4.4
SYSTEM 2000/80

4.4 SYSTEM 2000/80

4.4.0 GENERAL DESCRIPTION

SYSTEM 2000/80 is the product designation for a generalized DBMS sold by INTEL Systems Corporation, P.O. Box 9968, Austin, Texas 78766. This system is an evolved and expanded descendant of the SYSTEM 2000 DBMS developed by MRI Systems Corporation, a DBMS company which became a division of INTEL in 1979.

SYSTEM 2000 may be described as a multiple-indexed (inverted list) DBMS which processes hierarchically structured data bases with full inversion of user-selected data items. System capabilities include calculated (CALC) location mode, the ability to define and access network structures, and relational data access (by use of a WHERE clause). System architecture is basically divided into the "Programming Facility" and the "Self-Contained Facility" (which consists of the Data Base Definition, Data Base Administration, and End-User Facilities).

In addition to data base definition, security and privacy controls are available in SYSTEM 2000 through the Integrated Data Dictionary--Basic Data Dictionary (BDD) and optional Extended Data Dictionary (XDD). Data Base Definition in SYSTEM 2000 involves the use of the BDD DEFINE language and a DEFINE processor to describe and revise the data base structure. CONTROL can also be used for such operational functions as assigning passwords, saving and restoring data bases, and controlling Rollback. The "End-User Facility" processes three languages: QUEUE, REPORT, and ACCESS (including the Query/Update Extension, QUEST). These SYSTEM 2000 languages provide several methods of access to the data base for different applications.

All languages in the Self-Contained Facility are designed for natural language similarity and simplicity, enabling end-users to access the data base without programming skills. For more conventional programming applications, SYSTEM 2000 offers the Programming Facility with its "Programming Language Extension" (PLEX). PLEX permits the user to operate in a COBOL, FORTRAN, PL/1 or ALC host language environment and provides the means for manipulating and querying SYSTEM 2000 data bases within the context of conventional programming languages.

Various add-on features are available (some at extra cost) with SYSTEM 2000 to satisfy the individual user processing needs. Included among these options are the Report Writer Feature, a Generalized Communications Interface to several teleprocessing monitors, a function-driven, screen-oriented query/update facility QueX), and the multiple thread capability. It should be noted that some options may not be available for some host machinery. See Figure 4.4.2 for a list of module availability by machine. Figure 4.4.1 is a price list of System 2000/80 products and features.

4.4.0.1 System Characteristics

While SYSTEM 2000, as described above, is fundamentally hierarchical in structure, it is designed to support other logical data base structures as well. In order to model a relational data base the WHERE clause allows the user to relate any field in any record to any other field in the data base. For creation of network structures the LINK command in PLEX and the Query/Update by Example establish relationships between any record type in any data base and a record type in the same or another data base.

PAID-UP LICENSE PRICES*

	CDC	UNIVAC 1100	UNIVAC 1100/60	DOS/VS(E)	OS/VS	IBM ≠4300 VM/CMS	IBM =4300 VM/CMS
BASIC PRODUCT (Data Base Manager, BDD,QUEUE, 1 PLEX)	70,000	70,000	45,000	49,500	60,000.	60,000	49,500
Report Writer	20,000	20,000	10,000	15,000	20,000	20,000	15,000
QUEST	30,000	30,000	15,000	**	30,000	30,000	**
Additional PLEX	15,000	15,000	10,000	9,000	10,000	10,000	9,000
Multi-Thread		30,000	20,000	20,000	25,000	25,000	20,000
Genius	10,000	10,000	10,000		10,000	10,000	10,000
QueX		25,000	20,000		15,000		
CICS Interface				10,000	10,000		
TAPS/80				15,000	25,000		
XDD				10,000	15,000	15,000	10,000
Multiple Systems Coupling					80,000		
Data Base Assist Processor (requires FAST-3805					8,500		
Installation Fee	2,500	2,500	2,500	2,500	2,500	2,500	2,500

**In Basic Product

*Prices are quoted in U.S. dollars and are subject to change rental and maintenance prices are also available

+IBM Basic products also include some or all of the following Multi-User, Accounting Log, Recovery, QUEST, additional PLEXs, Universal Software Interface

