

# HEWLETT-PACKARD

## Query Operators Guide

**HP**

**260**

**HP Computer Museum**  
**[www.hpmuseum.net](http://www.hpmuseum.net)**

**For research and education purposes only.**

**HP 260 Computer Systems**

# **QUERY OPERATORS GUIDE**

**User's Manual**



HERRENBERGER STRASSE 130, D-7030 BOEBLINGEN

Part No. 45261-90005  
E0986

Printed in Federal Republic of Germany 09/86

**FEDERAL COMMUNICATION COMMISSION RADIO  
FREQUENCY INTERFERENCE STATEMENT**  
(for U.S.A. only)

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

**NOTICE**

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

# PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual or update was issued. Many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

First Edition . . . . .	Feb 1985. . . . .	B. 07.00
Second Edition . . . . .	Sep 1986. . . . .	B. 08.00





# LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Pages	Date
all .....	Sep 1986





## Table of Contents

### Section 1

#### INTRODUCTION TO QUERY

Introduction to Data Base Management . . . . .	1-2
Data Item . . . . .	1-2
Data Entry . . . . .	1-3
Data Set . . . . .	1-3
Types of Data Sets . . . . .	1-5
Data Base . . . . .	1-6
Data Base Schema . . . . .	1-7

### Section 2

#### GETTING STARTED

Running QUERY . . . . .	2-1
Specifying a Data Base . . . . .	2-2
Examples . . . . .	2-2
Specifying a Password . . . . .	2-3
Communicating With QUERY . . . . .	2-4
Using Softkeys . . . . .	2-4
Examples . . . . .	2-6
Changing Data Bases . . . . .	2-10
Changing Passwords . . . . .	2-11
Data Base Information . . . . .	2-12
Prewritten Procedures . . . . .	2-15
Exit . . . . .	2-16

### Section 3

#### CREATING REPORTS

Finding Entries . . . . .	3-1
Specifying a Workfile . . . . .	3-1
Finding Entries in One Data Set . . . . .	3-2
Item List . . . . .	3-2
Search Expression . . . . .	3-3
Using the FIND Softkey . . . . .	3-4
Finding Entries in Multiple Data Sets . . . . .	3-5
Threading Data Sets . . . . .	3-5
The FIND Command with Threaded Data Sets . . . . .	3-6
Listing the Workfile . . . . .	3-7
Specifying the Output Device . . . . .	3-7
Listing Data Items in Columnar Format . . . . .	3-8
Listing Data Items in a Linear Format . . . . .	3-9
Formatting the Listing . . . . .	3-10
Sorting Data Items . . . . .	3-11
Computing with Numeric Data Items . . . . .	3-12
Breaking a Listing into Groups . . . . .	3-13
Creating Subtotals within a List . . . . .	3-15
Running a Report Program . . . . .	3-16

Table of Contents

**Section 4**  
**MODIFYING THE DATA BASE**

Adding Entries . . . . . 4-1  
    Using Forms to Add Entries . . . . . 4-4  
Replacing Entries . . . . . 4-6  
Deleting Entries . . . . . 4-8  
    Deleting Key Items . . . . . 4-8

**Appendix A**  
**SYNTAX**

**Appendix B**  
**ERROR MESSAGES**

**Appendix C**  
**VOLUME SPECIFIERS**

Volume Label . . . . . C-1  
Unit Specifier . . . . . C-1  
    Example . . . . . C-2

**Appendix D**  
**MATH OPERATIONS**

Examples . . . . . D-1

**Appendix E**  
**INSTALLING QUERY**

# INTRODUCTION TO QUERY

SECTION

1

The QUERY system software provides a simple method of accessing an IMAGE data base without programming effort. (The IMAGE data base must have been created with the IMAGE system software supplied with one of the computer systems listed later on this page.) You may use QUERY to do the following:

- retrieve data which meets selection criteria
- report on the data retrieved
- add or delete data entries
- modify data entries

## NOTE

QUERY system software is included with the operating system and system software of the following computer systems:

- all HP 250 Business Computer Systems
- all HP 260 Business Computer Systems

This manual describes operation of QUERY system software supplied with the most recent version operating system and system software of the previously mentioned computer systems. (At this edition of the QUERY Operator's Guide, B.08 is the most recent version of operating system and system software for these computer systems.)

Before QUERY can be used on your system, it must be installed using a supplied installation program. If you are responsible for installing QUERY on your system, refer to this manual's appendix titled "Installing QUERY".

## INTRODUCTION TO DATA BASE MANAGEMENT

A **data base** is a collection of related information, commonly accessible, that has been stored in such a way that easy access to the information is made possible. The person who designs the data base decides what information will be in the data base and how each piece of information is related to every other piece of information. To use QUERY, you need not know the process the designer uses to create the data base, but you do need to know what the information is and how it is organized.

The information stored in the data base depends upon the use of the data base. Throughout this manual, a sales analysis example is used. This is only an example to show you how to work with the data base. After you have read this first section, ask your manager or the person in charge of the computer system what information your data base stores.

### Data Item

The smallest piece of information in a data base is called a **data item**. For example, a customer name would be a data item, a customer address would be another data item, etc. When the data base is designed, a certain amount of space is reserved for storing the contents of each data item. For example, assume the customer information is stored in a card file. The first line might be reserved for the customer's name, the next two lines for the address, and so on.

Jon Doe
ABC Construction
Company
•
•
•

**Example Data Items**

Thus the first data item uses the first line and the second data item uses the next two lines. Note that cards are not really used for storage. This is only an example to help you visualize data items.

When it becomes necessary to find the card which contains the customer Jon Doe, you need a way to specify which line on each card is to be searched. Consequently, each data item is assigned a name and length. For example, the first data item might be named "customer" and have a length of one line. (When stored in the data base, the length is actually specified as a number of characters or digits.)

# Data Entry

The next level in the data base system is the **data entry**. All the data items on a card might be defined as a data entry.

data item names	data item values
Order-No	0001
Name	Jon Doe
Address	ABC Construction Company
City	Midland
State	Colorado
Country	USA
Zip-Code	80001
Order-Date	
Ship-Date	
Region	West
Product-No	1000
Price	10.00
Salesperson	KK

data item →

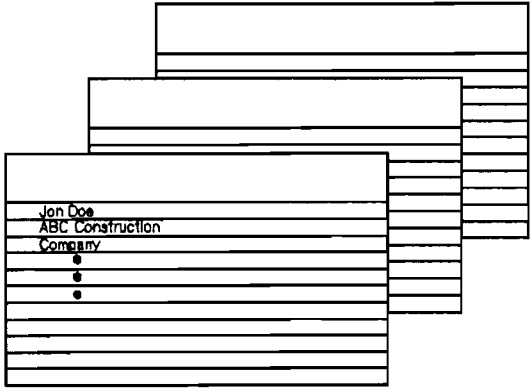
Data  
Entry

Data Entry Example

The figure shows a data entry that consists of 13 data items.

# Data Set

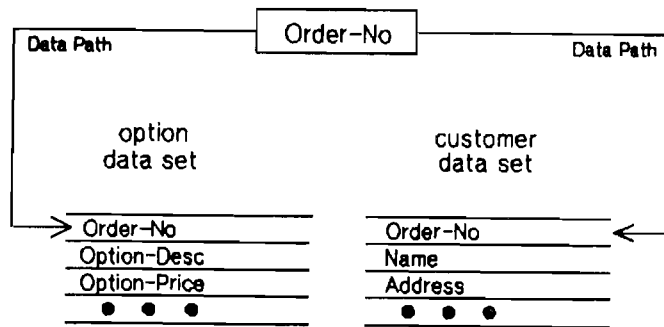
A collection of similar data entries is known as a **data set**. The following figure shows three data entries which together make up a particular data set that will be referred to as the CUSTOMER data set. A data set is similar to a file.



Example Data Sets

## Introduction to QUERY

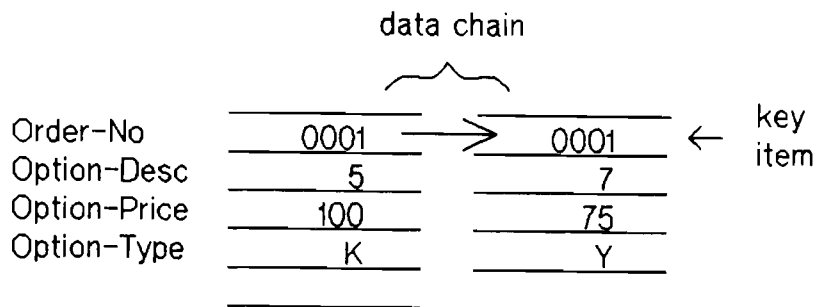
A collection of data sets is called a **data base**. The data sets in a data base are usually linked together. For example, a second data set in the Sales Analysis data base contains the data items **ORDER\_NO**, **OPTION\_DESC**, **OPTION\_PRICE** and **OPTION\_TYPE**; this data set is referred to as the **OPTION** data set. Because the information in this data set may be wanted with the information in the customer data set, the two are linked together through a third data set, **ORDER**, which has one data item **ORDER\_NO**. This link from one data set to another is called a **data path**.



Example Data Path

The data item which is linked is called a **key item**.

Now assume the customer specifies two options on the same order. This would cause the option data set to have two entries with the same order number. These entries are also linked through the key item. This link, within a data set, is called a **data chain**.



Example Data Chain

## Types of Data Sets

There are two types of data sets, **master data sets** and **detail data sets**.

In a master data set, there is only one data entry for each key item value. There may be up to eight data paths from this key item.

In a detail data set, there may be more than one data entry for each key item value. There may be up to eight key items in a data entry. For every value of a key item in a detail data set, there is a corresponding value in its associated master data set.

For example, the order data set has one data entry for each `ORDER_NO` value. The option data set has two data entries for `ORDER_NO 0001`. Since there is a value of 0001 in the option data set `ORDER_NO` data item, there must be a value of 0001 in the order data set `ORDER_NO` data item. However, there does not have to be the same value in the customer data set.

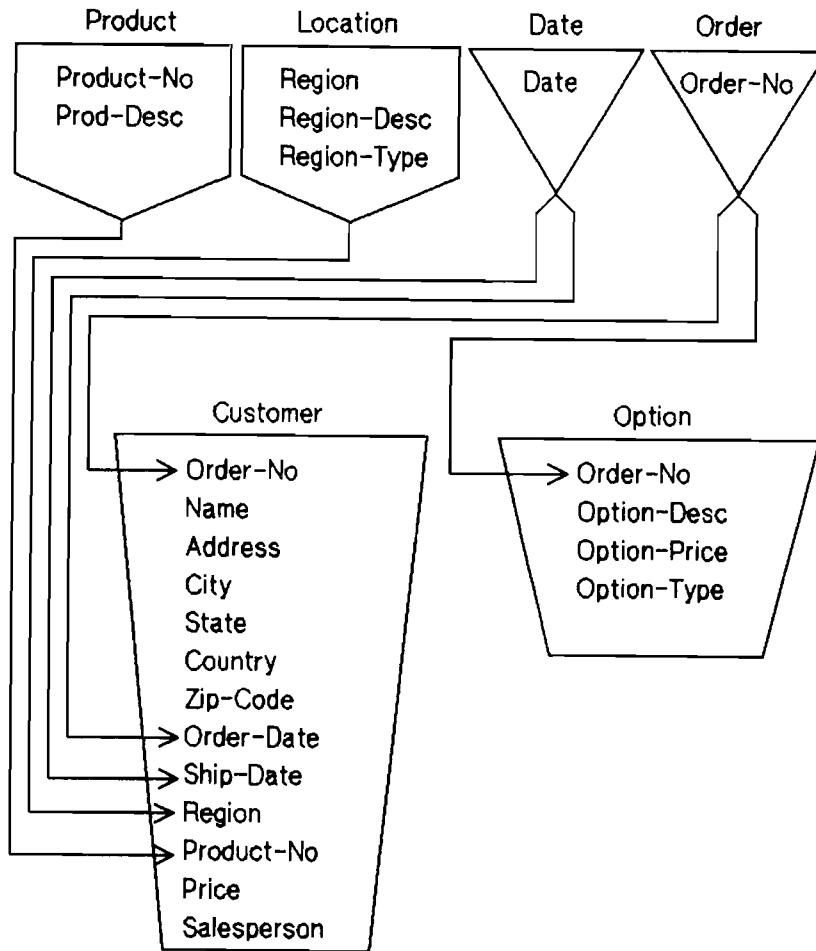
There are two types of master data sets. The **automatic master** data set contains one data item which is a key item. When a new value is added to the same key item in a detail data set, the value is automatically added to the automatic master data set. Deletions from the automatic master data set are made when the last data entry with a particular key item value is deleted from the detail data set.

