10-1-1984

A design of a data base system for an academic department

Joseph Yuanping Wan
Atlanta University

Follow this and additional works at: http://digitalcommons.auctr.edu/dissertations

Part of the Physical Sciences and Mathematics Commons

Recommended Citation
A DESIGN OF A DATA BASE SYSTEM
FOR AN ACADEMIC DEPARTMENT

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

BY
JOSEPH YUANPING WAN

DEPARTMENT OF MATHEMATICAL AND COMPUTER SCIENCES
ATLANTA UNIVERSITY
ATLANTA, GEORGIA
OCTOBER, 1984
ACKNOWLEDGEMENT

I would like to express my appreciation to all those who have supported me when I needed help and encouragement in my studies and research at Atlanta University.

My deepest gratitude goes to:

Dr. Benjamin J. Martin
Dr. C.B. Setzer
Dr. Nazir A. Warsi
Ms. Sharon M. Wallace

and all my classmates in the Department of Mathematical and Computer Sciences.
This thesis describes the design of an on-line database management system for an academic department. The implementation uses dBASE II to create a full information system containing student records, course listing, course scheduling, fiscal management, statistical data, faculty records, and inventory of textbooks and equipment. Application programs to maintain the data base are also included.
# TABLE OF CONTENTS

**ACKNOWLEDGEMENT** ........................................... 1

**ABSTRACT** ..................................................... 11

## CHAPTER

**INTRODUCTION** ............................................... 1

1. **DEFINITION AND STRUCTURE** ................................. 3
   1.1 **DEFINITION AND STRUCTURE OF DATABASE FILES**
   1.2 **DEFINITION OF TECHNICAL TERMS**

2. **DBASE II OVERVIEW AND FEATURES** ......................... 19

3. **DESCRIPTION OF PROGRAM DESIGN** .......................... 27

**CONCLUSION** ................................................... 40

**FOOTNOTES** ................................................... 42

**BIBLIOGRAPHY** ................................................ 43

**APPENDICES** ................................................... 44

A. **STRUCTURE OF DATA FILES**

B. **PROGRAMS**

C. **OUTPUT EXAMPLES**
INTRODUCTION

Everyone living in the world is affected by computers. All areas of our society have been and are being touched by computers. From the time we get up and read the morning news (which is typeset by computer), until we retire in the evening watching TV (computer allocated programs), we are constantly affected by computers either directly or indirectly. Computer are no longer the exclusive tool of mathematicians and scientists. Now they are used in business, education, government, and families.

The purpose of this effort is to design a data base as a management system, which includes student record file, fiscal management file, statistical data file, course listing file, equipment control file and textbooks management file, to simplify all the information processing functions. There are several advantages of using this database management system such as time saving, decreased expense, increased efficiency, and ease of maintenance and manipulation of the information in the data files.

There are three chapters in this thesis, the first chapter discusses the structure of each data file and the
relationship between files. Chapter two contains a general overview of dBASE II. Chapter three contains a description of programs created to carry out the functions of this database management system.

An appendix includes a listing from a test run and also programs.
CHAPTER 1
DEFINITION AND STRUCTURE

In designing a data base as a management system to computerize the manual functions for an academic department, first, we consider what would be the assumptions for this environment and what data files would be in the data base. Here, the Mathematical and Computer Sciences Department of Atlanta University is used as a practical case. We consider specific assumptions and functions after we described the data base kept by the system.

We assume the data files of student records, faculty records, course catalogs, course schedules, class listings, graduates, students on financial aid, object and sub-object budget codes of grants, textbook listings and equipment control constitute this data base. All the information for each data file will be stored in this data base and will be manipulated by this system. The structure of each data file and definition of each data item is considered in following sections.

1.1 Definition and Structure of Data Base Files
Student record file: This file stores information on all students. A data record contains 14 items as follows, the SSNO (social security number) is the primary key of a record.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSNO</td>
<td>social security number</td>
</tr>
<tr>
<td>NAME</td>
<td>name of student</td>
</tr>
<tr>
<td>ADDR</td>
<td>address of student</td>
</tr>
<tr>
<td>CITY</td>
<td>name of city</td>
</tr>
<tr>
<td>STATE</td>
<td>name of state</td>
</tr>
<tr>
<td>ZIP</td>
<td>zip code</td>
</tr>
<tr>
<td>BIR:DAT</td>
<td>date of birth</td>
</tr>
<tr>
<td>SEX</td>
<td>male/female</td>
</tr>
<tr>
<td>ENTRY</td>
<td>date student enrolled</td>
</tr>
<tr>
<td>MAJOR</td>
<td>student's major (computer science, mathematics and applied mathematics)</td>
</tr>
<tr>
<td>TEL</td>
<td>telephone number</td>
</tr>
<tr>
<td>NATION</td>
<td>nationality, foreign or domestic</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>expected date of graduation</td>
</tr>
</tbody>
</table>

This file is sorted by student's major and his social security number.

Course catalog file: This is a catalog of all courses offered by the Mathematical and Computer Sciences Department. There are four elements in a record, the data item of COR:NUM (course number) is unique.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COR:NUM</td>
<td>course number (i.e. MCS-575)</td>
</tr>
<tr>
<td>COR:NAME</td>
<td>course name</td>
</tr>
<tr>
<td>CREDIT</td>
<td>credit hours of this course</td>
</tr>
<tr>
<td>DESCRIP</td>
<td>brief description of this course</td>
</tr>
</tbody>
</table>

Course schedule file: This file contains all courses offered in a specific semester. Contents of each record are
as follows. At the end of the semester, we delete all records so a clean file can be ready for next semester. The CORNO is the primary key.

<table>
<thead>
<tr>
<th>SEMTR</th>
<th>semester course is offered</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>year course is offered</td>
</tr>
<tr>
<td>CNO</td>
<td>course number</td>
</tr>
<tr>
<td>TIME</td>
<td>from when to when course will be met</td>
</tr>
<tr>
<td>DAY</td>
<td>days course will be met</td>
</tr>
<tr>
<td>INSTR</td>
<td>instructor of course</td>
</tr>
</tbody>
</table>

**Class enrollment file:** This is a list of each offered course and information of students enrolled in it. Each record contains seven data items. Semester, year, course number, student name and social security number are entered at the beginning of the semester. The course grade will be entered at the end of the semester. Like the course schedule file, a listing of this file will be put in file when the semester is over and clean up all records from on-line for use next semester.

<table>
<thead>
<tr>
<th>SEMTR</th>
<th>semester course is offered</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>year course is offered</td>
</tr>
<tr>
<td>CORNO</td>
<td>course number</td>
</tr>
<tr>
<td>STUNT</td>
<td>student name</td>
</tr>
<tr>
<td>SSNO</td>
<td>student social security number</td>
</tr>
<tr>
<td>GRADE</td>
<td>course grade</td>
</tr>
<tr>
<td>POINT</td>
<td>point of grade (using a 4 point scale)</td>
</tr>
</tbody>
</table>

**Financial Aid file:** The information on students who have received financial aid are stored in this file. There are three types of aid: teaching assistantship, research
assistantship and tuition scholarship. Student social security number is unique, if we want to see students with a same type of aid, we can use the type of aid as a secondary key. The contents of record are as below.

<table>
<thead>
<tr>
<th>SSNO</th>
<th>student social security number</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>student name</td>
</tr>
<tr>
<td>TYPE</td>
<td>type of aid</td>
</tr>
<tr>
<td>GRANTIT</td>
<td>grant title of aid</td>
</tr>
<tr>
<td>BUDGNO</td>
<td>budget number of aid</td>
</tr>
<tr>
<td>AMT</td>
<td>amount of aid</td>
</tr>
<tr>
<td>TIME</td>
<td>when this student receives aid</td>
</tr>
</tbody>
</table>

Object budget file: This file stores current totals for department budgets. Each budget has a budget code which is the primary key. A budget record contains seven data items.

<table>
<thead>
<tr>
<th>BUDG:TIT</th>
<th>grant title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUDG:NO</td>
<td>grant number</td>
</tr>
<tr>
<td>CODE</td>
<td>budget code</td>
</tr>
<tr>
<td>SOURCE</td>
<td>budget source</td>
</tr>
<tr>
<td>DIRTOR</td>
<td>director of budget project</td>
</tr>
<tr>
<td>AULOG</td>
<td>Atlanta University log number</td>
</tr>
<tr>
<td>AMT</td>
<td>Amount of budget</td>
</tr>
</tbody>
</table>

Sub-object file: A object budget will have several sub-object codes for a budget. The current totals for all sub-objects are stored in a separate file (sub-object data file). There are three items in each record. The sub-object code is unique.

<table>
<thead>
<tr>
<th>SUB:CODE</th>
<th>code number of sub-object</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB:TIT</td>
<td>title of sub-object</td>
</tr>
</tbody>
</table>
Faculty record file: All the information on faculty and staff in the department are in this file. The employees social security number is the primary key.

