



DMS-170

**DDL
VERSION 3
REFERENCE MANUAL**

**Volume 3:
Sub-Schema Definition for
CYBER Record Manager Use With:
Query Update**

**CDC® OPERATING SYSTEMS:
NOS 1
NOS/BE 1**

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	-
Title Page	-
ii	C
iii/iv	C
v/vi	C
vii	C
viii	C
ix	C
1-1 thru 1-4	C
2-1	C
2-2	A
3-1 thru 3-3	C
4-1 thru 4-16	C
5-1 thru 5-8	C
6-1	C
6-2	C
A-1 thru A-4	C
B-1 thru B-11	C
C-1 thru C-5	C
D-1	C
D-2	C
E-1 thru E-3	C
F-1 thru F-6	C
F-7	B
G-1	C
G-2	A
G-3	A
H-1 thru H-3	C
I-1	C
I-2	C
Index-1 thru -3	C
Comment Sheet	C
Mailer	-
Back Cover	-



PREFACE

This manual defines the Data Description Language (DDL) for the Query Update sub-schema using CYBER Record Manager (CRM). DDL is a component of DMS-170, a data management system developed by Control Data Corporation.

As described in this publication, DDL Version 3.2 operates under control of the following operating systems:

- NOS 1 for the CONTROL DATA® CYBER 170 Series; CYBER 70 Models 71, 72, 73, and 74; and 6000 Series Computer Systems
- NOS/BE 1 for the CDC® CYBER 170 Series; CYBER 70 Models 71, 72, 73, and 74; and 6000 Series Computer Systems

This manual is designed for use by the data administrator or the staff member responsible for describing Query Update sub-schemas that are used by Query Update programs accessing the data base through CRM. It is assumed that the user is an experienced programmer and has used Control Data computers and software products. It is also assumed that the user is familiar with the Query Update language.

CDC offers guidelines for the use of the software described in this manual. These guidelines appear in appendix I. Before using the software described in this manual, the reader is strongly urged to review the content of this appendix. The guidelines recommend use of this software in a manner that reduces the effort required to migrate application programs to future hardware or software systems.

Material related to the use of this product is contained in the publications listed below; the publications are listed within groupings that indicate relative importance to readers of this manual.

The NOS manual abstracts and the NOS/BE manual abstracts are instant-sized manuals containing brief descriptions of the contents and intended audience of all NOS and NOS product set manuals, and NOS/BE and NOS/BE product set manuals, respectively. The abstracts manuals can be useful in determining which manuals are of greatest interest to a particular reader. The Software Publications Release History serves as a guide in determining which revision level of software documentation corresponds to the Programming Systems Report (PSR) level of installed site software.

The following publications are of primary interest:

<u>Publication</u>	<u>Publication Number</u>
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700
Data Base Utilities Version 1 Reference Manual	60498800
Query Update Version 3 Programmer User's Guide	60499000
Query Update Version 3 Reference Manual	60498300

The following publications are of secondary interest:

<u>Publication</u>	<u>Publication Number</u>
NOS Version 1 Manual Abstracts	84000420
NOS Version 1 Reference Manual Volume 1 of 2	60435400
NOS/BE Version 1 Manual Abstracts	84000470
NOS/BE Version 1 Reference Manual	60493800
Software Publications Release History	60481000

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

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

O

O

O

O

O

O

O

INDEX

FIGURES

1-1	Data Base Processing With DMS-170	1-2	4-27	Examples of Replacement Characters	4-11
1-2	Query Update Processing of the Data Base	1-3	4-28	Examples of Picture Editing	4-12
3-1	Identifier Format	3-2	4-29	OCCURS Clause Format	4-12
4-1	Identification Description Entry Format	4-1	4-30	USAGE Clause Format	4-13
4-2	SUB-SCHEMA NAME Clause Format	4-1	4-31	Relation Description Entry Format	4-14
4-3	PROCEDURE LIBRARY Clause Format	4-1	4-32	RELATION-NAME Clause Format	4-14
4-4	Formats of Area Description Entries	4-2	4-33	JOIN Clause Format	4-14
4-5	AREA-NAME Clause Format	4-2	4-34	RESTRICT Clause Format	4-15
4-6	BLOCK CONTAINS Clause Format	4-2	4-35	RESTRICT Clause Example	4-15
4-7	BLOCK-TYPE Clause Format	4-3	5-1	DDL Control Statement Format	5-1
4-8	RECORD CONTAINS Clause Format	4-3	5-2	REQUEST Control Statement Format	5-2
4-9	RECORD-TYPE Clause Format	4-3	5-3	CATALOG Control Statement Format	5-2
4-10	LOG Clause Format	4-4	5-4	DEFINE Control Statement Format	5-2
4-11	ORGANIZATION Clause Format	4-5	5-5	Compiling a Sub-Schema	5-2
4-12	FILE-LIMIT Clause Format	4-6	5-6	Creating a Sub-Schema Library	5-3
4-13	SEQUENCE Clause Format	4-6	5-7	Adding a Sub-Schema to the Library	5-3
4-14	INDEX-LEVEL Clause Format	4-6	5-8	Replacing a Sub-Schema in the Library	5-4
4-15	INDEX-BLOCK Clause Format	4-7	5-9	Deleting Sub-Schemas From the Library	5-5
4-16	INDEX-PADDING Clause Format	4-7	5-10	Compacting a Sub-Schema Library	5-5
4-17	DATA-PADDING Clause Format	4-7	5-11	Sub-Schema Source Listing	5-6
4-18	NUMBER OF BLOCKS Clause Format	4-7	5-12	Describing the Data Base	5-7
4-19	ON CALL Clause Format	4-7	5-13	Creating the Data Base	5-8
4-20	Formats of Record Description Entries	4-8	6-1	Creating a Sub-Schema Library and Compiling a Sub-Schema Under the NOS/BE Operating System	6-1
4-21	RECORD-NAME Clause Format	4-8	6-2	Creating a Sub-Schema Library and Compiling a Sub-Schema Under the NOS Operating System	6-2
4-22	PICTURE Clause Format	4-8			
4-23	Alphabetic Data Items	4-8			
4-24	Numeric Data Items	4-9			
4-25	Alphanumeric Data Items	4-10			
4-26	Examples of Insertion Characters	4-11			

TABLES

4-1	Record Type Descriptions	4-4
-----	--------------------------	-----

NOTATIONS

Each DDL entry is described in terms of a reference format. When more than one specific arrangement is permitted, the format is separated into numbered formats.

NOTATION USED IN REFERENCE FORMATS

UPPERCASE words are reserved words and must appear exactly as shown. Reserved words can be used in source statements only as specified in the reference formats.

**UNDERLINED
UPPERCASE** words are required when the format in which they appear is used.

lowercase words are generic terms that represent the words or symbols supplied by the user. When generic terms are repeated in a format, a number is appended to the term for identification.

[] Brackets enclose optional portions of a reference format. All of the format within the brackets can be omitted or included at the user's option.

{ } Braces enclose two or more vertically stacked items in a reference format when one and only one of the enclosed items must be used.

||| Vertical bars enclose two or more vertically stacked items in a reference format when at least one of the enclosed items must be used. Each of the vertically stacked items can be used once.

... Ellipses immediately follow a pair of brackets or braces to indicate that the enclosed material can be repeated at the user's option; also follow data-name, expression, or file-name to indicate that the user-supplied element can be repeated.

Punctuation symbols shown within the formats are required unless enclosed in brackets and specifically noted as optional. In general, commas and semicolons are optional. Periods are required to terminate division titles.

NOTATION USED IN EXAMPLES

↑ indicates the position of an assumed decimal point in an item.

A plus or minus sign above a numeric character indicates an operational sign is stored in combination with the numeric character.

Character positions in storage are shown by boxes.

A	B	C	D
---	---	---	---

Δ indicates a space (blank).

O

O

O

O

O

O

O

The DMS-170 software package functions as the data management system for Control Data computer systems. Through this data management system, a data base can be defined, maintained, and controlled in an environment totally independent of the applications that are accessing it. In this way, conventional files otherwise owned and processed by a number of distinct applications can be described through the data description language facilities of DMS-170. Consequently, the responsibility for tasks such as data description, data conversion, and validity checking is transferred from the application programmer to the data administrator.

The DMS-170 data management system is composed of the following elements:

- Data Description Language (DDL), which creates the schema definition, as well as the COBOL and Query Update sub-schema definitions
- CONTROL DATA CYBER Database Control System (CDCS), which controls, monitors, and interprets all data base requests from COBOL and FORTRAN application programs
- CONTROL DATA CYBER Record Manager (CRM), which handles all input/output processing requests on a data base from an application program
- FORTRAN Data Base Facility, which is composed of a FORTRAN sub-schema definition language and a Data Manipulation Language (DML)

Query Update functions within DMS-170 whenever a Query Update sub-schema is present. Query Update, using sub-schemas defined by DDL, can access the data base in either CDCS data base access mode or CRM data base access mode. The relationship between Query Update and the DMS-170 elements involved in processing the data base after it has been defined by DDL is shown in figure 1-1.

DATA BASE DEFINITION

The responsibility for the definition of a data base lies with the data administrator. The data administrator is a person or group of persons who have the task of developing and defining the data base as well as monitoring and controlling the day-to-day processing of that data base.

To define a data base, the data administrator uses DDL. Through this language, the schema and sub-schemas can be created. The data descriptions in the schema and in the sub-schemas follow specific structuring conventions, include unique clauses and statements, and conform to an individual set of rules.

SCHEMA DEFINITION

The schema is a detailed description in English-like syntax of the data in a data base. An installation can have many data bases, but only one schema is allowed for each data base. The schema description is generated by DDL statements that name the schema, organize the schema into files, describe each record type together with the

characteristics of the data comprising the record, and describe relationships among files. The DDL source statements describing the data are used as input to the DDL compiler and are compiled into an object schema, or schema directory. The data administrator then uses the schema description to create any number of sub-schemas.

SUB-SCHEMA DEFINITIONS

A sub-schema is a detailed description of selected portions of a data base to be used by application programs. Although only one schema definition is allowed for each data base, any number of sub-schemas can be defined to meet the needs of different types of applications.

Using DDL, the data administrator defines sub-schemas for use by application programs written in the Query Update language. A Query Update sub-schema describes the portion of a data base that can be accessed by a Query Update user in either CDCS data base access mode or CRM data base access mode.

CDCS Data Base Access Mode

The data descriptions in Query Update sub-schemas in CDCS data base access mode are written to correspond to data descriptions in the schema. Certain differences are allowed to exist; these differences are resolved by DDL and CDCS.

The schema must be compiled before the sub-schema is compiled. The DDL source statements describing the sub-schema are compiled into an object sub-schema, or sub-schema directory. The names and descriptions of data to be referenced in a Query Update program are obtained from a listing of the sub-schema.

CRM Data Base Access Mode

The data descriptions in Query Update sub-schemas in CRM data base access mode are not based on a schema definition. Schema-defined files can be accessed, however, providing the DDL source statements that define the sub-schema describe the data exactly as it is described in the schema. Appendix H contains a summary of data definition in DMS-170 to aid in accessing schema-defined files. For each sub-schema, the DDL source statements used as input to the DDL compiler name the sub-schema, specify needed files, define the content and structure of records, identify relations among files to be used, and specify record qualification for relation processing.

After the DDL source statements describing the sub-schema have been written, they are compiled into an object sub-schema, or sub-schema directory. The names and descriptions of data to be referenced in a Query Update program are obtained from a listing of the sub-schema.

The remainder of this manual describes Query Update sub-schemas in CRM data base access mode only. See volume 2 of the DDL 3 reference manual for information on Query Update sub-schemas in CDCS data base access mode.

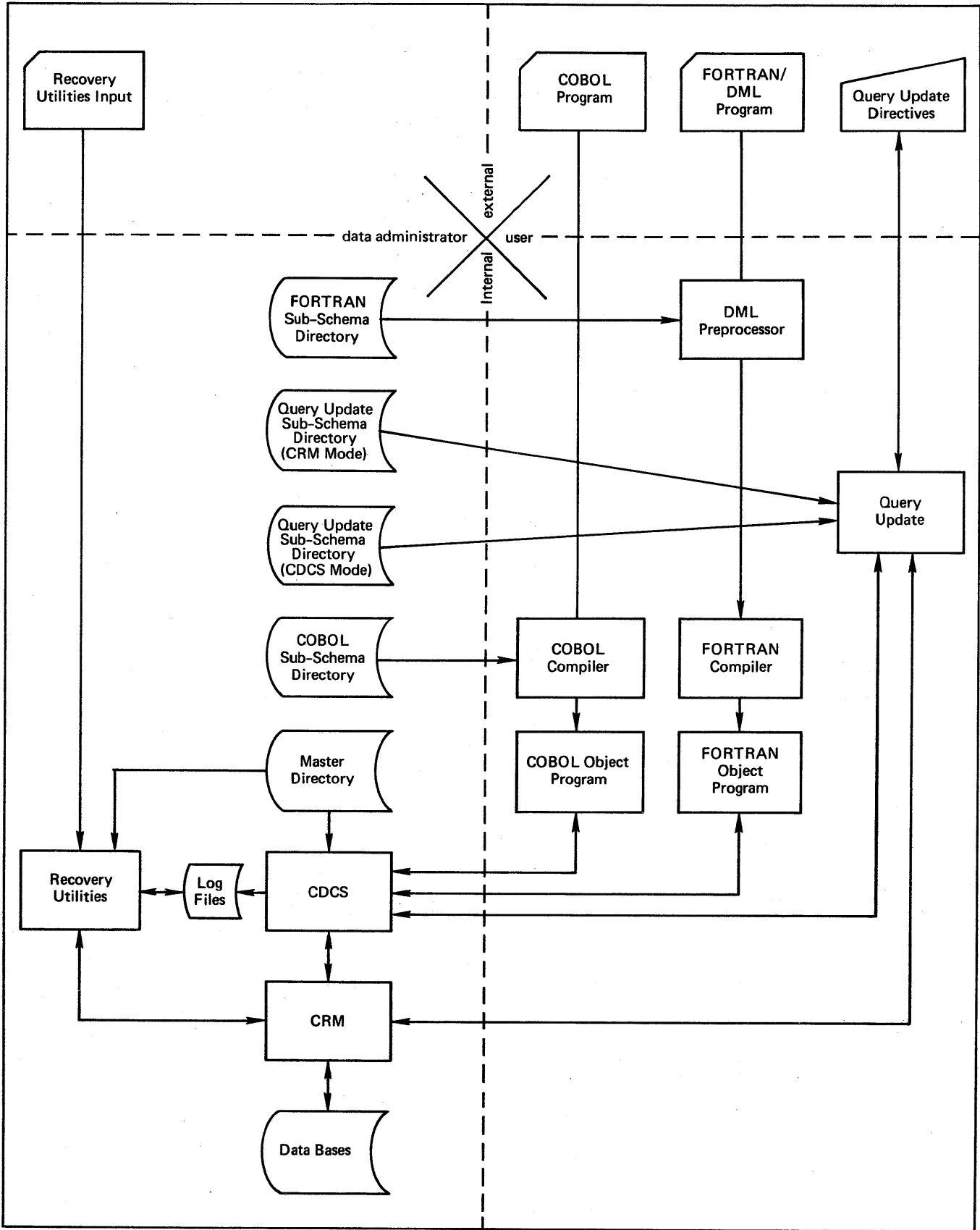


Figure 1-1. Data Base Processing With DMS-170

DATA BASE PROCESSING

Once a data base has been defined by the data administrator, it can be accessed by users of the Query Update language. Query Update directives can be used to perform read, write, update, and delete operations on data base files. Query Update can be used to access the data base in either batch or interactive mode. Query Update processing of the data base is shown in figure 1-2.

When Query Update accesses data base files in CRM data base access mode, input/output processing requests on data base files are handled by Advanced Access Methods (AAM) for direct access, indexed sequential, and actual key files and by Basic Access Methods (BAM) for sequential files. To access schema-defined files, the sub-schema must describe the data exactly as it is described in the schema (see appendix H).

Several special features are available for data base processing. Concurrency allows two Query Update users to access the same data base file at the same time. A relation facility enables the user to retrieve data from several linked files with a single read request. A data base procedure linkage allows special-purpose subprograms written by the data administrator to be called when specified situations occur during processing. The input/output capabilities of CRM handle all data base processing requests from Query Update application programs. Logging and recovery utilities provide for preservation of the integrity of a data base.

CONCURRENCY

Concurrent file access is allowed for Query Update users. Concurrency means that two or more Query Update application programs can access the same data base file at the same time. Programs can access a file concurrently for retrieval or update purposes. Concurrency is possible through the use of appropriate control statement parameters. Refer to the Query Update reference manual for more information.

RELATIONS

The relational data base facility of Query Update allows an application program to access data from related files with a single read request. In the Query Update sub-schema, the data administrator links the files together into a logical, meaningful relationship, called a relation, by specifying a relation entry. The relation entry assigns a name to the relation and specifies the data items to be used to link the files. The Query Update sub-schema can specify qualification criteria for retrieving only certain records from the data base files joined in the relation.

An application program accesses a relation by specifying a single read request with the name of the relation that is to be read. Query Update processes the request and returns a record occurrence from each file in the relation to the user's work area for the file.

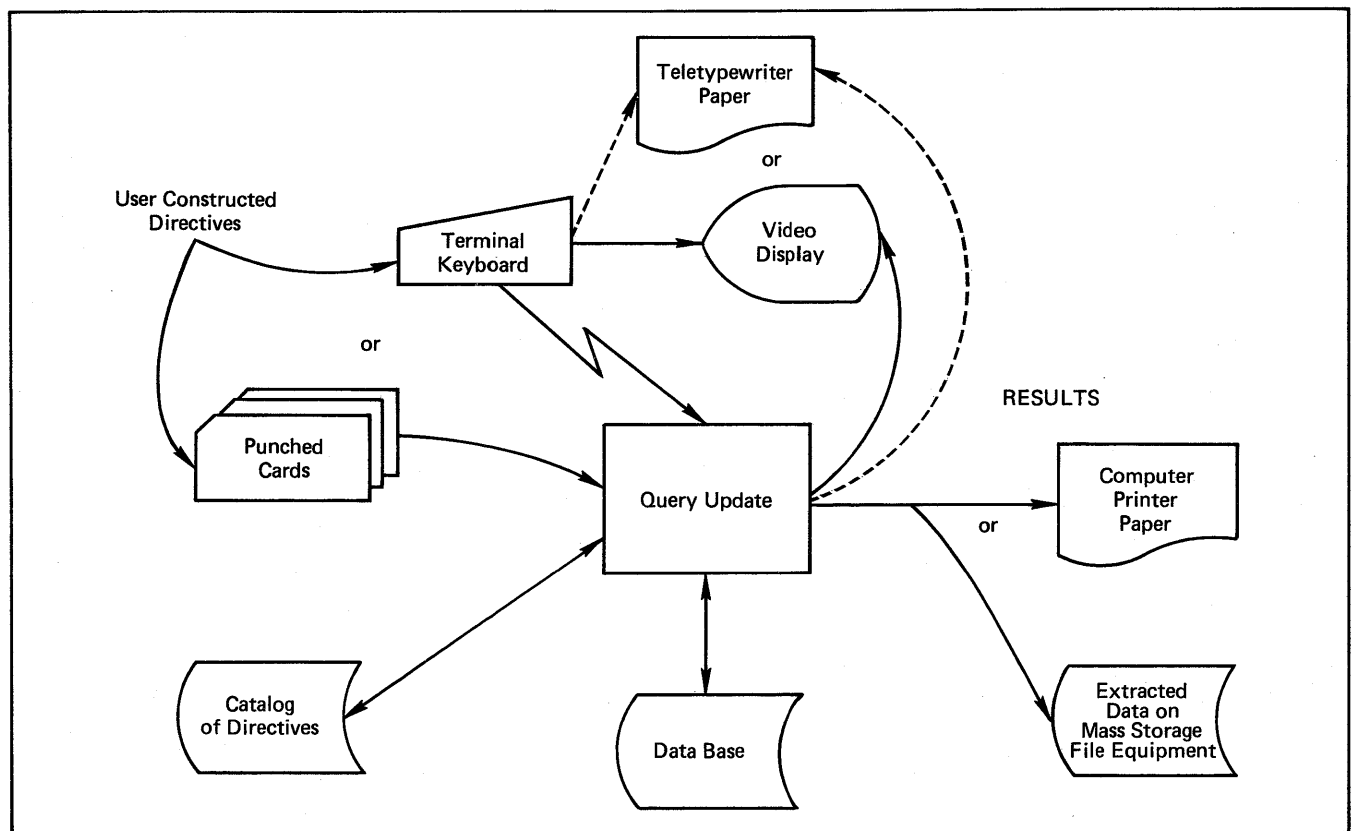


Figure 1-2. Query Update Processing of the Data Base

DATA BASE PROCEDURES

Data base procedures are special subprograms written by the data administrator to perform a variety of supplemental operations not otherwise performed by Query Update. The procedures are called when specific situations occur during processing. The conditions under which data base procedures are to be executed are specified in the Query Update sub-schema.

Some of the functions that can be performed by data base procedures are: data validation; calculation of values for actual or virtual data items; additional processing on creation, retrieval, or update of data base records; privacy checking; and special handling of detected error conditions. The use of data base procedures to perform these functions enables an installation to make the most efficient use of the DMS-170 system.

INPUT/OUTPUT PROCESSING

Processing of input/output statements that reference data base files is handled by CRM. All data base files accessed by Query Update are conventional CRM files; files are identified and described in the sub-schema.

File Organization

File organization of data base files accessed by Query Update application programs is specified in the sub-schema. The file organization information is stored in the sub-schema directory. The file organizations allowed for data base files that are to be accessed by Query Update are extended AAM files (which include extended direct access, extended actual key, and extended indexed sequential) and sequential BAM files.

Records in indexed sequential files are stored in ascending order by key. The records can be accessed either randomly by key or sequentially by position. This file organization is most efficient for very large mass storage files that are to be accessed both randomly and sequentially.

Records for direct access files are stored randomly in fixed length blocks. The number of the block to receive a record is determined by a calculation performed by the system on

the record key. Records can be accessed randomly by key; records can also be accessed sequentially for read-only processing. Direct access file organization is used most effectively when rapid random access of large mass storage files is required.

Actual key files contain records whose key values are assigned by the system. Records can be accessed randomly by actual key; records also can be accessed sequentially. Actual key file organization is used most effectively for large mass storage files when the user can keep track of system-assigned keys and when performance and file growth characteristics are of primary concern.

Records for sequential files are stored in the same physical order in which they are generated. Records are accessed by the issuing of successive reads. Sequential file organization is used most effectively with files that are to be read sequentially in the order in which they were created and that are to have records added at the end of the file.

Multiple-Index Processing

Multiple-index processing is performed if alternate keys are defined for indexed sequential, actual key, or direct access files. An index is created for each alternate key in a data file when the file is created. The indexes are updated automatically whenever the data file is updated. Records can then be retrieved by the primary key or by an alternate key. For detailed information refer to the Advanced Access Methods reference manual.

DATA BASE RECOVERY

The recovery facilities of Query Update supply the means to deal with a lost, partially destroyed, or invalid data base. Logging is specified in the sub-schema. Logging before and after images of records, along with transaction logging, is available for Query Update applications. The log files contain information that the data base administrator can use for off-line recovery of the data base or for statistical analysis. Query Update provides a utility that a data base administrator can execute to restore or reconstruct a data base.

The Query Update sub-schema describes the data that is needed by one or more Query Update application programs accessing the data base through CYBER Record Manager (CRM). The sub-schema consists of three divisions:

Identification Division

Contains one statement that names the sub-schema and the optional procedure library.

Data Division

Contains area and record description statements. The area statements describe area structure and include the optional transaction logging and data base procedures. The record statements describe the structure and content of each record type in the area.

Relation Division

Contains optional relational statements that identify the specific area relationships that are to be used when querying more than one area of the data base and specifies the qualification criteria that determines the projected records that can be selected for the Query Update user.

DATA DESCRIPTION

The basic unit of data description is the entry. The record description entry is the entire description of a data base record type. It consists of a record name entry followed by a series of data description entries that describe the specific data items within the record. Each data description entry has a level number and a data name. In addition, the entry can have one or more clauses describing the data item. Data items are described in terms of size, class, and usage.

The level number designates the level of the entry relative to other entries in the record description. The highest level entry is an implied 01 level and is reserved for the record-name entry. Other entries, which can be either group or elementary data items, are assigned level numbers 02 through 49 based on the position of the specific data item within the hierarchical structure of the record. Level numbers need not be consecutive but must be ordered so that the higher the number, the lower the entry in the hierarchy.

At least one clause must be included in the data description entry for an elementary item. A group item can also have one or more clauses. The order of clauses is not important except where explicitly stated in the clause description. Semicolons or commas can be used to separate clauses.

DATA ORGANIZATION

Data is organized into named subdivisions called areas. (Throughout this manual the term area is synonymous with file.) Each area, in turn, is organized into named subdivisions called records.

AREAS

An area description entry begins with the area name and includes a series of subordinate entries that describe the total structure of the area. A maximum of 64 areas can be included in a single sub-schema, and each area can contain occurrences of one or more types of records.

RECORDS

Data items in the record description statements are organized first into records and then into group and elementary items. A record description entry begins with the record name, which has an implied 01 level number, and includes a series of subordinate data description entries with level numbers 02 through 49.

Group Items

A group item is a collection of related items organized in a hierarchical structure. The group name data description entry has the lowest level number within the group itself. Other items within the group item have successively higher level numbers ending with the most elementary item, the highest numbered item in the group. All data items in a group item can be referenced collectively by the data name of the group item. A description must be written for each data item in the group, and the group name data description entry must have at least a level number and a data name. A group item can be part of a larger group (nested group items).

Elementary Items

An elementary item is an item that cannot be further subdivided. If it is part of a group item, the elementary item has the highest level number in the group item to which it belongs.

NONREPEATING DATA ITEMS

A nonrepeating elementary data item is a single data description entry. It consists of the level number, the data name, a PICTURE clause, and an optional USAGE clause.

Nonrepeating group items can be specified in the sub-schema by designating a series of related data items as subordinate entries to a group data item, which allows a Query Update program to access several data items with one request. A nonrepeating group item consists of three or more data description entries. The first entry specifies the data name for the entire group data item. The only restriction placed on nonrepeating group items is that the relative hierarchy of the sub-schema be preserved. A nonrepeating group data item is illustrated as follows:

```
02 ADDRESS
03 NUMBERS PIC 9(5)
03 STREET PIC X(23)
03 CITY PIC A(20)
03 ZIP-CODE PIC 9(5)
03 STATE PIC A(2)
```

REPEATING DATA ITEMS

Two types of repeating data items can be designated in the sub-schema: vectors and repeating groups. A vector is an elementary data item that is repeated a number of times in each record. A repeating group is a collection of data items that is repeated; the entire collection, not individual data items, is repeated a number of times in each record. Repeating data items are specified by including the OCCURS clause in the data description entry.

Vectors

A vector is a repeating elementary data item. A vector is described with the OCCURS clause and the PICTURE or USAGE clause. No data item can be subordinate to a vector. A vector is illustrated as follows:

```
02 MONTH-TOT PIC 9(4)V99
   OCCURS 12 TIMES
```

Repeating Groups

A repeating group is described with two or more data description entries. The first entry consists of the group data name and the OCCURS clause. Each additional entry is subordinate to the first entry and describes a repeating group, a vector, or an elementary data item. The vector or elementary data item contains the PICTURE clause. Up to three levels of nested groups can be specified. When repeating groups and vectors are nested, the vector is considered to be a repeating group and is included in the level count. A repeating group data item is illustrated as follows:

```
03 MONTHLY-ORDERS OCCURS 12 TIMES
   05 NUM-ORDERS PICTURE 99
   05 TOTAL-AMT PICTURE 9(6)V99
```

Fixed Occurrence Data Items

A fixed occurrence data item, which can be a group or a vector, occurs a fixed number of times. Format 1 of the OCCURS clause specification is used to describe a fixed occurrence data item (refer to section 4). The integer entered in the clause specifies the exact number of

occurrences for the data item. A fixed occurrence data item is illustrated as follows:

```
03 MONTHLY-ORDERS PIC 9999V99
   OCCURS 12 TIMES
```

Variable Occurrence Data Items

A variable occurrence data item, which can be a group or a vector, occurs a variable number of times. Format 2 of the OCCURS clause specification is used to describe a variable occurrence data item (refer to section 4). The integers entered in the clause specify the minimum and maximum number of occurrences for the data item and also specify the elementary data item that contains the actual number of occurrences for the record.

The data item that designates the actual number of occurrences must be an elementary data item and it must precede the data description entry for the variable occurrence data item. The variable occurrence data item must be the last item in the record. Only subordinate data description entries can follow the entry containing the OCCURS clause.

A variable occurrence data item is illustrated as follows:

```
03 NUM-ITEMS PICTURE 99
03 MONTHLY-ORDERS PIC 9999V99
   OCCURS 1 TO 15 TIMES DEPENDING
   ON NUM-ITEMS
```

DATA SIZE AND CLASS

The size and class of data items in the sub-schema are specified by the PICTURE and USAGE clauses. The PICTURE clause is used for formatting and specifying the class and size characteristics of a data item as the item appears in the user working area. The USAGE clause specifies whether a data item is used predominantly for computation or for display.

The class of a data item is specified by the character designators A (alphabetic), 9 (numeric), and X (alpha-numeric) in the picture-specification of the PICTURE clause. The size of a data item is specified by the number of character position designators (A, 9, and X) in the picture-specification of the PICTURE clause.

