

**THIS BOOK
CONTAINS
NUMEROUS
PAGES THAT ARE
CUT OFF**

**THIS IS AS
RECEIVED FROM
THE CUSTOMER**

IMPLEMENTATION OF A PROTOTYPE DATA BASE FOR
ADVISING COMPUTER SCIENCE STUDENTS

by

HARVEY A. LONG

B.S., Sterling College, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

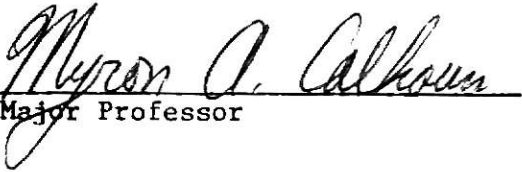
MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

Approved by:


Major Professor

LD
2668
R4
1976
L66
C.2

TABLE OF CONTENTS

List of Figures

Document

I. Introduction	1
1.1 Background	1
1.2 Objective	2
1.3 The Data Base Management System	2
1.4 IDMS Definitions	3
II. The Data Base Design	5
2.1 Schema Description	5
2.2 Storage Requirements	12
2.3 Subschema Description	15
III. The Application Programs	18
3.1 The Initial Load Program	18
3.2 The Add Department, Major-Course, or BS-REQ-AS Records Program	21
3.3 The Add Student Record Program	23
3.4 The Student Record Update Program	23
3.5 The Add Credits Record Program	28
3.6 The Credits Record Update Program	28
3.7 The Delete Record Program	32
3.8 The Advisor's Report Program	32
IV. Conclusion	35
Bibliography	37
Appendix A - Required Job Control Language	38
Appendix B - The Users Guide	41
Appendix C - Program Source Listings	64
Appendix D - Sample Program Reports	65

List of Figures

Figure

2.1	Schema Data Structures	6
2.2	Schema Description Statements	7
3.1	Keypunch Instructions for the Initial Load Program	19
3.2	Keypunch Instructions for the Department Record of Programs Addloads	22
3.3	Keypunch Instructions for the Add Student Record Program	24
3.4	Keypunch Instructions for the Student Record Update Program	26
3.5	Keypunch Instructions for the Add Credits Record Program	29
3.6	Keypunch Instructions for The Credits Record Update Program	30
3.7	Keypunch Instructions for the Delete Record Program	33
3.8	Keypunch Instructions for the Advisor's Report Program	34

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

This report is an outgrowth of a desire of the Computer Science Department to automate a portion of the files it maintains for students majoring Computer Science. Information from the automated files would then be used by faculty members to advise students on their plan of study.

Some previous work has been done at the University in this area. The College of Home Economics currently has a program which does a check of courses a student has taken to determine if graduation requirements are being met. While this is similiar to what is needed in the Computer Science Department it is not as comprehensive as what is planned. Also this program was not implemented using any type of data base management system. Since the Computer Science Department has the Integrated Database Management System (IDMS) available, it is intended that the student advising system be implemented using the facilities IDMS offers. Other work in this area and more relevant to this report was done in the Fall of 1975 by James H. Voelz and Robert L. Garland⁵. Their project consisted of designing a prototype data base management system for the University's academic records. This design was done using IDMS. The final data base design turned out to be very comprehensive and quite large.

These two factors made it uneconomical for the Computer Science Department to implement this deisng. It was determined, though, that by using Voelz and Garland's report as a starting point an economical system could be designed and implemented.

Conversations were held with Dr. Paul Fisher, the Computer Science Depart-

ment head, and Dr. Myron Calhoun, who maintains the course and curriculum information for the department, to determine what information the data base should contain. The requirements are as follows:

1. Some personal information on the student.
2. Requirements for the Bachelor of Science Degree.
3. Major course requirements.
4. Courses completed and/or currently enrolled in.

Given this information a suitable report can be generated which will provide faculty members with adequate information to advise students in academic matters.

A secondary reason for this report is to provide the Computer Science Department with an example of a data base management system. The Department wishes to have available examples of data base management systems using IDMS for future studies in minicomputer data base systems and distributed networks.

1.2 OBJECTIVE

It is the purpose of this report to show the design and implementation of a data base management system using the Integrated Database Management System. The data base will contain all the information indicated in the preceding section as being desirable for a useable system. This information will be used to generate a student advising report for use by faculty members when advising a student on his or her plan of study.

In addition to the program to generate the student advising report, all necessary application programs for creating and maintaining the data base will be provided. Also, information will be included which will allow a user to easily utilize the system.

1.3 THE DATA BASE MANAGEMENT SYSTEM^{1,2}

As indicated in the preceeding sections IDMS was chosen to implement the data base designed in this report. IDMS is a product of the Cullinane Corpora-

tion and is an implementation of a subset of the CODASYL Data Base Task Group Language Specifications. It will provide data base facilities for ANS COBOL programs and any other host language which supports a CALL statement or equivalent, operating in an IBM 360 or 370 OS/DOS/VS or UNIVAC SPECTRA TDOS/VMOS environment.

Separate language facilities are provided by IDMS for description of data and the manipulation of data. This feature separates the application program from the data description function giving data and program independence. This separation allows all data and data relations to be integrated into a data base which is common to all application programs which use it.

The IDMS Data Description Language (DDL) provides a complete description of the data base and is used to describe parts of a data base to various application programs accessing it. There is only one complete description for a data base, but there may be many sub-descriptions, each describing a specific combination of record segments, records, sets and areas which apply to a given application program.

A Data Manipulation Language (DML) is also provided by IDMS. It is used to store, retrieve and manipulate data stored on random access devices. DML statements can be used anywhere within the procedure of an application program.

1.4 IDMS DEFINITIONS^{1,2}

A SCHEMA consists of DDL statements and is a complete description of a data base. It includes descriptions of all the areas, set occurrences, record occurrences and associated data-items, defined below, that exist in the data base.

A SUBSCHEMA consists of separate DDL statements from the schema. It describes those areas, data-items, records and sets which are known to one or more specific application programs. It is not necessary for a subschema to describe the entire data base.

A DATA-ITEM is the smallest unit of named data. An occurrence of a data-item is a representation of a value.

A RECORD is a named collection of zero, one, or more data-items. There is no limit to the number of occurrences in the data base of each record-name specified in the schema for that data base. There is an important distinction between actual occurrences of a record and the type or name of the record. The type of a record is determined by the actual description of the structure. An occurrence is constituted by the contents of a record described by a given structure.

A SET is a named collection of record types. As a result, the characteristics of any number of occurrences of the named set are established. Each set-name specified in the schema must have one record type identified as its OWNER and one or more record types identified as its MEMBER records. The number of occurrences in the data base of each set-name specified is not limited.

An AREA is a named logical sub-division of the storage space in the data base. An area may contain occurrences of records of various types. This concept allows the data base to be sub-divided rather than having to consider the data base as a single unit. The opportunity is also provided to optimize access to the data base since the range of interest has been narrowed to a relatively small number of sub-divisions of the entire data base.

A FILE is an extent of addressable secondary storage known to the operating system. A file can consist of an area, a portion of an area, or be large enough to contain several areas.

The DEVICE-MEDIA CONTROL LANGUAGE (DMCL) consists of separate DDL statements from the schema and subschema DDL. The DMCL determines the areas of the data base and their associated files that are to be available to the data base management routines.

CHAPTER II

THE DATA BASE DESIGN

This chapter details the design chosen to implement a data base to fulfill the requirements stated in Chapter I. In addition to describing the schema, Section 2.2 will consider the storage requirements of the data base and Section 2.3 will describe the subschemas used in this report.

2.1 SCHEMA DESCRIPTION

The schema is a description of all the record types and the relationships between them which make up the data base. Figure 2.1 illustrates the structure of the schema and the schema is shown in Figure 2.2. There are five record types used in this schema: department (DEPARTMENT), courses taken (CREDITS), major course requirements (MAJOR-COURSE), student (STUDENT), and requirements for a BS degree (BS-REQ-AS).

Each record type in the schema is defined by a series of record description statements. A record is identified by an alphanumeric name (i.e. SB-REQ-AS) and by a numeric ID. The ID must be a unique unsigned integer of three or four digits, ranging in value from 100 to 9999. It is used by the data base management routines as code in place of the record name. The storage and retrieval criteria for occurrence of a record type is described in the location mode sentence. There are three location modes: CALC, VIA, and DIRECT. CALC and VIA are used in the schema being defined and will be described here. The CALC option stores the record occurrence based upon the value of one or more of the data items within the record. The transformation routine used in this procedure is furnished by the system. The VIA option specifies that the member record occurrence of a set will be placed physically as close as possible to the owner of the set. The final record description statement is the data-item sentence. It defines the group or elementary items of a sentence.

