6047760[illegible]

CONTROL DATA
CORPORATION

DMS-170

QUERY UPDATE VERSION 3 APPLICATION PROGRAMMING USER'S GUIDE

For Use With:

CYBER Record Manager

CDC® OPERATING SYSTEMS:

NOS 1
NOS 2
NOS/BE 1

REVISION RECORD

Revision	Description
A (04/01/77)	Original release.
B (08/15/78)	Updated to reflect Query Update Version 3.2, PSR level 472.
C (07/23/82)	Released at PSR level 564. This guide reflects Query Update Version 3.4 (or Version 3.3 for the NOS 1 user). The changes include the description of the MODIFY, REMOVE, STORE, and INVOKE directives, which replace the INSERT, UPDATE, DELETE, and USE directives respectively. The changes also include the description of the Query Update language under the NOS 2 operating system. This is a complete reprint.

REVISION LETTERS I, O, Q, AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

or use Comment Sheet in the back of this manual

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision
Front Cover	-
Inside Front Cover	C
Title Page	-
ii	C
iii/iv	C
v	C
vi	C
vii	C
viii	C
1-1	C
2-1 thru 2-10	C
3-1 thru 3-8	C
4-1 thru 4-4	C
5-1 thru 5-5	C
6-1 thru 6-24	C
7-1 thru 7-9	C
8-1 thru 8-6	C
A-1 thru A-4	C
B-1	C
B-2	C
Index-1 thru -3	C
Comment Sheet	C
Mailer	-
Back Cover	-

PREFACE

This user's guide describes the Query Update Version 3.4 language, which is designed for data storage and retrieval operations. Query Update operates under control of the following operating systems:

NOS 1 for the CONTROL DATA® CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; 6000 Computer Systems

NOS 2 for the CDC® CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; 6000 Computer Systems

NOS/BE 1 for the CDC CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; 6000 Computer Systems

The Query Update language supports a wide variety of applications ranging from simple data file query to complex report production. This guide is organized by application; directives are introduced as they become appropriate. Some feature enhancements are not available to the NOS 1 user; these enhancements are noted in the guide.

Sections 1 through 5 discuss file concepts and directives that apply to data access and manipulation. Section 6 details report writing facilities and directives that apply to report catalog maintenance and production. Section 7 presents operations within a multiple-file data base environment. Section 8 illustrates special Query Update utility operations. All data base access is described for use with CYBER Record Manager.

This guide is designed for programmers who are familiar with Control Data standard software products. The Query Update reference manual should be consulted for additional information regarding the Query Update language.

Related material is contained in the following publications. The NOS 1, NOS 2, and NOS/BE 1 manual abstracts are pocket-sized manuals containing brief descriptions of the contents and intended audience of all NOS 1 and NOS 1 product set manuals, NOS 2 and NOS 2 product set manuals, and NOS/BE 1 and NOS/BE 1 product set manuals, respectively. The abstract manuals can be used to determine which manuals are of greatest interest. The Software Publications Release History can be used to determine which revision level of software documentation corresponds to the Programming System Report (PSR) level of installed site software.

The following manuals are of primary interest:

Publication	Publication Number	NOS 1	NOS 2	NOS/BE 1
Query Update Version 3 Reference Manual	60498300	X	X	X
Query Update Version 3 User's Guide For Use With: CYBER Record Manager	60387700	X	X	X

The following manuals are of secondary interest:

Publication	Publication Number	NOS 1	NOS 2	NOS/BE 1
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300	X	X	X
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700	X	X	X
DMS-170 Query Update/CYBER Record Manager Data Administration Reference Manual	60482100	X	X	X
INTERCOM Version 5 Reference Manual	60455010			X
Network Products Interactive Facility Version 1 Reference Manual	60455250	X		

Publication	Publication Number	NOS 1	NOS 2	NOS/BE 1
NOS Version 1 Manual Abstracts	84000420	X		
NOS Version 1 Reference Manual, Volume 1 of 2	60435400	X		
NOS Version 2 Manual Abstracts	60485500		X	
NOS Version 2 Reference Set, Volume 3, System Commands	60459680		X	
NOS/BE Version 1 Manual Abstracts	84000470			X
NOS/BE Version 1 Reference Manual	60498300			X
Software Publications Release History	60481000	X	X	X

CDC manuals can be ordered from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

This guide describes a subset of the features and directives documented in the Query Update Version 3 Reference Manual. Control Data cannot be responsible for the proper functioning of features or directives not documented in the Query Update reference manual.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

QUERY UPDATE AS A PROGRAMMING LANGUAGE 1

Query Update is a programming language that can be used to perform the following data processing operations:

Store and remove data in existing files

Modify data in records in existing files

Sort and display data

Perform arithmetic operations

Generate special-purpose reports

Prepare report formats and preserve them for later use

Create data files

Instructions are input to the Query Update program in statement form. Each statement, called a Query Update directive, specifies one task to be performed. Query Update can be called to perform a single task or a group of tasks that collectively represent one Query Update session.

THE QUERY UPDATE USER

Query Update meets the needs of a variety of users. For example:

A nonprogrammer can access data files to display selected fields of information, perform minor modifications, prepare simple reports, or request previously generated reports with only a few easy-to-learn instructions.

A business programmer can search and manipulate data files through free-form directives rather than through complete program runs, and can construct and preserve various report layouts for subsequent reporting on data files.

A scientific programmer can build individual data files either from scratch or from existing files, and perform any number of calculations.

COMPARING QUERY UPDATE TO COBOL

Query Update is a free-form data processing language. It contains no program divisions and imposes few restrictions on the order in which statements are submitted.

The Query Update language is similar to COBOL, which is also a data processing language, in the following ways:

Both group English terms into sentence-type instructions.

Both use reserved words such as DISPLAY, SORT, EQ, and IF in each instruction.

Both require descriptions of the file organization and storage of data and the characteristics of data.

The terminology used in the descriptions is similar, but the methods for relaying the information to the program and operating system are different.

When Query Update is accessing a permanent data base file, the file organization and data description for the file must be predefined. Predefinition is performed by the data administrator; a program is written in DMS-170 Data Description Language (DDL), it is compiled into a DDL object directory, then it is stored into a permanent file via operating system procedures. The DDL object directory is called a Query Update subschema. One or more compiled Query Update subschemas are stored on a permanent file called a subschema library. The COBOL counterparts of a subschema are the Environment and Data Divisions.

When Query Update is accessing a sequential file that is not part of a permanent data base (this type of file is called a non-data-base file), the subschema concept does not apply. The physical storage characteristics of a non-data-base file are declared in the FILE control statement. The data description is generated within the series of directives that make up a Query Update session.

QUERY UPDATE PROCESSING

Before you consider the actual structure of the Query Update language, you should become familiar with how data is classified and handled within the Query Update data and processing environments.

THE DATA ENVIRONMENT

Query Update handles three distinct categories of data:

Data base file data items

Non-data-base file data items

Temporary data items for operations related to data base or non-data-base files

A data base file is a file whose organization and content are described by a Query Update subschema. As mentioned in the previous section, the subschema must have been written in DDL source language by the data administrator, compiled into a DDL object directory, and stored onto a permanent file (called a subschema library) before the data base can be accessed. A data base file is accessed during a Query Update session through the CREATE or INVOKE directive.

A non-data-base file is a sequential file whose organization and content is not described by a Query Update subschema. A typical non-data-base file might be a transaction file that has been read in from tape for one-time reporting or data base updating purposes. The directory of a non-data-base file must be generated through the DESCRIBE directive before it can be accessed. A non-data-base file is accessed during a Query Update session through the DISPLAY or EXTRACT directives.

A temporary data item is an item that is defined with a DEFINE, DESCRIBE, or SPECIFY directive during a Query Update session. Once a temporary data item is defined, it can be used for comparing data items in a data base or non-data-base file or for modifying data items in a data base file.

QUERY UPDATE LANGUAGE COMPONENTS

Query Update language components include Query Update reserved words, recognized symbols, punctuation, and user-supplied elements (data-names, literals, expressions, conditions, and picture specifications). These components are grouped together into statements for input to the Query Update program. Each complete statement is called a directive. Each directive represents one task to be performed by the computer.

A reserved word is always the first word in a directive and identifies the task to be performed. The keyword can be followed by a number of user-supplied elements as well as additional Query Update keywords. For example, the following directive asks Query Update to make a subschema and its associated data base files available to the tasks that will follow it:

```
INVOKE subschema-name
  FROM LIBRARY permanent-file-name
  (permanent-file-parameters [PW])
```

Operating system parameters are required when permanent files are referenced in a directive.

The following directive describes the first data item in a non-data-base file directory:

```
DESCRIBE non-data-base-file-name
  AND data-name
  AS CHARACTER BY $X(10)$
```

The $ delimiters are required when a non-numeric literal is specified.

The following directive defines a temporary item for use in subsequent tasks:

```
DEFINE data-name AS NUMERIC BY 999
```

The $ delimiters are not required when a numeric literal is specified.

A summary of Query Update directives is shown in table 2-1. The directives are listed in alphabetic order by reserved word, which identifies the purpose of the directive. The comments column provides some rules and default options for each directive. See the Query Update Version 3 Reference Manual for details about the directives.

TABLE 2-1. QUERY UPDATE DIRECTIVES

Directive	Description	Comments
ALTER	Locates retained report directives from the catalog for subsequent modification.	The designated report must have been created by a previous FORMAT directive and must reside on the current catalog.
BREAK	Indicates situations that cause interruption of the body of the report to insert footings and headings; interruption can occur when data name content changes or stated conditions are met.	BREAK is associated with a HEADING and FOOTING directive. Level number 0 cannot be specified.
COMPILE	Stores report specifications in encoded form on a table file.	After the table file has been generated by the COMPILE directive, it is available for input to the Report Utility. The REPORT control statement calls the Report Utility program to produce reports according to specifications in the table file.
CREATE	Establishes a data base file, known as an area, for subsequent insertion of data.	When alternate keys are defined for the area, the INDEX option is required if the index file name is not specified in the subschema. The FROM LIBRARY option is required if the subschema library permanent file name is different from the subschema name. The area and applicable index file must be made permanent if the files are to be saved for future use.
DATE	Specifies use and positioning of system-supplied date information within a report.	Default vertical positioning is line 1. Default horizontal positioning is column 2.
DEFINE	Establishes temporary data names and storage requirements.	Floating point is the default internal format. When the ITEMS option is included, an array is generated, and subscripting is subsequently required. When temporary data items are to be evaluated, the method of evaluation must be specified through the VALUE (or =) option.
DESCRIBE	Establishes a directory to the content of a non-data-base file.	FILLER should be used for those portions of the record that are not to be referenced.
DETAIL	Determines report line content and positioning of source data fields, literals, and computed values.	When tag numbers are included, SELECT directives are required. The AT LINE default is positioned one line beyond the preceding line. The ONCE option can be used only when no more than one numbered DETAIL line directive is specified in the report format. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width or column 1 and the section width if multiple sections are specified.
DIAGNOSTIC	Specifies whether or not consecutive duplicate diagnostic messages are to be displayed.	The default is no display of consecutive duplicate diagnostic messages.

TABLE 2-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	Description	Comments
DISPLAY	Displays information from a data base file, non-data-base file, or temporary storage.	The UPON default is the terminal or the output file. The FROM option requires a directory for the designated file; it must be produced by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive. Query Update rewinds the specified file before displaying FROM.
DUPLICATE	Copies recorded information between the user catalog and the default catalog.	Must be preceded by the VERSION directive, which attaches the user catalog.
END	Terminates Query Update operations and returns control to the operating system.	Default catalog, created area, and report dispositions are determined by the user and performed at this time through applicable operating system procedures.
ERASE	Removes DEFINE items or SPECIFY items. Removes one or more report specifications or session-IDs from the current catalog. Eliminates a directory created by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive.	A single item in a matrix or a literal cannot be erased.
EVALUATE	Performs arithmetic operations to compute data name content or a cumulative function result. Selects the working storage data names for which values are to be calculated when a particular report production step occurs.	The IF directive can be used in conjunction with EVALUATE for manipulative operations, but not for report operations. For report operations, the EVALUATE directive must be preceded by an ALTER or FORMAT directive. If EVALUATE is initializing items to be included as part of a detail line, the directive should be related to the DETAIL directive and not to SELECT.
EXECUTE	Causes execution of a procedure that is external to Query Update.	The procedure name must be 1 through 7 characters in length. The procedure must be in relocatable format.
EXHIBIT	Lists information that is recorded in the current catalog.	Default report specifications, working storage data names, report names with associated report specifications, session IDs with associated directives, relations with associated record names, item names in record, and area names in use can be displayed.
EXTRACT	Creates a subset of information in a data base file, non-data-base file, or temporary storage, and creates a directory to the subset.	Rewind operations before and after an EXTRACT directive ensure correct programming. When data is renamed through the AS option, the data must be referenced by the new name while the EXTRACT is still in effect. Data items can be selected by using the EXTRACT directive with an IF directive. Items in an array must be individually extracted.

TABLE 2-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	Description	Comments
FOOTING	Provides content and determines line and column positioning for informative footings.	A FOOTING directive is associated with a BREAK directive. Level number 0 is associated with the end of data. The AT LINE default is positioned one line beyond the preceding line. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.
FORMAT	Initiates grouping and retention of directives in the catalog under a report name for reference by other directives.	The report name must be unique in the current catalog.
HEADING	Provides content and determines line and column positioning for informative headings.	A HEADING directive is associated with a BREAK directive. Level number 0 occurs before any data is processed. The AT LINE default is positioned one line beyond the preceding line. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.
HELP	Presents descriptions of directives or explanations of diagnostic messages.	The HELP directive can be entered at any time during a Query Update session.
IF	Presents a test condition to determine whether subsequent directives are to be executed or bypassed.	An IF directive that references only temporary data items and literals can be used with any directive. An IF directive controls execution of the directives that follow it in the same transmission, up to but not including the next IF directive. IF directives cannot be nested.
INVOKE	Establishes areas, relations, and the subschema that is used for subsequent directives.	The FROM LIBRARY option identifies the subschema library that contains the subschema directory being used.
MODIFY	Modifies a data item value in an existing record in a data base.	Only one area can be modified at a time; each area joined in a relation must be modified separately. A record is selected for modification either by referencing the record key in the USING option or as a result of an IF directive with an associated MODIFY directive. The SETTING option specifies the names of the data items to be modified.
MOVE	Places values in data names. Places values in data items when a particular report production step occurs.	Defined data items that require evaluation must be evaluated before the MOVE directive is executed. When one record is being updated, MOVE is used in conjunction with the MODIFY directive. For report operations, the MOVE directive must be preceded by a FORMAT or ALTER directive.

TABLE 2-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	Description	Comments
NOTE	Allows user comments to be included in transmissions.	User comments do not appear in output or as part of a report.
OS	Allows the user to enter an operating system command during a Query Update session.	An OS directive can be recorded as part of a session. The directive is not allowed in batch mode.
PAGE-NUMBER	Specifies use and positioning of system-supplied page number within a report.	Positioning default is the rightmost 10 characters of the page, line 1.
PAGE-SIZE	Specifies maximum number of vertical lines, horizontal columns, horizontal or vertical sectional page divisions, and multiple copy images.	Default page length is 60 lines; default page width is 136 columns, 1 section, and 1 image. A maximum of 10 sections can be specified. If sections are specified, the default number of columns divided by the number of sections rounded down is the width of a section. The default PARALLEL option is alternating left and right sides of a composite page when a report page is larger than one printer page.
PERFORM	Retrieves and executes transmissions recorded in the current catalog.	When no options are included, the entire cataloged session is executed.
PREFACE	Causes lines of text or another report to precede the first page of a report.	A specified report name must be in the current catalog. A preface for the specified report name is included in the output report.
PREPARE	Initiates execution of report directives.	The specified report name must be in the current catalog. The source data file is automatically rewound before report preparation. The report output file is not rewound before or after report preparation.
PREVIEW	Causes sample execution of report directives.	The specified report name must be in the current catalog. The FROM default is dummy data values of X's and Y's for character information and 8's and 9's for numeric information.
RECAP	States content and positioning of recapitulative information generated at the end of each report page.	The AT LINE default is positioned one line beyond the last detail or footing line. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.
RECORDING	Initiates the recording of subsequent transmissions in the current catalog.	Recording continues until RECORDING OFF is specified. Report specification directives for a report format are not recorded; they are retained in the current catalog under the report name established by the FORMAT directive.
REMOVE	Removes specific records from a data base file.	Only records from one area can be removed at a time; records from each area joined in a relation must be removed separately. A complete record, not part of a record, is removed.

TABLE 2-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	Description	Comments
RETURN	Releases a file, relation, or subschema that is no longer needed by Query Update.	Multiple files can be returned with one RETURN directive. A file established with a CREATE directive requires two RETURN directives: the first to return the area information, the second to return the file.
REWIND	Logically positions a file at the beginning-of-information.	Multiple files can be rewound with one REWIND directive.
SELECT	Indicates alternative DETAIL specifications are to be selected when stated conditions are met.	When DETAIL directives include tag numbers, SELECT directives are required. The tag number must correspond to a DETAIL tag number. A maximum of five tag numbers can be specified; the number can be modified at installation time. If a MOVE or EVALUATE directive is used for initialization of items to be included as part of a detail line, the directive should be related to the DETAIL directive and not the SELECT directive. When multiple selects exist, only the first qualifying select is executed.
SEPARATOR	Defines a character to be used for delimiting nonnumeric literals.	The dollar sign ($) is the default separator. A character defined as a universal character cannot be defined as a separator. The ITEM-SIZE option is not used in report production.
SORT	Specifies and initiates the resequencing of source data.	Before a file can be sorted, it must have a directory produced by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive. The default collating sequence is COBOL. The default ordering sequence is ascending. A maximum of 25 sorting fields can be defined; each can be defined as either ascending or descending.
SPECIFY	Establishes a name for convenient reference to a condition.	Specified conditions can be tested by IF and SELECT directives.
STORE	Creates a new record and places it in a data base file.	Only one area can be modified at a time; each area joined in a relation must be modified separately. The SETTING option specifies the names of the data items to be stored.
SUMMARY	Causes lines of text or another report to follow the last page of a report.	A specified report name must be in the current catalog. A summary for the specified report name is included in the output report.
TABS	Relates tabular references to horizontal column numbers for report preparation.	Tab numbers need not be entered in sequence because the system sorts them in ascending order.
TIME	Specifies use and positioning of system-supplied time information within a report.	Default vertical positioning is line 1; default horizontal positioning is column 90. If the page size is less than 99 columns, the default horizontal positioning is the page width minus 29 columns.

TABLE 2-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	Description	Comments
TITLE	States content and positioning of title to start a report page.	Default vertical positioning is line 1; default horizontal positioning is column 1. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.
UNIVERSAL	Establishes a character that marks a character position to be ignored during comparison testing.	The default universal character is # in the ASCII character set and ≡ in the CDC graphic set. The current separator character cannot be defined as the universal character.
VERIFY	Specifies data names for terminal display in response to a previous VETO directive.	The specified data items are displayed in response to a directive that causes a record to be modified or deleted. The directive is not allowed in batch mode.
VERSION	Attaches a permanent file as the current catalog or reverts to the default catalog.	Permanent file parameters are required unless the subschema declares the area as TEMP.
VETO	Causes a terminal display of data subject to modification or deletion.	The directive is not allowed in batch mode.
VIA	Specifies which relation should be followed. VIA is needed when more than one relation is defined and the relation to be used cannot be determined by Query Update.	If more than one VIA directive is entered in a transmission, only the last one specified is used. A VIA directive remains in effect until another VIA directive is entered.

USING QUERY UPDATE

Query Update functions in interactive mode through a user terminal. Query Update asks for input, the user enters directives, then Query Update responds. The results of the operation requested by the directives are displayed or printed on output devices. Output devices can include a terminal display screen, terminal hardcopy printer, line printer, or any combination of the three.

One or more directives can be used to specify a complete operation. A complete operation is called a transmission. A transmission begins with a directive keyword and ends when a SEND, RETURN, or equivalent transmit key is depressed. Query Update executes or records the transmission and then returns one or more of the following:

The information requested by the transmission

An error message

A double hyphen (--), which indicates that Query Update is ready to receive the next transmission

A double greater-than symbol (>>), which indicates that Query Update is ready to receive data

A physical transmission (a transmission as defined by the operating system) can contain a maximum of 150 characters on NOS and can contain an unlimited number of characters on NOS/BE. A logical transmission (a transmission as defined by Query Update) can contain a maximum of 1030 characters. The maximum logical transmission length can be changed by setting the TL parameter on the Query Update control statement.

When a logical transmission exceeds the line width of the terminal, the system performs an automatic carriage return and line-feed, then continues to accept input on the next line.

Automatic continuation can be more convenient to use than manual continuation. Under automatic continuation, a user can backspace to the beginning of a physical transmission to correct a typing error.

Transmissions can be manually continued by entering one of two continuation characters (+ or =) as the last character on the line and depressing the transmit key. A manually continued transmission can include a maximum of 1030 characters or the number of characters specified by the TL parameter. At least one space is required between the last nonblank character and the continuation character.

Interactive mode is considered to be the most convenient method for using Query Update. Three exceptions might be the following:

When a large data base is being created.

When a long report format is being entered and terminal hardcopy output is not available.

When a long sequence of transmissions must be entered in the session.

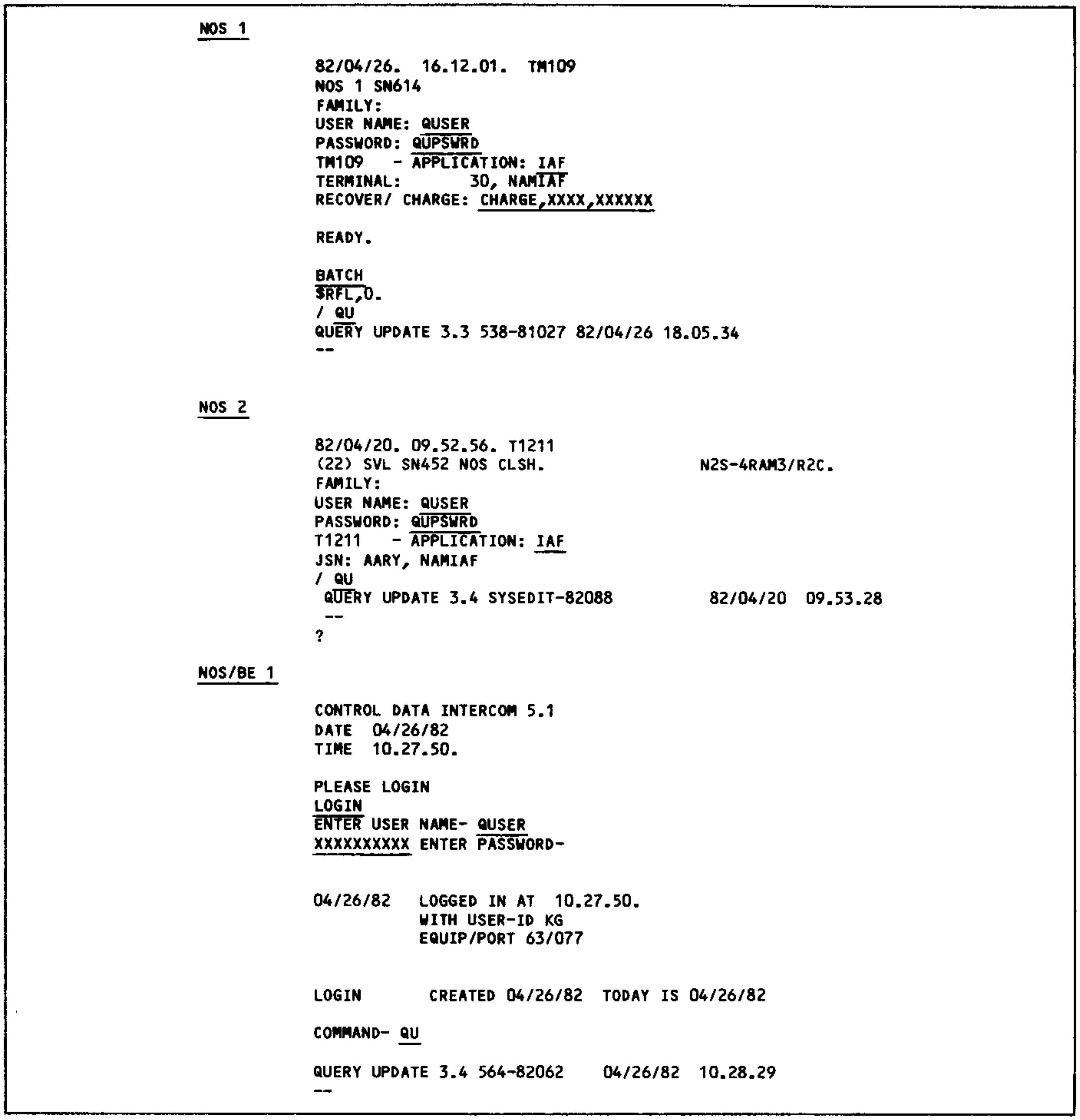
Beginning a Query Update Session

Before beginning a Query Update session you must establish communication between the terminal and the host computer.