The Query Update sub-schema DDL source program consists of a series of statements that describe a portion of the data base. The rules, conventions, and hierarchical structures of DDL are similar to those of COBOL.

LANGUAGE ELEMENTS

DDL source statements are composed of clauses that contain reserved words, user-defined names, and literals. The use of these elements is described in the following paragraphs. The specific formats of the clauses are defined in section 4.

RESERVED WORDS

Reserved words are English words and abbreviations that have special meanings to the DDL compiler. These words can be used only as shown in the format specifications. A reserved word must be spelled correctly; it cannot be replaced by another word. Over 275 words have been defined as reserved words. Appendix D contains a complete list of DDL reserved words used in Query Update sub-schema definition.

Two types of reserved words are recognized by the DDL compiler: keywords and optional words. A keyword is a reserved word that must be used in a specific clause. Keywords are essential to convey the meaning of a clause to the compiler. An optional word is a reserved word that can be included in a clause to improve readability. Optional words are recognized by the compiler but are not needed to compile the object coding. In the format specifications, keywords are shown as uppercase words that are underlined; optional words are shown as uppercase words that are not underlined.

USER-DEFINED NAMES

Many of the format specifications include names that are to be supplied by the user. User-defined names identify the sub-schema, areas, records, data items, and procedure names. The type of name to be supplied is indicated in the format specification by a lowercase word.

User-defined names are created when the sub-schema is written. The formation of these names is governed by the following rules:

- Sub-schema, area, index file, log file, procedure library, record, relation, and data names can each contain up to 30 characters; the first character must be alphabetic.
- Procedure names can contain up to 7 characters; the first character must be alphabetic.
- Letters (A-Z), digits (0-9), and the hyphen (-) can be used in a name.
- The hyphen can be used only in record, relation, and data names.
- The hyphen cannot be used to begin or end a name; adjacent hyphens are not allowed.

- Spaces (blanks) cannot be used in a name.
- A name cannot be spelled exactly the same as a reserved word.

LITERALS

In some formats, the user must supply a literal as part of the clause. A literal is a string of characters that represents a specific value. Literals are either numeric or nonnumeric.

Numeric Literals

An integer can contain the digits 0 through 9 and a leading plus or minus sign. If a sign is not included, the integer is treated as positive.

Examples: 3 +403 -07

A fixed-point (real) value can be formed by inserting a decimal point into a valid integer literal. The literal must not begin or end with the decimal point.

Examples: +2.4 0.06 -72.0

A real value in scientific notation is formed from a fixed-point literal followed by the character E followed by an integer. The trailing integer has the significance of the power of 10 by which the fixed-point literal is multiplied.

Examples: +14.0E-2 7.6E0 -0.56E4

The maximum size for an integer or fixed-point literal is 30 digits; however, only 14 digits can be significant. Up to 12 leading or trailing zeros can be specified for decimal alignment.

A complex value can be formed by bracketing the character I by valid real literals in scientific notation.

Examples: 43.0E1I7.6E1 0.0E1I-46.3E1
 -25.7E-1I-6.0E2 0.007E-4I+0.0E1

Nonnumeric Literals

A nonnumeric literal is a string of up to 255 characters. The string must be enclosed in quotation marks. Any character in the DDL character set, including the space, can be used in a nonnumeric literal. If a quotation mark is to be included in the literal, the quotation mark must be specified twice for each occurrence. For example, "A" "B" would yield the literal A"B.

DATA REFERENCE

Each user-defined name in the sub-schema must be capable of being uniquely referenced. Unless the name itself is unique because no other name has the identical spelling, a method for obtaining unique identification is necessary. Unique reference is recognized by DDL through the qualification, subscripting, and identifier concepts.

Qualification

Qualification is permitted in any clause that references a data name. When a name exists within a hierarchy of one or more names, the higher level names can be used to make the name unique. The data name is written followed by the word OF or IN and the qualifier. The choice between OF or IN is based on readability; the two words are logically equivalent. Qualification must be made to the level necessary to make the name unique; however, qualification can be used even when the name does not need to be qualified.

Subscripting

Subscripting within the DDL syntax is permitted only in the JOIN and RESTRICT clauses. Subscripts are used to indicate which occurrence of a repeating group or elementary item is to be referenced. The data name is written followed by a positive integer constant or the keyword ANY enclosed within parentheses. Specific rules regarding the use of subscripts are detailed in the following paragraphs.

Identifier

The term identifier is used to indicate a data name that is referenced uniquely through a combination of subscripts and qualifiers. When the term identifier appears in JOIN and RESTRICT clauses, it assumes the format shown in figure 3-1.

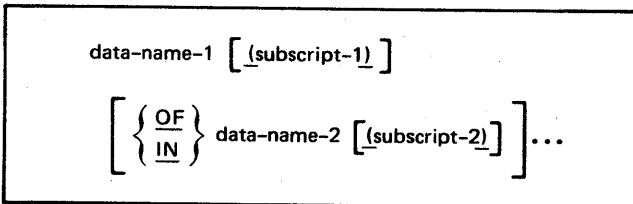


Figure 3-1. Identifier Format

A maximum of five data-name-2 qualifiers can be specified. They must be listed in the order of innermost to outermost level of nesting of the group; hence the leftmost qualifier (data-name-2) is the item with the highest level number.

Only one subscript can appear on each data-name that denotes an item defined with an OCCURS clause; only items specified with an OCCURS clause can have a subscript. When describing an identifier that is subordinate to a repeating group, the repeating group name must appear as a qualifier.

DDL CHARACTER SET

The set of characters recognized by the DDL compiler can be combined according to the specified rules to form names and values in the source program. The DDL character set consists of the letters A through Z, the numbers 0 through 9, and the following special characters:

Blank or Space

+ Plus Sign

- Minus Sign or Hyphen

, Comma

; Semicolon

* Asterisk

. Period or Decimal Point

" Quotation Mark

(Left Parenthesis

) Right Parenthesis

\$ Dollar Sign

/ Slash

PUNCTUATION

Most punctuation marks in a DDL source program are optional. When punctuation marks are used, the rules are precise and must be followed exactly. The rules for using punctuation marks are as follows:

- A period is required to terminate the Identification, Data, and Relation Division title lines.
- A period, comma, or semicolon must immediately follow a word and it must be followed by at least one space. If the punctuation mark is in the last column of the statement area (column 72), a blank is assumed to exist immediately following column 72 and prior to column 73 of the program identification area.
- At least one space, or a comma and a space, must separate successive words in a statement.
- Commas and semicolons can be used to separate clauses in a statement.

DDL CODING

DDL source programs can be written on standard 80-column coding sheets. Columns 8 through 72 are used to write DDL statements.

DDL STATEMENTS

The DDL source program that defines a Query Update sub-schema consists of three divisions: Identification, Data, and Relation Division. The division name begins in columns 8 through 11 and is followed by a space, the word DIVISION, and a period. The remainder of the line must be left blank. All other statements can begin in columns 8 through 72.

SEQUENCE NUMBERS

A sequence number consisting only of digits can be entered in columns 1 through 6. The sequence number is optional and has no effect on the source program.

CONTINUATION LINES

Words or literals can be continued from one line to the next line. When a continuation line is written on the coding sheet, a hyphen must be entered in column 7. The continuation must begin in column 12.

COMMENT LINES

Comments can be printed on the source listing. A comment line is written on the coding sheet by entering an asterisk in column 7. The comment begins in columns 8 through 72.

O

O

O

O

O

O

O

DDL statements consist of the reserved words of the DDL language in combination with user-supplied elements. The source statements are used as input to the DDL compiler for the purpose of generating a Query Update sub-schema.

The source program for a Query Update sub-schema contains three divisions: Identification Division, Data Division, and Relation Division. This section of the manual defines the format specifications for each clause that can be used in the source program.

IDENTIFICATION DIVISION

The Identification Division must be the first division in the DDL source program. This division identifies the sub-schema being described and names the optional procedure library. The format of the Identification Division is as follows:

IDENTIFICATION DIVISION.

```
{identification description entry}
```

The identification description entry is a statement containing one required clause that names the sub-schema and one optional clause that names the library in which data base procedures reside. The format of the identification description entry is shown in figure 4-1.

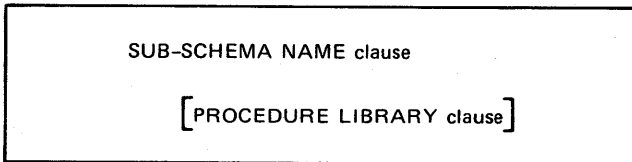


Figure 4-1. Identification Description Entry Format

The format of the SUB-SCHEMA NAME clause is shown in figure 4-2. This clause specifies the name of the sub-schema being created. The sub-schema-name entered in this clause is the name used whenever the sub-schema is referenced after it has been compiled and stored in the sub-schema library. The name must be unique among Query Update sub-schemas in the library.

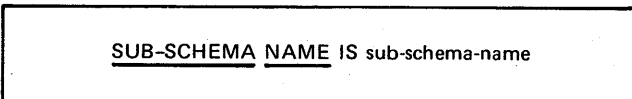


Figure 4-2. SUB-SCHEMA NAME Clause Format

The format of the PROCEDURE LIBRARY clause is shown in figure 4-3. This clause specifies the name of a permanent file user library in which all referenced data base procedures are stored. The library-name must be unique among all file names and must conform to operating system logical file naming conventions. Permanent file parameters are optional. NOS/BE keywords include ID, PW, CY, and SN. NOS keywords include UN, PW, M, and PN;

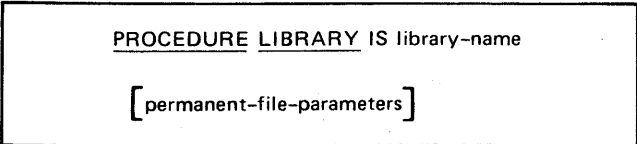


Figure 4-3. PROCEDURE LIBRARY Clause Format

values of M can be R, READ, W, WRITE, M, MODIFY, RM, or READMD. Refer to the appropriate operating system reference manual for explanations of permanent file parameters. When parameters are specified, keywords and their values must appear in the following format:

```
keyword IS value
```

When the PROCEDURE LIBRARY clause is present, Query Update attaches the specified permanent file, loads the procedures, and returns the permanent file. Data base procedure processing is detailed in appendix F.

DATA DIVISION

The Data Division must be the second division in the DDL source program. It identifies and describes areas as well as the records that are contained within the areas. The format of the Data Division is as follows:

DATA DIVISION.

```

{area description entry}
{record description entry}...
:
:
{area description entry}
{record description entry}...
    
```

AREA DESCRIPTION ENTRY

An area description entry precedes its associated record description entries in the Data Division. The area description is a statement consisting of one or more clauses that identify and describe the file. An entry can be qualified, restricted, or overridden by a subsequent record description entry. The format of the area description entry is shown in figure 4-4.

AREA-NAME Clause

The AREA-NAME clause is a required clause that identifies the area. The format of the AREA-NAME clause is shown in figure 4-5.

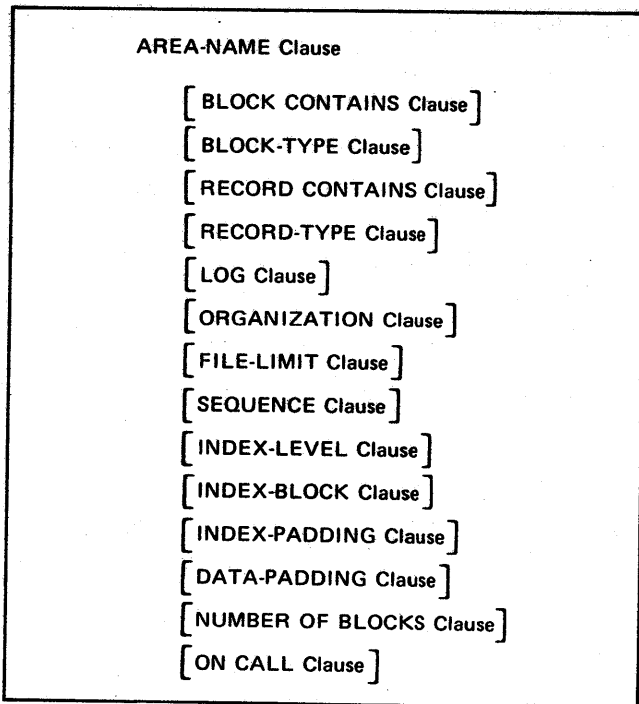


Figure 4-4. Formats of Area Description Entries

The area-name is user-assigned and must conform to DDL and operating system file naming conventions. The first seven characters of the name are used as the logical file name and must be unique within the sub-schema. Permanent file parameters are optional. NOS/BE keywords include ID, PW, CY, and SN. NOS keywords include UN, PW, M, and PN; values of M can be R, READ, W, WRITE, M, MODIFY, RM, or READMD. Refer to the appropriate operating system reference manual for explanations of permanent file parameters. When parameters are specified, keywords and their values must appear in the following format:

keyword IS value

When the TEMPORARY option is selected for AREA-NAME, both the area file and index file are local files for temporary use by one program. An area defined as temporary is similar to a scratch file. Items in the temporary area are not saved after the area is closed. Attach and return operations are not performed for a temporary area during a Query Update session.

The INDEX option names the associated index file for the area when the area contains alternate keys. Index file names must be unique and must conform to operating system file naming conventions. If this option is not included in the sub-schema, it becomes the responsibility of the Query Update user to declare the index file in the INVOKE (or USE) directive at Query Update execution time.

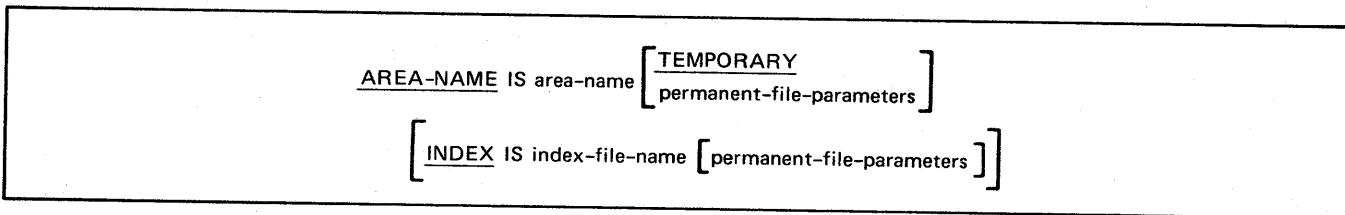


Figure 4-5. AREA-NAME Clause Format

Area and index file permanent file parameters are optional. They can be declared or overridden in the INVOKE (or USE) directive at Query Update execution time.

BLOCK CONTAINS Clause

The BLOCK CONTAINS clause describes the block size and CYBER Record Manager (CRM) block type. The format of the BLOCK CONTAINS clause is shown in figure 4-6.

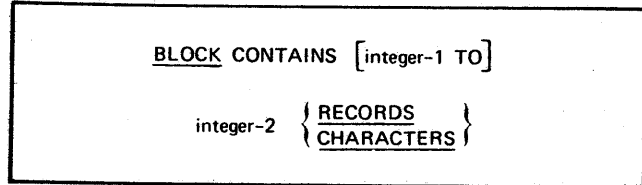


Figure 4-6. BLOCK CONTAINS Clause Format

The entry integer-1 does not affect block size and can be used for documentation purposes. The entry integer-2 is a positive number specifying the maximum number of records or characters per block.

The following statements apply to the BLOCK CONTAINS clause:

- If records are specified, CRM assumes K type blocking unless overridden by the BLOCK-TYPE clause. The K type blocking specification indicates that blocks always contain a fixed number of records whether records are fixed or variable length. If record lengths are variable, the maximum record size is multiplied by the number of records to determine the block size.
- If characters are specified, they are limited to a maximum of 327600. CRM assumes E type blocking unless overridden by the BLOCK-TYPE clause. The E type blocking specification indicates that blocks contain as many records as will fit within a given maximum block size.
- If the clause is omitted, CRM assumes C type blocking unless overridden by the BLOCK-TYPE clause. The C type blocking specification indicates that blocks always contain a fixed number of characters as determined by the storage device type.

BLOCK-TYPE Clause

The BLOCK-TYPE clause describes the CRM block type format and overrides a BLOCK CONTAINS clause. The clause can be specified only when organization is sequential. The format of the BLOCK-TYPE clause is shown in figure 4-7.

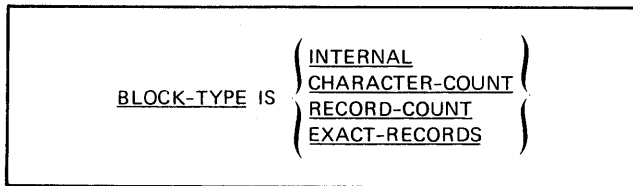


Figure 4-7. BLOCK-TYPE Clause Format

One of four block types recognized by CRM can be specified. Each type is defined as follows:

INTERNAL	Blocks are 5120 characters in length and contain binary records only.
CHARACTER-COUNT	Blocks always contain a fixed number of characters.
RECORD-COUNT	Blocks always contain a fixed number of records whether records are fixed or variable length.
EXACT-RECORDS	Blocks contain as many records as will fit within a given maximum block size.

If both BLOCK CONTAINS and BLOCK-TYPE clauses are omitted, CHARACTER-COUNT is assumed.

RECORD CONTAINS Clause

The RECORD CONTAINS clause establishes record size and CRM record type. The format of the RECORD CONTAINS clause is shown in figure 4-8.

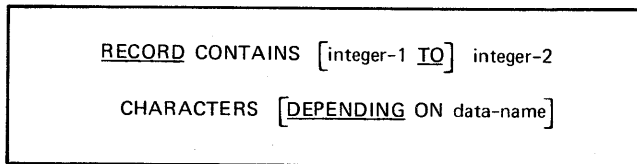


Figure 4-8. RECORD CONTAINS Clause Format

If records are fixed length, integer-1 is omitted and integer-2 must be a positive number specifying the number of characters in the record. The maximum size of integer-2 is 262142.

If records are variable length, integer-1 must be a positive number specifying the number of characters in the smallest record. Integer-1 should be large enough to include all the keys. If it is incorrect or omitted, DDL computes the correct value. Integer-2 specifies the number of characters in the largest record and cannot exceed 131071.

The following restrictions apply to the DEPENDING ON option:

- The option can be used only if CHARACTER-COUNT is specified in the RECORD-TYPE clause.
- The specified data-name, which determines record size, must be in the same location in every record of the file. The data-name entry must be elementary, must not exceed six characters, and must be included in a record description for the area.

If the clause is omitted, record size and type are determined as follows:

- Record size is determined by the RECORD-NAME clause as described under Record Description Entry.
- Record type is fixed length unless specified otherwise by the RECORD-TYPE clause.

RECORD-TYPE Clause

The RECORD-TYPE clause describes the CRM record type format and overrides a RECORD CONTAINS clause. The format of the RECORD-TYPE clause is shown in figure 4-9.

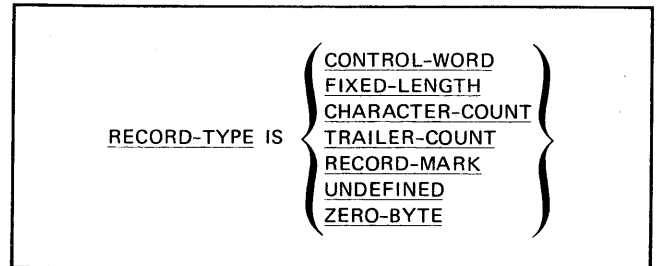


Figure 4-9. RECORD-TYPE Clause Format

One of seven record types recognized by CRM can be specified. Table 4-1 lists and describes each record type entry and indicates applicable rules and constraints.

LOG Clause

The LOG clause specifies the type of information that is to be recorded for information recovery and statistical analysis. Information consists of activity that occurs in response to Query Update directives following an INVOKE (or USE) directive for the area. The information is recorded in a log file. The format of the LOG clause is shown in figure 4-10.

A record image can be recorded as it appeared before or after a transaction. A transaction is defined as the activity that was imposed upon a record by a Query Update directive. The transaction itself can be recorded and consists of the text of the Query Update transmission. Transaction logging is restricted to transmissions that update or have the potential to update an area. Conditional transactions are logged whether or not the condition is satisfied. No logging is performed for query-only transactions.

If RECORD is not specified, the transaction is recorded. If only the words LOG, RECORD, and ASSIGN are specified, the before image of the record is recorded.

The log file must have been previously established as a permanent file. Query Update attaches the permanent log file as needed. The ASSIGN clause must specify the same information that is normally required for the attaching of a permanent file. The NOS operating system requires the M parameter with a permission mode of write or modify. Appropriate operating system documentation should be consulted for an explanation of the area-name qualifiers ID and UN and the permanent file parameters shown in the format.

Logging is performed by Query Update through the data base utility logging routines. Log file information is included in appendix G.

TABLE 4-1. RECORD TYPE DESCRIPTIONS

Option	CYBER Record Manager Type	Description	Comments
CONTROL-WORD	W	Binary records with a blocking factor of 5120 characters.	Must be specified if BLOCK-TYPE is INTERNAL.
FIXED-LENGTH	F	Records with fixed length.	Assumed if RECORD-TYPE clause is omitted.
CHARACTER-COUNT	D	Records containing a user-defined count field that gives the length of the records in characters.	Must be declared if RECORD CONTAINS clause with DEPENDING ON option is specified.
TRAILER-COUNT	T	Records containing a fixed header portion followed by a variable number of fixed-length trailer items.	Must be declared if both a RECORD-TYPE clause and a record section entry with OCCURS. . .DEPENDING ON are specified. Assumed if RECORD-TYPE clause is omitted and a record section entry with OCCURS. . .DEPENDING ON is specified.
RECORD-MARK	R	Records containing a special record delimiter.	
UNDEFINED	U	Records that impose no special action on the part of CRM.	
ZERO-BYTE	Z	Records in which trailing spaces are dropped and a zero (or terminating) byte is included.	

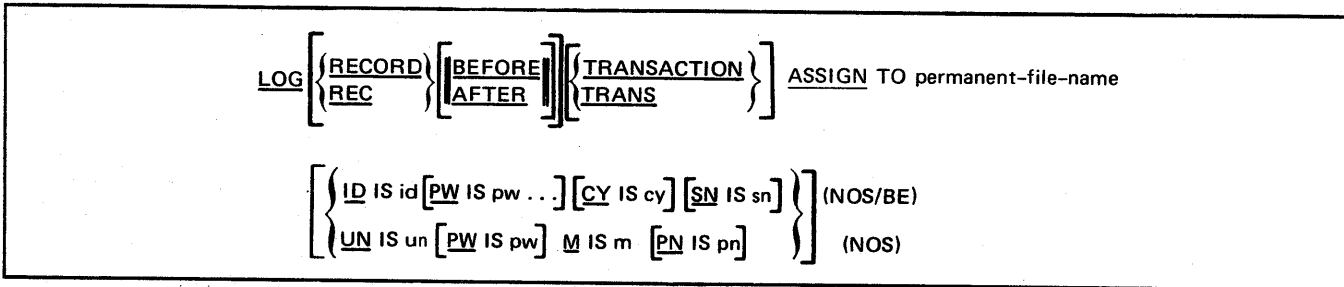


Figure 4-10. LOG Clause Format

ORGANIZATION Clause

The ORGANIZATION clause specifies the file organization and associated access method to be used when a data record is retrieved or updated. The format of the ORGANIZATION clause is shown in figure 4-11.

An area can be specified as direct access (DIRECT), sequential (SEQUENTIAL), indexed sequential (INDEXED), or actual key (ACTUAL). One mode must be specified for each area named in the data base. If the mode is omitted, unsorted sequential is assumed.

The OLD or NEW option can be used with indexed sequential, direct access, and actual key CRM Advanced Access Methods (AAM) files. OLD designates an initial AAM file; NEW designates an extended AAM file. For example, if the OLD and INDEXED options are selected, the file is considered to be an initial indexed sequential file; if the NEW and INDEXED options are selected, the file is considered to be an extended indexed sequential file. If neither OLD nor NEW is specified, OLD is assumed. All the rules that apply to initial AAM files apply to extended AAM files with the exception of computing key length in actual key files. See the Primary Key Option subsection for more information.

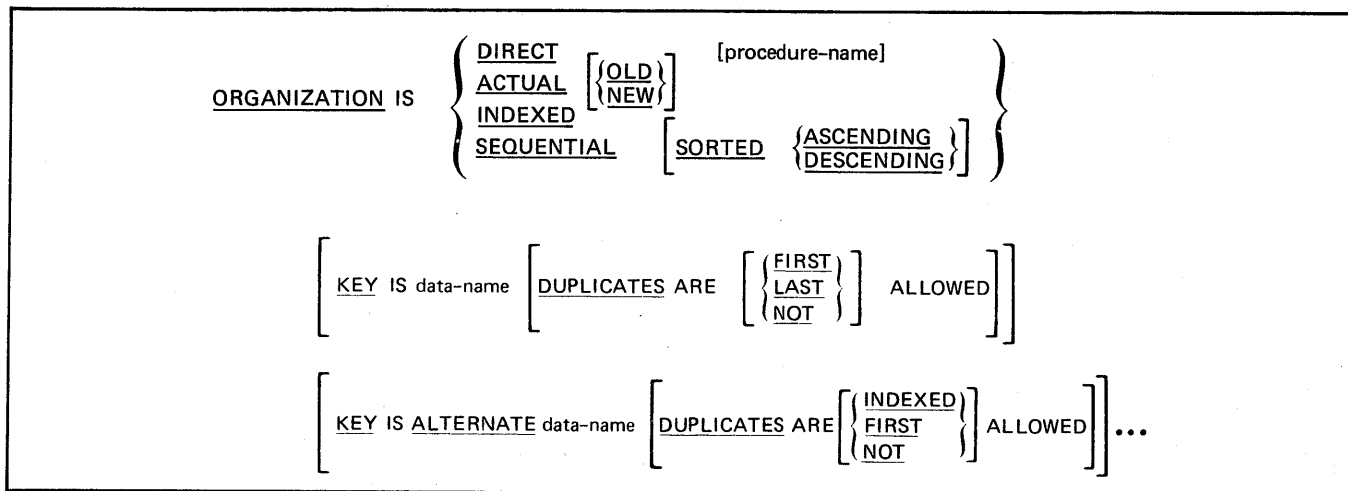


Figure 4-11. ORGANIZATION Clause Format

Direct access files are normally handled by AAM. A substitute hashing technique can be used by including the hashing routine name (procedure-name) option. This routine is loaded at the same time, from the same library, and according to the same sequence as data base procedures, described in appendix F. Query Update supplies the address of the hash routine to AAM. Refer to the CYBER Record Manager Advanced Access Methods reference manual for detailed specifications of hashing routine input and output.

The SORTED ASCENDING or DESCENDING option can be used only with sorted sequential files. A file can be sequenced in ascending order (low-to-high, A through Z) or descending order (high-to-low, Z through A). If the sort option is omitted, ascending sequence is assumed.

A key field cannot be an occurring field because it must be in the same position in every record. A key that is a group item cannot contain items of different usages. For indexed sequential files where the key is not part of the record, the name of the key is defined as the only field of the last record description for the area; the name of the record must be the word KEY-FIELD.

Primary Key Option

When file organization is DIRECT, SEQUENTIAL SORTED, INDEXED, or ACTUAL, the primary key must be specified within a record description. The primary key is declared by the KEY IS data-name option.

The primary key (data-name) is a sort control field that must be defined as one of the entries in an associated record description. When file organization is ACTUAL, the primary key must be specified in the record description with a USAGE IS INTEGER clause and can have a picture specification. Also, when the file organization is ACTUAL, the key length (KL) is computed in bits when OLD is specified or in characters when NEW is specified. If no PICTURE clause is specified, the default value for the key length is 48 (bits for OLD) or 8 (characters for NEW). When a PICTURE clause is specified, the picture value is used to compute key length up to a maximum of 48 (bits for OLD) or 8 (characters for NEW). In a picture specification resulting in a size greater than the maximum allowed, the key length defaults to the maximum value.

For example, assume the primary key item is as follows:

```
02 SAK-KEY PIC 9(5) USAGE IS INTEGER
```

If the file organization is ACTUAL OLD, the key length of SAK-KEY is 30 (bits). If the file organization is ACTUAL NEW, the key length of SAK-KEY is 5 (characters). If the file organization is DIRECT, SEQUENTIAL SORTED, or INDEXED, the key length is 10 (characters).

The DUPLICATES phrase refers to the retrieval of duplicate keys. Available combinations are as follows:

```
DUPLICATES
DUPLICATES ARE ALLOWED
DUPLICATES ARE LAST
```

Duplicate key occurrences are allowed. The initial occurrence of the key is retrieved first, followed by the duplicates in the order in which they were stored.

```
DUPLICATES ARE FIRST
```

Duplicate key occurrences are allowed. Duplicates are retrieved before the initial occurrence of the key. The last duplicate stored is the first retrieved.

```
DUPLICATES ARE NOT ALLOWED
```

Duplicate key occurrences are not allowed.

The DUPLICATES phrase is illegal under the following conditions:

- File organization is ACTUAL.
- File organization is DIRECT.
- File organization is INDEXED NEW.
- The multiple key option is selected.

If the DUPLICATES phrase is omitted, duplicates not allowed is assumed.