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

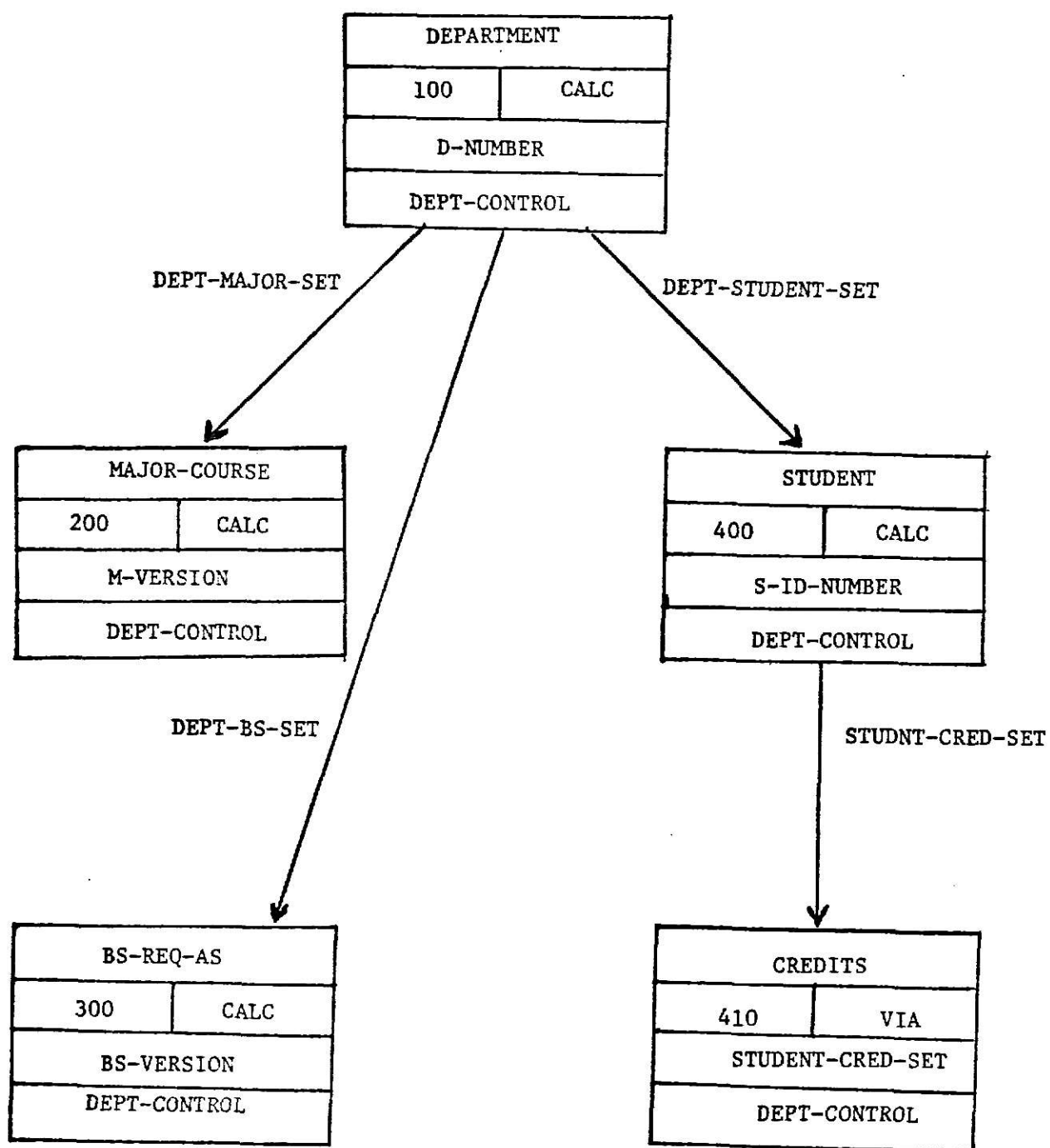


Figure 2.1 Schema Data Structures

SCHEMA DESCRIPTION.

SCHEMA NAME IS ADVISOR1.

AUTHOR. HARVEY A LONG.

DATE. 01/29/76.

INSTALLATION. KANSAS STATE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE.

REMARKS. THIS DATA BASE WILL CONTAIN INFORMATION TO
BE USED TO GENERATE REPORTS FOR ADVISING
COMPUTER SCIENCE STUDENTS.

FILE DESCRIPTION.

FILE NAME IS DEPT-FILE

ASSIGN TO ADVDEPT.

FILE NAME IS JOURNAL

ASSIGN TO ADVJRNL.

AREA DESCRIPTION.

AREA NAME IS DEPT-CONTROL

RANGE IS 1001 THRU 1050

WITHIN FILE DEPT-file

FROM 1 THRU 50.

RECORD DESCRIPTION.

RECORD NAME IS DEPARTMENT.

RECORD ID IS 100.

LOCATION MODE IS CALC

USING D-NUMBER

DUPLICATES ARE NOT ALLOWED.

WITHIN DEPT-CONTROL AREA.

02 D-Number PIC X(3).

COMMENT 'DEPARTMENT CATALOG NUMBER'.

02 D-NAME PIC X(25)

RECORD NAME IS STUDENT

RECORD ID IS 400.

LOCATION MODE IS CALC

USING SID-NUMBER

DUPLICATES ARE NOT ALLOWED.

WITHIN DEPT-CONTROL AREA.

02 S-ID-NUMBER PIC X(11).

02 S-NAME PIC X(25).

02 S-HOME-ADDR.

03 S-HA-STREET PIC X(20).

03 S-HA-CITY PIC X(20).

03 S-HA-STATE PIC XX.

02 S-PREV-SCHOOL.

03 S-PS-TYPE PIC XX.

COMMENT 'HS - HIGH SCHOOL

'JC - JUNIOR COLLEGE

'CU - COLLEGE OR UNIVERSITY'.

03 S-PS-NAME PIC X(30).

Figure 2.2 Schema Description Statements

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH THE ORIGINAL
PRINTING BEING
SKEWED
DIFFERANTLY FROM
THE TOP OF THE
PAGE TO THE
BOTTOM.**

**THIS IS AS RECEIVED
FROM THE
CUSTOMER.**

```

03 S-PS-CITY          PIC X(20).
03 S-PS-STATE         PIC XX.
03 S-DATE-GRAD        PIC X(4).
COMMENT 'MONTH AND YEAR
        'GRADUATED OR TRANSFERRED'.
02 S-DATE-ENROLL      PIC X(4).
COMMENT 'MONTH AND YEAR STARTED
        'ATTENDANCE AT KSU'.
02 S-ADVISOR          PIC X(20).
02 S-DEGREE-VER        PIC XX.
COMMENT 'TWO DIGIT DEPARTMENT CODE
        'AND LAST TWO DIGITS OF YEAR
        'REQUIREMENTS TAKE EFFECT'.
02 S-ACCESS           PIC 999.
02 FILLER             PIC XXX.

```

RECORD NAME IS CREDITS.

RECORD ID IS 410.

LOCATION MODE IS VIA STUDNT-CRED-SET SET.

WITHIN DEPT-CONTROL AREA.

```

02 C-Number
03 C-DEPT-NO          PIC XXX.
03 C-TITLE-NO         PIC XXX.
02 C-TITLE            PIC X(12).
02 C-CREDIT-HRS       PIC XX.
02 C-GRADE            PIC XXX.
COMMENT 'POSSIBLE GRADES: A, B,
        C, D, E, INC, WP, WF, CR, NCR'.
02 C-LAB              PIC X.
COMMENT 'FULFULL NATURAL SCIENCE
        'LAB REQUIREMENT, Y-YES, N-NO'.
02 C-REQ-FILLED       PIC X(4).
COMMENT 'TECH - TECHNICAL REQ
        'FREE - FREE ELECTIVE'.

```

RECORD NAME IS MAJOR-COURSE.

RECORD ID IS 200.

LOCATION MODE IS CALC

USING M-VERSION

DUPLICATES ARE NOT ALLOWED.

WITHIN DEPT-CONTROL AREA.

```

02 M-VERSION          PIC X(4).
COMMENT 'TWO DIGIT DEPARTMENT CODE
        'AND LAST TWO DIGITS OF YEAR
        'REQUIREMENTS TAKE EFFECT'.
02 M-COURSE           OCCURS 20 TIMES.
03 M-DEPT-NO          PIC XXX.
03 M-TITLE-NO         PIC XXX.
03 M-TITLE            PIC X(12).
03 M-CREDIT-HRS       PIC XX.
03 M-PREQ             PIC X(20).
03 M-SEM-OFFERED      PIC X(6).
COMMENT 'I - FALL SEMESTER
        'II - SPRING SEMESTER
        'S - SUMMER SCHOOL'.

```

Figure 2.2 Schema Description Statements (continued)

```

02 NO-MAJ-COURSES      PIC 99.
02 FILLER              PIC XX.

RECORD NAME IS BS-REQ-AS.
RECORD ID IS 300.
LOCATION MODE IS CALC
    USING BS-VERSION
    DUPLICATES ARE NOT ALLOWED.
WITHIN DEPT-CONTROL AREA.
    02 BS-VERSION      PIC XX
      COMMENT 'LAST TWO DIGIST OF YEAR
        'REQUIREMENTS TAKE EFFECT'.
    02 BS-GENERAL-REQ.
      03 ORAL-COM-1-2.
        04 OC-DEPT      PIC XXX.
        04 OC-COURSE-1  PIC XXX.
        04 OC-COURSE-2  PIC XXX.
      03 SPEECH
      03 HUMAN-SOC-SCI.
        04 HS-DEPT-NO   PIC XXX      OCCURS 18 TIMES.
      03 NATURAL-SCIENCE.
        04 NS-DEPT-NO   PIC XXX      OCCURS 9 TIMES.
      03 PHYSICAL-ED.
        04 PE-DEPT-NO   PIC XXX
        04 PE-CRSE-NO   PIC XXX
        COMMENT 'CHECKED FOR BY COMPARING
          'CREDITS TO MAJOR-COURSE
          'RECORD'.
      03 FILLER          PIC XXX.

```

Figure 2.2 Schema Description Statements (continued)

SET DESCRIPTION.

