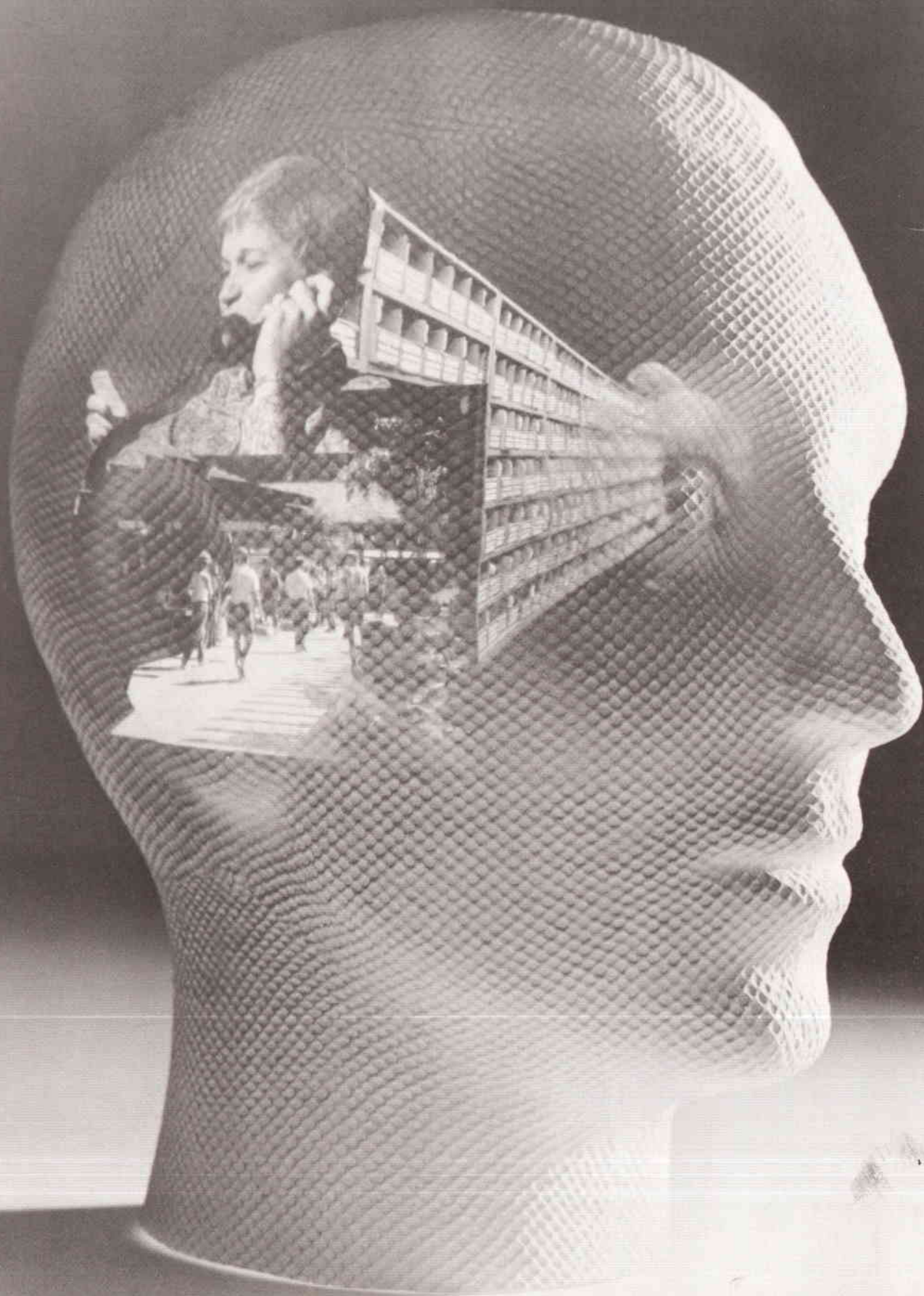


JULY 1974

# HEWLETT-PACKARD JOURNAL



# Powerful Data Base Management Systems for Small Computers

For the first time, sophisticated data management is available to the user of computers as small as the HP 2100 and HP 3000.

by Richard E. McIntire

**C**OMPUTER INFORMATION SYSTEMS harness the processing power of the computer to collect and organize data and make it easily accessible to the businessman, scientist, or other user who needs it. A computer information system may be thought of as a set of applications programs using common data bases through a data management system, as shown in Fig. 1. The data management system is composed of two separate software entities: the operating system and the data base management system.

In the past it has been generally accepted that computerized information systems and, similarly, data base management systems, required large-scale computers. But as computer performance has expanded, the conventional concept of a data base management system has changed. Increasingly evident is a shift away from large central processors designed to serve all needs and all users. The demand now is not necessarily a bigger or faster computer, but a less costly, more flexible system tailored to serve definite data base requirements and process the total expected workload at a lower cost.

## Small-Computer Data Management

Hewlett-Packard's new data base management system, IMAGE, and its companion data base inquiry facility, QUERY, for the first time make sophisticated information management available to the user of small, low-cost computer systems based on the HP 2100 and HP 3000 Computers. The design objective for these new software systems was to make common data bases and data management services accessible to users ranging from those who know nothing about the system and want to use it without learning complex programming languages to those who understand the system well and want to manipulate its inner workings to their advantage.

IMAGE is designed for the internal user, that is, the

information systems manager, the data base manager, the systems analyst, the computer system specialist, and the programmer—the user who is interested in the computer, its associated input/output devices, computer programs, and the influence of these components on data entry, organization, and retrieval. QUERY is designed for the external user, the non-



**Cover:** The images on the head represent information – credit card records, inventory records, student and course data, and so on – information that must be organized and made readily available to those who need it (and protected from those

who don't). IMAGE, a sophisticated new data base management system for HP 2100 and HP 3000 Computers, has information management capabilities formerly available only in large computer systems.