F:SSNO  : social security number
F:NAME  : name
F:TIT    : title or position
F:ADDRS  : address of faculty/staff member
F:CITY   : city name
F:ST     : state name
F:ZIP    : zip code
F:TEL    : telephone number

Textbook file: This file contains all information concerning textbooks which are used by department. Each book has a unique bookcode. The contents of record are:

BKCODE  : book code
BKNAME  : book name
COURSE  : course using this book
PUBSHER : publisher of book
BKTIME  : semester and year this book used

Equipment data file: This is an inventory of equipment and machines in the department. Each record contains following data items. Equipment code is the primary key.

CODE    : equipment code
CMY     : manufacturer
LOC     : location of equipment
SERV    : service representative
TEL     : telephone number of representative
We now consider the assumptions used in setting up a database management system of the above data base files for this environment.

1. Student information may be changed and updated.
2. Student record may be deleted.
3. Student academic records and GPA (average of grade point) results can be calculated and printed out.
4. Produce a course schedule listing for a specific semester.
5. Course schedule may be changed and updated.
6. Produce a class listing for each offered course and students who are taking the course.
7. Students may add or drop courses, class listing files need to be changed and updated.
8. There are three majors in Mathematical and Computer Sciences Department (Computer Science, Mathematics and Applied Mathematics).
9. There are three types of financial aid (Teaching Assistantship, Research Assistantship and Tuition Scholarship).
10. Produce statistics table including number of male and female, foreign and domestic, graduates, students on financial aid, and students of each major.
11. How to access an object budget and its sub-object.
12. Balance of budget amount may be changed and updated.
13. Faculty records may be deleted, added or modified.
14. A textbook is used for a specific course and semester.
15. Information about a book may be added, deleted, changed and updated.
16. Equipment can be classified into computer, terminal and others.
17. Equipment information may be added, changed and updated.

In chapter three, we discuss how these assumptions affect the functional design of this information management system.

In the database files, some data items have relationships (associations) with other data items. Generally, there are four relationships between data items: one to one, one to many, many to many and conditional mapping. For example, every social security number has one and only one name associated with it, so between social security number and name, there is one to one mapping. Each object budget code has several sub-object codes associated with it, but each sub-object code has only one object budget
code, this is one to many mapping between object budget code and sub-object code. The student name and course number are a many to many mapping relationship, because a student may take several courses and a course may be taken by a number of students. A conditional mapping means there may have one or no association between two data items. For instance, each offered course has a course number associated with it, but not every course is offered, some courses are offered in the current semester, some are not. Fig. 1-1 is a diagram to show the relationships of the conceptual model for those data files described above.

```
<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRESS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>STUDENT RECORD FILE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>TYPE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>STUDENT ON FINANCIAL AID FILE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE NO.</th>
<th>SSNO</th>
<th>NAME</th>
<th>GRADE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLASS ENROLLMENT FILE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRESS</th>
<th>TITLE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FACULTY RECORD FILE</td>
</tr>
</tbody>
</table>
```
1.2 Definition of Technical Terms

Some terminology of data base organization which have been used in this thesis are described in this section.

1.2.1 Logical and physical data:

Logical data refers to a unit of data which is operated on by the computer programmer. Physical data is the data
which is stored on a disk. A physical data record might contain several logical records, these records can be blocked and chained together in order to save memory space or access time. The programmer or user does not need to know how data records are chained. The conversion of how the data stored on disk from logical to physical record is done by operating system.

1.2.2 Data description:

Usually, we use BYTE, FIELD (DATA ITEM), RECORD, FILE, or DATA BASE to describe data. A byte is the smallest individually addressible group of bits, one byte consists of eight bits. Data item also called 'field', it is the smallest unit in data base; a field may consist of numbers of bytes. A record is composed of one or several data items. File is a collection of records that have the same given type and every record has the same number of data items, as in figure 1-2, a whole set of student records is a data file.

A data base is a collection of stored operational and interrelated data used by the application system of some particular enterprise. The 'enterprise' is a generic term for any commercial, educational, scientific or other operations, such as school, bank, hospital, business company and government. Any enterprise must maintain a lot of data
about its operation, this is its 'operational data'. The operational data for an enterprise could be such things as account data (for bank), patient data (for hospital), student data (for school), or inventory data (for business company).

<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRS</th>
<th>MAJOR</th>
</tr>
</thead>
</table>

Fig. 1-2 A view of data from a programmer.

A key is the data item or field which can identify a record. In figure 1-2, both social security number (SSNO) and NAME can be used as key to identify a record. We discuss primary key and secondary key here.

1. Primary key: A primary key is a data item or group of data items used to uniquely identify one record. Usually, it is used by the computer in searching the record by means of an index or other addressing technique. As in the student record, the SSNO (social security number) is the item that can identify a record uniquely, for it is a primary key.
2. Secondary key: Computer may use a key which does not identify a record uniquely, but can identify all records which have a same certain property. This kind of key is referred to as a secondary key. Figure 1-3 is an example using data item STATE as a secondary key to see all those students from the state of Georgia.

<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRESS</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>245355078</td>
<td>Joe</td>
<td>6, 10th Ave.</td>
<td>Atlanta</td>
<td>GA</td>
<td>30030</td>
</tr>
<tr>
<td>254138987</td>
<td>Liz</td>
<td>101 Bolton Dr.</td>
<td>Marietta</td>
<td>GA</td>
<td>30316</td>
</tr>
<tr>
<td>897037846</td>
<td>Lee</td>
<td>90 P'tree St.</td>
<td>Atlanta</td>
<td>GA</td>
<td>30038</td>
</tr>
<tr>
<td>871140983</td>
<td>Lin</td>
<td>10 Oak Pl.</td>
<td>Dallas</td>
<td>TX</td>
<td>90015</td>
</tr>
<tr>
<td>753478659</td>
<td>Dan</td>
<td>7 P'dmont Rd.</td>
<td>Atlanta</td>
<td>GA</td>
<td>30016</td>
</tr>
</tbody>
</table>

Fig. 1-3 We can use the data item 'STATE' as a secondary key to see those students from state of Georgia.

From the structure of data, we can find relationships between the files of a data base. There are three best-known structure types of data models: tree (hierarchical), network and relational. Since the writer used dBASE II in designing this system of a data base, and dBASE II uses the relational data structure, only the relational data model is discussed here.
To design a data base by relational structure, we need to find a way of describing the data. First, it can be understood easily by users with no training in programming; and secondly, it makes it possible to add new data items, records, and associations to the data base without changing the existing structure and application programs; and to permit the flexibility of handling unanticipated uses of data or casual inquiries at terminals.

The most natural way to represent data to a nonprogramming user is with a two-dimensional table such as in figure 1-3. Both the tree and network structure data base can also be reduced to a group of two-dimensional tables. It has the following properties:

1. Each entry in a table represents one data item, there are no repeating groups.
2. They are column-homogeneous, that means all items in the same column are of the same type.
3. Each column is assigned a different name.
4. All rows are distinct, each row is called a record, and duplicate rows are not allowed.
5. Both the rows and the columns can be viewed in any sequence at any time without affecting the contents.

Any data table as in figure 1-3 which has the
properties above, is referred to as a relation. A data base constructed by using relations is referred to as a relational data base.

Figure 1-4 is an example to explain how to represent a relational structure data file from a tree structure data file. A file which is flat except for a repeating group of fields can be normalized by removing the repeating group into a separate table or flat file. The new file or relation so formed is given a name. The records in it must have keys to uniquely identify them, as in figure 1-4, the data item SSNO is repeated in the new file and combined with SEM:YR and COURSE# to form a unique identifier (primary key).

When two relations share a data item (field) type, they can be joined together. The join operation will put rows together from different relations, figure 1-5 is an example to show some join operations on three relations. When relations are joined on a given data item type, only those records which share the same value of that data item appear in the result. The result relation may contain fewer records than either of the original relations.

A disadvantage sometimes cited for relational data bases is machine performance. If the fields are millions of
bytes long, the join operation will take substantial machine time. It is important to know that the relations and the operations such as join are a part of logical view and do not necessarily exist physically. The advantage of a relational data base are ease of use, flexibility in files, ease of implementation, and data independence (which means that the data and the application programs which use those data are independent, so that either one may be changed without changing the other).

