

PRIME

Initial Documentation Release

**IDR 3044
PRIME COMPUTER
REFERENCE GUIDE
FOR DBMS SCHEMA DATA
DEFINITION LANGUAGE (DDL)**

First Printing July 1977

Copyright 1977 by
Prime Computer, Incorporated
145 Pennsylvania Avenue
Framingham, Massachusetts 01701

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
SECTION 1	INTRODUCTION	1-1
	ABOUT THIS MANUAL	1-1
	DATABASE DOCUMENTATION	1-1
	Documentation Releases	1-1
	OTHER RELATED PRIME DOCUMENTS	1-2
	INTRODUCTION TO DDL COMPILER	1-5
	Invoking The DDL Compiler	1-5
	Listing	1-5
	Compiler Errors	1-5
	Schema/DBMS Linkage	1-5
	SCHEMA ORGANIZATION	1-6
	NOTATION	1-7
	CHARACTER SET	1-7
	Character	1-7
	WORDS	1-8
	Reserved Words	1-8
	Names	1-8
	LITERALS	1-9
	Non-numeric Literals	1-9
	Numeric Literals	1-9
	COMMENTS	1-12
	PUNCTUATION	1-12
	DATA-BASE-DATA-NAMES	1-12
	LOCK-NAMES	1-13
	SYSTEM ENVIRONMENT	1-13
	DDL REFERENCES TO DML OPERATIONS	1-13
SECTION 2	THE SCHEMA ENTRY	2-1
	INTRODUCTION	2-1
	SCHEMA ENTRY	2-1
	Function	2-1
	Schema Entry Skeleton	2-1
	General Format of Entry	2-1

CONTENTS (Cont)

SCHEMA NAME	2-2
Function	2-2
General Format	2-2
Syntax Rules	2-2
General Rules	2-2
SCHEMA ENTRY-PRIVACY LOCK	2-3
Function	2-3
Syntax Rules	2-3
General Rules	2-3
SECTION 3 THE AREA ENTRY	3-1
INTRODUCTION	3-1
AREA ENTRY	3-1
Function	3-1
Area Entry Skeleton	3-1
General Format of Entry	3-1
AREA NAME	3-2
Function	3-2
General Format	3-2
Syntax Rules	3-2
General Rules	3-2
PRIVACY LOCK	3-3
Function	3-3
General Format	3-3
Syntax Rules	3-3
General Rules	3-3
SECTION 4 THE RECORD ENTRY	4-1
INTRODUCTION	4-1
RECORD ENTRY	4-2
Function	4-2
Record Entry Skeleton	4-2
Record Sub-entry Skeleton	4-2
Data Sub-entry Skeleton	4-2
General Format of Record Sub-entry	4-3
General Format of Data Sub-Entry	4-4

CONTENTS (Cont)

RECORD NAME	4-6
Function	4-6
General Format	4-6
Syntax Rules	4-6
General Rules	4-6
LOCATION MODE	4-7
Function	4-7
General Format	4-7
Syntax Rules	4-8
General Rules	4-8
Usage Considerations	4-9
WITHIN	4-10
Function	4-10
General Format	4-10
Syntax Rules	4-10
General Rules	4-10
PRIVACY LOCK	4-12
Function	4-12
Syntax Rules	4-13
General Rules	4-13
DATA-BASE-DATA-NAME	4-14
Function	4-14
General Format	4-14
Syntax Rules	4-14
General Rules	4-14
TYPE	4-15
Function	4-15
Syntax Rules	4-15
General Rules	4-15
PICTURE	4-18
Function	4-18
General Format	4-18
Syntax Rules	4-18
General Rules	4-18
OCCURS	4-21
Function	4-21
Syntax Rules	4-21
General Rules	4-22

CONTENTS (Cont)

CHECK	4-23
Function	4-23
Syntax Rules	4-23
General Rules	4-23
SECTION 5 THE SET ENTRY	5-1
INTRODUCTION	5-1
SET ENTRY	5-1
Function	5-1
Set Entry Skeleton	5-1
Set Sub-entry Skeleton	5-1
Member Sub-entry Skeleton	5-1
SET NAME	5-2
Function	5-2
General Format	5-2
Syntax Rules	5-2
General Rules	5-2
SET SUB ENTRY	5-3
Function	5-3
General Format of Set Sub-entry	5-3
General Format of MEMBER Sub-entry	5-4
ORDER	5-6
Function	5-6
Syntax Rules	5-6
General Rules	5-6
PRIVACY LOCK	5-9
Function	5-9
Syntax Rules	5-9
General Rules	5-9
OWNER	5-11
Function	5-11
Syntax Rules	5-11
General Rules	5-11
MEMBER	5-12
Function	5-12
Syntax Rules	5-12

CONTENTS (Cont)

General Rules	5-12
ASCENDING/DESCENDING	5-14
Function	5-14
Syntax Rules	5-14
General Rules	5-15
SEARCH KEY	5-16
Function	5-16
General Format	5-16
Syntax Rules	5-16
General Rules	5-16
SET OCCURRENCE SELECTION	5-18
Function	5-18
Syntax Rules	5-18
General Rules	5-20
SECTION 6 END STATEMENT	6-1
END SCHEMA	6-1
Function	6-1
General Format	6-1
General Rules	6-1

ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1-1	Database Documentation	1-3
1-2	Documentation Releases	1-4

TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
1-1	Reserved Words	1-11

FOREWORD

The DBMS Schema Reference Manual was written to aid the Database Administrator in determining what resources are available, how Prime's DBMS is structured and the methods by which a database can be created. Although the content assumes a level of knowledge, reference is made to appropriate books and documents. This Reference Guide serves to:

1. describe the Schema Compiler Syntax, and
2. provide diagnostic instructions.

DOCUMENTATION EXCELLENCE

Prime is striving to maintain the highest documentation standards. To achieve this goal, the Database documentation will be published in three documentation releases as described in Section 1. This is the Initial Documentation Release. Prime asks that each serious Database user correspond his comments about this manual concerning technical accuracy and additional information needed to implement the task of Database Administrator.

Robert E. Dawes, Technical Writer
Technical Publications Department
Prime Computer Inc.
145 Pennsylvania Avenue,
Framingham, Ma. 01701

ACKNOWLEDGEMENT

Prime Computer, Inc. wishes to formally acknowledge the work of the CODASYL Programming Language Committee (PLC) and the Data Description Language Committee (DDL). The Data Base Task Group (DBTG) of the PLC produced in April, 1971 a report containing the specifications of a standardized data base management facility consisting of a Data Description Language for describing a database, a Data Description Language for describing that part of the database known to a COBOL program, and a Data Manipulation Language for COBOL.

The Prime DBMS, portions of which are described in this manual, is based almost completely on the April 1971 DBTG specifications. Prime Computer is also participating in the ongoing work of CODASYL in the area of data base management through its membership on the DDL and on the Data Base Language Task Group (DBLTG) of PLC.

SECTION 1

INTRODUCTION

ABOUT THIS MANUAL

This manual contains the complete specifications of the Schema Data Description Language for writing a schema.

A schema written in the DDL consists of five types of ENTRIES which serve to:

1. Identify the schema (Schema Entry - Section 2).
2. Define areas (Area Entry - Section 3).
3. Define records (Record Entry - Section 4).
4. Define sets (Set Entry - Section 5).
5. Define end (END Entry - Section 6).

DATABASE DOCUMENTATION

Database documentation (Figure 1-1) is provided for both the Database Administrator and the application programmer. (The Database Administrator uses two manuals: 1) The PRIME Computer User's Guide for the Database Administrator and 2) PRIME DBMS SCHEMA DESCRIPTION LANGUAGE (DDL) Reference manuals.

The application programmer uses two manuals per application language: 1) the PRIME FORTRAN REFERENCE MANUAL FOR DBMS and the COMPANION PRIME FORTRAN USER'S Guide; 2) The COBOL Reference Manual for DBMS and the Companion PRIME COBOL USER'S Manual.

Documentation Releases

Prime provides three documentation releases (see Figure 1-2) for every new product: The Initial Documentation Release (IDR), the Preliminary Documentation Release (PDR), and the Final Documentation Release (FDR).

The Initial Documentation Release (IDR) provides whatever information is available without regard to whether the information is grammatically correct, or properly formatted. Thus, the intent is to provide usable information when needed.

The Preliminary Documentation Release (PDR) is the second draft by the writer. It provides more complete and accurate information about the product, but does not represent the final product information. Customers having the IDR should request the PDR from his sales representative.

The Final Documentation Release (FDR) is the complete product description up to the stated software revision number. This release is edited, formatted and presented in Prime's highest professional standards. Users will be notified when this release is available and should contact the local sales representative for a copy.

OTHER RELATED PRIME DOCUMENTS

PRIMOS FILE SYSTEM USER'S GUIDE	MAN2604
PRIMOS INTERACTIVE USER'S GUIDE	MAN2602
PRIMOS COMPUTER ROOM USER'S GUIDE	MAN2603
PROGRAM DEVELOPMENT SOFTWARE USER'S GUIDE	MAN1879
COBOL IV USER'S GUIDE	MAN3057
COBOL USER'S GUIDE	MAN2797

