.WORLDINFO is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 193. Query Results

12. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
13. Tab to **Map** in the Charts section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

**Note:** The fields on the **Parameter Selection** panel contain the values from the previous mapping operation.

14. Set the parameters for your map as shown the following table:



Table 16. Parameter Selection for mapping world sales by country	
Field name	Parameter to set
Specify the column which contains the data to be mapped	SALES
Specify the column which links the data to the map coordinates	COUNTRY ID
Specify the location to be mapped	World
Chart Title	Enter a title or accept the default asterisk (*).  If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.  If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.

15. Press the Run function key (F2) to generate the map.

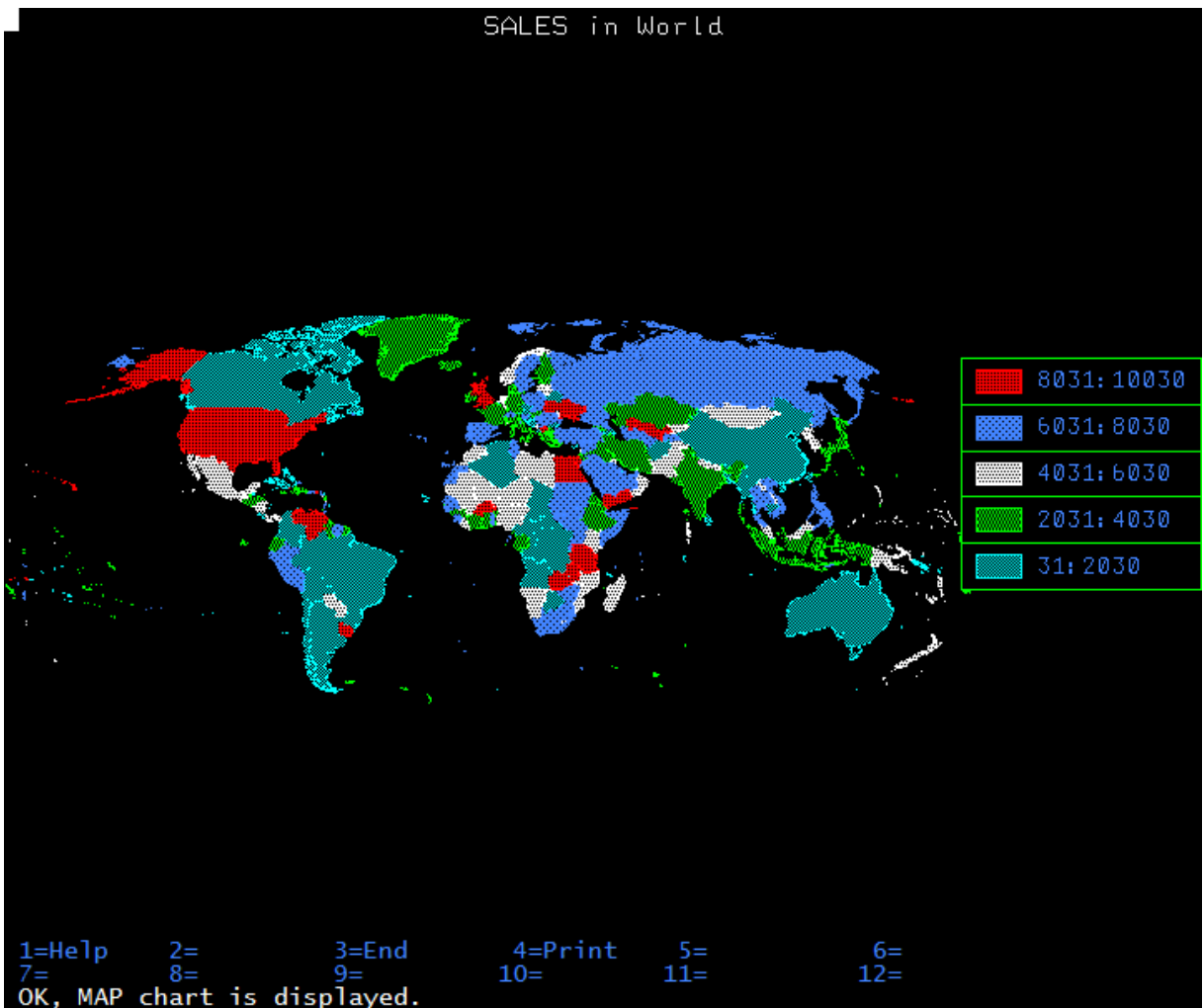


Figure 194. World Sales Data

The map generated is color-coded and includes a key for associating colors to sales revenue around the globe. In this example you can see that sales are highest in the United States.

16. Optional: If you have configured print capabilities in QMF, you can print the map.

17. Press the End function key (F3) to return to the Parameter Selection panel.
18. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Charting query results as a user-defined map

You can chart query results as a map by using QMF Analytics for TSO.

### About this task

Maps render query results on to defined area.

This topic provides an example for charting information geographically by using a user-defined map.

In the example you map rainfall data from a climate query to a defined geographic region (eight states of the United States). Say you worked for a company that manufactured and distributed foul-weather gear. A map that shows which of eight states have the heaviest rainfall might be useful for communicating and explaining sales strategy.

This procedure uses the Q . CLIMATE\_USA and Q . MAPUSA sample tables.

### Procedure

1. Start QMF for TSO.  
Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.
2. From the **QMF Home Panel**, enter **DISPLAY Q . CLIMATE\_USA** on the command line and press Enter.  
QMF for TSO displays the query result as a multi-columned report.

MONTH	STATE	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
1	AK	9	25	1	1
1	AL	23	61	11	34
1	AR	19	50	12	1
1	AZ	12	68	7	80
1	CA	19	72	18	41
1	CO	-13	50	1	51
1	CT	18	39	3	1
1	DE	25	37	3	11
1	FL	34	70	12	83
1	GA	25	55	9	45
1	HI	63	82	7	133
1	IA	-6	27	4	1
1	ID	12	50	4	29
1	IL	9	36	2	10
1	IN	9	32	4	1
1	KS	9	54	3	50
1	KY	18	36	6	1
1	LA	27	63	10	51
1	MA	12	39	5	1
1	MD	19	39	9	1
1	ME	5	28	4	1
1	MI	9	34	3	1
1	MN	-2	19	3	1
1	MO	9	37	10	1
1	MS	28	57	6	44
1	MT	1	39	5	1
1	NC	14	50	8	23
1	ND	-6	23	3	1
1	NE	3	41	3	10
1	NH	10	32	1	1
1	NJ	21	41	2	11
1	NM	5	54	4	48
1	NV	10	57	2	58

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_USA is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 195. Query Results

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Map** in the Charts section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

In the **Parameter Selection** panel, set cursor to the field **Specify the location to be mapped**, press F4 to open the **Coordinate Table Selection** dialog, and press F4 (New) to create a user-defined map.

In the **User Coordinate Table Selection** dialog, specify the table that you want to use for building the map:

- a. Select the database from that you want to use the coordinate table. If you select TSO, specify the CSV delimiter and header options.
- b. In the **Short Name** field enter the name for the table that you want use for it in the **Coordinate Table Selection** dialog.
- c. In the **Coordinate table name** field, type the table name in the database.
- d. Press F2 to save the user-defined map and return to the **Parameter Selection** panel.

Table 17. Parameter Settings for User-defined Map	
Parameter	Setting
Table Location	Db2

<i>Table 17. Parameter Settings for User-defined Map (continued)</i>	
<b>Parameter</b>	<b>Setting</b>
Short Name	USERMAP
Coordinate Table Name	Q.MAPUSA

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

<i>Table 18. Parameter Settings for Map</i>	
<b>Parameter</b>	<b>Setting</b>
Specify the column which contains the data to be mapped	RAINFALL
Specify the column which links the data to the map coordinates	STATE
Specify the location to be mapped	USERMAP
Chart Title	<p>Enter a title or accept the default asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2) to generate the map.

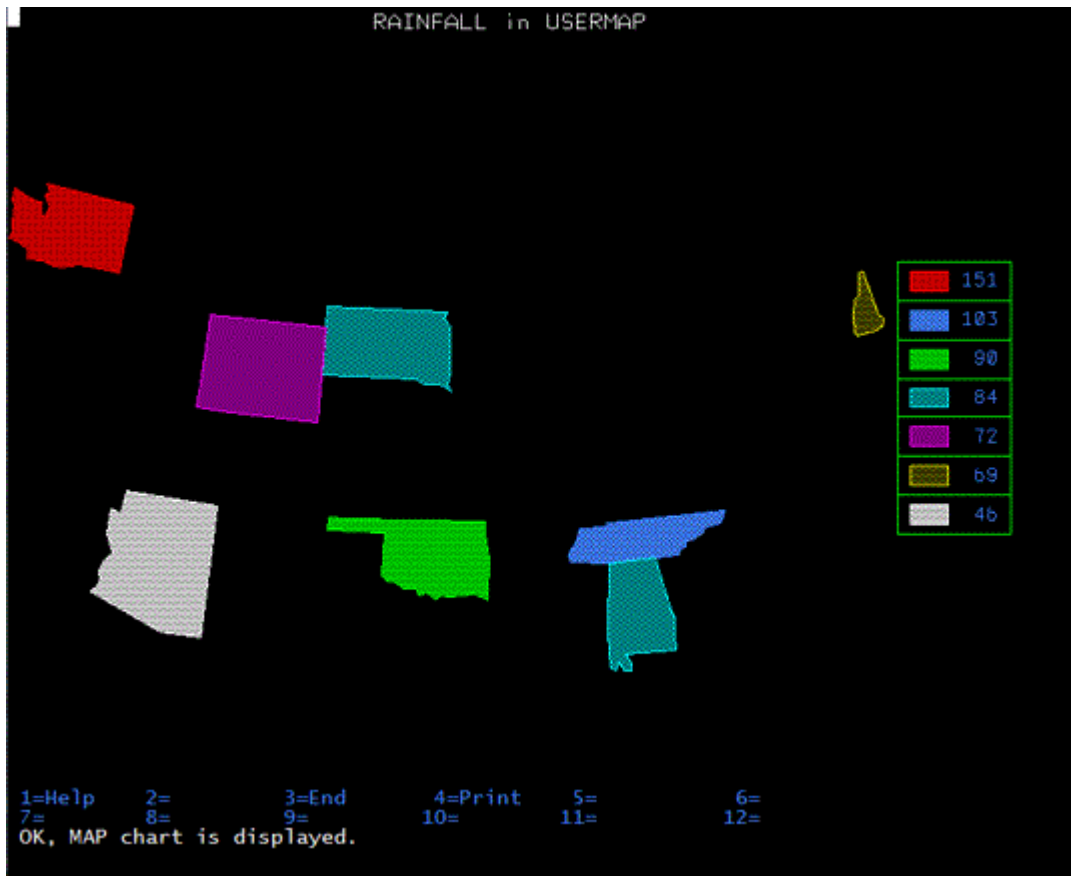


Figure 196. Map of Rainfall in the eight states of the Unites States

The map generated is color-coded and includes a key for associating colors to rainfall amount. In this example you can see that Washington state receives the highest amount of rainfall of all the states

8. Press the End function (F3) key to return to the **Parameter Selection** panel.
9. Press the End function key again to return to the QMF Analytics for TSO Home panel.
10. Press the End function key again to return to the query results.
11. Optional: If you have configured print capabilities in QMF, you can print the map.

## Charting query results as a mixed chart

You can chart query results as a mixed chart by using QMF Analytics for TSO.

### About this task

Mixed charts combine the elements of plot charts and histograms on one screen, making it easier for users to interpret two or three different measures.

In this example you will run a query on climate data and create a chart showing the minimum and maximum temperature data from the state of California.

For an example of how to chart query results as a mixed chart:

### Procedure

1. Start QMF for TSO.
 

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.
2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_USA** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

REPORT	Q.CLIMATE_USA		LINE 1	POS 1	79	
MONTH	STATE	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE	
1	AK	9	25	1	1	
1	AL	23	61	11	34	
1	AR	19	50	12	1	
1	AZ	12	68	7	80	
1	CA	19	72	18	41	
1	CO	-13	50	1	51	
1	CT	18	39	3	1	
1	DE	25	37	3	11	
1	FL	34	70	12	83	
1	GA	25	55	9	45	
1	HI	63	82	7	133	
1	IA	-6	27	4	1	
1	ID	12	50	4	29	
1	IL	9	36	2	10	
1	IN	9	32	4	1	
1	KS	9	54	3	50	
1	KY	18	36	6	1	
1	LA	27	63	10	51	
1	MA	12	39	5	1	
1	MD	19	39	9	1	
1	ME	5	28	4	1	
1	MI	9	34	3	1	
1	MN	-2	19	3	1	
1	MO	9	37	10	1	
1	MS	28	57	6	44	
1	MT	1	39	5	1	
1	NC	14	50	8	23	
1	ND	-6	23	3	1	
1	NE	3	41	3	10	
1	NH	10	32	1	1	
1	NJ	21	41	2	11	
1	NM	5	54	4	48	
1	NV	10	57	2	58	

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_USA is displayed.  
COMMAND ==>      SCROLL ==> PAGE

Figure 197. Query Results

To scroll the contents of the query result from left to right or vice versa, select the Left function key (F10) or the Right function key (F11).

3. Press the Query function (F6).
4. Tab to the data entry area of the SQL QUERY screen and enter the following SELECT statement:

```
SELECT * FROM Q.CLIMATE_USA
WHERE STATEID = 'CA'
```

Your screen will look like the following:

```
SQL QUERY                                MODIFIED LINE 1
SELECT * FROM Q.CLIMATE_USA
WHERE STATEID = 'CA'

*** END ***

1=Help      2=Run      3=End      4=Print    5=Chart    6=Draw
7=Backward  8=Forward  9=Form     10=Insert  11=Delete  12=Report
OK, QUERY is displayed.
COMMAND ==>                                SCROLL ==> PAGE
```

*Figure 198. SELECT statement for Q.CLIMATE\_USA*

5. Press the Run function (F2) to run the SELECT statement.

MONTH	STATEID	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
1	CA	19	72	18	41
2	CA	27	72	8	94
3	CA	16	82	6	122
4	CA	19	88	12	115
5	CA	30	90	8	175
6	CA	41	111	2	244
7	CA	48	120	4	279
8	CA	41	111	2	265
9	CA	37	111	1	243
10	CA	37	93	10	138
11	CA	25	77	11	91
12	CA	19	73	21	18

\*\*\* END \*\*\*

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
 7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
 OK, this is the REPORT from your RUN command.  
 COMMAND ==>      SCROLL ==> PAGE

Figure 199. Results of running SELECT statement on Q.CLIMATE\_USA

- Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
- Tab to **Mixed** in the Charts section of the QMF Analytics for TSO Home Panel and press Enter.

The **Parameter Selection** panel is displayed.

- Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Specify the column to form the vertical coordinates (y-axis)	Set two y-axis fields: <ul style="list-style-type: none"> <li>TEMPMAX</li> <li>TEMPMIN</li> </ul>
Specify the column to form the horizontal coordinates (x-axis)	MONTH



<i>Table 19. Parameter Settings for Mixed chart (continued)</i>	
<b>Parameter</b>	<b>Setting</b>
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

9. Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

10. Press the Run function key (F2) to generate the chart.

QMF Analytics for TSO generates a chart displaying the measures for maximum temperature and minimum temperature by month on a single chart.

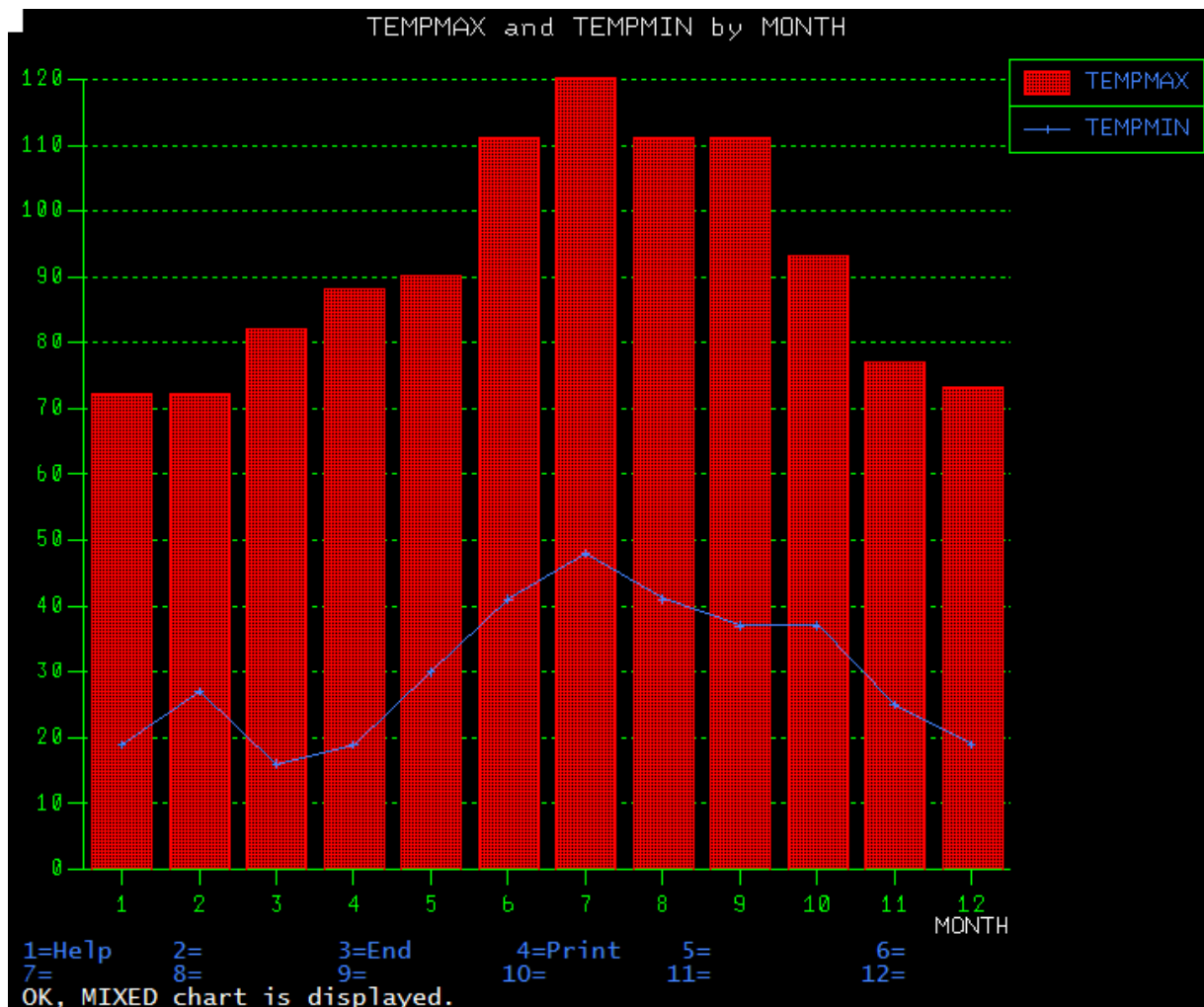


Figure 200. Mixed chart of query results

The mixed chart shows two measures, one for minimum temperature (as denoted by the plotted blue line) and one for maximum temperature (as denoted by the vertical bars).

11. Optional: If you have configured print capabilities in QMF, you can print the chart.
12. Press the End function key (F3) to return to the Parameter Selection panel.
13. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Charting query results as a pie chart

You can chart your query results as a pie chart by using QMF Analytics for TSO.

### About this task

Pie charts depict the size relationship between the parts to the whole.

For example, staying with the foul weather gear sales example, you might want to use a pie chart to illustrate which months are the wettest and use the chart to explain why you want to increase production of foul-weather gear during those months.

In this example you will query climate data for the USA and create a chart using data from the RAINFALL and MONTH columns of the query result.

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

- From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_USA** on the command line and press Enter. QMF for TSO displays the query result as a multi-columned report.

```

REPORT          Q.CLIMATE_USA  LINE 1    POS 1    79

  MONTH STATE  TEMPMIN  TEMPMAX  RAINFALL  SUNSHINE
-----
   1  AK      9         25         1          1
   1  AL     23         61        11         34
   1  AR     19         50        12          1
   1  AZ     12         68         7         80
   1  CA     19         72        18         41
   1  CO    -13         50         1         51
   1  CT     18         39         3          1
   1  DE     25         37         3         11
   1  FL     34         70        12         83
   1  GA     25         55         9         45
   1  HI     63         82         7        133
   1  IA     -6         27         4          1
   1  ID     12         50         4         29
   1  IL      9         36         2         10
   1  IN      9         32         4          1
   1  KS      9         54         3         50
   1  KY     18         36         6          1
   1  LA     27         63        10         51
   1  MA     12         39         5          1
   1  MD     19         39         9          1
   1  ME      5         28         4          1
   1  MI      9         34         3          1
   1  MN     -2         19         3          1
   1  MO      9         37        10          1
   1  MS     28         57         6         44
   1  MT      1         39         5          1
   1  NC     14         50         8         23
   1  ND     -6         23         3          1
   1  NE      3         41         3         10
   1  NH     10         32         1          1
   1  NJ     21         41         2         11
   1  NM      5         54         4         48
   1  NV     10         57         2         58

1=Help      2=          3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward  9=Form    10=Left    11=Right   12=
OK, Q.CLIMATE_USA is displayed.
COMMAND ==>
SCROLL ==> PAGE

```

Figure 201. Query Results

- Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
- Tab to **Pie** in the Charts section of the QMF Analytics for TSO Home Panel and press Enter.

The **Parameter Selection** panel is displayed.

- Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Table 20. Parameter Settings for Pie chart	
Parameter	Setting
Specify the column to form the accumulated values for each slice (Y)	RAINFALL

<i>Table 20. Parameter Settings for Pie chart (continued)</i>	
<b>Parameter</b>	<b>Setting</b>
Specify a column to determine the sector of each slice (X)	MONTH
Explode Slices?	None
3-D Pie Chart?	No
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2) to generate the pie chart.

QMF Analytics for TSO creates the chart.

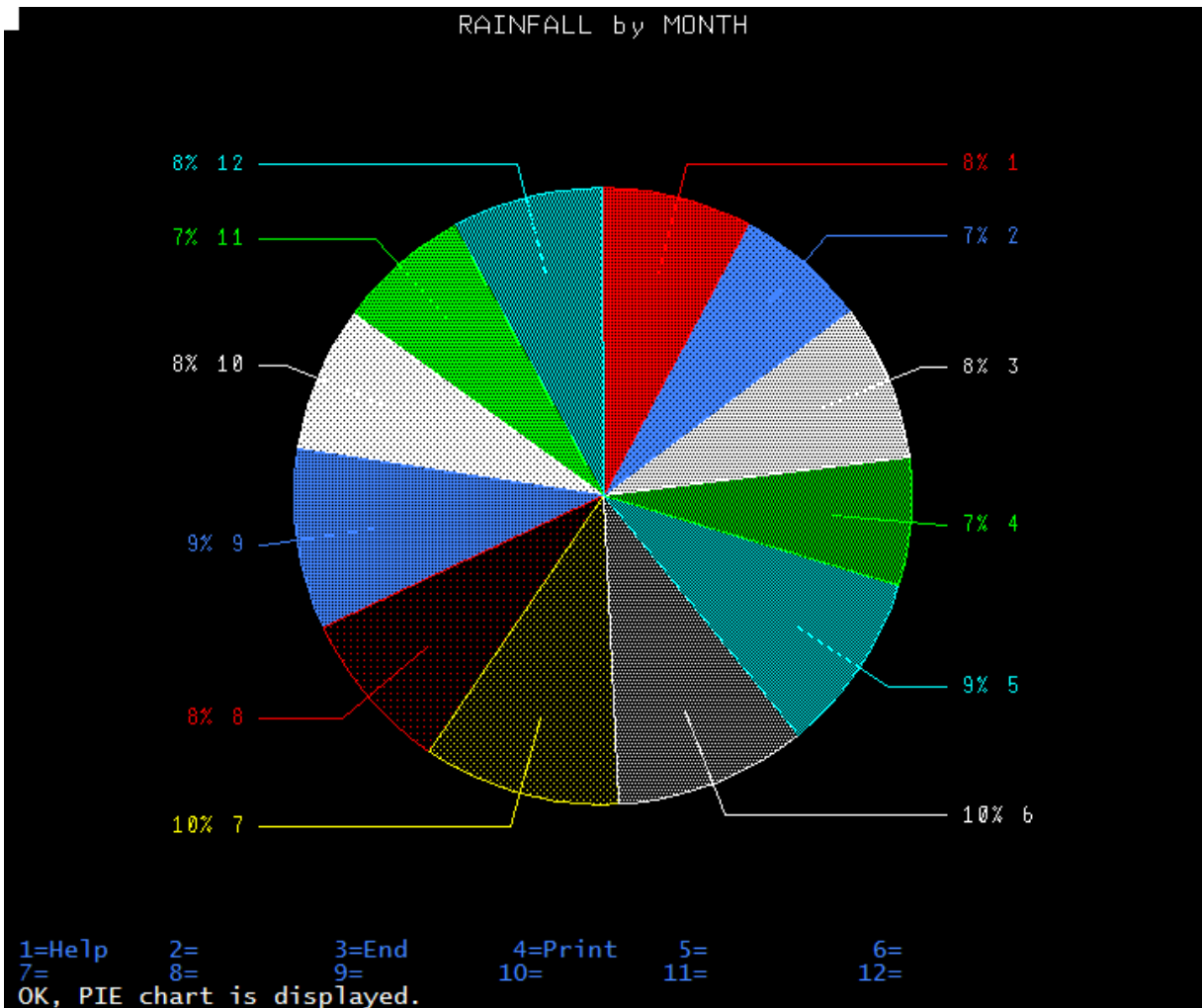


Figure 202. Rainfall by Year Pie Chart

The chart shows that June and July (month 6 and 7) are the rainiest months. To make this even more clear to your audience, you can accentuate the "slice" associated with the rainiest month.

8. Press the End function key to return to the **Parameter Selection** panel.
9. To highlight which month was the rainiest month, set **Explode Slices?** to Largest by tabbing to Largest and pressing Enter.
10. To apply a 3-dimensional effect to the chart, set **3-D Pie Chart?** to Yes by tabbing to Yes and pressing Enter.
11. Press the Run function key (F2) to generate an updated version of the pie chart.

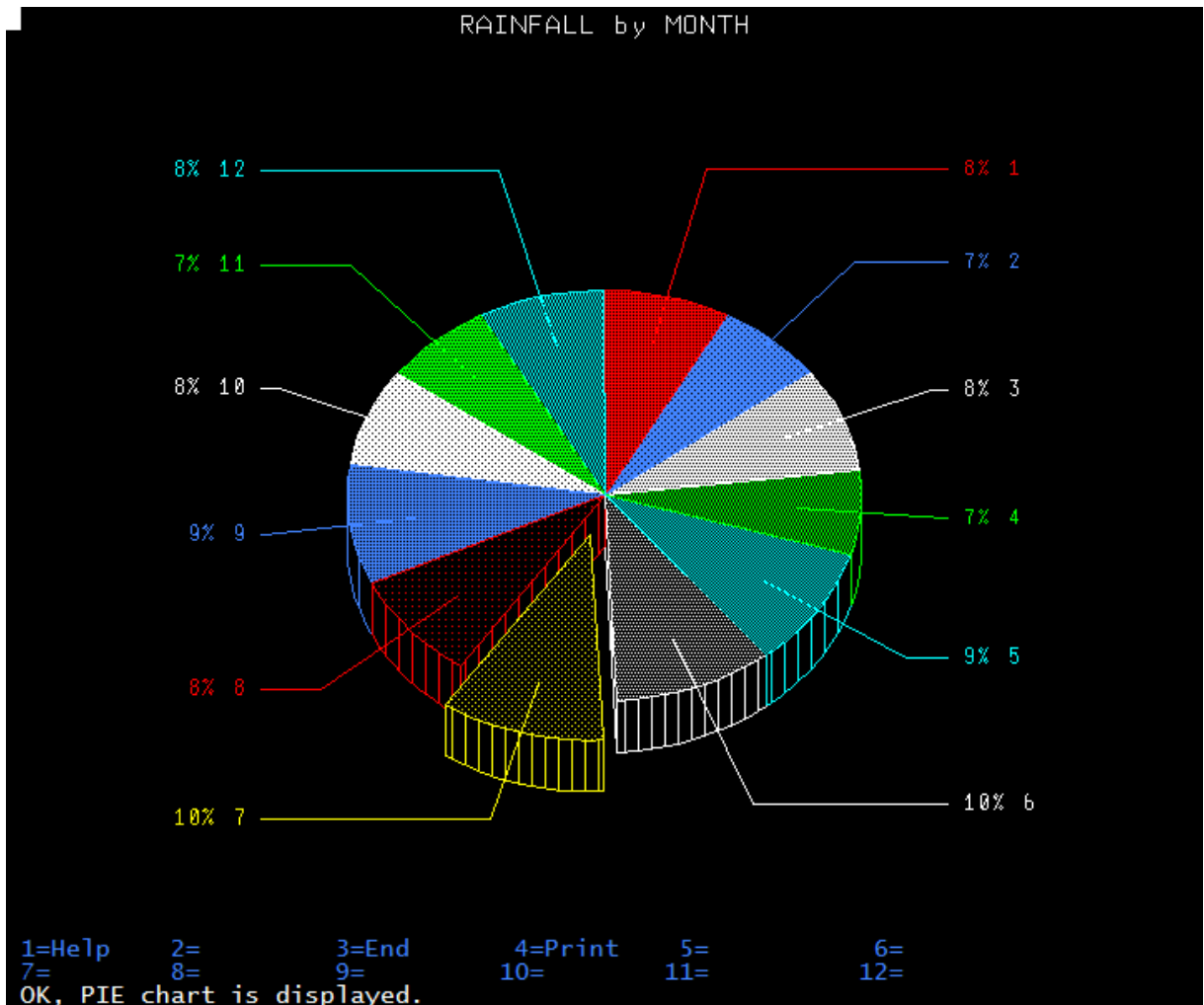


Figure 203. Rainfall by Year Pie Chart with Exploded Slice and 3-Dimensional Format

QMF Analytics for TSO creates the new pie chart in 3-dimensional format, highlighting the rainiest month 7 .

12. Optional: If you have configured print capabilities in QMF, you can print the chart.
13. Press the End function key (F3) to return to the Parameter Selection panel.
14. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Charting query results as a plot chart

You can chart your query results as a plot chart by using QMF Analytics for TSO.

### About this task

Plot charts are often used as an alternative to bar charts. In a plot chart, the dots depict the quantitative values (e.g. counts) associated with categorical variables. You might choose to use a plot chart when there are very few data values. Each value is recorded as a dot, so that the frequencies for each value can easily be counted.

In this example you will plot the minimum, maximum, and average salary by years of service.

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

- From the **QMF Home Panel**, enter **DISPLAY Q.STAFF** on the command line and press Enter. QMF for TSO displays the query result as a multi-columned report.

REPORT	Q.STAFF				LINE 1	POS 1	79
ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	
10	SANDERS	20	MGR	7	18357.50	-	
20	PERNAL	20	SALES	8	18171.25	612.45	
30	MARENGHI	38	MGR	5	17506.75	-	
40	O'BRIEN	38	SALES	6	18006.00	846.55	
50	HANES	15	MGR	10	20659.80	-	
60	QUIGLEY	38	SALES	-	16808.30	650.25	
70	ROTHMAN	15	SALES	7	16502.83	1152.00	
80	JAMES	20	CLERK	-	13504.60	128.20	
90	KOONITZ	42	SALES	6	18001.75	1386.70	
100	PLOTZ	42	MGR	7	18352.80	-	
110	NGAN	15	CLERK	5	12508.20	206.60	
120	NAUGHTON	38	CLERK	-	12954.75	180.00	
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60	
140	FRAYE	51	MGR	6	21150.00	-	
150	WILLIAMS	51	SALES	6	19456.50	637.65	
160	MOLINARE	10	MGR	7	22959.20	-	
170	KERMISCH	15	CLERK	4	12258.50	110.10	
180	ABRAHAMS	38	CLERK	3	12009.75	236.50	
190	SNEIDER	20	CLERK	8	14252.75	126.50	
200	SCOUTTEN	42	CLERK	-	11508.60	84.20	
210	LU	10	MGR	10	20010.00	-	
220	SMITH	51	SALES	7	17654.50	992.80	
230	LUNDQUIST	51	CLERK	3	13369.80	189.65	
240	DANIELS	10	MGR	5	19260.25	-	
250	WHEELER	51	CLERK	6	14460.00	513.30	
260	JONES	10	MGR	12	21234.00	-	
270	LEA	66	MGR	9	18555.50	-	
280	WILSON	66	SALES	9	18674.50	811.50	
290	QUILL	84	MGR	10	19818.00	-	
300	DAVIS	84	SALES	5	15454.50	806.10	
310	GRAHAM	66	SALES	13	21000.00	200.30	
320	GONZALES	66	SALES	4	16858.20	844.00	
330	BURKE	66	CLERK	1	10988.00	55.50	

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.STAFF is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 204. Query Results

- Press the Query function (F6).
- Tab to the data entry area of the SQL QUERY screen and enter the following SELECT statement:

```
SELECT
  MIN(SALARY) AS MIN_SALARY,
  AVG(SALARY) AS AVERAGE_SALARY,
  MAX(SALARY) AS MAX_SALARY,
  YEARS
FROM Q.STAFF
WHERE YEARS > 0
GROUP BY YEARS
```

Your screen will look like the following:

```

SQL QUERY                                LINE    1
SELECT
  MIN(SALARY) AS MIN_SALARY,
  AVG(SALARY) AS AVERAGE_SALARY,
  MAX(SALARY) AS MAX_SALARY,
  YEARS
FROM Q.STAFF
  WHERE YEARS > 0
GROUP BY YEARS

*** END ***

1=Help      2=Run      3=End      4=Print    5=Chart    6=Draw
7=Backward  8=Forward  9=Form     10=Insert  11=Delete  12=Report
OK, QUERY is displayed.
COMMAND ==>                                SCROLL ==> PAGE

```

*Figure 205. SELECT statement for Q.STAFF*

5. Press the Run function (F2) to run the SELECT statement.



MIN SALARY	AVERAGE SALARY	MAX SALARY	YEARS
10988.00	10988.0000000000	10988.00	1
12009.75	12689.7750000000	13369.80	3
12258.50	14558.3500000000	16858.20	4
12508.20	15552.0400000000	19260.25	5
10505.90	16930.0250000000	21150.00	6
16502.83	18611.8050000000	22959.20	7
14252.75	16212.0000000000	18171.25	8
18555.50	18615.0000000000	18674.50	9
19818.00	20162.6000000000	20659.80	10
21234.00	21234.0000000000	21234.00	12
21000.00	21000.0000000000	21000.00	13

\*\*\* END \*\*\*

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
 7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
 OK, this is the REPORT from your RUN command.  
 COMMAND ==>      SCROLL ==> PAGE

Figure 206. Results of running SELECT statement on Q.STAFF query results

6. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO..
7. Tab to **Plot** in the Charts section of the QMF Analytics for TSO Home Panel and press Enter.

The **Parameter Selection** panel is displayed.

8. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Specify the columns to form the vertical coordinates (y-axis)	MIN_SALARY AVERAGE_SALARY MAX_SALARY
Specify the column to form the horizontal coordinates (x-axis)	YEARS

Table 21. Parameter Settings for Plot chart (continued)	
Parameter	Setting
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2) to generate the chart.

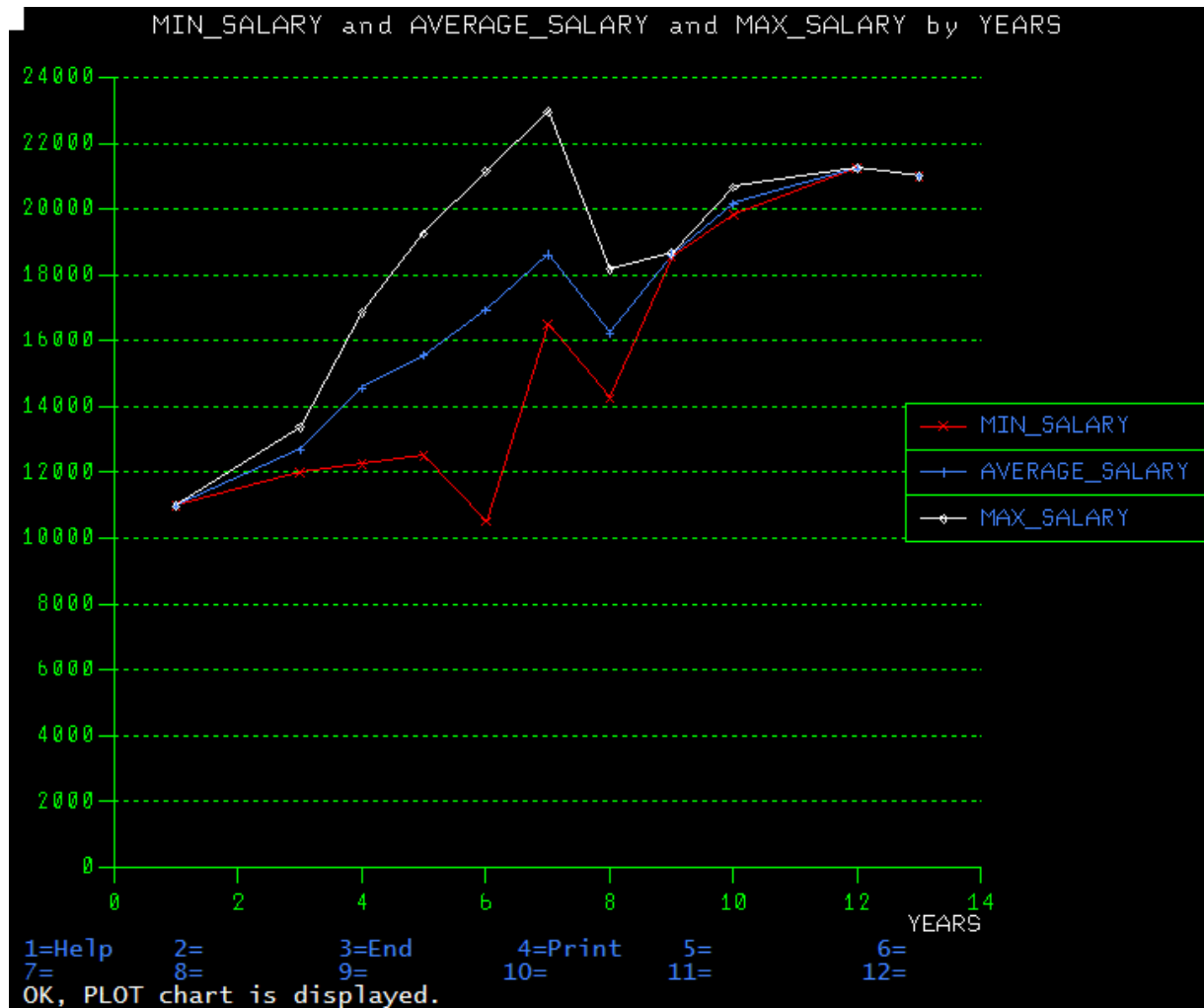


Figure 207. Plot chart of Minimum, Average, and Maximum Salary by Years of service

- Optional: If you have configured print capabilities in QMF, you can print the chart.
- Press the End function key (F3) to return to the Parameter Selection panel.

13. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Charting query results as a tower chart

You can chart your query results as a tower chart by using QMF Analytics for TSO.

### About this task

Tower charts show the frequency of data items in successive numerical intervals of equal size, or to show a comparison of categorical values in a query result.

QMF Analytics for TSO will plot data distributions or simple categorical values depending on the type of data in column to be used as the x-axis on the chart. QMF Analytics for TSO examines the X variable data type, data values, and number of rows before deciding whether to chart it as a distribution or as a set of categorical values.

In this example you will chart salary by department number. You might want to chart this type of data to understand which departments in a company have the highest cost.

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.STAFF** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

REPORT	Q.STAFF		LINE 1	POS 1	79	
ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	SANDERS	20	MGR	7	18357.50	-
20	PERNAL	20	SALES	8	18171.25	612.45
30	MARENGHI	38	MGR	5	17506.75	-
40	O'BRIEN	38	SALES	6	18006.00	846.55
50	HANES	15	MGR	10	20659.80	-
60	QUIGLEY	38	SALES	-	16808.30	650.25
70	ROTHMAN	15	SALES	7	16502.83	1152.00
80	JAMES	20	CLERK	-	13504.60	128.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
100	PLOTZ	42	MGR	7	18352.80	-
110	NGAN	15	CLERK	5	12508.20	206.60
120	NAUGHTON	38	CLERK	-	12954.75	180.00
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
140	FRAYE	51	MGR	6	21150.00	-
150	WILLIAMS	51	SALES	6	19456.50	637.65
160	MOLINARE	10	MGR	7	22959.20	-
170	KERMISCH	15	CLERK	4	12258.50	110.10
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
190	SNEIDER	20	CLERK	8	14252.75	126.50
200	SCOUTTEN	42	CLERK	-	11508.60	84.20
210	LU	10	MGR	10	20010.00	-
220	SMITH	51	SALES	7	17654.50	992.80
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
240	DANIELS	10	MGR	5	19260.25	-
250	WHEELER	51	CLERK	6	14460.00	513.30
260	JONES	10	MGR	12	21234.00	-
270	LEA	66	MGR	9	18555.50	-
280	WILSON	66	SALES	9	18674.50	811.50
290	QUILL	84	MGR	10	19818.00	-
300	DAVIS	84	SALES	5	15454.50	806.10
310	GRAHAM	66	SALES	13	21000.00	200.30
320	GONZALES	66	SALES	4	16858.20	844.00
330	BURKE	66	CLERK	1	10988.00	55.50

1=Help      2=  
7=Backward      8=Forward  
OK, Q.STAFF is displayed.  
COMMAND ==>>

3=End      4=Print      5=Chart      6=Query  
9=Form      10=Left      11=Right      12=  
SCROLL ==>> PAGE

Figure 208. Query Results

- Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
- Tab to **Tower** in the Charts section of the QMF Analytics for TSO Home Panel and press Enter.

The **Parameter Selection** panel is displayed.

- Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Specify the columns to form the vertical coordinates (y-axis)	SALARY
Specify the column to form the horizontal coordinates (x-axis)	DEPT
Tabulation?	No

Table 22. Parameter Settings for Tower chart (continued)	
Parameter	Setting
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) to save the specification to the QMF database as an ANALYTIC object.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2) to generate the chart.

QMF Analytics for TSO generates the following chart:

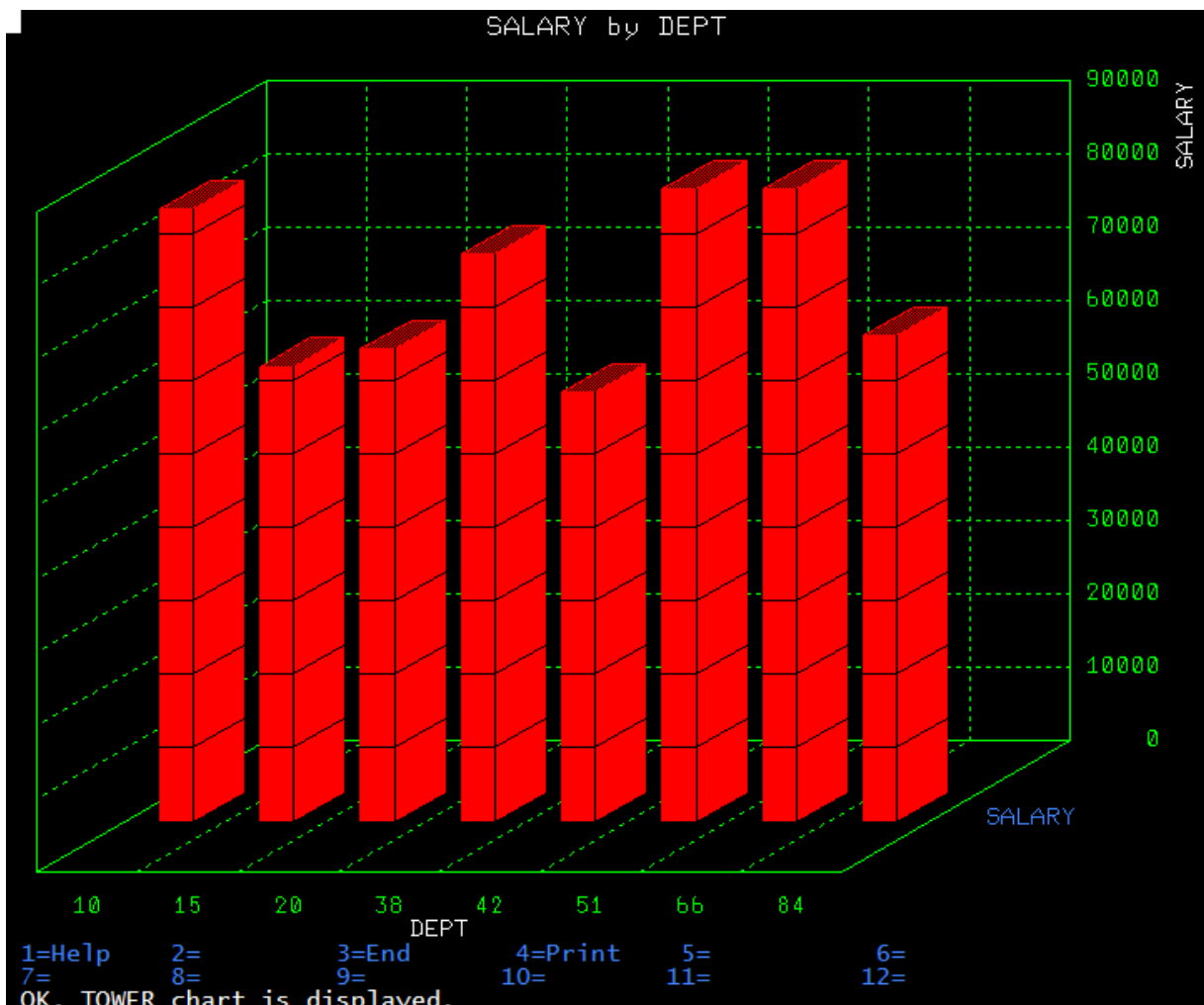


Figure 209. Salary by Department Tower Chart

The chart shows that the accumulated salary by department. It shows that department 10 is the costliest in terms of employee salary and department 66 as the least costly in terms of employee salary.

8. Optional: If you have configured print capabilities in QMF, you can print the chart.
9. Press the End function key (F3) to return to the Parameter Selection panel.
10. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Analyzing query results with QMF Analytics for TSO Statistics

Use the information in this section to learn how to use QMF Analytics for TSO to generate statistics from your query results.

### Before you begin

To use an example topic as a hands-on exercise, you must have QMF for TSO installed and QMF Analytics for TSO configured for use. The examples make use of sample data tables that are included with QMF for TSO.

### About this task

Table 23 on page 186 lists the types of statistical analyses supported by QMF Analytics for TSO. The table describes the purpose of type of analysis, when it might be appropriate to use the analysis, and a link to instructions on how to run the analysis.

The procedure in this topic demonstrates:

- How to run a query from QMF for TSO
- How to start QMF Analytics for TSO and select a statistical analysis type
- How to create the specification for your statistical analysis by using the Parameter Selection panel
- How to run the specification to generate the statistics

QMF Analytics for TSO has its own online help system and context-sensitive help for panels and fields. If you want to learn more about a field or panel as you go through the examples, use the Help and More Help functions.

Statistic	When would I use this type of statistic?	For an example, see
Basic	<p>This type of analysis provides sample summarization based on an average, the dispersion, and the range.</p> <p>Suppose that you are a marketing analyst at a company that manufactures small kitchen appliances. The director of marketing suggests that each of the appliances should include a one-year guarantee. Before implementing such a policy, the company's CEO requests a study to determine how many appliances break in a year. This becomes the business question.</p>	<p><a href="#">“Running a basic statistical analysis on query results” on page 188</a></p>

Table 23. Types of statistical analyses you can run on your query results with QMF Analytics for TSO (continued)

Statistic	When would I use this type of statistic?	For an example, see
Curve Fitting - Bivariate	<p>This type of analysis measures the curve derived from two variables, one dependent variable (sometimes referred to as the response variable) and one independent variable (sometimes referred to as the predictor variable).</p> <p>Suppose that you are a research analyst for a well drilling company. The board of executives wants to know whether the concentration of chemicals (used in the drilling process) in well water samples dissipates over time.</p>	<p><a href="#">“Analyzing query results using bivariate curve fitting analysis” on page 194</a></p>
Curve Fitting - Univariate	<p>This type of analysis shows the distribution (frequency of occurrence) of a single variable from the distribution (frequency of occurrence) of several variables.</p> <p>Suppose you work for a golf club manufacturer, and you would like to look at your sales data for the last ten years to forecast the demand for your product in the next few years.</p> <p>The sales of golf clubs fall off in some months or years, even though the sales of golf clubs grow steadily overall. Using QMF for TSO you would run a query to show the show the historical demand for your golf clubs. Your query result would have only one variable (Demand).</p>	<p><a href="#">“Analyzing query results using univariate curve fitting analysis” on page 199</a></p>
Linear Trend	<p>Linear trend analysis shows how the sales and price of a product have been related in the past and uses this relationship to predict how a specific price affects future sales.</p> <p>Suppose you work for automobile dealership, you might use Linear Regression to derive the relationship between advertising and sales, and then use this relationship to determine how a specific advertising budget would affect sales.</p>	<p><a href="#">“Running a linear trend analysis of your query results” on page 212</a></p>

Table 23. Types of statistical analyses you can run on your query results with QMF Analytics for TSO (continued)

Statistic	When would I use this type of statistic?	For an example, see
Discounted Cash Flow	<p>Discounted Cash Flow statistics compare costs and expenditures on new projects or indicate whether the internal rate of return on a project or investment meets financial guidelines established by your company.</p> <p>Suppose you want to launch a new product. Through market research you can assess the projected revenue and costs, and calculate the projected cash flow.</p> <p>Using QMF Analytics for TSO, you can run a discounted cash flow analysis to view graphs and data tables of the results of Discounted Cash Flow.</p>	<p><a href="#">“Running a discounted cash flow analysis of your query results” on page 204</a></p>

## Running a basic statistical analysis on query results

You can generate basic statistics from your query results by using QMF Analytics for TSO.

### About this task

Basic Statistics show sample size, minimum value, maximum value, range, median, mode, mean, standard error, standard deviation, skewness, kurtosis, quartiles, and sextiles for your query result data.

For an example of how to analyze query results using Basic Statistics:

### Procedure

1. Start QMF for TSO.  
Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.
2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_10YR** on the command line and press Enter.  
QMF for TSO displays the query result as a multi-columned report.



REPORT	Q.CLIMATE_10YR	LINE 1	POS 1	79	
YEAR	MONTH	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
2001	1	9	70	3	234
2001	2	18	72	7	205
2001	3	16	77	12	180
2001	4	32	91	3	230
2001	5	32	95	4	234
2001	6	41	115	3	230
2001	7	43	111	1	227
2001	8	39	115	3	238
2001	9	32	93	6	226
2001	10	27	88	6	221
2001	11	14	79	12	183
2001	12	19	73	9	204
2002	1	27	77	11	185
2002	2	25	75	45	25
2002	3	25	88	17	161
2002	4	32	90	3	226
2002	5	32	100	2	225
2002	6	43	108	4	241
2002	7	46	111	4	228
2002	8	45	113	2	238
2002	9	39	102	9	197
2002	10	32	90	3	227
2002	11	27	79	5	221
2002	12	23	66	8	219
2003	1	14	64	12	195
2003	2	16	75	8	190
2003	3	25	73	10	187
2003	4	27	97	1	233
2003	5	36	102	7	220
2003	6	45	115	3	242
2003	7	37	115	1	237
2003	8	41	117	1	240
2003	9	39	104	2	235

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward    8=Forward    9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_10YR is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 210. Query Results

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Basic** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Select all numeric columns	All  QMF Analytics for TSO lists all columns with numeric data in the <b>Which column(s) to be analyzed?</b> field.  <b>Note:</b> You can also specify which columns from the query result to analyze by entering the column names manually or by using the List function key (F4).

Parameter	Setting
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

6. Optional: Press the Save function key (F5) to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

7. Press the Run function key (F2) to analyze the data.

QMF Analytics for TSO returns a list of the variables (the numeric columns) in your analysis and for each one shows the sample size, minimum value, maximum value, and mean. The following columns are standard measures in basic statistics: Size, Minimum, Maximum, and Mean.

In [Figure 211 on page 190](#), the Size column is the amount of data samples collected during the observation period, which is the number of rows in the table. In this example, the values in the Size column is equivalent to the number of months if the data contains one sample per month.

Basic				
Name	Size	Minimum	Maximum	Mean
YEAR	120	2001.000	2010.000	2005.500
MONTH	120	1.000	12.000	6.500
TEMPMIN	120	9.000	48.000	29.350
TEMPMAX	120	64.000	120.000	91.550
RAINFALL	120	1.000	45.000	6.917
SUNSHINE	120	25.000	242.000	211.900

1=Help	2=	3=End	4=Print	5=Summary	6=
7=Back	8=Forward	9=	10=Statistics	11=Distribution	12=Chi-Square

Figure 211. Summary Data for Basic Statistical Analysis of Query Results

The largest observation for each variable is listed in the column titled **Maximum** and the smallest observation for each variable is listed in the column titled **Minimum**.

From the Summary page you can tab to a specific variable and run statistical analyses on the variable selected.

The following table lists and describes the functions available from the Summary page.

<i>Table 25. Statistical analysis functions available from the Summary page. Statistical analysis functions available from the Summary page</i>		
<b>Function</b>	<b>Result</b>	<b>Usage notes</b>
Statistics (F10)	Displays additional statistics for the column selected. See <a href="#">Basic statistics for Rainfall</a> for an example.	You can use the Forward and Backward function keys to scroll through basic statistics for each column.
Distribution (F11)	Displays a distribution graph of the column data being analyzed. See <a href="#">Distribution statistics for Rainfall</a> for an example.	You can use the Forward and Backward function keys to scroll through distribution statistics for each column.
Chi-square (F12)	<p>Displays a Chi-square graph of the column data being analyzed. See <a href="#">Chi-square</a> for an example.</p> <p>Chi-square is a value used in statistical analyses as a basis for rejecting or accepting the null hypothesis.</p> <p>By running a Chi-square Test, you can see whether a sample is normally distributed.</p>	You can use the Forward and Backward function keys to scroll through chi-square statistics for each column.

8. Tab to RAINFALL in the **Name** column and press the Statistics function key (F10).

Basic				
RAINFALL	Size	120		
	Minimum	1.000		
	Maximum	45.000		
	Range	44.000		
	Median	5.000	Quartile 1	3.000
	Mode	3.000	Quartile 2	5.000
	Mean	6.917	Quartile 3	9.000
	Standard Error	0.611	Sextile 1	2.000
	Standard Deviation	6.691	Sextile 2	3.000
	Skewness	2.583	Sextile 3	5.000
	Kurtosis	12.186	Sextile 4	8.000
			Sextile 5	10.000

1=Help 2= 3=End 4=Print 5=Summary 6=  
7=Back 8=Forward 9= 10=Statistics 11=Distribution 12=Chi-Square  
Press a function key for the appropriate display.

Figure 212. Additional Statistics - Rainfall

In addition to the standard measures for basic statistics, the Statistics function shows the following:

**Mode**

The value of a variable that occurs most often in a distribution.

**Median.**

A measure of central tendency; the midpoint of a distribution.

**Range**

The difference between the maximum and minimum values.

**Standard deviation**

A measure of dispersion. This measures the spread of your data relative to its arithmetic mean. The standard deviation is measured in the same units as the data.

**Standard error**

The standard deviation of the sampling distribution of a variable.

**Skewness**

A measure of the asymmetry of the values in your distribution. If the distribution has a longer tail to the high end of the scale, your data is positively skewed. If the distribution has a longer tail to the low end of the scale, your data is negatively skewed.

**Quartiles**

Three values that divide a distribution into four parts, such that a quarter of the numbers fall under the first quartile, half under the second quartile, and so on.

When speaking about Quartiles, the “numbers” represented in the analysis results have a direct relation to the size. For example, the quartile is a value which cuts 1/4 the “size” samples.

## Sextiles

Five values that divide a distribution into six parts, such that a sixth of the numbers fall under the first sextile, a third under the second sextile, half under the third sextile, and so on.

When speaking about Sextiles, the “numbers” represented in the analysis results have a direct relation to the size. For example, the sextile is a value which cuts 1/6 of the “size” samples.

9. Press the Distribution function key (F11) to display distribution statistics for the selected column.

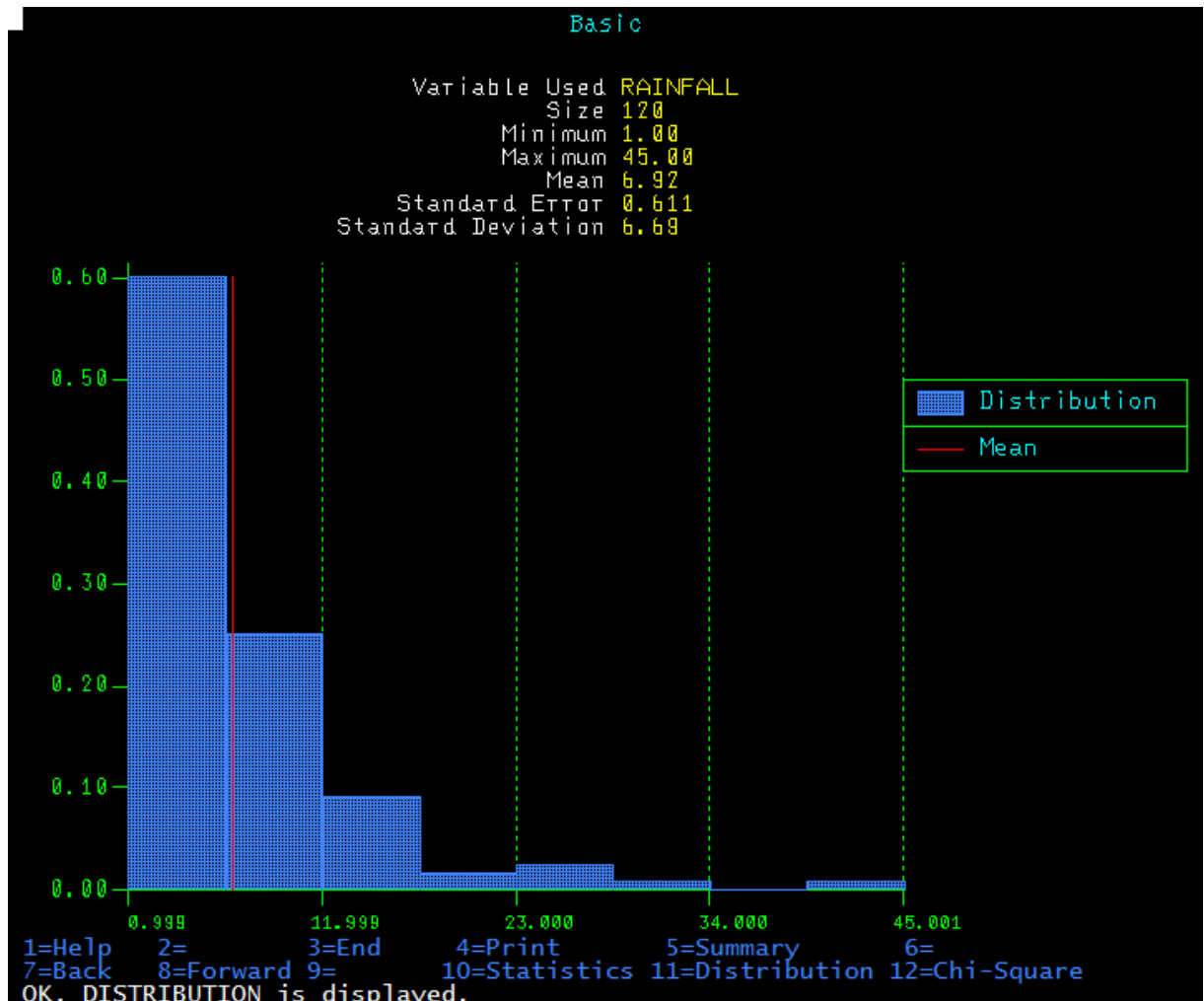


Figure 213. Distribution statistics - Rainfall

This graph shows the frequency of rainfall values from the query result during the 120 month period. It indicates that the bigger the rainfall is, the less likely it happens, and that the average rainfall is measured by 6.92 millimeter.

The X-axis is the rainfall sample value, the Y-axis is the frequency.

10. Press the Chi-Square function key (F12) to display Chi-Square statistics for the selected column.

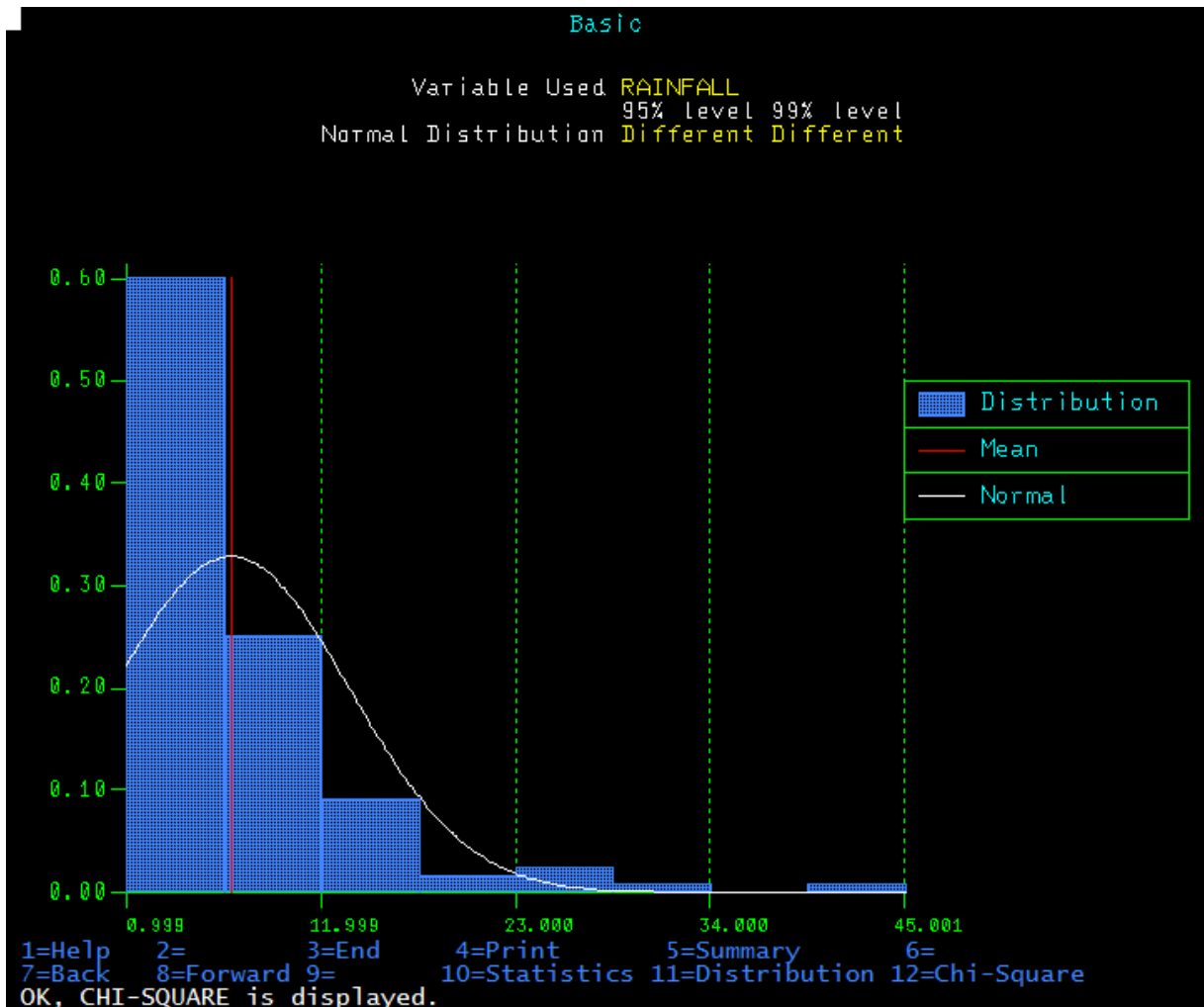


Figure 214. Chi-Square Statistics - Rainfall

Figure 214 on page 194 indicates that the suggested hypothesis, that the rainfall distribution is controlled by Normal statistics, is false.

The confidence levels 95% and 99% are the corresponding probability of the decisions made: Different and Different. That decision means that if we accept as normal the 5% error level by making a wrong guess about the rainfall distribution, then the Chi-Square criteria gives the answer NO. Therefore the distribution is different from normal.

The same holds true if we accept 1% error level.

11. Optional: If you have configured print capabilities in QMF, you can print the chart.
12. Press the End function key (F3) to return to the Parameter Selection panel.
13. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Analyzing query results using bivariate curve fitting analysis

You can run bivariate curve fitting analysis on your query results using QMF Analytics for TSO.

### About this task

Curve fitting is a statistical methodology commonly used as a quantitative forecasting technique. Many different types of curves can be used in fitting historical data for predicting future variables. The challenge is to determine *which form of curve* will best fit the available data and provide the basis for an accurate forecast of the future.

Bivariate curve fitting analysis can fit up to ten trend lines to a given time series, for two data variables. In this example you will query climate data over a 10 year period and perform an analysis on two variables (Rainfall and Sunshine).

## Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_10YR** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

REPORT	Q.CLIMATE_10YR						LINE 1	POS 1	79
	YEAR	MONTH	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE			
	2001	1	9	70	3	234			
	2001	2	18	72	7	205			
	2001	3	16	77	12	180			
	2001	4	32	91	3	230			
	2001	5	32	95	4	234			
	2001	6	41	115	3	230			
	2001	7	43	111	1	227			
	2001	8	39	115	3	238			
	2001	9	32	93	6	226			
	2001	10	27	88	6	221			
	2001	11	14	79	12	183			
	2001	12	19	73	9	204			
	2002	1	27	77	11	185			
	2002	2	25	75	45	25			
	2002	3	25	88	17	161			
	2002	4	32	90	3	226			
	2002	5	32	100	2	225			
	2002	6	43	108	4	241			
	2002	7	46	111	4	228			
	2002	8	45	113	2	238			
	2002	9	39	102	9	197			
	2002	10	32	90	3	227			
	2002	11	27	79	5	221			
	2002	12	23	66	8	219			
	2003	1	14	64	12	195			
	2003	2	16	75	8	190			
	2003	3	25	73	10	187			
	2003	4	27	97	1	233			
	2003	5	36	102	7	220			
	2003	6	45	115	3	242			
	2003	7	37	115	1	237			
	2003	8	41	117	1	240			
	2003	9	39	104	2	235			

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_10YR is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 215. Query Results

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Curve Fitting - Bivariate** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

<i>Table 26. Parameter Settings for Curve Fitting Bivariate Analysis</i>	
<b>Parameter</b>	<b>Setting</b>
Select x-column	RAINFALL
Select y-column	SUNSHINE
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

- Optional: Press the Save function key (F5) if you want to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2).

QMF Analytics for TSO applies the mathematical function for curve fitting analysis on the two variables (SUNSHINE and RAINFALL) specified and presents a list of curves that fit the series of data points and some statistical details.