Student Record File:

<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRESS</th>
<th>MAJOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COURSE RECORD FILE

<table>
<thead>
<tr>
<th>SSNO</th>
<th>SEM:YR</th>
<th>COURSE#</th>
<th>GRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REPEATING GROUP DATA ITEMS

Fig. 1-4 A repeating group is removed by splitting the file into two relational table files.
<table>
<thead>
<tr>
<th>SSNO</th>
<th>NAME</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>817-52-30931</td>
<td>Joseph Wan</td>
<td>22 Bolton Dr. Atl. GA</td>
</tr>
<tr>
<td>544-56-08931</td>
<td>Smith Lee</td>
<td>3400 P'tree St. Atl. GA</td>
</tr>
<tr>
<td>352-76-09361</td>
<td>Ben Fung</td>
<td>6 North Ave. Atl. GA</td>
</tr>
<tr>
<td>817-42-89231</td>
<td>John Chen</td>
<td>10 Oakwood Pl. Atl. GA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSNO</th>
<th>SEX</th>
<th>TEL.</th>
<th>OCCUPATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>252-45-9845</td>
<td>M</td>
<td>998-4500</td>
<td>Student</td>
</tr>
<tr>
<td>352-76-0936</td>
<td>M</td>
<td>352-2144</td>
<td>Cashier</td>
</tr>
<tr>
<td>879-74-8746</td>
<td>F</td>
<td>998-2766</td>
<td>Lawyer</td>
</tr>
</tbody>
</table>

JOIN = A1*A2*A3 (NAME, SSNO, TEL AND OCCUPATION)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SSNO</th>
<th>TEL.</th>
<th>OCCUPATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ben Fung</td>
<td>352-76-0936</td>
<td>352-2411</td>
<td>Student</td>
</tr>
</tbody>
</table>

Fig. 1-5 A join operation with three relations.
dBASE II was developed by Ashton Tate, with an English based, high level command language. It is widely used to create data base management systems (DBMS). DBASE II is very useful in establishing and maintaining a data base, and easy to manage small or medium sized data bases. A user can quickly learn how to create a full information system, and to add, delete, edit, display, print, and manipulate the information in the system. By the meantime, dBASE II data and programs are independent, a user can change either one without changing the other.

dBASE II requires 8080, 8050 or Z80 based microprocessor system with CP/M, CDOS, or CROMIX operating systems, or 8086, 8088 based microprocessor system with CP/M-86 or MS-DOS operating systems.

When using dBASE II as a data base management system, it is considerably different from a file handling system. The diagram following is an example of a file handling system, each program processes its own file (i.e. student record program processes student file). If the user wants to
get a report that combines data from different data files, a new program needs to be written. But with a DBMS (such as dBASE II), data can be integrated from different data files and make it much easier to get useful information from different dat files. The difference between a file handling system and a data base management system can be shown obviously in figure 2-1. In diagram 2-1a, it shows all data files are monitored and manipulated by the DBMS and not by the individual application programs.

Fig. 2-1a A data base management system.
The sequence of events when an application program needs a record by using a data base management system can be explained by figure 2-2. The meaning of each step is described as following:

1. An application programmer issues a call to the DBMS to read a record in a file; the program has to give the value of the key of that record the programmer wants.

2. The DBMS obtains the program data description (subschema) that is used by the application program and looks up the description of the data in question.

3. The DBMS obtains the global logical data description (schema) and decides which logical data types are needed.

4. The DBMS searches physical database description and finds the physical record.

5. The DBMS issues a command to the operating system to read the requisite record.
6. The operating system interacts with the physical storage where the data records are kept.
7. The required data is then transferred to a system buffer.
8. The DBMS derives from the data, the logical record needed by the application program.
9. The DBMS transfers the data from the system buffer to a work area.
10. The DBMS shows information to the application program on the outcome of its call, including error indications.
11. The application program can then operate with the data in the work area.

Fig. 2-2 A sequence events for an application program to call a record by DBMS.

As we said earlier, a relational data base like dBASE
II is a great deal simpler. Data is repeated as it is, and the relation between data elements can be considered a two dimensional table like in figure 2-3. Each row going across the table is called a record. Each column is called a field of the record. Each entry in the table must be a single value (no sets, no arrays, etc.). All the entries in a column must be of the same type. Each record (row) is unique, and the order of records (rows) in formal writing use doesn't matter.

2.1 dBASE II RECORDS, DATA TYPES, AND FILES

dBASE II was designed to run on a micro-computer so its scope stops short of infinity. A dBASE II data file can be stored up to 65,535 records, but with the memory and mass storage limitations of a microcomputer, the memory space is really no limitation at all. A record can be as large as 32 fields and 1,000 characters long, with a maximum of 254 characters for each field. Every data field has a name defined by the user, and must contain a single type of data. In dBASE II, there are three kinds of data types: character type (for all ASCII characters), numeric type (between \(1.8 \times 10^{63}\) to \(1.0 \times 10^{-63}\)), logical type (a value of of true or false, yes or no, occupies a field one character long). Each field name can be up to 10 characters long.

There are six different file types in dBASE II, the
file name being limited to 8 characters with a 3 character extension, they are '.DBF' (data base file), '.FRM' (report form file), '.CMD' (command file), '.NDX' (index file), '.MEM' (memory file), '.TXT' (text output file).

2.2. HOW TO USE dBASE II?

To execute a dBASE II program, place dBASE II system disk into any available disk drive. Set that drive to be the default drive (i.e. if the disk is in B drive, type in B:), then enter 'dBASE', the system program will be loaded into memory and request an option to enter the date (screen shows as MM/DD/YY). After the date is typed in, press the return key and the screen will display a dot '.', which means you are in dBASE II mode and the computer is ready to accept a dBASE command. We will see how to create a data base file and how to set up a command file (dBASE II program) in the following.

For example, suppose we want to create a data file for storing student's information and each record will contain student's name, social security number, address, city, state, zip code, telephone number and major. First, we type 'CREATE', dBASE II will respond with: 'ENTER FILENAME:'. The user then enters a filename (defined by user), string starting with a letter and up to 8 characters. Let us say
the filename is called STUREC.DBF. Since each record has eight fields as we wanted, dBASE II needs to know the name of each data field, data type, length of field, and how many decimal places if that data is numeric. The screen shows as:

```
.CREATE
ENTER FILENAME: STUREC
ENTER RECORD STRUCTURE AS FOLLOWS:
FIELD, NAME, TYPE, WIDTH, DECIMAL
001
```

A field name can be up to 10 characters long, starting with a letter and no spaces. Data type is specified by a single letter, C - character, N - numerical, and L - logical. The decimal point also takes one character position if the field is numeric and the decimal place is specified. We can now type in the record structure as follows:

```
001 NAME,C,20
002 SSNO,C,12
003 ADDRESS,C,30
004 CITY,C,20
005 STATE,C,2
006 ZIP,C,5
007 TEL,C,15
008 MAJOR,C,10
009
```

When dBASE II asks for a ninth field, press the return key to end the data definition. All the data structures typed in will be saved and the data file STUREC.DBF is ready
for data entry.

To set up a command file, the programmer lists all the dBASE II commands he wants to be performed and saves those commands. dBASE II starts at the top of the program list and processes one command at a time from left to right until it is done with the list. dBASE uses the carriage return to terminate a command line (BASIC uses a line number, PASCAL uses a semi-column ';' to separate program lines). If we want to create a command file called TEST, we simply type: 'MODIFY COMMAND TEST', and enter the program commands. When we want to run this command file, we type 'DO TEST' to execute the program.

Since its inception, dBASE II has been regarded as a high quality product. Although it is not particularly fast by today's microcomputer standards, there are very few things that dBASE II can not do.
CHAPTER 3

DESCRIPTION OF PROGRAM DESIGN

In this chapter, the writer uses the Mathematical and Computer Sciences Department of Atlanta University as a practical environment in discussing the design of a database management system for an academic department. We determine a method of accomplishing the major tasks to be done and divide them into five components: Student Records, Course Listing, Fiscal Management, Statistical Summary, and Department Information files. By using a database management system as dBASE II, we can design a main program as a user friendly menu to control and handle all problems of these five components. The description of each program design is listed below. A simulation program and list of test run of programs are attached in the appendix.

Program -- MAIN

This is a controlling program consisting of five modules: student records, course listing, fiscal management, statistical summary, and department information. To execute this program, type "DO MAIN". A menu-driven code table will display on the screen as on the next page:
The user may input a number indicating the module to be examined or input 'X' to back to dBASE mode, or input 'Z' to exit to system mode. A input number '1' will execute module of STU program, '2' will execute FIS module, '3' will execute COUR module, '4' will execute STA module, '5' will execute DPT module. Each module is discussed in following sections. Figure 3-1 is a structure chart of this program.

Program -- STU

The function of this program is to create and enter, modify, edit, delete, and extract information of a student record from the data base of the student record file. It can also execute and print a student's academic record report, which includes the grades of all courses he has been taken and the grade point average. A menu-driven code table will show on the screen like this:
Figure 3-2 is a visual table of the structure of this program. To change and delete information of a record, we manipulate the program using EDFIELD and DELEREC. After a command number is selected, the computer will ask the user to enter the student's security number (SSNO), then uses this SSNO as a search key and searches the student record in the data base file. If it can not be found, which means that is a new student, the screen gives the message: '* NOT IN FILE, IS A NEW STUDENT -- ENTER PERSONAL DATA : ' and creates a record in the data file for the user to type the student data in. If the SSNO is found in the file, then that student is an old student. The user can then follow the screen command and go to the next procedure to see all course taken and grades received, or see the courses being taken in the current semester.