SET NAME IS DEPT-MAJOR-SET.
 ORDER IS SORTED.
 MODE IS CHAIN LINKED PRIOR.
 OWNER IS DEPARTMENT
 NEXT POSITION IS 1
 PRIOR POSITION IS 2.
 MEMBER IS MAJOR-COURSE
 MANDATORY AUTOMATIC
 NEXT POSITION IS 1
 PRIOR POSITION IS 2
 ASCENDING KEY IS M-VERSION
 DUPLICATES ARE NOT ALLOWED.

SET NAME IS DEPT-BS-SET.
 ORDER IS SORTED.
 MODE IS CHAIN LINKED PRIOR.
 OWNER IS DEPARTMENT
 NEXT POSITION IS 3
 PRIOR POSITION IS 4.
 MEMBER IS BS-REQ-AS
 MANDATORY AUTOMATIC
 NEXT POSITION IS 1
 PRIOR POSITION IS 2
 ASCENDING KEY IS BS-VERSION
 DUPLICATES ARE NOT ALLOWED.

SET NAME IS DEPT-STUDENT-SET.
 ORDER IS SORTED.
 MODE IS CHAIN LINKED PRIOR.
 OWNER IS DEPARTMENT
 NEXT POSITION IS 5
 PRIOR POSITION IS 6.
 MEMBER IS STUDENT
 MANDATORY AUTOMATIC
 NEXT POSITION IS 1
 PRIOR POSITION IS 2
 ASCENDING KEY IS S-NAME
 DUPLICATES ARE NOT ALLOWED.

SET NAME IS STUDNT-CRED-SET.
 ORDER IS SORTED.
 MODE IS CHAIN LINKED PRIOR.
 OWNER IS STUDENT
 NEXT POSITION IS 3
 PRIOR POSITION IS 4.
 MEMBER IS CREDITS
 MANDATORY AUTOMATIC
 NEXT POSITION IS 1
 PRIOR POSITION IS 2
 ASCENDING KEY IS C-NUMBER
 DUPLICATES ARE NOT ALLOWED.

Figure 2.2 Schema Description Statements

The DEPARTMENT record is used to provide a description of department for report headings. Also, because of the set relationships it maintains, it ties together all occurrences of the five record types for a department. In addition to the department name the record contains a three character identification code.

The STUDENT record contains basic personal information for a student plus an eleven digit identification code. In addition information is maintained indicating the version of major course and degree requirements under which a student enrolled and the student's advisor. The STUDENT record is stored using the CALC mode with the identification code as the identifier.

The CREDITS record is set up to contain information about a course that has been completed or in which the student is currently enrolled. There is an occurrence of this record for each course taken. The record contains basic information about a course; course number, name, credit hours and the grade received. In addition to these items there is also provision to indicate whether or not the course meets any special requirements, such as the laboratory requirement for Natural Science. Since this record is closely associated with the STUDENT record the VIA location mode is used so that it will be stored near the STUDENT record.

The MAJOR-COURSE record provides information on courses required by a department for a major in that area. This information includes, course number, title, credit hours, prerequisites, and semester offered. There is also provision to indicate what version of the major course requirements a particular occurrence of the record represents. The record is stored using the CALC mode with the version used for identifier.

The BS-REQ-AS record contains information concerning the courses required by the University to obtain a Bachelor of Science degree from the College of Arts and Sciences. When specific courses are needed to meet requirements, appropriate identifiers are maintained. In those areas where requirements

are more general, such as Humanities and Social Science (HUMAN-SOC-SCI), only the department numbers are used for comparison with a CREDITS record. Since, as the the MAJOR-COURSE record, the requirements sometimes change, there is a provision to indicate what version each occurrence of the record indicates. This record is also stored using the CALC mode to allow easier access. The version is used as the identifier for the CALC transformations.

There are four set relationships described in the schema. Of the four, the STUDENT-CRED-SET which establishes a relationship between the STUDENT and CREDITS records is the most important. It provides the only access to occurrences of the CREDITS record. The other set relationships are not necessary for access purposes but are used to tie together all occurrences of the different record types within a department. Although not necessary for access, these relationships do allow access to be made through the DEPARTMENT record to all occurrences of the other record types within a department.

2.2 STORAGE REQUIREMENTS

Total space requirements for the data base are found by calculating the space requirements for each area and then adding them together. The following formulas are used to determine an area's storage requirements³.

1. $N = (L1*O1 + L2*O2 + \dots + Ln*On)/P$ rounded up
2. $I = (8(O1 + O2 + \dots + On) + 32*N)/P$ rounded up
3. $S = 2(N + I)/(P - 32)$ rounded up
4. $A = N + I + S$

where:

- L = length of records including pointers.
- O = number of occurrences of the record.
- P = page size.
- N = number of pages necessary to hold the anticipated number of record occurrences of each record type assigned to the area.
- I = the number of pages necessary to hold the required PAGE indexing
- S = the number of space management PAGES required.
- A = the total area page requirement.
- n = number of records.

Before any calculations can be made there are three things which must be determined. They are the length of each record, the number of occurrences of each record and the page size.

The length of each record is determined by totaling the byte requirements indicated by each PICTURE clause. The number of occurrences of each record type must be estimated. Since the data base is being implemented only for the Computer Science Department, only one occurrence of the DEPARTMENT record is being considered. The MAJOR-COURSE and BS-REQ-AS records are computed using on two occurrences each; one for current requirements and one for future changes. There are at present approximately 100 students majoring in Computer Science, but to allow for expansion calculations are being made on the basis of 150 STUDENT records. The number of occurrences of the CREDITS record is based on the assumption that each freshman will use ten records, each sophomore twenty records, each junior thirty records and each senior forty records. Statistics maintained by the department indicate that 33% of the majors are freshmen, 17% sophomore, 21% juniors, and 29% seniors. Figuring the combination of number of records per class and percent for each class on the basis of 150 students, the total number of occurrences for the CREDITS record would be 3770. Page size is the third item to be determined. Although it is an arbitrary number, it must be divisible by four, should be large enough to contain a significant number of records, and should fit on a IBM 3330 disk track with a minimum of wasted storage. A page length of 3156 bytes, as recommended by the IDMS user's manual, is being used since it meets the above requirements.

Since there is only one area in the data base a single application of the formulas is all that is required. The determinations for the DEPT-CONTROL area are as follows.

A = 53 + 11 + 1

A = 65 pages

Total Bytes = 205140

The 205140 bytes translates into about sixteen tracks of 3330 disk space. There are nineteen tracks per cylinder, so by allocating an entire cylinder for the data base it will give additional room for expansion.

2.3 SUBSCHEMA DESCRIPTION

The subschema is the mechanism through which an application program gains access to a data base. It also determines what information a program can add to or extract from the data base.

IDMS sets up its protection mechanism in the subschema. Various types of protection can be set up at the area, record, and set levels. At the area level there are two modes, update and retrieval. There are then two variations to the basic mode possible, protected and exclusive. Protection keys are available to allow or prohibit any or all of the access modes. IDMS provides for seven different actions that can be performed on a record. They are insert, remove, store, delete, modify, find, and get. Each record then can be set up to allow or prohibit all of any of the actions and different records can restrict different actions. Sets may have three functions performed on them: insert, remove and find. As in the previous cases, privacy locks are provided to allow or prohibit these functions.

There were seven subschemas used in the implementation of this project. They will be discussed in terms of what record types are accessed and what restrictions, if any, are placed on that access.

Subschema LOADER may be invoked by one program which will then have exclusive control of the DEPT-CONTROL area for update purposes. Access is

allowed to the DEPARTMENT, MAJOR-COURSE, and BS-REQ-AS records for execution of the store function only. This subschema is the only one which places any restrictions on the usage mode for the DEPT-CONTROL area.

Subschema SUBSTDNT allows a program to access the STUDENT record only for the purpose of storing new records. The DEPARTMENT record and the DEPT-STUDENT-SET set are included in the subschema so that the proper relationships can be established when a STUDENT record is stored.

Subschema SUBSTUPD is used by a program to make updates to a STUDENT record. Access is allowed to the STUDENT record for modification only. The DEPARTMENT record and DEPT-STUDENT-SET set are included to allow the maintenance of proper relationships when the modify function is used.

Subschema LDCRDIT is used to enter information on courses taken into the data base. The STUDENT and CREDITS records and the STUDNT-CRED-SET are included to allow the proper relationships to be established. Only "find" function can be used on the STUDENT record when this subschema is invoked.

Subschema SUBCDUPD is used to update occurrences of the CREDITS record. Only the find, get, and modify functions are permitted. The find and get functions allow a record to be moved into an applications program's work space where changes can be made. The modify function then insures that these changes are reflected in the data base. The STUDENT record and STUDNT-CRED-SET set are also included in the subschema. In addition to maintaining relationships, the STUDENT record is needed to enable the CREDITS record to be retrieved since it is stored by the VIA location mode.

Subschema SUBDELET is available to allow deletion of any or all occurrences of any of the record types in the DEPT-CONTROL area. This, then, means that great care must be taken in the use of this subschema. The potential for disaster to the data base is very great but the ability to delete occurrences of the

record types is a necessary part of the system. To afford the system maximum protection it is recommended that this subschema be kept in the possession of the Department and used by authorized personnel only.

Subschema SUBREPRT gives access to all record types in the data base. Information can be obtained from the records but no changes of any kind are permitted.

CHAPTER III

THE APPLICATION PROGRAMS

Chapter three describes the system of user programs used in the implementation of this project. There are eight programs used to fulfill the project requirements. Six of the programs are used either to store information in the data base or to update information previously stored. One program is used to delete occurrences of record types when they are no longer required and one program generates the advisor's report for a student.

The following sections will describe each program of the system in terms of its function, inputs required, and outputs produced.

