To the Graduate Council:

I am submitting herewith a thesis written by Ann Wayburn entitled "The Computer Science Database: A Comprehensive System for Student Information and Registration." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Charles P. Pfleeger, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

The Graduate School

THE COMPUTER SCIENCE DATABASE: A COMPREHENSIVE SYSTEM FOR STUDENT INFORMATION AND REGISTRATION

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Ann Wayburn

December 1984

To Bill

ACKNOWLEDGMENTS

I am indebted and grateful to the Department of Computer Science for preparing me for a career in computer science and for exposing me to the joys of teaching. I am especially grateful to Charles Pfleeger, David Straight, and Maria Leech for their endless patience, guidance and confidence while I developed this Computer Science Database System.

There are several people who deserve extra recognition. My friend, Wallace Mayo, has made contributions to my career too numerous to mention. Thanks. Chuck and David have pushed me from the beginning. I am grateful for their attention. Doyt Perry has been a model of responsibility and professionalism. I appreciate his enduring encouragement. Ethel Wittenberg smoothed my career as a teaching assistant for two and one-half years and continues to do everything she can to show me the ropes. Suzie Witenbarger has flawlessly prepared almost every letter and document I have written including this thesis. I truly appreciate her work and her contributions to mine.

The University of Tennessee Computing Center provided the computing resources necessary for this project. In integrating data from various campus agencies, I have gained valuable experience in the co-operative aspects of application development.

My husband, Bill Bryan, has been supportive, tolerant and loving. His parents, Elizabeth and Louis, have been the parents I always wanted. Bettina Hansen has been my silent partner. I owe them more than I could ever hope to repay.

To all my friends and colleagues--Thank You.

ABSTRACT

This thesis describes the Computer Science Database (CSDB), which was developed to be used by the faculty and staff of the University of Tennessee Computer Science department to address the information needs of the department. The CSDB system contains student and curriculum/registration information. The data kept can provide student information for advisors as well as the department regarding grades in computer science classes, progress through the major/minor program and pertinent background information. The thesis contains a description of the database and a user's guide to aid the beginning user. A database manager's manual for maintenance and further enhancements of the database is also provided.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

ABBREVIATIONS

CARS	Computer Assisted Registration Services
CM	Command Module
CMS	Conversational Monitor System
CSDB	Computer Science Database
DBA	Database Administrator
DEC	Digital Equipment Corporation
DI	Data Index
DS	Data Set
GPA	Grade Point Average
IBM	International Business Machines
OIS	Office of Information Systems
PM	Process Module
SD	Schema Definition
SDA	Student Data Analysis
SRM	Student Record Master
UTCC	University of Tennessee Computing Center

INTRODUCTION

1. DESCRIPTION OF THE COMPUTER SYSTEM

The Computer Science Database (CSDB) is a menu-driven system of programs that maintain a database of information for the Computer Science Department at The University of Tennessee, Knoxville. The project was developed in response to information management needs of this department.

The database includes student records and registration information. Each student record includes a list of all computer science courses taken by the student. university master record information and a profile of the student's academic standing within the Computer Science Department. The registration information includes courses offered by the Computer Science Department. student registration requests, class rolls, and waiting lists for the coming quarter.

The CSDB can produce a variety of student reports. including mailing labels, condensed transcripts, and complete student record displays, or a simple list of student names. There are several categories of students for which these reports can be produced. Some examples include: undergraduate majors or minors, graduate majors, co-op students, and students receiving warning letters or on probation. In addition to student reports, the CSDB system can produce registration reports such as class rolls and

waiting lists. or a list of all courses offered in a given quarter.

The CSDB system integrates student and registration information at the time the course requests are made by students. Every student is evaluated at the beginning of each quarter. Points are awarded to students to reflect overall successful progress in the Computer Science curriculum with the highest number of points going to the most successful students and to those who are nearing completion of the program. The objective of the registration point system is to give priority, at the time of registration, to those students whose past performance indicates a high probability of completing the degree requirements. As student requests for courses are processed by the CSDB system. class sections are filled based on the registration points. This technique will first fill classes with previously successful candidates and automatically order the remaining students for possible admission from a waiting list.

The whole project--database manipulation programs, report modules, and system management routines--was designed to be an integrated, "user-friendly" system. For this reason a menu-driven system was selected, and careful attention was given to making the system easy to use.

2. DESCRIPTION OF THE CHAPTER CONTENTS

This project was developed in response to the needs of the Department of Computer Science at The University of Tennessee, Knoxville The demands of a program with many interested students and limited faculty resources prompted discussions of information management needs for the Computer Science Department. In Chapter One, the needs of the department and resources available will be discussed.

Chapter Two contains a description of the host computer system and the development of the CSDB project. The CSDB project encompasses data storage and manipulation modules, report producing routines and system modules for printing reports, backing up the system, and reading and writing tapes. The project is menu-driven for ease of use by individuals at varying levels of expertise. In addition to the menu-driven system, the data can be accessed directly through Accent-R, the host database manager, via a command-driven interactive system.

Chapter Three is the User's Guide to the CSDB system. This guide is designed to familiarize the user with the system capabilities. Samples of the menus and prompts are included to familiarize the user with the system. A map of the menu system is provided with an index that indicates which function should be used to accomplish a specific task. A general time line is discussed as a guide for ordering database operations.

Chapter Four is the Database Administration (DBA) manual. This manual will be used by the system manager to maintain and upgrade existing programs or modules. This chapter contains the record formats. naming conventions, resource contacts, and the program module names and descriptions. This section also includes information regarding Accent-R, the host database manager.

Many ideas for extending the system could not be incorporated into this thesis project. Chapter Five addresses the ideas and modifications that could make this project a more powerful production system. Included are suggestions of additional data to the maintained, statistical analysis of student data. procedures to provide course scheduling projections, systematic checking of student prerequisites. and the possibility of the use of a similar system by other departments. System modifications to improve performance and security enhancements are included in this discussion.

Finally, Chapter Six includes a summary of the system capabilities and limitations. The author's ethical concerns about this project are raised.

CHAPTER 1

THE COMPUTER SCIENCE DEPARTMENT'S NEEDS

The Computer Science Department has undergone several changes in the last few years. The most significant change has been the increased demand for classes by undergraduate and graduate students. The combination of this demand with the limited resources has made it necessary for the department to look for ways to manage enrollment in its courses.

Serious discussion of a management information system to support this effort began in the Spring of 1983. At that time the department had extremely overcrowded classes, a labor intensive system for managing the waiting lists of students wanting to get into closed classes and complicated requirements for admission to the major. Classes were being filled with students in decreasing order of cumulative hours passed. That is, the method used by The Office of Computer Assisted Registration Services (CARS) for filling classes gives priority based on the number of hours a student has accumulated. This resulted in freshman-level classes of introductory computer science being filled with seniors first, and freshmen last. This situation caused many late starts into the major program because students with few hours were unable to get into classes. At that same time, many upperclassmen who had no intention of being computer

science majors were using the lower level classes designed for majors to "get a little experience with computers." Clearly the department's goals were not being met.

To control admission to the introductory classes, the department began taking applications for the first entry level courses. This partially relieved the problem. There were still overcrowded conditions in other lower division classes. In the past, there was no limit on the number of times a course could be repeated other than University regulations. Students kept registering again and again. This overcrowded classes with undesirable candidates. Again, the CARS priority system enhanced the enrollment odds of the marginal students, since they had accumulated an abundance of hours.

Effective Winter Quarter of 1982 the Computer Science Department invoked serious procedures to control enrollment in its classes. The following passage is from The Undergraduate Catalog of The University of Tennessee.

> Students who have completed the required lower-division courses with a minimum GPA of 3.0 and wish to progress to the major or minor program must apply to the departmental office. This must be done as soon as the stated requirements are met so that a decision can be reached prior to the Advanced Registration date for the next quarter. Those who are not accepted into the C.S. Degree program will be counseled and advised of educational alternatives.
>
> For computer science majors who have taken at least three computer science courses at UTK, grades in all Computer Science courses from UTK will be averaged. If a course is repeated, all

> grades received for this course will be counted. If a course in which a student has received an A or B is later retaken, only the first A or B is counted; no subsequent grades will be counted.
>
> If a student's computer science average as described above falls below 2.5, the student will be given a warning. If after one more quarter's grades have been received the student's Computer Science average has not risen to 2.5, the student will not be allowed to graduate with a major in computer science.
>
> A student who desires to be readmitted to the major after being withdrawn as described above must attain an average in computer science courses (computed as described above) of at least 2.7.

These requirements were devised to reduce the number of less successful students and at the same time strengthen the quality of the major.

Soon, the department became aware of the record-keeping load and its impact on the department. It was clear that this work needed to be done by a computer. Discussion of this project began in the Spring of 1983, six months after this policy was put into effect. The campus office of Student Data Analysis (SDA) was able to help with information on majors but could not address the problem of identifying or obtaining the records of pre-majors, i.e., students taking computer science courses but not yet admitted to the major. SDA was also unable to provide any information on graduate majors or pre-majors. Likewise, Administrative Data Services, now The Office of Information Systems (OIS), had difficulties getting the grades of the particular students desired. Currently, the CSDB system

cannot gain access to all the student records needed. The method used gets records for majors in computer science only. This does not include a substantial number of pre-majors because there is no way to identify these students as a group or on an individual basis.