## In this Issue:

- Powerful Data Base Management System for Small Computers, by Richard E. McIntire ..... **page 2**
- Quality Frequency Counters Designed for Minimum Cost, by Lewis W. Masters and Warren J. O'Buch ..... **page 11**
- A Versatile Bipolar Power Supply/Amplifier for Lab and Systems Use, by Santo Pecchio ..... **page 15**
- An Automatic Exposure Control for a Lab-Bench X-Ray Camera, by John L. Brewster ..... **page 20**

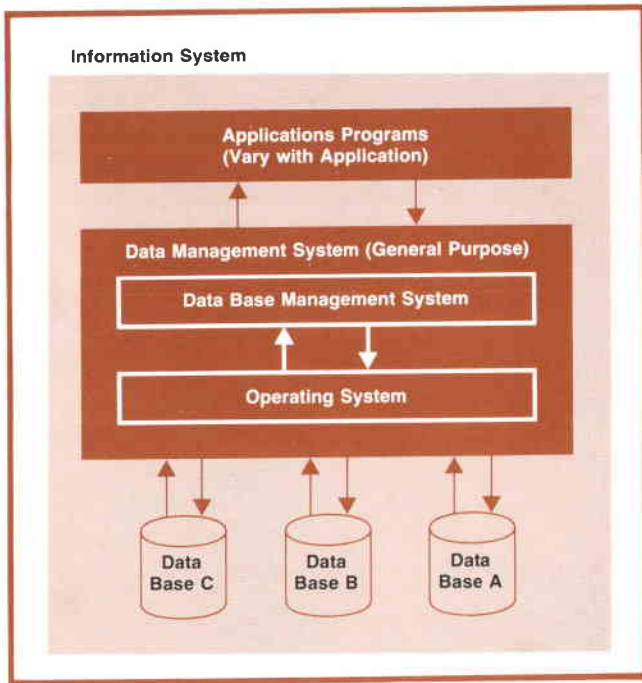


Fig. 1. General concept of a computer information system.

specialist, whose needs are to retrieve, analyze, and report information to support his function and decision processes.

IMAGE consists of a set of programs that create and maintain complex data structures known as data bases, and a set of library procedures that enable users to access, modify, and report on the data content of the data bases. There are two versions: IMAGE/3000 for HP 3000 Computers and IMAGE/2000 for systems based on HP 2100 Computers. IMAGE/2000, a subset of IMAGE/3000, has many but not all of the features of IMAGE/3000. IMAGE/2000 operates under the control of the disc operating system for HP 2100 Computers (DOS-III) and IMAGE/3000 is executed under the control of MPE/3000, the operating system for HP 3000 Computers.

### IMAGE/3000 Features

IMAGE provides powerful software tools that help the data base manager define and create a data base tailored to his requirements. It has a network data structure that allows cross-referenced access to collections of data down to the smallest unit.

The data base manager defines data sets and their interrelationships just once. Thereafter, applications programmers can search, retrieve, or update their data bases from host-language programs without concern for the details of accessing the data base. Host languages can be FORTRAN, COBOL, or SPL (the HP 3000 Systems Programming Language).

IMAGE provides facilities for combining files from many applications into one data base, so sets of data

that occur in two or more old files are consolidated and need be stored only once.

A flexible security scheme lets the data base designer control access to any subset of the data base, down to the smallest unit. The user must include a password or level word identifying his access. Reading and updating are treated as two separate operations, and the system checks each data transfer request to determine the user's access before allowing the read or update operation.

### IMAGE Subsystems

IMAGE consists of three basic elements (Fig. 2): a data base definition subsystem (DBDS), a data base management subsystem (DBMS), and a data base utility subsystem (DBUS).\* The data base manager uses DBDS to define the data base and DBUS to create and maintain the data base. The applications programmer, the principal user of IMAGE, writes procedural programs using a host language and the data base management language, DBML, which operates on the data base using DBMS.

DBDS is independent of the applications programs. This system allows the data base manager to define all aspects of data base organization. His data base definition is called a schema. Using the data base definition language, the data base manager defines data items, security levels, and relationships and mappings between data sets.

\* DBDS corresponds to the CODASYL data base task group's data base description statements and schema processor. DBMS corresponds to CODASYL's data base manipulation language and library routines. DBUS corresponds to various CODASYL utility routines.

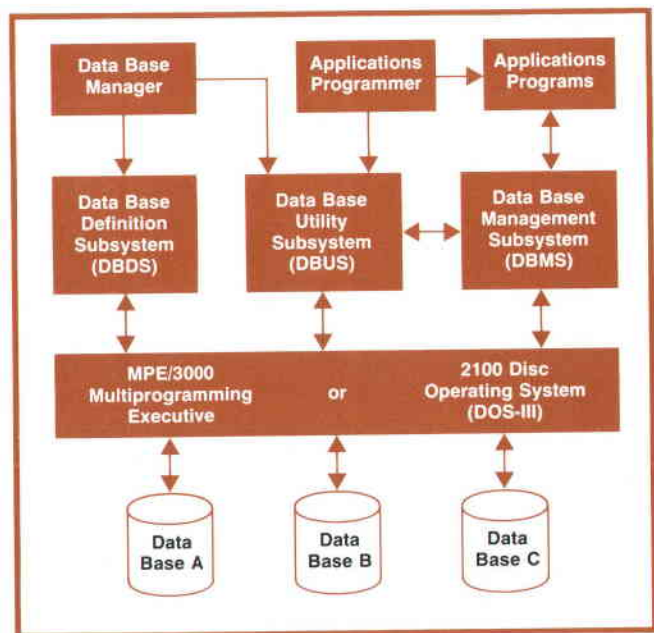


Fig. 2. IMAGE subsystems. IMAGE/2000, for HP 2100-Series Computers, is a subset of IMAGE/3000, which runs on an HP 3000 Computer System.

DBMS provides the means for applications programmers to access an IMAGE data base. DBMS is a set of stored library routines invoked by CALL statements in host-language applications programs. DBMS serves as the interface between the data base and the applications programs, and either can change without affecting the other. DBML is a non-procedural language. It relies on the host language to provide a framework and the capabilities required to manipulate the data. Thus, the application programmer uses the full data processing power of his host language and leaves data base structure and access activity to the DBMS. DBML is not an inquiry language and does not provide for selection criteria in the form of Boolean expressions; however, QUERY does allow such selection criteria.