Program -- EDFIELD

This program is to change and edit the information of a student record if the personal data is not correct or out-
dated. The screen will show:

' * Enter the SSNO need to EDIT or 0 to exit : ' 

The screen then gives a full screen for the user to edit the record if the SSNO was found in the data file, otherwise the computer will ask the user to enter again.

Program -- DELEREC

The function of this program is to delete a student (or faculty) record from data file. Use the social security number as the search key to delete a record. If the key is found in the file, the screen will ask:

' * Record is found, do you want to delete (y/n) ? '

When you mark a record for deletion, dBASE II will place an '*' by that record. Until you purge that file, the record will remain. It will also delete the whole course record of this student in the class enrollment file (CORLIS.DBF), otherwise the record will be unchanged.

Program -- COUR

This is a sub-control module. The prompts are allowed to set up a course schedule, a class enrollment listing, to add and drop course, and enter grade. A menu-driven code table will display on the screen as:
COMMAND NUMBER

1 - ENTER COURSE SCHEDULE
2 - DELETE COURSE FROM COURSE SCHEDULE
3 - LIST A COURSE SCHEDULE
4 - ADD / DROP, CLASS LISTING, ENTER GRADE
0 - EXIT TO MAIN MENU

ENTER A CODE YOU WANT --

The user can select a number to indicate the program to be executed. Figure 3-3 is an example of contents of this module. The computer will create a record, and give a full screen for the user to enter a course record in the course schedule file. A selection of number '2', dBASE II will execute DELECOUR program. A number of '3' will print out a course schedule of the current semester. Enter number '4' to execute CORLST program.

Program -- DELECOUR

This program is to delete a course record from the course schedule file. The computer will ask the user to enter the course number (i.e. MCS-575) and use this course number as a search key to locate in the data file. After a course has been deleted, and the file has been purged, it can not be brought back.

Program -- CORLST

The functions of this program is to set up and produce
A class enrollment listing for each course listed on the course schedule. Each record of a class listing file includes course name, semester and year, student name, S.S.NO., grade, and grade point average. The student's name and S.S.NO. are entered at the very beginning of a semester. At the end of that semester, after the grade has been submitted by the instructor, the student's grade will be entered accordingly and the grade point average is calculated. The prompts are set up as below:

**COMMAND NUMBER**

1 - ENTER STUDENT DATA  
2 - CLASS ENROLLED LISTING  
3 - ADD / DROP  
4 - ENTER STUDENT'S GRADE  
0 - SUB-MENU

**ENTER A NUMBER ----**

The user enters a number indicating the program to be executed.

Program -- ADDROP

This procedure is to update the class enrollment listing file. Some students might add a course or drop a course after registration. The class listing file then needs to be updated. When we want to add a student and course he added, the procedure is the same as in setting up a class listing file. To drop a course, we enter the student's S.S.NO. and the course number he wants to drop as the search key. The
record will then be deleted from the data file.

Program -- FIS

The functions of this program is to create, modify, edit, extract information and make transactions of an object budget record. A menu-driven code table as below will display on the screen:

COMMAND NUMBER

1 - ENTER BUDGET DATA
2 - BROWSE DATA FILE
3 - MAKE TRANSACTION (DEPOSIT / SUBSTRUCT)
0 - MAIN MENU

PLEASE ENTER A NUMBER -----

Figure 3-4 is a structure chart for this program. Each budget has a budget code which is the primary key, so the user can call a certain budget up by entering its budget code. A main object budget code represents the funding source of that budget record, and the sub-budget codes are the representatives of the sub-divisions (categories) with the main object budget. Program FISSUB is a procedure to make account transactions in a budget.

Program -- FISSUB

This program is to make account deposits or withdrawal transactions in a budget. If we want a certain budget
account, we enter the budget code, then dBASE II searches the budget data file and asks to input the information of transaction if it is found in the data file. The ending balance will also be calculated and show up on the screen.

Program -- STA

The functions of this program include: creating records and entering required information for the students on financial aid and graduates of the current semester, editing or deleting a record or listing the data files. The statistical summary will calculate the number of male and female, foreign and domestic students; and students with Computer Science, Mathematics, and Applied Mathematics majors. A menu-driven table as below will show on the screen.

COMMAND NUMBER

1 - STUDENT ON FINANCIAL AID
2 - GRADUATES
3 - STATISTICAL SUMMARY
0 - MAIN MENU

PLEASE ENTER A NUMBER -----

A visual table contents of this program is in figure 3-5.

PROGRAM -- DPT

This program is to set up files for the faculty, textbooks, and equipment in the department of Mathematical
and Computer Sciences. Screen will display as on next page.

COMMAND NUMBER

1 - FACULTY
2 - TEXTBOOKS
3 - EQUIPMENT
0 - MAIN MENU

PLEASE ENTER A NUMBER ----

The user can enter, change, and delete information of records, and browse these three data files (faculty, textbooks, and equipment) from on the screen. An example of the table contents of this program are in figure 3-6.
Figure 3-1 An example of the table contents of MAIN program.

Figure 3-2 An example of the table contents of STU program.
Fig. 3-3 An example of table contents of COUR program.
Figure 3-4 An example of table contents of FIS program.

Figure 3-5 An example of table contents of STA program.
Figure 3-6  An example of table contents of DPT program.
CONCLUSION

Since software applications have become more and more sophisticated, thousands of jobs are replaced by computers which were done by manual systems before. Approximately thirty years ago, computers were limited to large scale systems because of the high cost. But, with the advent of mini-computers and micro-computers in the 1960's - 1970's, the cost of computers has decreased. Now small organizations, even a family, can afford computers. Micro-computers are more and more widely used now in business applications.

Computers have invaded the schools, not only universities and colleges, but also secondary and elementary schools. These machines are used not only as a source of instruction or instructional purpose, but also to perform information processing operations, such as student registration, grade recording, payroll, course scheduling, student transcript updating, and other administrative tasks. Years ago, teachers and administrators had no choice, they had to squeeze these chores into an already busy schedule. Today, however, the application of computer technology to
these routines has freed many teachers and administrators from having to do them. The results: schools are improving the level of their service, teachers are devoting more time to teaching, administrators are spending more time with critical problems, and school records are more accurate and appear without fuss in standardized formats.
2. C.J. Date, "An introduction to data base system", Edisson Wesley Publishing Comp., N.Y. 1977, p.4-6
3. James Martin, op.cit., p.55-57
4. Ibid., p. 225-226
6. James Martin, op.cit., p.81-83
7. Ashton Tate, op.cit., p.99
8. Ibid., p.91
9. Ibid.,
10. Ibid., p.93, Part-II, p.5-8
11. Ibid., Part-II, p.57-58
12. Ibid.,

42
BIBLIOGRAPHY

2. C.J. Date, An introduction to data base system, Edison Wesley Publishing Company, N.Y. 1977
3. Ellis Hurowitz and Sartaj Sahni, Fundamentals of data structure, Computer Science Inc., Maryland, 1976
5. Tod Katz, dBASEing, the 'DBRUN' time package, PC magazine, Vol. 2, No. 1, pp. 547-552, June, 1983
### APPENDIX A  STRUCTURE OF DATA FILES

**STRUCTURE FOR FILE: A:STUREC .DBF**

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SSNO</td>
<td>C</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>NAME</td>
<td>C</td>
<td>045</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>ADDRS</td>
<td>C</td>
<td>035</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>CITY</td>
<td>C</td>
<td>015</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>STATE</td>
<td>C</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>ZIP:CODE</td>
<td>C</td>
<td>005</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>BIRDATE</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>BPLACE</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>009</td>
<td>SEX</td>
<td>C</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>ETRY</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>MAJOR</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>012</td>
<td>TEL</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>013</td>
<td>NATION</td>
<td>C</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>014</td>
<td>G:SEM</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>015</td>
<td>G:YR</td>
<td>C</td>
<td>004</td>
<td></td>
</tr>
<tr>
<td>016</td>
<td>ADVIOR</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>017</td>
<td>THESIS</td>
<td>C</td>
<td>050</td>
<td></td>
</tr>
<tr>
<td>018</td>
<td>APPON</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>019</td>
<td>THDATE</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL** 00285


### STRUCTURE FOR FILE: A:FACTY .DBF

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>NAME</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>SSNO</td>
<td>C</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>FTIT</td>
<td>C</td>
<td>015</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>FADDRS</td>
<td>C</td>
<td>030</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>FCITY</td>
<td>C</td>
<td>015</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>FSTATE</td>
<td>C</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>FZIP</td>
<td>C</td>
<td>005</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>FTEL</td>
<td>C</td>
<td>013</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL** 00117

| 44 |
### Structure for File: A:FISCMG .DBF

**Number of Records:** 00003  
**Date of Last Update:** 04/15/83

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>G:TIT</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>G:NO</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>B:CODE</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>FD:SOUCE</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>PRODIR</td>
<td>C</td>
<td>017</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>AUHG</td>
<td>C</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>AMT</td>
<td>N</td>
<td>010</td>
<td>002</td>
</tr>
</tbody>
</table>