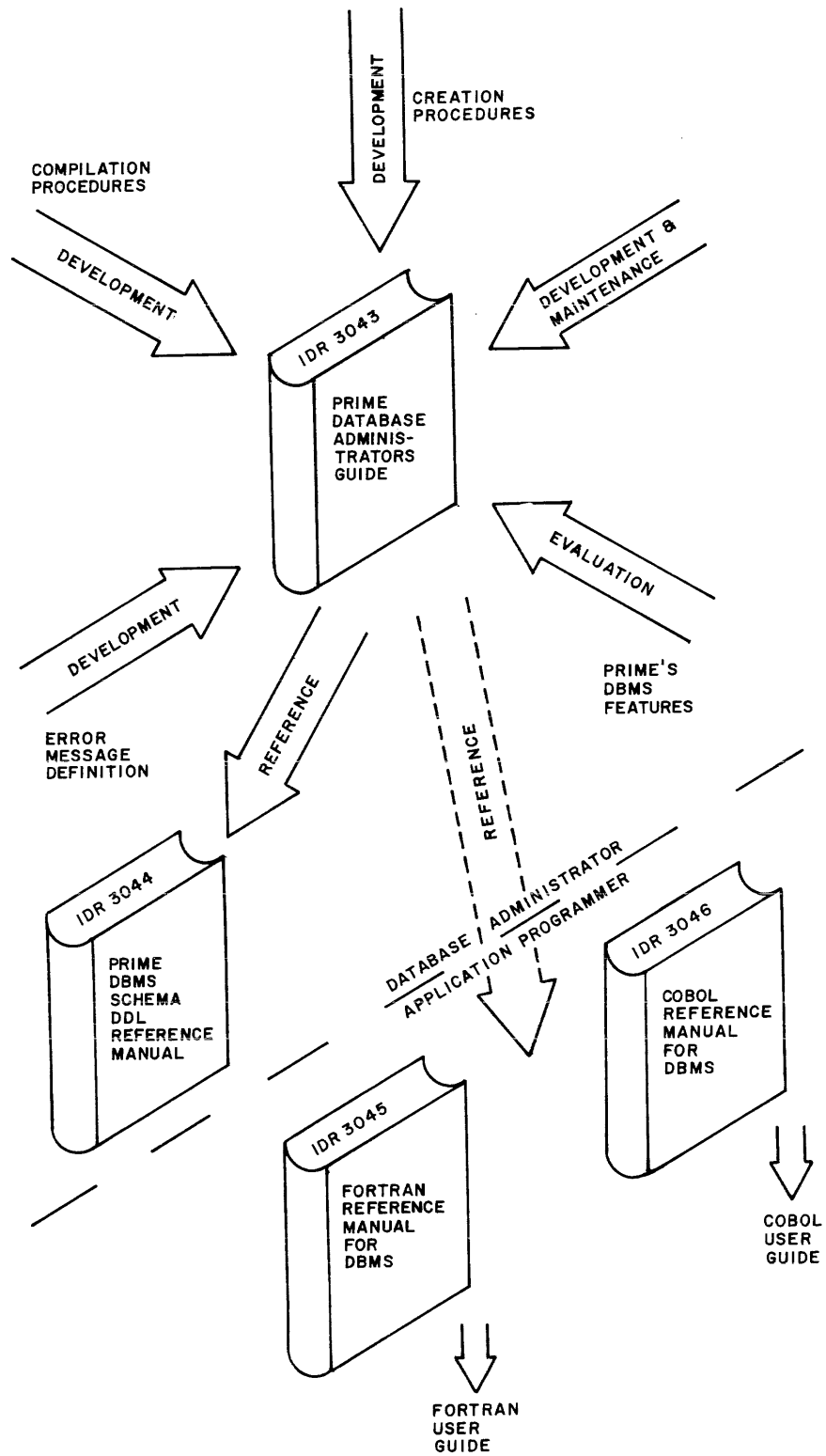


Figure 1-1 Database Documentation

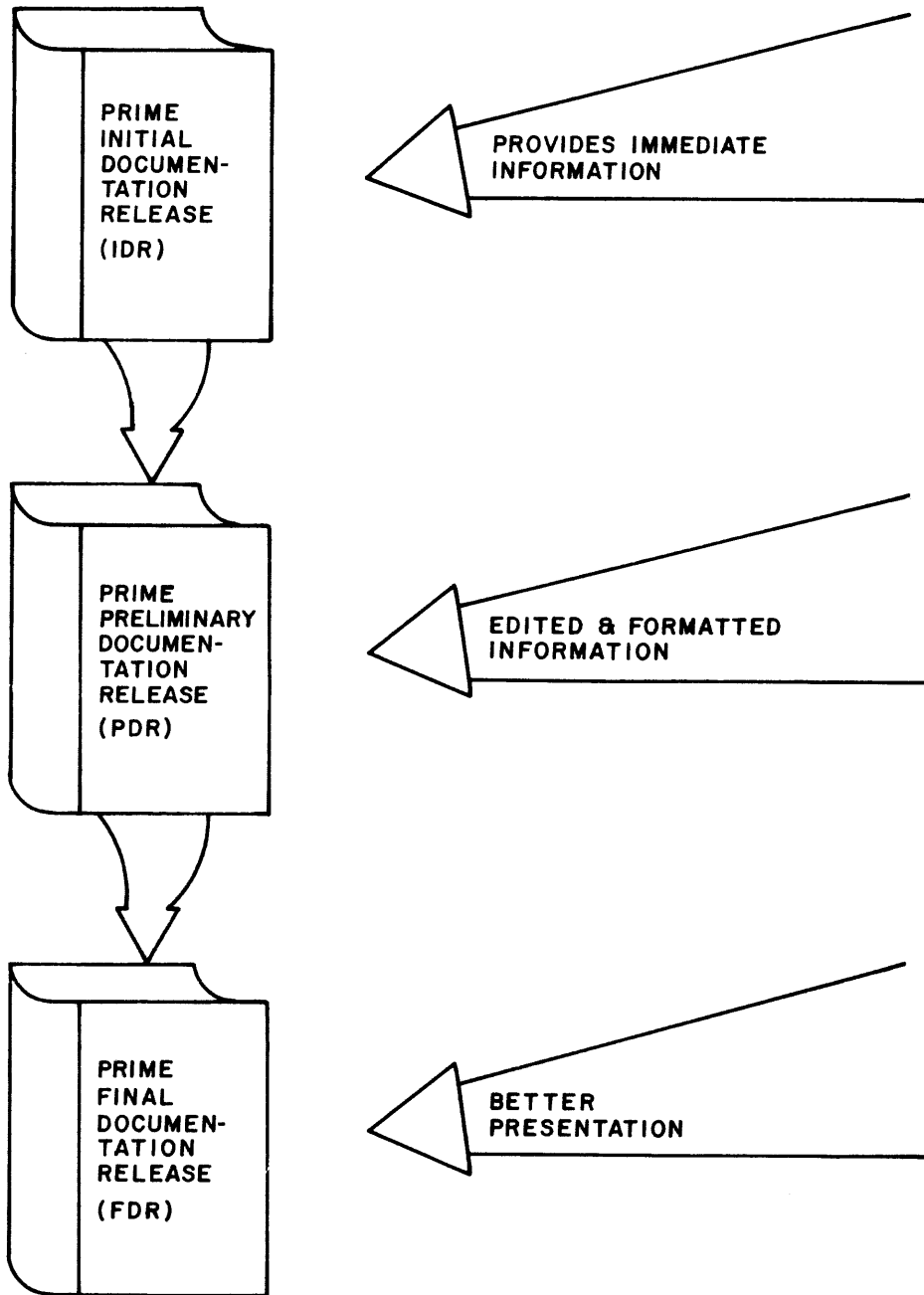


Figure 1-2. DOCUMENTATION RELEASES

INTRODUCTION TO DDL COMPILER

The schema DDL Compiler (SCHEMA) is a processor which translates the source schema data definition language into an internal tabular form called the schema table. SCHEMA also produces an output listing which includes the DDL text and error messages. The listing is found in the current UFD and is called "L<-name" where "name" is the input source file name.

Invoking The DDL Compiler

The Schema DDL Compiler is invoked with the command:

```
SCHEMA source-file [-VOL volume-name]
```

If volume-name for the output schema table is not specified, the volume of the schema directory (SCHDER) is assumed.

Listing

A listing is generated by the schema compiler which includes the input DDL and all error messages as well as compiler timings. The listing file name is L<-Name, where 'name' is the name of the input source file.

Compiler Errors

There are two classes of errors that may be encountered by the compiler.

1. Fatal - cannot parse further.
2. Nonfatal - can continue parsing for further errors after end of this clause, but final schema table will not be produced.

Error messages are displayed both on the user terminal with the line in which the error occurred, as well as in the output listing after the erroneous line.

Schema/DBMS Linkage

If the output schema table is completed, it is automatically entered in the DBMS schema directory and can be accessed by the Database Administrator Command Processor (DBACP), the Subschema Compilers, and the DML Command Processor.

SCHEMA ORGANIZATION

For each area, record type and set type in the schema, a separate entry is required. There must be only one Schema Entry in a schema. The following rules apply to the sequence of the various types of entries in a schema:

- o The Schema Entry must be the first entry.
- o An Area Entry must precede the Record Entry for all records in that area.
- o A Record Entry must precede the Set Entry for records included in that set.

An entry consists of one or more clauses which describe its attributes. In an entry describing a record or a set, clauses are grouped into sub-entries. Sub-entries may be repeated within an entry. Both entries and sub-entries are terminated by a period. Clauses may contain subordinate clauses.

The specifications for an entry consist of the following:

- o A narrative description of the function of the entry.
- o An Entry Skeleton representing the organization of the entry into clauses, or into sub-entries as applicable. Where the Entry Skeleton consists of sub-entries, a sub-entry skeleton is shown representing the organization of the sub-entry into clauses.
- o The general formats of the entry; that is the general format of each of the clauses which may be specified in the entry.
- o A separate description of each clause together with its subordinate clauses.

CLAUSES

The description of each clause consists of the following:

- o A narrative description of its function.
- o Its general format.
- o The syntax rules which apply.
- o The general rules which apply.

A general format is the arrangement of the elements which make up a clause. Inter-clause and intra-clause sequencing are governed by the sequence shown in the Entry Skeleton except when otherwise specified in syntax rules. A syntax rule amplifies or restricts the usage of the elements within a general format. A general rule amplifies or

restricts functions attributed to a general format or to its constituent elements.

NOTATION

The notation used in all formats and the rules which apply to all formats are:

- o The elements which make up a clause consist of upper-case words, lower-case words, special symbols, and special characters.
- o All underlined upper-case words are required when the format is used.
- o Upper-case words which are not underlined are optional words and need not be used.
- o Lower-case words are generic terms which must be replaced by appropriate names or values.
- o The meaning of enclosing a portion of a general format in special symbols as follows is:

a	at least no occurrences
b	
c	at most one occurrence

{ a }	at least one occurrence
{ b }	
{ c }	at most one occurrence

a	at least one occurrence of each
b	
c	at most one occurrence of each

An ellipsis (that is, '...') indicates repetition is allowed. The portion of the format which may be repeated is determined by the '[' or '{' which logically matches the ']' or '}' to the immediate left of the '...'.
'...'

CHARACTER SET

The character set for the DDL consists of the following characters:

Character

0,1, ...,9	digits
A,B, ...,Z	letters
	space
+	plus sign
-	minus sign or hyphen

,	comma
;	semi-colon
"	quotation mark
(left parenthesis
)	right parenthesis
/	stroke
*	asterisk

WORDS

A word is a sequence of not more than 30 characters. Each character is selected from the set 'A'... 'Z', '0'... '9', '-', '*', except that the '-' or '*' may not appear as the first or last character of a word.

Reserved Words

Reserved words are a list of words that may be used, but must not appear as user-defined words. The types of reserved words are described below.

- o A key word is a word that is required when the format in which the word appears is used. Within each format, such words are upper-case and underlined.
- o Within each format, upper-case words that are not underlined are called optional words and may appear at the user's option. The presence or absence of each optional word does not alter translation. Misspelling of an optional word, or its replacement by another word of any kind is not allowed.
- o Table 1-1 lists reserved words with their abbreviations enclosed in parentheses:

Names

A name is a word beginning with a letter. Lock-names have a maximum word length of 6 characters.

Types of names include:

- o Data-base-data-name
- o Record-name
- o Area-name
- o Set-name
- o Lock-name
- o Data-base-identifier
- o Schema-name

LITERALS

A literal is a string of characters, the value of which is implied by the order set of characters of which the literal is composed. Every literal belongs to one of two types, numeric and non-numeric.

Non-numeric Literals

A non-numeric literal is defined as a string of any allowable characters in the computer's character set, with a maximum length of 30 characters, delimited by quotation marks or apostrophe. This includes the quotation mark itself which, however, must be written twice consecutively for each of its occurrences within the string. The value of a non-numeric literal is the string of characters itself, excluding the delimiting quotation marks. Any spaces enclosed in the quotation marks are characters rather than separators; each such space is part of the non-numeric literal and is part of its value.

Numeric Literals

A numeric literal is defined as a string of characters chosen from the digits '0' through '9', the plus sign, the minus sign, the decimal point, and the character 'E'. Numeric literals may be expressed in two forms, integer and real. The rules for formation of numeric literals are as follows:

- o A literal must contain at least one digit.
- o An integer literal must not contain more than one sign character. If a sign is used, it must appear as the left-most character of the literal. If the literal is unsigned, the literal is positive.
- o An integer literal must not contain a decimal point or the character 'E'.
- o The value of an integer literal must be in the range
INTEGER * 2 : -32768 to + 32767.
- o A real literal consists of a string of digits with a decimal point and/or an exponent. The left-most characters of the literal may be a sign, and if it is omitted the literal is positive. The exponent consists of the letter E followed by a signed or unsigned integer that represents the power of ten by which the numeric value of the preceding digits is to be multiplied.
- o The value of a real literal must be in the range of REAL *4:
(10**-38 to 10**+38).
- o If a literal conforms to the rules for the formation of numeric literals, but is enclosed in quotation marks, it is a non-numeric literal and is treated as such by the compiler.

- o The value of a numeric literal is the algebraic decimal quantity represented by the characters in the numeric literal.

Table 1-1. Reserved Words.