The CSDB system was an idea devised to deal with these and other administrative problems. Both student data analysis and enrollment restrictions are addressed by this system.

Currently each request for a report from SDA, OIS, or CARS is handled separately. Because of this, the department has occasionally not received a response to a request, or has received a limited or incomplete report. By getting data from SDA and OIS on a regular quarterly basis, the department will be making only one request of each agency, but will have the ability to manipulate this data in a variety of ways without additional contact with these outside agencies. Quarterly data retrieval from CARS, after preregistration, completes the requests for data.

CSDB contains registration modules to further the effort in limiting enrollment. The CSDB will get student requests for classes from CARS. By gaining cooperation from CARS the department can establish its own scheme for prioritizing students for classes. The algorithm for prioritizing students for admission into computer science courses gives points to students based on past performance

in computer science classes, overall GPA and class standing. The CSDB ranks all students requesting to enroll in a given class by integrating student course requests with other information regarding the student. Students within the specified class limits will be admitted. The remaining names will form an ordered waiting list, and are returned to CARS to be excluded from the class roll. Should any enrolled student voluntarily withdraw from a class at a later time- the waiting list provides an ordered list for the instructor of the next most appropriate student requesting that class.

All of these data transfers can be achieved in machine readable format so that the only clerical activity is in determining the original list of student identification numbers for whom reports from SDA and OIS are requested. At this time- the CSDB is designed to use existing retrieval procedures that will retrieve records and grades for undergraduate majors only.

CHAPTER 2

PROJECT OVERVIEW

1. HOST COMPUTER SYSTEMS

Originally the CSDB project was planned for development using the PL/I language and the Conversational Monitor System (CMS) available on the IBM computer system at UTCC. During the early phases of project planning all modules were designed, and inputs and outputs were outlined. The plan to produce the system using interactive PL/I and stored command modules or automatically executed files looked very much like the system appears today. However, the lack of flexibility in this approach became evident. The complexity for users was an undesirable factor. During the development period, UTCC acquired a new system facility. Accent-R, a relational database manager, produced by National Information System was brought on line on the DEC-10. This was the first interactive relational database available on a mainframe at UTCC. After considering the alternatives, the decision was made to drop development on the IBM and to move to Accent-R on the DEC-10. Accent-R was selected for the development of the CSDB application modules because of its power as a relational database tool and because of the flexibility it provides when used to develop complete applications for end users.

Still, some modules remain in PL/I, although not on the

CMS system. Data tapes sent to the department from outside agencies, specifically OIS. SDA and CARS, are read using PL/I programs. Modules in PL/I also produce the machine readable data the department sends to CARS. PL/I was selected because it has excellent record input/output facilities and tape access using the IBM is a relatively straight-forward matter. The PL/I programs that read the tapes from SDA. OIS. and CARS are maintained on the DEC-10 and submitted for execution to the IBM 3081 processor at UTCC using the interface between the DEC-10 and the IBM 3081. The results of these programs are routed back to the CSDB area on the DEC-10 and can easily be loaded into CSDB datasets from within the CSDB menu system.

2. DATA ORGANIZATION AND MANIPULATION

The CSDB system maintains several diverse groups of data. The Accent-R database management system keeps data organized in flat files or data sets. That is, a relation can be viewed as a flat table as shown in Figure 2.1. The schema for this table is also given as an example (see Figure 2.2). The various views or logical organizations of a data set are managed by a data indexing system. For each view, a different index is maintained. Each alteration or entry to the data set automatically updates the data index. The index is kept in a B-tree format with the location of the actual record as the leaf elements. This method allows

NAME	ID	AGE . . .
JONES	190-60-5947	19
BROWN	375-26-5727	20
SMITH	495-87-1352	18
JONES	234-09-4295	20

FIGURE 2.1

Sample Data Relation

IDENTIFIER	TYPE
NAME	CHARACTER LENGTH 10
ID	CHARACTER LENGTH 11
AGE	INTEGER
.	.
.	.
.	.

FIGURE 2.2

Sample Schema Definition

various views to be maintained without repeatedly sorting data or storing multiple copies of the data for quick data access.

There are three files of student data and two files of registration information. For each dataset, the information maintained, the source of the data and where it is ultimately kept within CSDB are discussed below.

Permanent student records within the CSDB are comprised of data from the student record master (SRM), grade reports in computer science courses, and profile indicators generated by the manipulation of the data. The desired SRM records are written to tape by SDA. These are pulled from the university student record database. A PL/I program (READ.SDA) reads that tape. picks off the pertinent parts and forms a system file within the CSDB directory area on the DEC-10. Then, before any updating is done, the user must load the data into the SDA.DAT file which has the SDA.STUD.TAPE schema definition (SD) (see Appendix A).

The grade reports on all relevant computer science courses are written to tape by OIS. A PL/I program (READ.OIS) run on the IBM 3081. reads and reformats the OIS data and writes a system file to the area belonging to CSDB on the DEC-10. Before the quarterly updating is done, the user must load this data into the OIS.DAT file using the OIS.GRADE.TAPE schema definition (see Appendix A).

With the OIS.DAT and SDA.DAT files in place, updating

modules read and rewrite the CSDB.DAT file with updated information. This file houses the full student records used by the CSDB programs. The data from SDA and OIS are merged, and certain indicator fields are verified and altered, if necessary. The schema definition is CSDB.PERM.REC. This schema keeps the data logically separated into student information, generated departmental indicators, and computer science grade reports (see Appendix A). Student information includes name, address, phone number, etc. Departmental indicators include computer science grade point average (GPA), major standing, etc. These are generated according to current departmental policy. The computer science grade report includes course number, quarter taken, grade earned, etc. for each class taken in computer science and cross listed courses. The student information is obtained from SDA, indicators are generated by the department and grade histories are provided by OIS. Any field of this record may be altered interactively by the user. However, it is intended to be maintained systematically. Various data index (DI) files are maintained to implement ordered access to this data; for example, the data can be presented in alphabetic order by student name or in numeric order by student identification number, as well as several other ways.

Registration information is kept in two files. The sections offered by the Computer Science Department are

defined in the SECT.INFO schema definition (see Appendix A). The actual data is in the SECT.DAT file. This data must be entered by hand every quarter. Fortunately, this only amounts to forty or fifty entries of eight items each. The other seqment of registration information is course requests made by students. CARS will write a tape of all student requests for each section offered by computer science. This tape is accessed by a PL/I program (READ.CRS) and the data is reformatted for interaction with the CSDB registration modules. A system file is formed within the CSDB directory on the DEC-10. This data is then loaded into the CARS.DAT file with the schema definition CARS.ROLL.WAIT.TAPE (see Appendix A).

This data is then augmented with registration points for each student. Reqistration points are calculated according to an algorithm developed to model the hand selection process (see Figure 2.3). The specific points allocated to different categories can be changed as conditions in the department change. The registration points information is kept in CSDB.DAT.

With everything now prepared, the ORDR routine determines the relative position of each student based on registration points. Those students who cannot be accommodated in the class are deleted by writing each student identification number and requested section number to a tape which CARS reads and uses to remove the indicated

Set registration points to zero

+ 200 points to Graduate Majors and Undergraduate Majors

+ 190 points to Graduate with Nondegree status and Minors

+ 180 to Graduates--to give an advantage over Undergraduates

+ Cumulative Hours Passed

+ 9 to CO-OP students and Graduate Teaching Assistants

+ ((CS.GPA-2.00)*(CS-HRS/45))*40 up to maximum of 60
(If Undergraduate or Special)

+ ((CS.GPA-3.00)*(CS-HRS/45))*40 up to maximum of 40
(If Graduate)

+ 9 (If Senior Undergraduate and overall GPA >= 3.5)

+ 6 (If Junior Undergraduate and overall GPA >= 3.5)

+ 3 (If Sophomore Undergraduate and overall GPA >= 3.5)

FIGURE 2.3

Registration Points Algorithm

students prior to their being enrolled. The CSDB system maintains an IN.OUT mark for each student in the CARS.DAT file. The waiting list is formed using this mark. The mark indicates whether a student is in a particular course or not.

The data fields kept in the CSDB system are easily modifiable using Accent-R. Enhancements, mentioned in Chapter Five, can be developed swiftly and smoothly using Accent-R. The reports that can be produced using the data described in the above paragraphs are discussed later in this chapter.

3. DEC-10 FACILITIES USED BY CSDB

Accent-R permits system calls to execute programs outside of the database manager, but re-entry to Accent-R is impossible on the DEC-10. Therefore, there are several CSDB facilities which are kept outside of the menu system.