A major primary key, which is defined as the leading portion of a primary key field, can be specified in the following manner:

```
02 PRIMEKEY
    05 MAJORKEY PIC X(3)
    05 REST     PIC X(3)
```

Multiple Key Option

When file organization is INDEXED, ACTUAL, or DIRECT, multiple key fields can be specified within a record. When this option is selected, the primary key must not have duplicates. The Multiple-Index Processor accommodates record retrieval by alternate key data-name for these file organizations. The Multiple-Index Processor, which is an extension of CRM, maintains an index file for each alternate key.

Multiple keys are specified by the KEY IS ALTERNATE option. The name of the alternate key is specified by data-name. Any number of alternate keys can be defined. The only restriction applies to the alternate key size, which cannot exceed 255 characters.

The DUPLICATES phrase refers to the organization of the index file for the alternate key. Available combinations are as follows:

```
DUPLICATES
DUPLICATES ARE ALLOWED
DUPLICATES ARE INDEXED
```

Duplicate occurrences of the alternate key are allowed. In the index file, duplicates are stored in order (indexed) according to the primary key to which the alternate key corresponds.

```
DUPLICATES ARE FIRST
```

Duplicate occurrences of the alternate key are allowed. In the index file, duplicates are stored as they occur - first in, first out.

```
DUPLICATES ARE NOT ALLOWED
```

Duplicate occurrences of the alternate key are not allowed.

If the DUPLICATES phrase is omitted, duplicates not allowed is assumed.

A major alternate key, which is defined as the leading portion of an alternate key field, can be specified in the following manner:

```
02 ALTKEY
    05 MAJORALTKEY PIC X(3)
    05 REST-ALT   PIC X(3)
```

FILE-LIMIT Clause

The FILE-LIMIT clause specifies the maximum number of records in the file. The clause can be used when the area being created is an indexed sequential, direct access, or actual key file. The format of the FILE-LIMIT clause is shown in figure 4-12.

If this clause is omitted, no limit is placed on the number of records in the file.

FILE-LIMIT IS integer

Figure 4-12. FILE-LIMIT Clause Format

SEQUENCE Clause

The SEQUENCE clause specifies the collating sequence for the area. It is used in sort operations and in all comparisons involving items from the area. The format of the SEQUENCE clause is shown in figure 4-13.

SEQUENCE IS { FORTRAN }
 { COBOL }
 { literal }

Figure 4-13. SEQUENCE Clause Format

The COBOL collating sequence is the standard collating sequence and is the default assumed by DDL. The FORTRAN collating sequence is the same as display code.

The user can specify another collating sequence by defining a literal. The following rules apply to a user-defined collating sequence:

- The literal cannot exceed 64 characters.
- The literal can contain any characters from the standard character set, including blanks.
- The literal must be enclosed in quotation marks.
- The quotation mark character can be included and is specified by using double quotation marks.
- If the literal is split between two lines on the coding form, a hyphen must be specified in column 7 of the second line and a quotation mark must be specified in or after column 12 of the second line.
- All numeric and special characters not specified in the literal are treated as equivalents and are higher in the collation than the highest specified character.

INDEX-LEVEL Clause

The INDEX-LEVEL clause is used to determine the size of index blocks when an indexed sequential file is being created. The system uses the index level value and the maximum number of records given in the FILE-LIMIT clause to determine size. The format of the INDEX-LEVEL clause is shown in figure 4-14.

The specified integer range is 1 through 63 with a default value of 1.

INDEX-LEVEL IS integer

Figure 4-14. INDEX-LEVEL Clause Format

INDEX-BLOCK Clause

The INDEX-BLOCK clause can be used in place of the INDEX-LEVEL clause to specify the size of index blocks when an indexed sequential file is being created. The format of the INDEX-BLOCK clause is shown in figure 4-15.

```
INDEX-BLOCK CONTAINS integer CHARACTERS
```

Figure 4-15. INDEX-BLOCK Clause Format

The specified integer range is 1 through 32767 with a default value of 511.

INDEX-PADDING Clause

The INDEX-PADDING clause specifies the percentage of the index block to be reserved for padding when an indexed sequential file is being created. Padding permits file growth without additional index levels. The format of the INDEX-PADDING clause is shown in figure 4-16.

```
INDEX-PADDING IS integer PERCENT
```

Figure 4-16. INDEX-PADDING Clause Format

The specified integer range for the percentages is 0 through 99 with a default value of 5.

DATA-PADDING Clause

The DATA-PADDING clause specifies the percentage of the data block to be reserved for padding when an indexed sequential or actual key file is being created. This clause is also used to permit record expansion without increasing input/output time. The format of the DATA-PADDING clause is shown in figure 4-17.

```
DATA-PADDING IS integer PERCENT
```

Figure 4-17. DATA-PADDING Clause Format

The specified integer range for the percentages is 0 through 99 with a default value of 0.

NUMBER OF BLOCKS Clause

The NUMBER OF BLOCKS clause specifies the number of home blocks to be assigned when a direct file is being created. The format of the NUMBER OF BLOCKS clause is shown in figure 4-18.

```
NUMBER OF BLOCKS IS integer
```

Figure 4-18. NUMBER OF BLOCKS Clause Format

The maximum allowable value of the specified integer is 16777215. If the clause is omitted, a default value of 5 is used.

ON CALL Clause

The ON CALL clause specifies up to eight data base procedure exits to be taken by Query Update when specific conditions occur during processing. The format of the ON CALL clause is shown in figure 4-19.

```
ON | OPEN |  
   | SEARCH |  
   | MATCH |  
   | MISMATCH |  
   | UPDATE |  
   | DISPLAY |  
   | CLOSE |  
   | RETRIEVAL |  
CALL procedure-name
```

Figure 4-19. ON CALL Clause Format

The options in the ON CALL clause indicate both conditions and general procedure processing. The terms procedure and data base procedure are synonymous. The options are described as follows:

OPEN

Initialization is performed by the procedure when the area is opened at the beginning of processing of a transmission that accesses the area.

SEARCH

A record is supplied by the procedure in lieu of having Query Update read a record. When SEARCH is used, procedures must also be provided for the OPEN and CLOSE conditions.

MATCH

Processing of the selected record is performed by the procedure.

MISMATCH

Processing of a record that failed to satisfy the search is performed by the procedure.

UPDATE

Additional processing is performed by the procedure before the record is written to the area.

DISPLAY

The selected record is available in the record work area when this exit is called.

CLOSE

Processing is performed by the procedure prior to closing of the area at the end of a transmission that accessed the area.

RETRIEVAL

Additional processing is performed by the procedure to access a record immediately after CYBER Record Manager performs a read.

A detailed description of ON CALL statement usage is included in appendix F.

RECORD DESCRIPTION ENTRY

A record description entry follows its associated area description entry in the Data Division. One or more descriptions of a record can be specified within an area. The record description entry consists of the RECORD-NAME clause and a series of statements (data description entries) that name and describe records at the group and elementary levels. The format of the record description entry is shown in figure 4-20.

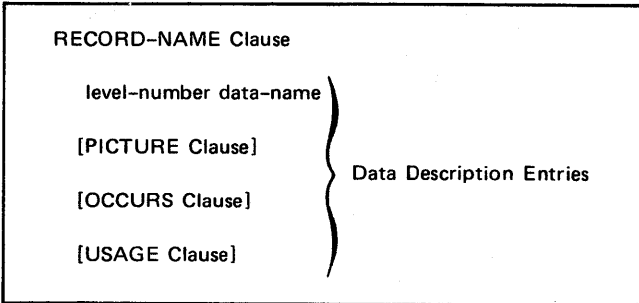


Figure 4-20. Formats of Record Description Entries

The data description entries describe the data as it is used by the Query Update program. Since the RECORD-NAME clause assumes a level of 01, data description entries begin with level numbers 02 through 49. DDL imposes a limit of 819 items per record.

The reserved word FILLER can be used in place of a data name to retain space in a record. No information can be referenced with the name FILLER.

RECORD-NAME Clause

The RECORD-NAME clause is a required clause that must be used for each record with a differing format. A level of 01 is implied for a record name. The format of the RECORD-NAME clause is shown in figure 4-21.

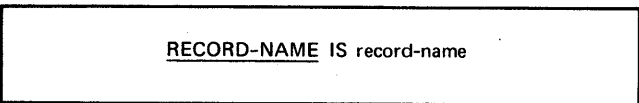


Figure 4-21. RECORD-NAME Clause Format

The record-name is user-assigned and must be unique to the sub-schema. The rules governing the formation of a record name are listed in section 3.

PICTURE Clause

The PICTURE clause describes the general characteristics of an elementary data item in terms of its size and class. The location of an operational sign or an assumed decimal point can also be indicated in the clause. The format of the PICTURE clause is shown in figure 4-22.

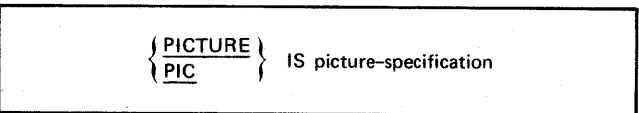


Figure 4-22. PICTURE Clause Format

This clause can be specified only for elementary data items. PIC is the legal abbreviation for PICTURE.

The class of a data item is determined by the type of characters in the picture-specification. The characters 9, S, V, and P are used to describe numeric data items. An alphabetic data item is described by the character A. The picture-specification for an alphanumeric data item contains the character X or any combination of the characters X, 9, and A.

The size of a data item is determined by the number of 9, X, or A characters in the picture-specification. The characters S, V, and P are not counted in determining the size. Consecutive identical characters in the string can be specified by a number in parentheses following the character. For example: 99999999 is equivalent to 9(8), XXXXXXXX is equivalent to X(8), and 9999AAAA is equivalent to 9(4)A(4); each indicates a data item with eight character positions.

The picture-specification can contain a maximum of 30 characters, including parentheses; however, a pictured item can be larger. A picture-specification containing the character A repeated 75 times is too long, but A(75) is a valid description for a data item with 75 alphabetic characters. Numeric data items can contain only 14 significant digits; additional leading or trailing zeros for decimal point alignment can be specified up to a total of 30 characters.

The format and punctuation applied to a numeric source data item at the time it is displayed are called editing. Editing characters are of two types:

- Insertion characters
- Replacement characters

The displayed format is referred to as a report item rather than a source data item. Insertion and replacement characters are not included in the size of a numeric source item. Insertion and replacement characters must be included in the size of a report item.

Alphabetic Data Items

The picture-specification to describe an alphabetic data item can contain only the character A. The A can be specified as many times as necessary as long as the size of the item does not exceed 2047 characters.

The function of the characters in a PICTURE clause picture-specification for an alphabetic data item is as follows:

- A Each A in the picture-specification represents a character position that can contain either a letter of the alphabet or a space (blank).

Some typical alphabetic data items are shown in figure 4-23.

Picture-Specification	Data Value	Display Code Stored
AAAAA or A(5)	COSTS	C O S T S
AAAA or A(4)	WXYZ	W X Y Z

Figure 4-23. Alphabetic Data Items

Numeric Data Items

The picture-specification to describe a numeric data item can contain a combination of the characters 9, S, V, and P as well as insertion and replacement characters for editing. Each 9 represents a significant digit; a maximum of 18 digits can be specified (only 14 digits are significant). The character P is used to indicate leading or trailing zeros. The combined number of positions indicated by the characters 9 and P cannot exceed 30 characters.

The function of the characters in a PICTURE clause picture-specification for a numeric data item is as follows:

- 9 Each 9 in the picture-specification represents a digit position that can contain a number. The 9 is counted in determining the size of the data item.
- S The character S is used in the picture-specification to indicate that the data item has an operational sign. It must be the leftmost character in the picture-specification; it cannot appear more than once. The operational sign does not occupy a character position in the data item and is not counted in its size. The operational sign cannot be used with the period editing character.
- V The character V is used in the picture-specification to indicate the position of an assumed decimal point. A V as the rightmost character in the picture-specification is redundant. The character V cannot appear more than once in the picture-specification. Since the assumed decimal point does not occupy a character position, the V is not counted in the size of the data item.
- P The character P in the picture-specification indicates an assumed decimal scaling position. It is used to specify an assumed decimal point when its position is not within the number that appears in the data item. If the assumed decimal point extends beyond the rightmost digit, one P is specified for each implied position between the rightmost digit and the assumed decimal point.

Similarly, if the assumed decimal point extends beyond the leftmost digit, one P is specified for each implied position between the leftmost digit and the assumed decimal point. The item is treated as if a zero were substituted for each P and a decimal point were placed before the P or after the last P. Since the character P indicates an assumed decimal point, the character V in the picture-specification would be redundant. The character P is not counted in determining the size of the data item; however, it is counted in determining the maximum number of digit positions (30) in numeric data items.

Some typical numeric data items are shown in figure 4-24.

Alphanumeric Data Items

The picture-specification to describe an alphanumeric data item contains either a combination of the characters 9, A, and X or only the character X. The size of the data item cannot exceed 2047 characters.

The function of the characters 9 and A is the same as for numeric and alphabetic data items; the function of the character X is as follows:

- X Each X in the picture-specification represents a character position that can contain any character in the DDL character set.

Some typical alphanumeric data items are shown in figure 4-25.

Insertion Characters

Insertion characters in the picture of an item appear in the edited data item and are counted in the size of the report item. A maximum of 63 characters is allowed. Insertion characters for the PICTURE clause are as follows:

- \$ A single dollar sign specified as the leftmost symbol in an item picture appears as the leftmost character in a report item.

Picture-Specification	Data Value	Display Code Stored
999	123	1 2 3
99V999	12345	1 2 3 4 5
S99V99	+1234	1 2 3 4 ⁺
PPP9999	.0001234	↑0 0 0 1 2 3 4
SPPP9999	-.0001234	↑0 0 0 1 2 3 4 ⁻
S999PPP	-123000.	1 2 3 ⁻ 0 0 0 ↑

Figure 4-24. Numeric Data Items

<u>Picture-Specification</u>	<u>Data Value</u>	<u>Display Code Stored</u>
XXXXXXXX or X(8)	ABCD-***	A B C D - * * *
XXXXXXXX or X(8)	123.4567	1 2 3 . 4 5 6 7
AAAA999	ABCD123	A B C D 1 2 3
A(4)9(3)	ABCD123	A B C D 1 2 3

Figure 4-25. Alphanumeric Data Items

- + A plus sign specified as the first or last symbol in an item picture is inserted in the indicated character position of the report item provided the data contains a positive operational sign or is unsigned. If the data is negative, a minus sign is inserted in the indicated character position.
- A minus sign specified as the first or last symbol in an item picture is inserted in the indicated character position of the report item provided the data contains a negative operational sign. If the data is positive, a blank is inserted in the indicated character position.
- .
- . A period is used in a report item picture to represent a decimal point. When it is used, a decimal point appears in the report item as a character in the indicated character position.
- ,
- . A comma used in the picture of an item is inserted in the corresponding character position of the report item.
- /
- / A slash used in the picture of an item is inserted in the corresponding character position of the report item.
- 0
- 0 A zero used in the picture of an item is inserted in the corresponding character position of the report item.
- B
- B The character B used in the picture of an item causes a blank to be inserted in the corresponding character position of the report item.
- CR
- CR The CR symbol represents credit and can be specified only at the rightmost position of the picture of an item. The symbol is inserted in the last two character positions of the report item provided the data contains a negative operational sign. If the data is positive or unsigned, the last two character positions are set to blanks. Since the credit symbol always results in two characters - CR or blanks - it is included as two characters in the size of the report item.
- DB
- DB The DB symbol represents debit and can be specified only at the rightmost position of the picture of an item. The debit symbol has the same results as the credit symbol.

Replacement Characters

A replacement character in the picture of an item suppresses leading zeros in the source data and replaces them with the specified character or a blank in the report item. Only one replacement character can be used in a picture. Replacement characters for the PICTURE clause are as follows:

- Z One character Z is specified as the leftmost symbol in an item picture for each leading zero to be suppressed and replaced by blanks. The character Z can be preceded by one of the insertion characters and interspersed with any of the insertion characters decimal point, comma, zero, or B.

No zeros are suppressed to the right of the first nonzero digit whether a Z is present or not; nor are any zeros to the right of an assumed or actual decimal point suppressed unless the value of the data is zero and all character positions in the item are described by a Z. In this special case, even the actual decimal point is suppressed and the edited item is all blanks.

If a \$ + or - precedes the Z characters, it is inserted in the far left character position of the item even if succeeding zeros in the item are suppressed. In the special case where the value of the data is zero and all the character positions following the \$ + or - are Zs, the \$ + or - is replaced by blanks.

If a comma, zero, or B is encountered before zero suppression terminates, the character is not inserted in the edited data item, but it is suppressed and a blank inserted in its place.

- * The asterisk causes leading zeros to be replaced by an asterisk instead of a blank. It is specified in the same way as the editing character Z and follows the same rules, except an actual decimal point is not replaced by an asterisk when the value of the data is zero.

- \$ When the dollar sign is used as a replacement character to suppress leading zeros, it acts as a floating sign and is inserted directly preceding the first nonsuppressed character. One more dollar sign than the number of zeros to be suppressed must be specified. This dollar sign is always present in the edited data whether or not any zero suppression occurs. The remaining dollar signs act in the same way as the Z characters to suppress leading zeros.

Examples of insertion characters are shown in figure 4-26.

<u>Picture-Specification</u>	<u>Data Value</u>	<u>Displayed Item</u>
\$99	48	\$ 4 8
\$99.99	4834	\$ 4 8 . 3 4
9,999	4834	4 , 8 3 4
+999	292	+ 2 9 2
+999	292	+ 2 9 2
+999	292	- 2 9 2
999-	292	2 9 2 -
-999	292	Δ 2 9 2
999-	292	2 9 2 Δ
\$BB999.99	24321	\$ Δ Δ 2 4 3 . 2 1
\$00999.99	24321	\$ 0 0 2 4 3 . 2 1
99.99CR	1134	1 1 . 3 4 C R
99.99CR	1134	1 1 . 3 4 Δ Δ
99.99DB	2376	2 3 . 7 6 D B
99.99DB	2376	2 3 . 7 6 Δ Δ

Figure 4-26. Examples of Insertion Characters

- + When a plus sign is used as a replacement character, it is a floating sign. The plus sign is specified one more time than the number of leading zeros to be suppressed. It functions in the same way as the floating dollar sign. A plus sign is placed directly preceding the first nonsuppressed character if the edited data is positive or unsigned; a minus sign is placed in this position if the edited data is negative.
- When a minus sign is used as a replacement character, it is a floating sign. The minus sign is specified one more time than the number of leading zeros to be suppressed. It functions in the same way as the floating plus sign. A minus sign is placed directly preceding the first nonsuppressed character if the edited data is negative; a blank is placed in this position if the edited data is positive or unsigned.

<u>Picture-Specification</u>	<u>Data Value</u>	<u>Displayed Item</u>
Z999	00923	Δ Δ 9 2 3
ZZ99	00923	Δ Δ 9 2 3
ZZZZ.ZZ	000000	Δ Δ Δ Δ Δ Δ Δ Δ
S***.99	00923	\$ * * 9 . 2 3
\$\$\$9.99	000824	Δ Δ \$ 8 . 2 4
---9.99	00526	Δ Δ - 5 . 2 6
\$\$\$99	3265	\$ 3 2 . 6 5

Figure 4-27. Examples of Replacement Characters

Examples of replacement characters are shown in figure 4-27. Examples of picture editing are shown in figure 4-28.

OCCURS Clause

The OCCURS clause is used to indicate a repeated data item where all occurrences of the data item are identical in every respect except value. The data item can be an elementary item or a group item. The format of the OCCURS clause is shown in figure 4-29.

A data description entry with format 1 of the OCCURS clause can be subordinate to another entry with either format of the OCCURS clause. An entry with format 2 cannot be subordinate to an entry with the OCCURS clause. Up to three levels of nested data items can be specified with the OCCURS clause. Each group or elementary item with an OCCURS clause constitutes a level of nesting if it is subordinate to a repeating group. When repeating groups and vectors are nested, the vector is considered to be a repeating group and is included in the level count.

Picture-Specification	Data Value	Displayed Item
ZZZ,999.99	12345	△ 1 2 , 3 4 5 . 0 0
Z99,999.99	1234	△ 0 0 , 0 1 2 . 3 4
\$ZZZ,ZZ9.99	↑ 123	\$ △ △ △ △ △ △ 1 . 2 3
\$ZZZ,ZZZ.99	↑ 12	\$ △ △ △ △ △ △ △ . 1 2
\$***,**9.99	↑ 1234	\$ * * 1 , 2 3 4 . 0 0
\$***,***.99	123456	\$ 1 2 3 , 4 5 6 . 0 0
\$***,***.99	↑ 123	\$ * * * * * 1 . 2 3
+999,999	↑ 12	+ 0 0 0 , 0 1 2
-ZZZ,ZZZ	↑ 12	- △ △ △ △ △ 1 2
\$ZZZ,ZZ9.99CR	123456	\$ 1 2 3 , 4 5 6 . 0 0 C R
\$ZZZ,ZZ9.99DB	↑ 123	\$ △ △ △ △ △ △ 1 . 2 3 △ △
S(4),S\$9.99	↑ 1234	△ △ △ △ \$ 1 2 3 . 4 0
S(4),S\$.99	↑ 0000	△ △ △ △ △ △ △ \$. 0 0
----- .99	↑ 12	△ △ △ △ △ △ △ - . 1 2
BBBB,BBB.99	↑ 12	△ △ △ △ △ △ △ △ . 1 2
SSSS,SZZ.99	↑ 12	illegal picture
S99.99	12	illegal picture

Figure 4-28. Examples of Picture Editing

<u>Format 1</u>
<u>OCCURS</u> integer-2 TIMES
<u>Format 2</u>
<u>OCCURS</u> [integer-1 <u>TO</u>] integer-2 TIMES <u>DEPENDING ON</u> data-name-1

Figure 4-29. OCCURS Clause Format

Integer-1 and integer-2 must be positive numbers and cannot exceed 262142. In format 2, integer-2 must be greater than integer-1. The value of integer-1 can be zero; integer-2 must never be zero.

An elementary data item described with the OCCURS clause must also be described with the PICTURE or USAGE clause. A group data item cannot be described with both the OCCURS clause and the PICTURE clause. FILLER can never be used with the OCCURS clause.

When an item occurs a fixed number of times, format 1 is used and integer-2 specifies the exact number of occurrences. When an item occurs a variable number of times, format 2 is used and the number of occurrences for each record is determined as follows:

- Integer-1 represents the minimum number of occurrences; default minimum is zero.
- Integer-2 represents the maximum number of occurrences.
- Data-name-1 references a data item whose current value represents the number of occurrences. The value of data-name-1 must be a positive value within the range of integer-1 through integer-2. Data-name-1 must appear in the area section in the same logical record as the entry containing the OCCURS clause and must be defined before it is referenced.

In format 2, data-name-1 names an elementary item that is unique or can be made unique by qualification. It cannot be subscripted and it must be described as COMPUTATIONAL, COMPUTATIONAL-1, or DISPLAY. The size of the data item cannot exceed six characters. The elementary data item must precede the group data item that references it.

A data item that occurs a variable number of times must be the last item in a record description entry. The data description entry that contain the OCCURS clause can only be followed by subordinate entries.

If both the OCCURS and RECORD CONTAINS clauses are specified and the RECORD CONTAINS clause does not include DEPENDING ON, the OCCURS specification overrides the RECORD CONTAINS clause. Specification of both OCCURS...DEPENDING ON and RECORD CONTAINS...DEPENDING ON is illegal.

The total number of characters in all occurrences of a data item must be included when calculating the record size. The maximum size of a record is 262142 characters.

USAGE Clause

The USAGE clause specifies the internal representation of a data item in terms of the primary use of the data item. The clause specifies whether an item is used predominantly for computation or for display, but it does not necessarily limit the item to the usage specified. For example, a numeric display item can be used for computation. The format of the USAGE clause is shown in figure 4-30.

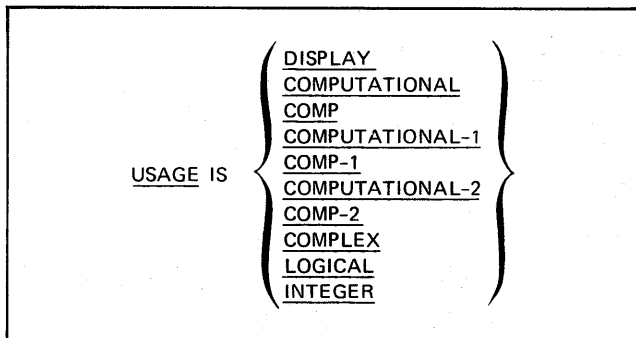


Figure 4-30. USAGE Clause Format

This clause can describe a data item at any level. When the USAGE clause is specified for a group item, it applies to each subordinate item in the group. The USAGE clause of an elementary item in a group cannot contradict the USAGE clause of the group item.

A data item described as COMPUTATIONAL, COMPUTATIONAL-1, or COMPUTATIONAL-2 must be numeric with a size not exceeding 18 digits. If a group item is described as computational, the elementary items within the group are all computational; however, the group item itself is not computational and cannot be used in computations.

COMP, COMP-1, and COMP-2 are legal abbreviations for COMPUTATIONAL, COMPUTATIONAL-1, and COMPUTATIONAL-2.

DISPLAY Option

A DISPLAY data item is stored in display code format. The data item can be alphabetic, numeric, or alphanumeric. When no USAGE clause is associated with a data item, DISPLAY is the default format.

COMPUTATIONAL Option

A COMPUTATIONAL data item has a decimal numeric value. The PICTURE picture-specification can contain the characters 9 (digit position), S (operational sign), V (implied decimal point), and P (assumed decimal scaling position); editing characters can also be included.

COMPUTATIONAL-1 Option

A COMPUTATIONAL-1 data item is stored as an unnormalized floating-point binary representation of a decimal number. When the size of the data item is 14 digits or less, the data item is single-precision and contained in one computer word. When the size of the data item is 15 to 18 digits, the data item is double-precision and requires two computer words. A PICTURE clause can be included with this option if editing is desired.

A COMPUTATIONAL-1 data item created by a COBOL 5 program must be read as an integer (refer to INTEGER option).

COMPUTATIONAL-2 Option

A COMPUTATIONAL-2 data item is stored as a normalized floating-point binary representation of a decimal number. The decimal point location is carried in the data item itself as a binary exponent. The data item is single-precision and is stored in one computer word. Size is immaterial for a COMPUTATIONAL-2 data item; a PICTURE clause can be included with this option and is meaningful only for editing.

COMPLEX Option

A COMPLEX data item is an ordered pair of signed or unsigned real constants; the first represents the real part of the complex number and the second represents the imaginary part of the complex number. A PICTURE clause can be included with this option if editing is desired.

LOGICAL Option

A LOGICAL data item assumes only the values of true or false. When this option is selected, Query Update displays the true/false condition rather than the value of the described data.

INTEGER Option

An INTEGER data item is a number written without a decimal point or exponent field. It can consist of a maximum of 18 decimal digits within the range of $-(2^{48}-1) \leq n \leq (2^{48}-1)$. Integers used as subscripts are limited to $2^{17}-2$. Integer constants can be positive, zero, or negative and must be within the allowed magnitude; they cannot contain embedded commas.

The integer option can be used to read COBOL 5 COMPUTATIONAL-1 data items that are scaled integers. To specify a COBOL 5 scaled integer in a Query Update sub-schema, the picture-specification must represent a fractional part, as in the following example:

```
02 DATA-ITEM PIC 99.99 USAGE IS INTEGER
```

If the data item is assigned the value 12.34, the data base area actually contains the integer 1234. When the data item is referenced by a program, Query Update scales the item by dividing it by 100, giving the value 12.34.

RELATION DIVISION

The Relation Division is the last division in the DDL source program. It is optional and if included it must immediately follow the Data Division. The Relation Division identifies the specific area relationships that are to be used when querying more than one area of the data base and specifies the qualification criteria that determine the projected records that can be selected for the Query Update user. The format of the Relation Division is as follows:

RELATION DIVISION.

```
{ relation description entry } ...
```

RELATION DESCRIPTION ENTRY

The relation description entry is a statement consisting of the **RELATION-NAME**, **JOIN**, and **RESTRICT** clauses, which specify the relation name, the join terms, and the record qualification restrictions, respectively. Each relation description entry identifies one relation. The format of the relation description entry is shown in figure 4-31.

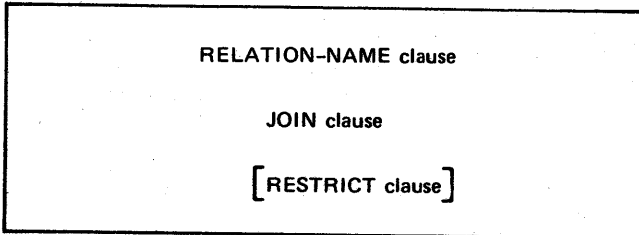


Figure 4-31. Relation Description Entry Format

RELATION-NAME Clause

The **RELATION-NAME** clause assigns a name to the relation. It must be the first clause specified in the relation description entry. The format of the **RELATION-NAME** clause is shown in figure 4-32.

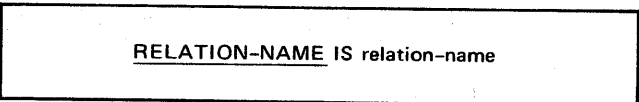


Figure 4-32. RELATION-NAME Clause Format

The relation-name must be unique among all area-names and relation-names in the sub-schema. It can be up to 30 characters in length. An area can be included in more than one relation. Areas that have sequential, indexed sequential, direct access, and actual key file organizations can be joined in a relation. An area that is joined in a relation can contain only one record description.