ALIAS	FIND	ORDER
ALL	FIRST	OWNER
ALLOWED	FOR	PICTURE
ALTER	FROM	PRIOR
ALWAYS	GET	PRIVACY
ANY	IN	
ARE	INCLUSION	PROTECTED
AREA	INSERT	
AREA-ID	INTEGER	REAL
ASCENDING	INTEGER*2	REAL*4
AUTOMATIC	INTEGER*4	REAL*8
BIT	INTO	RECORD
BY	IS	REMOVE
CALC	KEY	REST
CALL	LAST	
CHARACTER	LINKED	RETRIEVAL
CHECK	LIST	SCHEMA
CLOSE	LOCATION	SEARCH
CODE	LOCK	SELECTION
	LOCKS	SELECTIVE
	MANDATORY	SET
CURRENCY	MANUAL	SETS
CURRENT	MEMBER	SORTED
DATABASE-KEY	MEMBERS	START
DATE	MODIFY	STORE
DECIMAL	NAME	STRING
	NEXT	SYSTEM
DELETE	NON-EXCLUSIVE	THRU
DESCENDING	NOT	TIME
DIRECT	OCCURRENCE	TIMES
DUPLICATE	OF	TYPE
DUPLICATES	ON	UPDATE
	OPEN	UPDATES
EXCLUSIVE	OPTIONAL	USAGE
	OR	USING
		VALUE
		VIA
		WITHIN

COMMENTS

Comments may be included for documentation purposes. They are delimited on the left by the contiguous special characters /* and on the right by */. Comments may appear wherever the blank may be used as a separator.

PUNCTUATION

The following punctuation characters are used:

- o One or more consecutive spaces, when not contained in a comment or delimited string, is a separator.
- o The comma is used as a separator.
- o The period is a delimiter for an entry or sub-entry.
- o A semi-colon is used to separate clauses within a sub-entry.

DATA-BASE-DATA-NAMES

A data-base-data-name which appears in the schema, but not in a data sub-entry of a record, constitutes a global data-base-data-name, i.e., it is not part of any record. Its type is determined by the clause in which it first appears (for example, in the DIRECT phrase of the LOCATION MODE clause) and it cannot appear in a second clause in which the type would not be consistent.

A data-base-data-name which appears in a data sub-entry of a record constitutes a data-base-identifier (sometimes referred to as a data item) and is part of the record in which it is defined. Where the same data-base-data-name is declared in a data sub-entry of more than one Record Entry, its use as a data-base-identifier may have to be qualified to achieve uniqueness. Syntax rules will specify when qualification is necessary. A name can be qualified even though it does not need qualification. The format of a data-base-identifier in the DDL for the schema is:

$$\text{data-base-data-name} \left[\left\{ \begin{array}{c} \text{OF} \\ \text{IN} \end{array} \right\} \text{record-name} \right]$$

LOCK-NAMES

A lock-name which appears in a PRIVACY LOCK clause defines an item which is not part of any record. When the schema is compiled, all variable locks have an undefined value. Setting the value of each variable lock is a function of the Administrator and can be done after schema compilation, through the Data Administrator DBACP. Any privacy key will match the contents of a lock whose value is undefined.

SYSTEM ENVIRONMENT

DBMS includes various capabilities to permit the Data Administrator to organize, monitor, reorganize, and restructure the database as it evolves with changing requirements. In addition the Data Administrator requires certain facilities to use, maintain, and develop each schema under personal control. The following operations exist for this purpose:

ALTER

ALTER permits the alteration of all the schema with the exception of the privacy locks clauses. This includes initializing and changing the various database files with the DBACP.

INCLUSION

INCLUSION, which is specified in the DDL for writing a sub-schema, causes entries specified in a schema to be included in a sub-schema.

DISPLAY

DISPLAY permits viewing of the schema, with the exception of the privacy locks, via the DBACP.

LOCKS

LOCKS allows the viewing, allocation, or changing of the values of variable locks, via the DBACP.

DDL REFERENCES TO DML OPERATIONS

In the specification of this DDL, it is assumed that certain operations may be performed on the described data. This results in various interactions between data descriptions and data operations. Where relevant these interactions are indicated. The assumed DML operations are of the following generic types:

OPEN: CLOSE

Areas of the database for (from) user processing.

INSERT: REMOVE

A record-occurrence into (or from) set occurrences.

STORE: DELETE

A record-occurrence into (or from) the database.

FIND:

Locate a specified record-occurrence in the database using a record-selection-expression (rse) which may be based on set membership.

GET:

Move data from a previously specified record to the User Work Area.

FETCH:

Equivalent of a FIND followed by a GET.

MODIFY:

Change data in a previously specified record.

MOVE:

Extract values from system locations maintained by DBMS. This operation frequently refers to the currency status indicators; these show which record occurrences (under various conditions) were last accessed.

SECTION 2

THE SCHEMA ENTRY

INTRODUCTION

This section defines the SCHEMA entry with the corresponding PRIVACY LOCK clauses.

```
*****
* SCHEMA ENTRY *
*****
```

Function

To identify a SCHEMA of a database.

Schema Entry Skeleton

SCHEMA clause

PRIVACY clause

General Format of Entry

SCHEMA NAME IS schema-name

$$[;\text{PRIVACY LOCK } [\text{FOR } \left\{ \begin{array}{l} \text{REST} \\ \text{LOCKS} \\ \text{DISPLAY} \\ \text{INCLUSION} \\ \text{ALTER} \end{array} \right\}] \text{ IS } \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \left[\text{OR } \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right] \dots] \dots$$

* SCHEMA ENTRY-SCHEMA NAME *

Function

To name a schema.

General Format

SCHEMA NAME IS schema-name.

Syntax Rules

1. The schema-name must be unique among the schema-names known to DEMS.

General Rules

1. The schema identified by the specified schema-name consists of the DDL entries that appear after the SCHEMA NAME clause and before an END entry.

 * SCHEMA ENTRY - PRIVACY LOCK *

Function

To specify the privacy lock(s) for certain operations which apply to the schema.

$$\text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \left\| \begin{array}{c} \text{REST} \\ \text{LOCKS} \\ \text{DISPLAY} \\ \text{INCLUSION} \\ \text{ALTER} \end{array} \right\| \right\} \text{IS} \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right] \dots$$

Syntax Rules

1. A separate PRIVACY clause may be stated for each restricted operation (LOCKS, DISPLAY, INCLUSION, ALTER). However, the same operation must not be specified in more than one PRIVACY clause.

General Rules

1. The literals and the content of the lock-names are privacy locks, to be matched with the pertinent privacy key. The content of lock-names can be established, changed, or cancelled using the appropriate support-functions available in the DBACP.
2. The same literal or lock-name may be specified for one or more operations.
3. If the optional FOR clause is omitted, all literals or lock-names apply to all operations.
4. The REST option implies all operations not yet specified in a PRIVACY clause for the schema.

5. Multiple privacy locks connected by OR are considered satisfied if any one is satisfied. The privacy locks are processed in the order listed until the outcome of the PRIVACY clause is known.
6. If no PRIVACY clause has been specified for an operation, then the use of that operation is without restriction.
7. The privacy locks associated with each restricted operation (ALTER, DISPLAY,...) must be satisfied in order to perform that restricted operation on the schema. In the case of the restricted INCLUSION operation, the privacy key is associated with each sub-schema to be developed from the schema to which the privacy lock applies. For the other restricted operations the key is associated with the pertinent support-function available in the DBACP.

SECTION 3

THE AREA ENTRY

INTRODUCTION

This section defines the AREA Entry with the corresponding AREA Clause and PRIVACY Clause.

```
*****
* AREA ENTRY *
*****
```

Function

To name and give certain characteristics of an area within a database.

Area Entry Skeleton

AREA Clause

PRIVACY Clause

General Format of Entry

AREA NAME IS area-name-1

$$\left[\text{;PRIVACY LOCK} \left[\text{FOR} \left[\begin{array}{l} \left[\begin{array}{l} \text{EXCLUSIVE} \\ \text{PROTECTED} \end{array} \right] \text{RETRIEVAL} \\ \left[\begin{array}{l} \text{EXCLUSIVE} \\ \text{PROTECTED} \end{array} \right] \text{UPDATE} \end{array} \right] \right] \right] \text{ IS} \\ \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \dots \right] \dots$$

* AREA NAME *

Function

To name an area within the database.

General Format

AREA NAME IS area-name

Syntax Rules

1. All area-names must be unique among area-names within the schema.
2. At least one area-name must be specified in a schema.

General Rules

1. The assignment of an area to a specific device; the specification of storage space required; the specification of page size; the ability to extend areas dynamically; and other similar functions are specifically delegated to support-functions available in the Data Administrator Module, to be used in conjunction with the DDL/DML.

 * PRIVACY LOCK *

Function

To specify the privacy locks which apply to the use of an area.

General Format

$$\left[\text{;PRIVACY LOCK} \left[\text{FOR} \left\| \left\| \begin{array}{l} \text{EXCLUSIVE} \\ \text{PROTECTED} \end{array} \right\| \begin{array}{l} \text{RETRIEVAL} \\ \text{UPDATE} \end{array} \right\| \right\| \right] \text{ IS} \right.$$

$$\left. \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right] \dots \right] \dots$$

Syntax Rules

1. A separate PRIVACY clause may be stated for each usage-mode (EXCLUSIVE or PROTECTED RETRIEVAL or UPDATE, or non-restricted RETRIEVAL or UPDATE). However, the same usage-mode must not be specified in more than one PRIVACY clause.

General Rules

1. The literals and the content of the lock-names are privacy locks, to be matched with the pertinent privacy key. The values of variable locks can be established, changed, or cancelled using the appropriate support-functions in the DBACP.
2. The same literal, lock-name, may be specified for one or more options included in this clause.
3. If the optional FOR clause is omitted, literals, or lock-names apply to any use of the area.

4. The REST option implies all usage-modes not yet specified in a PRIVACY clause for the area.
5. Multiple privacy locks connected by OR are considered satisfied if any one is satisfied. The privacy locks are processed in the order listed until the outcome of the PRIVACY clause is known.
6. If a PRIVACY clause has not been specified for a given usage-mode, then the use of that usage-mode on the area being described, is without restriction.
7. The privacy locks associated with the EXCLUSIVE RETRIEVAL, PROTECTED RETRIEVAL, RETRIEVAL, EXCLUSIVE UPDATE, PROTECTED UPDATE, and UPDATE clauses must be satisfied by a run-unit to enable it to open the area with the corresponding usage-mode. (See the FORTRAN Reference Manual for DBMS or the COBOL Reference Manual for DBMS, as appropriate.)

SECTION 4

THE RECORD ENTRY

INTRODUCTION

This section defines the RECORD entry and the corresponding clauses and statements.

* RECORD ENTRY *

Function

To name and give certain characteristics of record occurrences and their subordinate data items within a database.

Record Entry Skeleton

Record Sub-entry.

[Data Sub-entry.]...