The **manual master** data set contains one or more data items, one of which is a key item. Before a new key item value can be added to a detail data set that is linked to a manual master data set, the value must be first added to the manual master data set.

## Data Base

The collection of multiple detail data sets and master data sets is collectively known as a **data base**. As the information is stored into the data base, no ordering is necessary since the links provide the ordering. All data chain maintenance is performed automatically.

The following is a diagram of the Sales Analysis data base showing the data paths between master and detail data sets.



Sales Analysis Data Base



## DATA BASE SCHEMA

A data base schema is used to define the data base. There are three parts to a schema. Part one gives the name of the data base and passwords used for access to the data base. Part two lists each data item, specifying its size and type. (The type of a data item refers to the type of data that will be stored in it, such as integer data. For a full description, refer to the appendix titled "DBML Syntax" in the IMAGE Programming Manual.) Part three describes the individual data sets and the data items they contain.

Now let's look at the schema for the Sales Analysis data base.

```

1 BEGIN DATA BASE      SAD;  <<CUSTOMER SALES ANALYSIS DATA BASE>>
2
3 PASSWORDS;
4           10      SALESMAN;
5           15      MANAGER;
6           3       SECRETARY;  <<WILL HAVE READ ACCESS ONLY>>
7
8 ITEMS:
9           ADDRESS,  2X30;  <<2 LINES OF ADDRESS ALLOWED>>
10          CITY,     X16;
11          COUNTRY,  X12;
12          DATE,     I;  <<PATH FOR ORDER-DATE, SHIP-DATE>>
13          NAME,     X30;
14          OPTION-DESC, X10;
15          OPTION-PRICE, L;
16          OPTION-TYPE, I;
17          ORDER-DATE, I;  <<MUST BE YYYY>>
18          ORDER-NO,  X10;
19          PRICE,     L;
20          PRODUCT-NO, I;
21          PROD-DESC, X30;
22          REGION,   X6;
23          REGION-DESC, X30;
24          REGION-TYPE, I;
25          SALESPERSON, X4;
26          SHIP-DATE, I;  <<MUST BE YYYY>>
27          STATE,    X6;
28          ZIP-CODE,  X8;
29
30 SETS:
31
32  NAME:    DATE,AUTOMATIC(3/10,15),SALES;
33  ENTRY:   DATE(2);
34  CAPACITY: 51;
35
36  NAME:    ORDER,A(3/10,15);
37  ENTRY:   ORDER-NO(2);
38  CAPACITY: 101;
39
40  NAME:    PRODUCT,MANUAL(3,10/15),SALES;
41  ENTRY:   PRODUCT-NO(1),
42           PRODUCT-DESC;

```

## Introduction to QUERY

```
43  CAPACITY: 11;
44
45  NAME:      LOCATION,M(3,10/15),SALES;
46  ENTRY:    REGION(1),
47            REGION-DESC,
48            REGION-TYPE;
49  CAPACITY: 17;
50
51  NAME:      OPTION,D(3/10,15);
52  ENTRY:    ORDER-NO(ORDER),
53            OPTION-DESC,
54            OPTION-PRICE,
55            OPTION-TYPE;
56  CAPACITY: 300;
57
58  NAME:      CUSTOMER,DETAIL(3/10,15);
59  ENTRY:    ORDER-NO(ORDER),
60            NAME,
61            ADDRESS,
62            CITY,
63            STATE,
64            COUNTRY,
65            ZIP-CODE,
66            ORDER-DATE (DATE),
67            SHIP-DATE (DATE),
68            REGION (LOCATION),
69            PRODUCT-NO (PRODUCT),
70            PRICE,
71            SALESPERSON;
72  CAPACITY: 100;
73
74          END.
```

**Line 1:** The name of the data base is given - SAD. When you use QUERY, you must give the data base name.

**Line 3, 4, 5, and 6:** The passwords are given. The numbers (10, 15, 3) are used later in the schema. You use the words SALESMAN, MANAGER, SECRETARY when you first begin using QUERY (as explained in this manual's section titled "Getting Started").

**Lines 8 through 28:** The name, size and type of each data item is given. 2X30 specifies a two item array, of which each item is 30 characters long. X16 specifies a data item 16 characters long. I specifies that the data item contains an integer. L specifies that the data item contains a long precision number. If a data item is a number, then you can only enter a number for its value; otherwise an error will occur. If data item is a string (specified by X) then you may enter up to the specified number of characters. (Entering a value is described in this manual's section titled "Modifying the Data Base".)

**Lines 32, 33, 34:** The automatic data set DATE is described. Line 32 gives the name (DATE), the type (AUTOMATIC), the read/write passwords (3/10, 15) and the volume (SALES) on which the data set is located. The passwords (3/10, 15) mean that anyone who uses the password SECRETARY (3) can read and list the data in this data set set, but cannot write new values, change values or delete values. Anyone who uses the password SALESMAN or MANAGER can read or write (add, delete, modify) the values in this data set. Line 33 gives the name of data item (DATE) and the number of data items in detail data sets to

which are pointed (2). Line 34 gives the maximum number of data entries that are possible in this data set.

**Lines 36 through 49:** The other master data sets are described.

**Lines 51 through 56:** The detail data set OPTION is described.

**Line 52:** This specifies the name of the first data item (ORDER\_\_NO) which is a key item pointed to from the master data set ORDER.

**Lines 58 through 72:** The detail data set CUSTOMER is described.

**Line 74:** This specifies the end of a schema.

Within the schema, comments are enclosed in << >> signs. These comments aid you in understanding the schema. They also provide additional information about data item values. For example, line 12 has the following comment:

```
<<PATH FOR ORDER-DATE, SHIP-DATE>>
```

This tells the reader of the schema that this data item is used as the link between ORDER-DATE data item and SHIP-DATE data item.

Lines 17 and 26 both have the following comment:


```
<<MUST BE YYMM>>
```

This tells the reader that the values entered for the data must be formatted as YYMM. (YY means a two digit representation of the year. MM means a two digit representation of the month. For example, October 1986 represented in YYMM format would be 8610.) This is not required by the schema, but because reports are written expecting this format, you should use it.

In the schema, some data items names contain a dash (e.g., ZIP-CODE). When you use this data item in QUERY, the dash must be changed to an underscore. Thus, ZIP-CODE would be written as ZIP\_CODE.



Throughout this manual, you are asked to enter commands. To enter a command, use your workstation's keyboard to type the command name and then press the RETURN key (RETURN).

The HP 2622D and HP 2649D workstations do not have a RETURN key. On these workstations, use the large  key located at the center right of the keyboard's typewriter block. This key is the ENTER key for these workstations and functions identically to the RETURN key.

If you cannot find your workstation's RETURN key, refer to your computer's Operating Manual.

## RUNNING QUERY

If QUERY is stored on a flexible disc, place the disc in a flexible disc drive before starting QUERY; the disc must remain in the disc drive while QUERY is running. Similarly, if the data base is stored on a flexible disc, place that disc in a flexible disc drive; the disc must remain in the disc drive the entire time you are using that data base.

Before running QUERY, you must first tell your computer system on which mass storage device the QUERY software is stored. To do this, you will need to know the volume specifier of both the disc on which QUERY is stored and the disc on which your data base is stored. (A volume specifier can be either a volume label or a unit specifier.) To determine the volume specifiers you will need, consult your system's Principal Operator or refer to this manual's appendix titled "Volume Specifiers". Tell the computer which volume contains QUERY by typing the following command (*volume spec* is the volume specifier of the mass storage device on which QUERY is stored):

```
MASS STORAGE IS volume spec (RETURN)
```

This command not only tells the computer where the QUERY software is stored, it also tells the computer which mass storage device to use whenever an optional volume specifier is not supplied; this device is called the **default mass storage device**.

Next, run QUERY by typing the following command:

```
RUN "QUERY" (RETURN)
```

## Specifying a Data Base

As QUERY begins execution it displays the following prompt, asking you to enter the name of the data base and to identify the volume on which it is located:

Enter the data base name and volume:

When this prompt is displayed, enter in the data base name and the volume specifier of the storage device, in the following form:

*data base name*[*volume spec*]

### NOTE

Information enclosed in square brackets ([ ]) is optional. If you do not provide this information, the system uses a default value.

If you do not provide a volume specifier when entering the name of the data base, the system assumes that the data base resides on the default mass storage device.

QUERY next finds the data base you specified and prepares it for use. If for some reason QUERY cannot find the data base you specified, it will display the following error message:

DATA BASE NOT FOUND OR IN USE

If this should happen, check that you have entered the correct data base name and volume specifier. If you have entered the information correctly and the system still displays the error message, contact your system's Principal Operator for assistance.

Note that all error messages QUERY uses are described in this manual's appendix titled "Error Messages".

### Examples

Assume that when starting QUERY, you executed the following commands:

MASS STORAGE IS ",DBASE" **RETURN**

RUN "QUERY" **RETURN**

The system then displays a prompt, asking you to enter the name of the data base and to identify the volume on which the data base resides. If the data base is named SAD and is stored on the volume whose label is DBASE, you need only type the following:

SAD **RETURN**

If the SAD data base is stored on a disc whose volume name is FILES (not the default mass storage device), you would enter the following:

SAD, FILES **RETURN**

Alternately, if the unit specifier of the FILES volume is :U2,5,0 you could enter the following (since a volume specifier can be either the volume's label or the volume's unit specifier):

SAD:U2,5,0 **RETURN**

## Specifying a Password

Once you have supplied the name of the data base, QUERY asks you to enter your password by displaying the following prompt:

Enter the password:

Type your password and then press **RETURN**; the characters you type in response to this prompt are not displayed.

For example, if your password is OPERATOR, then you would type the following:

OPERATOR **RETURN**

After you press **RETURN**, the password prompt is cleared from the display; you are now ready to use the data base.

### NOTE

The data base name, volume and the password must be specified before you use any QUERY command, except the DO command. (The DO command is described in the paragraphs titled "Prewritten Procedures".)

## COMMUNICATING WITH QUERY

QUERY provides access to an IMAGE data base through the execution of simple commands. QUERY commands can be executed by:

- using your workstation's softkeys. This procedure is explained in the paragraphs titled "Using Softkeys".
- typing in the command name and any information needed by the command (such as data base name), and then pressing **RETURN**. The syntax for QUERY commands is discussed throughout this manual and is summarized in the appendix titled "Syntax".
- storing a list of commands in a file; then execute QUERY's DO command by either of the preceding methods. The DO command is discussed in the paragraphs titled "Prewritten Procedures".