The third IMAGE subsystem, DBUS, consists of a set of support routines that run as stand-alone programs. DBUS permits building, dumping, and restoring data bases and assists in their restructuring.

### Preparing a Data Base

Four steps are required to prepare an IMAGE/3000 data base (see Fig. 3). The first step is for the data base manager to use the data base definition language to define the data base structure. Fig. 4 is an example of a typical data base definition, or schema. Second, a DBDS program is employed to process the schema, the result being a disc-resident table describing the data base. This table is called a root file. Third, a DBUS program is employed to build the empty disc files that constitute the framework of the data base; each such disc file is called a data set. Fourth, data is entered into the previously created data sets by means

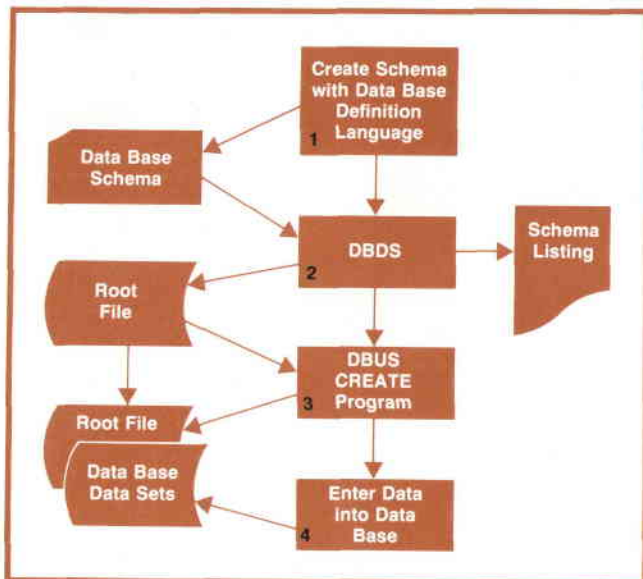


Fig. 3. The four steps in preparing an IMAGE data base so it can be used by applications programs.

```

BEGIN DATA BASE SCHOOL1
LEVELS:
  4 CLASS#
  5 TEACHID#
  6 STUDENT#
  8 TEACHER#
  10 CLASSUD#
  11 #TEACHID#
  12 STUDENT#
  14 TEACH#
  15 ZADH#Z#

ITEMS:
  SCHL-TEACH, X12(8,15)
  SCHL-CRSE, X6(4,15)
  NO-SEC, X7(4,14)
  DESCR, X14(4,14)
  CREDITS, X2(4,14)
  SEATS, P4(4,10)
  FEES, P4(4,10)
  MEET-ROUNTS, X6(4,10)
  SEMESTER, X2(4,12)
  ROOM-NO, X6(4,14)
  SCHL-CRSE-ID, X8(4,15)
  SCHL-STUDENT, X8(6,15)
  COM-CODE, X2(6,12)
  GRADE-LEVEL, X2(4,10)
  ENROLL, X2(6,12)
  MONTH, X2(6,12)
  DAY, X2(6,12)
  ABSENT, X2(6,12)
  COMMENTS, X10(8,15)
  SCORES, SP4(6,12)

SETS:
  NAME: TEACH-MSTR,A(8,14)
  ENTRY: SCHL-TEACH(1)
  CAPACITY: 300

  NAME: SECTION-MSTR,A(5,12)
  ENTRY: SCHL-CRSE-ID(2)
  CAPACITY: 1000

  NAME: STUDENT-MSTR,MANUAL(6,12)
  ENTRY: SCHL-STUDENT(1),ROOM-NO,COM-CODE,GRADE-LEVEL
  CAPACITY: 3000

  NAME: COURSE-MSTR,MANUAL(4,14)
  ENTRY: SCHL-CRSE(1),NO-SEC,DESCR,CREDITS,SEATS,FEES,
  MEET-ROUNTS,SEMESTER
  CAPACITY: 1000

  NAME: STUDENT-TEST,DETAIL(5,12)
  ENTRY: SCHL-STUDENT(STUDENT-MSTR),
  SCHL-CRSE-ID(SECTION-MSTR),COMMENTS,SCORES
  CAPACITY: 10000

  NAME: COURSE-SEC,DETAIL(5,11)
  ENTRY: SCHL-TEACH(TEACH-MSTR(SCHL-CRSE)),
  SCHL-CRSE(COURSE-MSTR),
  SCHL-CRSE-ID(SECTION-MSTR),
  ENROLL,ABSENT,MONTH,DAY
  CAPACITY: 30000

END.
  
```

### Summary of Processed Schema

DATA SET NAME	TYPE	LEVEL		FLD CNT	PT CT	ENTR LGTH	MED REC	CAPACITY	BLK FAC	BLK LGTH	DISC SPACE
		R	W								
TEACH-MSTR	A	8	14	1	1	6	16	300	31	498	44
SECTION-MSTR	A	5	12	1	2	4	19	1000	20	382	153
STUDENT-MSTR	M	6	12	4	1	13	23	3000	22	508	552
COURSE-MSTR	M	4	14	8	1	17	27	1000	14	379	219
STUDENT-TEST	D	5	12	4	2	18	26	10013	19	496	2112
COURSE-SEC	D	5	11	7	3	17	29	30004	13	378	6927
								TOTAL DISC SECTORS INCLUDING ROOT:	10015		

NUMBER OF ERROR MESSAGES: 0  
 HIGHEST LEVEL WORD: 15 ITEM NAME COUNT: 20 DATA SET COUNT: 6  
 ROOT LENGTH: 676 BUFFER LENGTH: 508 TRAILER LENGTH: 256  
 ROOT FILE SCHOOL CREATED.

Fig. 4. An example of a schema that defines an IMAGE/3000 data base.

of QUERY or applications programs.

When an applications program uses the data base, it must enter its data requirements into a data buffer, which defines the elements of the data base the program needs. When the program wants to access the data base, it communicates with IMAGE/3000 using DBML to access the DBMS subsystem. Using the program's data buffer and the data base root file to locate the desired data, DBMS gets the data for the applications program. If updating is required, DBMS will take the new data from the data buffer and put it into the data base.