There are CSDB programs to read and write various tapes, print certain files or backup/restore the system. The read/write tape jobs are stored programs that can execute independently if the proper command is given to the DEC-10. At that time the programs will be sent to the IBM for execution. BACKUP is a facility provided by UTCC. This facility is used to copy the complete CSDB directory area to tape. In the event of a problem or system failure, this backup tape could be used to reload the database from a

previously working condition. In order to protect the data fully, the BACKUP facility should be used after every alteration is made to the database. Additionally, BACKUP should be used before new data is loaded from tape during the massive update each quarter. Another system command, .PRINT. is used to print reports that have been generated by CSDB on UTCC line printers. The DEC-10 print command and UTCC facilities must be used because Accent-R has no builtin print command.

4. MENU SYSTEM

All of the CSDB system is directed by menus except the few system features previously named. The menu system operates from within Accent-R and is invoked automatically upon login to the CSDB account. The menu system is stored as records in a data file with schema definition MENU.SD. Each menu item is an entry in the MENU.DS. Each menu item has an associated help field, a series of commands to be executed, a parent, a next menu and a previous menu. By using this design the menu system can be driven in a hierarchical manner. For the more advanced user, special commands can be used to speed up menu traversal. The help command will give a listing of the various commands available for use from within the menu system. The help page for each menu item is a brief section of text designed to clarify the actions to follow.

In general, the menu system is divided into student record processing and registration activities. On the student side there are basically two activities to consider: the update modules and query modules. Updating can be done for the whole student file in batch mode, or for one record in interactive mode. Queries can be directed to the file or to one record. File directed queries return a report containing all students in a given category, or students arranged in a specified order. The report can have several formats. Record directed queries will present a choice among a variety of views of the student record for a specific student. For example, queries regarding students might include:

1. an alphabetical list of majors with their grades in computer science courses;
2. a list of computer science majors in zip code order on mailing labels;
3. a specific student record (by student name);
4. a specific student record (by ID number) giving only departmental indicators.

The registration modules can be used to order the names of students requesting courses. The user can also create or change a class section, or print a report of a single section or class roll information. The report formatter can produce a class roll, the waiting list, or section information, or all three for one class section or for all

class sections available at that time. A complete discussion of the reports available follows.

5. REPORTING

The CSDB system has extensive reporting facilities. The reports available that contain the student information fall into four general forms. Each report can be generated for a group of students or for students in a specified order. The mailing label report produces mailing labels for a category of students in zip code order. The name report will merely produce an ordered list of names of all students or students in a particular group. A grade report will produce an abbreviated transcript showing a student's computer science grades. A full report will display all known information for the student or students in question: name. grades, overall university and department information (see Appendix B). In addition. the user can obtain a count of the number of students in a group without printing any report.

The registration side of CSDB also produces reports. A report can be produced for one specific section or for all sections in the current quarter. In either case. the report can include section information such as classroom and instructor. Class rolls, waiting lists or both can also be included with this section information.

All reports in the student and registration areas can

be listed to the screen. sent to system files for printing on UTCC printers. or both.

CHAPTER 3

USER'S GUIDE

Before reading this section the user should have at least a casual acquaintance with the DEC-10 computer although perhaps not with Accent-R.

1. LOGGING IN AND LOGGING OUT

All programs, data sets and data indices for the CSDB system reside within the CSDB's DEC-10 directory area. The user will login to the CSDB account using the usual DEC-10 login procedures. The password should be kept as secure as possible to prevent access by unauthorized users. Logging out is accomplished with the .K command. Once logged in, the user has several options from which to choose. These are discussed in the remainder of this chapter.

2. DEC-10 MONITOR COMMANDS

Invoking Accent-R

The most usual course of action will be to use the monitor level command .R ACCENT to invoke the CSDB menu-system. This will be described later.

Backing Up the System

.R BACKUP will invoke the system backup to save the CSDB system on tape. This facility is fully documented by UTCC.

Deleting Files

Use the command .DEL file.ext to remove the file with the name "file.ext." The user should exercise caution with this command when removing unneeded files.

Print Options

.PRINT is the command used to print any desired reports on UTCC line printers. This command is also documented by UTCC. For example. .PRINT file.ext will print a file named "file.ext." Several options can follow the file name. An example of the use of an OPTION switch is shown below.

.PRINT file.ext/OPTION:LABELS

The switch /OPTION:LABELS when placed after file.ext will automatically have mailing labels mounted on the printer and the output routed back to Remote 2 for delivery. The switch /OPTION:TEXT will have output printed on letter size paper and routed back to Remote 2. The switch /OPTION:QUICK will have output printed on greenbar at Remote 2. In any case. the user will be notified when the print command has been completed by the DEC-10. At that time. the user may safely delete any report files no longer needed.

Submitting Files to the IBM

.HS is another command needed at the monitor level. This command will submit a program to the IBM 3081 for execution. The .HS command can be used to execute programs on the IBM 3081 that read tapes. For example. .HS READ.ext

would be used to read either the OIS. SDA or CARS tape; use the three letters OIS. SDA. or CRS that indicate the source of the tape you desire to read. Normally each tape would be read only once per quarter. This will be covered more thoroughly in the section Using the Menu System.

After you enter the .HS command, there is no need to wait. You may continue with other CSDB activity or logoff. You could return later to the system and verify visually with the .DIR command that the desired data has appeared in the CSDB directory area. The tape writing activity performed on the IBM side is also invoked by the .HS command. For example. to write the tape sending CARS the student identification numbers and course section numbers, you use the following command.

.HS WRITE.CRS

This will start that tape program. You needn't wait for completion of this job.

To review. the commands you will use are:

1. .R ACCENT to invoke the menu system.
2. .R BACKUP to save or restore tape backups.
3. .DEL file.ext to delete a file named "file.ext".
4. .PRINT file ext/OPTION:opt to print a report.
5. .HS READ.ext to read a tape.
6. .DIR to see the file names in the CSDB directory.
7. .HS WRITE.CRS to prepare the tape of names to be deleted.

3. ABOUT THE MENU SYSTEM

The Menu Options

When the user executes the monitor level command

.R ACCENT

the menu-system will be invoked automatically. The main menu will appear on the screen (see Figures 3.1 and 3.2).

```
[MAIN]

ITEM     DESCRIPTION

USER....CSDB MENUS
DBA ....NATIVE ACCENT-R
QUIT....EXIT

Enter ITEM or HELP :
```

FIGURE 3.1

Main Menu

```
[USER]

ITEM     DESCRIPTION

REG ....REGISTRATION PROGRAM
STUD....STUDENT INFO PROGRAM
QUIT....EXIT CSDB

Enter ITEM or HELP :
```

FIGURE 3.2

User Menu

For this and every other menu you may invoke your choice from a series of items or commands. You can choose one of the items presented on the menu or you can use any of the menu commands shown (see Figure 3.3). To view these menu commands on your screen, you should type HELP when responding to the menu prompt.

```
Enter ITEM or HELP:

HELP - THE FOLLOWING COMMANDS ARE AVAILABLE
BACK     Returns you to parent menu
GOTO     Transfers you to a certain menu
HELP     Prints this text
HOME     Returns you to home menu (NOTE MARK command)
LAST     Returns you to the previous menu
LIST     Prints a listing of all menus in system
MARK     Set current menu as home
QUIT     Returns you to native ACCENT-R
SHOW     Redisplays current menu
TIME     Shows current time of day
?ITEM    Prints any associated help information for item

Enter ITEM or HELP:
```

FIGURE 3.3

Menu Commands

For each menu item there is further explanation available. To see this, type "?" before the item name and you will see the additional text describing the action for the indicated item.

Leaving the Menu System and Quitting Accent-R

To leave the menu system. use the QUIT command. This will return you to native Accent-R. The prompt will be an asterisk "*". You must QUIT aqain to return to the DEC-10 monitor prompt '.'. When quitting Accent-R, you will want to clean up any temporary storage space used. You can do this by giving the *QUIT CLEAN command. You may notice a delay before being returned to the DEC-10 monitor prompt.

4. USING THE MENU SYSTEM

Menu Tree Overview

The menu-system can be viewed as an inverted tree. The top of the menu system is the trunk of the tree. The MAIN menu is the highest menu (see Figure 3.4). The menu selections that invoke database activities, as opposed to another menu, are known as "leaf nodes." The menu selections that invoke other levels with menus are known as "menu nodes."

Description of Each Menu Option

STUD will take you to the student portion. STUD, of the database. There are two primary activities to be considered, updates and queries.

UPDT will take you to the update portion of the menus. In updating the database. there are three menu choices.

