THE CONSTRUCTION OF A DATA BASE FOR THE
DEVELOPMENTAL STUDIES DEPARTMENT OF
ALBANY STATE COLLEGE

A THESIS
SUBMITTED TO THE FACULTY OF ATLANTA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

BY
JIMMY LEE RAMSEY

DEPARTMENT OF MATHEMATICAL SCIENCES

ATLANTA, GEORGIA
July 1982

R = 0 T = 59
TABLE OF CONTENTS

INTRODUCTION .................................................. 1

Chapter

I. A DESCRIPTION OF QUERY UPDATE .................. 3

II. CONSTRUCTION OF THE DATA BASE ................ 11
   Creation of the Subschemas and Data Files ...... 12
   Attributes of the Data Base ...................... 22
   The Data Dictionary ................................. 29

III. UTILIZATION OF THE DATA BASE .................. 37
   Exhibiting Information About the Subschema .... 38
   Displaying Actual Data in the Data Base ....... 43
   Updating the Database .............................. 45
   Generating Reports from the Data Base .......... 50

APPENDIX ..................................................... 54

BIBLIOGRAPHY ............................................... 60
INTRODUCTION

The construction of a data base to handle the data processing needs of the Developmental Studies Department (DSD) of Albany State College, Albany, Georgia, has been needed ever since that department came into existence. This is because of its size, the records it must keep on students, and the many complicated reports which it must submit to certain administrative offices from time to time. Up until the time of the construction of the data base described in this thesis, all of the information processing and the generation of reports was being done by hand. Needless to say, this required many man hours, and the need to ensure accuracy was often a very serious problem under these circumstances. The construction of a data base to solve the data processing needs of the DSD became imperative. This is the motivation of this thesis project.

When personnel, equipment needs, supplies, available funds, and the training of personnel to operate and maintain the data base had been carefully considered, it was decided to construct the data base using the Control Data Corporation (CDC) Cyber 730 computer of the University System of Georgia Computer Network of which Albany State
College is a node. It was further decided that the secretary of the DSD would be the person who would initially operate and maintain the data base. It was also decided that the operation and maintenance of the data base should not be of such complexity that the secretary would have to take a leave of absence from her job to acquire formal training in data processing in order to be able initially to operate and maintain the data base. Thus it was decided to give the secretary on-the-job training and to involve her as much as possible in the actual construction of the data base.

Having decided this, it was then decided to use the CDC Query Update interpretive system and the CDC Network Operating System (NOS) to implement the data base. The Query Update interpreter was chosen because its diversified structure lends itself to both professional computer programmers and nontechnical staff personnel. Further, it was chosen because of its capacity to rapidly generate reports.

It took about five months to construct the data base described in this thesis. This includes the time used in training the secretary. Work on the data base began in January, 1982 and proceeded through May, 1982. Plans to evaluate the effectiveness of the data base include keeping an accurate accounting of man hours used in the generation of reports as compared to those previously used.
CHAPTER I

A DESCRIPTION OF QUERY UPDATE

Because of the major role which Query Update plays in this project, a description of it is presented at this point. The description presented here is based on the Control Data Corporation Query Update Version 3 Reference Manual. This description, however, is not intended to be a full and complete description of Query Update; it focuses primarily on those parts of Query Update which are relevant to this project to achieve its objectives.

Query Update is an interpretive system which performs data storage and data retrieval operations. It has the capacity to update, delete, sort, selectively retrieve, display, and to generate reports from the data records which are stored in the data base. It also has the capacity to add new records to the data base. These operations are performed through commands called directives; Query Update has 65 such directives and options.

Query Update can operate within the CDC DMS-170 System or within the Information Management Facility of CDC. DMS-170 was chosen due to its availability. The overall description of the format of this data base was effected using a Query Update subschema definition. A
schema definition would have required the use of the Cyber Database Control System (CDCS) data base access mode. The Cyber Record Manager (CRM) data base access mode was desired for use with this data base due to its availability and for the purpose of making files generated from the data base available for use by high level languages under the NOS operating system.

In setting up the data base, the Query Update subschema must first be described using the CDC Data Description Language for the Query Update Subschema (DDL). This is accomplished by entering TEXT mode under the NOS operating system and creating a file which must contain the source code of the subschema. This code must then be compiled for actual use. When compiled, the binary code generated is placed on a permanent file to be used in accessing the data base for any reason.

After the subschema has been compiled and saved, a direct access permanent file is then created, using the NOS operating system for storing the records of the data base. When this file has been created, the Query Update system is then entered, and the CREATE directive is issued to establish the data base file for subsequent insertion of data. It determines whether subsequent directives are performed in the CRM data base access mode or in the CDCS data base access mode.
After the CREATE directive has been issued, the INSERT directive is then issued to create a record entry and place it in the data base file. The names of the fields of the record to be entered immediately follow the USING option of the INSERT command. Once these field names have been entered, the actual data items of a record may then be entered. When all the fields of a particular record have been entered, a user will then depress the RETURN key. Upon depressing the RETURN key, another data record may be entered. This process will continue until the user types *END to indicate to Query Update that he has finished entering records in the data base at this time. At this point, the data base has been created. A user would then enter END in order to exit Query Update. Subsequent records may be entered at a later time.

In order to use the data base, a user would enter Query Update, attach the data file in write mode, and proceed to issue Query Update directives. Of the many directives and options of Query Update the following ones are more prominent.

**EXHIBIT**

This directive displays certain information about the subschema.

**DISPLAY**

This directive displays data in various formats.
REWIND

This directive creates a temporary file during report preparation to hold the intermediate forms of the report.

IF

This directive permits the user to selectively manipulate the data.

DELETE

This directive permits a user to delete records from the data base.

EXTRACT

This directive permits a user to extract data from the data base and place it upon a selected file.

FORMAT

This directive permits a user to format a file during report preparation.

USE

This directive permits a user to select certain fields to be used in data manipulation.

INSERT

This directive permits a user to insert data in the fields of a record.
SORT
This directive permits a user to sort a file.

HELP
This directive enables a user to list the description of a directive or the explanation of a diagnostic message. It can also be used to list all available Query Update directives.

MOVE
This directive places a value in a data item.

OS
This directive allows a user to enter operating system control statements during a Query Update session.

TITLE
This directive is a report directive which permits a user to insert titles on the pages of reports.

UPDATE
This directive permits the user to identify a file for updating.

VETO
This directive causes data to be displayed at the terminal whenever a record is being deleted, inserted, or updated.
Examples illustrating the use of these commands are given in other sections of this thesis.

The directives listed above are said to be prominent because they permit most beginning users of Query Update to generate data from the data base in a fairly easy manner. Thus this set of directives should be mastered first.

A user may communicate with Query Update in interactive mode or in batch mode. Interactive mode permits a user to communicate with Query Update using a terminal keyboard or a pre-defined file. Batch mode permits a user to submit transmissions to Query Update on punched cards. The interactive mode is the mode which the users of the data base described in this thesis are presently using to communicate with Query Update. Each terminal used has a small printer attached for the purpose of receiving hard copy output.