Hardware communication is established by depressing the switches necessary to turn on the remote terminal equipment, dialing into the host computer, and completing the connection between the terminal and the computer.

Software communication is established by a login procedure. The login procedure is a dialog that is required for user identification and accounting purposes. Login procedures for NOS and NOS/BE are shown in figure 2-1. User input is underscored.

A Query Update session is begun when you enter a QU control statement.



```
NOS 1

            82/04/26.  16.12.01.  TM109
            NOS 1 SN614
            FAMILY:
            USER NAME: QUSER
            PASSWORD: QUPSWRD
            TM109    - APPLICATION: IAF
            TERMINAL:       30, NAMIAF
            RECOVER/ CHARGE: CHARGE,XXXX,XXXXXX

            READY.

            BATCH
            $RFL,0.
            / QU
            QUERY UPDATE 3.3 538-81027 82/04/26 18.05.34
            --


NOS 2

            82/04/20. 09.52.56. T1211
            (22) SVL SN452 NOS CLSH.                N2S-4RAM3/R2C.
            FAMILY:
            USER NAME: QUSER
            PASSWORD: QUPSWRD
            T1211    - APPLICATION: IAF
            JSN: AARY, NAMIAF
            / QU
             QUERY UPDATE 3.4 SYSEDIT-82088          82/04/20  09.53.28
             --
            ?

NOS/BE 1

            CONTROL DATA INTERCOM 5.1
            DATE  04/26/82
            TIME  10.27.50.

            PLEASE LOGIN
            LOGIN
            ENTER USER NAME- QUSER
            XXXXXXXXXX ENTER PASSWORD-


            04/26/82   LOGGED IN AT  10.27.50.
                       WITH USER-ID KG
                       EQUIP/PORT 63/077


            LOGIN       CREATED 04/26/82  TODAY IS 04/26/82

            COMMAND- QU

            QUERY UPDATE 3.4 564-82062    04/26/82  10.28.29
            --
```

Figure 2-1. Login Procedures

Ending a Query Update Session

A Query Update session is ended when you enter an END directive.

Software communication is terminated by a logout procedure. The logout procedure is a single user command. Logout procedures for NOS and NOS/BE are shown in figure 2-2. User input is underscored.

```
NOS

      --
    ? END

    / BYE

NOS/BE

      --
      END

    COMMAND- LOGOUT
```

Figure 2-2. Logout Procedures

Hardware communications are terminated by depressing the switches necessary to turn off the remote terminal equipment and releasing the telephone connection between the terminal and the computer.

THE PROCESSING ENVIRONMENT

The Query Update processing environment using CYBER Record Manager (CRM) varies depending on the type of file (data base or non-data-base) being accessed. The basic requirement for either type of file access is the existence of a directory that describes the physical storage and characteristics of individual data items. Query Update processing is shown in figure 2-3.

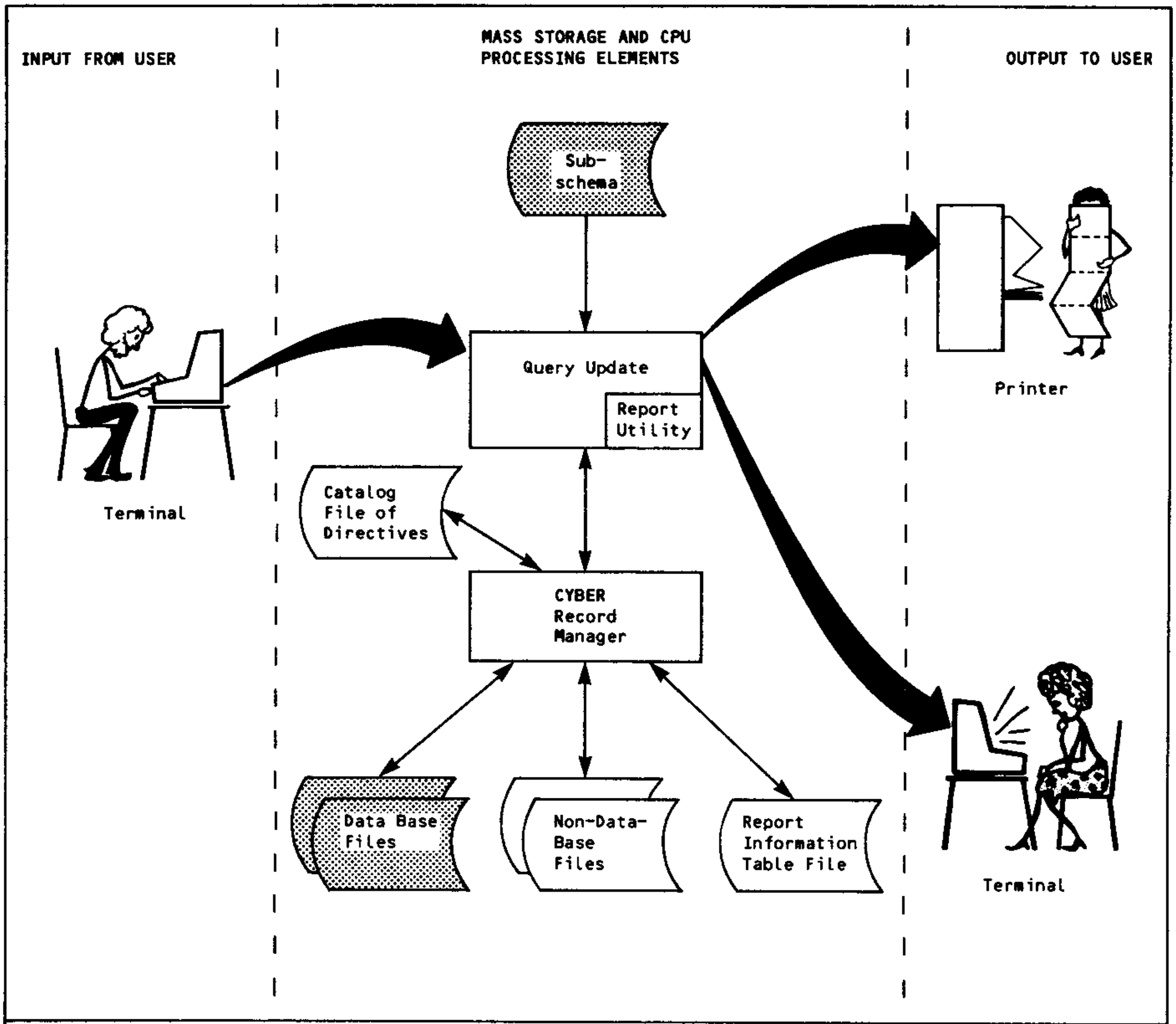
The left portion shows user input to the Query Update program via a user terminal. Input consists of Query Update directives and data entered as a series of transmissions.

The center portion shows the Query Update program and how it interfaces with the Report Utility and CRM.

The Report Utility program is callable through the REPORT control statement of the operating system. This utility automates report writing. The report information table file contains encoded report specifications that are generated by the COMPILE directive of the Query Update language and used by the Report utility. The catalog of directives is a series of recorded report specifications that have been generated and stored in mass storage for subsequent use.

CRM performs input/output processing for data base files according to the information passed through the subschema. CRM performs input/output processing for non-data-base files according to information provided by the user.

The right portion shows output to the user from the Query Update program to a line printer or terminal display screen.



Note:

Shaded portions apply to a data base file described by a previously compiled subschema directory.

Nonshaded portions apply to operations on both data base and non-data-base files.

Figure 2-3. Query Update Processing

DATA BASE FILE OPERATIONS 3

Every data base file accessed through the Query Update language must be described in a directory called a Query Update subschema. The subschema declares the organization of the file, indicates the physical storage of the data, defines data names, and describes the characteristics of each data item.

There are two types of Query Update subschemas: one in which a data base is defined by a subschema and controlled by CYBER Database Control System (CDCS), and one in which a data base is defined by a subschema and controlled by CYBER Record Manager (CRM). This guide only deals with Query Update in CRM data base access mode.

You can access an existing data base file after you generate and store a Query Update subschema that corresponds to the file. You can, alternatively, generate and store a Query Update subschema first, and then create a file to match the description set up in the subschema. This latter approach is taken in the following paragraphs.

GENERATING A QUERY UPDATE SUBSCHEMA

The generation of Query Update subschemas is normally performed under the direction of a data administrator. This conforms to the DMS-170 data base management concept that data base information should be centrally controlled. Detailed information regarding Query Update subschema generation is contained in the Query Update/CYBER Record Manager Data Administration reference manual.

The NOS job structure and DDL source statements required to generate a Query Update subschema are shown in figure 3-1.

The subschema contains the following divisions:

> The Identification Division, which names the subschema.
>
> The Data Division, which names the file, supplies file organization and key specification for CRM and describes the record.

The subschema is named QUSUB.

The data base file is named INVENTORY. Notice that the subschema refers to the file as an area. Several areas could have been included in this subschema, but only one was selected.

File organization is new indexed sequential (which means extended indexed sequential). The primary key is INV-NO, the alternate key is BACK-ORDER.

The index file is named INVIDX. The INDEX statement is optional; when it is not included in the subschema, the index for files with alternate keys must be declared at Query Update execution time.

```
Job statement
USER control statement
CHARGE control statement
DEFINE(QUSUB/CT=PU,M=R)
DDL3(QD,SB=QUSUB)
End-of-record
        IDENTIFICATION DIVISION.
        SUB-SCHEMA NAME IS QUSUB
        DATA DIVISION.
        AREA-NAME IS INVENTORY UN IS username M IS W
        INDEX IS INVIDX UN IS username M IS W
        ORGANIZATION IS INDEXED NEW
        KEY IS INV-NO
            KEY IS ALTERNATE BACK-ORDER DUPLICATES INDEXED
        RECORD-NAME IS INV-REC
            02 INV-NO PICTURE X(6)
            02 IN-STOCK PICTURE Z(3)9
            02 BACK-ORDER PICTURE Z(3)9
            02 ON-ORDER PICTURE Z(3)9
            02 REORDER-PT PICTURE Z(3)9
            02 UNIT-COST PICTURE Z(4).99
            02 UNIT-PRICE PICTURE Z(4).99
            02 DESCRIPTION PICTURE X(17)
End-of-information
```

Figure 3-1. Generating a Subschema Under NOS

The record is named INV-REC and has eight fields, which are alphanumeric, as indicated by the X descriptor. The remaining fields are numeric. Because reports are to be generated from the INVENTORY file, editing characters are included in appropriate picture specifications.

The DEFINE control statement effects permanent file storage. The file is declared public by the CT parameter and the file access mode is declared read-only by the M parameter. (The file access mode only affects users who have a different user number.)

The DDL control statement specifies two required parameters: QD specifies that a Query Update subschema is to be compiled and SB=QUSUB denotes the file name upon which the subschema is to be written.

The NOS/BE job structure and DDL statements required to generate a subschema are shown in figure 3-2.

The REQUEST and CATALOG control statements are required for permanent file storage of the subschema. The local file name is X and the permanent file name is QUSUB.

The DDL control statement specifies two required parameters: QD denotes that a Query Update subschema is to be compiled and SB=X denotes the local file name upon which the subschema is to be written.

CREATING A DATA BASE FILE

Now that a subschema exists, a data base file named INVENTORY can be created through the CREATE directive. The use of this directive is shown on NOS in figure 3-3.

The DEFINE control statement effects permanent file storage of the INVENTORY file and its associated index file; the M parameter provides user write permission mode. The QU control statement calls the Query Update program.

The hyphens (--) indicate that Query Update is expecting a directive.

The CREATE directive performs the following functions:

Names the data base file (INVENTORY)

Names the index file (INVIDX) (required when alternate keys are specified in the subschema and when the index file is not specified in the subschema)

Names the file name where the subschema is located

The STORE directive is required when data is to be stored in the file. If this directive had not been included, the INVENTORY file would have been created, but would be empty; data could be entered at a later time. The SETTING option allows multiple data items in a series of records to be inserted. Notice the continuation symbol (+). The space that appears between ON-ORDER and the + is required. The two greater-than symbols (>>) indicate that the program is expecting data.

```
Job statement
ACCOUNT control statement
REQUEST(X,*PF)
DDL3(QD,SB=X)
CATALOG(X,QUSUB,ID=QUSER)
End-of-record
        IDENTIFICATION DIVISION.
        SUB-SCHEMA NAME IS QUSUB
        DATA DIVISION.
        AREA-NAME IS INVENTORY ID IS QUSER
        INDEX IS INVIDX ID IS QUSER
        ORGANIZATION IS INDEXED NEW
        KEY IS INV-NO
            KEY IS ALTERNATE BACK-ORDER DUPLICATES INDEXED
        RECORD-NAME IS INV-REC
            02 INV-NO PICTURE X(6)
            02 IN-STOCK PICTURE Z(3)9
            02 BACK-ORDER PICTURE Z(3)9
            02 ON-ORDER PICTURE Z(3)9
            02 REORDER-PT PICTURE Z(3)9
            02 UNIT-COST PICTURE Z(4).99
            02 UNIT-PRICE PICTURE Z(4).99
            02 DESCRIPTION PICTURE X(17)
End-of-information
```

Figure 3-2. Generating a Subschema Under NOS/BE

```
/ DEFINE(INVENTO,INVIDX/CT=PU,M=W)
/ QU

 QUERY UPDATE 3.4 SYSEDIT-82110          82/04/27  12.23.30

 --
? CREATE INVENTORY INDEX IS INVIDX OF QUSUB (UN=username)

 --
? STORE SETTING INV-NO IN-STOCK BACK-ORDER ON-ORDER +
? REORDER-PT UNIT-COST UNIT-PRICE DESCRIPTION
 >> $AB5972$ 5 3 9 5 175.79 389.95 $METAL DESK$
 >> $AB5973$ 2 0 0 1 500.00 1282.50 $OAK DESK$
 >> $AB5975$ 2 1 3 1 900.00 1300.00 $WALNUT DESK$
 >> $BB0013$ 10 0 8 5 5.00 15.00 $BULLETIN BOARD$
 >> $CB0168$ 8 0 0 5 8.00 19.52 $CHALK BOARD$
 >> $CB1001$ 10 0 0 5 15.00 45.00 $1-DR FILE CABINET$
 >> $CB1003$ 10 0 0 5 20.00 60.00 $3-DR FILE CABINET$
 >> $CB1005$ 10 0 0 5 32.00 90.00 $5-DR FILE CABINET$
 >> $CH0059$ 7 1 6 3 130.00 295.00 $ARM CHAIR$
 >> $CH0060$ 5 0 0 3 89.00 149.95 $DESK CHAIR$
 >> $CH0080$ 9 0 0 3 35.00 96.00 $SWIVEL CHAIR$
 >> $CH0575$ 50 0 6 10 .98 3.98 $LETTER RACK$
 >> $SH0011$ 10 0 3 5 20.00 39.95 $3-SHELF BOOK CASE$
 >> $ST0592$ 7 0 0 3 9.50 16.20 $STOOL$
 >> $TY5015$ 2 0 0 1 329.00 369.00 $ELECT TYPEWRITER$
 >> $XM6158$ 10 0 0 2 38.00 95.00 $COFFEE TABLE$
 >> $YB0020$ 500 0 0 20 5.50 19.95 $DESK LAMP$
 >> $YB0059$ 25 0 0 20 18.50 69.95 $FLOOR LAMP$
 >> $YB0060$ 20 0 4 5 13.30 39.95 $TABLE LAMP$
 >> *END
     19 ACCESSES, 19 HITS, 19 IO-S

 --
? END
```

Figure 3-3. Creating a Data Base File Under NOS

Each data entry includes an item for each field specified in the STORE directive. Data is entered in the order indicated by the data names in the SETTING option; INV-NO data is first, IN-STOCK data is second, and so forth. Data fields are separated from each other by at least one space (a comma could be used instead of a space). Several additional points concerning the data entries should be noted:

The INV-NO and DESCRIPTION fields are described as alphanumeric in the subschema; therefore, they must be enclosed in $ delimiters. These fields are stored with the appropriate number of trailing blanks.

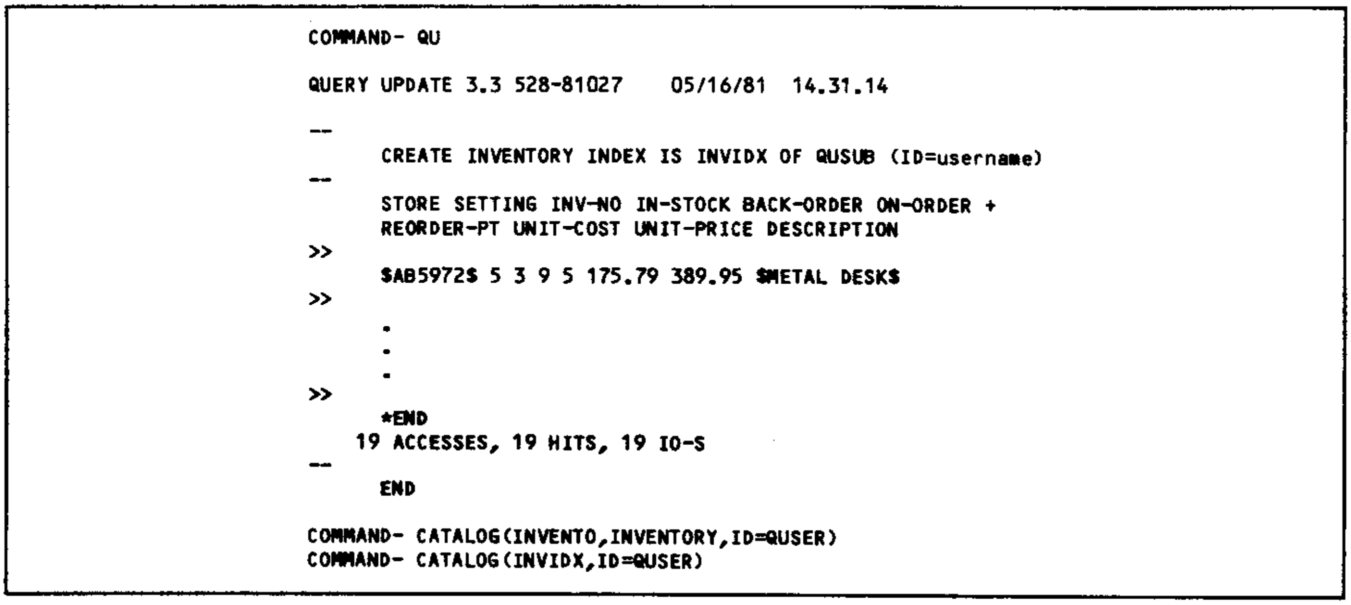
The numeric fields do not require $ delimiters. These fields are stored with the appropriate number of leading zeros. The Z replacement character in the subschema causes suppression of leading zeros when the fields are output in display mode.

The data entries are followed by an *END directive, which terminates the STORE directive.

The last entry is the END directive, which terminates the Query Update session.

The INVENTORY data base file can be created under NOS/BE with the job stream shown in figure 3-4.

Two CATALOG control statements effect permanent file storage of the INVENTORY file and its associated index file.



```
COMMAND- QU

QUERY UPDATE 3.3 528-81027    05/16/81  14.31.14

--
      CREATE INVENTORY INDEX IS INVIDX OF QUSUB (ID=username)
--
      STORE SETTING INV-NO IN-STOCK BACK-ORDER ON-ORDER +
      REORDER-PT UNIT-COST UNIT-PRICE DESCRIPTION
>>
      $AB5972$ 5 3 9 5 175.79 389.95 $METAL DESK$
>>
      .
      .
      .
>>
      *END
    19 ACCESSES, 19 HITS, 19 IO-S
--
      END

COMMAND- CATALOG(INVENTO,INVENTORY,ID=QUSER)
COMMAND- CATALOG(INVIDX,ID=QUSER)
```

Figure 3-4. Creating a Data Base File Under NOS/BE

QUERYING A DATA BASE FILE

Interactive query always begins with a call to the Query Update program. This is done by typing the letters QU in response to the operating system request for input. NOS requests input by printing a slash (/). NOS/BE requests input by printing the word COMMAND-.

When the Query Update program is loaded into memory, a Query Update heading is printed followed by a request for input. Query Update requests input by printing double hyphens (--) followed by a linefeed. On NOS, Query Update also prints a question mark (?).

A data base file must be attached before it can be accessed for interactive query. This is done through the INVOKE directive, which names the subschema and the subschema library (if the subschema library name is different from the subschema name) and provides user identification. Query Update automatically attaches the data base file, together with its associated index file, when it is needed and returns the files when they are not needed.

Sample directives for interactive query are shown in figure 3-5. The following points should be noted:

> The INVOKE directive references permanent files. Certain parameters are required as shown in the figure.
>
> Query Update displays only 14 lines at one time. After each group of lines, the program pauses and offers the option to continue the display or terminate the directive.
>
> The IF directive can be used in conjunction with the DISPLAY directive to provide conditional query.