## IMAGE/3000 Data Base Organization

Within an IMAGE/3000 data base are three basic structures: data items, data entries, and data sets.

The *data item* is the smallest accessible data element. Each data item is a value and is referenced by a *data item name*, which is a character string defined in the schema by the data base manager. Usually, many data item values are referenced by the same data item name.

Data Item Name from Schema	Data Item Values
NAME	SMITH, JONES, JOHNSON, GREEN, MEADE, DILLION
CITY	SAN JOSE, DENVER, PRESCOTT, AJO, GILROY, TRACY
STATE	CALIFORNIA, ARIZONA, COLORADO

A *data entry* is an ordered collection of related data items and is defined by an ordered listing of the data item names in the schema. Data entries are stored in physical locations on a direct-access storage device, such as a disc.

Data Entry Definition from Schema	NAME	CITY	STATE	DEGREE	SEX
$R_i$	SMITH	PRESCOTT	ARIZONA	BA	M
Data $R_{i+1}$	JONES	DENVER	COLORADO	MBA	M
Entries $R_{i+2}$	JOHNSON	SAN JOSE	CALIFORNIA	MS	F
$R_{i+3}$	GREEN	TRACY	CALIFORNIA	BA	F

A *data set* is a collection of data entries sharing a common definition. All data entries within a data set are of the same length (the maximum length in IMAGE/3000 is 4094 bytes). A *data set name*, a character string defined in the schema, references any or all of the data entries of a data set. The number of data entries in a data set is limited by available disc space.

Data Set Name: PERSONNEL

Data Entry Definition from

Schema: NAME, CITY, STATE, DEGREE, SEX

Smith Prescott Arizona BA M

Jones Denver Colorado MBA M

Johnson San Jose California MS F

Green Tracy California BA F

Physical  
Storage  
Locations  
on Disc

A *data base* is a named collection of related data sets. It is referenced by a *data base name*.

## Search Items and Chains

Data entries in a data set can be referenced by one or more data items known as search items. Search

items of a data set are specified in the schema. Data entries of a data set are linked together in subsets based on the values of their search items. For example, the data item STATE might be specified as one of the search items for a data set. A general reference to all data entries of the data set having the same value of STATE is then possible. If all fifty states of the United States of America are represented, the data set would be logically divided into fifty subsets, with all the entries of each subset containing the same value of STATE.

Each data entry of a data set is distinguished by an entry number. An entry number is a unique integer between 1 and N, where N is the data set's capacity or the total number of available data entry storage locations, as defined in the schema. Each storage location is initially empty. Whenever a new data entry is added to a data set, it is assigned one of the unused entry numbers and written in the corresponding storage location. Entry numbers are the means by which data entries with like search item values are linked together.

## IMAGE/3000 Chains

For each search item, pointers are maintained by IMAGE with each data entry, along with the data item

## Why Data Base Management Systems?

Historically, the cost/performance ratio of computer hardware has improved, while the cost of programming has increased. Just as a major reason for the development of high-level languages such as FORTRAN and COBOL was to reduce coding time and cost, the primary objective of information systems development is to reduce the time and cost of writing programs to store and retrieve information in a computer system.

In data management, there are certain functions that are repetitive, time-consuming, and error-prone, and if each programmer performed these functions every time he created a new program it would be extremely inefficient and uneconomical. For example, in an earlier approach to data processing using a batch mode of operation, all the data for a particular application had to accompany the applications program. Thus data in the personnel file might be repeated in the skills file, the payroll file, and the medical file. When these files are combined into a common data base, redundant data is eliminated, storage costs are lower, and the data is internally consistent, requiring only one standard procedure for updating or modification. Also, data base management can be performed independently, freeing the applications programs from this task.

Data base management concepts are an evolution of earlier EDP techniques and not a radical new method. The innovation in the data base approach is that the definition and control of the data base are independent of the applications. The common data base of logically connected files or items of data is then accessible to all programs of the proper security clearance by means of special software that permits more efficient data processing and therefore easier systems development and lower programming costs.