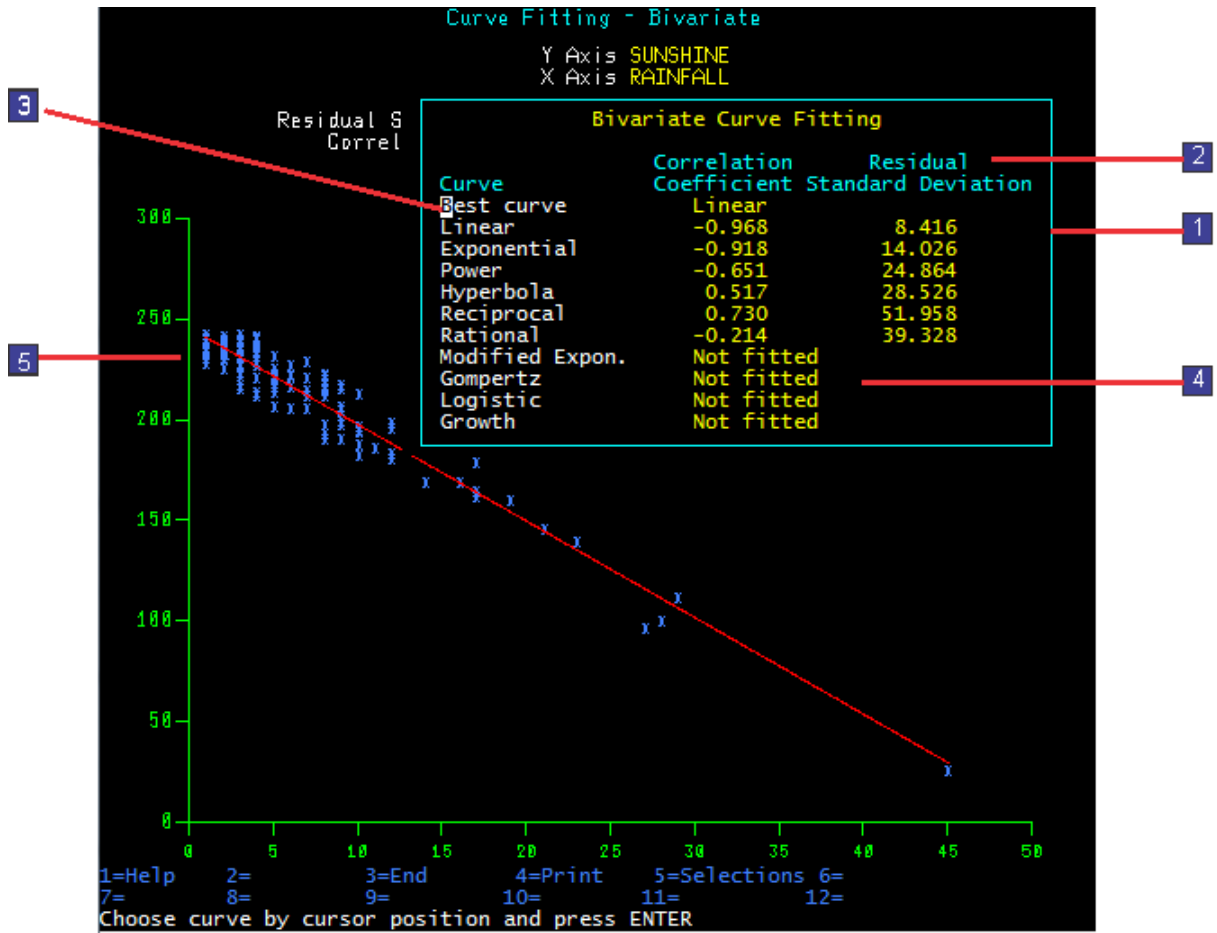


Figure 216. Initial Curve Fitting Selection Window

Callout Number	Description
1	<p>The <b>Bivariate Curve Fitting Selection</b> window that contains a "ranked" list of 10 curves that QMF Analytics for TSO has formulated from the query result columns specified.</p> <p>The items listed in the window are selectable, meaning you can tab to any of the curves listed and press Enter to see the details for the curve.</p>
2	<p>The <b>correlation coefficient</b> measures the degree of linear relationship between a pair of variables. The value of the correlation coefficient can range from -1 (when large values of one variable occur with small values of the other) to +1 (when large or small values of both variables occur together).</p> <p>The <b>residual standard deviation</b> measures the standard deviation of the residuals.</p>
3	<p>The Best curve is the curve form, as determined by the analysis, that will best fit the available data and provide the basis for the most accurate forecast of the future.</p> <p>The Best curve is always listed at the top of the list of curves in the <b>Bivariate Curve Fitting Selection</b> window. <a href="#">Figure 216 on page 197</a> indicates that for the data you are analyzing, the Linear curve form would be the best curve to use for predictive analysis.</p>

Table 27. Annotations for Curve Fitting Bivariate Analysis result screen (continued)	
Callout Number	Description
4	If the data being analyzed could not be fitted to a curve form, QMF Analytics for TSO lists the curve in the <b>Curve Fitting Selection</b> window, and categorizes it as <i>Not fitted</i> .
5	Data points (x) for the column data analyzed and trend line (—)

8. To draw one of the curves other than the best fit curve and re-display the statistical graph, position the cursor on the curve name and press Enter.

For example, tab to **Hyperbola** and press Enter.

QMF Analytics for TSO draws the curve and displays the graph.

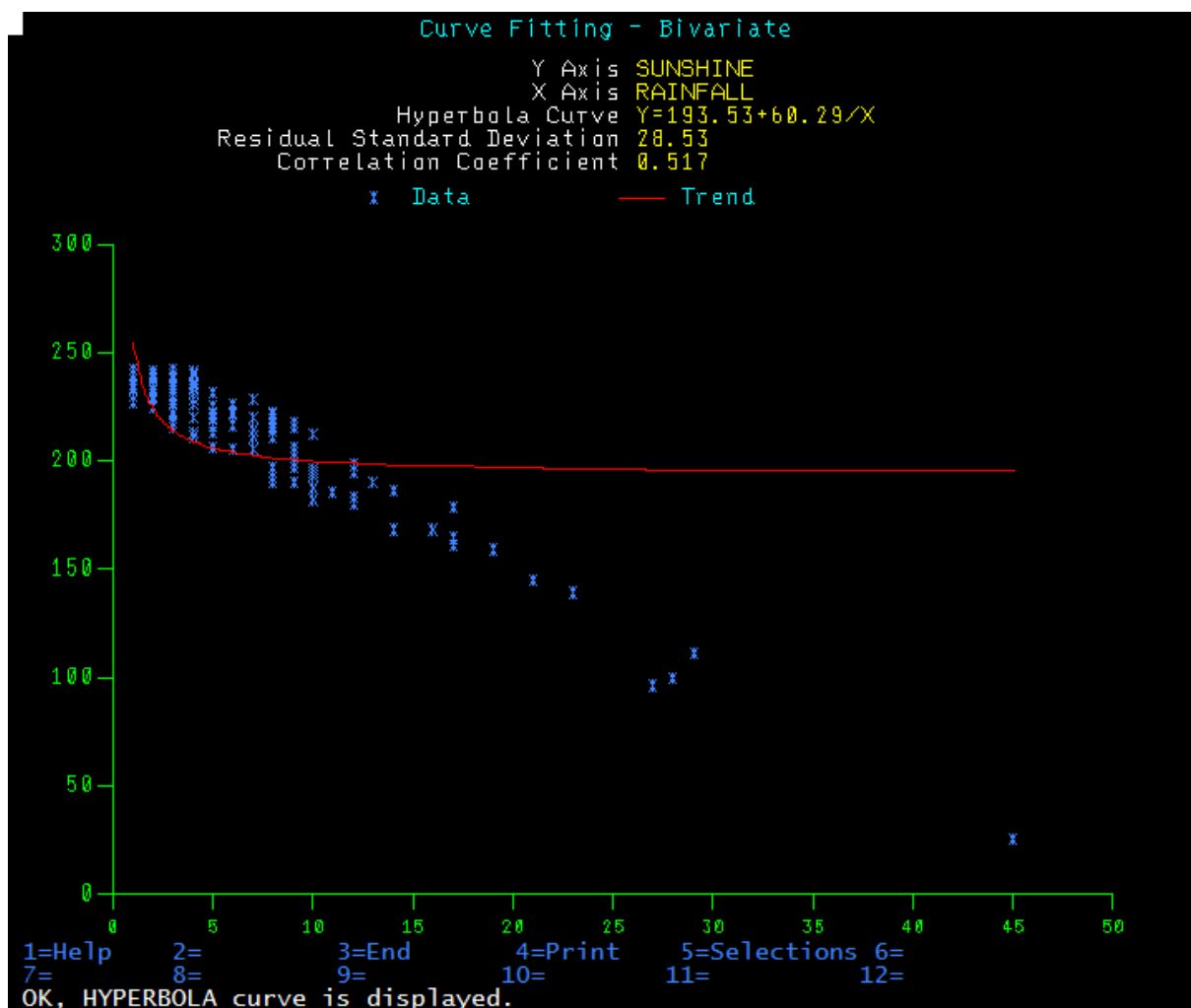


Figure 217. Hyperbola Curve for Bivariate Statistics

9. From the curve analysis page, you can:

- Go back to the Curve Fitting selection by pressing the Selections function key (F5).
- Print the analysis results by pressing the Print function key (F3).

10. Press the End function key (F3) to return to the Parameter Selection panel.

11. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

## Analyzing query results using univariate curve fitting analysis

You can run univariate curve fitting analysis on your query results using QMF Analytics for TSO.

### About this task

Univariate curve fitting takes time series data collected at fixed intervals (daily, weekly, monthly, quarterly, or yearly) and identifies observable patterns known as *trends*. Time series analysis is a mathematical way of finding observable patterns in the time series data. The four components of time that affect patterns in data are:

- Long-term trend
- seasonal variation
- cyclic variation
- irregular variation

For an example of how to analyze query results using Curve Fitting Univariate Statistics:

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_10YR** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

The following illustration shows the query result from QMF and notes the columns of interest.

REPORT	Q.CLIMATE_10YR	LINE 1	POS 1	79	
YEAR	MONTH	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
2001	1	9	70	3	234
2001	2	18	72	7	205
2001	3	16	77	12	180
2001	4	32	91	3	230
2001	5	32	95	4	234
2001	6	41	115	3	230
2001	7	43	111	1	227
2001	8	39	115	3	238
2001	9	32	93	6	226
2001	10	27	88	6	221
2001	11	14	79	12	183
2001	12	19	73	9	204
2002	1	27	77	11	185
2002	2	25	75	45	25
2002	3	25	88	17	161
2002	4	32	90	3	226
2002	5	32	100	2	225
2002	6	43	108	4	241
2002	7	46	111	4	228
2002	8	45	113	2	238
2002	9	39	102	9	197
2002	10	32	90	3	227
2002	11	27	79	5	221
2002	12	23	66	8	219
2003	1	14	64	12	195
2003	2	16	75	8	190
2003	3	25	73	10	187
2003	4	27	97	1	233
2003	5	36	102	7	220
2003	6	45	115	3	242
2003	7	37	115	1	237
2003	8	41	117	1	240
2003	9	39	104	2	235

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_10YR is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 218. Query Results

- Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
- Tab to **Curve Fitting - Univariate** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

- Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Select time-series data column	RAINFALL
Number of Periods in Year	12
Start Period	1
Start Year	2001

Table 28. Parameter Settings for Curve Fitting Univariate Analysis (continued)	
Parameter	Setting
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

Your Parameter Selection panel should look like this:

```

ANALYTICS -- UNIVARIATE CURVE FITTING

                                Parameter Selection

Select time series data column ====> ( RAINFALL                               )

Optional parameters:
    Number of Periods in Year ===> 12
      Start Period ===> 1
      Start Year ===> 2001

Chart Title ===> *

1=Help      2=Run      3=End      4=List      5=Save      6=
7=          8=          9=          10=         11=         12=

```

Figure 219. Parameter Selection Panel

- Optional: Press the Save function key (F5) if you want to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

- Press the Run function key (F2).

QMF Analytics for TSO displays a set of 10 curves in a **Curve Fitting** selection window.

The **Curve Fitting** window specifies which of the curves has the lowest residual standard deviation.

Additionally, the Curve Fitting window provides the correlation coefficient and residual standard deviation for each type of curve that can be fitted and indicates which curve is the best fitted curve. In this example, the Hyperbola curve is the best fitted curve.

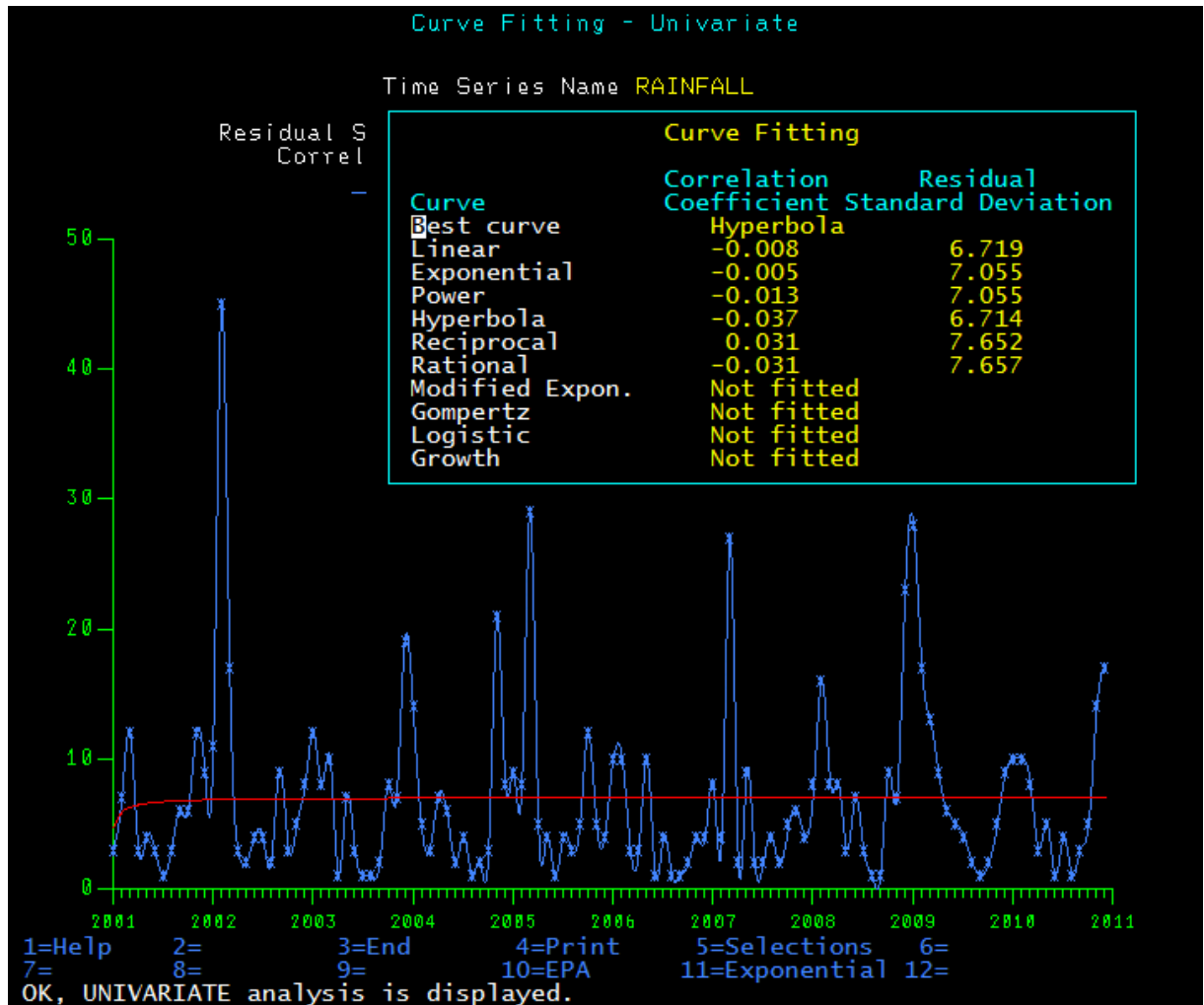


Figure 220. Curve Fitting Window for Univariate Analysis

- To draw one of the curves and re-display the statistical graph, position the cursor on the curve name and press Enter.

For example, tab to **Power** and press Enter.

QMF Analytics for TSO draws the Power® curve and displays the graph.

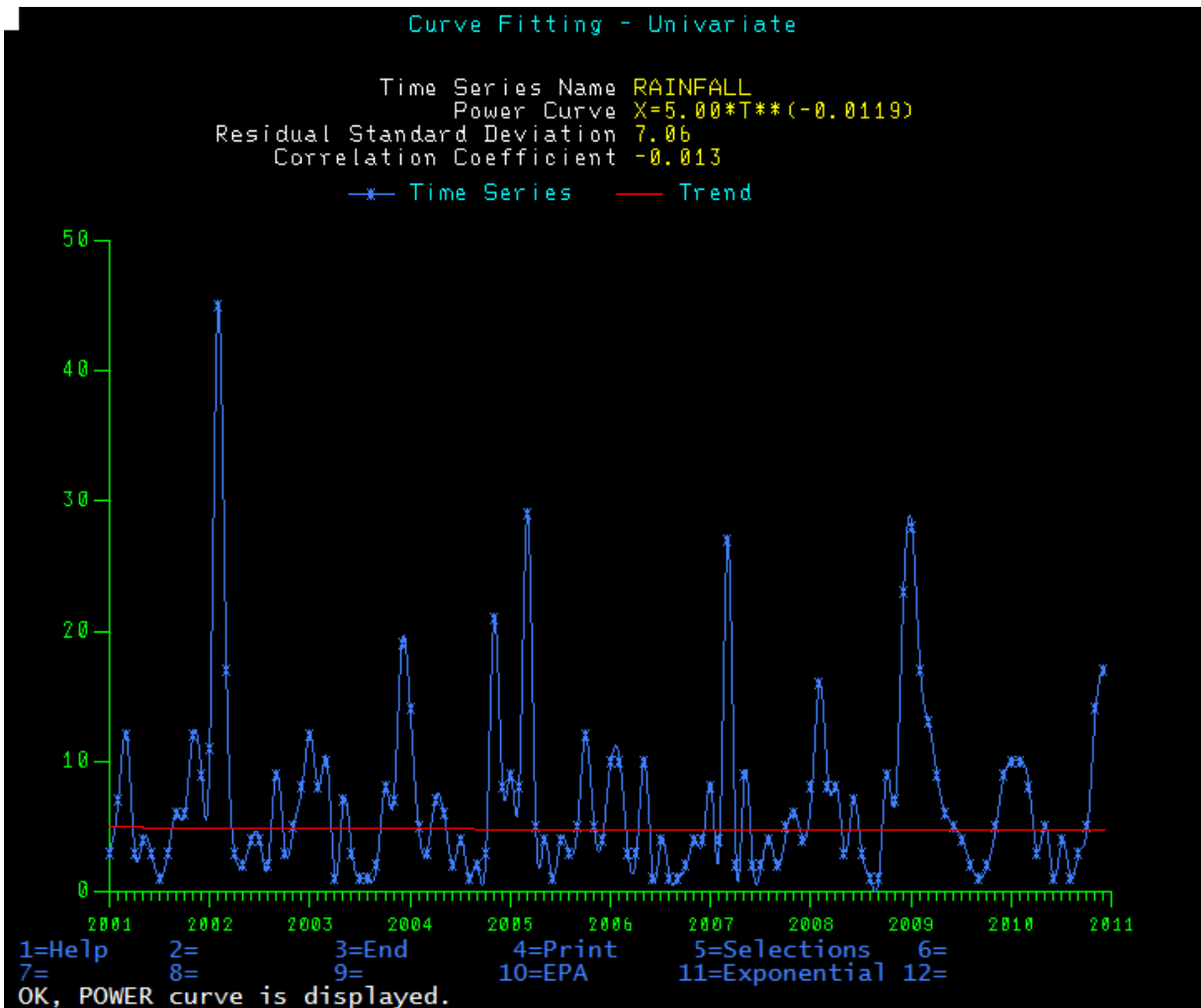


Figure 221. Curve Fitting Univariate for Power

9. From the curve analysis page, you can:

- Identify trend and seasonality data by pressing the EPA function key (10).

The Economic Planning Agency (EPA) method analyzes monthly or quarterly data.

EPA Seasonal Analysis breaks down a time series into its constituent parts and then gives you the results so that you can, for example, check for any seasonal factors. It also produces seasonal factors for the year after the data stops. The minimum amount of data required spans two years.

- Display exponential smoothing by pressing the Exponential function key (F11).

Exponential Smoothing does not remove seasonal or cyclic variation, but it *smooths* over the irregularities by averaging past values.

By giving more weight to recent values when averaging, Exponential Smoothing can be more sensitive than some other smoothing methods to shifts or cycles in the data.

- Go back to the Curve Fitting selection by pressing the Selections function key (F5).
- Print the analysis results by pressing the Print function key.

10. Press the End function key (F3) to return to the Parameter Selection panel.

11. Press the End function key (F3) to return to the QMF Analytics for TSO Home panel.

# Running a discounted cash flow analysis of your query results

You can run a discounted cash flow analysis on your query results using QMF Analytics for TSO.

## About this task

Discounted Cash Flow (DCF) analysis compares costs and expenditures on new projects or to see whether the internal rate of return on a project meets company standards.

You can also use DCF to calculate the net present value of a rate of return on your investment.

This procedure uses the Q.CASHFLOW sample table.

To run a discounted cash flow analysis:

## Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.CASHFLOW** on the command line and press Enter.

QMF for TSO displays the query result as a multi-columned report.

Figure 222. Query Results

REPORT	Q.CASHFLOW	LINE 1	POS 1	79
PERIOD	COSTS	REVENUE	CASHFLOW	
1	-800	0	-800	
2	-600	0	-600	
3	-100	200	100	
4	0	400	400	
5	0	500	500	
6	-800	300	-500	
7	0	300	300	
8	0	200	200	
9	-200	300	100	
10	0	300	300	
11	0	300	300	
12	-100	400	300	

\*\*\* END \*\*\*

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward    8=Forward    9=Form      10=Left      11=Right      12=  
OK, Q.CASHFLOW is displayed.  
COMMAND ==>

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Discounted Cash Flow** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.



Table 29. Parameter Settings for Discounted Cash Flow Analysis	
Parameter	Setting
Specify the columns to analyze	CASHFLOW
Discount rate	Entering a value in this field is optional. For this exercise, enter 100.
Chart Title	Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).  If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.  If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.

6. Optional: Press the Save function key (F5) to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

7. Press the Run function key (F2) to run the Discounted Cash Flow analysis.

By default, the **Discounted Cash Flow** analysis window includes two views. The top portion of the window shows the Figures view and the bottom portion shows the Graph view.

Use the F9 function key to toggle between the *figures-only*, *graph-only* and the default view.



Figure 223. Discounted Cash Flow Analysis

You can change the values in the **Cashflow** column and rerun the analysis.

The value shown in the **Discount Rate** field shows the value that you specified on the Parameter Selection panel.

The value shown in the **Payback Period** field, tells you the last period the cash flow becomes positive or the period when you get your money back.

The value shown in the **Double Life** field, tells you the number of periods it would take to double a sum of money if you had invested the money at that rate of return.

- Press the IRR function key (F10) to calculate the Internal Rate of Return.



Figure 224. Internal Rate of Return Analysis

IRR is a measurement for evaluating whether to proceed with a project or investment.

If the internal rate of return (IRR) on a project or investment is greater than the minimum required rate of return – the cost of capital – then the decision would generally be to go ahead with the project or investment. Conversely, if the IRR on a project or investment is lower than the cost of capital, then you might not want to proceed with the project or investment.

You can change the values in the **Cashflow** column and rerun the analysis.

9. Press the NPV function key (F11) to calculate the Net Present Value.

The NPV shows the difference between the present value of cash inflows and the present value of cash outflows.



Figure 225. Discounted Cash Flow Net Present Value (NPV) analysis

You can use NPV to analyze the profitability of an investment or project. You can change the values in the **Cashflow** column and rerun the analysis.

Table 30. Additional functions available from the Discounted Cash Flow page. Additional functions available from the Discounted Cash Flow page

Function	Result	Usage notes
Help (F1)	Opens the online help for the Discounted Cash Flow results page.	The help describes the functions that are available from the Discounted Cash Flow page.
End (F3)	Returns you to the <b>Parameter Selection</b> panel.	The <b>Parameter Selection</b> contains the values from the last analysis run. You can change the settings and rerun the analysis.
Print (F4)	The Print dialog window is displayed.	You must have print capabilities configured in QMF in order to print analysis results.
Backward (F7)	Scrolls backward.	

Table 30. Additional functions available from the Discounted Cash Flow page. Additional functions available from the Discounted Cash Flow page (continued)		
Function	Result	Usage notes
Forward (F8)	Scrolls backward.	

## Running an F-Test analysis of your query results

You can run an F-test on your query results using QMF Analytics for TSO.

### About this task

F-test tests the variances of the minimum and maximum monthly temperature.

This procedure uses the Q.CLIMATE\_10YR sample table.

To run an F-test:

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter F6 (Query) to open the query editor.
3. Type the following command and press Enter:

```
DISPLAY Q.CLIMATE_10YR
```

QMF for TSO displays the query result as a report.

```

REPORT          Q.CLIMATE_10YR          LINE 1      POS 1      79

YEAR  MONTH  TEMPMIN  TEMPMAX  RAINFALL  SUNSHINE
-----
2001   1       9        70       3         234
2001   2      18        72       7         205
2001   3      16        77      12         180
2001   4      32        91       3         230
2001   5      32        95       4         234
2001   6      41       115       3         230
2001   7      43       111       1         227
2001   8      39       115       3         238
2001   9      32        93       6         226
2001  10      27        88       6         221
2001  11      14        79      12         183
2001  12      19        73       9         204
2002   1      27        77      11         185
2002   2      25        75      45           25
2002   3      25        88      17         161
2002   4      32        90       3         226
2002   5      32       100       2         225
2002   6      43       108       4         241
2002   7      46       111       4         228
2002   8      45       113       2         238
2002   9      39       102       9         197
2002  10      32        90       3         227
2002  11      27        79       5         221
2002  12      23        66       8         219
2003   1      14        64      12         195
2003   2      16        75       8         190
2003   3      25        73      10         187
2003   4      27        97       1         233
2003   5      36       102       7         220
2003   6      45       115       3         242
2003   7      37       115       1         237
2003   8      41       117       1         240
2003   9      39       104       2         235
1=Help      2=          3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward  9=Form     10=Left    11=Right   12=
You are at the boundary of the panel.
COMMAND ==>

```

Figure 226. Query Results

4. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
5. Tab to **F-Test** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

6. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
X	TEMPMIN
Y	TEMPMAX

Table 31. Parameter Settings for F-Test (continued)	
Parameter	Setting
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

7. Optional: Press the Save function key (F5) to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

8. Press the Run function key (F2) to run the F-test.

```

                                F-Test

                                Variables TEMPMIN
                                    TEMPMAX
                                F-value 2.846256
                                Degrees of freedom 119, 119
                                Probability less than 0.0001
                                Type of test two-tailed

Null hypothesis: Variances are equal
Conclusion: The null hypothesis is rejected at 0.001 level

1=Help      2=          3=End      4=Print    5=         6=
7=          8=          9=        10=       11=       12=
OK, F-TEST analysis is displayed.

```

Figure 227. F-Test

The F-test shows the following fields:

**Variables**

Shows the names of columns under analysis.

**F-value**

The test statistic that was calculated by QMF Analytics for TSO.

**Degrees of freedom**

Degrees of freedom that were calculated by QMF Analytics for TSO.

**Probability**

Probability that was calculated by QMF Analytics for TSO for F-value by using Fisher's distribution with appropriate degrees of freedom. If the calculated probability is <0.0001, the string "less than 0.0001" is shown.

### Type of test

Shows that null hypothesis is testing apart from symmetrical alternative hypothesis. The application does not make any assumption about whether X or Y tends to have greater or smaller variance than other.

### Null hypothesis

Variances are equal.

### Conclusion

Shows you the following values:

- The null hypothesis is rejected at [numeric value] level
- The null hypothesis cannot be rejected at [numeric value] level

The [numeric value] can be 0.001, 0.01, 0.05.

The conclusion "The null hypothesis is rejected at 0.001 level" means that the variances of the minimum and maximum monthly temperature differ.

Function	Result	Usage notes
Help (F1)	Opens the online help for the F-test results page.	The help describes the functions that are available from the F-test page.
End (F3)	Returns you to the <b>Parameter Selection</b> panel.	The <b>Parameter Selection</b> contains the values from the last analysis run. You can change the settings and rerun the analysis.
Print (F4)	The Print dialog window is displayed.	You must have print capabilities configured in QMF in order to print analysis results.

## Running a linear trend analysis of your query results

You can run linear trend analysis on your query results using QMF Analytics for TSO.

### About this task

Linear trend analysis fits a straight line to your data, identifying the best linear relationship between a variable and time.

The linear trend analysis in QMF Analytics for TSO uses the *method of least-squares* mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets (the residuals) of the points from the curve.

For an example of how to analyze query results using Linear Statistics:

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter **DISPLAY Q.CLIMATE\_10YR** on the command line and press Enter.



REPORT	Q.CLIMATE_10YR	LINE 1	POS 1	79	
YEAR	MONTH	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
2001	1	9	70	3	234
2001	2	18	72	7	205
2001	3	16	77	12	180
2001	4	32	91	3	230
2001	5	32	95	4	234
2001	6	41	115	3	230
2001	7	43	111	1	227
2001	8	39	115	3	238
2001	9	32	93	6	226
2001	10	27	88	6	221
2001	11	14	79	12	183
2001	12	19	73	9	204
2002	1	27	77	11	185
2002	2	25	75	45	25
2002	3	25	88	17	161
2002	4	32	90	3	226
2002	5	32	100	2	225
2002	6	43	108	4	241
2002	7	46	111	4	228
2002	8	45	113	2	238
2002	9	39	102	9	197
2002	10	32	90	3	227
2002	11	27	79	5	221
2002	12	23	66	8	219
2003	1	14	64	12	195
2003	2	16	75	8	190
2003	3	25	73	10	187
2003	4	27	97	1	233
2003	5	36	102	7	220
2003	6	45	115	3	242
2003	7	37	115	1	237
2003	8	41	117	1	240
2003	9	39	104	2	235

1=Help      2=      3=End      4=Print      5=Chart      6=Query  
7=Backward      8=Forward      9=Form      10=Left      11=Right      12=  
OK, Q.CLIMATE\_10YR is displayed.  
COMMAND ==>>>      SCROLL ==>> PAGE

Figure 228. Query Results

QMF for TSO displays the query result as a multi-columned report.

3. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
4. Tab to **Linear Trend** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

5. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Parameter	Setting
Specify the column to forecast	RAINFALL
How many periods in a year?	12
In what period does your data start?	1

Table 33. Parameter Settings for Linear Trend analysis (continued)	
Parameter	Setting
In what year does your data start?	2001
How many periods ahead to forecast	4
Seasonal	<p>A seasonal effect is a recurring pattern in data that can be attributed to calendar event.</p> <p>An escalation in retail sales that occur in December in response to the Christmas, or an increase in water consumption in summer months are examples of seasonal effects.</p> <p>You can specify whether you want QMF Analytics for TSO to take into account the seasonal nature of data in its analysis.</p> <p>Yes</p> <p>Tab to <b>Seasonal</b> and press Enter.</p>
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

The fields **How many periods in a year?**, **In what period does your data start?** and **In what year does your data start?** provide the time series construct for data being analyzed. Data is considered to be *time series* in nature if it represents a collection of observations of well-defined data items obtained through repeated measurements over time. For example, measuring the amount of rainfall each month of the year would comprise a time series. This is because rainfall amount is well defined, and consistently measured at equally spaced intervals.

6. Press the Run function key (F2).

QMF Analytics for TSO interprets the specification, performs the analysis, and generates the output.

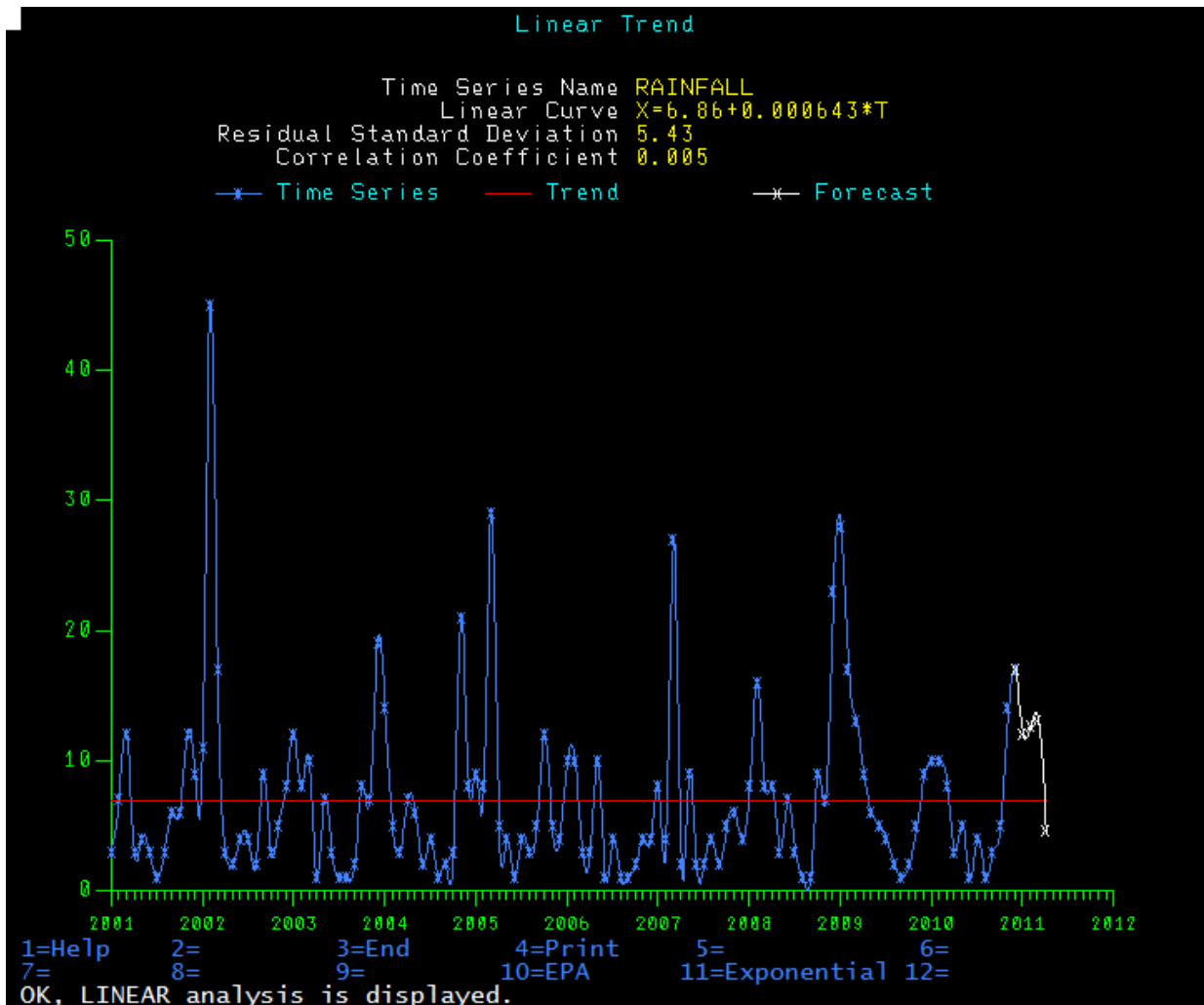


Figure 229. Linear Trend Analysis results

7. From the analysis page, you can:

- Identify trend and seasonality data by pressing the EPA function key (10).

The Economic Planning Agency (EPA) method analyzes monthly or quarterly data.

EPA Seasonal Analysis breaks down a time series into its constituent parts and then gives you the results so that you can, for example, check for any seasonal factors. It also produces seasonal factors for the year after the data stops. The minimum amount of data required spans two years. Go to step “8” on [page 215](#) for more information.

- Display exponential smoothing by pressing the Exponential function key (F11).

Exponential Smoothing does not remove seasonal or cyclic variation, but it *smooths* over the irregularities by averaging past values.

By giving more weight to recent values when averaging, Exponential Smoothing can be more sensitive than some other smoothing methods to shifts or cycles in the data. Go to step “9” on [page 217](#) for more information.

- Go back to the Parameter Selection panel by selecting the End function key (F5).
- Print the analysis results by pressing the Print function key (F3).

8. Select the EPA function key (F10) to identify trend and seasonality data.

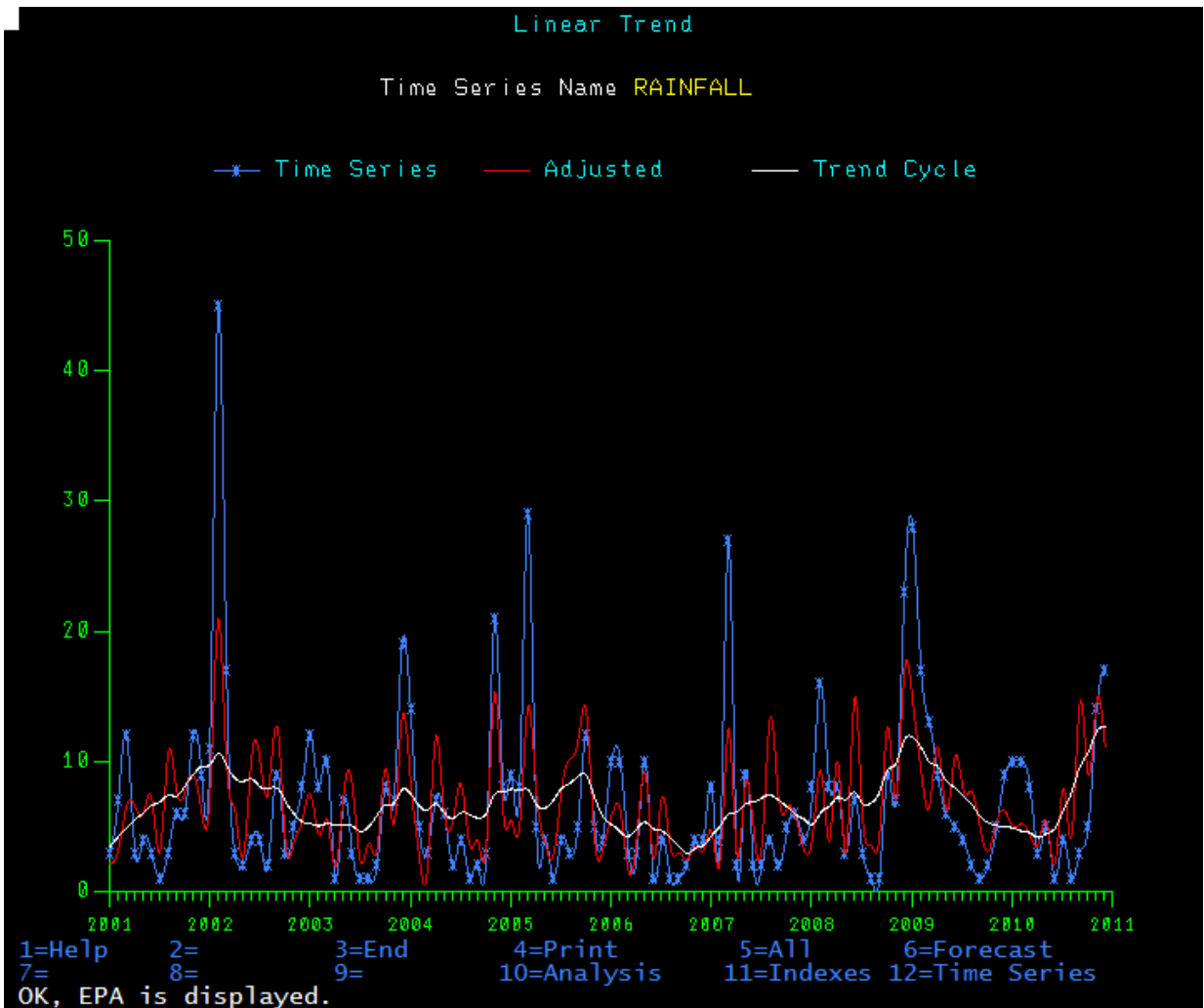


Figure 230. EPA Analysis results

a) From the EPA Analysis screen you can select from the following functions:

*Table 34. Functions from the EPA Analysis screen*

Function key	Results
All	QMF Analytics for TSO displays the graphs for analysis and indexes on one page.
Indexes	QMF Analytics for TSO displays the graphs for indexes.

<i>Table 34. Functions from the EPA Analysis screen (continued)</i>	
<b>Function key</b>	<b>Results</b>
Forecasts	<p>QMF Analytics for TSO displays the Forecast options window in the lower right corner of the graph.</p> <p>The following options are available:</p> <ul style="list-style-type: none"> <li>• Autoforecast Select this option to produce a forecast for the next year automatically.</li> <li>• Trend Selection Select this option to go to the Trend Selection window to select the trend that best corresponds to the trend cycle from EPA.</li> <li>• Curve Fitting Select this option to go to the Curve Fitting selection window to select the curve that best corresponds to the trend cycle from EPA.</li> <li>• Exponential Select this option to go to the Exponential Smoothing graph.</li> </ul>
Print	<p>Prints the EPA Analysis graph.</p> <p>You must have print capabilities configured in QMF in order to print the graph.</p>
Time Series	QMF Analytics for TSO displays data for the time series only.
End	To go back to the <b>Parameter Selection</b> panel.

9. Select the Exponential function key (F11) to smooth the time series data.

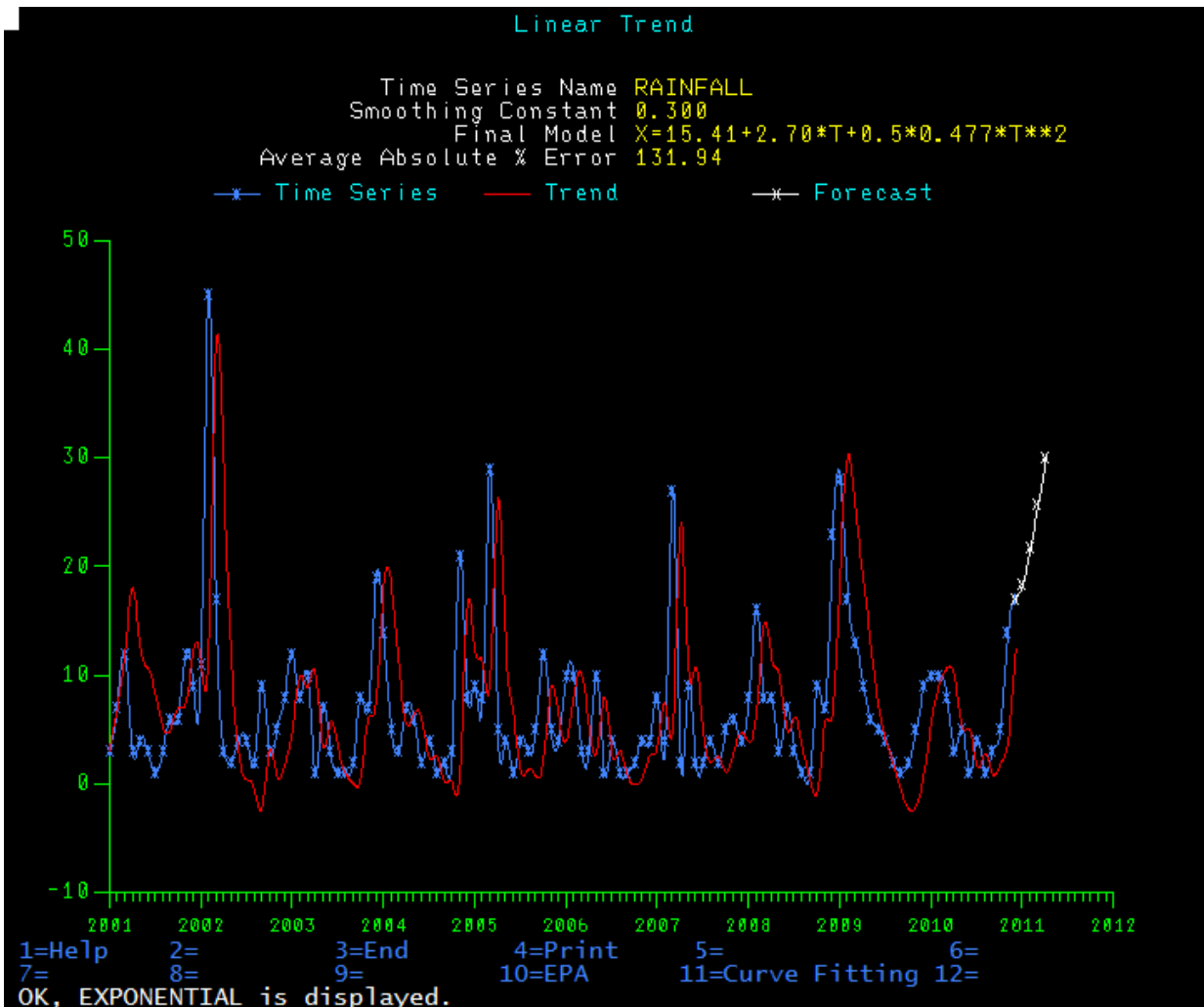


Figure 231. Exponential Analysis results

a) From the Exponential Smoothing screen you have the following options:

*Table 35. Functions from the Exponential Smoothing Analysis screen*

Function key	Results
EPA	QMF Analytics for TSO displays a graph that identifies trend and seasonality data.
Curve Fitting	QMF Analytics for TSO displays the Curve Fitting selection window from which you can select the curve that best corresponds to the analysis.
Print	Prints the Exponential Smoothing graph. You must have print capabilities configured in QMF in order to print the graph.
End	To go back to the Parameter Selection panel.

## Running a Mann-Whitney U test of your query results

You can run a Mann-Whitney U test on your query results using QMF Analytics for TSO.

### About this task

Mann-Whitney U test compares the sales volumes during two different promotional campaigns.

This procedure uses the Q.CLIMATE\_USA sample table.

**Note:** The minimum allowed sample size is three.

To run a Mann-Whitney U test:

## Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter F6 (Query) to open the query editor.
3. Type the following query and press F2 to run it:

```
SELECT (A.TEMPMIN+A.TEMPMAX)/2 AS TEMP_NC, (B.TEMPMIN+B.TEMPMAX)/2 AS TEMP_CA
FROM (SELECT ROW_NUMBER() OVER() AS MONTH, STATE, TEMPMIN, TEMPMAX
FROM Q.CLIMATE_USA WHERE STATE='NC') A
FULL JOIN
(SELECT ROW_NUMBER() OVER() AS MONTH, STATE, TEMPMIN, TEMPMAX
FROM Q.CLIMATE_USA WHERE STATE='CA') B
ON A.MONTH=B.MONTH
```

QMF for TSO displays the query result as a report.

TEMP	NC	TEMP	CA
	32		45
	39		49
	46		49
	54		53
	65		60
	74		76
	75		84
	70		76
	65		74
	58		65
	50		51
	33		46

\*\*\* END \*\*\*

1=Help            2=            3=End          4=Print          5=Chart          6=Query  
7=Backward       8=Forward     9=Form        10=Left         11=Right        12=  
OK, this is the REPORT from your RUN command.  
COMMAND ==>

Figure 232. Query Results

4. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
5. Tab to **Mann-Whitney U test** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

6. Create the specification for your analysis.

Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Table 36. Parameter Settings for Mann-Whitney U test	
Parameter	Setting
X	TEMP_NC
Y	TEMP_CA
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

7. Optional: Press the Save function key (F5) to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

8. Press the Run function key (F2) to run the Mann-Whitney U test.



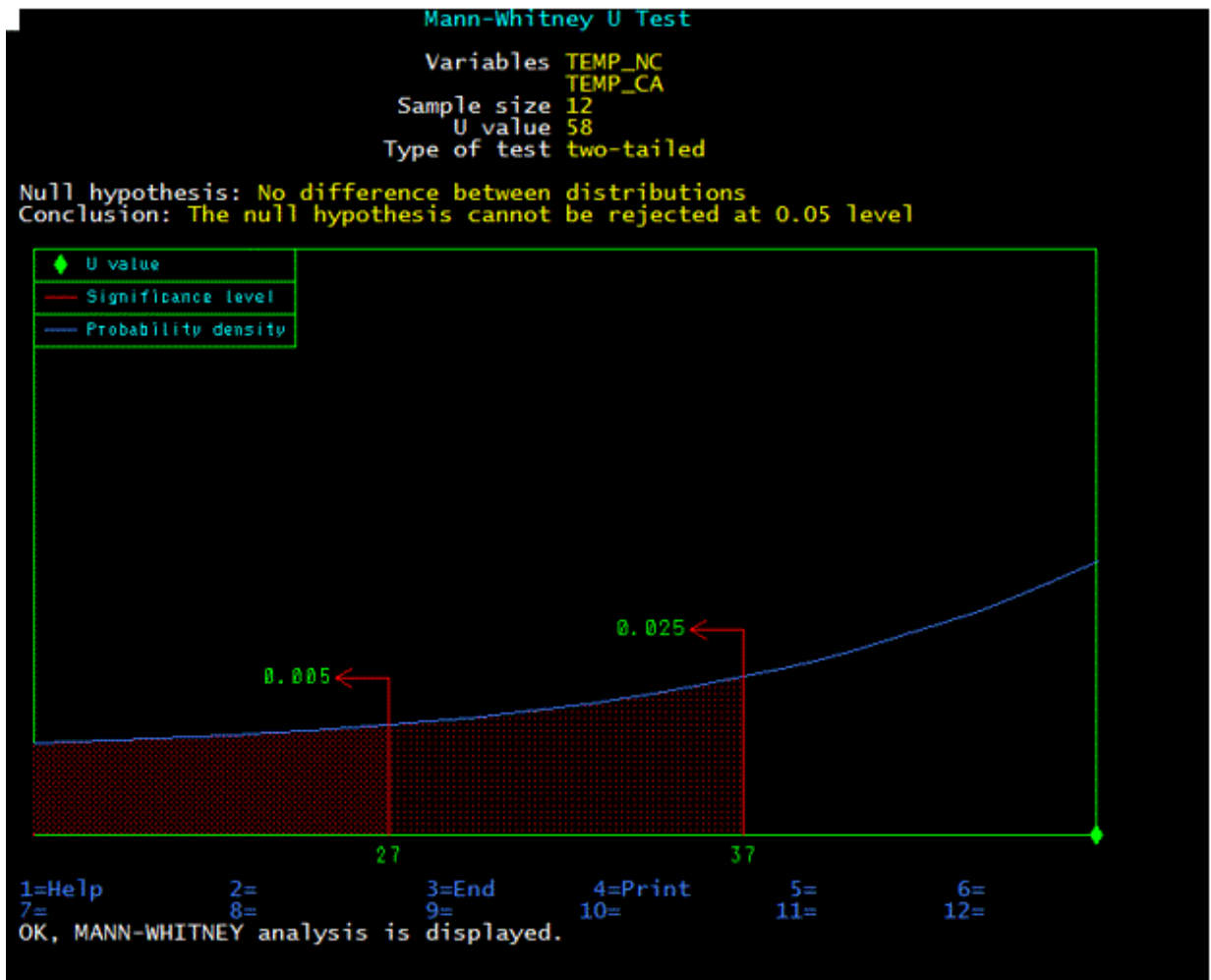


Figure 233. Mann-Whitney U test

The Mann-Whitney U test shows the following fields:

**Variables**

Shows the names of columns under analysis.

**Sample Size**

Shows the number of rows in input columns.

**U value (if sample size <=20) or Z value**

The test statistic that was calculated by QMF Analytics for TSO.

**Probability**

Appears only if sample size > 20. If calculated probability is <0.0001, the string “less than 0.0001” is shown.

**Type of test**

Shows that null hypothesis is testing apart from symmetrical alternative hypothesis. The application does not make any assumption about whether X or Y tends to have greater or smaller values than other.

**Null hypothesis**

Shows that there is no difference between distributions.

**Conclusion**

Shows you the following values:

- The null hypothesis is rejected at [numeric value] level
- The null hypothesis cannot be rejected at [numeric value] level

If sample size > 20, then the [numeric value] can be 0.001, 0.01, 0.05. If sample size <=20, then [numeric value] can be 0.01, 0.05.

**Note:** The chart appears only if sample size > 5 and shows graphical interpretation of analysis: some approximation of probability density, Z value, critical points for 0.0005, 0.005, 0.025 for sample size > 20 and critical points for 0.005, 0.025 otherwise.

U value lies beyond the red region in the example "Mann-Whitney U test" figure, so we cannot reject the null hypothesis. It means that temperature has the same distributions in both states.

<i>Table 37. Additional functions available from the Mann-Whitney U test page.</i> Additional functions available from the Mann-Whitney U test page		
<b>Function</b>	<b>Result</b>	<b>Usage notes</b>
Help (F1)	Opens the online help for the Mann-Whitney U test results page.	The help describes the functions that are available from the Mann-Whitney U test page.
End (F3)	Returns you to the <b>Parameter Selection</b> panel.	The <b>Parameter Selection</b> contains the values from the last analysis run. You can change the settings and rerun the analysis.
Print (F4)	The Print dialog window is displayed.	You must have print capabilities configured in QMF in order to print analysis results.

## Running a Wilcoxon Signed-Rank test of your query results

You can run a Wilcoxon Signed-Rank test on your query results using QMF Analytics for TSO.

### About this task

Wilcoxon Signed-Rank test compares the sales volume before and after a training course for marketing managers.

This procedure uses the Q.CLIMATE\_10YR sample table.

**Note:** The minimum allowed sample size is two.

To run a Wilcoxon Signed-Rank test:

### Procedure

1. Start QMF for TSO.

Your QMF administrator can tell you how to start a QMF session and give you a user ID. Check with your QMF administrator if you have any questions about getting started with QMF.

2. From the **QMF Home Panel**, enter F6 (Query) to open the query editor.
3. Type the following query and press F2 to run it:

```
SELECT (A.TEMPMIN+A.TEMPMAX)/2 AS T2007,(B.TEMPMIN+B.TEMPMAX)/2 AS T2008
FROM (SELECT ROW_NUMBER() OVER() AS YEAR, TEMPMIN, TEMPMAX, MONTH
FROM Q.CLIMATE_10YR WHERE YEAR='2007') A
FULL JOIN
(SELECT ROW_NUMBER() OVER() AS YEAR, TEMPMIN, TEMPMAX, MONTH
FROM Q.CLIMATE_10YR WHERE YEAR='2008') B
ON A.MONTH=B.MONTH
```

QMF for TSO displays the query result as a report.

```

      T2007      T2008
      -----      -----
          41          44
          50          48
          47          53
          57          62
          63          73
          68          73
          79          75
          78          78
          72          73
          65          61
          51          52
          47          42

*** END ***

1=Help      2=      3=End      4=Print      5=Chart      6=Query
7=Backward  8=Forward  9=Form     10=Left     11=Right    12=
OK, this is the REPORT from your RUN command.
COMMAND ==>

```

Figure 234. Query Results

4. Enter **SHOW ANALYTICS** on the command line and press Enter to start QMF Analytics for TSO.
5. Tab to **Wilcoxon Signed-Rank test** in the Statistics section of the QMF Analytics for TSO Home panel and press Enter.

The **Parameter Selection** panel is displayed.

6. Create the specification for your analysis.  
Set your parameters as shown in the following table.

You can enter column names manually or you can tab to the field and press the List function key (F4).

The List function opens a **Column Selection** window that lists all of the column names from the query result. To select a column from the **Column Selection** window, tab to the column name and press Enter. QMF Analytics for TSO adds the name to the Parameter Selection panel automatically.

Table 38. Parameter Settings for Wilcoxon Signed-Rank test	
Parameter	Setting
X	T2007
Y	T2008
Chart Title	<p>Enter a title for your chart in the <b>Chart Title</b> field or accept the default value of asterisk (*).</p> <p>If you accept the default value of asterisk (*), QMF Analytics for TSO creates a title based on the columns that you have selected.</p> <p>If you delete the asterisk (*) and leave this field blank, the analysis output will not include a title.</p>

7. Optional: Press the Save function key (F5) to save the specification.

The specification is saved to the QMF database as an *ANALYTIC* object. In Save Command Prompt, you can enter object names manually or you can tab to the field and press the List function key (F4) to see all available objects in Analytic Object List.

Saving the specification to the database allows you to run your analysis directly from the QMF command line at a later time, without having to navigate to the QMF Analytics for TSO Home panel and reenter the parameters. Saving specifications makes sense if you plan on running the analysis frequently or multiple times.

**Note:** The global variable *DSQEC\_SESSGLV\_SAV* in QMF controls how the autosave for parameter settings works. For information about Global Variables, see *Global Variable List* in the QMF help.

8. Press the Run function key (F2) to run the Wilcoxon Signed-Rank test.

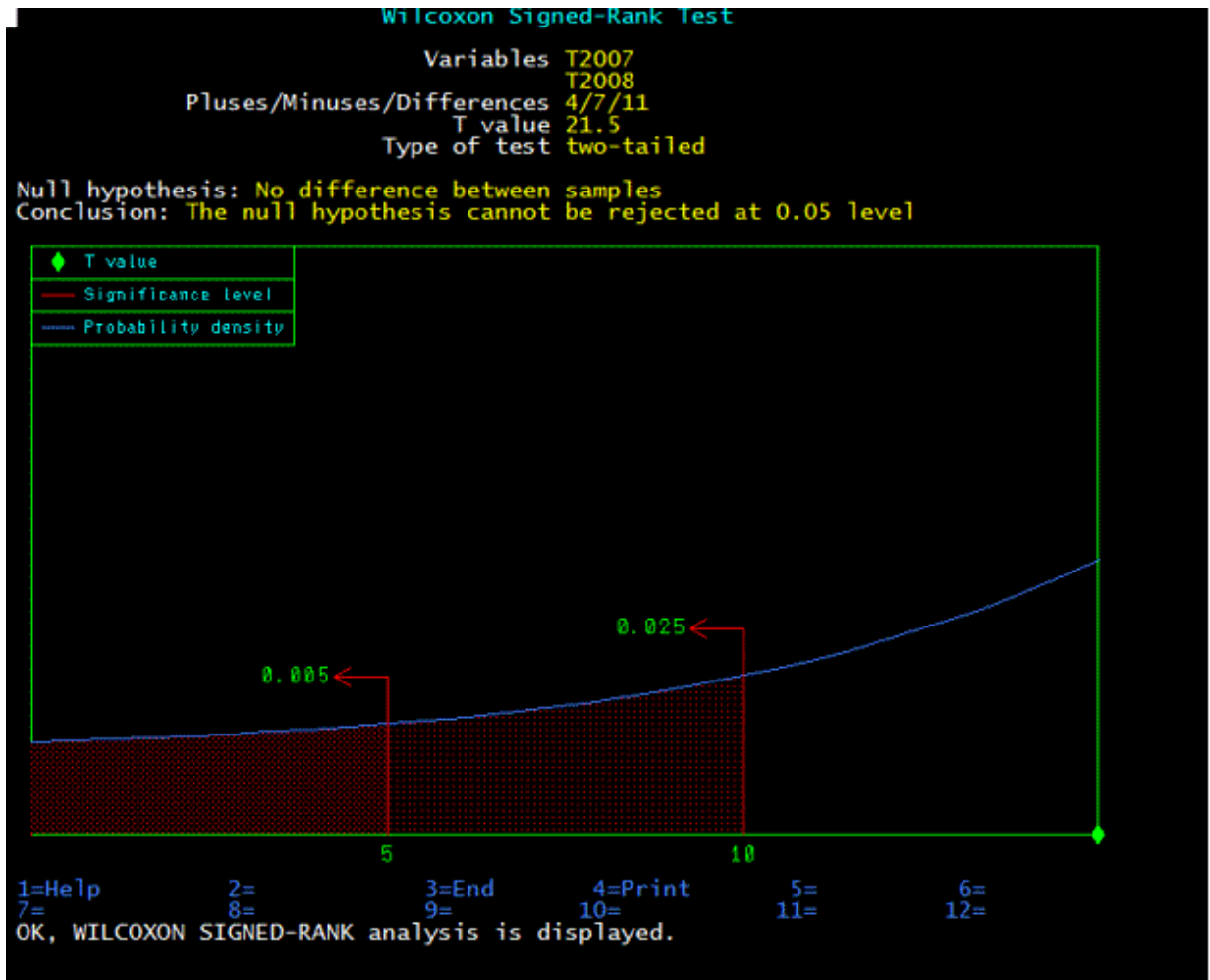


Figure 235. Wilcoxon Signed-Rank test

The Wilcoxon Signed-Rank test shows the following fields:

**Variables**

Shows the names of columns under analysis.

**Pluses/Minuses/Differences**

- Pluses – number of  $(X-Y) > 0$ .
- Minuses – number of  $(X-Y) < 0$
- Differences – number of non-zero difference between X and Y.

**T value (if sample size  $\leq 25$ ) or Z value**

The test statistic that was calculated by QMF Analytics for TSO.

**Probability**

Appears only if sample size  $> 25$ . If calculated probability is  $< 0.0001$ , the string “less than 0.0001” is shown.

**Type of test**

Shows that null hypothesis is testing apart from symmetrical alternative hypothesis. The application does not make any assumption about whether X or Y tends to have greater or smaller values than other.

### Null hypothesis

Shows that there is no difference between distributions.

### Conclusion

Shows you the following values:

- The null hypothesis is rejected at [numeric value] level
- The null hypothesis cannot be rejected at [numeric value] level

If sample size > 25, then the [numeric value] can be 0.001, 0.01, 0.05. If sample size <=25, then [numeric value] can be 0.01, 0.05.

**Note:** The chart appears only if sample size > 9 and shows graphical interpretation of analysis: some approximation of probability density, Z value, critical points for 0.0005, 0.005, 0.025 for sample size > 25 and critical points for 0.005, 0.025 otherwise.

T value lies beyond the red region in the example "Wilcoxon Signed-Rank test" figure, so we cannot reject the null hypothesis. It means that temperature has the same distributions in 2007 and 2008.

Function	Result	Usage notes
Help (F1)	Opens the online help for the Wilcoxon Signed-Rank results page.	The help describes the functions that are available from the Wilcoxon Signed-Rank page.
End (F3)	Returns you to the <b>Parameter Selection</b> panel.	The <b>Parameter Selection</b> contains the values from the last analysis run. You can change the settings and rerun the analysis.
Print (F4)	The Print dialog window is displayed.	You must have print capabilities configured in QMF in order to print analysis results.

## Running your chart or statistical analysis from QMF for TSO

You can run your QMF Analytics for TSO chart or statistical analysis from QMF for TSO, without being in the QMF Analytics for TSO component.

### Before you begin

In order to run your QMF Analytics for TSO chart or statistical analysis from QMF for TSO, it must exist as an ANALYTIC object in the QMF database.

You can save any chart or statistical analysis as an ANALYTIC object by using the Save function on the object's Parameter Selection panel.

### About this task

To run your chart or statistical analysis from QMF for TSO:

### Procedure

1. Enter one of the following commands on the QMF command line:

```
DISPLAY ANALYTICS object name
```

or

```
DISPLAY object name
```

where *object name* is the name you gave the specification when you saved it as an ANALYTIC object to the QMF database.

2. Press Enter.

QMF for TSO displays the ANALYTIC object as a chart or statistical graph, depending on the object type.

**Note:** If the columns specified and saved during Parameter Selection are not found in the current query result set, QMF issues an error message and returns you to the Parameter Selection panel where you can either select the appropriate column or exit QMF Analytics for TSO and rerun the query from QMF.

## What to do next

The following functions are available from the displayed ANALYTIC object page:

<b>Function</b>	<b>Description</b>
Help	Displays the help panel for the displayed ANALYTIC object.
End	Returns you to QMF for TSO.
Print	Opens the Print command window. You can press the Help function key from within the Print window to learn about printing ANALYTIC objects.
Parameters	Returns you to the Parameter Selection panel for the displayed ANALYTIC object.

## Chapter 8. Displaying your report as a chart

You can display your tabular data in a wide variety of charts. You can change QMF chart formats, or create new chart formats.

QMF can send your report data to the Interactive Chart Utility (ICU), which displays the data as a chart. You do not need to learn everything about the ICU to create charts. You can create many basic charts by using only the QMF interface to the ICU.

Your site might not support the use of charts. Check with your administrator before you try to create charts. You cannot chart data or tables that contain columns defined as BINARY, VARBINARY, or XML. You can chart data or tables containing DECFLOAT data only when the processor on which QMF is running supports decimal floating-point instructions.

### QMF chart formats

To use a chart format, specify its name as a parameter when you type the DISPLAY CHART command.

QMF provides the following chart formats:

- Bar (the QMF default chart)
- Pie
- Line
- Tower
- Table
- Polar
- Histogram
- Surface
- Scatter

### Where QMF report data appears on a chart

To understand how to present and label data in a chart, review where the report data appears in the chart.

Compare the report in Figure 1 and the bar chart in Figure 2 to see how QMF displays report data on a chart. Both were created using the default report form.

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
120	NAUGHTON	38	CLERK	-	12954.75	180.00
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
40	O'BRIEN	38	SALES	6	18006.00	846.55
60	QUIGLEY	38	SALES	-	16808.30	650.25
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
200	SCOUTTEN	42	CLERK	-	11508.60	84.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
250	WHEELER	51	CLERK	6	14460.00	513.30
220	SMITH	51	SALES	7	17654.50	992.80
150	WILLIAMS	51	SALES	6	19456.50	637.65

Figure 236. This report shows employee data.

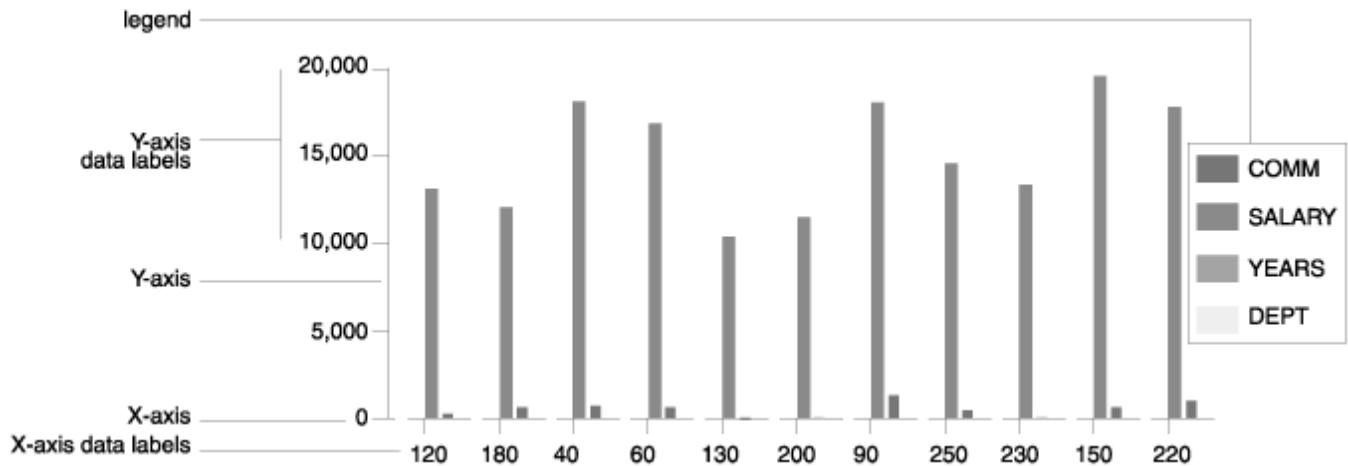


Figure 237. This bar chart shows the same employee data as the report.

You specify the following parts of a chart using a QMF form:

- Chart heading
- X-axis
- X-axis data labels
- Y-axis
- Y-axis data labels
- Legend

The following table shows the correspondence between report and chart data.

This part of the report:	Corresponds to this part of the chart:
Report heading	Chart heading
First (left-most) column of the report. If you define a GROUP or BREAK column, the data in that column appears on the X-axis	X-axis data
Values in the left-most column, or the GROUP or BREAK column	X-axis data labels
Remaining numeric columns	Y-axis data
Values in the remaining numeric columns	Y-axis data labels
Column headings for the Y-axis data	Legend

## How QMF spaces data along the X-axis

The formats that QMF provides for bar, tower, and polar charts space both numeric values and non-numeric values at even intervals. If you specify one of these chart types in the ICU, rather than using the QMF DISPLAY command, your data might be spaced unevenly along the X-axis

The following rules apply to how QMF spaces data along the X-axis for the chart formats that it provides:

- QMF plots numeric data from a single report column by its actual numeric value.
- QMF spaces non-numeric data from a single report column at even intervals.
- QMF spaces numeric data or non-numeric data from multiple report columns at even intervals.