3.1 THE INITIAL LOAD PROGRAM

Program LOAD1 is used to create occurrences of the DEPARTMENT, MAJOR-COURSE, AND BS-REQ-AS records. The subschema LOADER is invoked and allows the program access to the data base in the exclusive update mode. The subschema only allows the program to store information in the data base.

This program is intended to be used only once, at the time the data base is set up. Occurrences of these record types, once correctly loaded, are not subject to update. If any changed in major course requirements or BS requirements take place a new occurrence of the record type is created using a different version number.

Input to the program is the information required for the DEPARTMENT, MAJOR-COURSE, and BS-REQ-AS records. Key punch instructions are shown in figure 3.1. Each input card must contain a three character transaction code. This indicates to the program what record type is being entered so that it can be stored properly.

Figure 3.1 shows two key punch formats for the MAJOR-COURSE and BS-REQ-AS

KEYPUNCH INSTRUCTIONSFORTHE INITIAL LOAD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
DEPARTMENT RECORD INFORMATION			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-7	3	
Filler	8	1	
Department Name	9-3	25	
MAJOR-COURSE INFORMATION			
FORMAT 1			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Version	5-8	4	Department ID and last two digits of year effective
Filler	9	1	
Number of Courses	10-11	2	
FORMAT 2			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Department Number	5-7	3	
Course Number	8-10	3	
Title	11-12	12	
Credit Hours	23-24	2	right justified
Prerequisites	25-44	20	
Semester Offered	45-50	6	I, II, S

Figure 3.1

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
BS REQUIREMENTS INFORMATION			
FORMAT 1			
Transaction Code	1-3	3	'REQ'
Filler	4	1	
Version	5-6	2	last two digits of year effective
Filler	7	1	
English Dept. No.	8-10	3	
Filler	11	1	
Course 1	12-14	3	
Course 2	16-18	3	
Filler	19	1	
Speech	20-22	3	
Filler	23	1	
PE Dept Number	24-26	3	
Filler	27	1	
PE Course Number	28-30	3	
FORMAT 2			
Transaction Code	1-3	3	'REQ'
Filler	4	1	
Department Number	5-7	3	

Figure 3.1 (continued)

records. Format 1 for the MAJOR-COURSE contains the version and how many courses are being entered. Format 2 is to enter information for each of the courses. There will be one card of format 2 for each major course. For the BS-REQ-AS record format 1 indicates the version and all specific requirements. Format 2 is used to enter department numbers for those departments in the Natural Science, Humanities and Social Science area. In each case when entering information format 1 cards must precede format 2 entries.

Output from the program is stored in the data base and a printed report of what has been stored is produced. The printed output then should be checked for errors. Any corrections necessary are accomplished by reloading the data base.

3.2 THE ADD DEPARTMENT, MAJOR-COURSE, OR BS-REQ-AS RECORDS PROGRAM

ADDLOAD1 is used to add new occurrences of the DEPARTMENT, MAJOR-COURSE, or BS-REQ-AS record types to the data base once it has been established. It used the same subschema (LOADER) as the initial load program (LOAD1).

Inputs to the program are also the same as for LOAD1. The keypunch instructions in figure 3.1 are to be used for the MAJOR-COURSE and the BS-REQ-AS records. Figure 3.2 shows the keypunch instructions for the DEPARTMENT record. A new field has been added to indicate if a new occurrence of the record type is being added. This is necessary since department information must be entered, even if the proper occurrence of the record type exists in the data base, so that proper relationships can be maintained.

Information input into the program is stored in the data base and a printed report showing each record added. To any errors detected corrections are done by deleting the incorrect occurrence and rerunning the program with the correct information.

KEYPUNCH INSTRUCTIONSFORTHE DEPARTMENT RECORD OF PROGRAM ADDLOAD1

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-9	5	
Filler	10	1	
Department Name	11-37	27	
Filler	38	1	
Type Code	39-41	3	'NEW' if new occurrence, blank otherwise

Figure 3.2

3.3 THE ADD STUDENT RECORD PROGRAM

This program (STDNTLD) stores new occurrences of the STUDENT record type in the data base. Subschema SUBSTDNT is invoked to allow the program proper access to the data base.

The program accepts information in the formats shown in figure 3.3. If any of the information indicated is not applicable to the student the space is left blank.

Output is stored in the data base and a printed report is produced. The printed report is then used for error checking. Any corrections necessary are made using the student record update program (STDUPTE).

3.4 THE STUDENT RECORD UPDATE PROGRAM

Program STDUPTE is used to correct errors detected in the program STDNTLD and to make modifications of occurrences of the STUDENT record when any information contained therein becomes obsolete. Invoking subschema SUBSTUPD allows the program to accomplish these tasks.

Inputs for this program are the same as the the STDNTLD program. Formats 2 thru 7 of figure 3.3 are used for keypunching. In addition to these, two different formats are shown in figure 3.4. These are necessary to allow the program to successfully complete its function. To allow for all possible changes provision is made to enter the student ID twice. The first time with the FID transaction code, shown in figure 3.4, is to use it to access the proper occurrence of the STUDENT record. The second time is to permit it to be changed if necessary. The transaction code (MDY) of the second format in figure 3.4 indicates that all changes have been made to this occurrence of the STUDENT record and the modification function can be executed.

KEYPUNCH INSTRUCTIONS
FOR
THE ADD STUDENT RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department ID	5-7	3	
FORMAT 2			
Transaction Code	1-3	3	'SID'
Filler	4	1	
Student ID	5-15	11	
Filler	16	1	
Name	17-41	25	
FORMAT 3			
Transaction Code	1-3	3	'HAD'
Filler	4	1	
Home Street Addr	5-24	20	
Filler	25	1	
City	26-45	20	
Filler	46	1	
State	47-48	2	
FORMAT 4			
Transaction Code	1-3	3	'PSC'
Filler	4	1	
Type of School	5-6	2	'HS', 'JC', or 'CU'
Filler	7	1	
Name	8-37	30	

Figure 3.3

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 5			
Transaction	1-3	3	'PLC'
Filler	4	1	
City School Loacted	5-24	20	
Filler	25	1	
State	26-27	2	
Filler	28	1	
Date Grad or Transfer	29-32	4	month and year
FORMAT 6			
Transaction Code	1-3	3	'DAD'
Filler	4	1	
Date Enrolled	5-8	4	month and year
Filler	9	1	
Advisor	10-29	20	
FORMAT 7			
Transaction Code	1-3	3	'CDV'
Filler	4	1	
Degree Version	5-6	2	last two digits of year effective
Filler	7	1	
Course Version	8-11	4	Department ID and last two digits of year effective

Figure 3.3 (continued)

KEYPUNCH INSTRUCTIONS
FOR
THE STUDENT RECORD UPDATE PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'FID'
Filler	4	1	
Student ID	5-15	11	
FORMAT 2			
Transaction Code	1-3	3	'MDY'
Filler	4-80	77	

Figure 3.4

Outputs are the updated information in the data base and a printed report showing the changes made. Error corrections are made by rerunning the program.

3.5 THE ADD CREDITS RECORD PROGRAM

Program CREDITLD stores new occurrences of the CREDITS record type in the data base. Access is gained to the data base by invoking subschema LDCRDIT.

Information is input in the formats shown in Figure 3.5. The program must receive input in the order established in Figure 3.5.

The program stores output in the data base and produces a printed report of the information stored. Errors discovered in the printed report can be corrected using the update program for the CREDITS record (CRDUPDTE).

3.6 THE CREDITS RECORD UPDATE PROGRAM

Program CRDUPDTE is used to correct errors detected in the add CREDITS record program (CREDITLD) and to make modifications of occurrences of the CREDITS record when any information in an occurrence becomes outdated. Subschema SUBCDUPD is invoked by this program to gain access to the data base.

The information input to this program is the same as that for the add CREDITS records program. Transaction codes are added so that a user can input only that information which is to be changed. Key punch instructions are shown in Figure 3.6. The student ID is used to access the STUDENT record to establish the proper relationships so that the CREDITS record can be retrieved since it is stored by the VIA location mode. The department number and course number make up the record identifier. They must be input once to locate the record occurrence and can be input a second time if they need to be changed. The last card input for a record occurrence must have the transaction code, MODY, to indicate that changes have been completed and the modification function can be executed.

At the completion of the program the changes desired are recorded in the data base and a report printed showing the record with changes.

KEYPUNCH INSTRUCTIONSFORTHE ADD CREDITS RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Filler	1-2	2	
Student ID	3-13	11	
FORMAT 2			
Filler	1-2	2	
Department Number	3-5	3	
Filler	6	1	
Course Number	7-9	3	
FORMAT 3			
Filler	1-2	2	
Course Name	3-14	12	
FORMAT 4			
Filler	1-2	2	
Credit Hours	3-4	2	right justified
Filler	5	1	
Grade	6-8	3	right justified, possible grades: A,B, C,D,F,WP,WF,INC,CR,NC
FORMAT 5			
Filler	1-2	2	
Lab Requirement	3	1	'Y' or 'N'
Filler	4	1	
Special Requirement	5-8	4	'TECH', 'FREE' or Blank

Figure 3.5

KEYPUNCH INSTRUCTION
FOR
THE CREDITS RECORD UPDATE PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-4	4	'STID'
Filler	5	1	
Student ID	6-16	11	
FORMAT 2			
Transaction Code	1-4	4	'FREC' for locating proper occurrence, 'CNUM' if department number or course number is to be changed.
Filler	5	1	
Department Number	6-8	3	
Filler	9	1	
Course Number	10-12	3	
FORMAT 3			
Transaction Code	1-4	4	'TITL'
Filler	5	1	
Course Name	6-17	12	
FORMAT 4			
Transaction Code	1-4	4	'CRGR'
Filler	5	1	
Credit Hours	6-7	2	right justified
Filler	8	1	
Grade	9-11	3	right justified possible grades A,B,C,D,F,WP,WF,INC CR, NCR

Figure 3.6

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 5			
Transaction Code	1-4	4	'LBRQ'
Filler	5	1	
Lab Requirement	6	1	'Y' or 'N'
Filler	7	1	
Special Requirement	8-11	4	'TECH', 'FREE' or Blank
FORMAT 6			
Transaction Code	1-4	4	'MODY'
Filler	5-80	76	

Figure 3.6 (continued)

3.7 THE DELETE RECORD PROGRAM

Program DELETREC gives the user the ability to delete any or all occurrences of the record types comprising the data base. Subschema SUBDELET is invoked to allow access to the data base.