JOIN Clause

The **JOIN** clause specifies the data items that Query Update must inspect to join the areas in which the data items reside. The order in which the join terms are specified determines the direction of the relationship. A join term can be a full primary key, a major primary key, a full alternate key, a major alternate key, or a non-key data item. The **JOIN** clause is required if a **RELATION-NAME** clause is present. The format of the **JOIN** clause is shown in figure 4-33.

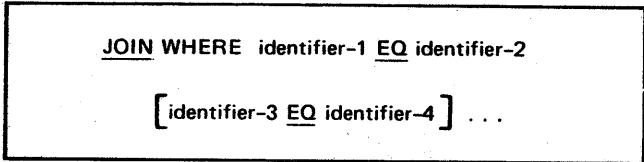


Figure 4-33. JOIN Clause Format

Within the **JOIN** clause a specified data item in one area is equated with an identical data item in another area; the two areas are thus related through specification of a common data item. The relational operator **EQ** must appear between each pair of identifiers included in the **JOIN** clause; the source identifier appears to the left of the **EQ** and the target identifier appears to the right.

The term identifier reflects a unique reference to a data-name; it implies that the data-name is referenced uniquely through a combination of subscripts and qualifiers. Refer to section 3 for the identifier format.

Each source identifier (except the first) must be in the same area as the previous target identifier, so that a continuous path from one area to the next area is defined. Cycling is not allowed; an area cannot be joined to itself directly or indirectly.

The identifiers to the left and to the right of a relational operator must have identical characteristics in the picture-specification and in the **USAGE** specification. The position of the data items within the records need not be identical. The identifier must not specify a data item that is more than 255 characters long. Subscripting and qualification can be specified for an identifier. The data item referenced by an identifier can be any of the following types of data items:

- A nonrepeating elementary item (no subscript)
- A specific occurrence of a repeating elementary item or an occurrence of a repeating group item (up to three subscripts)
- All occurrences of a repeating data item if it is an alternate key or the major part of an alternate key and is specified to the right of the **EQ** in the **JOIN** clause

If a repeating elementary item or group item is specified without a subscript, the first occurrence is used to join the relationship.

Generally, if a target identifier is defined as a key field, retrieval of record occurrences from the file containing the target identifier is by that key. This rule holds for repeating or nonrepeating elementary items. Only the first occurrence of a repeating item can be defined as a primary key. If a repeating group item or a repeating elementary item is defined as an alternate key, all occurrences of the item or group item are used as the key, rather than just the first occurrence. The ANY option used as a subscript identifies this type of alternate key field as a join term. If any one of the item occurrences satisfies the equality, the record occurrence is included in the relationship occurrence.

If the target identifier is a major part of a repeating alternate key, then a given record occurrence could be retrieved more than once. CHILD-NAME, in the example, is defined as an alternate key and TWO-LETTERS as the target identifier (a major part of the alternate key):

```
02 CHILD-NAME OCCURS 3 TIMES
    03 TWO-LETTERS PICTURE X(2)
    03 OTHERS PICTURE X(8)
```

Now, if a particular record occurrence contains the values JOAN, JOE, and JOHN, it will be delivered three times (rather than once) when retrieved by the relational mechanism.

More than one relation can be defined joining two given areas, thus enabling network data structures to be described where multiple relations exist between two record descriptions. Since a query can be made against only one relation at a time, user intervention is required when multiple relations exist. To minimize the need for user intervention, only one relation joining any given pair of areas should be specified in a sub-schema. Having one sub-schema per relation, with appropriate RESTRICT clauses for the records and passwords for the areas, is a good method of tailoring the sub-schema to a specific application or user population and enforcing privacy and security rules.

RESTRICT Clause

The optional RESTRICT clause specifies the qualification criteria that must be satisfied before a projected record can be selected for the Query Update user. The format of the RESTRICT clause is shown in figure 4-34.

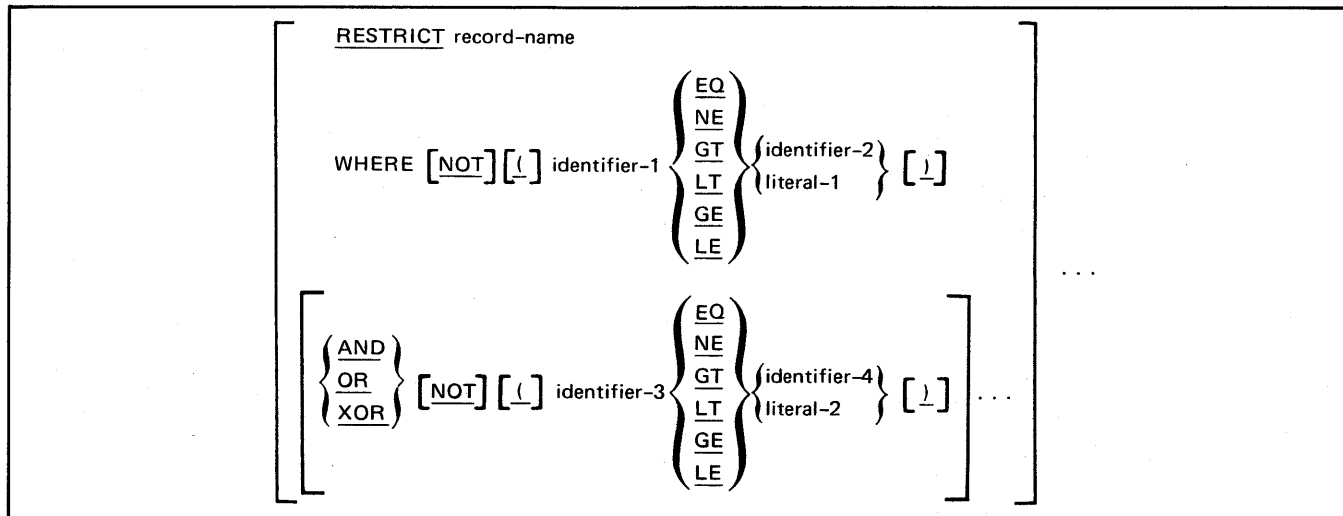


Figure 4-34. RESTRICT Clause Format

Record-name designates the record description to which the qualification restrictions pertain. The record description must appear in the Data Division of the sub-schema. The record-name must be contained in an area that has been joined by the relation named in the preceding RELATION-NAME clause.

A separate RESTRICT clause must be specified for each record description in which qualification is required. A given record can appear in only one RESTRICT clause for a particular relation. The RESTRICT clauses must appear in the same order in which the areas containing the record descriptions are joined in the relation definition. Refer to the example in figure 4-35.

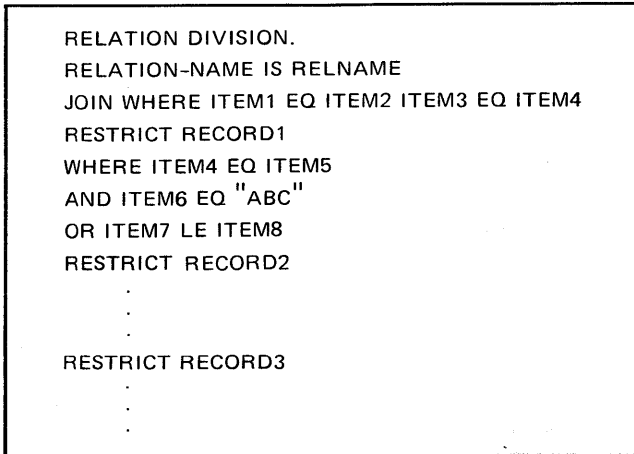


Figure 4-35. RESTRICT Clause Example

The identifiers represent data items whose values are examined by Query Update to determine whether the projected record qualifies for selection. They are termed qualifier identifiers. Refer to section 3 for the identifier format.

All qualifier identifiers must be defined in the record named in the RESTRICT clause. An identifier can be used as both a join term and a qualifier identifier in the relation description entry.

A qualifier identifier can be a subscripted data item but cannot be a data item that is within a repeating group subordinate to another repeating group. For example in the following data description, Z(1) or Z(2) can be defined as qualifier identifiers, but occurrences of A, B, or C cannot be qualifier identifiers because they are subordinate to Z.

- 02 Z OCCURS 2 TIMES
- 03 A OCCURS 3 TIMES
- 04 B PIC X(10)
- 04 C PIC X(5)

Qualifier identifiers cannot be described with the LOGICAL option in the USAGE clause. Only constant integers are allowed as item subscripts.

Literals specified in the RESTRICT clause must be compatible with the representation of the identifier corresponding to the literal. Literals cannot be figurative constants such as LAST, NEXT, and so on.

A maximum of 1024 entities (identifiers, operators, and literals) can appear in any one RESTRICT clause.

The Query Update sub-schema is coded on a standard 80-column coding sheet according to the specifications in this manual. Coding information is punched on 80-column cards or entered through a terminal. The resulting source program and various control statements are input to the DDL compiler to compile the sub-schema and to store it in the sub-schema library. The control statements provide information for the operating system and for the DDL compiler. A set of control statements, beginning with a job statement and ending with a 7/8/9 card or its equivalent, precedes the DDL source program.

SUB-SCHEMA LIBRARY

One or more compiled sub-schemas are stored in a permanent file called the sub-schema library. The library is created when the first sub-schema is stored in it. Subsequent sub-schemas can be added to the library or can replace existing sub-schemas in the library. A sub-schema can be deleted from the library through the use of the purge parameter in the DDL control statement. A sub-schema that has had sub-schemas replaced or purged can be transferred to a new, compacted sub-schema library.

Data security can be maintained by creating more than one sub-schema library to control the availability of the sub-schemas to the Query Update programs. Sub-schemas providing access to data that is restricted to specific

applications can be stored in one sub-schema library while sub-schemas for general use can be stored in a different library. Each library is identified by a unique permanent file name. This permanent file name is specified in a Query Update INVOKE (or USE), CREATE, or VERSION directive that uses a sub-schema in the library.

DDL CONTROL STATEMENT

The DDL control statement must be included in the set of control statements preceding the DDL source program. It provides the DDL compiler with information related to a specific Query Update sub-schema. The format and description of the DDL control statement are shown in figure 5-1.

Only one library manipulation function can be performed in an execution of DDL. If more than one function is specified, a control statement error is issued. If a library manipulation function is attempted on a nonempty file that does not contain a library, a diagnostic is issued and the job is aborted.

NOS/BE CONTROL STATEMENTS

In addition to the DDL control statement, standard NOS/BE control statements are included in the set of control statements preceding the DDL source statements.

DDL3(QD,SB=Ifn,p1,p2,p3,p4,p5,p6,p7)		
DDL3		Identifies the DDL control statement.
QD		Specifies a Query Update sub-schema in CRM data base access mode.
SB=Ifn		Specifies the logical file name (Ifn) of the sub-schema library.
p1	I=Ifn	An optional parameter that specifies a nondefault input file; Ifn is the logical file name of the source input file; default is INPUT.
p2	L=Ifn	An optional parameter that specifies a nondefault output file; Ifn is the logical file name of the source listing file; default is OUTPUT. If L=0 is specified, only diagnostics and associated statements are listed.
p3	R	An optional parameter that replaces the existing sub-schema in the sub-schema library with the sub-schema compiled from the source program in the input deck. Replacement takes place only if no compilation errors other than informative diagnostics are encountered.
p4	N	An optional parameter that compiles the sub-schema but does not add it to the sub-schema library.
p5	P	An optional parameter that purges the specified sub-schemas from the sub-schema library; no compilation takes place.
p6	A	An optional parameter that produces a list of sub-schemas, together with their creation dates, from the library indicated by the SB parameter.
p7	NL=Ifn	An optional parameter that transfers the active sub-schemas in the sub-schema library (indicated by the SB parameter) to a new sub-schema library (indicated by the NL parameter). Also, use of the NL parameter produces a list of sub-schemas, together with their creation dates.

Figure 5-1. DDL Control Statement Format

Refer to the NOS/BE reference manual for complete descriptions of the NOS/BE control statements. Two control statements that must be used in specific instances are described in the following paragraphs.

REQUEST CONTROL STATEMENT

The REQUEST control statement is included in the set of control statements for the first sub-schema to be stored in the sub-schema library. This statement specifies the logical file name of the sub-schema library. The format and description of the REQUEST control statement are shown in figure 5-2.

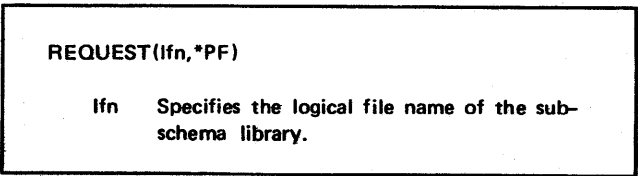


Figure 5-2. REQUEST Control Statement Format

CATALOG CONTROL STATEMENT

The CATALOG control statement is used to catalog the sub-schema library in a permanent file. This statement must be included in the set of control statements for the first sub-schema to be stored in the library. If more than one sub-schema is to be stored in the library, the extend (EX) and modify (MD) parameters must be specified. The format and description of the CATALOG control statement are shown in figure 5-3.

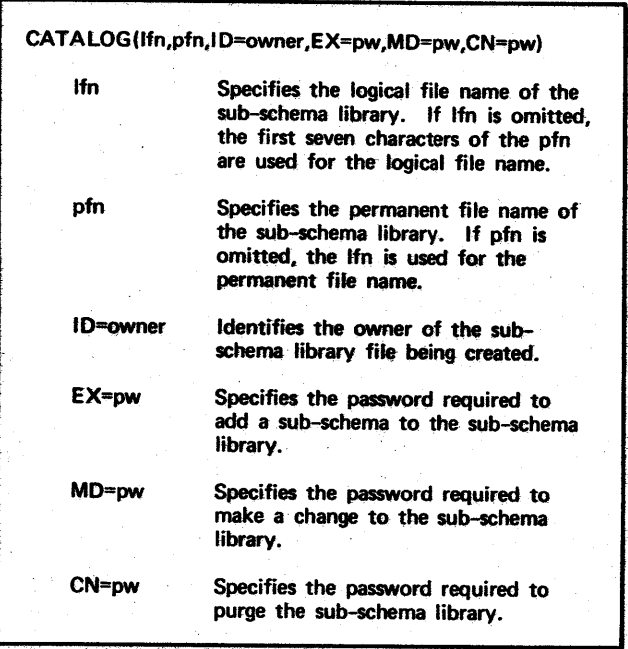


Figure 5-3. CATALOG Control Statement Format

NOS CONTROL STATEMENTS

The set of control statements preceding the DDL source statements include the DDL control statement and standard NOS control statements. Refer to the NOS reference manual for complete descriptions of the control

statements. The DEFINE control statement is used to create a sub-schema library file. The format and description of the DEFINE control statement are shown in figure 5-4.

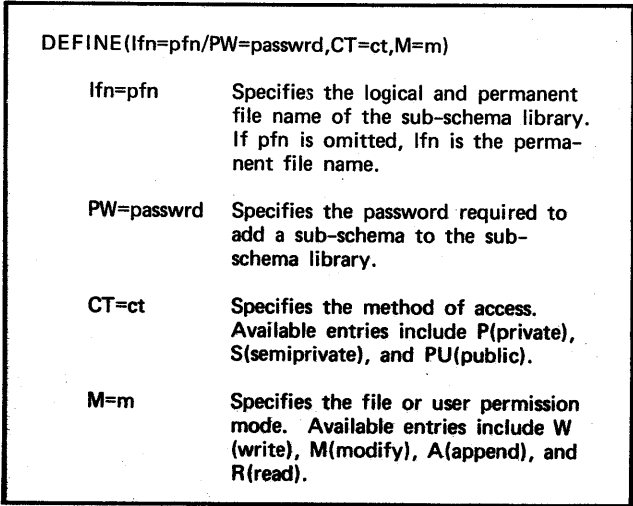


Figure 5-4. DEFINE Control Statement Format

SAMPLE DECK STRUCTURES

The deck structures shown in the following paragraphs illustrate the control statements required to compile, store, and delete Query Update sub-schemas. Control statements are shown for both the NOS/BE and NOS operating systems.

COMPILING A SUB-SCHEMA

The source program for a Query Update sub-schema can be compiled without storing the sub-schema in a library. This is accomplished with a deck structure similar to the one shown in figure 5-5. The DDL3 control statement must be included in the control statements, and the compile parameter (N) must be specified.

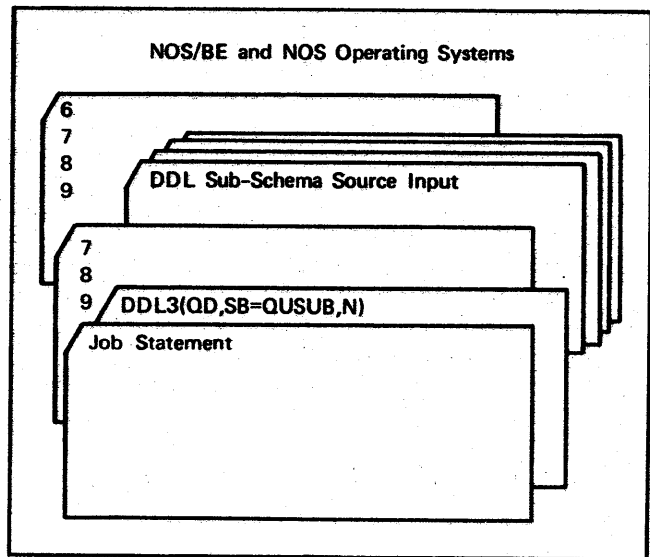


Figure 5-5. Compiling a Sub-Schema

CREATING A SUB-SCHEMA LIBRARY

The sub-schema library is created when the first sub-schema is stored in it. Figure 5-6 illustrates a deck structure used to compile a sub-schema and create a sub-schema library.

In this example, the REQUEST/CATALOG and DEFINE control statements specify the logical file name of the sub-schema library (QUSUB) and assign it to a permanent file device. If only one sub-schema is to be stored in the library, the sub-schema name can be used for the library file name. The DDL3 control statement names the sub-schema library file.

ADDING TO THE SUB-SCHEMA LIBRARY

Once the sub-schema library has been created and made a permanent file as described in the Creating a Sub-Schema Library subsection, new sub-schemas can be added to the library. The deck structure illustrated in figure 5-7 adds a sub-schema to the sub-schema library created by the example in figure 5-6.

The ATTACH control statement in this example is required to attach the sub-schema library file QUSUB. The DDL3 control statement requires only the QD and SB parameters.

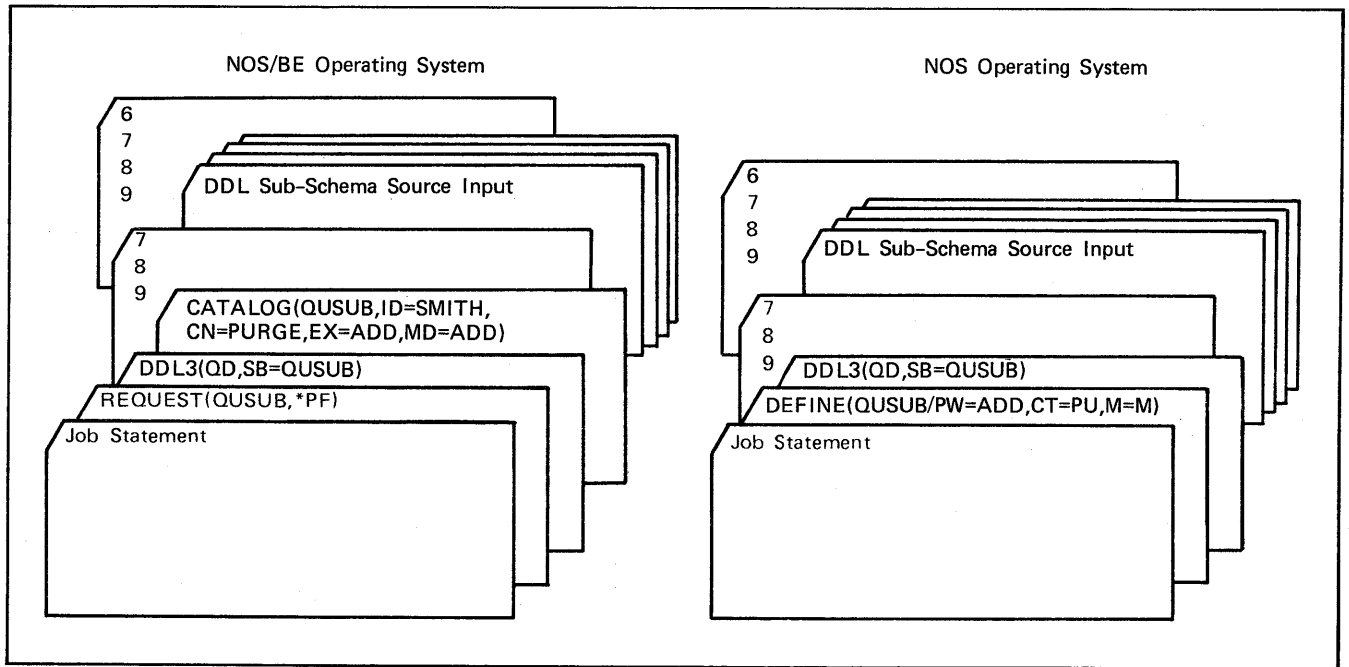


Figure 5-6. Creating a Sub-Schema Library

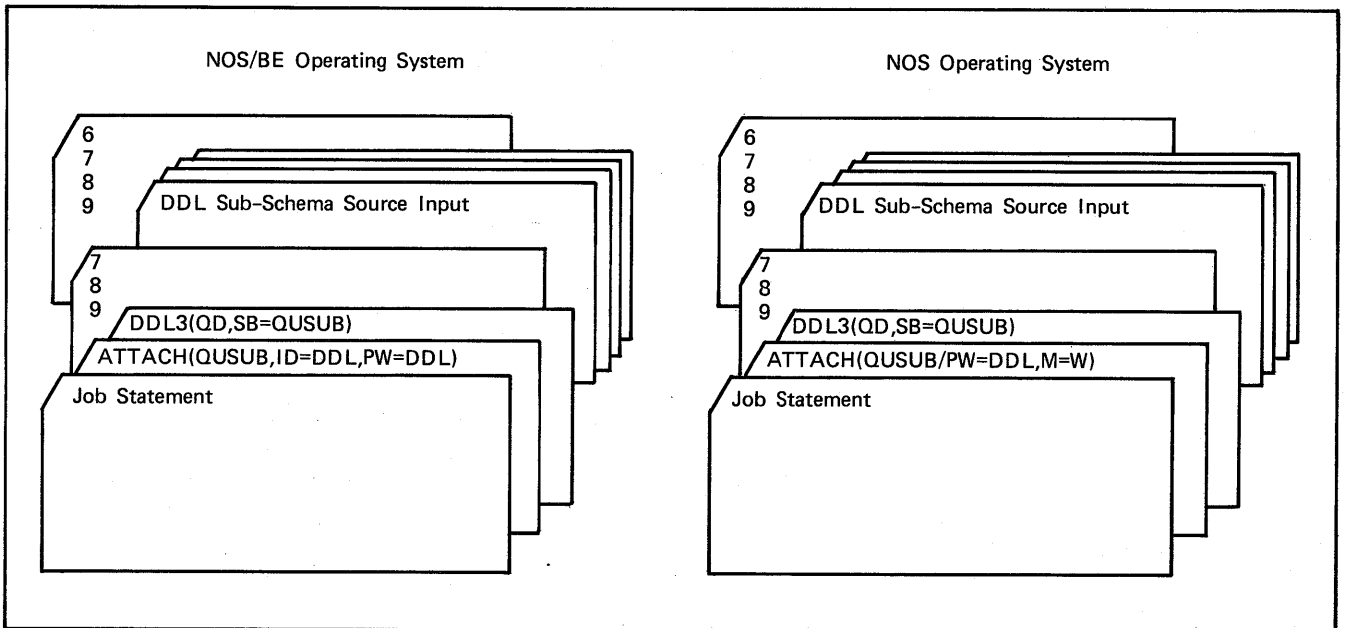


Figure 5-7. Adding a Sub-Schema to the Library

Each sub-schema stored in the library must have a unique name. If the sub-schema being added to the library has the same name as a sub-schema already stored in the library, a diagnostic is issued and the job is aborted.

REPLACING A SUB-SCHEMA

A new sub-schema can replace one that is stored in the sub-schema library. Figure 5-8 illustrates a deck structure for replacing a sub-schema in the library.

The sub-schema library file with the applicable password is attached. In addition to the QD and SB parameters in the DDL3 control statement, the replacement parameter (R) must be specified. If the sub-schema to be replaced cannot be found in the sub-schema library, an informative diagnostic is issued and the new sub-schema is added to the library.

DELETING A SUB-SCHEMA

A sub-schema stored in the sub-schema library can be deleted from the library by specifying the purge parameter (P) in the DDL3 control statement. A deck structure for deleting two sub-schemas is shown in figure 5-9.

The sub-schema library file must be attached. The QD, SB, and purge parameters are specified in the DDL3 control statement. The 7/8/9 card or its equivalent designates the end of the control statements. This is followed by statements that specify the sub-schemas to be deleted. The sub-schema name is entered anywhere from column 8 through 72. If more than one sub-schema name is entered, a space or comma must follow each sub-schema name.

COMPACTING A SUB-SCHEMA LIBRARY

Sub-schema library compaction is an optional facility that generates a new sub-schema library by copying only the active sub-schemas from the specified sub-schema library.

Figure 5-10 illustrates the deck structure for compacting the sub-schema library. The sub-schema library is attached with the appropriate password. The REQUEST/CATALOG and DEFINE control statements specify the logical file name for the new, compacted sub-schema library and assign the new library to a permanent file device. The DDL3 control statement must specify the QD parameter, the SB parameter with the name of the current sub-schema library to be compacted, and the NL parameter with the name of the new sub-schema library.

This facility is intended to be used on a sub-schema library that has had a number of sub-schemas purged or replaced and, therefore, contains wasted space. DDL eliminates the wasted space in the new sub-schema library. After the new sub-schema library is stored as a permanent file, the user should purge the old sub-schema library. This facility does not allow for compilation of sub-schemas.

COMPILATION OUTPUT

A listing of the DDL source program is provided whenever a sub-schema is compiled. Each line of the listing corresponds to one statement in the source program. The format and order of each line on the listing are identical to the format and order of the statements in the source program. Figure 5-11 is a sample source listing for a sub-schema compilation.