## Where data appears on pie charts

Pie charts do not have a typical X-axis and Y-axis.

Compare the report in Figure 1 and the pie chart in Figure 2 to see how QMF displays data on a pie chart.

<----- JOB ----->			
DEPT	<- CLERK --> AVERAGE SALARY	<- SALES --> AVERAGE SALARY	<- TOTAL --> AVERAGE SALARY
15	12383.35	16502.83	13756.51
20	13878.68	18171.25	15309.53
38	12482.25	17407.15	14944.70
	=====	=====	=====
	12914.76	17372.10	14697.69

Figure 238. This report shows department salary averages.

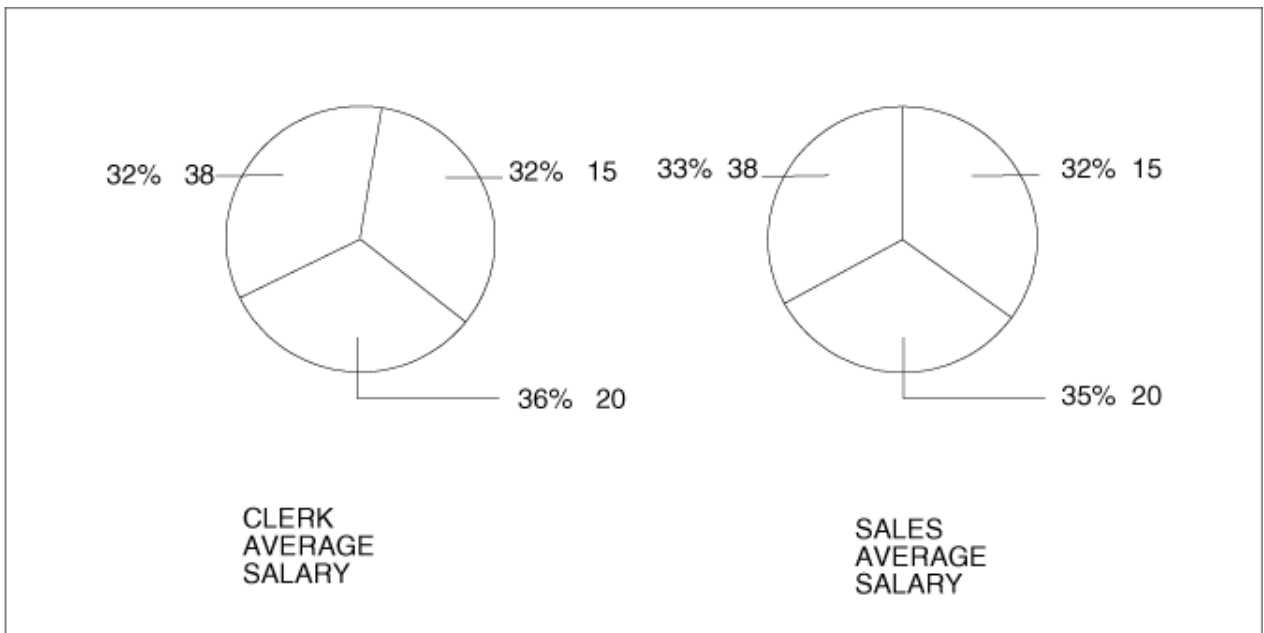


Figure 239. The pie chart displays the same data.

Each numeric column (called a Y-data column) produces a separate pie. QMF displays column data as labels attached to the pie slices. It displays column headings as titles for the pies.

Because each numeric column on a report produces a separate pie, you probably will not want to use a pie chart to display data from reports with more than two numeric columns.

## Chart data size limits

QMF and the Interactive Chart Utility (ICU) both have restrictions on the data size of charts.

The following size limits apply to chart data.

- 132 bytes for chart title, X-axis label, and legend label
- 999 X-data values (report data rows)
- 999 Y-data groups (report data columns)
- 8,192 Y-data values (number of X-data rows times the number of Y-data columns per row)

QMF defines the last restriction. The others are ICU limits that are validated by QMF.

## Displaying report data as a chart

You can use the Chart function key to display report data as a chart.

### Procedure

To display report data as a chart:

1. Run the query to display the report.

For this example, run the prompted query that is shown in the following figure:

```
PROMPTED QUERY          MODIFIED LINE 1

Tables:
  Q.STAFF

Columns:
  ID
  NAME
  DEPT
  JOB
  YEARS
  SALARY
  COMM

Row Conditions:
  If DEPT Is Equal To 38, 42 Or 51
  And JOB Is Not Equal To 'MGR'

Sort:
  Ascending by DEPT
  Ascending by JOB

*** END ***
```

Figure 240. Use this query to produce the charts for this example procedure.

This query produces the report that is shown in the following figure:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
120	NAUGHTON	38	CLERK	-	12954.75	180.00
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
40	O'BRIEN	38	SALES	6	18006.00	846.55
60	QUIGLEY	38	SALES	-	16808.30	650.25
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
200	SCOUTTEN	42	CLERK	-	11508.60	84.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
250	WHEELER	51	CLERK	6	14460.00	513.30
220	SMITH	51	SALES	7	17654.50	992.80
150	WILLIAMS	51	SALES	6	19456.50	637.65

Figure 241. Report produced from a typical prompted query

2. Press the Chart function key. (Or you can enter DISPLAY CHART.)

If you do not specify a chart type, QMF creates the chart with the GDDM default chart type. The following examples assume that the GDDM default chart type is bar.

To specify a different chart type, enter:

```
DISPLAY CHART (ICUFORM=charttype)
```

While the ICU is creating the chart, you see a panel like the one in the following figure.

## GRAPHICS BRIDGE STATUS PANEL

Your request is currently being processed by the  
GDDM\* Interactive Chart Utility (ICU).

When the chart is displayed, you will be within the ICU Environment.  
To return to the QMF environment, press the Exit key.

Figure 242. The **Graphics Bridge Status** panel

QMF creates the chart that is shown in the following figure by using the default report form and the default chart format. You can change the appearance of a chart by changing the QMF form and the ICU values.

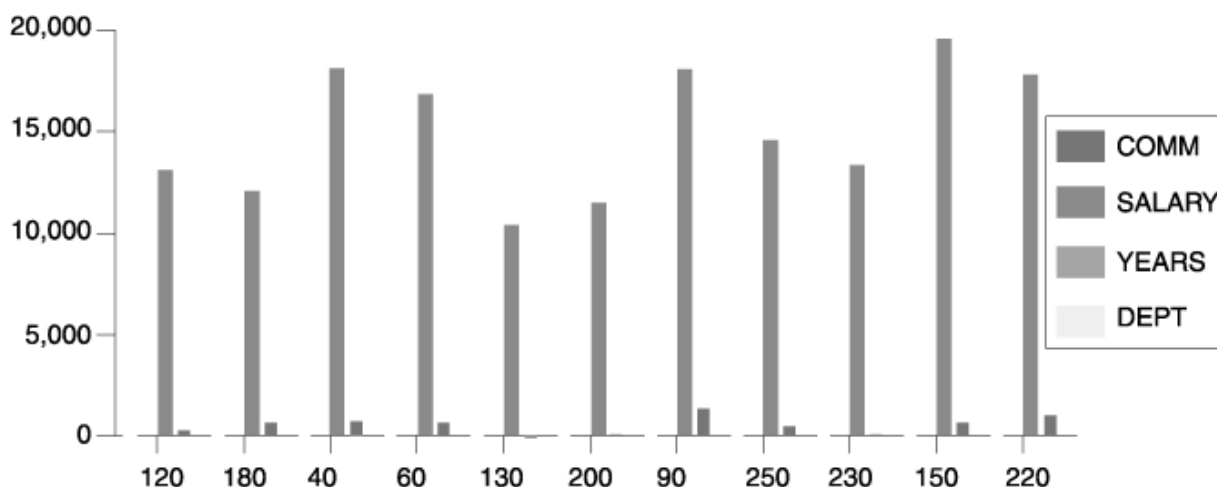


Figure 243. Format of the default bar chart

3. Press the Home function key followed by the Exit function key to return to the **QMF** panel where you issued the DISPLAY CHART command.

### Related concepts

#### QMF chart formats

To use a chart format, specify its name as a parameter when you type the DISPLAY CHART command.

## Changing a chart with QMF forms

In this example, you modify the QMF default form to display a bar chart that shows average salaries for clerks and salespeople by department.

### About this task

You can change the format and content of your chart by using QMF form panels. For example, you can make the following types of changes on a form panel:

- Use the OMIT usage code for columns you do not want to display on the chart.
- Use the GROUP usage code to group all the entries in a particular column and display the grouped data on the chart.
- Use the AVERAGE usage code to calculate the average value for grouped data and display the average on the chart.
- Change the PAGE HEADING field to change the chart heading.
- Change the chart legend by changing column headings.

Use QMF forms to change any chart property that is determined by the report data. See the following quick reference for ways that you can change chart properties.

Change	Object	Comments
Chart type	CHART	Use ICUFORM parameter
X-axis data label text	FORM	Left-most column (BREAK or GROUP)
X-axis data label length	FORM	WIDTH of left-most column
Y-axis chart data	FORM	Numeric data columns (not OMIT)
Legend text*	FORM	Column headings of Y-data
Legend case	PROFILE	Select UPPER or STRING for the CASE option
Chart heading text	FORM	Page heading
Chart heading case	PROFILE	Select UPPER or STRING for the CASE option

\* Legend text for pie charts is an exception. The data from the report column that normally appears as labels along the X-axis appears as labels attached to the pie slices.

## Procedure

1. Display the FORM panel that you need to change.

In this example, you make all the form changes on the **FORM.MAIN** panel. Enter SHOW FORM on the QMF command line.

2. Type the changes to the form.

For this example, type the changes that are shown on the **FORM.MAIN** panel in the following figure.

```

FORM.MAIN                                MODIFIED
          WARNING
COLUMNS:          Total Width of Report Columns: 24
  NUM COLUMN HEADING          USAGE  INDENT WIDTH EDIT  SEQ
-----
   1 ID                      OMIT    2     6    L    1
   2 NAME                     OMIT    2     9    C    2
   3 DEPT                     GROUP   2     6    L    3
   4 JOB                       GROUP   2     5    C    4
   5 YEARS                    OMIT    2     6    L    5

PAGE:   HEADING  ==>> SALES AND CLERK AVERAGE SALARIES FOR 1997
        FOOTING  ==>>
FINAL:   TEXT    ==>>
BREAK1:  NEW PAGE FOR BREAK? ==>> NO
        FOOTING  ==>>
BREAK2:  NEW PAGE FOR BREAK? ==>> NO
        FOOTING  ==>>
OPTIONS: OUTLINE? ==>> YES           DEFAULT BREAK TEXT? ==>> YES

```

Figure 244. Change the chart format on the **FORM.MAIN** panel.

If the sixth and seventh columns do not appear on the panel, scroll forward to see them. Here is the information that you need to change for those columns:

NUM	COLUMN HEADING	USAGE
-----	----------------	-------

6	<b>SALARY</b>	<b>AVERAGE</b>
7	<b>COMMISSION</b>	<b>OMIT</b>

3. Press the Report function key to see the changed report, which is shown in the following figure.

SALES AND CLERK AVERAGE SALARIES FOR 1997		
DEPT	JOB	AVERAGE SALARY
38	CLERK	12482.25
38	SALES	17407.15
42	CLERK	11007.25
42	SALES	18001.75
51	CLERK	13914.90
51	SALES	18555.50
		=====
		14975.99

Figure 245. The report reflects the changes you made on the **FORM.MAIN** panel.

4. Enter DISPLAY CHART or press the Chart function key to display your new chart, which is shown in the following figure:

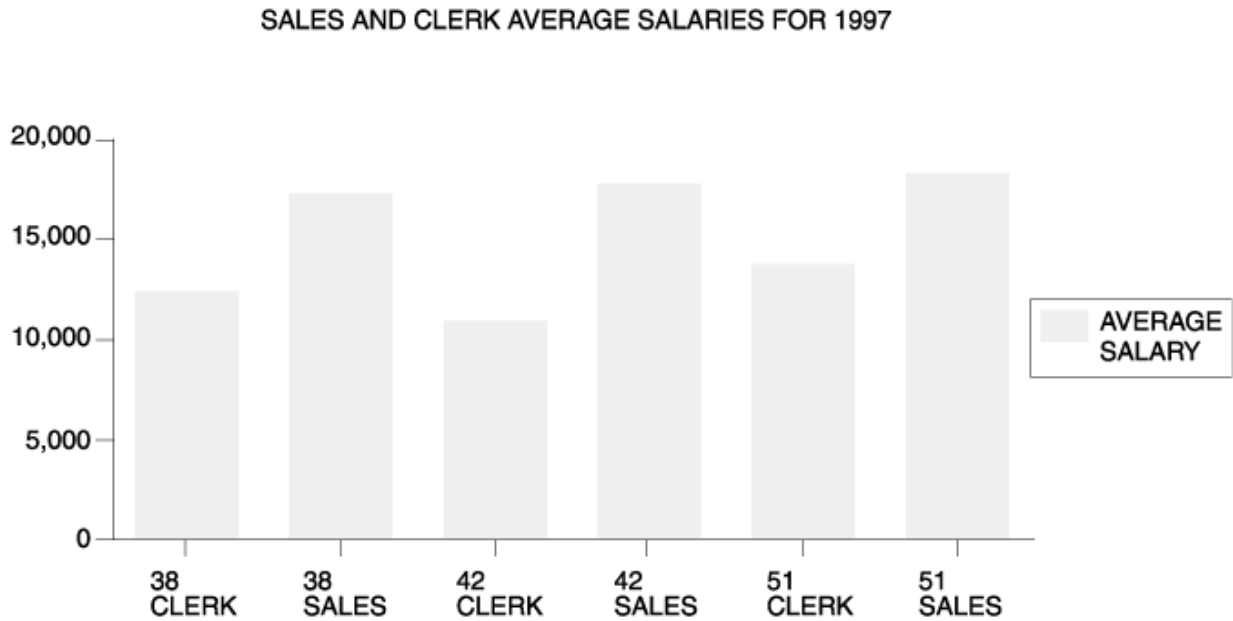


Figure 246. The chart reflects the same changes.

### Example

This chart makes it difficult to see divisions by department. You can change the QMF form to display the same data in a way that makes it easier to compare the salaries in the chart.

For this example, type the information on the **FORM.MAIN** panel. The following figure shows the information to type. The GROUP usage code groups all the information in a particular category (such as DEPT) together, and the ACROSS usage code displays a category of data (such as JOB) broken down into subcategories (such as SALES and CLERK) within a larger category (such as DEPT).

```

FORM.MAIN                                MODIFIED
COLUMNS:
NUM COLUMN HEADING                       USAGE  INDENT WIDTH EDIT  SEQ
-----
 1 ID                                     OMIT   2     6    L    1
 2 NAME                                   OMIT   2     9    C    2
 3 DEPT                                   GROUP  2     6    L    3
 4 JOB                                     ACROSS 2     5    C    4
 5 YEARS                                   OMIT   2     6    L    5

PAGE:  HEADING  ==> SALES AND CLERK AVERAGE SALARIES FOR 1997
      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=Check    3=End        4=Show    5=Chart    6=Query
7=Backward 8=Forward  9=          10=Insert 11=Delete 12=Report
OK, FORM.MAIN is shown.
COMMAND ==>                                SCROLL ==> PAGE

```

Figure 247. Group data for the chart on the **FORM.MAIN** panel.

The report looks like the following figure.

<----- JOB ----->			
DEPT	<- CLERK --> AVERAGE SALARY	<- SALES --> AVERAGE SALARY	<- TOTAL --> AVERAGE SALARY
38	12482.25	17407.15	14944.70
42	11007.25	18001.75	13338.75
51	13914.90	18555.50	16235.20
	=====	=====	=====
	12468.13	17985.41	14975.99

Figure 248. The report shows average salaries for salespeople and clerks.

Within each department, the report displays two different salary columns: an average salary for clerks and an average salary for salespeople. Each column is a bar on the chart. The TOTAL column and the final summary line do not appear on a chart. The chart is shown in the following figure.

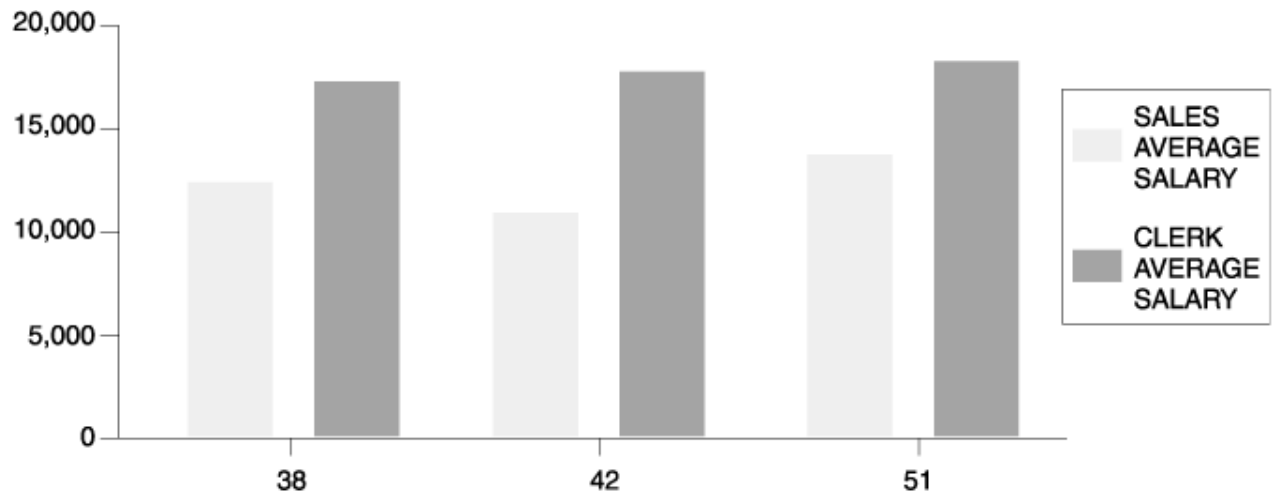


Figure 249. The chart makes comparing average salaries easier.

## Working with charts using the ICU

The changes you make to a chart by using QMF forms usually affect the data you want to display on the chart. The changes you make to a chart in the Interactive Chart Utility (ICU) usually affect the format of the chart itself.

Use the ICU to change chart presentation properties, such as color, position, and size. The following table identifies changes you can make to charts using the ICU.

*Table 43. Quick reference for changing a chart in the ICU*

Change	ICU home panel	Select
Axis titles	AXIS OPTIONS	X-axis; function key for Y-axis
Legend position	HEADING, LEGEND, AND LAYOUT	Legend Position and Format
Color and size of legend text	DATA ENTRY & IMPORT	Group Name Attributes
Chart size	HEADING, LEGEND, AND LAYOUT	Chart Dimensions
Color and appearance of data	DATA ENTRY & IMPORT	Data Attributes
Position of data along X-axis	DATA ENTRY & IMPORT	Data Interpretation
X-axis label color and size	DATA ENTRY & IMPORT	Data Label Attributes
Chart heading color and size	HEADING, LEGEND, AND LAYOUT	Chart Heading (answer YES)
Heading and axis positions	HEADING, LEGEND, AND LAYOUT	Heading and Axis Positions

### Changing a chart format

You can change a chart format in the ICU to meet your requirements, such as to move the legend, or to change the size of the pie in a pie chart.

#### Procedure

To change a chart format in the Interactive Chart Utility (ICU):

1. On the QMF command line, enter:

```
DISPLAY CHART (ICUFORM=charttype)
```

Where *charttype* is the type of the chart you want to change. The chart of the type you specified displays.

2. Press the Home function key; the ICU home panel displays.
3. Select the menu for the chart properties you want to change.
4. Make the appropriate selections for the changes.

### Saving a chart format

You can save a chart from any ICU panel that has a Save function key or a Save/Load function key.

#### Procedure

To save a chart in the Interactive Chart Utility (ICU):

1. Return to the ICU home panel or any panel that has a Save function key or a Save/Load function key.
2. Press the Save or Save/Load key. The Save and Load Chart panel displays.
3. In the **What do you want to do?** field, type 3.
4. Because you want to save only the chart format, type 1 in the **Which part of chart?** field.
5. In the **Format** field under **Filename**, type the name of the chart (for example, MYTOWER).

6. Press Enter.

## What to do next

To display the chart again from the QMF command line, type its name for the ICUFORM parameter. For example, enter:

```
DISPLAY CHART (ICUFORM=MYTOWER
```

## Specifying a new default chart format

You can change the default chart format from one chart format that QMF provides to another. In this example, you change the default chart format from BAR to LINE.

### Procedure

1. On the QMF command line, enter:

```
DISPLAY CHART (ICUFORM=charttype
```

Where *charttype* is the chart type of the chart you want to use as the default.

For this example, enter:

```
DISPLAY CHART (ICUFORM=LINE
```

2. Set the REPLACE option to YES, and save the chart as DSQCFORM (the QMF default report format) on the ICU Save panel.

## Fixing problems with charts

When you display your QMF report data as a chart in the ICU, you may not see exactly what you expect.

This topic covers some common problems and how to fix them.

- Chart does not display X-axis labels or Y-axis labels

Do one of the following:

- Use the QMF form to truncate the labels by reducing the column widths for those particular columns.
- Make the labels smaller or set them at an angle (or both) using the ICU menu for data label attributes.

- Chart does not display all the pie charts indicated by the data

There is not enough space on your screen to display all the pies at a reasonable size. In the ICU, reduce the margins by using the menu that is associated with headings, legends, and layout.

- Omitted data value labels

The data labels cannot fit on your chart. Do one of the following:

- In QMF, reorder the data in your query so that QMF can group values appropriately, which requires fewer labels on the X-axis.
- In the ICU, reduce the margins of your chart.
- In the ICU, put the labels in a legend rather than attach them to the pie slices. Use the menu that allows you to specify chart options for each chart type.

- Wrong spacing of X-axis data

In this case, the position of data along the X-axis is either spaced at equal intervals and you want it spaced according to numeric value, or just the opposite. Do one of the following:

- In QMF, specify an alternate chart type by using a different chart format that QMF provides. Each chart format provides the type of X-axis that is most typically used with its given chart type.



- In the ICU, change the way the data is spaced along the X-axis by using the menu for data interpretation.
- Error messages indicating that the values for one or more of the chart axes exceeds numeric bounds  
If the data you are charting contains a column defined with the BIGINT or DECFLOAT data type, values in this column can exceed the supported numeric limits, resulting in either a GDDM or QMF error message. To prevent this problem, you can:
  - Assign the OMIT usage code to the column to omit the column from the report and chart.
  - Change your query so that it does not select data from the column.
  - Change the data in the column so that it is within the expected range for the data type (if a range is shown in the error message).

## Printing your chart

---

In addition to displaying your ICU chart online, you can print your chart. The chart must be in temporary storage.

### About this task

The rules for printing QMF charts vary depending on which operating system you use and how you set up your printer. See your administrator if you need help printing charts.

### Procedure

To print a chart, enter the following command on the QMF command line:

```
PRINT CHART (PRINTER=printer
```

In this statement, *printer* is the printer nickname your administrator set up for you.

When you initiate the PRINT command from a chart panel, you do not need to specify the object type in the command. From a chart panel, the object type for the PRINT command defaults to CHART.

### Related reference

[PRINT in CICS](#)The PRINT command in CICS prints a copy of an object in the QMF temporary storage area or an object that is stored in the database.

[PRINT in TSO](#)The PRINT command prints a copy of an object in the QMF temporary storage area or an object that is stored in the database.



---

## Chapter 9. Procedures to run QMF commands

You can create two types of procedures: procedures with logic or linear procedures.

**Restriction:** If you are using QMF in the CICS environment, you can only use linear procedures. If you are using QMF in the TSO environment, you can use linear procedures and procedures with logic.

### Linear procedures

---

You can create a linear procedure to runs a series of QMF commands.

#### An example linear procedure

This example linear procedure selects data, saves the data in a separate table, and prints a report.

##### Procedure

1. Create and save any objects that the procedure must retrieve from the database.

In this example, REPT4QRY is the name of the saved query, and REPT4FORM is the name of the saved form. Here is the query for this example:

```
SELECT NAME, ID, COMM
FROM Q.STAFF
```

2. On the QMF command line, enter:

```
RESET PROC
```

The **PROC** panel displays.

3. Type the QMF commands that you want this procedure to run in the order you want them to run.

If you want to display and interact with panels just as you would if you entered a command on the QMF command line, type **INTERACT** before the command name.

4. Add, modify, or delete lines in the procedure:

- Type comment lines if you need them. A comment line is a line that begins with two hyphens (--). These lines are ignored when the procedure runs.
- To insert lines in a procedure, move the cursor to the line you want to precede the new line, and press the Insert function key. You can also type **INSERT** on the QMF command line, move the cursor to the line you want to precede the new line, and press Enter.
- To delete lines from a procedure, move the cursor to the line you want to delete and press the Delete function key. You can also type **DELETE** on the QMF command line, move the cursor to the line you want to delete, and press Enter.

The following figure shows a procedure that runs the REPT4QRY.

```

PROC                                MODIFIED    LINE 1
-- MONDAY MORNING REPORT.
-- PROCEDURES MAY CONTAIN COMMENT LINES; THEY BEGIN
-- WITH TWO HYPHENS.
-- A TITLE OR IDENTIFIER AT THE BEGINNING IS USEFUL.

RUN QUERY REPT4QRY (FORM=REPT4FORM
-- THIS COMMAND RUNS YOUR QUERY AND FORMATS THE REPORT.

SAVE DATA AS LASTWEEKDATA (CONFIRM=NO
-- THIS COMMAND SAVES YOUR DATA AND OVERRIDES THE VALUE OF
-- CONFIRM IN YOUR PROFILE FOR THE DURATION OF THE COMMAND.

PRINT REPORT (LENGTH=50
-- THIS COMMAND PRINTS THE REPORT.
-- YOU MAY OR MAY NOT WANT TO CHANGE PRINTING
-- SPECIFICATIONS BY USING OPTIONS OF THE PRINT COMMAND.
MESSAGE (TEXT 'OK, LASTWEEKDATA HAS BEEN SAVED AND PRINTED.'
--THE MESSAGE COMMAND CAN BE USED TO DISPLAY A MESSAGE WHEN THE
--PROCEDURE HAS FINISHED.

*** END ***
1=Help      2=Run      3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward  9=Form    10=Insert 11=Delete  12=Report
OK, cursor positioned.
COMMAND ===>                                SCROLL ===> PAGE

```

Figure 250. Enter your procedure on the **PROC** panel.

- To save the procedure in the database, enter:

```
SAVE
```

QMF prompts you for the name you want to assign to the procedure.

You can also enter: `SAVE AS procname`. For this example, enter:

```
SAVE AS MONDAY
```

### Related reference

[INTERACT](#)The INTERACT command enables user interaction while a procedure or application is running.

## Guidelines for writing linear procedures

When writing linear procedures, keep some specific guidelines in mind.

- A linear procedure can contain QMF commands, comment lines beginning with two hyphens (--), and blank lines.
- Use complete names for commands, options, and values, rather than abbreviated names.
- Do not specify a command of more than 2,000 characters (or the equivalent in DBCS). QMF stops running a procedure when it finds a command of over 2,000 characters.
- You can include comments on the same line as a command, but place them after the command.

If a command spans more than one line, type + at the beginning of the continuation line. The + sign is the continuation character. For example:

```
RUN QUERY (&&VAR1 = 'THIS IS A VALUE FOR VAR1' &&VAR2 = 'THIS
+IS A VALUE FOR VAR2')
```

QMF does not insert a space between the last character of the first line and the first character of the second line, unless either of the following is true:

- The command includes an open quotation mark
- You included a space at the end of the first line

You cannot use a continuation character in a comment line, command keyword, or substitution variable name. You can use a continuation character in a substitution variable value if you enclose the value in single quotation marks.

You can type comments and blank lines between continuation lines.

- To continue an object name across one or more lines, enclose each part of the name in double quotation marks and enter a + sign (which is the continuation character) when you split the object name. The following figure shows an example that uses a long name to refer to an object:
  - The name of the object owner is MIRIAM\_SMITH\_AT\_OFFFC1234
  - The object name is OVERSEAS\_PARTS\_LIST\_ASIA\_VIEW

```
PROC          MIRIAM_>.ASIAPARTSPROC>          LINE    1
  DISPLAY "MIRIAM_SMITH_AT_OFFFC1234".
+"OVERSEAS_PARTS_LIST_ASIA_VIEW"

*** END ***

1=Help      2=Run      3=End      4=Print    5=Chart    6=Query
7=Backward  8=Forward  9=Form    10=Insert  11=Delete  12=Report
COMMAND ==>
```

Figure 251. A QMF command that spans two lines in a linear procedure

## Procedures with logic

You can use both QMF and REXX commands when you create procedures with logic.

### An example procedure with logic

This example procedure with logic automatically prints a report on a particular day.

#### About this task

Procedures with logic can run both QMF and REXX commands. The rules and the structure of procedures with logic follow the rules of any REXX program.

This example shows how to print the commissions report shown in [Figure 250 on page 240](#), but adds REXX logic to check whether the day is Monday. If it is Monday, the procedure automatically prints the report.

#### Procedure

1. Create and save the query and form.
2. Enter:

```
RESET PROC
```

The **PROC** panel displays.

3. Type a REXX comment line as the first line of the procedure so that QMF recognizes the procedure as a procedure with logic.

REXX comment lines begin with `"/*` and end with `*/`. The first line of a procedure must be a comment, but you can also use them anywhere to explain the commands in the procedure.

4. Type the QMF commands that you want the procedure to run in the order you want them to run.

Because QMF does not convert any text in a procedure, type all QMF commands in uppercase or they will not run.

Enclose all QMF commands in quotes to distinguish them from REXX commands; any QMF command that is identical to a REXX command (such as EXIT) is processed as a REXX command.

If you want to display and interact with panels just as you would if you entered a command on the QMF command line, type INTERACT before the command name.

5. Type the logic statements for the procedure.

You can use any REXX function in a procedure with logic.

You can also include internal functions for arithmetic operations, character manipulation, data conversion, and information gathering, and you can write your own external functions.

6. Type a REXX exit statement at the end of the procedure.

The procedure in the following figure has two exit statements. One has an exit code of 0, meaning that the procedure ran successfully. The other has a return code of 8, meaning that an error occurred while the procedure was running. The procedure uses the QMF MESSAGE command to issue message text to help users diagnose the error.

```
PROC                                MODIFIED    LINE    1
/* This procedure checks to see what day it is.  If it's
Monday, it runs a query and prints a report.  If it
isn't, a message is displayed informing the user.  */
signal on error
if date('w') = 'Monday' then
do
  "RUN QUERY MYQUERY (FORM = MYFORM"
  "PRINT REPORT"
  "MESSAGE (TEXT='OK, MONDAY report has been created and sent to printer.'"
end
else
do
  "MESSAGE (TEXT='Sorry, it is not Monday.  Report cannot be created.'"
end
exit 0          /*Exit without errors */
error:
  "MESSAGE (TEXT = '"dsq_message_text"'"
exit 8         /*Exit with error condition*/
*** END ***
```

*Figure 252. This procedure produces a commission report on Mondays.*

7. To insert lines in a procedure, move the cursor to the line you want to precede the new line, and press the Insert function key.

You can also type INSERT on the QMF command line, move the cursor to the line you want to precede the new line, and press Enter.

8. To delete lines from a procedure, move the cursor to the line you want to delete and press the Delete function key.

Alternatively, you can type DELETE on the QMF command line, move the cursor to the line you want to delete, and press Enter.

9. To save the procedure in the database, enter:

```
SAVE AS procname
```

In the procedure that is shown in the previous figure, the REXX DATE function provides the day of the week. The rest of the procedure includes QMF commands that are run depending on the day of the week.

### Related reference

[INTERACT](#)The INTERACT command enables user interaction while a procedure or application is running.

### Related information

[The IBM Publications Center](#)Search for information about REXX.

## Guidelines for writing procedures with logic

When you write procedures with logic, keep some specific guidelines in mind.

- A procedure with logic can contain QMF commands, REXX logic statements, and comment lines.

- Use the complete names for commands, options, and values, rather than the abbreviated names.
- Do not specify a command of more than 2,000 characters (or the equivalent in DBCS). QMF stops running a procedure when it finds a command of over 2,000 characters.
- You can include comments on the same line as a command, but place them after the command.
- If a command spans more than one line, type a comma as a continuation character at the end of the first line. For example:

```
"RUN QUERY MYQUERY (&&DEPT=38, ",
"&&DIV='EASTERN' "
```

Because this statement is a QMF command that is split into two lines, both lines are enclosed in quotes. The continuation character is placed at the end of the first line, outside the quotes.

You cannot use a continuation character in a comment line, command keyword, or substitution variable name. You can use a continuation character in a substitution variable value if you enclose the value in single quotation marks.

You can type comments between continuation lines.

If the object name is long enough to continue it on multiple lines, follow the REXX syntax rules for continuation and concatenation. Enclose each part of the object name in single quotation marks and use the concatenation symbol (||) at the beginning of each new line. The following figure shows an example of an object name that spans two lines in a procedure with logic:

```
/* REXX PROC */
"DISPLAY TABLE",
"DALLAS.MIRIAM_SMITH_AT_OFFFC1234.",
||"OVERSEAS_PARTS_LIST_ASIA_VIEW"
```

*Figure 253. Spanning an object name across multiple lines in a procedure with logic*

When you save a procedure, the procedure name might not be fully displayed at the top of the **PROC** panel if the authorization ID is longer than eight characters or the object name is longer than 18 characters. Enter the SHOW NAME command from the command line to display a pop-up screen that contains the complete object name.

#### **Related reference**

**SHOW**The SHOW command has many uses. For example, you can use the SHOW command to navigate among object panels and show a variation of the FORM.DETAIL panel.

## **Using REXX variables in procedures with logic**

You can use REXX variables in a procedure with logic. The values for these variables are known only within the procedure in which you defined them.

You can do the following:

- Copy a REXX variable to a QMF variable with the SET GLOBAL command
- Copy a global variable to a REXX variable with the GET GLOBAL command
- Use REXX variables in your REXX statements

QMF also provides a group of REXX variables for the callable interface that QMF sets after processing each QMF command. These variables provide important information about the results of each command. You can use them in your procedures with logic. For example, DSQ\_RETURN\_CODE is a return code of QMF, and DSQ\_MESSAGE\_ID is a completion message of QMF.

#### **Related reference**

Interface communications variables for REXXThe interface communications variables consist of several REXX variables.

GET GLOBALThe linear syntax of the GET GLOBAL command assigns values of QMF global variables to REXX variables in applications and procedures written in REXX.

**SET GLOBAL**The SET GLOBAL command assigns values to global variables from the QMF command line, from a procedure, or through the callable interface.

### Related information

[The IBM Publications Center](#) Search for information about REXX.

## Specifying REXX variables using SAY and PULL statements

In a procedure with logic, you can use REXX SAY and PULL statements to prompt for variable values.

Use a SAY statement (or a sequence of SAY statements) to display text on the screen. The following figure shows some sample SAY statements:

```
say 'Hello,' whoisuser'.'  
say 'Please enter the letter of the weekly report you would like, '  
say 'or NONE to exit:'  
say  
say '          A. Sales results (Monday Only)'  
say '          B. Tax figures'  
say '          C. Cumulative salaries'
```

Figure 254. SAY statements prompt users to enter text.

When you use these SAY statements, the output shown in the following figure appears:

```
Hello, username.  
Please enter the letter of the weekly report you would like,  
or NONE to exit:  
  
          A. Sales results (Monday Only)  
          B. Tax figures  
          C. Cumulative salaries
```

Figure 255. The user prompts appear on the screen.

Specify a REXX PULL statement to retrieve the input and place it in the REXX variable *answer* as shown in the following figure.

```
/* This procedure can produce any of three weekly reports  
regularly produced by the Acme Company (Sales Results,  
Tax Figures, or Cumulative Salaries). It prompts the user  
for the type of report wanted, runs the necessary  
queries, and checks for errors. */  
  
arg report . /* get any arguments from RUN PROC */  
ok = 'NO' /* set variable for do loop */  
"GET GLOBAL (WHOISUSER = DSQAO_CONNECT_ID" /* identify user */  
  
if report = '' then /* check to see if no arg entered */  
  
/* if no arg entered, prompt user until A,B,C, or NONE is entered */  
do until ok = 'YES'  
  
say 'Hello,' whoisuser'.'  
say 'Please enter the letter of the weekly report you would like, '  
say 'or None to exit:'  
say  
say '          A. Sales results (Monday Only)'  
say '          B. Tax figures'  
say '          C. Cumulative salaries'  
  
pull answer /* get answer from user */  
answer = strip(answer) /* strip any leading or trailing blanks */  
  
if answer = 'NONE' then exit 3 /* exit immediately if NONE */  
if pos(answer,'ABC') = 0 then ok = 'YES' /* if invalid value, */  
end /* keep prompting. */  
else answer = report
```

Figure 256. PULL statements retrieve user input.



An exit code of 3 was selected here to indicate the exit condition when the user enters None. As with any exit code, you choose the number to indicate an exit condition.

## Passing values to a procedure with logic

For procedures with logic, use the ARG option on the RUN PROC command to pass arguments, or values, to a procedure with logic. You can also use the ARG option to pass values between procedures.

Use the ARG option when you are running a procedure that contains a REXX PARSE ARG statement or ARG statement, as in the following figure.

```

PROC                WILDE.SHOW_ARGS                MODIFIED    LINE 1
/*****
/* This procedure shows you how to use the 'ARG=' option on the RUN    */
/* PROC command.                                                    */
/*****
parse upper arg query_name form_name
"RUN QUERY" query_name "(FORM="form_name

```

Figure 257. The ARG option passes values to a procedure with logic.

The RUN command for this procedure is:

```

RUN PROC SHOW_ARGS (ARG=(query_name form_name)

```

In this command, *query\_name* and *form\_name* are REXX variable names that describe the parameters being passed to the procedure with logic; they contain the object names for a query and a form, respectively. Using these variable names allows you to reference the parameters that were passed to the procedure with logic.

## Differences between REXX variables and substitution variables

A summary of the differences between REXX variables and substitution variables in procedures with logic.

The following table shows the differences between REXX variables and substitution variables. It also shows how each is used in a procedure with logic.

Table 44. REXX variables versus substitution variables in a procedure with logic	
REXX variables	Substitution variables
Name is made up of alphanumeric characters in lower case or upper case. For example: <i>what_2_do</i>	Name must begin with an ampersand (&), followed by alphanumeric or special characters. For example:  <div style="background-color: #f0f0f0; padding: 2px;">           &amp;DEPARTMENT         </div>
Can be used in REXX statements. For example:  <div style="background-color: #f0f0f0; padding: 2px;">           if progname = ' ' then         </div>	Can be used in QMF commands. For example:  <div style="background-color: #f0f0f0; padding: 2px;">           "RUN QUERY MYQUERY (FORM =            &amp;FORMNAME"         </div>
Can be given a value on the RUN PROC command using the QMF ARG parameter and the REXX ARG parameter. For example:  <div style="background-color: #f0f0f0; padding: 2px;">           RUN PROC MYPROC (ARG=MONDAY            arg whichday         </div>	Can be given a value on the RUN PROC command. For example:  <div style="background-color: #f0f0f0; padding: 2px;">           "RUN PROC MYPROC (&amp;&amp;FORMNAME =            MYFORM"         </div>

Table 44. REXX variables versus substitution variables in a procedure with logic (continued)

REXX variables	Substitution variables
<p>Can be given a value using a QMF global variable and the QMF GET GLOBAL command. For example:</p> <pre>"GET GLOBAL (WHO_IS_IT = DSQAO_CONNECT_ID"</pre>	<p>Automatically assigned a value by QMF at the time the command is run if a global variable by that name has been set (if the substitution variable has not already been given a value).</p>
<p>Can be used to set a global variable value using the QMF SET GLOBAL command. For example:</p> <pre>"SET GLOBAL (JOBTYPE =" JOBVAR</pre>	<p>Cannot be used to set a global variable value.</p>

### Related reference

**RUN**The RUN command runs queries or procedures from QMF temporary storage or from the database at the current location.

## REXX error-handling instructions in procedures with logic

You can use REXX error-handling techniques, such as the REXX SIGNAL instruction, in a procedure with logic. In addition, you can use QMF commands and variables with the REXX EXIT instruction to help clarify nonzero return codes.

### Branching to error-handling subroutines

The REXX *signal on error* instruction tells REXX to leave the current line and branch to a label marked **error** when a nonzero return code is encountered. This instruction requires two parts:

- Signal on error

After every command, REXX puts the return code of the command in a variable called `rc`.

If a command has a nonzero return code, REXX branches to the **error** label.

In TSO, signal on error returns errors from the QMF REXX procedure (ADDRESS QRW) command environment, but not the REXX callable interface.

- **Error label**

The signal on error instruction requires that you provide a label that the procedure can branch to if it encounters a nonzero return code. The label precedes your error-handling code. The return code is in the variable `rc`. You can use this variable to branch to another subroutine or you can use it in your EXIT instruction, as shown in the following figure.

```
/* error handling code for a procedure with logic */
error:
  exit rc
```

Figure 258. QMF exits with a nonzero return code.

### Using messages with the REXX EXIT instruction

You can use the REXX EXIT instruction to exit a procedure with logic. QMF always issues a message when it finishes running a procedure with logic. If you use the EXIT instruction, the message you see depends on these factors:

- Whether the last QMF command encountered an error
- Whether the return code was zero

The following table shows which message you see based on the conditions.

Table 45. Messages returned from QMF commands in procedures

Return code from last QMF command	Procedure return code	Message (or examples of messages) at completion of procedure
0	0	OK, your procedure was run
0	Nonzero	The return code from your procedure was 8
Nonzero	0	The error message provided by QMF
Nonzero	Nonzero	The error message provided by QMF

A QMF error message takes precedence over the return code message from the program if you have an incorrect QMF command and a nonzero return code.

If you want to show the error message from the last command and exit with a nonzero return code, use the MESSAGE command as in the following figure.

```
"MESSAGE (TEXT='dsq_message_text'"
exit rc
```

Figure 259. Specify MESSAGE to see the error message from the last command.

The variable dsq\_message\_text is a REXX variable that is provided by QMF. You can use the MESSAGE command and the dsq\_message\_text variable to store and display a message after further processing has occurred, as shown in the following figure.

```
/* Monthly report */
Signal on error
"DISPLAY TABLE JUNE_INFO"
"PRINT REPORT"
Exit(0);
Error:
Original_msg = dsq_message_text /* Saves error message. */
"RUN PROC GENERAL_RECOVERY" /* This proc generates */
/* new dsq_message_text. */
"MESSAGE (TEXT=' Original_msg '" /* Display original error msg. */
Exit(8);
```

Figure 260. The MESSAGE command displays the original error message.

### Related reference

**MESSAGE** Use the MESSAGE command from user applications (procedures, programs, execs, and CLISTs) to pass a message to the QMF message area.

## Calling programs from a procedure with logic

You might have procedures that call applications.

When you call your callable interface application from a procedure with logic, be careful about the number of ampersands you specify for the substitution variables in your application. This is important if the program being called contains a RUN command with substitution variables, as in RUN QUERY WEEKLY\_Q (&&DEPT=58.

### Calling programs without substitution variables

If your program does not contain a RUN command that includes substitution variables, use one of the following methods to start your program:

- The ADDRESS instruction

This instruction establishes a command environment. For example, if you want to call a program that is named PANDA from within the TSO environment, enter this command:

```
ADDRESS TSO "PANDA"
```

- The CALL instruction

This instruction starts a program. For the program named PANDA, the command is:

```
CALL PANDA
```

- A function

You also can call the program PANDA as a function:

```
ANSWER = PANDA()
```

You might consider removing the substitution variables from the RUN command if you want to call your programs by using one of the REXX invocation calls. In that case, QMF prompts the user for the variables.

## Calling programs that contain substitution variables

Whether you are running a procedure with logic or a callable interface program that is invoked by a procedure with logic, commands come into QMF the same way. In this context, the callable interface program becomes a logical extension of the procedure itself.

If your application contains QMF RUN commands with substitution variables, you have three choices for how to resolve those variables:

- Use the TSO command to call the application as follows:

```
TSO <program name>
```

When you call the application using this command, QMF does not process any substitution variables it encounters. For example, the substitution variable in the following RUN QUERY command embedded in the program is passed to the query rather than being interpreted as a variable of the program:

```
RUN QUERY WEEKLY_Q (&DEPT=58
```

- Treat all substitution variables in your application as though you were using them in a procedure with logic.

Add an extra ampersand to every substitution variable so the procedure with logic does not resolve it. For example, consider the following command:

```
RUN QUERY WEEKLY_Q (&&DEPT=58
```

In a procedure with logic, use two ampersands on the substitution variable to pass the variable to the query:

```
"RUN QUERY WEEKLY_Q (&&DEPT=58"
```

If a substitution variable has only one ampersand, QMF resolves the variable for the procedure itself, and cannot pass the variable to the query.

- Use global variables.

You can define global variables at the start of your application and use them throughout your QMF session.

### Related concepts

[ADDRESS QRW and the QMF command environment](#) When QMF is started in TSO, ISPF, or native z/OS, QMF creates a REXX command environment called QRW.

### Related reference

[SET GLOBAL](#)The SET GLOBAL command assigns values to global variables from the QMF command line, from a procedure, or through the callable interface.

### Related information

[The IBM Publications Center](#)Search for information about REXX.

## Running a procedure

---

The QMF commands that you specify in a linear procedure run in the order that they appear in the procedure. The QMF commands that you specify in a procedure with logic run in the order that is specified by the logic of the procedure.

### About this task

If the QMF commands in the procedure run a query or display a query or form, they change the contents of the temporary storage areas REPORT, FORM, or QUERY. This change occurs as if you entered each command separately on the QMF command line.

If an error occurs while a linear procedure is running, QMF stops running the procedure. The **PROC** panel displays the command that contains the error. The error message at the bottom of the screen provides information on correcting the error.

If an error occurs while a procedure with logic is running, the logic of the procedure determines when the procedure ends and what displays.

If ISPF is available on your system, you can use the QMF batch application to run the procedure while you are doing other work on your system.

If you are using QMF for TSO, you can also run the procedure by starting QMF as a stored procedure and passing the procedure name as a parameter on the CALL statement. The procedure that is passed to the interface can also start a batch job, which allows any Db2 client that can issue a CALL statement to access QMF for TSO batch services.

### Procedure

To run a procedure, enter the following on the QMF command line:

```
RUN PROC procname
```

### Related concepts

[REXX error-handling instructions in procedures with logic](#)

You can use REXX error-handling techniques, such as the REXX SIGNAL instruction, in a procedure with logic. In addition, you can use QMF commands and variables with the REXX EXIT instruction to help clarify nonzero return codes.

[Installing and managing QMF for TSO and CICST](#)The stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.

### Related reference

[BATCH](#)BATCH is a command synonym supplied by QMF that accesses the QMF batch application. This application lets you run queries and procedures as QMF batch jobs rather than interactively.

## Sharing a procedure with other QMF users

---

You can share a procedure with other QMF users just as you do other QMF objects, by saving it with the SHARE=YES parameter. Make sure that you also save any other QMF objects you specify in the procedure with the SHARE=YES parameter.

You can also set the DSQEC\_SHARE global variable to share objects with other users. To set this global variable to allow other users to share your objects, enter this command:

```
SET GLOBAL (DSQEC_SHARE=1
```

To check whether an object is shared, use the LIST command to display the object. Move the cursor to the object name and press the Describe function key. If the object is shared, the value in the **Restricted** field on the object description panel is No.

Make sure that you qualify the name of every shared object in a procedure with your user ID or the user ID of the person who owns it. Doing this ensures that other people use the correct procedure, in case they have procedures with identical names.

### Related concepts

[Displaying a list of database objects](#)

To quickly view information about database objects, you can display a list of those objects.

## Creating reusable procedures with substitution variables

You can use substitution variables in linear procedures and procedures with logic, just as you can in queries.

A substitution variable is any variable that you can use in a QMF command. A substitution variable is always preceded by an ampersand (&).

You can supply a value for a variable in the following ways:

- On the RUN command
- On a prompt panel
- On the SET GLOBAL command

## Specifying values for variables on the RUN command

This value for the substitution variable is active only within the procedure that defines it. The value is not active in any procedure or module called from the defining procedure.

### Procedure

You can assign a value to a substitution variable by using the RUN command:

- When running a linear procedure:

```
RUN PROC SCHEDULE (&&TYPE='VACATION'
```

- When running a procedure with logic:

```
"RUN PROC SCHEDULE (&&TYPE='VACATION' "
```

You enclose the variable value VACATION in single quotes because the value is a character string. Precede the variable with && to set the value on the RUN statement, or with & if the procedure prompts you for the value.

In the example above, the value of &&TYPE is available only to the procedure that is called SCHEDULE.

### Related concepts

[Specifying values for variables using global variables](#)

You can specify values for substitution variables by defining global variables with the SET GLOBAL command.

[Specifying values on the RUN command prompt panel](#)

If you run a procedure that contains a substitution variable and you do not assign a value to the variable using a global variable or the RUN command, QMF displays the RUN Command Prompt panel. You can specify the value for the variable on this panel.

## Specifying values for variables using global variables

You can specify values for substitution variables by defining global variables with the SET GLOBAL command.

For example, to set a global variable value for the &DEPARTMENT variable, enter the following command on the QMF command line:

```
SET GLOBAL (DEPARTMENT=38
```

You can assign values to up to 10 variables with a single SET GLOBAL command. Separate the variable assignments with commas or with blanks.

By default, values for global variables persist for the duration of the QMF session or until you reset them. However, the DSQEC\_USERGLV\_SAV global variable can be set to save global variable values from one session to another.

You can also use the SET GLOBAL command to prompt for all the values in your procedure at the same time, as in the following example, which is how the command would appear in a procedure with logic:

```
"SET GLOBAL (LASTNAME=&LASTNAME,DEPT_NUM=&DEPT_NUM";
```

### Related concepts

[Specifying values on the RUN command prompt panel](#)

If you run a procedure that contains a substitution variable and you do not assign a value to the variable using a global variable or the RUN command, QMF displays the RUN Command Prompt panel. You can specify the value for the variable on this panel.

### Related tasks

[Specifying values for variables on the RUN command](#)

This value for the substitution variable is active only within the procedure that defines it. The value is not active in any procedure or module called from the defining procedure.

## Specifying values on the RUN command prompt panel

If you run a procedure that contains a substitution variable and you do not assign a value to the variable using a global variable or the RUN command, QMF displays the RUN Command Prompt panel. You can specify the value for the variable on this panel.

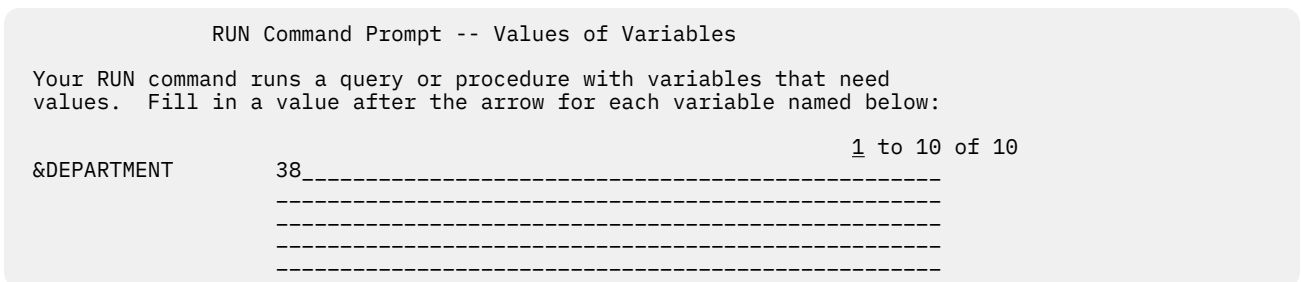


Figure 261. Entering a value for a substitution variable

This value for the substitution variable is active only within the procedure that defines it. The value is not active in any procedure or module called from the defining procedure.

In a linear procedure, QMF scans the procedure for substitution variables and resolves them before it processes any commands. QMF prompts you for all variable values before the procedure runs.

In a procedure with logic, QMF does not prompt you for variable values until REXX encounters the statement that contains the variables. For example, if your procedure with logic includes three statements that contain variables that QMF must prompt you for, QMF prompts you three times – once for each statement.

If you want a procedure with logic to prompt you for all the necessary variable values at one time, as the linear procedure does, use a dummy procedure. Suppose you want to be prompted once for the substitution variables LASTNAME and DEPT\_NUM, which occur on two different lines in your procedure with logic as shown in the following figure.

```
/* This procedure runs two queries, displaying the report after each */
/* procedure has run. */
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";
"INTERACT"
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

*Figure 262. This procedure requires two substitution variables.*

Add the following line to the beginning of your procedure with logic, immediately following the comment lines. (All procedures with logic must begin with at least one comment.)

```
"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
```

In this line, PROMPT\_ME is a procedure with logic that contains a comment line and no instructions. The PROMPT\_ME procedure is shown in the following figure.

```
/* This procedure is a dummy procedure that provides prompting */
```

*Figure 263. This procedure prompts you to enter substitution variables.*

The following figure shows the completed procedure with the RUN PROC command for the PROMPT\_ME procedure that prompts for variables:

```
/* This procedure runs two queries, displaying the report after each */
/* procedure has run */
"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";
"INTERACT"
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

*Figure 264. Procedure with logic that prompts for variables*

### **Related concepts**

[Specifying values for variables using global variables](#)

You can specify values for substitution variables by defining global variables with the SET GLOBAL command.

### **Related tasks**

[Specifying values for variables on the RUN command](#)

This value for the substitution variable is active only within the procedure that defines it. The value is not active in any procedure or module called from the defining procedure.

## **Connecting to a remote location from a procedure**

The QMF CONNECT command lets you connect to a remote database.

You can use this command within a linear procedure or a procedure with logic, except when the procedure has been passed as a parameter on a CALL statement that starts QMF as a stored procedure.

In the following example, suppose that you are an administrator in Miami, and you want to write a procedure that:



1. Connects to a remote location (DALLAS)
2. Issues a series of QMF commands
3. Prints a report
4. Reconnects to the originating location (MIAMI)

The procedure looks like the one that is shown in the following figure.

```
CONNECT TO DALLAS          -- Connect to a database named DALLAS
RUN PROC GENERATE_REPORT (FORM=GEN_FORM  -- Issue QMF commands
PRINT REPORT              -- Print report on printer whose name is specified in QMF
profile
CONNECT TO MIAMI         -- Connect to a database named MIAMI
```

Figure 265. This procedure uses the `CONNECT` command.

Make sure that you store the procedure at the remote location in the same database to which you are connected when you issue the `RUN PROC` command. When you connect to a new location, QMF re-initializes your profile, except for the value of `TRACE`. It also re-initializes command synonyms and function keys to the values at the new (remote) location.

When you write procedures that use the QMF `CONNECT` command to access remote databases, keep the following guidelines in mind:

- If you are connected to a remote database and issue a `RUN PROC` command, that procedure and all the objects used in that procedure must be stored at the remote database.
- All QMF commands in the procedure run in QMF temporary storage at the system where QMF is running (the local system). However, all objects used by these QMF commands (such as queries, procedures, or forms) must be defined in the database at the remote location.
- All commands that affect the database (for example, SQL statements, QMF queries, or `EDIT TABLE` updates) run at the remote location.
- If the procedure contains system-specific commands (CICS or TSO), these commands run at the system where QMF is running (the local system).

If your procedures contain system-specific commands that cannot run on the system where QMF runs, your procedure will not run successfully.

- Any files, data sets, or data queues that are used in a system-specific command must exist on the system where QMF is running (the local system).
- If your site uses TSO and takes advantage of RACF® support for mixed-case passwords, ensure that the `CASE` option of your QMF profile is set to `MIXED`. Otherwise, QMF converts all input to upper case, causing the `CONNECT` command to fail. In this situation, ensure that you enter all input in upper case, because QMF only recognizes commands in upper case.

### Related concepts

#### The QMF `CONNECT` command and access data at a remote database

When you are connected to a remote database, you access data and objects in the same way that you access them at a local database. QMF continues to use programs that reside on the local system.

Installing and managing QMF for TSO and CICSThe stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.

### Related tasks

#### Accessing data at a remote database

When you start QMF, the system from which you start the program is known as the *local* system. The Db2 database that resides on this system is known as the *local* database. Databases that do not reside on this system are known as *remote* databases, or remote servers.

### Related reference

`CONNECT` in CICSWith the `CONNECT` command, you can connect to any database server that is part of the distributed network from within a QMF session.

[CONNECT in TSO](#) You can use the CONNECT command from within a QMF session to connect to any database server that is part of the distributed network.

## Writing a procedure that runs a generic query

The example in this topic shows you how to write a procedure with logic that uses a generic query that contains substitution variables for the columns in the SELECT statement as well as the conditions in the WHERE clause.

The sample procedure:

- Checks the day of the week
- Sets the values of the variables passed to the query if the day is Friday
- Runs the query

## Writing a generic SQL statement

You can write a generic SQL statement that can accept different values for the column names and row conditions.

For this scenario, create the following query and save it as SENIORSTAFF:

```
SELECT &SELECT1
FROM Q.STAFF
WHERE &COND1
```

This query allows the user or a procedure to specify the column names and row conditions just before running the query.

Using a procedure, you can assign values to the QMF query substitution variables (&SELECT1 and &COND1) by using one of the following methods:

- Pass the substitution variable values to the query on the RUN QUERY command.
- Set global variable values.

Both of the procedures that are described in this scenario produce the same results.

### Related concepts

#### [Passing variables to the generic query](#)

You can write a procedure that sets REXX variable values and passes these values to your generic SQL statement.

#### [Writing a procedure that sets global variables for the generic query](#)

You can write a procedure that sets global variable values according to REXX logic. These values are then available to the generic query when the procedure issues the QMF RUN QUERY command.

## Passing variables to the generic query

You can write a procedure that sets REXX variable values and passes these values to your generic SQL statement.

The QMF procedure in the following figure passes the substitution variable values to the query on the RUN QUERY command.

```

/* REXX PROC */
IF DATE('W') = 'Friday' THEN
  DO
    sel = '(NAME, JOB, SALARY, COMM)'
    con1 = '((SALARY > 15000) OR (JOB = 'MGR'))'
  END
ELSE
  DO
    sel = '*'
    con1 = '(DEPT=51)'
  END

"RUN QUERY SENIORSTAFF (&SELECT1 ="sel", &COND1 ="con1

```

Figure 266. The procedure passes values on the RUN QUERY command.

Because this procedure assigns values to the substitution variables (SELECT1 and COND1) on the RUN QUERY command, you must use a double ampersand before the variable names to tell REXX that these variables are assigned in the procedure, but not used in the procedure.

If you use only one ampersand before the variable name, as in the following statement, QMF assumes that the variables are procedure variables, rather than variables to be passed to the query, and prompts you for their values when you run the procedure:

```
"RUN QUERY (&SELECT1 ="sel",&COND1 ="con1
```

In the following lines from this procedure, the procedure assigns a character string to a REXX variable:

```

con1 = '((SALARY > 15000) OR (JOB = 'MGR'))'
con1 = '(DEPT=51)'

```

These values are then passed to the query on the RUN QUERY command. The values in the first REXX variable assignment, SALARY and JOB, are enclosed in double parentheses because the character strings passed to the query contain single parentheses and an equal sign.

When you run this procedure on a Friday, the procedure sets the substitution variables and passes the values to the query so that QMF runs the following query:

```

SELECT NAME, JOB, SALARY, COMM
FROM Q.STAFF
WHERE (SALARY > 15000) OR (JOB='MGR')

```

If you run this procedure on any day other than Friday, QMF runs the following query:

```

SELECT *
FROM Q.STAFF
WHERE DEPT = 51

```

### Related reference

**RUN**The RUN command runs queries or procedures from QMF temporary storage or from the database at the current location.

## Writing a procedure that sets global variables for the generic query

You can write a procedure that sets global variable values according to REXX logic. These values are then available to the generic query when the procedure issues the QMF RUN QUERY command.

The following procedure uses the SET GLOBAL command to set values for the variables in the query. The results are the same as those explained in [“Passing variables to the generic query”](#) on page 254.

```

/* REXX PROC */
IF DATE('W') = 'Friday' THEN
DO
  "SET GLOBAL (SELECT1 = 'NAME, JOB, SALARY, COMM'"
  "SET GLOBAL (COND1 = '(SALARY > 15000) OR (JOB = 'MGR')'"
END
ELSE
DO
  "SET GLOBAL (SELECT1 = '*'"
  "SET GLOBAL (COND1 = '(DEPT = 51)'"
END
"RUN QUERY SENIORSTAFF"

```

Figure 267. The procedure sets query variables as global variables.

## Writing procedures that can run unattended

In QMF batch mode, you can run both linear procedures and procedures with logic while you perform other work on your system.

You can run procedures in batch mode at any time, and you do not have to interact with QMF while the procedure is running. ISPF is required to use the QMF BATCH application. ISPF is not available in CICS.

To run a procedure in batch mode, first create and save the procedure, just as you would to run it interactively. Then use the QMF batch application, which simplifies batch processing. The application prepares and submits the batch job from information that you enter on the BATCH Command Prompt panel. You only need to know the name of the procedure and a few details about the batch machine on your system. However, it still might be necessary to tailor the application to your needs.

If you are using QMF for TSO, you can also run the procedure using the QMF stored procedure interface. This interface allows any software program that can call a stored procedure to start QMF as a Db2 for z/OS stored procedure and run a QMF query or procedure. The procedure that is passed to the interface can also start a batch job, which allows any Db2 client that can issue a CALL statement to access QMF for TSO batch services. If you run a procedure, commands that display panels or require user interaction are not allowed.

The following terms are used in these guidelines for writing a procedure that will be used in batch mode, start a batch job through the stored procedure interface, or will be run through the QMF stored procedure interface.

- The *main procedure* is the procedure that is identified on the CALL statement or the ISPSTART command.
- A *subordinate procedure* is any procedure that is called directly from the main procedure or from another subordinate procedure.

### Related concepts

[Installing and managing QMF for TSO and CICS](#)The stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.

## Inhibiting the display of panels

QMF cannot display panels of any kind when running in batch mode or as a Db2 for z/OS stored procedure under TSO. Thus, any procedures that include commands that display panels will not run.

To avoid errors when you run your procedures, ensure that you:

- Do not write incomplete commands.
  - QMF has no way of prompting you for information to complete the command.
- Do not try to directly access command prompt panels.
  - Do not issue commands that use the question mark to obtain the command prompt panels.
- Turn off the confirmation panel display.

You can issue commands that erase or change data objects, but you must inhibit the confirmation prompt.

To inhibit the confirmation panel, include the CONFIRM=NO option on the command or issue the following command at the beginning of the procedure:

```
SET PROFILE (CONFIRM=NO
```

- Avoid situations that might display the Incomplete Data prompt.
- Share the main procedure by specifying SHARE=YES when you save it.

If you have issued a SET GLOBAL command with the value DSQEC\_SHARE=1 prior to this SAVE command, you do not need to specify the SHARE=YES parameter.

## Using IMPORT or EXPORT commands

When you export an object and later import it, refer to the data set name consistently. Always refer to the data set using the unqualified or fully-qualified name. Otherwise, problems can arise.

### Related reference

[EXPORT in CICS](#) Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

[EXPORT in TSO](#) The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

[IMPORT in CICS](#) The IMPORT command copies the contents of a CICS data queue into QMF temporary storage or into the database.

[IMPORT in TSO](#) The IMPORT command copies the contents of a TSO data set or UNIX file into QMF temporary storage or into the database.

## Using the EXIT command

QMF stops after the EXIT command runs.

A procedure also stops after it runs the command on its bottom line. If this command is not EXIT, one of the following happens:

- For a subordinate procedure, control is returned to the calling procedure without ending QMF. (This is also true in interactive mode.)
- For a main procedure, QMF ends.

Ending a main procedure that is running in batch mode or through the QMF for TSO stored procedure interface always ends QMF. This is why the sample procedure shown in [“Example of a procedure that requires no user interaction”](#) on page 258 does not need the EXIT command.

### Related reference

[EXIT](#) The EXIT command stops your QMF session.

## Writing procedures for a National Language Feature

If you are writing the procedure for a National Language Feature, the verbs and keywords in the commands must be the translated versions of their English-language counterparts: for example, a German batch-mode procedure would use the word ANZEIGEN for the DISPLAY command and PROZEDUR for PROC.

## Error handling

Any error encountered while running a linear procedure ends the procedure.

### Related concepts

[REXX error-handling instructions in procedures with logic](#)

You can use REXX error-handling techniques, such as the REXX SIGNAL instruction, in a procedure with logic. In addition, you can use QMF commands and variables with the REXX EXIT instruction to help clarify nonzero return codes.

## Example of a procedure that requires no user interaction

This example procedure passes a value of a variable to a query, prints a report of the results and saves the data in a table. Finally, the main procedure runs a procedure against the table that was just created.

The following main procedure illustrates how to write a successful procedure that can run in batch mode or through the QMF for TSO stored procedure interface, or start a batch job through the stored procedure interface:

```
SET PROFILE (CONFIRM=NO
RUN QUERYA (&&LICENSE='007'
PRINT REPORT (PRINTER='
SAVE DATA AS TABLEA
RUN PROCA (&&TABLE=TABLEA
EXIT
```

### **SET PROFILE (CONFIRM=NO**

Eliminates the possible display of confirmation panels.

### **RUN QUERYA**

Passes the value 007 to QUERYA for the substitution variable &LICENSE.

The object names in this command are not qualified with the owner's name. Their owner is, therefore, the current QMF user.

### **PRINT**

Prints a report based on the query results; the output goes to the DSQPRINT data set.

### **SAVE**

Saves the data in the database.

The SAVE command does not need to contain CONFIRM=NO because of the SET PROFILE command at the start of the procedure. However, if the DATA object is too large for the storage that is reserved for it, the SAVE command might end the procedure because of an incomplete-data condition.

### **RUN PROCA**

Runs a procedure that does something with TABLEA (the table that was just created or replaced by the SAVE command).

The name of this table is passed to the procedure through the &TABLE variable. This command fails if the procedure called contains other substitution variables that have not been set.

### **EXIT**

Ends the procedure and exits QMF.

Depending on the commands you use in your procedures, there might be some restrictions or special considerations you need to know about.

### **Related reference**

[QMF commands](#) Look up syntax, option descriptions, and usage information for commands used with QMF.

## Chapter 10. Reusing QMF variables and input values

Global variables can be used for multiple purposes such as providing values for substitution variables or saving user input within and across QMF sessions. You can also use global variables to change certain behavioral aspects of your QMF session, such as displaying confirmation panels in the Table Editor.

By default, values for global variables persist for the duration of the QMF session or until you reset them. However, the DSQEC\_USERGLV\_SAV global variable can be set to save global variable values from one session to another.

When you write queries or procedures or use QMF forms, you can include substitution variables that make these objects generic enough to be used with different sets of data. If you use substitution variables in QMF commands such as procedures and queries, you can either assign the values to these variables at run time or you can set a global variable that then provides the value.

### The global variable list

One way to display, change, add, or delete global variables is to use the SHOW GLOBALS command.

When you enter SHOW GLOBALS on the QMF command line, QMF displays a global variable list panel similar to the one shown in the following figure.

```
GLOBALS

Type a value for a global variable and press Enter or press a function
key. Variable values may be changed if they are enclosed in parentheses
or brackets.

Variable Name:      Value:
-----
                                  1 to 11 of 97
DXYEPRST_01        ( GLOBAL                                )
EMPLOYEE_NAME      ( SANDERS                                )
LOCATION_LIST        ( 'NEW YORK', 'BOSTON', 'WASHINGTON', 'ATLANTA', >
MAXIMUM_SALARY     ( 18999                                  )
MINIMUM_SALARY     ( 17000                                  )
TABLE_NAME         ( Q.STAFF                                )
DSQAO_APPL_TRACE   0
DSQAO_ATTENTION    0
DSQAO_BATCH        1
DSQAO_CICS_SQNAME
DSQAO_CICS_SQTYPE
DSQAO_CICS_TQNAME
1=Help            2=
3=End             4=
5=Show Field     6=Query
7=Backward       8=Forward    9=Form    10=Add    11=Delete  12=Report
COMMAND ==>
```

Figure 268. The GLOBALS panel

The global variable list panel uses one row on the screen for each global variable. The variable name appears on the left, and up to the first 50 characters of the variable value appear on the right. Variable values that you can change appear within parentheses. Otherwise, the value is read-only. Variables that you or an administrator defined and session variables appear in alphabetical order first. Then, predefined QMF variables, which start with "DSQ", follow in alphabetical order.