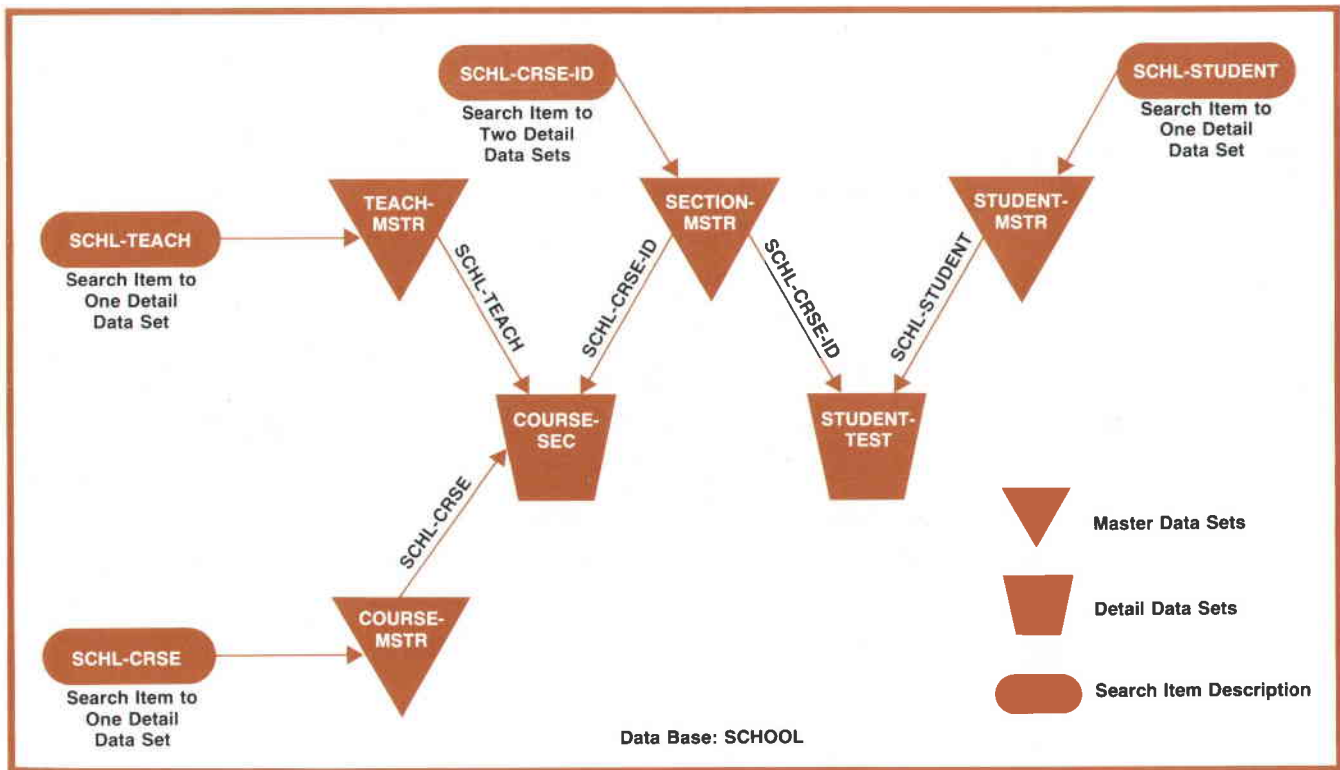


Fig. 5. IMAGE/3000 data sets are of two types, master and detail. An important purpose of master data sets is to serve as indexes to detail data set chains.

values for that entry. These pointers contain the entry numbers of the preceding and succeeding data entries within the data set that have the same search item value. All data entries having the same search item value are linked together in this way, and are referred to collectively as a chain.

In a set of employees, for example, engineers form a subset of the employee set, based on the search item JOB-TITLE. All engineers would be linked together to form an engineer chain. Thus, members of a data chain have in common the value of a specified search item.

A chain may be in sorted order if each new data entry is inserted into the chain at a point determined by the value of a specified data item called a sort item. Sort items are defined in the schema.

#### IMAGE/3000 Data Set Types

There are two types of data sets in IMAGE/3000: master data sets and detail data sets (Fig. 5). Detail data sets contain "line item" information. For example, in the detail data set PERSONNEL, each person's location, educational experience, and similar information is stored. An important purpose of master data sets is to serve as indexes to detail data set chains.

The data entries of a master data set have just one search item and unique search item values. Thus in a

master data set with the search item STATE, there are at most fifty entries, one for each of the United States that appears in a related detail data set.

Data entries of a master data set contain pointers to corresponding chains in related detail data sets. For example, if the specified search item is STATE, the "New York" entry in the master data set will contain the entry numbers of the beginning and end of the "New York" chain in each related detail data set. The "New York" entry (and all other entries) in the master data set may also contain master information about the state, such as population. This information then does not have to be duplicated with each "New York" entry of the detail data set or sets.

For each search item defined in a detail data set, an existing master data set is specified by name. This association establishes a master-detail data set relationship. A master data set may be related to more than one detail data set, and a detail data set may be related to more than one master data set.

#### IMAGE/3000 Data Access

All data base operations are accomplished through the facilities of the DBML that interface with a host language such as COBOL, FORTRAN, or SPL. The DBML is structured so that each command consists of a DBMS procedure call followed by a set of parameters.

In general, access to data within a data base is carried out on the data entry level. That is, each call to a DBMS procedure accesses some or all of the data items within a data entry. The table below presents an outline of the parameter set.

```
CALL <DBMS procedure> USING
<base><dset><mode><status>[<list><buffer><arg>]
```

Where:	<base>	is the data base of interest
	<dset>	is the data set of interest
	<mode>	various modes within each DBMS procedure
	<status>	status area containing the results of the execution of the DBMS procedure in the application program.
	<list>	is a list of data item names of interest.
	<buffer>	is the address of a buffer in the user's data area
	<arg>	is the search item of interest
	[ ]	indicates optional parameters

Some of the functions that can be accomplished through use of the DBMS procedures include adding a new data entry to a data set, deleting a data entry from a data set, reading some or all of the data items of a data entry, and changing the values of items of a data entry.

#### Master Data Set Access

Data in a master data set may be accessed in serial, directed, calculated, or chained fashion.

In serial access, DBMS accesses the data entry whose entry number is one greater than the last until a data entry is located and read or until an end of file is encountered. When a data set is accessed by a program for the first time, a search begins at entry number one. Reverse serial access is also possible.

Directed access is accomplished by an applications program's supplying an entry number. If present, the data entry with the specified entry number is read. If no such data entry exists, the program is notified by an exceptional condition return.

Calculated access is based on a search item value. It involves mapping the applications-program-supplied item value into a primary entry number by means of what is known as a key transformation. The data entry at that location is then accessed to determine if it contains the matching search item value. The key transformation may map more than one search item value into the same primary entry number. When this occurs the search item values are called synonyms. If the data entry at the location specified by the primary entry number does not contain the desired search item value but does contain a synonym, an exhaustive search of all synonyms with the same primary address is made to locate the desired data entry, if it exists in the data set. To eliminate ambiguity, data entries of a master data set must

have unique search item values. Calculated access is used to retrieve a selected data entry or obtain the chain head pointer of a detail data set chain from the master data entry without an exhaustive search of the entire master data set. Obtaining the chain head pointer is generally done as a prelude to accessing the data entries of a chain in a detail data set.

In chained access of a data entry in a master data set, the applications program may read the next data entry in a synonym chain in either the forward or the backward direction.

#### Detail Data Set Access

The first data entry in a detail data set is assigned entry number one and subsequent data entries are assigned entry numbers 2, 3, 4 and so on in sequence. However, DBMS keeps track of deleted data entries and always reallocates deleted entry numbers. This modified sequential allocation is called serial allocation; it applies only to detail data sets.

Data entries are logically linked to other similar data entries in the detail data set, as well as to the master data entry to which they belong. Detail data entries may be retrieved directly through chains related to the master data set.