Record Sub-entry Skeleton

RECORD clause

LOCATION clause

WITHIN clause

PRIVACY clause

Data Sub-entry Skeleton

data-base-data-name

TYPE clause

OCCURS clause

CHECK clause

PRIVACY clause

General Format of Record Sub-entry

RECORD NAME IS record-name-1

[;LOCATION MODE IS {
DIRECT { data-base-data-name-1
data-base-identifier-1 }
CALC USING data-base-identifier-2
[,data-base-identifier-3] ... DUPLICATES ARE [NOT] ALLOWED
VIA set-name-1 SET }]

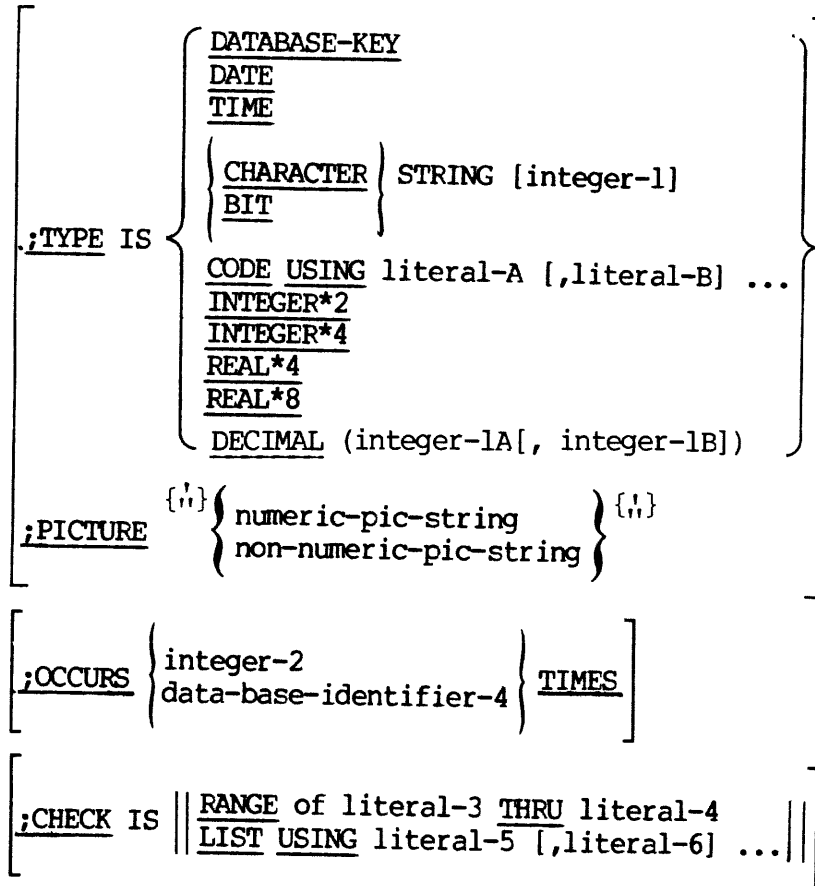
[;WITHIN area-name-1 [{ ,area-name-2 } ... AREA-ID IS data-base-data-name-2]]

[;PRIVACY LOCK {
{
INSERT
REMOVE
STORE
DELETE
DELETE MANDATORY
DELETE SELECTIVE
DELETE ALL
MODIFY
FIND
GET
REST
} } IS { literal-1
lock-name-1 }]

[OR { literal-2
lock-name-2 }] ...] ...

General Format of Data Sub-Entry

[level-number] data-base-data-name-3



$$\left[\text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \left\| \begin{array}{c} \text{REST} \\ \text{STORE} \\ \text{GET} \\ \text{MODIFY} \end{array} \right\| \right\} \right] \text{IS} \left\{ \begin{array}{l} \text{literal-7} \\ \text{lock-name-3} \end{array} \right\} \left[\text{OR} \left\{ \begin{array}{l} \text{literal-8} \\ \text{lock-name-4} \end{array} \right\} \right] \dots \right] \dots$$

RECORD NAME

Function

To name a record type in the schema; that is, to specify a generic name for all occurrences of the record type in the database.

General Format

RECORD NAME IS record-name-1

Syntax Rules

1. Record-names must be unique within the records of a schema.

General Rules

None.

LOCATION MODE

Function

To define the criteria for selecting a record occurrence and to advise DEMS and desired placement for a record occurrence in an area.

General Format

;LOCATION MODE IS	{	<table border="0"> <tr> <td style="vertical-align: middle;"><u>DIRECT</u></td> <td style="font-size: 2em; vertical-align: middle;">{</td> <td style="vertical-align: middle;">data-base-data-name-1 data-base-identifier-1</td> <td style="font-size: 2em; vertical-align: middle;">}</td> <td rowspan="3" style="font-size: 4em; vertical-align: middle;">}</td> </tr> <tr> <td style="vertical-align: middle;"><u>CALC USING</u></td> <td style="font-size: 2em; vertical-align: middle;">{</td> <td style="vertical-align: middle;">data-base-identifier-2 [,data-base-identifier-3] ... <u>DUPLICATES ARE [NOT] ALLOWED</u></td> <td style="font-size: 2em; vertical-align: middle;">}</td> </tr> <tr> <td style="vertical-align: middle;"><u>VIA</u></td> <td style="font-size: 2em; vertical-align: middle;">{</td> <td style="vertical-align: middle;">set-name-1 <u>SET</u></td> <td style="font-size: 2em; vertical-align: middle;">}</td> </tr> </table>	<u>DIRECT</u>	{	data-base-data-name-1 data-base-identifier-1	}	}	<u>CALC USING</u>	{	data-base-identifier-2 [,data-base-identifier-3] ... <u>DUPLICATES ARE [NOT] ALLOWED</u>	}	<u>VIA</u>	{	set-name-1 <u>SET</u>	}
<u>DIRECT</u>	{	data-base-data-name-1 data-base-identifier-1	}	}											
<u>CALC USING</u>	{	data-base-identifier-2 [,data-base-identifier-3] ... <u>DUPLICATES ARE [NOT] ALLOWED</u>	}												
<u>VIA</u>	{	set-name-1 <u>SET</u>	}												

Syntax Rules

1. By its appearance in a LOCATION MODE clause data-base-data-name-1 is implicitly declared as a data-base-key and is not part of a record.
2. Data-base-identifier-1 must be qualified with a record-name and must refer to a data item defined as a database-key.
3. Data-base-identifier-2, and data-base-identifier-3... must refer to data items included in the record being described.
4. Set-name-1 must be a set in which the record is defined as being a member. If set-name-1 is not a system owned set, then the member entry for the record must contain a SET OCCURRENCE SELECTION clause.

General Rules

1. If a LOCATION MODE IS DIRECT clause is specified then the data item specified in the clause must be initialized with a database-key prior to the execution of a command that selects the record on the basis of its LOCATION MODE clause. For a STORE command however, the occurrence number portion of the data item specified must be initialized either with a valid occurrence number or with a null-value.
2. If a LOCATION MODE IS CALC clause is specified, then data-base-identifier-2, data-base-identifier-3... must be initialized prior to the execution of a command that stores a record or selects it on the basis of its LOCATION MODE clause.
3. If a LOCATION MODE IS VIA set-name SET clause is specified, then all data items used to select a unique set occurrence as specified in the SET OCCURRENCE SELECTION clause for the set named must be initialized prior to the execution of a command that stores a record or selects it on the basis of its LOCATION MODE clause.
4. The LOCATION MODE clause controls placement of records according to the options selected. Specification of DIRECT or CALC causes placement to be controlled by data-base-data-name-1 or data-base-identifier-1, or by data-base identifier-2, data-base-identifier-3... respectively as constrained by the WITHIN clause. Specification of VIA set-name-1 causes placement to be as close as possible to the logical insert point of the record in set-name-1 as constrained by the WITHIN clause. If the optional LOCATION MODE clause is not used, ISIS controls placement within the constraints of the WITHIN clause.
5. The DUPLICATES clause, which must be stated if the LOCATION MODE IS CALC clause is used, refers to the CALC keys, that is, to the

input arguments (data-base-identifier 2...) of the CALC procedure.

6. If the DUPLICATES ARE NOT ALLOWED clause is specified , no additional occurrence of this record type will be allowed to exist if all of its CALC key values are identical to those of an occurrence already in the same area.

Usage Considerations

Please note that LOCATION MODE DIRECT has been deleted by CODASYL in current JOD specifications. If the PRIME DBMS is revised to comply with these specifications, this feature (i.e. LOCATION MODE DIRECT) will not be supported in future releases.

 * WITHIN *

Function

To define an area, or multiple areas, in which a record occurrence of the record type may be placed.

General Format

```
;WITHIN area-name-1 [{area-name-2}...AREA-ID IS data-base-data-name-2]
```

Syntax Rules

1. WITHIN clause must be specified if multiple AREAS have been defined in the AREA entry for the SCHEMA.
2. Area-name-1, area-name-2 must be the names of areas for which an area entry is included in the schema.
3. By its appearance in an AREA-ID clause, data-base-data-name-1 is implicitly defined to be a data-item which can contain a 30-character-string; i.e., an area name.
4. If the LOCATION MODE IS VIA set-name clause is specified in this Record Entry, the WITHIN clause for this Record Entry must include all areas named in the WITHIN clause of the OWNER record of the set named.

General Rules

1. When only area-name is specified, all record occurrences of the type being described will be placed in the named area (area-name-1).
2. When more than one area-name is specified, the contents of data-base-data-name-1 determine the area into which a record occurrence is placed.
3. Data-base-data-name-1 must be initialized with an appropriate area-name or with a NULL-VALUE prior to the execution of a command which stores a record or one which selects a record on the basis of its LOCATION MODE clause.
 - a. If the LOCATION MODE clause specifies DIRECT or CALC, data-base data-name-1 must be initialized with an area-name included in the WITHIN clause; otherwise an error will occur.

- b. If the LOCATION MODE clause specifies VIA set-name SET, data-base-data-name-1 must be initialized with an area-name included in the WITHIN clause or with a NULL-VALUE. If it is initialized with a legal area-name the record occurrence is placed in the named-area. If it is initialized with a NULL-VALUE the record occurrence is placed as close as possible, within the constraints of the areas named in its WITHIN clause, to its logical insert point in the set named.

 * PRIVACY LOCK *

Function

To specify the privacy locks which apply to the use of a record and the privacy locks which apply to the use of data items or data aggregates included in a record.

Format 1

$$\begin{array}{c}
 \text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \begin{array}{l} \text{REST} \\ \text{INSERT} \\ \text{REMOVE} \\ \text{STORE} \\ \text{DELETE} \\ \text{DELETE MANDATORY} \\ \text{DELETE SELECTIVE} \\ \text{DELETE ALL} \\ \text{GET} \\ \text{MODIFY} \\ \text{FIND} \end{array} \right\} \right] \text{ IS } \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \\
 \\
 \left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \dots \right]
 \end{array}$$

Format 2

$$\begin{array}{c}
 \text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \begin{array}{l} \text{REST} \\ \text{STORE} \\ \text{GET} \\ \text{MODIFY} \end{array} \right\} \right] \text{ IS } \left\{ \begin{array}{l} \text{literal-7} \\ \text{lock-name-3} \end{array} \right\} \left[\text{OR} \left\{ \begin{array}{l} \text{literal-8} \\ \text{lock-name-4} \end{array} \right\} \right] \dots
 \end{array}$$

Syntax Rules

1. A separate PRIVACY clause may be stated for each command included in each of the formats. However, the same command must not be specified in more than one PRIVACY clause. This latter rule applies to each of the two formats separately.
2. Format 1 applies to records, while Format 2 applies to data items or data aggregates.

General Rules

1. The literals and the content of the lock-names are privacy locks, to be matched with the pertinent privacy key. The values of variable locks can be established, changed, or cancelled using the appropriate support-functions available in the Data Administrator Module.
2. The same literal, or lock-name, may be specified for one or more of the commands included in this clause.
3. If the optional FOR clause is omitted, all literals, or lock-names apply to all commands included in the relevant format.
4. The REST option implies that all literals, or lock-names apply to all commands not yet specified in a PRIVACY clause in the data sub-entry or record entry in which this clause appears.
5. Multiple privacy locks connected by OR clauses are considered satisfied if any one is satisfied. The privacy locks are processed in the order listed until the outcome of the PRIVACY clause is known.
6. If a PRIVACY clause has not been specified for a command, then the use of that command on occurrences of the record or data item being described, is without restriction.
7. The privacy locks associated with the various commands (INSERT REMOVE ...) must be satisfied in order to execute the respective DML command on the record or data item to which the privacy lock applies.
8. A PRIVACY clause for a record which specifies a STORE, GET, or MODIFY command must be satisfied in order to execute the respective command on any of its components. A PRIVACY clause for a repeating-group which specifies a STORE, GET, or MODIFY command must be satisfied in order to execute the respective command on any of its components.