Global variables added on the GLOBALS panel can have a length of up to 32,768 characters. Variable values that are longer than a single line are indicated by a greater-than (>) sign to the right of the value.

#### Related reference

[QMF global variables](#) QMF provides many global variables that help you control aspects of your QMF session, QMF commands, and panel display. The global variables also help you control behavior of QMF functions in procedures and applications.

## Changing a global variable value

---

Global variable values that you can change appear within parentheses on the GLOBALS panel. If a value is not enclosed in parentheses, it is read-only.

### About this task

Some variable names are reserved for use by QMF. These names begin with the letters DSQ. Some DSQ variables have a restricted set of acceptable values. For example, the variable DSQDC\_COST\_EST (which controls the display of the database cost estimate) must have a value of either 0 or 1.

### Procedure

To change a variable value, use one of the following options:

- On the QMF command line, enter the SET GLOBAL command with the following syntax:

```
SET GLOBAL (variable_name = value)
```

For example, to set a new global variable called JOBTYP to the value SECRETARY, enter the following command:

```
SET GLOBAL (JOBTYP='SECRETARY')
```

You can also set the value of a global variable by copying a value from another global variable with the following syntax:

```
SET GLOBAL (variable_name = &variable_name)
```

For the SET GLOBAL command, the maximum length of the value is 55 characters, and you can create or change up to ten global variables. For more information on the SET GLOBAL command, see QMF Reference.

- On the QMF command line, enter the SHOW GLOBALS command to display the GLOBALS panel. To change a global variable value, type the new value over the displayed value and press Enter.

If the variable value is too long to be displayed completely, a greater-than sign (>) displays in the right margin. To change or assign a value that is longer than 50 bytes, use the **Show Global Variable** panel as follows:

- a. Move the cursor to the line that contains the variable name.
- b. Press the **Show Field** function key, which displays the **Show Global Variable** panel. The entire variable value appears in a scrollable area.
- c. Type the new value over the displayed value and press Enter.

### Related reference

[QMF global variables](#) QMF provides many global variables that help you control aspects of your QMF session, QMF commands, and panel display. The global variables also help you control behavior of QMF functions in procedures and applications.

## Adding a global variable

---

You can add and new global variables with names that do not begin with "DSQ."

### Procedure

To add a global variable, use one of the following options:



- On the QMF command line, enter the SET GLOBAL command with the following syntax:

```
SET GLOBAL (variable_name = value)
```

For example, to create a new global variable called JOBTYPY with a value of SECRETARY, enter the following command:

```
SET GLOBAL (JOBTYPY='SECRETARY')
```

You can create a global variable by copying a value from another global variable with the following syntax:

```
SET GLOBAL (variable_name = &variable_name)
```

For the SET GLOBAL command, the maximum length of the value is 55 characters, and you can create up to ten global variables. For more information on the SET GLOBAL command, see QMF Reference.

- On the QMF command line, enter the SHOW GLOBALS command to display the GLOBALS panel.  
To add a variable, press the Add function key to display the **Add Global Variable** panel. Type the variable name and value and press Enter.

## Deleting a global variable

---

Deleting a global variable deletes both the name and value of the variable. You can delete variables that were created by an administrator or user, which have names that do not begin with "DSQ."

### Procedure

To delete a global variable, use one of the following options:

- On the QMF command line, enter the RESET GLOBAL command with the following syntax:

```
RESET GLOBAL (variable_name)
```

For example, to delete a global variable called JOBTYPY, enter the following command:

```
RESET GLOBAL (JOBTYPY)
```

You can delete all global variables that were created by an administrator or user by entering the following command:

```
RESET GLOBAL ALL
```

- On the QMF command line, enter the SHOW GLOBALS command to display the GLOBALS panel. To delete a global variable from the global variable list, move the cursor to the line that contains the variable name and press the Delete function key.

## Using the global variable list and the CASE option

---

When you change or add a global variable from the global variable list, the name and value might be changed to upper case, depending on the CASE option in your QMF profile.

The variable name is changed to upper case if the CASE option is set to UPPER or STRING. The variable value is changed only if you enter it in upper case (or the CASE option of the profile is set to UPPER).

## Saving global variables across QMF sessions

---

By default, global variables that are set during a QMF session are retained only until you reset the values or end the session. However, you can use the DSQEC\_USERGLV\_SAV global variable to save the values of your global variables when you exit QMF and restore those values when you start QMF again.

### About this task

DSQEC\_USERGLV\_SAV applies to global variables that were added or modified during your session, including global variables that start with “DSQ.”

### Procedure

Use the SET GLOBAL command to set the value of DSQEC\_USERGLV\_SAV to 2.

## Retaining panel input values

---

By default, command prompt panels display input fields either with no value or with a predefined QMF value. To save values that you enter on a panel and have your values displayed the next time that you access the same panel, you can set the DSQEC\_SESSGLV\_SAV global variable.

### About this task

Your input values can be retained for the remainder of the current session or for the remainder of the current session and subsequent sessions. In either case, your input values are saved as session variables, which have a prefix of DXY, in the Q.GLOBAL\_VARS table. Like other variables in the Q.GLOBAL\_VARS table, these session variables can be changed and deleted. Modify these variables with caution because doing so changes the values that are generated on command prompt panels.

Some input values on some panels are not saved, regardless of the DSQEC\_SESSGLV\_SAV setting. For information about the panels and fields that DSQEC\_SESSGLV\_SAV applies to, see QMF Reference.

### Procedure

Use the SET GLOBAL command to set the value of DSQEC\_SESSGLV\_SAV as follows:

- To save your input values for the remainder of the current session, set DSQEC\_SESSGLV\_SAV to 1.
- To save your input values for the remainder of the current session and for the next session, set DSQEC\_SESSGLV\_SAV to 2.

---

# Chapter 11. Creating tables

You can create tables in the database by using SQL statements on the SQL Query panels.

The syntax of the SQL statements that are shown might vary slightly depending on what database management system you use.

You can create tables only in the database to which you are currently connected. To create tables at a remote location, use the CONNECT command. The remote location becomes the current location, from which you can create tables.

## Related reference

[CONNECT in CICS](#) With the CONNECT command, you can connect to any database server that is part of the distributed network from within a QMF session.

[CONNECT in TSO](#) You can use the CONNECT command from within a QMF session to connect to any database server that is part of the distributed network.

## Related information

[The IBM Publications Center](#) To find information about the syntax of SQL statements, see the SQL reference information for the database you are using.

---

## Planning for your table

Make sure that you are authorized to create tables.

Before you create a table, you need the following information:

- The storage space for the table (called *table space name* in Db2 for z/OS and *dbspace name* in DB2 Server for VSE and VM) in which you can save your tables; check with your administrator for this information.
- The name of the table.
- The columns you want to include in the table, and the data type of each column.

---

## Creating a table

You can create a table by issuing the SQL CREATE TABLE statement.

### Procedure

- Issue a SQL CREATE TABLE statement to create a table:

```
CREATE TABLE tablename
  (columnname datatype definition,
   columnname datatype definition)
  IN spacename
```

In this statement:

- *tablename* is the name of the table.
- *columnname* is the name of a column.
- *datatype* is the data type of the data you are using in that column.
- *definition* (optional) describes whether the column can contain nulls.
- *spacename* is the dbspace name (DB2 Server for VSE and VM) or table space name (Db2 for z/OS and Db2 for Linux, UNIX, and Windows) where the table is stored.

## Example

The example in the following figure shows a table for an appointment calendar. The table name is CALENDAR. There are columns for the month, day, time, location, and reason for the event.

MONTH	DAY	TIME	LOCATION	REASON
5	24	15.30	BIG CONF. RM.	ANNE'S BIRTHDAY PARTY
5	25	10.45	BRIEFING CTR.	SALES CAMPAIGN KICK-OFF

Figure 269. This table contains data for an appointment calendar.

There are often several ways to specify columns and data types for a table. In this example, you can combine the MONTH and DAY columns into one column and use the DATE data type. You can also use the TIME data type for the time column.

If you use DATE and TIME data types, your CREATE TABLE statement looks like the one that is shown here:

```
CREATE TABLE CALENDAR
(CALDATE DATE NOT NULL,
 TIME TIME NOT NULL,
 LOCATION VARCHAR(15) NOT NULL,
 REASON VARCHAR(36))
IN spacename
```

Specifying NOT NULL prevents you from entering an appointment without a MONTH, DAY, TIME, and LOCATION. Specify a data type (character, numeric, or date/time) for each column. You must specify spacename.

### Related information

The IBM Publications CenterSearch for information about data types by referring to the SQL reference information provided for your database.

## Saving and appending to a table

You can save a table or append the table to another table with the SAVE DATA AS command.

### About this task

When you append new rows to an existing table and the corresponding columns do not have the same data type or length, QMF might automatically convert the data from one data type to another to save the data. Whether QMF converts the data depends on whether the database to which the command is directed supports this type of conversion.

If you replace or append data in an existing temporal table, the table remains temporal. However, you cannot save data into a new temporal table. If the specified table does not exist, a new table is created using the column names and labels in the QMF data object. When you save data into a new or existing table, new values are created for columns that were defined with the GENERATED ALWAYS attribute.

### Procedure

Choose one of the following options, depending on whether you want to save a table in the database or append a table to an existing table.

- To save your table in the database, enter:

```
SAVE DATA AS tablename
```

- If you want to append the table to an existing table, enter:

```
SAVE DATA AS tablename (ACTION=APPEND
```

Where *tablename* is the name of the table to which you want to append the new table.

For example, to append a table that is called NEWAPPTS to the existing table CALENDAR, enter:

```
DISPLAY TABLE NEWAPPTS
SAVE DATA AS CALENDAR (ACTION=APPEND)
```

### Related reference

**SAVE**The SAVE command saves in the database at the current location objects that are currently in QMF™ temporary storage.

## Creating a copy of a table

---

You can create a table by copying the data from an existing table into a new table.

### Procedure

To create a copy of a table:

1. Enter `DISPLAY TABLE tablename`, where *tablename* is the name of the table you want to copy.

For example, if you want to create a new table from a copy of the CALENDAR table, enter `DISPLAY TABLE CALENDAR`.

The table you want to copy displays.

2. Enter `SAVE DATA AS tablename`, where *tablename* is the name of the new table.

For example, enter `SAVE DATA AS MYCAL` to name the new table MYCAL.

QMF creates a new table with the same data as the old table. In this example, MYCAL and CALENDAR both exist in the database with the same data.

### Related reference

**CREATE TABLE**The CREATE TABLE statement defines a table. You provide the name of the table and the names and attributes of its columns. You can grant or revoke authorization for other people to use a table you created.

## Creating a view of a table

---

You can create a view that includes some or all of the columns in one or more tables. You can use a view just as you would a table. When you update the table (or tables) you used to create the view, the view is also updated. When you update the view, the tables are updated.

### About this task

You might find it useful to create a view of a table when you want a portion of a table to remain hidden. Creating a view from portions of several tables can simplify query development because you only have to specify that view instead of selecting several tables and joining them.

### Procedure

To create a view of a table, issue the CREATE command with the VIEW keyword.

The following example shows how to create a view of the CALENDAR table, called MYCAL, with the REASON column omitted.

```
CREATE VIEW MYCAL
(CALDATE, TIME, LOCATION)
AS SELECT CALDATE, TIME, LOCATION
FROM CALENDAR
```

## Synonyms for tables or views

---

You can create a synonym for the name of a table or view by using the CREATE SYNONYM statement.

When you refer to that table or view, you do not have to specify the fully qualified table name.

For example, to create a synonym for the CALENDAR table, enter:

```
CREATE SYNONYM CALEN FOR CALENDAR
```

Depending on your database configuration, you might need to specify an owner qualifier when you specify the table. In that case, enter:

```
CREATE SYNONYM CALEN FOR userid.CALENDAR
```

Now you can use the synonym instead of the table name. For example, you can specify CALEN instead of CALENDAR in all commands that refer to the CALENDAR table.

If you share a query that uses a synonym, the users you share it with must define the same synonym before they can run the query.

## Aliases for tables or views

---

If you have the CREATEALIAS privilege or SYSADM or SYSCTRL authority, you can create an alias for the name of a table or view by using the CREATE ALIAS statement.

For example, to create an alias for the CALENDAR table, enter:

```
CREATE ALIAS CALEN FOR CALENDAR
```

Depending on your database configuration, you might need to specify an owner qualifier when you specify the table. In that case, enter:

```
CREATE ALIAS CALEN FOR userid.CALENDAR
```

You can use an alias in the same way you use a synonym. The difference between a synonym and an alias, however, is that a synonym can be used only by its owner, and an alias can be used by its owner and other users.

If you share a query that uses an alias, the users you share it with do not have to define the same alias before they can run the query.

## Deletion of tables, views, synonyms, and aliases

---

You can use either the QMF ERASE command or the SQL DROP statement to erase tables, views, synonyms, and aliases from the database.

For example, to use the QMF ERASE command to delete the CALENDAR table, enter:

```
ERASE TABLE CALENDAR
```

To use the SQL DROP statement to erase the same table, run this query:

```
DROP TABLE CALENDAR
```

When you use either the DROP statement or the ERASE command to delete a table from the database, any views or synonyms you created from them are also dropped.

The privileges or authorities that are required for the ERASE command depend on the type of object you are trying to erase:

- To erase a table, you must be the owner of the table or have DBADM authority.
- To erase a view or an alias, you must be the owner or have SYSADM or SYSCTRL authority.

---

## Chapter 12. Maintaining the data in your tables

After you create your tables, you can add to or change the data in them.

QMF for CICS and QMF for TSO provide two ways for you to easily maintain the data in your tables:

- The Table Editor provides an interactive interface for maintaining your tables that steps you through the process of changing the data.
- If you are familiar with Structured Query Language (SQL), QMF provides an SQL Query panel for you to write and run SQL statements.

---

### Maintaining data with the Table Editor

By using the QMF Table Editor, you can easily make updates to the information in your tables.

The Table Editor operates on all tables except tables that contain columns that are defined as BINARY, VARBINARY, BLOB, or XML. If a table contains CLOB or DBCLOB columns, you cannot edit those columns, but you can edit other columns in the table. To edit a table that contains DECFLOAT data, the processor on which QMF is running must support decimal floating point instructions.

### Deciding when to save your data

When you start a Table Editor session, you might be able to override the default setting for when to save additions or changes to your data. You specify when to save your data by using the SAVE keyword when you type the EDIT TABLE command that begins a Table Editor session.

If you want to save additions or changes as you make them, use SAVE=IMMEDIATE. This option is only available if your database management system supports CURSOR HOLD. See your administrator to find out whether you can use the SAVE=IMMEDIATE option.

If you want to save additions or changes only at the end of the Table Editor session, use SAVE=END. Because SAVE=END is the default for the EDIT TABLE command, you do not have to type anything on the EDIT TABLE command.

If you specify that you want to see confirmation panels, those confirmation panels differ based on when the data is saved. You specify that you want to see confirmation panels (CONFIRM=YES) either in your QMF user profile or at the beginning of a Table Editor session.

### New table rows using the Table Editor

You can use the Table Editor to add rows to a table.

#### Adding rows to a table

When you are in ADD mode in the Table Editor you can add rows to a table.

#### Before you begin

Make sure the text case (UPPER, STRING, or MIXED) that you specify with the CASE parameter in your QMF profile is the same for this session as the text in the table. A MIXED value for the CASE parameter might have implications in TSO for the QMF CONNECT command.

#### Procedure

To add rows to a table by using the Table Editor:

1. Do one of the following tasks, depending on where you are starting:

- From the QMF home panel, type:

```
tablename (MODE=ADD
```

Then, press the Edit Table function key.

- From any other QMF panel with a command line, enter:

```
EDIT TABLE tablename (MODE=ADD
```

For example, to add rows to the PERS table from the QMF home panel, type PERS (MODE=ADD, and press the Edit Table key.

To save each addition as you make it, enter PERS (MODE=ADD SAVE=IMMEDIATE.

The Table Editor **ADD** panel displays as shown in the following figure. The panel shows the name of each column in the table, followed by an entry field where you enter new data for that column.

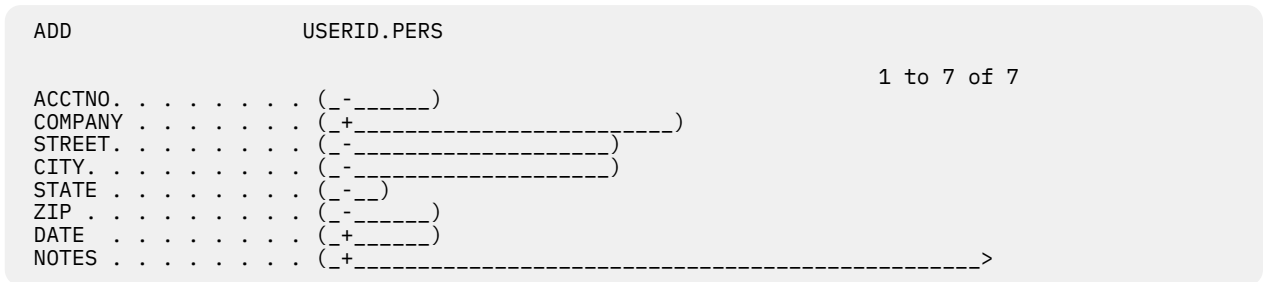


Figure 270. The Table Editor **ADD** panel

On this panel:

- The name of the table you are editing and the user ID of the table owner appear at the top of the panel.
- The columns that are displayed on this panel make up one row in the table.
- A null indicator (which is not the same as zero or blank) or a column default indicator (if available for the column) is displayed in each field to indicate that nothing was entered. These indicators are configurable.

A blank or a zero means that the column is preset to a blank or zero value.

- The scroll indicator indicates how many columns are in a row, and how many columns appear on the panel.

To move a specific column to the top of the panel, type its number in the first position of the scroll indicator. Press the Forward key to see the rest of the columns. Table Editor panels do not have a command line, so press the appropriate function key for the command you want to issue.

You can display the default settings for the fields by using the Show Field function key. Seeing the default settings is helpful when you type over the default values and forget what they were.

2. Type the information for the fields in the row. For this example, type the information in each field as it appears in the following figure.

Use the Tab key to move from field to field.

To see what values are valid for a field, press the Show Field function key.



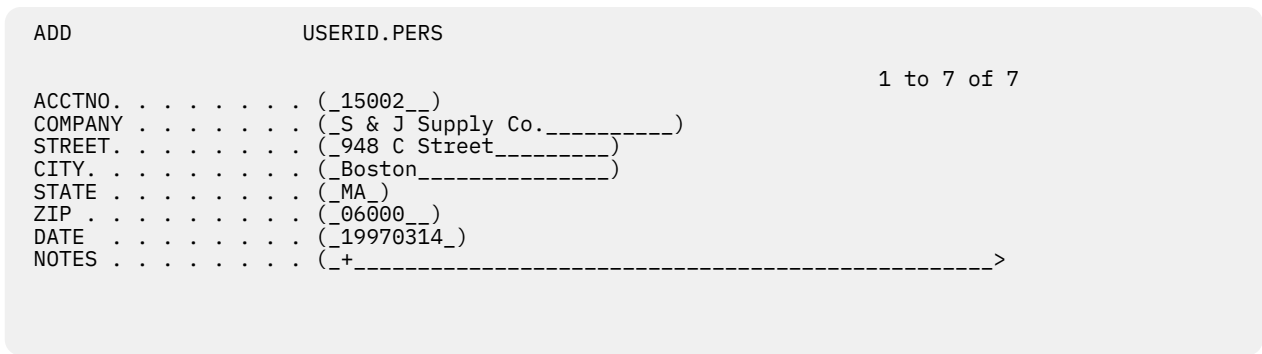


Figure 271. Use the **ADD** panel to type data for rows you want to insert into a table.

3. Press the Add function key after you enter all the data for the row.
  - If you specified that you want to save each row as you add it (SAVE=IMMEDIATE), QMF adds the new row to the table.
  - If you specified to save changes only at the end of the Table Editor session (SAVE=END), the new row is held until you end the Table Editor session.

The Table Editor **ADD** panel is reset as is shown in [Figure 270 on page 268](#) and is ready for you to add another row.

### Related concepts

#### Column default and null indicators

You can specify a default indicator for columns that can support a default or a null indicator for columns that support null values.

### Related reference

[SET PROFILE](#)The SET PROFILE command changes values in your QMF profile. These values influence the behavior of your QMF session.

## Adding a row by copying the previous row

If the row you want to add contains much of the same information as the previous row, you can save keystrokes and time by using the previous row as a model.

### Procedure

To add a row from a copy of the previous row:

1. Press the Previous function key.
  - The last row that you entered displays on the Table Editor **ADD** panel.
2. Type the information for the new row over the information that is displayed on the panel.
  - Make sure that you erase any information you do not need.

## Column default and null indicators

You can specify a default indicator for columns that can support a default or a null indicator for columns that support null values.

For example, when you type the column default indicator in a Table Editor field that supports it, QMF uses the default value for the field. If you specify the column default indicator for a column that has the system date defined as its default, QMF uses the system date in that column. The following table describes the column default and null indicators.

Table 46. Column default and null indicators

Indicator	Character initially provided with QMF	Global variables that define the indicator character
Column default	+	DSQCP_TEDFLT, DSQCP_TEDFLT_DBCS
Null	-	DSQCP_TENULL, DSQCP_TENULL_DBCS

If you specify the column default for a column that does not have a default, QMF returns an error message, and prompts you to correct your entry. Possible reasons QMF might not be able to detect a default value are:

- No default exists for the column.
- The EDIT TABLE command referenced a three-part table or view name.
- The table that you are editing is a view that is located at a Db2 for z/OS server.

## Redefining the column default and null indicators

You can change the column default and null indicators by changing the DSQCP\_TEDFLT and DSQCP\_TENULL global variables.

### Procedure

You can redefine the column default and null indicators from the command line by using the SET GLOBAL command. For example, to change the column default indicator to ? and the null indicator to #, enter the following at the command line:

```
SET GLOBAL (DSQCP_TEDFLT=?, DSQCP_TENULL=#)
```

Once you have issued this command, the Table Editor session illustrated in the following figure would reflect values similar to those shown in the following figure:

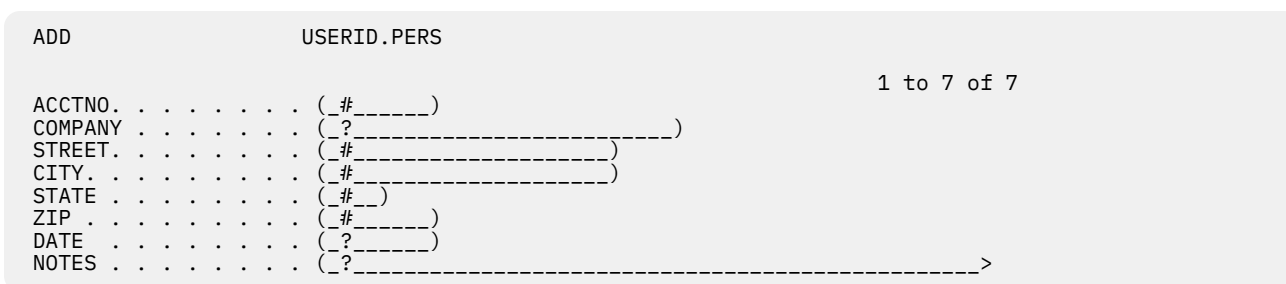


Figure 272. The Table Editor ADD panel with new values for the column default and null indicators

## Adding data to long fields

If a field is followed by a greater-than symbol (>) rather than a right parenthesis, the entire field is longer than 50 characters. If the information you need to type for this field is more than 50 characters, QMF provides a way to display the entire field.

### Procedure

To add more than 50 characters of data in a field:

1. Move the cursor to the field you want to display.

In the PERS sample table, the NOTES field is longer than 50 characters.

2. Press the Show Field function key.

The Show Field panel for the field displays.

The valid values for the field appear at the bottom of the panel on the message line.

### 3. Type the data for the field.

When you get to the end of one line, just continue typing. Your data will automatically wrap to the next line, as shown in the following figure.

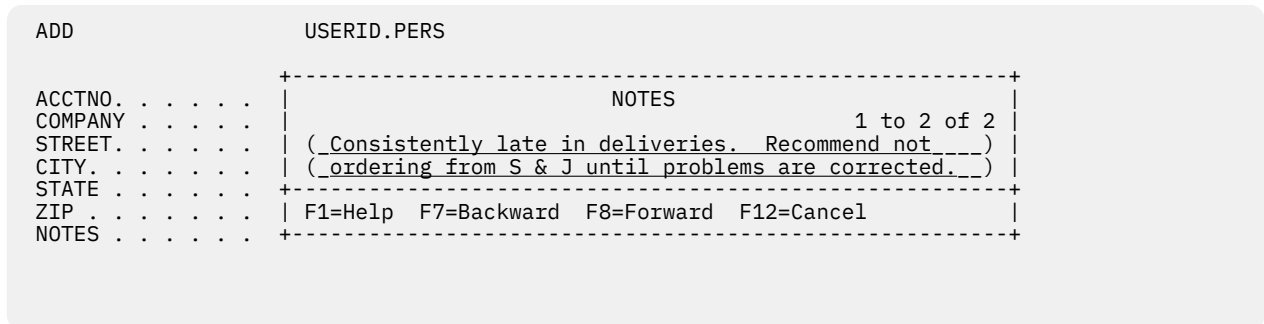


Figure 273. You can enter more data in long fields with the Show Field key.

### 4. Press Enter to save the data in the field.

The Table Editor ADD panel displays with the first 50 characters of the field displayed.

## Row changes using the Table Editor

You can use the Table Editor to change the rows in a table.

### Starting the Table Editor in CHANGE mode

When you are in CHANGE mode in the Table Editor you can change the data in a table.

#### Before you begin

Make sure the text case (UPPER, STRING, or MIXED) that you specify with the CASE parameter in your QMF profile is the same for this session as the text in the table. A MIXED value for the CASE parameter might have implications in TSO for the QMF CONNECT command.

#### About this task

If a table contains CLOB or DBCLOB columns, those columns cannot be edited. Also, only up to 32 K of CLOB data and up to 16 K of DBCLOB data is displayed.

#### Procedure

To make changes to the data in a table, do one of the following, depending on where you are starting:

- From the QMF home panel, type:

```
tablename (MODE=CHANGE
```

Then press the Edit Table function key.

- From any other QMF panel with a command line, enter:

```
EDIT TABLE tablename (MODE=CHANGE
```

#### Results

The Table Editor SEARCH panel displays so that you can search for the first row you want to change.

## Example

For example, to change rows in the PERS table, enter the following on the QMF command line:

```
EDIT TABLE PERS (MODE=CHANGE
```

To enter the same command and save each addition as you make it, enter:

```
EDIT TABLE PERS (MODE=CHANGE SAVE=IMMEDIATE
```

## Related tasks

### [Setting up and changing your QMF user profile](#)

Your QMF user profile tells QMF how you want to display information and process commands, and what printer to use when you print reports or charts.

## Related reference

**SET PROFILE**The SET PROFILE command changes values in your QMF profile. These values influence the behavior of your QMF session.

## Searching for the rows you want to change

After you start the Table Editor in CHANGE mode, you can search for rows to change.

## About this task

The Table Editor **SEARCH** panel, which is shown in the following figure, shows the name of each column in the table, followed by an entry field where you can enter search criteria to select the rows you want to change.

SEARCH	USERID.PERS	
		1 to 7 of 7
ACCTNO. . . . .	( _15002_ )	
COMPANY . . . . .	( _S & J Supply Co._____ )	
STREET. . . . .	( _948 C Street_____ )	
CITY. . . . .	( _Boston_____ )	
STATE . . . . .	( _MA_ )	
ZIP . . . . .	( _06000_ )	
DATE . . . . .	( _-_____ )	
NOTES . . . . .	( _-_____ )	>

Figure 274. The Table Editor **SEARCH** panel

## Procedure

To search for the rows you want to change:

1. Specify the search criteria:
  - Type the criteria that you want to use to select the rows to change.
  - Leave a null in any field for which you are not specifying selection criteria.
  - Press the Clear function key to clear all the fields and set them to the null or column default indicator.
  - Press the Show Field function key to see the data type for a column.
  - Press Enter to select all the rows in the table.
  - To select a specific set of rows to change, you can either use an exact value or use the underscore ( \_ ) and the percent sign ( % ) as selection symbols to specify selection criteria for any column that contains character or graphic data.
    - Use an underscore for one character.

- Use the percent sign for zero or more characters.
2. Press the Search function key.

The Table Editor **CHANGE** panel displays with the first row you selected.

## Changing rows

You can change rows by using the Table Editor **CHANGE** panel. You can change information in any field that is enclosed in parentheses.

### Procedure

To change a row:

1. On the Table Editor **CHANGE** panel, type the changes to that row.
  - To display the default settings for the fields, press the Show Field function key.
  - To change a value to the default value for a field, if a default value is available, type the default indicator in the field.
  - To remove changes you type and return the original data to the fields, press the Refresh function key.
  - To see the next row without changing this row, press the Next function key.
2. Press the Change function key.
  - If you specified that you want to save changes as you make them (SAVE=IMMEDIATE), QMF adds the changes to the table.
  - If you specified to save changes only at the end of the Table Editor session (SAVE=END), the changes are held until you end the Table Editor session.

If there are more rows to display that meet your search criteria, the Table Editor **CHANGE** panel displays with the next row in the search set. If there are no more rows that are left to display, the Table Editor **SEARCH** panel displays.

## Results

### What to do next

If you want to search for a different set of rows, press the Show Search function key.

## Deleting rows from a table

You can delete rows by using the Table Editor **CHANGE** panel.

### Procedure

To use the Table Editor to delete rows from a table:

1. Search for the rows you want to delete as explained in [“Searching for the rows you want to change” on page 272](#).
2. Make sure that the row you want to delete appears on the Table Editor **CHANGE** panel.
3. Press the Delete function key.
  - If you specified that you want to save changes as you make them (SAVE=IMMEDIATE), QMF deletes the row from the table when you press the Delete function key.
  - If you specified to save changes only at the end of the Table Editor session (SAVE=END), the changes are held until you end the Table Editor session.

If there are more rows to display that meet your search criteria, the Table Editor **CHANGE** panel displays. If there are no more rows that are left to display, the Table Editor **SEARCH** panel displays.

## Ending a Table Editor session

You can end a Table Editor session either with the End or the Cancel function keys.

### Procedure

- To end a Table Editor session, press one of the following function keys:
  - Press the End function key. If you specified SAVE=END when you began the Table Editor session, QMF saves any held changes in the database.
  - Press the Cancel function key. You can cancel a Table Editor session only if you specified SAVE=END. QMF does not save any held rows in the database.

### Results

The QMF panel from which you began the Table Editor session displays.

### What to do next

If you want to see the changed table, enter:

```
DISPLAY TABLE tablename
```

## Using SQL statements to maintain data

---

You can maintain and update the data in your tables by entering Structured Query Language (SQL) statements on the QMF SQL query panel.

Using SQL statements, you can:

- Add rows to a table
- Change rows in a table
- Delete rows from a table
- Copy rows from one table to another
- Add a column to a table
- Authorize others to access your tables
- Enter date and time values into a table

Except for CALL and CREATE PROCEDURE statements, QMF SQL queries can include multiple statements. However, only one SELECT statement can be included in a multistatement query.

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

## New table rows using SQL statements

If you want to use SQL statements to add rows to a table, you can use the QMF DRAW command to create a query that adds the data to the table. You can also add rows to a table by typing an SQL INSERT statement on the SQL query panel and running the query.

### Using the QMF DRAW command to add rows

One method of adding rows to a table is with the DRAW command, which adds one row for each DRAW command.

#### Procedure

To use the QMF DRAW command to add rows:

1. On the QMF command line, enter:

```
RESET QUERY (LANGUAGE=SQL
```

The SQL query panel displays.

2. On the QMF command line, enter:

```
DRAW tablename (TYPE=INSERT
```

The INSERT statement template for the table displays.

3. With your keyboard in text-insert mode, type to the left of the dash characters (--) the data that you want to insert for each column.

The highlighted text in the following figure is inserted as the row that is shown in Step 4.

```
INSERT INTO Q.ORG
(DEPTNUMB, DEPTNAME, MANAGER, DIVISION, LOCATION)
VALUES (
-- COLUMN NAME           DATA TYPE      LENGTH  NULLS
30-- DEPTNUMB           SMALLINT      10      NO
'MIDWEST'-- DEPTNAME     VARCHAR       14      YES
130-- MANAGER           SMALLINT      10      YES
'MIDWEST'-- DIVISION     VARCHAR       10      YES
'WISCONSIN'-- LOCATION   VARCHAR       13      YES
)
```

Figure 275. Inserting data using the DRAW command

4. Press the Run function key.

The example shown in Step “3” on page 275 inserts the following row into the Q.ORG table:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
30	MIDWEST	130	MIDWEST	WISCONSIN

#### What to do next

Repeat these steps to add additional rows to the table.

## Writing your own INSERT statement to add rows

In addition to the QMF DRAW command, you can use an SQL INSERT statement to add rows.

### About this task

The syntax of the SQL INSERT statement is:

```
INSERT INTO tablename
VALUES (value1, value2, value3, ...)
```

In this INSERT statement:

- *tablename* is the name of the table into which you are inserting data
- *value1, value2, value3* are the data values you are adding to each column

If you are adding data to only certain columns in the table, name those columns in a parenthetical statement after the table name.

### Procedure

To write an INSERT query:

1. Issue the following command:

```
RESET QUERY (LANGUAGE=SQL
```

The SQL query panel displays.

2. Type one or more SQL INSERT statements to add data to the table.

If you do not specify data for a column, QMF adds a null value.

3. Press the Run function key to run the query.

### Results

QMF adds the new row to the table.

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

### Related reference

[INSERT](#) is an SQL statement that adds data to a table.

## Row changes using SQL statements

You can change rows in a table by using the QMF DRAW command to create a query that updates the data in the table or by typing an SQL UPDATE statement on the SQL query panel and running the query.

### Using the QMF DRAW command to change rows

You can create a query that changes the rows in a table.

### Procedure

To use the QMF DRAW command to change rows in a table:

1. On the QMF command line, enter:

```
RESET QUERY (LANGUAGE=SQL
```

The SQL query panel displays.



2. Issue the DRAW command:

```
DRAW tablename (TYPE=UPDATE
```

For example:

```
DRAW Q.STAFF (TYPE=UPDATE
```

The UPDATE query template for the table displays, as shown in the SQL QUERY panel in the following table.

SQL QUERY	MODIFIED	LINE	1
UPDATE Q.STAFF			
SET			
-- COLUMN NAME	DATA TYPE	LENGTH	NULLS
ID=	-- SMALLINT		NO
, "NAME"=	-- VARCHAR	9	YES
, DEPT=	-- SMALLINT		YES
, JOB=	-- CHAR	5	YES
, "YEARS"=	-- SMALLINT		YES
, SALARY=	-- DECIMAL	( 7, 2)	YES
, COMM=	-- DECIMAL	( 7, 2)	YES
WHERE			

Figure 276. A template UPDATE query drawn for the Q.STAFF table

3. Indicate which rows in the table you want to change by typing a clause after the WHERE keyword at the bottom of the drawn UPDATE statement. For this example, you will change a single row for employee 30. ID is the name of the column in the Q.STAFF table that contains employees' ID numbers, so type ID=30 after the WHERE keyword in the drawn statement. The finished clause reads WHERE ID=30.
4. For each column value you want to change in the row indicated in the WHERE clause, type the new data after the = sign. For this example, change employee 30's last name from Marenghi to Glock by typing GLOCK after "NAME"= in the drawn UPDATE statement. The name is surrounded by single quotes because the NAME column contains character data and all character data must be surrounded by single quotes.
5. Press the Delete function key to delete any columns not needed in the UPDATE statement (in this example, all columns but the NAME column), and ensure that the first column has no comma before the column name. The following figure shows the finished statement for this example.

```
SQL
QUERY

UPDATE Q.STAFF
SET
-- COLUMN NAME
"NAME"=' GLOCK'
WHERE ID=30

*** END ***
```

Figure 277. The finished UPDATE statement changes the value of the NAME column for employee 30 to GLOCK.

6. Press the Run function key to run the query.

## Results

QMF updates the table. Repeat these steps to update additional rows in the table.

## Writing your own UPDATE statement to change rows

You can enter an SQL UPDATE statement in the SQL query panel to update a table.

### About this task

The format of the SQL UPDATE statement is:

```
UPDATE tablename
SET column=value
WHERE clause
```

where:

- *tablename* is the name of the table that contains the data you want to update.
- *column=value* specifies the column to change and the new value to use for the column.
- *clause* is a SQL clause indicating the row in which the update is to be made.

For example, the following statement changes the employee ID of an employee named GRIMLEY in the Q.STAFF sample table to 30:

```
UPDATE Q.STAFF
SET ID=30
WHERE NAME='GRIMLEY'
```

### Procedure

To issue an UPDATE statement:

1. On the QMF command line, enter:

```
RESET QUERY (LANGUAGE=SQL
```

The SQL query panel displays.

2. Type one or more SQL UPDATE statements to change the necessary rows in the table.
3. Press the Run function key to run the query.

## Results

QMF makes the updates to the table.

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

## Deleting rows from a table

You can type an SQL DELETE statement on the SQL Query panel to delete one or more rows from a table.

For example, the following statement deletes the row for employee number 410 from the MYSTAFF table:

```
DELETE FROM MYSTAFF
WHERE ID = 410
```

This statement deletes all rows that are associated with Department 38:

```
DELETE FROM MYSTAFF
WHERE DEPT = 38
```

### Related tasks

[Including multiple SQL statements in one query](#)

You can include multiple SQL statements on the SQL query panel. The exceptions are CALL and CREATE PROCEDURE statements. These statements must be used alone in a query.

## Copying rows from one table into another

You can use an INSERT statement to copy certain rows and columns from an existing table into another table.

You can add the rows to an existing table or specify a new table name and create a table to contain the rows you specify.

For example, the following INSERT statement inserts into a table that is called MYSTAFF the ID, NAME, DEPT, and JOB data for all employees in Department 38 in the Q.STAFF table:

```
INSERT INTO MYSTAFF (ID, NAME, DEPT, JOB)
SELECT ID, NAME, DEPT, JOB
FROM Q.STAFF
WHERE DEPT = 38
```

After you run this statement, QMF adds five new rows to the MYSTAFF table. For these employees, the YEARS, SALARY, and COMM columns contain null values, because the SELECT statement does not select these columns. If you want to include all the data for a row, you must select all the columns in the table.

## Adding a column to a table

Use the ALTER TABLE SQL statement to add a column to a table.

For example, to add a NOTES column to the CALENDAR table, run the following SQL statement:

```
ALTER TABLE CALENDAR
ADD NOTES VARCHAR(40)
```

In this statement, NOTES is the name of the new column, VARCHAR is the data type, and 40 is the number of characters in the column.

If the table is stored in a Db2 for z/OS database, you can specify a default value other than null for the column. If the table is stored in a DB2 Server for VSE and VM database, the default value must be null for the column.

### Related information

[The IBM Publications Center](#)To find information about the ALTER TABLE SQL statement, see the SQL reference information for the database you are using.

## Table access

After you create a table and add data to it, you can use SQL statements to specify how you want other users to access the information. For example, you can authorize users to make changes to your tables, or you can limit their access so that they can only view the data.

### Giving users access to your tables

You can give table-wide authority to users by using the GRANT command.

### About this task

You can give users authority to do any of the following with your tables:

- View the data in a table
- Add new rows to the table
- Change the rows in the table
- Delete rows from a table

**Restriction:** In CICS, you can only allow users authority to view data (not update data) in tables at remote locations.

## Procedure

Use the GRANT command to give users access to your tables.

### Example

For example, to give a user with user ID LINDSAY the authority to view, add, change, and delete the data in the CALENDAR table, run the following query:

```
GRANT ALL ON TABLE CALENDAR
TO LINDSAY
```

To allow LINDSAY the authority to view the data in the PERS table, run this query:

```
GRANT SELECT ON TABLE PERS
TO LINDSAY
```

To allow all users (regardless of location) to have all privileges for the MYSTAFF table, run this query:

```
GRANT ALL ON TABLE MYSTAFF
TO PUBLIC AT ALL LOCATIONS
```

## Allowing users to update only specific columns in your tables

You can permit a user to update only certain columns in your tables by qualifying the UPDATE privileges with the names of specific columns.

### Procedure

The following example shows you how to give user LINDSAY the authority to update the LOCATION column in the CALENDAR table:

```
GRANT UPDATE(LOCATION) ON TABLE CALENDAR TO LINDSAY
```

This example grants user LINDSAY the authority to change the contents of the LOCATION column in the CALENDAR table.

### Related information

[The IBM Publications Center](#) To find information about the GRANT keyword, see the SQL reference information for the database you are using.

## Revoking access to a table

You can revoke privileges you have granted for a particular table.

For example, to prevent LINDSAY from deleting rows from the CALENDAR table, run this query:

```
REVOKE DELETE ON CALENDAR
FROM LINDSAY
```

### Related information

[The IBM Publications Center](#) To find information about the REVOKE keyword, see the SQL reference information for the database you are using.

## Entering date and time values with QMF

By default, or if the DSQSFISO program parameter is set to YES, QMF application programs are precompiled with the date and time formats of the International Standards Organization (ISO), which represents dates as *yyyy-mm-dd* and times as *hh.mm.ss*.

For the default situation, , if you insert a date or time value into a character column using a special register such as CURRENT DATE or CURRENT TIME, the character-string representation of the value is in the ISO format.

To insert the value in a format other than ISO (such as EUR format), you can use a statement like this:

```
INSERT INTO date_table
  SELECT CHAR(CURRENT DATE, EUR)
  FROM any_table
  WHERE any_table.unique_column = 'unique_value'
```

In this statement, *date\_table* is the name of the table into which you want to insert the current date value, *any\_table* is any table (preferably not one that is subject to changes) with a column that contains unique values, and 'unique\_value' is a value of the unique column. In these examples, *date\_table* has one character column, which contains the character representation of a date value.

To insert the default format, you can enter an SQL statement like the following:

```
INSERT INTO date_table
  VALUES(CURRENT DATE)
```

To update a character column with the CURRENT DATE or CURRENT TIME value in a format other than ISO, use a statement such as the following:

```
UPDATE date_table
  SET date_column = CHAR(CURRENT DATE, EUR)
  WHERE (clause identifying row to be updated)
```

In this statement, *date\_column* is a column defined with the DATE data type.

This behavior may be changed within a QMF session by setting the DSQEC\_DSQSFISO global variable. Changing DSQEC\_DSQSFISO to '0' will format CHAR(*datetime-expression*) in the format that is specified in the DATE FORMAT and TIME FORMAT fields on Db2 installation panel DSNTIP4. To determine the DATE FORMAT and TIME FORMAT values defined in the current Db2 subsystem, reference the DSQAO\_DATE\_FORMAT and DSQAO\_TIME\_FORMAT global variables.

### Related reference

**INSERT**The INSERT command inserts lines onto specific panels.

**UPDATE**The UPDATE statement changes the values of specified existing columns in rows of a table. You can update a table only if you created the table or are authorized to update the table.



## Chapter 13. Exporting and importing objects

Use the QMF EXPORT command to place objects outside the QMF environment, and use the IMPORT command to bring objects into QMF.

### Related reference

**EXPORT in CICS** Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

**EXPORT in TSO** The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

**IMPORT in CICS** The IMPORT command copies the contents of a CICS data queue into QMF temporary storage or into the database.

**IMPORT in TSO** The IMPORT command copies the contents of a TSO data set or UNIX file into QMF temporary storage or into the database.

## Exporting objects

Using the EXPORT command, you can export data and tables to a UNIX file, TSO data set, or CICS data queue and export QMF objects such as forms, reports, and queries to a TSO data set or CICS data queue.

You can export objects from either temporary storage or from the database, depending on the object type, as shown in the following table.

Object	Export from database	Export from temporary storage
TABLE	X	
QUERY	X	X
FORM	X	X
PROCEDURE	X	X
DATA		X
REPORT		X
CHART		X

You can export reports, charts, and data in comma-separated value (CSV) format from temporary storage, but you cannot import these objects into temporary storage.

When you export a QMF object from an object panel, you do not need to specify the object type for the export. For example, if you enter the EXPORT command from a form panel, the form currently displayed on the panel is exported with the default object type FORM. If you enter the EXPORT command from a chart panel, the chart currently displayed on the panel is exported with the default object type CHART. If you attempt to export from a panel that does not have a valid object type for export, and do not specify a valid object type, QMF prompts you for a valid object type.

The syntax of the EXPORT command varies depending on the object you are exporting and the environment you are working with. Additionally, some restrictions exist for exporting data or tables containing certain data types.

## Exporting objects from QMF for TSO

You can export an object to a TSO data set. If you are using QMF in TSO, you can also export data and tables to UNIX files.

### Procedure

Choose one of the following options, depending on whether you are exporting an object that is in QMF temporary storage or an object that is stored in the database.

- To export to a TSO data set an object that is in QMF temporary storage, use the following basic command:

```
EXPORT objecttype TO dataset
```

For example, to export a query that is in QMF temporary storage (which means that it is displayed when you enter SHOW QUERY) to a TSO data set that is named REPORTX, enter:

```
EXPORT QUERY TO REPORTX
```

- To export to a TSO data set an object that is stored in the database, use the following basic command:

```
EXPORT objecttype objectname TO dataset
```

You can use either a fully qualified or partially qualified name in TSO.

For example, to export a query that is named MYREP4Q (and is saved in the database) to a TSO data set named RPT4Q, enter:

```
EXPORT QUERY MYREP4Q TO RPT4Q
```

### Related reference

[EXPORT in TSO](#)The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

## Exporting objects from QMF for CICS

You can export an object to a CICS data queue or to a CICS temporary storage queue.

### Procedure

To export an object from QMF for CICS, choose one of the following options:

- To export to a CICS data queue an object that is currently in QMF temporary storage, use the following basic command:

```
EXPORT objecttype TO queuename (queuetype=TS|TD
```

For example, to export a query that is currently in QMF temporary storage to a transient data queue named REPORTX, enter the following:

```
EXPORT QUERY TO REPORTX (queuetype=TD
```

- To export the same query to a CICS temporary storage queue, you do not need to specify the *queuetype* parameter because temporary storage (TS) is the default queue type:

```
EXPORT QUERY TO REPORTX
```

- To export an object in the database to a CICS data queue, use the following command:

```
EXPORT objecttype objectname TO dataqueue (queuetype=TS|TD
```



For example, to export a query that is named MYREP4Q (and is saved in the database) to a CICS temporary storage queue named RPT4Q, enter:

```
EXPORT QUERY MYREP4Q TO RPT4Q
```

### Related reference

[EXPORT in CICS](#) Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

## Exporting QMF reports for use on the Internet

You can export reports for use on the Internet by specifying the HTML parameter with your EXPORT REPORT command.

### Procedure

To export QMF reports for use on the Internet, choose one of the following options:

- To export an HTML report to a TSO data set, enter:

```
EXPORT REPORT TO dataset (DATAFORMAT=HTML
```

- To export an HTML report to a CICS data queue, use the following command syntax. Specify either TS (temporary storage) or TD (transient data) for the **queuetype** parameter.

```
EXPORT REPORT TO queuname (QUEUETYPE=TS|TD DATAFORMAT=HTML
```

### Results

The resulting report contains HTML code that allows your report to be viewed with a web browser.

### Related reference

[EXPORT in CICS](#) Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

[EXPORT in TSO](#) The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

## Exporting QMF report text

You can export just the text of reports by specifying the TEXT parameter with your EXPORT REPORT command. This option excludes the control information that QMF typically adds to exported reports.

### Procedure

Enter the following command:

```
EXPORT REPORT TO datasetname (DATAFORMAT=TEXT
```

## Scenario: Exporting data in CSV format

You can use the EXPORT command to retrieve data and view it in a spreadsheet format.

### About this task

Suppose that a colleague wants to know which of your sales representatives sold less than \$1000 of your product so far in the third quarter. Your colleague wants the information about these sales representatives in comma-separated value (CSV) format so that the data can be viewed and manipulated in a spreadsheet program. First, you must run a query to retrieve the information. Then, you can export the data in CSV format and download the CSV file to your workstation.

## Procedure

To retrieve and export the data in CSV format:

1. From the QMF home panel, press the Query function key.
2. Enter the following query:

```
SELECT * FROM SALESTOTALS WHERE Q3 < 1000
```

3. Press the Run function key.
4. On the command line of the QMF home panel, enter the following command:

```
EXPORT DATA TO Q3SALES (DATAFORMAT=CSV
```

The data is exported to a data set named *userid.Q3SALES.DATA*.

By default, column headings are exported along with the data. To exclude the headings, you can use the HEADER option, as in the following example:

```
EXPORT DATA TO Q3SALES (DATAFORMAT=CSV HEADER=NO
```

## Downloading the exported data

### About this task

You can now download the data to your workstation in CSV format.

### Procedure

To download the exported data to your Windows workstation:

1. Open a DOS command prompt.
2. Navigate to the directory where you want to download the file:

```
cd C:\mycsv
```

3. Open a connection to your z/OS host:

```
ftp host.name
```

4. Enter your user name:

```
User <host>: userid
```

5. Enter your password:

```
Password: password
```

6. Transfer the file to your workstation:

```
get userid.q3sales.data q3sales.csv
```

The data is downloaded to your workstation as C:\mycsv\q3sales.csv.

### Results

When you open the CSV file, your spreadsheet program might require you to specify the type of column separator that the file uses. The column separator that QMF uses is determined by the DECIMAL option of your QMF profile. If the DECIMAL option is set to PERIOD, the column separator is a comma. If the DECIMAL option is set to COMMA or FRENCH, the column separator is a semicolon.

## Importing objects

Use the IMPORT command to bring the contents of a UNIX file, a TSO data set, or a CICS data queue into a QMF temporary storage area or the database.

You can import objects into either temporary storage or the database depending on the object type, as is shown in the following table.

*Table 48. Objects you can import*

Object type	Import to database	Import to temporary storage
TABLE	X	
QUERY	X	X
FORM	X	X
PROC	X	X
DATA		X

### Restrictions:

- You cannot import reports, charts, or CSV files into QMF. For reports and charts, you can instead import the data and the forms that were used to create them.
- If you import a QBE query that was exported from a QMF Version 11.1 or earlier system, the query is converted in temporary storage with long-name characteristics and cannot be used if you connect to a QMF Version 11.1 or earlier system. Furthermore, if you save the imported query, it cannot be used in QMF Version 11.1 or earlier systems.

The syntax of the IMPORT command varies depending on the object you are importing, and the environment and location from which you are importing it. Additionally, some restrictions exist for importing data or tables containing certain data types.

### Related reference

**IMPORT in CICS**The IMPORT command copies the contents of a CICS data queue into QMF temporary storage or into the database.

**IMPORT in TSO**The IMPORT command copies the contents of a TSO data set or UNIX file into QMF temporary storage or into the database.

## Importing objects from TSO data sets

You can use the IMPORT command to import objects from TSO data sets. You can use either a fully qualified or partially qualified name in TSO.

### Procedure

- To import the contents of a TSO data set into QMF temporary storage, use the following command syntax:

```
IMPORT objecttype FROM dataset
```

For example, to import a query in a data set that is named REPORTX to temporary storage, enter:

```
IMPORT QUERY FROM REPORTX
```

- To import an object in a data set to the database, enter:

```
IMPORT objecttype objectname FROM dataset
```

For example, to import a query from a data set named RPT4Q and save the query as MYREP4Q in the database, enter:

```
IMPORT QUERY MYREP4Q FROM RPT4Q
```

## Importing objects from CICS data queues

You can use the IMPORT command to import objects from CICS data queues.

### Procedure

- To import an object into QMF temporary storage from a CICS data queue, use the following command syntax.

Temporary storage (TS) is the default queue type.

```
IMPORT objecttype FROM queuename (queuetype=TS|TD
```

For example, to import a query from a temporary storage queue named REPORTX into QMF temporary storage, enter:

```
IMPORT QUERY FROM REPORTX
```

- To import an object from a data queue into the database, use the following syntax:

```
IMPORT objecttype objectname FROM dataqueue (queuetype=TS|TD
```

For example, to import a query from a CICS temporary storage queue named RPT4Q and save it as MYREP4Q in the database, enter:

```
IMPORT QUERY MYREP4Q FROM RPT4Q
```

---

## Chapter 14. Remote data access

You can use QMF to access data in remote database.

### Accessing data at a remote database

---

When you start QMF, the system from which you start the program is known as the *local* system. The Db2 database that resides on this system is known as the *local* database. Databases that do not reside on this system are known as *remote* databases, or remote servers.

#### About this task

You can access data at a remote Db2 database using QMF. Then you can create reports and charts to view the data on your local system.

**Restriction:** Different databases have varying levels of support for data types and various functions. When you display a table or view on a remote database, some operations such as SAVE or EXPORT might be unsupported if the object includes or refers to a data type that is not supported by that database.

If you have started QMF as a stored procedure, the query or procedure that you pass on the CALL statement cannot access data at a remote location.

#### Procedure

QMF provides two ways to access data that is stored in remote databases:

- Connect to the remote database by using the QMF CONNECT command.
- While using QMF on the system where you started the program, issue a QMF command that refers to an object by a three-part name, part of which references the name of the remote database in which the data is stored.

You can use this method to access tables and views that are stored in remote databases. To access QMF queries, procedures, and forms, you must use the QMF CONNECT command to connect to the remote database. You can use QMF commands with three-part names only when the name of the database that you see beneath Connected to on the QMF home panel is a Db2 for z/OS database.

#### Related concepts

[The QMF CONNECT command and access data at a remote database](#)

When you are connected to a remote database, you access data and objects in the same way that you access them at a local database. QMF continues to use programs that reside on the local system.

[Three-part names in QMF commands to access data in a remote database](#)

You can use QMF to access tables and views that are stored in a remote database by using a three-part name to refer to the table or view. By using three-part names, you can access and manipulate the data that you need without first having to use the QMF CONNECT command to connect to the remote database.

[Installing and managing QMF for TSO and CICSThe stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.](#)

### The QMF CONNECT command and access data at a remote database

---

When you are connected to a remote database, you access data and objects in the same way that you access them at a local database. QMF continues to use programs that reside on the local system.

You can use the QMF CONNECT command to connect to and access data at any of the following types of remote databases:

- Db2 for z/OS

- Db2 for Linux, UNIX, and Windows
- DB2 for iSeries
- DB2 for VSE and VM

## Determining the database to which you are connected

QMF provides several ways for you to view the name of the database to which you are currently connected. Viewing the current database name can help you orient yourself if you are accessing data in more than one location.

### Procedure

To determine the database to which you are connected, choose one of the following options:

- View the current database location on the QMF **home** panel.

When you connect to a new database, that location name is displayed below the **Connected to** heading on the QMF **home** panel.

- View the current database location by using a global variable. Enter `SHOW GLOBALS`.

If the location name does not appear on the QMF **home** panel (for example, if you are connected to a Db2 subsystem without a location name), you can view the database location name by displaying the global variable `DSQAO_CONNECT_LOC`.

The **GLOBALS** panel displays with a list of all the global variables that are provided by QMF and defined by users. The value of the `DSQAO_CONNECT_LOC` variable is the name of the database to which you are connected.

### Related concepts

[Starting from the QMF home panel](#)

After you start QMF, you see the QMF home panel. The home panel displays basic information about your session.

### Related reference

[QMF global variables](#) QMF provides many global variables that help you control aspects of your QMF session, QMF commands, and panel display. The global variables also help you control behavior of QMF functions in procedures and applications.

## Guidelines for the QMF Connect command

Keep in mind certain guidelines when you use the `CONNECT` command.

- Commands, queries, and SQL statements that create or access objects that are stored in the database, such as `DISPLAY TABLE tablename`, go to the database to which you are currently connected (the current location). Thus, any objects that are referenced in any commands or SQL statements you issue after you connect must reside at the current location.

For example, if a query called `MYQUERY` exists in a database named `CHICAGO` and you connect to a database named `NEW_YORK` and enter `RUN MYQUERY`, the command fails unless `MYQUERY` is also stored in the `NEW_YORK` database.

Objects can reside in temporary storage on the system where QMF is running. For example, if you enter `DISPLAY MYQUERY` while connected to `CHICAGO` and then you connect to `NEW_YORK` and enter `RUN QUERY`, QMF runs `MYQUERY`. The query runs even though it is not stored in the `NEW_YORK` database because it is in the `QUERY` temporary storage area.

If you are using QMF under CICS and connect to a database in a VM or TSO environment, procedures with logic and report calculations do not run from your CICS session because REXX is not available in CICS.

- In Db2 for z/OS, the `SQLID` you were using is no longer active after you connect to a different location. If you need to use the same `SQLID` with multiple Db2 application servers from a single QMF session, you might need to reset the `SQLID` after you connect to each server.

- After a successful connection, your profile (except for the value of the TRACE option), resource control table, command synonyms, and function keys are reinitialized to the values at the current location.
- When you use the CONNECT command to connect to a remote database, all programs started by QMF run under the operating system in which QMF is running (the local operating system). Such programs might include commands for other operating environments. For example, QMF is running in TSO and you are connected to a DB2 for VSE and VM database. You attempt to run a procedure that contains CMS or CICS commands. The TSO commands work, but the CMS and CICS commands do not.
- Before it connects to the remote database, QMF completes any work (for example, a large report) at the current location.

Your QMF administrator can specify the DSQSDBNM program parameter in the program logic so that you can start QMF. When this parameter is specified, QMF issues a CONNECT command to connect to the remote database before the QMF home panel is displayed. See your administrator for more information about configuring QMF to start in this manner. If you use a command to start QMF, such as QMFE to start the English version of QMF for CICS, you can specify the program parameter directly on this command. For example, the following command starts QMF for CICS and connects to a database named DB24 before the home panel displays:

```
QMFE DSQSDBNM=DB24
```

### Related tasks

[Specifying the initial database connection](#) You can start QMF in the local Db2 for z/OS subsystem and connect immediately to another database before the QMF home panel is displayed.

### Related reference

[CONNECT in CICS](#) With the CONNECT command, you can connect to any database server that is part of the distributed network from within a QMF session.

[CONNECT in TSO](#) You can use the CONNECT command from within a QMF session to connect to any database server that is part of the distributed network.

## How to issue the CONNECT command

You can use the QMF CONNECT command to connect to a remote database during a QMF session.

### About this task

You can issue the CONNECT command from:

- The command line
- Within a procedure (a linear procedure or a procedure with logic)
- The callable or command interface

The *current location* is the database location to which you are currently connected. The system where QMF is running is the operating system from which you started QMF, also known as the *local system*. The Db2 database at this location is known as the local database.

**Restriction:** You cannot use the CONNECT command to access a remote database if you started QMF as a stored procedure.

### Procedure

To use the CONNECT command:

1. On the QMF command line, enter:

```
CONNECT ?
```

The **CONNECT Command Prompt** panel displays, as shown in the following figure:

```

CONNECT Command Prompt
1 to 13 of 13
User ( )
.... ( )
.... ( )
Enter the userid
attempting to connect to the application server.
Password ( )
Enter the password of the userid
attempting to connect to the application server.
TO
Location ( ) +
Enter the location name
of the application server to be connected.
F1=Help F3=End F4=List F7=Backward F8=Forward
Please follow the directions on the command prompt panel.

```

Figure 278. The **CONNECT Command Prompt** panel

2. Type the information that you need to connect to the remote database, then press Enter.  
If a plus sign appears after the **Location** field, you can press the List function key to display a list of database names to which you can connect.  
QMF connects you to the remote location and displays the following message on the message line:  
"OK, CONNECT performed. Please proceed."

## What to do next

If you select a database from the list but cannot connect to it, check the following items:

- You have the authority to connect to the database.
- You correctly entered your user ID and password for the remote database.
- The CASE value in your QMF profile is set to MIXED if your site uses mixed-case passwords (otherwise, QMF converts your password to uppercase and fails to connect you to the database).
- The database location supports connections of this type.

Ask your administrator if the database server to which you are trying to connect is prepared for access by the QMF CONNECT command.

- The database is running.

The database might be stopped for maintenance or other reasons. Check with your administrator to make sure that the database is started.

### Related concepts

[Procedures to run QMF commands](#)

You can create two types of procedures: procedures with logic or linear procedures.

[Application programming interfaces to QMF](#) There are two application programming interfaces to QMF: Callable interface and command interface.

[Installing and managing QMF for TSO and CIC](#) The stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.

### Related tasks

[Determining the database to which you are connected](#)



QMF provides several ways for you to view the name of the database to which you are currently connected. Viewing the current database name can help you orient yourself if you are accessing data in more than one location.

## Reconnecting to a remote database

You can reconnect to a remote database if the connection is lost. The steps you need to take to reconnect vary depending on whether you are using QMF interactively or in batch mode.

### Reconnecting if you are running QMF interactively

If you are entering QMF commands interactively and the connection to the remote location is lost, the **Lost Connection Prompt** panel displays.

#### About this task

If you are running QMF commands in a linear procedure and the connection to the remote location is lost, the procedure ends. If you are running the procedure interactively, the **Lost Connection Prompt** panel displays so you can reconnect.

If you are running QMF commands in a procedure with logic, the logic of the procedure determines how the procedure ends. When the procedure finishes, the **Lost Connection Prompt** panel displays so you can reconnect.

#### Procedure

To reconnect to the remote database:

1. Type 1 to reconnect to the remote database, or type 2 to exit QMF.

If you type 1 to reconnect to the remote database, the **CONNECT Command Prompt** panel displays.

The name of the location to which you were previously connected appears in the **TO Location** field.

2. Enter the information to reconnect to the remote database.

If QMF cannot connect to the location you specify, the **CONNECT Command Prompt** panel displays so you can try to connect again.

### Reconnecting if you are running QMF commands in batch mode

If the connection to the remote location is lost when QMF is running an application or procedure in batch mode, QMF ends.

#### Procedure

To reconnect to the remote location:

1. Start your QMF session.
2. Unless you automatically connect to the remote database when you start QMF, use the **CONNECT** command to connect to the remote database.
3. Restart the batch application or procedure.

### What you can expect when you reconnect

If you are using the Table Editor to update a table in a remote database and you lose the connection to the database, any updates you have not saved (for example, if you specified **SAVE=END** when you started the Table Editor) are lost.

If you are running QMF commands from a list of database objects and the connection to the database is lost, the database list is obsolete. You can still display the list, but if you type a command on the list, you will receive an error.

## Three-part names in QMF commands to access data in a remote database

You can use QMF to access tables and views that are stored in a remote database by using a three-part name to refer to the table or view. By using three-part names, you can access and manipulate the data that you need without first having to use the QMF CONNECT command to connect to the remote database.

The name takes the following form:

```
location.owner.name
```

The parts of the name syntax have the following meanings:

### **location**

The name of the database in which the table or view is stored. Location names can be up to 16 characters.

### **owner**

The ID of the QMF user who created the table.

### **name**

The name of the table or view.

The following example displays a table named STAFF that is owned by user Q and in a database named NEW\_YORK:

```
DISPLAY TABLE NEW_YORK.Q.STAFF
```

You can also use three-part names in SQL queries to select data from tables in remote databases. For example, this query displays the same table as the previous DISPLAY TABLE command.

```
SELECT * FROM NEW_YORK.Q.STAFF
```

If you use three-part-names to perform operations at remote databases, keep in mind the following:

- You can retrieve data from more than one table only if the tables are at the same location.
- You can update tables that are at remote locations, but you can create tables at remote locations only if you first issue a CONNECT command to connect to the remote location.
- An alias for the three-part name can be assigned for use in querying a remote table. For example, the alias NYSTAFF can be assigned to the table NEW\_YORK.Q.STAFF. See your administrator for a list of aliases for tables and views that you need to access.
- You can issue a GRANT statement at a remote location if you first connect to the remote location. You can grant privileges on a table that resides at the current location to users at other locations by using the clause PUBLIC AT ALL LOCATIONS on the GRANT statement. You cannot use a three-part name in GRANT statements if the three-part name refers to an object at the local Db2 database.
- By default, three-part names cannot be used to access remote tables that contain LOB data. However, you can set the DSQEC\_LOB\_RETRV global variable to 2 or 3 to access LOB metadata or data with a three-part name. Or, you can use the CONNECT command to connect to the database, and then run the query to access the remote table.
- Commands that include three-part names cannot:
  - Be directed to DB2 for VSE and VM databases
  - Be issued in queries or procedures that are run after QMF is started as a stored procedure

### **Related concepts**

[Installing and managing QMF for TSO and CIC](#)The stored procedure interface to QMF for TSO allows any software program that can call a Db2 for z/OS stored procedure to start QMF for TSO.

---

## Chapter 15. Introducing QMF Enhanced Editor

The QMF Enhanced Editor adds a number of tailored facilities to the standard ISPF edit environment to aid with query and procedure development.

- QMF objects can be edited and saved with the help of the standard ISPF editor commands. Named objects are saved back to QMF temporary storage and the database. Unnamed objects are saved back only to QMF temporary storage.
- QMF objects can be saved during the edit session by using the standard ISPF SAVE command, or saved to another name using the SAVEAS command.
- The standard ISPF Edit commands COPY, MOVE, CREATE, and REPLACE can be used to read and write QMF objects directly to and from the database.
- QMF pattern matching can be used to generate a list of selectable objects when using the COPY, MOVE, and REPLACE commands.
- QMF object syntax coloring detects the language type and can individually color many syntactical elements. These settings can be customized and saved in your ISPF profile by using the COLORS command.
- QMF objects can be run with or without a command prompt by using the RUN command. For QMF queries, the final report object is displayed in QMF.
- For QMF queries, the PREVIEW command allows generating and scrolling the first 100 rows of any result table on the edit panel itself.
- Attributes that are associated with the edited object, such as comments and share option, can be viewed and updated within the editor by using the DESCRIBE command. Object attributes are automatically saved back to the database with the object data.
- QMF queries can also use the Assist feature, which is automatically activated when a table or column name is selected on the edit panel.
- QMF queries can be syntax-checked without the calling Db2 by using the SYNTAX command.

The QMF Enhanced Editor requires that QMF be started using ISPF services. For Enhanced Editor configuration instructions, see the "Configuring QMF Enhanced Editor" topic in the Installing and Managing Db2 QMF for TSO and CICS guide.

---

### Coloring object syntax

You can customize QMF object syntax coloring to your requirements by using the Color settings panel.

#### About this task

Open the panel by entering the COLORS command on the command line or by selecting the **Color settings** item from the **Settings** menu. Color settings are saved in your QMF ISPF profile.

The coloring logic is sensitive to the QMF object type and assigns a color to each syntax element. The elements apply to the SQL, REXX, and QMF linear procedure languages. Each syntax element can be assigned a specified color from the set of ISPF highlighting colors. They are RED, PINK, GREEN, YELLOW, BLUE, TURQ, and WHITE.

#### Procedure

1. To change a color:
  - a) Navigate to the color field next to the syntax element.
  - b) Enter a new color and press ENTER.
  - c) Clear the color value to blank to automatically set it to the default ISPF color, GREEN.

2. To remove coloring completely by setting all colors to GREEN, enter the `COLORS OFF` command or select the **Remove point-and shoot** field on the **Color settings** panel.
3. To reset coloring to the initial default color scheme, enter the `COLORS DEFAULT` command or select the **Defaults point-and-shoot** field on the **Color settings** panel.
4. To save updated color settings to your QMF ISPF profile from the **Color settings** panel, enter the `END` command or enter `CANCEL` to exit without saving.