Figure 4.4.1/SYSTEM 2000/80 Products

SYSTEM 2000

MODULE AVAILABILITY

MODULE	MVT	VS	DOS	VM	CDC	UNIVAC	
BASIC S2K	X	X	X	X	X	X	
QUEST	X	X	X	X	X	X	
REPORT WRITER	X	X	X	X	X	X	
GENIUS	X	X	NA	X	X	X	
GRAPHICS INTERFACE	X	X	NA	X	X	X	
PLEX: COBOL	X	X	X	X	X	X	
PLEX: FORTRAN	X	X	X	X	X	X	
PLEX: ALC	X	X	X	X	NA	NA	
PLEX: PL/I	X	X	X	X	NA	NA	
ACCOUNTING LOG	X	X	X	X	NA	NA	
RECOVERY	X	X	X	X	NA	NA	
MULTI-THREAD	X	X	X	X	NA	X	(2.80)
EXTENDED DATA DICTIONARY	X	X	X	X	NA	NA	
QUEX (REQUIRES CICS)	NA	X	NA	NA	NA	X	
TAPS/80	NA	X	X	NA	NA	NA	
CICS INTERFACE	X	X	X	NA	NA	NA	
DBAP	NA	X	NA	NA	NA	NA	
MULTIPLE SYSTEMS COUPLING	NA	X	NA	NA	NA	NA	
UNIVERSAL SOFTWARE INTERFACE	X	X	X	X	NA	X	

X = AVAILABLE
NA = NOT AVAILABLE

Figure 4.4.2/SYSTEM 2000/80 Products

Records can be located via logical indices (see "schema" discussion below) without accessing the data records themselves. They can also be directly accessed based on a calculation (CALC location mode). User restrictions are imposed at definition time on the number of records that can be located using the CALC access and on the number of items that can be indexed. By providing faster, more direct data access, these features can significantly improve system efficiency in highly interactive environments.

A SYSTEM 2000 data base is defined as the "schema" and the total collection of all "data trees." The schema describes acceptable data and hierarchical relationships. Data trees are valid data organized to conform with the schema.

A data tree may be viewed as a tree structure with one trunk and several levels of branching. Figure 4.4.4 illustrates a data base with two data trees--one for Jones and one for Brown. "Records" correspond to each of the boxes (trunk or branches) in Figures 4.4.3 and 4.4.4. Each record is comprised of defined data fields call "items." In SYSTEM 2000 records and items are considered to be "components" of the data base and are assigned user-specified names and numbers. (The third component, "stored commands," will be discussed subsequently.) Thus, in Figure 4.4.3, NAME, ADDR, and CITY are examples of items within the ROOT record of the schema. In Figure 4.4.4, the JONES and BROWN data records are data tree occurrences of the ROOT record. Similarly, PRESS OPERATOR and LATHE OPERATOR are data tree occurrences of the SKILL record belonging to JONES. BROWN would be the root for another data tree.

A structural terminology would be useful in understanding relationships within the hierarchy:

1) Every record will be a member of a "family" of related records. (See Figure 4.4.5.)
2) Each record except the root record has one and only one "parent" occurring one level above it.
3) An "ancestor" will occur at each level higher, including the parent, except the root record which has no parent.
4) All records which trace their ancestry to a common record can be considered "descendants" of that common record.
5) "Siblings" are records that share a common parent.

4.4.0.2 Defining the Data Base

In the Data Dictionary Facility the DEFINE language is used to describe the data base schema to the system. Figure 4.4.6 demonstrates an example of DEFINE module syntax suitable for describing the schema pictured in Figure 4.4.3 Components 4, 8, 11 and 13 are all identified as records; no explicit identification of the root, or "entry record, " will be necessary. Components 1, 2, and 3 are the items of the entry record, while the IN syntax for 6 indicates that 4 is its parent. Numbers describing components are used for identification purposes only and need not be ordered. Also, the NON-KEY specification denotes that no index will be created for that item. KEY, the default, need not be unique even at the entry level, so it would be possible to employ two "E.Z. Jones" within the same company.

Ten possible data types are allowed in SYSTEM 2000. They are: CHARACTER, TEXT, DATE, INTEGER, DECIMAL, MONEY, REAL (single precision), DOUBLE PRECISION, BINARY, and OCTAL/HEXADECIMAL. Field lengths can be chosen for all data types except DATE, with an overflow available for alpha-numeric CHARACTER and TEXT.