**Total:** 00133

### Structure for File: A:SUBPRO .DBF

**Number of Records:** 00007  
**Date of Last Update:** 04/15/83

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SNO</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>S:CODE</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>S:TIT</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>S:AMT</td>
<td>N</td>
<td>010</td>
<td>002</td>
</tr>
</tbody>
</table>

**Total:** 00086

### Structure for File: A:CORSCH .DBF

**Number of Records:** 00003  
**Date of Last Update:** 04/15/83

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SEMTR</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>YEAR</td>
<td>C</td>
<td>004</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>CTIT</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>CNO</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>CHUR</td>
<td>N</td>
<td>003</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>TIME</td>
<td>C</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>RM</td>
<td>C</td>
<td>005</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>DAY</td>
<td>C</td>
<td>006</td>
<td></td>
</tr>
<tr>
<td>009</td>
<td>CINSTR</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
</tbody>
</table>

**Total:** 00091
### STRUCTURE FOR FILE: A: EQUIP .DBF

**NUMBER OF RECORDS:** 00003  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>CODE</td>
<td>C</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>NAME</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>CMY</td>
<td>C</td>
<td>030</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>SERV</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>LOC</td>
<td>C</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>TEL</td>
<td>C</td>
<td>012</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL:** 00108

### STRUCTURE FOR FILE: A: FINCE .DBF

**NUMBER OF RECORDS:** 00002  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>FNAME</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>FSSNO</td>
<td>C</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>TYPE</td>
<td>C</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>GRANT: NO</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>BUDGT: NO</td>
<td>C</td>
<td>020</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>AMOUNT</td>
<td>N</td>
<td>008</td>
<td>002</td>
</tr>
<tr>
<td>007</td>
<td>SEMTR</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>YR</td>
<td>C</td>
<td>004</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL:** 00099

### STRUCTURE FOR FILE: A: CORLIS .DBF

**NUMBER OF RECORDS:** 00003  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

<table>
<thead>
<tr>
<th>FLD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SEMTR</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>YR</td>
<td>C</td>
<td>004</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>CNO</td>
<td>C</td>
<td>008</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>CNAME</td>
<td>C</td>
<td>030</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>SNAME</td>
<td>C</td>
<td>025</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>SSSNO</td>
<td>C</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>GRADE</td>
<td>C</td>
<td>001</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>POINT</td>
<td>N</td>
<td>005</td>
<td>002</td>
</tr>
</tbody>
</table>

**TOTAL:** 00093
### STRUCTURE FOR FILE: A:TEXTB .DBF

**NUMBER OF RECORDS:** 00001  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

| FLD | NAME    | TYPE | WIDTH | DEC |  |
|-----|---------|------|-------|-----|-
| 001 | B:NAME  | C    | 032   |     |  |
| 002 | B:COUR  | C    | 007   |     |  |
| 003 | PUBSHER | C    | 035   |     |  |
| 004 | B:TIME  | C    | 012   |     |  |
| 005 | CODE    | C    | 050   |     |  |

**TOTAL:** 00137

### STRUCTURE FOR FILE: A:TRANSFIL.DBF

**NUMBER OF RECORDS:** 00001  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

| FLD | NAME    | TYPE | WIDTH | DEC |  |
|-----|---------|------|-------|-----|-
| 001 | CODE    | C    | 020   |     |  |
| 002 | CHKNO   | C    | 005   |     |  |
| 003 | TIME    | C    | 008   |     |  |
| 004 | TO:WHO  | C    | 020   |     |  |
| 005 | AMT     | N    | 010   | 002 |  |
| 006 | PURPOSE | C    | 050   |     |  |
| 007 | PURPOSE1| C    | 050   |     |  |

**TOTAL:** 00164

### STRUCTURE FOR FILE: A:COURFIL .DBF

**NUMBER OF RECORDS:** 00000  
**DATE OF LAST UPDATE:** 04/15/83  
**PRIMARY USE DATABASE**

| FLD | NAME      | TYPE | WIDTH | DEC |  |
|-----|-----------|------|-------|-----|-
| 001 | NUMBER    | C    | 008   |     |  |
| 002 | COR:NAME  | C    | 030   |     |  |
| 003 | CREDIT    | C    | 002   |     |  |
| 004 | DESCRIPT  | C    | 120   |     |  |

**TOTAL:** 00161
APPENDIX B PROGRAMS

NOTE - THIS IS THE MAIN PROGRAM AND THE CONTROL MODULE
SET TALK OFF
SET DELETED ON
ERASE
DO WHILE T
ERASE
@ 5,2 SAY 'COMMAND'
@ 7,5 SAY '1 - STUDENT RECORDS'
@ 8,5 SAY '2 - FISCAL MANAGEMENT'
@ 9,5 SAY '3 - STATISTICAL SUMMARY (GRADUATES,';
     'FINANCIAL AID)'
@ 10,5 SAY '4 - COURSE LISTING (COURSE SCHEDULING,';
     'CLASS LISTING,';
     'ADD-DROP'
@ 11,10 SAY 'GRADE UPDATING)'
@ 12,5 SAY '5 - DEPARTMENT DATA (FACULTY, TEXTBOOK,';
     'EQUIPMENT)'
@ 13,5 SAY '6 - PRINT MODULE'
@ 15,5 SAY 'X - DBASE MODE'
@ 16,5 SAY 'Z - SYSTEM MODE'
STORE ' ' TO ACT
@ 19,2 SAY '* PLEASE ENTER A NUMBER ' GET ACT
READ
STORE !(ACT) TO ACT
IF ACT='X'
   * exit to dBASE prompt **
   ? ' ** ENTER dBASE COMMAND ***
   CANCEL
ENDIF
IF ACT='Z'
   * exit to system
   ? ' *********************************************
   ? ' PROCESS FINISHED '*********************************************
   QUIT
ENDIF
IF ACT='6'
   DO B:PRT
ENDIF
IF ACT='1'
   DO B:STU
ENDIF
IF ACT='2'
   DO B:FIS
ENDIF
IF ACT='3'
   DO B:STA
ENDIF
IF ACT='M'
   DO B:COUR
ENDIF
IF ACT='5'
   DO B:DPT
ENDIF
ENDDO
NOTE - B:STU THIS PROGRAM IS TO CREATE A NEW STUDENT
NOTE - RECORD OR VIEW STUDENT RECORD FILE
*
ERASE
SET TALK OFF
DO WHILE T
ERASE
@ 2,2 SAY ' COMMAND NUMBER ' 
@ 4,5 SAY ' 0 - MAIN MENU'
@ 5,5 SAY ' 1 - ENTER DATA / SEE STUDENT GRADE RECORD'
@ 6,5 SAY ' 2 - EDIT A RECORD'
@ 7,5 SAY ' 3 - DELETE A RECORD'
@ 8,5 SAY ' 4 - BROWSE FILE '
STORE ' ' TO A
@ 11,2 SAY '* PLEASE ENTER A NUMBER ' GET A
READ
IF A='0'
   * ( back to main menu ) *
   RETURN
ENDIF
IF A='2'
   * edit a record *****
   USE B:STUREC
   DO B:EDFIELD
ENDIF
IF A='3'
   ** delete a record **
   USE B:STUREC
   DO B:DELEREC
ENDIF
IF A='4'
   STORE ' ' TO D
   ACCEPT ' > DO YOU WANT SORT THE FILE ?(Y/N)' TO ASW
   IF !(ASW) = 'Y'
      USE
      USE B:STUREC
      INDEX ON SSNO TO B:STUREC
      USE B:STUREC INDEX B:STUREC
   ENDIF
   IF !(ASW)= 'N'
      USE B:STUREC
   ENDIF
   DO WHILE .NOT. EOF .AND. D # 'Q'
      DISP NEXT 7
      ?
      ? " ENTER 'Q' TO QUIT"
      ?
      WAIT TO B
   IF !(B) = 'Q'
      STORE 'Q' TO D
ENDIF
ENDDO