**Note:** Coloring is refreshed each time the edit panel itself is redisplayed. When the edit panel is scrolled to the right and earlier sections of a line are hidden from view, coloring can sometimes be incorrect.

## Copying and moving objects

---

Use the `COPY` and `MOVE` primary commands to copy and move QMF objects of the same type (SQL query or QMF procedure) into the data currently being edited.

### About this task

All moved objects are copied and then erased from the database. A confirmation prompt is displayed if `CONFIRM=YES` is set in your QMF profile.

All the copied object data records are read from the database. The attribute values of the copied object are NOT copied from the database. The attributes of the edited object remain unchanged.

### Procedure

1. Use the A (after) or B (before) line command or the `AFTER` or `BEFORE` keyword, along with a label to specify the location where the copied or moved data is to be placed in the edited data.
2. Specify the object name as a command parameter on the command line or on a prompt panel if you enter the command without a parameter.
3. Unqualify or qualify the object name with the object owner. If the owner or the name contains special characters, they must be enclosed by double quotation marks.
4. If there is insufficient space to enter an object name on the command line or the prompt panel, you can use the `ISPF EXPAND` command.
5. To copy QMF object `BOYD.AJB2` into the edit data where a location was properly specified:

```
Command ==>> copy boyd.ajb2
```

Copies the object `boyd.ajb2` into the specified location. The object owner defaults to the current authorization id.

6. To move QMF object `PROC.1` into the edit data where a location was properly specified: The object `PROC.1` is erased from the database. A confirmation prompt is displayed if `CONFIRM=YES` is set in your QMF profile.

```
Command ==>> move "proc.1"
```

Moves object `proc.1` into the specified location before it erases the object from QMF.

### Results

**Note:** These commands operate ONLY on QMF objects and always copy the complete QMF object. They do NOT operate on TSO data sets or members.

## Creating and replacing objects

---

Use the CREATE and REPLACE commands (can be abbreviated to CRE and REPL) to create and replace objects in the database with the object data currently being edited.

### About this task

You can only create and replace objects of the same type (SQL query or QMF procedure) from the object data currently being edited.

The confirmation prompt is not used when the objects are written to the database by using these commands.

To specify the lines that are used to create or replace the object, use:

- Line commands C or CC or else a range operand to specify lines.
- Line commands M or MM to specify lines that are then deleted.

The current attribute values of the edited object are written back to the database with the newly created or replaced object.

### Procedure

1. Specify the object name as a command parameter on the command line or on a prompt panel if you enter the command without a parameter.
2. Unqualify or qualify the object name with the object owner. If the owner or the name contains special characters, they must be enclosed by double quotation marks.
3. If there is insufficient space to enter an object name on the command line or the prompt panel, you can use the ZEXPAND command.
4. To create an object `newproc` in the database, where a range of lines was properly specified:

```
Command ==> create newproc
```

Creates the object `newproc` and saves it in the database. The object owner defaults to the current authorization id.

5. To replace the existing object `old.query` in the database, where a range of lines was properly specified:

```
Command ==> replace "old.query"
```

Replaces the object `old.query` in the database.

### Results

**Note:** These commands operate ONLY on QMF objects and always copy the complete QMF object. They do NOT operate on TSO data sets or members.

## Listing objects and Db2 tables

---

You can create a list of Db2 table owners, Db2 tables, QMF queries, QMF procedures, or all objects and tables using the **LIST** command or by selecting an option from the **List** menu.

### About this task

The list display is a scrollable table display that shows further information about each item. From the list you can select items to display more detailed information or to browse table or object data.

The **LIST** command accepts up to two optional parameters: a **type** parameter and an object **argument** parameter. If no parameters are entered, then a prompt panel is displayed. If no parameters are entered

on the prompt panel, then a default list of all QMF objects and Db2 tables visible to you will be displayed. Only objects that are visible to your authorization id can be listed.

- The **type** parameter determines the type of object to list. Valid values are OWNERS, QUERIES, PROCS, PROCEDURES, TABLES or ALL. A one-character abbreviation of any of these values is also valid.
- The **argument** parameter determines the list argument. Use the QMF pattern characters to specify a subset of matching objects of the type specified in the **type** parameter. The ALL parameter specifies that any owner and object name are included. This is also the default if an **argument** parameter is not specified. For a list of table owners, only pass one qualifier.

**Note:** When the object name is unqualified, the list operation uses the default owner value found in the DSQEC\_LIST\_OWNER global variable. If this is blank, then the value of DSQAO\_CONNECT\_ID is used instead. The default initial list sort order follows the current value of the DSQDC\_LIST\_ORDER global variable. The current values of DSQEC\_LIST\_OWNER and DSQDC\_LIST\_ORDER can be updated using the QMF Enhanced Editor Settings dialog.

## Procedure

1. Enter a **LIST** command from the Enhanced Editor command line or select a list option from the **List** menu.

For example:

- The command below lists all Db2 table owners starting with A.

```
Command ==> list owners A%
```

- The command below lists all QMF SQL queries.

```
Command ==> list q
```

- The command below lists all Db2 tables belonging to the owner boyd.

```
Command ==> list tables boyd.%
```

- The command below lists all QMF objects and Db2 tables belonging to you that start with B.

```
Command ==> list all B%
```

2. Press END to cancel the command.

## Updating object attributes

---

You can view and update the associated attributes of the QMF object that you are editing on the Object attributes panel.

### About this task

QMF object attributes are properties such as owner, name, comments, folder or share option, that are associated with QMF SQL queries or QMF procedures.

Enter the DESCRIBE command on the command line or select the **Describe Object** option from the **Actions** menu to view and update the associated attributes of the QMF object.

## Procedure

1. To update object attributes:
  - a) Navigate to the attribute input field.
  - b) Enter new data.
  - c) Press ENTER.

- d) Enter the END command to complete the update or enter the CANCEL command to exit without updating.
- e) The object type and subtype remain constant during the edit session and cannot be updated.

**Note:** Any attribute changes that you make are saved back to the database ONLY when the object you are currently editing is saved back to the database.

2. You can update the object name and owner of the currently edited object on the **Object attributes** panel. The object name at the top of the editor panel is also updated. If you save the object, the new object name is saved automatically .
3. Objects must always have an owner. If you clear the object owner attribute field, the owner defaults to your QMF connect ID instead.
4. Objects can be unnamed. This can occur either because you are editing an unnamed object in QMF temporary storage or because you cleared the name attribute field of the object. Unnamed objects are saved back only to the QMF temporary storage area and not back to the database. An unnamed object cannot be saved with associated attributes such as comments, share option, or folder name.

## Running objects

---

You can run the QMF object that you are currently editing by entering the RUN command on the command line or selecting the **Run object** item from the **Actions** menu. Both SQL queries and QMF procedures can be run.

### About this task

You do not need to save the object data or specify the object type to do it. QMF uses the object data on the edit panel.

### Procedure

1. Enter the RUN command on the command line or select the **Run object** item from the **Actions** menu. When the QMF RUN command is run successfully, QMF displays the current Report in a QMF mini-session where you can browse the results. Press the END key to return to the edit panel.
2. If the run was not successful, a diagnostic message opens. Where it is possible, the cursor is set to the line and column position of the error. Enter the QMFHELP command to obtain more error information from QMF.
3. To display the QMF command prompt, enter the RUN ? command.  
If there are variables in your query or QMF linear procedure, QMF prompts you for variable values.

## Saving objects

---

The QMF Enhanced Editor automatically saves the QMF object data and associated attributes at the end of the edit session if you updated the data or an attribute.

### About this task

The save type will depend on the exit behavior setting found on the Enhanced Editor settings dialog. The setting can be:

<i>Table 49. Save types</i>	
Option	Description
I (ISPF style)	Data is saved back to both the QMF temporary storage area and the database automatically.

<i>Table 49. Save types (continued)</i>	
<b>Option</b>	<b>Description</b>
Q (QMF style)	Data is only saved back to the QMF temporary storage area. You must use the <b>SAVE</b> command in the Enhanced Editor or QMF to save changes back to the database.

If you do not update the object data or attributes, the edit session ends without saving. If the object has a name, the object and any associated attributes, such as comments or share option, are saved back to the database. QMF temporary storage is updated to display the object data. A confirmation prompt is displayed if your QMF CONFIRM profile option is set to YES.

If the object has no name, the object data is NOT saved back to the database, although QMF temporary storage is still updated to display the object data. Any other associated object attributes cannot be saved and are ignored.

To save the current object and specify the object name:

## Procedure

1. Enter the SAVEAS command with or without an object name parameter or select the **Save as** item from the **File** menu.
2. You can provide a new object name or use the existing object name. A confirmation prompt is presented if your QMF CONFIRM profile option is set to YES. If the SAVEAS command is successful, the current object owner and name are changed to that of the saved object.
3. To save the current object without ending the edit session, enter the SAVE command or select the **Save** item from the **File** menu.
4. To end the edit session and save any changes, enter the EXIT command or select the **Exit** item from the **File** menu.
5. To end the edit session without saving any changes, enter the CANCEL command or select the **Cancel** item from the **File** menu.

## Primary commands

The following primary edit commands are new commands that were added for the QMF Enhanced Editor or existing ISPF commands that were modified to behave differently for QMF objects. ISPF commands that do not appear in this list remain unchanged.

<i>Table 50. QMF Enhanced Editor commands.</i>		
<b>Command</b>	<b>Parameter</b>	<b>Description</b>
COLORS		Displays Color settings panel.
COLORS	OFF	Removes coloring.
COLORS	DEFAULT	Resets coloring to default values.
COPY	<object_name>	Copies QMF object into edited object.
CREATE	<object_name>	Creates a QMF object from edited object.
DESCRIBE		Displays Object attributes panel.
EESET		Displays the Enhanced Editor Settings panel.
FORMAT		Displays Format settings panel.



Table 50. QMF Enhanced Editor commands. (continued)

Command	Parameter	Description
FORMAT	DEFAULT	Resets formatting to default values.
LIST	type argument	Displays a list of QMF objects or Db2 tables.
MOVE	<object_name>	Moves QMF object into edited object and erases it from database.
NEXT		Fastpath to next object in the parent list.
PREV		Fastpath to previous object in the parent list.
PREVIEW		Previews query object results.
QMF	QMF command	Executes a QMF command from the ISPF command line.
QMFHELP		Displays QMF Help.
REPLACE	<object_name>	Replaces QMF object with edited object.
RUN		Runs the edited object.
RUN	?	Runs the edited object with the QMF command prompt.
SAVE		Saves edited object.
SAVEAS	<object_name>	Saves edited object to alternative name.
SIZE		Displays Preview size panel
SIZE	DEFAULT	Sets preview area size to default value.
SYNTAX		Checks syntax of query objects.

## Useful notes

Review the following miscellaneous notes:

### Using special characters in object names

The ISPF editor command line can fail to accept some character sequences when you specify a QMF object name as a parameter for the COPY, MOVE, CREATE, REPLACE, and SAVEAS commands. Combinations of strings that are enclosed in quotation marks or special characters that cannot be parsed on the ISPF command line can cause the procedure failure. You can bypass this restriction by entering a blank parameter to launch the editor object name prompt panel and entering the object name there.

### Saving empty or zero-length objects

Use the standard QMF SAVE command to save empty (zero-length) objects by automatically saving a blank line to the QMF catalog. The QMF Enhanced Editor re-creates this behavior by also adding a blank line when it saves any objects that have no data records.

## The QMF temporary storage area

The QMF temporary storage area contains data for the current query or procedure object. This area is updated by the SAVE command or when you end the editor after data changes. If the object is named, the QMF catalog is also updated. The CREATE, MOVE, REPLACE, and SAVEAS commands read and write only to the QMF catalog and do not update the QMF temporary storage area.

## Editing large objects

When editing very large queries or procedures, it is possible that a specification will fail, either due to insufficient TSO user region, or to a QMF failure. A message is displayed warning you that the current object data is too large to be processed. If an object fails to save for this reason, it is strongly suggested that you attempt to immediately resave the object again with less data to avoid the possibility of a truncated or otherwise corrupted object remaining in the QMF catalog.

## Table display area

---

The QMF Enhanced Editor uses a table display to display data in tabular format when the preview, list and assist commands are used. The table display allows users to navigate through displays and perform most functions using a mouse.

The table display presents data in tabular format using rows and columns.

- The number of rows and columns present in the data is indicated in the top right-hand corner of the screen.
- The last column and row in the table data is marked with the <End> indicator.
- Directly below the command input field, a < symbol indicates which display you will return to when exiting this display. You can also select this field to exit the display.
- On the same line, the valid line commands that can be used on this table display are listed. The first line command shown is always the default line command and will be invoked when the mouse or enter key alone is used to select a particular table row.

For Assist displays, the edit data from the edit line that contains the assist target word is displayed along with the preceding and following edit lines at the top of the table display. Table data is normally displayed without leading and trailing blanks.

The table display area can display up to 100 rows and 250 columns of table data. The display can be scrolled up and down and left and right using the standard ISPF commands. The left and right scroll behavior is determined by the Shift input field which accepts a numeric value (in columns) or PAGE or MAX values.

Use the fastpath methods << Prev and Next >> to quickly move to the next item in a parent list without exiting the display. For example, after using Assist to display a list of Db2 tables you have chosen to browse the Db2 data for a particular table. You can then use fastpath to browse the table data for the next table in the parent list without first returning to the parent list.

Table data can be sorted dynamically at any time by using the mouse or cursor to select a column name and pressing Enter. The sort direction will alternate between ascending and descending sequence. A small indicator will appear in front of the column name to display the current sort sequence. Only one table column can be sorted at a time.

### Note:

- When lists of Db2 tables or QMF objects are shown using the table display, the values of QMF Globals DSQDC\_LIST\_ORDER and DSQEC\_LIST\_OWNER are used for the initial sort order and default owner, respectively. The values of these QMF Globals can be updated using the Enhanced Editor settings panel by entering the **EESET** command or selecting the **Enhanced Editor settings** option from the **Settings** menu.
- If your table data is too large to be displayed in a cell (and is truncated), position the cursor on the data cell and enter the **ISPF EXPAND** command. The PF4 function key is often set to the **ISPF EXPAND** command.

## Previewing query results

---

Use the **PREVIEW** command or select the **Preview report** option from the **Actions** menu to preview up to 100 rows and 250 columns from the result set generated by the current query data, without leaving the Enhanced Editor.

### About this task

The preview function is available only when you are editing a QMF query object and the report is displayed on the edit panel.

**Note:** The preview function can be used only for SQL queries and cannot be used with SQL statements that change the database. Previews are inhibited where certain SQL keywords in the data indicate that the database might be changed when the object is run.

### Procedure

1. Enter the **PREVIEW** command on the command line or select the **Preview report** item from the **Actions** menu.
2. QMF runs the **RUN** command and, if successful, diverts the output to the preview area of the screen. Data is presented in a similar manner to a default QMF Report. However, each column can display only up to a maximum of 70 characters of data.
3. If the **RUN** was not successful, a diagnostic message opens. Where possible, the cursor is set to the line and column position of the error. Enter the **QMFHELP** command to obtain more error information from QMF.
4. If QMF finds variables in your query, it prompts you for variable values .

## Using Assist

---

The Assist facility allows you to easily display more detailed information about SQL identifiers and other keywords that appear in your QMF queries and procedures.

### About this task

For example, if you select a Db2 table name, Assist displays the columns belonging to that table. If you select a QMF keyword, Assist displays the QMF help data for that keyword.

If you position the cursor on the character immediately ahead of the target word and invoke Assist, a list of alternatives to the target word are displayed. For example, if you select a table name, Assist displays other tables belonging to that owner. If you select a QMF or SQL keyword, Assist displays other keywords of a similar type.

Assist will display data on a table or browse panel depending on the data type. The target word you select is always shown in context at the top of the Assist panel. The edit data from the edit line that contains the target word is shown along with the preceding and following edit lines. The target word is shown in reverse green unless you have selected ahead of the word (in which case, the target word is shown in reverse red to indicate that the word can be replaced by any of the items from the first column in the list by using the **R** replace line command.

To setup the method by which you invoke Assist, enter the **EESET** command or select the **Enhanced Editor settings** option from the **Settings** menu. Assist is invoked by positioning the cursor on or just ahead of a target word in the edit data and pressing the key you have indicated in the Settings dialog. Available options are:

- The Enter key (alternatively, you can click the target word)
- The Help function key (alternatively, you can enter the **HELP** command)
- Inhibit (turns the Assist feature off)

In a query, the target word can be an SQL identifier such as a Db2 table owner, table name, column name, column value or an SQL keyword or function. In a procedure it can only be a QMF command.

**Note:**

- Assist cannot be invoked by selecting a blank character. You must select a non-blank character in your target word to invoke Assist. If the target is a blank value, you can select a surrounding quote.
- Browsable Db2 table data is limited to the first 100 rows and 250 columns. Db2 tables containing unsupported data types XML, CLOB and BLOB cannot be browsed. Up to 1000 distinct values can be listed on a column value display.
- Help information for some SQL keywords and REXX keywords is not included in the QMF Enhanced Editor.
- Assist is primarily designed for use within SQL SELECT statements. Other SQL statements might not always produce correct results. Assist does not access Db2 data when analyzing an identifier that you have selected. This means that ambiguities can arise in multi-table queries (or in other intricate queries) where aliases are not used and Assist might subsequently fail to correctly identify the parent identifier of a selected token.

## Formatting queries

---

You can format the SELECT statements in queries by using the Q (Query format) line command in the prefix area of the edit panel.

### About this task

Format settings determine how query formatting must behave and can be customized to your requirements and saved in your ISPF profile for future sessions. The formatting process uses the settings that you selected and rearranges the SQL by inserting new lines where necessary.

### Procedure

1. Use the query format line command to target one, several, or all the lines in the editor for formatting. For example, use Q to target one line, QQ (blocked format) to identify a range of lines, or Qnn to target a fixed number of lines.
2. To display or update format settings, enter the FORMAT command on the command line or select the **Format query** item from the **Settings** menu.
3. On the format settings panel, you can select the format options that you need, including indentation and line width.
4. To reset format settings to the initial default scheme, enter the FORMAT DEFAULT command or select the **Defaults point-and-shoot** field on the **Format settings** panel.
5. To save updated format settings to your QMF ISPF profile, from the Format settings panel, enter the END command or enter CANCEL to exit without saving.

### Results

**Note:**

- Formatting can fail if too much data is passed. An error message is displayed if it happens.
- Strings or identifiers that are too long to be formatted by the current format settings are written to a new line without indentation and wrap through as many complete edit lines as necessary.
- Formatting is used with SQL SELECT statements. Other common SQL statements are recognized but might not format correctly. These SQL statements are treated the same as a SELECT clause.
- SQL comments in '--' style cannot be formatted and are moved to their own line.

## Syntax-checking queries

---

You can use the SYNTAX command to check the query syntax.

### About this task

The SYNTAX command detects only SQL syntax errors and does not perform semantic verification such as checking that column names exist in specified table names.

**Note:** Syntax checking is designed to support SELECT statements and other common SQL statements, not all SQL statements are supported.

### Procedure

1. Enter the SYNTAX command or select the **Check syntax** option from the **Actions** menu to check the syntax.
2. If the syntax check fails, a diagnostic message opens on the panel. Where possible, the cursor is set to the line and column position of the error.



---

## Chapter 16. Multicultural support in QMF

QMF supports bilingual commands, bilingual forms, and double-byte character set (DBCS) data. Ask your administrator whether you have the proper hardware and software necessary to operate with DBCS.

### Bilingual command support

---

With the exception of the Uppercase feature, a QMF National Language Feature (NLF) is a non-English version of QMF. When running a QMF NLF, you can issue QMF commands in the NLF language or in English by setting the DSQEC\_NLFCMD\_LANG global variable.

When you set this variable to a value of 1, the QMF panels appear in the presiding language, but only English commands are accepted. Additionally, when the variable is set to English, any NLF session can run a procedure that is written in English (if all the QMF commands in the procedure are in English). This increases the portability of procedures among the various NLFs by providing a common language for QMF commands.

You can use the following English commands in any QMF NLF session without switching to English:

- INTERACT
- MESSAGE
- GET GLOBAL
- SET GLOBAL

#### Related concepts

Writing bilingual applications You can write one English application and run it in any national language that QMF supports.

#### Related reference

Global variables that control how commands and procedures are executed DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

### Exporting and importing bilingual forms

---

If you are using a QMF NLF, you can choose to export a form in either your own language or in English. If you export a form in English, you can translate it into the NLF language when you import it. This allows portability of forms among the different NLFs.

For example, if you create a form in a French NLF, you can export it in English, and then import it into a Spanish NLF.

You use the LANGUAGE parameter of both the EXPORT and IMPORT commands to specify whether a form is exported or imported in English or in the current session (non-English) language. QMF handles all the necessary translations.

#### Related reference

EXPORT in CICS Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

EXPORT in TSO The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

IMPORT in CICS The IMPORT command copies the contents of a CICS data queue into QMF temporary storage or into the database.

IMPORT in TSO The IMPORT command copies the contents of a TSO data set or UNIX file into QMF temporary storage or into the database.

## Working with double-byte character set (DBCS) data

---

In languages that use a double-byte character set (DBCS), the internal representation for each character requires two bytes of storage. Writing systems such as Kanji and Chinese require such double-byte representations.

In some cases, the Katakana writing system is considered a single-byte character set (SBCS) because it can be represented internally in single bytes. English, German, and French languages fit the category of single-byte character sets.

References made in this topic to "mixed" data mean that strings of DBCS data and strings of SBCS data appear in one data field. When data is mixed, the DBCS data is preceded by an SO (shift out) delimiter character and followed by an SI (shift in) delimiter character.

### How DBCS data looks when displayed

DBCS data differs from SBCS data when it is displayed. It occupies twice as much space on the screen as SBCS data. When double-byte characters are displayed, the SO and SI characters take up one space each.

If you are using a display device that supports DBCS data, you can choose to display the SO and SI delimiters in your data or make them appear as spaces.

When QMF displays DBCS data in the Table Editor, it adjusts the length of the input field for a column to allow for the SI and SO characters. This is especially evident in the Show Field window, where QMF inserts an SI or SO character (or both) on every line of the window. If you type over the SI and the SO characters, you can create an error condition.

You can display any QMF objects that contain DBCS data from the QMF Database Object List with or without a DBCS display device. However, if the object name contains double-byte characters, and you have a non-DBCS display device, the double-byte characters are not displayed correctly. When you enter the DISPLAY command next to the DBCS object you want to display, clear the rest of the row by pressing the Clear key before you press Enter. (If you are using a terminal emulator to simulate the 3270 environment, right-click on the session you want to cancel to display a set of key names.) When a DISPLAY TABLE command is directed to a Unicode database and the table referenced in the command contains columns with graphic data types, QMF might automatically convert the graphic data types to character data types in certain situations to avoid incompatibilities between coded character set identifiers (CCSIDs). A coded character set identifier (CCSID) contains all of the information necessary to assign and preserve the meaning and rendering of characters through various stages of processing and interchange.

Although QMF can display DBCS data in the Table Editor on a non-DBCS display device, you cannot change the data. If you want to change DBCS data by using the Table Editor, you must use a display device that supports DBCS data.

#### Related concepts

[How QMF recasts certain data types when displaying data](#) When a DISPLAY TABLE command is directed to a Unicode database and the table referenced in the command contains columns with graphic data types, QMF converts the graphic data types to character data types.

### How DBCS data changes the length of names and fields

Generally, when you use double-byte characters in QMF, you enter fewer characters than when you use only single-byte characters.

#### Procedure

- To calculate the length of names and fields containing only double-byte characters (assuming that each double-byte character is represented internally by two bytes):
  - a) Count the number of single-byte characters possible.
  - b) Subtract 2 characters, one each for the SO and SI delimiters that are automatically generated.



- c) Divide the remaining number of characters by 2 to get the number of double-byte characters the name or field can contain. If the number is odd, drop the remainder after doing the division.
- To determine whether a name or field can contain a particular mix of double-byte and single-byte characters, use a similar process. First, for each string of double-byte characters in the name or field:
  - a) Count the number of double-byte characters in the string.
  - b) Multiply the number of double-byte characters by 2.
  - c) Add 2 (one each for the SO and SI delimiters).
  - d) Add the sums from all the individual strings of double-byte characters, then count the number of single-byte characters.
  - e) Add the number of single-byte characters to the sums of the double-byte characters.  
The total cannot exceed the maximum length of the name or field that is stated for single-byte characters.

### **Related concepts**

#### DBCS data and QMF objects

Double-byte character set (DBCS) data adheres to different rules depending on where it is used.

#### Data types you can use for DBCS data

You can save DBCS data in your database if you define the columns in which you save the data as character or graphic.

## **Data types you can use for DBCS data**

You can save DBCS data in your database if you define the columns in which you save the data as character or graphic.

Whether you save your DBCS data in character or graphic columns depends on your needs:

- If the column contains DBCS data strings and SBCS data strings, or if it contains a string with both DBCS and SBCS data, define the column as character.
- If the column contains only DBCS data, define the column as character if the SO and SI delimiters must be saved in the database with the double-byte characters. Otherwise, define the column as graphic.

### **Standard data types for DBCS data**

QMF can save DBCS data in database columns that are defined with these data types:

- Character

When surrounded by single-byte, single quotation marks, DBCS data can appear in columns with a character data type. QMF also allows DBCS data strings that are mixed with SBCS data strings. Use this data type if all entries in the column have the same length, up to a maximum of 255 bytes.

- Graphic

QMF can put only DBCS data of a fixed length into columns that are defined with a graphic data type. Use this data type if all entries in the column have the same length, up to a maximum of 127 double-byte characters.

- Variable character

Use this for variable-length entries of up to 32,704 bytes. When surrounded by single-byte, single quotation marks, DBCS data can appear in columns with a variable data type. QMF also allows DBCS data strings that are mixed with SBCS data strings.

- Variable graphic

QMF can put only DBCS data of variable length up to 16,352 DBCS characters into a column that is defined with a VARGRAPHIC data type.

## LOB data types

By default, QMF assigns an edit code of M to any columns that contain BLOB, CLOB, or DBCLOB data. The ability to change the edit code for LOB data is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default.

Unless the DSQEC\_LOB\_RETRV global variable is set to 2, the following edit codes are valid for LOB data:

- For CLOB data, you can specify any edit code that is valid for character data.
- For BLOB data, you can specify B, BW, X, or XW.
- For DBCLOB data, you can specify G or GW.

To display LOB data that is longer than the column width, specify edit codes that allow column wrapping, as follows:

- For CLOB data, set the column width on FORM.MAIN or FORM.COLUMNS to a value of up to 32767 and specify the CW edit code.
- For BLOB data, set the column width on FORM.MAIN or FORM.COLUMNS to a value of up to 32767 and specify the BW or XW edit code.
- For DBCLOB data, set the column width on FORM.MAIN or FORM.COLUMNS to a value of up to 16383 and specify the GW edit code.

If you are working with LOB data and you receive out-of-storage errors while using an edit code other than M, you can change the edit code to M to clear the error and display the report.

Be aware of the following restrictions when you work with LOB data types:

- By default, three-part names cannot be used to access remote tables that contain LOB data. However, you can set the DSQEC\_LOB\_RETRV global variable to 2 or 3 to access LOB metadata or data with a three-part name. Or, you can use the CONNECT command to connect to the database, and then run the query to access the remote table.
- You can use the EDIT TABLE command to edit a table or a view that contains columns with large-object data types; however, you cannot change columns that have CLOB or DBCLOB data types.
- Large-object data cannot be passed in a parameter on the CALL statement.

### Related concepts

Edit codes An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

## DBCS data and QMF objects

Double-byte character set (DBCS) data adheres to different rules depending on where it is used.

### Using DBCS data in commands and procedures

You must issue QMF commands in English, which uses a single-byte character set. However, you can write the following parts of commands and procedures by using double-byte characters:

- Substitution variable names and values
- Comments
- Object names

Object names are the names you supply with commands such as CONVERT, DRAW, and DISPLAY. If your database manager specifically supports double-byte characters in table names, you can use double-byte characters in object names only if you surround the characters with SO and SI delimiters and do not include any DBCS character that is represented internally with a single-byte, double quotation mark character (EBCDIC code x'7F').

- Table names

Unless your database specifically supports double-byte characters in table names, table names cannot contain any double-byte characters that are internally represented with single-byte, double quotation marks.

## Using DBCS data in input fields

All QMF input fields allow DBCS data if you are using a DBCS display device.

Your keyboard can lock while you are keying DBCS data. This indicates that you might not have allowed for the SI character at the end of a field. If this happens, press the Reset key and then press Enter to continue. In the Show Field window of the Table Editor and on SQL query and procedure panels, the SI/SO and SI/blank/SO characters are stripped out each time you press Enter. This means that you might have extra space in these input fields after you press Enter.

## Using DBCS data in queries

In queries, the following items can be represented in either double-byte characters or mixed single-byte and double-byte characters:

- Column, table, and query names

Unless your database specifically supports double-byte characters in table names, column names cannot contain any double-byte characters that are internally represented with single-byte, double quotation marks.

- Substitution variable names and values
- Strings enclosed in quotation marks in character data type fields
- Comments
- QBE example elements

The first character must be a single-byte underscore character. Length limits are the same for SBCS or DBCS data, even though a double-byte character is twice the length of a single-byte character.

In queries, graphic strings that you want to enter or compare to graphic data type fields must be in double-byte characters only. A graphic string consists of either a G or N literal, a single quotation mark followed by the double-byte character string, and a single quotation mark at the end.

If you are writing a prompted query with a LIKE operator and enter a left-side value with an N literal, when you issue a CONVERT TO SQL command, the query shows a G instead of the N that you entered.

## Using DBCS in form panels

You can use DBCS or mixed data in form panels as:

- Column headings
- Break text
- Page text
- Final text
- Form names

Double-byte characters can also appear in QMF forms as column labels.

The following descriptions show ways in which DBCS (or mixed) data differs from the SBCS data in QMF forms.

- Report width

The report width shown at the top of the FORM.MAIN panel indicates the width of the current report in character lengths. Indents, SO and SI delimiters, and characters are counted to calculate this width. A single-byte character counts as one position; a double-byte character counts as two positions. SO and SI delimiters count as one position each.

- Column headings

The underscore ( \_ ) character on the form indicates where an SBCS column heading breaks and continues on another line. DBCS data strings can be broken if the underscore used is a single-byte character.

The DSQDC\_COL\_LABELS global variable controls whether the column heading shown in QMF forms defaults to the database label assigned to the column or the name of the column in the table from which it was selected.

- USAGE

Form usage codes must be single-byte characters.

- INDENT

The leading SO value in a column with a graphic data type appears in the indent space. Therefore, the indent value for a graphic data column must be 1 or more when the leading character in the column is SO.

- WIDTH

The width of a column is specified in number of characters. Although a double-byte character is twice as wide as a single-byte character, a character of either type is always counted as one character when calculating the column width.

With mixed SBCS and DBCS data (in columns with a data type of character), the SO and SI delimiters are counted as part of the column width. If you are using double-byte characters in a column with a data type of character, the width of that column shown on FORM.MAIN and FORM.COLUMNS should be 4 or larger. The minimum column width to display one double-byte character is 4.

With DBCS data (in columns with a data type of graphic), the leading SO is not counted in the column width, but the SI character is part of the width. Therefore, the minimum column width shown on FORM.MAIN and FORM.COLUMNS for columns with a data type of graphic is 1.

- EDIT

Edit codes must be entered on the form in single-byte characters. Edit codes that begin with G can only be used with DBCS data. Codes that begin with C can be used with either DBCS or mixed data. The following figure shows how different edit codes operate on DBCS data.

<i>Table 51. How DBCS characters display with different edit codes</i>		
<b>Edit code</b>	<b>Purpose</b>	<b>Effect on display</b>
C	Columns of data defined as character type	The display of a value is unchanged.
CW	Columns of character data you want wrapped	<p>The display of a value is unchanged, but if the value cannot fit on one line in the column, CW tells QMF to wrap the text according to the width of the column. Instead of cutting off the data at the end of the column, QMF puts as much data as it can on a line in the column, then wraps the data onto the next line.</p> <p>When you use the CW edit code for a column that contains mixed data, the minimum width for the column is 4.</p>

Table 51. How DBCS characters display with different edit codes (continued)

Edit code	Purpose	Effect on display
CT	Columns of character data you want wrapped according to blank characters in the text	<p>The value itself is unchanged, but if the value cannot fit on one line in the column, CT tells QMF to wrap the column according to the text in the column. Instead of cutting off the data at the end of the column, QMF fits as much data as possible on a line, interrupts the line when it finds a single-byte blank, and continues wrapping the data on the next line. If a string of data is too long to fit in the column and does not contain a single-byte blank, QMF wraps the data by width until it finds a single-byte blank and can continue wrapping by text.</p> <p>When you use the CT edit code for a column that contains mixed data, the minimum width for the column is 4.</p>
CDx	Columns of character data you want wrapped according to a delimiter	<p>QMF begins a new line in the column each time it sees a special delimiter in the text. For this edit code, x indicates a special delimiter that can be any single-byte character, including a blank. It does not appear in the output.</p> <p>QMF does not permit column wrapping of graphic data according to a delimiter. QMF will wrap columns of mixed data by delimiter if the delimiter is outside the DBCS data string. When you use this edit code for a column containing mixed data, the minimum width for the column is 4.</p> <p>If a string of data is too long to fit in the column and does not contain a delimiter, QMF wraps the data by width until it finds a delimiter and can continue wrapping by it. If a string of data contains multiple successive delimiters, QMF inserts a blank line for each one after the first. For example, if the data contains two delimiters, QMF begins a new line when it gets to the first delimiter, skips a line when it gets to the second delimiter, and then continues wrapping the output.</p>
G	Columns of data defined with a graphic data type	The display of the value is unchanged.
GW	Columns of graphic data you want wrapped	The value itself is unchanged, but if the value cannot fit on one line in the column, GW tells QMF to wrap the text according to the width of the column. Instead of cutting off the data at the end of the column, QMF puts as much data as it can on a line in the column, then wraps the remaining data on subsequent lines.

Table 51. How DBCS characters display with different edit codes (continued)

Edit code	Purpose	Effect on display
Uxxxx and Vxxxx	Custom edit codes defined at your site	<p>These codes format data in ways that are custom-defined by your site. Replace xxxx in this code with an identifier that names a unique code. See your administrator for a description of your available custom edit codes.</p> <p>You cannot use these edit codes to format data containing XML, BLOB, CLOB, and DBCLOB data. User-defined edit codes can be used to format DECFLOAT data only if the processor on which QMF is running supports decimal floating-point instructions.</p>
M	Displays the data type and length of the column instead of the data itself	The data is not displayed.

When an SO or SI character is missing from a DBCS data string, the existing SO or SI character appears as a question mark. All other double-byte data appears as single-byte characters and is meaningless.

QMF breaks displayed DBCS data at a field or a screen boundary in a way that avoids splitting double-byte characters. Scrolling is necessary to view the characters on the truncated lines. SO or SI delimiters are added where these breaks occur. Set the scrolling value to less than the screen width on report panels and QBE query panels to ensure that you do not miss characters that are out of the regular screen view.

#### Related concepts

[Query-by-Example](#)

[Edit codes](#)An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

#### Related reference

[FORM.COLUMNS](#)Use FORM.COLUMNS to make choices about the uses of the columns. What you specify on FORM.COLUMNS is reflected on FORM.MAIN.

[Global variables that control various displays](#)DSQDC global variables control the display of certain kinds of information. All of these global variables can be modified by the SET GLOBAL command.

## Exporting and importing DBCS data

You can export data that is defined as graphic and variable graphic.

The column width of exported data is the number of double-byte characters in it, which is half the number of bytes that are used to store it. Column data is stored in the data record exactly as it comes from the database, except that SO and SI delimiters are added.

DBCS data can be imported in queries, procedures, and forms. When importing DBCS data in this indirect way, be certain that the record length does not exceed 79 bytes. Also be sure that the data is enclosed in SO and SI delimiters. Data that does not meet these requirements appears as meaningless single-byte characters.

You can also import DBCS data by using the IMPORT DATA command. QMF validates the data as it is imported. If the DBCS data is not valid, the import stops.

#### Related reference

[Exporting data objects and database tables](#)You can export data and table objects in the QMF, Integrated Exchange Format (IXF), XML, or comma-separated value (CSV) format. The QMF format is the default.

EXPORT in CICS Use the EXPORT command in CICS to send queries, forms, procedures, reports, and data from QMF temporary storage to a CICS data queue.

EXPORT in TSO The EXPORT command sends queries, forms, procedures, reports, tables, data, and charts to certain data sets and files.

IMPORT in CICS The IMPORT command copies the contents of a CICS data queue into QMF temporary storage or into the database.

IMPORT in TSO The IMPORT command copies the contents of a TSO data set or UNIX file into QMF temporary storage or into the database.

## Printing DBCS reports

With a DBCS printer, you can print reports that contain DBCS data even if you do not have a device that is capable of displaying DBCS data.

Your QMF administrator can start QMF using the DSQSDBCS program parameter to allow this support.

You can also print any objects that contain DBCS data from the Database Object List panel whether or not you have a DBCS display device. However, if the object name contains double-byte characters, and you have a non-DBCS display device, all double-byte characters are altered. When you enter the PRINT command next to a DBCS object, clear the rest of the row for that object before pressing Enter.

If you are printing DBCS data and QMF splits the page, printing on the second and subsequent pages of the report resumes at the fourth byte position from the left side of the page.

### Related tasks

Printing DBCS data from non-DBCS display devices If you use the Uppercase, Japanese, or Korean National Language Feature (NLF), you might need to print double-byte character set (DBCS) data.





---

## Chapter 17. QMF with other products

You can use QMF with other products to enhance the manipulation and reporting of data. By using other products with QMF you access a broad range of functions and services.

You can access other products from the QMF home panel or from any other QMF panel. When you do, you might see panels that are not QMF panels. For example, if you use ISPF, you might see the primary option menu in the Interactive System Productivity Facility-Program Development Facility (ISPF-PDF). Using other products from QMF has no effect on any other QMF operation and does not disrupt the normal sequence of events. When you exit the product, you return to QMF at the point where you left it.

---

### Edit objects external to QMF with ISPF

You can edit an existing QMF procedure or SQL statement from QMF. The QMF object that you edit can be a new, changed, or imported procedure or query. You cannot edit QBE and prompted queries.

**Restriction:** You cannot use an editor from QMF under CICS. You can, however, change a QMF object while you view it in temporary storage.

QMF supports the ISPF-PDF editor. You can name a CLIST that initializes another editor and optionally performs housekeeping functions. The ISPF-PDF editor is the default editor, but if you want to use the ISPF-PDF editor, you must do one of the following:

- Start QMF as an ISPF-PDF dialog.
- Name a CLIST to set up ISPF and start the PDF editor.

To find out about editors you can use, see your administrator.

### Editing an object by using the ISPF-PDF editor

To use the ISPF-PDF editor, you must be using ISPF.

#### Procedure

To edit an object with the ISPF-PDF editor:

- To display the ISPF-PDF editor and the current query or procedure, enter:

```
EDIT object
```

where *object* is either PROC or QUERY.

- From a **PROC** or **QUERY** panel, you can enter the EDIT command without specifying a value for *object*. The procedure or query displayed on the panel is edited. Entering EDIT ? prompts you with the default *object*, either PROC or QUERY, depending on which type of panel you are using when you initiate the command.

You can edit your SQL statements or procedure in a different ISPF application ID by using a CLIST as the editor name of the QMF EDIT command.

When you end the edit session, you return to QMF with the edited object in QMF temporary storage.

### Editing an object using a CLIST

To use a CLIST, you must be using TSO. The named editor represents your CLIST.

For example, enter the following command, where the editor is named MYCLIST:

```
EDIT object (EDITOR=MYCLIST)
```

In this command, *object* is either PROC or QUERY.

Using an editor of your choice, run this CLIST to edit the current query or procedure.

When you end the edit session, you return to QMF with the edited object in QMF temporary storage.

## Starting an edit session

You can use the EDIT keyword to start an edit session.

### Procedure

To start an edit session:

1. On the QMF command line, enter:

```
EDIT ?
```

The **EDIT Command Prompt** panel displays.

2. Enter either QUERY or PROC to start an edit session.

Another **EDIT Command Prompt** panel appears.

3. Specify the editor that you want to use.

PDF is the default editor.

4. Press Enter.

QMF displays the appropriate panel for the editor you requested. The panel contains the current QUERY or PROC object (the object you last worked on).

5. To return to QMF, exit the editor.

## Using ISPF from QMF

---

To access the ISPF-PDF product from QMF, you must start QMF as an ISPF dialog.

You can access the ISPF-PDF product from QMF in two ways:

- Access the ISPF-PDF primary option menu panel from which you can choose an application.
- Display a specific ISPF-PDF panel.

To access the ISPF-PDF primary option menu panel, enter the following command on the QMF command line:

```
ISPF
```

From the ISPF-PDF primary option menu panel, you can start whatever applications you normally use in ISPF. All available command options appear on the menu. You can select any of them by entering a letter on the command line or by pressing a function key.

To display a specific ISPF-PDF panel, enter the panel identifier as a parameter to the ISPF command. For example:

```
ISPF 3
```

This starts the application identified as **Option 3** on the ISPF-PDF primary option menu panel. The specific panel you see depends on how ISPF has been configured.

To return to QMF, exit ISPF-PDF.

## Inserting a QMF report in a document

---

In an edit session, you can insert a QMF report into the document you are editing without leaving the session.

**Restriction:** You cannot use the document interface from QMF under CICS.

Use the GETQMF macro to insert the report. The GETQMF macro is not a QMF command.

You can insert an existing QMF report into a document or generate a new QMF report by using QMF either interactively or through the command interface. You can also format the QMF report by using SCRIPT/VS control words that are used by the Document Composition Facility (DCF).

Before you insert the QMF report into a document, you must print it from within a QMF session.

The syntax of the GETQMF macro is:

```
GETQMF type option
```

In this statement, *type* specifies whether SCRIPT/VS control words are also inserted.

**DCF**

For a SCRIPT/VS document

**ASIS**

For inserting a QMF report "as is"

The *option* parameter specifies whether you are creating a new report or inserting an existing one.

**USEQMF**

To create a QMF report dynamically

**DSN**

To insert an existing QMF report

**Related concepts**

[Formatting the report](#)

The *type* parameter of the GETQMF statement specifies whether you want your report formatted for a DCF document or left as it is.

[Specifying whether to insert a new or existing report](#)

You can insert a new or existing QMF report into another document.

## Formatting the report

The *type* parameter of the GETQMF statement specifies whether you want your report formatted for a DCF document or left as it is.

### DCF type

The QMF report you identified or produced is inserted into your document with SCRIPT/VS control words. For example, enter the following from your editor:

```
GETQMF DCF USEQMF
```

DCF places SCRIPT/VS control words before and after the QMF report. Additionally, each printer page eject is replaced by a SCRIPT/VS page eject. SCRIPT/VS control words are placed at the heading and footing of each page.

QMF report length and width must be considered when QMF reports are included within a SCRIPT/VS document. Editor settings always override QMF report characteristics. Use the following specifications on the QMF PRINT command:

- Use a length of 56 lines per page.
- The number of characters per line varies with the DCF print arrangement selected. If the report is too wide to fit in the document, it is inserted anyway. However, a warning message is issued, and the lines that are too long to fit are wrapped for ISPF-PDF. Wrapping or truncating occurs only when you are inserting an existing QMF report into a document. When you create a new report interactively in QMF, the lines are not too long.

## ASIS type

The QMF report you identified or produced is inserted into your document with no alterations, "as is". For example, enter the following from your editor:

```
GETQMF ASIS USEQMF
```

ASIS is the default.

## Specifying whether to insert a new or existing report

You can insert a new or existing QMF report into another document.

- The USEQMF option inserts a new report
- The DSN option inserts an existing report

### The USEQMF option

The USEQMF option allows you to insert a new QMF report into another document without leaving your QMF session. You might need to initialize the system environments.

The command you use differs depending on whether or not QMF is active:

- When QMF is not active

If you are using ISPF-PDF or PS/TSO and you want to generate a report from QMF and insert it into the document you are working on, enter the following from your editor:

```
GETQMF DCF USEQMF
```

This causes the GETQMF macro (with the USEQMF option) to start an interactive QMF session. QMF uses a default initial procedure when it starts. When you are in QMF, you have the full interactive capability available to produce your report. After your report is finished, remember to print it using the PRINT REPORT command. QMF issues ISPF messages and does not allow you to leave QMF with the END command until a QMF report is printed. The ISPF messages and associated help panels tell you how to print a report for the document interface and return to your editor.

If you specified a procedure name after USEQMF, it runs as an initial procedure when you start QMF. You must specify an EXIT command in the procedure to end QMF, or you must manually exit from the QMF session. The END command runs the procedure again.

- When QMF is active

If you are using QMF and you want to insert a report into a document outside the QMF environment, you can access ISPF-PDF through the ISPF bridge. Then edit your target document outside the QMF environment. After you start the editor, prepare it to receive the new report in the proper place in the document.

With QMF active, you must enter a QMF procedure name after the USEQMF option. For example, enter the following from your editor:

```
GETQMF DCF USEQMF MYPROC
```

Where MYPROC is the name of a QMF procedure that runs through the QMF command interface and generates a report. If you want to run a shared procedure that you do not own, specify it as *owner.procname*. You must specify USEQMF to use the procedure. To call the document interface, enter GETQMF. If your procedure printed a report, the report appears in your document. You can save the document and return to QMF.

Your QMF session ends if you use a procedure that issues the EXIT command.

You must use a QMF procedure, not a query, to produce your QMF report.

### Related concepts

[Information about your editor](#)

You can insert a QMF report into a document while using ISPF-PDF or PS/TSO.

## Using the DSN option

Use DSN if you want to insert an existing QMF report.

### Procedure

You must follow DSN by the fully-qualified data set name.

For example, enter the following from your editor:

```
GETQMF ASIS DSN dataset_name
```

Where *dataset\_name* is the name of the data set that contains the chart or report to be inserted. Lines in the inserted data might be truncated or wrapped.

You can also create a report interactively and export it to a data set (which becomes an "existing" report) in one step by including USEQMF before the DSN option:

```
GETQMF ASIS USEQMF DSN dataset name
```

The report is then inserted into your document.

## Information about your editor

You can insert a QMF report into a document while using ISPF-PDF or PS/TSO.

### ISPF-PDF

When you use ISPF-PDF, the QMF report is inserted into your document after the line where you enter A, or before the line where you enter B, in the prefix area. If you do not choose a line, the report is inserted at the end of the document. The top line displayed after insertion is the line that immediately precedes the inserted report. This is similar to the ISPF-PDF COPY command.

### PS/TSO

If you are using the Personal Services for TSO Extensions (PS/TSO), you are using the ISPF-PDF editor.

#### Related information

[The IBM Publications Center](#) Search for information about ISPF-PDF.

## Restrictions on the document interface

When you use the document editing interface, keep in mind the certain restrictions.

- When you format a report to insert into a document, you cannot use a GDDM printer nickname. The QMF document interface sets a PROFILE value of PRINTER=' ' if you enter QMF through the QMF command interface or interactively by using the default initial procedure. When you are running your own initial procedure, make sure that your PROFILE setting contains a string of blanks for the PRINTER option. (Alternatively, you can specify it on the PRINT command.)
- You cannot shorten GETQMF, but you can enter its parameters with minimum unique representation. You need two characters (in English). The exception is when you specify USEQMF and DSN instead of a procedure name. In this case, anything other than DSN is taken as a procedure name.
- You cannot nest the document interface.
- Do not use the ISPF-PDF DEFINE command to redefine current ISPF-PDF commands.
- No prompt panel or help panel appears with the GETQMF macro because GETQMF is not a QMF command. If QMF uses the default initial procedure, there are help panels for the document interface messages in QMF.

After you install QMF, and it is running successfully, you need to tailor the document interface.

## Using the QMF document interface

---

Although you might not use all the products and environments, you should look at each one to see the different ways the document interface can be used.

This topic shows examples of inserting QMF reports into documents under two conditions:

- Accessing QMF from an editor
- Accessing an editor from QMF

### Accessing QMF from an editor

There are multiple ways to issue the GETQMF command from ISPF-PDF and PS/TSO.

- Example 1 – From ISPF-PDF:

The USEQMF option specifies that QMF is used to produce the report during the edit session.

1. Choose where you want the document inserted by using the A (after) or B (before) prefix commands.
2. From your ISPF-PDF edit session, enter the GETQMF command:

```
GETQMF USEQMF MYPROC
```

Your screen is blank for a few minutes while MYPROC creates and prints a report.

3. When the QMF object panel appears, enter an EXIT command to exit QMF.

The report is inserted into your document as-is when QMF ends.

Use EXIT to leave QMF; the END command runs the initial procedure again.

- Example 2 – From ISPF-PDF or PS/TSO:

The QMF report data set *userid.MYREPORT* is inserted into the user's document as is.

1. From a document in ISPF-PDF or PS/TSO, insert a prefix command A (after) or B (before) to receive the new report at the proper place.
2. On the command line, enter the GETQMF command:

```
GETQMF ASIS DSN userid.MYREPORT
```

The macro gets the existing report, *userid.MYREPORT*. Then it inserts it into your document and returns you to the ISPF-PDF or PS/TSO editor you were using before you issued the GETQMF command.

- Example 3 – From ISPF-PDF:

The QMF report is produced interactively in QMF.

1. Enter the GETQMF command:

```
GETQMF ASIS USEQMF
```

Your screen is blank for a few minutes while the macro is running.

2. When the QMF home panel appears with a document interface message, produce a report as you normally would in QMF.
3. Alter the form of the report if necessary.
4. Display the report to check it.
5. Enter PRINT REPORT.
6. Enter an END or EXIT command to exit QMF.

The report is inserted into your document, and you return to ISPF where you were before issuing the GETQMF macro.

If you type EXIT on the QMF command line without printing a report, you are returned to ISPF. An error message displays, and no report is inserted.

## Accessing an editor from QMF

When you are using QMF, you can create a report and insert it into a document with PS/TSO.

### About this task

To run this example, use the ISPF command to bridge to ISPF-PDF and define the data set where your target document is located.

### Procedure

From your PS/TSO session:

1. Prepare your document to insert the new report at the correct location in the document.
2. Enter the GETQMF command:

```
GETQMF ASIS USEQMF MYPROC4
```

The GETQMF macro runs the MYPROC4 routine in QMF, and the report is inserted into your document.

3. Save your document.

You return to QMF at the point where you issued the ISPF command.

You lose your QMF session if you use a procedure that issues an EXIT command.

### Related concepts

[Information about your editor](#)

You can insert a QMF report into a document while using ISPF-PDF or PS/TSO.





---

## Chapter 18. Db2 for z/OS stored procedures and QMF

You can run stored procedures from QMF for TSO and CICS.

### Related concepts

[Generate different types of programs](#) Program Generator generates stored procedure programs from QMF queries.

---

### Running a CALL statement from the SQL QUERY panel

When you run a stored procedure from QMF, you enter your CALL statement on the SQL QUERY panel and then use the RUN command as you would when running any QMF query. The database to which the CALL statement is directed must support the ability to call a stored procedure.

Most simple CALL statements use the following format:

```
CALL spname(parm1, parm2, parm $x$ )
```

Example: This basic CALL statement is for a stored procedure that has two input parameters that are defined as decimal data with a precision of five and a scale of two:

```
CALL USERAPROC1(123.45,1.22)
```

### Related concepts

[Viewing the data in the database using SQL statements](#)

You can select and view data in the database by using SQL statements.

### Related reference

**CALL**To run a stored procedure from within a QMF session, you must issue a CALL statement from the SQL Query panel. The database to which the CALL statement is directed must support the ability to call a stored procedure.

---

### Examples of running a stored procedure

You can develop the stored procedures on your system with Db2 QMF High Performance Option User's Guide for TSO and CICS or development tools native to the environment in which you are working.

Details of the stored procedure's input and output parameters and returned result sets are provided in these examples so that you can develop similar stored procedures to follow along with the examples.

QMF does not provide the source code for the stored procedures that are used in these examples.

### Related concepts

[Generate different types of programs](#) Program Generator generates stored procedure programs from QMF queries.

### A stored procedure that returns no output parameters

To run a stored procedure with no parameters, enter the CALL statement on the **SQL QUERY** panel.

#### About this task

The stored procedure that is used in this example performs some Db2 maintenance functions. You can set up your stored procedure to do maintenance functions that are required at your site.

#### Procedure

Issue the CALL statement on the **SQL QUERY** panel:

```
CALL USERA.MYPROC1
```

Upon completion, you see this message:

OK, your stored procedure has successfully completed.

### Related concepts

[A stored procedure that contains input and output parameters](#)

In this example, the USERA.IOVCHAR stored procedure has two parameters (an input and an output parameter), which are both defined as VARCHAR 32 KB character strings. The stored procedure copies the input string to the output string.

[A stored procedure that returns a result set](#)

An example stored procedure that displays a result set after it completes.

[A stored procedure that returns multiple result sets](#)

An example stored procedure that has two output parameters and that returns multiple result sets.

## A stored procedure that contains input and output parameters

In this example, the USERA.IOVCHAR stored procedure has two parameters (an input and an output parameter), which are both defined as VARCHAR 32 KB character strings. The stored procedure copies the input string to the output string.

You can enter your CALL statement with QMF global variables to handle the input and output parameters. The global variables can be declared by using the SET GLOBAL command before you run the CALL statement. If you do not set the value before you run the CALL statement, you are prompted to enter a value. Issue the CALL statement on the **SQL QUERY** panel. For example:

```
CALL USERA.IOVCHAR(&A01, &A02)
```

You enter your data on the lines that are provided in the prompt panel. You must enter a blank for the output parameter regardless of whether you use prompting or you set the global variables before you run the CALL statement. A blank value is shown for the output parameter &A02 in the following figure:

```
SQL QUERY                                MODIFIED LINE 1
                                         1 to 10 of 10
RUN Command Prompt - Values of Variables

Your RUN command runs a query or procedure with variables that need
values. Fill in a value for each variable named below:

&A01          'This is my string'
&A02          ' '

F1=Help  F3=End  F7=Backward  F8=Forward
Please give a variable for each variable name.
```

Figure 279. **Run Command Prompt** panel that shows input and output parameters

Upon completion, you see this message:

OK, your stored procedure has successfully completed.

You can view the output parameter in the global variable pool by using the SHOW GLOBALS command. This command displays the **GLOBALS** panel, which is shown in the following figure:

## GLOBALS

Type a value for a global variable and press Enter or press a function key. Variable values may be changed if they are enclosed in parentheses or brackets.

```
Variable Name:      Value:
-----
A02                ( THIS IS MY STRING          1 to 11 of 113
DSQAO_APPL_TRACE   0
DSQAO_ATTENTION    0
DSQA)_BATCH        1
DSQAO_CICS_SQNAME
DSQAO_CICS_SQTYPE
DSQAO_CICS_TQNAME
DSQAO_CICS_TQTYPE
DSQAO_CONNECT_ID   CACLARK
DSQAO_CONNECT_LOC  zOS1DB2M
DSQAO_CURSOR_OPEN  2

1=Help  2=          3=End      4=          5=Show Field  6=Query
7=Backward 8=Forward 9=Form 10=Add 11=Delete 12=Report
OK, GLOBALS IS SHOWN
COMMAND ==>
```

Figure 280. Viewing the output parameters with the SHOW GLOBALS command

### Related concepts

[A stored procedure that returns a result set](#)

An example stored procedure that displays a result set after it completes.

[A stored procedure that returns multiple result sets](#)

An example stored procedure that has two output parameters and that returns multiple result sets.

### Related tasks

[A stored procedure that returns no output parameters](#)

To run a stored procedure with no parameters, enter the CALL statement on the **SQL QUERY** panel.

## A stored procedure that returns a result set

An example stored procedure that displays a result set after it completes.

In this example, the USERA.RET1RS stored procedure has two output parameters. The first, defined as an integer, returns the SQL code. The second, defined as a character string, returns the SQL state. This stored procedure also returns a result set based on the following SELECT statement from the QMF supplied table Q.STAFF:

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY, COMM FROM Q.STAFF
```

The DSQEC\_SP\_RS\_NUM global variable indicates which result set is used to create the report. Verify that DSQEC\_SP\_RS\_NUM is set to 1. Issue the CALL statement on the SQL QUERY panel:

```
CALL USERA.RET1RS(&A01, &A02)
```

Upon completion, you see this message:

```
OK, your stored procedure has successfully completed.
```

Your result set is displayed, as shown in the following figure; you can page forward or back to view the entire report.

```

REPORT                                     LINE 1   POS 1   79

  ID  NAME      DEPT  JOB   YEARS  SALARY  COMM
-----
  10  SANDERS    20  MGR   7      99999.99  -
  20  PERNAL     20  SALES 8     18171.25 612.45
  30  MARENGHI   38  MGR   5     17506.75  -
  40  O'BRIEN    38  SALES 6     18006.00 846.55
  50  HANES      15  MGR   10    20659.80  -
  60  QUIGLEY    38  SALES -     16808.30 650.25
  70  ROTHMAN    15  SALES 7     16502.83 1152.00
  80  JAMES      20  CLERK -     13504.60 128.20
  90  KOONITZ   42  SALES 6     18001.75 1386.70
 100  PLOTZ      42  MGR   7     18352.80  -
 110  NGAN       15  CLERK 5     12508.20 206.60
 120  NAUGHTON   38  CLERK -     12954.75 180.00
 130  YAMAGUCHI  42  CLERK 6     10505.90 75.60
 140  FRAYE      51  MGR   6     21150.00  -

1=Help   2=      3=End   4=Print  5=Chart  6=Query
7=Backward 8=Forward 9=Form 10=Left 11=Right 12=
OK, this is the REPORT from your RUN command
COMMAND ==>

```

Figure 281. Result set from `USERA.RET1RS`

You can view the output parameters by issuing the `SHOW GLOBALS` command. This command displays a list of all global variables and their values, including those global variables that are provided by QMF and those variables that are defined by users.

### Related concepts

[A stored procedure that contains input and output parameters](#)

In this example, the `USERA.IOVCHAR` stored procedure has two parameters (an input and an output parameter), which are both defined as `VARCHAR 32 KB` character strings. The stored procedure copies the input string to the output string.

[A stored procedure that returns multiple result sets](#)

An example stored procedure that has two output parameters and that returns multiple result sets.

### Related tasks

[A stored procedure that returns no output parameters](#)

To run a stored procedure with no parameters, enter the `CALL` statement on the **SQL QUERY** panel.

### Related reference

[Global variables that control how commands and procedures are executed](#) DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the `SET GLOBAL` command.

## A stored procedure that returns multiple result sets

An example stored procedure that has two output parameters and that returns multiple result sets.

The first output parameter in this example stored procedure, named `USERA.RET3RS`, is defined as an integer and returns the SQL code. The second output parameter, which is defined as a character string, returns the SQL state. This stored procedure also returns three result sets based on the following `SELECT` statements from the QMF supplied tables `Q.STAFF`, `Q.ORG`, and `Q.INTERVIEW`:

```

SELECT NAME, DEPT, SALARY, COMM FROM Q.STAFF
SELECT DEPTNUMB, DEPTNAME, MANAGER, DIVISION LOCATION FROM Q.ORG
SELECT TEMPIDL, INTDATE, STARTTIME, ENDTIME, MANAGER, DISP, LASTNAME,
FIRSTNAM FROM Q.INTERVIEW

```

You can display a result set each time that you run the stored procedure. Indicate the result set to be displayed by setting the QMF global variable `DSQEC_SP_RS_NUM`. A setting of 1 displays the first result

set returned, a setting of 2 displays the second result set returned, and so on. In the following example, DSQEC\_SP\_RS\_NUM is set to 3 to display the third result set returned:

```
SET GLOBAL (DSQEC_SP_RS_NUM=3
```

Issue the CALL statement on the **SQL QUERY** panel:

```
CALL USERA.RET1RS(&A01, &A02)
```

Upon completion, you see this message:

OK, your stored procedure has successfully completed.

The following figure shows an example of a displayed result set:

REPORT	LINE 1	POS 1	79				
TEMPID	INTDATE	STARTTIME	ENDTIME	MANAGER	DISP	LASTNAME	FIRSTNAM
-----	-----	-----	-----	-----	---	-----	-----
400	1996-02-05	13.00.00	15.12.00	270	NOHIRE	FROMMHERZ	RICHARD
410	1996-02-11	15.00.00	16.18.00	10	HIRE	JACOBS	SUSAN
420	1996-04-07	09.00.00	09.58.00	140	HIRE	MOTNEZ	RITA
430	1996-04-24	10.30.00	11.30.00	290	NOHIRE	RICHOWSKI	JOHN
440	1996-03-13	10.15.00	11.23.00	160	HIRE	REID	CATHY
450	1996-09-19	09.45.00	11.00.00	50	HIRE	JEFFREYS	PAUL
460	1996-10-06	14.45.00	16.22.00	100	HIRE	STANLEY	JOHN
470	1996-02-05	16.30.00	18.00.00	270	HIRE	CASALS	DAVID
480	1996-03-13	13.30.00	14.45.00	160	NOHIRE	LEEDS	DIANE
490	1996-09-30	15.00.00	15.44.00	140	NOHIRE	GASPARD	PIERRE

\*\*\* END \*\*\*

1=Help 2= 3=End 4=Print 5=Chart 6=Query  
7=Backward 8=Forward 0=Form 10=Left 11=Right 12=  
OK, this is the REPORT from your RUN command.  
COMMAND ==>

SCROLL ==> PAGE

Figure 282. The result set from USERA.RET3RS

The output parameters can be viewed in the global variable pool by using the SHOW GLOBALS command.

### Related concepts

[A stored procedure that contains input and output parameters](#)

In this example, the USERA.IOVCHAR stored procedure has two parameters (an input and an output parameter), which are both defined as VARCHAR 32 KB character strings. The stored procedure copies the input string to the output string.

[A stored procedure that returns a result set](#)

An example stored procedure that displays a result set after it completes.

### Related tasks

[A stored procedure that returns no output parameters](#)

To run a stored procedure with no parameters, enter the CALL statement on the **SQL QUERY** panel.

### Related reference

Global variables that control how commands and procedures are executed DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## QMF forms with result sets returned from stored procedures

You can change the appearance of the reports that are created from running your stored procedures by changing the report information on the QMF form panels.

### Related concepts

[Customizing your reports](#)

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

---

## Appendix A. Query-by-Example

QBE is a language for querying relational data by graphically representing it.

Db2 for z/OS Version 13 Release 1 increases maximum column name lengths from 30 bytes to 128 bytes. Db2 for z/OS Version 13 Release 1 increase to 128 byte column name support is limited for those applications that use an SQLDA to obtain column name information. The QMF for TSO and CICS application falls into this category. Due to this Db2 for z/OS limitation, column names longer than 30 bytes are truncated. If a column name longer than 30 bytes is present, the QBE command is blocked from usage.

---

### Starting a QBE query

Before you can write a query in QBE, you need to display the QBE Query panel.

#### Procedure

There are two ways to display the QBE Query panel from the command line on the QMF home panel. The method you choose depends on whether you plan to use QBE most of the time or switch back and forth between query languages.

- If you plan to write queries in QBE most of the time, enter:

```
SET PROFILE (LANGUAGE=QBE
RESET QUERY
SAVE PROFILE
```

- If you prefer to have another language set in your profile, you can specify QBE for your current session with the following command:

```
RESET QUERY (LANGUAGE=QBE
```

---

### Running and saving queries

When you enter a RUN or SAVE command from a query panel, you do not need to specify the object type as QUERY. The object type defaults to QUERY when these commands are entered from a query panel.

#### Procedure

When you finish composing your query, you will want to run it and possibly save it.

- To run a query, press the Run function key or enter the command:

```
RUN QUERY
```

- To save a query, choose a name (for example, MYQUERY), and enter the command:

```
SAVE QUERY AS MYQUERY
```

---

### Listing existing queries

You can use the LIST QUERIES command to display your saved queries.

#### Procedure

Enter the following command to display your saved queries:

```
LIST QUERIES (OWNER =userid
```

where *userid* is your QMF user ID. The default for the OWNER parameter of the LIST command is taken from the DSQEC\_LIST\_OWNER global variable. The variable has a blank value by default, which causes QMF to display objects owned by the ID of the current QMF user.

If you want additional information about any command, enter the command name and follow it with a question mark. For example:

```
LIST ?
```

If the DSQEC\_CURR\_FOLDER global variable is set to a folder name when the LIST QUERIES command is executed, only the QMF query objects in that folder are listed. If DSQEC\_CURR\_FOLDER is not set, all query objects are listed.

You can list query objects that are in a specific folder by specifying the FOLDER keyword, or you can use the FOLDER keyword to override the current DSQEC\_CURR\_FOLDER setting. For example, to list queries in the SALES folder, either set DSQEC\_CURR\_FOLDER to SALES or issue the following command:

```
LIST QUERIES (FOLDER=SALES
```

### Related concepts

[QMF commands specific to QBE](#)

### Related reference

[Global variables that control how commands and procedures are executed](#) DSQEC global variables control how commands and procedures are executed. All of these global variables can be modified by the SET GLOBAL command.

## Drawing example tables

In QBE, you create queries in an example table.

Db2 for z/OS Version 13 Release 1 increases maximum column name lengths from 30 bytes to 128 bytes. Db2 for z/OS Version 13 Release 1 increase to 128 byte column name support is limited for those applications that use an SQLDA to obtain column name information. The QMF for TSO and CICS application falls into this category. Due to this Db2 for z/OS limitation, column names longer than 30 bytes are truncated. If a column name longer than 30 bytes is present, the QBE command is blocked from usage.

An *example table* is a framework in which you enter instructions about how you want data presented in your report. (If you have the appropriate authorization, you can also use the keywords D., I. and U. in an example table to make changes to a database.) For example, from a RESET QBE QUERY panel, the Q.ORG example table displays when you issue the following command:

```
DRAW Q.ORG
```

The example table appears as shown in the following figure:

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
| | | | | | |
```

Figure 283. A QBE example table drawn for the Q.ORG table

Within this framework, you can select the columns you want to see by using the P. keyword and other QBE keywords to control the presentation of the report data and make changes to the database.

The program function keys shown at the bottom of the screen make it easier to perform certain functions. Your site may have changed the setting of the function keys. The initial default settings are as follows:

- 1 Shows help information about your last action.



- 2** Runs your query.
- 3** Returns to the QMF home panel.
- 4** Makes the object larger.
- 5** Makes the object smaller.
- 6** Draws an empty example table.
- 7** Moves the display backward.
- 8** Moves the display forward.
- 9** Shows the form panel last used.
- 10** Scrolls the display to the left.
- 11** Scrolls the display to the right.
- 12** Displays your report.

#### **Related concepts**

[ENLARGE command](#)

[REDUCE command](#)

## Displaying all the columns of a table

---

To retrieve data from a table in the database and display it in a report, use the **P.** keyword. You can use the **D.**, **I.**, and **U.** keywords in a similar way to delete, insert, and update data in the database.

### **Before you begin**

1. [Start a QBE query](#)
2. [Draw an example table](#)

### **About this task**

By default, metadata (the data type and length of the column) is displayed for columns with binary, XML, or large object (LOB) data types. Default metadata is also displayed if the table contains columns with decimal-floating point data and the processor on which QMF is running does not support decimal floating-point instructions. For all columns except columns with decimal floating-point data, in QMF for TSO you can change the edit code for the column to display the actual data. However, the ability to change the edit code for LOB data is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default. The same is true in QMF for CICS except for XML data over 32 KB, for which you cannot change the edit code. You can assign the OMIT usage code to columns for which metadata displays to keep them from appearing in your report.

**Restriction:** Different databases have varying levels of support for data types.

### **Procedure**

1. To display the data in all the columns of a table, put **P.** under the table name and do not remove any of the column headings, as shown in the following figure:

```

Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
P.    |           |           |         |         |         |

```

Figure 284. Presenting all columns of the Q.ORG table

2. Enter RUN QUERY on the command line (or press the Run function key).

For example, if you run the query against the Q.ORG table, you produce the following report:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	HEAD OFFICE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
42	GREAT LAKES	100	MIDWEST	CHICAGO
51	PLAINS	140	MIDWEST	DALLAS
66	PACIFIC	270	WESTERN	SAN FRANCISCO
84	MOUNTAIN	290	WESTERN	DENVER

### Related concepts

**Edit codes**An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

**Usage codes**QMF usage codes can be entered in the USAGE field on QMF FORM.MAIN or FORM.COLUMNS to define how to use column data to produce reports and charts.