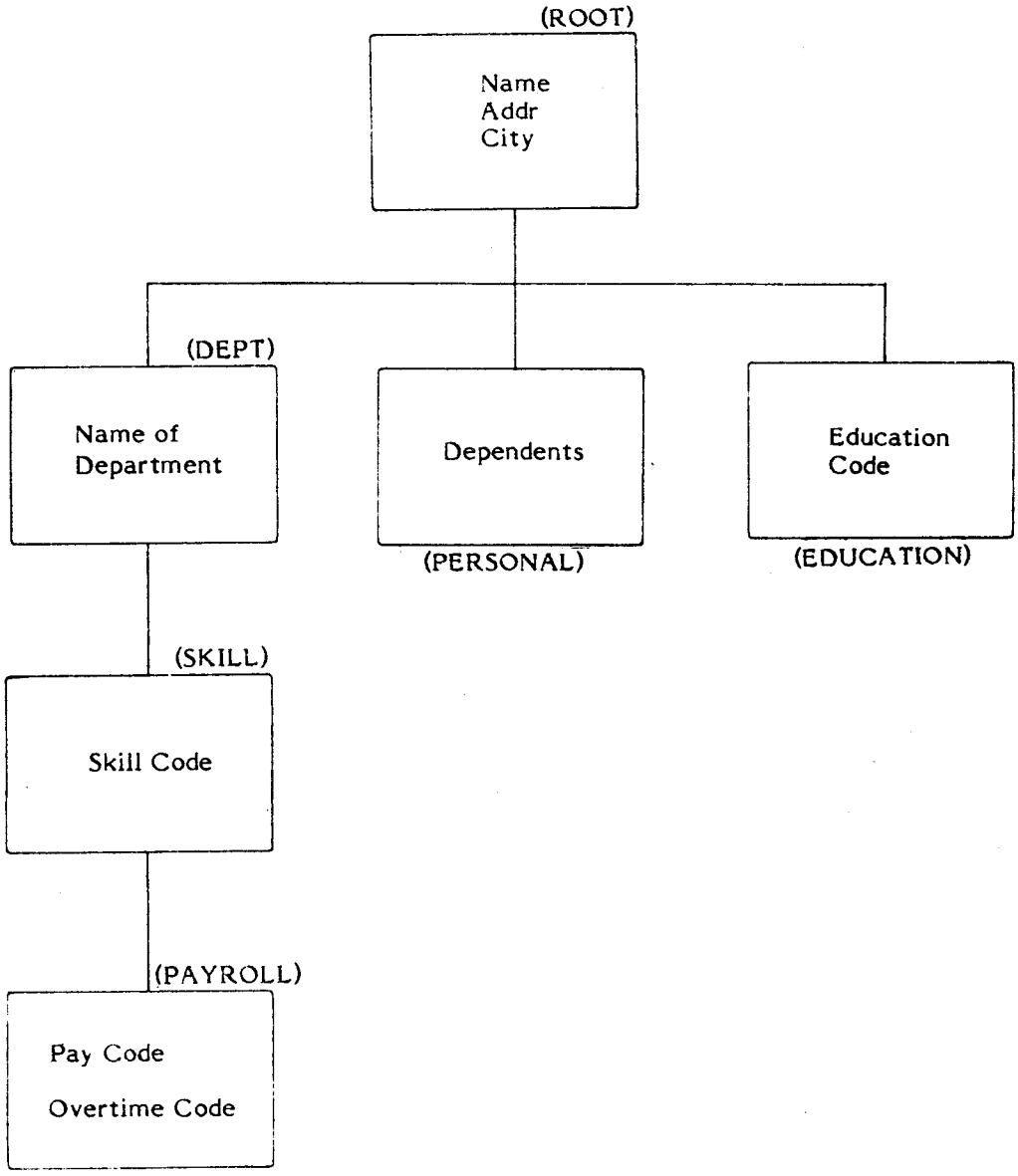


Figure 4.4.3/SCHEMA

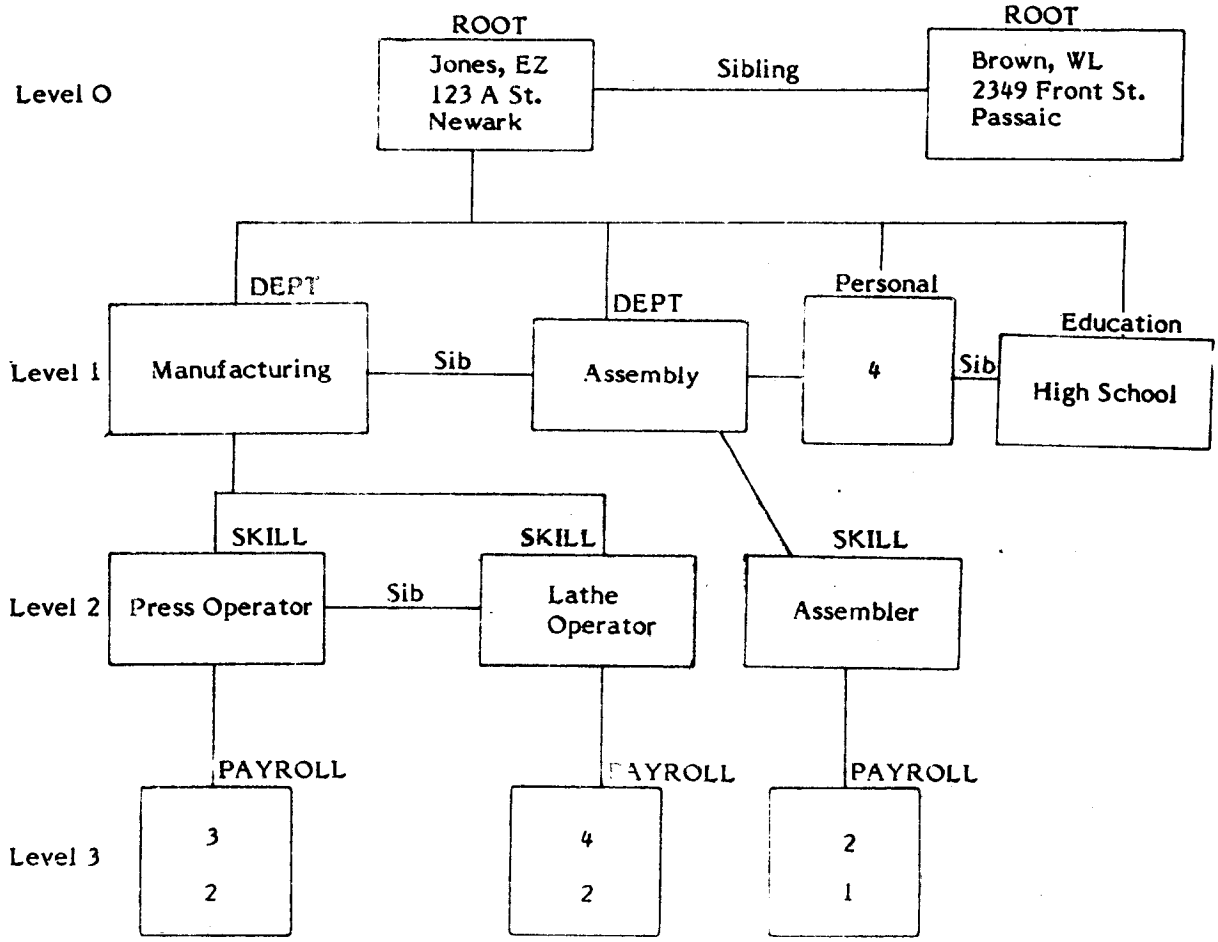


Figure 4.4.4/Data Trees

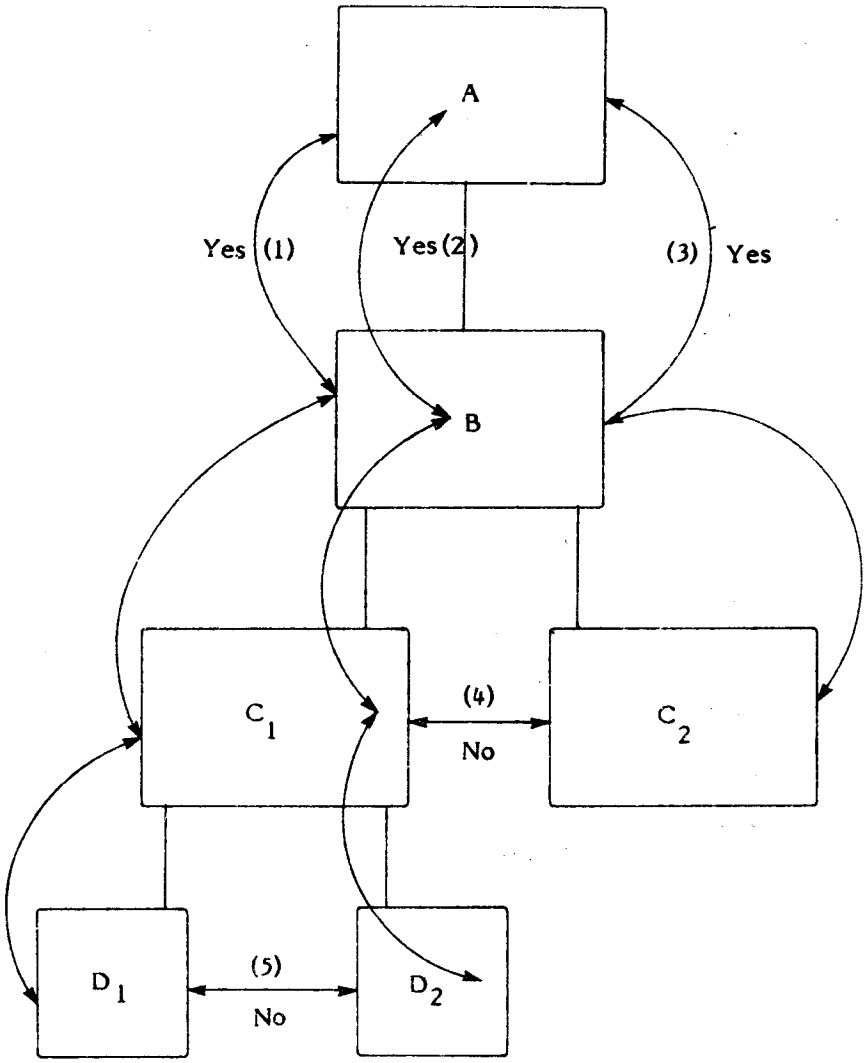


Figure 4.4.5/A Hierarchical "Family"

```
LEVEL

      ┌1*   NAME      (TEXT X (25));
  0   │2*   ADDRESS (NON-KEY CHARACTER X (12);
      └3*   CITY      NAME X 12));

      ┌4*   DEPT      RECORD
  1   │
      │     5*  NAME OF DEPT (CHARACTER X (12) IN 4);
      │
      │   2 ┌6*  SKILL     RECORD IN 4);
      │     │
      │     │     7*  SKILL CODE (CHARACTER X (5) IN 6);
      │     │
      │     │   3 ┌8*  PAYROLL RECORD IN 6);
      │     │     │
      │     │     │     9*  PAYCODE(CHARACTER XX IN 8);
      │     │     │
      │     │     │     10* OVERTIME CODE (CHARACTER XX N 8);

      ┌11*  PERSONAL    (RECORD)
  1   │
      │     12* DEPENDENTS  (NON-KEY INT 99 IN 11);

      ┌13*  EDUCATION   (RECORD);
  1   │
      │     14* EDUCATION CODE (TEXT XXX IN 13);
```

Figure 4.4.6/Typical Data Base Definition: SYSTEM 2000

The user may also specify an ARRAY for any of the ten types.

This summary has outlined the hierarchical structure, types of facilities, and logical relationships within SYSTEM 2000, citing also the user options for defining the data base. Distinctions will be made in subsequent sections between the Self-Contained Facility and the Programming Language Extension, as well as distinctions between the batch and teleprocessing modes.

4.4.1 DATA MANIPULATION