Inputs to the program are a transaction code indicating which record type is to be deleted and an identifier so that the proper occurrence of the record type can be accessed. Key punch instructions for this program are shown in Figure 3.7.

Output from the program is a message indicating whether or not the deletion was successful.

Because of the nature of this program there is a high potential for serious damage to the data base if it is not used with great care. Therefore, it is recommended that it and the subschema SUBDELET be kept in the possession of the Department for use by authorized personnel only.

3.8 THE ADVISOR'S REPORT PROGRAM

Program ADVISRPT produces a report that gives a student's current status in regard to completing the requirements for a BS Degree in Computer Science. This report is used by faculty members for aid in advising a student on his or her plan of study. The report contains some personal information, a list of courses taken, the requirements for graduation the courses fulfill, and major courses remaining to be taken.

Input to the program is the department number and the student ID of the student requiring the report. Key punch instructions are shown in Figure 3.8.

The output from the program is the advisor's report. Examples of the advisor's report are included in Appendix D.

KEYPUNCH INSTRUCTIONSFORTHE DELETE RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-7	3	
FORMAT 2			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Version	5-8	4	
FORMAT 3			
Transaction Code	1-3	3	'BSR'
Filler	4	1	
BS Version	5-6	2	
FORMAT 4			
Transaction Code	1-3	3	'STD'
Filler	4	1	
Student ID	5-15	11	
FORMAT 5			
Transaction Code	1-3	3	'CRD'
Filler	4	1	
Student ID	5-12	11	
Filler	16	1	
Course number	17-22	6	Department & Course Number

Figure 3.7

KEYPUNCH INSTRUCTIONSFORTHE ADVISOR'S REPORT PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
Department Number	1-3	3	
Filler	4	1	
Studnet ID	5-15	11	

Figure 3.8

CHAPTER IV

CONCLUSION

This report has covered the basic aspects of the design and implementation of a data base system which provides information used in advising students majoring in Computer Science. A complete set of programs is provided to set up, maintain, and update the data base in addition to generating the required report. This, then, fulfills the requirements for the system set forth in Chapter I.

A review of the system indicates that the most volatile record type in terms of additions and deletions will be the CREDITS record. This is one of the records intended to be maintained by the student. At present the only means of deleting an occurrence of a record is the program DELETREC. It is being recommended, though, that students not be allowed access to this program, for reasons of data integrity and security. Therefore, as an enhancement to the system it seems desirable to have an application program available which would allow an occurrence of the CREDITS record to be deleted.

Another area that needs to be considered is protection of the data base from system failure and from the malicious or inadvertent actions of users. IDMS provides utilities for dumping and restoring copies of the data base to and from tape. Protection from the user is much more difficult. The subschemas in use limit the functions performed by the application programs but there is no way to completely restrict the other avenues of approach to the data base. The main protection this system seems to have is that it is completely unofficial, so there is no information of value to any intruder.

Although, the system designed is intended for use by the Computer Science Department, consideration was given to the possibility of extending the system to serve all departments in the College of Arts and Sciences. In surveying the system from the College point of view, two problems were observed. The first

involves the BS-REQ-AS record. In an extended system one occurrence of each version of the requirements for a BS degree could be used by all department. At present the mechanism for establishing the set relationship between the BS-REQ-AS record and the DEPARTMENT record does not allow the set relationship to be established with new occurrences of the DEPARTMENT record. A careful study of the set relationship shows that it is not necessary for this implementation nor would it be needed in an extended data base since access to the BS-REQ-AS record is gained thru information contained in the STUDENT record. This mechanism could easily be deleted from the schema without affecting any application programs. The BS-REQ-AS record would then function as an independent record.

The second problem concerns the major course requirements. The present system is set up to handle requirements where the courses to be taken are fixed by the department and are the same for all students. There are some departments within Arts and Sciences whose requirements are more flexible. For example, the History Department has only a basic course and seminar as fixed requirements. The course work is then divided into five fields of study and the student must design his or her plan of study to include at least three of them. A situation like this can not be handled by the present system. A redesign of the schema and new application programs would be necessary to handle this problem.

The development of this system consisted of three distinct phases: the design of the schema, development of the subschemas and application programs, and the testing of the system. Approximately thirty hours were spent in designing the schema. This includes the initial discussion period thru the successful compilation of the completed design. Developing the software programs to use the data base required 120 hours. This includes the subschema associated with each program. Once a permanent data base was established, forty hours were spent testing and debugging the system.

BIBLIOGRAPHY

1. Data Definitions Languages, Utilities and GEI Reference Guide;
Release 3.1, Cullinane Corporation, Boston, Mass., April 1975.
2. Data Manipulation Language Programmer's Reference Guide;
Release 3.1, Cullinane Corporation, Boston, Mass., April 1975.
3. Date, C.J.; An Introduction To Database Systems; Addison Wesley
Publishing Company, Reading, Mass., 1975.
4. Martin, James; Computer Data-Bases Organization; Prentice Hall
Publishing Company, Reading, Mass., 1975.
5. Voelz, James H. and Garland, Robert L.; Prototype of a Database Manage-
ment System for Academic Records Utilizing the Integrated Database
Management System (IDMS), A Masters Report, Department of Computer
Science, Kansas State University, Manhattan, Kansas.
6. Weber, William P.; An Implementation of a Budgetary System for the
College of Arts and Sciences, A Masters Report, Department of
Computer Science, Kansas State University, Manhattan, Kansas.

APPENDIX A

REQUIRED JOB CONTROL LANGUAGE

ALLOCATE STORAGE FOR THE
DATA DICTIONARY AND THE PROGRAM LIBRARY

```
//ALLOC EXEC PGM=IEFBR14
//LIB DD DSN=DS8H8.ADVD2,DISP=(,CATLG) , UNIT=3330,VOL=SER=KSCC04,
// SPACE=(TRK(10,5,5)),DCB=(BLKSIZE=13030,RECFM=U)
//DDICT DD DSN=DS8H8.DIC2,DISP=(,CATLG),UNIT=3330,
// VOL=SER=KSCC04,SPACE=(1512,1000)
```

SCHEMA PROCESSOR RUN

```
//SCHEMA EXEC IDMSCHEM,DDICT='DS8H8.DIC2'
//IDMSCHEM.SYSIN DD *
```

schema DDL cards

DMCL PROCESSOR RUN

```
//DMCL EXEC IDMSDMCL,DMCL=ADVDMCL,DDICT='DS8H8.DIC2'
//IDMSDMCL.SYSIN DD *
```

DMCL DDL cards

```
//LKED.SYSLMOD DD DSN=DS8H8.ADVD2(&DMCL), DISP=OLD
```

SUBSCHEMA PROCESSOR RUN

```
//SUBSC EXEC IDMSURSC,SUBSC=subsname, DDICT='DS8H8.DIC2'
//IDMSUBSC.SYSIN DD *
    subschema DDL cards
//LKED.SYSLMOD DD DEN=DE8H8.ADVA2(&SUSC),DISP=OLD
```

DATA BASE INITIALIZATION

```
//DBINIT EXEC IDMSINIT,LIB='DS8H8=DS8H8.ADVD2'
//IDMSINIT.ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=(,CATLG),
//          UNIT=3330,VOL=SER=KSCC04, SPACE=(3156,65)
//IDMSINIT,SYSIN DD *
PROCESS=TOTAL,DMCL=ADVDMCL
```

APPLICATION PROGRAM PROCESSOR RUN

```
//APPL EXEC IDMSDML5,PROG=programe,LIB='DS8H8.ADVD2',
//          DDICT='DS8H8.DIC2'
//IDMSDML5.SYSIN DD *
```

COBOL source program

```
//LKED,SYSMOD DD DSN=DS8H8.ADVD2(&PROG), DISP=OLD
//LKED,SYSIN DD *
INCLUDE SYSLIM (subscname)
```

PROGRAM EXECUTION RUN

```
//GO EXEC IDMSUTIL,PROG=programe, LIB='DS8H8.ADVD2'
//CDREADER DD *
```

data cards

```
/*
//PRTFILE DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
//ADVJRNL DD DUMMY
/*
```

APPENDIX B

THE USER'S GUIDE

APPENDIX B

THE USER'S GUIDE

This supplement to the report, "Implementation of a Prototype Data Base for Advising Computer Science Students", is intended to provide a user with sufficient information to utilize the system implemented for the project. It is included as an appendix to the report but is written so as to be useable by itself.

There are two classes of programs in the system. One is for use by the Department only and the other is intended to be utilized by students majoring in Computer Science. A section of this supplement will be devoted to each.

The description and use of the programs are based on the following assumptions.