MODIFYING A DATA BASE FILE

Modifications to a data base file are made by three Query Update directives: MODIFY, STORE, and REMOVE. Specific operations and general directive formats are listed as follows:

Modify like fields in several records.

```
MODIFY USING key-name
    MOVE expression TO data-name-1
>> key-name-value
```

Store values in one or more fields for one new record.

```
STORE MOVE expression-1 TO data-name-1
    AND expression-2 TO data-name-2
```

Store values for like fields in several new records.

```
STORE SETTING
    data-name-1 data-name-2 data-name-3
>> data-name-1-value data-name-2-value
    data-name-3-value
```

Remove one or more records.

```
REMOVE USING key-name
>> key-name-value
```

```
/ QU                                                          Call Query Update.
 QUERY UPDATE 3.4 SYSEDIT-82110        82/04/29  13.05.46
 --
? INVOKE QUSUB (UN=username)                                  Name the subschema.  NOS/BE
 --                                                           requires an ID parameter
? DISPLAY INV-NO DESCRIPTION IN-STOCK +                       instead of a UN parameter.
? REORDER-PT UNIT-COST UNIT-PRICE
                                                              Display six fields.
 AB5972 METAL DESKS          5    5  175.79  389.95
 AB5973 OAK DESKS            2    1  500.00 1282.50
 AB5975 WALNUT DESK          2    1  900.00 1300.00
 BB0013 BULLETIN BOARD      10    5    5.00   15.00
 CB0168 CHALK BOARD          8    5    8.00   19.52
 CB1001 1-DR FILE CABINET   10    5   15.00   45.00
 CB1003 3-DR FILE CABINET   10    5   20.00   60.00
 CB1005 5-DR FILE CABINET   10    5   32.00   90.00
 CH0059 ARM CHAIR            7    3  130.00  295.00
 CH0060 DESK CHAIR           5    3   89.00  149.95
 CH0080 SWIVEL CHAIR         9    3   35.00   96.00
 CH0575 LETTER RACK         50   10     .98    3.98
 SH0011 3-SHELF BOOK CASE   10    5   20.00   39.95
 ST0592 STOOL                7    3    9.50   16.20
 (MORE... ANSWER Y OR N)
? Y                                                           A Y response continues the
 TY5015 ELECT TYPEWRITER     2    1  329.00  369.00           display.
 XN6158 COFFEE TABLE        10    2   38.00   95.00
 YB0020 DESK LAMP          500   20    5.50   19.95
 YB0059 FLOOR LAMP          25   20   18.50   69.95
 YB0060 TABLE LAMP          20    5   13.30   39.95
       19 ACCESSES, 19 HITS, 19 IO-S

 --
? IF INV-NO EQ $YB0060$ DISPLAY INV-REC                       Request a conditional
                                                              display by primary key.
 (209) REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT            The primary key is alphanu-
 YB006000200000000400050013300003995TABLE LAMP                meric and requires the $'s.
       1 ACCESSES, 1 HITS, 1 IO-S                             Full record display is never
                                                              in display format.
 --
? IF UNIT-PRICE LE 40 DISPLAY DESCRIPTION +                   Request a conditional display
? UNIT-PRICE                                                  using the relational operator
 BULLETIN BOARD       15.00                                   less than or equal to.
 CHALK BOARD          19.52
 LETTER RACK           3.98
 3-SHELF BOOK CASE    39.95
 STOOL                16.20
 DESK LAMP            19.95
 TABLE LAMP           39.95
       19 ACCESSES, 7 HITS, 19 IO-S

 --
? IF IN-STOCK GE 5 DISPLAY DESCRIPTION +                      Request a conditional display
? IN-STOCK                                                    using the relational operator
 METAL DESKS           5                                      greater than or equal to.
 BULLETIN BOARD       10
 CHALK BOARD           8
 1-DR FILE CABINET    10
 3-DR FILE CABINET    10
 5-DR FILE CABINET    10
 ARM CHAIR             7
 DESK CHAIR            5
 SWIVEL CHAIR          9
 LETTER RACK          50
 3-SHELF BOOK CASE    10
 STOOL                 7
 COFFEE TABLE         10
 DESK LAMP           500
 (MORE... ANSWER Y OR N)
? N                                                           An N response terminates the
       18 ACCESSES, 14 HITS, 18 IO-S                          display.
```

Figure 3-5. Querying a Data Base File (Sheet 1 of 2)

```
 --
? DISPLAY DESCRIPTION $$$$ UNIT-PRICE                                      Display the $ literal.  The
 METAL DESKS       $  389.95                                               $ is the separator character
 OAK DESKS         $ 1282.50                                               and must be specified twice
 WALNUT DESK       $ 1300.00                                               for each occurrence.
 BULLETIN BOARD    $   15.00
 CHALK BOARD       $   19.52
 1-DR FILE CABINET $   45.00
 3-DR FILE CABINET $   60.00
 5-DR FILE CABINET $   90.00
 ARM CHAIR         $  295.00
 DESK CHAIR        $  149.95
 SWIVEL CHAIR      $   96.00
 LETTER RACK       $    3.98
 3-SHELF BOOK CASE $   39.95
 STOOL             $   16.20
 (MORE... ANSWER Y OR N)
? Y
 ELECT TYPEWRITER  $  369.00
 COFFEE TABLE      $   95.00
 DESK LAMP         $   19.95
 FLOOR LAMP        $   69.95
 TABLE LAMP        $   39.95
       19 ACCESSES, 19 HITS, 19 IO-S

 --
? IF REORDER-PT = 3 AND UNIT-COST NE 130 +                                 Request a conditional dis-
? DISPLAY DESCRIPTION REORDER-PT UNIT-COST                                 play using the logical
 DESK CHAIR            3   89.00                                           operator AND.
 SWIVEL CHAIR          3   35.00
 STOOL                 3    9.50
       19 ACCESSES, 3 HITS, 19 IO-S

 --
? IF REORDER-PT = 3 OR UNIT-COST = 89 +                                    Request a conditional dis-
? DISPLAY DESCRIPTION REORDER-PT UNIT-COST                                 play using the logical
 ARM CHAIR             3  130.00                                           operator OR.
 DESK CHAIR            3   89.00
 SWIVEL CHAIR          3   35.00
 STOOL                 3    9.50
       19 ACCESSES, 4 HITS, 19 IO-S

 --
? DISPLAY (UNIT-PRICE - UNIT-COST) +                                       Request the display of an
? / UNIT-COST * 100                                                        arithmetic expression.
  000000000121.83                                                          Default display is floating-
  000000000156.50                                                          point format.
  000000000044.44
  000000000200.00
  000000000144.00
  000000000200.00
  000000000200.00
  000000000181.25
  000000000126.92
  000000000068.48
  000000000174.29
  000000000306.12
  000000000099.75
  000000000070.53
 (MORE... ANSWER Y OR N)
? N
       15 ACCESSES, 14 HITS, 15 IO-S

 --
? END                                                                      Terminate Query Update.
```

Figure 3-5. Querying a Data Base File (Sheet 2 of 2)

Sample directives for data base file modification are shown in figure 3-6. The following points should be noted:

> The REMOVE directive deletes an entire record, not part of a record.
>
> The IF directive can be used in conjunction with the MODIFY and REMOVE directives to provide conditional modification.

The ACCESS message indicates the total number of records accessed for modification; the HITS message indicates the total number of successful operations; the IO-S message indicates the total number of input/output operations required.

```
/ QU                                                        Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110        82/04/29  13.05.46

 --
? INVOKE QUSUB (UN=username)                                Name the subschema.  NOS/BE requires an
                                                            ID parameter instead of a UN parameter.
 --
? MODIFY USING INV-NO SETTING ON-ORDER                      Modify one field in three records.

 >> $AB5972$ 12

 >> $AB5973$ 7

 >> $AB5975$ 12

 >> *END                                                    Terminate the MODIFY USING directive.
       3 ACCESSES, 3 HITS, 6 IO-S

 --
? IF INV-NO EQ $SH0011$ MODIFY MOVE 15 TO IN-STOCK          Request a conditional modification.
       1 ACCESSES, 1 HITS, 3 IO-S

 --
? STORE SETTING INV-NO IN-STOCK BACK-ORDER ON-ORDER REORDER-PT +   Store three records.
? UNIT-COST UNIT-PRICE DESCRIPTION

 >> $CH0100$ 5 0 2 2 52.00 80.00 $TYPING CHAIR$

 >> $CB1002$ 10 0 0 5 22.00 40.00 $2-DR FILE CABINET$

 >> $CB1004$ 10 0 0 5 37.00 75.00 $4-DR FILE CABINET$

 >> *END                                                    Terminate the STORE SETTING directive.
       3 ACCESSES, 3 HITS, 3 IO-S

 --
? REMOVE USING INV-NO                                       Remove three records.

 >> $YB0059$

 >> $CB1005$

 >> $CB1001$

 >> *END                                                    Terminate the REMOVE USING directive.
       3 ACCESSES, 3 HITS, 6 IO-S

 --
? IF INV-NO EQ $XN6158$ REMOVE VETO                         The VETO directive displays the first
 XN61580010000000000002003800009500COFFEE                   40 characters.
 VETO- NO                                                   A NO response cancels the remove re-
       1 ACCESSES, 1 HITS, 2 IO-S                           quest.
```

Figure 3-6. Modifying a Data Base File (Sheet 1 of 2)

```
 --
? VERIFY DESCRIPTION
```

The VERIFY directive requests display of the description field, rather than the first 40 characters.

```
 --
? REMOVE USING INV-NO VETO

 >> $ST0592$

 STOOL
 VETO- YES
```

A YES response effects the remove operation.

```
 >> *END
       1 ACCESSES, 1 HITS, 2 IO-S

 --
? IF INV-NO EQ $XN6158$ REMOVE VETO
```

Request a conditional remove with the VERIFY option in effect.

```
 COFFEE TABLE
 VETO- NO
       1 ACCESSES, 1 HITS, 2 IO-S

 --
? DISPLAY INV-NO DESCRIPTION
```

A DISPLAY operation shows that three records have been stored (CB1002, CB1004, and CH0100) and four records have been removed (CB1001, CB1005, ST0592, and YB0059).

```
 AB5972 METAL DESK
 AB5973 OAK DESK
 AB5975 WALNUT DESK
 BB0013 BULLETIN BOARD
 CB0168 CHALK BOARD
 CB1002 2-DR FILE CABINET
 CB1003 3-DR FILE CABINET
 CB1004 4-DR FILE CABINET
 CH0059 ARM CHAIR
 CH0060 DESK CHAIR
 CH0080 SWIVEL CHAIR
 CH0100 TYPING CHAIR
 CH0575 LETTER RACK
 SH0011 3-SHELF BOOK CASE
 (MORE... ANSWER Y OR N)
? Y
 TY5015 ELECT TYPEWRITER
 XN6158 COFFEE TABLE
 YB0020 DESK LAMP
 YB0060 TABLE LAMP
       18 ACCESSES, 18 HITS, 18 IO-S

 --
? END
```

Terminate Query Update.

Figure 3-6. Modifying a Data Base File (Sheet 2 of 2)

A non-data-base file, as defined within the context of this guide, is a sequential file that has no corresponding Query Update subschema. When a non-data-base file is to be accessed by Query Update, the following requirements must be met:

The file must be a local file. If the file is a permanent file, it must be attached via the operating system permanent file attach procedure.

The storage characteristics of the file must be conveyed to CYBER Record Manager (CRM) with a FILE control statement.

The file must be described for Query Update with the DESCRIBE directive.

For purposes of illustration, a non-data-base file has been stored on disk. The layout of the file is shown in figure 4-1. The file has the following characteristics:

The file name is INDFILE.

The file contains six records.

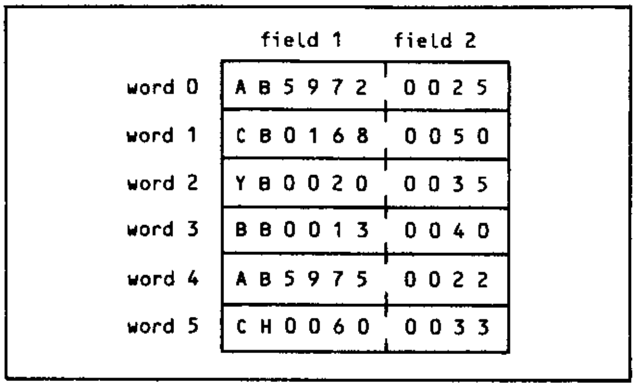
Each record contains two fields.

Field 1 contains six alphanumeric characters and field 2 contains four numeric characters.

This file is a transaction file related to the data base file INVENTORY. INDFILE includes values that correspond to values within the primary key field of INVENTORY.

DESCRIBING A NON-DATA-BASE FILE

Query Update accesses a non-data-base file according to the rules specified in a directory. Unlike a data base file subschema directory that is generated by DDL, a non-data-base file directory is generated by the Query Update DESCRIBE directive.



	field 1	field 2
word 0	A B 5 9 7 2	0 0 2 5
word 1	C B 0 1 6 8	0 0 5 0
word 2	Y B 0 0 2 0	0 0 3 5
word 3	B B 0 0 1 3	0 0 4 0
word 4	A B 5 9 7 5	0 0 2 2
word 5	C H 0 0 6 0	0 0 3 3

Figure 4-1. INDFILE Format

The operating system control statements and Query Update directive required to establish access to the non-data-base file INDFILE are shown in figure 4-2. The ATTACH and FILE control statements are required; they can precede the QU control statement and be entered in the usual manner or they can be entered through the Query Update OS directive. Only the BT and RT parameters are included on the FILE control statement; default values for other required parameters apply.

The DESCRIBE directive includes the file name and assigns names for the two fields (IND-INV-NO and IND-ON-ORDER). The fields correspond to the INVENTORY file fields INV-NO and ON-ORDER. Data item IND-INV-NO is described as a six-character alphanumeric field and IND-ON-ORDER is described as a four-character zero-suppressed numeric field. The $ delimiters are required because of the non-numeric characters X Z) (.

```
/ QU                                                            Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110          82/05/03  09.26.41

 --
? OS,ATTACH,INDFILE                                             Issue an operating system ATTACH com-
                                                                mand for INDFILE.
 --
? OS,FILE,INDFILE BT=C,RT=F                                     Issue an operating system FILE command
                                                                to declare INDFILE file organization.
 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +           Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
```

Figure 4-2. Describing a Non-Data-Base File

QUERYING A NON-DATA-BASE FILE

You can query a non-data-base file after it has been attached, defined for CRM, and described for Query Update. Sample display operations are shown in figure 4-3.

The following points should be noted for non-data-base file query operations:

When the OS directives reference permanent files, certain parameters are required, as shown in the figure.

The word FROM and the file name must be included in the DISPLAY directive.

The file is always automatically rewound before a DISPLAY FROM directive is executed.

Conditional testing with the IF directive is not allowed.

ENTERING DATA IN A NON-DATA-BASE FILE

Data can be placed in a non-data-base file by using the DISPLAY UPON format of the DISPLAY directive. Data is entered and displayed upon a non-data-base file rather than upon the terminal display screen. Data entered with the DISPLAY UPON directive under NOS is retained in a direct access file automatically, whereas data entered under NOS/BE with the directive must be retained with permanent file operations.

Sample DISPLAY UPON operations are shown in figure 4-4 for NOS and in figure 4-5 for NOS/BE. The following points should be noted for the DISPLAY UPON directive:

The non-data-base file must be positioned at the end of information before the DISPLAY UPON directive is issued. Figure 4-4 shows the DISPLAY FROM directive positioning the file; figure 4-5 shows the OS directive positioning the file. The DISPLAY FROM directive for positioning purposes is practical only when the file is small.

The DISPLAY UPON directive eliminates any previously established directory and creates a directory for the file being displayed (directory not created on NOS 1).

The SEPARATOR (SEP ITEM-SIZE) directive tells Query Update to concatenate the fields in INDFILE. If SEP ITEM-SIZE had not been specified, Query Update would have written one blank character before each item.

```
 --
? DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER          Display the INDFILE fields.
 AB5972   25
 CB0168   50
 YB0020   35
 BB0013   40
 AB5975   22
 CH0060   33
       6 ACCESSES, 6 HITS, 6 IO-S

 --
? DISPLAY FROM INDFILE $INVENTORY NUMBER IS$ +          Include nonnumeric literals in a display.
? IND-INV-NO $AND ORDER AMOUNT IS$ IND-ON-ORDER
 INVENTORY NUMBER IS AB5972 AND ORDER AMOUNT IS   25
 INVENTORY NUMBER IS CB0168 AND ORDER AMOUNT IS   50
 INVENTORY NUMBER IS YB0020 AND ORDER AMOUNT IS   35
 INVENTORY NUMBER IS BB0013 AND ORDER AMOUNT IS   40
 INVENTORY NUMBER IS AB5975 AND ORDER AMOUNT IS   22
 INVENTORY NUMBER IS CH0060 AND ORDER AMOUNT IS   33
       6 ACCESSES, 6 HITS, 6 IO-S
 --
? END
```

Figure 4-3. Querying a Non-Data-Base File

```
/ QU                                                      Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110        82/05/03  10.09.16

 --
? OS,ATTACH,INDFILE/M=W                                   Issue operating system ATTACH and FILE
                                                          control statements.
 --
? OS,FILE,INDFILE,BT=C,RT=F

 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +     Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
? DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER            Position INDFILE at the end-of-
AB5972   25                                               information.
CB0168   50
YB0020   35
BB0013   40
AB5975   22
CH0060   33
       6 ACCESSES, 6 HITS, 6 IO-S

 --
? SEP ITEM-SIZE                                           Indicate to Query Update to concate-
                                                          nate the fields.
 --
? DISPLAY UPON INDFILE $CH0575$ $0006$                    Enter four records in INDFILE.  Under
                                                          NOS 1, the DISPLAY UPON directive
 --                                                       eliminates the old directory.  Under
? DISPLAY UPON INDFILE $SH0011$ $0004$                    NOS 2 and NOS/BE, the directive
                                                          creates a new directory to INDFILE.
 --
? DISPLAY UPON INDFILE $TY5015$ $0012$

 --
? DISPLAY UPON INDFILE $XN6158$ $0003$

 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +     Establish the directory for INDFILE
? AND IND-ON-ORDER AS NUM BY $Z(3)9$                      under NOS 1.

 --
? DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER            The DISPLAY operation verifies that
 AB5972   25                                              four records have been added to the
 CB0168   50                                              file.
 YB0020   35
 BB0013   40
 AB5975   22
 CH0060   33
 CH0575    6
 SH0011    4
 TY5015   12
 XN6158    3
      10 ACCESSES, 10 HITS, 10 IO-S

 --
? END                                                     The added data is made permanent.
```

Figure 4-4. Entering Data Into a Non-Data-Base File Under NOS

```
COMMAND- QU                                               Call Query Update.

 QUERY UPDATE 3.3 538-81089    82/05/03  10.09.16

--
      OS,ATTACH,INDFILE,ID=QUSER                          Issue operating system ATTACH and FILE con-
                                                          trol statements.
--
      OS,FILE,INDFILE,BT=C,RT=F

--
      OS,SKIPF,INDFILE,300000                             Skip forward to position INDFILE at end-of-
                                                          information.
--
      SEP ITEM-SIZE                                       Indicate to Query Update to concatenate the
                                                          fields.
--
      DISPLAY UPON INDFILE $CM0575$ $0006$                Enter four records in INDFILE and create a
                                                          directory for INDFILE.
--
      DISPLAY UPON INDFILE $SH0011$ $0004$

--
      DISPLAY UPON INDFILE $TY5015$ $0012$

--
      DISPLAY UPON INDFILE $XN6158$ $0003$

--
      DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER        A DISPLAY operation verifies that four
                                                          records have been added to the file.
      AB5972   25
      CB0168   50
      YB0020   35
      BB0013   40
      AB5975   22
      CH0060   33
      CM0575    6
      SH0011    4
      TY5015   12
      XN6158    3
      10 ACCESSES, 10 HITS, 10 IO-S

--
      OS,EXTEND,INDFILE                                   Extend INDFILE and make the data permanent.

--
      END
```

Figure 4-5. Entering Data Into a Non-Data-Base File Under NOS/BE

COMBINING DATA BASE AND NON-DATA-BASE FILE OPERATIONS 5

Section 3 introduces the MODIFY, STORE, and REMOVE directives and shows their use with data base files. The input source in each case was the user terminal. This section presents a variation of these directives and shows how they can be used to modify a data base file using a non-data-base file as the input source. Three operations can be performed:

MODIFY ... FROM

Modifies a data base file using data from a non-data-base file.

REMOVE ... FROM

Removes records from a data base file based on values contained in a non-data-base file.

STORE ... FROM

Stores data in a data base file using data from a non-data-base file.

Whenever combined data base and non-data-base file operations are anticipated, the following requirements must be met:

A data base file key field value must appear in the non-data-base file.

Information in a non-data-base file must be arranged in the order in which it is to be referenced in a Query Update directive.

Data base file INVENTORY and non-data-base file INDFILE contain common fields: inventory number and amount on order. The following paragraphs describe modification of INVENTORY based on values contained in INDFILE.

MODIFYING ACROSS FILES

The INVENTORY ON-ORDER fields, which are assumed to be noncumulative, can be modified with values contained in the corresponding INDFILE IND-ON-ORDER fields. The MODIFY ... FROM directive performs this type of operation as shown in figure 5-1.

INDFILE is appropriately attached and described and the subschema is named with the INVOKE directive. Displays against INVENTORY and INDFILE indicate the present status of the ON-ORDER fields.

The SEP ITEM-SIZE directive tells Query Update to use the directory description (established by the DESCRIBE directive) to determine the size of each item in INDFILE. The MODIFY ... FROM directive specifies the INVENTORY file primary key field and the field to be modified. CYBER Record Manager (CRM) locates the appropriate data base record and Query Update moves the new value into the corresponding ON-ORDER field. A second display against INVENTORY shows that data records with primary keys AB5972, AB5975, BB0013, CB0168, CH0060, and YB0020 reflect new totals for the ON-ORDER fields.

The following should be noted about the MODIFY ... FROM directive:

The fields specified in the USING option must be a primary or an alternate key field in the data base file. If a primary key is specified, only one record will be modified because the key value is unique. If an alternate key is specified, all records with the same value as the key are modified.

The non-data-base file is automatically rewound before a MODIFY ... FROM directive is executed.

Conditional testing on a data base field with the IF directive is not allowed when the USING option has been specified.

```
/ QU                                                          Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110          82/05/03  10.09.16

 --
? OS,ATTACH,INDFILE (UN=username)                             Issue operating system ATTACH and FILE
                                                              control statements.  NOS/BE requires an
                                                              ID parameter instead of a UN parameter.
 --
? OS,FILE,INDFILE,BT=C,RT=F


 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +         Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
? INVOKE QUSUB (UN=CAH0220)                                   Name the subschema.

 --
? DISPLAY INV-NO ON-ORDER
 AB5972   12                                                  This is the present status of INVENTORY
 AB5973    7                                                  ON-ORDER fields.
 AB5975   12
 BB0013    8
 CB0168    0
 CB1002    0
 CB1003    0
 CB1004    0
 CH0059    6
 CH0060    0
 CH0080    0
 CH0100    2
 CH0575    6
 SH0011    3
 (MORE... ANSWER Y OR N)
? Y
 TY5015    0
 XN6158    0
 YB0020    0
 YB0060    4
       18 ACCESSES, 18 HITS, 18 IO-S

 --
? DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER
 AB5972   25                                                  This is the present status of INDFILE
 CB0168   50                                                  IND-ON-ORDER fields.
 YB0020   35
 BB0013   40
 AB5975   22
 CH0060   33
 CH0575    6
 SH0011    4
 TY5015   12
 XN6158    3
      10 ACCESSES, 10 HITS, 10 IO-S

 --
? SEP ITEM-SIZE                                               Indicate that the $ is not present in
                                                              INDFILE; Query Update must use the
                                                              directory descriptions.
 --
? MODIFY USING INV-NO SETTING ON-ORDER FROM INDFILE           Modify INVENTORY from INDFILE.
       9 ACCESSES, 9 HITS, 18 IO-S
```

Figure 5-1. Modifying INVENTORY From INDFILE (Sheet 1 of 2)

```
 --
? DISPLAY INV-NO ON-ORDER                         A display of INVENTORY shows that the
 AB5972   25                                      contents of nine ON-ORDER fields have
 AB5973    7                                      changed.
 AB5975   22
 BB0013   40
 CB0168   50
 CB1002    0
 CB1003    0
 CB1004    0
 CH0059    6
 CH0060   33
 CH0080    0
 CH0100    2
 CH0575    6
 SH0011    4
 (MORE... ANSWER Y OR N)
? Y
 TY5015   12
 XN6158    3
 YB0020   35
 YB0060    4
       18 ACCESSES, 18 HITS, 18 IO-S

 --
? END                                             Terminate Query Update.
```

Figure 5-1. Modifying INVENTORY From INDFILE (Sheet 2 of 2)

REMOVING ACROSS FILES

Records can be removed from INVENTORY based on information contained in INDFILE. Removal is performed by the REMOVE ... FROM directive. Key field values in the non-data-base file are compared to key field values in the data base file; when the values match, the appropriate data base file record is removed. This type of operation is shown in figure 5-2.

INDFILE is appropriately attached and described, and the subschema QUSUB is named with the INVOKE directive. A display of INVENTORY indicates the total number of records currently stored.

The REMOVE ... FROM directive specifies the INVENTORY primary key field. CRM locates the data base record whose key field matches the non-data-base file field, and Query Update removes the data base record. A second display of INVENTORY reflects a new total, indicating that the removal took place.

The following should be noted about the REMOVE ... FROM directive:

A field specified in the USING option must be a primary or an alternate key field in the data base file. Additional fields, if included, are used for spacing in the file specified in the FROM clause.

A non-data-base file is automatically rewound before a REMOVE ... FROM directive is executed.

Conditional testing on a data base field with an IF directive is not allowed when the USING option has been specified.

STORING ACROSS FILES

Records in INDFILE can be stored into INVENTORY. The STORE ... FROM directive performs this type of operation as shown in figure 5-3.

INDFILE is appropriately attached and described, and the subschema QUSUB is named with the INVOKE directive. A display of INVENTORY indicates the total number of records currently stored.

The STORE ... FROM directive specifies the INVENTORY file primary key INV-NO field and the ON-ORDER field. Query Update stores the new record entries in the data base file. A second display of INVENTORY reflects a new total, indicating that the store took place. A display of inventory record AB5975 shows that Query Update numeric fields are zero filled and alphanumeric fields are blank filled.

The following should be noted about the STORE ... FROM directive:

The field specified in the SETTING option must be a primary or an alternate key field in the data base file.

The non-data-base file is automatically rewound before a STORE ... FROM directive is executed.