The record managing tasks of the data base are handled by the Cyber Record Manager through an interface with Query Update. The information regarding file organization, file limit, index padding, data padding, and index level is used by the Cyber Record Manager when extracted from the Query Update subschema to set up tables and to set appropriate parameters for various input/output operations performed on the files in the data base. This information is also used by the Cyber Record Manager to
initiate various macro calls which are used in searching the data base files.

The interface which Query has with the NOS operating system is extremely important for the operation of the data base. It permits a user to issue NOS commands while operating in Query Update. This is important in that a user may wish to handle an urgent task involving files and processes which operate under the NOS operating system without having to leave Query Update and undergo the unpleasant task of having to reinitiate a process which he may have nearly completed. Files which are created while in Query Update may be manipulated, edited, and saved under the NOS operating system. The interface with the NOS operating system also permits a user to cause output generated by Query Update to be printed on a high speed printer of the host computer or on any of the high speed printers of any node in the network. This is accomplished using the ROUTE command of NOS.

Many times when a user is logged in under Query Update, the operator of the host computer may send out a bulletin which is of an urgent nature, such as, informing all users that the host computer will be going down within a short period of time. If this should happen, it may become necessary for a user to communicate directly with the operator to ensure that certain matters are properly handled. The NOS interface with Query Update handles this.
Frequently, the operator of the host computer may wish to communicate directly with a particular user who may be logged in under Query Update. This may be necessary to correct certain problems which are peculiar to that user. If such an occasion should arise, such communication can be effected through the NOS interface without the user having to exit Query Update.

In using Query Update in the interactive mode for the purpose of creating a data base or in updating the data base, the QU control statement, with or without any of its optional parameters, must be submitted by a user in order to establish communications with the Query Update interpreter. See the Appendix for the format of the QU statement.
CHAPTER II

CONSTRUCTION OF THE DATA BASE

The data base consists of two files which were formed using two Query Update subschemas. It was initially decided to have one major file but this idea was abandoned in order to provide for ease of use in entering the records in the data base. Each record in the data base is a student record which contains data items which are necessary to generate numerous reports of the Developmental Studies Department of Albany State College. The first file consists of data which is used very frequently, and the second file consists of data which is used less frequently. Therefore a student's record is spread across the two files. The student’s ID number identifies each segment of his record in the two files.

Before a file can be formed, it is first necessary to define a Query Update subschema for the file. This can be done using the CDC Text Editor of the NOS operating system. The CDC Data Description Language for the Query Update Subschema is used to describe the subschema. After having described the subschema, it is then necessary to have the DDL compiler to compile the subschema. After the subschema has been compiled, Query Update is then entered, 11
and the CREATE and INSERT directives of Query Update are then issued in order to insert records in the data base files. The following sequence details how the subschemas and data base files for this project were created using the facilities indicated above.

Creation of the Subschemas and Data Files

In the material which follows, the numbers which appear in parentheses at the right margin are not a part of the commands, directives, or statements given on the same line; they are to be used in a subsequent discussion to refer to the commands, directives, or statements which appear on the same line with them to explain their meaning or operation.

Creation of Subschema # 1 and Its Data File

The NOS operating system was entered in interactive mode and the following commands were issued in response to the NOS RECOVER/SYSTEM query. Note that NOS generated lines are underlined. Those issued by the user are not. BATCH refers to a general interactive user mode of NOS.

RECOVER/SYSTEM BATCH (1)
OLD, NEW, OR LIB FILE NEW, TAQUTE (2)
READY TEXT (3)
ENTER TEXT MODE

After the NOS response ENTER TEXT MODE, the following sub-schema was then entered:

IDENTIFICATION DIVISION.

SUB-SCHEMA NAME IS TUESTUD.

DATA DIVISION.

AREA-NAME IS TRENDAEES

ORGANIZATION IS INDEXED KEY IS STD-ID

FILE-LIMIT IS 7000

DATA-PADDING IS 20 PERCENT

INDEX-PADDING IS 20 PERCENT

INDEX-LEVEL IS 4

RECORD-NAME IS STD-RECORD

02 STD-ID PICTURE X(5)

02 STD-NAME

03 LAST-NAME PICTURE X(10)

03 FIRST-NAME PICTURE X(10)

03 MIDDLE-NAME PICTURE X(10)

02 SEX PICTURE X(10)

02 RACE PICTURE X(1)

02 CUR-QTR PICTURE X(3)

02 SAT-VERBAL-SCORE PICTURE 9(3)

02 SAT-MATH-SCORE PICTURE 9(3)

02 BSE-PRETEST-READING-SCORE PICTURE Z(3)

02 BSE-PRETEST-ENGLISH-SCORE PICTURE Z(3)
Upon completion of the entering of this first sub-schema (TUESTUD), the following commands were entered to exit from TEXT mode and to save the sub-schema on a permanent file named TAQUTE:

\( (CTRL): T \) \( (41) \)

The NOS operating system then responded with the following:

\( \text{EXIT TEXT MODE} \) \( (42) \)

Then the following command was entered:

\( \text{SAVE, TAQUTE} \) \( (43) \)

At this point, the creation of the first sub-schema is
complete. The subschema is now ready to be compiled and used for insertion of records into the first file. The following sequence of commands effects this.

BATCH

DEFINE(TUDENT=TUESTUD/CT=P,M=R)  \(44\)

DDL,QD,I=TAQUTE,SB=TTUDENT  \(45\)

Upon the execution of \(46\) above, the DDL compiler then compiled the subschema, and the following command was then issued to return the temporary file TTUDENT to the NOS operating system:

RETURN,TTUDENT.  \(47\)

At this point, the permanent file on which the student records reside was created with the following command:

DEFINE(TRENDAE/CT=P,M=W).  \(48\)

Then Query Update was entered for the purpose of creating the file with the command

QU\(^{-}\).  \(49\)

Query Update then responded with two dashes and a question mark (command mode for Query Update):

--  \(50\)

?  \(51\)

The Query Update directive to create the file was then issued:

CREATE TRENDAEES OF TUESTUD  \(52\)
Query Update then responded with two dashes and a question mark:

```
--
?
```

(53) (54)

The Query Update directive INSERT was then issued to permit the entering of data into the data base (Note that the symbol + indicates continuation of the command to the following line):

```
INSERT USING STD-ID, LAST-NAME, FIRST-NAME, +
MIDDLE-NAME, SEX, RACE, CUR-QTR, SAT-VERBAL- +
SCORE, SAT-MATH-SCORE, BSE-PRETEST-READING- +
SCORE, BSE-PRETEST-ENGLISH-SCORE, BSE- +
PRETEST-MATH-SCORE, BSE-PRETEST-ARITH-SCORE, +
BSE-POSTTEST-READING-SCORE, BSE-POSTTEST- +
BSE-POSTTEST-ENGLISH-SCORE, BSE-POSTTEST- +
MATH-SCORE, BSE-POSTTEST-ARITH-SCORE, +
DIAGNOSTIC-READING-PRETEST-ND, +
DIAGNOSTIC-READING-POSTTEST-ND, MATH- +
PLACEMENT-PRETEST, MATH-PLACEMENT-POSTTEST, +
CUR-DSD-MATH-CLASS, CUR-DSD-ENGLISH-CLASS, +
CUR-DSD-READING-CLASS, SOC-SEC-NO
```

(55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67)

Upon the completion of the issuance of the INSERT directive, Query Update responded with the following (Note that the symbol >> indicates insert mode):

```
>>
```

(68)
The insert mode symbol >> is a prompt for the user to enter a record into the data base. Student records were then entered into the data base. Non-numeric items are delimited by the dollar mark ($). Numeric items are entered normally. Items are separated by single spaces. The entry of each record in the data base causes the insert prompt >> to be regenerated.