The Self-Contained Facility (SCF) and the Programming Language Extension (PLEX) contain the data manipulation languages for populating, retrieving, and modifying data bases. QUEUE, REPORT, ACCESS (including QUEST), and Query/Update By Example (QueX) comprise the SCF languages. These languages can be executed in two ways: a batch-oriented system, which grants the user access to the SYSTEM 2000 data bases one job at a time; or a conversational mode, executed interactively, which provides the user with on-line access to the data base files. The PLEX feature allows the user access through the facilities of the host language in either batch or transaction-oriented environments.

SCF and PLEX utilize two access techniques--queued and immediate--that correspond to modules in SCF. QUEST is used for processing individual commands, and QUEUE is used to group commands into one access to the data base files. In PLEX, however, there are no separate modules but rather a capability for queued processing built into the syntax of the language itself. The following paragraphs address the differences and similarities among the modules. Unless otherwise noted, the information described will be pertinent to most configurations.

4.4.1.1 Data Entry, Update and Deletion

The system provides full entry, update, and delete facilities for each of its configurations. QUEST provides seven basic commands--three for retrieval and four for updating. QUEUE offers the same capabilities with essentially the same set of commands. In PLEX there are four basic retrieval commands and four update commands. These modules and their data manipulation languages are compared in Figure 4.4.7. These modules differ primarily in language format and data presentation methods, both discussed in following sections. QueX provides user access through a menu-driven screen-oriented system, freeing the user from all but the most elementary syntax requirements.

LANGUAGE FORMAT

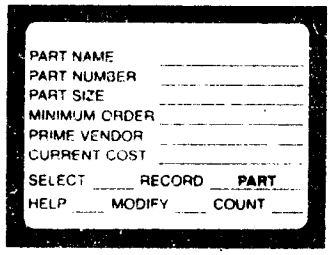
The principal difference between PLEX and SCF is the unit of data accessed by each call to the data base system. In SCF an entire search and operation to be performed on the data base will be specified within one syntactic unit, and no reference will be made to positioning that has resulted from a prior command. This does not imply that the user is unable to carry over position between commands, but rather that commands tend to be more independent of one another and positional control more explicit.

Query/Update by Example (QueX) is an application program designed to give each user easy access to his own data, or "user view. The date is presented in a fill-in-the-blank format, with the specific subset of data allowed for each user. The user will select an operation to be performed, the name of a record which he is allowed to access, and the search criteria in the format described. (See Figure 4.4.8.) In response to a retrieval