LOAD will cause system files (already prepared) that

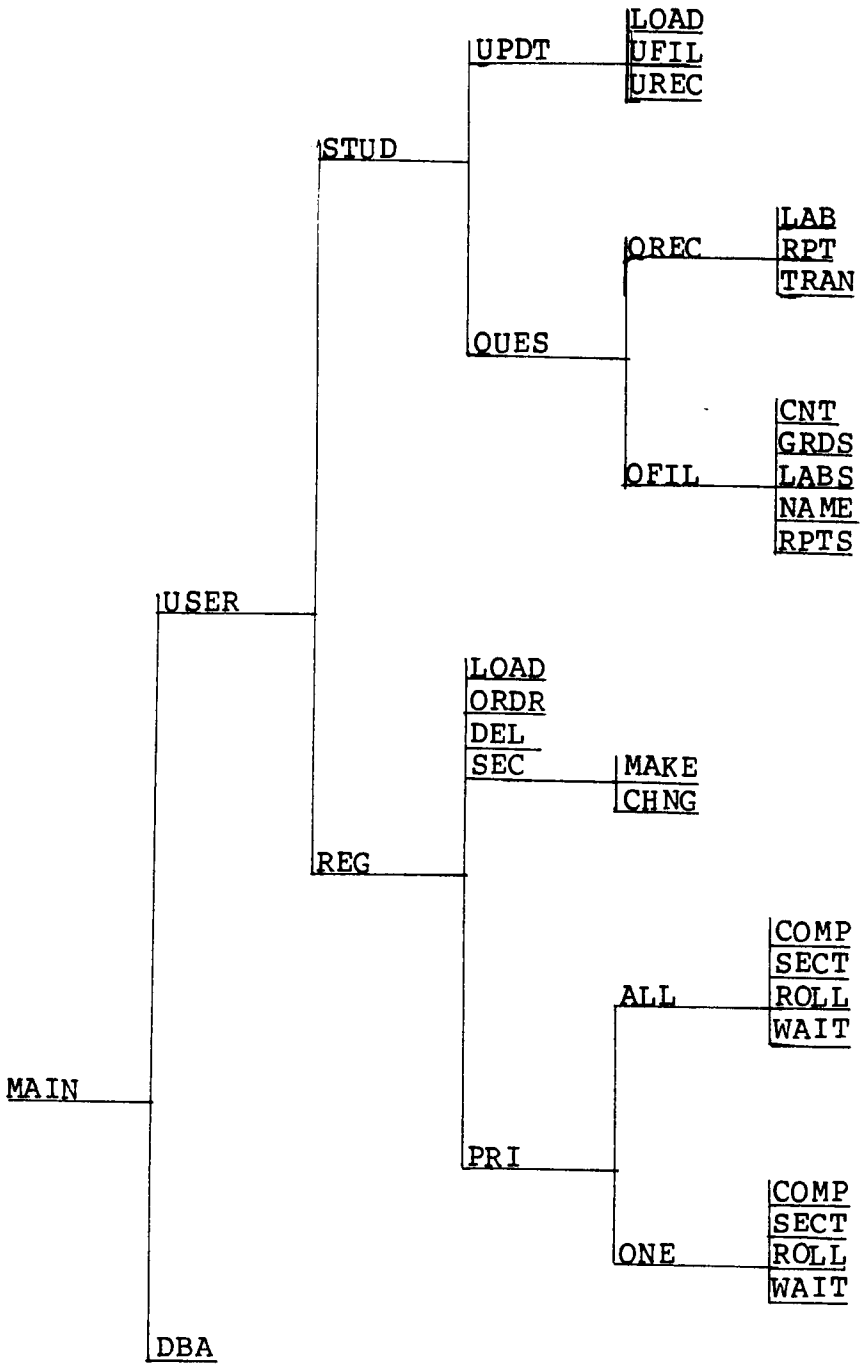

FIGURE 3.4

Menu Tree With Item Names

contain data from the SDA and OIS to be loaded into database files.

UFIL will do automatic updating of the student database with new information. New data will be written to the permanent student records. Departmental indicators are determined, grade averages are calculated and registration points are determined. This is a long and somewhat slow procedure. You will be given messages to indicate progress through the phases of this module.

UREC will allow you to interactively modify any one student record. You will first be asked whether you wish to identify the student record by student name or student ID number. Next you will be prompted to indicate what part of the student record you wish to modify.

QUES will take you to the querying portion of the menu system. When querying the database, you have the option to look at one student record or a group of student records.

QREC is the menu of options for querying the database regarding one particular student. You will first have to identify the student. This is the same procedure as found in updating one record. After identifying the student you have three options regarding the data you can view:

LAB will produce a mailing label for that student.

RPT will produce a full report of the student record.

TRAN will produce an abbreviated transcript for that student.

QFIL is the menu option for querying the entire collection of student records for a particular grouping or ordering of students. In any case, you will be prompted to indicate the category or ordering for the desired response.

CNT will count the number of students in a certain group.

GRDS will produce abbreviated transcripts for students in a particular category.

LABS will produce mailing labels for students in a group. The labels will be in zip code order.

NAME will produce a list of all student names in a certain order or all students in a certain group.

RPTS will produce a full report for each student identified as belonging to a named group.

In each case where a report is produced you will have the option of getting the output on the screen, in a file, or both. Screen output can be redirected to an adjacent hardcopy device. Files produced and placed in the CSDB directory area can be printed using the DEC-10 .PRINT command mentioned in the discussion of print options for the DEC-10 monitor commands. When creating files, you will be prompted to give a unique name for the report.

There are several options available regarding registration information for the REG branch of the main menu.

LOAD will cause the system file (that had already been prepared) containing data regarding student course requests from CARS to be loaded into a database file. This will automatically remove any previous quarter's data. The user should be careful when using this option.

ORDR will evaluate each student request for each course and order the students. The students within the enrollment capacity will be marked for admission; the others will be marked for deletion.

DEL will make a file of the students and section numbers marked for deletion. This is the file you will later produce on a tape for return to CARS.

SEC will take you to a menu with selections for manipulating or creating class sections.

MAKE will invoke an interactive routine that will prompt you for information about a new class section.

CHNG will invoke an interactive routine to allow you to change any information regarding a certain section. You will select the class section by entering the section number.

PRI will take you to a menu for selecting the report you want produced. The routines for printing registration

information can be invoked for a single section or all sections for the current quarter. The current or active quarter is the one for which the rolls are still available. The current quarter is a period that begins with the LOADing and ORDRing of the data. Normally this occurs just after preregistration and just before the end of the academic quarter period. The data from the next quarter period will overwrite the data space when the LOAD of the dataset is activated.

ONE will invoke the menu defining the possible reports available for a particular section.

ALL will invoke the menu defining the possible reports available for all current sections.

In either case. the following are the selections available for indicating the desired report.

COMP produces a complete report including section information. class roll and waiting list.

SECT produces a report of section information.

ROLL produces a report with the roll and section information.

WAIT produces a report with the waiting list and section information.

Activity Sequencing Considerations

Some activities must be performed in a certain sequence

to produce the desired results (see Appendix C). The two situations where this is of greatest consequence are in updating the student records and in ordering student requests for registration.

When updating the whole file of student records you should follow this sequence of steps:

1. Use the backup facilities to preserve the current status of the database as well as other files. This will make a snapshot of the CSDB system. This is time consuming. You may leave the system and return later since this will execute in batch mode.
2. Use the .HS command to read the tapes from SDA and OIS using the IBM 3081 computer. This will create new system files in the CSDB directory area. This may be slow. You may leave the system and return later, or you may perform other CSDB tasks.
3. Run Accent-R and select the STUD branch of the MAIN menu. Select the UPDT branch. At this point you should select LOAD. This will put the new information from the tapes into the database in preparation for the massive update. You will be returned to the UPDATE menu when this process completes. This is slow. Unfortunately, you may not leave the system.
4. Next select the UFIL option on the UPDATE menu. You will see a sequence of messages to indicate the

progress of this module. This is very slow. You may not leave the system.

When this module completes, the student records are now ready to be accessed. They have the most recent grades as well as the student data from the most recent registration process.

5. Now you may delete the OIS.LP0 and SDA.LP0 system files using the .DEL command to the DEC-10 computer.
6. At this point, and after every significant change to the database, you should back up the system using the DEC-10 command .R BACKUP.

Fresh grades for a quarter are available on tape at about the time the next quarter begins. The student data is gathered at registration just prior to the beginning of the quarter. This is available shortly thereafter. You should have fresh data from both OIS and SDA before running the UFIL module.

When processing the registration requests you should follow this sequence of steps.

1. Use the HS command to read the tape from CARS. This will create a system file in the CSDB directory area. This can be slow. You may leave the system and return later.
2. Next you should invoke Accent-R and the CSDB system. Be certain that the capacity for each

section is accurate. Do this by selecting the PRI option. Select ALL to get all the information on sections for the desired quarter. The SECT option will produce a report of section information for the quarter you indicate. Verify the capacity for each section and then use the CHNG option if necessary.

3. Select the REG branch on the MAIN menu. At this point you should select LOAD. This will put the data from the tapes into the database. This may be slow. Do not leave the system.
4. Next you should select ORDR. This will compare the registration points for all students and then order the requests based on points. Each student will be marked as in or out of a given section.
5. Select the DEL module next. This will prepare a system file that indicates which students should be deleted from which sections.
6. PRI may be selected to get a printout.
7. You have now completed using the CSDB system. You may quit Accent-R, and then you may use the .HS command to write the system file to tape. Notify CARS that the tape is prepared.
8. You may now delete the system file CARS.LP0.
9. You should backup the system after this and every significant modification.

CHAPTER 4

DATABASE ADMINISTRATOR'S MANUAL

1. NAMING CONVENTIONS

The CSDB system is comprised of data and programs. All reside in the CSDB directory area on the DEC-10 computer. The database administrator is expected to be very familiar with the DEC-10 computer running the TOPS-10 operating system. In addition. the DBA is expected to have access to the Accent-R manuals.