* DATA-BASE-DATA-NAME *

Function

To name a data item or data aggregate and indicate its structural level within the record.

General Format

[level-number] data-base-data-name-3

Syntax Rules

1. A data-base-data-name is a name that is not identical to any reserved word.
2. A data-base data-name must be unique within a record.
3. A level-number is an unsigned decimal integer greater than 0 and less than 8.

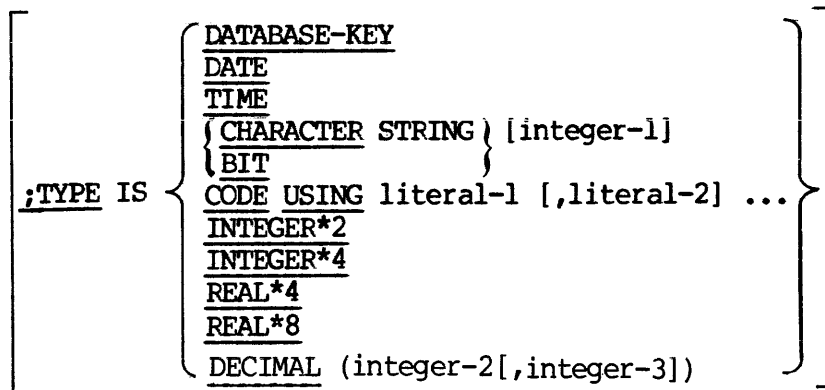
General Rules

1. A data-base-data-name followed by one or more clauses constitutes a data sub-entry. The content of a record is defined by a series of one or more data sub-entries.
2. A data sub-entry names and describes a data item, vector or repeating group. Additional sub-entries are required to name and describe the components of a repeating group.
3. If a level-number is not specified, level 1 is assumed.
4. The sub-entry for a data item must include a TYPE or a PICTURE clause. The sub-entry may also include a PRIVACY clause, and a CHECK clause.
5. To describe a vector, the sub-entry must include a TYPE or a PICTURE clause and an OCCURS clause. The sub-entry may also include a PRIVACY clause and a CHECK clause.
6. To describe a repeating group, the sub-entry may include an OCCURS clause and, a PRIVACY clause, neither a TYPE nor a PICTURE clause may be specified. If the OCCURS clause is omitted, "OCCURS 1 TIME" is assumed for the repeating group. (Also called a naming group.) The sub-entry must be followed by at least one sub-entry with a higher-valued level number (i.e., a subordinate sub-entry). A naming or repeating group is the only type of sub-entry that may be followed by subordinate sub-entries. Subordinate sub-entries may, in turn, describe repeating groups.

 * TYPE *

Function

To describe the characteristics of a data item.



Syntax Rules

1. Integer-1, integer-2, and integer-3 must be unsigned decimal constants with a value greater than zero.
2. Literal-1 and literal-2 must be nonnumeric literals.
3. For each item, either a PICTURE specification or a TYPE specification may be used but not both.

General Rules

1. The TYPE clause is used to define an arithmetic data item, a string item, a database-key 2 coded item, a date, or a time. A picture clause is used to define an arithmetic or string data item with implicit character checking.
2. INTEGER data are a precise representation of the range of binary integers from -2(15) to +2(15) if INTEGER*2 or INTEGER and range -(2(31)-1) to 2(31) if INTEGER*4. Integer data may only be assigned integral values within this range.
3. REAL data can be assigned approximations of real numbers with magnitudes within the range of -8,388,607 to +8,388,607 if REAL

or REAL*4 and +140,737,488,355,327 if REAL*8. A real datum may acquire positive or negative values within this range, or the value zero. Real data have an associated precision of 6+ or 14th significant digits. That is, the sixth or fourteenth most significant digit will be accurate, while the seventh or fifteenth will sometimes be accurate, depending on the value assigned to the datum.

4. CHARACTER is used to specify a character string; that is, one or more characters of the ASCII character set. BIT specifies a bit-string; that is, one or more binary digits.
5. Integer-1 is used to specify the length of a string data item; that is, the number of bits or characters in the string. If integer-1 is omitted for a character string, the string is assumed to be of variable length and terminated with trailing blanks. The trailing character may be redefined in the Data Manipulation Language. If integer-1 is omitted for a bit string, the string is assumed to be one bit long.
6. DATABASE-KEY defines a data item as 6 bytes in length designed to hold a database key. A database-key is a unique identifier of a record occurrence.
7. The characteristics of a data item as defined in the schema may differ from the characteristics of the data item as defined in a sub-schema. Where the characteristics differ, a conversion occurs from the schema defined characteristics to the sub-schema defined characteristics whenever a GET command is issued for the data item; and a conversion occurs from the sub-schema defined characteristics to the schema defined characteristics whenever the data item is involved in a STORE command or in a MODIFY command.
8. The conversion rules may vary, depending on the data item characteristics (i.e., TYPES) available in the host language of the sub-schema. The procedures applied when the schema characteristics differ from the sub-schema characteristics are described in the PRIME FORTRAN Reference Manual for DBMS (Section 2).
9. Whenever a conversion cannot occur (e.g., it would cause a loss of significant digits or significant characters), an Error Status condition results and the value in the database and the run-unit's User Work Area remains unchanged.
10. CODE defines a data item designed to hold one of a fixed list of string literals (i.e., multiple choice). The format is compact since the internal representation is a pointer (2 bytes) into a literal table rather than the entire string (1 byte/char.). The corresponding sub-schema data type may be either a character string, in which case the literal string is put in the User Work Area, or an integer representing the relative location of the

corresponding literal in the CODE USING clause (starting with 1).

11. DECIMAL specifies that the base of a numeric data item is decimal, and that the number is to be stored in packed format with 4 bits representing each decimal digit. Integer-2 specifies the total number of decimal digits and must be in the range 1 to 18. Integer-3 is the scale factor, which may be either negative or positive. A negative scale factor, $-n$, describes an integer, with the point assumed to be located n places to the right of the right-most actual digit. A positive scale factor, n , that is larger than the value of integer-2, describes a fraction with the point assumed to be located n places to the left of the right-most actual digit. If integer-3 is not specified, 0 is assumed.

 * PICTURE *

Function

The PICTURE clause describes the characteristics of elementary items as they appear in the User Work Area.

General Format

$$\text{;PICTURE } \left\{ \begin{array}{l} \{ \text{ } \} \\ \{ \text{ } \} \end{array} \right\} \left\{ \begin{array}{l} \text{non-numeric picture specification} \\ \text{numeric picture specification} \end{array} \right\} \{ \text{ } \}$$

Syntax Rules

1. A PICTURE clause can be specified only at the elementary item level.
2. A PICTURE specification consists of certain allowable combinations of characters as described in the general rules below enclosed in quotation marks or apostrophes.
3. The maximum number of symbols allowed in the PICTURE specification is 30.
4. The PICTURE clause describes fixed length data items.
5. A numeric PICTURE specification cannot contain the characters A or X, and a non-numeric picture specification must include at least one A or one X, plus 9's.
6. Either the PICTURE clause or a TYPE clause must be specified for every elementary item.

General Rules

1. There are two categories of data that can be described with a PICTURE clause: Numeric and Non-numeric.
2. To define an item as Numeric:
 - a. Its PICTURE character-string can only contain the symbols '9', 'P', 'S' and 'V'. The number of digit positions that

can be described in the PICTURE character-string must range from 1 to 18 inclusive; and

- b. If unsigned, its contents when represented in Standard Data Format must be a combination of the Arabic numerals '0', '1', '2', '3', '4', '5', '6', '7', '8', and '9'; if signed, the item may also contain a '+' or '-'.
3. To define an item as Non-numeric:
 - a. Its PICTURE character-string is restricted to certain combinations of the symbols 'A', 'X', and '9'.
 - b. Its contents when represented in Standard Data Format are allowable characters in the computer's character set.
 4. The size of an elementary item, where size means the number of character positions occupied by the elementary items in Standard Data Format, is determined by the number of allowable symbols that represent character positions. An unsigned nonzero integer which is enclosed in parentheses following the symbols 'A', 'X', '9', indicates the number of consecutive occurrences of the symbol. Note that the symbols 'S' and 'V' may appear only once in a given PICTURE.
 5. The function of the symbols used to describe an elementary item are explained as follows:
 - A Each 'A' in the character-string represents a character position which can contain only a letter of the Roman alphabet or a space.
 - P The 'P' indicates an assumed decimal scaling position and is used to specify the location of an assumed decimal point, when the point is not within the number that appears in the data item. The scaling position character 'P' is not counted in the size of the data item. Scaling position characters are counted in determining the maximum number of digit positions (18) in numeric items which appear as operands in arithmetic statements. The scaling position character 'P' can appear only to the left or right as a continuous string of 'P's within a PICTURE description; since the scaling position character 'P' implies an assumed decimal point (to the left of 'P', if 'P's are the left-most PICTURE characters and to the right of 'P's if 'P's are the right-most PICTURE characters), the assumed decimal point symbol 'V' is redundant as either the left-most or right-most character within such a PICTURE description.
 - S The letter 'S' is used in a character-string to indicate the presence, but neither the representation nor, necessarily, the position of an operational sign; it must be written as the left-most character in the PICTURE. The 'S' is not

counted in determining the size (in terms of Standard Data Format characters) of the elementary item.

- V The 'V' is used in a character-string to indicate the location of the assumed decimal point and may only appear once in a character-string. The 'V' does not represent a character position and therefore is not counted in the size of the elementary item. When the assumed decimal point is to the right of the right-most symbol in the string, the 'V' is redundant.
- X Each 'X' in the character-string is used to represent a character position which contains any allowable character from the computer's character set.
- 9 Each '9' in the character-string represents a character position which contains a decimal numeric digit and is counted in the size of the item.

The characteristics of a data item as defined in the schema may differ from the characteristics of the data item as defined in a sub-schema. When the characteristics differ, a conversion occurs from the schema defined characteristics to the sub-schema defined characteristics whenever a GET command is issued for the data item; and a conversion occurs from the sub-schema defined characteristics to the schema defined characteristics whenever the data item is involved in a STORE or MODIFY command.

 * OCCURS *

Function

To specify the number of times the data item or group occurs within a record.

$$\text{;OCCURS } \left\{ \begin{array}{l} \text{integer-2} \\ \text{data-base-identifier-4} \end{array} \right\} \text{TIMES}$$

Syntax Rules

1. The OCCURS clause cannot be used in the sub-entry as a RESULT clause.
2. The value of integer-2 must be greater than 0.
3. The data item referred to by data-base-identifier-4 must be previously defined as an integer in the same record as the subject data aggregate and cannot contain an OCCURS clause in its data sub-entry. Furthermore, data-base-identifier-4 must satisfy at least one of the following conditions:
 - o It must be defined that there is only one occurrence of the item in each occurrence of the record type.
 - o It must have a level-number exactly one greater than the level number of the innermost repeating group (the group with largest level-number) to which both data-base-identifier-4 and the data item being described are subordinate.

General Rules

1. The value of integer-2 or data-base-identifier-4 specifies the dimension of the data aggregate; that is, the number of occurrences of the data item or group.
2. The use of integer-2 indicates that the specified dimension is the same for all record occurrences.
3. The use of data-base-identifier-4 defines a data aggregate of variable dimension. If data-base-identifier-4 is defined to have only one occurrence in the record type, then for a given record occurrence, the dimension of the data aggregate is given by the value of data-base-identifier-4 in that record occurrence. If data-base-identifier-4 and the data aggregate are subordinate to a common repeating group, then for a given occurrence of the repeating group within a given record occurrence, the dimension of the data aggregate is given by the value of data-base-identifier-4 in that repeating group occurrence.
4. All values of data-base-identifier-4 must be positive or zero.