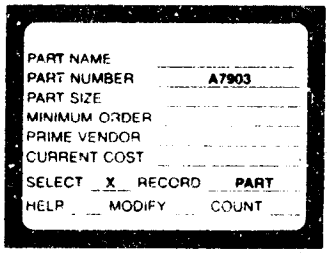
SCF

QUEST	QUEUE	PLEX
DATE FORMAT		QUEUE TERMINATE CANCEL QUEUE
Session-Wide STRINGs Functions FUNCTIONs In-line Summary (MIN MAX SUM) (AVG COUNT SIGMA) Command Files COMMAND DATA REPORT MESSAGE	**Session-Wide** STRINGs TERMINATE CANCEL QUEUE REPEAT *DATA*	**Auxiliary** LOCATE ORDER BY LINK
Updates LOAD ADD CHANGE ASSIGN REMOVE INSERT TREE REMOVE TREE ASSIGN TREE	**Updates** ADD CHANGE ASSIGN REMOVE APPEND TREE REMOVE TREE	**Updates** MODIFY REMOVE INSERT REMOVE TREE LOAD
Retrievals PRINT LIST TALLY UNLOAD DESCRIBE STRINGS FUNCTIONS OB BY	**Retrievals** PRINT PRINT TREE TERMINATE/UNLOAD IF THEN ELSE	**Retrevials** GET GETI GETD GETA
WHERE KEY NON-KEY Binary Operators (EQ NE LT) (GT GE LE) Unary Operators (EXISTS FAILS) Range Operators (EQ NE SPANS) Boolean Operators (AND OR NOT) Tree (HAS AT) Miscellaneous Text search CONTAINS SAME	**WHERE** KEY NON-KEY Binary Operators (EQ NE LT) (GT GE LE) Unary Operators (EXISTS) Boolean Operators (ANY OF ALL OF) Tree HAS	**WHERE** KEY NON-KEY Binary Operators (EQ NE LT) (GT GE LE) Unary Operators (EXISTS FAILS) Range Operators (EQ NE SPANS) Boolean Operators (AND OR NOT) Tree (HAS AT) Miscellaneous Text Search CONTAINS

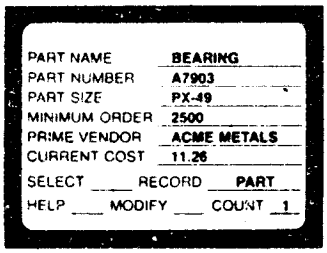
Figure 4.4.7/QUEST, QUEUE, and PLEX Modules



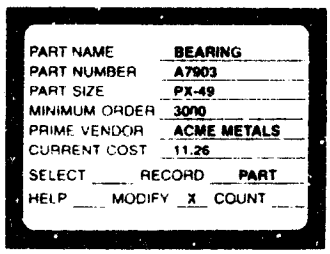
1. DURING A TYPICAL SESSION, QUEX SUPPLIES A RECORD DESCRIPTION.



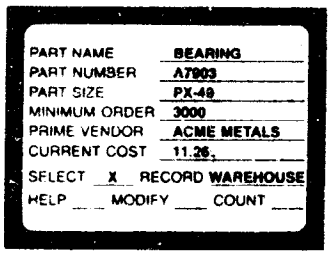
2. THEN THE USER FILLS IN WHAT HE KNOWS.



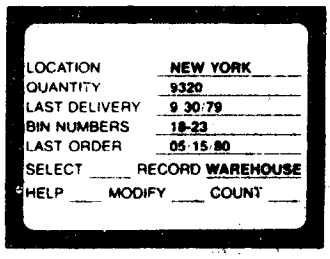
3. QUEX FILLS IN THE BLANKS.



4. A USER CAN MODIFY DATA.



5. A USER CAN REQUEST ANY RELATED RECORD.



6. QUEX SUPPLIES ANY RECORD FROM ANY DATA BASE.

Figure 4.4.8/QueX Session

request, one record at a time will be returned in menu form. For updating, the user changes the fields on the form. All positioning is accomplished by predefined processes established by the data administrator.