1. All programs are assumed to be load modules as in the test system. JCL given is for this arrangement.
2. All error checking is done by the user from the printed output.
3. Data set names used to illustrate the JCL may be changed. A check for this should be made before running a program.

DEPARTMENT

LOAD1

LOAD1 is used to store initial occurrences of the DEPARTMENT, BS-REQ-AS, and MAJOR-COURSE record types in the data base. Key punch instructions are shown in Figure B-1. To correct errors that may occur it is necessary to reload the data base by rerunning the program. Information to input is as follows:

DEPARTMENT RECORD

Department number
Department name

BS-REQ-AS RECORD

Version
English Department Number
 Course Numbers of 2 required courses
Speech Department Number
PE Department Number
 Course Number of required course
Department Number of courses
 in Humanities and Social Science area
Department Numbers of courses
 in Natural Science area

MAJOR-COURSE RECORD

Version
Number of Courses
For each course:
 Department Number
 Course Number
 Title
 Credit Hours
 Prerequisites
 Semester Offered

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIO,PROG=LOAD1,LIB='DS8H8.ADVD2'
```

```
//CHREADER DD *
```

data cards

```
/*
```

```
//PRTFILE DD SYSOUT=A
```

```
//SYSOUT DD SYSOUT=A
```

```
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
```

```
//ADVJRNL DD DUMMY
```

```
/*
```

KEYPUNCH INSTRUCTIONS
FOR
THE INITIAL LOAD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
DEPARTMENT RECORD INFORMATION			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-7	3	
Filler	8	1	
Department Name	9-3	25	
MAJOR-COURSE INFORMATION			
FORMAT 1			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Version	5-8	4	Department ID and last two digits of year effective
Filler	9	1	
Number of Courses	10-11	2	
FORMAT 2			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Department Number	5-7	3	
Course Number	8-10	3	
Title	11-12	12	
Credit Hours	23-24	2	right justified
Prerequisites	25-44	20	
Semester Offered	45-50	6	I, II, S

Figure B-1

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
BS REQUIREMENTS INFORMATION			
FORMAT 1			
Transaction Code	1-3	3	'REQ'
Filler	4	1	
Version	5-6	2	last two digits of year effective
Filler	7	1	
English Dept. No.	8-10	3	
Filler	11	1	
Course 1	12-14	3	
Course 2	16-18	3	
Filler	19	1	
Speech	20-22	3	
Filler	23	1	
PE Dept Number	24-26	3	
Filler	27	1	
PE Course Number	28-30	3	
FORMAT 2			
Transaction Code	1-3	3	'REQ'
Filler	4	1	
Department Number 5-7	5-7	3	

Figure B-1 (continued)

ADDLOAD 1

ADDLOAD1 stores new occurrences of the DEPARTMENT, BS-REQ-AS, and MAJOR-COURSE records after the data base has been established. Information which can be stored is the same as that in LOAD1. Key punch instructions for the BS-REQ-AS and MAJOR-COURSE records are shown in Figure B-1. Key punch instructions for the DEPARTMENT record are shown in Figure B-2. The department number must be entered, even if it represents an old occurrence of that record, so that proper relationships can be maintained. To correct errors, delete bad record occurrences and rerun the program.

JACL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=ADDLOAD1,LIB='DS8H8.ADVD2'
```

```
//CDREADER DD *
```

```
data cards
```

```
/*
```

```
//PRTFILE DD SYSOUT=A
```

```
//SYSOUT DD SYSOUT=A
```

```
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
```

```
//ADVJRNL DD DUMMY
```

```
/*
```

KEYPUNCH INSTRUCTIONS
FOR
THE DEPARTMENT RECORD OF PROGRAM ADDLOAD1

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-9	5	
Filler	10	1	
Department Name	11-37	27	
Filler	38	1	
Type Code	39-41	3	'NEW' if new occurrence, blank otherwise

Figure B-2

DELETREC

This program is used to delete occurrences of any or all of the record types in the data base. Key punch instructions are shown in figure B-3. The input information shown is that which is required to delete the given record.

DEPARTMENT - department number

MAJOR-COURSE - major course version

BS-REQ-AS - BS requirements version

STUDENT - student ID

CREDITS - student ID

- course number

JCL necessary to execute the program

```
//GO EXEC IDMSUTIL,PROG=ELECTREC,LIB='DS8H8.ADVD2'
```

```
//CDREADER DD *
```

data cards

```
/*
```

```
//PRTFILE DD SYSOUT=A
```

```
//SYSOUT DD SYSOUT=A
```

```
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
```

```
//ADVJRNL DD DUMMY
```

```
/*
```

KEYPUNCH INSTRUCTIONSFORTHE DELETE RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department Number	5-7	3	
FORMAT 2			
Transaction Code	1-3	3	'MAJ'
Filler	4	1	
Version	5-8	4	
FORMAT 3			
Transaction Code	1-3	3	'BSR'
Filler	4	1	
BS Version	5-6	2	
FORMAT 4			
Transaction Code	1-3	3	'STD'
Filler	4	1	
Student ID	5-15	11	
FORMAT 5			
Transaction Code	1-3	3	'CRD'
Filler	4	1	
Student ID	5-12	11	
Filler	16	1	
Course number	17-22	6	Department & Course Number

Figure B-3

STUDENT

STDNTLD

STDNTLD stores new occurrences of the STUDENT record types in the data base. Key punch instructions are shown in Figure B-4. The program, STDUPTE, is used for error correction.

INPUTS:

Department ID - needed so that proper set relationships can be maintained.

Student ID

Name

Home address (street, city, state)

Previous school (type, name, city, state)

Date graduated or transferred

Date enrolled at KSU

Advisor's name

BS Degree version enrolled under

Major course version enrolled under

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=STDNTLD,LIB='DS8H8.ADVD2'
```

```
//CDREADER DD *
```

data cards

```
/*
```

```
//PRTFILE DD SYSOUT=A
```

```
//SYSOUT DD SYSOUT=A
```

```
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
```

```
//ADVJRNL DD DUMMY
```

```
/*
```

KEYPUNCH INSTRUCTIONS
FOR
THE ADD STUDENT RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'DEP'
Filler	4	1	
Department ID	5-7	3	
FORMAT 2			
Transaction Code	1-3	3	'SID'
Filler	4	1	
Student ID	5-15	11	
Filler	16	1	
Name	17-41	25	
FORMAT 3			
Transaction Code	1-3	3	'HAD'
Filler	4	1	
Home Street Addr	5-24	20	
Filler	25	1	
City	26-45	20	
Filler	46	1	
State	47-48	2	
FORMAT 4			
Transaction Code	1-3	3	'PSC'
Filler	4	1	
Type of School	5-6	2	'HS', 'JC', or 'CU'
Filler	7	1	
Name	8-37	30	

Figure B-4

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 5			
Transaction	1-3	3	'PLC'
Filler	4	1	
City School Loacted	5-24	20	
Filler	25	1	
State	26-27	2	
Filler	28	1	
Date Grad or Transfer	29-32	4	month and year
FORMAT 6			
Transaction Code	1-3	3	'DAD'
Filler	4	1	
Date Enrolled	5-8	4	month and year
Filler	9	1	
Advisor	10-29	20	
FORMAT 7			
Transaction Code	1-3	3	'CDV'
Filler	4	1	
Degree Version	5-6	2	last two digits of year effective
Filler	7	1	
Course Version	8-11	4	Department ID and last two digits of year effective

Figure B-4 (continued)

STDUPTE

STDUPTE is used to make modifications to occurrences of the STUDENT record type. Input information to this program is the same as that for STDNTLD. Blank input should be used for any fields which do not need to be changed. Key punch instructions are shown in formats 2 thru 7 of Figure B-4 and in Figure B-5. Allowance is made to enter the student ID twice so that it can be changed if necessary. The transaction code MDY indicates all changes to the record have been made and the modification of the data base can be completed.

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=STDUPTE,LIB='DS8H8.ADVD2'  
//CDREADER DD *
```

data cards

```
/*  
//PRTFILE DD SYSOUT=A  
//SYSOUT DD SYSOUT=A  
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD  
//ADVJRNL DD DUMMY  
/*
```

KEYPUNCH INSTRUCTIONSFORTHE STUDENT RECORD UPDATE PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-3	3	'FID'
Filler	4	1	
Student ID	5-15	11	
FORMAT 2			
Transaction Code	1-3	3	'MDY'
Filler	4-80	77	

Figure B-5

CREDITLD

CREDITLD stores new occurrences of the CREDITS record in the data base.

The CREDITS record contains information about courses a student has completed or is currently taking. Courses currently being taken should be entered with a blank grade field. Key punch instructions are shown in Figure B-6. Data cards must be input in the order shown in Figure B-6. Use CRDUPDTE for error correction.