## Displaying certain columns of a table

To retrieve data from a table in the database, enter the **P.** keyword under the columns you want to see.

### Before you begin

1. [Start a QBE query](#)
2. [Draw an example table](#)

### About this task

By default, metadata (the data type and length of the column) is displayed for columns with binary, XML, or large object (LOB) data types. Default metadata is also displayed if the table contains columns with decimal-floating point data and the processor on which QMF is running does not support decimal floating-point instructions. For all columns except those columns with decimal floating-point data, in QMF for TSO you can change the edit code for the column to display the actual data. However, the ability to change the edit code for LOB data is controlled by the value of the DSQEC\_LOB\_RETRV global variable. This global variable can also be set to display LOB data instead of metadata by default. The same is true in QMF for CICS except for XML data over 32 KB, for which you cannot change the edit code. You can assign the OMIT usage code to columns for which metadata displays to keep them from appearing in your report.

**Restriction:** Different databases have varying levels of support for data types.

### Procedure

1. Enter the **P.** keyword under the names of the columns you want to see in an example table, as shown in the following figure:

```

Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+
      | P.       | P.       |         |         |         |

```

2. Enter RUN QUERY on the command line (or press the Run function key).

For example, if you run the query against the Q.ORG table, you produce the following report:

QMF produces this report:

```
DEPTNUMB  DEPTNAME
-----
      84  MOUNTAIN
      66  PACIFIC
      10  HEAD OFFICE
      15  NEW ENGLAND
      20  MID ATLANTIC
      38  SOUTH ATLANTIC
      42  GREAT LAKES
      51  PLAINS
```

The columns are displayed by default in the same order in which they appear in the table.

3. To change the order of the columns that are displayed, type over the names of the columns in the example table.

For example, to reverse the column names, type LOCATION over **DIVISION** and vice versa.

The table that is shown in the following figure reverses the DIVISION and LOCATION column names:

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | LOCATION | DIVISION |
-----+-----+-----+-----+-----+-----
      | P.      |          |          | P.      | P.      |
```

Figure 285. Reversing the positions of column names in a QBE query

4. To display a column more than once, type the name of the column a second time over the name of an unused column.

Alternatively, you can use the Enlarge function key to add a column to the example table. Then, type the name of the column you want to display in the new column. Put **P.** under the column name.

### Related concepts

[ENLARGE command](#)

[Edit codes](#)An edit code is a set of characters that tells QMF how to format and punctuate the data in a specific column of a report.

[Usage codes](#)QMF usage codes can be entered in the USAGE field on QMF FORM.MAIN or FORM.COLUMNS to define how to use column data to produce reports and charts.

## Displaying certain rows of a table

There are many ways to specify which rows of a table you want to present.

### Displaying rows that contain a certain value

To display only those rows of a table that have a certain value in a column, put the value under the column in the example table.

That value is then a *condition*. The query selects just those rows of the table that contain that value in the indicated column.

You can, for example, display all the column names that are shown in the example table, and select only the rows with 5 in the YEARS column, as shown below:

```
Q.STAFF | DEPT | NAME | JOB | YEARS |
-----+-----+-----+-----+-----
P.      |      |      |      | 5      |
```

QMF produces this report:

```
DEPT  NAME      JOB  YEARS
-----
  38  MARENGHI  MGR   5
  15  NGAN     CLERK 5
  10  DANIELS  MGR   5
```

84	DAVIS	SALES	5
84	GAFNEY	CLERK	5

You can display only the columns DEPT, NAME, and JOB and select only the rows with 20 in the DEPT column. (You could get the report without the DEPT column by not including the **P.** in that column of the example table.) For example, when you run this query:

```
Q.STAFF | ID | DEPT | NAME | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----|
          |   | P. 20 | P.   | P.   |         |         |         |
```

QMF produces this report:

DEPT	NAME	JOB
20	SANDERS	MGR
20	PERNAL	SALES
20	JAMES	CLERK
20	SNEIDER	CLERK

You cannot enter any values for columns that contain binary or XML data.

## Example elements

An *example element* is a symbol that is used to represent data in a column. It must appear in a named column before it can be used with a column function (AVG., COUNT., MAX., MIN., SUM.) in an unnamed column.

An example element must start with an underscore character (\_). Any string of letters and digits up to 17 characters can follow.

Include the example element in your query at least twice: once to define it in the example table and at least once in conditions or calculations (either in the example table or in a conditions box).

An example element might be similar to the name of the column to which it refers. For instance, an example element in the SALARY column might be `_S`, `_SAL`, or `_SALARY`. Similarity is not necessary, however. Someone who is accustomed to writing algebraic expressions might want `_X` and `_Y` as example elements.

The following example query defines `_S` as "any salary". Then, in the unnamed column, it calculates and selects the average of all the salaries in the Q.STAFF table.

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY |
-----+-----+-----+-----+-----+-----+-----|
          |   |     |     |     |         | _S     | P. AVG. _S |
```

Figure 286. Defining a new element in the QBE query

### Related concepts

[Adding a CONDITIONS box](#)

[ENLARGE command](#)

## Writing expressions

You can write expressions in conditions that operate on numeric columns by using valid QBE keywords operators for arithmetic expressions.

Condition	QBE keyword
Equal	=
Not equal	≠

Table 52. Valid operators for arithmetic expressions in QBE (continued)

Condition	QBE keyword
Greater than	>
Greater than or equal	>=
Less than	<
Less than or equal	<=
Multiple conditions	AND OR
Values within a range	BETWEEN
Values in a list	IN (x, y, z)
A certain string of characters	LIKE '%abc%'
Ignore certain characters	LIKE '_abc_'
Negative conditions	NOT

The QBE language does not recognize the following operators:

- Concatenation (||) operator
- Not greater than ( $\neg$ >)
- Not less than ( $\neg$ <)
- Not equal to ( $\neg$ <=)

If you use one of these operators, QMF displays an error message.

Keep in mind the following points as you write expressions in QBE:

- Order of evaluation

The following order of evaluation is used in QBE queries:

1. Built-in column functions
2. A plus sign or minus sign before a single value
3. Multiplication or division of two values
4. Addition or subtraction of two values

QMF evaluates operations at the same level of precedence from left to right. You can change the order of evaluation with parentheses just as you would use them in a mathematical formula. For example, the following two expressions are equivalent:

```
A * - B / C + D / E
((A*(-B))/C) + (D/E)
```

When you create a table, each column in it holds a certain type of data. QMF performs arithmetic operations only on numeric data types.

- Rules for quotation marks

Do not enclose numeric data in quotation marks. You need to enclose character data that is used in conditions in quotation marks only when:

- The data contains blanks (as in 'ROOM 27') or any characters other than digits, letters, #, \$, or @ (as in 'P.D.Q.' or 'BOW-WOW').
- The data contains a single quotation mark or apostrophe. (In this case you have to double the quotation mark inside the data, as in 'O' 'BRIEN').

- To distinguish the constants 'NULL' and 'USER' from the keywords NULL and USER.
- The data contains all double-byte characters.
- The data is character data of entirely digits, as in '849276552'.
- The data type is DATE, TIME, TIMESTAMP, or TIMESTAMP WITH TIME ZONE.

Do not enclose values to compare with columns of numeric data in quotation marks.

- Arithmetic overflow

When an operation in a query produces a result outside an allowable range, the situation is called "arithmetic overflow". It is possible for the result of an arithmetic operation to be outside the range that is allowed for the data type of the result. For example, 1000000 is an allowable value in a column with a data type of INTEGER, but the result of 1000000 \* 1000000 is too large to be stored in a field that is defined with this data type. Also, division of any number by 0 produces an overflow. Overflow conditions are depicted with "?????????" error symbols.

## Unnamed columns in an example table

The named columns in the tables in many of the examples in this information adequately represent reports that you might want to create. But for more complex queries, you need to add new "unnamed" columns or use target tables

To add an empty column to your query, put your cursor beside the column name to the left of where you want to add the column. Then, press the Enlarge function key. You can also erase an existing unwanted column name to create a new (unnamed) column.

You can add a column of descriptive information to your report by putting a constant in an example table in an added (unnamed) column. The following example query lists the names and addresses of the people that are listed in the Q.APPLICANT table with 14 years of education, and identifies each with the character constant APPLICANT.

```
Q.APPLICANT | NAME | ADDRESS | EDLEVEL |
-----+-----+-----+-----+
| P.AO. | P. | 14 | P. APPLICANT |
```

QMF produces this report:

```
NAME          ADDRESS          COL1
-----+-----+-----
CASALS        PALO ALTO,CA    APPLICANT
REID          ENDICOTT,NY     APPLICANT
RICHOWSKI     TUCSON,AZ      APPLICANT
```

You could also use a numeric constant. A constant can be up to 254 characters in length and, in addition to alphabetic and numeric characters, it can contain #, \$, and @.

Use example elements to refer to the columns in an example table that are the source of the data for the expression in an unnamed column. For example, the following query uses `_S` to refer to the values in the SALARY column and `_C` to refer to the values in the COMM column:

```
Q.STAFF | ID | DEPT | SALARY | COMM |
-----+-----+-----+-----+-----+
| P. | 20 | P._S + _C | _S | _C |
```

QMF produces a report that begins with these rows:

```
ID          COL1
-----+-----
10          -
20          18783.70
80          13632.80
190         14379.25
```

By using `_S` and `_C`, you can then create an expression from the values in two columns and put the sum of the two into the report, via the unnamed column.

There is no restriction on the location of the unnamed column. Like other data columns, however, the unnamed column must be to the right of the column with the table name.

- Example 1:

To list yearly, monthly, and weekly salaries, run this query:

```
Q.STAFF | ID | NAME | SALARY |          |          |
-----+---+-----+-----+-----+-----+
P.      |   |   |   _S   |   _S/12  |   _S/52  |
```

- Example 2:

Run the following query to perform these tasks: list the IDs, the commission, and the sum of the salary and commission columns; show the percentage of total earnings the commission represents; and list in descending order (DO.) by the percentage.

```
Q.STAFF | ID | SALARY | COMM |          |          |          |
-----+---+-----+-----+-----+-----+-----+
P.      | _S | P._C | P._S + _C | P.100*_C/(_S+_C) DO. |
```

### Related tasks

[Adding a target table](#)

## Adding conditions to the example table

You can write expressions in your example table that set conditions on which rows are selected.

For example, the query below selects only the rows where the commission is greater than or equal to 1000:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |          |
-----+---+-----+-----+-----+-----+-----+-----+
P.      |   |   |   |   |   |   |   |   >= 1000 P. |
```

QMF produces this report:

```
  ID      COMM
-----
  70     1152.00
  90     1386.70
 340     1285.00
```

If a column contains letters or special characters, it must have a character data type. (If it contains double-byte characters, it can have a graphic data type.)

If a column contains only numbers or mostly numbers, it might still have a character data type. For example, a column of part numbers might contain mostly digits. However, if a part number like "1390X" is in the column, the column must have a character data type.

## Adding a CONDITIONS box

You can express simple conditions in an example table. However, more complicated conditions require the use of example elements and a CONDITIONS box.

You can also specify expressions in an example table. However, it is generally more convenient to define example elements in the example table and specify the expressions in a CONDITIONS box.

To add a CONDITIONS box to your query, enter this command:

```
COMMAND====> DRAW COND
```

If you enter DRAW CONDITION or DRAW CONDITIONS instead, an example table of that name rather than a CONDITIONS box displays.

Use a CONDITIONS box to do any of the following:

- Refer to two or more columns in the condition. For example:

```
_S + _C > 20000
```

- Use a column function in the condition. For example:

```
AVG. _S > 20000
```

- Refer to a column in the example table more than once. For example:

```
_SAL > 10000 AND _SAL > _COMM
```

- Use the AND or OR operator in a condition requiring example elements. For example:

```
_Y=10 OR _S>2000
```

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

```
(_SAL > 20000 OR _COMM < 2000) AND DEPT = 84
```

- Avoid widening a column of the example table to hold a long condition.

The CONDITIONS box in the following query uses the example elements (`_S` and `_C`) defined in the example table to select rows where salary plus commission (`_S + _C`) is greater than \$20,000.

```
Q.STAFF | NAME | SALARY | COMM |
-----+-----+-----+-----+
P.      |AO.   | _S     | _C   |
|          CONDITIONS          |
|-----+-----+-----+
| _S + _C > 20000                |
```

QMF produces this report:

NAME	SALARY	COMM
GRAHAM	21000.00	200.30
WILLIAMS	19456.50	637.65

QMF presents the names in ascending order (AO.). (The result does not include anyone whose salary alone is greater than \$20,000 if the commission is null.)

This query selects anyone whose weekly salary is less than \$300:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+---+-----+-----+
P.      |   |     |     |    |      | _SAL   |     |
|          CONDITIONS          |
|-----+-----+-----+
| _SAL/52 < 300                |
```

This query selects anyone whose commission is 5% or more of total earnings:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+---+-----+-----+
P.      |   |     |     |    |      | _S     | _C   |
|          CONDITIONS          |
|-----+-----+-----+
| _C >= .05 * (_S+_C)          |
```

You can use more than one CONDITIONS box or more than one condition in any box. However, you must place each condition on a single row in the box.

Multiple conditions in a query are implicitly connected by the AND operator. That is, in the example below, the AND keyword is assumed between the two conditions `_Y = 10 OR _S > 20000` and `_C >= 1000`.



QMF evaluates the OR condition ( $\_Y = 10$  OR  $\_S > 20000$ ) before it connects and evaluates the two conditions.

Q.STAFF	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
P.					$\_Y$	$\_S$	$\_C$

CONDITIONS
$\_Y = 10$ OR $\_S > 20000$

CONDITIONS
$\_C \leq 1000$

### Related concepts

[Writing expressions](#)

[Adding conditions to the example table](#)

## Adding a target table

An alternative to adding a new unnamed column to your example table is to use a target table. A target table is an empty example table that uses example elements to refer to other example tables. Anything that you can use in an unnamed column added to an example table you can also use in a target table.

### About this task

You cannot use target tables (or unnamed columns in example tables) to do the following:

- Name the column for your report. QMF names columns that are created by expressions (like the one in the previous example). You can change the name of a column in a report by using a form.
- Write a condition. (If you need to write a condition, write it in a named column or a CONDITIONS box.)
- Define an example element. You must define example elements in a named column of the example table.

### Procedure

To use a target table to combine information from two columns, display your table and issue the DRAW command:

```
COMMAND====> DRAW
```

The following figure shows the results of the DRAW command.

Q.STAFF	ID	DEPT	SALARY	COMM
	$\_I$	20	$\_S$	$\_C$

P.	$\_I$	$\_S + \_C$		
----	-------	-------------	--	--

Figure 287. A target table drawn with the DRAW command

### Related concepts

[Adding a CONDITIONS box](#)

[Customizing your reports](#)

When you run a query, QMF applies a default format to the data and presents it in a report. You can change this default report format by changing options on the QMF form panels.

## Eliminating duplicate rows

QMF displays all rows, including duplicate rows, by default if you have just one P. row in your query. To eliminate duplicate rows, specify UNQ. (unique) under the table name in the row with the P. operator.

If your example table has two or more P. rows, QMF does not display duplicate rows.

The following examples show queries with and without the UNQ. keyword.

- Example 1: This query presents all the rows, including duplicates.

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
| | | | | P. | |
```

QMF produces this report:

```
DIVISION
-----
CORPORATE
EASTERN
EASTERN
EASTERN
MIDWEST
MIDWEST
WESTERN
WESTERN
```

- Example 2: This query specifies UNQ. under the table name. Therefore, QMF eliminates all columns that contain duplicate data in the presented column.

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
UNQ. | | | | P. | |
```

QMF produces this report:

```
DIVISION
-----
CORPORATE
EASTERN
MIDWEST
WESTERN
```

### Related reference

[QBE keyword reference](#)

## Displaying data from more than one table

Sometimes you need information from two different tables. In this example, you link columns in two sample tables, and produce one report.

### About this task

You can display data from more than one table only if there is a link between the two tables (that is, a column in each table that contains identical information). For example, both Q.STAFF and Q.ORG have a column that contains employee numbers. In Q.STAFF this column is ID; in Q.ORG it is MANAGER. With this link, you can combine information from both tables into one report by using the following process.

### Procedure

1. In QMF, enter RESET QUERY to display an empty QBE Query panel.
2. Enter DRAW Q. STAFF, which draws the table that is shown in the following figure:

Q.STAFF	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
---------	----	------	------	-----	-------	--------	------

Figure 288. A representation of the Q.STAFF sample table drawn with the DRAW command

- This query uses only the table name and the first two columns. Delete the other columns by using the REDUCE command, as shown in the following figure.

Q.STAFF	ID	NAME
---------	----	------

Figure 289. You can delete columns of an example table by using the REDUCE command.

- Place the cursor on the command line and enter DRAW Q.ORG. The Q.ORG example table displays, as shown in the following figure:

Q.STAFF	ID	NAME			
Q.ORG	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION

Figure 290. Preparing to present data from two drawn tables

- Delete the **DIVISION** and **LOCATION** columns from the Q.ORG table, as shown in the following figure.

Q.STAFF	ID	NAME	
Q.ORG	DEPTNUMB	DEPTNAME	MANAGER

Figure 291. Deleting columns from the Q.ORG table in preparation for presenting data from both tables

- Add an unnamed column to the Q.ORG example table and increase its size by using the ENLARGE command, as shown in the following figure.

Q.STAFF	ID	NAME		
Q.ORG	DEPTNUMB	DEPTNAME	MANAGER	

Figure 292. Adding an unnamed column to the Q.ORG table and increasing its size with the ENLARGE command

- Add some example elements, as shown in the following figure:

Q.STAFF	ID	NAME
	_ID	_NM

Q.ORG	DEPTNUMB	DEPTNAME	MANAGER	
P.			_ID	_NM

Figure 293. Adding example elements in preparation for presenting data from two tables at once

The same example element (in this case **\_ID**) must be used in two example tables to select only the rows where **MANAGER** (the manager ID) in Q.ORG is equal to ID in Q.STAFF. The columns that you select for the condition cannot contain BINARY, VARBINARY, or XML data.

P. can appear in only one table. The example element \_NM is added to the unnamed column of the Q.ORG example table so that it is presented from the Q.STAFF table even though no P. appears in the Q.STAFF example table.

8. Press the Run function key.

This query displays the DEPTNUMB, DEPTNAME, and MANAGER columns from Q.ORG and the NAME column from Q.STAFF. It displays the rows where the data in the MANAGER column in Q.ORG is the same as the data in the ID column in Q.STAFF.

DEPTNUMB	DEPTNAME	MANAGER	NAME
20	MID ATLANTIC	10	SANDERS
38	SOUTH ATLANTIC	30	MARENGHI
15	NEW ENGLAND	50	HANES
42	GREAT LAKES	100	PLOTZ
51	PLAINS	140	FRAYE
10	HEAD OFFICE	160	MOLINARE
66	PACIFIC	270	LEA
84	MOUNTAIN	290	QUILL

Figure 294. A report that results from joining two tables in QBE

### Related concepts

[ENLARGE command](#)

## Writing queries to be shared

To make it possible to share a query with another user, you can use a model query, substitution variables, or the USER variable.

### Model query

A model is a copy of a query that contains conditions. With a model query, you or other users can produce different reports by specifying different conditions in a copy of the model.

Suppose, for example, that you are the sales manager of Department 38, and you write a query that lists the name, job, and commission for everyone in your department. The query might look like the one shown in the following figure:

Q.STAFF	NAME	DEPT	JOB	COMM
	P. A0.	38	P.	P.

Figure 295. A model query that presents information from the Q.STAFF table

Other sales managers can get a report for their departments by using your model query. They can display, change, and run the query, or change it and run it later.

### Substitution variables

Substitution variables make it possible for other people to use your query. Other users can substitute any value in place of the variable and produce a report specific to their needs.

A substitution variable can represent anything that you can write in a query, such as a column name (except for BINARY, VARBINARY, and XML columns), a search condition, or a specific value. You supply the value for a substitution variable on the RUN command or the **RUN Command Prompt** panel. You can also specify the substitution variable with a SET GLOBAL command (instead of RUN) before you run the query.

For example, you want a list of the employee names, jobs, and commissions of everyone in each of several different departments. You can construct a query that is generic enough to be used regardless of the department for which you need the information:

```

Q.STAFF | NAME      | DEPT  | JOB  | COMM  |
-----+-----+-----+-----+-----+
        | P. A0.    | &DEPT | P.   | P.    |

```

Figure 296. Substitution variables in a QBE query

If you run this query without a value on your RUN command, a prompt panel displays. On the prompt panel, you enter a value to substitute for the variable (&DEPT) in the query. For example, if you specify 38 for the value, QMF reads the query like this:

```

Q.STAFF | NAME      | DEPT  | JOB  | COMM  |
-----+-----+-----+-----+
        | P. A0.    | 38    | P.   | P.    |

```

The following table shows how you need to specify certain variable values.

Table 53. How to specify substitution variables in QBE	
Value that you want to specify for the variable	What to enter on the RUN command
A single valid numeric value	Specify the value that you want.
Text without embedded quotes, parentheses, blanks, equal signs, or commas	Specify exactly.
Text with embedded quotes	Enclose the entire value in quotes. (Quotation marks are not removed when QMF makes the substitution.)
Text with embedded parentheses, blanks, equal signs, or commas	Enclose the entire value in parentheses. (The outer parentheses are removed when QMF makes the substitution.)

Substitution variable names:

- Can be no longer than 18 characters, and the first character must be an ampersand (&).
- Can contain only these characters:
  - Letters of the alphabet
  - National characters: @ # \$
  - Special characters: ! % ? ~ ` { } \ | ! ¢
  - Numbers
  - Underscores ( \_ )
- Can be separated from another variable or command word by any of the characters that are not previously listed, such as commas, blanks, or parentheses.

## The USER variable

Another method of sharing a query is to create a query with USER under the NAME column (or any column that contains user identification (*user ID*) numbers. You can then share the query with other users, who can run it without further changes because their user ID is substituted for the word USER as a condition in the query.

### Related reference

[QBE keyword reference](#)

## QMF commands specific to QBE

The QMF commands in this topic are either unique to QBE or function differently with QBE queries than with SQL queries.

### CONVERT command

The CONVERT command converts a QBE query to an SQL query.

If you specify CONVERT ?, the prompt panel shown in the following figure displays. You can complete the command on the prompt panel.

```

                                CONVERT Command Prompt
type    ===> QUERY
name    ===>
        To convert an object from temporary storage, enter QUERY
        as its type.

        To convert an object from the database, enter its name (and
        optionally its type).
TARGET  ===> QUERY
        You can type QUERY to place the SQL query text on the SQL
        Query panel, or VARS to place it in the global variable pool.
        If you specify no target, the default is QUERY.
CONFIRM ===> YES
        Display the Confirmation panel before converting the current
        query to the SQL Query panel. YES or NO.

        Press Enter to execute the command from this panel.

13=Help    15=End
Please follow the directions on the Command Prompt panel.
Command ===>

```

Figure 297. The CONVERT Command Prompt panel

If your query contains substitution variables, and you do not provide values for them on your CONVERT command, a prompt panel displays. You can use this panel to supply the values for the variables. For example, assume that you write the following query, and save it as THISONE:

Q.STAFF	NAME	DEPT	JOB	COMM
-----	-----	-----	-----	-----
P. A0.		&DEPT	P.	P.

Now, suppose that you want to convert it to SQL, but specify only:

```
CONVERT THISONE
```

The prompt panel shown in the following figure displays:

### CONVERT Command Prompt -- Values of Variables

Your CONVERT command converts a query with variables that need values. Fill in a value after the arrow for each variable named below:

```
&DEPT          ===>
                ===>
                ===>
                ===>
                ===>
                ===>
                ===>
                ===>
                ===>
                ===>
```

Then press Enter to execute the command from this panel.

```
13=Help      15=End
Please give a value for each variable name.
Command ===>
```

Figure 298. Prompt panel that displays when you convert a query with substitution variables

When you enter the department number (84), the following SQL query displays:

```
SELECT "NAME", "JOB", "COMM"
FROM "Q"."STAFF"
WHERE ("DEPT" = 84)
ORDER BY 0000001
```

The CONVERT command does not operate on a QBE query at a remote location.

A single QBE insert or delete query can result in multiple SQL statements after the conversion. These statements are all placed on the SQL Query panel. However, all of the statements after the first are turned into query comments (each line is preceded by two hyphens). Use the Delete key to remove the hyphens from any statements that you want to run. You must place a semicolon at the end of every SQL statement except the last. The DSQEC\_RUN\_MQ global variable controls support for multi-statement queries.

#### Related reference

Global variables that control how commands and procedures are executed [DSQEC global variables control how commands and procedures are executed](#). All of these global variables can be modified by the SET GLOBAL command.

## DELETE command

The keyword D. is different from the DELETE command.

### About this task

The DELETE command removes the following from a QBE query:

- An example table
- A COMMENTS box
- A CONDITIONS box
- Error messages on the Query panel

### Procedure

To delete one of the above items:

1. Type DELETE on the command line. Do not press Enter yet.
2. Move the cursor to a position anywhere within any of the items that are listed above.

## QBE commands

3. Press Enter. QMF deletes the item.

### Related reference

[QBE keyword reference](#)

## DRAW command

The DRAW command creates an example table or adds a COMMENTS box, CONDITIONS box, or target table to a QBE query.

If you specify the DRAW command by itself (or press the Draw function key), an empty table appears. DRAW has the following forms:

- DRAW

Draws an empty target table:

-----+	-----+	-----+	-----+	-----+
--------	--------	--------	--------	--------

- DRAW *tablename*

Draws an example table and places the name of the table or view in the first column.

If *tablename* specifies an existing table or view, QMF draws an example of that table or view. The example table has the same number of columns with the same column names as the table or view name has. The width of each column in the example table depends on the data type for each column.

For example, DRAW Q.STAFF produces this example table:

Q.STAFF	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+

If *tablename* specifies a nonexistent table, QMF draws an example table and places the specified table name at the left of the panel. For example, if there is no table in the database that is named EMPTYBOX, the command DRAW EMPTYBOX produces this example table:

EMPTYBOX	-----+	-----+	-----+	-----+
----------	--------	--------	--------	--------

- DRAW COMM

Adds an empty COMMENTS box:

COMMENTS	-----+
----------	--------

- DRAW COND

Adds an empty CONDITIONS box:

CONDITIONS	-----+
------------	--------

## ENLARGE command

The ENLARGE command increases the size of an example table, COMMENTS box, or CONDITIONS box. The maximum allowable table width depends on the number of columns that are selected and the size of the column names. Longer names use more space. You can select up to 300 columns.

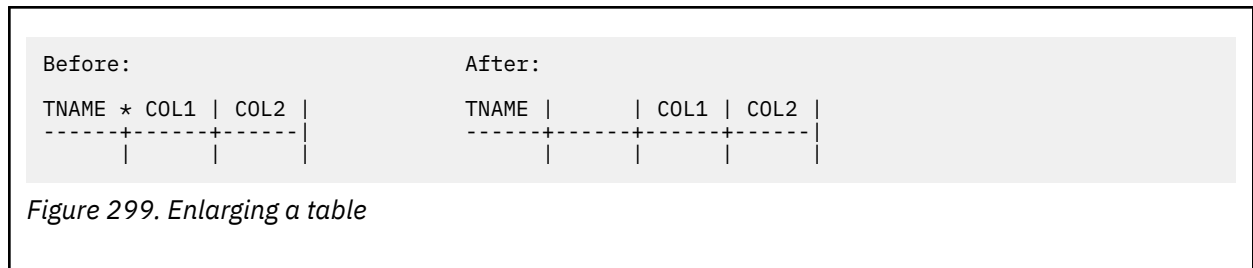
To enlarge a table, position your cursor appropriately (see the asterisk in the examples) and enter ENLARGE on the command line or press the Enlarge function key.



**Examples**

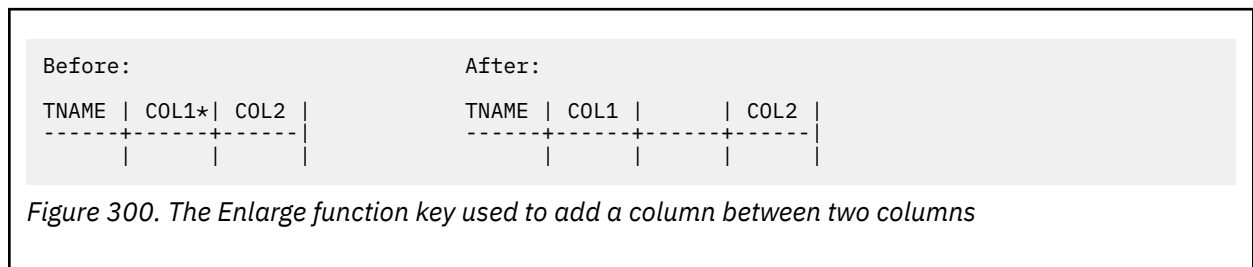
- Example 1:

Add a column to the right of the column that shows the table name. Put the cursor above the line, on the vertical bar, and press the Enlarge function key. The following figure shows a table before and after it is enlarged.



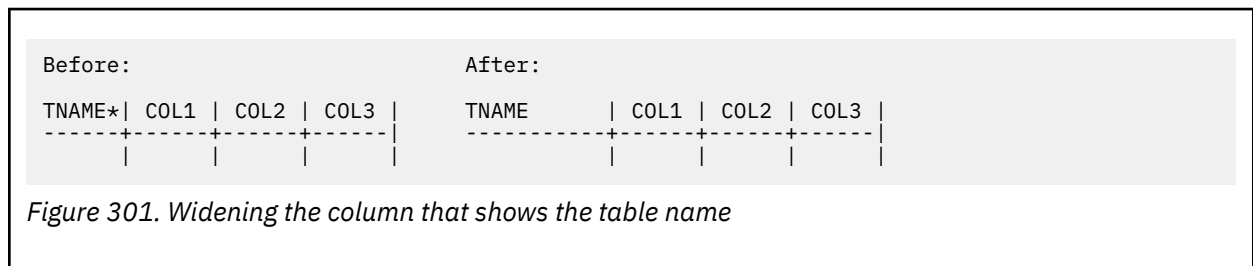
- Example 2:

Add a column to the right of any other column. Put the cursor above the line, within the left adjacent column, and press the Enlarge function key. The result is shown in the following figure:



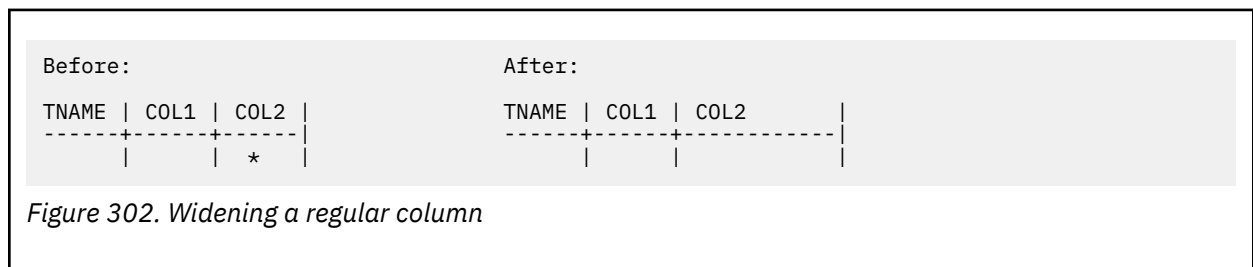
- Example 3:

Widen the column that shows the table name. Put the cursor in that column, above the line, and press the Enlarge function key. The result is shown in the following figure:



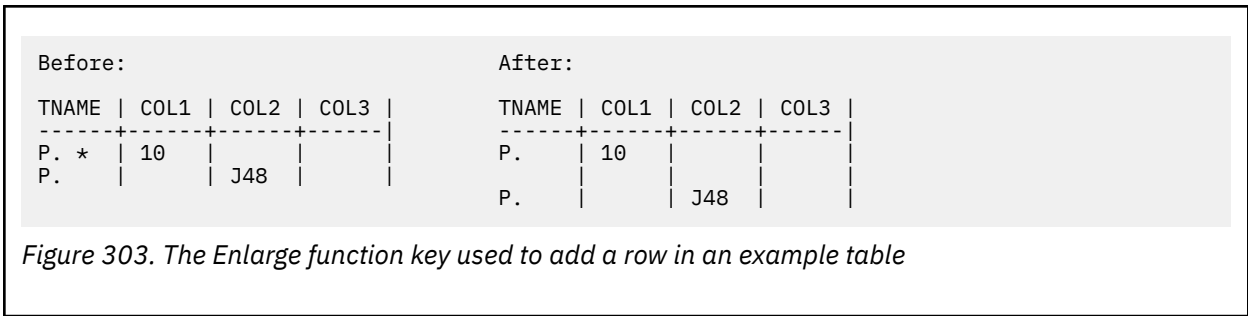
- Example 4:

Widen any other column. Put the cursor in that column, on or below the line, and press the Enlarge function key. The result is shown in the following figure:



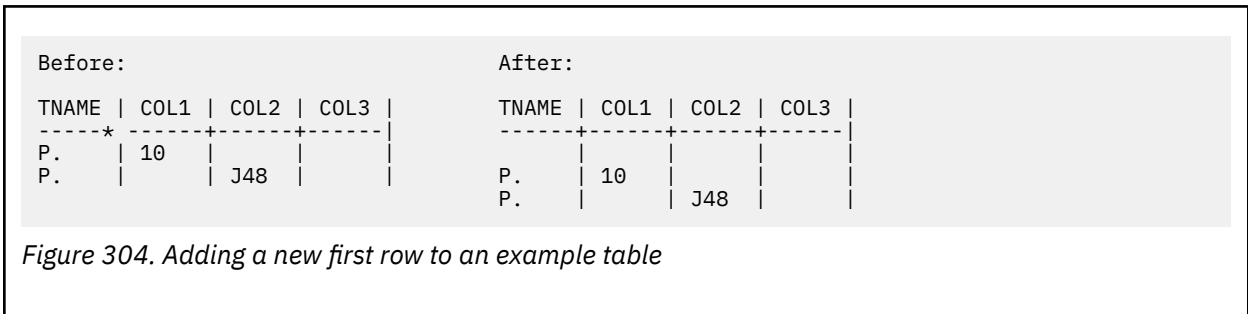
- Example 5:

Add a row below any row. Put the cursor below the line, under the table name, and press the Enlarge function key. The result is shown in the following figure:



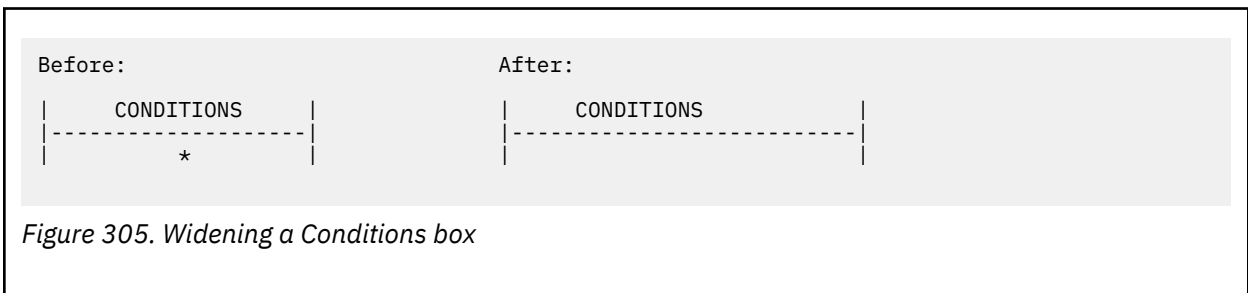
• Example 6:

Add a new first row. Put the cursor on the line, under the table name, and press the Enlarge function key. The result is shown in the following figure.



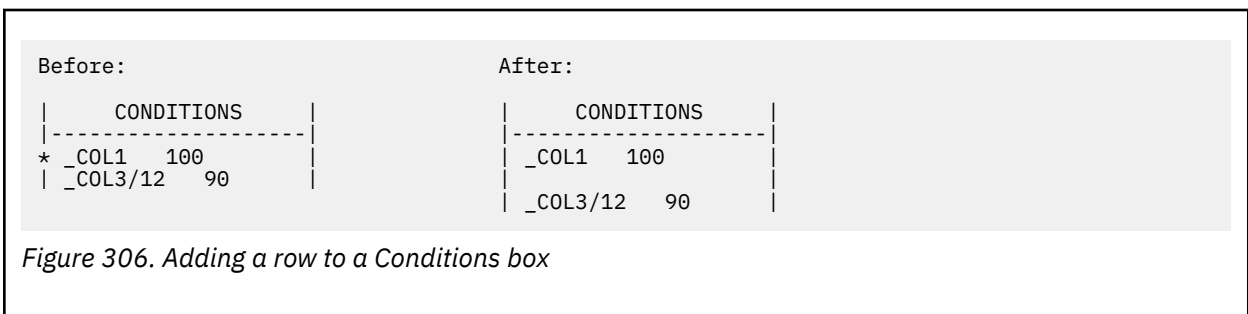
• Example 7:

Widen a CONDITIONS or COMMENTS box. Put the cursor below the line, inside the box, and press the Enlarge function key. The result is shown in the following figure.



• Example 8:

Add a row to a CONDITIONS or COMMENTS box. Put the cursor below the line, on the vertical bar at the left, and press the Enlarge function key. The result is shown in the following figure.



**REDUCE command**

The REDUCE command reduces the size of an example table, COMMENTS box, or CONDITIONS box.

To reduce a table, position your cursor appropriately (see the asterisk in the examples) and enter REDUCE on the command line or press the Reduce function key.

## Examples

- Example 1:

Remove a column. Put the cursor above the line, within the column, and press the Reduce function key. The result is shown in the following figure.

Before:				After:		
TNAME	COL1 *	COL2	COL3	TNAME	COL2	COL3
-----+	-----+	-----+	-----+	-----+	-----+	-----+

*Figure 307. Removing a column from an example table*

- Example 2:

Narrow the column that is labeled with the table name. Put the cursor above the line, in that column, and press the Reduce function key. The result is shown in the following figure.

Before:				After:			
TNAME *	COL1	COL2	COL3	TNAME	COL1	COL2	COL3
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+

*Figure 308. Narrowing the table-name column in an example table*

- Example 3:

Narrow any other column. Put the cursor on or below the line, in that column, and press the Reduce function key. The result is shown in the following figure:

Before:				After:			
TNAME	COL1	COL2	COL3	TNAME	COL1	COL2	COL3
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+
		*					

*Figure 309. Narrowing a column in an example table*

- Example 4:

Remove a row. Put the cursor under the table name, in the row to be removed, and press the Reduce function key. The result is shown in the following figure.

Before:				After:			
TNAME	COL1	COL2	COL3	TNAME	COL1	COL2	COL3
-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+
P.	10			P.	10		
P. *		J48					

*Figure 310. Removing a row from an example table*

- Example 5:

Narrow a CONDITIONS or COMMENTS box. Put the cursor below the line, inside the box, and press the Reduce function key. The result is shown in the following figure.



Table 54. Keywords used in QBE (continued)

Keyword	Action	Topic
IN (x, y, z)	Present certain values in a list	<a href="#">“IN (x,y,z) – Present certain values in a set” on page 360</a>
LIKE	Present on part of a character or graphic value	<a href="#">“LIKE – Present on part of a value” on page 361</a>
MAX.	Calculate the maximum value	<a href="#">“MAX. – Calculate the maximum value” on page 362</a>
MIN.	Calculate the minimum value	<a href="#">“MIN. – Calculate the minimum value” on page 362</a>
NOT	Present the opposite condition	<a href="#">“NOT – Present on the opposite of the condition” on page 363</a>
NULL	Present rows with missing entries	<a href="#">“NULL – Present rows with missing entries” on page 364</a>
OR	Present either of two conditions	<a href="#">“OR – Present on either of two conditions” on page 365</a>
P.	Present information in a table	<a href="#">“P. – Present data in a table” on page 365</a>
SUM.	Calculate the sum	<a href="#">“SUM. – Calculate the total” on page 367</a>
U.	Update a row in a table	<a href="#">“U. – Update a row in a table” on page 368</a>
UNQ.	Eliminate duplicate rows	<a href="#">“UNQ. – Eliminate duplicate rows” on page 369</a>
USER	Present rows with a value of <i>user ID</i>	<a href="#">“USER – Present rows with a value equal to a user identification” on page 369</a>
+ - */	Calculate arithmetic expressions	<a href="#">“+, -, *, / – Calculated values” on page 369</a>
= < >	Present on equality and inequality	<a href="#">“=, &lt;, &gt; (equality and inequality)” on page 370</a>

### ALL. – Display duplicate rows

**ALL.** ensures that all rows, including duplicate rows, display on the report. Specify ALL. under the table name in the row with the **P.** operator. You can use ALL. only when a **P.** operator is also specified.

**ALL.** is the default operator if the example table has only one **P.** operator.

However, if an example table has two or more **P.** operators, QMF excludes duplicates.

### AND – Present on two conditions

Two conditions that are connected by **AND** cause the query to select only rows that satisfy both conditions. The following query selects rows where the YEARS column is equal to 10 and the SALARY column is greater than 20000. The query selects only the two rows that satisfy both of these conditions.

```

Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----+
| P. | P. | | | | P. _Y | P. _S | |

```

```

|          CONDITIONS          |
|-----|
| _Y = 10 AND _S > 20000     |

```

QMF produces this report:

ID	NAME	YEARS	SALARY
50	HANES	10	20659.80
210	LU	10	20010.00

You can specify two conditions on the same row of a table. For example, if you want to display every clerk in Department 20, you can use the query that is shown in the following figure, which is the same as connecting the two conditions by **AND**.

```

Q.STAFF | ID | NAME | DEPT | JOB |
|-----+-----+-----+-----+-----|
P.      |   |     | 20   | CLERK |

```

Figure 313. How to specify the AND connector between two conditions in a QBE example table

The AND connector cannot be part of a row condition for a BINARY, VARBINARY, or XML column of a table.

### AO., AO(n). – Sort rows in ascending order

To put rows in a report in ascending order by the values in some column, put **AO.** in that column. (Make sure that you use the letter O.)

The sorting sequence for character data, in ascending order, is as follows:

1. Special characters, including blanks
2. Lowercase letters, in alphabetical order
3. Uppercase letters, in alphabetical order
4. Numbers, in ascending order
5. Null values

The sorting sequence for DATE, TIME, TIMESTAMP, and TIMESTAMP WITH TIME ZONE values is chronological.

The internal value of the data determines the sorting sequence for double-byte character set (DBCS) data. It is generally not meaningful.

The following query produces a report that lists the name, job, and years of employment for each employee in Department 84 in ascending alphabetical order by job.

```

Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
|-----+-----+-----+-----+-----+-----+-----+-----|
|   | P. | 84 | P.AO. | P. |   |   |   |

```

QMF produces this report:

NAME	JOB	YEARS
GAFNEY	CLERK	5
QUILL	MGR	10
DAVIS	SALES	5
EDWARDS	SALES	7

To order by more than one column, put **AO(1).** under the column to be ordered first. Then, put **AO(2).** under the next most significant column, and so on.

The number that follows **AO.** indicates the sort priority. The sequence of sort priorities you use does not need to be complete. For example, you can use 1, 2, and 4 without using 3, but no two columns can have the same priority.

The following query sorts by job first (in ascending order). Then, within each job classification, it sorts the rows by years of employment (in ascending order). After drawing the Q.STAFF table, enlarge the JOB and YEARS columns in the example table to accommodate the sort specifications shown.

```
Q.STAFF | NAME | DEPT | JOB          | YEARS      |
-----+-----+-----+-----+-----+
      P. | 84   | P. AO(1). | P. AO(2). |
```

QMF produces this report:

```
NAME      JOB      YEARS
-----+-----+-----
GAFNEY    CLERK    5
QUILL     MGR      10
DAVIS     SALES    5
EDWARDS   SALES    7
```

You can sort only on columns that the query selects. If you use **AO.** in a row and column of an example table, you must use **P.** either in the same row and column, or in the same row under the table name (which displays every column).

You cannot use the AO. keyword on a column that contains XML data.

### AVG. – Calculate the average value

The AVG. column function calculates the average of all values in a column for rows selected. It applies to numeric data and returns a single value for that data. You can use the UNQ. operator with AVG. to use only unique values when calculating the average. A column function does not include null values in its calculation.

To select only the SALARY column, define an example element for SALARY in the SALARY column. Then, add an unnamed column, request the average, and put the example element in the unnamed column.

For example, when you run this query:

```
Q.STAFF | SALARY |          |
-----+-----+-----+
      _S | P.AVG._S |
```

QMF produces this report:

```
          COL1
-----+-----
16675.6422857142
```

To find the average of the values in the SALARY column for clerks only, add a condition to your query. For example, when you run this query:

```
Q.STAFF | SALARY | JOB |          |
-----+-----+-----+-----+
      _S | CLERK | P.AVG._S |
```

QMF produces this report:

```
          COL1
-----+-----
12612.6125000000
```

Rules for AVG.:

- You can use AVG. only on numeric columns.

## Keyword reference

- In an unnamed column, you must specify `AVG.` along with the example element that identifies the column to be averaged.
- `AVG.` can be followed by an example element, an arithmetic expression that contains at least one example element, or the `UNQ.` operator followed by an example element. If the data in a column you want to average is defined by an arithmetic expression, enclose the expression in parentheses.
- When you apply `AVG.` to one column that is named in an example table, you must apply a column function (`AVG.`, `MIN.`, `MAX.`, `COUNT.`, OR `SUM.`) or the `G.` (group) operator to every other column selected.
- You cannot use the `AVG.` function on a column if the sum of the data in the column would cause an overflow condition.

### BETWEEN x AND y – Present values within a range

You can select all the rows that have a value between two limits. The limits are inclusive. You can abbreviate `BETWEEN` as `BT`. Comparisons that use `BETWEEN` do not work unless the smaller value comes before the larger value. In the following example, notice that the smaller value, 20000, appears immediately after `BT`.

```
Q.STAFF | ID | NAME | SALARY
-----+-----+-----+-----|
P.      |   |     | BT 20000 AND 21000 |
```

When you run the previous query, QMF produces this report:

```
  ID  NAME      SALARY
-----+-----+-----|
  50  HANES      20659.80
 210  LU          20010.00
 310  GRAHAM     21000.00
```

As another example, you can select all rows that have `YEARS` equal to 8, 9, or 10 by using the following clause:

```
Q.STAFF | ID | NAME | YEARS          | SALARY
-----+-----+-----+-----+-----|
P.      |   |     | BETWEEN 8 AND 10 |
```

QMF produces this report:

```
  ID  NAME      YEARS  SALARY
-----+-----+-----+-----|
  20  PERNAL      8      18171.25
  50  HANES      10     20659.80
 190  SNEIDER     8      14252.75
 210  LU          10     20010.00
 270  LEA         9      18555.50
 280  WILSON     9      18674.50
 290  QUILL      10     19818.00
```

Use `BETWEEN` either in an example table or in a `CONDITIONS` box. You can enter `_Y` in the `YEARS` column and `_Y BETWEEN 8 AND 10` in a `CONDITIONS` box to produce the previous report; these entries are shown in the following figure:

```
Q.STAFF | ID | NAME | YEARS          | SALARY
-----+-----+-----+-----+-----|
P.      |   |     | _Y             |
|
| CONDITIONS
|-----+-----|
| _Y BETWEEN 8 AND 10
```

Figure 314. How to specify a `BETWEEN` condition in a QBE example table



`_Y BETWEEN 8 and 10` produces the same results as `_Y >= 8 AND _Y <= 10`, but is easier to specify

The BETWEEN keyword cannot be part of a row condition for a BINARY, VARBINARY, or XML column of a table.

### COUNT. – Count the number of values in a column

The COUNT. column function finds the number of unique values in a column. Specify COUNT. either in an unnamed column or in a target table. You can abbreviate COUNT. in the following way:

```
CNT.
```

The following query finds the number of unique values in the SALARY column.

```
Q.STAFF | ID      | DEPT  | SALARY |          |
-----+-----+-----+-----+-----+
          |          |      |      |      | P.COUNT._S
```

QMF produces the following report:

```
      COL1
-----
      35
```

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

Rules for COUNT.:

- COUNT. counts only unique values.
- An example element must follow COUNT.
- You cannot follow COUNT. with an expression or an example element within an expression.
- You can use COUNT. only in reference to a specific column. Follow COUNT. with an example element alone.
- The COUNT. function operates on all data types except XML.

### D. – Delete rows from a table

To delete one or more rows from a table, put the D. operator under the table name in the row you want to delete.

You can delete rows from a table that you created or from a copy of a table that someone else created. (You need authorization to create or copy a table.) To copy the Q.STAFF sample table, for example, enter `DISPLAY Q.STAFF`. When Q.STAFF displays, enter `SAVE DATA AS PERS`.

The following query deletes the row that contains the ID number 140 from the PERS table:

```
PERS | ID      | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+
D.   | 140    |      |      |      |      |      |      |
```

Figure 315. Using QBE to delete a row from a database table

You can delete more than one row with one DELETE statement.

The following query deletes everyone in Department 10:

```
PERS | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+-----+-----+-----+-----+
D.   |   |      | 10   |      |      |      |      |
```

Figure 316. Using QBE to delete several rows at once from a database table

The example table with D. can have multiple rows, but you cannot mix the operators D., I., P., or U. in a single example table.



**Attention:** If D. appears under the table name with no conditions in other columns, QMF deletes the entire contents of the table.

Keep the following rules in mind when you use the D. keyword:

- An example table can have multiple D. rows.
- You cannot delete rows if the deletion is dependent on values in other rows of the same table.
- You cannot use QBE to delete data from BINARY, VARBINARY, or XML columns.

### **DO., DO(n). – Sort rows in descending order**

To put rows in a report in descending order by the values in a column, put DO. in that column. Use the letter "O" (not the digit zero).

The sorting sequence for character data, in descending order, is as follows:

1. Null values
2. Numbers, in descending order
3. Uppercase letters, in descending alphabetical order
4. Lowercase letters, in descending alphabetical order
5. Special characters, including blanks

With DO., the sorting sequence for DATE, TIME, TIMESTAMP, and TIMESTAMP WITH TIME ZONE values is reverse-chronological.

The internal value of the data determines the sorting sequence for DBCS data. The sorting sequence generally is not meaningful.

The following query produces a report that lists the name, job, and years of employment for each employee in Department 84 in descending order by job.

```
Q.STAFF | ID | NAME | DEPT | JOB          | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----+
          |   | P.   | 84   | P. DO.     | P.    |        |      |
```

QMF produces this report:

```
NAME      JOB      YEARS
-----+-----+-----
EDWARDS   SALES    7
DAVIS     SALES    5
QUILL     MGR      10
GAFNEY    CLERK    5
```

To order by more than one column, put DO(1). under the column to be ordered first. Then, put DO(2). under the next most significant column, and so on.

The number that follows DO. is called the sort priority. The sequence of sort priorities you use does not need to be complete. For example, you can use 1, 2, and 4 without using 3, but no two columns can have the same priority.

The following query sorts by job first (in descending order). Then, within each job classification, it orders the rows by years of employment, beginning with the greatest number of years (in descending order).

```
Q.STAFF | NAME | DEPT | JOB          | YEARS |
-----+-----+-----+-----+-----+
          | P.   | 84   | P. DO(1).  | P. DO(2). |
```

QMF produces this report:

```
NAME      JOB      YEARS
-----+-----+-----
```

EDWARDS	SALES	7
DAVIS	SALES	5
QUILL	MGR	10
GAFNEY	CLERK	5

You can sort on only those columns that the query selects. If you use `DO .` in a row and column of an example table, you must use `P .` either in the same row and column, or in the same row under the table name (which displays every column).

You cannot use the `DO .` keyword on a column that contains XML data.

## G. — Grouping data

The `G .` keyword groups selected rows by a specified column for performing operations on each group. The `G .` keyword accumulates the results by group, but it does not order the groups. (Use `AO .` or `DO .` to ensure the order that you want.)

For example, you can group by department to determine each department's average salary with the following steps:

1. Group the rows by department number (`G .` under `DEPT`).
2. Specify one average for each department (`_S` under `SALARY` and `AVG . _S` in an unnamed column to link `_S` to the `SALARY` column).
3. Add `P .` in the columns for which you want the results selected.
4. Add `AO .` to put them in ascending order by department.

When you run this query:

```
Q.STAFF | DEPT      | SALARY |
-----+-----+-----+
          | G.P.AO. | _S      | P.AVG._S |
```

QMF produces this report:

```
DEPT      COL1
-----+-----
 10 20865.862500000
 15 15482.332500000
 20 16071.525000000
 38 15457.110000000
 42 14592.262500000
 51 17218.160000000
 66 17215.240000000
 84 16536.750000000
```

Generally, `G .` produces one group for each set of identical values in a column. If there are null values in the column, they form a single group.

Keep the following rules in mind when you use the `G .` keyword:

- Any example element that does not refer to a `G .` column must have an associated column function.
- When you use grouping, you can select only data that refers to the groups. Only columns that contain `G .` or an aggregating function can contain `P .`
- A row of an example table that uses the `G .` keyword cannot use the `I .`, `U .`, or `D .` keywords.
- If more than one column contains `G .`, QMF groups the selected rows by each unique value of the combined columns. For example, if `G .` appears in both the `DEPT` column and the `LOCATION` column, each row of a group has the same `DEPT` value and `LOCATION` value.
- You cannot use the `G .` keyword on columns that contain XML data.

## I. — Insert rows into a table

To insert one or more rows into a table, put the `I .` operator under the table name and the values you want to insert under their respective columns. Each row that you want to insert must contain the `I .` operator.

If you leave a blank under a column, or omit a column from the example table, a null value is inserted in that column in the database. You must specify values for all columns that are defined as NOT NULL.

You can insert rows into a table you created or into a copy of a table someone else created. (You need authorization to create or copy a table.) To copy the Q.STAFF sample table, for example, enter DISPLAY Q.STAFF. When Q.STAFF displays, enter SAVE DATA AS PERS.

The following query inserts two rows into the PERS table:

PERS	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
I.	400	HARRISON	20	SALES		18000.66	0
I.	455	STONER	17			19000.00	540.00

*Figure 317. Using QBE to insert rows into a database table*

The following query inserts date and time values in a table called TEST.DATETIME:

TEST.DATETIME	SMALLINTEGER	DATE	TIME
I.		'1987-11-11'	'14.22.00'

*Figure 318. Using QBE to insert date and time values into a database table*

If authorized, you can copy rows from one table to another using the I. operator. In the following query, the example elements show which columns the query copies from Q.STAFF into PERS. The DEPT column in Q.STAFF is duplicated; one DEPT column contains a condition that limits one set of rows to Department 38. The YEARS column is also duplicated; one YEARS column contains a condition that limits the second set of rows to YEARS > 10. If employees in Department 38 have more than ten years of experience, they are in the report twice.

Q.STAFF	ID	NAME	DEPT	DEPT	JOB	YEARS	YEARS
	_I1	_N1	_D1	38	_J1	_Y1	
	_I2	_N2	_D2		_J2	_Y2	>10

PERS	ID	NAME	DEPT	JOB	YEARS
I.	_I1	_N1	_D1	_J1	_Y1
I.	_I2	_N2	_D2	_J2	_Y2

*Figure 319. Using the I. keyword to copy rows*

A CONDITIONS box that contains the conditions \_D1 = 38 and \_Y2 > 10 could be used instead of duplicating the DEPT and YEARS columns.

Keep the following rules in mind when you use the I. keyword:

- You cannot use a column name more than once in a table that receives inserted rows.
- You cannot insert a row in a table into the same table.
- You cannot use QBE to insert data into BINARY or VARBINARY columns.
- Surround XML data in single quotation marks when you insert it into a table by using the I. keyword.

### **IN (x,y,z) – Present certain values in a set**

You can select all the rows that contain any value in a set of values. Enclose the values in parentheses and separate one value from the next with a comma. A blank between values is optional. (You cannot specify NULL in a set of values.)

In the following query, the condition `IN (20, 38, 42)` in the `DEPTNUMB` column means "select every row with a department number of 20, 38, or 42." It is equivalent to, but simpler than, writing the condition `_D=20 OR _D=38 OR _D=42`.

```
Q.ORG | DEPTNUMB      | DEPTNAME |
-----+-----+-----+
P.    | IN (20, 38, 42) |          |
```

QMF produces this report:

```
DEPTNUMB  DEPTNAME
-----
      20  MID ATLANTIC
      38  SOUTH ATLANTIC
      42  GREAT LAKES
```

The `IN` keyword cannot be part of a row condition for a `BINARY`, `VARBINARY`, or `XML` column of a table.

### LIKE – Present on part of a value

To select data when you know only part of the value, use `LIKE` plus one of the following symbols for the unknown data:

- An underscore (`_`) is the symbol for any single character.

The underscore (`_`) in the following condition means to ignore the character between `LE` and `DS`. In other words, "search for `LE`, followed by any one character, followed by `DS`".

```
LIKE 'LE_DS'
```

This singles out the name `LEEDS` from the `NAME` column in the `Q.APPLICANT` sample table.

Enclose values containing underscore characters in single quotation marks to prevent search values from being mistaken for example elements.

Use a specific number of underscores to indicate that you want that number of characters ignored. For example, you can use the following condition to search the `PROJNO` column of the `Q.PROJECT` table for the combination `40` in positions 2 and 3. The first and last characters can be any characters.

```
LIKE '_40_'
```

- A percent sign (`%`) is the symbol for any number of characters (or none).

In the following query, `LIKE %NY` in the `ADDRESS` column means "where the address ends with `NY`, with anything at all before that." The *percent sign* (`%`) stands for any number of preceding characters (or none).

```
Q.APPLICANT | NAME | ADDRESS |
-----+-----+-----+
P.          | AO. | LIKE %NY |
```

QMF produces this report:

```
NAME          ADDRESS
-----
JACOBS       POUGHKEEPSIE, NY
LEEDS        EAST FISHKILL, NY
REID         ENDICOTT, NY
```

You can use both symbols in the same value.

You can use `LIKE` on columns with character or graphic data types only. For character data, you must always enclose the value after `LIKE` in single quotation marks. (Db2 for z/OS requires single quotation marks around an all-digit value of character data.)

When the data type of a column is VARCHAR, you do not need to know how many blanks to specify with LIKE. With VARCHAR, there are no blanks in the column. The size of the column varies with the size of its data.

When the data type of a column is CHAR, however, the size of the column is fixed. The column contains blanks, so use the appropriate number of blanks (or use the percent sign in your search criteria) when you specify LIKE.

If the data type of a column is LONG VARCHAR or LONG VARGRAPHIC, you cannot use it with LIKE (or any other search conditions).

## **MAX. – Calculate the maximum value**

The MAX. column function returns the greatest value in the group of numbers or characters in a specified column.

If MAX. is applied to a CHAR or VARCHAR type column, alphanumeric ordering is used:

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

QMF ignores null values when searching for the maximum. If all values specified in a column are null, QMF returns no value.

You can use an example element with MAX. to select the maximum years of employment and maximum salary from the Q.STAFF table. For example:

```
Q.STAFF | YEARS | SALARY |-----|-----|
-----+-----+-----+-----+-----+
      _Y |   _S   | P. MAX. _Y | P. MAX. _S |
```

QMF produces this report:

```
COL1      COL2
-----
    13    22959.20
```

Keep the following rules in mind when you use the MAX. keyword:

- You can follow MAX. with an example element or an arithmetic expression that contains at least one example element.
- In an unnamed column, you must specify MAX. with the example element that identifies the column from which the largest value is to be retrieved. This example element also appears in the column that contains the value to retrieve.
- When you apply MAX. to a column named in an example table, you must apply a column function (AVG., SUM., MIN., MAX., COUNT.) or the G. (group) operator to every other column to select.
- The MAX. function operates on all data types except XML.

## **MIN. – Calculate the minimum value**

The MIN. column function returns the smallest value in the group of numbers or characters in a specified column.

If MIN. is applied to a CHAR or VARCHAR type column, alphanumeric ordering is used:

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

QMF ignores null values when searching for the minimum. If all values specified in a column are null, QMF returns no value.

You can use an example element with MIN. to select the minimum years of employment for employees in the Q.STAFF table. For example:

```
Q.STAFF | NAME | DEPT | YEARS |
-----+-----+-----+-----+
| | | | _Y | P. MIN. _Y |
```

QMF produces this report:

```
COL1
-----
1
```

Keep the following rules in mind when you use the MIN. keyword:

- You can follow MIN. with an example element or an arithmetic expression that contains at least one example element.
- In an unnamed column, specify MIN. with the example element that identifies the column from which the smallest value is to be retrieved. This example element also appears in the column that contains the value to retrieve.
- When you apply MIN. to a column named in an example table, you must apply a column function (AVG., SUM., MIN., MAX., COUNT.) or the G. (group) operator to every other column to select.
- The MIN. function operates on all data types except XML.

### NOT – Present on the opposite of the condition

You can use the opposite of any condition by putting NOT before it. NOT takes precedence over AND and OR. For example, in the following query, rows are selected that do not contain 38 in the DEPTNUMB column but do contain EASTERN in the DIVISION column. The other row in the Q.ORG table that contains EASTERN in the DIVISION column contains 38 in the DEPTNUMB column, so it is not presented.

```
Q.ORG | DEPTNUMB | DIVISION | LOCATION |
-----+-----+-----+-----+
P. | _DEP | _DIV | |
| CONDITIONS |
| NOT _DEP=38 AND _DIV=EASTERN |
```

QMF produces this report:

```
DEPTNUMB DIVISION LOCATION
-----
15 EASTERN BOSTON
20 EASTERN WASHINGTON
```

To illustrate how parentheses can change the results of a query, consider the following examples.

When you run this query:

```
Q.ORG | DEPTNUMB | DIVISION | LOCATION |
-----+-----+-----+-----+
P. | _DEP | _DIV | _LOC |
| CONDITIONS |
| NOT _DEP=51 AND _DIV=MIDWEST OR _LOC=BOSTON |
```

QMF produces this report:

```
DEPTNUMB DIVISION LOCATION
-----
```

## Keyword reference

```
15 EASTERN      BOSTON
42 MIDWEST     CHICAGO
```

When you place parentheses as follows, your report is the same as in the previous example.

```
(NOT _DEP=51 AND _DIV=MIDWEST) OR _LOC=BOSTON
```

However, if you move the left parenthesis after NOT, as in the following query, you get different results.

When you run this query:

```
Q.ORG | DEPTNUMB | DIVISION | LOCATION |
-----+-----+-----+-----+
P.    | _DEP     | _DIV     | _LOC     |
-----+-----+-----+-----+
|          CONDITIONS          |
|-----+-----+-----+-----+
| NOT (_DEP=51 AND _DIV=MIDWEST) OR _LOC=BOSTON |
```

QMF produces this report:

```
DEPTNUMB  DIVISION  LOCATION
-----
10 CORPORATE  NEW YORK
15 EASTERN   BOSTON
20 EASTERN   WASHINGTON
38 EASTERN   ATLANTA
42 MIDWEST   CHICAGO
66 WESTERN   SAN FRANCISCO
84 WESTERN   DENVER
```

Keep the following rules in mind when you use the NOT keyword:

- You can type NOT =, NOT NULL, NOT LIKE, NOT IN, or NOT BETWEEN.
- With greater than or less than, NOT must precede the entire condition, for example, NOT \_YEARS > 10.
- Only the NOT NULL clause is a valid use of the NOT operator for columns that contain binary (BINARY or VARBINARY) or XML data.

### NULL – Present rows with missing entries

A null indicator is used in any field of a table that does not contain a value or any field where the value is explicitly set to NULL. This indicator prints and displays as a single hyphen (-) by default.

Do not confuse null fields with any of these values:

- A numeric value of zero
- A character string of all blanks
- A character string of length zero
- The character string NULL (of length 4)

Each of the previous items is a value that you can enter in some row and column of a table.

To select rows that have no entry in a column, use the NULL keyword in that column of the example table. For example, the following query displays the IDs and names of employees in Department 38 for whom the YEARS column is null.

When you run this query:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+
| P. | P. | 38 | | NULL | | |
```

QMF produces this report:

```
ID  NAME
-----
```



60 QUIGLEY  
120 NAUGHTON

You cannot use NULL with an example element in an example table. Use a CONDITIONS box instead. For example, this query is incorrect:

```
Q.STAFF | NAME | COMM          | SALARY |
-----+-----+-----+-----+
P.      |      | _C  $\neq$  NULL | _S     | _C + _S |
```

This query is correct:

```
Q.STAFF | NAME | COMM          | SALARY |
-----+-----+-----+-----+
P.      |      | _C            | _S     | _C + _S |

|          CONDITIONS          |
|-----|
| _C  $\neq$  NULL                   |
```

QMF interprets the NULL keyword as "unknown". The result of an operation on an unknown value is also unknown, so the result of any operation on NULL is NULL.

Remember that NULL is not zero. NULL is the absence of a value. In the sample table Q.STAFF, there is no value for COMM for managers because managers do not make commissions. In some examples, earnings is shown as SALARY + COMM. If this calculation is done for managers, the result is always a null value.

Follow these rules when you use the NULL keyword:

- You can use NULL alone or with =,  $\neq$ , or NOT.
- In a CONDITIONS box, you can use NULL only with a column name or an example element.

## OR – Present on either of two conditions

Two conditions that are connected by OR allow the query to select every row that satisfies one or the other of the conditions. The following query selects rows where either the YEARS column is equal to 10 or the SALARY column is greater than 20000.

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+---+-----+-----+
| P. | P. |      |      |      | P. _Y | P. _S |      |
|          CONDITIONS          |
| _Y = 10 OR _S > 20000        |
```

QMF produces this report:

ID	NAME	YEARS	SALARY
50	HANES	10	20659.80
140	FRAYE	6	21150.00
160	MOLINARE	7	22959.20
210	LU	10	20010.00
260	JONES	12	21234.00
290	QUILL	10	19818.00
310	GRAHAM	13	21000.00

The OR connector cannot be part of a row condition for a BINARY, VARBINARY, or XML column of a table.

## P. – Present data in a table

You can use P. to present all the columns or some of the columns in a table. You can use the P. keyword in a number of ways:

- Present all the columns in a table

## Keyword reference

To see all the columns in a table, put P. under the table name in the example table. All the columns that are shown in the example table appear.

When you run this query:

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
P.    |           |           |           |           |           |
```

QMF produces this report:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
84	MOUNTAIN	290	WESTERN	DENVER
66	PACIFIC	270	WESTERN	SAN FRANCISCO
10	HEAD OFFICE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
42	GREAT LAKES	100	MIDWEST	CHICAGO
51	PLAINS	140	MIDWEST	DALLAS

- Present some of the columns in a table

To see selected columns, put P. under the names of the columns you want to see. You can put the P. before or after other things you put under the column heading.

When you run this query:

```
Q.ORG | DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION |
-----+-----+-----+-----+-----+-----+
      | P.        |           |           | P.        | P.        |
```

QMF produces this report:

- Present some of the rows in a table

To see only certain rows of a table, add conditions to your query. For example, you can present all the Q.STAFF table columns, but only the rows that contain SALES in the JOB column.