Records were entered during this initial session in response to the prompt >> until the following directive was given to terminate insert mode:

*END

(69)

Query Update then responded with the following:

--

? (70)

The command to exit from Query Update was then given as follows:

END

(71)

The NOS operating system then responded with its command mode prompt:

/ (72)

The initial file for Subschema # 1 was then complete.

Creation of Subschema # 2 and its Data File

The creation of Subschema # 2 is accomplished in exactly the same manner as for the creation of Subschema
# 1, the only difference being the file names and the data fields. The following is a listing of Subschema # 2:

IDENTIFICATION DIVISION.

SUB-SCHEMA NAME IS WUESTUD.

DATA DIVISION.

AREA-NAME IS WRENDABES.

ORGANIZATION IS INDEXED KEY IS STD-ID

FILE-LIMIT IS 7000

DATA-PADDING IS 20 PERCENT

INDEX-PADDING IS 20 PERCENT

INDEX-LEVEL IS 4

RECORD-NAME IS INFM-RECORD

02 STD-ID PICTURE X(5)

02 QTR-1 PICTURE X(3)

02 QTR-2 PICTURE X(3)

02 QTR-3 PICTURE X(3)

02 QTR-4 PICTURE X(3)

02 QTR-5 PICTURE X(2)

02 ENROLLMENT-STATUS PICTURE X(2)

02 ADMISSION-STATUS PICTURE X(2)

02 WITHDRAWAL-REASON PICTURE Z(2)

02 MAJOR PICTURE X(3)

02 GRADE-IN-110-MATH PICTURE X(2)

02 GRADE-IN-131-MATH PICTURE X(2)
02 GRADE-IN-201-SOCIOLOGY PICTURE X(2)
02 GRADE-IN-111-BIOLOGY PICTURE X(2)
02 GRADE-IN-112-BIOLOGY PICTURE X(2)
02 GRADE-IN-101-HISTORY PICTURE X(2)
02 GRADE-IN-102-HISTORY PICTURE X(2)
02 GRADE-IN-101-ENGLISH PICTURE X(2)
02 GRADE-IN-102-ENGLISH PICTURE X(2)
02 GRADE-IN-097-MATH PICTURE X(2)
02 GRADE-IN-098-MATH PICTURE X(2)
02 GRADE-IN-099-MATH-PICTURE X(2)
02 GRADE-IN-021-READING PICTURE X(2)
02 GRADE-IN-022-READING PICTURE X(2)
02 GRADE-IN-023-READING PICTURE X(2)
02 GRADE-IN-097-ENGLISH PICTURE X(2)
02 GRADE-IN-098-ENGLISH PICTURE X(2)
02 GRADE-IN-099-ENGLISH PICTURE X(2)
02 EXIT-CODE-1 PICTURE 9(1)
02 EXIT-CODE-2 PICTURE 9(1)
02 EXIT-CODE-3 PICTURE 9(1)
02 EXIT-CODE-4 PICTURE 9(1)
02 ADDRESS
  03 NUMBERS PICTURE Z(5)
  03 STREET PICTURE X(20)
  03 CITY PICTURE X(15)
  03 STATE PICTURE X(2)
  03 ZIP-CODE PICTURE 9(5)
Sub-Schema # 2's name is WUESTUD; it resides on a permanent file named WAQUTE. Its data file name is WRENDAE. During the time of its compilation by the DDL compiler, Sub-Schema WUESTUD'S binary code was placed on a temporary file named WTUDENT.

Upon completion of entering records in the data base, the user is now ready to either manipulate the data, extract some information from the data base, create some procedure files for the data base, prepare some formal reports from the data base, or to log out. In the event the user wishes to log out, he simply types BYE while in BATCH mode.

Now that the process for creating the sub-schemas and the files of the data base has been described, a full discussion of its structure, and an explanation of some of the statements used in its creation is appropriate at this point. The line numbers used in this discussion refer to those numbers which appeared earlier in parentheses in the right hand margin of this section.

Sub-Schema # 1 is listed in lines (5) through (40); its name is given in line (6). Line (8) gives the name of the file on which data is to be stored and formatted according to the sub-schema TUESTUD. Line (9) indicates how the file is organized; it is an indexed sequential file and its key is STD-ID, i.e., the student's ID number. Line (10) indicates the maximum number of records that the
file can hold.

Line (11) indicates the percentage of the data block to be reserved for padding during file creation. Data padding is also used to permit record expansion without increasing input/output time.

Line (12) indicates the percentage of the index block to be reserved for padding to allow for future addition of key entries. The main purpose for padding is to permit file growth without requiring additional index levels.

Line (13) is used by the system along with the maximum number or records given in the FILE LIMIT statement to determine the size of the index blocks.

The remainder of the lines in the sub-schema, (14) through (40), were constructed in the same manner that a record in a COBOL program would be constructed; these lines describe the data record in terms of its data fields. It is to be noted that an effort was made to describe each data field as clearly as possible to provide for ease in entering these data fields when the INSERT directive is used.

In typing the lines of the sub-schema at the terminal, lines (5) through (14) begin in column 8. Lines (15) through (16) and lines (20) through (40) begin in column 12. Lines (17) through (19) begin in column 16.
In the picture clauses given in lines (15) through (32) and lines (35) through (36), the number enclosed in parentheses indicates the width of each data field; a 9 appearing to the left of a parenthesis indicates that that particular data field must be a pure number; an X appearing to the left of a parenthesis indicates that that particular data field may be a pure number or alphabetic data or alphanumeric data; a Z appearing to the left of a parenthesis indicates that that particular data item is a pure number and must be zero suppressed. Each 9 appearing to the left of a V and to the right of a V denote a decimal digit, and the V denotes the placement of the decimal point, and the data field is therefore a pure number.

Attributes of the Data Base

In designing the data base, it was desired that the data base should have certain attributes. A real effort was made to realize these attributes in the data base. Some of these attributes are discussed below.