The data entries of a detail data set may be accessed in serial, directed, or chained fashion. Serial and directed access to data entries in a detail data set are identical to that for master data sets. Chained access to data entries is only applicable to detail data sets having one or more search items. When a new data entry is added to such a data set it is linked into the existing chain of data entries whose search item values match that of the new data entry. If the new data entry has more than one search item this linking process is done for each search item.

Chained access to data entries of detail data sets enables rapid access to all data entries having a common search item value but, in itself, does not assist in locating the initial occurrence of a value. Retrieval of detail data entries in a given chain is generally preceded by a calculated access to the corresponding data entry in a master data set to obtain the chain head pointers. Once the application program has located itself on a chain, it may read the next data entry in either the forward or backward direction of the chain.

#### QUERY/3000 Subsystem

QUERY/3000 is a self-contained subsystem that interfaces with the DBMS of IMAGE/3000. A major development problem was to design a suitable language that would respond to spontaneous and unanticipated inquiry concerning data in an IMAGE/3000 data base. It was critical that the user be given a communication tool that relates to his problems rather than to the underlying programming problems, and that the language be the user's natural lan-

guage or something very close to it.

By entering English-like commands to QUERY/3000, the user can access data in the data base without learning complicated programming languages. The QUERY language, QL, allows the user to specify the information he wants using logical Boolean expressions of key-value pairs. The user may write procedures, have them executed, and, if desired, have them stored for repeated use at a later time.

### QUERY/3000 Functions

QUERY/3000 is composed of six functions (see Fig. 6): QUERY language (QL), QUERY interpreter, retrieve subsystem, report writer subsystem, update subsystem, and auxiliary functions.

The QUERY interpreter initially displays a standard HP 3000 program identity message, an initialization message, and a prompt character on the user's terminal. The prompt character tells the user that he has correctly requested QUERY/3000 and the QUERY interpreter is active and awaiting further input. The QUERY interpreter is the switching module of QUERY/3000. It interprets requests from the user, passes control to the appropriate module for further processing, and outputs the results to the user.

The retrieve subsystem does both inter-record processing, in which specified data entries are selected from the data base, and intra-record processing, in which the selection criteria are more restrictive. Data entries satisfying the selection criteria are extracted from the data base and their entry numbers are placed in a selection file. The number of data entries meeting the retrieve criteria is then displayed to the user. The user may now allow the report writer or update subsystems to execute, or may limit the number of data entries retrieved by issuing another FIND command.

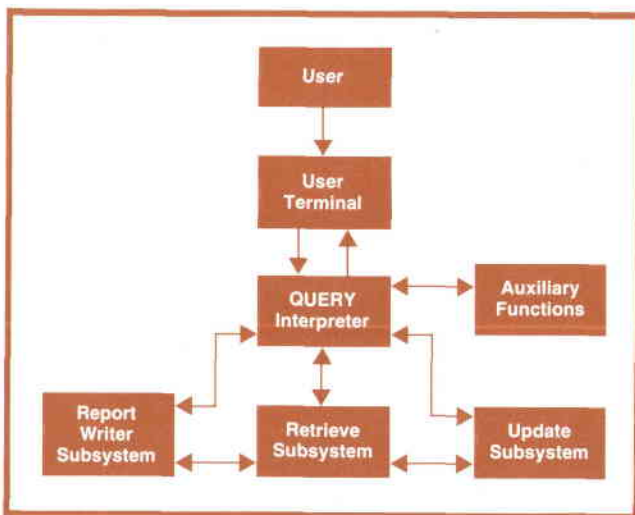


Fig. 6. QUERY/3000 functions make it easy for the non-specialist to retrieve and update data in IMAGE/3000 data bases.

QUERY Language		
Utility Commands	Main Commands	Stored Procedure Commands
DEFINE	FIND	CREATE
EXIT	REPORT	DISPLAY
HELP	UPDATE	DESTROY
FORM		ALTER

Fig. 7. QUERY/3000 commands.

The report subsystem provides flexibility in the format of reports. Reports may include page headings, column headings, and page numbers. Data item values may be subtotaled and totaled, and data entries may be sorted by multiple categories.

The update subsystem allows on-line update to the data base. The updated information is usable immediately after updating.

The auxiliary functions allow the user to create procedures for changing specific data item values in a data entry, for locating data entries that qualify according to specified search conditions and for writing reports about data entries that were retrieved. The user can also list, modify, or delete stored procedures.

### QUERY-Language Commands

The QL commands, shown in Fig. 7, may be used in either an on-line or a batch environment.

The HELP command is a tutorial aid; it may be used whenever the user wants to know the required syntax for a particular request.

The EXIT command may be used anytime an input response is expected; it causes immediate termination of QUERY/3000 and returns control to MPE/3000. It is the proper method of notifying the system of exiting from QUERY/3000.

The purpose of the FORM command is to display the structure of a data base. When used, it lists the data item names, data set names, and relationships defined in the schema. These names may then be used in other commands.

The DEFINE command is used to inform QUERY/3000 of the data base of interest, the data sets of interest, a SPEC-FILE containing stored procedures, and the output device name. These remain valid for any other QL command, but may be changed at any time. If a required define type is not present in the define table, QUERY/3000 will display an error message and the user must supply the required define type before being allowed to continue.

### Data Retrieval

Data entries are retrieved from the data base as

specified in the FIND command. When there is more than one comparison for each data entry, each comparison must be connected logically to the next by AND or OR. The FIND command is of the general form:

FIND <a data item name> <with a specified relation to> <a specified data item value> END

A list of available data item names may be obtained by using the FORM command. The type of comparison to be made is indicated by the relational operators listed below:

RELATIONAL OPERATOR	MEANING
IS IE	=
ISNOT INE	≠
ILT	<
INLT	⋆
IGT	>
INGT	⋆
IB	$n_1 \leq a \leq n_2$ , where a is a data item name defined in the schema, and $n_1$ and $n_2$ are specific limits.

When multiple logical connectors are used, ANDs are satisfied first and then ORs. Parentheses are not allowed.