Conditional testing on a data base field with the IF directive is not allowed when the SETTING option has been specified.

```
/ QU

 QUERY UPDATE 3.4 SYSEDIT-82110         82/05/03  10.09.16

 --
? OS,ATTACH,INDFILE (UN=username)                               Issue operating system ATTACH and FILE
                                                                control statements.  NOS/BE requires
                                                                an ID parameter instead of a UN
 --                                                             parameter.
? OS,FILE,INDFILE,BT=C,RT=F

 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +           Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
? INVOKE QUSUB (UN=CAH0220)                                     Name the subschema.  NOS/BE requires an
                                                                ID parameter instead of a UN parameter.
 --
? DISPLAY INV-NO                                                A display of INVENTORY primary key
 AB5972                                                         fields shows that 18 records are stored
 AB5973                                                         in the file.
 AB5975
 BB0013
 CB0168
 CB1002
 CB1003
 CB1004
 CH0059
 CH0060
 CH0080
 CH0100
 CH0575
 SH0011
 (MORE... ANSWER Y OR N)
? Y
 TY5015
 XN6158
 YB0020
 YB0060
       18 ACCESSES, 18 HITS, 18 IO-S

 --
? SEP ITEM-SIZE                                                 Indicate that the $ is not present in
                                                                INDFILE; Query Update must use the
                                                                directory descriptions.

 --
? REMOVE USING INV-NO FROM INDFILE                              Remove INVENTORY records whose primary
      10 ACCESSES, 10 HITS, 20 IO-S                             key values equal those in INDFILE.
                                                                The file is positioned at the beginning-
                                                                of-information because no operation has
                                                                been performed against INDFILE.
 --
? DISPLAY INV-NO                                                A display of INVENTORY primary key
 AB5973                                                         fields shows that 10 records have been
 CB1002                                                         removed.
 CB1003
 CB1004
 CH0059
 CH0080
 CH0100
 YB0060
       8 ACCESSES, 8 HITS, 8 IO-S

 --
? END                                                           Terminate Query Update.
```

Figure 5-2. Removing INVENTORY Records Via INDFILE

```
/ QU

 QUERY UPDATE 3.4 SYSEDIT-82110          82/05/03  10.09.16

 --
? OS,ATTACH,INDFILE (UN=username)                              Issue operating system ATTACH and FILE
                                                               control statements.  NOS/BE requires an
                                                               ID parameters instead of a UN parameter.

 --
? OS,FILE,INDFILE,BT=C,RT=F

 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +          Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
? INVOKE QUSUB (UN=XXXXXXX)                                    Name the subschema.  NOS/BE requires an
                                                               ID parameter instead of a UN parameter.
 --
? DISPLAY INV-NO ON-ORDER                                      A display of INVENTORY shows that 8
 AB5973    7                                                   records are stored in the file.
 CB1002    0
 CB1003    0
 CB1004    0
 CH0059    6
 CH0080    0
 CH0100    2
 YB0060    4
       8 ACCESSES, 8 HITS, 8 IO-S

 --
? SEP ITEM-SIZE                                                Indicate that the $ is not present in
                                                               INDFILE; Query Update must use the
                                                               directory descriptions.

 --
? STORE SETTING INV-NO ON-ORDER FROM INDFILE                   Store INDFILE records into INVENTORY.
      10 ACCESSES, 10 HITS, 10 IO-S

--
? DISPLAY INV-NO ON-ORDER                                      A display of INVENTORY shows that 10
 AB5972   25                                                   records have been stored.
 AB5973    7
 AB5975   22
 BB0013   40
 CB0168   50
 CB1002    0
 CB1003    0
 CB1004    0
 CH0059    6
 CH0060   33
 CH0080    0
 CH0100    2
 CH0575    6
 SH0011    4
 (MORE... ANSWER Y OR N)
? Y
 TY5015   12
 XN6158    3
 YB0020    0
 YB0060    4
       18 ACCESSES, 18 HITS, 18 IO-S

 --
? IF INV-NO EQ $AB5975$ DISPLAY INV-REC                        A display of a stored record shows that
 (209) REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT             Query Update zero filled numeric fields
  AB5975000000000022000000000000000                            and blank filled alphanumeric fields.
       1 ACCESSES, 1 HITS, 2 IO-S

 --
? END                                                          Terminate Query Update.
```

Figure 5-3. Storing INVENTORY Records Via INDFILE

QUERY UPDATE REPORTS 6

A variety of Query Update directives are available for preparing and generating reports. Reports can be displayed at the terminal, printed on either a terminal hardcopy printer or a line printer, stored as permanent files, or any combination of the four.

Reports can be generated from a data base or non-data-base file. Data for the report is retrieved according to a directory established at report time.

Data base file report operations use a temporary directory that points to a subset of the data base file. The subset is a sequential file composed of the fields selected for the report. Both the file subset and the temporary directory are created by an EXTRACT directive.

Non-data-base file report operations use the directory established by a DESCRIBE directive.

Reports are named by the FORMAT directive. The layout directives that follow the FORMAT directive provide the report titles, column headings, column positioning, page numbering, and so forth.

Production of the report is initiated through a PREPARE directive. Operating system commands are used to direct the finished report to the desired output device.

Reports shown in this section combine operations under NOS and NOS/BE. Some differences are the following:

The operating system request for input. NOS prints a slash (/), NOS/BE prints the word COMMAND-.

The permanent file parameters. Directives that reference permanent files require certain parameters, as shown in the figures.

The permanent file storage commands. NOS requires DEFINE and COPY control statements, NOS/BE requires a CATALOG control statement.

The report output commands. NOS requires a ROUTE control statement, NOS/BE requires a BATCH control statement.

The commands for viewing reports at the terminal (not shown in figures).

NOS requires you to enter the word EDIT, a comma, and the name of the report to be displayed. Then the text editor prints a heading and waits for input. You enter the word PRINT followed by an asterisk to display the entire report. Printing is terminated when you enter the word END.

NOS/BE requires you to enter the word PAGE, a comma, and the name of the report to be displayed. Then the PAGE system prints a Ready message and waits for input. You enter the number of the first line to be displayed. Paging continues if you enter a plus sign (+), paging terminates if you enter the letter E for end.

The general sequence of commands and directives for report generation is summarized in table 6-1. Reports prepared in this manner are referred to as one-time reports because they are available only during the current session.

TABLE 6-1. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR ONE-TIME REPORTING

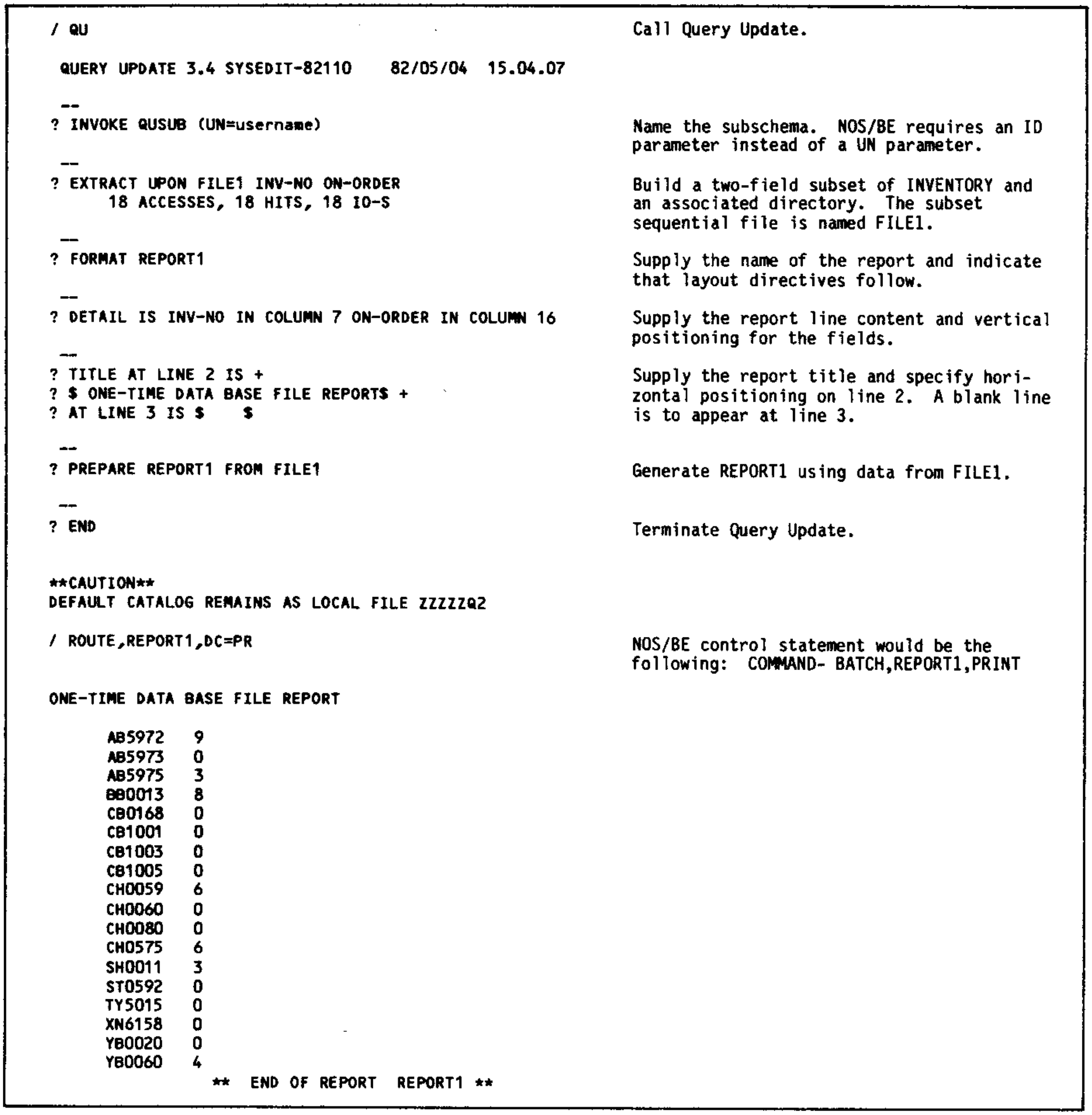
Non-Data-Base File Operations	Data Base File Operations
QU	QU
OS†,ATTACH,non-data-base file ...	INVOKE,subschema ...
OS†,FILE ...	EXTRACT
DESCRIBE	FORMAT
FORMAT	. . (layout directives) .
. . (layout directives) .	PREPARE
PREPARE	END
END	Output Commands
Output Commands	

†You can enter the operating system ATTACH and FILE control statements, rather than the Query Update OS directives, before the QU control statement.

The basic commands and directives for a one-time report from INVENTORY are shown in figure 6-1.

The basic commands and directives for a one-time report from INDFILE are shown in figure 6-2.

In most cases it is better to prepare report formats and store them as permanent files so they can be used over and over again. This convention is called cataloging and is described in the following paragraphs.



```
/ QU                                                    Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/04  15.04.07

 --
? INVOKE QUSUB (UN=username)                            Name the subschema.  NOS/BE requires an ID
                                                        parameter instead of a UN parameter.
 --
? EXTRACT UPON FILE1 INV-NO ON-ORDER                    Build a two-field subset of INVENTORY and
      18 ACCESSES, 18 HITS, 18 IO-S                     an associated directory.  The subset
                                                        sequential file is named FILE1.
 --
? FORMAT REPORT1                                        Supply the name of the report and indicate
                                                        that layout directives follow.
 --
? DETAIL IS INV-NO IN COLUMN 7 ON-ORDER IN COLUMN 16     Supply the report line content and vertical
                                                        positioning for the fields.
 --
? TITLE AT LINE 2 IS +                                  Supply the report title and specify hori-
? $ ONE-TIME DATA BASE FILE REPORT$ +                   zontal positioning on line 2.  A blank line
? AT LINE 3 IS $    $                                   is to appear at line 3.

 --
? PREPARE REPORT1 FROM FILE1                            Generate REPORT1 using data from FILE1.

 --
? END                                                   Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ ROUTE,REPORT1,DC=PR                                   NOS/BE control statement would be the
                                                        following:  COMMAND- BATCH,REPORT1,PRINT

ONE-TIME DATA BASE FILE REPORT

      AB5972   9
      AB5973   0
      AB5975   3
      BB0013   8
      CB0168   0
      CB1001   0
      CB1003   0
      CB1005   0
      CH0059   6
      CH0060   0
      CH0080   0
      CH0575   6
      SH0011   3
      ST0592   0
      TY5015   0
      XN6158   0
      YB0020   0
      YB0060   4
                 **  END OF REPORT  REPORT1 **
```

Figure 6-1. Generating a One-Time Report From INVENTORY

```
/ QU                                                  Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/12  15.32.30

 --
? OS,ATTACH,INDFILE (UN=username)                     Attach INDFILE and declare file organiza-
                                                      tion for CRM.  NOS/BE requires an ID param-
 --                                                   eter instead of a UN parameter.
? OS,FILE,INDFILE,BT=C,RT=F

 --
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +  Establish the directory for INDFILE.
? AND IND-ON-ORDER AS NUM BY $Z(3)9$

 --
? FORMAT REPORT2                                      Supply the name of the report and indicate
                                                      that layout directives follow.
 --
? DETAIL IS IND-INV-NO IN COLUMN 7 +                  Supply report line content and vertical
? IND-ON-ORDER IN COLUMN 16                           positioning for the fields.

 --
? TITLE AT LINE 2 IS +                                Supply report title and specify horizontal
? $ONE-TIME INDEPENDENT FILE REPORT$ +                positioning on line 2.  A blank line is to
? AT LINE 3 IS $    $                                 appear at line 3.

 --
? PREPARE REPORT2 FROM INDFILE                        Rewind INDFILE and execute REPORT2
                                                      directives.
 --
? END                                                 Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ ROUTE,REPORT2,DC=PR                                 NOS/BE control statement would be the
                                                      following:  COMMAND- BATCH,REPORT2,PRINT

ONE-TIME INDEPENDENT FILE REPORT

      AB5972   25
      CB0168   50
      YB0020   35
      BB0013   40
      AB5975   33
      CH0060   33
                  **  END OF REPORT  REPORT2 **
```

Figure 6-2. Generating a One-Time Report From INDFILE

QUERY UPDATE CATALOGS

Report directives are stored on a local file named ZZZZZQ2, which is known as the default catalog. At the beginning of each new Query Update session (that is, whenever the QU control statement is entered), the previous default catalog is returned to the operating system and a new default catalog is made available. This means that the report directives entered on the default catalog for one session cannot be made available for another session. All report directives for that specific report must be entered again.

This problem is solved by storing the default catalog as a permanent file. When the report is needed at a later time, the permanent catalog can be attached, the Query Update session directives initiated, and the report generated with minimum effort.

CATALOGING REPORTS

Recording Query Update sessions on a permanent catalog represents an efficient method of generating reports. Report directives need to be entered once, then the report itself can be generated over and over again. Reports can always be changed, even after they are cataloged. The ALTER directive, which is discussed later in this section, accepts subsequent directives that remove, replace, or add report specifications to the cataloged report.

The general sequence of commands and directives for cataloging reports is summarized in table 6-2.

TABLE 6-2. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR CATALOGING REPORTS

Non-Data-Base File Operations	Data Base File Operations
QU OS[†],ATTACH,non-data-base file, ...[†] OS,FILE, ...[†] RECORDING DESCRIBE PREPARE END[†††] RECORDING OFF FORMAT . . (layout directives) . END Permanent File Storage Commands	QU RECORDING INVOKE subschema ...[††] EXTRACT PREPARE END[†††] RECORDING OFF FORMAT . . (layout directives) . END Permanent File Storage Commands

[†] You can use the operating system ATTACH and FILE statements, rather than the Query Update OS directives, before the QU control statement. The OS directives can be positioned immediately after the RECORDING directive and be recorded as part of a session.

[††] The INVOKE directive can follow the QU control statement and remain outside of the recording limits.

[†††] The END directive within the recording limits is optional. When this directive is included, control reverts to the operating system immediately after the cataloged session is subsequently performed. When this directive is omitted, control remains with Query Update.

Notice that the FORMAT directive initiates retention of the associated layout directives. These directives are retained in the catalog under the report name specified in the FORMAT directive.

Figure 6-3 shows the basic commands and directives for cataloging a report from INVENTORY.

Figure 6-4 shows the basic commands and directives for cataloging a report from INDFILE.

USING CATALOGED REPORTS

Cataloged reports can be produced with a minimum of two directives: VERSION and PERFORM. The VERSION directive attaches the named permanent catalog. The PERFORM directive retrieves and executes the directives recorded under the named session. The general sequence of commands and directives for using cataloged reports is summarized in table 6-3.

TABLE 6-3. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR USING CATALOGED REPORTS

Non-Data-Base File Operations	Data Base File Operations
QU OS,ATTACH,non-data-base file ...[†] VERSION PERFORM Report Disposition Commands	QU VERSION ... PERFORM Report Disposition Commands

[†] The ATTACH and FILE control statements can assume three positions: before the QU control statement as normal operating system control statements, after the QU control statement as Query Update OS directives, or within the recorded session called by the Query Update VERSION directive.

Figure 6-5 shows the generation of data base file REPORT3 from the QUCAT catalog.

Figure 6-6 shows the generation of non-data-base file REPORT4 from the QUCAT catalog.

When the VERSION directive references the permanent catalog, some permanent file parameters are required as shown in the figures.

```
/ QU                                                        Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/12  16.42.18

 --
? RECORDING SESS3                                           Initiate recording of session and name it
                                                            SESS3.
 001--
? INVOKE QUSUB (UN=username)                                Name the subschema.  NOS/BE requires an ID
                                                            parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE3 INV-NO ON-ORDER

 003--
? PREPARE REPORT3 FROM FILE3

 004--
? END                                                       Terminate execution of SESS3.

 005--
? RECORDING OFF                                             Terminate the recording session.
     END OF SESSION SESS3

 --
? FORMAT REPORT3                                            Supply the name of the report.

 --
? DETAIL IS INV-NO IN COLUMN 7 +
? ON-ORDER IN COLUMN 16

 --
? TITLE AT LINE 2 IS +
? $RECORDED DATA BASE FILE REPORT$ +
? AT LINE 3 IS $    $

 --
? END                                                       Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ DEFINE,QUCAT                                              NOS/BE control statements would be the
/ COPY,ZZZZZQ2,QUCAT                                        following:
                                                            COMMAND- CATALOG,ZZZZZQ2,QUCAT,ID=QUSER
```

Figure 6-3. Cataloging a Report From INVENTORY

```
/ QU                                                        Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/12  16.55.44

 --
? OS,ATTACH,INDFILE (UN=username)                           Attach INDFILE.  NOS/BE requires an ID
                                                            parameter instead of a UN parameter.
 --
? OS,FILE,INDFILE,BT=C,RT=F                                 Declare file organization for CRM.

 --
? RECORDING SESS4                                           Initiate recording and name the session
                                                            SESS4.
 001--
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +       Query Update prints transmission IDs
? AND IND-ON-ORDER AS NUM BY $Z(3)9$                        numbered sequentially from 001.  Each
                                                            executable directive is entered for a
                                                            report named REPORT4.
 002--
? PREPARE REPORT4 FROM INDFILE
```

Figure 6-4. Cataloging a Report From INDFILE (Sheet 1 of 2)

```
 003--
? END                                                        Terminate execution of SESS4.

 004--
? RECORDING OFF                                              Terminate the recording session.
      END OF SESSION SESS4

 --
? FORMAT REPORT4                                             Supply the name of the report.

 --
? DETAIL IS IND-INV-NO IN COLUMN 7 +
? IND-ON-ORDER IN COLUMN 16

 --
? TITLE AT LINE 2 IS +
? $RECORDED NON-DATA-BASE FILE REPORT$ +
? AT LINE 3 IS $    $

 --
? END                                                        Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ DEFINE,QUCAT                                               NOS/BE control statement would be the
/ COPY,ZZZZZQ2,QUCAT                                         following:
                                                             COMMAND- CATALOG,ZZZZZQ2,QUCAT,ID=QUSER
```

Figure 6-4. Cataloging a Report From INDFILE (Sheet 2 of 2)

```
/ QU                                                         Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  10.35.49

 --
? VERSION QUCAT (UN=username)                                Attach the permanent catalog QUCAT.
                                                             NOS/BE requires an ID parameter instead
                                                             of the UN parameter.
 --
? PERFORM SESS3                                              Retrieve and execute the directives
                                                             recorded under SESS3.
 --
? END                                                        Terminate Query Update.


/ ROUTE,REPORT3,DC=PR                                        NOS/BE control statement would be the
                                                             following:  COMMAND- BATCH,REPORT3,PRINT

RECORDED DATA BASE FILE REPORT

      AB5972   9
      AB5973   0
      AB5975   3
      BB0013   8
      CB0168   0
      CB1001   0
      CB1003   0
      CB1005   0
      CH0059   6
      CH0060   0
      CH0080   0
      CM0575   6
      SH0011   3
      ST0592   0
      TY5015   0
      XN6158   0
      YB0020   0
      YB0059   0
      YB0060   4
             **  END OF REPORT  REPORT3 **
```

Figure 6-5. Generating Data Base File REPORT3 From a Permanent Catalog

```
/ QU                                                    Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  10.58.11

 --
? OS,ATTACH,INDFILE (UN=username)                       Attach INDFILE.  NOS/BE requires an ID
                                                        parameter instead of a UN parameter.
 --
? OS,FILE,INDFILE,BT=C,RT=F                             Declare the file organization for CRM.

 --
? VERSION QUCAT (UN=username)                           Attach the permanent file catalog.  NOS/BE
                                                        requires an ID parameter instead of a UN
                                                        parameter.
 --
? PERFORM SESS4                                         Retrieve and execute the directives recorded
                                                        under SESS4 (refer to figure 6-4).
 --
? END                                                   Terminate Query Update.


/ ROUTE,REPORT4,DC=PR                                   NOS/BE control statement would be the
                                                        following:  COMMAND- BATCH,REPORT4,PRINT

RECORDED NON-DATA-BASE FILE REPORT

      AB5972   25
      CB0168   50
      YB0020   35
      BB0013   40
      AB5975   22
      CH0060   33
                  **  END OF REPORT  REPORT4 **
```

Figure 6-6. Generating Non-Data-Base File REPORT4 From a Permanent Catalog

DUPLICATING CATALOGED REPORTS

Reports that are recorded on a permanent catalog can be duplicated on the default catalog. Conversely, reports that are recorded on the default catalog can be duplicated on a permanent catalog. Session names, report names, and transmission IDs (the three-digit number appended to a directive at recording time) can be selected for duplication.

A duplicate report operation from the default catalog to the permanent catalog is shown in figure 6-7. This option is a convenient way to add reports to an existing permanent catalog.

A duplicate report operation from the permanent catalog to the default catalog is shown in figure 6-8. This option is a convenient way to access cataloged report formats that require modification for one-time use. Changes can be made to the duplicated report format on the default catalog without affecting the report format stored on the permanent file catalog.

```
/ QU                                                    Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  13.24.17

 --
? RECORDING SESS5                                       Initiate recording of a session and name
                                                        it SESS5.
 001--
? INVOKE QUSUB (UN=username)                            Name the subschema.  NOS/BE requires an ID
                                                        parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE5 INV-NO ON-ORDER

 003--
? PREPARE REPORT5 FROM FILE5
```

Figure 6-7. Duplicating a Report From a Default to a Permanent Catalog (Sheet 1 of 2)

```
 004--
? END

 005--
? RECORDING OFF
     END OF SESSION SESS5

 --
? FORMAT REPORT5                                              REPORT5 is formatted on the default
                                                              catalog.
 --
? DETAIL IS INV-NO IN COLUMN 7 +
? ON-ORDER IN COLUMN 16

 --
? TITLE AT LINE 2 IS +
? $DUPLICATED REPORT$ IN COLUMN 7 +
? AT LINE 3 IS $    $

 --
? VERSION QUCAT (UN=username/M=W)                             Attach the QUCAT catalog.  NOS/BE requires
                                                              an ID parameter instead of a UN parameter.
 --
? DUPLICATE SESSION SESS5                                     Copy SESS5.

 --
? DUPLICATE REPORT REPORT5                                    Copy REPORT5.

 --
? PERFORM SESS5                                               Retrieve and execute the directives
                                                              recorded on permanent file catalog QUCAT.

 --
? END                                                         Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ ROUTE,REPORT5,DC=PR                                         NOS/BE control statement would be the
                                                              following:  COMMAND- BATCH,REPORT5,PRINT

DUPLICATED REPORT

      AB5972   9
      AB5973   0
      AB5975   3
      BB0013   8
      CB0168   0
      CB1001   0
      CB1003   0
      CB1005   0
      CH0059   6
      CH0060   0
      CH0080   0
      CM0575   6
      SH0011   3
      ST0592   0
      TY5015   0
      XN6158   0
      YB0020   0
      YB0059   0
      YB0060   4
                 **  END OF REPORT  REPORT5 **
```

Figure 6-7. Duplicating a Report From a Default to a Permanent Catalog (Sheet 2 of 2)

```
/ QU                                                      Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  14.12.10

 --
? VERSION QUCAT (UN=username)                             Attach the permanent file catalog QUCAT.
                                                          NOS/BE requires an ID parameter instead of
                                                          a UN parameter.
 --
? DUP UPON DEFAULT SESSION SESS3 AS DUP3                  Copy SESS3.

 --
? DUP UPON DEFAULT REPORT REPORT3                         Copy REPORT4.

 --
? VERSION DEFAULT                                         Future PERFORM directives will use the
                                                          default catalog.
 --
? PERFORM DUP3                                            Retrieve and execute the directives dupli-
                                                          cated on the default catalog.  Notice that
                                                          the new session name DUP3 is used; SESS3
 --                                                       would not be found on the default catalog.
? END                                                     Terminate Query Update.


**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ ROUTE,REPORT3,DC=PR                                     NOS/BE control statement would be the
                                                          following:  COMMAND- BATCH,REPORT3,PRINT

RECORDED DATA BASE FILE REPORT

      AB5972   9
      AB5973   0
      AB5975   3
      BB0013   8
      CB0168   0
      CB1001   0
      CB1003   0
      CB1005   0
      CH0059   6
      CH0060   0
      CH0080   0
      CM0575   6
      SH0011   3
      ST0592   0
      TY5015   0
      XN6158   0
      YB0020   0
      YB0059   0
      YB0060   4
                 **  END OF REPORT  REPORT3 **
```

Figure 6-8. Duplicating a Report From a Permanent to a Default Catalog

ALTERING REPORTS

Report formats can be altered by adding or removing one or more layout directives. The report whose format is to be altered can reside on the permanent or default catalog. The operation is initiated by the ALTER directive, which names the report. The directives that immediately follow effect the changes.