Privacy

Privacy for the data base files is provided for in line (45) by the parameter CT=P which indicates that the file is a private file, thereby precluding access to it by unauthorized users.
Protection of the Data Files from Loss or Damage

In order to protect the data files from machine failures, incompetents, vandals, criminals, and other persons who may falsely update them, the data files have been copied to a reel of tape. Upon updating the database, the tape file is also updated.

Controlled Redundancy

As is evidenced by the two subschemas, no data item is listed more than once except the student ID number which is listed only once on each file for the purpose of identifying each segment of a student's record. The listing of all the other data items only once was done to provide for efficiency and to cut storage costs.

Fast Access

The records in the data base are managed by the CDC Cyber Record Manager (CRM) which interfaces with Query Update and the NOS operating system. CRM provides the data base with extremely fast data access, storage and record handling capabilities.

Standardization of Data

The codes which are used in this data base are consistent with the codes which are used with the offices of the Institution which interface with the DSD.
Integrity Controls

Controls on the integrity of the data is provided for by Query Update and the NOS operating system. When a numeric value is out of the normal range or if the wrong type of data is submitted by a user to be put in a specific data field, Query Update immediately disallows this and so informs the user with a diagnostic.

Fast Recovery from Failures

The design of Query Update and its interface with the NOS operating system and the Cyber Record Manager makes provision for recovery from failures. The NOS operating system gives a user a certain time limit to log back in and resume processing should he be prematurely disconnected from the computer due to power failures or other interferences. If a user is prematurely disconnected from the computer during a given session, he may within 10 minutes log back in and issue the RECOVER command along with the terminal number which he was assigned by the NOS operating system at the time of the premature disconnection.

Monitoring Aids

The NOS operating system has provisions for the monitoring of the utilization of the data base. This may be done through the use of the DAYFILE facility. The DAYFILE facility keeps a record of each NOS command
which a user issues when connected to the computer; it
also keeps a record of the diagnostics which the computer
issues to a user. A user may save the DAYFILE in his
library for later examination by himself or by system
analysts who will have access to it at the central site
by issuing the following commands prior to logging out:

DAYFILE,ERROR

REPLACE,ERROR

A user may also save a copy of his session with Query
Update by entering the following command prior to log-
ging out:

REPLACE,ZZZZZQU

The file named ZZZZZQU is a default file which is used
by Query Update to record all directives and diagnostics
during a Query Update session.

High Level Programmer Interface

The manner in which this data base interfaces with
high level programmers is through Query Update. Query
Update will permit a programmer to extract the data
which he needs from a file and place it on another file
using the EXTRACT directive of Query Update. The pro-
grammer may then exit from Query Update and save the
file containing the extracted information with the SAVE
command of the NOS operating system. The file may then
be accessed by high level languages. It may also be
modified using the Text Editor of the NOS operating system.

Physical Data Independence

The data base is presently residing on the storage devices which are connected to the CDC Cyber 730 computer of the University System of Georgia Computer Network. A copy of it is also stored on a reel of tape. The changing of the devices on which the data base presently resides will not require that application programs be re-written.

Fast Searching

Searching for the records in the data base is provided for by the Cyber Record Manager through its interface with Query Update. The Cyber Record Manager is extremely efficient in locating a given record.

Quick Handling of Unanticipated Requests

Unanticipated spontaneous requests for data can be handled without application programs having to be written by use of the Query Update language.

Availability of Data

The data in this data base are readily available to a user at any time that it is needed.
Multiple Uses of the Data

Although the data may be perceived differently by different users, its residence on the storage devices remains the same. Different users may extract data from the data base and place it on different storage devices for use in any manner that is desired.

Protection of Intellectual Investment

When applications programs are written, these programs may be saved in the library for the data base or they may be saved on tape. With the data base's capacity to generate files which application programs access, modification of the data base will not require rewriting of application programs.

Clarity

A user of the data base is made aware of the data which is available to him by the data dictionary and the two sub-schemas.

Flexible Usage

The data base can be searched in flexible ways using the conditional IF directive of Query Update.

Less Data Proliferation

With the data base's capacity to generate files and immediately purge them from storage, there is no need for programmers to have to create large files which must be
stored on tape or other devices. This eliminates many problems which are associated with data storage, including cost.

Performance

With the aid of the Cyber Record Manager, requests through Query Update are handled almost instantaneously.

Accuracy and Consistency

The data base's needs for accuracy and consistency are met in part by the DDL compiler. At the time that the sub-schemas are compiled, if a duplicate data item name appears, a diagnostic is issued, and the compiler will discontinue compilation of the sub-schemas. Also, the Query Update interpreter, in its execution of the INSERT command, will not accept duplicate data item names; further, to ensure accuracy, it will not accept a data item name which does not agree exactly with the data item named in the sub-schemas.

Low Cost

The low cost to operate the data base is due to controlled redundancy and the fact that no new personnel had to be hired or a change in job title with a corresponding salary increase had to be made.
The Data Dictionary

The following is a complete description of all of the data items contained in each file of each sub-schema of the data base:

Subschema Name: TUESSTUD
File Name: TRENDAE
Record Name: STD-RECORD

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Type</th>
<th>Field Length</th>
<th>Definition of Codes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD-ID</td>
<td>AFN</td>
<td>5</td>
<td>Code</td>
<td>Student ID number</td>
</tr>
<tr>
<td>STD-NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST-NAME</td>
<td>AFN</td>
<td>10</td>
<td>Code</td>
<td>Student last name</td>
</tr>
<tr>
<td>FIRST-NAME</td>
<td>AFN</td>
<td>10</td>
<td>Code</td>
<td>Student first name</td>
</tr>
<tr>
<td>MIDDLE-NAME</td>
<td>AFN</td>
<td>10</td>
<td>Code</td>
<td>Student middle name</td>
</tr>
<tr>
<td>SEX</td>
<td>AFN</td>
<td>1</td>
<td>M</td>
<td>Male</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td>Female</td>
</tr>
<tr>
<td>RACE</td>
<td>AFN</td>
<td>1</td>
<td>1</td>
<td>Black American</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>Caucasian American</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>African</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>Indian</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>Chinese</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>Other</td>
</tr>
<tr>
<td>CUR-QTR</td>
<td>AFN</td>
<td>3</td>
<td>1XX</td>
<td>1=Fall Qtr; 2=Win. Qtr; 3=Spr. Qtr; 4=Sum Qtr; XX=Year</td>
</tr>
<tr>
<td>SAT VERBAL-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>SAT verbal score</td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Length</td>
<td>Definition of Codes</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>---------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>SAT-MATH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>SAT math score</td>
</tr>
<tr>
<td>BSE-PRETEST-READING-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE pretest reading score</td>
</tr>
<tr>
<td>BSE-PRETEST-ENGLISH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE pretest English score</td>
</tr>
<tr>
<td>BSE-PRETEST-MATH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE math pretest score</td>
</tr>
<tr>
<td>BSE-PRETEST-ARITH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE pretest arithmetic score</td>
</tr>
<tr>
<td>BSE-POSTTEST-READING-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE posttest reading score</td>
</tr>
<tr>
<td>BSE-POSTTEST-ENGLISH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE posttest English score</td>
</tr>
<tr>
<td>BSE-POSTTEST-MATH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE posttest math score</td>
</tr>
<tr>
<td>BSE-POSTTEST-ARITH-SCORE</td>
<td>N</td>
<td>3</td>
<td></td>
<td>BSE posttest arithmetic score</td>
</tr>
<tr>
<td>DIAGNOSTIC-READING-PRETEST-ND</td>
<td>N</td>
<td>4</td>
<td></td>
<td>Diagnostic reading pretest score</td>
</tr>
<tr>
<td>DIAGNOSTIC-READING-POSTTEST-ND</td>
<td>N</td>
<td>4</td>
<td></td>
<td>Diagnostic reading posttest score</td>
</tr>
<tr>
<td>MATH-PLACEMENT-PRETEST</td>
<td>N</td>
<td>3</td>
<td></td>
<td>Math placement pretest score</td>
</tr>
<tr>
<td>MATH-PLACEMENT-POSTTEST</td>
<td>N</td>
<td>3</td>
<td></td>
<td>Math placement posttest score</td>
</tr>
<tr>
<td>CUR-DSD-MATH-CLASS</td>
<td>N</td>
<td>6</td>
<td></td>
<td>Current DSD math class</td>
</tr>
<tr>
<td>CUR-DSD-ENGLISH-CLASS</td>
<td>N</td>
<td>6</td>
<td></td>
<td>Current DSD English class</td>
</tr>
<tr>
<td>CUR-DSD-READING-CLASS</td>
<td>N</td>
<td>6</td>
<td></td>
<td>Current DSD reading class</td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Length</td>
<td>Definition of Codes</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>--------------</td>
<td>---------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>SOC-SEC-NO</td>
<td>AFN</td>
<td>11</td>
<td></td>
<td>Social Security number</td>
</tr>
</tbody>
</table>

Subschema Name: WUESTUD

File Name: WRENDAE

Record Name: INFM-RECORD

DATA RECORD

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Type</th>
<th>Field Length</th>
<th>Definition of Codes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD-ID</td>
<td>AFN</td>
<td>5</td>
<td></td>
<td>Student ID number</td>
</tr>
<tr>
<td>QTR-1</td>
<td>AFN</td>
<td>3</td>
<td>1XX</td>
<td>1=Fall Qtr; XX=yr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2XX</td>
<td>2=Win. Qtr; XX=yr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3XX</td>
<td>3=Spr. Qtr; XX=yr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4XX</td>
<td>4=Sum. Qtr; XX=yr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>This field denotes the first quarter of enrollment in the DSD.</td>
</tr>
<tr>
<td>QTR-2</td>
<td>AFN</td>
<td>3</td>
<td>Same as above</td>
<td>This field denotes the second quarter of enrollment in the DSD. The codes are the same as above</td>
</tr>
<tr>
<td>QTR-3</td>
<td>AFN</td>
<td>3</td>
<td>Same as above</td>
<td>This field denotes the third quarter of enrollment in the DSD. The codes are the same as above</td>
</tr>
<tr>
<td>QTR-4</td>
<td>AFN</td>
<td>3</td>
<td>Same as above</td>
<td>This field denotes the fourth quarter of enrollment in the DSD. The codes are the same as above</td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Length</td>
<td>Definition of Codes</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>--------------</td>
<td>----------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>QTR-5</td>
<td>AFN</td>
<td>3</td>
<td>Same as the fifth quarter above of enrollment in the DSD. The codes are the same as above</td>
<td></td>
</tr>
<tr>
<td>ENROLLMENT-STATUS</td>
<td>AFN</td>
<td>2</td>
<td>01 Currently enrolled in the DSD and in good standing in the DSD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>02 Currently enrolled in the DSD but not in good standing with the DSD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03 Not currently enrolled in the DSD but in good standing with the DSD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>04 Not currently enrolled in the DSD and not in good standing with the DSD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>05 Not currently enrolled in the College but in good standing with the College.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>06 Not currently enrolled in the College and not in good standing with the College.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>07 Currently enrolled in the College and in good standing with the College.</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Length</td>
<td>Definition of Codes</td>
<td>Code</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>--------------</td>
<td>---------------------</td>
<td>------</td>
</tr>
<tr>
<td>ADMISSION-STATUS</td>
<td>AFN</td>
<td>2</td>
<td>08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>09</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>WITHDRAWAL-REASON</td>
<td>N</td>
<td>2</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>06</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>06</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>08</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Length</td>
<td>Definition of Codes</td>
<td>Code</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>--------------</td>
<td>---------------------</td>
<td>------</td>
</tr>
<tr>
<td>MAJOR</td>
<td>AFN</td>
<td>3</td>
<td>DSD</td>
<td>DSD</td>
</tr>
<tr>
<td>GRADE-IN-110-MATH</td>
<td>AFN</td>
<td>2</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>GRADE-IN-131-MATH</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-201-SOCIOLOGY</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-111-BIOLOGY</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-112-BIOLOGY</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-101-HISTORY</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-102-HISTORY</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>GRADE-IN-101-ENGLISH</td>
<td>AFN</td>
<td>2</td>
<td>Same as above</td>
<td>Same</td>
</tr>
<tr>
<td>Field Name</td>
<td>Type</td>
<td>Length</td>
<td>Code</td>
<td>Definition of Codes</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>GRADE-IN-102-ENGLISH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-097-MATH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-098-MATH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-099-MATH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-021-READING</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-022-READING</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-023-READING</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-097-ENGLISH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-098-ENGLISH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>GRADE-IN-099-ENGLISH</td>
<td>AFN</td>
<td>2</td>
<td>Same</td>
<td>Same as above as above</td>
</tr>
<tr>
<td>EXIT-CODE-1</td>
<td>N</td>
<td>1</td>
<td>1</td>
<td>Completed all DSD requirements</td>
</tr>
<tr>
<td>EXIT-CODE-2</td>
<td>N</td>
<td>1</td>
<td>2</td>
<td>Exit DSD math</td>
</tr>
<tr>
<td>EXIT-CODE-3</td>
<td>N</td>
<td>1</td>
<td>3</td>
<td>Exit DSD reading</td>
</tr>
<tr>
<td>EXIT-CODE-4</td>
<td>N</td>
<td>1</td>
<td>4</td>
<td>Exit DSD English</td>
</tr>
<tr>
<td>Field Name</td>
<td>Type</td>
<td>Length</td>
<td>Code</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>--------</td>
<td>--------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>ADDRESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBERS</td>
<td>N</td>
<td>5</td>
<td></td>
<td>Street number</td>
</tr>
<tr>
<td>STREET</td>
<td>AFN</td>
<td>20</td>
<td></td>
<td>Street</td>
</tr>
<tr>
<td>CITY</td>
<td>AFN</td>
<td>15</td>
<td></td>
<td>City</td>
</tr>
<tr>
<td>STATE</td>
<td>AFN</td>
<td>2</td>
<td></td>
<td>State; a two letter code prepared by the U.S. Gov.</td>
</tr>
<tr>
<td>ZIP-CODE</td>
<td>N</td>
<td>5</td>
<td></td>
<td>Zip codes as prepared by the U.S. Gov.</td>
</tr>
</tbody>
</table>
CHAPTER III

UTILIZATION OF THE DATA BASE

The data base is accessed through Query Update. To access the data in file TRENDAE of sub-schema TUESTUD, a user would log in under the BATCH subsystem of the NOS operating system and enter the following command:

QU

This is the call for the Query Update interpreter. Query Update would then respond with its identification, and then it would issue the following prompts:

--

? 