 * CHECK *

Function

To specify a validity-checking procedure to be executed whenever a value is changed or added to the database.

```

;CHECK IS || RANGE OF literal-3 THRU literal-4 ||
          || LIST USING literal-5 [,literal-6] ... ||
  
```

Syntax Rules

1. The value of literal-3 must be less than or equal to the value of literal-4. Literal-3 and literal-4 must be numeric and of the same type as the data item. Literal-5 and literal-6 must be strings.

General Rules

1. If RANGE is specified, the value of the data item is checked against the specified range. The value is valid if it is null or if it is greater than or equal to literal-3 and less than or equal to literal -4.
2. If LIST is specified, the data item must be equal to at least one of the literals specified in the clause. The data item must be type CHARACTER.
3. The specified validity checking occurs whenever a new value of the data item is placed in the database as a result of a STORE command or whenever a value of the data item is changed as a result of a MODIFY command. The check is performed after the value of the data has been converted to the 'type' specified for it in the schema definition.
4. If an invalid value is detected, an Error Status Condition will occur on the DELETE, GET, INSERT, MODIFY, REMOVE, or STORE command. Location ERROR-STATUS will contain the code associated with the first detected error.

SECTION 5

THE SET ENTRY

INTRODUCTION

This section defines the SET entry, the sub-entries and the clauses.

```
*****
* SET ENTRY *
*****
```

Function

To name and give certain characteristics of the set occurrences within a database.

Set Entry Skeleton

Set Sub-entry.

{Member sub-entry.}...

Set Sub-entry Skeleton.SET clause

ORDER clause

PRIVACY clause

OWNER clause

Member Sub-entry Skeleton.MEMBER clause

ASC/DES clause

SEARCH clause

SET OCCURRENCE SELECTION clause

* SET NAME *

Function

To name a set type in the schema, that is, to specify a generic name for all occurrences of the set type in the database.

General Format

SET NAME IS set-name-1.

Syntax Rules

1. The set-name must be unique among the set-names of the schema.

General Rules

None.

 * SET SUB ENTRY *

Function

To name the set occurrences within a database.

General Format of Set Sub Entry

SET NAME IS Set-name-1

Format 1

;ORDER IS $\left\{ \begin{array}{c} \underline{\text{FIRST}} \\ \underline{\text{LAST}} \\ \underline{\text{NEXT}} \\ \underline{\text{PRIOR}} \end{array} \right\}$

Format 2

;ORDER IS SORTED $\left[\begin{array}{l} \text{BY } \underline{\text{DATABASE-KEY}} \\ \text{ON COMMON KEY } \underline{\text{DUPLICATES}} \text{ ARE } \left[\begin{array}{c} \underline{\text{FIRST}} \\ \underline{\text{LAST}} \\ \underline{\text{NOT}} \end{array} \right] \text{ ALLOWED} \end{array} \right]$

$$\left[\text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \begin{array}{l} \text{REST} \\ \text{FIND} \\ \text{INSERT} \\ \text{REMOVE} \end{array} \right\} \right] \text{IS} \left\{ \begin{array}{l} \text{literal-1} \\ \text{lock-name-1} \end{array} \right\} \right]$$

$$\left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right] \dots \left. \vphantom{\left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right]} \right] \dots$$

$$\text{;OWNER IS} \left\{ \begin{array}{l} \text{record-name-1} \\ \text{SYSTEM} \end{array} \right\}$$

General Format of MEMBER Sub-entry

$$\text{MEMBER IS record-name-2} \left[\begin{array}{l} \text{MANDATORY} \\ \text{OPTIONAL} \end{array} \right] \left[\begin{array}{l} \text{AUTOMATIC} \\ \text{MANUAL} \end{array} \right] \left[\text{LINKED TO OWNER} \right]$$

$$\left[\left\{ \begin{array}{l} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \text{KEY IS data-base-identifier-3} \right]$$

$$\left[\text{DUPLICATES ARE} \left[\begin{array}{l} \text{FIRST} \\ \text{LAST} \\ \text{NOT} \end{array} \right] \text{ALLOWED} \right] \left[\text{,data-base-identifier-4} \right] \dots$$

$$\left[\text{;SEARCH KEY IS data-base-identifier-5} \left[\text{,data-base-identifier-6} \right] \dots \right]$$

$$\left[\text{DUPLICATES ARE} \left[\text{NOT} \right] \text{ALLOWED} \right] \dots$$

Format 1 of the SET OCCURRENCE SELECTION clause

```
[
;SET OCCURRENCE SELECTION IS THRU
]
{
CURRENT OF SET
}
{
LOCATION MODE OF OWNER
}
[
[ USING data-base-identifier-7 [,data-base-identifier-8] ...
{ ALIAS FOR data-base-identifier-9 IS data-base-data-name-1 } ... ]
]
]
```

Format 2 of the SET OCCURRENCE SELECTION clause

```
[
;SET OCCURRENCE SELECTION IS THRU set-name-2 USING
]
{
CURRENT OF SET
}
{
LOCATION MODE OF OWNER
}
[
[ ALIAS FOR data-base-identifier-10 IS data-base-data-name-2 ] ...
]
{
set-name-3 {
[ USING data-base-identifier-11 ,data-base-identifier-12 ...
{ ALIAS FOR data-base-identifier-13 IS
data-base-data-name-3 } ... ]
} ...
}
]
```

 * ORDER *

Function

To specify the insertion point of a member record occurrence within a set occurrence, and thereby define the order of sequential progression.

Format 1

;ORDER IS $\left(\begin{array}{c} \underline{\text{FIRST}} \\ \underline{\text{LAST}} \\ \underline{\text{NEXT}} \\ \underline{\text{PRIOR}} \end{array} \right)$

Format 2

;ORDER IS SORTED $\left[\begin{array}{l} \text{BY } \underline{\text{DATABASE-KEY}} \\ \text{ON COMMON KEY } \underline{\text{DUPLICATES}} \text{ ARE } \left[\begin{array}{c} \underline{\text{FIRST}} \\ \underline{\text{LAST}} \\ \underline{\text{NOT}} \end{array} \right] \text{ ALLOWED} \end{array} \right]$

Syntax Rules

1. If the ORDER IS SORTED clause is specified with the COMMON KEY option then, an ASCENDING/DESCENDING clause without a clause must be stated for each Member Sub-entry for this set with keys that are identical in item types.

General Rules

1. ORDER FIRST refers to the position within the set occurrence that immediately follows the OWNER record occurrence; this is a reversed chronological sequencing; the last member record inserted into the set occurrence becomes the first member of the set occurrence.

2. ORDER LAST refers to the position within the set occurrence that immediately precedes the owner record occurrence. This is a chronological sequencing; the newest member record occurrence becomes the last member in the set occurrence.
3. ORDER PRIOR/NEXT refers to insertion points relative to the current record of the set. Whenever the current record of a given set-name is not known to the run-unit, no new member records can be inserted into an occurrence of that set-name without first establishing a current record for it. If the current record of the set is the owner record, ORDER PRIOR is equivalent to ORDER LAST and ORDER NEXT is equivalent to ORDER FIRST.
4. The ORDER IS SORTED clause allows specification of a set order based on one of the following:
 - o record-types, ordered by the appearance of their member entries within the set entry. Additionally, data items of a given record type may be specified as minor sort keys within that record type.
 - o data-base-keys of member record occurrences.
 - o by common sort control data items specified for all member record types of the set.
5. If the ORDER IS SORTED is stated and the BY DATABASE-KEY or common key is not used, the order of the member entry definitions within the set entry is used as the major sort key. In other words, member record occurrences are grouped by record-type with all occurrences of the record-type whose member entry appears first in the set entry appearing first in the set occurrence. Minor sort keys, which specify how record occurrences are sorted within a record type, may be defined in ASCENDING/DESCENDING clause of the member entry for the record. If no such minor keys are defined for a particular member record, record occurrences within that record-type will be sorted by DATABASE-KEY.
6. The optional BY DATABASE-KEY clause specifies that the member records of a set occurrence are kept in ascending sequence by their database-key. If this option is specified, the ASCENDING/DESCENDING clause may not appear in any member entry of the set.
7. The optional COMMON KEY clause specifies the action to be taken when a new record occurrence is to be added to the set. The values of its sort control data items are duplicates of sort control data items of record occurrence. This currently participates as members of the set occurrence. Use of the common key clause also specifies that the member records in a set occurrence are to be maintained in a single sequence regardless of the number of different member record-types specified in the

SET ENTRY. The common sort key(s) are specified in the ASCENDING/DESCENDING clauses for each member record-type. 'Common sort keys' implies that:

- o There are the same number of data items specified in every member record's ASCENDING/DESCENDING clause.
- o The corresponding data items from each clause have identical data characteristics and match in terms of whether they are ascending or descending.

 * PRIVACY LOCK *

Function

To specify the privacy locks which apply to the use of a set.

$$\begin{array}{c}
 \text{;PRIVACY LOCK} \left[\text{FOR} \left\{ \begin{array}{l} \text{REST} \\ \text{FIND} \\ \text{INSERT} \\ \text{REMOVE} \end{array} \right\} \right] \text{IS} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \\
 \\
 \left[\text{OR} \left\{ \begin{array}{l} \text{literal-2} \\ \text{lock-name-2} \end{array} \right\} \right] \dots
 \end{array}$$

Syntax Rules

1. A separate PRIVACY clause may be stated for each command. (INSERT REMOVE, and/or FIND). However, the same command must not be specified in more than one PRIVACY clause.

General Rules

1. The literals and the content of the lock-names are privacy locks to be matched with the pertinent privacy key. Lock-names can be established, changed, or cancelled using the appropriate support-functions available in the Data Administrator Module.
2. The same literal, or lock-name, may be specified for one or more commands.
3. If the optional FOR clause is omitted, all literals, or lock-names apply to all commands included in the format.
4. The REST option implies that all, literals, or lock-names apply to all commands not yet specified in a PRIVACY clause in the set entry in which this clause appears.
5. Multiple privacy locks connected by OR clauses are considered satisfied if any one is satisfied. The privacy locks are processed in the order listed until the outcome of the PRIVACY clause is known.

6. If a PRIVACY clause has not been specified for a command, then the use of that command on occurrences of the set being described is without the restriction.
7. The privacy locks associated with the various commands (REMOVE INSERT ...) must be satisfied in order to execute the respective commands on any occurrence of the set being described. A PRIVACY LOCK FOR FIND clause applies to any record-selection-expression which includes the set- name in which this clause appears.

 * OWNER *

Function

To specify the name of a record, each occurrence of which establishes the existence of a set occurrence of the set named in this Set Entry.

$$;\underline{\text{OWNER}} \text{ IS } \left\{ \begin{array}{l} \text{record-name-1} \\ \underline{\text{SYSTEM}} \end{array} \right\}$$

Syntax Rules

1. Record-name-1 must be previously defined in a Record Entry.
2. If the OWNER IS SYSTEM clause is a Set Entry, none of its Member Sub-Entries can include a SET OCCURRENCE SELECTION clause.

General Rules

1. A record type may be specified as an owner in more than one Set Entry. It may also be defined as a member in one or more Set Entries.
2. The OWNER IS SYSTEM clause defines a singular set. A singular set has exactly one occurrence and no user specified owner record type.

 * MEMBER *

Function

To specify the name of a record, the occurrences of which may be members in set occurrences of the set named in this Set Entry.

To specify the type of membership in that set and optionally to check and reject the insertion within the same set occurrence of member record occurrences that have duplicate values for specified data items.

$$\underline{\text{MEMBER}} \text{ IS record-name-2 } \left[\begin{array}{c} \underline{\text{MANDATORY}} \\ \underline{\text{OPTIONAL}} \end{array} \right] \left[\begin{array}{c} \underline{\text{AUTOMATIC}} \\ \underline{\text{MANUAL}} \end{array} \right]$$

Syntax Rules

1. Record-name-2 must be previously defined in a Record Entry.
2. Data-base-identifier-1, data-base-identifier-2 ... must refer to data items included in record-name-2.