Use the method of command execution that is most convenient for you.


### Using Softkeys

Each workstation keyboard is equipped with eight Special Function Keys, also referred to as softkeys. These keys are labeled "f1" through "f8" on the keyboard. (If you cannot locate the softkeys, refer to your computer's Operating and Managing Manual for assistance.) QUERY defines each softkey to perform a certain function when that key is pressed.



QUERY indicates the current function of each softkey by displaying a label for each key. For example, once you successfully supply the data base name and password, QUERY defines the softkeys, displaying the following on your workstation:

QUERY							
Enter the data base name and volume: SAD,SALES							
							
THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT

The position of a label on the screen corresponds with the location of a Special Function Key on the keyboard. For example, the left-most label on the screen (THREAD) defines the function of the left-most Special Function Key (key f1). With these softkey definitions, pressing  executes QUERY's THREAD command. Similarly, the right-most label on the screen (EXIT) defines the function of the right-most Special Function Key, (key f8). With each different screen displayed, QUERY can modify the definition and the label of the function key, providing only those responses that are relevant for the information displayed.

## Getting Started

### Examples

Notice that the preceding display is identical to the display on your workstation once the data base name and password are entered. The softkey label second from the right on this display is "MORE COMMANDS"; this corresponds to softkey f7.

Press **f7**. Notice that the displayed labels change; the softkeys have new definitions, as shown below.

ADD	DELETE	REPLACE			LINEAR LIST	MORE COMMANDS	EXIT
-----	--------	---------	--	--	----------------	------------------	------

Press **f7** again. Once more the displayed labels change; the softkeys have new definitions, as shown below.

DATA BASE	PASSWORD	DO	RUN	WORKFILE	INFO	MORE COMMANDS	EXIT
--------------	----------	----	-----	----------	------	------------------	------

Press **f7** once more to change the softkey definitions back to their original values. All possible QUERY commands are represented on the softkey labels and are accessed by using the softkey labeled "MORE COMMANDS". A QUERY command is accessed by first using the softkey labeled "MORE COMMANDS" until the command name is displayed as a softkey label; then press the softkey that corresponds with the label identifying the desired QUERY command.



## Getting Started

DATA BASE command							
The DATA BASE command defines the data base that subsequent Query commands will act on.							
Enter the data base name and volume:							
SAD, SALES							
DATA BASE SAD, SALES							
EXECUTE							EXIT

Notice that the command, complete with the data base name, is written on the lower part of the display; the cursor remains in the inverse video field. If the data base name displayed with the command name is not correct, simply type the correct name in the inverse video field and press **RETURN** again.

When the command is printed exactly as you want it, press the EXECUTE softkey. The command menu is displayed again, while the command to be executed is displayed at the top of the screen; QUERY then executes the command.

QUERY							
DATA BASE SAD,SALES							
-							
THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT

In many of the command displays, there is more than one field. QUERY makes it easy to move the cursor from one field to another. To move the cursor from its current field to the next field, press either **TAB** or **RETURN** key. To move the cursor from its current field to the previous field, press **SHIFT** **TAB**. Additionally, you can use the keys **←**, **→**, **↑** and **↓** to move the cursor to the desired field. Only the **RETURN** key, however, will cause the command to be displayed.

To change what you have entered, position the cursor in the field containing the incorrect information and type the correct information in the field; then press **RETURN** to display the changed command.

## CHANGING DATA BASES

When you want to work with a different data base, execute QUERY's DATA BASE command to tell QUERY the new name of the data base. To execute this command, use your workstation's softkeys or type the following (where name is the name of the new data base):

**DATA BASE** *name* [*volume spec*]

The words in upper case must be entered exactly as shown. Remember, anything shown in brackets [ ] is optional. For example, if the name of the new data base is SAD and it is stored on the volume whose label is FILES, you would type the following:

DATA BASE SAD, FILES **RETURN**

## CHANGING PASSWORDS

Frequently, a data base has several passwords. Each different password can provide a different set of capabilities. For example, a user who provides one password might only be allowed to read certain records in the data base; a user providing a different password might have the ability to read and modify all records in the data base. Changing your password can provide you with different access capabilities.

Additionally, different data bases often use different passwords to provide better data security. After changing data bases with QUERY's DATA BASE command, you may also have to change your password before you can work with the new data base.

To enter a new password, use your workstation's softkeys to execute the PASSWORD command, or type the following (where *password* is the new password):

```
PASSWORD password RETURN
```

## DATA BASE INFORMATION

If you want information about the data base you are using, execute QUERY's INFO command by using your workstation's softkeys or by typing the following:

**INFO** **RETURN**

The function of the INFO command is to list a modified schema. The information is listed to the current output device (the device specified with the OUTPUT TO command); if no output device has yet been specified, the information is listed on the display. For example, the listing for the data base SAD would appear as follows:

DATA BASE: SAD, SALES							
SETS:							
NAME:	DATE, A ON SALES						
ITEMS:	DATE		I				
ENTRIES:	14						
CAPACITY:	51						
NAME:	ORDER, A ON SALES						
ITEMS:	ORDER_NO		X10				
ENTRIES:	20						
CAPACITY:	101						
NAME:	PRODUCT, M ON SALES						
ITEMS:	PRODUCT_NO		I				
	PROD_DESC		X30				
ENTRIES:	5						
CAPACITY:	11						
							Page 1
CONTINUE							EXIT

Note the softkey labeled CONTINUE. Since the entire listing won't fit on one display page. QUERY pauses after each page. Press the CONTINUE key to display the next page of information.



NAME: LOCATION, M ON SALES  
 ITEMS: REGION X6  
 REGION\_DESC X30  
 REGION\_TYPE I  
 ENTRIES: 9  
 CAPACITY: 17

NAME: OPTION, D ON SALES  
 ITEMS: ORDER\_NO X10  
 OPTION\_DESC X10  
 OPTION\_PRICE L  
 OPTION\_TYPE I  
 ENTRIES: 41  
 CAPACITY: 300

NAME: CUSTOMER, D ON SALES  
 ITEMS: ORDER\_NO X10

CONTINUE

EXIT

NAME X30  
 ADDRESS 2X30  
 CITY X16  
 STATE X6  
 COUNTRY X12  
 ZIP\_CODE X8  
 ORDER\_DATE I  
 SHIP\_DATE I  
 REGION X6  
 PRODUCT\_NO I  
 PRICE L  
 SALESPERSON X4  
 ENTRIES: 20  
 CAPACITY: 100

PATHS:

MASTER

ASSOCIATED DETAIL(S)

=====

CONTINUE

EXIT

DATE . DATE		CUSTOMER . ORDER_DATE CUSTOMER . SHIP_DATE					
ORDER . ORDER_NO		OPTION . ORDER_NO CUSTOMER . ORDER_NO					
PRODUCT . PRODUCT_NO		CUSTOMER . PRODUCT_NO					
LOCATION . REGION		CUSTOMER . REGION					
Page 4							
DATA BASE	PASSWORD	DO	RUN	WORKFILE	INFO	MORE COMMANDS	EXIT

This first line displays the current date in the upper right corner of the screen (not shown in the example). The second line of this listing displays the name of the data base and the volume on which it is located. The next set of lines gives you the name and type of each data set and the volume on which each is located. The item names, the number of entries currently in the data set and the maximum capacity of the data set are listed. The third set of lines lists the data paths between data sets. In the data base SAD, six data paths exist.

You can compare this listing with the SAD data base schema listed in this manual's section titled "Introduction to QUERY". Note that the INFO listing does not show the data base passwords. It does show the number of entries that have been made to the data set and shows the paths between master and detail data sets.

## PREWRITTEN PROCEDURES

Sometimes a sequence of commands is done repeatedly. This sequence of commands is called a **procedure** and can be put into a file and then executed with a single command. Once the procedure is in a file you can access it at any time. The process of writing the procedure and putting it into a file is discussed in the QUERY Programming Manual.

To execute the procedure, use your workstation's softkeys or type the following to execute QUERY's DO command:

```
DO " filename [volume spec]" RETURN
```

*filename* is the name of the file in which the procedure is stored.

QUERY looks at the file on the volume specified and expects to find a procedure. For example, a procedure might consist of the following (the commands used are explained in the next section):

```
FIND OPTION FOR OPTION PRICE>"100"  
SORT BY OPTION__DESC, ORDER__NO  
OUTPUT TO PRINTER  
TOTAL OPTION__PRICE  
LIST
```

QUERY will use the commands to find certain options, sort them and list them on the printer, giving a total of OPTION\_\_PRICE. While QUERY is busy, you will not be able to type commands.

Each command is displayed as it is processed (except the password command). The cursor reappears when QUERY has finished the procedure, allowing you to enter the next command.

Getting Started

## EXIT

When you have finished using QUERY, use your workstation's softkeys to execute the EXIT command, or type the following:

**EXIT** **RETURN**

Executing EXIT logically terminates QUERY and closes the data base.

This chapter shows you how to list or report data in many forms and on different devices.

In order to report any data you must tell QUERY:

- how to find the data items and data entries that you want to list,
- where the data items and data entries are to be listed (for example, to your workstation's display or to a specific printer), and
- how the data items and data entries are to appear when listed.

## FINDING ENTRIES

When telling QUERY how to find data items and data entries, you must first specify a location (called a workfile) where QUERY can store the information it finds; this is done with the **WORKFILE** command.

If the data you want to report is contained in one data set, you can then use QUERY's **FIND** command to find the data and copy it into the workfile. If the data is stored in more than one data set, you must first use the **THREAD** command to tell QUERY how the data sets are connected and in which order they are to be searched; then use the **FIND** command to find the data items and copy them into the workfile.

### Specifying a Workfile

When finding data items and data entries for you, QUERY needs a place on a mass storage device to temporarily store the information it finds; this location is called a **workfile**. QUERY automatically creates a workfile when told to find the first data item and then deletes this workfile when you exit the QUERY program.

On systems where mass storage is limited (for example, on older systems that have only flexible disc mass storage), the principal operator often creates a workfile of a fixed size (using BASIC's **CREATE** command). Check with your system's principal operator to learn if he has already created a workfile. If so, you can tell QUERY to use that workfile by executing the following command:

```
WORKFILE ["file name [volume spec]MS"]
```

Notice that the file name is optional. If you don't supply a file name, QUERY creates its own workfile and then removes that file when you exit the QUERY program.

For example, to tell QUERY to use the file **TEMP**, located on the disc labeled **FILES**, as the workfile you would enter the following:

```
WORKFILE "TEMP, FILES" (RETURN)
```

## Finding Entries in One Data Set

Once you have specified the workfile (or decided to let QUERY create its own), you are ready to find entries and copy them into the workfile. To find entries, you must know the names of the data sets and data items. If you don't know them, use the INFO command as described in the paragraphs titled "Data Base Information".

To find entries, execute the FIND command:

```
FIND item list FOR search expression
```

The parameter *item list* specifies which data items are to be found; the parameter *search expression* specifies what conditions the data items must satisfy. A maximum of 60 data items may be specified in the item list and search expression combined.

For example, using the example data base, assume you wanted to find the name of all customers who ordered product number 92640. You would enter the following command:

```
FIND NAME FOR CUSTOMER.PRODUCT_NO = "92640" (RETURN)
```

QUERY would find all the values of the NAME data item that have a corresponding value of 92640 for the PRODUCT\_NO data item in the CUSTOMER data set. Then QUERY copies all these values into the workfile.