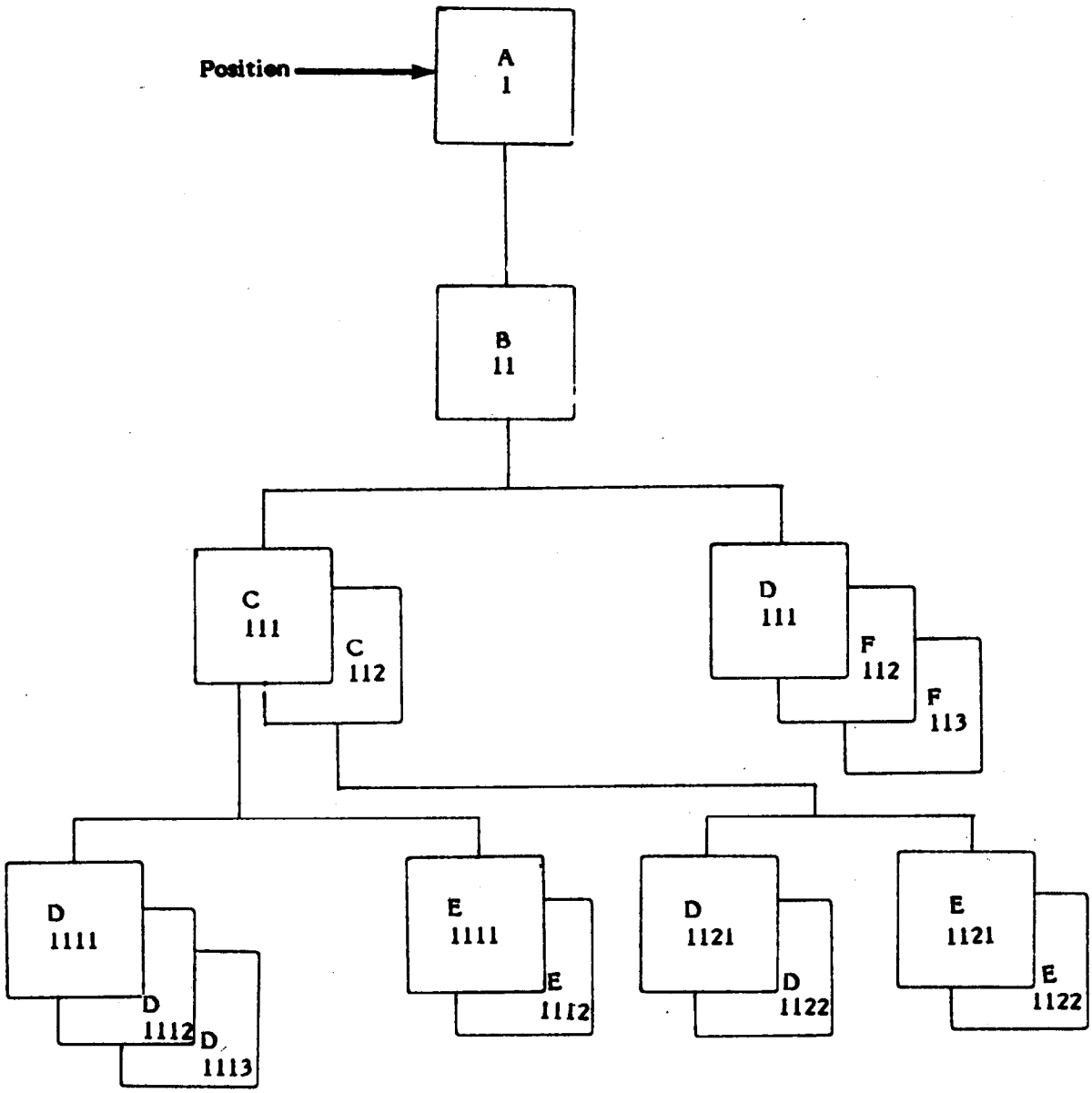
As with QueX, PLEX generally operates on one data record at a time and relies on the position within the data base established by prior operations. The position determines the effective context of any further operation. So the PLEX user has more control over navigation in the data base, and he can better coordinate the actual placement of data while manipulating the data base with a finer tuning of logic than the SCF user. Figures 4.4.9 and 4.4.10 demonstrate what is meant by "universe of data." From any position in the data base, all dependents or ancestors of that data record can be accessed without relinquishing the position currently held.

DATA PRESENTATION

The second major difference relates to the method of presenting data values for insertion into the data base. In both the queued and the immediate modules of SCF, data values are presented as either embedded in the statement in the form of a value string or through a separate data input stream, as detailed in Figures 4.4.11 and 4.4.12. When operating through PLEX the input/output area for a data base record is called a "subschema record" (SSR). A subschema record is a program resident definition of a record, and it consists of a subset or the entire set of items in the record in any order. All data to be inserted, removed, or modified in the data bases are moved into the appropriate subschema record skeleton. Conversely, all data retrieved by the system are brought from the data base to a specific subschema record. Multiple SSRs can be used to redefine a data base record.

All data accesses are ultimately dependent on locating the record of interest. In QUEST and QUEUE, the WHERE clause—the technique for qualifying records—is normally appended to most commands to establish position or qualify a subset of the data base. (See Figures 4.4.13 and 4.4.14.) The clause qualifies records from one or more positions in the data base and incorporates boolean logic, text search, and relational structure, as well as indexed and non-indexed search features. The user may also process records sequentially or in value order. The REPORT facility of SCF selects data records to be included in one or more reports with the same syntax in the "GENERATE (reports) WHERE ..." statement. PLEX offers the same features, except it cannot process selected records all at once. Instead, the language uses a "stack" mechanism that keeps track of position in the data base.

Earlier it was mentioned that SYSTEM 2000 recognizes three component types—records, items, and stored commands (STRINGS and FUNCTIONS). STRINGS and FUNCTIONS are available only through SCF. A STRING may be viewed as a pre-defined processing request that may in turn contain other embedded STRINGS or may be specified with variable parameters supplied at the time the STRING is invoked. Figures 4.4.15 and 4.4.16 demonstrate the definition and usage of STRINGS. The FUNCTION capability of data base definition allows the user to define "virtual" fields which do not exist in the data base, but which are computed at the time of retrieval. If the user, for instance, needed to maintain a data field that equalled the result of "data field a" divided by "data field b," he could declare this FUNCTION on the data base definition. The resulting field would not occupy any space in the data base, but it could be referenced by access commands as if it were a real component. Figure 4.4.17 depicts examples of user-defined FUNCTIONS. In addition, the QUEST feature of SCF provides a set of system FUNCTIONS that enable the user to obtain arithmetic statistics about data values in the data base. A list of these FUNCTIONS is shown in Figure 4.4.18.