General Rules

1. If the optional word `AUTOMATIC` is used occurrences of record-name-2 are inserted unconditionally into the selected occurrence of the set at the time the record occurrence is stored. If `AUTOMATIC` or `MANUAL` is not specified, `MANUAL` is assumed.
2. If the word `OPTIONAL` is used, occurrences of record-name-2 may be inserted into or removed from the appropriate set occurrences by the execution of the `INSERT` and `REMOVE` commands respectively. If `OPTIONAL` or `MANDATORY` is not specified, `OPTIONAL` is assumed.
3. A `MEMBER` clause must be specified for each record type that can participate as a member in the set being described.

4. More than one record type can be declared as a member of any given set.
5. A record can be defined as a member in more than one set. It may also be defined as an owner in one or more sets.
6. Each occurrence of a member record participates in only one occurrence of a set. That is, within a set, each member record occurrence may have one and only one owner.
7. A record may not be defined as both an owner and a member of the sets such that a cycle is formed in which all records participate as automatic members in the sets included in the cycle.
8. Cycles are permitted, however, if one or more of the records involved is a manual member of a set included in the cycle. At least one of the manual members must in its Record Entry have a LOCATION MODE clause other than VIA a set involved in the cycle. This allows the manual member to be stored independently of its owner record. Each subsequent member record in the sequence of the cycle can be stored since its owner record will have been stored previously.

 * ASCENDING/DESCENDING *

Function

To specify the sort control keys for the member records of a sorted set.

To optionally check and reject the insertion within the same set of occurrence of member record occurrences that contain duplicate values for the specified sort control keys.

{ ASCENDING
 ; DESCENDING } KEY IS data-base-identifier-3
 [,data-base-identifier-4]...
 [DUPLICATES ARE [FIRST
 LAST
 NOT]]

Syntax Rules

1. Data-base-identifier-3, -base-identifier-4, ... must refer to data items specified in the Record Entry for the record named in the MEMBER clause of this sub-entry. They must not be RESULT data items and may not be variable length strings.
2. This clause must be included in all Member Sub-Entries of any Set-Entry which includes an ORDER IS SORTED clause specifying the optional DUPLICATES clause. This clause must not be used if the Set-Entry does not include the ORDER IS SORTED clause or if the ORDER IS BY DATABASE-KEY clause is used. If the Set-Entry includes the ORDER IS SORTED clause without additional qualifying phrases, specification of ASCENDING/DESCENDING clause is optional.
3. The DUPLICATES clause must be stated if and only if the ASCENDING/DESCENDING clause is used and an ORDER IS SORTED clause does not include a DUPLICATES clause (i.e. Sort on common key).

General Rules

1. The order in which the keys are specified will define the major to minor sequence for sorting.
2. Within a member record some data items can be defined as ascending keys and some can be defined as descending keys. That is, they can be intermixed.
3. If there are multiple member record types defined for a set, the ASCENDING/DESCENDING KEY clauses can be intermixed between record types. That is, record type can be defined with an ASCENDING KEY clause, while another can have a DESCENDING KEY clause.
4. IF ORDER IS SORTED clause is used in a Set Entry and no ASCENDING/DESCENDING clause is specified in one or more of its Member Sub-Entries, the database-key of such member records is used as an ascending key.
5. If the ORDER IS SORTED clause includes the (COMMON KEY) clause, the member records in a set occurrence are maintained in a single sequence regardless of the number of different member record-types. The ASCENDING/DESCENDING clause for each member record type must have the same number of sort control data items. The corresponding sort keys specified for each member record type must have identical data characteristics and must also match in terms of whether they are ascending or descending.
6. If the DUPLICATES ARE NOT ALLOWED clause is used, DBMS will reject the insertion into any given set occurrence of member record occurrences with duplicate values for the specified ascending/descending keys. This may occur during an attempt to store a new record occurrence in the database, to insert an existing record occurrence into a set, or to modify the value of a data item specified in an ASCENDING or DESCENDING KEY clause.
7. If the DUPLICATES ARE FIRST or the DUPLICATES ARE LAST clause is used, member record occurrences with duplicate values for the specified ascending/descending keys will be inserted before or after, as specified, any existing member occurrences with such duplicate values.
8. If the DUPLICATES clause does not include any of the optional words FIRST, LAST or NOT, the insertion point of 'duplicate' member occurrences relative to existing 'duplicates' is unpredictable.
9. If a data item specified as a KEY IS TYPE code, the sorted order will be determined by the unique internal coding corresponding to the appropriate literal in the CODE USING list.

* SEARCH KEY *

Function

To declare that for each set occurrence of a specific type, an index is required of all of its member records of a given type.

To specify the data items for which indexing is required.

To optionally check and reject the insertion within the same set occurrence of member record occurrences that contain duplicate values for the specified search keys.

General Format

;SEARCH KEY IS data-base-identifier-5[,data-base-identifier-6]...

DUPLICATES ARE [NOT] ALLOWED

Syntax Rules

1. Data-base-identifier-5, data-base-identifier-6...must refer to data items included in the record named in the MEMBER clause of this Sub-Entry. SEARCH key may not be variable length strings.

General Rules

1. A search key may appear as an argument in a SET OCCURRENCE SELECTION clause or in a record-selection-expression of a FIND command. Where such arguments have been declared with a SEARCH KEY clause, the indexing provided will be used to speed the required search.
2. The data items specified in one SEARCH KEY clause will be concatenated to form a single search argument. The SEARCH KEY clause must be repeated for each search argument for which indexing is to be provided.
3. If the DUPLICATES ARE NOT ALLOWED clause is used, the DBMS will reject the insertion into any given set occurrence of member record occurrences with duplicate values for the specified search keys. This may occur during an attempt to store a new record occurrence in the database, or to insert an existing record occurrence into a set or to modify the value of a data item to be a search key.

4. Where the DUPLICATES ARE ALLOWED clause is used, the record selected on the basis of an argument specified as a search key, will be the first record encountered which satisfies the argument. The order of the duplicates is unpredictable.

 * SET OCCURRENCE SELECTION *

Function

To define the rules governing the selection of the appropriate occurrence of a set for the purpose of inserting an occurrence of a member record, or accessing a desired member record.

Format 1

SET OCCURRENCE SELECTION IS THRU

{ CURRENT OF SET
 { LOCATION MODE OF OWNER

[USING data-base-identifier-7 ,data-base-identifier-8 ...
 { ALIAS FOR data-base-identifier-9 IS data-base-dasta-name-1}...] }

Format 2

SET OCCURRENCE SELECTION IS THRU set-name-2 USING

{ CURRENT OF SET
 { LOCATION MODE OF OWNER
 [ALIAS FOR data-base-identifier-10 IS data-base-data-name-2]... }
 { set-name-3 { USING data-base-identifier-11 ,data-base-identifier-12 ...
 { ALIAS FOR data-base-identifier-13 IS data-base-data-name-3}... } }

Syntax Rules

1. The SET OCCURRENCE SELECTION clause may not appear in any member entry of a singular set (i.e., OWNER IS SYSTEM was specified in the set entry).

2. The SELECTION clause must be specified for any member record-type whose LOCATION MODE is VIA set-name SET, where the set named is the set defined by this set entry. A singular set is an exception to this rule.
3. The SELECTION clause must be specified for any member record-type defined as an AUTOMATIC member of the set. A singular set is an exception to this rule.
4. In Format 1 a USING clause may qualify the LOCATION MODE OF OWNER clause only when the LOCATION MODE clause in the Record Entry for the owner record of the set in which this clause appears, is VIA set-name SET. When this is the case, either an ALIAS or a USING clause must be specified.
5. All data-base-identifiers must refer to declared data items of the owner record of the set(s) referenced.
6. Data-base-identifier-9 ... must, if the LOCATION MODE clause in the Record Entry for the OWNER record of the set is DIRECT or CALC, refer to data items specified in that LOCATION MODE clause. If, however, the LOCATION MODE clause is VIA set-name SET, data-base- identifier-9 ... must refer to data items specified in the USING clause of a SET OCCURRENCE SELECTION clause for another set with the same defined owner record type.
7. Data-base-identifier-10 ... must refer to data items specified in the LOCATION MODE clause in the Record Entry for the owner record of set-name-2.
8. By their appearance in an ALIAS clause, all data-base-data-names are implicitly defined as having the same characteristics as their data- base-identifiers, and are not part of any record. All data-base-identifiers appearing in an ALIAS clause must not be defined in a data sub-entry as a data aggregate or a variable length string.
9. Set-name-2, set-name-3 ... must form a continuous path in the sense that the owner of set-name-3 is a member of set-name-2..., with set-name- 2 as a start point or root. The same set-name cannot appear more than once. The last set named must be the subject of the set entry which this clause is part.
10. The LOCATION MODE OF OWNER option cannot be used if the owner being referenced does not have a LOCATION MODE clause specified for it in its Record Entry.
11. If the LOCATION MODE OF OWNER option is used the Record Entry for the owner record type being referenced must have a DUPLICATES NOT ALLOWED clause if its LOCATION MODE IS CALC.

12. All other data-base-identifiers explicitly named in a SET OCCURRENCE clause must have a DUPLICATES NOT ALLOWED clause in the appropriate Set Entries. The DUPLICATES NOT ALLOWED clause may be specified in the ASCENDING/DESCENDING KEY clause, the SEARCH KEY clause or the MEMBER clause of the relevant Set Entries.

General Rules

1. The SET OCCURRENCE SELECTION clauses for the appropriate member record and set combinations will govern the selection of specific set occurrences whenever:
 - a. Format 5 of the FIND command (without the optional word CURRENT) is executed.
 - b. A STORE command is executed and the object record is an automatic member of one or more sets.
 - c. A STORE command is executed and the LOCATION MODE of object record is VIA set-name SET.
 - d. A MODIFY command is executed which changes the value of a data item specified in a SET OCCURRENCE SELECTION clause.
2. Prior to the execution of any command involving set occurrence selection the data items specified in this clause must be initialized. The data items specified by this clause are those which are explicitly named in it and/or which are not explicitly named but which are implied by the LOCATION MODE option of this clause.
3. Format 1 applies where:
 - o the owner record of the set occurrence to be selected is either procedurally pre-selected (the CURRENT OF SET option) or can be uniquely identified on the basis of its LOCATION MODE clause alone (the LOCATION MODE OF OWNER option is used in this clause and the Record Entry for the owner record of the set specifies that its LOCATION MODE IS DIRECT or CALC).
 - o the owner record of the set occurrence to be selected cannot be determined except in terms of its membership in some other set and its associated SET OCCURRENCE SELECTION clause (the LOCATION MODE OF OWNER option is used in this clause and the Record Entry for the owner record of the set specifies that its LOCATION MODE IS VIA set-name SET).
4. Format 2 applies where the immediate owner record of the set occurrence to be selected cannot itself be uniquely selected except in terms of its membership in another set, and, the

criteria for selecting that set are included in the SET OCCURRENCE SELECTION clause. This condition may occur to an arbitrary number of levels forming a continuous path from a starting point of the specific owner of set-name-2 to the specific owner of the set-name of which the Format 2 SET OCCURRENCE SELECTION clause is a part. The owner of set-name-2 must either be procedurally pre-selected (the CURRENT OF SET option) or be uniquely identifiable on the basis of its LOCATION MODE clause alone (the LOCATION MODE OF OWNER option is used in this clause and the Record Entry for the owner of set-name-2 specifies that its LOCATION MODE IS DIRECT or CALC).