The syntax of the item list and search expression parameters and a description of their use is provided in the following paragraphs.

### Item List

The item list parameter is a list of data items and data sets that you want QUERY to find. QUERY finds each data entry that satisfies the search expression. It then copies the values of all data items specified in the item list for that entry into the workfile. If only a data set is specified in the item list, the entire data entry is copied in the workfile.

There are two ways to specify a data item. If a data item's name is in only one data set, you need enter only the data item name. If the data item name is in two or more data sets, precede the item name with the set name and a period: *setname.itemname*

For example, suppose that you wanted to know the customer order number for every customer that ordered product number 92640. In other words, you want to copy into the workfile the ORDER\_NO and PRODUCT\_NO data items from the CUSTOMER data set each time that the PRODUCT\_NO data item in the CUSTOMER data set equals "92640". To do this you would type the following:

```
FIND CUSTOMER.ORDER_NO FOR CUSTOMER.PRODUCT_NO = "92640" (RETURN)
```

Alternately, if you want to copy the entire data entry into the workfile each time the CUSTOMER data set's PRODUCT\_NO data item equals "92640", you would type the following:

```
FIND CUSTOMER FOR CUSTOMER.PRODUCT_NO = "92640" (RETURN)
```

As a final example, if you want to copy the values for the data items `ORDER_NO`, `NAME` and `PRODUCT_NO` into the workfile each time that the `CUSTOMER` data set's `PRODUCT_NO` data item equals "92640", you would type the following:

```
FIND CUSTOMER.ORDER_NO,NAME FOR CUSTOMER.PRODUCT_NO = "92640" RETURN
```

If you fail to specify the data set and the requested data item exists in more than one data set, `QUERY` asks you from which data set you want the data item value. For example, using the example data base, suppose that you entered the following:

```
FIND ORDER_NO FOR PRODUCT_NO="92640" RETURN
```

`QUERY` would then then display the following message:

```
ORDER_NO is a member of these sets:
  1) ORDER
  2) OPTION
  3) CUSTOMER
```

Enter the number of the set you wish to use:

You would then type the number identifying the data set to be used and press `RETURN`.

If a data set and a data item have the same name and you want `QUERY` to copy the entire data set into the workfile, add a period to the data set name:

```
FIND setname. FOR search expression RETURN
```

Otherwise, in this situation `QUERY` interprets a name without a period as a data item name.

### Search Expression

The search expression is a mathematical expression that is evaluated true or false. Data item names are used as variables. These data items cannot be arrays (such as the `ADDRESS` data item in the `CUSTOMER` data set).

Some simple examples of search expressions are:

```
SALESPERSON="SAM"
NAME > "A" AND NAME < "B"
NAME > "A" AND CUSTOMER.PART_NO < "999"
```

Notice that the values (such as `SAM`, `A`, `B` and `999`) to which data items are compared must be enclosed in quotes.

`QUERY` looks at a data entry and inserts the value of the data items specified into the search expression. If the expression evaluates true, `QUERY` copies all the data items specified in the item list and in the search expression from that data entry into the workfile. If the expression evaluates false, `QUERY` looks







## Creating Reports

If you don't know how the data sets are connected, use the INFO command. The data paths are listed.

Note that you can only thread data sets from master to detail and detail to master.

You can extend the thread through ten data sets as long as there are no breaks. For example, assume you had sets A, B, C, D, and E, and A is connected to B, B is connected to C, and D is connected to E. You can thread A, B and C together, but D and E cannot be on the thread. If you could give a second THREAD command which would cancel the first.

The formal syntax is:

```
THREAD set1, set2[; set2, set3 [; set3, set4 ...]]
```

If one master data set has paths to more than one item in a detail data set, the data item to be threaded must be specified.

```
THREAD set1, item1, set2, item2; ...
```

If you don't give the item name, QUERY issues an error message.

The THREAD command stays in effect until the next THREAD command is given or until a single set FIND or ADD command is executed.

### The FIND Command with Threaded Data Sets

After the THREAD command is given, you can find entries in more than one data set. For example, assume you want to know what options were ordered with a specific product. You would type the following:

```
THREAD CUSTOMER, ORDER; ORDER, OPTION RETURN
```

```
FIND OPTION FOR CUSTOMER. PRODUCT__NO="100" RETURN
```

QUERY looks at the first data entry in the data set CUSTOMER and tests the search expression. If the expression is true, QUERY chains to the OPTION data set through the ORDER data set and copies all the values into the workfile. If the expression is false, or after the chain operation, QUERY tests the next entry, in CUSTOMER. If the chain length is zero (there are no entries in the OPTION data set), then no values are copied into the workfile even if the search expression (CUSTOMER < PRODUCT\_\_NO = "100") is satisfied.

In any multiple data set find, QUERY searches the first data set sequentially and all other data sets on the thread via the data paths and data chains.

## LISTING THE WORKFILE

Once QUERY has found entries and copied them into the workfile, the contents of the workfile can be listed to a display, to a printer or to a spool file. When listing the contents of the workfile, you must first tell QUERY to which device the information is to be listed; this is done with the OUTPUT TO command. Once QUERY knows which device to use, you can execute the LIST command to list data items in a columnar format or the LINEAR LIST command to list data items in a linear format.

### Specifying the Output Device

To specify the device on which reports and the output from the INFO command are written, execute the OUTPUT TO command:

```
OUTPUT TO device [, "width" [, "length"]]
```

The parameter *device* tells QUERY the physical location of the output device. The following are acceptable values for *device*:

value	description
DISPLAY	specifies that your workstation's display is the output device.
PRINTER	specifies that an HP-IB printer connected at device address 0 is the output device.
"spool file name"	specifies that the listing is to be stored in a spool file to be printed later.
"device address"	an integer number specifying the physical address of the output device. (For more information on device addresses, refer to this manual's appendix titled "Volume Specifiers".)

The device specified by an OUTPUT TO command remains the designated output device until another OUTPUT TO command is executed.

The parameter *width* specifies the number of characters per line; valid values for *width* range from 20 through 264. If you don't specify it, QUERY chooses a default width depending on the output device specified: 80 characters for the display, 130 characters for the printer, 130 characters for spool files, and 80 characters for a device address.

The parameter *length* is used to specify the number of lines per page on the output device. If you don't specify it, QUERY chooses a default device page length depending on the device specified: 20 lines for the display, 66 lines for the printer or spool file and 66 lines for a device address.

For example, assume you want to list the workfile to an HP-IB printer whose device address is 0; the printer contains narrow paper that holds 80 characters per line. You would type the following:

```
OUTPUT TO PRINTER, "80"
```

## Listing Data Items in Columnar Format

Once you have found the data items you want to list, you can copy them from the workfile to the output device with the LIST command. This command prints the data items in columnar format and allows you to specify a page heading, which is printed at the top of each page of the report. The syntax of the LIST command is:

```
LIST ["heading"] [, item list]
```

The *heading* is printed at the top of each page in the report. It can consist of at most 80 characters.

The *item list* is a list of the data sets and/or data items to be listed from the workfile. If you do not supply an item list, QUERY assumes that you want all values in the workfile listed.

The values copied from the workfile are listed in columns. The head of the column lists the data set name and data item name.

For example, in the previous FIND command, values for the OPTION data set with corresponding PRODUCT\_NO value of 100 were placed in the workfile. Using the following OUTPUT TO and LIST commands, the sample listing below could be generated:

```
OUTPUT TO DISPLAY RETURN
```

```
LIST RETURN
```

OPTION ORDER_NO =====	OPTION OPTION_DESC =====	OPTION OPTION_PRICE =====	OPTION OPTION_TYPE =====	CUSTOMER PRODUCT_NO =====
101		75	0	100
101	Horn	2.5	569	100
103		75	0	100
103	Light	5	552	100
103	Mud Flaps	7.25	589	100
103	Horn	10	6987	100
103	Stripes	2.5	44	100
103	Fan	10	459	100
108		75	0	100
108	Horn	5	589	100
113		75	0	100
113	Basket	10	5	100

Page 1

THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT
--------	------	------------	-------------	--------------	------	------------------	------

The data items are listed in the same order that they are specified in the *item list* parameter. Or, if a data set is specified (or all items are to be listed), the order each data items was listed in the **FIND** command or in the schema determines the order they are listed with the **LIST** command.

If there are more columns than will fit on the output device width, **QUERY** wraps the columns around. For example:

OPTION ORDER_NO =====	OPTION OPTION_DESC =====	OPTION OPTION_PRICE =====	OPTION OPTION_TYPE =====	CUSTOMER PRODUCT_NO =====
CUSTOMER NAME =====		CUSTOMER SALESPERSON =====		
101 Noname, Joseph		75 12	0	100
101 Noname, Joseph	Horn	2.5 12	569	100
103 Hernandes, Jose		75 56	0	100
				Page 1
CONTINUE				EXIT

Another list command, **LINEAR LIST**, can be used to avoid wrap around.

## Listing Data Items in a Linear Format

**QUERY**'s **LINEAR LIST** command allows you to copy items from the workfile and list them, one item per line, on the output device. Optionally, you can also supply a page heading with this command, causing the heading to be printed at the top of each page of the report. The syntax for the **LINEAR LIST** command is:

```
LINEAR LIST ["heading"] [, item list]
```

The *heading* is printed at the top of each page in the report. It can consist of at most 80 characters.

The *item list* is a list of the data sets and/or data items to be listed from the workfile. If you do not supply an item list, **QUERY** assumes that you want all values in the workfile listed.

## Creating Reports

The **LINEAR LIST** command lists the data items and data sets specified in *item list* in a linear format (one data item per line). For example:

```
OPTION.ORDER_NO = 101
OPTION.OPTION_DESC =
OPTION.OPTION_PRICE = 75
OPTION.OPTION_TYPE = 0
CUSTOMER.PRODUCT_NO = 100
CUSTOMER.NAME = Noname, Joseph
CUSTOMER.SALESPERSON = 12
```

```
OPTION.ORDER_NO = 101
OPTION.OPTION_DESC = Horn
OPTION.OPTION_PRICE = 2.5
OPTION.OPTION_TYPE = 569
CUSTOMER.PRODUCT_NO = 100
CUSTOMER.NAME = Noname, Joseph
CUSTOMER.SALESPERSON = 12
```

```
OPTION.ORDER_NO = 103
OPTION.OPTION_DESC =
```

Page 1

CONTINUE							EXIT
----------	--	--	--	--	--	--	------

The data items are listed in the same order that they are specified in the *item list* parameter. Or if a data set is specified (or all items are to be listed), the order each data item was listed in the **FIND** command or in the schema determines the order they are listed with the **LIST** command.

## Formatting the Listing

In addition to page headings and a choice between columnar or linear listings, **QUERY** provides other commands for controlling the format of a listing:

- the **SORT BY** command sorts the data to be listed alphabetically or numerically, in either ascending or descending order.
- the **TOTAL** command computes a total for each numeric data item specified in its parameter list. The total is listed with the data item when the **LIST** command is executed.
- the **BREAK ON** command allows you to break a listing into subgroups based on the value of a data item. This command can be used together with the **TOTAL** command to create subtotals of data items.

### Sorting Data Items

Once items have been placed in the workfile with the FIND command, you can sort the items alphabetically or numerically with the SORT BY command. Its syntax is:

**SORT BY** *sort list*

The *sort list* is a list of data items to be sorted; items in the list are separated by commas. Items are sorted in ascending order unless the name of an item is followed by a D (specifying to sort in descending order). A maximum of 10 items may be sorted. An array cannot be sorted.