Adding a Directive

A new directive is to be added to REPORT3. Each time the report is produced, the system is to supply the data and position it on the title line at column 45. The addition of the new directive is shown in figure 6-9.

The QUCAT catalog on which REPORT3 resides is attached with the VERSION directive. The report to be altered is named with the ALTER directive. The horizontal and vertical positioning of the date is supplied with the DATE directive.

Report modification ends when a directive not associated with report modification is entered. Since the PERFORM directive is not related to report modification, no further modifications are made. The subsequent report reflects the current date.

```
/ QU                                                  Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  14.45.09

 --
? VERSION QUCAT (UN=username/M=W)                     Attach the permanent file catalog QUCAT.
                                                      NOS/BE requires an ID parameter instead of
                                                      a UN parameter.
 --
? ALTER REPORT3                                       Locate the report directives for REPORT3.

 --
? DATE AT TITLE-LINE COLUMN 45                        Add the DATE directive to the report.  DATE
                                                      specifies the current date, which is sup-
                                                      plied by the system.
 --
? PERFORM SESS3                                       Terminate report modification, retrieve and
                                                      execute SESS3.  The report will now have
                                                      the current date each time it is prepared
                                                      and output.
 --
? END                                                 Terminate Query Update.


/ ROUTE,REPORT3,DC=PR                                 NOS/BE control statement would be the
                                                      following:  COMMAND- BATCH,REPORT3,PRINT

RECORDED DATA BASE FILE REPORT              81/06/17

      AB5972   9
      AB5973   0
      AB5975   3
      BB0013   8
      CB0168   0
      CB1001   0
      CB1003   0
      CB1005   0
      CH0059   6
      CH0060   0
      CH0080   0
      CM0575   6
      SH0011   3
      ST0592   0
      TY5015   0
      XN6158   0
      YB0020   0
      YB0059   0
      YB0060   4
              **  END OF REPORT  REPORT3 **
```

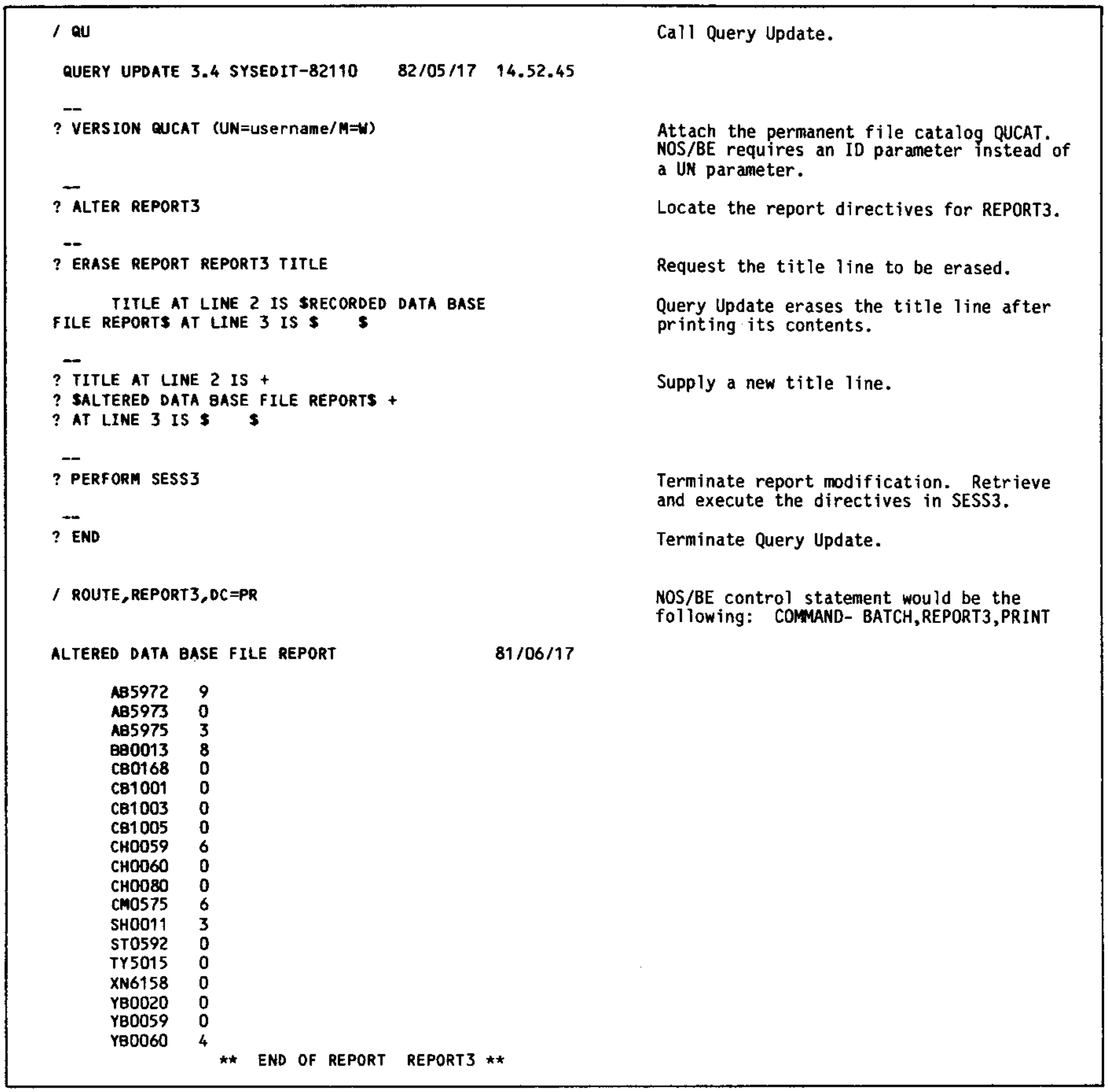
Figure 6-9. Adding a Directive to a Report

Erasing a Directive

A new title is to be given to REPORT3. The present TITLE directive is to be removed and a new TITLE directive is to be stored. The use of the ERASE directive is shown in figure 6-10.

The QUCAT catalog on which REPORT3 resides is attached with the VERSION directive and the report to be altered is named with the ALTER directive. The ERASE directive indicates the TITLE directive is to be removed. Query Update erases the directive and prints its contents. The TITLE directive adds a new title. The PERFORM directive terminates modification and executes the session.

The subsequent report reflects the new title.



```
/ QU                                                Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  14.52.45

 --
? VERSION QUCAT (UN=username/M=W)                   Attach the permanent file catalog QUCAT.
                                                    NOS/BE requires an ID parameter instead of
                                                    a UN parameter.
 --
? ALTER REPORT3                                     Locate the report directives for REPORT3.

 --
? ERASE REPORT REPORT3 TITLE                        Request the title line to be erased.

     TITLE AT LINE 2 IS $RECORDED DATA BASE         Query Update erases the title line after
FILE REPORT$ AT LINE 3 IS $    $                    printing its contents.

 --
? TITLE AT LINE 2 IS +                              Supply a new title line.
? $ALTERED DATA BASE FILE REPORT$ +
? AT LINE 3 IS $    $

 --
? PERFORM SESS3                                     Terminate report modification.  Retrieve
                                                    and execute the directives in SESS3.
 --
? END                                               Terminate Query Update.


/ ROUTE,REPORT3,DC=PR                               NOS/BE control statement would be the
                                                    following:  COMMAND- BATCH,REPORT3,PRINT

ALTERED DATA BASE FILE REPORT              81/06/17

     AB5972   9
     AB5973   0
     AB5975   3
     BB0013   8
     CB0168   0
     CB1001   0
     CB1003   0
     CB1005   0
     CH0059   6
     CH0060   0
     CH0080   0
     CM0575   6
     SH0011   3
     ST0592   0
     TY5015   0
     XN6158   0
     YB0020   0
     YB0059   0
     YB0060   4
               **  END OF REPORT  REPORT3 **
```

Figure 6-10. Erasing a Directive From a Report

SAMPLE REPORTS

Sample reports are shown on the following pages. All reports are recorded on the permanent catalog QUCAT and the reference data base file INVENTORY. For NOS, the write permission mode parameter (M=W) must be included on the VERSION directive for both the owner and the alternate user. Many of the Query Update directives that appear in the reports will be familiar because they have already been discussed in relation to cataloging operations. For this reason, only new directives are detailed in the figures.

Figure 6-11 is a sample report that extracts information from four fields, sorts the information on the IN-STOCK field, positions the information in four columns, and supplies the title, date, and time. This report shows the most commonly used report directives and introduces two other directives: SORT and TIME.

Figure 6-12 is a sample report that shows the definition of a temporary item. As mentioned in an earlier section, a temporary item is one of the three categories of data handled by Query Update. This report defines temporary item PROFIT-MARGIN and uses it to calculate entries for one of the report columns. Three directives are introduced: DEFINE, TABS, and EVALUATE.

Figure 6-13 is a sample report that presents one of many possible applications for the universal character. This report declares the asterisk as the universal character and uses it for special testing against the INV-NO field. Three directives are introduced: UNIVERSAL, SPECIFY, and SELECT.

Figure 6-14 is a sample report that extracts information from four fields, sorts the information on the IN-STOCK field, and specifies a break condition based on the contents of that field. Three directives are introduced: HEADING, PAGE-NUMBER, and BREAK.

Figure 6-15 is a sample report that specifies vertical positioning for the output report. Two directives are introduced: PAGE-SIZE and RECAP.

Figure 6-16 is a sample report that shows how lines of text can be designated to precede and follow a report. Two directives are introduced: PREFACE and SUMMARY.

```
/ QU                                                  Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/17  15.51.19

 --
? VERSION QUCAT (UN=username)                         Attach the permanent catalog.  NOS/BE
                                                      requires an ID parameter instead of a UN
                                                      parameter.
 --
? RECORDING SESS6                                     Begin recording session SESS6.

 001--
? INVOKE QUSUB (UN=username)                          Name the subschema.  NOS/BE requires an ID
                                                      parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE6 INV-NO DESCRIPTION +             Extract four fields from FILE6.
? IN-STOCK UNIT-PRICE

 003--
? REWIND FILE6 SORT6                                  Rewind FILE6 and the file to be used for
                                                      sorting operations.
 004--
? SORT FILE6 UPON SORT6 ON IN-STOCK                   Resequence records in FILE6 and place them
                                                      on file SORT6.
 005--
? PREPARE REPORT6 FROM SORT6                          Prepare REPORT6 from the sorted file.

 006--
? RECORDING OFF                                       Terminate recording of SESS6.
     END OF SESSION SESS6

 --
? FORMAT REPORT6                                      Format a report named REPORT6.

 --
? DETAIL IS IN-STOCK IN COLUMN 1 +                    Supply the report line content.
? INV-NO IN COLUMN 11 +
? DESCRIPTION IN COLUMN 24 +
? UNIT-PRICE IN COLUMN 48
```

Figure 6-11. Sample Report Illustrating the Basic Directives (Sheet 1 of 2)

```
 --
? TITLE AT LINE 1 IS +                                       Supply the report title.
? $BASIC REPORT$ IN COLUMN 24 +
? AT LINE 2 IS $    $

 --
? DATE AT TITLE-LINE COLUMN 60                               Request output of the current date.

 --
? TIME AT LINE 2 COLULMN 60                                  Request output of the current time.

 --
? PERFORM SESS6                                              Retrieve and execute SESS6, which contains
                                                             the directives in REPORT6.

 --
? END                                                        Terminate Query Update.

/ ROUTE,REPORT6,DC=PR                                        NOS/BE control statement would be the
                                                             following:  COMMAND- BATCH,REPORT6,PRINT

                       BASIC REPORT                    81/06/17
                                                       15.57.27
   2      AB5973       OAK DESK               1282.50
   2      AB5975       WALNUT DESK            1300.00
   2      TY5015       ELECT TYPEWRITER        369.00
   5      AB5972       METAL DESK              389.95
   5      CH0060       DESK CHAIR              149.95
   7      CH0059       ARM CHAIR               295.00
   7      ST0592       STOOL                    16.20
   8      CB0168       CHALK BOARD              19.52
   9      CH0080       SWIVEL CHAIR             96.00
  10      BB0013       BULLETIN BOARD           15.00
  10      CB1001       1-DR FILE CABINET        45.00
  10      CB1003       3-DR FILE CABINET        60.00
  10      CB1005       5-DR FILE CABINET        90.00
  10      SH0011       3-SHELF BOOK CASE        39.95
  10      XN6158       COFFEE TABLE             95.00
  20      YB0060       TABLE LAMP               39.95
  25      YB0059       FLOOR LAMP               69.95
  50      CM0575       LETTER RACK               3.98
 500      YB0020       DESK LAMP                19.95
              **  END OF REPORT  REPORT6 **
```

Figure 6-11. Sample Report Illustrating the Basic Directives (Sheet 2 of 2)

```
/ QU                                                         Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/18  09.15.12

 --
? VERSION QUCAT (UN=username)                                Attach the permanent file catalog.  NOS/BE
                                                             requires an ID parameter instead of a UN
                                                             parameter.
 --
? RECORDING SESS7                                            Begin recording session SESS7.

 001--
? INVOKE QUSUB (UN=username)                                 Name the subschema.  NOS/BE requires an ID
                                                             parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE7 DESCRIPTION +                           Extract three fields upon FILE7.
? UNIT-PRICE UNIT-COST
```

Figure 6-12. Sample Report Illustrating the Definition of a Temporary Data Item (Sheet 1 of 2)

```
 003--
? DEFINE PROFIT-MARGIN BY $Z(6)99$ = +                 Establish a temporary data item named
? (UNIT-PRICE - UNIT-COST) / UNIT-COST * 100           PROFIT-MARGIN.  The value of the item is a
                                                       mathematical expression that is to be
                                                       evaluated by Query Update.

 004--
? PREPARE REPORT7 FROM FILE7                           Prepare REPORT7 from the extracted file.

 005--
? END                                                  Terminate execution of SESS7.

 006--
? RECORDING OFF                                        Terminate recording of SESS7.
     END OF SESSION SESS7

 --
? FORMAT REPORT7                                       Format a report named REPORT7.

 --
? TABS AT 1 20                                         Establish tab positions at columns 1 and
                                                       20 for the report layout.

 --
? EVALUATE BEFORE ANY DETAIL PROFIT-MARGIN             Perform the mathematical operation indi-
                                                       cated in the above DEFINE directive.

 --
? DETAIL IS DESCRIPTION IN TAB 1 +                     Supply the report line content.
? PROFIT-MARGIN IN TAB 2

 --
? TITLE AT LINE 1 IS $TEMPORARY ITEM REPORT$ +         Supply the report title and set up two
? IN COLUMN 6 AT LINE 2 IS $    $ +                    column headings.
? AT LINE 3 IS $ITEM$ IN TAB 1 +
? IS $PROFIT MARGIN$ IN TAB 2 AT LINE 4 IS $    $

 --
? PERFORM SESS7                                        Retrieve and execute SESS7.

 --
? END                                                  Terminate Query Update.

/ ROUTE,REPORT7,DC=PR                                  NOS/BE control statement would be the
                                                       following:  COMMAND- BATCH,REPORT7,PRINT

     TEMPORARY ITEM REPORT

ITEM               PROFIT MARGIN

METAL DESK         122
OAK DESK           157
WALNUT DESK         44
BULLETIN BOARD     200
CHALK BOARD        144
1-DR FILE CABINET  200
3-DR FILE CABINET  200
5-DR FILE CABINET  181
ARM CHAIR          127
DESK CHAIR          68
SWIVEL CHAIR       174
LETTER RACK        306
3-SHELF BOOK CASE  100
STOOL               71
ELECT TYPEWRITER    12
COFFEE TABLE       150
DESK LAMP          263
FLOOR LAMP         278
TABLE LAMP         200
                        **  END OF REPORT  REPORT7 **
```

Figure 6-12. Sample Report Illustrating the Definition of a Temporary Data Item (Sheet 2 of 2)

```
/ QU                                                          Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/18  09.58.53

 --
? VERSION QUCAT (UN=username)                                 Attach the permanent file catalog.  NOS/BE
                                                              requires an ID parameter instead of a UN
                                                              parameter.

 --
? RECORDING SESS8                                             Begin recording SESS8.

 001--
? INVOKE QUSUB (UN=username)                                  Name the subschema.  NOS/BE requires an ID
                                                              parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE8 DESCRIPTION +                            Extract three fields upon FILE8.
? INV-NO IN-STOCK

 003--
? UNIVERSAL IS *                                              Establish a character (*) that marks a
                                                              character position to be ignored during
                                                              comparison testing.
 004--
? SPECIFY CHAIRS AS INV-NO EQ $CH****$                        Establish a name for convenient reference
                                                              to a condition.  The condition name in
                                                              this directive is CHAIRS.  The condition
                                                              is an inventory number (INV-NO) whose
                                                              first two characters are equal to CH.  The
                                                              universal character prohibits testing on
                                                              the last four character positions of INV-NO.
 005--
? PREPARE REPORT8 FROM FILE8                                  Prepare REPORT8 from the extracted file.

 006--
? RECORDING OFF                                               Terminate recording of SESS8.
      END OF SESSION SESS8

 --
? FORMAT REPORT8                                              Format a report named REPORT8.

 --
? SELECT 1 ON CHAIRS                                          Test each record to see if it satisfies
                                                              the condition in the SPECIFY directive.
 --
? DETAIL 1 IS DESCRIPTION IN COLUMN 3 +                       Supply the report line content.
? IN-STOCK IN COLUMN 21

 --
? TITLE AT LINE 1 IS $UNIVERSAL CHARACTER REPORT$ +           Supply the report title and set up two
? AT LINE 2 IS $    $ AT LINE 3 IS $ITEM$ IN COLUMN 3 +       column headings.
? IS $IN STOCK$ IN COLUMN 21 AT LINE 4 IS $    $

 --
? PERFORM SESS8                                               Retrieve and execute the directives in
                                                              SESS8.
 --
? END                                                         Terminate Query Update.


/ ROUTE,REPORT8,DC=PR                                         NOS/BE control statement would be the
                                                              following:  COMMAND- BATCH,REPORT8,PRINT
UNIVERSAL CHARACTER REPORT

  ITEM              IN STOCK

  ARM CHAIR         7
  DESK CHAIR        5
  SWIVEL CHAIR      9
                      **  END OF REPORT  REPORT8 **
```

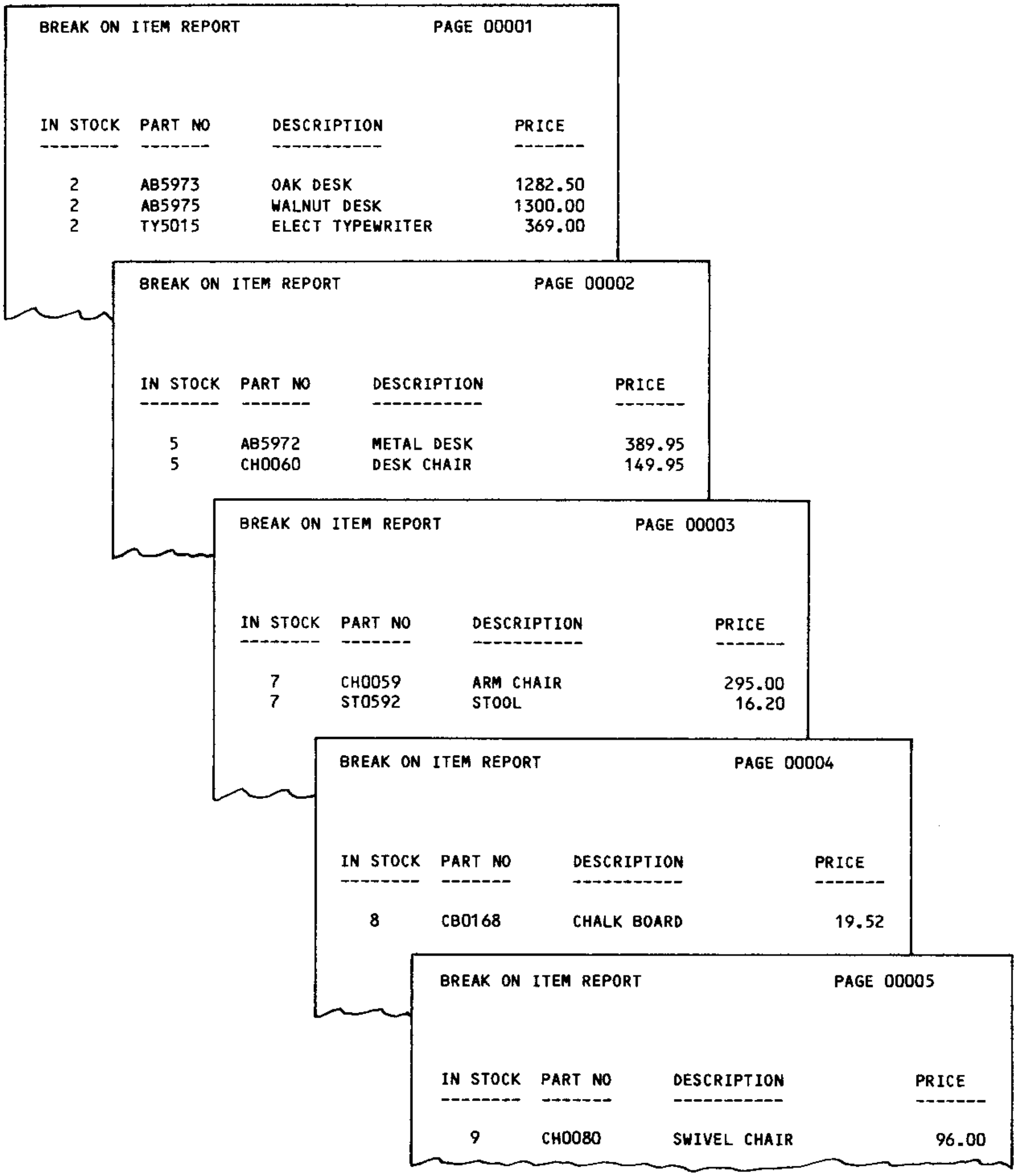
Figure 6-13. Sample Report Illustrating the Universal Character

```
/ QU                                                          Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/18  14.50.02

 --
? VERSION QUCAT (UN=username)                                 Attach the permanent file catalog.  NOS/BE
                                                              requires an ID parameter instead of the UN
                                                              parameter.

 --
? RECORDING SESS9                                             Begin recording SESS9.

 001--
? INVOKE QUSUB (UN=username)                                  Name the subschema.  NOS/BE requires an ID
                                                              parameter instead of the UN parameter.

 002--
? EXTRACT UPON FILE9 INV-NO DESCRIPTION +                     Extract four fields upon FILE9.
? IN-STOCK UNIT-PRICE

 003--
? REWIND FILE9 SORT9                                          Rewind FILE9 and the file to be used for
                                                              sorting operations.

 004--
? SORT FILE9 UPON SORT9 ON IN-STOCK                           Sort FILE9 on the IN-STOCK field.

 005--
? PREPARE REPORT9 FROM SORT9                                  Prepare REPORT9 from the sorted file.

 006--                                                        Terminate recording of SESS9.
? RECORDING OFF
     END OF SESSION SESS9

 --
? FORMAT REPORT9                                              Format a report named REPORT9.

 --
? DETAIL IS IN-STOCK IN COLUMN 1 +                            Supply the report line content.
? INV-NO IN COLUMN 11 +
? DESCRIPTION IN COLUMN 24 +
? UNIT-PRICE IN COLUMN 48

 --
? TITLE AT LINE 1 IS +                                        Supply the report title.
? $BREAK ON ITEM REPORT$ IN COLUMN 21

 --
? HEADING 0 AT LINE 5 BEYOND ON ALL PAGES +                   Specify the content and positioning of a
? IS $IN STOCK$ IN COLUMN 1 +                                 heading.  The level number 0 indicates
? IS $PART NO$ IN COLUMN 11 +                                 that the heading is to appear before any
? IS $DESCRIPTION$ IN COLUMN 24 +                             input data is processed.  The heading is to
? IS $PRICE$ IN COLUMN 48 +                                   appear on each page of the report.
? AT LINE 6 BEYOND ON ALL PAGES +
? IS $--------$ IN COLUMN 1 +
? IS $-------$ IN COLUMN 11 +
? IS $-----------$ IN COLUMN 24 +
? IS $-------$ IN COLUMN 48 +
? AT LINE 7 BEYOND ON ALL PAGES IS $    $

 --
? PAGE-NUMBER AT TITLE-LINE COLUMN 60                         Indicate vertical and horizontal position-
                                                              ing of page numbers.
 --
? BREAK 1 AND EJECT ON ITEM IN-STOCK                          Establish when the body of the report page
                                                              is to be interrupted for a heading or foot-
                                                              ing.  The break occurs when the content of
                                                              IN-STOCK changes.  A printer page-eject is
                                                              included.

 --
? PERFORM SESS9                                               Retrieve and execute SESS9.

 --
? END                                                         Terminate Query Update.
```

Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 1 of 3)

```
/ ROUTE,REPORT9,DC=PR
```

NOS/BE control statement would be the following:
COMMAND- BATCH,REPORT9,PRINT



```
BREAK ON ITEM REPORT                    PAGE 00001



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

    2     AB5973       OAK DESK               1282.50
    2     AB5975       WALNUT DESK            1300.00
    2     TY5015       ELECT TYPEWRITER        369.00
```

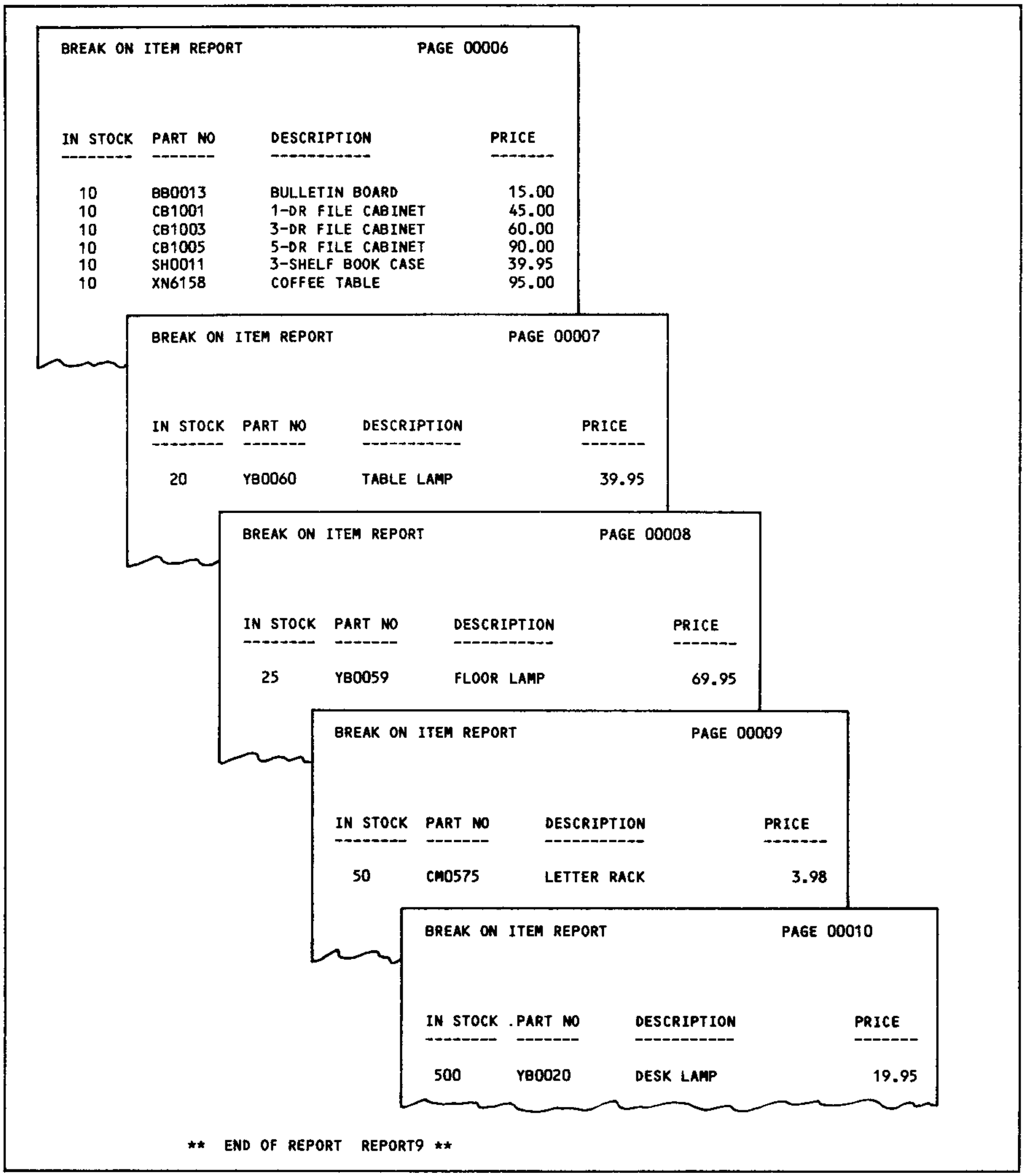
```
BREAK ON ITEM REPORT                    PAGE 00002



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

    5     AB5972       METAL DESK              389.95
    5     CH0060       DESK CHAIR              149.95
```

```
BREAK ON ITEM REPORT                    PAGE 00003



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

    7     CH0059       ARM CHAIR               295.00
    7     ST0592       STOOL                    16.20
```

```
BREAK ON ITEM REPORT                    PAGE 00004



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

    8     CB0168       CHALK BOARD              19.52
```

```
BREAK ON ITEM REPORT                    PAGE 00005



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

    9     CH0080       SWIVEL CHAIR             96.00
```

Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 2 of 3)



```
BREAK ON ITEM REPORT                    PAGE 00006



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

  10      BB0013       BULLETIN BOARD           15.00
  10      CB1001       1-DR FILE CABINET        45.00
  10      CB1003       3-DR FILE CABINET        60.00
  10      CB1005       5-DR FILE CABINET        90.00
  10      SH0011       3-SHELF BOOK CASE        39.95
  10      XN6158       COFFEE TABLE             95.00
```

```
BREAK ON ITEM REPORT                    PAGE 00007



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

  20      YB0060       TABLE LAMP               39.95
```

```
BREAK ON ITEM REPORT                    PAGE 00008



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

  25      YB0059       FLOOR LAMP               69.95
```

```
BREAK ON ITEM REPORT                    PAGE 00009



IN STOCK  PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

  50      CM0575       LETTER RACK               3.98
```

```
BREAK ON ITEM REPORT                    PAGE 00010



IN STOCK .PART NO      DESCRIPTION             PRICE
--------  -------      -----------            -------

 500      YB0020       DESK LAMP                19.95
```

```
** END OF REPORT  REPORT9 **
```

Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 3 of 3)

```
/ QU                                              Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110   82/05/18  15.37.30

 --
? VERSION QUCAT (UN=username)                     Attach the permanent file catalog.  NOS/BE
                                                  requires an ID parameter instead of a UN
                                                  parameter.
 --
? RECORDING SESS10                                Begin recording SESS10.

 001--
? INVOKE QUSUB (UN=username)                      Name the subschema.  NOS/BE requires an ID
                                                  parameter instead of a UN parameter.
 002--
? EXTRACT UPON FILE10 DESCRIPTION UNIT-PRICE      Extract two fields upon FILE10.

 003--
? REWIND FILE10 SORT10                            Rewind FILE10 and the file to be used for
                                                  sorting operations.
 004--
? SORT FILE10 UPON SORT10 ON DESCRIPTION          Sort the file on the description field.

 005--
? PREPARE REPT10 FROM SORT10                      Prepare REPT10 from the sorted file.

 006--
? RECORDING OFF                                   Terminate recording of SESS10.
     END OF SESSION SESS10

 --
? FORMAT REPT10                                   Format a report named REPT10.

 --
? DETAIL IS DESCRIPTION IN COLUMN 1 +             Supply the report line content.
? UNIT-PRICE IN COLUMN 19

 --
? TITLE AT LINE 1 IS $VERTICAL SECTIONS REPORT$ + Supply the report title.
? AT LINE 2 IS $    $

 --
? PAGE-SIZE IS 2 VERTICAL SECTIONS                Establish special placement of detail lines
                                                  on a report page.  Two sectional divisions
                                                  are to appear across the page.  The VERTICAL
                                                  option places successive entries in one
                                                  section and then continues to the second
                                                  section.
 --
? RECAP AT LINE 2 BEYOND IS +                     Generate a line of information at the com-
? $END OF VERTICAL SECTIONS REPORT$               pletion of a report page.  A literal expres-
                                                  sion is selected to appear two lines after
                                                  the last detail line.
 --
? PERFORM SESS10                                  Retrieve and execute SESS10.

 --
? END                                             Terminate Query Update.
```

Figure 6-15. Sample Report Illustrating Report Page Positioning (Sheet 1 of 2)

```
/ ROUTE,REPT10,DC=PR                                      NOS/BE control statement would be the
                                                          following:  COMMAND- BATCH,REPT10,PRINT


VERTICAL SECTIONS REPORT

ARM CHAIR           295.00                    OAK DESK          1282.50
BULLETIN BOARD       15.00                    STOOL               16.20
CHALK BOARD          19.52                    SWIVEL CHAIR        96.00
COFFEE TABLE         95.00                    TABLE LAMP          39.95
DESK CHAIR          149.95                    WALNUT DESK       1300.00
DESK LAMP            19.95                    1-DR FILE CABINET   45.00
ELECT TYPEWRITER    369.00                    3-DR FILE CABINET   60.00
FLOOR LAMP           69.95                    3-SHELF BOOK CASE   39.95
LETTER RACK           3.98                    5-DR FILE CABINET   90.00
METAL DESK          389.95

END OF VERTICAL SECTIONS REPORT
              **  END OF REPORT  REPT10  **
```

Figure 6-15. Sample Report Illustrating Report Page Positioning (Sheet 2 of 2)

```
COMMAND- ATTACH,PREFILE,ID=QUSER                            NOS/BE control statements.  The files con-
COMMAND- FILE,PREFILE,BT=C,RT=Z                             tain preface and summary information for
COMMAND- ATTACH,SUMFILE,ID=QUSER                            the report.
COMMAND- FILE,SUMFILE,BT=C,RT=Z

/ ATTACH,PREFILE,SUMFILE                                    NOS control statements.
/ FILE,PREFILE,BT=C,RT=Z
/ FILE,SUMFILE,BT=C,RT=Z

/ QU                                                        Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/18  17.16.34

 --
? VERSION QUCAT (UN=username)                               Attach the permanent file catalog.  NOS/BE
                                                            requires an ID parameter instead of a UN
                                                            parameter.
 --
? RECORDING SESS11                                          Begin recording SESS11.

 001--
? INVOKE QUSUB (UN=username)                                Name the subschema.  NOS/BE requires an ID
                                                            parameter instead of a UN parameter.

 002--
? EXTRACT UPON FILE11 DESCRIPTION UNIT-PRICE                Extract two fields upon FILE11.

 003--
? SPECIFY SALE-ITEMS AS UNIT-PRICE LT 50                    Specify a condition named SALE-ITEMS.  The
                                                            condition is a price (UNIT-PRICE) less
                                                            than $50 dollars.
 004--
? PREPARE REPT11 FROM FILE11                                Prepare REPT11 from the extracted file.
```

Figure 6-16. Sample Report Illustrating a Report Preface and Summary (Sheet 1 of 2)

```
 005--
? RECORDING OFF                                      Terminate recording of SESS11.
      END OF SESSION SESS11

 --
? FORMAT REPT11                                      Format a report named REPT11.

 --
? PREFACE IS TEXT FROM PREFILE                       Cause lines of text from permanent file
                                                     PREFILE to precede the current report.
 --
? SUMMARY IS TEXT FROM SUMFILE                       Cause lines of text from permanent file
                                                     SUMFILE to follow the current report.
 --
? SELECT 1 ON SALE-ITEMS                             Select the items that satisfy the SALE-
                                                     ITEMS condition.
 --
? DETAIL 1 IS DESCRIPTION IN COLUMN 1 +              Supply the report line content.
? UNIT-PRICE IN COLUMN 20

 --
? TITLE AT LINE 1 IS $PREFACE/SUMMARY REPORT$ +      Supply the report title and one column
? IN COLUMN 3 AT LINE 2 IS $    $ AT LINE 3 +        heading.
? IS $SALE ITEMS$ IN COLUMN 9 +
? AT LINE 4 IS $    $

 --
? PERFORM SESS11                                     Retrieve and execute the directives in
                                                     SESS11.
 --
? END                                                Terminate Query Update.

/ ROUTE,REPT11,CD=PR QU.                             NOS/BE control statement would be the
                                                     following:  COMMAND- BATCH,REPT11,PRINT


DISTRIBUTE TO ALL OUTLETS BY THE 16TH OF THE MONTH

  PREFACE/SUMMARY REPORT

        SALE ITEMS

BULLETIN BOARD       15.00
CHALK BOARD          19.52
1-DR FILE CABINET    45.00
LETTER RACK           3.98
3-SHELF BOOK CASE    39.95
STOOL                16.20
DESK LAMP            19.95
TABLE LAMP           39.95

SALE ENDS ON THE LAST DAY OF THE MONTH
              **  END OF REPORT  REPT11  **
```

Figure 6-16. Sample Report Illustrating a Report Preface and Summary (Sheet 2 of 2)

PREVIEWING REPORTS

A two-page sample printout can be produced for any report that has an existing directory. A data base file requires a directory created with the EXTRACT directive; a non-data-base file requires a directory created with the DESCRIBE directive. The sample report is produced with the PREVIEW directive.

When the PREVIEW directive is issued with no options, dummy values of X's and Y's are substituted for alphanumeric data and dummy values of 8's and 9's are substituted for numeric data. When the PREVIEW directive is issued with reference to a file name, actual data values are used.

Use of the PREVIEW directive is shown in figure 6-17. A report entitled PREVIEWED REPORT is previewed before it is prepared from FILE12. This application shows how a report format can be viewed and adjusted, if necessary, before preparation.

```
/ QU                                                        Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/19  10.22.37

 --
? INVOKE QUSUB (UN=username)                                Issue a sequence of directives that format
                                                            a three-column report.  Preview the report
                                                            before issuing the PREPARE directive to
                                                            ensure proper alignment.

 --
? EXTRACT UPON FILE12 INV-NO +
? DESCRIPTION UNIT-COST
      19 ACCESSES, 19 HITS, 19 IO-S

 --
? FORMAT REPT12

 --
? DETAIL IS INV-NO IN COLUMN 1 +
? DESCRIPTION IN COLUMN 8 +
? UNIT-COST IN COLUMN 26

 --
? TITLE AT LINE 1 IS $PREVIEWED REPORT$ +
? AT LINE 2 IS $    $

 --
? PREVIEW REPT12                                            Generate a sample report with dummy values.

 --
? OS,EDIT,REPT12                                            Display the previewed report.

PRINT,1

PREVIEWED REPORT

XXXXXX YYYYYYYYYYYYYYYYY 9999999                            X's and Y's represent alphabetic informa-
XXXXXX YYYYYYYYYYYYYYYYY 9999999                            tion; 9's represent numeric information.
XXXXXX YYYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYYY 9999999

 --
? END
              **  END OF REPORT  REPT12  **

**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2
```

Figure 6-17. Previewing a Report

THE REPORT UTILITY

All reports illustrated up to this point are shown being prepared by Query Update through the PREPARE directive. When a PREPARE directive is encountered, Query Update compiles internal tables and uses them to produce the report. The tables, which are collectively referred to as a report information table, serve the following purposes:

Describe the sequential file that is created as a result of an EXTRACT or DESCRIBE directive.

Associate the sequential file with the report layout that is established by the FORMAT directive.

Query Update compiles a report information table each time a report is prepared and discards the table after the report is generated.

Reports can be prepared for use by the Report Utility through the COMPILE directive. When a COMPILE directive is used in lieu of the PREPARE directive, Query Update compiles the report information table and stores it on a local file. When the local file is made permanent, reports can be subsequently generated by the following steps:

Attach the permanent file containing the compiled report information table.

Attach the sequential file.

Define the sequential file characters for CRM.

Issue a control statement call to the Report Utility.

When a report format is prepared in this manner, compilation of the report information table occurs only once; a new report information table has to be generated only when the report layout is changed. This approach results in savings in central processor time and central memory usage.

Use of the COMPILE directive is shown in figure 6-18. A sequential file named COMFILE is generated from data base file INVENTORY and the report information table is compiled on local file TBLFILE. Both files are made permanent for subsequent use by the Report Utility.

Use of the Report Utility is shown in figure 6-19. Two sets of control statements are shown. The first set uses COMFILE as the input data file; the second set uses non-data-base file INDFILE to obtain the same results.

```
COMMAND- REQUEST,COMFILE,*PF                                 NOS/BE control statements.
COMMAND- REQUEST,TBLFILE,*PF

/ DEFINE,COMFILE/CT=PU,M=W                                   NOS control statements.
/ DEFINE,TBLFILE/CT=PU,M=W

COMMAND- QU                                                  Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110    82/05/19  10.41.32

 --
  INVOKE QUSUB (UN=username)                                 Name the subschema.

 --
  EXTRACT UPON COMFILE INV-NO ON-ORDER                       Extract two fields from COMFILE.
      19 ACCESSES, 19 HITS, 19 IO-S

 --
  FORMAT COMRPT                                              Format a report named COMRPT.

 --
  DETAIL IS INV-NO IN COLUMN 1 +                             Supply the report line content.
  ON-ORDER IN COLUMN 12

 --
  TITLE AT LINE 1 IS $COMPILED REPORT$                       Supply the report title.
  AT LINE 2 IS $    $

 --
  COMPILE COMRPT UPON TBLFILE                                Generate the report information table on
                                                             local file TBLFILE.

PREFACES/SUMMARIES                                           Query Update prints a message that indi-
(NONE)                                                       cates prefaces and summaries are not asso-
                                                             ciated with COMRPT; no additional COMPILE
                                                             directives are needed.

 --
  END                                                        Terminate Query Update.
```

Figure 6-18. Compiling Report Specifications (Sheet 1 of 2)

```
**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

COMMAND- CATALOG,COMFILE,ID=QUSER,RP=999
COMMAND- CATALOG,TBLFILE,ID=QUSER,RP=999
```

NOS/BE control statements.

Figure 6-18. Compiling Report Specifications (Sheet 2 of 2)

```
COMMAND- ATTACH,COMFILE,ID=QUSER
COMMAND- FILE,COMFILE,RT=F
COMMAND- ATTACH,TBLFILE,ID=QUSER
COMMAND- REPORT,R=COMRPT,T=TBLFILE,I=COMFILE
COMMAND- BATCH,COMRPT,PRINT
```

ATTACH control statements are issued for sequential file COMFILE and report information table TBLFILE. A file control statement is issued for COMFILE to describe the file for CRM. A REPORT control statement is issued to call the Report Utility program to produce the report according to specifications in TBLFILE. The R parameter names the report; the T parameter names the report information table; and the I parameter names the input data file. Appropriate commands direct the finished report to the line printer.

```
/ ATTACH,COMFILE,TBLFILE
/ FILE,COMFILE,RT=F
/ REPORT,R=COMRPT,T=TBLFILE,I=COMFILE
/ ROUTE,COMRPT,DC=PR

COMPILED REPORT

AB5972        9
AB5973        0
AB5975        3
BB0013        8
CB0168        0
CB1001        0
CB1003        0
CB1005        0
CH0059        6
CH0060        0
CH0080        0
CM0575        6
SH0011        3
ST0592        0
TY5015        0
XN6158        0
YB0020        0
YB0059        0
YB0060        4
              **  END OF REPORT  COMRPT  **

COMMAND- ATTACH,INDFILE,ID=QUSER
COMMAND- FILE,INDFILE,RT=F
COMMAND- REPORT,R=COMRPT,T=TBLFILE,I=INDFILE
COMMAND- BATCH,COMRPT,PRINT
```

The data file can vary from run to run; only its characteristics must remain the same. These control statements attach INDFILE and use it as input data to the Report Utility.

```
/ ATTACH,INDFILE
/ FILE,INDFILE,RT=F
/ REPORT,R=COMRPT,T=TBLFILE,I=INDFILE
/ ROUTE,COMRPT,DC=PR

COMPILED REPORT

AB5972       25
CB0168       50
YB0020       35
BB0013       40
AB5975       22
CH0060       33
              **  END OF REPORT  COMRPT  **
```

Figure 6-19. Using the Report Utility

MULTIPLE FILES AND RELATIONSHIPS 7

One Query Update subschema can reference up to 64 separate data base files. This section of the guide shows the declaration of multiple files and how they can be individually accessed, and presents the relational data base capability in which files can be accessed simultaneously.

DECLARING MULTIPLE FILES

A subschema named MULTSUB is shown in figure 7-1. Unlike subschema QUSUB, which was presented in earlier sections, this subschema references multiple files. MULTSUB describes the original INVENTORY file along with three additional files named ORDERS, LINEITEMS, and OUTLET.

For ease of illustration, assume that information is stored in the new files. Figure 7-2 shows sample display operations for the new files and introduces the RETURN directive. RETURN allows files to be released to the operating system when they are no longer needed by Query Update. Returning a file frees a portion of central memory.

If the returned file is a data base area, its internal tables are released; the file itself is not released to the operating system. If the file is a non-data-base file, a RETURN releases the file to the operating system. There is no need to release a data base area specified through an INVOKE directive, which is not a temporary area, because INVOKE does not attach the file; directives that reference the area (such as DISPLAY) attach the file, reference it, and release it.

Two RETURN directives will release to the operating system a TEMPORARY area or an area specified through a CREATE directive. The first RETURN releases internal tables; the second performs a RETURN on a non-data-base file and releases it to the operating system.

The LINEITEMS file lends itself to arithmetic operations. Figure 7-3 is a sample report that illustrates subtotaling facilities. Two directives are introduced: MOVE and FOOTING.