Examples:

1. Retrieve all data entries of persons who are between the ages of 21 and 25, live in the state of Iowa, and have blood type AB.

FIND STATE IS "IOWA" AND AGE IB "21", "25" AND B-T IS "AB" END

2. Retrieve all students majoring in German or Spanish, whose cumulative grade average is greater than 2.9.

FIND MAJOR IS "GERMAN", "SPANISH" AND AVGRADE IGT "2.9" END

### Reports

The REPORT command is an extension of the FIND command. Its purpose is to generate a report of the data entries retrieved by specifying either a report procedure, the name of a report procedure stored on a SPEC-FILE, or the keyword ALL, which prints the data item name and data item value for each data item in every data entry retrieved, without any report formatting or data editing.

If the user wants to print the retrieved data on an output device in a formatted report, he may specify a report procedure. There are six statements that enable him to do this. These specify the header information to be printed at the top of each page, the order in which data is to be sorted, the data to be printed in each report column and the way in which it is to be punctuated, and any summaries to be printed of information contained in parts or all of the report.

Fig. 8 shows an example of a QUERY/3000 inquiry, report definition, and report.

```

QUERY/3000 HEADY
NEXT?
DEFINE
DATA-BASE = SCHOOL
LEVEL = ZADMINZ
MODE = 2
DATA-SETS = COURSE-SEC
SPEC-FILE =
OUTPUT = TERM
NEXT?

FIND MONTH IS "5" AND DAY IS "6","7","8","9","5" AND
SCHL-CRSE-ID IS "CHEM1","ENG2","HIST5","MATH2","MATH3","SHOP3" END
18 ENTRIES QUALIFIED
NEXT?

REPORT
M1,"M D A I L Y ", 18
M1,"M A T T E R N A N C E ",42,SPACE B1,SPACE A2
M1,"R E G I S T E R ", 60
M2,"WEEK OF MAY 5, 1974", 20
M2,"PAGE", 63
M2,"PAGE", 66, SPACE A2
M3,"DAY", 4
M3,"COURSE", 15
M3,"TEACHER", 38
M3,"ENROLLED", 55
M3,"ABSENT", 65, SPACE A2
S1, DAY 5, SCHL-CRSE-ID
D, DAY, 3
D, SCHL-CRSE-ID, 20
D, SCHL-TEACH, 40
D, ENROLL, 52
D, ABSENT, 63
T1," ",65,SPACE B2, SPACE A2
END

D A I L Y A T T E N D A N C E R E G I S T E R
WEEK OF MAY 5, 1974 PAGE 1

DAY COURSE TEACHER ENROLLED ABSENT
5 CHEM1 BASS 18 2
5 ENG2 JOHNSON 30 5
5 HIST5 CORCORAN 28 2
5 MATH2 WHITE 15 1

6 CHEM1 BASS 18 2
6 MATH3 BROWN 25 3
6 SHOP3 DOLAN 25 4

7 CHEM1 BASS 18 3
7 ENG2 JOHNSON 30 5
7 HIST5 CORCORAN 28 2
7 MATH2 WHITE 15 4

8 CHEM1 BASS 18 0
8 MATH3 BROWN 25 3
8 SHOP3 DOLAN 25 6

9 CHEM1 BASS 18 1
9 ENG2 JOHNSON 30 7
9 HIST5 CORCORAN 28 2
9 MATH2 WHITE 15 0

NEXT?
EXIT

```

Fig. 8. An example of a QUERY/3000 request, response, report definition, and report.

### Update Commands

The UPDATE command may be an extension of the FIND command if data items are to be deleted or replaced. The UPDATE command may also be used to add a new data entry. The update procedure allows three different types of update statements: ADD, DELETE, and REPLACE.

The ADD statement is used to add a data entry to the data set. The user need not include all values for every data entry; data items values not included will be declared null by the system. The system will prompt the user with the data item names of the data set when

requesting input. If a data item name is a key or search item, a value of null will not be accepted.

The DELETE statement in the UPDATE command is executed after the FIND command has been executed and the data entries have been retrieved. All retrieved data entries will be deleted from the data base.

To replace specified data item values, the REPLACE statement of the UPDATE command is used. It is executed after a FIND command has been executed and the data entries having data item values to be replaced have been selected. If the REPLACE statement is applied to data entries that do not contain the data item to be replaced, the system will not modify the retrieved data entries.

#### Stored Procedure Commands

The CREATE command is used to store a FIND, REPORT, or UPDATE command into a SPEC-FILE. FIND, REPORT and UPDATE commands may all be stored in one SPEC-FILE.


The DISPLAY command will display a stored procedure generated with a CREATE command. The keyword LIST will print all the procedure names in the SPEC-FILE.

The ALTER command allows the user to make modifications to a stored procedure in the SPEC-FILE. In-

dividual lines may be deleted, replaced, or inserted.

The DESTROY command causes the destruction of a stored procedure generated with a CREATE command. This command will not destroy any data sets of a data base.

#### Acknowledgments

The author wishes to acknowledge the following people who were involved in the design and implementation of the IMAGE and QUERY projects for both the HP 2100 and HP 3000 computers, and whose efforts made these projects a reality: Lee Johnson, Fred White, Lee Bollinger, Jon Bale, Ed Estes, Bob Mayer, Bob Brown, Carol Fuquay, Mary Berner, and Phil Taylor. 

#### References

1. "Feature Analysis of Generalized Data Base Management Systems," Association for Computing Machinery, New York, May 1971.
2. Report of the CODASYL Data Base Task Group, ACM, April 1971.
3. J.K. Lyon, "An Introduction to Data Base Design," John Wiley and Sons, Inc., 1971.
4. D. Lefkowitz, "File Structures for On-Line Systems," Spartan Books, 1969.

## SPECIFICATIONS

### IMAGE/3000      QUERY/3000

**DATA ITEM NAMES PER DATA BASE:** 255

**DATA SETS PER DATA BASE:** 99 (The space occupied by any single data set cannot exceed the capacity of any one disc drive. However, the total data base is limited only by the total available storage.)