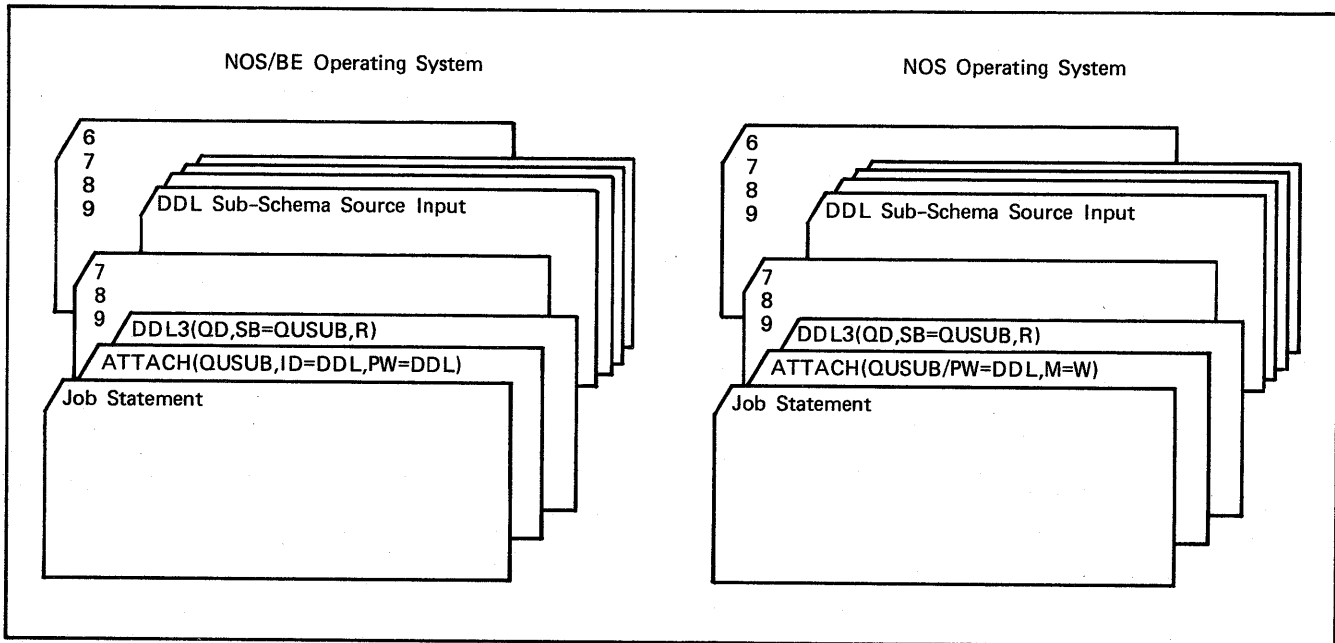


Figure 5-8. Replacing a Sub-Schema in the Library

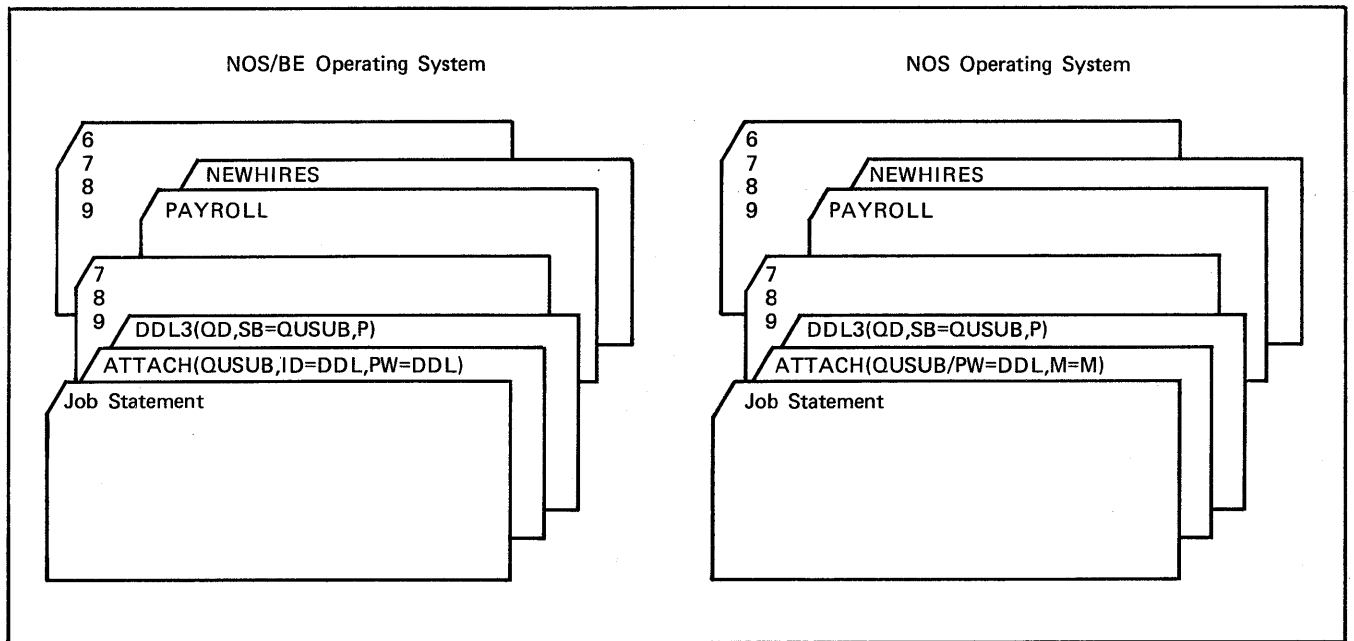


Figure 5-9. Deleting Sub-Schemas From the Library

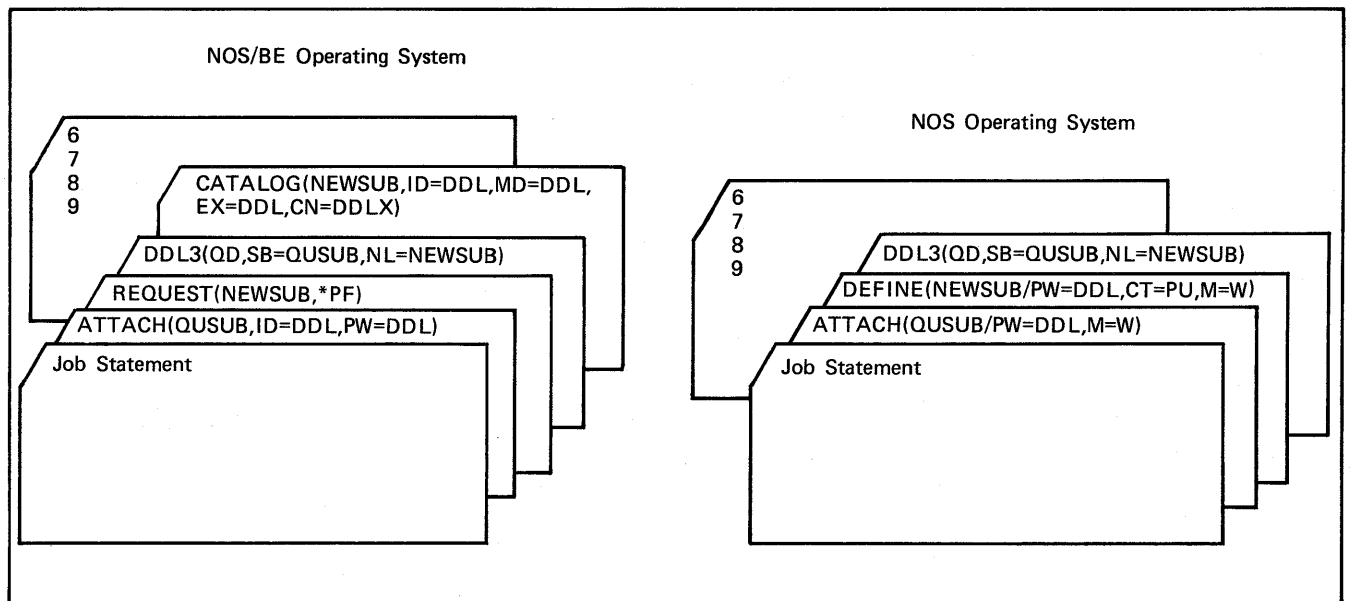


Figure 5-10. Compacting a Sub-Schema Library

The DDL compiler assigns a line number to each input statement beginning with 00001. The line numbers are printed on the source listing starting in column 16. Diagnostic messages begin in column 3 of the listing. After the last input statement is listed, a compilation summary is printed. The source listing can be suppressed by specifying L=0 in the DDL control statement; only diagnostic messages and the compilation summary are printed.

FIELD LENGTH REQUIREMENTS

Typical field length requirements for generating a Query Update sub-schema are 45gK. For more complex source input, additional core requirements are approximately ten words per data item contained in the record description entries.

```

00001 IDENTIFICATION DIVISION.
00002 SUB-SCHEMA NAME IS FINANCE
00003 DATA DIVISION.
00004 AREA-NAME IS ACCOUNTING
00005 LOG REC BEFORE,AFTER TRANS ASSIGN TO LOGFILE
00006 ORGANIZATION IS DIRECT KEY IS EMP-NO
00007 ON UPDATE CALL VALCHCK
00008 RECORD-NAME IS PAYROLL
00009 02 NAMES
00010 03 FIRST-INITIAL PIC X
00011 03 SECOND-INITIAL PIC X
00012 03 LAST-NAME PIC X(20)
00013 02 EMP-NO PIC 9(5)
00014 02 DEPT-NO PIC 9(4)
00015 02 RATE PIC 9V99
00016 02 HOURS-WORKED PIC 99
00017 02 ADDRESS
00018 03 NUMBERS PIC 9(5)
00019 03 STREET PIC X(23)
00020 03 CITY PIC A(20)
00021 03 ZIP-CODE PIC 9(5)
00022 03 STATE PIC A(2)
-----
BEGIN SUB-SCHEMA FILE MAINTENANCE -----
-----
END OF FILE MAINTENANCE -----
0 DIAGNOSTICS.
0.531 CP SECS.

```

DDL COMPLETE.
45100B CM USED.

Figure 5-11. Sub-Schema Source Listing

CREATING A DATA BASE

A data base can be generated for Query Update by using the CREATE directive as detailed in the Query Update reference manual. An alternate method involves the use of a program to read an input deck and write the data to mass storage. This method is discussed in the following paragraphs.

Data can be stored in the data base providing it is stored according to the descriptions and techniques declared in a sub-schema. After the sub-schema is compiled and stored, the data can be accessed by Query Update.

Figure 5-12 illustrates a sub-schema named FINANCE prepared for input to mass storage. Three areas are defined: ACCOUNTING, REPORTA, and VENDORD. The sub-schema is entered through the NOS/BE operating system with a sub-schema library name of DIRFILE.

Figure 5-13 illustrates three data decks containing the actual data that is to be stored according to the specifications of sub-schema FINANCE. Data for the ACCOUNTING area is entered by a COBOL program; data for the REPORTA area is entered by a FORTRAN program; and data for the VENDORD area is entered by a COMPASS program.

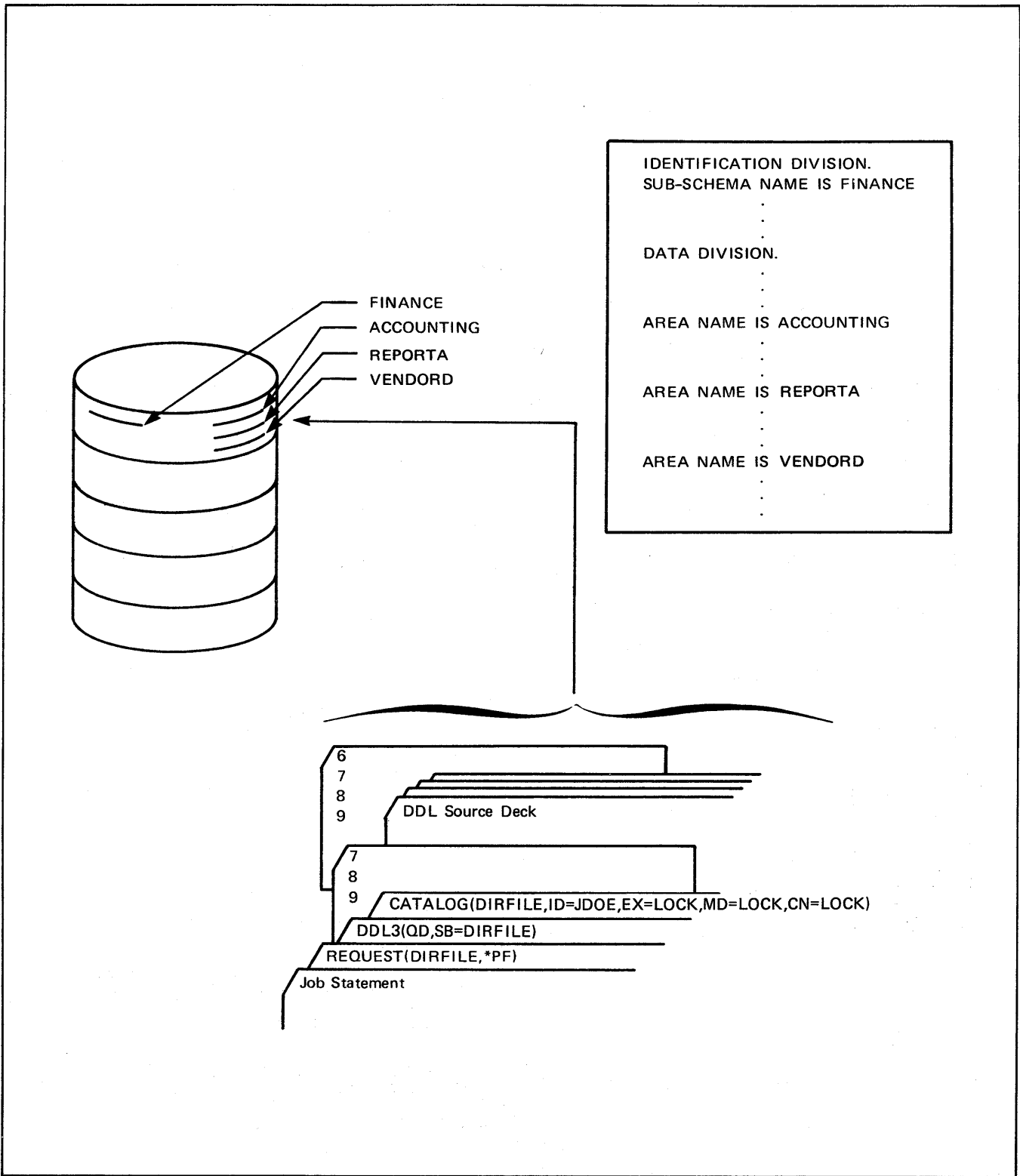


Figure 5-12. Describing the Data Base

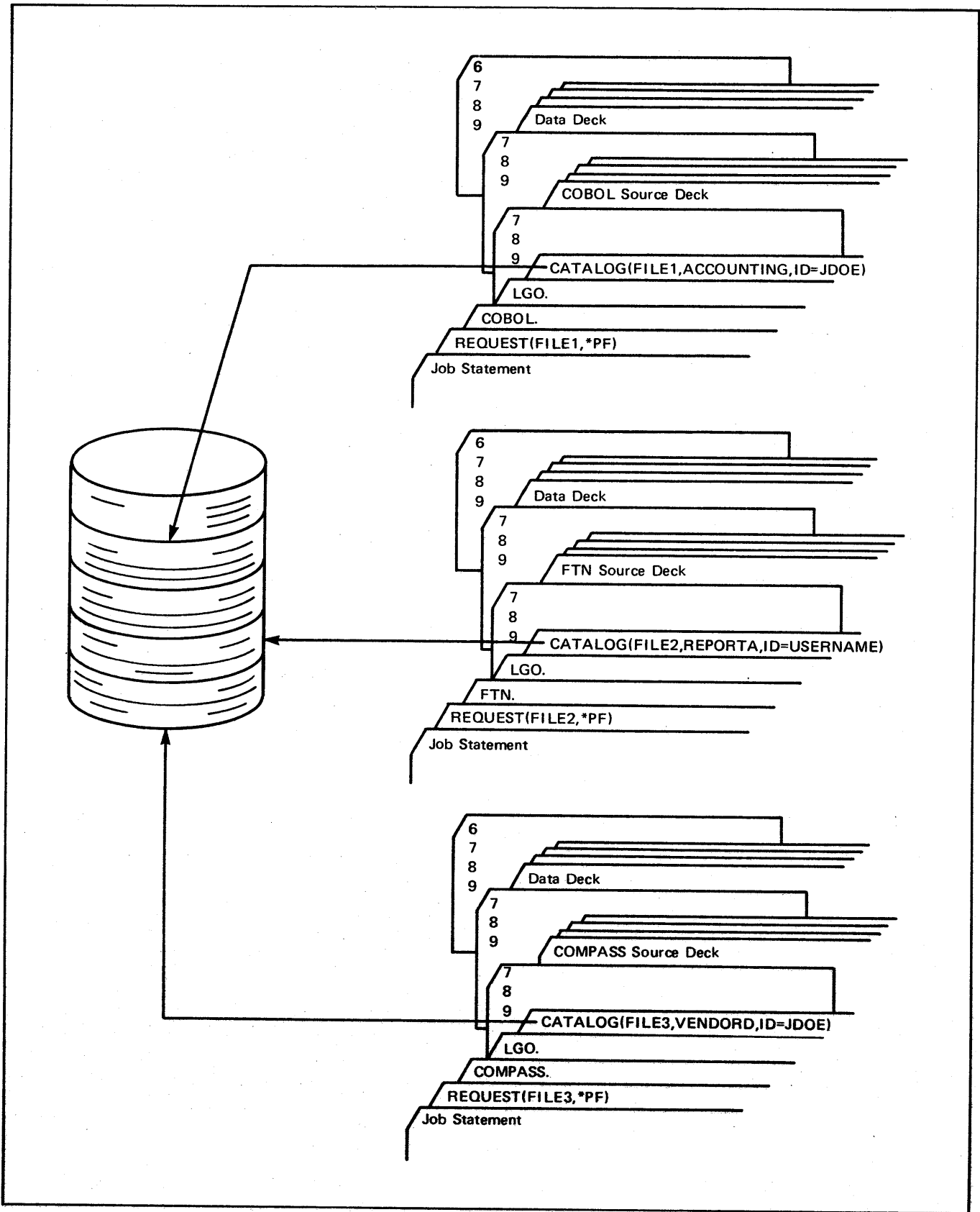


Figure 5-13. Creating the Data Base

A sample sub-schema named ACCOUNTING is shown in figure 6-1. The sub-schema library QUSUBS is created to store the sub-schema. The sub-schema contains three areas named ORDERS, LINEITEMS, and CUSTOMERS. Each area is stored under an ID of XYZ as indicated by the permanent file parameters.

A procedure library named PROCLIB is associated with the sub-schema. This library contains the data base procedure named CHECK that is called to perform special processing each time the ORDERS file is updated. PROCLIB is stored under an ID of ABC. Cycle number 3 is to be attached.

The ORDERS and CUSTOMERS areas are declared indexed sequential. Their index files, which are named ORDX and CUSTX respectively, are stored under an ID of XYZ. The ORDERS area defines ORDER-NO as a primary key and CUST-NO as an alternate key. The CUSTOMERS area defines CUST-NO as a primary key. The LINEITEMS area is declared direct access and CUST-NO is defined as a primary key.

ACCOUNTING is a relational sub-schema. The relation name is RELEXAMPLE and the join term is CUST-NO.

```

Job Statement
REQUEST(QUSUBS,*PF)
DDL3(QD,SB=QUSUBS)
CATALOG(QUSUBS,ID=XYZ,CN=PURGE,EX=ADD,MD=CHG)
7/8/9 Card

IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS ACCOUNTING
PROCEDURE LIBRARY IS PROCLIB ID IS ABC CY IS 003
DATA DIVISION.
AREA-NAME IS ORDERS ID IS XYZ
INDEX IS ORDX ID IS XYZ
ORGANIZATION IS INDEXED
KEY IS ORDER-NO KEY IS ALTERNATE CUST-NO
ON UPDATE CALL CHECK
RECORD-NAME IS ORDER-REC
    02 ORDER-NO          PIC 9(6)
    02 ORDER-DATE       PIC X(10)
    02 CUST-NO          PIC 9(4)
    02 TOTAL-VALUE      PIC Z(12).99
AREA-NAME IS LINEITEMS ID IS XYZ
ORGANIZATION IS DIRECT
KEY IS CUST-NO
RECORD-NAME IS ITEM-REC
    02 CUST-NO          PIC 9(4)
    02 ITEM-NUM         PIC X(16)
    02 QUANTITY         PIC Z(7)
    02 ITEM-PRICE       PIC Z(4).99
AREA-NAME IS CUSTOMERS ID IS XYZ
INDEX IS CUSTX ID IS XYZ
ORGANIZATION IS INDEXED
KEY IS CUST-NO
RECORD-NAME IS CUST-REC
    03 CUST-NO          PIC 9(4)
    03 BILLING-ADDR
        05 CUST-NAME     PIC X(20)
        05 CUST-STREET  PIC X(14)
        05 CUST-STATE   PIC AA
        05 CUST-ZIP     PIC 9(5)
RELATION DIVISION.
RELATION-NAME IS RELEXAMPLE
JOIN WHERE CUST-NO OF ORDER-REC EQ CUST-NO OF ITEM-REC
          CUST-NO OF ITEM-REC EQ CUST-NO OF CUST-REC
RESTRICT CUST-REC WHERE CUST-STATE EQ "CA"
    
```

6/7/8/9 Card

Figure 6-1. Creating a Sub-Schema Library and Compiling a Sub-Schema Under the NOS/BE Operating System

When the Query Update program is executing in relation mode under this sub-schema, records within the three areas are made available whenever the contents of CUST-NO compare. The RESTRICT clause imposes a restriction on the CUSTOMERS area. Only customers with billing addresses in the state of California can qualify for retrieval.

A sample sub-schema named SALES, stored in the sub-schema library QUSUB1, is shown in figure 6-2. The sub-schema contains one area named INVOICING with one record named SALES-RECORD. Sub-schema SALES is a basic example illustrating the declaration of a primary and an alternate key.

```
Job Statement
DEFINE(QUSUB1/PW=ADD,CT=PU,M=M)
DDL3(QD,SB=QUSUB1)
7/8/9 Card

IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS SALES
DATA DIVISION.
AREA-NAME IS INVOICING
BLOCK CONTAINS 10000 CHARACTERS
ORGANIZATION IS INDEXED
KEY IS INVOICE-NUM KEY IS ALTERNATE CUSTOMER-ID
    DUPLICATES ARE FIRST
FILE-LIMIT IS 10000
INDEX-LEVEL IS 4
INDEX-PADDING IS 20 PERCENT
DATA-PADDING IS 20 PERCENT
RECORD-NAME IS SALES-RECORD
    03 INVOICE-NUM      PICTURE 9(5)          USAGE IS COMP-1
    03 SALE-DATE
        05 MONTH      PICTURE 99
        05 DA         PICTURE 99
        05 YEAR       PICTURE 99
    03 CUSTOMER-ID    PICTURE X(5)
    03 CHG-ACCT-NUM  PICTURE X(10)
    03 TOTALS
        05 AMOUNT-DUE PICTURE Z(6).99      USAGE IS COMP-1
        05 AMOUNT-REC PICTURE Z(6).99      USAGE IS COMP-1
        05 SALES-TAX  PICTURE Z(6).99      USAGE IS COMP-1
    03 NUM-ITEMS     PICTURE 99            USAGE IS COMP-1
    03 ITEMX        OCCURS 15 TIMES
        05 QUANTITY   PICTURE 9(4)
        05 DESCRIPTION PICTURE X(16)
        05 UNIT-PRICE PICTURE Z(5).99
        05 EXT-PRICE  PICTURE Z(6).99

6/7/8/9 Card
```

Figure 6-2. Creating a Sub-Schema Library and Compiling A Sub-Schema Under the NOS Operating System

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

Table A-1 shows these character sets. The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in 026 or in 029 mode (regardless of the character set in use). Under NOS/BE, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job statement or any

7/8/9 card. The specified mode remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Under NOS, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of any 6/7/9 card, as described for a 7/8/9 card. In addition, 026 mode can be specified by a card with 5/7/9 multipunched in column 1, and 029 mode can be specified by a card with 5/7/9 multipunched in column 1 and a 9 punched in column 2.

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 table A-1 are applicable to BCD terminals; ASCII graphic characters are applicable to ASCII-CRT and ASCII-TTY terminals.

Several graphics are not common for all codes. Where these differences in graphics appear, assignment of collation positions and translations between codes must be made. Tables A-2 and A-3 show the CDC and ASCII character set collating sequences.

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	v	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 (55g).

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	v	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	J	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 †	nonet
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.

DIAGNOSTICS

B

All diagnostic messages that can be issued during compilation of a Query Update sub-schema source program are listed in table B-1. When a diagnostic message is printed on the source listing, it is preceded by a three-digit number enclosed in asterisks. The diagnostics are listed in order by this number. One of the following type codes is included for each diagnostic:

C Catastrophic: syntax error. Compilation is terminated.

T Trivial: syntax error. Compilation continues and a sub-schema is created.

Type Meaning

E Error: syntax error. Compilation is not necessarily terminated, but a sub-schema is not created.

The sub-schema library maintenance diagnostics listed in table B-2 are issued after compilation during an attempt to update a sub-schema library. In most cases, it is necessary to recompile the sub-schema.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS

Error Code	Type	Message	Significance	Action
035	E	NO ON CALL OPTION SPECIFIED	At least one option for the ON CALL clause must be specified when the clause is used.	Correct or delete the ON CALL clause and resubmit job.
036	E	DUPLICATE ON CALL OPTION	An option in the ON CALL clause has been specified twice.	Remove the duplicate entry in the ON CALL clause. Resubmit job.
037	T	INTEGER-1 FOR RECORD CONTAINS MUST INCLUDE KEYS	The value specified for integer-1 should be large enough to include all the keys. As a default, DDL has computed the correct value.	None.
050	E	RELATION-NAME IS MISSING	A RELATION-NAME clause is required in the Relation Division.	Place RELATION-NAME clause in Relation Division. Resubmit job.
051	E	JOIN MISSING	A JOIN clause is required if a RELATION-NAME clause is present.	Add JOIN clause to Relation Division. Resubmit job.
052	E	EQ MISSING	The relational operator EQ must appear between each pair of identifiers in JOIN clause.	Correct error and resubmit job.
053	E	INVALID RELATIONAL OPERATORS	Relational operators include: EQ, NE, GT, LT, GE, and LE.	Change illegal relational operator to a legal one. Resubmit job.
054	E	INVALID DATA NAME	An identifier references an invalid data name.	Correct error and resubmit job.
055	E	RIGHT PARENTHESIS MISSING	Subscripts must have a closing parenthesis.	Correct error and resubmit job.
056	E	INVALID RELATION NAME	The first character of a relation name is numeric or the name is the same as a reserved word.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
060	T	RELATION NAME LONGER THAN 30 CHARACTERS-NAME TRUNCATED	Relation names are limited to 30 characters; the first 30 characters of the declared name are used.	None.
061	E	RELATION NAME NOT UNIQUE	The relation name must be unique among all relation names in the sub-schema.	Make relation name unique. Resubmit job.
062	E	MAX QUALIFICATION LEVELS FOR DATA NAME REACHED	Qualification is made only to the level necessary to make the name unique.	Correct error and resubmit job.
063	E	ERROR IN SUBSCRIPT	A subscript must be a positive integer constant.	Correct error and resubmit job.
064	E	ERROR IN PARENTHESIS	A left parenthesis is missing on the subscript.	Correct error and resubmit job.
065	E	CYCLING NOT ALLOWED	An area cannot be joined to itself directly or indirectly.	Correct error and resubmit job.
066	E	KEYWORD -DIVISION- EXPECTED, NOT FOUND	The keyword DIVISION is missing or misspelled in the Relation Division.	Correct error and resubmit job.
067	E	DBI BELONGS TO AREA WITH MULTIPLE RECORDS - NOT ALLOWED IN RELATION	A JOIN clause identifier is in an area that contains more than one record description. Multiple record descriptions are not legal for areas joined in a relationship.	Correct error and resubmit job.
068	E	SOURCE DBI NOT IN SAME AREA AS PREVIOUS TARGET DBI	Each JOIN clause source identifier (except the first) must be in the same area as the previous target identifier.	Correct error and resubmit job.
069	E	SOURCE AND TARGET DBI IN SAME FILE	The source identifier in a JOIN clause cannot be in the same area as the target identifier.	Correct error and resubmit job.
070	E	SOURCE AND TARGET DBI CHARACTERISTICS MUST BE IDENTICAL	Source and target identifiers in a JOIN clause must be identical.	Correct error and resubmit job.
071	E	DBI SIZE EXCEEDS THE MAXIMUM OF 255 CHARACTERS	JOIN clause identifiers are limited to 255 characters.	Correct error and resubmit job.
072	E	-ANY- ONLY LEGAL ON TARGET ITEM IN JOIN	The ANY option can be used as a subscript for a target identifier only in a JOIN clause.	Correct error and resubmit job.
073	E	TARGET DBI WITH SUBSCRIPT ANY MUST BE REPEATING ALTERNATE KEY	The ANY option used as a subscript for a target identifier in a JOIN clause is restricted to repeating alternate key fields.	Correct error and resubmit job.
074	E	DATA NAME UNDEFINED	The identifier in a JOIN clause references an undefined data name.	Correct error and resubmit job.
075	E	MULTIPLY DEFINED/IMPROPERLY QUALIFIED DATA NAME	A data name is not unique.	Qualify data name with a higher level and resubmit job.
076	E	SUBSCRIPT MUST APPEAR ON SUBSCRIPTABLE ITEM	Only items specified with an OCCURS clause can have a subscript.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
077	E	INVALID RESTRICT CLAUSE. SCANNING RESUMES AT NEXT RELATION ENTRY	The construction of the RESTRICT clause does not conform to the RESTRICT clause rules.	Correct the RESTRICT clause and resubmit job.
078	E	SUBSCRIPT VALUE TOO LARGE	The value of the subscript is larger than the value specified in the OCCURS clause applying to the data description.	Correct error and resubmit job.
080	E	TOO MANY SUBSCRIPTS	A maximum of three subscripts can be specified.	Correct error and resubmit job.
081	E	SUBSCRIPTABLE QUALIFIER OMITTED	A higher level of qualification is needed for the subscripted data name.	Qualify data name with higher level and resubmit job.
082	E	DATA ITEM NOT IN RECORD OF RESTRICT	A data item referenced in a RESTRICT clause is not a defined item in the record.	Correct error and resubmit job.
083	E	LITERAL IS OF WRONG DATA TYPE	The literal specified in a RESTRICT clause is not compatible with the data type of the referenced data item.	Correct error and resubmit job.
084	E	RECORD NOT IN RELATION OR OUT OF ORDER	The record named in the RESTRICT clause must be within an area named in the Relation, and the order of the RESTRICT clauses must agree with the order of the Areas in the Relation definition.	Correct error and resubmit job.
085	E	MORE THAN ONE RESTRICT CLAUSE FOR THIS RECORD	A given record can appear in only one RESTRICT clause for a particular relation.	Correct error and resubmit job.
086	E	INVALID BLOCK TYPE	The block type in the BLOCK TYPE clause is unrecognizable.	Correct error and resubmit job.
087	E	INVALID RECORD TYPE	The record type in the RECORD TYPE clause is unrecognizable.	Correct error and resubmit job.
100	C	IDENTIFICATION HEADER NOT FIRST CARD OF PROGRAM	The Identification Division is misspelled or missing.	Correct error and resubmit job.
101	E	ERROR IN SUB-SCHEMA CLAUSE	The source statement defining the sub-schema name is missing or reserved word SUB-SCHEMA is misspelled or the sub-schema name is a reserved word.	Correct error and resubmit job.
102	E	DATA-NAME NOT UNIQUE	This data name was already used in the current hierarchy.	Make data name unique and resubmit job.
103	C	DATA DIVISION HEADER NOT SPECIFIED	The reserved words in the Data Division declaration are misspelled or the declaration is missing.	Correct error and resubmit job.
104	E	THE FOLLOWING WORD IS NOT VALID IN THE INDICATED STATEMENT LINE	An item name is the same as a reserved word or improper coding conventions have been used.	Correct error and resubmit job.
105	E	AREA NAME NOT SPECIFIED	The expected area name was not found.	Specify area name and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
106	E	FILE-LIMIT CLAUSE ILLEGAL WITH SEQUENTIAL FILES	FILE-LIMIT does not apply to sequential files.	Delete FILE-LIMIT clause and resubmit job.
107	E	INVALID FILE ORGANIZATION	File organization reserved words are either missing or misspelled.	Correct error and resubmit job.
108	E	INTEGER FOR FILE LIMIT NOT VALID	The integer value specified in the FILE-LIMIT clause is not a valid numeric literal.	Correct error and resubmit job.
109	T	COMPUTED RECORD SIZE IS GREATER OR LESS THAN THE RECORD CONTAINS SIZE, COMPUTED SIZE IS USED	The record size computed from the item definitions is larger than the value specified in the RECORD CONTAINS clause. The computed size replaces the specified size.	None.
110	E	INTEGER FOR INDEX LEVEL NOT VALID	The integer value specified in the INDEX-LEVEL clause is not a valid numeric literal.	Correct error and resubmit job.
111	E	INTEGER FOR INDEX BLOCK NOT VALID	The integer value specified in the INDEX-BLOCK clause is not a valid numeric literal.	Correct error and resubmit job.
112	E	INTEGER FOR INDEX PADDING NOT VALID	The integer value specified in the INDEX-PADDING clause is not a valid numeric literal.	Correct error and resubmit job.
113	E	INTEGER FOR DATA PADDING NOT VALID	The integer value specified in the DATA-PADDING clause is not a valid numeric literal.	Correct error and resubmit job.
114	E	INTEGER FOR NUMBER OF BLOCKS NOT VALID	The integer value specified in the NUMBER OF BLOCKS clause is not a valid numeric literal.	Correct error and resubmit job.
115	E	DATA-PADDING CLAUSE LEGAL FOR IS/AK FILES ONLY	DATA-PADDING applies to indexed sequential and actual key files only.	Correct error and resubmit job.
116	E	IMPLEMENTOR NAME FOR LOG FILE INVALID	The area name specified in the LOG clause does not conform to the naming conventions.	Correct error and resubmit job.
117	E	INTEGER FOR BLOCK CONTAINS NOT VALID	Integer-1 and/or integer-2 value specified in the BLOCK CONTAINS clause is not a valid numeric literal.	Correct error and resubmit job.
118	E	INTEGER FOR RECORD CONTAINS NOT VALID	Integer-1 and/or integer-2 value specified in the RECORD CONTAINS clause is not a valid numeric literal.	Correct error and resubmit job.
119	T	COMPUTED RECORD SIZE GREATER THAN BLOCK SIZE, COMPUTED SIZE ASSUMED	The record size computed from the item definitions is larger than the value specified in the BLOCK CONTAINS clause. The computed size replaces the specified size.	None.
120	E	INDEX-PADDING CLAUSE LEGAL FOR IS FILES ONLY	INDEX-PADDING applies to indexed sequential files only.	Correct error and resubmit job.
121	E	INDEX-LEVEL CLAUSE LEGAL FOR IS FILES ONLY	INDEX-LEVEL applies to indexed sequential files only.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
122	E	RECORD NAME INVALID	The expected record name does not conform to the naming conventions.	Correct error and resubmit job.
123	E	SDA PROC NAME GREATER THAN 7 CHARACTERS	The hashing routine procedure-name specified in the ORGANIZATION clause can only be from 1 through 7 characters in length. The procedure name is not stored in the sub-schema.	Correct error and resubmit job.
124	E	INDEX-BLOCK CLAUSE LEGAL FOR IS FILES ONLY	INDEX-BLOCK applies to indexed sequential files only.	Correct error and resubmit job.
125	E	ILLEGAL CHARACTER IN PERM FILE NAME	Permanent file names must be 1 through 30 alphanumeric characters; the first character must be alphabetic.	Correct error and resubmit job.
126	E	THE NUMBER OF ON CALL STATEMENTS EXCEEDS 8	Only one through eight conditions are allowed in the ON CALL clause.	Correct error and resubmit job.
127	E	INVALID KEY NAME	The expected key name does not conform to the naming conventions.	Correct error and resubmit job.
128	E	ON CALL PROCEDURE NAME GREATER THAN 7 CHARACTERS	The procedure name specified in the ON CALL clause can only be from 1 through 7 characters in length.	Correct error and resubmit job.
129	E	CALL NOT SPECIFIED	The reserved word CALL is either missing or misspelled.	Correct error and resubmit job.
130	E	ON CALL PROCEDURE NAME NOT SPECIFIED	The expected procedure name in the ON CALL clause is either missing or is the same as a reserved word.	Correct procedure name and resubmit job.
131	E	NUMBER OF BLOCKS CLAUSE LEGAL FOR DA FILES ONLY	NUMBER OF BLOCKS applies to direct access files only.	Correct error and resubmit job.
132	E	LEVEL NUMBER NOT SPECIFIED	The expected level number is not a valid numeric literal.	Correct error and resubmit job.
133	E	INVALID LEVEL NUMBER	The valid range of level numbers is from 2 through 49.	Correct error and resubmit job.
134	E	DATA NAME OR FILLER NOT SPECIFIED OR RESERVED WORD USED	The expected item name either does not conform to the naming conventions or is the same as a reserved word.	Correct item name and resubmit job.
135	E	DEPENDING ON ITEM PIC SIZE EXCEEDS 6	The DEPENDING ON item specified in the OCCURS clause must have a picture size of 1 through 6 characters.	Correct error and resubmit job.
136	E	INTEGER FOR OCCURS NOT SPECIFIED	The integer-1 value specified in the OCCURS clause is not a valid numeric literal.	Correct error and resubmit job.
137	E	DEPENDING ON NAME NOT SPECIFIED	The DEPENDING ON name specified in the OCCURS clause does not conform to the naming conventions.	Correct name and resubmit job.
138	E	NUMBER OF BLOCKS VALUE NOT BETWEEN 1 AND 1677215	NUMBER OF BLOCKS value not between 1 and 1677215.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
139	E	DEPENDING ON ITEM HAS WRONG USAGE	DEPENDING ON ITEM usage must be DISPLAY, COMPUTATIONAL or COMP-1.	Correct error and resubmit job.
140	C	DATA NAME TABLE OVERFLOW - INCREASE YOUR FIELD LENGTH	Not enough field length was specified to complete the compilation.	Increase field length and resubmit job.
141	E	BAD LEVEL NUMBER - ITEM IGNORED	A sign was encountered in the level number.	Delete sign and resubmit job.
142	E	NO ASSIGN CLAUSE SPECIFIED FOR LOG FILE	ASSIGN sub-clause missing in the LOG clause.	Correct error and resubmit job.
143	E	WORD OCCURS NOT FOLLOWED BY AN INTEGER	A sign was encountered in the integer-1 or integer-2 value in the OCCURS clause.	Delete sign and resubmit job.
144	E	PICTURE TYPE DOES NOT CORRESPOND TO USAGE TYPE	The item requires a numeric picture specification.	Correct error and resubmit job.
145	E	OCCURS INTEGER GREATER THAN 262142	The maximum value that is allowed in the OCCURS clause is 262142.	Correct value and resubmit job.
146	T	RECORD SIZE NOT EQUAL THROUGHOUT THE AREA, LARGEST ONE IS USED FOR MRL	All records throughout the area should be defined as being the same length.	None.
147	E	LITERAL GREATER THAN 256	The length of literals is from 1 through 256 characters.	Shorten literal and resubmit job.
148	E	PF R VALUE INVALID	The R value, specifying the device on which a permanent file resides or is to reside, is invalid.	Correct error and resubmit job.
149	E	NO PICTURE SPECIFICATION FOR ELEMENTARY ITEM	An elementary item must be described with a picture specification.	Correct error and resubmit job.
150	T	INDEX PADDING PERCENT GREATER THAN 2 CHARACTERS, 5 ASSUMED	The allowable range of the integer value in the INDEX-PADDING clause is 1 through 99. The value is now defaulted to 5.	None.
151	E	RECORD SIZE GREATER THAN 6	The maximum size of the DEPENDING ON name specified in the RECORD CONTAINS clause is 6 characters.	Shorten size and resubmit job.
152	T	VALUE OF RECORD SIZE GREATER THAN 262142, 512 ASSUMED	The value of integer-1 and/or integer-2 in the RECORD CONTAINS clause exceeds the maximum value of 262142. It is defaulted to 512.	None.
153	E	INDEX LEVEL SPECIFIED GREATER THAN 2 CHARACTERS	The allowable range of the integer value in the INDEX-LEVEL clause is 1 through 63.	Correct level number and resubmit job.
154	T	INDEX LEVEL VALUE NOT BETWEEN 1 AND 64, 1 ASSUMED	The allowable range of the integer value in the INDEX-LEVEL clause is 1 through 63. The value is now defaulted to 1.	None.
155	E	INDEX BLOCK SPECIFIED GREATER THAN 5 CHARACTERS	The allowable range of the integer value in the INDEX-BLOCK clause is 1 through 32767.	Correct value and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
156	T	INDEX BLOCK VALUE NOT BETWEEN 1 AND 32767, 512 ASSUMED	The allowable range of the integer value in the INDEX-BLOCK clause is 1 through 32767. The value is now defaulted to 512.	None.
157	T	DATA PADDING PERCENT GREATER THAN 2 CHARACTERS, 5 ASSUMED	The allowable range of the integer value in the INDEX-PADDING clause is 1 through 99. The value is now defaulted to 5.	None.
158	E	ILLEGAL CHARACTER IN PICTURE	Invalid character specified in the picture string.	Correct string and resubmit job.
159	E	REPEAT COUNT IN PARENTHESES EXCEEDS MAXIMUM OF 2047	The allowable range for the repeat count is 1 through 2047.	Correct count and resubmit job.
160	E	SYNTACTICAL ERROR IN PICTURE	Character placement in the picture string is invalid.	Look for parenthesis placement.
161	E	EXCESS REPEAT COUNT IN EDITED PICTURE	The total allowable values for repeat counts in a picture string is 2047. The maximum number of characters defining the picture string is 30.	Correct error and resubmit job.
162	E	MAXIMUM OF 63 INSERTION CHARACTERS ALLOWED IN PICTURE	The allowable number of insertion characters specified in a picture string is 1 through 63.	Correct error and resubmit job.
163	E	NON NUMERIC WITHIN PARENTHESES	Only numeric values are allowed within parentheses.	Correct error and resubmit job.
164	E	NUMERIC ITEM GREATER THAN 18	The picture for a numeric item must not specify more than 18 digits, of which 14 are significant.	Correct error and resubmit job.
165	E	DEPENDING ON NAME IN RECORD CONTAINS CLAUSE IS INVALID	The name in the RECORD CONTAINS... DEPENDING ON clause must be 1 through 30 alphanumeric characters and embedded hyphens with at least one alphabetic character.	Correct name and resubmit job.
166	E	INTEGER IN RECORD BLOCK CONTAINS CLAUSE IS INVALID	The integer-2 entry in the BLOCK CONTAINS clause must be a positive number. If characters are specified, the maximum is 327600.	Correct error and resubmit job.
167	T	INTEGER VALUE IN RECORD BLOCK CONTAINS IS GREATER THAN 262142, 5120 ASSUMED	The allowable value of integer-1 and/or integer-2 in the BLOCK CONTAINS clause is 1 through 262142. The value is now defaulted to 5120.	None.
168	E	USAGE CLAUSE IS INCOMPLETE	The USAGE clause must specify the internal representation of a data item with one of 7 available options.	Specify internal representation and resubmit job.
169	C	EMPTY INPUT FILE	No information is in the input file.	Make corrections to the input file and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
170	E	DUPLICATES INVALID FOR ACTUAL OR DIRECT ORGANIZATION OR INDEXED WHERE ALTERNATE KEY EXISTS	Duplicates cannot be specified for an actual key or direct access file. For an indexed sequential file, duplicate keys cannot be specified for the primary key if alternate keys are specified.	Correct error and resubmit job.
171	E	UNRECOGNIZABLE/MISPLACED SOURCE STATEMENT	Reserved word is misspelled or required source statement is missing.	Correct error and resubmit job.
173	E	PICTURE ON GROUP ITEM IS INVALID	It is invalid to specify a picture string on a group item.	Correct error and resubmit job.
174	E	OCCURS CANNOT BE SPECIFIED FOR FILLER ITEMS	FILLER cannot be an occurring item.	Correct error and resubmit job.
175	C	HASH OVERFLOW TABLE SIZE EXCEEDED	Not enough core was specified to compile sub-schema.	Increase field length and resubmit job.
176	E	INDICATED POINT LOCATION GREATER THAN 31	The maximum length of a numeric item is 30 digits.	Correct error and resubmit job.
177	E	DEPENDING ON ITEM NOT IN FIXED PART OF RECORD	DEPENDING ON name in the item entry was not defined before it was referenced.	Correct error and resubmit job.
178	E	KEY NAME OR DEPENDING ON NAME UNDEFINED	Key name in the area description entry or DEPENDING ON name specified in the item entry could not be found.	Check spelling.
179	E	RECORD CONTAINS DATA NAME NOT WITHIN RECORD	DEPENDING ON name specified in the RECORD CONTAINS clause could not be found.	Check spelling.
180	E	KEY NAME NOT WITHIN SPECIFIED AREA	Key name specified in the area description entry could not be found in the subordinate records.	Check spelling.
181	E	PF ID/UN NAME INVALID	The user identification following the ID or UN keyword does not conform to Query Update naming conventions.	Correct identification and resubmit job.
182	E	PF PW NAME INVALID	Maximum length of a permanent file password is 20 characters.	Correct error and resubmit job.
183	E	PF CY NUMBER INVALID	The number following the CY keyword must be integer literal.	Correct error and resubmit job.
184	E	INVALID SEQUENCE OPTION	The SEQUENCE clause must specify FORTRAN, COBOL, or a character literal for the collating sequence.	Correct the SEQUENCE clause and resubmit job.
185	E	RECORD CONTAINS AND RECORD TYPE DO NOT AGREE	Character-count must be declared if RECORD CONTAINS DEPENDING ON and RECORD TYPE are specified.	Correct error and resubmit job.
186	E	BLOCK CONTAINS AND BLOCK TYPE DO NOT AGREE	Entries in the BLOCK CONTAINS and BLOCK TYPE clauses are conflicting.	Resolve conflict and resubmit.
187	E	INCOMPATIBLE BLOCK AND RECORD TYPE	If record type is CONTROL-WORD, block type must be INTERNAL.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
189	E	BOTH REC CONTAINS DEP ON AND OCCURS DEP ON CAN'T BE SPECIFIED	It is illegal to define a variable length item within a record that has a DEPENDING ON specified.	Correct error and resubmit job.
190	E	RECORD-TYPE MUST BE TRAILER-COUNT	When OCCURS DEPENDING ON is specified, record type must be TRAILER-COUNT.	Correct error and resubmit job.
192	E	DEPENDING ON ITEM NOT EQUALLY DEFINED THROUGHOUT THE AREA	When more than one record is specified for an area, there must be an item in each record whose relative position must correspond to the DEPENDING ON item.	Correct error and resubmit job.
193	E	DEPENDING ON ITEM POSITION DIFFERS WITHIN THE AREA	If an area contains more than one record description and one record description contains an item that occurs a number of times depending on a data-name, each record description in the area must contain an occurring item depending on a data-name. In each record, this data-name must be in the same location.	Correct error and resubmit job.
194	E	TRAILER LENGTH OF A T TYPE RECORD IS NOT EQUALLY DEFINED THROUGHOUT THE AREA	If more than one record is specified for an area, the trailer portion size of the T type record must be defined the same in all of the records.	Correct error and resubmit job.
195	E	HEADER LENGTH OF A T TYPE RECORD IS NOT EQUALLY DEFINED THROUGHOUT THE AREA	If more than one record is specified for an area, the header portion size of the T type record must be defined the same in all of the records.	Correct error and resubmit job.
196	E	KEY SIZE IS NOT EQUALLY DEFINED WITHIN THE AREA	When more than one record is specified in an area, the size of the primary key must be the same in all of the records.	Correct error and resubmit job.
197	E	RELATIVE KEY WORD POSITION IS NOT EQUALLY DEFINED WITHIN THE AREA	When more than one record is specified in an area, the word position of the primary key must be defined the same in all of the records.	Correct error and resubmit job.
198	E	RELATIVE KEY POSITION IN RKW IS NOT EQUALLY DEFINED WITHIN THE AREA	When more than one record is specified in an area, the relative position of the primary key must be defined the same in all of the records.	Correct error and resubmit job.
199	T	BLOCK TYPE TAKES PRECEDENCE OVER BLOCK-CONTAINS	When both BLOCK-TYPE and BLOCK-CONTAINS are specified, BLOCK-TYPE takes precedence.	None.
200	C	SEQUENTIAL FILE MAY NOT HAVE ALTERNATE KEY	Alternate keys cannot be specified for sequential files.	Correct error and resubmit job.
201	C	FOR ACTUAL KEY FILE, KEY MUST BE INTEGER	For an actual key file, the primary key must be a data item described with a USAGE IS INTEGER clause.	Correct error and resubmit job.

TABLE B-1. SUB-SCHEMA COMPILATION DIAGNOSTICS (Contd)

Error Code	Type	Message	Significance	Action
202	E	PF M NAME INVALID	Legal M parameter values include: R(READ), W(WRITE), M(MODIFY), and RM(READMD).	Correct error and resubmit job.
203	E	PF SN OR PN NAME INVALID	An invalid set name (SN) for NOS/BE or pack name (PN) for NOS has been specified as a permanent file parameter.	Correct error and resubmit job.
204	E	INDEX FILE NAME INVALID	An index file name is limited to 1 through 7 characters (letters or digits) and must begin with a letter.	Correct error and resubmit job.
205	E	PF PARAMETER INVALID	Valid permanent file parameters are ID, PW, CY, and SN for NOS/BE, and UN, PW, M, and PN for NOS.	Correct error and resubmit job.
206	E	LIBRARY NAME INVALID	A procedure library name is limited to 1 through 7 characters (letters or digits), must begin with a letter, and must be unique among all file names.	Correct error and resubmit job.
207	E	DUPLICATE LFN	Logical file names must be unique within a job.	Correct error and resubmit job.
208	C	ONLY 64 AREAS ALLOWED IN A SUBSCHEMA	A single sub-schema is limited to 64 areas.	Correct error and resubmit job.
209	E	INDEX FILE PF PARAMETER INVALID WITH AREA TEMPORARY	An index file associated with a TEMPORARY area is a local file for temporary use only.	Delete permanent file parameters.
210	E	LIBRARY NOT SPECIFIED	The word LIBRARY in the PROCEDURE LIBRARY clause is missing.	Correct error and resubmit job.
211	E	PRIMARY KEY PREVIOUSLY DEFINED	Only one primary key can be defined for a file.	Correct error and resubmit job.
212	E	NO PRIMARY KEY SPECIFIED FOR AAM OR SORTED SEQ FILE	An advanced access method file (indexed sequential, direct, actual key) and a sorted sequential file must have a primary key.	Define primary key and resubmit job.
213	E	NO KEY ALLOWED FOR THIS FILE	A key cannot be specified for a nonsorted sequential file.	Correct error and resubmit job.
214	E	OPEN AND CLOSE PROCEDURES MUST BE SPECIFIED WITH SEARCH	When an ON SEARCH clause is used, an ON OPEN and ON CLOSE clause must be specified also.	Correct error and resubmit job.
215	E	MORE THAN ONE RECORD NAMED KEY-FIELD FOR SAME AREA	Any one area can have only one record named KEY-FIELD.	Correct error and resubmit job.
216	E	RECORD NAMED KEY-FIELD MUST BE LAST RECORD IN AREA	KEY-FIELD record must be the last record declared for any one area.	Correct error and resubmit job.
217	E	RECORD NAMED KEY-FIELD INVALID WITH SEQUENTIAL FILES	Record named KEY-FIELD cannot be declared for sequential files.	Correct error and resubmit job.
218	E	MORE THAN ONE ITEM/GROUP SPECIFIED FOR RECORD NAMED KEY-FIELD	Only one item can be declared in a KEY-FIELD record declaration.	Correct error and resubmit job.

TABLE B-2. SUB-SCHEMA LIBRARY MAINTENANCE DIAGNOSTICS

Message	Significance	Action
DID NOT LOCATE aaaaa... - PURGE NOT POSSIBLE	The sub-schema name specified for the purge could not be located in the sub-schema library.	Specify the correct sub-schema name and recompile.
DID NOT LOCATE SUB-SCHEMA TO BE REPLACED --- NEW SUB-SCHEMA HAS BEEN ADDED	The sub-schema to be replaced in the library could not be located; the current sub-schema has been added to the library.	Specify the correct library or sub-schema name and recompile.
EMPTY INPUT FILE --- PURGE NOT POSSIBLE	The names of the sub-schemas to be purged must be in the input file.	Make appropriate corrections to the input file and recompile.
EMPTY SUB-SCHEMA FILE, DDL ABORTED	While executing an audit, collect, or purge of a sub-schema library, DDL found it empty and terminated execution.	Specify the correct library and recompile.
ILL-FORMATTED LIBRARY -- NOT UPDATABLE, DDL ABORTED	The sub-schema library contains an error and cannot be updated. DDL terminates.	Check for empty library or file that is not a library. Correct the error and recompile.
OLD SUB-SCHEMA FILE BAD, SUB-SCHEMA LENGTH IS ZERO. DDL ABORTED	During execution of the facility for compacting the sub-schema library, DDL found that a sub-schema in the old library had a length of zero, which indicates an error in the file.	Specify the correct sub-schema library and recompile.
SUB-SCHEMA WITH THE SAME NAME AS THE NEW SUB-SCHEMA ALREADY EXISTS --- FILE NOT UPDATED	The current sub-schema could not be added to the library since the library contains a sub-schema with the same name.	Change the sub-schema name and recompile.
WARNING -- EMPTY SUB-SCHEMA FILE	The sub-schema library contains no sub-schemas. The sub-schemas may have been purged prior to this compilation. The new sub-schema is added to the library and is the only sub-schema in the library.	If sub-schemas have been purged, create a new library.

0

0

0

0

0

0

0

GLOSSARY

C

Actual Key -

A file organization in which records are stored according to their key values.

Advanced Access Methods (AAM) -

A file manager that processes initial and extended files of the following organizations: indexed sequential, direct access, and actual key. AAM supports the Multiple-Index Processor. See CYBER Record Manager.

Area -

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

Basic Access Methods (BAM) -

A file manager that processes sequential and word addressable file organizations. See CYBER Record Manager.

Beginning-of-Information (BOI) -

CYBER Record Manager defines beginning-of-information as the start of the first user record in a file. System-supplied information, such as an index block or control word, does not affect beginning-of-information. Any label on a tape exists prior to beginning-of-information.

Blocks -

The term block has several meanings depending on context. On tape, a block is information between interrecord gaps on tape. CYBER Record Manager defines several blocks depending on organization, as shown in table C-1.

TABLE C-1. BLOCK TYPES

Organization	Blocks
Indexed sequential	Data block; index block
Direct access	Home block; overflow block
Actual key	Data block
Sequential	Block type I, C, K, E

Clause -

A combination of DDL elements that formulate a description of the data base.

Compression -

The process of condensing a record to reduce the amount of storage space required. The user can supply a compression routine or use a system-supplied routine. Contrast with Decompression.

Control Word -

A system-supplied word that precedes each W type record in storage.

CYBER Database Control System (CDCS) -

The controlling module of DMS-170.

CYBER Record Manager (CRM) -

A generic term relating to the common products AAM and BAM, which run under the NOS and NOS/BE operating systems and allow a variety of record types, blocking types, and file organizations to be created and accessed. The execution time input/output of COBOL, FORTRAN, Sort/Merge 4, ALGOL, and the DMS-170 products is implemented through CYBER Record Manager. Neither the input/output of the NOS and NOS/BE operating systems themselves nor any of the system utilities such as COPY or SKIPF is implemented through CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

Data Administrator -

A person who defines the format and organization of the data base.

Data Base -

A systematically organized, central pool of information; organization is described by a sub-schema for Query Update.

Data Base Procedure -

A special-purpose routine that performs a predefined operation; specified in the sub-schema and initiated by Query Update.

Data Description Entry -

A source program statement that describes the attributes of a data item.

Data Description Language (DDL) -

The language used to structure the sub-schema.

Data Item -

A unit of data within a record; can be an elementary or group data item.

Data Manipulation Language (DML) -

A component of the FORTRAN Data Base Facility. The language consists of statements that are specified within a FORTRAN program to provide access to a data base.

Data Name -

A name identifying a group or elementary data item in the data base; can contain up to 30 letters, digits, or embedded hyphens, and must begin with a letter.

Decompression -

The process of expanding a compressed record to restore it to its original size. The user can supply a decompression routine or use a system-supplied routine. Contrast with Compression.

Direct Access -

A file organization in which records are stored randomly in fixed length blocks. The number of the block to receive a record is determined by a hashing operation on the record key.

Direct Access File -

In the context of CYBER Record Manager, a direct access file is one of the five file organizations. It is characterized by the system hashing of the unique key within each file record to distribute records randomly in blocks called home blocks of the file.

In the context of NOS permanent files, a direct access file is a file that is accessed and modified directly, as contrasted with an indirect access permanent file.

Directed Relationship -

The logical relational structure that defines a specific order in which the files in a relation are traversed and the order in which the record occurrences are retrieved. The relational structure is formed by the join terms.

Directive -

A Query Update command that describes an operation to be performed; consists of the reserved words of the Query Update language in combination with user-supplied elements.

Directory -

A file that contains area and record attributes of the data base; created when the sub-schema is compiled; an object sub-schema.

Elementary Item -

A data item that is not subdivided into other data items; an elementary item that is part of a group item has the highest level number in the group item.

End-of-Information (EOI) -

CYBER Record Manager defines end-of-information in terms of the file organization and file residence, as shown in table C-2.

Entry -

One or more clauses that form a source program statement.

File -

A collection of records treated as a unit; an area in the sub-schema.

Fixed Occurrence Data Item -

A data item that is repeated the same number of times in all records; uses format 1 of the OCCURS clause.

Group Item -

A data item that is subdivided into other items; a collection of data items.

Hashing -

The transformation of record keys to produce relative data block addresses for records in a direct access file.

Home Block -

Mass storage allocated for a file with direct access organization at the time the file is created.

Identifier -

A data-name that is referenced uniquely through a combination of subscripts and qualifiers.

TABLE C-2. END-OF-INFORMATION BOUNDARIES

File Organization	File Residence	Physical Position
Sequential	Mass storage	After the last user record.
	Labeled tape in SI, I, S, or L format	After the last user record and before any file trailer labels.
	Unlabeled tape in SI or I format	After the last user record and before any file trailer labels.
Word Addressable	Unlabeled tape in S or L format	Undefined.
	Mass storage	After the last word allocated to the file, which might be beyond the last user record.
Indexed Sequential, Actual Key	Mass storage	After the record with the highest key value.
Direct Access	Mass storage	After the last record in the most recently created overflow block or home block with the highest relative address.

Indexed Sequential -

A file organization in which records are stored in ascending order by key.

Interrelated Files -

Those data base files that are connected logically through a relationship defined in the sub-schema using join terms.

Join Terms -

The common identifiers that are used to join two files in a relationship.

Joining Files -

The logical linkage of one file to another in a relation through the use of data items called join terms or identifiers.

Key -

One or more data items, the contents of which identify the type or location of a record, or the sequence of data.

Keyword -

A reserved word that is required in a DDL source program clause.

Level -

For system-logical-records, an octal number 0 through 17 in the system-supplied 48-bit marker that terminates a short or zero-length PRU.

Level Number -

A number defining the structure of data within a record.

Literal -

A constant completely defined by its own identity.

Logging -

The recording of activity that occurs in response to all Query Update directives following an INVOKE (or USE) directive for the area.

Logical Record -

Under NOS, a data grouping that consists of one or more PRUs terminated by a short PRU or zero-length PRU. Equivalent to a system-logical-record under NOS/BE.

Major Alternate Key -

The leading portion of an alternate key field.

Major Primary Key -

The leading portion of a primary key field.

Nested Group Item -

A group item that is subordinate to another group item.

Noise Record -

Number of characters the tape drivers discard as being extraneous noise rather than a valid record. Value depends on installation settings.

Nonrepeating Group Item -

A sub-schema data item that contains subordinate data items; this group item occurs only once in each record; used to identify a series of related data items.

Overflow Block -

Mass storage the system adds to a file with direct access organization when records cannot be accommodated in the home block.

Partition -

CYBER Record Manager defines a partition as a division within a file with sequential organization. Generally, a partition contains several records or sections. Implementation of a partition boundary is affected by file structure and residence, as shown in table C-3.

Notice that in a file with W type records, a short PRU of level 0 terminates both a section and a partition.

TABLE C-3. PARTITION BOUNDARIES

Device	Record Type (RT)	Block Type (BT)	Physical Boundary
PRU device	W	I	A short PRU of level 0 containing a one-word deleted record pointing back to the last I block boundary, followed by a control word with a flag indicating a partition boundary.
	W	C	A short PRU of level 0 containing a control word with a flag indicating a partition boundary.
	D,F,R, T,U,Z	C	A short PRU of level 0 followed by a zero-length PRU of level 17g.
	S	-	A zero-length PRU of level number 17g.
S or L format tape	W	I	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a deleted one-word record pointing back to the last I block boundary, followed by a control word with a flag indicating a partition boundary.
	W	C	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a control word with a flag indicating a partition boundary.
Any other tape format	D,F,T, R,U,Z	C,K,E	A tapemark.
	S	-	A tapemark.
	-	-	Undefined.