ENDIF
****** IF = 4 browse data file

IF A='1'
****** enter or pull up a record  ****
STORE ' ' TO SEE
STORE ' ' TO SES
DO WHILE I(SEE) <> 'Q'
  ERASE
  STORE ' ' TO OLD
  STORE ' ' TO T:SSNO
  @ 5,5 SAY '* ENTER STUDENT ID NO, 0 TO EXIT *'
  GET T:SSNO PICT '###-##-####'
  READ
  IF $(T:SSNO,1,1) = 'O' .AND. $(T:SSNO,2,10) = ' - - ' 
    STORE 'Q ' TO SEE
    STORE 'Q ' TO SES
    RETURN
  ENDIF
  IF LEN(TRIM(T:SSNO)) < 11
    ? ' ******** BAD INPUT, INPUT AGAIN ********
    STORE 1 TO B
    DO WHILE B < 20
      STORE B+1 TO B
    ENDDO
    release B
    LOOP
    ** back and enter again **
  ENDIF
  ****««**«« ID INPUT IS RIGHT  ***********
  IF SES <> 'Q'
    STORE ' ' TO T:NAME
    @ 7,5 SAY ' STUDENT NAME ' GET T:NAME
    READ
    USE B:STUREC
    *** search record ****
    LOCATE FOR SSNO=T:SSNO .AND. !(TRIM(T:NAME)) $(1(NAME))
    IF SSNO=T:SSNO .AND. !(TRIM(T:NAME)) $(1(NAME))
      STORE 'Y' TO OLD
      ERASE
      @ 3,2 SAY ' *** RECORD IS IN FILE ****
    ELSE
      ERASE
      ******** A NEW STUDENT ********
      @ 3,2 SAY ' * A NEW STUDENT, PLEASE ENTER HIS DATA *'
      APPEND BLANK
      REPLACE SSNO WITH T:SSNO
REPLACE NAME WITH TRIM(T:NAME)
ENDIF
STORE 'MAJOR: CS-COMPUTER SCI. MATH-MATHEMATICS';
' APM-APPLIED MATH.' TO L
STORE 'NATIONALITY: D - DOMESTIC STUDENT, ';
' F - FOREIGN STUDENT ' TO L1
@ 5,0
@ 20,1 SAY '*' + L
@ 21,1 SAY '*' + L1
@ 5,3 SAY 'SSNO' GET SSNO
@ 5,25 SAY 'NAME' GET NAME
@ 7,3 SAY 'ADDRESS' GET ADDR
@ 8,3 SAY 'CITY' GET CITY
@ 8,30 SAY 'STATE' GET STATE
@ 8,45 SAY 'ZIP CODE' GET ZIP:CODE
@ 9,3 SAY 'BIRTH DAY' ' GET BIRDAT PICT'##-##-##'
@ 9,30 SAY 'BIRTH PLACE' ' GET BPLACE
@ 11,3 SAY 'SEX' GET SEX
@ 11,12 SAY 'ENTRY DATE' ' GET ETRY PICT '##-##-##'
@ 13,3 SAY 'MAJOR' GET MAJOR
@ 13,30 SAY 'TEL' GET TEL PICT '##-##-##'
@ 13,50 SAY 'NATIONALITY(F/D)' GET NATION
READ
ERASE
@ 1,5 SAY '*** GRADUATION INFORMATION ***'
?
*** enter semester and year of graduation ****
STORE '* 1- FALL, 2 - SPRING, 3 - SUMMER '* TO L
@ 3,3 SAY L
STORE ' ' TO TIME
ACCEPT '* ENTER SEMESTER CODE OR RETURN ' TO TIME
? CHR(7)
?
IF TIME='1'
REPLACE G:SEM WITH 'FALL'
ENDIF
IF TIME='2'
REPLACE G:SEM WITH 'SPRING'
ENDIF
IF TIME='3'
REPLACE G:SEM WITH 'SUMMER'
ENDIF
**** enter graduate information ****
?
@ 6,0
@ 6,3 SAY 'SEMESTER OF GRADUATION' ' GET G:SEM
@ 6,39 SAY 'YEAR' ' GET G:YR
@ 8,3 SAY 'ADVISOR' ' GET ADVISOR
@ 10,3 SAY 'THESIS TITLE' ' GET THESIS
@ 12,3 SAY 'APPROVED BY' ' GET APPON
@ 12, 40 SAY 'DATE' GET THDATE PICT '#/#/#/'
READ
ENDIF
IF OLD='Y' .OR. OLD='y'
@ 20, 0
@ 21, 0
STORE ' ' TO B
@ 20, 1 SAY " ** DO YOU WANT STUDENT'S GRADE";
"REPORTING?(Y/N)" GET B
READ
IF B = 'N'
STORE 'Q' TO SEE
ELSE
ERASE
USE B:CORLIS
STORE '1 - CURRENT SEMESTER  2 - WHOLE RECORDS' TO L
@ 4, 2 SAY '*' + L
STORE ' ' TO SEC
@ 8, 3 SAY ' ENTER A NUMBER' GET SEC
READ
IF SEC = '1'
STORE '1 - FALL,  2 - SPRING,  3 - SUMMER' TO L1
@ 6, 2 SAY '*' + L1
STORE ' ' TO SE
STORE ' ' TO YEAR
@ 9, 3 SAY '* YEAR ' GET YEAR
@ 9, 20 SAY '* SEMESTER CODE ' GET SE
READ
STORE ' ' TO SEM
IF SE = '1'
STORE 'FALL ' TO SEM
ENDIF
IF SE = '2'
STORE 'SPRING ' TO SEM
ENDIF
IF SE = '3'
STORE 'SUMMER ' TO SEM
ENDIF
COUNT TO X FOR !(TRIM(SEMTR))=TRIM(SEM);
.AND. YR=YEAR .AND. SSSNO=T:SSNO .AND. !(GRADE) <> 'I';
.AND. !(GRADE)<> 'W' .AND. GRADE <> 'F';
SUM POINT TO SUMT FOR !(TRIM(SEMTR))=TRIM(SEM);
.AND. YR=YEAR;
.AND. SSSNO=T:SSNO
STORE SUMT/X TO GPA
ENDIF
IF SEC = '2'
COUNT TO X FOR SSSNO=T:SSNO .AND. !(GRADE)#'I';
   .AND. !(GRADE)#'W' .AND. !(GRADE) # ' '
SUM POINT TO SUMT FOR SSSNO=T:SSNO
STORE SUMT/X TO GPA
ENDIF
STORE 'YR=YEAR .AND. SEMTR=SEM' TO STING
?
? ' SEMESTER S.S.NO COURSE';
? ' GRADE ';
? '=============================================';
? '============================================'
? IF SEC='1'
   DISP ' ', SEMTR, YR, SSSNO, ' ', GRADE;
   FOR SSSNO=T:SSNO .AND. &STING OFF
ENDIF
IF SEC='2'
   DISP ' ', SEMTR, YR, SSSNO, ' ', GRADE;
   FOR SSSNO=T:SSNO OFF
ENDIF
?
DISP OFF '
*------- GPA = ', STR(GPA,4,2) ENDIF
?
' Q - QUIT or <Return> - CONTINUE '
?
WAIT TO SEE
STORE I(SEE) TO SEE
ERASE
ENDIF
ENDDO
ENDIF
ENDDO
NOTE EDFIELD.PRG  THIS PROGRAM IS TO EDIT A RECORD IN  
*  
STUDENT OR FACULTY FILE.  
*  
SET TALK OFF  
ERASE  
DO WHILE T  
ERASE  
@ 3,5 SAY '***** EDIT A RECORD *****';  
STORE ' ' TO KEY  
@ 6,5 SAY 'ENTER SSNO (0 TO EXIT)' GET KEY PICT '###-##-####'  
READ  
IF $(KEY,1,1)='0' .AND. $(KEY,2,10)=' - - '  
RETURN  
ENDIF  
IF LEN(TRIM(KEY))<11  
? '********* BAD INPUT, ENTER AGAIN *********';  
LOOP  
ENDIF  
LOCATE FOR SSNO=KEY  
IF SSNO=KEY  
@ 9,5 SAY '*** RECORD IS FOUND ***';  
?  
DISP OFF ' ' RECORD NUMBER IS--',#,', 'SSNO,NAME  
ACCEPT ' ' ENTER RECORD NUMBER ' TO K  
STORE ' >>> PRESS CTRL+W TO EXIT <<< ' TO P  
@ 15,2 SAY ' ' NOTE : ' GET P  
STORE 1 TO X  
DO WHILE X < 30  
STORE X+1 TO X  
ENDDO  
EDIT &K  
ELSE  
@ 9,5 SAY '*** RECORD IS NOT IN FILE *****';  
ENDIF  
STORE ' ' TO ANO  
ERASE  
@ 12,3 SAY ' * DO YOU WANT EDIT ANOTHER RECORD?(Y/N)' GET ANO  
READ  
STORE !(ANO) TO ANO  
IF ANO='N'  
RETURN  
ENDIF  
ENDDO
* DELREC.PRG THIS PROGRAM IS TO DELETE A RECORD IN
* STUDENT FILE OR FACULTY FILE
*
SET TALK OFF
ERASE
DO WHILE T
ERASE
STORE ' ' TO KEY
@ 5,2 SAY '********** DELETE A RECORD ***************
@ 7,3 SAY 'ENTER SSNO WANT TO DELETE, 0 TO EXIT';
GET KEY PICT '###-###-####';
READ
IF $(KEY,1,1)='0'. AND. $(KEY,2,10)=' ' RETURN
ENDIF
LOCATE FOR SSNO=KEY
IF SSNO=KEY
@ 10,3 SAY '**** RECORD FOUND ********
?
DISP
STORE ' ' TO SEC
@ 17,3 SAY 'IS THIS THE RECORD YOU WANT';
'TO DELETE?(Y/N)' GET SEC
READ
IF I(SEC)='Y'
DELETE
@ 19,3 SAY 'RECORD HAS BEEN DELETED *
ENDIF
ELSE
@ 10,3 SAY '**** RECORD IS NOT IN FILE *****
STORE 1 TO X
DO WHILE X < 20
STORE X+1 TO X
ENDDO
ENDIF
STORE ' ' TO SEC
@ 22,2 SAY 'DO YOU WANT DELETE ANOTHER RECORD ' GET SEC
READ
IF I(SEC)='N'
RETURN
ENDIF