**CHARACTERS PER ITEM NAME:** 16

**CHARACTERS PER DATA SET NAME:** 16

**DATA ITEMS PER DATA ENTRY:** 127

**MAXIMUM DATA ENTRY SIZE:** 4094 bytes

**KEYS PER DETAIL DATA SET:** 16

**DETAIL DATA SETS PER MASTER DATA SET:** 16

**ENTRIES PER CHAIN:** 65000

#### ORDERING INFORMATION:

32215A IMAGE/3000 Data Base Management System. Includes 1600 bpi magnetic tape and manual (additional manuals optionally available). Price in U.S.A., \$10,000.

32215A-130 Same as 32215A but on 800 bpi magnetic tape.

32216A QUERY/3000 Data Base Inquiry Facility. Includes 1600 bpi magnetic tape and manual (additional manuals optionally available). Price in U.S.A., \$1,000.

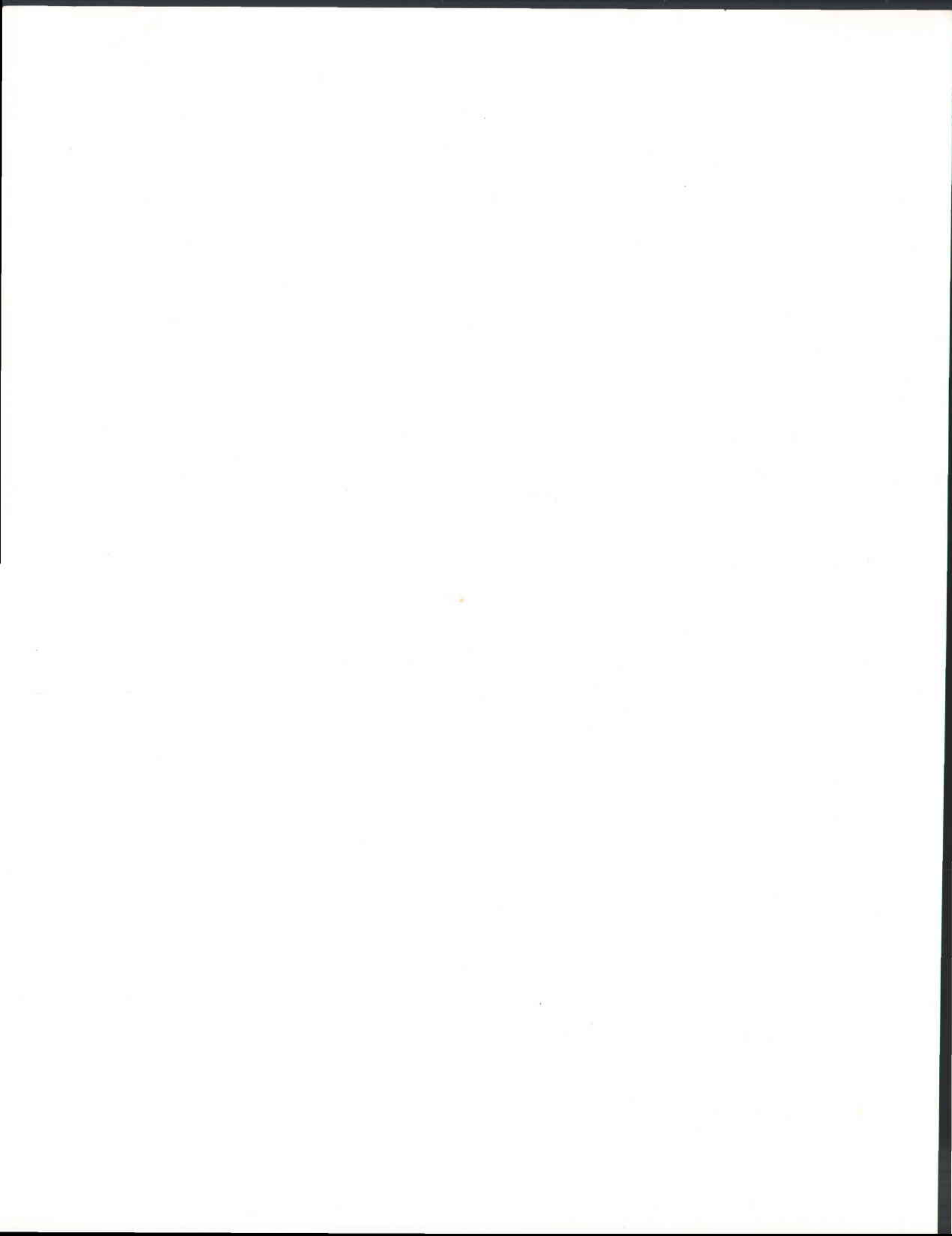
32216A-130 Same as 32216A but on 800 bpi magnetic tape.

**MANUFACTURING DIVISION:** GENERAL SYSTEMS DIVISION  
5303 Stevens Creek Blvd.  
Santa Clara, California 95050 U.S.A.



#### Richard E. McIntire

Dick McIntire has been with HP since 1969, serving as the first project manager for IMAGE and QUERY, laboratory section manager for data base management systems, and special projects manager in data systems marketing. He is currently conducting feasibility studies of in-house on-line data base applications using IMAGE. Dick spent three years in the U.S. Navy before enrolling at Arizona State University to study mathematics. He received his BA degree in 1964, and for the next five years designed and implemented computer programs for satellite navigation and oil-field applications. Resuming his studies after joining HP, he received his MS degree in mathematics in 1971 and the MBA degree in 1973, both from the University of Santa Clara. He's a member of ACM and the Mathematical Association of America, and an associate professor in computer science at California State University, Sacramento. Bachelor McIntire lives in Los Gatos, California, and enjoys hiking and flying in his spare time.





Sales and service from 172 offices in 65 countries.  
1501 Page Mill Road, Palo Alto, California 94304

# HP Archive

This vintage Hewlett-Packard document was  
preserved and distributed by

**[www.hparchive.com](http://www.hparchive.com)**

Please visit us on the web!

On-line curator: John Miles, KE5FX

[jmiles@pop.net](mailto:jmiles@pop.net)