For example, using the sample SAD data base, assume that you want to list the CUSTOMER data set in ascending order by NAME, for every customer whose order number is larger than "106"; if a customer has placed more than one order, you want to list these orders in ascending order based on order number. To do this, you would execute the following commands:

```
FIND CUSTOMER FOR CUSTOMER.ORDER_NO>"106"

SORT BY NAME,CUSTOMER.ORDER_NO

LIST NAME,CUSTOMER.ORDER_NO,CUSTOMER.PRODUCT_NO
```

Executing these commands causes the following to be displayed:

CUSTOMER NAME =====	CUSTOMER ORDER_NO =====	CUSTOMER PRODUCT_NO =====
Arauja, Luciano A.	108	100
Aspinall,Dave	116	500
Bekker, Bart	109	500
Dalling, Jimmy	107	1000
Dolittle, Howard	117	500
Gissing, Malcomb	110	50
McMillan, Donald	119	1000
Rimer, Mike	114	300
Ross, John	112	50
Smith, Alexander	113	100
Wadsworth, John	118	500
Whitespoon, Mark	120	1000
Wright, Brian	111	50

Page 1

THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT
--------	------	------------	-------------	--------------	------	------------------	------

## Creating Reports

Alternately, assume that you wanted to list the CUSTOMER data set in reverse alphabetic order; like before, if a customer has placed more than one order, list those orders in ascending order based on order number. To do this, you would execute the following commands:

```
FIND CUSTOMER FOR CUSTOMER.ORDER_NO>"106"  
  
SORT BY NAME D,CUSTOMER.ORDER_NO  
  
LIST NAME,CUSTOMER.ORDER_NO,CUSTOMER.PRODUCT_NO
```

This would cause the following to be displayed:

CUSTOMER NAME =====	CUSTOMER ORDER_NO =====	CUSTOMER PRODUCT_NO =====
Wright, Brian	111	50
Whitespoon, Mark	120	1000
Wadsworth, John	118	500
Smith, Alexander	113	100
Ross, John	112	50
Rimer, Mike	114	300
McMillan, Donald	119	1000
Gissing, Malcomb	110	50
Dolittle, Howard	117	500
Dalling, Jimmy	107	1000
Bekker, Bart	109	500
Aspinall,Dave	116	500
Arauja, Luciano A.	108	100

Page 1

THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT
--------	------	------------	-------------	--------------	------	------------------	------

## Computing with Numeric Data Items

To compute and list the total of a numeric data item, use the TOTAL command.

### **TOTAL** *item list*

The *item list* is a list of the data items you want to be totaled. The items must be a numeric. A maximum of ten items may be totaled, and no arrays may be totaled.



For example, assume you want to know the total dollar amount that one salesperson sold in a certain month. Using the SAD data base, you would type the following:

```
FIND PRICE FOR SALESPERSON = "15"

TOTAL PRICE

LIST
```

CUSTOMER PRICE	CUSTOMER SALESPERSON						
=====	=====						
179.63	15						
46.58	15						
46.58	15						
GRAND TOTAL OF CUSTOMER.PRICE = 272.79							
Page 1							
THREAD	FIND	SORT BY	BREAK ON	OUTPUT TO	LIST	MORE COMMANDS	EXIT

While the SORT BY command remains in effect until the next SORT BY command or FIND command is executed, the TOTAL command is in effect only until the next LIST command. Thus, you must re-enter the TOTAL command for every list.

### Breaking a Listing into Groups

To break up a list by a change in the value of a single data item, use the BREAK ON command.

**BREAK ON** *item*

The data item must be in the workfile.

## Creating Reports

For example, assume you want a list of the products each salesperson has sold and you want to list the salespeople in groups. You would execute the following commands:

```
SORT BY SALESPERSON  
  
BREAK ON SALESPERSON  
  
LIST CUSTOMER.PRODUCT_NO
```

The following would be displayed:

CUSTOMER PRODUCT_NO =====							
	CUSTOMER.SALESPERSON = 10						
	1000						
	CUSTOMER.SALESPERSON = 12						
	100						
	1000						
	CUSTOMER.SALESPERSON = 125						
	100						
	CUSTOMER.SALESPERSON = 15						
	500						
	50						
							Page 1
CONTINUE							EXIT

Even though SALESPERSON was not specified in the LIST command, the breaks still occur. Also note that a sort was done prior to the LIST. If a sort had not been done, the values of SALESPERSON would not have been in order and the breaks would not be what you wanted.



## RUNNING A REPORT PROGRAM

Using **SORT BY** and **BREAK ON** allow some formatting of the lists. If you want editing capabilities and more complex formatting, however, the **REPORT WRITER** software can be used.

A subprogram can be written using **REPORT WRITER** statements and accessing the workfile. You run this subprogram from **QUERY** with the **RUN** command.

```
RUN "subprogram name [volume spec]" RETURN
```

How to write this subprogram is described in the **QUERY Programming Manual**.

### NOTE

If you have used the **DATA BASE** command to change data bases, you must first execute one of the following commands before running a subprogram with the **RUN** command: **ADD**, **FIND** or **INFO**. These commands open the data base, ensuring that the subprogram can access it.

For example, assume you want to run the subprogram **REPORT** which is located on the disc whose volume label is "PROGRAMS". You type the following:

```
RUN "REPORT, PROGRAMS" RETURN
```

QUERY provides several commands that allow you to add new data items, delete data items and replace data items. You can use these commands **only if** you have supplied a data base password that gives you permission to modify the data base.

## ADDING ENTRIES

To add entries to a data set, use QUERY's ADD command. Its syntax is:

```
ADD item list
```

If you specify a data set in *item list*, then QUERY will ask you for each data item in the data set. Entries are added to one data set at a time. For example, suppose that you want to add a new entry to the OPTION data set, which consists of four data items: ORDER\_NO, OPTION\_DESC, OPTION\_PRICE and OPTION\_TYPE. The following shows the initial command executed to add the new entry; it also shows QUERY's prompts and the supplied responses:

```
ADD OPTION
```

```
OPTION
ORDER_NO ?
190
OPTION_DESC ?
Super Tire
OPTION_PRICE ?
18
OPTION_TYPE ?
33
```

```
OPTION
ORDER_NO ?
-
```

## Modifying the Data Base

QUERY continues to prompt for new entries until the EXIT softkey is pressed. If you press EXIT before you have entered a value for every data in the list, then QUERY will not add any of the data item values to the data base. For example:

### ADD OPTION

```
OPTION
ORDER_NO ?
190
OPTION_DESC ?
Super Tire
OPTION_PRICE ?
18
OPTION_TYPE ?
33
```

```
OPTION
ORDER_NO ?
200
OPTION_DESC ?
_
```

← The EXIT softkey was pressed at this point.

Order number 190 and its associated data items are added; however, order 200 is not added because the EXIT key was pressed before values for all four data items were supplied.

You can add specific data items in a data entry. The items not entered are given a null value (0 for numeric items and blank for alphabetic items). For example, the following shows a new addition to the OPTION data set. Notice that while the value "250" is supplied for ORDER\_NO, nulls are supplied for the three remaining data items (a null is entered by simply pressing RETURN without typing in a value). The example also shows the result of executing FIND and LIST for this data entry:

**ADD OPTION**

```
OPTION
ORDER_NO ?
250
OPTION_DESC ?

OPTION_PRICE ?

OPTION_TYPE ?
```

```
OPTION
ORDER_NO ?
```

```
FIND OPTION FOR OPTION.ORDER_NO="250"
** 1 entry found
LIST
```

OPTION ORDER_NO =====	OPTION OPTION_DESC =====	OPTION OPTION_PRICE =====	OPTION OPTION_TYPE =====
250		0	0

Later, the REPLACE command can be executed to give OPTION\_DESC, OPTION\_PRICE and OPTION\_TYPE values. Note that the REPLACE command does not operate on key items. Therefore, you should not enter a null value for a key item.

You can add entries to detail data sets and manual master data sets but entries are added to an automatic master data set for you. Further, if you are adding a new key item to a detail data set and that item is pointed to by a manual master, then you must first add the new key item value to the manual master.

If you make an error in entering a value, you can either press EXIT to terminate the command (if the error was not made on the last data item in the list), or use the REPLACE command to change the value.

## Using Forms to Add Entries

If you frequently add the same data items to a data base, you may find it more convenient to enter the data items into a standard form. A version of QUERY's ADD command makes it easy to display a form and then update the data base with the data you type on the form:

```
ADD item list FROM "form name[volume spec]"
```

The *item list* is a list of specific data items or data sets to which new values will be added.

<b>NOTE</b>
-------------

When using a form to add entries to a data base, the number of items in the *item list* must equal the number of input fields on the form. (Arrays require one field for each element.) The form input order must correspond to the order of the data items in the input list.

If a data set name is supplied in the *item list*, QUERY expects an input field on the form for each data item in the data set; again, the form input field order must correspond to the data item order in the data set.

*form name* is the name of a form file, stored on the volume specified by *volume spec*. The form is created using the FORMS software supplied with your computer's operating system. For information on creating forms for use with QUERY, refer first to your system's QUERY Programming Manual and then to the FORMS Programming Manual.



When this form of the ADD command is used, the specified form is displayed and the softkeys take a new definition. For example:

```

*****
*                                     *
*                   ADD OPTION       *
*                                     *
*   ORDER NUMBER   [XXXXXXXXXXXX]    *
*                                     *
*   OPTION DESCRIPTION [XXXXXXXXXXXX] *
*                                     *
*   OPTION PRICE   [XXXXXXXXXXXX]    *
*                                     *
*   OPTION TYPE   [XXXXXX]          *
*                                     *
*****

```

ENTER					CLEAR FORM		EXIT
-------	--	--	--	--	---------------	--	------

When the form is displayed, the cursor is placed in the first input field. Type in the correct information for that field; then press **RETURN** or **TAB** to move the cursor to the next input field. If you make a mistake and want to start over, press the **CLEAR FORM** softkey.

When you tab through the last field on the form or press the **ENTER** softkey, **QUERY** assumes that you are finished entering data on the form and asks if the information is correct. If it is, enter **Y** (yes). If not, enter **N** (no) and then re-enter any incorrect value. Once the form is correct, **QUERY** reads the form and adds the data item values to the data base. **QUERY** then erases the entries, leaving a blank form so that you can continue to update the data base. When you have made all the entries, press the **EXIT** softkey.

## REPLACING ENTRIES

If you would like to change the value of one or more data items in the data base, you must first copy the data item(s) into the workfile using the FIND command. Then, you can execute QUERY's REPLACE command to modify the value(s):

### **REPLACE** *item list*

The *item list* is a list of data items whose values are to be changed. All the data items must be in the same data set. If a data set name is supplied in *item list*, QUERY assumes that every data item in the data set is to be modified.

#### **NOTE**

The REPLACE command cannot modify the value of a key item. If the *item list* contains a key item, QUERY redisplay the command and item list, positioning an error message and pointer under the key item.

One data item at a time, QUERY displays the name of each data item and that item's current value. To change the data item's value, type the new value and then press **(RETURN)**; if you want the data item to retain its current value, simply press **(RETURN)**. If while replacing entries you make a mistake and want to start again, press the EXIT softkey. Pressing the EXIT softkey while the REPLACE command is executing causes QUERY to ignore all modifications to the data base made since the REPLACE command was executed.