At this point, the user may enter the following Query Update directive:

USE TRENDAEES(M=W) OF TUESTUD.

The file TRENDAE is here referenced using two additional characters (ES), the same two characters which were used in the CREATE directive. The use of the two additional characters in Query Update directives relative to files created in the manner previously described is a peculiarity of the Query Update interpretive system. The (M=W) parameter directs Query Update to attach the file in write mode. Upon receiving the above USE directive,
Query Update responds with the following:

--

?

The user may now enter any valid Query Update directive; there are 65 of them.

Exhibiting Information About the Sub-Schema

If a user of the data base wishes to have the Query Update information which gives the characteristics of Sub-schema TUESTUD, he would enter the following directive:

EXHIBIT

When this directive is given, Query Update responds with the following:

MAXIMUM TRANSMISSION LENGTH   1030
TL OF CATALOG FILE   1030
SEPARATOR $  
UNIVERSAL OFF
MAX NUMBER OF LINES   060
MAX NUM. OF COLUMNS   136
MAX NO. OF SECTIONS   010
MAX IMAGES PER PAGE   004

AREA NAME(S):

TRENDAEES

SUBSCHEMA NAME = TUESTUD

SUBSCHEMA LIBRARY NAME = TUESTUD

--

?
In the above block of information, MAXIMUM TRANSMISSION LENGTH refers to the total number of characters which may be transmitted to Query Update via the terminal in any one transmission. UNIVERSAL OFF indicates that no universal character is to be recognized by Query Update which would mark a character position to be ignored during comparing and testing. SEPARATOR $ indicates that the character $ is to immediately precede and immediately follow any alphabetic or alphanumeric data field, that is, it delimits these types of data fields. The maximum number of lines, columns, images per page, and sections are indicated by the MAX statements. The AREA NAME is the file name (in this case TRENDAEES). Both the SUBSCHEMA name and the SUBSCHEMA LIBRARY NAME are coded as TUESTUD.

A user may have Query Update display the record name, key, area name (file name), and mode in which the database file TRENDAEES is attached by entering the following directive:

```
EXHIBIT TRENDAEES
```

Query Update then would respond as follows:

```
RECORD NAME IS STD-RECOR
KEY IS STD-ID
AREA PF NAME = TRENDAEES
   M = W
--
?
```
If a user wishes to see the names of the fields in, say, the record named STD-RECORD, he enters the following directive:

EXHIBIT STD-RECORD

Query Update, in response to the above directive, produces the following:

02 (KEY) STD-ID
02 (GRP) STD-NAME
03 (ELM) LAST-NAME
03 (ELM) FIRST-NAME
03 (ELM) MIDDLE-NAME
03 (ELM) SEX
02 (ELM) RACE
02 (ELM) CUR-QTR
02 (ELM) SAT-VERBAL-SCORE
02 (ELM) SAT-MATH-SCORE
02 (ELM) BSE-PRETEST-READING-SCORE
02 (ELM) BSE-PRETEST-ENGLISH-SCORE
02 (ELM) BSE-PRETEST-MATH-SCORE
02 (ELS) BSE-PRETEST-ARITH-SCORE
02 (ELM) BSE-PRETEST-ARITH-SCORE
MORE... ANSWER Y OR N)
? Y
02 (ELM) BSE-POSTTEST-READING-SCORE
02 (ELM) BSE-POSTTEST-ENGLISH-SCORE
02 (ELM) BSE-POSTTEST-MATH-SCORE
02 (ELM) BSE-POSTTEST-ARITH-SCORE
02 (ELM) DIAGNOSTIC-READING-PRETEST-ND
02 (ELM) DIAGNOSTIC-READING-POSTTEST-ND
02 (ELM) MATH-PLACEMENT-PRETEST
02 (ELM) MATH-PLACEMENT-POSTTEST
02 (ELM) CUR-DSD-MATH-CLASS
02 (ELM) CUR-DSD-ENGLISH-CLASS
02 (ELM) CUR-DSD-READING-CLASS
02 (ELM) SOC-SEC-NO

Note that these are the fields of STD-RECORD. The expression (KEY) indicates that the data name STD-ID is the key field; (GRP) indicates that the data item STD-NAME is a group field name; (ELM) indicates that the given field stores an elementary data item. After listing 14 data field names of the record, Query Update responded:

(MORE...ANYWER Y OR N)

Since a Y was then entered, the remaining 12 data item names of the record were listed. Had there been more than 12 data item names remaining, Query Update would have only printed up to 14 of them before inquiring as to whether or not to continue. This printing of groups of 14 data name items continues in this manner until all of them have been printed.

To exhibit any data field name of a record, the user
would enter the directive EXHIBIT followed by the name of the data field. The following are some instances of using the EXHIBIT directive to display some of the data field names from the data base. The underlined words indicates what the user entered, and those not underlined indicate the response by Query Update:

```
EXHIBIT STD-NAME
STD-NAME OF STD-RECORD
CONTAINS LAST-NAME
CONTAINS FIRST-NAME
CONTAINS MIDDLE-NAME
--
?
```

```
EXHIBIT SAT-VERBAL-SCORE
SAT-VERBAL-SCORE OF STD-RECORD
TYPE NUMERIC PIC SIZE 0003
--
?
```

```
EXHIBIT LAST-NAME
LAST-NAME OF STD-NAME OF STD-RECORD
TYPE CHAR ITEM PIC SIZE 0010
--
?
```
Displaying Actual Data in the Data Base

The DISPLAY directive permits the user to display actual data items in the data base. If a user wishes to display the last name, first name, and the SAT verbal score of all students who have records in the data base, he would enter the following directive:

DISPLAY LAST-NAME, FIRST-NAME, SAT-VERBAL-SCORE

Query Update responds as follows when the above directive was entered (Note: The following output data is test data):

JOHNSON     TIM     600
RAMSEY      LEE      900
JAMES       KING     400
RAMSEY      MARLENE  500
REESE       KENNY    300
PRIMER      JACKIE   900
PEARSON     KIM      600
PETERS      TISA     300
URSERY      LORI     200
EIFFORD     LESTER   100
MILLER      DANNY    400
MARTIN      KEVIN    500
BACHELOR    PAM      600

(MORE... ANSWER Y OR N)

?
If the user had entered Y, the next 14 names would have been listed, and this cycle would have been repeated until all of the names in the file are printed out. If the user had entered N, the listing of the names would have terminated. Note that Query Update lists fields in the order specified. It also determines the output format when given this type of request.