ENDDO
* STASUB.PRG STATISTICAL SUMMARY OF GRADUATES
SET TALK OFF
USE B:STUREC
STORE "G:SEM # \n" TO D
COUNT TO X1 FOR TRIM(MAJOR)='CS' .AND. &D
COUNT TO X2 FOR TRIM(MAJOR)='MATH' .AND. &D
COUNT TO X3 FOR TRIM(MAJOR)='APM' .AND. &D
ERASE
? ' GRADUATES SUMMARY '
? ' ==--------------------------'
DISP OFF ' COMPUTER SCI. ' +STR(X1,4)
DISP OFF ' MATHEMATICS ' +STR(X2,4)
DISP OFF ' APPLIED MATH. ' +STR(X3,4)
DISP OFF ' ** TOTAL : ' +STR(X1+X2+X3,3)
?
WAIT
* ADDROP.PRG  ADD AND DROP COURSE TO CLASS LISTING FILE
SET TALK OFF
ERASE
DO WHILE T
   ERASE
   @ 5,5 SAY 'COMMAND NUMBER '
   @ 7,8 SAY '0 - MENU '
   @ 8,8 SAY '1 - ADD '
   @ 9,8 SAY '2 - DROP'
   STORE ' ' TO AD
   @ 11,5 SAY '* ENTER A NUMBER ---' GET AD
   READ
   IF AD='0'
      RETURN
   ENDIF
   DO B:SEMTYR
      IF AD='2'
         USE B:CORLIS
         STORE ' ' TO COLIST
         DO WHILE COLIST <> 'N'
            ERASE
            @ 3,5 SAY '***** DROP COURSE *****'
            STORE ' ' TO TSNO
            @ 5,5 SAY '* ENTER STUDENT SSNO ' GET TSNO;
            PICT 'XXX-XX-XXXX'
            STORE ' ' TO TCO
            @ 7,5 SAY '* COURSE WANT TO DROP ' GET TCO;
            PICT 'XXX-XXXX'
            READ
            LOCATE FOR SSSNO=TSNO .AND. CNO=TCO;
            .AND. SEMTR=TSEMR .AND. YR=TYR
            IF SSSNO=TSNO .AND. CNO=TCO .AND. SEMTR=TSEMR;
            .AND. YR=TYR
               ?
               ? '****** RECORD FOUND AND BEEN DELETED ******'
               ?
               GOTO #
               DELETE
               PACK
               STORE 1 TO X
               DO WHILE X < 15
                  STORE X+1 TO X
               ENDDO
            ELSE
               ? '****** STUDENT HAS NOT TAKEN THIS COURSE ******'
               STORE 1 TO X
DO WHILE X < 16
    STORE X+1 TO X
ENDDO
ENDIF
?
ACCEPT * MORE TO DROP?(Y/N) ' TO COLIST
STORE I(COLIST) TO COLIST
ENDDO
ENDIF
IF AD = '1' 
STORE ' ' TO COLIST
USE B:CORLIS
GOTO BOTTOM
DO WHILE COLIST <> 'N'
    ERASE
    @ 5,5 SAY '***** ADD A COURSE *****'
    APPEND BLANK
    REPLACE SEMTR WITH TSEMR
    REPLACE YR WITH TYR
    @ 7,5 SAY 'SEMESTER : '+TSEMR
    @ 7,5 SAY 'YEAR : '+TYR
    @ 9,5 SAY 'SSNO ' GET SSSNO PICT '###-####'
    @ 9,5 SAY 'NAME' GET SNAME
    @ 11,5 SAY 'COURSE NO. ' GET CNO PICT 'XXX-XXXX'
    READ
    STORE ' ' TO CONT
    @ 15,5 SAY ' * MORE TO INPUT ---' GET CONT
    READ
    STORE I(CONT) TO CONT
    IF CONT = 'N'
    STORE 'N' TO COLIST
    ENDDO
ENDIF
ENDDO
NOTE CORLST.PRG CLASS LISTING PROGRAM
* SET UP A CLASS LISTING, ADD/DROP COURSE, AND ENTER
* STUDENT GRADE
SET TALK OFF
ERASE
DO WHILE T
ERASE
@ 5,5 SAY 'COMMAND NUMBER '
@ 7,8 SAY '1 - ENTER DATA FOR A CLASS'
@ 8,8 SAY '2 - CLASS LISTING '
@ 9,8 SAY '3 - ADD / DROP'
@ 10,8 SAY '4 - ENTER GRADE '
@ 11,8 SAY '0 - MENU '
STORE ' ' TO RESP
@ 14,5 SAY '** ENTER A NUMBER ' GET RESP
READ
IF RESP = '0'
  *( return to menu )*RETURN
ENDIF
IF RESP='3'
  DO B:ADDROP
ENDIF
IF RESP='4'
  DO B:GRADE
ENDIF
IF RESP='1'
  ERASE
  USE B:CORLIS
  STORE ' ' TO TSE
  STORE ' ' TO TYR
  STORE '* 1 - FALL, 2 - SPRING, 3 - SUMMER **' TO P
  @ 3,1 SAY P
  @ 5,0 SAY 'CURRENT SEMESTER IS ' GET TSE
  @ 5,30 SAY 'YEAR ' GET TYR
  READ
  IF TSE='1'
    STORE 'FALL ' TO TSEM
  ENDIF
  IF TSE='2'
    STORE 'SPRING ' TO TSEM
  ENDIF
  IF TSE='3'
    STORE 'SUMMER ' TO TSEM
  ENDIF
  STORE ' ' TO NLIST
DO WHILE NLIST <> 'N'
STORE ' ' TO TCO
* ( enter course number ) *
@ 7,5 SAY 'ENTER COURSE NO. ( 0 EXIT ) '; GET TCO PICT 'XXX-XXXX'
READ
IF TCO='0' -'
   STORE 'Q' TO NLIST
   RETURN
ENDIF
STORE ' ' TO COLIST
DO WHILE COLIST <> 'N'
*( create a new record )*
APPEND BLANK
REPLACE SEMTR WITH TSEM, YR WITH TYR, CNO WITH TCO
@ 7,0
@ 7,5 SAY 'COURSE NO. : ' + TCO
@ 9,5 SAY 'STUDENT ID.NO ' GET SSSNO PICT '###-##-####'
@ 10,5 SAY 'NAME ' GET SNAME
READ
@ 13,5 SAY 'MORE STUDENT TAKING THIS';
   'COURSE?(Y/N) ' GET COLIST
READ
STORE !(COLIST) TO COLIST
ENDDO
@ 15,5 SAY 'WANT INPUT ANOTHER COURSE CLASS ?(Y/N)';
   GET NLIST
READ
STORE !(NLIST) TO NLIST
ERASE
ENDDO
ENDIF = 1 INPUT DATA
IF RESP = '2'
STORE ' ' TO COLIST
DO WHILE COLIST <> 'N'
   USE B:CORLIS
   ERASE
   @ 1,5 SAY 'CLASS LISTING ***'
   STORE ' ' TO TSEM
   STORE ' ' TO TYR
   STORE ' ' TO TCO
   @ 3,5 SAY '1 - FALL 2 - SPRING 3 - SUMMER '
   @ 5,5 SAY 'ENTER SEMESTER (NUMBER) ' GET TSEM
   @ 5,37 SAY 'YEAR ' GET TYR
   @ 7,5 SAY 'ENTER COURSE NO.( 0 EXIT ) ';
      GET TCO PICT 'XXX-XXXX'
READ
IF TCO='O - ' RETURN
ENDIF
IF TSEM='1'
STORE 'FALL ' TO TSETR
ENDIF
IF TSEM='2'
STORE 'SPRING ' TO TSETR
ENDIF
IF TSEM='3'
STORE 'SUMMER ' TO TSETR
ENDIF
LOCATE FOR CNO=TCO .AND. SEMTR=TSETR .AND. YR=TYR
IF CNO=TCO .AND. SEMTR=TSETR .AND. YR=TYR
COPY TO B:TEMP FOR CNO=TCO .AND. SEMTR=TSETR .AND. YR=TYR
ELSE
?
? '***** NOT IN FILE, CHECK YOUR INPUT *****
STORE 1 TO X
DO WHILE X < 19
  STORE X+1 TO X
ENDDO
LOOP
ENDIF
USE
USE B:TEMP
COUNT TO X
DISP OFF TSETR, TYR, ' COURSE NO. : '+CNO
STORE 1 TO NUM
DO WHILE NUM <=X
  GOTO NUM
  ? '--------------------------------------------------------'
  DISP OFF ' ',SNAME,SSSNO,GRADE
  STORE NUM+1 TO NUM
ENDDO
WAIT
USE
ERASE
STORE ' ' TO CO
@ 12,2 SAY '* WANT ANOTHER CLASS LISTING ?(Y/N) GET CO
READ
STORE !(CO) TO CO
IF CO='N'
  STORE 'N' TO COLIST
ENDIF
ENDDO
ENDIF
ENDDO
* STASUB.PRG STATISTICAL SUMMARY
SET TALK OFF
COUNT TO X1 FOR !(TRIM(TYPE))= 'RA'
COUNT TO X2 FOR !(TRIM(TYPE))= 'TA'
COUNT TO X3 FOR !(TRIM(TYPE))= 'TS'
ERASE
?
? ' STUDENT ON FINANCIAL AID'
? '==================================='
DISP OFF ' RESEARCH ASSISTANTSHIP ' +STR(X1,4)
DISP OFF ' TEACHING ASSISTANTSHIP ' +STR(X2,4)
DISP OFF ' TUITION SCHOLARSHIP ' +STR(X3,4)
DISP OFF ' * TOTAL : ' +STR(X1+X2+X3,4)
?
USE
NOTE DPT.PRG THIS PROGRAM IS TO SET UP FILES FOR FACULTY, * TEXTBOOKS, EQUIPMENT. FUNCTIONS: CREATE, ENTER, * EDIT, AND DELETE A RECORD, ALSO LIST THE FILE