MODIFYING MULTIPLE FILES

When modifying files that are data base areas in a multiple file subschema, the UPDATE directive tells Query Update which file is to be modified. The UPDATE directive is shown in figure 7-4.

```
IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS MULTSUB
DATA DIVISION.
AREA-NAME IN INVENTORY M IS W
INDEX IS INVIDX M IS W
ORGANIZATION IS INDEXED NEW
KEY IS INV-NO
KEY IS ALTERNATE BACK-ORDER DUPLICATES INDEXED
RECORD-NAME IS INV-REC
    02 INV-NO           PIC X(6)
    02 IN-STOCK         PIC Z(3)9
    02 BACK-ORDER       PIC Z(3)9
    02 ON-ORDER         PIC Z(3)9
    02 REORDER-PT       PIC Z(3)9
    02 UNIT-COST        PIC Z(4).99
    02 UNIT-PRICE       PIC Z(4).99
    02 DESCRIPTION      PIC X(17)
AREA-NAME IS ORDERS M IS W
INDEX IS ORDIDX M IS W
ORGANIZATION IS INDEXED NEW
KEY IS ORDER-NO
KEY IS ALTERNATE OUTLET-CODE DUPLICATES INDEXED
RECORD-NAME IS ORDER-REC
    02 ORDER-NO         PIC 9(6)
    02 ORDER-DATE       PIC X(10)
    02 TOT-ORDER        PIC Z(6).99
    02 OUTLET-CODE      PIC X(4)
AREA-NAME IS LINEITEMS M IS W
INDEX IS ITMIDX M IS W
ORGANIZATION IS ACTUAL NEW
KEY IS ACT-KEY
    KEY IS ALTERNATE ORDER-NO DUPLICATES INDEXED
    KEY IS ALTERNATE INV-NO DUPLICATES INDEXED
RECORD-NAME IS ITEM-REC
    02 ACT-KEY          PIC 9(5) USAGE IS INTEGER
    02 ORDER-NO         PIC 9(6)
    02 INV-NO           PIC X(6)
    02 QUANTITY         PIC Z(4)9
    02 ITEM-PRICE       PIC Z(4).99
    02 OUTLET-CODE      PIC X(4)
AREA-NAME IS OUTLET M IS W
ORGANIZATION IS DIRECT NEW
KEY IS OUTLET-CODE
RECORD-NAME IS OUTLET-REC
    02 OUTLET-CODE      PIC X(4)
    02 OUTLET-ADDR
        05 OUTLET-STREET PIC X(16)
        05 OUTLET-CITY   PIC X(16)
        05 OUTLET-STATE  PIC A(2)
        05 OUTLET-ZIP    PIC 9(5)
```

Figure 7-1. A Multiple File Subschema

```
/ QU                                                                   Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110           81/06/22  10.06.52

 --
? INVOKE MULTSUB (UN=username)                                         Name the subschema.  NOS/BE requires an
                                                                       ID parameter instead of a UN parameter.
 --
? DISPLAY ORDER-NO OF ORDER-REC +                                      Attach ORDERS and ORDIDX, display all
? ORDER-DATE TOT-ORDER +                                               fields of ORDERS, and release ORDERS and
? OUTLET-CODE OF ORDER-REC +                                           ORDIDX to the operating system.
 430001 07/01/81      2374.45 AZ43
 440001 07/02/81      8647.20 NV44
 460001 07/03/81       754.00 CA46
       3 ACCESSES, 3 HITS, 3 IO-S

 --
? RETURN ORDERS                                                        Release the internal tables of file
                                                                       ORDERS.
 --
? DISPLAY ORDER-NO OF ITEM-REC +                                       Attach LINEITEMS and ITMIDX, display
? INV-NO OF ITEM-REC QUANTITY +                                        all LINEITEMS, and release LINEITEMS
? ITEM-PRICE OUTLET-CODE OF ITEM-REC                                   and ITMIDX to the operating system.
 430001 AB5972      3  389.95 AZ43
 430001 TY5015      3  369.00 AZ43
 430001 CB0168      5   19.52 AZ43
 440001 AB5975      3 1300.00 NV44
 440001 CH0060      6  149.95 NV44
 440001 AB5973      3 1282.50 NV44
 460001 CB1003      7   60.00 CA46
 460001 BB0013      9   15.00 CA46
 460001 CM0575     50    3.98 CA46
       9 ACCESSES, 9 HITS, 9 IO-S

 --
? RETURN LINEITEMS                                                     Release the internal tables of file
                                                                       LINEITEMS.
 --
? DISPLAY OUTLET-CODE OF OUTLET-REC +                                  Attach OUTLET, display all fields of
? OUTLET-STREET OUTLET-CITY +                                          OUTLET, and release OUTLET to the
? OUTLET-STATE OUTLET-ZIP                                              operating system.
 AZ43 3440 N CENTRAL    PHOENIX          AZ 85015
 CA46 1000 W MAIN       SAN FRANCISCO    CA 94111
 NV44 1231 E SAHARA     LAS VEGAS        NV 89105
       3 ACCESSES, 3 HITS, 3 IO-S

 --
? EXTRACT UPON FILE1 OUTLET-STREET                                     Build a one-field subset of OUTLET and
       3 ACCESSES, 3 HITS, 3 IO-S                                      an associated directory.

 --
? RETURN FILE1                                                         Release the internal tables associated
                                                                       with file FILE1 and release FILE1 to the
                                                                       operating system.
 --
```

Figure 7-2. Querying Multiple Files

```
/ QU                                                             Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110         81/06/23  14.16.12

 --
? VERSION QUCAT (UN=username)                                    Attach the permanent catalog.

 --
? RECORDING MSESS                                                Begin recording session MSESS.

 001--
? INVOKE MULTSUB (UN=username)                                   Name the subschema.  NOS/BE requires an
                                                                 ID parameter instead of a UN parameter.
 002--
? EXTRACT UPON MFILE ORDER-NO OF ITEM-REC AS O +                 Extract 5 fields upon MFILE.  The AS
? INV-NO OF ITEM-REC AS I QUANTITY AS Q +                        option renames each field.
? ITEM-PRICE AS P OUTLET-CODE OF ITEM-REC AS C

 003--
? REWIND MFILE MSORT                                             Rewind MFILE and the file to be used
                                                                 for sorting operations.
 004--
? SORT MFILE UPON MSORT ON O                                     Sort the file on the ON-ORDER field,
                                                                 renamed O.
 005--
? DEFINE TOTAL BY $Z(5).99$ = Q * P                              Define a temporary item named TOTAL
                                                                 whose value is evaluated by Query
                                                                 Update.
 006--
? DEFINE SUBTOTAL BY $Z(6).99$                                   Define a temporary item named SUBTOTAL.

 007--
? DEFINE GRANDTOTAL BY $Z(7).99$                                 Define a temporary item named
                                                                 GRANDTOTAL.
 008--
? PREPARE MRPT FROM MSORT                                        Prepare MRPT from the sorted file.

 009--
? RECORDING OFF                                                  Terminate recording of MSESS.  Query
 END OF SESSION MSESS                                            Update acknowledges the end of the
                                                                 session.
 --
? FORMAT MRPT                                                    Format a report named MRPT.

 --
? EVALUATE BEFORE ANY DETAIL TOTAL                               Evaluate temporary item TOTAL before
                                                                 the DETAIL production step.
 --
? TITLE AT LINE 1 IS $SUBTOTAL REPORT$ IN COLUMN 16              Supply the report title.

 --
? DETAIL IS O IN COLUMN 1 I IN COLUMN 9 +                        Supply the report line content.  The
? Q IN COLUMN 15 P IN COLUMN 24 +                                new field names must be used.
? TOTAL IN COLUMN 33 C IN COLUMN 43

 --
? BREAK 1 ON ITEM O                                              Break when the content of field O (the
                                                                 renamed ORDER-NO field) changes.
 --
? HEADING O AT LINE 4 BEYOND IS $ORDER$ IN COLUMN 1 +            Specify the heading that is to appear
? IS $ITEM$ IN COLUMN 10 IS $QTY$ IN COLUMN 18 +                 before any input data is processed.
? IS $PRICE$ IN COLUMN 25 IS $TOTAL$ IN COLUMN 35 +
? IS $CODE$ IN COLUMN 43 AT LINE 5 BEYOND IS $    $
```

Figure 7-3. Sample Report Illustrating Subtotaling Operations (Sheet 1 of 2)

```
 --
? FOOTING 0 AT LINE 2 BEYOND IS +
? $GRAND TOTAL $ GRANDTOTAL IN COLUMN 19
```

The FOOTING directive shows the content and positioning of a footing. Level 0 indicates that this footing is associated with the end of data.

```
 --
? FOOTING 1 AT LINE 2 BEYOND IS +
? $SUBTOTAL $ SUBTOTAL IN COLUMN 23 AT LINE 3 BEYOND IS $    $
```

FOOTING 1 is associated with BREAK 1. This footing will appear when the content of field 0 (the renamed ORDER-NO field) changes.

```
 --
? MOVE BEFORE ANY DETAIL SUBTOTAL + TOTAL TO SUBTOTAL +
? AND GRANDTOTAL + TOTAL TO GRANDTOTAL
```

The MOVE directive moves values to data names. Values are moved before the DETAIL production step.

```
 --
? MOVE AFTER FOOTING 1 0 TO SUBTOTAL
```

Set the content of SUBTOTAL to zero after a level 1 footing is executed.

```
 --
? PERFORM MSESS

 --
? END
```

Terminate Query Update.

```
/ ROUTE,MRPT,DC=PR
```

Print the report. Under NOS/BE the control statement would be COMMAND- BATCH,MRPT,PRINT

```
                        SUBTOTAL REPORT


ORDER     ITEM     QTY    PRICE      TOTAL    CODE

430001   AB5972     3     389.95    1169.85   AZ43
430001   TY5015     3     369.00    1107.00   AZ43
430001   CB0168     5      19.52      97.60   AZ43

                        SUBTOTAL    2374.45

440001   AB5975     3    1300.00    3900.00   NV44
440001   CH0060     6     149.95     899.70   NV44
440001   AB5973     3    1282.50    3847.50   NV44

                        SUBTOTAL    8647.20

460001   CB1003     7      60.00     420.00   CA46
460001   BB0013     9      15.00     135.00   CA46
460001   CM0575    50       3.98     199.00   CA46

                        SUBTOTAL     754.00


                        GRAND TOTAL   11775.65

                              **  END OF REPORT  MRPT    **
```

Figure 7-3. Sample Report Illustrating Subtotaling Operations (Sheet 2 of 2)

```
/ QU                                                              Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110         81/06/24  09.23.10

 --
? INVOKE MULTSUB (UN=username)                                    Name the subschema.  NOS/BE requires an
                                                                  ID parameter instead of a UN parameter.
 --
? UPDATE OUTLET                                                   Name the file.

 --
? MODIFY USING OUTLET-CODE SETTING OUTLET-STREET +                Modify two fields.
? OUTLET-ZIP
 >> $CA46$ $210 E MARKET$ 94115
 >> *END                                                          Terminate the MODIFY directive.
    1 ACCESSES, 1 HITS, 2 IO-S

 --
? DISPLAY KEY $CA46$ OUTLET-STREET OUTLET-ZIP                     Verify that outlet CA46 has a new
 210 E MARKET     94115                                           address.
    1 ACCESSES, 1 HITS, 1 IO-S

 --
? UPDATE ORDERS                                                   Name another file.

 --
? STORE SETTING ORDER-NO ORDER-DATE +                             Store a new record.
? TOT-ORDER OUTLET-CODE
 >> 450001 $07/08/81$ 398.95 $WA45$
 >> *END                                                          Terminate the STORE directive.
    1 ACCESSES, 1 HITS, 1 IO-S

 --
? END                                                             Terminate Query Update.
```

Figure 7-4. Modifying Multiple Files

ESTABLISHING RELATIONSHIPS

The multiple file subschema MULTSUB presented in the preceding paragraphs described several files. Each file could be accessed for query, update, or report operations -- but data could be retrieved from only one file at a time.

Inherent relationships exist among INVENTORY, ORDERS, LINEITEMS, and OUTLET. Some data items appear in more than one file. For example, data item INV-NO appears in INVENTORY and LINEITEMS, and data item OUTLET-CODE appears in ORDERS, LINEITEMS, and OUTLET. A logical connection can be established within the subschema to allow two or more files to be queried as one file.

A relational subschema is shown in figure 7-5. MULTSUB, renamed RELSUB, is expanded to include a Relation Division that specifies two relations: CONTEST and ACTIVITY-CHECK.

```
IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS RELSUB
DATA DIVISION.
AREA-NAME IS INVENTORY M IS W
INDEX IS INVIDX M IS W
ORGANIZATION IS INDEXED NEW
KEY IS INV-NO
KEY IS ALTERNATE BACK-ORDER DUPLICATES INDEXED
RECORD-NAME IS INV-REC
    02 INV-NO          PIC X(6)
    02 IN-STOCK        PIC Z(3)9
    02 BACK-ORDER      PIC Z(3)9
    02 ON-ORDER        PIC Z(3)9
    02 REORDER-PT      PIC Z(3)9
    02 UNIT-COST       PIC Z(4).99
    02 UNIT-PRICE      PIC Z(4).99
    02 DESCRIPTION     PIC X(17)
AREA-NAME IS ORDERS M IS W
INDEX IS ORDIDX M IS W
ORGANIZATION IS INDEXED NEW
KEY IS ORDER-NO
KEY IS ALTERNATE OUTLET-CODE DUPLICATES INDEXED
RECORD-NAME IS ORDER-REC
    02 ORDER-NO        PIC 9(6)
    02 ORDER-DATE      PIC X(10)
    02 TOT-ORDER       PIC Z(6).99
    02 OUTLET-CODE     PIC X(4)
AREA-NAME IS LINEITEMS M IS W
INDEX IS ITMIDX M IS W
ORGANIZATION IS ACTUAL NEW
KEY IS ACT-KEY
KEY IS ALTERNATE ORDER-NO DUPLICATES INDEXED
KEY IS ALTERNATE INV-NO DUPLICATES INDEXED
RECORD-NAME IS ITEM-REC
    02 ACT-KEY         PIC 9(5) USAGE IS INTEGER
    02 ORDER-NO        PIC 9(6)
    02 INV-NO          PIC X(6)
    02 QUANTITY        PIC Z(4)9
    02 ITEM-PRICE      PIC Z(4).99
    02 OUTLET-CODE     PIC X(4)
AREA-NAME IS OUTLET M IS W
ORGANIZATION IS DIRECT NEW
KEY IS OUTLET-CODE
RECORD-NAME IS OUTLET-REC
    02 OUTLET-CODE     PIC X(4)
    02 OUTLET-ADDR
        05 OUTLET-STREET  PIC X(16)
        05 OUTLET-CITY    PIC X(16)
        05 OUTLET-STATE   PIC A(2)
        05 OUTLET-ZIP     PIC 9(5)
RELATION DIVISION.
RELATION-NAME IS CONTEST
JOIN WHERE OUTLET-CODE OF ORDER-REC EQ OUTLET-CODE OF OUTLET-REC
RESTRICT ORDER-REC WHERE TOT-ORDER GE 5000
RELATION-NAME IS ACTIVITY-CHECK
JOIN WHERE INV-NO OF INV-REC EQ INV-NO OF ITEM-REC
OUTLET-CODE OF ITEM-REC EQ OUTLET-CODE OF OUTLET-REC
RESTRICT OUTLET-REC WHERE OUTLET-CITY EQ "PHOENIX"
```

Figure 7-5. A Relational Subschema

The CONTEST relation joins the ORDERS and OUTLET areas. Data from both files can be returned whenever these two files are traversed by Query Update. The files are traversed when at least one data item in each file is referenced in a Query Update transmission. The CONTEST relation imposes one restriction on the ORDERS file. Data is returned only when the TOT-ORDER field contains a value greater than or equal to 5000.

The ACTIVITY-CHECK relation joins the INVENTORY, LINEITEMS, and OUTLET areas. Data can be returned from these files whenever they are traversed by Query Update. The ACTIVITY-CHECK relation imposes one restriction on the OUTLET file. Data is returned only when the OUTLET-CITY field contains the value PHOENIX.

The relationships are shown in the diagram in figure 7-6. The solid arrow traces CONTEST. The shaded arrows trace ACTIVITY-CHECK.

QUERYING RELATED FILES

Related files are queried in the same manner as individual files. When queries are issued against files joined in a relation, Query Update searches the joined files and returns the qualifying data as one projected record. Query Update determines which relation to use by the data names specified in a directive.

Interactive query begins with the INVOKE directive, which makes all areas and relations described in the subschema available. Sample queries involving relations are shown in figure 7-7.

RECONCILING AMBIGUITIES

Ambiguities can exist if the same files are joined in different relations and the data items named in the query are from these files only. The VIA directive specifies the relation that should be followed when such ambiguities exist. The directive can be entered alone as a single transmission or included in a query transmission. The VIA directive is illustrated in figure 7-8.

MODIFYING RELATED FILES

Relations are limited to read-only operations. When files joined in a relationship are to be modified, the modification is made on each individual file, one at a time.

Care should be taken when joined files are modified. If, for example, an inventory number (INV-NO) in the INVENTORY file is changed, appropriate changes must be reflected in the LINEITEMS file to maintain the integrity of projected records.

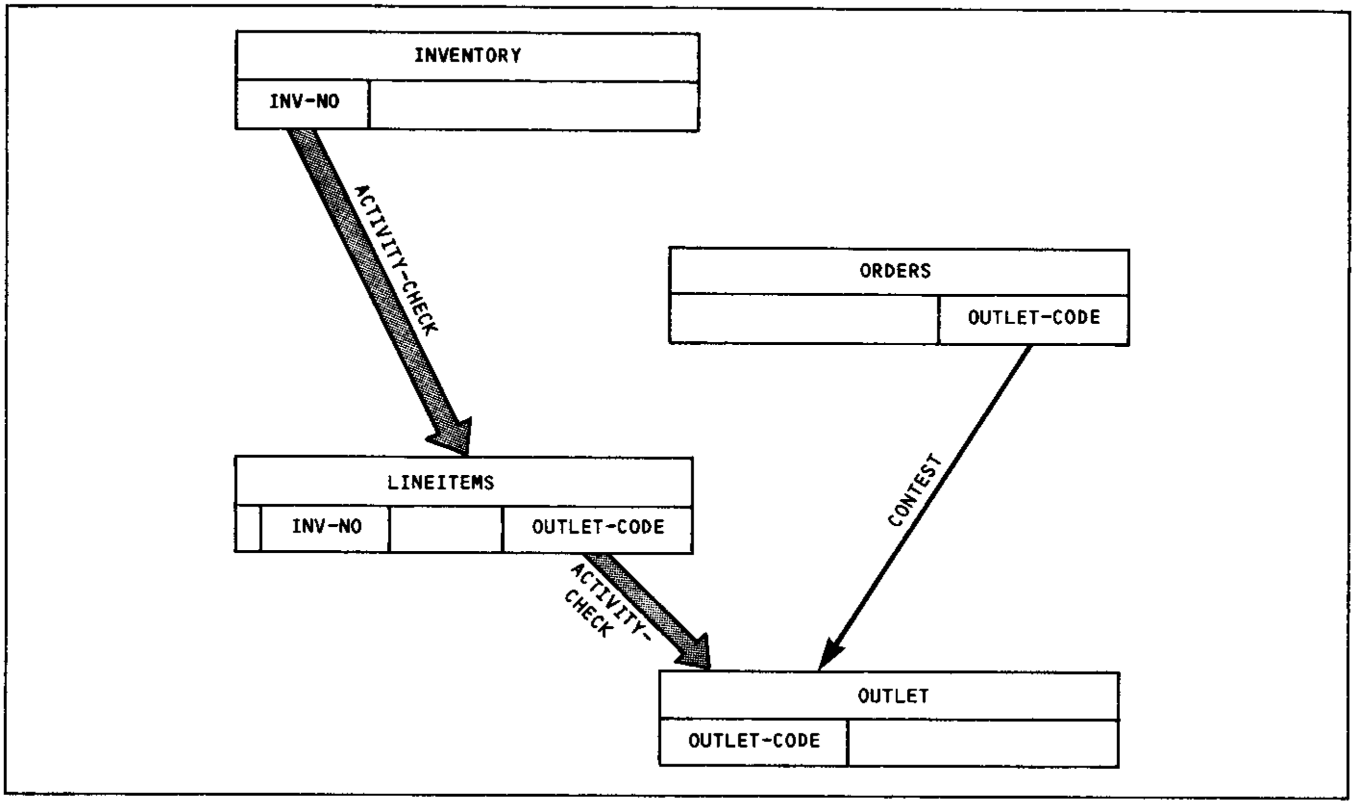


Figure 7-6. The Relationships Between Areas

```
/ QU                                                                  Call Query Update.

 QUERY UPDATE 3.4 SYSEDIT-82110         81/06/26  13.30.26

 --
? INVOKE RELSUB (UN=username)                                         Name the subschema.  NOS/BE
                                                                      requires an ID parameter instead
                                                                      of a UN parameter.

 --
? DISPLAY TOT-ORDER OUTLET-CITY                                       Display across the ORDERS and
                                                                      OUTLET files.  Query Update auto-
                                                                      matically ses the CONTEST relation.

 8647.20 LAS VEGAS                                                    Query Update imposes the restric-
       1 ACCESSES, 1 HITS, 4 IO-S                                     tion on TOT-ORDER.

 --
? DISPLAY $THE ORDER THAT WAS EQUAL TO +                              Clarify the display with literals.
? OR GREATER THAN $$5000 WAS$ TOT-ORDER +
? $*****THE WINNER IS$ OUTLET-CITY

 THE ORDER THAT WAS EQUAL TO OR GREATER THAN $5000 WAS   8647.20
 *****  THE WINNER IS LAS VEGAS
       1 ACCESSES, 1 HITS, 4 IO-S

 --
? DISPLAY DESCRIPTION QUANTITY OUTLET-CITY                            Display across the INVENTORY,
 METAL DESK              3 PHOENIX                                    LINEITEMS, and OUTLET files.  Query
 CHALK BOARD             5 PHOENIX                                    Update automatically uses the
 ELECT TYPEWRITER        3 PHOENIX                                    ACTIVITY-CHECK relation.  Query
       19 ACCESSES, 3 HITS, 35 IO-S                                   Update imposes the restriction on
                                                                      OUTLET-CITY.
 --
? DISPLAY DESCRPTION INV-NO OF ITEM-REC +                             This display references INV-NO,
? OUTLET-CITY                                                         which is not unique.  The data
 METAL DESK          AB5972 PHOENIX                                   item must be qualified.
 CHALK BOARD         CB0168 PHOENIX
 ELECT TYPEWRITER    TY5015 PHOENIX
       19 ACCESSES, 3 HITS, 35 IO-S

 --
? EXTRACT UPON RELFILE DESCRIPTION QUANTITY +                         Prepare a report using the
? ITEM-PRICE OUTLET-CITY                                              ACTIVITY-CHECK relation.  The
       19 ACCESSES, 3 HITS, 35 IO-S                                   report is concerned only with the
                                                                      Phoenix outlet only because of the
 --                                                                   restriction on OUTLET-CITY.
? FORMAT RELRPT

 --
? TITLE AT LINE 1 IS +
? $PHOENIX ACTIVITY REPORT$ IN COLUMN 11 +
? AT LINE 2 IS $    $

 --
? DETAIL IS DESCRIPTION IN COLUMN 1 +
? QUANTITY IN COLUMN 20 +
? ITEM-PRICE IN COLUMN 26 +
? OUTLET-CITY IN COLUMN 35

 --
? PREPARE RELRPT FROM RELFILE

 --
? END                                                                 Terminate Query Update.
```

Figure 7-7. Querying a Relation (Sheet 1 of 2)

```
 **CAUTION**
 DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ ROUTE,RELRPT,DC=PR                                             Print the report.  Under NOS/BE the
                                                                 control statement would be COMMAND-
                                                                 BATCH,RELRPT,PRINT.


          PHOENIX ACTIVITY REPORT

 METAL DESK               3  389.95  PHOENIX
 CHALK BOARD              5   19.52  PHOENIX
 ELECT TYPEWRITER         3  369.00  PHOENIX
                        **  END OF REPORT  RELRPT  **
```

Figure 7-7. Querying a Relation (Sheet 2 of 2)

```
 --
? INVOKE RELSUB (UN=username)

 --
? VIA ACTIVITY-CHECK                                       VIA can be a separate transmission.

 --
? DISPLAY ...                                              Where the ellipses indicate data names from
                                                           areas INVENTORY and LINE ITEMS, which are
                                                           joined in relation ACTIVITY-CHECK.

 --
? VIA ACTIVITY-CHECK DISPLAY ...                           VIA can precede a query.

 --
? DISPLAY ... VIA ACTIVITY-CHECK                           VIA can follow a query.
```

Figure 7-8. The VIA Directive

A number of special Query Update utilities are available to assist programmers by providing access to and clarification of stored information and offering subscripting capabilities.

Query Update uses a large amount of stored reference information within its processing environment. Typical information includes subschema descriptions and permanent catalog contents. This type of information is made available to the programmer through several informative directives.

Subscripting operations can be performed within Query Update. Elements within an array or repeating group can be accessed by supplying an integer, data name, or figurative subscript.

INFORMATIVE DIRECTIVES

Query Update includes the following four directives that supply information that can be helpful to the new user:

DIAGNOSTIC

Expands diagnostic messages to aid in debugging

EXHIBIT

Displays current information related to permanent and temporary data names, relations, reports, and sessions

HELP

Provides descriptions of directives and explains diagnostic messages

NOTE

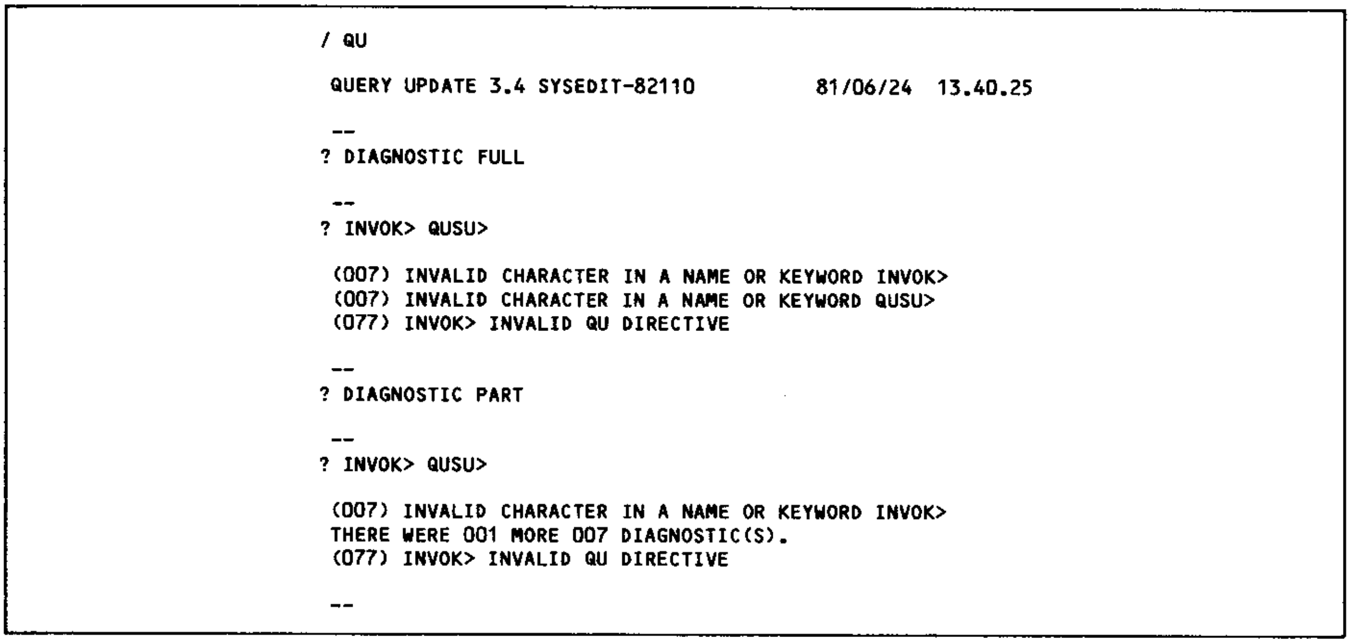
Allows comments to be included in transmissions

These directives can be issued at any time without affecting the operation in progress. Each is described in the following paragraphs.