When you run this query:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+
P.      |   |     |     | SALES |     |     |     |
```

QMF produces this report:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
20	PERNAL	20	SALES	8	18171.25	612.45
40	O'BRIEN	38	SALES	6	18006.00	846.55
60	QUIGLEY	38	SALES	-	16808.30	650.25
70	ROTHMAN	15	SALES	7	16502.83	1152.00
90	KOONITZ	42	SALES	6	18001.75	1386.70
150	WILLIAMS	51	SALES	6	19456.50	637.65
220	SMITH	51	SALES	7	17654.50	992.80
280	WILSON	66	SALES	9	18674.50	811.50
300	DAVIS	84	SALES	5	15454.50	806.10
310	GRAHAM	66	SALES	13	21000.00	200.30
320	GONZALES	66	SALES	4	16858.20	844.00
340	EDWARDS	84	SALES	7	17844.00	1285.00

- Present data from multiple tables

To present data from two tables, draw two example tables that have at least one column that contains the same data (for example, ID in the Q.STAFF table and MANAGER in the Q.ORG table). Add one or more unnamed columns to one of the tables. Enter the same example element in each table in the columns that contain the same data. Then, enter another example element in an unnamed column of the first table, and enter that same example element in a named column of the second table. (P. can appear only in the table with the unnamed column.)

When you run this query:

```

Q.STAFF | ID | NAME |      |
-----+-----+-----+-----+
P.      |  _I |      |  _D |

Q.ORG   | DEPTNUMB | DEPTNAME | MANAGER |
-----+-----+-----+-----+
        |  _D      |          |  _I     |

```

QMF produces this report:

ID	NAME	DEPTNUMB
10	SANDERS	20
30	MARENGHI	38
50	HANES	15
100	PLOTZ	42
140	FRAYE	51
160	MOLINARE	10
270	LEA	66
290	QUILL	84

- Present data dependent on non-presented data

A query that uses multiple tables can present data from one table that is dependent on data in another table. For example, by using the example element `_D` in the `DEPT` column of `Q.STAFF` and the `DEPTNUMB` column of `Q.ORG`, you can present the ID, name, and department of only those employees who are in Dallas.

When you run this query:

```

Q.STAFF | ID | NAME | DEPT |
-----+-----+-----+-----+
P.      |    |      |  _D  |

Q.ORG   | DEPTNUMB | LOCATION |
-----+-----+-----+
        |  _D      | DALLAS   |

```

QMF produces this report:

ID	NAME	DEPT
140	FRAYE	51
150	WILLIAMS	51
220	SMITH	51
230	LUNDQUIST	51
250	WHEELER	51

You cannot use the `D.` (delete), `I.` (insert), and `U.` (update) keywords in the same query with `P.`

## SUM. – Calculate the total

The `SUM.` column function calculates the total of all values in a column for selected rows. It applies to a group of numeric values and returns a single value for each group of numbers to which it is applied. You can use the `UNQ.` operator with `SUM.` to request that QMF uses only unique values when calculating the sum. QMF ignores nulls. If all values in the specified column are null, the total is null.

You can use an arithmetic expression with `SUM.`. The following example calculates the total earnings (salaries plus commissions) for every selected row in `Q.STAFF`:

```

Q.STAFF | NAME | SALARY | COMM |
-----+-----+-----+-----+
        |      |  _S    |  _C    | P. SUM. (_S+_C) |

```

Figure 320. Using arithmetic expressions with the `SUM.` keyword in QBE

## Keyword reference

All the columns referred to in an unnamed column are either grouped or have a column function specified. For example, you can select the total, average, and maximum salaries by department.

When you run this query:

```
Q.STAFF | DEPT | SALARY |           |           |           |
-----+-----+-----+-----+-----+-----+
          | P. G. | _S   | P. SUM._S | P. AVG. _S | P. MAX. _S |
```

QMF produces this report:

DEPT	COL1	COL2	COL3
10	83463.45	20865.8625000000	22959.20
15	61929.33	15482.3325000000	20659.80
20	64286.10	16071.5250000000	18357.50
38	77285.55	15457.1100000000	18006.00
42	58369.05	14592.2625000000	18352.80
51	86090.80	17218.1600000000	21150.00
66	86076.20	17215.2400000000	21000.00
84	66147.00	16536.7500000000	19818.00

Keep the following rules in mind when you use the SUM. keyword:

- You can use SUM. only on columns that have numeric data types.
- In an unnamed column, specify SUM. with the example element that identifies the column to be totaled.
- SUM. can be followed by an example element, an arithmetic expression that contains at least one example element, or the UNQ. operator followed by an example element. If the data in a column to be totaled is defined by an arithmetic expression, enclose the expression in parentheses.
- When you apply SUM. to a column named in an example table, you must apply a column function (AVG., MIN., MAX., COUNT. or SUM.) or the G. (group) operator to every other column to select.

## U. – Update a row in a table

To update one or more values in an existing row of a table, put the U. operator, with the new value, in each column you want to change. An example table that uses U. can have more than one row. However, all rows must contain the U. operator. Values in other columns identify the row or rows to change.

You can update rows in a table that you created or in a copy of a table someone else created. (You need authorization to create or copy a table.) For example, to copy the Q.STAFF sample table, enter DISPLAY Q.STAFF. When Q.STAFF displays, enter SAVE DATA AS PERS.

The following query updates the PERS table for employees 250 and 330. It changes data in the JOB column to SALES and increases the value of the SALARY column by 15%.

```
PERS | ID | JOB | SALARY | SALARY |
-----+-----+-----+-----+-----+
      | 250 | U. SALES | _S1 | U. _S1*1.15 |
      | 330 | U. SALES | _S2 | U. _S2*1.15 |
```

Figure 321. Using the U. keyword in QBE to update rows in a table

To see the changed rows in your PERS table, enter DISPLAY PERS. The updated PERS table looks like this:

ID	NAME	JOB	SALARY
250	WHEELER	SALES	16629.00
330	BURKE	SALES	12636.00

To update date and time values in a QBE update query, enclose them in single quotation marks, as shown in the following figure:

```

MY.INTERVIEW |      INTDATE      |  STARTTIME  |  MANAGER  |
-----+-----+-----+-----+
              | U. '1987-04-04'   | U. '14.22.00' | 270      |

```

Figure 322. Using quotation marks with date and time values in an update query in QBE

Keep the following rules in mind when you use the U. keyword:

- You can update a column only with a constant or with values from other columns in the same row.
- You cannot update a column in a row from columns in other rows in the same table.
- You cannot update a row if it is dependent on other rows in the same table.
- Surround XML data in single quotation marks when submitting updates.
- You cannot use the U. operator with columns that contain binary (BINARY or VARBINARY) data. To update binary data, use an SQL query with either the BX or BINARY() function.

## UNQ. – Eliminate duplicate rows

UNQ. eliminates duplicate rows from query results. If your example table contains two or more P. rows, QMF deletes duplicate rows from the query results by default. 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 the P. operator.

When you run this query:

```

Q.ORG | DEPTNUMB | DEPTNAME |  MANAGER  |  DIVISION  |  LOCATION  |
-----+-----+-----+-----+-----+-----+
UNQ.  |           |           |           | P.         |           |

```

QMF produces this result:

```

DIVISION
-----
CORPORATE
EASTERN
MIDWEST
WESTERN

```

You cannot use the UNQ. keyword with columns that contain XML data.

## USER – Present rows with a value equal to a user identification

When you run a query with USER in the NAME column (or in any column that contains user ID numbers), your own user ID is substituted for the word USER as a condition in the query. You can then share the query with other users, who can run it without further changes. QMF automatically substitutes their user IDs for the USER keyword. (USER is not preceded by &.)

Suppose that you usually run a query once a month (using the Q.STAFF table) to find out what your commission is to the current day. You discover that your colleagues also want to check the same information. You can write the query that is shown in the following figure and share it with them.

```

Q.STAFF | NAME | COMM |
-----+-----+-----+
              | USER | P.   |

```

Figure 323. How to use the USER keyword to make a query generic enough to share with other users

## +, -, \*, / – Calculated values

A QBE query can present not only data already in a table, but also results that can be calculated using data in numeric columns.

\_S/12 is an example of an expression; it is the result of dividing SALARY by 12. You can form expressions by using symbols for operations. The following figure shows which symbols to use for which operations:

Symbol	Operation
+	Add
-	Subtract
*	Multiply
/	Divide

Within expressions you can use column headings (RATE\*HOURS), constants (RATE\*1.07), and column functions (AVG. (\_S)/2).

In the report, column names for calculated values are different depending on whether you are using DB2 Server for VSE and VM or Db2 for z/OS. You might see, for example:

- 1, 2, or 3
- COL1, COL2, or COL3
- EXPRESSION 1, EXPRESSION 2, or EXPRESSION 3
- AVG (EXPRESSION 2)

You can produce reports with columns that contain the values of expressions. To do so, put the expression in an unnamed column as in the following query.

To show the total earnings of employees in Department 20, include \_S + \_C in the unnamed column.

When you run this query:

```
Q.STAFF | ID | NAME | DEPT | SALARY | COMM |
-----+-----+-----+-----+-----+-----+
| P. | P. | 20 | _S | _C | P._S + _C |
```

QMF produces this report:

ID	NAME	COL1
10	SANDERS	-
20	PERNAL	18783.70
80	JAMES	13632.80
190	SNEIDER	14379.25

The value of SALARY+COMM for employee 10 is null because the value of COMM is null, and the result of any calculation with a null value is null.

You can get a report for everyone in Department 38 and their monthly salaries. When you run this query:

```
Q.STAFF | NAME | DEPT | SALARY |
-----+-----+-----+-----+
| P. | P.38 | _S | P._S/12 |
```

QMF produces this report:

NAME	DEPT	COL1
MARENGHI	38	1458.8958333333
O'BRIEN	38	1500.5000000000
QUIGLEY	38	1400.6916666666
NAUGHTON	38	1079.5625000000
ABRAHAMS	38	1000.8125000000

### =, <math>\neq</math>, >, < (equality and inequality)

To select rows that satisfy a condition that is based on equality or inequality, put the condition under the appropriate column.

For example, the following query displays a report with everyone who has 10 or more years of service.

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----+
| P. | P. | | | | P. >=10 | | |
```

QMF produces this report:

```
   ID  NAME      YEARS
-----+-----+-----
   50  HANES        10
  210  LU            10
  260  JONES         12
  290  QUILL         10
  310  GRAHAM        13
```

If you do not specify an operator this way, the operator defaults to equality. You can write the following query to produce a report that contains all the managers.

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----+
| P. | P. | | | MGR | | |
```

QMF produces this report:

```
   ID  NAME
-----+-----
   10  SANDERS
   30  MARENGHI
   50  HANES
  100  PLOTZ
  140  FRAYE
  160  MOLINARE
  210  LU
  240  DANIELS
  260  JONES
  270  LEA
  290  QUILL
```

You can display a report that contains everyone in the alphabet after SMITH. For example, when you run this query:

```
Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----+
| P. | P. >SMITH A0. | | | | | |
```

QMF produces this report:

```
   ID  NAME
-----+-----
  190  SNEIDER
  250  WHEELER
  150  WILLIAMS
  280  WILSON
  130  YAMAGUCHI
```

Equality or greater than or less than symbols cannot be part of a row condition for columns that contain BINARY, VARBINARY, or XML data.

**Related concepts**

[Writing expressions](#)

**Related information**

[The IBM Publications Center](#)To find information about the BX or BINARY() function, see the SQL reference information for the database you are using.

## QBE exercises with solutions

---

This topic provides exercises and solutions to help you learn the QBE query method.

### Exercises

Exercises 1 through 4 use the Q.STAFF table.

#### Exercise 1

1. Write a query to produce a list of employee names and jobs for every employee in Department 84.
2. After you have successfully run the query in Step “1” on page 372, use the Query function key to bring it back to your display panel. Change it to produce a list of employee numbers, employee names, years of service, and salaries for every employee in Department 51.
3. Change the query to show all of the columns for employees in Department 51.
4. Produce a report that contains the employee identification number, name, department, and years of service for each person who has no data in the YEARS column.
5. Write a query to produce a list that shows employee identification number, name, job, and years of service for everyone with 10 or more years of service. Would someone with exactly 10 years of service appear on your list?
6. Produce a report that contains the name and commission for any manager whose row contains a commission amount.

#### Exercise 2

1. Produce a report that contains each clerk's name, department, and years of service. Arrange the report in ascending alphabetic order by employee name.
2. Produce a report that contains the name, department, and years of service for every clerk. Put the department numbers in ascending order and, within each department, put the years of service in ascending order.
3. Write a query to produce a list that shows employee number, employee name, and years of service for all the clerks. Arrange the report by years of service with the most senior clerk first.
4. Change Step “3” on page 372 to again arrange the report in descending order by years of service but, within each year, in ascending order by department number. Include the department numbers in your report.
5. Produce a report that contains each employee whose name contains the letter Z.
6. Produce a report that contains each employee whose name begins with S.
7. Produce a report that contains each employee whose name has an A as its third character.

#### Exercise 3

1. Produce a report that contains the name, salary, and commission of any employee whose salary is greater than \$18,000 or whose commission exceeds \$1,000.
2. Produce a report that lists all employees who have no data in their years of service column or no data in their commission column. Show the employee's name, years of service, and commission. (Hint: Remember that you must use the equal (=) or not equal ( $\neq$ ) symbols when comparing for null values in QBE.)
3. Write a query to produce a list that shows employee number, name, and salary for everyone with a salary between \$20,000 and \$21,000. Did people with a salary of exactly \$20,000 or \$21,000 appear on your list?
  - If they did appear, how could you exclude them?
  - If they did not appear, how could you include them?



- Produce a report that lists all managers who have been with the company fewer than 10 years, but whose salary is at least \$20,000. Show the name, job title, years of service, and salary.
- Show the name, years of service, salary, and commission of those employees with fewer than 10 years of service and either a salary over \$20,000 or a commission over \$1,000.

### Exercise 4

- Write a query to produce the name, employee number, salary, commission, and total earnings (salary plus commission) of everyone with a job of sales.
- Write a query to produce the name, number, salary, commission, and total earnings of everyone with a job of sales whose total earnings are less than \$17,500.
- Produce a report that lists each salesperson's name as well as commission as a percentage of salary. (For example, if a person's salary is \$20,000 and commission is \$2,000, the commission percentage is 10%.) Arrange the report in descending order by the commission percentage.
- Change Step “3” on page 373 so that commission percentage is based on total earnings (salary plus commission = 100%).

### Exercise 5

- Write a query that will access both the Q.STAFF and Q.ORG tables (DRAW Q.STAFF and DRAW Q.ORG). Produce a report that contains each department name, location, and manager's name.
- Change Step “1” on page 373 to list only those departments in the Eastern division.
- Change Step “2” on page 373 to list any managers in the Eastern division who have 10 or more years of service. For each manager, list the department name, location, and manager's name.

### Exercise 6

- Make a copy of the Q.STAFF table and call it MYTABLE.
- Write a query to update MYTABLE. Change the name of the manager for Department 66 to RAMOTH, years of service to 7, and salary to \$18,238.50. Write a query to retrieve the row after you have updated it.
- Write a query that increases the salaries in MYTABLE by 10%. Retrieve all the rows for clerks. Because MYTABLE began with data identical to Q.STAFF, you can randomly check YEARS and SALARY against the Q.STAFF table, to ensure that the correct employees received the salary increases.
- Insert a new row into MYTABLE. The new employee's information is as follows:

```
ID      = 275
NAME    = ROGERS
DEPT    = 66
JOB     = SALES
YEARS   = NULL
SALARY  = $14,000.00
COMM    = NULL
```

After you have inserted the row, write and run a query that will display it.

- Delete from MYTABLE the rows for salespeople in Department 66.

## Solutions to exercises

Solutions show minimum columns. Your answer may include unused columns that have been deleted here.

### Solutions to Exercise 1

- The finished query is as follows:

## QBE exercises with solutions

```

Q.STAFF | NAME      | DEPT  | JOB  |
-----+-----+-----+-----+
          | P.        | 84    | P.   |
    
```

The report resulting from the query is as follows:

```

NAME      JOB
-----+-----
QUILL     MGR
DAVIS     SALES
EDWARDS   SALES
GAFNEY    CLERK
    
```

2. The finished query is as follows:

```

Q.STAFF | ID      | NAME  | DEPT  | YEARS | SALARY |
-----+-----+-----+-----+-----+-----+
          | P.     | P.    | 51    | P.    | P.     |
    
```

The report resulting from the query is as follows:

```

  ID  NAME      YEARS  SALARY
-----+-----+-----+-----
 140  FRAYE        6    21150.00
 150  WILLIAMS     6    19456.50
 220  SMITH        7    17654.50
 230  LUNDQUIST    3    13369.80
 250  WHEELER     6    14460.00
    
```

3. The finished query is as follows:

```

Q.STAFF | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+----+-----+-----+-----+
P.      |   |     | 51   |    |       |        |      |
    
```

The report resulting from the query is as follows:

```

  ID  NAME      DEPT  JOB    YEARS  SALARY  COMM
-----+-----+-----+-----+-----+-----+-----
 140  FRAYE      51    MGR    6      21150.00  -
 150  WILLIAMS  51    SALES  6      19456.50  637.65
 220  SMITH     51    SALES  7      17654.50  992.80
 230  LUNDQUIST 51    CLERK  3      13369.80  189.65
 250  WHEELER   51    CLERK  6      14460.00  513.30
    
```

4. The finished query is as follows:

```

Q.STAFF | ID      | NAME  | DEPT  | YEARS |
-----+-----+-----+-----+-----+
          | P.     | P.    | P.    | P.NULL |
    
```

The report resulting from the query is as follows:

```

  ID  NAME      DEPT  YEARS
-----+-----+-----+-----
  60  QUIGLEY    38    -
  80  JAMES      20    -
 120  NAUGHTON   38    -
 200  SCOUTTEN   42    -
    
```

5. The finished query is as follows:

```

Q.STAFF | ID      | NAME  | JOB  | YEARS |
-----+-----+-----+-----+-----+
          | P.     | P.    | P.   | P.>=10 |
    
```

The report resulting from the query is as follows:

ID	NAME	JOB	YEARS
50	HANES	MGR	10
210	LU	MGR	10
260	JONES	MGR	12
290	QUILL	MGR	10
310	GRAHAM	SALES	13

6. The finished query is as follows:

Q.STAFF	NAME	JOB	COMM
	P.	MGR	P. NULL

The report resulting from the query is as follows:

NAME	COMM
SANDERS	-
MARENGHI	-
HANES	-
PLOTZ	-
FRAYE	-
MOLINARE	-
LU	-
DANIELS	-
JONES	-
LEA	-
QUILL	-

This is a correct result; it is an empty set. None of the managers is on commission, so no data exists in the COMM column.

## Solutions to Exercise 2

1. The finished query is as follows:

Q.STAFF	NAME	DEPT	JOB	YEARS
	P. A0.	P.	CLERK	P.

The report resulting from the query is as follows:

NAME	DEPT	YEARS
ABRAHAMS	38	3
BURKE	66	1
GAFNEY	84	5
JAMES	20	-
KERMISCH	15	4
LUNDQUIST	51	3
NAUGHTON	38	-
NGAN	15	5
SCOUTTEN	42	-
SNEIDER	20	8
WHEELER	51	6
YAMAGUCHI	42	6

2. The finished query is as follows:

Q.STAFF	NAME	DEPT	JOB	YEARS
	P.	P. A0(1).	CLERK	P. A0(2).

The report resulting from the query is as follows:

NAME	DEPT	YEARS
KERMISCH	15	4
NGAN	15	5
SNEIDER	20	8
JAMES	20	-

## QBE exercises with solutions

ABRAHAMS	38	3
NAUGHTON	38	-
YAMAGUCHI	42	6
SCOUTTEN	42	-
LUNDQUIST	51	3
WHEELER	51	6
BURKE	66	1
GAFNEY	84	5

3. The finished query is as follows:

```

Q.STAFF | ID      | NAME      | JOB      | YEARS |
-----+-----+-----+-----+-----+
          | P.      | P.        | CLERK    | P. DO. |
    
```

The report resulting from the query is as follows:

```

      ID  NAME      YEARS
-----+-----+-----
      80  JAMES      -
     200  SCOUTTEN   -
     120  NAUGHTON   -
     190  SNEIDER     8
     130  YAMAGUCHI   6
     250  WHEELER     6
     350  GAFNEY      5
     110  NGAN        5
     170  KERMISCH    4
     230  LUNDQUIST   3
     180  ABRAHAMS    3
     330  BURKE       1
    
```

- Null years sort to the top when you specify descending order.
- The order of names may vary within order by fields. For example, Scoutten may be listed before James, because both have null years.

4. The finished query is as follows:

```

Q.STAFF | ID      | NAME      | DEPT      | JOB      | YEARS |
-----+-----+-----+-----+-----+-----+
          | P.      | P.        | P. A0(2). | CLERK    | P. DO(1). |
    
```

The report resulting from the query is as follows:

```

      ID  NAME      DEPT  YEARS
-----+-----+-----+-----
      80  JAMES      20    -
     120  NAUGHTON   38    -
     200  SCOUTTEN   42    -
     190  SNEIDER    20    8
     130  YAMAGUCHI  42    6
     250  WHEELER    51    6
     110  NGAN       15    5
     350  GAFNEY     84    5
     170  KERMISCH   15    4
     180  ABRAHAMS   38    3
     230  LUNDQUIST  51    3
     330  BURKE     66    1
    
```

5. The finished query is as follows:

```

Q.STAFF | NAME      |
-----+-----+
          | P. LIKE '@(#)' |
    
```

The report resulting from the query is as follows:

```

NAME
-----
KOONITZ
PLOTZ
GONZALES
    
```

6. The finished query is as follows:

```

Q.STAFF | NAME |
-----+-----+
          | P. LIKE 'S%' |
    
```

The report resulting from the query is as follows:

```

NAME
-----
SANDERS
SNEIDER
SCOUTTEN
SMITH
    
```

7. The finished query is as follows:

```

Q.STAFF | NAME |
-----+-----+
          | P. LIKE ' _A%' |
    
```

The report resulting from the query is as follows:

```

NAME
-----
NGAN
FRAYE
LEA
GRAHAM
    
```

### Solutions to Exercise 3

1. The finished query is as follows:

```

Q.STAFF | NAME | SALARY | COMM |
-----+-----+-----+-----+
          | P. | P. _S | P. _C |
          |
          | CONDITIONS |
          |-----+-----+
          | _S > 18000 OR _C > 1000 |
    
```

The report resulting from the query is as follows:

NAME	SALARY	COMM
SANDERS	18357.50	-
PERNAL	18171.25	612.45
O'BRIEN	18006.00	846.55
HANES	20659.80	-
ROTHMAN	16502.83	1152.00
KOONITZ	18001.75	1386.70
PLOTZ	18352.80	-
FRAYE	21150.00	-
WILLIAMS	19456.50	637.65
MOLINARE	22959.20	-
LU	20010.00	-
DANIELS	19260.25	-
JONES	21234.00	-
LEA	18555.50	-
WILSON	18674.50	811.50
QUILL	19818.00	-
GRAHAM	21000.00	200.30
EDWARDS	17844.00	1285.00

2. The finished query is as follows:

```

Q.STAFF | NAME | YEARS | COMM |
-----+-----+-----+-----+
          | P. | P. _Y | P. _C |
          |
          | CONDITIONS |
          |-----+-----+
    
```

```

|-----|
| _Y = NULL OR _C = NULL |
|-----|
    
```

The report resulting from the query is as follows:

NAME	YEARS	COMM
SANDERS	7	-
MARENGHI	5	-
HANES	10	-
QUIGLEY	-	650.25
JAMES	-	128.20
PLOTZ	7	-
NAUGHTON	-	180.00
FRAYE	6	-
MOLINARE	7	-
SCOUTTEN	-	84.20
LU	10	-
DANIELS	5	-
JONES	12	-
LEA	9	-
QUILL	10	-

3. The finished query is as follows:

```

Q.STAFF | ID | NAME | SALARY |
|-----|---|-----|-----|
| P. | P. | P. | P. |
|-----|---|-----|-----|

Exclusive BETWEEN:                               Inclusive BETWEEN:

|-----|           |           |           |           |
| CONDITIONS |           | CONDITIONS |           |
|-----|           |-----|           |
| _S > 20000 AND _S < 21000 | | _S >= 20000 AND _S <= 21000 |
|-----|           |-----|           |

OR                                                 OR

|-----|           |           |           |           |
| CONDITIONS |           | CONDITIONS |           |
|-----|           |-----|           |
| _S BT 20001 AND 20999 | | _S BETWEEN 20000 AND 21000 |
|-----|           |-----|           |
    
```

The report resulting from the query is as follows:

ID	NAME	SALARY	ID	NAME	SALARY
50	HANES	20659.80	50	HANES	20659.80
210	LU	20010.00	210	LU	20010.00

4. The finished query is as follows:

```

Q.STAFF | NAME | JOB | YEARS | SALARY |
|-----|-----|-----|-----|-----|
| P. | P. =MGR | P. <10 | P. >=20000 |
|-----|-----|-----|-----|
    
```

The report resulting from the query is as follows:

NAME	JOB	YEARS	SALARY
FRAYE	MGR	6	21150.00
MOLINARE	MGR	7	22959.20

5. The finished query is as follows:

```

Q.STAFF | NAME | YEARS | SALARY | COMM |
|-----|-----|-----|-----|-----|
| P. | P. <10 | P. _S | P. _C |
|-----|-----|-----|-----|

|-----|
| CONDITIONS |
|-----|
| _S > 20000 OR _C > 1000 |
|-----|
    
```

The report resulting from the query is as follows:

NAME	YEARS	SALARY	COMM
ROTHMAN	7	16502.83	1152.00
KOONITZ	6	18001.75	1386.70
FRAYE	6	21150.00	-
MOLINARE	7	22959.20	-
EDWARDS	7	17844.00	1285.00

## Solutions to Exercise 4

1. The finished query is as follows:

Q.STAFF	ID	NAME	JOB	SALARY	COMM	
	P.	P.	SALES	P. _S	P. _C	P. _S+_C

The report resulting from the query is as follows:

ID	NAME	SALARY	COMM	COL1
20	PERNAL	18171.25	612.45	18783.70
40	O'BRIEN	18006.00	846.55	18852.55
60	QUIGLEY	16808.30	650.25	17458.55
70	ROTHMAN	16502.83	1152.00	17654.83
90	KOONITZ	18001.75	1386.70	19388.45
150	WILLIAMS	19456.50	637.65	20094.15
220	SMITH	17654.50	992.80	18647.30
280	WILSON	18674.50	811.50	19486.00
300	DAVIS	15454.50	806.10	16260.60
310	GRAHAM	21000.00	200.30	21200.30
320	GONZALES	16858.20	844.00	17702.20
340	EDWARDS	17844.00	1285.00	19129.00

2. The finished query is as follows:

Q.STAFF	ID	NAME	JOB	SALARY	COMM	
	P.	P.	SALES	P. _S	P. _C	P. _S+_C
CONDITIONS						
	(_S + _C) < 17500					

The report resulting from the query is as follows:

ID	NAME	SALARY	COMM	COL1
60	QUIGLEY	16808.30	650.25	17458.55
300	DAVIS	15454.50	806.10	16260.60

3. The finished query is as follows:

Q.STAFF	NAME	JOB	SALARY	COMM	
	P.	SALES	_S	_C	P. DO. 100*( _C/_S)

The report resulting from the query is as follows:

NAME	COL1
KOONITZ	7.70313900
EDWARDS	7.20130000
ROTHMAN	6.98062000
SMITH	5.62349500
DAVIS	5.21595600
GONZALES	5.00646500
O'BRIEN	4.70148800
WILSON	4.34549700
QUIGLEY	3.86862400
PERNAL	3.37043400
WILLIAMS	3.27731000
GRAHAM	0.95380900

You might try using this alternate query instead:

```

Q.STAFF | NAME | JOB | SALARY | COMM |
-----+-----+-----+-----+-----+
          | P. | SALES | _S | _C | P. DO. (100*_C)/_S |
    
```

The report resulting from this query is as follows:

NAME	COL1
KOONITZ	7.703
EDWARDS	7.201
ROTHMAN	6.980
SMITH	5.623
DAVIS	5.215
GONZALES	5.006
O'BRIEN	4.701
WILSON	4.345
QUIGLEY	3.868
PERNAL	3.370
WILLIAMS	3.277
GRAHAM	0.953

Note what happens to the precision of the ratio that you calculated. This truncation can also affect the order into which your rows are sorted.

4. The finished query is as follows:

```

Q.STAFF | NAME | JOB | SALARY | COMM |
-----+-----+-----+-----+-----+
          | P. | SALES | _S | _C | P. DO. 100*( _C/(_S+_C)) |
    
```

The report resulting from the query is as follows:

NAME	COL1
KOONITZ	7.15219600
EDWARDS	6.71754900
ROTHMAN	6.52512600
SMITH	5.32409500
DAVIS	4.95738100
GONZALES	4.76776800
O'BRIEN	4.49037300
WILSON	4.16452800
QUIGLEY	3.72453600
PERNAL	3.26053900
WILLIAMS	3.17331100
GRAHAM	0.94479700

### Solutions to Exercise 5

1. The finished query is as follows:

```

Q.STAFF | ID | NAME |
-----+-----+
          | _MID | _MNM |

Q.ORG | DEPTNAME | MANAGER | LOCATION |
-----+-----+-----+-----+
          | P. | _MID | P. | P. _MNM |
    
```

The report resulting from this query is as follows:

DEPTNAME	LOCATION	NAME
MID ATLANTIC	WASHINGTON	SANDERS
SOUTH ATLANTIC	ATLANTA	MARENGHI
NEW ENGLAND	BOSTON	HANES
GREAT LAKES	CHICAGO	PLOTZ
PLAINS	DALLAS	FRAYE
HEAD OFFICE	NEW YORK	MOLINARE



```
PACIFIC      SAN FRANCISCO  LEA
MOUNTAIN    DENVER         QUILL
```

2. The finished query is as follows:

```
Q.STAFF | ID      | NAME      |
-----+-----+-----+
          | _MID    | _MNM     |

Q.ORG   | DEPTNAME | MANAGER  | DIVISION | LOCATION |
-----+-----+-----+-----+-----+
          | P.      | _MID    | EASTERN  | P.      | P. _MNM |
```

The report resulting from this query is as follows:

```
DEPTNAME      LOCATION      NAME
-----+-----+-----
MID ATLANTIC  WASHINGTON    SANDERS
SOUTH ATLANTIC ATLANTA      MARENGHI
NEW ENGLAND   BOSTON       HANES
```

3. The finished query is as follows:

```
Q.STAFF | ID      | NAME      | YEARS |
-----+-----+-----+-----+
          | _MID    | _MNM     | >=10  |

Q.ORG   | DEPTNAME | MANAGER  | DIVISION | LOCATION |
-----+-----+-----+-----+-----+
          | P.      | _MID    | EASTERN  | P.      | P. _MNM |
```

The report resulting from the query is as follows:

```
DEPTNAME      LOCATION      NAME
-----+-----+-----
NEW ENGLAND   BOSTON       HANES
```

## Solutions to Exercise 6

1. To copy Q.STAFF as MYTABLE, enter:

```
DISPLAY Q.STAFF
SAVE DATA AS MYTABLE
```

2. The query that updates MYTABLE is as follows:

```
MYTABLE | NAME      | DEPT | JOB | YEARS | SALARY |
-----+-----+-----+-----+-----+
          | U.RAMOTH | 66   | MGR | U. 7  | U.18238.50 |
```

The following query retrieves the data just updated:

```
MYTABLE | ID | NAME      | DEPT | JOB | YEARS | SALARY | COMM |
-----+---+-----+-----+-----+-----+-----+-----+
P.      |   | RAMOTH   |      |    |       |        |      |
```

The report resulting from the query is as follows:

```
   ID  NAME      DEPT  JOB   YEARS  SALARY  COMM
-----+-----+-----+-----+-----+-----+-----
   270  RAMOTH      66    MGR    7     18238.50  -
```

3. The following query increases the salaries of all clerks by 10%:

```
MYTABLE | JOB   | YEARS | SALARY | SALARY |
-----+-----+-----+-----+-----+
          | CLERK | > 5   | _S     | U. _S * 1.1 |
```

## QBE exercises with solutions

The following query retrieves the rows just updated:

```
MYTABLE | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----|
P.      |   |     |     | CLERK |     |     |     |
```

The report resulting from the query is as follows:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
80	JAMES	20	CLERK	-	13504.60	128.20
110	NGAN	15	CLERK	5	12508.20	206.60
120	NAUGHTON	38	CLERK	-	12954.75	180.00
130	YAMAGUCHI	42	CLERK	6	11556.49	75.60
170	KERMISCH	15	CLERK	4	12258.50	110.10
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
190	SNEIDER	20	CLERK	8	15678.02	126.50
200	SCOUTTEN	42	CLERK	-	11508.60	84.20
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
250	WHEELER	51	CLERK	6	15906.00	513.30
330	BURKE	66	CLERK	1	10988.00	55.50
350	GAFNEY	84	CLERK	5	13030.50	188.00

4. The finished query that inserts a new employee row into MYTABLE is as follows:

```
MYTABLE | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----|
I.      | 275 | ROGERS | 66 | SALES | NULL | 14000 | NULL |
```

Instead of 14000 you can use 14000.00, but not 14,000 or 14,000.00 (commas are not valid numeric input characters).

The query to run to retrieve the data just inserted is as follows:

```
MYTABLE | ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
-----+-----+-----+-----+-----+-----+-----+-----|
P.      | 275 |     |     |     |     |     |     |
```

The report resulting from the query is as follows:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
275	ROGERS	66	SALES	-	14000.00	-

5. The finished query is as follows:

```
MYTABLE | DEPT | JOB |
-----+-----+-----|
D.      | 66   | SALES |
```

### Related concepts

[QMF sample tables](#)

## Appendix B. QMF sample tables

QMF provides sample tables that you can use to help you learn and test product functions. These tables contain data about applicants, interviews, parts, products, employees, and suppliers of a fictitious electrical parts manufacturer, J & H Supply Company.

Additionally, QMF Analytics for TSO provides the following sample tables that you can use to learn about QMF Analytics for TSO functions:

- Q.CASHFLOW
- Q.CLIMATE\_10YR
- Q.CLIMATE\_USA
- Q.WORLDINFO

### Q.APPLICANT

This table provides information about people who have applied for jobs with the company. Each row represents an applicant.

The columns are as follows:

**TEMPID**

Temporary identification of the applicant

**NAME**

Last name of the applicant

**ADDRESS**

City and state in which the applicant lives

**EDLEVEL**

Education level of the applicant

**COMMENTS**

Notes made by the interviewer

The Q.APPLICANT table is shown in the following figure:

TEMPID	NAME	ADDRESS	EDLEVEL	COMMENTS
400	FROMMHERZ	SAN JOSE,CA	12	NO SALES EXPERIENCE
410	JACOBS	POUGHKEEPSIE,NY	16	GOOD CANDIDATE FOR WASHINGTON
420	MONTEZ	DALLAS,TX	13	OFFER SALES POSITION
430	RICHOWSKI	TUCSON,AZ	14	CAN'T START WORK UNTIL 12/96
440	REID	ENDICOTT,NY	14	1 YEAR SALES EXPERIENCE
450	JEFFREYS	PHILADELPHIA,PA	12	GOOD CLERICAL BACKGROUND
460	STANLEY	CHICAGO,IL	11	WANTS PARTIME JOB
470	CASALS	PALO ALTO,CA	14	EXPERIENCED SALESMAN
480	LEEDS	EAST FISHKILL,NY	12	NEEDS INTERVIEW WITH BROWN
490	GASPARD	PARIS,TX	16	WORKED HERE FROM 1/94 TO 6/94

Figure 324. The Q.APPLICANT table

### Q.INTERVIEW

This table is for sites that support date/time data. It shows dates and times in ISO format. The format of DATE, TIME, and TIMESTAMP data in your reports depends on the format chosen as your site's default. It can be modified with edit codes that can be used with date, time, and timestamp data.

The columns in the Q.INTERVIEW table are as follows:

**TEMPID**

Temporary identification of the applicant

**INTDATE**

Date of the interview

**STARTTIME**

Time the interview started

**ENDTIME**

Time the interview ended

**MANAGER**

Employee number of the manager who interviewed the applicant

**DISP**

Whether or not the applicant will be hired

**LASTNAME**

Last name of the applicant

**FIRSTNAME**

First name of the applicant

The Q.INTERVIEW table is shown in the following table:

TEMPID	INTDATE	STARTTIME	ENDTIME	MANAGER	DISP	LASTNAME	FIRSTNAME
400	1996-02-05	13.00.00	15.12.00	270	NOHIRE	FROMMHERZ	RICHARD
410	1996-02-11	15.00.00	16.18.00	10	HIRE	JACOBS	SUSAN
420	1996-04-07	09.00.00	09.58.00	140	HIRE	MONTEZ	RITA
430	1996-04-24	10.30.00	11.30.00	290	NOHIRE	RICHOWSKI	JOHN
440	1996-03-13	10.15.00	11.23.00	160	HIRE	REID	CATHY
450	1996-09-19	09.45.00	11.00.00	50	HIRE	JEFFREYS	PAUL
460	1996-10-06	14.45.00	16.22.00	100	HIRE	STANLEY	JOHN
470	1996-02-05	16.30.00	18.00.00	270	HIRE	CASALS	DAVID
480	1996-03-13	13.30.00	14.45.00	160	NOHIRE	LEEDS	DIANE
490	1996-09-30	15.00.00	15.44.00	140	NOHIRE	GASPARD	PIERRE

Figure 325. The Q.INTERVIEW table

## Q.ORG

This table provides information on the organization of the company.

Each row represents a department. The columns are as follows:

**DEPTNUMB**

Number of the department (must be unique)

**DEPTNAME**

Descriptive name of the department

**MANAGER**

Employee number of the manager of the department

**DIVISION**

Division to which the department belongs

**LOCATION**

Name of the city in which the department is located

The Q.ORG table is shown in the following figure:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	HEAD OFFICE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
42	GREAT LAKES	100	MIDWEST	CHICAGO
51	PLAINS	140	MIDWEST	DALLAS
66	PACIFIC	270	WESTERN	SAN FRANCISCO
84	MOUNTAIN	290	WESTERN	DENVER

Figure 326. The Q.ORG table

## Q.PARTS

This table provides information about parts.

The columns are as follows:

**SUPPNO**

Number of the supplier

**PARTNAME**

Name of the part

**PRODUCT**

Product for which the part is needed

**PRODNO**

Number of the product

**PROJNO**

Number of the project

The Q.PARTS table is shown in the following figure:

SUPPNO	PARTNAME	PRODUCT	PRODNO	PROJNO
1100P	PLASTIC	RELAY	30	1501
1100P	STEEL	WRENCHSET	509	1520
1200S	WIRE	GENERATOR	10	1401
1200S	BEARINGS	MOTOR	50	1402
1300S	COPPER	RELAY	30	1501
1300S	BLADES	SAW	205	1510
1400P	MAGNETS	GENERATOR	10	1409
1400P	VALVES	MOTOR	50	1407
1400P	OIL	GEAR	160	1405

Figure 327. The Q.PARTS table

## Q.PRODUCTS

This table provides information about products and their prices.

The columns are as follows:

**PRODNUM**

Number of the product

**PRODNAME**

Descriptive name of the product

**PRODGRP**

General type of product

**PRODPRICE**

Price of the product

The Q.PRODUCTS table is shown in the following table:

PRODNUM	PRODNAME	PRODGRP	PRODPRICE
10	GENERATOR	ELECTRICAL	45.75
505	SCREWDRIVER	TOOL	3.70
101	SHAFT	MECHANICAL	8.65
20	SWITCH	ELECTRICAL	2.60
30	RELAY	ELECTRICAL	7.55
40	SOCKET	ELECTRICAL	1.40
50	MOTOR	ELECTRICAL	35.80
150	CAM	MECHANICAL	1.15
160	GEAR	MECHANICAL	9.65
190	BUSHING	MECHANICAL	5.90
205	SAW	TOOL	18.90
330	HAMMER	TOOL	9.35
450	CHISEL	TOOL	7.75
509	WRENCHSET	TOOL	25.90

Figure 328. The Q.PRODUCTS table

## Q.PROJECT

This table provides information about project schedules.

The columns are as follows:

**PROJNO**

Number of the project (must be unique)

**PRODNUM**

Number of the product

**DEPT**

Number of the department responsible for the project

**STARTD**

Date the project is to start

**ENDD**

Date the project is to end

**TIMESTAMP**

Year, month, day, and time of the report

The Q.PROJECT table includes date/time data, showing dates and times in ISO format. This format is an arbitrary choice. The table you see depends on the choice made by your administrator. The Q.PROJECT table is shown in the following figure:

PROJNO	PRODNUM	DEPT	STARTD	ENDD	TIMESTAMP
1401	10	20	1996-01-01	1998-03-31	1994-12-18-10.14.44.000001
1402	50	66	1996-01-30	1997-06-30	1994-12-18-10.15.01.999998
1403	150	51	1996-02-02	1999-05-29	1994-12-18-10.22.23.000001
1404	190	38	1997-01-04	1999-06-30	1994-12-18-10.25.43.999999
1405	160	15	1997-04-29	1999-10-30	1995-12-31-14.23.00.999999
1406	20	20	1997-07-11	1998-12-31	1996-01-05-13.31.18.009999
1407	50	42	1997-12-12	2000-06-15	1996-01-05-13.42.27.000000
1408	30	42	1999-03-13	2000-09-30	1996-01-05-13.44.16.999999
1409	10	66	1998-06-15	1999-12-31	1996-03-13-09.12.57.149572
1410	190	10	1998-09-29	2000-03-31	1996-03-13-12.18.23.402917
1501	30	51	1999-01-04	1999-12-31	1996-03-13-12.22.14.201966
1502	150	38	1999-03-01	2000-07-17	1996-03-13-13.17.48.948276

Figure 329. The Q.PROJECT table

## Q.SALES

This table provides data on sales orders made by sales representatives.

Its columns are as follows:

**ORDERNO**

The unique number of the parts order taken by the representative

**SALESREPNO**

Unique employee serial number of the sales representative who made the sale

**PRODNO**

The unique product number of the product sold

**QUANTITY**

The number of products ordered by the customer in the CUSTNO column

**CUSTNO**

Unique numeric identifier for each customer

The Q.SALES table is shown in the following figure:

ORDERNO	SALESREPNO	PRODNO	QUANTITY	CUSTNO
3456	20	10	50	1200
6667	20	160	120	4400
1991	40	150	600	4500
7777	60	30	150	8500
1020	60	30	150	8500
3333	70	50	240	9600
1115	70	101	120	8300
3580	20	190	360	4900
2345	90	450	360	2500
5770	70	205	100	8300
6432	40	150	120	8900
4432	90	505	150	2550
3455	150	190	360	8800
4477	220	330	480	5600
6540	150	150	200	8850
6688	280	150	300	6600
4080	300	101	500	5900
5456	300	20	60	6300
3360	310	101	120	3600
4596	310	160	100	2000
4321	340	330	200	3000
4567	40	450	100	4100
7010	20	505	150	3500
1550	90	160	200	4000
2888	90	50	240	5000
5432	220	20	100	6000
6677	40	10	150	9111
5521	60	50	150	9666
4010	150	205	225	4297
3968	220	509	200	7329
5832	280	509	300	7299
4491	300	50	100	5581
3962	340	10	240	3681

Figure 330. The Q.SALES table

## Q.STAFF

---

This table provides data on the employees of the J&H Supply Company.

The columns are as follows:

**ID**

Employee serial number (must be unique)

**NAME**

Name of the employee

**DEPT**

Department number of the employee

**JOB**

Classification of the employee's job

**YEARS**

Number of years the employee has worked for the company

**SALARY**

Employee's annual salary in dollars and cents

**COMM**

Employee's commission in dollars and cents

The Q.STAFF table is shown in the following figure:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	SANDERS	20	MGR	7	18357.50	-
20	PERNAL	20	SALES	8	18171.25	612.45
30	MARENGHI	38	MGR	5	17506.75	-
40	O'BRIEN	38	SALES	6	18006.00	846.55
50	HANES	15	MGR	10	20659.80	-
60	QUIGLEY	38	SALES	-	16808.30	650.25
70	ROTHMAN	15	SALES	7	16502.83	1152.00
80	JAMES	20	CLERK	-	13504.60	128.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
100	PLOTZ	42	MGR	7	18352.80	-
110	NGAN	15	CLERK	5	12508.20	206.60
120	NAUGHTON	38	CLERK	-	12954.75	180.00
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
140	FRAYE	51	MGR	6	21150.00	-
150	WILLIAMS	51	SALES	6	19456.50	637.65
160	MOLINARE	10	MGR	7	22959.20	-
170	KERMISCH	15	CLERK	4	12258.50	110.10
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
190	SNEIDER	20	CLERK	8	14252.75	126.50
200	SCOUTTEN	42	CLERK	-	11508.60	84.20
210	LU	10	MGR	10	20010.00	-
220	SMITH	51	SALES	7	17654.50	992.80
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
240	DANIELS	10	MGR	5	19260.25	-
250	WHEELER	51	CLERK	6	14460.00	513.30
260	JONES	10	MGR	12	21234.00	-
270	LEA	66	MGR	9	18555.50	-
280	WILSON	66	SALES	9	18674.50	811.50
290	QUILL	84	MGR	10	19818.00	-
300	DAVIS	84	SALES	5	15454.50	806.10
310	GRAHAM	66	SALES	13	21000.00	200.30
320	GONZALES	66	SALES	4	16858.20	844.00
330	BURKE	66	CLERK	1	10988.00	55.50
340	EDWARDS	84	SALES	7	17844.00	1285.00
350	GAFNEY	84	CLERK	5	13030.50	188.00

Figure 331. The Q.STAFF table

## Q.SUPPLIER

This table provides data on the suppliers of the J&H Supply Company.

The columns are as follows:

**ACCTNO**

The account number of the company

**COMPANY**

The name of the company

**STREET**

The street address of the company

**CITY**

The city in which the company is located

**STATE**

The state in which the company is located

**ZIP**

The company's zip code



## NOTES

Other information about the supplier

The Q.SUPPLIER table is shown in the following figure:

ACCTNO	COMPANY	STREET	CITY	STATE	ZIP	NOTES
1100P	WESTCO, INC.	1900 115TH ST.	EMERYVILLE	CA	16600	THIS COMPANY HAS A STRONG HISTORY OF ON-TIME DELIVERY. WESTCO IS GROWING QUICKLY.
1200S	MAJOR ELECTRICS	4250 BENSON ST.	DALLAS	TX	87050	MAJOR ELECTRICS DECLARED BANKRUPTCY IN 1987, BUT HAS RECOVERED. FORESEE NO FURTHER PROBLEMS.
1300S	FRANKLIN, INC.	40025 EASTLAND	DOVER	DE	99000	DUE TO ITS LOCATION ON EASTERN SEABOARD, FRANKLIN HAS EXCELLENT TRANSPORTATION FACILITIES.
1400P	MOTORWORKS , INC.	19503 BESWICK	JOLIET	IL	12000	PROXIMITY TO CHICAGO ENSURES GOOD TRANSPORTATION, BOTH BY RAIL AND TRUCK. A RELIABLE SUPPLIER.

Figure 332. The Q.SUPPLIER table

You might have to adjust the column width on either FORM.MAIN or FORM.COLUMNS to see all the information in the NOTES column.

## Q.CASHFLOW

This sample table provides data about cost and revenue that can be used with QMF Analytics for TSO discounted cash flow analysis.

The columns are as follows:

### PERIOD

The accounting period

### COSTS

Costs for the period

### REVENUE

Revenue for the period

### CASHFLOW

The calculated cashflow for the period

An excerpt of the Q.CASHFLOW table is shown in the following figure:

PERIOD	COSTS	REVENUE	CASHFLOW
1	-800	0	-800
2	-600	0	-600
3	-100	200	100
4	0	400	400
5	0	500	500
6	-800	300	-500

Figure 333. Excerpt from the Q.CASHFLOW table

## Q.CLIMATE\_10YR

This QMF Analytics for TSO sample table provides data about climate over a 10-year period.

The columns are as follows:

### YEAR

The year that the climate data pertains to

### MONTH

The month that the climate data pertains to

### TEMPMIN

The lowest temperature (in Fahrenheit) for the month

### TEMPMAX

The highest temperature (in Fahrenheit) for the month

### RAINFALL

The amount of rainfall for the month in inches

### SUNSHINE

Number of hours of sunshine for the month

An excerpt of the Q.CLIMATE\_10YR table is shown in the following figure:

YEAR	MONTH	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
2001	1	9	70	3	234
2001	2	18	72	7	205
2001	3	16	77	12	180
2001	4	32	91	3	230
2001	5	32	95	4	234
2001	6	41	115	3	230
2001	7	43	111	1	227
2001	8	39	115	3	238
2001	9	32	93	6	226
2001	10	27	88	6	221
2001	11	14	79	12	183
2001	12	19	73	9	204
2002	1	27	77	11	185
2002	2	25	75	45	25
2002	3	25	88	17	161
2002	4	32	90	3	226
2002	5	32	100	2	225
2002	6	43	108	4	241
2002	7	46	111	4	228
2002	8	45	113	2	238
2002	9	39	102	9	197
2002	10	32	90	3	227
2002	11	27	79	5	221
2002	12	23	66	8	219

Figure 334. Excerpt from the Q.CLIMATE\_10YR table

## Q.CLIMATE\_USA

This QMF Analytics for TSO sample table provides data about climate in the United States, including data about rainfall and sunshine.

The columns are as follows:

**MONTH**

The month that the climate data pertains to

**STATE**

The two-letter abbreviation for the state that the climate data pertains to

**TEMPMIN**

The lowest temperature (in Fahrenheit) for the month

**TEMPMAX**

The highest temperature (in Fahrenheit) for the month

**RAINFALL**

The amount of rainfall for the month in inches

**SUNSHINE**

Number of hours of sunshine for the month

An excerpt from the Q.CLIMATE\_USA table is shown in the following figure:

MONTH	STATE	TEMPMIN	TEMPMAX	RAINFALL	SUNSHINE
1	AK	9	25	1	1
1	AL	23	61	11	34
1	AR	19	50	12	1
1	AZ	12	68	7	80
1	CA	19	72	18	41
1	CO	-13	50	1	51
1	CT	18	39	3	1
1	DE	25	37	3	11
1	FL	34	70	12	83
1	GA	25	55	9	45
1	HI	63	82	7	133
1	IA	-6	27	4	1
1	ID	12	50	4	29
1	IL	9	36	2	10
1	IN	9	32	4	1
1	KS	9	54	3	50
1	KY	18	36	6	1
1	LA	27	63	10	51
1	MA	12	39	5	1
1	MD	19	39	9	1
1	ME	5	28	4	1
1	MI	9	34	3	1
1	MN	-2	19	3	1
1	MO	9	37	10	1
1	MS	28	57	6	44
1	MT	1	39	5	1
1	NC	14	50	8	23
1	ND	-6	23	3	1
1	NE	3	41	3	10
1	NH	10	32	1	1
1	NJ	21	41	2	11
1	NM	5	54	4	48
1	NV	10	57	2	58
1	NY	5	41	6	1
1	OH	18	39	2	1
1	OK	18	55	3	41
1	OR	23	52	15	9
1	PA	12	37	7	1
1	RI	16	34	3	1
1	SC	27	55	11	34
1	SD	0	34	1	3
1	TN	18	46	8	2
1	TX	19	66	4	97
1	UT	-2	54	3	41
1	VA	21	43	12	1
1	VT	12	34	3	1
1	WA	19	50	28	1
1	WI	1	25	3	1
1	WV	18	39	4	1
1	WY	0	37	1	16

Figure 335. Excerpt from the Q.CLIMATE\_USA table

## Q.WORLDINFO

This QMF Analytics for TSO sample table provides data about geographic regions in which the J & H Supply Company conducts business.

The columns are as follows:

### **COUNTRY ID**

The three-digit numeric (ISO 3166-1) country code for the country

### **ALPHA 3**

The three-letter (ISO 3166-1) country code for the country

### **ALPHA2**

The two-letter (ISO 3166-1) country code for the country

### **COUNTRY**

The country name

An excerpt of the Q.WORLDINFO table is shown in the following figure:

COUNTRY ID	ALPHA3	ALPHA2	COUNTRY
4	AFG	AF	AFGHANISTAN
8	ALB	AL	ALBANIA
10	ATA	AQ	ANTARCTICA
12	DZA	DZ	ALGERIA
16	ASM	AS	AMERICAN SAMOA
20	AND	AD	ANDORRA
24	AGO	AO	ANGOLA
28	ATG	AG	ANTIGUA AND BARBUDA
31	AZE	AZ	AZERBAIJAN
32	ARG	AR	ARGENTINA
36	AUS	AU	AUSTRALIA
40	AUT	AT	AUSTRIA
44	BHS	BS	BAHAMAS
48	BHR	BH	BAHRAIN
50	BGD	BD	BANGLADESH
51	ARM	AM	ARMENIA
52	BRB	BB	BARBADOS
56	BEL	BE	BELGIUM
60	BMU	BM	BERMUDA
64	BTN	BT	BHUTAN
68	BOL	BO	BOLIVIA
70	BIH	BA	BOSNIA AND HERZEGOVINA
72	BWA	BW	BOTSWANA
74	BVT	BV	BOUVET ISLAND
76	BRA	BR	BRAZIL
84	BLZ	BZ	BELIZE
86	IOT	IO	BRITISH INDIAN OCEAN TERRITORY
90	SLB	SB	SOLOMON ISLANDS
92	VGB	VG	VIRGIN ISLANDS, BRITISH
96	BRN	BN	BRUNEI DARUSSALAM
100	BGR	BG	BULGARIA
104	MMR	MM	MYANMAR
108	BDI	BI	BURUNDI
112	BLR	BY	BELARUS
116	KHM	KH	CAMBODIA
120	CMR	CM	CAMEROON
124	CAN	CA	CANADA
132	CPV	CV	CAPE VERDE
136	CYM	KY	CAYMAN ISLANDS
140	CAF	CF	CENTRAL AFRICAN REPUBLIC
144	LKA	LK	SRI LANKA
148	TCD	TD	CHAD
152	CHL	CL	CHILE
156	CHN	CN	CHINA

Figure 336. Excerpt from the Q.WORLDINFO table

## Appendix C. QMF functions that require specific support

Support for these functions varies with the database or environment.

### Functions that vary according to database type

Support for these functions varies by database.

Function supported	Db2 for z/OS	Db2 for Linux, UNIX, and Windows	DB2 for iSeries	DB2 for VSE and VM
Length of query statement supported	2 MB*	65 KB*	65 KB*	8 KB
Number of columns in SELECT statement	750	255	255	255
Importing single-precision floating point numbers	X			X
Long fields with LIKE statement	X			X
Database synonyms				X
Database aliases for tables or views	X	X	X	
SAVE=IMMEDIATE option available in Table Editor (supports CURSOR HOLD)	X	X	X	
Setting Db2 global variables	X	X		
QMF commands that include three-part names	Commands with three-part names can be initiated from this type of database. They can also be directed to this type of database unless QMF was started as a stored procedure.	Commands with three-part names can be directed to this type of server unless QMF was started as a stored procedure.	Commands with three-part names can be directed to this type of server unless QMF was started as a stored procedure.	Commands with three-part names cannot be directed to these server types.

\* To activate support for SQL queries up to 2 MB on Db2 for z/OS databases and up to 65 KB on Db2 for Linux, UNIX, and Windows databases, set the DSQEC\_SQLQRYSZ\_2M global variable to 1 before running the query.

## Functions not available in CICS

---

Certain functions are supported by TSO only.

The following functions are not available in CICS:

- Use of multiple thread support.
- Use of QMF Analytics for TSO
- Use of the QMF Enhanced Editor.
- Use of extended storage for spilling report data no longer needed in active storage; a spill file must be used instead
- Ability to start QMF as a Db2 for z/OS stored procedure
- Interfaces:
  - Command interface
  - Document interface
- Program parameters:
  - DSQSCMD (QMF callable interface only)
  - DSQSMTHD
  - DSQSPLAN
  - DSQSPRID
  - DSQSPTYP
  - DSQSRSTG
  - DSQSSUBS
- Commands:
  - BATCH (and its associated application)
  - DPRE (and its associated REXX exec)
  - EDIT QUERY
  - EDIT PROC
  - ISPF (and its associated application)
  - LAYOUT (and its associated application)
  - SET GLOBAL commands that reference the following global variable:
    - DSQEC\_EXTND\_STG
  - SHOW GLOBAL commands that reference the following global variables:
    - DSQEC\_EXTND\_STG
    - DSQAO\_STO\_PROC\_INT
  - STATE (requires the command interface)
- Macros: GETQMF
- Form functions:
  - Report calculations or expressions that require REXX
  - Conditional formatting
  - Column definition
  - Locally defined edit codes TDL and TTL (for formatting dates and times, respectively)
- Procedures with logic (which require REXX)
- The ability to cancel transactions

- The ability to update data at remote locations (all tables and views at remote locations are read-only in QMF for CICS)
- External variables





## Notices

---

This information was developed for products and services offered in the US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at <http://www.ibm.com/legal/copytrade.shtml>.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions:

**Applicability:** These terms and conditions are in addition to any terms of use for the IBM website.

**Personal use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

**Commercial use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

**Rights:** Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## Privacy policy considerations

---

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience,

to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



# Glossary of terms and acronyms

---

**abnormal end of task (abend)**

The termination of a task, job, or subsystem because of an error condition that recovery facilities cannot resolve during execution.

**address space**

The range of addresses available to a computer program or process. Address space can refer to physical storage, virtual storage, or both.

**Advanced Program-to-Program Communication**

See *APPC*.

**aggregate function**

Any of a group of functions that summarizes data in a column. They are requested with these usage codes on the form panels: AVERAGE, CALC, COUNT, FIRST, LAST, MAXIMUM, MINIMUM, STDEV, SUM, CSUM, PCT, CPCT, TPCT, TCPCT.

**aggregation variable**

An aggregation function that is placed in a report using the FORM.BREAK, FORM.CALC, FORM.DETAIL, or FORM.FINAL panels. Its value appears as part of the break footing, detail block text, or final text when the report is produced.

**alias**

An alternative name used to identify a table, view, database, or nickname. An alias can be used in SQL statements to refer to a table, view, or database in the same DB2 system or subsystem or in a remote DB2 system or subsystem.

**APAR (Authorized Program Analysis Report)**

A request for correction of a defect in a supported release of an program supplied by IBM.

**APF (authorized program facility)**

In a z/OS environment, a facility that permits the identification of programs that are authorized to use restricted functions.

**API (application programming interface)**

An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

**application**

One or more computer programs or software components that use QMF services to provide functionality in direct support of a specific business process or processes.

**APPC (Advanced Program-to-Program Communication)**

An implementation of the SNA LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

**application plan**

The control structure that is produced during the bind process. The default name for the QMF Version 13 Release 1 application plan is QMF1310.

**application programming interface**

See *API*.

**application requester**

The source of a request to a remote DRDA-enabled relational database management system (RDBMS). Only Db2 for z/OS databases can function as application requesters because this is the only type of database in which QMF can be started.

**application server**

The target of a request from an application requester. The database management system (DBMS) at the application server site services the request. Connectivity with remote servers is not supported when QMF for TSO is running as a Db2 for z/OS stored procedure.

**argument**

A value passed to or returned from a function or procedure at run time.

**authorization identifier (authorization ID)**

A character string that designates a set of privileges and can be used to verify authority. An authorization ID can represent an object, an individual user, an organizational group, a function, or a database role. QMF authenticates either the database authorization ID or, optionally, the QMF TSO logon ID, against the CREATOR column of the Q.PROFILES table during QMF initialization.

**Authorized Program Analysis Report**

See *APAR*.

**Authorized program facility**

See *APF*.

**auxiliary table**

A table that stores columns outside the table in which they are defined. See also *base table*.

**base product**

The English-language version of QMF, established when QMF is installed. Any other language environment is established after installation by installing the National Language Feature (NLF) associated with that language.

**base table**

A table that is created by the SQL CREATE TABLE statement and that holds persistent data.

**binary string**

A sequence of bytes that is not associated with a coded character set and therefore is never converted. For example, the BLOB data type is a binary string. See also *CCSID*.

**bind**

To convert the output from the DBMS precompiler to a usable control structure, such as an access plan, an application plan, or a package.

**bit data**

Data with a data type of CHAR or VARCHAR that is not associated with a coded character set and therefore is never converted.

**buffer pool**

An area of memory into which data pages are read and in which they are modified and held during processing. See also *address space*.

**built-in function**

A strongly typed, high-performance function that is integral to the DB2 database. A built-in function can be referenced in SQL statements anywhere that an expression is valid.

**CAF (call attachment facility)**

A Db2 for z/OS attachment facility for application programs that run in TSO or z/OS batch. The CAF is an alternative to the DSN command processor and provides greater control over the execution environment.

**call attachment facility**

See *CAF*.

**callable interface**

A programming interface that provides access to QMF objects and services.

**cascade delete**

A process by which the DB2 database manager enforces referential constraints by deleting all descendent rows of a deleted parent row.

**catalog**

A collection of tables and views that contains descriptions of objects such as tables, views, and indexes. See also *QMF object catalog*.

**CCSID (coded character set identifier)**

A 16-bit number that includes a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic-character

representation. Because QMF uses display services provided by GDDM, the GDDM application code page must agree with the CCSIDs in use for the database. See also *binary string*.

**character string**

A sequence of bytes that represents bit data, single-byte characters, or a mixture of single-byte and multibyte characters.

**check constraint**

A user-defined constraint that specifies the values that specific columns of a base table can contain. See also *constraint*.

**CICS (Customer Information Control System)**

An IBM licensed program that provides online transaction-processing services and management for business applications.

**clause**

In SQL, a distinct part of a statement in the language structure, such as a SELECT clause or a WHERE clause.

**CM (Compatibility Mode)**

An installation mode of QMF Version 8.1 and QMF Version 9.1 that limited owner and object names in the QMF object catalog to eight and 18 characters, respectively. See also *NFM*.

**code page**

A particular assignment of code points to graphic characters. Within a given code page, a code point can have only one specific meaning. A code page also identifies how undefined code points are handled.

**coded character set identifier**

See *CCSID*.

**coexistence**

The state during which two QMF releases exist in the same DB2 subsystem. QMF Version 13.1 can coexist with QMF Version 9.1 New Function Mode or QMF Version 8.1 New Function Mode only.

**column**

The vertical component of a database table. A column has a name and a particular data type (for example, character, decimal, or integer).

**column function**

See *aggregate function*.

**column wrapping**

The value formatting in a report where the values occupy several lines within a column. Column wrapping is often used when a column contains values whose length exceeds the column width, such as cases requiring the display of XML data.

**command interface**

An interface for issuing QMF commands. The command interface allows you to issue QMF commands from an ISPF dialog running under QMF. Using this interface, QMF communicates with the dialog through the ISPF variable pool.

**command synonym**

The verb or verb/object part of a site-defined command. After command synonyms are defined and activated in the QMF profile, users can enter the synonyms on the QMF command line as they do with regular QMF commands.

**command synonym table**

A table that stores one site-defined command in each row. You assign a set of command synonyms to a user by storing the name of this table in the user's profile.

**comparison operator**

In SQL, a symbol used in comparison expressions to specify a relationship between two values. Comparison operators are = (equal to), <> (not equal to), < (less than), > (greater than), <= (less than or equal to), and >= (greater than or equal to).

**Compatibility Mode**

See *CM*.

**commit**

To end a unit of work by releasing locks so that the database changes made by that unit of work can be perceived by other processes. This operation makes the data changes permanent.

**concatenation**

Joining two characters or strings to form one string.

**connection**

In data communication, an association established between entities for conveying information. See also *SQL connection*. Connectivity with remote servers is not supported when QMF for TSO is running as a Db2 for z/OS stored procedure.

**constant**

A language element that specifies an unchanging value. Constants are classified as string constants or numeric constants.

**constraint**

A rule that limits the values that can be inserted, deleted, or updated in a table.

**control section**

See *CSECT*.

**control tables**

A set of tables that QMF uses to store information about QMF objects and manage QMF operations. See also *QMF object catalog*.

**correlated reference**

A reference to a column of a table or view that is outside a subquery.

**correlation name**

An identifier specified and used within a single SQL statement as the exposed name for objects such as a table, view, table function reference, nested table expression, or data change table reference. Correlation names are useful in an SQL statement to allow two distinct references to the same base table and to allow an alternative name to be used to represent an object.

**CSECT (control section)**

The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

**current location**

The application server to which the QMF session is currently connected. After the connection is made, this server processes all SQL statements. When initializing QMF, the current location can be indicated using the DSQSDBNM startup parameter. Connectivity with remote servers is not supported when QMF for TSO is running as a Db2 for z/OS stored procedure.

**current object**

A QMF object that is held in temporary storage so that, with each use, it can be readily accessed without requiring database retrieval. There are seven temporary storage areas: QUERY, FORM, PROC, PROFILE, REPORT, DATA, and CHART. Users can navigate to all areas but the DATA area using the SHOW and DISPLAY commands. See also *temporary storage*.

**cursor**

A named control structure used by an application program to point to and select a row of data from a set.

**Customer Information Control System**

See *CICS*.

**data type**

A classification identifying one of various kinds of data. In SQL, the data type is an attribute of columns, literals, host variables, special registers, parameters, and the results of functions and expressions.

**database**

A collection of interrelated or independent data items that are stored together to serve one or more applications.



**database administrator**

A person who is responsible for the design, development, operation, security, maintenance, and use of a database.

**database management system**

See *DBMS*.

**database manager**

A program that manages data by providing centralized control, data independence, and complex physical structures for efficient access, integrity, recovery, concurrency control, privacy, and security.

**database server**

A software program that uses a database manager to provide database services to other software programs or computers.

**DBCS (double-byte character set)**

A set of characters in which each character is represented by two bytes. These character sets are commonly used by national languages such as Japanese and Chinese, which have more symbols than can be represented by a single byte. See also *SBCS*.

**DBMS (database management system)**

A software system that controls the creation, organization, and modification of a database and the access to the data that is stored within it.

**DCT (destination control table)**

A table describing each of the transient data destinations used in CICS. This table contains an entry for each extrapartition, intrapartition, and indirect destination.

**default form**

The QMF form created when a saved form is not specified on the RUN QUERY command.

**default value**

A predetermined value, attribute, or option that is assumed when no other value is specified. A default value can be defined for column data in DB2 tables by specifying the DEFAULT keyword in an SQL statement that changes data (such as INSERT, UPDATE, and MERGE).

**dependent row**

A row that contains a foreign key that matches the value of a parent key in the parent row. The foreign key value represents a reference from the dependent row to the parent row.

**dependent table**

A table that is a dependent of an object. For example, a table with a foreign key is a dependent of the table containing the corresponding primary key.

**destination control table**

See *DCT*.

**detail block text**

The text in the body of a report that is associated with a particular row of data.

**detail heading text**

The text in the heading of a report.

**detail variation**

A data formatting definition specified on a FORM.DETAIL panel that can be used to conditionally format a report or part of a report.

**distinct type**

A user-defined data type that shares a common representation with built-in data types.

**distributed data**

Data that is stored on more than one system and is available to remote users and application programs.

**distributed database**

A database that appears to users as a logical whole, locally accessible database, but consists of databases in multiple locations that are connected by a data communications network.

**Distributed Relational Database Architecture™**

See *DRDA*.

**distributed unit of work**

A form of distributed relational database processing that enables a user or application program to read or update data at multiple locations within a unit of work. Within one unit of work, an application, such as QMF, running in one system can direct SQL requests to multiple remote database management systems using the SQL supported by those systems. The request is made through a QMF command that includes a three-part table or view name. QMF commands with three-part names cannot be directed to DB2 for VM or VSE databases or used when QMF for TSO has been started as a Db2 for z/OS stored procedure. Three-part names in QMF commands also cannot refer to a table that contains large object (LOB) data types.

**double-byte character set**

See *DBCS*.

**double-precision floating-point number**

A 64-bit approximate representation of a real number.

**DRDA (Distributed Relational Database Architecture)**

The architecture that defines formats and protocols for providing transparent access to remote data. DRDA defines two types of functions: the application requester function and the application server function.

**environment**

A named collection of logical and physical resources used to support the performance of a function.

**exit routine**

A program that receives control from another program to perform specific functions.

**Extensible Markup Language**

See *XML*.

**extended syntax**

Syntax that is used for the QMF SET GLOBAL and GET GLOBAL commands and certain function calls. Extended syntax defines parameters used by QMF callable interface applications written in Assembler, C, COBOL, Fortran, or PL/I.

**fallback**

The process of returning to a prior release of a software program after attempting or completing migration to a current release.

**fetch**

The process of retrieving rows from the database or a file to create a QMF DATA object. QMF supports multirow fetch through the use of the DSQSMRFI parameter.

**foreign key**

In a relational database, a key in one table that references the primary key in another table.

**GDDM (Graphical Data Display Manager)**

Graphics software that defines and displays text and graphics for output on a display device or printer.

**global variable**

A named entity whose value persists for the duration of a QMF session. QMF uses global variables to manage both session and database activity. Some global variables can be set with the SET GLOBAL command, while others record information about the state of the current QMF session and therefore cannot be set.

**graphic string**

A sequence of double-byte character set (DBCS) characters.

**Graphical Data Display Manager**

See *GDDM*.

**host**

The controlling or highest-level system in a data communications configuration.

**HTML (hypertext markup language)**

A markup language that conforms to the Standard Generalized Markup Language (SGML) standard and was designed primarily to support the online display of textual and graphical information, including hypertext links.

**hypertext markup language**

See *HTML*.

**ICU (Interactive Chart Utility)**

A menu-driven component of IBM's Graphical Data Display Manager (GDDM) product that allows non-programmers to display, print, or plot charts, graphs, and diagrams.

**identity column**

A column that provides a way for the DB2 database manager to automatically generate a numeric value for each row that is inserted into a table. Identity columns are defined with the AS IDENTITY clause. A table can have no more than one identity column.

**index**

A set of pointers that is logically ordered by the values of a key. Indexes provide quick access to data and can enforce uniqueness of the key values for the rows in the table.

**inner join**

The result of a join operation that includes only the matched rows of both tables that are being joined. See also *outer join*.

**installation verification procedure**

See *IVP*.

**Integrated Exchange Format**

See *IXF*.

**Interactive Chart Utility**

See *ICU*.

**Interactive System Productivity Facility**

See *ISPF*.

**ISPF (Interactive System Productivity Facility)**

An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and terminal user.

**IVP (installation verification procedure)**

A procedure or program whose purpose is to verify that a product has been correctly installed.

**IXF (Integrated Exchange Format)**

A protocol for transferring tabular data among various software products.

**JCL (job control language)**

A command language that identifies a job to an operating system and describes the job's requirements.

**job control language**

See *JCL*.

**join**

An SQL relational operation that allows retrieval of data from two or more tables based on matching column values.

**key**

A column or an ordered collection of columns that is identified in the description of a table, index, or referential constraint. The same column can be part of more than one key.

**keyword**

One of the predefined words of a programming language, artificial language, application, or command.

**keyword parameter**

A parameter that consists of a keyword followed by one or more values. See also *positional parameter*.

**large object**

See *LOB*.

**link-edit**

To create a loadable computer program by means of a linkage editor.

**linkage editor**

A computer program for creating load modules from one or more object modules or load modules by resolving cross-references among the modules and, if necessary, adjusting addresses.

**literal**

A character string whose value is defined by the characters themselves. For example, the numeric constant 7 has the value 7, and the character constant 'CHARACTERS' has the value CHARACTERS.

**linear procedure**

A sequenced set of QMF commands or command synonyms that can be used to perform several operations at once. See also *procedure with logic*.

**linear syntax**

QMF command syntax that is entered in one statement of a program or procedure, or that can be entered on the QMF command line.

**load module**

A program in a form suitable for loading into main storage for execution.

**LOB (large object)**

A sequence of bytes with a size ranging from 0 bytes to 2 gigabytes (less 1 byte). There are three LOB data types: binary large object (BLOB), character large object (CLOB, which can include single-byte characters only or a mixture of single-byte and double-byte characters), and double-byte character large object (DBCLOB). QMF supports a LOB column size of up to 32 KB.

**local**

Pertaining to databases, objects, or applications that are installed or stored in the system in which QMF is currently running.

**location**

A specific relational database server in a distributed relational database system. Each location has a unique location name.

**location name**

The unique name of a database server. An application uses the location name to access a DB2 database server.

**lock**

A means of serializing a sequence of events or serializing access to data.

**log**

A collection of records that sequentially describes the events that occur in a system.

**LUW**

An abbreviation for Linux, UNIX, and Windows.

**National Language Feature**

See *NLF*.

**New Function Mode**

See *NFM*.

**NFM (New Function Mode)**

An installation mode of QMF Version 8.1 and QMF Version 9.1 that allowed owner and object names in the QMF object catalog to be the maximum length allowed by the database. QMF Version 13 Release 1 allows owner and object names to be as long as the database allows as well. See also *CM*.

**NLF (National Language Feature)**

Any of several optional features available with QMF. NLFs allow users to interact with QMF in specific native languages.

**object**

A named storage space that consists of a set of characteristics that describe the space and, in some cases, data. An object is anything that occupies space in storage, can be located in a library or directory, can be secured, and on which defined operations can be performed. See also *QMF object*.

**outer join**

The result of a join operation that includes the matched rows of both tables that are being joined and preserves some or all of the unmatched rows of the tables that are being joined. See also *inner join*.

**package**

A control-structure database object produced during program preparation that can contain both executable forms of static SQL statements or XQuery expressions and placement holders for executable forms of dynamic SQL statements.

**panel**

A formatted display of information on a screen that can also include entry fields.

**parameter**

A value or reference passed to a function, command, or program that serves as input or controls actions. The value is supplied by a user or by another program or process.

**partition**

A portion of a page set. Each partition corresponds to a single, independently extendable data set. Partitions can be extended to a maximum size of 1, 2, or 4 gigabytes, depending on the number of partitions in the partitioned page set. All partitions of a given page set have the same maximum size.

**plan**

See *application plan*.

**positional parameter**

A parameter that must appear in a specified location, relative to other parameters. See also *keyword parameter*.

**precision**

An attribute of a number that describes the total number of significant digits.

**predicate**

An element of a search condition that expresses or implies a comparison operation.

**primary authorization ID**

The authorization identifier used to identify an application process to DB2 for z/OS.

**primary key**

In a relational database, a key that uniquely identifies one row of a database table.

**privilege**

In SQL, a capability given to a user by the processing of a GRANT statement.

**procedure**

A sequenced set of statements or commands used to perform one or more tasks. See also *linear procedure* and *procedure with logic*.

**procedure with logic**

A set of statements that performs one or more tasks. A procedure with logic begins with a REXX comment and allows conditional logic (which uses REXX), calculations, build strings, and TSO or CICS commands. See also *linear procedure*.

**profile**

An object that contains information about the characteristics of the user's session.

**program temporary fix**

See *PTF*.

**prompted query**

A menu-driven query controlled by user-provided parameters.

**PTF (program temporary fix)**

For System i®, System p, and System z®, products, a fix that is tested by IBM and is made available to all customers.

**QBE (Query-by-Example)**

A component of QMF that allows users to create queries graphically.

**QMF administrator authority**

Authority that allows a user to insert or delete rows in the Q.PROFILES control table. Users with this authority can perform the following commands on QMF queries, forms, and procedures that are owned by other users without forcing the owners to share these objects with all users: SAVE, ERASE, IMPORT, EXPORT, and DISPLAY. QMF checks each user ID for administrator authority during

initialization; you can disable this checking by setting the DSQEC\_DISABLEADM variable in the DSQUOPTS exit routine or in another program of your choice.

**QMF administrator**

A user who has QMF administrator authority.

**Query-by-Example**

See *QBE*.

**QMF object**

An object used by QMF users to query, format, and present data or otherwise manage interaction between QMF and the database. QMF objects include queries and query result data, forms, procedures, reports, charts, and the QMF profile. Each QMF object has a named temporary storage area that is used to display the object. All objects except reports and charts can be saved in the database; reports and charts are created dynamically upon user request by applying the formatting specifications of a particular QMF form to result data that has been returned from the database. See also *temporary storage*.

**QMF object catalog**

A set of control tables that stores information about QMF queries, procedures, and forms. These control tables include Q.OBJECT\_DIRECTORY, Q.OBJECT\_DATA, and Q.OBJECT\_REMARKS.

**qualifier**

When referring to a QMF object, the part of the name that identifies the owner or the location of an object. When referring to a TSO data set, any part of the name that is separated from the rest of the name by periods. For example, 'TCK', 'XYZ', and 'QUERY' are all qualifiers in the data set name 'TCK.XYZ.QUERY'.

**query**

A request for information from a database based on specific conditions: for example, a request for a list of all customers in a customer table whose balances are greater than \$1000. In QMF, a query also refers to SQL statements submitted from the Prompted Query, QBE, or SQL query panel, even if these statements are not requests for information (SELECT statements).

**RCT (resource control table)**

A DB2 control table that defines the relationship between CICS transactions and DB2 resources.

**RDBMS (relational database management system)**

A collection of hardware and software that organizes and provides access to a relational database.

**RDO (resource definition online)**

In CICS, a facility that allows the user to define certain CICS resources interactively while CICS is running. Specifically, RDO allows the user to define terminals, programs, and transactions interactively.

**record**

The storage representation of a row or other data.

**record length**

The length of storage that represents a row or other data.

**reentrant**

Executable code that can reside in storage as one shared copy for all database threads. Reentrant code is not self-modifying and provides separate storage areas for each thread.

**referential constraint**

The requirement that the nonnull values of a designated foreign key are valid only if they also appear as values of the primary key of the parent table. The referential constraint is always defined from the perspective of the dependent file.

**relational database**

A database that can be perceived as a set of tables and manipulated in accordance with the relational model of data. Each database includes a set of system catalog tables that describe the logical and physical structure of the data, a configuration file containing the parameter values allocated for the database, and a recovery log with ongoing transactions and archivable transactions.

**relational database management system**

See *RDBMS*.

**remote**

Pertaining to databases, objects, or applications that are installed or stored on a system other than the system where QMF is currently executing. You can access objects (including QMF queries, forms, and procedures) at a remote server by using the QMF CONNECT command. You can also use a QMF command with a three-part table or view name if you want to access just tables or views at a remote location. Remote access is not permitted when QMF for TSO is running as a Db2 for z/OS stored procedure.

**remote unit of work**

A form of distributed relational database processing in which an application program, such as QMF, can access data on a remote database within a unit of work. The connection is established by the QMF CONNECT command. The CONNECT command cannot be used when QMF for TSO is running as a Db2 for z/OS stored procedure.

**requester**

See *application requester*.

**resource**

The object of a lock or claim, which could be a table space, an index space, a data partition, an index partition, or a logical partition.

**resource control table**

See *RCT*.

**resource definition online**

See *RDO*.

**Restructured Extended Executor**

See *REXX*.

**REXX (Restructured Extended Executor)**

A general-purpose, high-level programming language, particularly suitable for EXEC procedures or programs for personal computing.

**roll back**

To restore data that is changed by an SQL statement to the state at its last commit point. Changes made by all SQL statements in a multistatement query, except those that affect the QMF session (such as SET), are rolled back in the event of a failure.

**routine**

A program or sequence of instructions called by a program. Typically, a routine has a general purpose and is frequently used.

**row**

The horizontal component of a table, consisting of a sequence of values, one for each column of the table.

**runtime variable**

A variable in a procedure or query whose value is specified by the user when the procedure or query is run. The value of a runtime variable is only available in the current procedure or query. See also *global variable*.

**SBCS (single-byte character set)**

A coded character set in which each character is represented by a 1-byte code. A 1-byte code point allows representation of up to 256 characters. See also *double-byte character set*.

**scalar function**

An SQL function that optionally accepts arguments and that returns a single scalar value each time that it is invoked. A scalar function can be referenced in an SQL statement wherever an expression is valid.

**scratchpad area**

A work area used in conversational processing to retain information from an application program across executions of the program.

**search condition**

A criterion for selecting rows from a table. A search condition consists of one or more predicates.

**secondary authorization ID**

In DB2 for z/OS, an authorization identifier that is associated with a primary authorization ID by an authorization exit routine. See also *primary authorization ID*.

**segmented table space**

A table space that is divided into equal-sized groups of pages called segments. Segments are assigned to tables so that rows of different tables are never stored in the same segment. See also *table space*.

**server**

See *application server*.

**session**

All interactions between the user and QMF from the time the user invokes QMF until the EXIT command is issued.

**shift-in character**

A control character (X'0F') that is used in EBCDIC systems to denote that the subsequent bytes represent SBCS characters. See also *shift-out character*.

**shift-out character**

A control character (X'0E') that is used in EBCDIC systems to denote that the subsequent bytes, up to the next shift-in control character, represent DBCS characters. See also *shift-in character*.

**single-byte character set**

See *SBCS*.

**single-precision floating-point number**

A 32-bit approximate representation of a real number.

**SQL (Structured Query Language)**

A standardized language for defining and manipulating data in a relational database.

**SQL authorization ID**

See *SQLID*.

**SQL connection**

An association between an application process and a local or remote application server or database server. See also *remote unit of work*, *distributed unit of work*.

**SQL function**

A function that is implemented entirely by using a subset of SQL statements and SQL PL statements.

**SQL ID (SQL authorization ID)**

In DB2 for z/OS, the ID that is used for checking the authorization of dynamic SQL statements in some situations.

**SQL return code**

The SQLSTATE or SQLCODE that indicates whether the previously run SQL statement completed successfully, with one or more warnings, or with an error.

**SQLCA (Structured Query Language Communication Area)**

A set of variables that provides an application program with information about the execution of its SQL statements or requests from the database manager. When an error is associated with an SQL code, the QMF message help (available by pressing the Help key) displays the contents of the SQLCA.

**stored procedure**

A routine that can be invoked using the SQL CALL statement to perform operations that can include both host language statements and SQL statements.

**stored procedure interface**

An interface to QMF for TSO that allows you to start QMF as a Db2 for z/OS stored procedure, pass the name of a QMF query or procedure that performs the work you require, and receive up to 21 result sets back, including a result set for trace output. QMF for TSO can be started in this manner from any product that can run a Db2 for z/OS stored procedure.

**Structured Query Language**

See *SQL*.



## **Structured Query Language Communication Area**

See *SQLCA*.

### **subquery**

A complete SQL query that appears in a WHERE or HAVING clause of another query.

### **substitution variable**

(1) A variable in a procedure or query whose value is specified either by a global variable or by a runtime variable. (2) A variable in a QMF form whose value is specified by a global variable.

### **substring**

A part of a character string.

### **subsystem**

In DB2 for z/OS, a distinct instance of a relational database management system (RDBMS).

### **table**

In a relational database, a database object that consists of a specific number of columns and is used to store an unordered set of rows. See also *base table*.

### **table space**

A logical unit of storage in a database. In DB2 for z/OS, a table space is a page set and can contain one or more tables. In Db2 for Linux, UNIX, and Windows, a table space is a collection of containers, and the data, index, long field, and LOB portions of a table can be stored in the same table space or in separate table spaces.

### **temporary storage**

An area used to store a QMF object temporarily while the user is working on it so that, with each use, it can be readily accessed without further database retrieval. There are seven temporary storage areas: QUERY, DATA, FORM, PROC, REPORT, CHART, or PROFILE. With the exception of query result data (the DATA object), the QMF objects in these areas can be displayed using the SHOW command followed by the name of the storage area. Though the contents of the DATA area cannot be directly displayed, users can issue the SHOW REPORT or SHOW CHART commands to see the query result data formatted with the specifications of the form currently in the FORM area. See also *QMF object*, *current object*.

### **temporary storage queue**

In CICS, a queue of data items which can be read and reread, in any sequence. The queue is created by a task, and persists until the same task or a another task deletes it. See also *transient data queue*.

### **thread**

The DB2 structure that describes an application's connection, traces its progress, processes resource functions, and delimits its accessibility to DB2 resources and services. Most DB2 functions execute under a thread structure.

### **three-part name**

The full name of a table, view, or alias that consists of a location name, an authorization identifier, and an object name, separated by periods. QMF commands that include three-part names can be initiated only from Db2 for z/OS databases and can be directed to all databases except DB2 for VM or VSE. When QMF for TSO has been started as a Db2 for z/OS stored procedure, QMF commands with three-part names are not supported.

### **Time Sharing Option**

See *TSO*.

### **trace**

A record of the processing of a computer program or transaction. The information collected from a trace can be used to assess problems and performance.

### **transaction**

A unit of processing consisting of one or more application programs, affecting one or more objects, that is initiated by a single request.

### **transient data queue**

A CICS storage area where objects are stored for subsequent internal or external processing. See also *temporary storage queue*.

**trigger**

A database object that is associated with a single base table or view and that defines a rule. The rule consists of a set of SQL statements that runs when an insert, update, or delete database operation occurs on the associated base table or view.

**TSO (Time Sharing Option)**

A base element of the z/OS operating system that allows users to work interactively with the system.

**two-phase commit**

A two-step process by which recoverable resources in an external subsystem are committed. During the first step, the database manager subsystems are polled to ensure that they are ready to commit. If all subsystems respond positively, the database manager instructs them to commit.

**UDF (user-defined function)**

A function that is defined to the DB2 database system by using the CREATE FUNCTION statement and that can be referenced thereafter in SQL statements. A UDF can be an external function or an SQL function.

**Unicode**

A character encoding standard that supports the interchange, processing, and display of text that is written in the common languages around the world, plus some classical and historical texts. The Unicode standard has a 16-bit character set defined by ISO 10646.

**unit of work**

A recoverable sequence of operations within an application process. At any time, an application process is a single UOW, but the life of an application process can involve many UOWs as a result of commit or rollback operations. In a multisite update operation, a single UOW can include several units of recovery. In QMF SQL queries that include multiple statements, all statements comprise a single unit of work, so all statements except those that affect the session (such as SET) are rolled back in the event of a failure.

**user-defined function**

See *UDF*.

**view**

A logical table that is based on data stored in an underlying set of tables. The data returned by a view is determined by a SELECT statement that is run on the underlying tables.

**XML (Extensible Markup Language)**

A standard metalanguage for defining markup languages that is based on Standard Generalized Markup Language (SGML).

**z/OS**

An IBM mainframe operating system that uses 64-bit real storage.

# Index

## A

add  
break heading/footering [124](#)  
break segments and text [123](#)  
columns [279](#), [338](#)  
conditions  
    in a CONDITIONS box [339](#)  
    to an example table [339](#)  
    using AND [352](#)  
data to long fields [270](#)  
date, time, page number to report [122](#)  
dates and times [87](#)  
dates/times [87](#)  
expressions [339](#)  
information to queries [54](#)  
lines to a query [70](#)  
new column to report [104](#)  
page heading/footering to report [116](#)  
rows  
    I (insert) [352](#)  
    subtotals to report [114](#)  
    target tables [341](#)  
addition of dates and times [87](#)  
alias  
    deleting [266](#)  
    for table [266](#)  
    for view [266](#)  
    in place of three-part name [293](#)  
ALL keyword (QBE) [352](#)  
ANALYTIC database object [3](#)  
ANALYTIC objects  
    displaying [225](#)  
    printing [225](#)  
analyzing  
    query results [186](#)  
Analyzing data  
    creating specifications [151](#)  
    data characteristics [150](#)  
    number of [150](#)  
    observations [150](#)  
    parameter selection [151](#)  
    QMF Analytics for TSO [151](#)  
    scale of measurement [150](#)  
    variables [150](#)  
AND keyword (QBE) [352](#)  
AO keyword [352](#)  
application period  
    in Prompted Query [47](#)  
application requester, *See* connectivity  
application server, *See* connectivity  
arithmetic  
    expressions [352](#)  
    overflow [336](#)  
ascending order, sorting sequence [41](#), [352](#)  
ASIS formatting type [319](#)  
authorization

authorization (*continued*)  
    granting on table [279](#)  
    revoking on table [280](#)  
    to use objects [3](#)  
automatic interrupt [54](#)  
AVG column function [77](#), [352](#)

## B

BAR chart format [227](#)  
batch mode  
    procedures  
        errors [257](#)  
        example [258](#)  
        restrictions [256](#)  
        termination [257](#)  
        using IMPORT/EXPORT commands [257](#)  
        using the QMF EXIT command [257](#)  
        writing [256](#)  
BETWEEN keyword  
    QBE [352](#)  
    restrictions [352](#)  
BIGINT data type  
    chart axes exceed numeric bounds [236](#)  
    length [84](#)  
    scalar functions [79](#)  
bilingual command [307](#)  
BINARY data type  
    edit code [109](#)  
    restrictions  
        AND connector [352](#)  
        BETWEEN keyword in QBE [352](#)  
        charts [227](#)  
        D. keyword in QBE [352](#)  
        equality, greater and less than [352](#)  
        I. keyword in QBE [352](#)  
        IN keyword in QBE [352](#)  
        OR connector [352](#)  
        P. keyword in QBE [335](#), [342](#)  
        substitution variables [344](#)  
        Table Editor [28](#)  
        U. keyword in QBE [352](#)  
    scalar functions [79](#)  
BLOB data type  
    edit code [109](#), [113](#)  
break segments, adding to reports [123](#)  
break text, adding to reports [124](#)

## C

calculated values  
    average [352](#)  
    columns of expressions [338](#), [352](#)  
    count [352](#)  
    displaying on reports [132](#)  
    for groups [352](#)  
    maximum [352](#)

- calculated values (*continued*)
  - minimum [352](#)
  - of expressions [352](#)
  - total [352](#)
- CALL statements
  - restrictions in multistatement queries [274](#)
- CANCEL command [54](#)
- canceling a command or query [54](#)
- case operand for Prompted Query [33](#)
- casting of graphic to character data [308](#)
- changing
  - appearance of a report [20](#)
  - column names in queries [54](#)
  - date/time values using scalar functions [79](#)
  - default chart format [236](#)
  - information in queries [54](#)
  - row conditions in queries [54](#)
  - rows
    - U operator [352](#)
  - saved query [54](#)
  - sort order in queries [54](#)
  - table names in queries [54](#)
- CHAR scalar function [79](#)
- character
  - constants [338](#)
  - data
    - in conditions [336](#)
    - in descriptive columns [338](#)
    - with LIKE [352](#)
- CHART database object [3](#)
- charting
  - QMF Analytics for TSO [153](#)
  - query results [153](#)
- Charting
  - chart types [154](#)
  - QMF Analytics for TSO [154](#)
  - query results [153](#)
  - selecting a chart type [154](#)
  - using QMF Analytics for TSO [153](#)
- charts
  - changing
    - data, using QMF forms [231](#)
    - formats, using the ICU [235](#)
    - properties, using the ICU [235](#)
  - creating [227](#), [237](#)
  - creating histograms [155](#)
  - creating maps [161](#)
  - creating mixed charts [169](#)
  - creating pie charts [174](#)
  - creating plot charts [178](#)
  - creating tower charts [183](#)
  - creating user-defined maps [166](#)
  - fixing problems [236](#)
  - format
    - changing [235](#), [236](#)
    - creating [227](#)
    - saving [235](#)
  - location of data
    - on pie charts [229](#)
    - on the X-axis [228](#)
    - rules for specifying [227](#)
  - printing [237](#)
  - QMF Analytics for TSO
    - displaying [225](#)
- charts (*continued*)
  - QMF Analytics for TSO (*continued*)
    - running [225](#)
    - restrictions [227](#)
    - size limits for data [229](#)
    - types of [227](#), [230](#)
  - CHECK command [142](#)
  - CICS environment
    - restrictions
      - calculated values in reports [132](#)
      - document interface [318](#)
      - external editors [317](#)
      - QMF BATCH command [256](#)
      - REXX support and connectivity [291](#)
      - viewing data in tables [279](#)
      - unsupported functions [394](#)
  - CLIST used with editor [317](#)
  - CLOB data type
    - edit code [109](#), [113](#)
  - codes
    - edit [109](#)
    - usage [112](#)
  - column
    - number supported in queries [393](#)
  - column names
    - changing
      - in queries [54](#)
    - distinguishing between
      - using correlation names [70](#)
      - using qualifiers [70](#)
    - finding [63](#)
  - columns
    - adding to reports [104](#)
    - changing
      - alignment of headings and data [108](#)
      - column spacing [107](#)
      - headings [106](#)
      - on reports [103](#)
      - order (reports) [105](#)
      - spacing [107](#)
      - widths [107](#)
    - creating [64](#)
    - edit codes in [109](#)
    - expressions [352](#)
    - fixed, on reports [117](#)
    - functions
      - AVG [77](#), [352](#)
      - COUNT [77](#), [352](#)
      - MAX [77](#), [352](#)
      - MIN [77](#), [352](#)
      - nesting within scalar functions [86](#)
      - SUM [77](#), [352](#)
    - headings
      - labels vs. names [20](#)
    - joining
      - in Prompted Query [43](#)
      - multiple [50](#), [70](#)
    - punctuating [109](#)
    - select
      - using P. [352](#)
    - selecting [63](#), [334](#), [352](#)
    - specifying, on reports [117](#)
    - with calculated values [352](#)
  - command

command (*continued*)

- CANCEL [54](#)
- CHECK [142](#)
- CONNECT
  - compared to the DSQSDBNM parameter [290](#)
  - passwords in RACF [7](#)
- CONVERT [346](#)
- DBCS data [310](#)
- DELETE [347](#)
- DISPLAY
  - ANALYTICS [225](#)
  - using QBE [344](#)
- DRAW
  - using QBE [341](#), [348](#)
- EDIT [318](#)
- ENLARGE [348](#)
- FORM.COLUMNNS [103](#)
- governor interrupt [54](#)
- how to issue [5](#)
- interrupt [54](#)
- LAYOUT [119](#)
- LIST
  - under QBE [331](#)
- QBE-specific [346](#)
- REDUCE [350](#)
- RUN
  - substitution variables [344](#)
- SHOW FORM [103](#)

command line [4](#)

comment
 

- in QBE query [347](#)

COMMENTS box [347](#)

comments, sending to IBM [xi](#)

concatenation, rules for [94](#)

conditions
 

- character data in [336](#)
- data types in [339](#)
- DBCS characters [339](#)
- equalities [352](#)
- example elements [336](#)
- grouping [67](#)
- in a CONDITIONS box [339](#)
- in an example table [339](#)
- inequalities [352](#)
- multiple
  - AND [67](#), [352](#)
  - BETWEEN [352](#)
  - IN [67](#)
  - in Prompted Query [40](#)
  - OR [67](#), [352](#)
- negative [352](#)
- opposite [66](#)
- quotation marks in [336](#)
- row [65](#)
- selection symbols in [67](#)
- special characters in [339](#)
- values in a set [352](#)
- writing [335](#), [336](#)

CONDITIONS box (QBE)
 

- deleting [347](#)
- drawing [348](#)
- reducing [350](#)
- restricting data presentation [339](#)

CONNECT command (*continued*)
 

- comparison with DSQSDBNM parameter [290](#)
- passwords and RACF [7](#)
- prompt panels for [291](#)
- stored procedure interface restrictions [252](#)

connectivity with remote databases
 

- CONNECT command [289](#), [291](#)
- general considerations [290](#)
- lost connection [293](#)
- three-part names [294](#)

constants in reports [338](#)

conventions for highlighting [xi](#)

conversion of data types
 

- scalar functions [79](#)
- Unicode databases with DISPLAY TABLE [308](#)

CONVERT command [346](#)

converting queries to SQL [346](#)

copying a table [265](#)

correcting, saved query [54](#)

correlation names
 

- rules for [76](#)
- used in a subquery [74](#)
- used to distinguish between columns [70](#)

COUNT column function [77](#), [352](#)

counting number of values in a column [352](#)

CREATE PROCEDURE statements
 

- restrictions in multistatement queries [274](#)

creating
 

- expressions [36](#)
- new column in report [36](#), [64](#)
- reports [20](#)
- reusable procedures [250](#)
- row conditions [18](#), [37](#)

currency symbol, changing [109](#)

current location
 

- changing [291](#)
- DSQAO\_CONNECT\_LOC [290](#)
- procedures, forms, and queries [291](#)
- QMF objects [291](#)

CURRENT SQLID [291](#)

## D

D keyword in QBE [352](#)

data
 

- deletion [352](#)
- entry
  - inserting rows [352](#)
  - updating rows [352](#)
- retrieval by remote unit of work [291](#)

DATA database object [3](#)

data types
 

- BIGINT data type, *See* BIGINT data type
- BINARY data type, *See* BINARY data type
- conversion using scalar functions [78](#)
- DBCS support [309](#)
- DECFLOAT data type, *See* DECFLOAT data type
- dependencies with LIKE [352](#)
- GRAPHIC, *See* graphic data, GRAPHIC data type
- result of operation [336](#)
- VARBINARY data type, *See* VARBINARY data type
- VARGRAPHIC, *See* VARGRAPHIC data type
- XML data type, *See* XML data type

database

database (*continued*)

- cost estimate, suppressing [260](#)
- erasing queries from [56](#)
- functions that vary from one to another [393](#)
- objects, types of [3](#)
- retrieving saved query from [54](#)

database labels for column headings in forms [20](#)

DATE data type

- length [84](#)
- quoting in query conditions [336](#)
- scalar function [79](#)
- sorting sequence [352](#)

date/time

- adding
  - to page headings/footings [122](#)
- arithmetic, date/time [87](#), [93](#)
- data [87](#)
- edit codes [79](#)
- formats [79](#)
- scalar functions [79](#), [83](#)

DAY scalar function [81](#)

DAYS scalar function [88](#)

DBCLOB data type

- edit code [109](#), [113](#)

DBCS (double-byte character set) support

- changing lengths of names and fields [308](#)
- conversion of graphic to character data in Unicode [308](#)
- delimiters [308](#), [310](#)
- description [308](#), [315](#)
- display [308](#)
- exporting [314](#)
- form usages [310](#)
- importing [314](#)
- input fields that allow [310](#)
- QBE example elements [336](#)
- query usages [310](#)
- sorting [352](#)
- truncation [310](#)
- See also* GRAPHIC data type and VARGRAPHIC data type

DCF (Document Composition Facility)

- formatting type [319](#)
- how to insert a QMF report [319](#)

DECFLOAT data type

- chart axes exceed numeric bounds [236](#)
- edit codes [109](#)
- length [84](#)
- requirements for support
  - charting [227](#)
  - Table Editor [28](#)
  - user edit codes [310](#)
- scalar function [79](#)

DECIMAL data type

- length [84](#)
- scalar function [79](#)

decimal floating-point data type, *See* DECFLOAT data type

decrementing dates, times, or timestamps [87](#)

DELETE command [347](#)

deleting

- aliases [266](#)
- COMMENTS box [347](#)
- CONDITIONS box [347](#)
- example table [347](#)
- information from queries [54](#)

deleting (*continued*)

- lines from a query [70](#)
- query [56](#), [347](#)
- rows from a table
  - using QBE [352](#)
- synonyms [266](#)
- tables [266](#)
- views [266](#)

delimiters

- DBCS data types [308](#)
- multistatement queries [61](#)
- quoting in query conditions [336](#)
- XML data [352](#)

descending order, sorting sequence [41](#), [352](#)

detail blocks

- using panel variations to change [127](#)
- using to refine report format [127](#)

DIGITS scalar function [79](#)

DISPLAY command

- using QBE [344](#)

DISPLAY TABLE command

- conversion of graphic to character data in Unicode [308](#)

displaying

- calculated values on reports [132](#)
- database object list [23](#)
- duplicate rows [352](#)
- form panels [101](#)
- QBE Query panel [331](#), [342](#)
- report layout without data [119](#)
- reports [52](#)
- special conditions on reports [134](#)

distributed unit of work

- databases that do not support [393](#)
- VM/VSE restrictions [294](#)
- See also* three-part names

division sign (/) [336](#)

DO keyword [352](#)

Document Composition Facility [319](#)

document interface

- restrictions [321](#)
- using [321](#)

DRAW command

- using QBE [341](#), [348](#)

DSN option, GETQMF [321](#)

DSQAO\_CONNECT\_LOC global variable [290](#)

DSQDC\_COL\_LABELS global variable [20](#)

DSQDC\_COST\_EST global variable [260](#)

DSQEC\_LIST\_OWNER global variable [23](#)

DSQEC\_RUN\_MQ global variable [61](#)

DSQEC\_SQLQRYSZ\_2M global variable [61](#)

DSQSDBNM program parameter [290](#)

duplicate rows in reports, eliminating [45](#), [342](#), [352](#)

durations

- date/time arithmetic [89](#)
- incrementing and decrementing
  - dates [89](#)
  - times [92](#)
  - timestamps [93](#)
- making easier to read [92](#)

**E**

echo area, definition [15](#)

edit codes

- edit codes (*continued*)
  - changing [109](#)
  - character data [109](#)
  - currency symbol [109](#)
  - date/time [79](#)
  - definition [109](#)
  - metadata [109](#)
  - numeric data [109](#)
  - percent data [109](#)
  - specifying punctuation for values, in report columns [109](#)
  - suppressing zero values [109](#)
  - user-defined, restrictions [310](#)
- EDIT command [318](#)
- editing
  - in ISPF-PDF [321](#)
  - in PS/TSO [321](#)
  - QMF objects [317](#)
  - using CLIST [317](#)
  - using CLIST as editor name [317](#)
  - using editors with QMF [317](#)
  - using ISPF-PDF [317](#)
  - within QMF
    - ISPF [317](#)
- eliminating duplicate rows in reports [45](#), [352](#)
- ending QMF session [5](#)
- ENLARGE command [348](#)
- equalities [352](#)
- error messages, getting help for [13](#)
- European format, date/time edit codes [79](#)
- evaluation of expressions, rules [336](#)
- example elements [336](#)
- example table
  - conditions in [339](#)
  - deleting [347](#)
  - description [332](#)
  - drawing [348](#)
  - enlarging [348](#)
  - example elements [341](#)
  - expressions in [339](#)
  - reducing [350](#)
  - restrictions on [338](#), [341](#)
  - with added columns [338](#)
  - with example elements [338](#)
  - with target table [341](#)
  - with unnamed columns [338](#)
- exercises for Query-by-Example [372](#)
- EXPORT command
  - CICS [284](#)
  - HTML reports [285](#)
  - language [307](#)
  - TSO [284](#)
- exporting
  - DBCS data [314](#)
  - HTML reports [285](#)
  - objects
    - into TSO [284](#)
  - report text [285](#)
  - to UNIX [284](#)
- expressions
  - arithmetic [352](#)
  - definition [36](#)
  - evaluating [336](#)
  - in a CONDITIONS box [339](#)
  - in conditions [339](#)

- expressions (*continued*)
  - in example tables [339](#)
  - numeric data in [336](#)
  - order of evaluating [336](#)
  - results with nulls [352](#)
  - summary functions [36](#)
  - use of parentheses [336](#)
- Extensible Markup Language data type, *See* XML data type

## F

- feedback, sending to IBM [xi](#)
- final text, specifying on reports [130](#)
- FLOAT data type
  - length [84](#)
  - scalar function [79](#)
- floating-point numbers
  - importing single-precision [393](#)
- FOLDER database object [3](#)
- footings
  - adding
    - date, time, page number [122](#)
    - to reports [116](#)
  - changing alignment of [123](#)
  - refining, on reports [121](#)
  - using global variables in [121](#)
- FORM database object [3](#)
- form panels
  - changing column names [338](#)
  - displaying [103](#)
- FORM.BREAKn panel [115](#)
- FORM.CALC panel [132](#)
- FORM.COLUMNS panel [103](#)
- FORM.CONDITIONS panel [136](#)
- FORM.DETAIL panel [127](#)
- FORM.FINAL panel [130](#)
- FORM.MAIN panel [103](#)
- FORM.OPTIONS panel [117](#)
- FORM.PAGE panel [116](#)
- formats for time and date functions [79](#)
- formatting reports
  - for document type
    - ASIS [319](#)
    - DCF [319](#)
  - using QMF forms [101](#), [143](#)
- forms
  - correcting errors on [142](#)
  - displaying [101](#)
  - resetting to default values [143](#)
  - saving [142](#)
  - stored procedure result sets [329](#)
  - using DBCS data in [310](#)
  - using to create reports [101](#), [143](#)
- fractional seconds of a timestamp, scalar function [83](#)
- function keys
  - defined [4](#)
  - enlarging [348](#)
  - on home panel [4](#)
  - QBE initial settings [332](#)
  - reducing [350](#)

## G

- G keyword [352](#)
- G literal [310](#)
- GETQMF editor macro [318](#)
- global variables
  - add [259](#), [260](#)
  - change [259](#)
  - delete [259](#)
  - display [259](#)
  - DSQDC\_COL\_LABELS [20](#)
  - DSQEC\_LIST\_OWNER [23](#)
  - DSQEC\_RUN\_MQ [61](#)
  - DSQEC\_SQLQRYSZ\_2M [61](#)
  - list
    - and the CASE option [260](#)
    - to add a variable [260](#)
  - location [290](#)
  - panel [259](#)
  - set [259](#)
- governor interrupt routine [54](#)
- GRANT statement [294](#)
- graphic data
  - conversion to character with DISPLAY TABLE, Unicode [308](#)
  - GRAPHIC data type [309](#)
  - Prompted Query [33](#)
  - with LIKE [352](#)
- grouping data [352](#)

## H

- headings for columns, default [20](#)
- HEX scalar function [79](#)
- highlighting conventions [xi](#)
- HISTOGRAM chart format [227](#)
- home panel
  - starting a session [4](#)
- HOUR scalar function [83](#)

## I

- I keyword [352](#)
- I operator [352](#)
- ICU (Interactive Chart Utility)
  - used with editor [317](#)
- IMPORT command
  - single-precision floating point support [393](#)
- importing
  - DBCS data [314](#)
  - objects
    - from CICS [288](#)
    - from TSO [287](#)
- IN keyword
  - using in QBE [352](#)
  - with NOT [352](#)
- IN with NOT [352](#)
- incrementing dates, times, or timestamps [87](#)
- inequalities
  - in conditions [352](#)
- information, adding or changing in queries [54](#)
- input fields [310](#)
- inserting

- inserting (*continued*)
  - a row into a table [352](#)
  - GETQMF macro [320](#)
  - ISPF [318](#)
  - PS/TSO [318](#)
  - QMF report from an editor [318](#)
  - reports [320](#), [321](#)
  - rows into tables [352](#)
  - USEQMF option [320](#)
- INTEGER data type
  - length [84](#)
  - scalar function [79](#)
- interrupt
  - a command or query [54](#)
  - by the governor [54](#)
- iSeries platform
  - minimum supported database level [1](#)
  - query size maximum [61](#)
- ISO format, date/time edit codes [79](#)
- ISPF commands
  - COLORS [300](#)
  - COPY [300](#)
  - CREATE [300](#)
  - DESCRIBE [300](#)
  - FORMAT [300](#)
  - MOVE [300](#)
  - PREVIEW [300](#)
  - QMFHELP [300](#)
  - REPLACE [300](#)
  - RUN [300](#)
  - SAVE [300](#)
  - SAVEAS [300](#)
  - SIZE [300](#)
  - SYNTAX [300](#)
- ISPF-PDF (Interactive System Productivity Facility-Program Development Facility)
  - accessing from QMF [318](#)
  - primary option menu panel [318](#)
  - supported environments [321](#)
  - used while in QMF [321](#)

## J

- JIS format, date/time edit codes [79](#)
- joining
  - columns
    - in Prompted Query [43](#)
    - in QBE [342](#)
    - in SQL [70](#)
  - multiple tables [47](#), [70](#), [342](#)
  - strings [94](#)
  - tables in Prompted Query [43](#)

## K

- keys, function [4](#)
- keywords, QBE
  - ALL [352](#)
  - AND [352](#)
  - AO [352](#)
  - AVG [352](#)
  - BETWEEN [352](#)
  - COUNT [352](#)



keywords, QBE (*continued*)

- [D](#) [352](#)
- [DO](#) [352](#)
- [G](#) [352](#)
- [I](#) [352](#)
- [IN](#) [352](#)
- [LIKE](#) [352](#)
- [MAX](#) [352](#)
- [MIN](#) [352](#)
- [NOT](#) [352](#)
- [NULL](#) [352](#)
- [OR](#) [352](#)
- [P](#) [352](#)
- [SUM](#) [352](#)
- [U](#) [352](#)
- [UNQ](#) [352](#)

## L

- labels for column headings in forms [20](#)
- language operand for Prompted Query [33](#)
- LANGUAGE parameter on SET PROFILE command [331](#)
- LAYOUT command [119](#)
- LENGTH string function [84](#)
- LIKE keyword
  - any number of characters (%) [352](#)
  - any single character ( ) [352](#)
  - data type dependencies [352](#)
  - graphic data [352](#)
  - how support varies by database [393](#)
  - selecting on conditions [352](#)
  - with NOT [352](#)
- limits for size of chart data [229](#)
- LINE chart format [227](#)
- linear procedures
  - example [239](#)
  - guidelines [240](#)
- links
  - non-IBM Web sites [398](#)
- Linux, UNIX, and Windows platform
  - minimum supported database level [1](#)
  - SQL query size maximum [61](#)
- LIST command
  - OWNER parameter default [23](#)
  - under QBE [331](#)
- List function key
  - command prompt panel [23](#)
  - QMF CONNECT prompt panel [291](#)
- list of database objects
  - displaying [23](#)
  - tables [33](#)
- locating data on charts
  - in general [227](#)
  - on pie charts [229](#)
  - on the X-axis [228](#)
- location name
  - global variable for [290](#)
  - list panel [291](#)
  - QMF CONNECT prompt panel [291](#)
  - viewing [290](#)
- lost connection [293](#)

## M

- manipulating character/graphic strings [78](#)
- MAX column function
  - in QBE [352](#)
  - in SQL [77](#)
- message line [4](#)
- metadata edit code [109](#)
- MICROSECOND scalar function [83](#)
- MIN column function [77](#), [352](#)
- minimum unique abbreviations, for commands [5](#)
- minus sign (-)
  - in expressions [352](#)
  - order of evaluation [336](#)
- MINUTE scalar function [83](#)
- model queries [344](#)
- MONTH scalar function [81](#)
- More Help, for error messages [13](#)
- multiple
  - columns, joining [50](#)
  - conditions
    - under QBE [352](#)
  - table queries [70](#)
  - tables, joining [47](#)
- multiplication operator (\*) [336](#)
- multistatement queries
  - how to create [61](#)
  - QBE to SQL conversion [346](#)

## N

- N literal [310](#)
- name
  - column [1](#), [338](#)
  - for columns of calculated values [338](#)
  - for form panels, displaying list of [101](#)
  - for tables [1](#)
  - location qualifier [1](#)
  - owner qualifier [1](#)
  - tables [1](#)
- National Language Feature (NLF)
  - batch mode and [257](#)
  - bilingual forms [307](#)
- negative conditions [352](#)
- NOT keyword [352](#)
- notices
  - legal [397](#)
- null
  - definition of [352](#)
  - in condition with IN keyword [352](#)
  - passing in expressions [104](#)
  - replacing with data [104](#)
  - values
    - defined [352](#)
    - prints and displays as [352](#)
    - replacing using the VALUE function [85](#)
    - results in expressions [352](#)
    - using the VALUE function [85](#)
    - with conditions [339](#)
    - with G [352](#)
    - with I [352](#)
    - with NOT [352](#)
- numeric data

numeric (*continued*)  
  data (*continued*)  
    importing single-precision floating point [393](#)  
numeric data  
  chart axes exceed numeric bounds [236](#)  
  constants [338](#)  
  expressions [336](#)

## O

object owner [3](#)  
objects  
  CHART [3](#)  
  current location [291](#)  
  DATA [3](#)  
  database [3](#)  
  displaying  
    saved ANALYTIC objects [225](#)  
  displaying a list of [23](#)  
  exporting  
    HTML reports [285](#)  
    into CICS [284](#)  
    into TSO [284](#)  
  FOLDER [3](#)  
  FORM [3](#)  
  importing  
    from CICS [288](#)  
    from TSO [287](#)  
  listing [33](#)  
  PROC [3](#)  
  PROFILE [3](#)  
  QUERY [3](#)  
  REPORT [3](#)  
  retrieving from the database [11](#)  
  saving in the database [9](#)  
  sharing with other users [9](#)  
operators  
  ALL [352](#)  
  AO [352](#)  
  D [352](#)  
  DO [352](#)  
  G [352](#)  
  I [352](#)  
  P [352](#)  
  U [352](#)  
  UNQ [352](#)  
OR keyword [352](#)  
order  
  evaluating expressions [336](#)  
  of columns  
    ascending [352](#)  
    descending [352](#)  
    reversing [334](#)  
  rows in a report  
    ascending [352](#)  
    descending [352](#)  
OWNER parameter of LIST command, default [23](#)

## P

P (present) keyword [333](#), [335](#), [352](#)  
page heading and footing  
  adding

page heading and footing (*continued*)  
  adding (*continued*)  
    date, time, page number [122](#)  
    to reports [116](#)  
  adding to a report [20](#)  
  changing alignment of [123](#)  
  refining, on reports [121](#)  
  using global variables in [121](#)  
panel  
  CONVERT Command Prompt [346](#)  
  display form panels using the SHOW and DISPLAY  
  commands [101](#)  
  home [4](#)  
  Prompted Query dialog [31](#)  
  QBE QUERY [332](#)  
parentheses in expressions [336](#)  
PASS NULLS field [104](#)  
passwords on CONNECT command [7](#)  
percent sign (%) with LIKE [352](#)  
PIE chart format [227](#)  
plus sign (+)  
  in expressions [352](#)  
  order of evaluation [336](#)  
POLAR chart format [227](#)  
precision of TIMESTAMP values [83](#)  
presenting  
  all columns of a table [333](#), [352](#)  
  certain values in a set [352](#)  
  data [352](#)  
  data from more than one table [342](#), [352](#)  
  maximum number of specific columns [333](#)  
  on either of two conditions [352](#)  
  on part of a value [352](#)  
  on the opposite of a condition [352](#)  
  on two conditions [352](#)  
  rows with a certain value [335](#)  
  rows with missing entries [352](#)  
  some columns in a table [352](#)  
  some rows in a table [352](#)  
  specific columns of a table [334](#), [352](#)  
  specific rows of a table [335](#)  
  values within a range [352](#)  
previewing reports [96](#)  
printing  
  charts [237](#)  
  DBCS reports [315](#)  
  reports [143](#)  
PROC database object [3](#)  
procedures  
  batch  
    errors [257](#)  
    example [258](#)  
    termination [257](#)  
    using EXIT command [257](#)  
    using IMPORT/EXPORT commands [257](#)  
    writing [256](#)  
  bilingual command [307](#)  
  DBCS data [310](#)  
  for QMF  
    in CICS environment [239](#)  
    in TSO environment [239](#)  
  linear  
    connecting from, to a remote location [252](#)  
    creating [239](#)

- procedures (*continued*)
  - reusable, creating [250](#)
  - sharing with other QMF users [249](#)
  - to create queries
    - using generic SQL statements [254](#)
    - using global variables [255](#)
    - using REXX variable values [254](#)
  - to run [249](#)
  - with logic
    - connecting from, to a remote location [252](#)
    - creating [239](#)
    - example [241](#)
    - using REXX variables in [243](#)
- processing date and time values [78](#)
- processing order [336](#)
- profile
  - saving changes in the database [7](#)
  - SET PROFILE command [331](#)
  - setting up [7](#), [15](#), [331](#)
  - viewing [7](#)
- PROFILE database object [3](#)
- prompt panel for variable data [344](#)
- Prompted Query
  - and echo areas [31](#)
  - dialog panels [31](#)
  - displaying a report [52](#)
  - eliminating duplicate rows in reports [45](#)
  - finding comments about tables [33](#)
  - general rules [33](#)
  - joining multiple columns in [50](#)
  - joining tables in [43](#)
  - listing tables [33](#)
  - main panel
    - command line [31](#)
    - echo area [31](#)
    - function key area [31](#)
    - scroll indicator [31](#)
  - profile requirements [33](#)
  - running a query [52](#)
  - selecting tables [33](#)
  - SQL equivalent of [57](#)
  - starting [33](#)
  - substitution variables in [50](#)
- PS/TSO, used with QMF document interface [318](#)
- PULL statements to specify REXX variables [244](#)

## Q

- Q.APPLICANT sample table [383](#)
- Q.CASHFLOW sample table [389](#)
- Q.CLIMATE\_10YR sample table [390](#)
- Q.CLIMATE\_USA sample table [390](#)
- Q.INTERVIEW sample table [383](#)
- Q.ORG sample table [384](#)
- Q.PARTS sample table [385](#)
- Q.PRODUCTS sample table [385](#)
- Q.PROJECT sample table [386](#)
- Q.SALES sample table [386](#)
- Q.STAFF sample table [387](#)
- Q.SUPPLIER sample table [388](#)
- Q.WORLDINFO sample table [392](#)
- QBE (Query-by-Example)
  - calculated values in expressions [352](#)
  - commands [346](#)

- QBE (Query-by-Example) (*continued*)
  - conversion of multiple statements to SQL [346](#)
  - keywords [352](#)
  - Query panel [332](#)
- QMF Analytics for TSO
  - determining the purpose of your analysis [147](#)
  - planning [147](#), [150](#)
  - selecting an analysis [150](#)
  - task flow [147](#)
  - using [147](#)
- qualifiers, to distinguish between columns [70](#)
- query
  - adding
    - lines to [54](#), [70](#)
    - specifications to [54](#)
  - calculated values
    - columns of expressions [352](#)
    - for groups [352](#)
  - calculated values in [352](#)
  - changing [54](#)
  - conditions
    - selecting on [335](#), [339](#)
  - converting
    - to SQL [346](#)
  - cost estimate, suppressing [260](#)
  - creating using Prompted Query [15](#), [31](#), [57](#)
  - creating using SQL [59](#)
  - data entry
    - insert rows [352](#)
    - update rows [352](#)
  - deleting
    - DELETE command [347](#)
    - from database [56](#), [347](#)
    - information from [54](#)
    - lines from [70](#)
    - rows [352](#)
  - eliminating duplicate rows [352](#)
  - erasing from database [56](#)
  - example elements [336](#)
  - expressions
    - arithmetic [352](#)
    - definition [352](#)
    - in conditions [339](#)
  - format [59](#)
  - interfaces [3](#)
  - joining multiple columns in [50](#)
  - listing [331](#)
  - making reusable [50](#), [95](#)
  - model [344](#)
  - multiple table [70](#)
  - non-displaying, correcting [54](#)
  - ordering rows in a report [352](#)
  - Prompted Query [3](#)
  - Query-by-Example [3](#)
  - resetting [331](#)
  - retrieving from database [54](#)
  - reusing [344](#)
  - row conditions in [37](#), [65](#), [335](#), [339](#)
  - rows [339](#)
  - running
    - under QBE [331](#)
  - saving [331](#)
  - selecting
    - all columns [34](#), [63](#), [352](#)

- query (*continued*)
  - selecting (*continued*)
    - columns for a Prompted Query [17](#)
    - from multiple tables [43](#), [72](#), [352](#)
    - rows for a Prompted Query [18](#)
    - specific columns [34](#), [63](#), [334](#)
    - specific rows [37](#), [65](#), [335](#), [352](#)
    - table for a Prompted Query [15](#)
  - selecting on conditions
    - BETWEEN [352](#)
    - IN keyword [352](#)
    - multiple [352](#)
    - negative [352](#)
    - specific columns [352](#)
    - specific rows [335](#), [352](#)
    - values within a range [352](#)
    - with a certain string of characters [352](#)
    - with equality and inequality [352](#)
  - sharing with other users [352](#)
  - size restrictions [61](#)
  - sorting rows in [41](#), [69](#)
  - SQL [3](#)
  - starting [33](#), [60](#)
  - statement length [393](#)
  - substitution variables in [50](#), [95](#)
  - with DBCS data
    - graphic strings [310](#)
    - with substitution variable data [344](#)
- QUERY database object [3](#)
- Query results
  - analyzing [147](#), [148](#), [151](#)
  - data characteristics [148](#)
  - QMF Analytics for TSO [147](#)
- Query-by-Example [331](#), [372](#)
- quotation marks
  - when needed [336](#)
  - with constants [338](#)
  - with LIKE [352](#)

## R

- RACF and passwords [7](#)
- REDUCE command [350](#)
- reformatted text, mixing with tabular data [139](#)
- relative cost estimate, suppressing [260](#)
- remote data access, Db2 environment [1](#)
- remote location, connecting from a procedure [252](#)
- remote unit of work
  - accessing the current location name [290](#)
  - granting privileges to other locations [294](#)
  - Lost Connection prompt panel [293](#)
  - QMF objects [291](#)
  - reconnecting to a location [293](#)
  - states of QMF when a connection is lost [293](#)
  - stored procedure interface restrictions [252](#)
  - tables and views [293](#)
  - tips and techniques [290](#), [294](#)
  - tips for procedures [290](#)
  - using QMF with [291](#)
- replacing null values using the VALUE function [85](#)
- REPORT database object [3](#)
- reports
  - calculating values on [132](#)
  - changing

- reports (*continued*)
  - changing (*continued*)
    - alignment of headings and data [108](#)
    - column heading [106](#)
    - column names [20](#)
    - column order [105](#)
    - column width [20](#), [107](#)
    - columns in [103](#)
    - default format [102](#)
    - completing before connecting to remote location [290](#)
    - correcting errors before display [142](#)
    - creating [101](#), [143](#)
    - customizing
      - break segments [123](#)
      - break text [124](#)
      - new column to [104](#)
      - page headings and footings [20](#)
      - subtotals to [114](#)
    - default format [102](#)
    - default report format [20](#)
    - definition [20](#)
    - displaying
      - calculated values in [132](#)
      - FORMS panels for [103](#)
      - representative [119](#)
      - special conditions on [134](#)
    - duplicate rows, eliminating [342](#)
    - edit codes in [109](#)
    - eliminating duplicate rows from [45](#)
    - exporting without control information [285](#)
    - final text in [130](#)
    - footings in [116](#), [121](#), [123](#)
    - FORM.COLUMNS command [103](#)
    - FORM.MAIN command [103](#)
    - formatting with detail blocks [127](#)
    - headings in [116](#), [121](#), [123](#)
    - mixing tabular data with reformatted text [139](#)
    - page headings [121](#)
    - printing [143](#)
    - SELECT with change statements [96](#)
    - SHOW FORM command [103](#)
    - showing totals across rows [140](#)
    - specifying fixed columns [117](#)
    - specifying punctuation for values in columns [109](#)
    - specifying text for subtotals [115](#)
    - specifying usage codes [112](#)
  - RESET QUERY command [331](#)
  - resetting forms to default values [143](#)
  - retrieving
    - data
      - from multiple tables [70](#), [74](#)
      - with QBE [333](#)
    - query from database [54](#)
    - saved query [54](#)
  - reusable procedures, creating [250](#)
  - reusing queries [344](#)
  - reversing order of columns [334](#)
  - REXX support
    - calling programs from procedure with logic [247](#)
    - error handling, branching to subroutines [246](#)
    - procedures with logic, creating [241](#)–[243](#)
    - SAY and PULL statements [244](#)
    - substitution variables [245](#)
    - using messages with the EXIT instruction [246](#)

REXX support (*continued*)  
 writing programs [104](#)

rollbacks  
 failure of multistatement queries [61](#)

ROWID data type  
 length [84](#)

rows  
 adding [352](#)  
 conditions  
   changing in queries [54](#)  
   specifying [65](#)  
 deleting [352](#)  
 duplicates in reports, eliminating [45](#)  
 eliminating duplicates [69](#), [352](#)  
 from multiple tables [352](#)  
 inserting [352](#)  
 ordering [352](#)  
 select  
   specific [335](#)  
 select on conditions  
   AND [352](#)  
   BETWEEN [352](#)  
   OR [352](#)  
 selecting  
   both conditions true [67](#)  
   one of two conditions true [67](#)  
   specific [37](#), [65](#)  
   using character values [65](#)  
   using conditions [65](#)  
   using grouped conditions [67](#)  
   using multiple conditions [67](#)  
   using multiple OR conditions [67](#)  
   using multiple row conditions [40](#)  
   using opposite conditions [66](#)  
   using selection symbols [67](#)  
   with no data [65](#)  
 selecting on conditions [335](#)  
 selecting some [352](#)  
 sorting [69](#)  
 updating [352](#)  
 with nulls [352](#)

rules  
 for creating a subquery [74](#)  
 for date/time addition [87](#)  
 for date/time subtraction [87](#)  
 for location of data on charts [227](#)

RUN command  
 description [331](#)  
 multistatement queries [61](#)  
 substitution variables [344](#)  
 to run a query [18](#)

**S**

sample tables [2](#), [383](#), [389](#), [390](#), [392](#)

SAVE command  
 under QBE [331](#)

SAVE option  
 EDIT TABLE command  
 unsupported situations [393](#)

saved query, changing [54](#)

saving  
 chart format [235](#)  
 queries in the database [331](#)

saving (*continued*)  
 report forms [142](#)

SAY statements to specify REXX variables [244](#)

scalar functions  
 datetime data [79](#)  
 nesting [86](#)  
 uses of [78](#)

SCATTER chart format [227](#)

SCRIPT/VS  
 how to insert a QMF report [319](#)

SECOND scalar function [83](#)

select  
 all columns [63](#), [352](#)  
 columns for a Prompted Query [17](#)  
 maximum number of specific columns [333](#)  
 multiple tables [352](#)  
 on conditions  
   introduction [335](#)  
   multiple [352](#)  
   negative [339](#), [352](#)  
   values in a set [352](#)  
   values within a range [352](#)  
   with a certain string of characters [352](#)  
   with equality and inequality [352](#)  
 rows [18](#), [37](#), [65](#)  
 some columns [63](#), [352](#)  
 some rows [352](#)  
 specific columns [334](#)  
 specific rows [335](#)  
 table [15](#)  
 tables [33](#), [63](#)

SELECT statements  
 restrictions in multistatement queries [274](#)

SELECT with change statements  
 previewing reports [96](#)

selection symbols [27](#)

semicolon delimiters in multistatement queries [61](#)

service information [xi](#)

SET GLOBAL command [260](#)

SET PROFILE command  
 LANGUAGE parameter [331](#)

sharing queries [352](#)

shift in delimiter (SI) [308](#), [310](#)

shift out delimiter (SO) [308](#), [310](#)

shortened form panel names [101](#)

SHOW command  
 to display forms [101](#)

SHOW FIELD function key [260](#)

SHOW FORM command [103](#)

SHOW GLOBALS command [259](#), [260](#)

SI character [308](#)

single-precision floating point numbers  
 support [393](#)

size limits for chart data [229](#)

slash (/)  
 in expressions [352](#)

SMALLINT data type  
 length [84](#)

SO character [308](#)

solutions to exercises for QBE [372](#)

sorting sequence  
 AO (ascending order) [352](#)  
 changing in queries [54](#)  
 DO (descending order) [352](#)

- sorting sequence (*continued*)
  - specifying [54](#), [352](#)
- special characters in Prompted Query [33](#)
- special conditions
  - displaying on reports [134](#)
  - identifying using expressions [136](#)
  - identifying using REXX program [134](#)
- SQL
  - converting queries to [346](#)
  - equivalent of prompted query [57](#)
  - functions, advanced
    - date/time arithmetic [87](#)
    - joining strings [94](#)
    - multiple table queries [70](#)
  - statements
    - for adding columns [279](#)
    - for adding rows [275](#), [276](#)
    - for authorizing access to tables [279](#)
    - for changing rows [276](#), [278](#)
    - for copying rows from one table to another [279](#)
    - for deleting rows [278](#)
    - restrictions in multistatement queries [274](#)
    - substitution variables in [95](#)
- SQL queries
  - converting queries from QBE [346](#)
  - inserting results into tables [96](#)
  - multiple statements [61](#), [274](#)
  - size restrictions [61](#)
- starting QMF
  - home panel [4](#)
  - stored procedure interface
    - restrictions [25](#)
- statement, query
  - lengths supported [393](#)
- statistics
  - generating [186](#)
  - QMF Analytics for TSO
    - displaying [225](#)
    - running [225](#)
- stored procedure interface to QMF for TSO
  - restrictions [25](#)
  - running QMF procedures [249](#)
  - writing QMF procedures [256](#)
- stored procedures
  - examples [325](#)
  - formatting in result sets [329](#)
  - issuing CALL statements from SQL query panel [325](#)
- string functions
  - LENGTH [84](#)
  - SUBSTR [85](#)
  - VALUE [85](#)
- string of characters with LIKE [352](#)
- subquery
  - rules for creating [74](#)
  - to retrieve data from multiple tables [74](#)
  - to retrieve more than one value [74](#)
  - to satisfy a condition [74](#)
  - using a correlation name [74](#)
- substitution values [344](#)
- substitution variables
  - differences from REXX variables [245](#)
  - in procedures with logic [250](#)
  - in Prompted Query [50](#)
  - in QBE [344](#)
- substitution variables (*continued*)
  - in SQL [95](#)
  - restrictions [344](#)
  - specifying values for
    - as part of RUN command [52](#)
    - on RUN Command Prompt panel [52](#)
    - using global variables [52](#)
  - supplying values
    - using the RUN command [250](#)
    - using the RUN command prompt panel [251](#)
    - using the SET GLOBAL command [251](#)
- SUBSTR scalar function [85](#)
- SUBSTR string function [85](#)
- subtotals
  - adding to reports [114](#)
  - specifying text for [115](#)
- subtraction of dates and times [87](#), [93](#)
- SUM
  - column function [77](#)
- SUM column function
  - in QBE [352](#)
  - in SQL [77](#)
- summary conditions [352](#)
- summary functions [36](#)
- support information [xi](#)
- SURFACE chart format [227](#)
- synonym
  - deleting [266](#)
  - for table [266](#)
  - for view [266](#)
- system period
  - in Prompted Query [47](#)

## T

- TABLE chart format [227](#)
- Table Editor
  - ending a session [274](#)
  - for adding data to long fields [270](#)
  - for adding rows [269](#)
  - for changing rows [271](#), [273](#)
  - for deleting rows [273](#)
  - search [272](#)
  - using column defaults [269](#)
  - using nulls [269](#)
- TABLE option of RUN command [96](#)
- tables
  - adding columns using SQL statements [279](#)
  - adding rows
    - using SQL statements [275](#), [276](#)
    - using the QMF DRAW command [275](#)
    - using the Table Editor [269](#)
  - appending to existing [264](#)
  - authorizing access
    - to add new rows [279](#)
    - to change rows [279](#)
    - to delete rows [279](#)
    - to update columns [280](#)
    - to view [279](#)
  - changing rows
    - using SQL statements [276](#), [278](#)
    - using the QMF DRAW command [276](#)
    - using the Table Editor [271](#), [273](#)
  - columns, relationship to [1](#)

tables (*continued*)

- copying [265](#), [352](#)
- copying rows from one to another
  - using SQL statements [279](#)
- creating
  - a view from [265](#)
  - alias for [266](#)
  - synonym for [266](#)
  - using SQL statements [263](#)
- deleting [266](#)
- deleting rows
  - using QBE [352](#)
  - using SQL statements [278](#)
  - using the Table Editor [273](#)
- inserting rows
  - using QBE [352](#)
  - using SQL statements [275](#)
- joining [43](#), [70](#)
- joining multiple [47](#), [70](#)
- listing [33](#)
- location qualifier [1](#)
- multiple
  - joining columns from [70](#)
  - merging data from [72](#)
  - retrieving data from [74](#)
  - used to create a query [70](#)
- names
  - changing in queries [54](#)
  - specifying [1](#)
- naming conventions [1](#)
- owner qualifier [1](#)
- planning to create [263](#)
- presenting data from more than one [342](#)
- revoking access [280](#)
- rows, relationship to [1](#)
- sample
  - Q.APPLICANT [383](#)
  - Q.CASHFLOW [389](#)
  - Q.CLIMATE\_10YR [390](#)
  - Q.CLIMATE\_USA [390](#)
  - Q.INTERVIEW [383](#)
  - Q.ORG [384](#)
  - Q.PARTS [385](#)
  - Q.PRODUCTS [385](#)
  - Q.PROJECT [386](#)
  - Q.SALES [386](#)
  - Q.STAFF [387](#)
  - Q.SUPPLIER [388](#)
  - Q.WORLDINFO [392](#)
- saving [264](#)
- selecting [33](#), [63](#)
- target [341](#), [348](#)
- updating rows [352](#)
- using column defaults with Table Editor [269](#)
- with null data [352](#)

tabular data, mixing with reformatted text [139](#)

target table, drawing [341](#), [348](#)

temporal data
 

- in Prompted Query [47](#)

temporary storage
 

- QMF objects in [9](#)
- saving to [9](#)

three-part names
 

- database support [393](#)

three-part names (*continued*)
 

- GRANT statements [294](#)
- QMF objects [291](#)
- restrictions [25](#), [294](#)
- use of an alias [293](#)

TIME data type
 

- length [84](#)
- quoting in query conditions [336](#)
- scalar function [79](#)
- sorting sequence [352](#)

time period
 

- specifying
  - in Prompted Query [47](#)

times sign (\*) in expressions [352](#)

TIMESTAMP data type
 

- fractional seconds function [83](#)
- length [84](#)
- quoting in query conditions [336](#)
- scalar function [79](#)
- sorting sequence [352](#)

TIMESTAMP WITH TIME ZONE data type
 

- length [84](#)
- quoting in query conditions [336](#)
- sorting sequence [352](#)

timestamps, adding or subtracting duration [93](#)

totals, showing across rows in reports [140](#)

TOWER chart format [227](#)

## U

U edit code restrictions [310](#)

U keyword and operator [352](#)

underscore ()
 

- example element [336](#)
- with LIKE [352](#)

Unicode considerations for graphic data [308](#)

UNIX, exporting to [284](#)

UNQ keyword (QBE) [352](#)

updating rows [271](#), [352](#)

USA format, date/time edit codes [79](#)

usage codes
 

- definition [112](#)
- specifying [112](#)

USEQMF option, GETQMF [320](#)

user ID to use QMF [4](#)

USER variable [344](#), [352](#)

## V

V edit code restrictions [310](#)

VALUE string function [85](#)

values
 

- calculated
  - on reports [338](#), [352](#)
- expressions [352](#)
- on reports
  - sources [132](#)
  - specifying calculations for [132](#)

VARBINARY data type
 

- edit code [109](#)
- restrictions
  - AND connector [352](#)
  - BETWEEN keyword in QBE [352](#)

VARBINARY data type *(continued)*  
   restrictions *(continued)*  
     charts [227](#)  
     D. keyword in QBE [352](#)  
     equality, greater and less than [352](#)  
     I. keyword in QBE [352](#)  
     IN keyword in QBE [352](#)  
     OR connector [352](#)  
     P. keyword in QBE [342](#)  
     substitution variables [344](#)  
     Table Editor [28](#)  
     U. keyword in QBE [352](#)  
     scalar functions [79](#)  
 VARGRAPHIC data type  
   scalar function [79](#)  
   support [309](#)  
 variables  
   data [344](#)  
   resolution in multistatement queries [61](#)  
   substituting user ID for [352](#)  
   substitution [344](#)  
   USER [344](#), [352](#)  
   value [260](#)  
 view  
   creating alias for [266](#)  
   creating from a table [265](#)  
   creating synonym for [266](#)  
   deleting [266](#)  
   location name [290](#)  
 VM platform  
   minimum supported database level [1](#)  
   SQL query size maximum [61](#)  
   three-part name restrictions [294](#)  
 VSE platform  
   minimum supported database level [1](#)  
   SQL query size maximum [61](#)  
   three-part name restrictions [294](#)

## W

wildcards [27](#)

## X

XML data type  
   edit code [33](#), [109](#)  
   quote delimiters [352](#)  
   restrictions  
     AND connector [352](#)  
     AO. keyword in QBE [352](#)  
     BETWEEN keyword in QBE [352](#)  
     charts [227](#)  
     CICS [333](#)  
     COUNT. keyword [352](#)  
     D. keyword in QBE [352](#)  
     DO. keyword in QBE [352](#)  
     equality, greater and less than [352](#)  
     G. keyword in QBE [352](#)  
     IN keyword in QBE [352](#)  
     MAX function [352](#)  
     MIN function [352](#)  
     OR connector [352](#)  
     P. keyword in QBE [335](#), [342](#)

XML data type *(continued)*  
   restrictions *(continued)*  
     substitution variables [344](#)  
     Table Editor [28](#), [267](#)  
     UNQ. keyword in QBE [352](#)  
     user edit codes [310](#)  
     wrapping [107](#), [113](#)

## Y

YEAR scalar function [81](#)

## Z

z/OS platform  
   minimum supported database level [1](#)  
   SQL query size maximum [61](#)  
   starting QMF from ISPF-PDF [322](#)  
 zero values, suppressing [109](#)







Product Number: 5698-DB2  
5698-DB2  
5698-DB2

SC28-2794-00