Data from tapes is first fetched and written to system files in the directory. These will be named according to the data source. CARS.LP0 is raw data from CARS. SDA.LP0 is raw data from SDA. OIS.LP0 is raw data from OIS.

This data is prepared to fit the schema definitions of the corresponding data sets of the database, so DS CARS.DAT. DS SDA.DAT. and DS OIS.DAT contain this same information after it has been loaded into the database.

Other data set names also indicate their contents. DS CSDB.DAT contains the student data records. DS SECT.DAT contains the records of information for a section.

The menu system and additional narrative help information are also kept in data sets. DS MENU contains the menu system. DS HELP contains the help pages for the

menu system. Schema definitions follow the same naming convention, as do the data indices (see Figure 4.1).

DS	SD	DI
CSDB.DAT	CSDB.PERM.REC	CSDB
SDA.DAT	SDA.STUD.TAPE	SDA
OIS.DAT	OIS-GRADES.TAPE	OIS
SECT-DAT	SECT.INF	SECT
CARS.DAT	CARS.ROLL.WAIT.TAPE	CARS
MENU.	MENU	MENU
HELP.DAT	HELP.SD	HELP.INF

FIGURE 4.1

Naming Conventions for Data

The routines in the CSDB menu-system access one global storage area for global variables. This is GS MENU. The schema definitions are found in Appendix A.

The program modules are driven from menu selections. Each menu item, when selected, can either invoke another menu or invoke a command. In the CSDB system each "leaf node" or command driver invokes a command module (CM) with the same prefix. For example, selecting the menu item TRAN invokes CM TRAN with the following command.

USE CM TRAN

Each primary CM directs the entire action for that menu selection (see Figure 4.2). The primary CM may call other service CMs or may invoke process modules (PM). A PM is a compiled program whereas a CM contains Accent-R commands that are read and performed one at a time.

Most PM's are report producing modules. The names of these have "OUT" as the suffix; for example. PM LABSOUT is a report formatter that produces mailing labels (see Figure 4.3). PM UPDATE and PM ALTER are used by the CM UFIL in the massive file update. PM ORDRCARS and PM SECT.IN.OUT are used in the CM ORDR to rearrange the order of students requesting classes. CM MENU is the menu formatter and PM MENU is the terminal controlling routine. These are provided by Accent-R.

The programs kept outside the menu-system are written in PL/I. They are:

1. READ.CRS (reads CARS tape)
2. READ.SDA (reads SDA tape)
3. READ.OIS (reads OIS tape)
4. WRITE.CRS (writes CARS tape)

2. DATA INPUT/OUTPUT

The CSDB system requests data transfers from several agencies on a regular quarterly basis.

```
 10
 20 ! CM TRAN
 30 ! THIS MODULE PRODUCES ONE STUDENT MINI-TRANSCRIPT.
 40
 50 USE DS CSDB.DAT     DI CSDB
 60 USE CM ID_OR NAME           !TO SET %DOM
 70 USE CM SET ID_OR NAME      !TO SET %ID OR %NAME
 80
 90 TYPE @CR. @CR
100 TYPE 'PLACE PAPER IN THE SPINWRITER IF YOU WANT A HARD
    COPY.'
110 TYPE 'FLIP THE SWITCH THEN PRESS ANY KEY.  REMEMBER TO
    FLIP BACK.'. @CR
120 ACCEPT @STRING(5)
130 IF %DOM = 'STUDENT.ID'
140   EXTRACT WHEN $%DOM = $%ID        &
                              USE DOMAIN $%DOM
150 OR IF %DOM = 'STUDENT.NAME'
160   EXTRACT WHEN $%DOM = $%NAME      &
                              USE DOMAIN $%DOM
170 ELSE
180   TYPE ' INVALID SELECTION CRITERIA'
190 CONTINUE
```

FIGURE 4.2

Sample Command Module

```
 10
 20 !PM LABSOUT
 30 !PRODUCES MAILING LABELS
 40
 50 CONTROL SECTION
 60 RELATE SF FROM COMMAND AS REPORT 1
 70 RELATE SD CSDB.PERM.REC AS MASTER
 80 RELATE GS MENU
 90
100 INITIAL SECTION
110 6 TO @LINES
120 EJECT
130
150 DETAIL SECTION
160 PRINT @TOF, @CR
170 PRINT ' ', STUDENT.NAME
180 PRINT ' ', ST.ADDRESS
190 PRINT ' ', CITY.STATE, 2S, ZIP
200
210 FINAL SECTION
220 IF %IO = '2'.'3'
230   TYPE "THE REPORT IS AVAILABLE IN FILE ", %FILENAME,
    '.REP'
240 CONTINUE
```

FIGURE 4.3

Sample Process Module

The first retrieval is grades for the most recent quarter. This data comes from OIS just after grades are posted, usually around the first week of the following quarter. Contact OIS at (615) 974-5143, P225 Andy Holt Tower, University of Tennessee.

The next data request to arrive is student record master (SRM) information. This data is collected and updated by SDA each quarter during registration. The data comes to the Computer Science Department just after the quarter begins. Contact SDA at (615) 974-2108, 301 Student Services Building, University of Tennessee.

The last data to arrive is course requests by students. This data is collected by CARS during preregistration week (week six). This data is available to the department just prior to the end of the quarter. The data the department writes that indicates which students are to be deleted from classes must get to CARS before schedules are finalized and printed. The record formats for the various tapes are given in Appendix D.

3. SYSTEM REOUIREMENTS

This system relies on SWITCH.INI and BEGIN.ACC files to do system initialization. They are found in Appendix E. The SWITCH.INI pre-formats the print command. The BEGIN.ACC initializes Accent-R and automatically invokes the menu system when the user issues the .R ACCENT command. These

routines must have two pages of core allotted on the DEC-10 (1024K) and a user area of 5000 blocks.

4. ERROR DETECTION AND CORRECTION

The most disconcerting area of problems is errors in the data coming to CSDB from SDA and OIS. The PL/I modules do limited error checking and verification for reasonableness of data. Certain phone number prefixes represent area codes, so if a number has that prefix, the sequence is not a valid seven digit phone number, and it is replaced with blanks. Certain indicator fields have a range of expected values. These. too, are verified in the PL/I programs. Grade information is validated only for reasonableness, that is, the grade assigned must be a valid grade. Any grade that is invalid is set to blanks. There is no provision within the CSDB system for the user to alter the class grade field. Errors or inaccuracies detected here should be reported to the student records keeper for changing on the University's master file.

There is an area of problems for the database that the user can create. When the user is interactively MAKEing or CHNGing the DS SECT.DAT, there can be a problem should the user enter a duplicate section number. This will be detected as an error. but there is no protection from it. If this situation should occur, the record will need to be

repaired using native Accent-R commands which are fully explained in the Accent-R User's Guide.

The CSDB attempts to detect invalid data as it is entered by the user when operating in an interactive mode. In these situations, the user will be reprompted for a more acceptable data item, or perhaps a more acceptable format.

There is no protection for the database in the situation where the user modifies the record of the wrong student. It is hoped that when identifying a student record by student name- the record will be visually checked for additional fields of identification. Care should be taken to prevent mistaken identities.

5. PROGRAM MODULES

The remainder of Chapter Four contains the names of the program modules used by the CSDB and briefly describes the action of each.

Menu Modules

CHNG This module is used to modify DS SECT.DAT. The section is identified by section number.

CNT This module will produce a count of students belonging to a certain category. It uses DS CSDB.DAT. The user sets the category using CM WHATCAT and CM CNTCAT.

COMPALL This module will produce a report of all sections with class rolls and waiting lists for the coming

quarter. DS SECT.DAT and DS CARS.DAT are used. The data from CARS determines which sections are to be printed since it indicates the active quarter. PM COMPOUT is called to format the report.

COMPONE This module produces a display of section information. the class roll and the waiting list for one class section in the active quarter. DS SECT.DAT and DS CARS.DAT are used. The user is prompted to select the desired section. The display is routed to the terminal.

DEL This module formats a report of student numbers and corresponding section numbers, one pair to a line. This data will be rerouted to tape for access by CARS. The ORDR routines must have been used to establish which students are enrolled and which ones are to be deleted. The PM WRITE.CARS.DELETE is invoked. Output is directed to system file CARS.DEL.

GRDS This module produces a report containing abbreviated transcripts for students in a certain category. The dataset is CSDB.DAT. The user selects the desired category with CM WHATCAT and the system uses this to set the DI. The report is formatted by PM GRDSOUT. Output destination is set with CM WHATFILE.

LAB produces one mailing label for the student indicated. The user selects the method of identifying the student. then identifies the student. The output is routed to the terminal. DS CSDB.DAT is used.

LABS produces mailing labels, in zip code order, for students in a certain category. The CSDB.DAT data set is used. The user selects the category with CM WHATCAT. Output to a system file is set with CM WHATFILE.

LOAD is used to fetch data from system files that are created when the data tapes are read. This module will place this data into SDA.DAT and OIS.DAT. The data indices are then formed and any errors are written to error files for later correction and inclusion. The system files are then deleted.