For example, when introducing the ADD command earlier in this section, ORDER\_NO 250 was added to the OPTION data set; the values for OPTION\_DESC, OPTION\_PRICE and OPTION\_TYPE were not entered. To change the values from null, first find the entry and then use the REPLACE command, as shown in the following example:

```
FIND OPTION FOR OPTION.ORDER_NO="250"
** 1 entry found.
REPLACE OPTION_DESC,OPTION_PRICE,OPTION_TYPE

OPTION
OPTION_DESC =
Light
OPTION_PRICE =
75
OPTION_TYPE =
0
```

In this example, only one entry was found. QUERY then prompted for each data item specified with the REPLACE command. If more than one entry would have been found in the workfile, QUERY would have prompted for each data item of each entry until either all data items were modified or the EXIT softkey was pressed.

**NOTE**

If there is more than one data entry in the workfile, the order in which QUERY prompts for the items is the same as the order in which the data entries were placed in the workfile.

## DELETING ENTRIES

If you would like to delete one or more data items, you must first use the **FIND** command to copy the data item(s) into the workfile. Then, you can execute **QUERY**'s **DELETE** command to delete the item(s):

### **DELETE** *item list*

The *item list* is a list of data items to be deleted. All data items must be in the same data set. When a data item is deleted, its value is changed to null character(s). If all the data items in a data set are specified in the item list, the entire data entry is deleted; the values of the data items are not simply changed to nulls.

For example, assume the **ORDER\_NO** 25 was cancelled. The following shows how the entry is deleted from the data base:

```
FIND CUSTOMER FOR CUSTOMER.ORDER_NO="25"  
** 1 entry found  
DELETE CUSTOMER
```

```
FIND OPTION FOR OPTION.ORDER_NO="25"  
** 1 entry found  
DELETE OPTION
```

<b>NOTE</b>
-------------

If one of the entries in a list cannot be deleted, **QUERY** issues an error message; it then goes to the next entry and continues deleting.

## Deleting Key Items

It is not possible to delete just the key item in a detail data set; the entire data entry must be deleted.

The entries in a manual master data set are deleted in the same manner as deleting an entry from a detail data set. To delete a key item from a manual master data set you must first delete all entries in associated detail data sets that have the same key item value as that of the manual master you wish to delete.

Entries are deleted from an automatic master data set for you.

**RUN "QUERY***volume spec***"**

Begins QUERY operation.

**ADD** *item list* **[FROM "form name***volume spec***"]**

Adds values to the data items or data set listed. A form can be used to input the values.

**[BREAK ON** *item***][TOTAL** *item list***]**

Sets report breaks and their associated totals for the LIST or LINEAR LIST commands. TOTAL alone causes a grand total to be printed. A maximum of ten items can be totaled.

**DATA BASE** *data base***[volume spec]**

Causes future commands to operate on the specified data base.

**DELETE** *item list*

Deletes data item values or entries from the data set. The data items must have been found by the previous FIND command.

**DO "file name***volume spec***"**

Transfers control to a file containing QUERY commands. At the end of the file, control transfers back to operator.

**EXIT**

Terminates QUERY.

**FIND** *item list* **FOR** *search expression*

Finds entries which satisfy the search expression and copies the data items listed in the item list or the entire data entry (if the set name is in the item list) into the workfile.

**INFO**

Prints information about the data base on the current output device.

**LINEAR LIST** **[ "heading" [,** *item list***]**

Lists data items from the workfile in a linear format (one item per line) on the current output device using all BREAKS and TOTALS specified.

**LIST** **[ "heading" [,** *item list***]**

Lists data items from the workfile in columnar format (one data entry per line) on the current output device using all BREAKS and TOTALS specified.

**OUTPUT TO** *device* **[, "width" [, "length"]]**

Changes the output device or spool file for future LIST, LINEAR LIST and INFO commands.

## Syntax

**PASSWORD** *password*

Defines the password to be used for subsequent commands.

**REPLACE** *item list*

Replaces values for the data items specified which are in the workfile.

**RUN** "*subprogram name[volume spec]*"

Causes a report subprogram to be run.

**SORT BY** *item list*

Sorts the entries in the workfile by the data items listed. Data items are sorted in ascending order unless a D follows the data item name.

**THREAD** *set1, set2[, set2, set3 ...]*

Defines the order in which data sets are accessed during the FIND command.

**WORKFILE** "*file name[volume spec]*"

Specifies the workfile to be used for subsequent commands.

If an error is made when entering a command, QUERY points to the error with a ^ sign and outputs a message.

For example:

```
OUTPUT PRINTER
      ^
EXPECTED 'TO'
```

The different error messages that QUERY might issue are listed alphabetically on the following pages, along with descriptions of the error messages. Where possible, the error message description includes suggestions for corrective actions.

## A QUERY FILE IS MISSING FROM THE DISC

One of the programs or data files that QUERY uses is not on the QUERY disc.

## ABNORMAL QUERY TERMINATION

You have caused QUERY to abort operation. This may have caused the data base to be destroyed. Consult your system's Principal Operator for assistance.

## ARITHMETIC OPERATOR IN STRING EXPRESSION

In the FIND search expression a string type data item is being evaluated to a numeric expression (e.g., NAME="1" + "1") or a string is being used in numeric expression (e.g., "ABC" + "3" = PRICE).

## ATTEMPTED DIVISION BY ZERO

The divisor in an equation is zero. Check all the values for the data items used in the search expression.

## ARRAY ITEM NOT ALLOWED HERE

The data item in the FIND search expression and sort list, as well as the data item in the item list of a BREAK ON or TOTAL command must be a simple item. The INFO command lists all the data items and their types.

## AUTOMATIC MASTER IS FULL

You cannot add any more entries which have a key item value not in the automatic master data set. The schema must be changed and the entire data base reloaded to correct this.

## CANNOT MODIFY KEY ITEM

A key item value cannot be changed with the REPLACE or DELETE command. You must delete the entire entry and then add it with the change.

## Error Messages

### **CLOSING PARENTHESIS MISSING**

You have omitted a closing parenthesis.

### **CLOSING QUOTE MISSING**

You have omitted a closing quotation mark.

### **DATA BASE CORRUPT**

The data base is not stored correctly. Do not attempt further use of the data base. The data base needs to be unloaded then re-loaded (refer to the IMAGE Programming Manual).

### **DATA BASE NOT CREATED**

The data base has not been prepared for use. The DBCREATE utility must be run (refer to the IMAGE Programming Manual).

### **DATA BASE NOT FOUND OR IN USE**

QUERY cannot open the data base, either because it could not find it on the disc or because it is open to another user. Using the CATALOGUE command, check the disc directory to see if the data base is on the disc. Also check that there is no other user.

### **DATA BASE ROOT FILE NOT FOUND**

QUERY cannot find the root (main) file of the data base. Check the disc file directory with a CATALOG command. If this error occurs during an EXIT, the disc with the root file must be returned to the disc drive and the EXIT performed again. Otherwise, the data base will be corrupt and you will not be able to use it again.

### **DATA SET IN USE**

Someone or some program is using the data set you specified. QUERY must have exclusive use of the data set.

### **DATA SET IS FULL**

You cannot add any more entries to the data set. The schema must be changed to correct this.

### **DISC DIRECTORY OVERFLOW**

The file directory on the disc is full.

### **DISC IS WRITE PROTECTED**

You cannot write data to the specified disc. This includes creating files on the disc.

### **DISC NOT READY**

The drive door is open or the disc is not prepared for use.

### **'DO' COMMAND NOT ALLOWED IN A 'DO' FILE**

The DO file has a DO command in it. The DO command cannot be stacked.

### **DUPLICATE KEY ITEM IN MANUAL MASTER**

You are attempting to add a key item value which already exists in the manual data set.



**EXCEEDS ITEM LIMIT OF 10**

Only 10 items can be totaled. Enter the command again limiting the number of items.

**EXCEEDS ITEM LIMIT OF 60**

The total number of data items in the FIND item list and search expression cannot exceed 60.

**EXPECTED A COMMA**

Check the syntax of the command and then re-enter it correctly.

**EXPECTED A DETAIL SET**

The THREAD command links detail sets to master sets. Two master sets cannot be directly linked together.

**EXPECTED A LOGIC OPERATOR**

A logic operator (AND, OR) must be used between expressions in the FIND search expression.

**EXPECTED A MASTER SET**

The THREAD command links detail sets to master sets. Two detail sets cannot be directly linked together.

**EXPECTED A RELATIONAL OPERATOR**

A relational operator (>, <, =, #) is required in each expression of the FIND search expression.

**EXPECTED A SET OR ITEM NAME**

The value entered for the name of the set or data item is not a valid set or item name.

**EXPECTED 'BASE'**

The word BASE was omitted from the DATA BASE command.



**EXPECTED 'BY'**

The word BY was omitted from SORT BY command.

**EXPECTED END OF LINE**

The end of the command was read, but more characters followed. Check the syntax of the command.

**EXPECTED 'FOR'**

The word FOR was omitted from the FIND command.

**EXPECTED 'LIST'**

The word LIST was omitted from the LINEAR LIST command.

**EXPECTED 'ON'**

The word ON was omitted from the BREAK ON command.

## Error Messages

### **EXPECTED 'TO'**

The word TO was omitted from the OUTPUT TO command.

### **EXPECTED 'TOTAL'**

Something other than the word TOTAL follows the break item in the BREAK ON command.

### **EXPRESSION MUST CONTAIN AN ITEM OR 'POS'**

The search expression in the FIND command must contain a data item or a POS function.

### **FATAL COMMAND xxx ENCOUNTERED**

An error was found during the execution of the command. The number -xxx- following the message is the error code. Refer to your computer's Operating and Managing Manual for an explanation of the error code.

### **FILE NOT FOUND, WRONG TYPE, OR BUSY**

The DO file, workfile, RUN program file or FORM file cannot be found or the type is wrong, or some one else is using it. Using the CATALOGUE command, check the disc directory for file names and file types.

### **FIND COMMAND TOO LONG**

There are too many characters in the command.

### **INCOMPATIBLE DATA BASE VERSION**

The data base was created on another version of IMAGE and cannot be used on this version.

### **INPUT EXCEEDS 160 CHARACTERS**

A command can consist of at most 160 characters.

### **INSUFFICIENT DISC SPACE FOR FILES**

QUERY cannot create a workfile, scratch file or spool file because of insufficient room on the specified disc.

### **INSUFFICIENT MEMORY FOR DATA SET LOCK**

The Common Block of memory is full and another data set cannot be locked at this time.

### **INSUFFICIENT NUMBER OF FIELDS IN FORM**

There are not enough fields in the form to enter values for all the data items.

### **INTERMEDIATE RESULT OVERFLOW**

While the search expression was being evaluated, an overflow occurred. The expression must be modified and the FIND command entered again.

### **INVALID COMMAND**

You entered a command unknown to QUERY. Check that the command is spelled correctly.

**INVALID DATA BASE NAME**

The name you entered cannot be a data base name. Check that you have entered it correctly.

**INVALID DATA IN 'DO' FILE**

The entry in the file is not a valid QUERY command. This file may not be the DO file. Check the contents of the file.

**INVALID DATE**

The date must be entered in the format MM/DD/YY for the U.S. or DD/MM/YY for the U.K. The date may be from Jan 1, 1978 to Dec 31, 1999.

**INVALID DEVICE SPECIFICATION**

The device address in the OUTPUT TO command has been specified incorrectly, or is not PRINTER or DISPLAY.

**INVALID EXPRESSION**

The expression in the FIND search expression cannot be used by QUERY. Refer to the paragraphs titled "Search Expression" in in this manual's section titled "Creating Reports" for more information.

**INVALID FILE NAME**

The name you specified for a file is not a valid disc file name.

**INVALID FILE PROTECT KEY**

The file you specified is protected and you did not enter the correct protect code.