INPUTS:

Student ID - needed to maintain proper set relationship

Department number of course taken

Course number

Name

Credit hours

Grade

Does course fulfill lab requirement - 'Y' or 'N'

Is course a technical elective or free elective - 'TECH',
'FREE', or blank

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=CREDITLD,LIB='DS8H8.ADV2'
```

```
//CDREADER DD *
```

data cards

```
/*
```

```
//PRTFILE DD SYSOUT=A
```

```
//SYSOUT DD SYSOUT=A
```

```
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD
```

```
//ADVJRNL DD DUMMY
```

```
/*
```

KEYPUNCH INSTRUCTIONSFORTHE ADD CREDITS RECORD PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Filler	1-2	2	
Student ID	3-13	11	
FORMAT 2			
Filler	1-2	2	
Department Number	3-5	3	
Filler	6	1	
Course Number	7-9	3	
FORMAT 3			
Filler	1-2	2	
Course Name	3-14	12	
FORMAT 4			
Filler	1-2	2	
Credit Hours	3-4	2	right justified
Filler	5	1	
Grade	6-8	3	right justified, possible grades: A,B C,D,F,WP,WF,INC,CR,N
FORMAT 5			
Filler	1-2	2	
Lab Requirement	3	1	'Y' or 'N'
Filler	4	1	
Special Requirement	5-8	4	'TECH', 'FREE' or Blank

Figure B-6

CRDUPDTE

CRDUPDTE is used to make modifications to the occurrences of the CREDITS record type. Inputs to this program are the same as those for CREDITLD. Blank input should be used for any fields which do not need to be changed. Key punch instructions are shown in Figure B-7. Provision is made for entering the department number and course number twice; once for locating the proper occurrence and a second time if either of the numbers needs to be changed. To correct errors rerun the program.

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=CRDUPDTE,LIB='DS8H8.ADVD2'  
//CDREADER DD *  
    data cards  
/*  
//PRTFILE DD SYSOUT=A  
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD  
//SYSOUT DD SYSOUT=A  
//ADVJRNL DD DUMMY  
/*
```

KEYPUNCH INSTRUCTION
FOR
THE CREDITS RECORD UPDATE PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 1			
Transaction Code	1-4	4	'STID'
Filler	5	1	
Student ID	6-16	11	
FORMAT 2			
Transaction Code	1-4	4	'FREC' for locatin proper occurrence, 'CNUM' if departme number or course number is to be changed.
Filler	5	1	
Department Number	6-8	3	
Filler	9	1	
Course Number	10-12	3	
FORMAT 3			
Transaction Code	1-4	4	'TITL'
Filler	5	1	
Course Name	6-17	12	
FORMAT 4			
Transaction Code	1-4	4	'CRGR'
Filler	5	1	
Credit Hours	6-7	2	right justified
Filler	8	1	
Grade	9-11	3	right justified possible grades A,B,C,D,F,WP,WF,I: CR, NCR

Figure B-7

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
FORMAT 5			
Transaction Code	1-4	4	'LBRQ'
Filler	5	1	
Lab Requirement	6	1	'Y' or 'N'
Filler	7	1	
Special Requirement	8-11	4	'TECH', 'FREE' or Blank
FORMAT 6			
Transaction Code	1-4	4	'MODY'
Filler	5-80	76	

Figure B-7 (continued)

ADVISRPT

ADVISRPT is the program which produces the advisor's report. This is used by faculty members when advising students on their plan of study. Input to the program is the department number and the student number in the format shown in Figure B-8. If an error causes a wrong report or no report to be produced, correct the input and rerun the program.

JCL necessary to execute the program.

```
//GO EXEC IDMSUTIL,PROG=ADVISRPT,LIB='DS8H8.ADV2'  
//CDREADER DD *  
    data cards  
/*  
//PRTFILE DD SYSOUT=A  
//SYSOUT DD SYSOUT=A  
//ADVDEPT DD DSN=DS8H8.ADVBASE,DISP=OLD  
//ADVJRNL DD DUMMY  
/*
```


KEYPUNCH INSTRUCTIONSFORTHE ADVISOR'S REPORT PROGRAM

<u>FIELD NAME</u>	<u>POSITION</u>	<u>LENGTH</u>	<u>REMARKS</u>
Department Number	1-3	3	
Filler	4	1	
Studnet ID	5-15	11	

Figure B-8

APPENDIX C
PROGRAM SOURCE LISTINGS

The sound listing have not been appended to this report due to their excessive volume. They have been placed in a special binder and are available for inspection in the Computer Science Office.

APPENDIX D
SAMPLE PROGRAM REPORTS

ILLEGIBLE DOCUMENT

**THE FOLLOWING
DOCUMENT(S) IS OF
POOR LEGIBILITY IN
THE ORIGINAL**

**THIS IS THE BEST
COPY AVAILABLE**

ADVISORS GUIDE FOR BACHELOR OF SCIENCE DEGREE
KANSAS STATE UNIVERSITY

DATE PREPARED: 07/20/76		DEPARTMENT OF COMPUTER SCIENCE		PAGE 1	
STUDENT: JOE C SMITH		ADVISOR: DR MYRON CALHOUN			
HOME ADDRESS		ENROLLMENT DATE AT KSU: 08/72		PREVIOUS SCHOOL	
STREET: BOX 925		NAME: BROKEN BOW HIGH SCHOOL			
CITY: BROKEN BOW		CITY: BROKEN BOW			
STATE: OK		STATE: OK			
		DATE GRADUATED OR TRANSFERRED: 05/72			
REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE					
A. ENGLISH		NUMBER	NAME	CREDIT HOURS	POINTS
		229-110	ENG CCMP 1	3	9
		229-120	ENGL COMP 2	3	
B. HUMANITIES AND SOCIAL SCIENCE					
SEVEN COURSES INCLUDING 1 PHILOSOPHY AND 2 ADVANCED LEVEL COURSES 1500 UP OR 2ND YEAR OF A FOREIGN LANGUAGE.					
1		-			
2		-			
3		-			
4		-			
5		-			
6		-			
7		-			
C. NATURAL SCIENCE					
FOUR COURSES INCLUDING 1 LABORATORY AND 1 COURSE ABOVE THE INTRODUCTORY LEVEL WITH PREREQUISITE IN THE SAME DEPARTMENT.					
1		265-113	GEN PHYS I	4	16
2		265-114	GEN PHYS II	4	
3		-			
4		-			
D. PHYSICAL EDUCATION					
1		261-101	CONC IN PE	1	4
2		-			
3		-			
4		-			

DATE PREPARED: 07/20/76

DEPARTMENT OF COMPUTER SCIENCE

STUDENT: JOE C SMITH

ADVISOR: DR MYRON CALHOUN

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

POINTS

GRADE

CREDIT HOURS

NAME

NUMBER
E. MAJOR COURSES

1	245-220	AN GM CAL 1	4	A	12
2	245-221	AN GM CAL 2	4		
3	286-200	FUN COMP PGM	2	A	8
4	286-202	PL/1	1	A	4
5	286-300	ALGORITHM PROC	3		
6	530-241	INTRO COMP E	3		

F. TECHNICAL ELECTIVES

1	286-201	FORTAN	1		
---	---------	--------	---	--	--

G. FREE ELECTIVES

TOTAL CREDIT HOURS: 15 CURRENT GRADE POINT AVERAGE: 3.53 CURRENT STUDENT CLASSIFICATION: FRESHMAN

DEPARTMENT	COURSE	COURSE NAME	MAJOR COURSES REMAINING HOURS	PREFREQUISITES	SEMESTER
245	224	FL AP LIA AN	03	245-220, 221	I, II, S
245	551	APP MATR TH	03	JR STANDING	I, II
286	201	FORTAN	01	286-200 (or CONCUR	I, II, S
286	203	APL	01	286-200 (or CONCUR	I, II
286	205	CJROL	01	286-200 (or CONCUR	I, II
286	305	CP ORG PGM 1	03	CS LAB/CS790 E, EE241	I
286	405	INT PGM LANG	03	CS300 OR CS790	I
286	420	OPER SYS 1	03	CS305 OR CS790	II
286	560	DATA STRUCT	03	MA220 E, CS300/CS790	I, II
286	580	NUMERIC COMP	03	CS LAB/790, MA224/551	I, II

END OF REPORT ***

DATE PREPARED: 07/21/76

DEPARTMENT OF COMPUTER SCIENCE

PAGE 2

STUDENT: JOE C SMITH

ADVISOR: DR HYRON CALHOUN

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

F. MAJOR COURSES	NUMBER	NAME	CREDIT HOURS	GRADE	POINTS
1	245-220	AN GM CAL 1	4	A	12
2	245-221	AN GM CAL 2	4	A	12
3	246-200	FUN COMP PG4	2	A	8
4	246-202	PL/1	1	A	4
5	246-300	ALGTHM PROC	3	A	12
6	246-305	CP DRG PG4 1	3	A	12
7	533-241	INTRO COMP E	3	A	12

F. TECHNICAL ELECTIVES

1	503-160	ENG COMP	2	A	8
---	---------	----------	---	---	---

G. FREE ELECTIVES

1	206-201	FORTHAN	1	A	4
---	---------	---------	---	---	---

TOTAL CREDIT HOURS: 51

CURRENT GRADE POINT AVERAGE: 3.31

CURRENT STUDENT CLASSIFICATION: SOPHOMORE

DEPARTMENT	COURSE	COURSE NAME	MAJOR COURSES REMAINING HOURS	PREREQUISITES	SCHESTER
245	224	PL AP LTR AN	03	245-220, 221	1,11,5
245	351	APP MATH TH	03	JP STAT0146	1,11
246	201	FORTHAN	01	246-200 OR CONCUR	1,11,5
246	203	APL	01	246-200 OR CONCUR	1,11
246	205	CONROL	01	246-200 OR CONCUR	1,11
286	405	INT PGM LANG	03	CS300 OR CS790	1
286	420	OPER SYS I	03	CS305 OR CS790	1
286	563	DATA STRUCT	03	MA220 & CS300/CS790	1,11
286	580	NUMERIC COMP	03	CS LAN/190, MA224/551	1,11

** END OF REPORT **

ADVISORS GUIDE FOR BACHELOR OF SCIENCE DEGREE
KANSAS STATE UNIVERSITY

DATE PREPARED: 07/21/76	DEPARTMENT OF COMPUTER SCIENCE	PAGE 1
STUDENT: JOE C SMITH	ADVISOR: DR MYRON CALHOUN	
ENROLLMENT DATE AT KSU: 08/72		
HOME ADDRESS	PREVIOUS SCHOOL	
STREET: BOX 925	NAME: BROKEN BOW HIGH SCHOOL	
CITY: BROKEN BOW	CITY: BROKEN BOW	
STATE: OK	STATE: OK	
	DATE GRADUATED	
	OR TRANSFERRED: 05/72	

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

A. ENGLISH	NUMBER	NAME	CREDIT HOURS	GRADE	POINTS
	229-110	ENG COMP 1	3	B	9
	229-120	ENGL COMP 2	3	C	6
	291-105	OPAL COMM 1	2	C	4

B. HUMANITIES AND SOCIAL SCIENCE

SEVEN COURSES INCLUDING 1 PHILOSOPHY AND 2
ADVANCED LEVEL COURSES (500 UP OR 2ND YEAR
OF A FOREIGN LANGUAGE).