LOADCARS is used to fetch the data from the CARS system file and place it into CARS.DAT. This would be done as soon as the CARS tape was read and the file prepared. The data index is formed and any errors are written to an error file for later correction and inclusion. The system file is then deleted.

MAKE The user adds data to the SECT.DAT dataset by use of the interactive enter facility. Prompts are provided.

NAME This module produces an ordered list of student names in a certain category. The category is selected using CM WHATCAT or CM OTHERCAT. OTHERCAT produces an ordered list of names, not necessarily an alphabetical listing, by setting the domain of the data index. The order will depend on the category selected. The DS CSDB.DAT is used. The output file can be set with CM

WHATFILE. The report is formatted by the PM NAMEOUT.

ORDR This module rearranges the student names based on registration points. The student is then identified with an IN.OUT mark that indicates whether or not the student is to be enrolled in the class. The PM ORDR.CARS process the data in DS CARS.DAT and DS CSDB.DAT.

ROLLALL This module produces a report of students enrolled in all sections for the active quarter.

ROLLONE This module produces section information and the list of enrolled students for one section. The section information is from DS SECT.DAT. The roll is from CARS.DAT. If the user selects a quarter that does not have an active roll only the section information is displayed. The display is to the screen.

RPT produces a full display of information regarding one student. The user sets the student identification method and then selects the student. The data set is CSDB.DAT. The report is routed for display on the terminal.

RPTS This module produces a full display of information for each student in a group. The category is set for the user in CM WHATCAT. The output file for the report is set for the user in CM WHATFILE. The PM RPTSOUT formats the report.

SECTALL This module will produce a printout of information about a group of class sections offered by the department. The user selects the group by quarter and year in the CM WHATQTR. The dataset is SECT.DAT. The report is routed by using CM WHATFILE.

SECTONE This module will produce a display of information regarding one class section. The user selects the section by section numbers. The report is routed to the terminal.

TRAN This module produces an abbreviated transcript for one student. The user selects the student by identification number or name. The grade report is routed to the terminal.

UFIL This module invokes the programs that do the quarterly update to the database using data returned on tape from OIS and SDA. The entire student file is modified. New department indicators are generated and a fresh value for student registration points is determined.

UREC This sets up an interactive state within the database. The user is prompted to indicate which student record is desired, either by name or by identification number. The user is then prompted to select the data fields that are to be modified. Finally the user is prompted to modify those fields.

WAITALL This module produces a printout. by name. of all

students on the waiting lists in all class sections for the active quarter. The waiting list is formed in descending order of registration points.

WAITONE This module produces a report of the students on the waiting list for one specific class section for the active quarter.

Service Modules

CNT This module counts the number of students in a certain category by counting the records in the database.

CNTCAT This module prompts the user to select the search category for counting the number of students in a group. Fourteen categories are presented.

ID_OR_NAME This is the module that prompts the user for identifying a student record by student name or identification number. It sets a flag to indicate the user's choice.

IS012 In this module the user sets a flag to indicate whether the desired group consists of those not in the category- those capable of being in the category, or those in the selected category.

MENU This is the module that drives the menu system. It is provided by Accent-R.

OTHERCAT This module provides three additional categories for selection of a student group and ranking the students by a certain field value such as GPA.

SET_ID_OR_NAME In this module the user is prompted to indicate the actual student name or student identification number for the desired student record.

WHATCAT In this module the user selects categories of students for CSDB activities other than a count of student records in a group. Eleven categories are presented.

WHATFILE This module prompts the user to name the output file for a report that is to be produced.

WHATQTR In this module the user is prompted to indicate the quarter and year for certain section information that is desired.

PL/I Modules

READ.CRS This module will be produced to read a tape from CARS.

READ.OIS This program accesses a tape provided by OIS that contains computer science grade information. The data is reformatted to fit CSDB schema definitions.

READ.SDA This program accesses a tape provided by SDA that contains student master records. The data is read for the desired student records and reformatted to fit the CSDB schema formats.

WRITE.CRS This module will produce a tape for CARS that has student numbers and the section numbers from which they are to be deleted from the roll.

CHAPTER 5

PROJECT ENHANCEMENTS

The CSDB project has produced many refreshing ideas and suggestions for modification and improvement. These are arranged for discussion in groups of similar items.

1. NEW FIELDS

The student database was modified just prior to the project completion to include the data fields that indicate previously earned degrees. This was accomplished by modification of schema definitions. The additional data has not yet been fully integrated into printout reports or categorical student reports. This item, like some of the ideas mentioned later in this chapter, was seen to be useful, so the item was included during the last stages of program modification. New fields can be added readily, and programs can be developed or modified to take advantage of the newly available data fields.

The database table that maintains information on class sections offered by the Computer Science Department presently includes only registration details such as room number, capacity and section number for all courses offered every quarter. This could be expanded to include textbook titles, syllabi, grade averages and teacher evaluations. This information could be a resource for instructors.

2. STATISTICAL ASPECTS

This project does no statistical analysis, although such information could be useful to the Computer Science Department. For example, programs could be developed to print quarterly reports of means and medians for grades earned in computer science classes. Quarterly summaries of the number of students graduating, number of new majors, average grades earned in classes, etc, could be generated. An interesting idea would be a study that examines the correlation of various items to student success in computer science. This could provide insight into the factors which should be considered in the selection of future students.

3. SCHEDULING PROJECTION

Another area of study could be the development of programs to evaluate the flow of students through the courses and project enrollments in classes. This could help the department plan the timing and class size of course offerings.

4. PREREQUISITE CHECKING

The department has established prerequisite requirements for many of the courses offered. Students who do not adhere to this policy often impede the progress of a class because they are not fully prepared. Currently, the

only method of checking enrolled students for successful completion of all prerequisites is manual examination of old grade reports. This is an extremely tedious process; therefore it is often left undone.

The CSDB maintains each student's course history in computer science. Program modules could be developed to search the record of each student enrolled in a specific class for compliance with that class's prerequisite requirements. This module could be run in advance of the beqinning of the quarter so that non-complying students could be detected before the first class meeting and their space given to a more appropriate student. This would mean that late additions to class rolls would have to be done separately. either by hand or by using a simplified version of the database routines.

5. ADAPTATION FOR USE BY OTHER DEPARTMENTS

Another useful facility would be an interactive program for setting the variables and constants used to determine the registration points value for each student. This procedure could provide greater flexibility in the quarterly registration process. Perhaps even variable measures could be developed for different classes. The development of this module would greatly ease adaptation of this system to the needs of other departments. The report formatting programs would need to be modified to meet the departmental needs.

6. PRODUCTION ENHANCEMENTS

Additional future considerations include system modifications to improve the project's ease of use. Currently, student records and grade reports are requested for undergraduate majors only. Records of undergraduate pre-majors and graduate students are not provided. This is due in part to the department's inability to indicate specifically which student records are desired. Many university students enroll in lower level computer science courses, some with no intention of majoring in computer science. There is currently no satisfactory method of identifying the pre-majors as a group or on an individual basis. If students could be identified on an individual basis, the student identification numbers could be used for requesting reports from SDA and OIS on a quarterly basis. The request could be sent on tape (in machine readable format) and the errors in typing the numbers for each student could be reduced. Of course, OIS and SDA, the agencies providing the student data, would have to agree to the new procedure. Gathering student identification numbers of the desired students is not a trivial undertaking, yet it is not very realistic to consider retrieving records only for majors already well into the program, as is now the case.

One idea to handle the data entry problems is to

develop a logon procedure that students would use to type in their own names and identification numbers. Another technique could be scan forms that students could fill out. The problem is truly significant only in the initial phases as the number of new students is small when compared with the munber on the initial list. Recurring students could be processed quarterly; students that are inactive would eventually be dropped. The current list of active students in CSDB.DAT. merged with any new students' identification numbers. could generate the next quarter's list of requests to OIS for records. This change in request format would need to be approved by OIS and SDA. and all the details of the tape formats defined.

7. SECURITY

Security is another area for possible modification or upgrading. The sensitive nature of a student's personal and academic record, of course. leads to security concerns. This department has especially talented and precocious students. so security is more than a casual concern for the production aspect of the CSDB system. These students may view breaking into a computer science database as a special challenge. As an added protection. the department could request CARS, SDA and OIS to require a password to access the tapes and the tape reading programs could be designed to destroy the data on the tapes as soon as it is acquired.

The time between the tape being written by the agency and its being read by CSDB could be narrowed to reduce the possibility of exposure. Once the data goes into the CSDB system. each file could be encrypted, or perhaps just the identificiation fields in the files. Use of passwords within the Accent-R menu system is too easy to circumvent. (This is an Accent-R shortcoming.)

Changing grades or any other information within the CSDB system will not affect a student's transcript except within the Computer Science Department. However, it is possible that this could be used to increase a student's preregistration points. Even then. those altered fields would be overwritten the next quarter. No student data is ever transfered back to the university's databases. so the real risk is indeed exposure. There is no permanent risk to the integrity of the data.