**INVALID INTEGER PRECISION NUMBER**

The data item is integer precision and the value you entered for the data item is not an integer precision number. Integer precision is described in this manual's appendix titled "Math Operations".

**INVALID LENGHT SPECIFICATION**

The page length specified in the OUTPUT TO command is not valid. Check that it is numeric and between 20 and 30000.

**INVALID LONG PRECISION NUMBER**

The data item is long precision and the value you entered for the data item is not a long precision number. Long precision is described in this manual's appendix titled "Math Operations".

**INVALID PASSWORD**

The password you specified is not recognized by QUERY.

**INVALID REPORT SUBPROGRAM**

The subprogram has a name other then REPORT, the parameters are incorrect, the subprogram is binary, or there is no subprogram in the file specified. Refer to the QUERY Programming Manual for a description of the report subprogram.

## Error Messages

### **INVALID SHORT PRECISION NUMBER**

The data item is short precision and the value entered for the data item is not a short precision number.

### **INVALID WIDTH SPECIFICATION**

The page width specified in the OUTPUT TO command cannot be used. Check that the value is an integer between 20 and 264.

### **ITEM NOT IN SPECIFIED SET**

The data item specified is not in the specified data set.

### **KEY ITEM POINTS TO DETAIL ENTRY**

You are attempting to delete an entry from a manual master which still points to an entry in a detail set.

### **LIST TITLE EXCEEDS 80 CHARACTERS**

The optional heading cannot be more than 80 characters in length.

### **MORE THAN ONE PATH BETWEEN THESE SETS**

You must specify the key data items to show which path to use.

### **MORE THAN ONE SET NOT ALLOWED HERE**

You can only add, delete or replace items in one data set at a time.

### **NAME EXCEEDS 15 CHARACTERS**

The data set or data item name exceeds 15 characters. Use the INFO command to check the schema for the correct name and then re-enter it.

### **NO ENTRIES HAVE BEEN FOUND**

There are no data entries in the workfile.

### **NO MATCHING KEY ITEM IN MANUAL MASTER**

You cannot add a new key item value to a detail data set unless the value already exists in the manual master set which points to it.

### **NO WRITE ACCESS WITH CURRENT PASSWORD**

You cannot add, replace or delete entries in this data set with your password.

### **NOT A VALID PATH**

The only paths are those connecting a detail set and a master set together. The key item must be in both sets.

### **NUMBER NOT IN QUOTES OR INVALID CHARACTER**

You entered a character which QUERY cannot use (e.g., lowercase or line drawing characters) or there are no quotes around a number. Re-enter the command using only the standard character set (in upper-case).

**NUMERIC ITEM OR CONSTANT REQUIRED**

A numeric data item is being tested for a string value, or you are adding a string value to a numeric data item.

**NUMERIC ITEM REQUIRED**

Only numeric data items may be totaled.

**OUTPUT DEVICE IS IN USE**

Someone else is using the output device. Choose another printer, or use a spool file for output.

**OUTPUT DEVICE NOT AVAILABLE**

The device specified is not currently ready for output, or the spool DROM is not loaded.

**POSSIBLE VOLUME LABEL CONFLICT**

The flexible disc was replaced after the last disc operation. Either replace the last flexible disc used, or re-enter the command. This is only a warning.

**QUERY DISC DIRECTORY OVERFLOW**

The file directory is full. Include a different volume specifier with the WORKFILE command.

**REPORT BREAK PREVIOUSLY SET ON THIS ITEM**

The data item in the current BREAK ON command was used in a previous BREAK ON command.

**SET NAME ONLY NOT ALLOWED HERE**

A set name cannot be used alone in the FIND search expression, the BREAK ON list, the TOTAL list, or the SORT BY list.

**SET OR ITEM NOT FOUND**

QUERY cannot find the data set or data item name in this data base.

**SET OR ITEM NOT IN PREVIOUS FIND**

The set or item specified in the command is not in the workfile.

**SPOOL FILE CANNOT BE EXPANDED**

The spool file is full and there is no room on the disc to expand it. Use another for further output.

**STRING ITEM OR STRING REQUIRED**

A numeric data item or constant is being used where a string is required.

**STRING TOO LONG**

The string entered is too long for the data item.

**THIS SET NOT IN THE SPECIFIED THREAD**

One or more of the sets in the FIND command were not in the previous THREAD command.

**VOLUME NOT FOUND OR WRITE PROTECTED**

The specified volume could not be found or is write protected. Check that the volume is correctly specified. If the volume is on a removable medium (such as a flexible disc), verify that the medium is correctly inserted and that the medium is not write protected.

**WORKFILE IS TOO SMALL**

The workfile is not large enough to store all the data entries.

**WORKFILE OR SCRATCH FILE NAME CONFLICT**

QUERY cannot create the workfile or scratch the file because the name it uses has already been used on another file.

**WORK RECORD EXCEEDS 256 BYTES**

A maximum of 10 data items may be sorted. Further, the total length of the data items plus 2 bytes times the length of the thread must be less than or equal to 256 bytes. How to calculate the length of data items (variables) is described in the section titled "Memory Usage", in the BASIC Programming Manual.

**WRITE TO AUTOMATIC MASTER NOT ALLOWED**

You cannot use ADD, DELETE or REPLACE commands on data items in the automatic master.

**WRITE TO THIS SET OR ITEM NOT ALLOWED**

You cannot use the ADD, DELETE, or REPLACE commands on the item or set specified because the control number is set to write inhibit (refer to the QUERY Programming Manual).

Whenever you use a program, subprogram, or other file you must specify the name of file and (in most cases) identify the volume where it is located. You identify the volume by supplying a volume specifier with the file name. A volume specifier is either a volume label or a unit specifier.

## Volume Label

A volume label is a one to eight character string assigned to a storage medium via the PRINT LABEL statement. The first character of a volume label cannot be "," or ":", since these characters are used in commands to delimit a volume label or file specifier from other parameters.

## Unit Specifier

A unit specifier consists of several parameters in the following format (optional parameters are enclosed in brackets):

```
device type[2[device address[unit code]]]
```

*device type* a single character which specifies the type of mass storage device. The currently supported values for *device type* are shown below:

Device Description	<i>device type</i>
HP 250's built-in 5 Mbyte Disc Drive . . . . .	G
HP 250's built-in 10 Mbyte Disc Drive . . . . .	H
HP 250's built-in 15 Mbyte Disc Drive . . . . .	I
HP 7902 Flexible Disc Drive . . . . .	F
HP 7906A/S Removable Disc Cartridge . . . . .	C
HP 7906A/S Fixed Disc . . . . .	D
HP 7908P Disc Drive (fixed disc) . . . . .	Q
HP 7909 Disc Drive. . . . .	E
HP 7910 Disc Drive. . . . .	L
HP 7911P Disc Drive (fixed disc) . . . . .	R
HP 7912P Disc Drive (fixed disc) . . . . .	S
HP 7941A/7942A Disc Drive (fixed disc) . . . . .	T
HP 7945A/7946A Disc Drive (fixed disc) . . . . .	U
HP 7957/7958 Disc Drive . . . . .	V
HP 9895 Flexible Disc Drive . . . . .	F
HP 9133 Disc Drive (built-in 3.5" flexible disc drive) . . . . .	A
HP 9133/9134 Disc Drive (fixed disc) . . . . .	M
HP 9144 Cartridge Tape Drive. . . . .	K
HP 9153 Disc Drive (built-in 3.5" flexible disc drive) . . . . .	A
HP 9153/9154 Disc Drive (fixed disc) . . . . .	N
Cartridge Tape Drive (built-in on the HP 7908P, HP 7911P, HP 7912P, HP 7942A and HP 7946A). . . . .	K

## Volume Specifiers

**2** select code of the HP-IB Interface card.

*device address* the physical address of the device; this can be an integer expression with a value 0 through 7. If the device address is not specified, the system uses the highest address of all devices of type *device type* currently connected to the system.

For example, suppose that two HP 7908P disc drives are connected to a system (addresses 0 and 4). Then if the unit specifier ":Q" is entered, the system "assumes" that the device specified is the device at address 4 (the higher address of the two type "Q" devices).

*unit code* a single digit used to distinguish between two or more HP-IB devices having the same device address. (For example, the HP 7908P contains a disc drive and a tape drive which are located at the same device address.) The unit code can be an integer expression with value 0 through 7. For an the HP 7908P, HP 7911P, HP 7912P, HP 7942A and HP 7946A Disc Drives (with cartridge tape drive), the disc unit code is always 0 and the tape unit code 1. For the HP 9895A Disc Drive, the unit code of the left drive is 0 while the unit code of the right drive is 1 (if the drive is equipped with two drives).

The default unit code for disc devices is "0"; the default unit code for tape devices is "1".

### Example

Suppose that you are using a system equipped with the following mass storage devices:

- HP 7946A Disc Drive (device address is set to 5).
- HP 7945A Disc Drive (device address is set to 4),
- HP 9895A Dual Drive Flexible Disc Drive (device address is set to 3).

The unit specifier for the HP 7946A disc drive is ":U2,5,0". Because the device type tells your system that the device is a fixed disc drive, the unit code can be omitted. Thus, the device specifier is ":U2,5". And since the HP 7946A has the highest device address of all type "U" devices connected to the system, the device address can also be omitted. Thus the device specifier is ":U2". However, since "2" is no longer needed, the device specifier for the HP 7946A can be further shortened to ":U".

The unit specifier for the HP 7945A disc drive is ":U2,4,0". Because the device type tells your system that the device is a fixed disc drive, the unit code can be omitted. Thus, the device specifier is ":U2,4". Note that this specifier cannot be shortened any further. If the device address was omitted, the system would assume the address to be the highest of all type "U" devices connected to the system and would select the address of the HP 7946A by default.

The unit specifiers for the HP 9895A disc drive are ":F2,3,0" (left drive of the dual drives) and ":F2,3,1" (right drive of the dual drives). The unit specifier ":F" specifies the left hand drive since the default unit code for disc devices is "0".



# MATH OPERATIONS

Mathematical operations are used in the FIND command's search expression. The following describes valid operators for mathematical operations in a search expression:

+	addition	/	division
-	subtraction	*	multiplication
AND	logical AND	POS	position function
OR	logical OR	=	equal to
>	greater than	#	not equal to
<	less than		

## Examples

Suppose that you wanted to find data entries where the data item NAME equals "Hewlett-Packard". The search expression to be used with the FIND command would be:

```
NAME = "Hewlett-Packard"
```

Now suppose that you only want to find data entries where the data item NAME equals "Hewlett-Packard" and also where the data item ORDER\_NO equals "1000". The search expression would be:

```
NAME = "Hewlett-Packard" AND ORDER_NO = "1000"
```

If you wanted to find all data entries where the data item NAME does not equal "Hewlett-Packard" or where the data item ORDER\_NO equals "1000", you would use the following search expression with the FIND command:

```
NAME # "Hewlett-Packard" OR ORDER_NO = "1000"
```

Suppose that you wanted to find all data entries where the data item PRICE is greater than or equal to "1000". The search expression would be:

```
PRICE >= "1000"
```

## Math Operations

Now suppose that we have a large data base of products and want to find all products where the profit is less than "100". You might use the following search expression:

$$(PRICE - DISCOUNT) < (COST + 100)$$

As a final example, suppose that you want to find all data entries where the data item NAME contains "John"; NAME may contain more than "John", but you only want to know if "John" is somewhere in NAME. The search expression would be:

$$POS(NAME, "John") > 0$$

If the string "John" is located anywhere in the value of NAME, POS returns a numeric value indicating the location where "John" begins. Thus if "john" is part of the value of NAME, the value will be greater than 0 and the expression is true.