In addition to the above uses of the DISPLAY directive, a user may use the DISPLAY directive to extract data from the data base based on a key. The following instances illustrate how data was extracted from the data base using a key. The underlined words indicate what the user entered; the words not underlined indicate the responses by Query Update:

```
DISPLAY KEY $00020$ LAST-NAME, FIRST-NAME, MIDDLE-NAME
CRAFT CORNELIUS ROBERT
  1 ACCESSES, 1 HITS, 1 IO-S

DISPLAY KEY $00035$ FIRST-NAME
HOLLIS
  1 ACCESSES, 1 HITS, 1 IO-S

DISPLAY KEY $00025$ FIRST-NAME, MIDDLE-NAME, LAST-NAME, SEX
GWEN BERNICE BLOCKER F
  1 ACCESSES, 1 HITS, 1 IO-S
```
DISPLAY KEY $00020$ STD-ID.LAST-NAME,FIRST-NAME,MIDDLE-NAME,SOC-SEC-NO

00020 GRAFT CORNELIUS ROBERT 864-42-4132

1 ACCESSES, 1 HITS, 1 IO-S

--

?  

Thus, the user has great flexibility in displaying data from the data base.

**Updating the Data Base**

A user may update a record in the data base by changing any data item in that record without affecting any other data items. For example, if a user wishes to change the social security number of a student whose ID number is 30, he may enter the following directive:

```
UPDATE $00030$ MOVE $230-30-6787$ TO SOC-SEC-NO
```

When the above directive was given, Query Update responded as follows:

1 ACCESSES, 1 HITS, 2 IO-s

--

?  

At this point, the user may enter the DISPLAY directive to see whether or not the social security number of student number 30 was changed. When this was done, the social security number of student number 30 was changed.
There are several ways a user could add a new record to the data base. In one case, he may simply wish to enter just the ID number of a new student and no other information; in this case, he simply enters the following if the ID number is 4000:

```
INSERT $4000$
```
Provision would then be made for student 4000 in the data base; the actual information may be entered at a subsequent time.

If a user wishes to add a new record to the data base and the only information he has on a student is the ID number and the student's name, the user would enter the following:

```
INSERT $5000$ MOVE $JIM$ TO FIRST-NAME
```
In this case, only the student's first name would be entered into the data base along with his ID number 5000.

If a user wishes to add several new records where only the ID numbers and the last names of the students are known, the user would enter the following:

```
INSERT USING STD-ID,LAST-NAME
```
At this point, the user would be permitted to enter the actual ID numbers and names of the students into the data base.

If a user wishes to delete a record from the data base, say record number 1000, he simply enters the following:

```
DELETE $1000$
```
DELETE $1000$

The record would then be deleted from the data base.

Another way to delete a record from the data base gives the user VETO power, that is, it permits a user to change his mind if he decides not to delete a given record. To delete record 2000 with VETO power a user would enter the following:

DELETE $2000$ VETO

At this point, the first 40 characters of record 2000 would be displayed at the terminal along with the word VETO. If the user decides to go ahead with deleting the record, he would enter YES; otherwise, he would enter NO.

If a user wished to delete several records from the data base, he would enter the following:

DELETE USING STD-ID

At this point, the user would be permitted to delete as many records as he wishes, simply by entering the ID numbers of the students, one at a time. After deleting as many records as desired, the user would enter *END.

A user may selectively add, delete, and display the records in the data base; he may also selectively delete records with veto power. These selective operations are performed using the IF statement. For example, if a user wishes to print a list of all female students of the data base, he may enter the following:

IF SEX EQ $F$ DISPLAY FIRST-NAME,MIDDLE-NAME,LAST-
NAME

At this point, a list of the names of all females in the data base would be displayed at the terminal.

If a user wishes to obtain a list of all the male students from the state of Georgia, he would enter the following:

IF SEX EQ "$M" AND STATE EQ "$GA" DISPLAY FIRST-NAME, MIDDLE-NAME, LAST-NAME

At this point, the requested list would be printed. With a little imagination, a user may request various types of combinations of the data using the IF statement.

To display all of the students' last names whose ID numbers are less than 5000, a user would enter the following:

IF STD-ID LT "$5000" DISPLAY LAST-NAME

The requested list would then be printed at this point.

If a user wishes to list all students' last names whose ID numbers are greater than 6000, he would enter the following:

IF STD-ID GT "$6000" DISPLAY LAST-NAME

The requested list would then be printed at this point.

A user may request similar information based on the operators greater than or equal to, equal to, and less than or equal to. These codes for these operators are, GE, EQ, and LE.
If a user wishes to print a list of all student last names in the database who are from Georgia or from Florida, he would enter the following:

IF STATE EQ $GA$ OR $FL$ DISPLAY LAST-NAME

The requested information would be displayed at this time.

To delete a student's record using the IF statement and the VETO option (say student number 3000), a user would enter the following:

IF STD-ID EQ $3000$ DELETE VETO

At this point, the first 40 characters of record 3000 would be displayed along with the word VETO. If the user enters YES, the record will be deleted; if the user enters NO, the record will not be deleted.

To delete, say record 4000, using the IF statement but without the VETO option, a user would enter the following:

IF STD-ID EQ $4000$ DELETE

The database would be searched until record 4000 is found; then record 4000 would be deleted.

If a user wishes to change a student's middle name using the IF statement with the VETO option, he would enter the following (say, student 3000):

IF STD-ID EQ $3000$ UPDATE VETO MOVE $LEE$ TO MIDDLE-NAME

The first 40 characters of the record would be displayed at this time along with the word VETO. If the user enters
YES, the student's middle name will be changed to Lee; if NO is entered, the student's middle name will remain the same.

To use the IF statement to change a student's last name without VETO power (say, student number 3000), the user would enter the following:

IF STD-ID EQ "$3000$" UPDATE MOVE "$LEE$" TO LAST-NAME
The student's last name will be changed to Lee at this point.