Physical Record Unit (PRU) -
Under NOS and NOS/BE, the amount of information transmitted by a single physical operation of a specified device. The size of a PRU depends on the device, as shown in table C-4.

TABLE C-4. PRU SIZES

Device	Size in Number of 60-Bit Words
Mass storage (NOS and NOS/BE only).	64
Tape in SI format with coded data (NOS/BE only).	128
Tape in SI format with binary data.	512
Tape in I format (NOS only).	512
Tape in any other format.	Undefined.

A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

PRU Device -
Under NOS and NOS/BE, a mass storage device or a tape in SI or I format, so called because records on these devices are written in PRUs.

Procedure Name -
The name assigned to operations to be performed by an object program.

Projected Record Occurrence -
The record occurrence returned by Query Update that is the logical concatenation of selected information from all record descriptions specified in the relation.

Qualification -
The method whereby a nonunique name can be made unique. If the name exists within a hierarchy of names, it can be made unique by mentioning one or more of the higher levels of the hierarchy.

Qualifier Identifier -
A data item that is used to restrict which record occurrences are to be retrieved when reading a relation.

Random File -
In the context of CYBER Record Manager, a file with word addressable, indexed sequential, direct access, or actual key organization in which individual records can be accessed by the values of their keys.

In the context of the NOS or NOS/BE operating systems, a file with the random bit set in the file environment table in which individual records are accessed by their relative PRU numbers.

Record -
CYBER Record Manager defines a record as a group of related characters. A record or a portion thereof is the smallest collection of information passed between CYBER Record Manager and a user program. Eight different record types exist, as defined by the RT field of the file information table.

Other parts of the operating systems and their products might have an additional or different definition of records.

Record Occurrence -
An actual data base record that conforms to a record description in the sub-schema.

Record Qualification -
The method used to restrict which records are to be returned to the user by specifying criteria that must be satisfied by a record occurrence. Record qualification is allowed only for relation reads.

Record Type -
The term record type can have one of several meanings, depending on the context. CYBER Record Manager defines eight record types established by an RT field in the file information table. Tables output by the loader are classified as record types such as text, relocatable, or absolute, depending on the first few words of the tables.

Relation -
The logical structure formed by the joining of records based on common identifiers.

Relation Occurrence -
The logical concatenation of a record occurrence from each record description specified in the relation.

Relational Data Base -
A data base of files joined in relations through common data item identifiers.

Repeating Group -
A collection of data items that occur a number of times within a record; can consist of elementary items, group items, and vectors.

Schema -
A detailed description of the internal structure of the complete data base within the CDCS environment; synonymous with sub-schema for Query Update applications.

Section -
CYBER Record Manager defines a section as a division within a file with sequential organization. Generally, a section contains more than one record and is a division within a partition of a file. A section terminates with a physical representation of a section boundary, as shown in table C-5.

The NOS and NOS/BE operating systems equate a section with a system-logical-record of level 0 through 16 octal.

Sequential -
A file organization in which records are stored in the order they were generated.

TABLE C-5. SECTION BOUNDARIES

Device	Record Type (RT)	Block Type (BT)	Physical Representation
PRU device	W	I	A deleted one-word record pointing back to the last I block boundary followed by a control word with flags indicating a section boundary. At least the control word is in a short PRU of level 0.
	W	C	A control word with flags indicating a section boundary. The control word is in a short PRU of level 0.
	D,F,R,T,U,Z	C	A short PRU with a level less than 178.
	S	-	Undefined.
	W	I	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a deleted one-word record pointing back to the last I block boundary, followed by a control word with flags indicating a section boundary.
S or L format tape	W	C	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a control word with flags indicating a section boundary.
	D,F,R,T,U,Z	C,K,E	Undefined.
	S	-	Undefined.
	-	-	Undefined.
Any other tape format	-	-	Undefined.

Short PRU -

A PRU that does not contain as much user data as the PRU can hold and that is terminated by a system terminator with a level number.

Under NOS, a short PRU defines EOR.

Under NOS/BE, a short PRU defines the end of a system-logical-record. In the CYBER Record Manager context, a short PRU can have several interpretations depending on the record and blocking types.

Source Identifier -

A common identifier that is positioned on the left of a relational operator in a JOIN or RESTRICT clause.

Statement -

A source program entry that contains one or more DDL clauses.

Sub-Schema -

A detailed description of the data base as generated by DDL statements.

Sub-Schema Library -

A permanent file containing one or more sub-schemas.

System-Logical-Record -

Under NOS/BE, a data grouping that consists of one or more PRUs terminated by a short PRU or zero-length PRU. These records can be transferred between devices without loss of structure.

Equivalent to a logical record under NOS.

Equivalent to a CYBER Record Manager S type record.

Target Identifier -

A common identifier that is positioned on the right of a relational operator in a JOIN or RESTRICT clause.

Transaction -

The activity that was imposed upon a record by a Query Update directive.

Variable Occurrence Data Item -

A data item that is repeated a specific number of times for each record; the number of occurrences is controlled by a preceding elementary data item; uses format 2 of the OCCURS clause.

Vector -

An elementary data item that is repeated a number of times in each record.

W Type Record -

One of the eight record types supported by CYBER Record Manager. Such records appear in storage preceded by a system-supplied control word. The existence of the control word allows files with sequential organization to have both partition and section boundaries.

Zero-Byte Terminator -

Twelve bits of zero in the low-order position of a word that marks the end of the line to be displayed at a terminal or printed on a line printer. The image of cards input through the card reader or terminal also has such a terminator.

Zero-Length PRU -

A PRU that contains system information, but no user data. Under CYBER Record Manager, a zero-length PRU of level 17 is a partition boundary. Under NOS, a zero-length PRU defines EOF.

0

0

0

0

0

0

0

RESERVED WORD LIST

D

<p>ABS ACCESS ACTUAL AFTER ALL ALLOWED ALTER ALTERNATE AND ANY ARE AREA AREA-NAME AREAS AS ASCENDING ASSETS ASSIGN AT ATTACH</p> <p>BATCH BEFORE BEYOND BLOCK BLOCK-TYPE BLOCKS BREAK BY</p> <p>CALL CATALOG CENTER CENTERED CHAR CHARACTER CHARACTER-COUNT CHARACTERS CLOSE COBOL COLUMN COLUMNS COMP COMP-1 COMP-2 COMPILE COMPLEX COMPUTATIONAL COMPUTATIONAL-1 COMPUTATIONAL-2 CONTAINS CONTROL-WORD COUNT CRE CREATE CURRENT-ANY CURRENT-AREA CURRENT-DATE CURRENT-KEY CURRENT-LINE CURRENT-PAGE CURRENT-REPORT CURRENT-SESSION CURRENT-TIME CY</p>	<p>DATA DATA-PADDING DATE DECODE DEF DEFAULT DEFINE DEL DELETE DEPEND DEPENDING DESCE DESCENDING DESCR DESCRIBE DESCRIBED DETAIL DIAG DIAGNOSTIC DIRECT DIS DISCARD DISPLAY DIVISION DOUB DOUBLE DUP DUPLICATE DUPLICATES</p> <p>EJECT END EQ ERASE EVAL EVALUATE EXACT-RECORDS EXEC EXECUTE EXH EXHIBIT EXTRACT</p> <p>F FALSE FETCH FILE FILE-LIMIT FILES FILLER FIRST FIX FIXED FIXED-LENGTH FL FLOATING FOOTING FOR FORMAT FORTRAN FROM FULL</p>	<p>GE GREG GT</p> <p>HEADING HELP HIGH-VALUE HORIZ HORIZONTAL</p> <p>I-O ID IDENTIFICATION IF IMAGES IN INDEX INDEX-BLOCK INDEX-LEVEL INDEX-PADDING INDEXED INPUT INS INSERT INT INTEGER INTERNAL INV INVOKE IS ITEM ITEM-SIZE ITEMS</p> <p>JOIN JULIAN</p> <p>KEY KEY-FIELD</p> <p>LAND LAST LE LIBRARY LINE LINES LOCATION LOG LOGICAL LOR LOW-VALUE LT LXOR</p> <p>M MASK MATCH MAX MAXS MEAN MIN</p>	<p>MINS MINUS MISMATCH MOD MODE MODIFIED MODIFY MOVE MR</p> <p>NAME NE NEGATIVE NEXT NO NOT NOTE NULL-VALUE NUM NUMBER NUMERIC</p> <p>OCCURS OF OFF ON ONCE OPEN OR ORGANIZATION OS OVER OVERFLOW</p> <p>PA PAGE PAGE-NUMBER PAGE-SIZE PAGES PAR PARALLEL PART PASS PERCENT PERF PERFORM PIC PICTURE PLUS PN POINT PREFACE PREPARE PREVIEW PROCEDURE PURGE PW</p> <p>Q</p>
--	--	--	--

R
RAISED
REC
RECAP
RECAP-LINE
RECORD
RECORD-COUNT
RECORD-MARK
RECORD-NAME
RECORD-TYPE
RECORDING
RECORDS
RECOVERY
REL
RELATION
RELATION-NAME
REM
REMOVE
REPEAT
REPORT
REPORTS
RESTRIC
RET
RETRIEVAL
RETURN
REW
REWIND
RM
RW

SAME
SCAN
SCHEMA
SDA
SEARCH
SECT
SECTIONS
SELECT
SEND
SEP
SEPARATOR
SEQ
SEQUENCE
SEQUENTIAL
SESSION
SESSIONS
SETTING
SIS
SITUATE
SN
SORT
SORTED
SPACE
SPEC
SPECIFY
STOP
STORE
STRING
SUB-SCHEMA

SUM
SUMMARY
SWITCH

T
TAB
TABS
TEMP
TEMPORARY
TEXT
TIME
TIMES
TITLE
TITLE-LINE
TO
TRAILER-COUNT
TRANS
TRANSACTION
TRUE
TYPE

UN
UNDEFINED
UNI
UNIQUE
UNIVERSAL
UNTIL

UPD
UPDATE
UPON
USAGE
USE
USING

VAL
VALUE
VERIFY
VERSION
VERT
VERTICAL
VETO
VIA

WHERE

XOR

YCLEPT

ZERO
ZERO-BYTE

The format specifications for all DDL statements used to define a Query Update sub-schema are summarized and listed in this appendix. Detailed information for each format is referenced by page number and appears in section 4 of this manual.

IDENTIFICATION DIVISION

<u>IDENTIFICATION DIVISION.</u>	4-1
<u>SUB-SCHEMA NAME</u> IS sub-schema-name	4-1
[<u>PROCEDURE LIBRARY</u> IS library-name [permanent-file-parameters]]	4-1

DATA DIVISION

<u>DATA DIVISION.</u>	4-1
<u>AREA-NAME</u> IS area-name [<u>TEMPORARY</u> permanent-file-parameters]	4-2
[<u>INDEX</u> IS index-file-name [permanent-file-parameters]]	4-2
[<u>BLOCK CONTAINS</u> [integer-1 TO] integer-2 { <u>RECORDS</u> <u>CHARACTERS</u> }]	4-2
[<u>BLOCK-TYPE</u> IS { <u>INTERNAL</u> <u>CHARACTER-COUNT</u> <u>RECORD-COUNT</u> <u>EXACT-RECORDS</u> }]	4-3
[<u>RECORD CONTAINS</u> [integer-1 TO] integer-2 CHARACTERS [<u>DEPENDING ON</u> data-name]]	4-3
[<u>RECORD-TYPE</u> IS { <u>CONTROL-WORD</u> <u>FIXED-LENGTH</u> <u>CHARACTER-COUNT</u> <u>TRAILER-COUNT</u> <u>RECORD-MARK</u> <u>UNDEFINED</u> <u>ZERO-BYTE</u> }]	4-3
[<u>LOG</u> [{ <u>RECORD</u> <u>REC</u> } [[<u>BEFORE</u> <u>AFTER</u>]]] [{ <u>TRANSACTION</u> <u>TRANS</u> }]]	4-4
[<u>ASSIGN TO</u> permanent-file-name [{ <u>ID</u> IS id [<u>PW</u> IS pw . . .] [<u>CY</u> IS cy] [<u>SN</u> IS sn] }] [{ <u>UN</u> IS un [<u>PW</u> IS pw] <u>M</u> IS m [<u>PN</u> IS pn }]]	4-4

[ORGANIZATION IS { DIRECT [procedure-name] }
 ACTUAL { OLD }
 INDEXED { NEW }
 SEQUENTIAL [SORTED] { ASCENDING }
 DESCENDING }] 4-5
 [KEY IS data-name [DUPLICATES ARE { FIRST }
 LAST } ALLOWED]
 [KEY IS ALTERNATE data-name [DUPLICATES ARE { INDEXED }
 FIRST }
 NOT } ALLOWED] ...]

[FILE-LIMIT IS integer] 4-6

[SEQUENCE IS { FORTRAN }
 COBOL }
 literal] 4-6

[INDEX-LEVEL IS integer] 4-6

[INDEX-BLOCK CONTAINS integer CHARACTERS] 4-7

[INDEX-PADDING IS integer PERCENT] 4-7

[DATA-PADDING IS integer PERCENT] 4-7

[NUMBER OF BLOCKS IS integer] 4-7

[ON || OPEN ||
 SEARCH ||
 MATCH ||
 MISMATCH ||
 UPDATE ||
 DISPLAY ||
 CLOSE ||
 RETRIEVAL ||] CALL procedure-name ... 4-7

level number { data-name }
 FILLER } 4-8

RECORD-NAME IS record-name 4-8

[{ PICTURE } IS picture-specification]
 PIC } 4-8

[OCCURS integer-2 TIMES]

4-12

[OCCURS [integer-1 TO] integer-2 TIMES DEPENDING ON data-name-1]

4-12

[<u>USAGE IS</u>]	}	DISPLAY <u>COMPUTATIONAL</u> COMP <u>COMPUTATIONAL-1</u> COMP-1 <u>COMPUTATIONAL-2</u> COMP-2 COMPLEX LOGICAL <u>INTEGER</u>	}
---------------------------	---	---	---

4-13

RELATION DIVISION

RELATION DIVISION.

4-14

RELATION-NAME IS relation-name

4-14

JOIN WHERE identifier-1 EQ identifier-2

4-14

[identifier-3 EQ identifier-4] ...

[<u>RESTRICT</u> record-name WHERE [NOT] [(] identifier-1	{	EQ NE GT LT GE LE	}	{ identifier-2 } literal-1	[]	...	
[<u>AND</u> <u>OR</u> <u>XOR</u>	[NOT] [(] identifier-3	{	EQ NE GT LT GE LE	}	{ identifier-4 } literal-2	[]

4-15

O

O

O

O

O

O

O

The DDL ON CALL clause permits transfer of control to special coding, called data base procedures, during a Query Update session. The procedures are written, compiled, and stored as permanent files by the data administrator. Exits to the procedures allow special processing to be performed on the area level (OPEN, CLOSE) and on the record level (SEARCH, MATCH, MISMATCH, UPDATE, DISPLAY, and RETRIEVAL).

Data base procedures are relocatable binary routines. They can be:

- COBOL subprograms (compiled with the subcompile parameter SUBM in the COBOL control statement)
- FORTRAN subprograms
- COMPASS subprograms (no transfer address in the END statement)

LOADING OF PROCEDURES

Query Update loads data base procedures from the procedure library when one is specified in the sub-schema. If no procedure library is specified, Query Update loads the procedures from local file QUEXITS if such a file exists at the time of the INVOKE (or USE) or CREATE directive. If QUEXITS does not exist, Query Update loads the procedures from system library SYSMISC.

Externals from the procedure library, if one exists, are satisfied first, followed in order by externals from the system library SYSMISC, the global library set, and the local library set. If a procedure for the areas to be used is not located, the INVOKE (or USE) or CREATE directive is rejected, and the following diagnostic is issued:

ENTRY POINT name NOT FOUND FOR AREA name

LINKAGE AND COMMUNICATION

Query Update and data base procedure communication under version 3.1 and later versions is handled through a parameter list similar to that used by CDCS. This is in contrast to the method employed for version 3.0 in which communication was handled through a common block named CCOMMON. Version 3.0 procedures are being maintained through a linkage module named QUEXIT, which copies the parameters into CCOMMON and performs integer to COMPUTATIONAL-2 conversion as necessary. Details regarding QUEXIT appear in the Maintaining Version 3.0 Procedures subsection. Since additional overhead is required for calls to version 3.0 procedures, it is recommended that all procedures be converted to use parameter lists.

A parameter list consists of one-word addresses, which are stored right-justified with leading zeros. The entire parameter list is terminated by a word of zeros. COMPASS procedures obtain the address of the parameter list in register A. COBOL procedures use the Linkage Section and the USING clause in the PROCEDURE DIVISION statement to handle the parameter list. FORTRAN procedures use the standard subroutine linkage.

The parameter list is described in table F-1. The CCOMMON block used for version 3.0 is included for comparison purposes. The parameter list is identical to that used with CDCS procedures with two exceptions: the meanings of EC and RC values vary, and the third parameter is the directive code DIRCODE instead of an error code.

TABLE F-1. PARAMETER LIST AND CCOMMON ORGANIZATION

Parameter List		CCOMMON	
Parameter Number	Parameter Name	Word	Name
1	EC (I)	1	EC,RC (R)
2	RC (I)	2	DIRCODE (R)
3	DIRCODE (I)	3-27	FIT
4	FIT	28	KEYLG (R)
5	RECAREA	29-54	KEYAREA
6	RECLG (I)	55	RECLG (R)
7	KEYAREA	56-n	RECAREA
8	KEYPOS (I)		
9	KEYLG (I)		

In CCOMMON, EC and RC both refer to the first word.

In CCOMMON, the length of RECAREA is set by the first procedure loaded that declares it. RECAREA should be defined large enough to hold the longest possible record for any area that will be in use.

(I) denotes INTEGER format.

(R) denotes REAL (COMPUTATIONAL-2) format.

Each parameter is identified as follows:

Entry Code (EC)

Query Update sets EC to an entry code (1 through 8) to represent the condition that caused an exit to the procedure. Entry codes are detailed in the Procedure Processing subsection.

Return Code (RC)

The procedure sets RC to a return code (0 through 3) when return is to Query Update. Return codes are detailed in the Procedure Processing subsection.

Current Directive Code (DIRCODE)

Query Update sets DIRCODE to a code (1 through 11) to indicate the current directive. DIRCODE values and their meanings are listed in table F-2.

TABLE F-2. DIRCODE VALUES

Query Update Directive	Corresponding DIRCODE Value
DISPLAY	1
REMOVE (or DELETE)	2 3
STORE (or INSERT)	4 5
MODIFY (or UPDATE)	6 7
END (or STOP)	8
EXTRACT	9
CREATE	10
OTHER	11 (OTHER indicates multiple directive transmissions)

File Information Table (FIT)

Query Update sets the 25-word file information table of the area in use to provide reference for the procedure.

Record Area (RECAREA)

RECAREA is a storage space that must be large enough to store the longest record that can be found in the area. Query Update stores the current record in RECAREA when a MATCH, MISMATCH, UPDATE, DISPLAY, or RETRIEVAL procedure is entered. A SEARCH procedure stores records in RECAREA for Query Update. An OPEN or CLOSE procedure does not use RECAREA.

Record Length (RECLG)

Query Update sets the record length (in characters) in RECLG. When a record is supplied by a SEARCH procedure, the procedure sets the record length in RECLG.

Key Area (KEYAREA)

Query Update stores the current record key in KEYAREA, which is in an array of up to 26 words. When a record is supplied by a SEARCH procedure, the procedure stores the current record key in KEYAREA.

Key Position (KEYPOS)

Query Update sets the starting character position (0 through 9) of the current record in KEYPOS.

Key Length (KEYLG)

Query Update sets the current record key length (in characters) left-justified in KEYLG. When a record is supplied by a SEARCH procedure, the procedure sets the current record key length in KEYLG.

PROCEDURE PROCESSING

Data base procedures are not restricted in the operations to be performed; however, a return code must be set in RC before control is returned to Query Update.

Entry codes are established as a convenience to the programmer. They can be examined by a procedure to determine the condition causing a call to the procedure. Similarly, a procedure can examine the Current Directive Code in DIRCODE to determine the directive causing an entry to the procedure.

Operations are performed within data base procedures at programmer option. Table F-3 lists entry codes, conditions, return codes, and actions subsequent to data base procedure processing. If an action releases an area and is to get the next directive, a user-submitted INVOKE (or USE) directive is required.

OPEN PROCEDURE EXIT

The OPEN procedure exit is taken at the beginning of a transmission in which an area is accessed. FIT and KEYLG are updated to reflect the current area being opened.

Before the OPEN procedure exit is taken, Query Update performs a CYBER Record Manager (CRM) open of the area unless a SEARCH procedure exists. If a SEARCH procedure exists, the OPEN procedure rather than Query Update must perform the actual CRM open of the appropriate area.

SEARCH PROCEDURE EXIT

The SEARCH procedure exit is taken to provide a means of reading and writing records that are not acceptable to CRM. The SEARCH procedure rather than Query Update accesses the area.

The SEARCH procedure must supply records for the following directives: DISPLAY, REMOVE (or DELETE), MODIFY (or UPDATE), and EXTRACT. The directive code field is set to the appropriate value (refer to table F-2), which indicates the Query Update directive that caused the SEARCH exit to be taken.

When returning records to Query Update, the procedure must set KEYLG and RECLG to the correct values and fill KEYAREA and RECAREA with the correct data.

TABLE F-3. PROCEDURE PROCESSING CODES

Entry Code	Condition	Return Codes			
		0	1	2	3
1	OPEN	Proceed normally.	Release area; get next directive. ① ②	Terminate Query Update session. ③	Terminate Query Update session. ③
2	SEARCH	Procedure has provided record.	End of area reached; no more records are available.	No more records are available; get next directive. ②	Release area, get next directive; area will be closed and cannot be referenced until a subsequent INVOKE (or USE) statement is issued; if a CLOSE procedure is also specified for this area, it will be executed. ① ②
3	MATCH	Proceed normally.	Drop the record; record is treated as if it did not meet qualification criteria.	Get next directive. ②	Release area, get next directive. ① ②
4	MISMATCH	Proceed normally.	Accept the record; record is treated as if it did not meet qualification criteria.	Get next directive. ②	Release area, get next directive. ① ②
5	UPDATE	Proceed normally.	Do not alter area. ④	Get next directive. ②	Release area, get next directive. ① ②
6	DISPLAY	Proceed normally.	Do not display message.	Do not display message; get next directive. ②	Release area, get next directive. ① ②
7	CLOSE	Get next directive.	Terminate Query Update session. ① ③	Terminate Query Update session. ① ③	Terminate Query Update session. ① ③
8	RETRIEVAL	Continue.	Reject record.	Get next directive. ②	Close area, get next directive. ① ②

① AREA CLOSED BY DATA BASE PROCEDURE message is issued.
 ② CURRENT DIRECTIVE TERMINATED BY DATA BASE PROCEDURE message is issued.
 ③ SESSION TERMINATED BY DATA BASE PROCEDURE message is issued.
 ④ RECORD IGNORED BY DATA BASE PROCEDURE message is issued.

MATCH PROCEDURE EXIT

The MATCH procedure exit is taken when a record has been retrieved either by Query Update or a SEARCH procedure. This occurs when the DISPLAY, REMOVE (or DELETE), MODIFY (or UPDATE), and EXTRACT directives cause records to be retrieved. If the retrieval directive is the object of a conditional statement, the MATCH exit is taken only when the records meet the qualification criteria. If the retrieval directive is not the object of a conditional statement, the MATCH exit

is taken for every record retrieved. RECAREA, RECLG, and KEYAREA reflect the record retrieved.

If the MATCH procedure directs Query Update to reject a record, the following message is displayed upon completion of the transmission:

p FORCED q REJECTED

The letter q represents the number of records rejected by the MATCH procedure.

MISMATCH PROCEDURE EXIT

The MISMATCH procedure exit is taken when a record has been retrieved by Query Update or by a SEARCH procedure and the record does not qualify for the current IF condition. RECAREA, RECLG, and KEYAREA reflect the record retrieved.

If the MISMATCH procedure forces Query Update to accept a record, the following message is displayed upon completion of the transmission:

p FORCED q REJECTED

The letter p represents the number of records forced to be accepted by the MISMATCH procedure.

UPDATE PROCEDURE EXIT

The UPDATE procedure exit is taken when Query Update receives a REMOVE, STORE, or MODIFY (or DELETE, INSERT, or UPDATE) directive. The directive code reflects which condition is currently being processed.

When a REMOVE (or DELETE) directive is involved, the appropriate record has been copied into RECAREA; it has not been deleted from the area at the time the procedure assumes control.

When the directive MODIFY (or UPDATE), MOVE, or MODIFY USING (or UPDATE USING) is involved, the selected record is copied and modified in RECAREA. The record is not rewritten to the area in use at the time that the procedure assumes control.

When the directive STORE (or INSERT) or STORE USING (or INSERT USING) is involved, the new record content is in RECAREA, but is not placed in the area in use when the procedure assumes control. The record in RECAREA is formatted with blanks and zeros or specified data for the STORE (or INSERT) or STORE USING (or INSERT USING) directives.

If the user specifies verification with a VETO option, the record placed in RECAREA has passed the VETO. The MOVE, REMOVE, MODIFY, or STORE (or DELETE, UPDATE, or INSERT) directive does not modify the area before an exit to a procedure.

If both a SEARCH and an UPDATE procedure have been specified, the procedures rather than Query Update are performing all I/O. The SEARCH procedure is retrieving all records and the UPDATE procedure is writing and rewriting all records. If logging has been specified, the procedure must ensure that the return code reflects the action taken and that the FIT contains all information regarding record length, key length, key position, and so forth.

DISPLAY PROCEDURE EXIT

The DISPLAY procedure exit is taken when the current directive is DISPLAY. The exit is taken for each record.

CLOSE PROCEDURE EXIT

The CLOSE procedure exit is taken at the end of a transmission in which an area is accessed. When the CLOSE exit is taken, Query Update has not performed the CRM close of the area.

Upon return from the CLOSE exit, Query Update performs the CRM close unless a SEARCH procedure has been specified; if the SEARCH procedure exists, the CLOSE procedure must perform the close of the area.

RETRIEVAL PROCEDURE EXIT

The RETRIEVAL procedure exit is taken after a record has been read by CRM but before Query Update has applied any qualification criteria.

The exit enables the user to specify compression or decompression of records or to speed up relation processing by allowing Query Update to reject records during processing. The capability for compression or decompression of records is available only for extended indexed sequential, extended direct access, and extended actual key files.

PROCEDURE RELATIONSHIPS

The manner in which conditions relate to one another is illustrated in table F-4. The table indicates a series of typical Query Update directives for an area that has ON CALL procedure exits for OPEN, RETRIEVAL, MATCH, MISMATCH, UPDATE, DISPLAY, and CLOSE.

TABLE F-4. PROCEDURE RELATIONSHIPS

Query Update Directive Executed	Sequence of Data Base Procedure Exits Taken
CREATE	OPEN CLOSE
STORE (or INSERT) n records	OPEN UPDATE n times CLOSE
DISPLAY these n records	OPEN RETRIEVAL } MATCH } n times DISPLAY } CLOSE
IF condition REMOVE (or DELETE) where first n records fail and remaining p records qualify	OPEN RETRIEVAL n+p times MISMATCH n times MATCH } UPDATE } p times CLOSE

PROCEDURE INPUT/OUTPUT

Query Update provides certain entry points to allow the procedure to communicate with the user or CRM. Parameters are passed to the entry points according to the standard rules governing subprogram calls as follows:

In COBOL:

ENTER entry point name USING parameter name list.

In FORTRAN:

CALL entry point name (parameter name list).

In COMPASS:

SA1 List of parameter addresses, terminated by a word of zeros.

RJ = X entry point name.

CRM ENTRY POINTS

Query Update supports the following entry points to the CRM FORTRAN interface routines:

OPENM	PUT	IFETCH	FILEAK
CLOSEM	DLTE	STOREF	FILEDA
GET	REPLC	FILEIS	FILESQ
GETN			

Calling sequences are described in the CYBER Record Manager user's guides. Calls to routines that do not exist in the current Query Update overlay are trapped and diagnosed; the called entry point and the procedure most recently called by Query Update are identified.

If the PUT routine is called by a data base procedure to process any file other than a sequential file, the file cannot be accessed in read-only mode. If the DLTE or REPLC routine is called by a data base procedure to process any file, the file cannot be accessed in read-only mode.

An attempt to use CRM functions not provided by the supported entry points results in a large amount of CRM relocatable code being loaded into memory along with the user's data base procedures.

QUERY UPDATE UTILITY ROUTINE ENTRY POINTS

Entry points for utility routines supplied by Query Update include:

QUREAD	D.PACK	QUCINI	QUCONF
QUWRITE	D.UNPAK	QUCIN2	

QUREAD and QUWRITE allow the procedure to communicate with the terminal user (refer to the Query Update reference manual for background information on the concepts of batch or interactive mode, transmissions, or continuation symbols). QUREAD and QUWRITE are both called with two parameters, the second of which is described as COMP-2 in COBOL, REAL in FORTRAN, and normalized floating-point in COMPASS.

QUREAD

Working storage area. The area designated by this parameter should be large enough to hold the maximum number of characters the user expects to enter; QUREAD cannot deliver more than the maximum transmission length for each instance of the call.