1	225-110	ECONOMICS 1	3		
2	275-510	MONEY BANK	3		
3	241-101	RISC OF EOA	3	B	9
4	241-501	HEPT WEST	4		
5	241-526	AMER REVOL	3		
6	259-110	INTRO LOGIC	3	C	6
7					

C. NATURAL SCIENCE

FOUR COURSES INCLUDING 1 LABORATORY AND 1
COURSE ABOVE THE INTRODUCTORY LEVEL WITH
PREREQUISITE IN THE SAME DEPARTMENT.

1	221-210	CHEM 1	4	B	12
2	245-222	AN GM CAL 3	4	A	16
3	265-113	GEN PHYS I	4	A	16
4	265-114	GEN PHYS II	4	B	12

D. PHYSICAL EDUCATION

1	261-101	CONC IN PE	1	A	4
---	---------	------------	---	---	---

DATE PREPARED: 07/21/76

DEPARTMENT OF COMPUTER SCIENCE

PAGE 2

STUDENT: JOE C SMITH

ADVISOR: DR MYRON CALHOUN

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

5. MAJOR COURSES	NUMBER	NAME	CREDIT HOURS	GRADE	POINTS
1	245-220	AN GM CAL 1	4	B	12
2	245-221	AN GM CAL 2	4	B	12
3	245-274	EL AP LIN AN	3	A	12
4	286-200	FUN COMP PGM	2	A	8
5	286-202	PL/1	1	A	4
6	245-300	ALGORITHM PROG	3	A	12
7	286-305	CP DRG PGM 1	3	A	12
8	245-405	JIT PGM LANG	3	A	12
9	286-420	OPER SYS 1	3	A	12
10	286-560	DATA STRUCT	3		
11	245-500	NUMERIC COMP	3		
12	530-241	INTRO COMP E	3	A	12

6. TECHNICAL ELECTIVES

1	245-240	SER DIFF EQU	4		
2	245-551	APP MATR TH	3		
3	286-306	OP SYSTEM	3	B	9
4	286-505	CP DRG PGM 2	3		

G. FREE ELECTIVES

1	221-220	CHEM 2	4	B	12
2	286-201	FORTRAN	1	A	4
3	500-160	ENG COMP	2	A	8

TOTAL CREDIT HOURS: 67 CURRENT GRADE POINT AVERAGE: 3.33 CURRENT STUDENT CLASSIFICATION: JUNIOR

DEPARTMENT	COURSE	COURSE NAME	MAJOR COURSES REMAINING HOURS	PREREQUISITES	SEMESTER
245	551	APP MATR TH	03	JR STANDING	1,11
286	201	FORTRAN	01	286-200 OR CONCUR	1,11,5
286	203	APL	01	286-200 OR CONCUR	1,11
286	205	COBOL	01	286-200 OR CONCUR	1,11

** END OF REPORT **

ADVISORS GUIDE FOR BACHELOR OF SCIENCE DEGREE
KANSAS STATE UNIVERSITY

DATE PREPARED: 07/22/76 DEPARTMENT OF COMPUTER SCIENCE

PAGE 1

STUDENT: JOE C SMITH

ADVISOR: DR MYRON CALHOUN

ENROLLMENT DATE AT KSU: 08/72

HOME ADDRESS

PREVIOUS SCHOOL

STREET: 765 S LAWTON DRIVE
CITY: OKLAHOMA CITY
STATE: OK

NAME: BROKEN BOW HIGH SCHOOL
CITY: BROKEN BOW
STATE: OK
DATE GRADUATED:
OR TRANSFERRED: 05/72

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

A. ENGLISH

CREDIT HOURS

GRADE

POINTS

COURSE	NAME	CREDIT HOURS	GRADE	POINTS
229-110	ENG COMP 1	3	A	9
229-120	ENG COMP 2	3	C	6
231-105	ORAL COMM 1	2	C	4

B. HUMANITIES AND SOCIAL SCIENCE

SELECT COURSES INCLUDING 1 PHILOSOPHY AND 2
ADVANCED LEVEL COURSES (500 UP OR 2ND YEAR
OF A FOREIGN LANGUAGE).

COURSE	NAME	CREDIT HOURS	GRADE	POINTS
1	225-110 ECONOMICS 1	3		
2	225-330 MONEY BANK	3	A	9
3	241-101 RISE OF EUR	3	B	9
4	241-501 HEPT WEST	4	C	8
5	241-526 AMER REVOL	3	C	6
6	259-113 INTRO LOGIC	3	C	6
7	259-220 SYM LOGIC	3	B	9

C. NATURAL SCIENCE

FOUR COURSES INCLUDING 1 LABORATORY AND 1
COURSE ABOVE THE INTRODUCTORY LEVEL WITH
PREREQUISITE IN THE SAME DEPARTMENT.

COURSE	NAME	CREDIT HOURS	GRADE	POINTS
1	221-210 CHEM 1	4	B	12
2	245-222 AIR GM CAL 3	4	A	16
3	265-113 GEN PHYS I	4	A	16
4	265-114 GEN PHYS II	4	B	12

D. PHYSICAL EDUCATION

COURSE	NAME	CREDIT HOURS	GRADE	POINTS
1	261-101 CONC IN PE	1	A	4

DATE PREPARED: 07/22/76

DEPARTMENT OF COMPUTER SCIENCE

PAGE 2

STUDENT: JOE C SMITH

ADVISOR: DR MYRON CALHOON

REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE

POINTS

CREDIT HOURS

GRADE

NAME

E. MAJOR COURSES

NUMBER

1	245-220	AN GM CAL 1	4	B	12
2	245-221	AN GM CAL 2	4	B	12
3	245-224	CL AP LIT AL	3	A	12
4	246-200	FUN COMP PGM	2	A	8
5	246-202	PL/1	1	A	4
6	246-303	ALGORITHM PROC	3	A	12
7	246-305	CP RPG PGM 1	3	A	12
8	246-405	INT PGM LANG	3	A	12
9	246-423	OPEN SYS 1	3	A	12
10	246-570	DATA STRUCT	3	A	12
11	246-583	NUMERIC COMP	3	B	9
12	533-241	INTRO COMP E	3	A	12

F. TECHNICAL ELECTIVES

1	245-240	SFR DIFF EQU	4	A	16
2	245-511	INT ALG SYS	3	A	12
3	245-553	IN COMP AU	3	A	12
4	245-551	APP MATR TH	3	A	12
5	245-552	OP FCALL PDE	3	B	9
6	245-555	NUM ANAL	3	B	9
7	246-660	INT SOLT ENG	3	A	12
8	246-650	D AUT DIG SV	3	B	9
9	246-670	DIS COMP ST	3	B	9
10	246-670	PROJECTS	3	A	12

G. FREE ELECTIVES

1	221-230	CHEM 2	4	B	12
2	246-201	FORTRAN	1	A	4
3	246-306	OP SYSTEM	3	B	9
4	246-505	CP DRG PGM 2	3	A	12
5	503-160	ENG CONC	2	A	8

TOTAL CREDIT HOURS: 123 CURRENT GRADE POINT AVERAGE: 3.35 CURRENT STUDENT CLASSIFICATION: SENIOR

MAJOR COURSES REMAINING

DEPARTMENT	COURSE	COURSE NAME	APP MATR TH	HOURS	PREREQUISITES	SEMESTER
245	551	APP MATR TH		03	JR STANDING	1,11
246	201	FORTRAN		01	246-200 OR CONCUR	1,11,5
246	203	APL		01	246-200 OR CONCUR	1,11
246	205	C980L		01	246-200 OR CONCUR	1,11

** END OF REPORT **

IMPLEMENTATION OF A PROTOTYPE DATA BASE FOR
ADVISING COMPUTER SCIENCE STUDENTS

by

HARVEY A. LONG

B.S., Sterling College, 1966

AN ABSTRACT OF A MASTER'S REPORT
submitted in partial fulfillment of the
requirements for the degree
MASTER OF SCIENCE
Department of Computer Science

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1976

The purpose of this paper is to show the implementation of a data base using the Integrated Database Management System (IDMS) which will contain current academic information on all students majoring in computer science.

This implementation will produce a report that will show the students progress toward fulfilling the requirements for a Bachelor of Science Degree. The basis for determining the requirements for a degree is the University's 1975-1976 General Catalog. The report will show courses that have been taken and which requirements they fulfill and indicate requirements that remain. Also included will be credit hours, a grade for each course and current grade point average for all courses. This information will be made available to faculty members to aid them in counselling students about their course of study.

The system is intended to be on-line so that each student will be able to keep his own records current.