DIAGNOSTIC

Query Update analyzes the syntax of a directive after it is transmitted. When errors appear in the input string, Query Update prints a three-digit number followed by a diagnostic message. Full or partial diagnostics can be displayed. When diagnostics are full, Query Update displays all diagnostic messages. When diagnostics are partial (default), Query Update does not display consecutive duplicate messages but indicates instead the number of times the diagnostic occurred.

Use of the DIAGNOSTIC directive with both the FULL and PART options is shown in figure 8-1. In both cases, the INVOKE directive includes an invalid character. When the FULL option is in effect, message number 007 is repeated. When the PART option is in effect, consecutive appearances of message number 007 are not repeated.



```
/ QU

 QUERY UPDATE 3.4 SYSEDIT-82110          81/06/24  13.40.25

 --
? DIAGNOSTIC FULL

 --
? INVOK> QUSU>

 (007) INVALID CHARACTER IN A NAME OR KEYWORD INVOK>
 (007) INVALID CHARACTER IN A NAME OR KEYWORD QUSU>
 (077) INVOK> INVALID QU DIRECTIVE

 --
? DIAGNOSTIC PART

 --
? INVOK> QUSU>

 (007) INVALID CHARACTER IN A NAME OR KEYWORD INVOK>
 THERE WERE 001 MORE 007 DIAGNOSTIC(S).
 (077) INVOK> INVALID QU DIRECTIVE

 --
```

Figure 8-1. The DIAGNOSTIC Directive

EXHIBIT

Query Update deals with large amounts of information. Any number of subschemas can be stored in the system, and each subschema can reference up to 64 files. Any number of permanent file catalogs can also be stored in the system, and no limitation is imposed on the number of recorded sessions and report formats that can be retained. All information referenced by and contained within these files can be made available through the EXHIBIT directive.

Use of the EXHIBIT directive is shown in figure 8-2. Selected information concerning the subschema and the QUCAT catalog is displayed.

```
--
? INVOKE RELSUB (UN=username)                           Name the subschema.  NOS/BE requires an ID
                                                        parameter instead of a UN parameter.
--
? EXHIBIT                                               Exhibit subschema information.
 MAXIMUM TRANSMISSION LENGTH     1030
 TL OF CATALOG FILE    1030
 SEPARATOR  $
 UNIVERSAL OFF
 MAX NUMBER OF LINES  060
 MAX NUM. OF COLUMNS  136
 MAX NO. OF SECTIONS  010
 MAX IMAGES PER PAGE  004
 AREA NAME(S):
    INVENTORY
    ORDERS
    LINEITEMS
    OUTLET
 SUBSCHEMA NAME = RELSUB
 SUBSCHEMA LIBRARY NAME = RELSUB
     UN = username

--
? EXHIBIT RELATION                                      Exhibit relation information.
 CONTEST RELATES THE RECORDS:
    ORDER-REC IN ORDERS
    OUTLET-REC IN OUTLET
 ACTIVITY-CHECK RELATES THE RECORDS:
    INV-REC IN INVENTORY
    ITEM-REC IN LINEITEMS
    OUTLET-REC IN OUTLET

--
? EXHIBIT INVENTORY                                     Exhibit information for area INVENTORY.
 RECORD NAME IS     INV-REC
 KEY IS    INV-NO
 ALT KEY  BACK-ORDER
 AREA PF NAME = INVENTORY
     UN = username
 INDEX PF NAME = INVIDX
     M  = W

--
? EXHIBIT DESCRIPTION                                   Exhibit information for elementary item
 DESCRIPTION OF INV-REC                                 DESCRIPTION.
 TYPE     CHAR ITEM  PIC SIZE 0017

--
? VERSION QUCAT (UN=username)                           Attach the permanent catalog.  NOS/BE
                                                        requires an ID parameter instead of a UN
                                                        parameter.

--
? EXHIBIT REPORTS REPORT7 TITLE                         Exhibit the TITLE directive for report
         TITLE AT LINE 1 IS $TEMPORARY ITEM             REPORT7.
REPORT$ IN COLUMN 6 AT LINE 2 IS $    $ AT
LINE 3 IS $ITEM$ IN TAB 1 IS $PROFIT MARGIN$
IN TAB 2 AT LINE 4 IS $    $

--
? EXHIBIT SESSIONS SESS8 001 TO 003                     Exhibit transmission IDs 001 through 003
 1      INVOKE QSUB (UN=username)                       for session SESS8.
 2      EXTRACT UPON FILE8 DESCRIPTION INV-NO IN-STOCK
 3      UNIVERSAL IS *
```

Figure 8-2. The EXHIBIT Directive

HELP

Query Update provides over 60 directives and issues over 500 diagnostic messages. Definitions of the directives and detailed explanations of the diagnostic messages can be made available through the HELP directive.

Use of the HELP directive is shown in figure 8-3. Selected information concerning directives and diagnostics is displayed.

```
 --
? HELP PREFACE                                             Ask for the definition of the PREFACE
                                                           directive.
 PREFACE  CAUSES LINES OF TEXT OR ANOTHER REPORT TO
 PRECEDE  THE CURRENT REPORT.

 --
? HELP SELLECT                                             Ask for the definition of the SELECT
                                                           directive and misspell its name.
 (212) -SELLECT- IS NOT VALID FOLLOWING HELP.
 (029) ERRONEOUS DIRECTIVE AND REST OF TRANSMISSION IGNORED

 --
? HELP 212                                                 Ask for a more detailed explanation
                                                           of diagnostic message 212.
 THE ONLY VALID HELP PARAMETERS ARE EITHER A DIRECTIVE NAME
 OR A DIAGNOSTIC NUMBER.

 --
? HELP SELECT                                              Ask for the definition of the SELECT
                                                           directive.
 SELECT STATES CRITERIA FOR SELECTION OF DETAIL
 SPECIFICATIONS.

 --
? HELP                                                     Ask for a list of available
                                                           directives.
 THE DIRECTIVES IMPLEMENTED IN THIS RELEASE ARE
     ACCESS           ALTER            BREAK
     COMPILE          CREATE
     DATE             DEFINE           DELETE
     DESCRIBE         DETAIL           DIAG
     DISPLAY          DUPLICATE        END
     ERASE            EVALUATE
     EXECUTE          EXHIBIT          EXTRACT
     FOOTING          FORMAT           HEADING
     HELP             IF               INSERT
     INVOKE
     MODIFY           MOVE             NOTE
     OS               PAGE-NUMBER
     PAGE-SIZE        PERFORM          PREFACE
     PREPARE          PREVIEW          RECAP
     PRINT
     RECORDING        RECOVERY         REMOVE
     RETURN           REWIND           SELECT
     SEPARATOR        SORT             SPECIFY
     STOP             STORE            SUMMARY
     TABS
     TIME             TITLE            UNIVERSAL
     UPDATE           USE              VERIFY
     VERSION          VETO             VIA
     WRITE

 --
```

Figure 8-3. The HELP Directive

NOTE

Notes and comments can be useful, particularly when recorded sessions and report formats must be maintained by someone other than the originator. Notes can be incorporated through the NOTE directive as complete transmissions or can be appended as the last portion of a transmission containing other directives. All characters between keyword NOTE and the end of transmission are accepted as comments.

Use of the NOTE directive is shown in figure 8-4. Two notes are recorded and exhibited.

```
 --
? VERSION QUCAT (UN=username)

 --
? RECORDING SESS13

 001--
? INVOKE QUSUB (UN=username)

 002--
? NOTE THIS REPORT EXTRACTS DATA FROM ONE FIELD

 003--
? EXTRACT UPON FILE13 DESCRIPTION NOTE THIS IS +
? A 17-CHARACTER ALPHANUMERIC FIELD

 004--
? PREPARE REPT13 FROM FILE13

 005--
? RECORDING OFF
       END OF SESSION SESS13

 --
? FORMAT RPT13

 --
? DETAIL IS DESCRIPTION IN COLUMN 1

 --
? EXHIBIT SESSION SESS13
  1      INVOKE QUSUB (UN=username)
  2      NOTE THIS REPORT EXTRACTS DATA FROM ONE FIELD
  3      EXTRACT UPON FILE13 DESCRIPTION NOTE THIS IS A 17-CHARACTER A
LPHANUMERIC FIELD
  4      PREPARE REPT13 FROM FILE13

 --
```

NOS/BE requires an ID parameter instead of a UN parameter.

Insert a note as a complete transmission.

Append a note to a transmission.

Notes are printed through the EXHIBIT directive.

Figure 8-4. The NOTE Directive

SUBSCRIPTING

Subscripting operations can be performed when a series of data items is referenced by one data name. The data items can be represented as repeating group or elementary data items in the subschema, or as an array entered through the DEFINE directive. The subscript can be an integer, the data name of an item containing an integer, or one of four figurative subscripts: ALL, LAST, NEXT, or ANY. Figurative subscript ANY is restricted to the IF, BREAK, SELECT, SPECIFY, and PERFORM UNTIL directives. Figurative subscript NEXT is restricted to the MOVE and STORE SETTING directives.

An application illustrating the subscripting capability is shown in figure 8-5. The subschema contains a repeating elementary item named CHILD. The CHILD matrix is constructed by including figurative subscript NEXT in the STORE SETTING directive that creates the sample data base. The display operations that follow illustrate subscripts specified as integers and as figurative constants ALL and LAST.

```
IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS PERS
DATA DIVISION.
AREA-NAME IS PERSONNEL M IS W
ORGANIZATION IS INDEXED NEW
KEY IS SSN
RECORD-NAME IS PERSRECORD
    02  SSN             PIC X(9)
    02  EMPNAME         PIC X(20)
    02  KIDSNO          PIC 9
    02  CHILD           PIC X(10) OCCURS 0 TO 9 TIMES
                        DEPENDING ON KIDSNO


/ QU

 QUERY UPDATE 3.4 SYSEDIT-82110       04/25/82  14.43.02

 --
? CREATE PERSONNEL OF PERS (UN=username)

 --
? STORE SETTING SSN EMPNAME CHILD(NEXT) CHILD(NEXT) CHILD(NEXT)
 >> $111223333$ $JOHN R. DOE$ $JEFF$ $LARRY$ $CEDRIC$
 >> $222334444$ $JACK C. JONES$
 >> $333445555$ $GIL F. GULLES$ $ALICE$ $ERICA$
 >> $444556666$ $ABE A. ABBOTT$ $JONAS$
 >> *END
       4 ACCESSES, 4 HITS, 4 IO-S

 --
? DISPLAY KIDSNO CHILD(ALL)
 3 JEFF       LARRY      CEDRIC
 0
 2 ALICE      ERICA
 1 JONAS
       4 ACCESSES, 4 HITS, 4 IO-S

 --
? IF KIDSNO GE 1 DISPLAY SSN KIDSNO CHILD(LAST)
 111223333 3 CEDRIC
 333445555 2 ERICA
 444556666 1 JONAS
       4 ACCESSES, 3 HITS, 4 IO-S
```

Figure 8-5. Subscripting (Sheet 1 of 2)

```
 --
? DISPLAY CHILD(1) CHILD(2) CHILD(3)
 JEFF       LARRY      CEDRIC
 (941) SUBSCRIPT OUT OF BOUNDS
 THERE WERE 002 MORE 941 DIAGNOSTIC(S).
       4 ACCESSES, 4 HITS, 4 IO-S

 --
? DISPLAY PERSRECORD NOTE - THIS IS TO SEE THE COMPLETE RECORD
 (209) REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT
 111223333JOHN R. DOE         3JEFF      LARRY     CEDRIC
 222334444JACK C. JONES       0
 333445555GIL F. GULLES       2ALICE     ERICA
 444556666ABE A. ABBOTT       1JONAS
       4 ACCESSES, 4 HITS, 4 IO-S

 --
? END
```

Figure 8-5. Subscripting (Sheet 2 of 2)

STANDARD CHARACTER SETS A

Control Data operating systems offer the following variations of a basic character set:

CDC 64-character set

CDC 63-character set

ASCII 64-character set

ASCII 63-character set

The set in use at a particular installation was specified when the operating system was installed.

Graphic character representation appearing at a terminal or printer depends on the installation character set and the terminal type. Characters shown in the CDC Graphic column of the standard character set table (table A-1) are applicable to BCD terminals; ASCII graphic characters are applicable to ASCII-CRT and ASCII-TTY terminals.

Standard collating sequences for the two printer character sets are shown in tables A-2 and A-3.

TABLE A-1. STANDARD CHARACTER SETS

Display Code (octal)	CDC			ASCII		
	Graphic	Hollerith Punch (026)	External BCD Code	Graphic Subset	Punch (029)	Code (octal)
00[†]	: (colon)[††]	8-2	00	: (colon)[††]	8-2	072
01	A	12-1	61	A	12-1	101
02	B	12-2	62	B	12-2	102
03	C	12-3	63	C	12-3	103
04	D	12-4	64	D	12-4	104
05	E	12-5	65	E	12-5	105
06	F	12-6	66	F	12-6	106
07	G	12-7	67	G	12-7	107
10	H	12-8	70	H	12-8	110
11	I	12-9	71	I	12-9	111
12	J	11-1	41	J	11-1	112
13	K	11-2	42	K	11-2	113
14	L	11-3	43	L	11-3	114
15	M	11-4	44	M	11-4	115
16	N	11-5	45	N	11-5	116
17	O	11-6	46	O	11-6	117
20	P	11-7	47	P	11-7	120
21	Q	11-8	50	Q	11-8	121
22	R	11-9	51	R	11-9	122
23	S	0-2	22	S	0-2	123
24	T	0-3	23	T	0-3	124
25	U	0-4	24	U	0-4	125
26	V	0-5	25	V	0-5	126
27	W	0-6	26	W	0-6	127
30	X	0-7	27	X	0-7	130
31	Y	0-8	30	Y	0-8	131
32	Z	0-9	31	Z	0-9	132
33	0	0	12	0	0	060
34	1	1	01	1	1	061
35	2	2	02	2	2	062
36	3	3	03	3	3	063
37	4	4	04	4	4	064
40	5	5	05	5	5	065
41	6	6	06	6	6	066
42	7	7	07	7	7	067
43	8	8	10	8	8	070
44	9	9	11	9	9	071
45	+	12	60	+	12-8-6	053
46	-	11	40	-	11	055
47	*	11-8-4	54	*	11-8-4	052
50	/	0-1	21	/	0-1	057
51	(	0-8-4	34	(	12-8-5	050
52	)	12-8-4	74	)	11-8-5	051
53	$	11-8-3	53	$	11-8-3	044
54	=	8-3	13	=	8-6	075
55	blank	no punch	20	blank	no punch	040
56	, (comma)	0-8-3	33	, (comma)	0-8-3	054
57	. (period)	12-8-3	73	. (period)	12-8-3	056
60	≡	0-8-6	36	#	8-3	043
61	[	8-7	17	[	12-8-2	133
62	]	0-8-2	32	]	11-8-2	135
63	%[††]	8-6	16	%[††]	0-8-4	045
64	≠	8-4	14	" (quote)	8-7	042
65	↱	0-8-5	35	_ (underline)	0-8-5	137
66	∨	11-0	52	!	12-8-7	041
67	∧	0-8-7	37	&	12	046
70	↑	11-8-5	55	' (apostrophe)	8-5	047
71	↓	11-8-6	56	?	0-8-7	077
72	<	12-0	72	<	12-8-4	074
73	>	11-8-7	57	>	0-8-6	076
74	≤	8-5	15	@	8-4	100
75	≥	12-8-5	75	\	0-8-2	134
76	¬	12-8-6	76	^ (circumflex)	11-8-7	136
77	; (semicolon)	12-8-7	77	; (semicolon)	11-8-6	073

[†] Twelve zero bits at the end of a 60-bit word in a zero byte record are an end of record mark rather than two colons.

[††] In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations yield a blank (55_8).

TABLE A-2. CDC CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD	Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD
00	00	blank	55	20	32	40	H	10	70
01	01	≤	74	15	33	41	I	11	71
02	02	%	63 †	16 †	34	42	∨	66	52
03	03	[	61	17	35	43	J	12	41
04	04	→	65	35	36	44	K	13	42
05	05	≡	60	36	37	45	L	14	43
06	06	∧	67	37	38	46	M	15	44
07	07	↑	70	55	39	47	N	16	45
08	10	↓	71	56	40	50	O	17	46
09	11	>	73	57	41	51	P	20	47
10	12	≥	75	75	42	52	Q	21	50
11	13	¬	76	76	43	53	R	22	51
12	14	.	57	73	44	54	]	62	32
13	15	)	52	74	45	55	S	23	22
14	16	;	77	77	46	56	T	24	23
15	17	+	45	60	47	57	U	25	24
16	20	$	53	53	48	60	V	26	25
17	21	*	47	54	49	61	W	27	26
18	22	–	46	40	50	62	X	30	27
19	23	/	50	21	51	63	Y	31	30
20	24	,	56	33	52	64	Z	32	31
21	25	(	51	34	53	65	:	00 †	none†
22	26	=	54	13	54	66	0	33	12
23	27	≠	64	14	55	67	1	34	01
24	30	<	72	72	56	70	2	35	02
25	31	A	01	61	57	71	3	36	03
26	32	B	02	62	58	72	4	37	04
27	33	C	03	63	59	73	5	40	05
28	34	D	04	64	60	74	6	41	06
29	35	E	05	65	61	75	7	42	07
30	36	F	06	66	62	76	8	43	10
31	37	G	07	67	63	77	9	44	11

†In installations using the 63-graphic set, the % graphic does not exist. The : graphic is display code 63, External BCD code 16.

TABLE A-3. ASCII CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code	Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code
00	00	blank	55	20	32	40	@	74	40
01	01	!	66	21	33	41	A	01	41
02	02	"	64	22	34	42	B	02	42
03	03	#	60	23	35	43	C	03	43
04	04	$	53	24	36	44	D	04	44
05	05	%	63†	25	37	45	E	05	45
06	06	&	67	26	38	46	F	06	46
07	07	'	70	27	39	47	G	07	47
08	10	(	51	28	40	50	H	10	48
09	11	)	52	29	41	51	I	11	49
10	12	*	47	2A	42	52	J	12	4A
11	13	+	45	2B	43	53	K	13	4B
12	14	,	56	2C	44	54	L	14	4C
13	15	-	46	2D	45	55	M	15	4D
14	16	.	57	2E	46	56	N	16	4E
15	17	/	50	2F	47	57	O	17	4F
16	20	0	33	30	48	60	P	20	50
17	21	1	34	31	49	61	Q	21	51
18	22	2	35	32	50	62	R	22	52
19	23	3	36	33	51	63	S	23	53
20	24	4	37	34	52	64	T	24	54
21	25	5	40	35	53	65	U	25	55
22	26	6	41	36	54	66	V	26	56
23	27	7	42	37	55	67	W	27	57
24	30	8	43	38	56	70	X	30	58
25	31	9	44	39	57	71	Y	31	59
26	32	:	00†	3A	58	72	Z	32	5A
27	33	;	77	3B	59	73	[	61	5B
28	34	<	72	3C	60	74	\	75	5C
29	35	=	54	3D	61	75	]	62	5D
30	36	>	73	3E	62	76	^	76	5E
31	37	?	71	3F	63	77	_	65	5F

†In installations using a 63-graphic set, the % graphic does not exist. The : graphic is display code 63.

GLOSSARY B

Alphanumeric -
The description of a data item that can be any character A through Z, digit 0 through 9, or special character recognized by Query Update.

Area -
A uniquely named data base subdivision that contains data records; a file.

Beginning-of-Information -
The start of the first user record in a file.

Break -
The point during preparation of a report page when headings and/or footings are to be inserted.

Character Set -
Set of graphic and/or control characters.

Code Set -
System of symbols used to represent each character within a character set.

Condition -
One of a set of specified values for which a data item can be tested.

Current Catalog -
The catalog (default or permanent) that is available for recording Query Update transmissions.

CYBER Record Manager (CRM) -
A software package running under the NOS and NOS/BE operating systems that allows a variety of record types, blocking types, and file organizations to be created and accessed. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

Data Administrator -
The person, or group of people, that lead the design, programming, implementation, and maintenance efforts associated with a DMS-170 controlled data base.

Data Base File -
A file whose organization and content is described by one or more subschemas.

Data Description Language (DDL) -
The language used to generate a subschema.

Default Catalog -
A local file (ZZZZZQ2) that is available for recording Query Update transmissions.

Delimiter -
One of a set of characters used to separate and organize data items; synonymous with separator.

Directive -
A Query Update statement that describes an operation to be performed. A directive consists of a reserved word in the Query Update language and a combination of recognized symbols, punctuation, and user-supplied elements.

Directory -
A file that contains area and record attributes. The permanent file directory (subschema) for a data base file is created by DDL; the temporary file directory for a data base file is created by an EXTRACT or DISPLAY UPON (directory not generated on NOS 1) directive. The temporary file directory for a non-data-base file is created by a DESCRIBE or DISPLAY UPON (directory not generated on NOS 1) directive.

File Organization -
The predetermined arrangement of stored data. CYBER Record Manager recognizes the following organizations: sequential, extended indexed sequential, extended actual key, extended direct access, initial indexed sequential, initial actual key, and initial direct access.

Footing -
Lines of print that comprise a caption and occur after a break.

Heading -
Lines of print that comprise a caption and occur after a break.

Index File -
A file that contains an entry for each unique value of alternate key and associates it with a list of primary keys for all records containing that value.

Key -
One or more data items, the contents of which identify a record or set of records.

Layout Directives -
Directives that supply arrangement and structure of a report on a printed page.

Logical Operator -
A word defining the logical connections between two terms. Query Update recognizes AND, OR, XOR, and NOT.

Non-Data-Base File -
A sequential file whose organization and content is not described by a subschema.

Numeric -
The description of a data item that can be any digit 0 through 9.

Permanent Catalog -
A permanent file that contains recorded Query Update transmissions.

Permanent File -
A disk file known to the system because the file name has been cataloged in a permanent file table.

Picture -
The description of the general characteristics and editing requirements of a data item.

Record -
CYBER Record Manager defines a record as a group of related characters. A record or a portion of a record is the smallest collection of information passed between CYBER Record Manager and a user program.

Register -
Query Update locations that retain current data for display purposes.

Relation -
The logical structure formed by the joining of records.

Relational Operator -
An abbreviation or correspondence symbol that is used to describe a relationship between two terms. Query Update recognizes EQ, NE, GT, LT, GE, and LE.

Report Information Table -
An internal table that is generated by Query Update and used to produce a report.

Reserved Word -
The first word of a Query Update directive.

Separator -
A character used by Query Update as a delimiter.

Session -
Series of transmissions sent by a user between the QU control statement and the END directive.

Session Id -
The six-character session name assigned by the user.

Subschema -
Plan or outline described with DDL statements regarding names and characteristics of data items, records, areas, and relationships that must be maintained among data base elements.

Subscripting -
Use of an integer or variable to identify a particular element in an array.

Temporary Data Item -
An item established through a DEFINE, DESCRIBE, or SPECIFY directive for temporary use with a data base or non-data-base file.

Transmission -
One or more directives submitted as a unit.

Transmission Id -
The three-digit system-supplied identifier assigned to one or more directives in a session catalog.

INDEX

COMMENT SHEET

MANUAL TITLE: Query Update Version 3 Application Programming User's Guide

PUBLICATION NO.: 60499000 REVISION: C

NAME:

COMPANY:

STREET ADDRESS:

CITY: STATE: ZIP CODE:

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

____ Please reply ____ No reply necessary

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE

TAPE

TAPE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division

215 Moffett Park Drive
Sunnyvale, California 94086

CUT ALONG LINE

FOLD

FOLD