In conclusion. many changes could be made to improve the CSDB system. The production model will certainly develop over time. The overall system design has sought to provide for as much flexibility as possible. so that it can be tuned to satisfy the user's requirements that will surface only when the system is in production.

CHAPTER 6

CONCLUSION

1. SUMMARY

The CSDB system of programs has great promise as a tool for organizing information for the Computer Science Department. It can facilitate departmental management decisions and simplify the processing of student records. The department has stringent requirements for admission to the major. which requires substantial record checking for the staff and faculty. In addition. academic records of students must be carefully monitored on a quarterly basis for academic warning, probation and termination. Many times the department needs a list of student names for a certain category or a count of all students in a certain group. Requests of this kind can now be handled in a quick and efficient manner. Interaction with the university database can be kept to once per quarter. This represents a substantial reduction in requests from this department.

The CSDB system will also be a useful tool for departmental management of student registration. Course demand has been high for several years. and many students' needs have not been met promptly. By ranking the students for registration and controlling the process. the Computer Science Department can assure the more promising students a

place in the courses desired and reduce the probability of admitting a marginal student. The overall effect should be a more sophisticated student body and a higher caliber graduate. This is in accordance with the University's overall upgrading of standards.

The CSDB system is modifiable in several ways. The relational data organization allows for relatively simple additions and deletions of data fields. The menu system is itself organized as entries in a relational table so that new options and branches can be added rather easily, and the organization of the entire menu system can be altered, if necessary. New program options can be developed to manage additional data and can then be quickly attached to the menu system. Accent-R provides a good foundation for system development for a dynamic system such as this one.

One of the concerns is the discontinuation of the DEC-10 computer system. Although the Accent-R manager runs on a variety of systems, unfortunately the DEC-10 is the only one available at this installation that supports Accent-R. Digital Equipment Corporation has dropped support for the DEC-10. and the University expects to discontinue it eventually. At that time the CSDB system will either be discontinued or modified to conform to another database manager unless the University has acquired a computer that supports Accent-R.

The DEC-10 is often an overused resource with slow

response time. This will not significantly affect the use of CSDB since much of the heavier processing is done early in the academic quarter when system utilization is quite low. The query and report formatting modules will be used throughout the quarter but do not require significant computer resources to operate, so their use should not be seriously degraded.

2. ETHICAL CONSIDERATIONS

One of the usual concerns in the management of information systems is the misuse of information. The CSDB system has as much potential for abuse as any system of student information, whether computerized or manually maintained. The system will be protected from outside use with the usual login protection of a private DEC-10 account. The primary concern here, however, is not outside invasion, but misuse from within the Computer Science Department. For example, registration points are a valid means to compare one student to another only with regard to admission to courses. The registration points should not be used for any purpose other than determining admission to classes. Once students are admitted to a class, the relative position of students becomes meaningless. It is hoped that all users will regard registration points as a relative measure of appropriateness of admission to classes and not as any measure of academic standing or performance in the

classroom. Ranking students by other criteria will be possible and, of course, categorized groupings can easily be made. It will be the responsibility of the individual user to use the available information in a conscientious and reasonable way.

3. BENEFITS

During the initial development of the CSDB, I had many sessions with the prospective users of such a system. These sessions were directed toward defining the desired appearance of the product and the goals of the project regarding the information needs of the users. I also had several meetings with the service agencies where I attempted to gain the cooperation of these groups in supplying the necessary information.

This experience proved to be very illuminating. I learned quite a bit about project design and development. I also gained valuable experience in the area of cooperative efforts. My exposure to the university database was also valuable. These experiences contributed to the overall value of the project as a "real world" experience.

My understanding of relational database technology and "user-friendly" systems has been greatly enhanced by my work on this project.

LIST OF REFERENCES

LIST OF REFERENCES

Barnes, Robert A., PL/1 for Programmers. Elsevier Science Publishing Co., Inc., New York, NY, 1984.

Cardenas, Alfonso F., Database Management Systems. Allyn and Bacon, Inc., Boston, MA, 1979.

Date, C. J., An Introduction to Database Systems, Third Edition. Addison-Wesley Publishing Co., Reading, MA, 1982.

Kroenke, David, Database Processing, Second Edition. Science Research Associates, Inc., Chicago, IL, 1983.

"Accent-R User's Manual." National Information Systems, Inc., Cupertino, CA. 1983.

"Accent-R Getting Started." National Information Systems, Inc., Cupertino, CA. 1983.

Ullman, J.D., Principles of Database Systems. Computer Science Press, Inc., Rockville, MD, 1980.

"UTCC User's Guide." The University of Tennessee Computing Center, The University of Tennessee, Knoxville, TN, 1984.

"UTCC DEC-10 User's Guide." The University of Tennessee Computing Center, The University of Tennessee, Knoxville, TN, 1982.

"The University of Tennessee Undergraduate Catalog/1984/Knoxville." The University of Tennessee, Knoxville, TN, 1984. (E01-0425-002-84).

APPENDIXES

APPENDIX A

SCHEMA DEFINITIONS

SCHEMA DEFINITION SECT.INFO
FOR DATASET SECT.DAT

QTR-OFFERED	C	2
YEAR.OFFERED	I	2
DEPT.CODE	C	3
COURSE.NUM	I	4
SECTION.NUM	I	5
CREDIT-HRS	I	2
CAPACITY	I	3
INSTRUCTOR	C	3

SCHEMA DEFINITION CARS.ROLL.WAIT.TAPE
FOR DATASET CARS.DAT

STUDENT.ID	C	9
STUDENT.NAME	C	25
DEPT.CODE	C	3
COURSE.NUM	I	4
SECTION.NUM	I	5
SPECIAL.CREDIT	C	3
CARS.PRIORITY	I	3
CARS.POSITION	I	3
REG.PTS	I	4
CS.POSIT	I	3
IN.OUT	C	1

SCHEMA DEFINITION SDA.STUD.TAPE
FOR DATASET STUD.DAT

STUDENT.ID	C	9	
STUDENT.NAME	C	25	
ST.ADDRESS	C	25	
CITY.STATE	C	20	
ZIP	C	5	
PHONE	C	7	
SEX.CODE	C	1	
HS.GPA	R	4	2
ENGLISH.ACT	I	2	
MATH.ACT	I	2	
COMPOSITE.ACT	I	2	
DEGREED	C	1	
CLASSIFICATION	C	1	
LOCATION	C	1	
DIVISION	C	1	
COLLEGE	C	1	
MAJOR	C	3	
CUM.HRS.PASS	I	4	
CUM.GPA	R	4	2
THIS.QTR.HRS	I	3	

SCHEMA DEFINITION OIS.GRADE.TAPE
FOR DATASET OIS.DAT

STUDENT.ID	C	9	
DEPT.CODE	C	3	
COURSE.NUM	I	4	
SECTION.NUM	I	5	
GRADE	C	2	
QUALITY.PTS	R	5	2
CREDIT.HRS	I	2	

SCHEMA DEFINITION CSDB.PERM.REC
FOR DATASET CSDB.DAT

```
UPDATE.DAT          D 6
STUDENT.ID          C 9
STUDENT.NAME        C 25
ST.ADDRESS          C 25
CITY.STATE          C 20
ZIP                 C 5
PHONE               C 7
SEX.CODE            C 1
HS.GPA              R 4 2
ENGLISH.ACT         I 2
MATH.ACT            I 2
COMPOSITE.ACT       I 2
DEGREED             C 1
CLASSIFICATION      C 1
LOCATION            C 1
DIVISION            C 1
COLLEGE             C 1
MAJOR               C 3
CUM.HRS.PASS        I 4
CUM.GPA             R 4 2
THIS-QTR.HRS        I 3
ADVISOR             C 3
REG.PTS             I 4
CS-TOT.HRS          I 3
PREV.CS-GPA         R 4 2
CURR.CS-GPA         R 4 2
CO OP               C 1
GTA                 C 1
CS-GRAD             C 1
CS-MAJOR            C 1
CS-MINOR            C 1
CS-HONOR            C 1
CS-PROBATION        C 1
CS-TERMINATE        C 1
CS.RE ADMIT         C 1
REMOVE.MARK         I 1
NEW.GRADES          C 1
GROUP COURSE-HIST OCCURS 25
DEPT.CODE           C 3
COURSE.NUM          I 4
SECTION.NUM         I 5
GRADE               C 2
QUALITY.PTS         R 5 2
CREDIT.HRS          I 2
QTR-TAKEN           C 2
YEAR-TAKEN          I 2
GROUP END
```

SCHEMA DEFINITION MENU
FOR DATASET MENU

```
MENU.NAME                C 10
ITEM.ID                  C 4
ITEM.DESCRIPTION         C 20
THE.PASSWORD             C 6
THE.PASSWORD.PROMPT      C 15
HELP.ID                  C 4
ITEM.EXECUTION.TEXT      C 80 OCCURS 5
NEXT.MENU                C 10
PREVIOUS-MENU            C 10
```