5. The CURRENT OF SET option causes the DBMS to select the current set occurrence as defined by the current of the appropriate set-name. In case of Format 1, this is the set-name of which this clause is a part. In the case of Format 2, it is set-name-2.
6. The LOCATION MODE OF OWNER option causes the DBMS to select a set occurrence on the basis of the LOCATION MODE clause specified in the Record Entry for the owner of appropriate set name. In the case of Format 1, this is the set name of which this clause is a part. For Format 2, it is the owner of set-name-2. Unless the ALIAS option is used in the SET OCCURRENCE SELECTION clause, or the LOCATION MODE clause in the Record Entry for the owner record is VIA set-name SET, the arguments used for selection of the owner record are the arguments specified in the LOCATION MODE clause of the Record Entry. These arguments must therefore be initialized with the actual argument values prior to each execution of any command controlled by the SET OCCURRENCE SELECTION clause.
7. In Format 1 if the LOCATION MODE OF OWNER option is used and the LOCATION MODE clause in the Record Entry for the owner record is VIA set-name SET, the owner record to be selected must be located in terms of its membership in another set. This selection is governed by the SET OCCURRENCE SELECTION clause for the set named in the LOCATION MODE clause of the owner record, and by the USING option of the SET OCCURRENCE SELECTION clause. The data items in the USING clause uniquely identify a specific record occurrence within an occurrence of the set named in the LOCATION MODE clause of the owner record. The SET OCCURRENCE SELECTION clause for the set named in the LOCATION MODE clause may, in turn, specify LOCATION MODE OF OWNER and the LOCATION MODE may again be VIA set-name SET. This condition may occur to an arbitrary number of levels, but must eventually terminate with a SET OCCURRENCE SELECTION clause that does not specify LOCATION MODE OF OWNER where the LOCATION MODE is VIA set-name SET. At each level other than the first, the arguments specified in a USING clause are used to select an owner record in its capacity as a member of another set. The arguments specified may be any items in the records to be selected.

8. In Format 2 specific occurrences of each of the sets named must be selected by DBMS. Set-name-2 is selected as described in general rule 6. Set-name-3 is selected on the basis of the initialized value for its argument, that is data-base-identifier-11. All subsequent sets named are also selected on the basis of the initialized values for their argument names. In each case the value for the argument name is used by the system to select the owner record of the named set in its capacity as a member of the previously named set. As with Format 1 the arguments specified may be any data items in the records to be selected. However, they uniquely identify the owner record occurrences to be selected.

9. The option ALIAS clause provides for the situation where a given record is defined as a member in more than one set type, and each type such set type has the same owner record type. In this situation, more than one argument value may be required for the data item named as an argument. The ALIAS clause provides the locations for such values.

SECTION 6

END STATEMENT

* END SCHEMA *

Function

To delineate the end of the schema DDL source text.

General FormatEND SCHEMAGeneral Rules

1. This statement must appear at the end of the schema DDL source text or an "EOF ON INPUT FILE" error will occur.

INDEX

<p>'9' 4-20</p> <p>'S' 4-19</p> <p>'V' 4-20</p> <p>'X' 4-20</p> <p>ALIAS CLAUSE 5-22</p> <p>ALTER 1-10</p> <p>ALTER 2-4</p> <p>AREA ENTRY 1-3</p> <p>AREA ENTRY 3-1</p> <p>AREA NAME 3-2</p> <p>AREA-ID 4-10</p> <p>ASCENDING KEY CLAUSE 5-15 5-14</p> <p>ASCENDING/DESCENDING CLAUSE 5-7</p> <p>ASCENDING/DESCENDING KEYS 5-15</p> <p>ASCENDING/DESCENDING 5-14</p> <p>ASCII CHARACTER SET 4-16</p> <p>AUTOMATIC 5-12</p> <p>BIT 4-17</p> <p>CALC CLAUSE 4-8</p> <p>CALC KEYS 4-8</p> <p>CALC PROCEDURE 4-9</p> <p>CHARACTER SET 1-5</p> <p>CHARACTER STRING 4-16</p> <p>CHECK CLAUSE 4-14</p> <p>CLAUSES 1-4</p> <p>CLOSE 1-11</p>	<p>CODE USING 4-17</p> <p>COMMON KEY OPTION 5-7</p> <p>COMMON SORT CONTROL DATA 5-8</p> <p>COMMON SORT KEY(S) 5-8</p> <p>COMPILER ERRORS 1-3</p> <p>DATA SUB-ENTRY SKELETON 4-2</p> <p>DATA SUB-ENTRY 4-14</p> <p>DATA-BASE KEY 4-16</p> <p>DATA-BASE-DATA-NAMES 1-9</p> <p>DATA-BASE-DATA-NAME 4-14</p> <p>DATA-BASE-KEY 4-8</p> <p>DATABASE DOCUMENTATION 1-1</p> <p>DATABASE-KEY CLAUSE 5-7</p> <p>DBACP 2-4</p> <p>DDL COMPILER 1-1</p> <p>DDL REFERENCES TO DML OPERATIONS 1-10</p> <p>DELETE 1-11</p> <p>DESCENDING KEY CLAUSE 5-15</p> <p>DISPLAY 1-10</p> <p>DISPLAY 2-4</p> <p>DUPLICATES ARE FIRST CLAUSE 5-15</p> <p>DUPLICATES ARE LAST CLAUSE 5-15</p> <p>DUPLICATES ARE NOT ALLOWED CLAUSE 5-20</p> <p>DUPLICATES ARE NOT ALLOWED CLAUSE 5-15</p> <p>DUPLICATES CLAUSE 4-8</p>
--	--

INDEX

DUPLICATES CLAUSE	5-15	INTEGER*4	4-15
ELLIPSIS	1-5	INTEGER-1	4-16
END STATEMENT	6-1	INTERNAL TABULAR FORM	1-1
ENTRY SKELETON	1-4	INTRODUCTION TO DDL COMPILER	1-1
EOF ON INPUT FILE ERROR	6-1	INVOKING THE DDL COMPILER	1-3
ERROR MESSAGES	1-3	LEVEL-NUMBER	4-14
EXCLUSIVE RETRIEVAL	3-4	LIST	4-23
EXCLUSIVE UPDATE	3-4	LITERAL	1-7
EXCLUSIVE	3-4	LITERAL	3-4
FETCH	1-11	LITERALS	1-6
FIND CLAUSE	5-10	LOCATION MODE CLAUSE	4-8
FIND COMMAND	5-20	LOCATION MODE CLAUSE	5-13
FIND	1-11	LOCATION MODE CLAUSE	4-10
FIND	5-9	LOCATION MODE	1-9
FIRST ENTRY	1-3	LOCATION MODE	4-7
GENERAL FORMAT	1-5	LOCATION MODE	4-8
GENERIC TYPES	1-10	LOCATION MODE	5-20
GET COMMAND	4-16	LOCK-NAME	2-3
INCLUSION	1-11	LOCK-NAME	3-3
INCLUSION	2-4	LOCK-NAME	5-9
INDEXING	5-16	LOCK-NAMES	1-6
INSERT REMOVE	5-9	LOCK-NAMES	1-9
INSERT	1-10	LOCK-NAMES	4-13
INSERT	5-12	LOCKS	1-10
INTEGER DATA	4-16	LOCKS	2-3
INTEGER LITERAL	1-7	LOWER CASE WORDS	1-5
INTEGER*2	4-15		

INDEX

MANDATORY	5-12	OPEN	1-11
MANUAL	5-12	OPTIONAL	5-12
MEMBER CLAUSE	5-14	ORDER FIRST	5-7
MEMBER CLAUSE	5-14	ORDER IS SORTED CLAUSE	5-7
MEMBER CLAUSE	5-15	ORDER IS SORTED CLAUSE	5-15
MEMBER RECORD OCCURRENCES	5-13	ORDER LAST	5-7
MEMBER RECORD OCCURRENCES	5-14	ORDER NEXT	5-7
MEMBER SUB-ENTRY SKELETON	5-1	ORDER PRIOR/NEXT	5-7
MEMBER	5-12	ORDER	5-6
MODIFY COMMAND	4-16	OUTPUT LISTING	1-1
MODIFY COMMAND	5-20	OWNER RECORD OCCURRENCE	5-7
MODIFY	1-11	OWNER RECORD	4-10
MOVE	1-11	OWNER	5-11
MULTIPLE PRIVACY LOCKS	5-9	PICTURE CLAUSE	4-19
MULTIPLE PRIVACY LOCKS	2-5	PICTURE SPECIFICATION	4-18
MULTIPLE PRIVACY LOCKS	4-13	PRIVACY CLAUSE	2-4
NAMES	1-6	PRIVACY CLAUSE	3-3
NON-NUMERIC LITERALS	1-6	PRIVACY CLAUSE	4-14
NON-NUMERIC	4-19	PRIVACY CLAUSE	5-9
NUMERIC LITERALS	1-7	PRIVACY KEY	1-9
NUMERIC PICTURE	4-18	PRIVACY LOCK CLAUSE	1-9
NUMERIC	4-18	PRIVACY LOCK	3-4
OCCURRENCE OF THE RECORD TYPE	4-21	PRIVACY LOCK(S)	2-4
OCCURRENCE OF A MEMBER RECORD	5-20	PRIVACY LOCKS	3-5
OCCURS CLAUSE	4-14	PRIVACY LOCKS	4-12
OCCURS CLAUSE	4-21	PRIVACY LOCKS	5-9
		PROTECTED RETRIEVAL	3-4

INDEX

PROTECTED UPDATE	3-4	'P'	4-19
PROTECTED	3-4	SCHEMA CLAUSE	2-1
RANGE	3-23	SCHEMA ENTRY	1-3
REAL DATA	4-16	SCHEMA ORGANIZATION	1-3
REAL LITERAL	1-7	SCHEMA TABLE	1-1
RECORD ENTRY SKELETON	4-2	SCHEMA-NAME	2-3
RECORD ENTRY	1-3	SCHEMA/DBMS LINKAGE	1-3
RECORD ENTRY	4-1	SEARCH KEY CLAUSE	5-20
RECORD ENTRY	4-10	SEARCH KEY	5-16
RECORD ENTRY	4-13	SET ENTRY SKELETON	5-1
RECORD NAME	4-6	SET ENTRY	1-3
RECORD OCCURRENCE	4-7	SET ENTRY	5-1
RECORD OCCURRENCE	1-10	SET ENTRY	5-12
RECORD OCCURRENCE	5-7	SET NAME	5-2
RECORD SUB-ENTRY SKELETON	4-2	SET OCCURRENCE SELECTION CLAUSE	5-18
RECORD TYPE	1-3	SET OCCURRENCE SELECTION	5-20
RECORD TYPES	5-7	SET OCCURRENCE SELECTION CLAUSE	4-8
REMOVE	1-11	SET OCCURRENCE	5-7
REMOVE	5-12	SET OCCURRENCES	1-10
REPEATING GROUP	4-14	SET OCCURRENCES	5-16
RESERVED WORDS	1-6	SET SUB-ENTRY SKELETON	5-1
REST OPTION	3-4	SET SUBENTRY	5-3
REST OPTION	5-9	SET TYPE	1-3
RESULT CLAUSE	4-21	STORE COMMAND	4-16
RETRIEVAL	3-4	STORE COMMAND	5-20
RUN/UNIT	5-7	STORE	1-11
SCALING POSITION CHARACTER			

INDEX

SUPPORT-FUNCTIONS 3-4
SYSTEM ENVIRONMENT 1-10
TRAILING CHARACTER 4-16
TYPE CLAUSE 4-16
TYPE CLAUSE 4-18
TYPE SPECIFICATION 4-15
TYPE 4-15
TYPES OF ENTRIES 1-1
UPDATE CLAUSES 3-4
UPDATE 3-4
USAGE-MODE 3-4
USAGE-MODES 3-5
USER WORK AREA 1-11
USER WORK AREA 4-18
USING CLAUSE 5-21
VALIDITY-CHECKING
PROCEDURE 4-23
VARIABLE LENGTH STRINGS 5-14
VARIABLE LENGTH 4-16
VARIABLE LOCKS 3-4
WITHIN 4-10