In this example, the user is "positioned" at data set A1, and therefore has all of the data sets in his "universe of data."

Figure 4.4.9/Positioning and Universe of Data in Plex

The user is "positioned" at data set A1, and therefore has all of the data sets in his "universe of data."

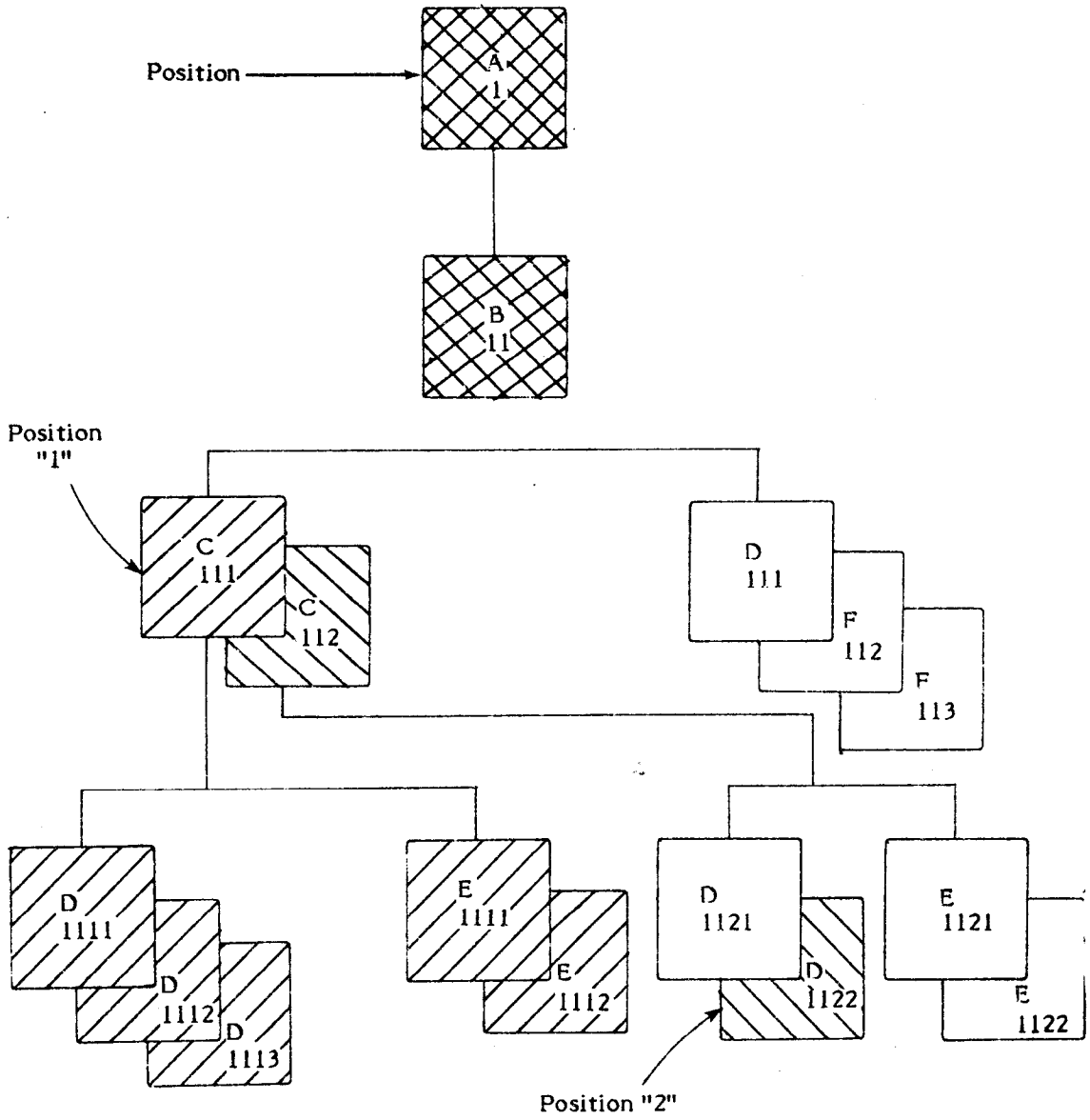


Figure 4.4.10/Positioning and Universe of Data in Plex

```
APPEND TREE REC1 EQ 9*ABC*10*DEF*END*
WHERE C1 EQ ALPHABET*:
```

This statement, or command, will add a record REC1 to a data hierarchy that has "ALPHABET" in the first entry, C1. The line notations in that data definition for the entries of REC1 were 9, 10, 11. The system will create a new record so that ABC is stored in 9, DEF in 10, and a null entry in 11. If either or both entries had been declared keys at data definition time, they would be added to the inverted files for occurrences of REC1.

Figure 4.4.11/Data Values