Generating Reports From the Data Base

Generating reports from the data base is very easily accomplished using Query Update. If a user wishes to generate, say, an enrollment roster from the data base, he may enter the following sequence of directives to Query Update (Note: the symbols -- and ? are prompts generated by Query Update):

QU
--

? USE TRENAEES(M=W) OF TUESTUD
--

? REWIND FILE4
--

? EXTRACT UPON FILE4 LAST-NAME, FIRST-NAME, MIDDLE-NAME, SOC-SEC-NO
? FORMAT REPORT4
51

? DETAIL IS LAST-NAME IN COLUMN 10, FIRST-NAME IN COLUMN 25, MIDDLE-NAME IN COLUMN 40, SOC-SEC-NO IN COLUMN 55

--

? TITLE AT LINE 1 IS $ENROLLMENT ROSTER$ IN COLUMN 25 AT LINE 2 IS $ $

--

? REWIND FILE4 SRTFILE

--

? SORT.FILE4 UPON SRTFILE ON LAST-NAME

--

? PREPARE REPORT4 FROM SRTFILE

--

? END

Upon ending the END statement, Query Update will sort the file and indicate that it has finished. Control is then passed to the NOS operating system which enters BATCH mode. Now the file on which the report resides is named REPORT4. Note that the report was generated by first extracting the last name, first name, and social security number from the data base permanent file TRENDAEES of sub-schema TUESTUD. File TRENDAEES was attached in write mode (M=W). The extracted information was then placed on a temporary file named FILE4, which first had to be wound. The extracted information was placed on the temporary file FILE4 with the EXTRACT
statement. Query Update was then directed to FORMAT REPORT 4 on which the information would ultimately reside. Then the details for the formatting were given using the DETAIL statement. Also the details for generating the title were then given. After this, the temporary file, FILE4, and the sort file, SRTFILE, were both rewound. Then Query Update was directed to sort the information on the temporary file, FILE4, and to place the sorted information on a file named SRTFILE; this was done using the SORT statement. Once the information had been sorted, the report was prepared from the information on the file named SRTFILE and it was placed on a file named REPORT4.

If the user desires to save the report, REPORT4, which has just been generated, he enters the NOS command SAVE, REPORT4. This will place a permanent copy of REPORT4 in the user's library. To have a copy of the report printed at the terminal, the user may enter the following commands:

EDIT, REPORT4

LIST,n (where n is the number of records in the report)

When the LIST,n command was given, the first n records of the report were printed at the terminal with a title. The following is a partial listing of the report:
ENROLLMENT ROSTER

BACHELOR       PAM       SHERRY       782-17-3864
BISHOP         RUFUS     CLYDE       960-39-1842
BLACK          SUSAN     REE         332-85-1842
BLOCKER        GWEN      BERNICE     448-22-4412

(Note: the above data was test data)

Thus it is clear that a user may prepare a file in any manner which he so desires from the data base.

The data base is capable of accomplishing much more than has been pointed out in this thesis; but this will depend largely upon the imagination and ingenuity of its users.
APPENDIX

THE QUERY UPDATE CONTROL CARD

In using Query Update in the interactive mode for the purpose of creating a database or in updating the database, the QU control statement, with or without any of its optional parameters, must be submitted by a user in order to establish communications with the Query Update interpreter. The optional parameters of the Query Update interpreter gives a user tremendous flexibility and control in manipulating the data in the database. The QU control statement, as is indicated in the CDC Query Update Reference Manual, takes the following format:

\[ \text{QU(AF)}(I(=\text{file-name})))(O(=\text{file-name}))(,TL=n)(,RO)(,U) \]

A brief description for interactive mode of each of these parameters is given below pursuant to the data base described in this thesis project:

AF This parameter directs Query Update to accept a file on which a certain type of error occurs and to continue processing. The type of error referred to here indicates that the file which contains the error has not been closed since it was last opened. This type of error may not

54
be of any significant concern to a user in that it may not adversely affect his query of the database. Thus a user may include this parameter for this reason. However, if this parameter is omitted, Query Update will ask the user whether or not he desires to have the file accepted for processing. If the user has specified this parameter, Query Update will not give him this choice but will accept the file for processing and will print a message to identify the file as containing a CRM 052 error.

`I=file-name` This parameter directs Query Update to accept its input directives from a file rather than from the terminal. If a file name is not given, the directives will be accepted from a file named COMPILE which receives its input from the default input file named ZZZZZZIN which automatically is connected to the terminal throughout any Query Update session.
O=file-name

This parameter directs Query Update to put all of its responses on a file (named by file-name). The Query Update responses are displayed at the terminal only if the user connects an output file when the O parameter is indicated. If no file name is given following the O parameter, the Query Update responses are placed on the default output file named ZZZZZOU which is automatically connected to the terminal throughout any Query Update session.

T=file-name

This parameter identifies the transaction file which receives a copy of all directive terminal input and system output other than generated reports. This file may be printed at the end of a terminal session so that a user can have a copy of what he submitted to Query Update and a copy of all system diagnostic messages.

PL=n

This parameter indicates the number of lines per page for output. The
maximum value allowed is 255. But if a value is given for n which is greater than the value set for the page size of the user's interactive system, then that value is overridden by the system. If a user specifies PL=0, then the paging facility is disabled. In this event, all lines of output are transmitted for display in response to the More request by Query Update.

**PW=n**

This parameter specifies the number of characters per line for output. The value for n must include two characters per line to be used for carriage control supplied by Query Update. The maximum value allowed is 255, but a value specified for n that is greater than the value set for the page size of the user's interactive system is overridden by the system. The default page width of the NOS operating system is 72 characters per line.

**TL=n**

This parameter indicates in charac-
ters the maximum length of a transmission by a user. A decimal integer which must not exceed six digits must be specified for n. The minimum transmission length allowed is 20 characters. The maximum transmission length allowed is 1030 characters.

RO

This parameter indicates that directives are not allowed for interactive or batch mode.

U

This parameter indicates that Query Update is not to abort processing upon encountering a fatal diagnostic in batch mode. It does not apply to interactive mode. It is included here only for completeness.

The afore stated parameters of the QU control statement bounds a Query Update session prior to the submission of any Query Update directives. The real power of the data base described herein, in its use of Query Update, lies in the submission of the directives of Query Update. The application of many of these directives to the data base described herein is covered in another
section of this thesis. The great flexibility of these
directives was a major influence on the decision to let
Query Update be the query language of the data base de-
scribed herein.
Bibliography

Data Description Language for the Query Update Sub-

Data Description Language (DDL) Version 1 Reference
Manual. Control Data Corporation. Sunnyvale, 
Calif. 1975.

Data Description Language (DDL) Version 1 User's
Guide. Control Data Corporation. Sunnyvale, 

DMS 170 Cyber Record Manager Version 1 Reference
Manual. Control Data Corporation. Sunnyvale,  
Calif. 1977.

DMS 170 Data Administrator User's Guide. Control Data
Corporation. Sunnyvale, Calif.


Update Sub-Schema Definition. Control Data Cor-

DMS 170 DDL Version 3 Reference Manual Volume 2 Sub-
Schema Definition for Cyber Data Base Control
System Use with COBOL Query Update. Control Data

DMS 170 Query Update Version 3 Reference Manual. Con-

DMS 170 Relational Data Base Facility Programmers System

Martin, James. Computer Data Base Organization. Prentice

NOS 1.0 Text Editor Reference Manual. Control Data Cor-
poration. Arden Hills, Minn. 1975

NOS 1.0 Time-Sharing User's Reference Manual. Control
Data Corporation. Arden Hills, Minn.

Tischritzis, Dionysios C. and Frederich H. Lochousky. Data