SCHEMA DEFINITION HELP.SD
FOR DATASET HELP.DAT

```
HELP.ID          C 4
HELP.SEQ.NO      I 4
HELP.TEXT        C 70
```

APPENDIX B

SAMPLE REPORT

Sep 13, 1984 COMPLETE REPORT OF STUDENT RECORDS IN CS FOR SOPHOMORES

LAST UPDATE: 09/11/84
200-00-0000
SUSAN BACKWARD
12 SECOND ST
KNOXVILLE TN

PHONE # 234-7654

37919

MARRIED FEMALE
LIBERAL ARTS
SOPHOMORE
DAY
KNOXVILLE

HS GPA :3.60
ENG ACT :23
MATH ACT:24

CUM HOURS : 50
HRS THIS QTR: 15
CUM GPA :3.00

CS HOURS : 12
PREV CS GPA:3.30
CURR CS GPA:3.00

ADVISOR:BBB
REG PTS: 60
REMOVE : 1

GTA :0
CO-OP:0
HONOR:0

CS MAJOR :0
CS MINOR :0
UT MAJOR :

PROBATION:0
TERMINATE:0
RE-ADMIT :0

DEPT.	COURSE	SECTION	GRADE	QTR	YR
233	1610	12345	B+	FA	83
233	1620	12346	B	WI	84
233	2710	12347	B+	SP	84
233	2215	12348	C	SU	84

APPENDIX C

SEQUENCE OF EVENTS

Fall (A)		Winter (B)		Spring (C)
GRAD(A)	Registration(B)	PRE-Registration(C)	Graduation(B)	Registration(C)
Grades (A) Posted	OIS Avail (A) SDA Avail (B) Generate Depart Info (B) Generate REG pts for (C) Send out letters (B) Change Marks on majors/minors terminates, etc. (B)	Set SECT (C)	CARS Avail (C) Set IN.OUT (C) Produce Rolls and waits (C) Send DEL tape (C)	
FUTURE: Old grades used to check prereqs	Grades from (A) checked for prereqs			

APPENDIX D

TAPE FORMATS

COMPUTER SCIENCES COURSE HISTORY FILE

RECORD DESCRIPTIONS:

		COLUMN	LENGTH
TYPE 1:	STUDENT NAME	1	30
	STUDENT ID	31	9
	FILLER(SPACE-FILLED)	40	37
	SORT FIELD	77	3
TYPE 2:	STUDENT NAME	1	30
	DEPARTMENT CODE	31	3
	COURSE NUMBER	34	5
	SECTION NUMBER	39	5
	GRADE	44	2
	MESSAGE*	46	31
	SORT-FIELD	77	3
TYPE 3:	STUDENT NAME	1	30
	CS GPA	31	4
	FILLER (SPACE-FILLED)	35	11
	MESSAGE**	46	31
	SORT-FIELD	77	3

*Field contains blanks or '** COURSE GRADE NOT A THRU F'

Field contains blanks or ' NO COMPUTER SCIENCE COURSE HISTORY'

COMPUTER SCIENCES STUDENT RECORD MASTER FILE

RECORD DESCRIPTIONS:

		COLUMN	LENGTH
TYPE 1:	STUDENT ID	1	9
	STUDENT NAME	10	25
	FILLER (SPACE-FILLED)	36	56
	LOCAL STREET ADDRESS	93	25
	LOCAL CITY AND STATE	118	20
	LOCAL ZIP CODE	138	5
	LOCAL PHONE	143	7
	FILLER (SPACE-FILLED)	150	77
	SEX	227	1
	FILLER (SPACE-FILLED)	228	1
	BIRTH DATE	229	6
	FILLER (SPACE-FILLED)	235	55
	HIGH SCHOOL GPA	290	3
	FILLER (SPACE-FILLED)	293	8
	ACT ENGLISH SCORE	301	2
	ACT MATH SCORE	303	2
	FILLER (SPACE-FILLED)	305	4
	ACT COMPOSITE SCORE	309	2
	FILLER (SPACE-FILLED)	311	16
	CLASSIFICATION	327	2
	LOCATION	329	1
	DIVISION	330	1
	COLLEGE	331	1
	MAJOR	332	3
	FILLER (SPACE-FILLED)	335	34
	CUMULATIVE HOURS CARRIED	369	4
	CUMULATIVE HOURS PASSED	373	4
	CUMULATIVE QUALITY POINTS	377	5
	CUMULATIVE GPA	382	3
	LAST QUARTER GPA	385	3
	HOURS THIS QUARTER	388	3
	HOURS CARRIED LAST QUARTER	391	3
	HOURS PASSED LAST QUARTER	394	3
	QUALITY POINTS LAST QUARTER	397	4
	FILLER (SPACE-FILLED)	401	67
	LAST QUARTER	468	1
	FILLER (SPACE-FILLED)	469	82

APPENDIX E

BEGIN.ACC AND SWITCH.INI

BEGIN.ACC

```
TYPE @CR. NOSKIP
TYPE @LTRIM @FTIME. " ", @LTRIM @WEEKDAY, " ", @FDATE
USE DBL CSDB
SET1000 TO @TERM.LINE
USE CM MENU
```

SWITCH.INI

```
;       SWITCH.INI
;
;       Comments must begin with a ';' .
;       No blanks lines are allowed.
;       To continue a command line put a '-' at the end
;       of the line to be continued and then begin next
;       line with a '/'.
;       Switches for a particular monitor command begin
;       with that monitor command name.
;
;-------------------------------------------------------------
;       LOGIN
;
LOGIN   /UFDPROT:700/SFDPROT:700/DEFPROT:477/DEST:RMT2-
        /ASSIGN:DSKA:A/ASSIGN:DSKB:B/ASSIGN:DSKC:C-
        /ASSIGN:DSKD:D/ASSIGN:DSKE:E-
        /SCAN/SYS/NEW/NON/NOTICE:SOMETIMES-
        /NOTE:ALL:ACCESS.LOG[,]-
        /TERMINAL:(LC,NOGAG,PAGE,PAGESIZE:60,TYPE:TELEVI)-
        /ACCOUNT:"C S D B"
;set user file directory protection at 700 (activate
;       access.usr)
;set file protection at 477 (activate access.usr)
;set default destination to Ayres Hall
;allow C to be ersatz name for DSKC, etc;
;(.PRI C:SWITCH:INI[,] is legal)
;enable scanning to higher directories when file not found
;enable scanning of SYS:
;enable scanning of NEW:
;enable scanning of NON:
;print new system notices only
;notify on login if access.log has been created on DSKC and
;       type it (treated like notice, only new access.log
;       is typed)
;set terminal to lower case
;enable printing of right curly bracket
```

```
;set terminal mode to permit 'send'ing of messages
;set pagesize so pauses every 66 lines for inserting forms
;get rid of ^S ^O marks when typing out a file
;set terminal type to TELEVIDEO for SED
;-------------------------------------------------------------
;       DIRECT
;
DIRECT/W:84/ALLOCATE
;
;set width to 84 so can have 2 columns (brief)
;show physically allocated (rather than written) file size
;
;-------------------------------------------------------------
;       HSUBMIT
;
HSUBMIT/NOTIFY
;
;buzz when job crosses interface to IBM
;
;-------------------------------------------------------------
;       PRINT (for files printed on greenbar)
;
PRINT:QUICK/HEADER:0/NOTIFY/DEST:RMT2
;
;use with .PRINT FILE.EXT/OPTION:QUICK
;no burst page
;buzz when across interface
;send to Ayres
;
;-------------------------------------------------------------
;       PRINT (for text)
;
PRINT:TEXT/FORMS:0416/TN/DEST:RMT2/NOHEADER/NOTIFY
;
;use with .PRINT FILE.EXT/OPTION:TEXT
;put on 8 by 11 white paper
;use upper and lower cae
;route to Ayres
;no burst page
;buzz when crosses interface
;
;-------------------------------------------------------------
;
PRINT:LABELS/FORMS:3008/TN/DEST:RMT2/NOHEADER/NOTIFY-
/NODE:77/LINES:1000
;
;use with .PRINT FILE.EXT/OPTION:LABELS
;print on mailing labels
;upper/lower cae
;route back to Ayres for delivery
;no burst page
```

```
;buzz when crosses interface
;get plenty of pages for output - each label is a page
;
;-------------------------------------------------------------
;       TYPE
;
;
TYPE/WILD/NAME/PAUSE:1:60/NOTABS/BACKSPACE/BELL/ENAM-
/INDENT:8
```

VITA

Ann Wayburn was born in Columbia, South Carolina. She was educated in public schools and attended The University of South Carolina. Ms. Wayburn relocated to Knoxville, Tennessee in the early 70's and completed her undergraduate studies at The University of Tennessee, Knoxville. She graduated in 1978 with a Bachelor of Arts degree majoring in Interpersonal Relationships, a program of study she developed through The Office of Individualized Programs. After working in the mental health field, she returned to The University of Tennessee to pursue a Master of Science degree with a major in Computer Science. While in graduate school she was a graduate teaching assistant with The Department of Computer Science. She later worked as system analyst for The Office for Research in High Technology Education developing a national on-line data base for The United States Department of Education.

Ms. Wayburn will graduate in December 1984. She is now an Instructor with The Computer Science Department at The University of Tennessee and continues to consult with The United States Department of Education.

Ms. Wayburn is a member of The Association for Computing Machinery.