Number of characters read. This parameter is filled in by QUREAD to indicate the number of characters actually transmitted into the working storage area. If QUREAD returns a zero, the end-of-file condition was encountered and no user input is available.

QUWRITE

Storage area. QUWRITE picks up the character string from the area designated by this parameter to deliver to the user; the first character of the string is treated as a control character, according to the standard conventions:

blank Print on the next line.

1 Skip to the top of the page or screen before printing.

0 Skip one line before printing.

Number of characters in the string. The number designated by this parameter includes the leading control character.

QUCIN1, QUCIN2, and QUCONF are numeric conversion routines provided for COBOL users. D.PACK and D.UNPAK are alternate names for QUCIN1 and QUCONF, respectively, and are included for commonality with CDCS.

QUCINI (D.PACK)

QUCINI converts a list of integers to COBOL4 COMPUTATIONAL-1 (unnormalized floating-point) format.

QUCIN2

QUCIN2 converts a list of integers to COMPUTATIONAL-2 (normalized floating-point) format.

QUCONF (D.UNPAK)

QUCONF converts a list of floating-point numbers to integer format.

The result of any conversion is stored directly in the original data item. For example:

ENTER QUCONF USING A B C.

results in the contents of A, B, and C being converted from floating-point to integer.

MAINTAINING VERSION 3.0 PROCEDURES

The QUEXIT linkage module is retained for processing data base procedures that were written for version 3.0.

QUEXIT copies parameters from the parameter list to common block CCOMMON before passing control to the procedure. Version 3.0 and later version procedures cannot be mixed; if any 3.0 procedures are loaded, then all calls to all procedures pass through QUEXIT and are processed as 3.0 procedure calls.

Reference to a QUEXIT entry point forces QUEXIT to be loaded. The following entry points are available:

QUOPEN	QUPUT	QUFETCH	QUFAK
QUCLOSE	QUDLTE	QUSTORE	QUFDA
QUGET	QUREPLC	QUFIX	QUFSQ
QUGETN			

SAMPLE DATA BASE PROCEDURE

A sample data base procedure written in COBOL 5 and compiled under the NOS operating system is shown in figure F-1. The LIBGEN control statement is included to illustrate the storage of the procedure in a user library.

The LIBGEN control statement specifies that the routine to be made a library file is found in file LGO, the default binary file of the COBOL program. The library is created in permanent file QUPROC0 with the name of the library being QUPROC0.

The deck structure for an equivalent job under the NOS/BE operating system is as follows:

```
Job Statement
REQUEST(QUPROC0,*PF)
COBOL5(MSB,LO=S/R)
EDITLIB(USER)
CATALOG(QUPROC0,ID=DPB,RP=999)
7/8/9 card
```

Program, DBPROC

```
7/8/9 card
LIBRARY(QUPROC0,NEW)
REWIND(LGO)
ADD(*,LGO)
FINISH.
ENDRUN.
6/7/8/9 card
```



```

1      IDENTIFICATION DIVISION.
2      PROGRAM-ID. DBPRCC.
3
4      ENVIRONMENT DIVISION.
5      CONFIGURATION SECTION.
6      SOURCE-COMPUTER. CYBER74.
7      OBJECT-COMPUTER. CYBER74.
8
9      DATA DIVISION.
10     WORKING-STORAGE SECTION.
11     01 LENGTH-1                COMP-2.
12     01 MSG1.
13         02 IBLANK1            PIC X VALUE IS # #.
14         02 FILE-NAME          PIC X(7).
15         02 IBLANK2            PIC X VALUE IS # #.
16         02 PRT                 PIC X(25).
17     01 PW                      PIC X(3).
18
19     LINKAGE SECTION.
20     77 ENTRY-CODE             PIC 9(10) COMP-1.
21     77 RETURN-CODE           COMP-2.
22     77 DIRCODE               PIC 9(10) COMP-1.
23     01 FILE-INFO-TABLE       PIC X(07).
24     01 CURRENT-RECORD        PIC X(25).
25     01 RECORD-LENGTH         PIC 9(10) COMP-1.
26     01 CURRENT-KEY           PIC X.
27     01 KEY-POS                PIC 9(10) COMP-1.
28     01 KEY-LENGTH            PIC 9(10) COMP-1.
29
30     PROCEDURE DIVISION USING ENTRY-CODE, RETURN-CODE, DIRCODE,
31     FILE-INFO-TABLE, CURRENT-RECORD, RECORD-LENGTH,
32     CURRENT-KEY, KEY-POS, KEY-LENGTH.
33     START-1.
34         MOVE 34 TO LENGTH-1.
35         MOVE FILE-INFO-TABLE TO FILE-NAME.
36         MOVE CURRENT-RECORD TO PRT.
37         ENTER QUWRITE USING MSG1 LENGTH-1.
38         MOVE 0 TO RETURN-CODE.
39         ENTER QUCONF USING RETURN-CODE.
40     END-DBP-A.
41     EXIT PROGRAM.

```

ACMIBGQ. 79/06/06.(22) SVL SN614 NCS

```

12.44.12.DDL3QU.
12.44.12.UCCR, 6125,      0.047KCS.
12.44.12.USER(          )
12.44.12.CHARGE(        )
12.44.13.COBOL5(MSB,LO=S/R)
12.44.23.0667008 CM REQUIRED,      .986 CPU SECS U
12.44.23.SED, 000008 ECS USED
12.44.23.DEFINE(QUPROC/PW=,CT=PU,M=W)
12.44.24.LIBGEN(F=LGO,P=QUPROC)
12.44.24. LIBRARY GENERATION COMPLETE.
12.44.24.UEAD,      0.012KUNS.
12.44.24.UEPF,      0.013KUNS.
12.44.24.UEMS,      2.364KUNS.
12.44.24.UECP,      1.036SECS.
12.44.24.AESR,      4.376UNTS.
12.46.50.UCLP, 6122,      0.256KUNS.

```

Figure F-1. Sample Data Base Procedure

O

O

O

O

O

O

O

The DDL LOG clause is used to maintain a record of user interaction with the data base. When this clause is included in a sub-schema, information vital to the total recovery or minor restoration of the data base is recorded in a log file. Logging operations for Query Update are restricted to disk. A permanent log file is established by the ASSIGN phrase in the LOG clause.

Each area named in a sub-schema (64 maximum) can specify logging, and each area can have a separate log file. Logging can be performed for any of the four file organizations recognized by Query Update and CRM: actual key, direct access, indexed sequential, and sequential. Logging operations do not occur for query-only transmissions. Before image, after image, and transaction logging are performed when transmissions have the potential to update a file; that is, whenever a MODIFY, STORE, or REMOVE (or UPDATE, INSERT, or DELETE) directive appears in a transmission.

Query Update updates one file at a time and concurrently writes the logging information directly to the log file. Log file entries produced by Query Update 3.1 and later versions are not compatible with entries produced by version 3.0. New log files must be established for Query Update 3.1 and later versions, and applications written to interpret the log file must be changed to recognize the new format.

SYSTEM FLOW

The data administrator assumes the major role in the series of programming operations that are performed to prepare for recovery of the data base and to call the recover and restore utilities. Figure G-1 shows the definition, execution and logging, and recovery phases.

A data base area named in the sub-schema can be selected for logging operations via DDL. This definition phase occurs prior to the execution and logging phase indicated in the illustration. The sub-schema directory is produced by the DDL compiler and holds the logging options.

Query Update automatically attaches the log file; no action is required on the part of the user. During the execution of a transmission, Query Update updates one area at a time and calls the data base utility logging routines to write logging information directly to the log file. Log records are recorded on the disk log file for the data base area being accessed. The log file can contain five types of log records or entries:

- Open
- Close
- Transaction
- Before image
- After image

Backup copies of the data base can be made on a periodic basis using the operating system utility routine for dumping permanent files. Before initiating the recover or restore

utility, the exact status of the data base can be determined by employing FORM, Query Update, or a COBOL program to read the log file and to create a log report. The report can be used to decide which area or areas of the data base require recovery or restoration, as well as which Query Update operations must be retransmitted.

Control statements are used to call the recover or restore utility. A backup copy of the data base, which can be loaded using the operating system utility for loading permanent files, is required for recovery. If the data base is to be restored, a backup copy of the data base need not be loaded. The restore utility works from the current state of the data base.

LOG FILE STRUCTURE

Logging is performed by the data base logging routines as detailed in the Data Base Utilities reference manual. Query Update log file structure must conform to the formats described in that document. The following should be noted for Query Update:

- The open and close log records are generated each time Query Update performs an open and a close on an area that is being updated.
- Word 0 of all log record formats. Type code 9 and directive codes A, B, D, and N do not apply for Query Update.
- Word 2 of all log record formats. User identification is the 7-character job name by which the job is known in the system dayfile.
- Transaction log record format. Word 5 is zero filled, and words 6-n contain the Query Update transmission.

INITIALIZING THE LOG FILE

A log file is initialized through a REQUEST/REWIND/CATALOG operation under NOS/BE and through a DEFINE operation under NOS. Figure G-2 shows the necessary control statements for permanent file initialization in a separate job. This step can be accomplished either in the same job sequence prior to the execution of the application program using the log file or in a separate job.

One exception should be noted. If the log file is being defined within a Query Update program run under NOS, the log file must be returned to the operating system before the QU control statement is issued. It is recommended that all log files be initialized by separate job runs.

LOG PROCESSING ERRORS

Errors encountered by the logging routines during the processing of a log file are not fatal to Query Update. All errors are handled by the Query Update program, and error messages are not applicable to the user.

RECOVER/RESTORE UTILITIES

The recover and restore off-line utilities are used to accomplish data base recovery. The utilities are called by the DFRCV and DFRST control statements. The recover utility reads after record images from the log file and inserts them in a loaded backup copy of the data base. The restore utility reads before record images in reverse chronological order and replaces existing data base records.

Detailed information regarding the utilities and the control statements through which they are called appears in the Data Base Utilities reference manual.

The following should be noted for Query Update:

The user-id entered in the DFRCV and DFRST control statements is the 7-character job name by which the job is known in the system dayfile.

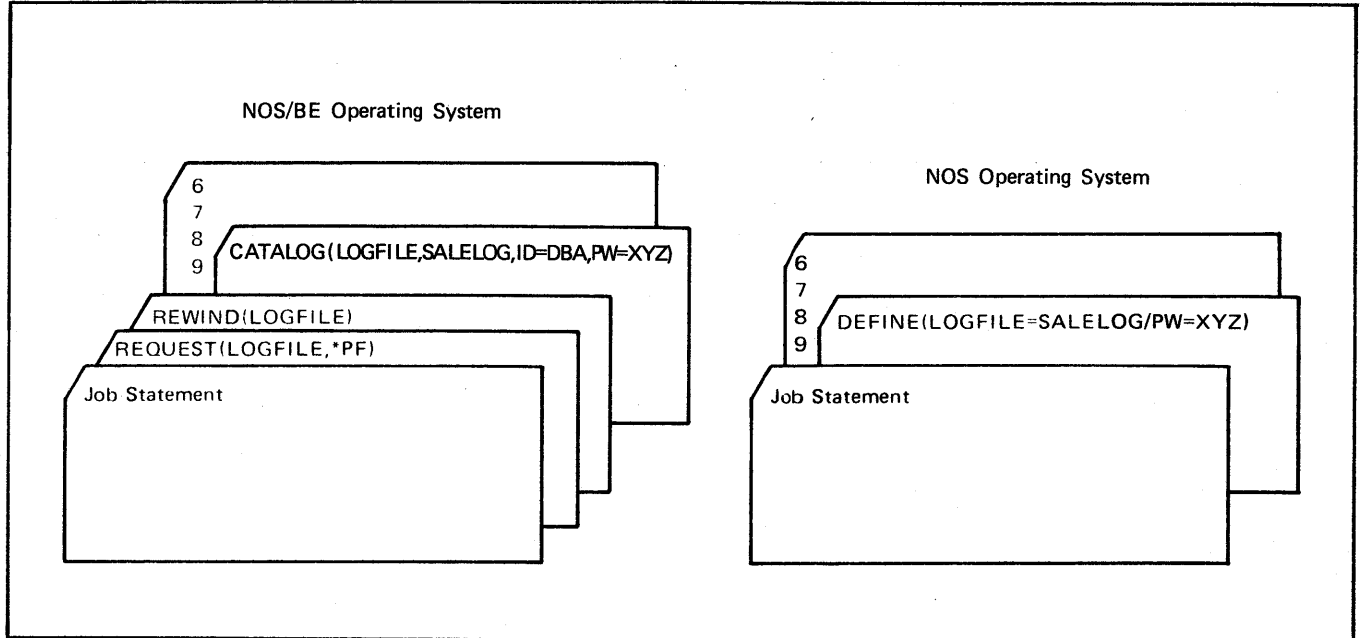


Figure G-2. Permanent File Log File Initialization

O

O

O

O

O

O

O

This summary includes all the clauses or statements that can be used in defining data items. Table H-1 shows the schema definition required for data items of each schema data class and the sub-schema definitions that correspond to each schema data class.

For Query Update access to schema-defined data base files in CYBER Record Manager (CRM) data base access mode, the data base must be defined in the Query Update sub-schema exactly as the data base is defined in the schema. Therefore, every data item must be defined to correspond in size and class to the schema definition of the item.

For COBOL, FORTRAN, and Query Update access to schema-defined files in CYBER Database Control System (CDCS) data base access mode, the definition of data items in the sub-schemas does not have to correspond exactly to the schema definition of data items. Through mapping, CDCS can generate a record image conforming to the sub-schema format from a record in schema format, or can perform the conversion of data from sub-schema format to schema format. Detailed information about conversions allowed is included in the CDCS 2, the FORTRAN Data Base Facility, the DDL 3 volume 1, and the DDL 3 volume 2 reference manuals.

TABLE H-1. DATA DEFINITION IN DMS-170

Data Class No.	Data Class Name	SCHEMA			COBOL Sub-Schema		FORTRAN 5 Sub-Schema Type Statement	FORTRAN Extended 4 Sub-Schema Type Statement	Query Update Sub-Schema in CDCS Data Base Access Mode		Query Update Sub-Schema in CRM Data Base Access Mode	
		PICTURE Clause	TYPE Clause	Internal Representation	PICTURE Clause	USAGE Clause			PICTURE Clause	USAGE Clause	PICTURE Clause	USAGE Clause
0	Display alpha-numeric	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	CHARACTER	Display code, alphanumeric	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)	CHARACTER	INTEGER	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)
1	Display alphabetic	Alphabetic (A)	None	Display code, alphabetic	Alphabetic (A)	DISPLAY (or none)	CHARACTER	INTEGER	Alphabetic (A)	DISPLAY (or none)	Alphabetic (A) (or none)	DISPLAY (or none)
3	Display integer	Numeric (9 T)	None	Display code numeric, can have sign overpunch in last character position	Numeric (9 S)	DISPLAY COMP (or none)	None†	None†	Numeric (9 S and insertion and replacement characters)	DISPLAY COMP (or none)	Numeric (9 S and insertion and replacement characters)	DISPLAY COMP (or none)
4	Display fixed	Numeric (9 P T V .)	None	Display code numeric plus implicit decimal or scaling position	Numeric (9 S V P)	DISPLAY COMP (or none)	None†	None†	Numeric (9 S V P and insertion and replacement characters)	DISPLAY COMP (or none)	Numeric (9 S V P and insertion and replacement characters)	DISPLAY COMP (or none)
10	Coded binary integer	None	FIXED integer-1 integer-2 (where the integer value is 1 thru 18)	Binary integer	Numeric (9 S V P)	COMP-1 INDEX††	INTEGER LOGICAL BOOLEAN	INTEGER LOGICAL	Numeric (9 S V P and insertion and replacement characters)	COMP-1 INDEX†† LOGICAL	Numeric (9 S V P and insertion and replacement characters)	COMP-1 INTEGER LOGICAL
13	Coded floating-point normalized	None	FLOAT integer-1 (where the integer value is 1 thru 14)	Signed, normalized floating-point (1 word)	Numeric (9 S V P)	COMP-2	REAL BOOLEAN	REAL	Numeric (9 S V P and insertion and replacement characters)	COMP-2	Numeric (9 S V P and insertion and replacement characters)	COMP-2

TABLE H-1. DATA DEFINITION IN DMS-170 (Contd)

Data Class No.	Data Class Name	SCHEMA			COBOL Sub-Schema		FORTRAN 5 Sub-Schema Type Statement	FORTRAN Extended 4 Sub-Schema Type Statement	Query Update Sub-Schema in CDCS Data Base Access Mode		Query Update Sub-Schema in CRM Data Base Access Mode	
		PICTURE Clause	TYPE Clause	Internal Representation	PICTURE Clause	USAGE Clause			PICTURE Clause	USAGE Clause	PICTURE Clause	USAGE Clause
14	Coded double-precision	None	FLOAT integer-1 (where the integer value is 15 thru 29)	Signed, normalized floating-point (2 words)	None†	None†	DOUBLE PRECISION	DOUBLE PRECISION	Numeric (9 S V P and insertion and replacement characters)	DOUBLE	Numeric (9 S V P and insertion and replacement characters)	DOUBLE
15	Coded complex	None	COMPLEX	Floating-point with real part and imaginary part (2 words)	None†	None†	COMPLEX	COMPLEX	Numeric (9 S V P and insertion and replacement characters)	COMPLEX	Numeric (9 S V P and insertion and replacement characters)	COMPLEX

†No corresponding sub-schema type. Valid conversion is shown in the DDL3 volume 2 or FORTRAN Data Base Facility reference manual.
 ††No picture clause allowed.



This appendix contains programming practices recommended by CDC for users of the software described in this manual. When possible, application programs based on this software should be designed and coded in conformance with these recommendations.

Two forms of guidelines are given. The general guidelines minimize application program dependence on the specific characteristics of a hardware system. The feature use guidelines ensure the easiest migration of an application program to future hardware or software systems.

GENERAL GUIDELINES

Good programming techniques always include the following practices to avoid hardware dependency:

- Avoid programming with hardcoded constants. Manipulation of data should never depend on the occurrence of a type of data in a fixed multiple such as 6, 10, or 60.
- Do not manipulate data based on the binary representation of that data. Characters should be manipulated as characters, rather than as octal display-coded values or as 6-bit binary digits. Numbers should be manipulated as numeric data of a known type, rather than as binary patterns within a central memory word.
- Do not identify or classify information based on the location of a specific value within a specific set of central memory word bits.
- Avoid using COMPASS in application programs. COMPASS and other machine-dependent languages can complicate migration to future hardware or software systems. Migration is restricted by continued use of COMPASS for stand-alone programs, by COMPASS subroutines embedded in programs using higher-level languages, and by COMPASS owncode routines used with CDC standard products. COMPASS should only be used to create part or all of an application program when the function cannot be performed in a higher-level language or when execution efficiency is more important than any other consideration.

FEATURE USE GUIDELINES

The recommendations in the remainder of this appendix ensure the easiest migration of an application program for use on future hardware or software systems. These recommendations are based on known or anticipated changes in the hardware or software system, or comply with proposed new industry standards or proposed changes to existing industry standards.

ADVANCED ACCESS METHODS

The Advanced Access Methods (AAM) offer several features within which choices must be made. The following paragraphs indicate preferred usage.

Access Methods

The recommended access methods are indexed sequential (IS), direct access (DA), and Multiple-Index Processor (MIP).

Record Types

The recommended record types are either F for fixed length records, or W for variable length records. Record length for W records is indicated in the control word; the length must be supplied by the user in the RL FIT field on a put operation and is returned to the user in RL on a get operation.

BASIC ACCESS METHODS

The Basic Access Methods (BAM) offer several features within which choices must be made. The following paragraphs indicate preferred usage.

File Organizations

The recommended file organization is sequential (SQ). For files with word-addressable (WA) organization, use an accessing technique that can be easily modified to byte addresses.

Block Types

The recommended block type is C.

Record Types

The recommended record types are F for fixed length records and W for variable length records. For purely coded files that are to be listed, Z type records can be used.

Block Size

Set the maximum block length (MBL) to 640 characters for mass storage files and 5120 characters for tape files.

DMS-170

DMS-170 offers several features among which choices must be made. The following paragraphs indicate preferred usage of CDCS, DDL, and of Query Update in support of CDCS.

Multiple Record Descriptions

Do not include multiple record descriptions on a single file.

Repeating Groups

Avoid the use of the OCCURS clause, repeating groups, or arrays within records; as an alternative, the repeating data can be normalized into separate records on a different file. If repeating data must be used, limit usage to fixed length groups (no OCCURS DEPENDING ON clause) and to simple (unnested) OCCURS clauses.

Alternate Keys on Repeating Groups

Avoid the specification of alternate keys on repeating groups. The data can be normalized as indicated under Repeating Groups.

Collating Sequence

Use the default collating sequence or the ASCII collating sequence.

REDEFINES Clause

Use the REDEFINES clause only for alphanumeric-to-alphanumeric redefinitions, where the term alphanumeric has the meaning assigned by COBOL to data. In general, avoid the use of REDEFINES where use is based on a knowledge of the internal representation of data (floating-point layout, number of characters per word, and so forth).

Query Update Syntax

Use the new directives INVOKE, STORE, MODIFY, and REMOVE instead of the directives USE, INSERT, UPDATE, and DELETE.

INDEX

- Actual key file organization 1-4, 4-3
- Advanced Access Methods (AAM) 1-3
- Alphabetic data items 4-8
- Alphanumeric data items 4-9
- Alternate keys 4-6
- Area description entry
 - Definition 2-1, 4-1
 - Formats 4-2
- AREA-NAME clause 4-1

- Basic Access Methods (BAM) 1-3
- BLOCK CONTAINS clause 4-2
- BLOCK-TYPE clause 4-2
- Blocking 4-2

- CATALOG control statement 5-2
- Character set 3-2, A-1
- Clause
 - AREA-NAME 4-1
 - BLOCK CONTAINS 4-2
 - BLOCK-TYPE 4-2
 - DATA-PADDING 4-7
 - FILE-LIMIT 4-6
 - INDEX-BLOCK 4-7
 - INDEX-LEVEL 4-6
 - INDEX-PADDING 4-7
 - JOIN 4-14
 - LOG 4-3
 - NUMBER OF BLOCKS 4-7
 - OCCURS 4-11
 - ON CALL 4-7
 - ORGANIZATION 4-4
 - PICTURE 4-8
 - PROCEDURE LIBRARY 4-1
 - RECORD CONTAINS 4-3
 - RECORD-NAME 4-8
 - RECORD-TYPE 4-3
 - RELATION-NAME 4-14
 - RESTRICT 4-15
 - SEQUENCE 4-6
 - SUB-SCHEMA NAME 4-1
 - USAGE 4-13
- CLOSE condition 4-7, F-4
- Coding 3-2
- Comment lines 3-3
- COMPLEX items 4-13
- COMPUTATIONAL items 4-13
- Concurrency 1-3
- Continuation lines 3-3
- Control statements
 - DDL 5-1
 - NOS
 - DEFINE 5-2
 - Requirements 5-2
 - NOS/BE
 - CATALOG 5-2
 - REQUEST 5-2
 - Requirements 5-1
- CYBER Database Control System (CDCS) 1-1
- CYBER Record Manager (CRM)
 - Block types 4-3
 - Function 1-1
 - Record types 4-4

- Data administrator 1-1
- Data base
 - Creation 5-6
 - Definition 1-1
 - Recovery 1-4, G-3
 - Restoration 1-4, G-3
- Data base procedures
 - CLOSE 4-7, F-4
 - Definition 1-4
 - DISPLAY 4-7, F-4
 - MATCH 4-7, F-3
 - MISMATCH 4-7, F-4
 - OPEN 4-7, F-2
 - RETRIEVAL 4-7, F-4
 - SEARCH 4-7, F-2
 - Summary 4-7, F-2
 - UPDATE 4-7, F-4
- Data class specification 2-2, H-1
- Data description entry
 - Definition 2-1
 - Format 4-8
- Data Division
 - Definition 2-1
 - Format 4-1
- Data items
 - Alphabetic 4-8
 - Alphanumeric 4-9
 - COMPLEX 4-13
 - COMPUTATIONAL 4-13
 - DISPLAY 4-13
 - Fixed occurrence 2-2
 - INTEGER 4-13
 - LOGICAL 4-13
 - Nonrepeating 2-1
 - Numeric 4-8
 - Repeating 2-2
 - Variable occurrence 2-2
- Data size 2-2
- DATA-PADDING clause 4-7
- DDL
 - Character set 3-2
 - Coding 3-2
 - Control statement 5-1
 - Function 1-1
 - Language elements 3-1
 - Literals 3-1
 - Punctuation 3-2
 - Reserved words 3-1, D-1
 - Sequence numbers 3-2
 - Statements 3-2, 4-1, E-1
 - User-defined names 3-1
- Deck structures 5-2
- DEFINE control statement 5-2
- Diagnostics B-1
- Direct access file organization 1-4, 4-4
- DISPLAY condition 4-7, F-4
- DISPLAY items 4-13

- Divisions
 - Data 2-1, 4-1
 - Identification 2-1, 4-1
 - Relation 2-1, 4-13
- DMS-170 1-1
- Duplicate keys 4-5
- Editing 4-8
 - Elementary items
 - Definition 2-1
 - Vectors 2-2
 - Examples 6-1
 - Extended AAM files 1-4, 4-4
- Field length requirements 5-5
 - File organizations
 - Description 1-4
 - ORGANIZATION clause 4-4
 - FILE-LIMIT clause 4-6
 - FILLER 4-8
 - Fixed occurrence data items 2-2
- Group items
 - Definition 2-1
 - Nesting 2-1
 - Nonrepeating 2-1
 - Repeating 2-2
- Hashing 4-5
- Identification description entry 4-1
 - Identification Division
 - Definition 2-1
 - Format 4-1
 - Identifier 3-2
 - Index file 4-2
 - INDEX-BLOCK clause 4-7
 - INDEX-LEVEL clause 4-6
 - INDEX-PADDING clause 4-7
 - Indexed sequential file organization 1-4, 4-4
 - Initial AAM files 4-4
 - Insertion characters 4-9
 - INTEGER items 4-13
- JOIN clause 4-14
- Key field
 - Alternate 4-6
 - KEY-FIELD 4-5
 - Major alternate 4-6
 - Major primary 4-6
 - Primary 4-5
 - KEY-FIELD 4-5
- Level number assignments 2-1, 4-8
- Library, sub-schema (see Sub-schema library)
- Literals 3-1
- LOG clause 4-3

- Log file initialization G-1
- Log file structure G-1
- Logging G-1
- LOGICAL items 4-13
- Major alternate keys 4-6
- Major primary keys 4-6
- MATCH condition 4-7, F-3
- Migration guidelines 1-1
- MISMATCH condition 4-7, F-4
- Multiple-index processing 1-4, 4-6
- Multiple keys 4-6

- Nested group items 2-1
- Nonnumeric literals 3-1
- Nonrepeating data items 2-1
- NOS control statements 5-2
- NOS/BE control statements
 - CATALOG 5-2
 - REQUEST 5-2
- NUMBER OF BLOCKS clause 4-7
- Numeric data items
 - Assumed decimal point 4-9
 - Assumed decimal scaling 4-9
 - Definition 4-9
 - Operational sign 4-9
- Numeric literals 3-1

- OCCURS clause 4-11
- ON CALL clause 4-7
- OPEN condition 4-7, F-2
- ORGANIZATION clause 4-4

- PICTURE clause 4-8
- Primary keys 4-5
- PROCEDURE LIBRARY clause 4-1
- Punctuation 3-2

- Qualification 3-2
- Query Update description 1-3

- RECORD CONTAINS clause 4-3
- Record description entry
 - Definition 2-1
 - Format 4-8
- Record types, CYBER Record Manager (see CYBER Record Manager record types)

- RECORD-NAME clause 4-8
- RECORD-TYPE clause 4-3
- RECOVER utility G-3
- Relation description entry
 - Definition 2-1
 - Format 4-14
- Relation Division 2-1, 4-14
- RELATION-NAME clause 4-14
- Relational data base 1-3
- Repeating data items 2-2
- Repeating groups 2-2
- Replacement characters 4-10
- REQUEST control statement 5-2
- Reserved words 3-1, D-1
- RESTORE utility G-3
- RESTRICT clause 4-15
- RETRIEVAL condition 4-7, F-4

Schema definition 1-1
SEARCH condition 4-7, F-2
SEQUENCE clause 4-6
Sequence numbers 3-2
Sequential file organization 1-4, 4-3
Statements
 Components 3-2
 Formats 4-1, E-1
Sub-schema
 Compilation 5-2
 Definitions
 CDCS data base access mode 1-1
 CRM data base access mode 1-1
 Divisions 2-1
Sub-schema library 5-1

SUB-SCHEMA NAME clause 4-1
Subscripting 3-2

Temporary area 4-2

UPDATE condition 4-7, F-4
USAGE clause 4-13
User-defined names 3-1

Variable occurrence data items 2-2
Vectors 2-2

O

O

O

O

O

O

O

COMMENT SHEET

MANUAL TITLE: DDL Version 3 Reference Manual
Volume 3: Sub-Schema Definition for
CYBER Record Manager Use With:
Query Update

PUBLICATION NO.: 60482100

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

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

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

FOLD ON DOTTED LINES

TAPE

TAPE

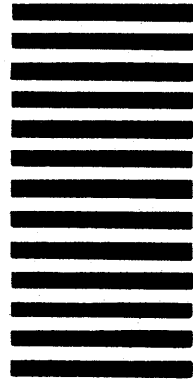
FOLD

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

TAPE

TAPE



CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.



CONTROL DATA CORPORATION