The version of QUERY supplied with the system software is a "bundled" software version, containing prompts and messages in six different languages (French, German, Italian, Katakana, U.K. English and U.S. English). Before running QUERY, use the following procedure to select the language with which you want to work, and to copy this single language version of the software from its distribution medium to the disc from which QUERY will be run (this can be either a fixed disc or a flexible disc):

1. If your system is not powered up and running, power up the workstations and mass storage devices connected to the system; then power up the system SPU. Refer to your system's Operating and Managing Manual if you need assistance with powering up the system.
2. Your computer's system software is supplied either on cartridge tape or on a set of flexible discs. Locate the cartridge tape or set of flexible discs containing the most recent version of the system software.

**If your computer's system software is supplied on cartridge tape, insert the cartridge tape into the cartridge tape drive; then execute the following command:**

```
RUN "QUERIN ,SYSTEM"
```

**If your computer's system software is supplied on flexible discs, locate the flexible disc labeled "QUERY"; insert this disc into the system's flexible disc drive. Then execute the following command:**

```
RUN "QUERIN ,QUERY"
```

## NOTE

Flexible disc drives using 3.5" media do not lock the flexible disc in the drive while in use. To avoid possible data loss, do not remove the flexible disc from the drive until instructed to do so.

## Installing QUERY

3. The installation program displays the following screen, asking you to select the language that you want QUERY to use when displaying messages and when prompting for information. Press the softkey labeled with the language in which you want to work.

### NOTE

The difference between U.S. and U.K. versions of QUERY are in date handling and currency symbols.

QUERY INSTALLATION PROGRAM							
The QUERY program has been written in 6 languages.							
Press the softkey labeled with the language that you prefer for your prompts and messages.							
U.K. ENGLISH and U.S. ENGLISH differ in date and currency symbol representation							
Please select the language for your QUERY program messages and prompts.							
FRENCH	GERMAN	ITALIAN	KATAKANA	U.K. ENGLISH	U.S. ENGLISH		EXIT PROGRAM

4. The installation program next displays the language you selected and asks you to verify that it is correct. If the language displayed is not correct, press  (labeled "NO"); the installation program prompts you once again for the operating language. If the language displayed is correct, press  (labeled "YES"); then proceed to the next step.

5. Next the installation program asks you to identify the "SOURCE" volume (the volume containing the "bundled" QUERY software). A list of all currently available mass storage devices and their volumes is displayed (an example is shown below).

QUERY INSTALLATION PROGRAM							
<u>LABEL</u>	<u>DEVICE</u>			<u>COMMENT</u>			
SYSTEM	CTD	:	K2,5,1				
LOAD	7945/46	:	U2,5,0				
DBASE	7945/46	:	U2,4,0				
	9133/34	:	M2,2,0				

Please select the SOURCE volume.

SYSTEM :K2,5,1	LOAD :U2,5,0	DBASE :U2,4,0	:M2,2,0				EXIT
-------------------	-----------------	------------------	---------	--	--	--	------

If your computer's system software is supplied on cartridge tape, press the softkey whose label includes the volume label "SYSTEM".

If your computer's system software is supplied on flexible disc, press the softkey whose label includes the volume label "QUERY".

## Installing QUERY

The word "SOURCE" is added to the comment column in the list on the display, identifying the volume you specified (similar to the following example).

QUERY INSTALLATION PROGRAM							
<u>LABEL</u>	<u>DEVICE</u>		<u>COMMENT</u>				
SYSTEM	CTD	:	K2,5,1	SOURCE			
LOAD	7945/46	:	U2,5,0				
DBASE	7945/46	:	U2,4,0				
	9133/34	:	M2,2,0				
Please select the DESTINATION volume.							
	LOAD	DBASE					EXIT
	:U2,5,0	:U2,4,0	:M2,2,0				

- The installation program now asks you to identify the "DESTINATION" volume (the volume to which the QUERY software should be copied). This device is called the DESTINATION device. Press the softkey identifying the disc to which QUERY should be copied.

The word "DESTINATION" is added to the comment column on the display, identifying the volume on which QUERY will be installed.

QUERY INSTALLATION PROGRAM			
<u>LABEL</u>	<u>DEVICE</u>		<u>COMMENT</u>
SYSTEM	CTD	:K2,5,1	SOURCE
LOAD	7945/46	:U2,5,0	
DBASE	7945/46	:U2,4,0	DESTINATION
	9133/34	:M2,2,0	

---

Are SOURCE and DESTINATION correct?

NO				YES			EXIT
----	--	--	--	-----	--	--	------

- As shown in the previous example screen, the installation program next asks you to verify that the SOURCE and DESTINATION volumes are correctly specified. If you wish to specify a different SOURCE or DESTINATION volume, press  (labeled "NO"); the program will once again ask you to select these volumes. If the SOURCE and DESTINATION volumes are correct, press  (labeled "YES"); then proceed to the next step.
- The installation program now copies QUERY from the SOURCE volume to the DESTINATION volume. When the installation is complete, the following message is displayed:

END OF QUERIN.

Remove the SOURCE volume (flexible disc or cartridge tape) from its mass storage device and store it in a safe place. You will need this volume should you ever have to re-install QUERY. (If you simply want to copy the installed single language version of QUERY to a different volume, use the ROUTIL, BACKUP, DUPL or FVBACK utilities as described in your system's Utilities Manual.)





**A**

adding entries (using forms), 4-4  
adding entries, 4-1  
addition, D-1  
address, C-1  
automatic master, 1-5  
ADD ... FROM ..., 4-4, A-1  
ADD, 4-1  
AND, D-1

**B**

breaking a listing into groups, 3-13  
BREAK ON ... TOTAL ..., A-1  
BREAK ON ... TOTAL, 3-15  
BREAK ON, 3-10, 3-13, A-1

**C**

changing a data entry, 4-6  
changing data bases, 2-10  
changing passwords, 2-11  
closing the data base, 2-16  
columnar format, 3-8  
command entry, 2-4  
    softkeys, 2-4  
command syntax, A-1  
commands, syntax of, A-1  
computing with numeric data items, 3-12

**D**

data base schema, 1-7  
data base, 1-2, 1-4, 1-6, 2-10  
  adding data entries, 4-1  
  adding entries (using forms), 4-4  
  adding entries, 4-1  
  changing data bases, 2-10  
  closing the data base, 2-16  
  data base schema, 1-7  
  data chain, 1-4  
  data entry, 1-3  
  data item, 1-2  
  data path, 1-4  
  data set, 1-3, 1-5  
  deleting key items, 4-8  
  deleting entries, 4-8  
  information, 2-12  
  key item, 1-4  
  modifying, 4-1  
  password, 1-7, 1-8, 2-3, 2-11  
  replacing entries, 4-6  
  schema, 1-7  
  specifying, 2-2  
data chain, 1-4  
data entry, 1-3  
data item, 1-2  
data path, 1-4  
data set, 1-3, 1-5  
  adding entries (using forms), 4-4  
  adding entries, 4-1  
  automatic master, 1-5  
  deleting entries, 4-8  
  deleting key items, 4-8  
  detail data set, 1-5  
  manual master, 1-5  
  master data set, 1-5  
  replacing entries, 4-6  
  threading multiple data sets, 3-5  
default mass storage device, 2-1  
deleting entries, 4-8  
deleting key items, 4-8  
detail data set, 1-5  
device address, 3-7, C-1  
device type, C-1  
division, D-1  
DATA BASE, 2-10, A-1  
DELETE, 4-8, A-1  
DISPLAY, 3-7  
DO, 2-15, A-11

**E**

entering commands, 2-4  
equal to function, D-1  
error messages, B-1  
EXIT, 2-16, 4-2, A-1

**F**

finding entries in multiple data sets, 3-5  
finding entries in one data set, 3-2  
finding entries, 3-1  
formatting the report, 3-8, 3-9, 3-10  
    breaking a listing into groups, 3-13  
    subtotaling data items, 3-15  
    totaling data items, 3-12  
FIND ... FOR ..., 3-2, A-1

**G**

generating reports, 3-1, 3-7  
greater than, D-1

**H**

heading, 3-8, 3-9

**I**

information, 2-12  
installation, 1-1, E-1  
item list, 3-2  
itemname, 3-2  
INFO, 2-12, A-1

**K**

key item, 1-4, 3-5, 4-8  
    deleting, 4-8

## L

- label (softkey), 2-5
- label (volume), C-1
- language, E-1
- leaving QUERY, 2-16
- less than, D-1
- linear format, 3-9
- link, 1-4
  - data chain, 1-4
  - data path, 1-4
  - key item, 1-4
- listing the workfile in columnar format, 3-8
- listing the workfile in linear format, 3-9
- listing the workfile, 3-7
- loading QUERY, 2-1
- logical AND, D-1
- logical OR, D-1
- LINEAR LIST, 3-9, A-1
- LIST, 3-8, 3-9, A-1

## M

- manual master, 1-5
- mass storage device, C-1
- master data set, 1-5
  - automatic master, 1-5
  - manual master, 1-5
- math operations, D-1
- messages, B-1
- modifying the data base, 4-1
  - adding entries, 4-1, 4-4
  - deleting entries, 4-8
  - deleting key items, 4-8
  - replacing entries, 4-6
- multiplication, D-1
- MASS STORAGE IS, 2-1

## N

- not equal to function, D-1

## O

- operating language, E-1
- output device, 3-7
- OR, D-1
- OUTPUT TO, 3-7, A-1

**P**

password, 1-7, 1-8, 2-3, 2-11, A-2  
position function, D-1  
prewritten procedure, 2-15  
procedure, 2-15  
PASSWORD, A-2  
POS, D-1  
PRINTER, 3-7

**R**

removing a data entry, 4-8  
replacing entries, 4-6  
reporting, 3-1, 3-7  
    creating page headings, 3-8, 3-9  
    creating subtotals, 3-15  
    formatting, 3-8, 3-9, 3-10  
    totaling data items, 3-12  
running QUERY, 2-1  
REPLACE, 4-6, A-2  
RUN, A-1, A-2

**S**

schema, 1-7  
    password, 1-7, 1-8  
search expression, 3-2, 3-3, D-1  
selecting operating language, E-1  
setname, 3-2  
softkey label, 2-5  
softkeys, 2-4  
sort list, 3-11  
sorting the workfile, 3-10, 3-11  
specifier (unit), C-1  
specifier (volume), C-1  
specifying a data base, 2-2  
specifying a workfile, 3-1  
specifying the output device, 3-7  
spool file, 3-7  
starting QUERY, 2-1  
stopping QUERY, 2-16  
subtotals, 3-10  
subtraction, D-1  
syntax, A-1  
SORT BY, 3-10, 3-11, A-2  
Special Function Keys, 2-4

## T

temporary storage, 3-1  
terminating QUERY, 2-16  
threading data sets, 3-5  
THREAD, 3-5, A-2  
TOTAL, 3-10, 3-12

## U

unit code, C-1  
unit specifier, 2-1, C-1  
using softkeys, 2-4  
using Special Function Keys, 2-4  
using the FIND softkey, 3-4

## V

volume label, 2-1, C-1  
volume specifier, 2-1, C-1  
  unit specifier, 2-1  
  volume label, 2-1

## W

workfile, 3-1  
wrap around, 3-9  
WORKFILE, A-2

## SPECIAL CHARACTERS

-, D-1  
#, D-1  
\*, D-1  
/, D-1  
+, D-1  
<, D-1  
=, D-1  
>, D-1  
[ ], 2-2