SET TALK OFF
ERASE
DO WHILE T
ERASE
?

COMMAND
?
0 - EXIT
1 - FACULTY FILE
2 - TEXT BOOK FILE
3 - EQUIPMENT FILE

ACCEPT * SELECT A NUMBER PLEASE ------- ' TO W

? CHR(7)
IF W = 'O'
RETURN
ENDIF
IF W = '1'
USE B:FACTY
STORE T TO AX
DO WHILE AX
ERASE
@ 1,5 SAY 'COMMAND NUMBER'
@ 3,10 SAY '1 - ENTER DATA'
@ 4,10 SAY '2 - EDIT RECORD'
@ 5,10 SAY '3 - DELETE RECORD'
@ 6,10 SAY '4 - SEE FILE'
@ 8,10 SAY '0 - EXIT'

ACCEPT * ENTER A NUMBER --- ' TO MOVE + CHR(7)
IF MOVE= 'O'
STORE F TO AX
ENDIF
IF MOVE= '2'
DO B:EDFIELD
ENDIF
IF MOVE= '3'
DO B:DELEREC
ENDIF
IF MOVE= '4'
ERASE
?
**** FACULTY / STAFF LISTING FILE ****
?
?
TITLE NAME SSNO';
TEL. '
?
'DISPLAY ALL '+FTIT, NAME, SSNO,' ', FTEL OFF
?
ENTER <RETURN> TO CONTINUE'
WAIT
ENDIF
IF MOVE='1'
ERASE
?
'**** FILE OF DEPARTMENT FACULTY MEMBERS ****'
?
'STORE ' ' TO CONTINUE
DO WHILE CONTINUE <> 'Q' .AND. CONTINUE <> 'q'
APPEND BLANK
@ 3,0 SAY '** ENTER THE FOLLOWING INFORMATION '
@ 5,0 SAY ' NAME ' GET NAME
@ 6,0 SAY ' S.S.NO ' GET SSNO PICT '###-##-####'
@ 8,0 SAY ' TITLE ' GET FTIT
@ 10,0 SAY ' ADDRESS ' GET FADDRS
@ 11,0 SAY ' CITY ' GET FCITY
@ 11,30 SAY ' STATE ' GET FSTATE
@ 11,40 SAY ' ZIP:CODE ' GET FZIP
@ 13,0 SAY ' TEL. ' GET FTEL PICT '###-####'
READ
@ 15,0
ACCEPT ' ** WANT TO CONTINUE, (Y/N) ' TO CONT
? CHR(7)
STORE I(CONT) TO CONT
IF CONT = 'N'
STORE 'Q' TO CONTINUE
ENDIF
ERASE
ENDDO
ENDIF

**** END OF IF SELECT='1' <FACULTY FILE> ****

IF W='2'
ERASE
USE B:TEXTB
STORE T TO CBOOK
DO WHILE CBOOK
ERASE
@ 5,5 SAY 'COMMAND NUMBER'
@ 7,7 SAY ' 0 - EXIT'
@ 8,7 SAY ' 1 - SEE FILE'
@ 9,7 SAY ' 2 - ENTER DATA'
@ 10,7 SAY ' 3 - EDIT or DELETE A RECORD '
? ACCEPT ' ENTER A NUMBER---' TO BOOK
? CHR(7)
IF BOOK=0'0'
    STORE F TO CBOOK
ENDIF
IF BOOK=1'
    USE B:TEXTB
    COUNT TO X
    STORE 1 TO NUM
? '----------------------------------------'
DO WHILE NUM <= X
GOTO NUM
DISPLAY OFF ' RECORD NO: ',STR(NUM,3)
DISP OFF ' COURSE NUME: ',B:NAME, ' CODE : '+CODE
DISP OFF ' SEMESTER USING THIS BOOK: • + B:TIME
DISP OFF ' PUBLISHER: '+PUBSER
? '----------------------------------------'
STORE NUM+1 TO NUM
IF NUM > X

? ' **** END OF FILE ****
    WAIT
    STORE F TO CBOOK
ENDIF
ENDDO
ENDIF
IF BOOK=3'
ERASE
USE B:TEXTB
COUNT TO LAST
GOTO TOP
STORE ' ' TO CONT
DO WHILE CONT <> 'N'
? ' **** EDIT or DELETE A RECORD ****
INPUT ' ** RECORD NUMBER YOU WANT -- ' TO NUM
IF NUM > LAST

? ' **** RECORD OUT OF RANGE ****
LOOP
ELSE
   GOTO NUM
@ 10,5 SAY 'BOOK ' GET B:NAME
@ 11,5 SAY 'CODE ' GET CODE
@ 12,30 SAY 'COURSE USE ' GET B:COUR
@ 14,5 SAY 'PUBLISHER ' GET PUBLISHER
@ 15,5 SAY 'SEMESTER USE ' GET B:TIME
?
ACCEPT ' ** EDIT or DELETE ( E or D ) ' TO CH
? CHR(7)
IF !(CH) = 'E'
   READ
ELSE
   IF !(CH)='D'
      ACCEPT ' ** ARE YOU SURE (Y/N) ' TO SURE
      IF !(SURE)='Y'
         DELETE
      ENDIF
   ENDIF
   ACCEPT ' ** DO YOU WANT ANOTHER RECORD (Y/N) ';
   TO CONT + CHR(7)
   STORE !(CONT) TO CONT
ENDDO
ENDIF
IF BO0K='2'
ERASE
? ' ***** TEXT BOOK FILE *****
?
STORE ' ' TO CONTINUE
?
DO WHILE CONTINUE <> 'Q' .AND. CONTINUE <> 'q'
   APPEND BLANK
   @ 5,0 SAY '* INFORMATION OF TEXTBOOK *
   @ 8,0 SAY ' NAME OF BOOK ' GET B:NAME
   @ 9,0 SAY ' COURSE USE ' GET B:COUR PICT ' - ' 
   @ 10,0 SAY ' SEMESTER/YR ' GET B:TIME PICT 'XXXX/XXXX'
   @ 11,0 SAY ' PUBLISHER ' GET PUBLISHER
   @ 12,0 SAY ' ADDRESS ' GET CODE
   @ 15,0 SAY '*** MAKE SURE YOUR INPUT IS CORRECT ***'
   READ
?
? ' Q TO STOP THE PROCEDURE'
? ' <RETURN> TO CONTINUE.'
?
WAIT TO CONTINUE
ERASE
ENDDO
ENDIF
ENDDO
ENDIF
ENDIF
**********END OF IF SELECTION=’2’ (TEXTBOOK FILE) ******
*
( Equipment control )
*
IF W=’3’
DO WHILE T
ERASE
?
?  * EQUIPMENT ( COMPUTER, TERMINALS, AND OTHERS ) *’
?
?  0 - EXIT ’
?  1 - ENTER DATA’
?  2 - SEE FILE ’
?  3 - EDIT or DELETE A RECORD ’
? ACCEPT ‘* PLEASE ENTER A NUMBER’ TO SELECT
USE B:EQUIP
IF SELECT =’0’
  RETURN
ENDIF
IF SELECT=’2’
ERASE
?
’  ******** EQUIPMENTS OF DEPARTMENT ********
?
COUNT TO X
STORE 1 TO NUM
STORE ’ ’ TO CONT
DO WHILE NUM <= X .AND. CONT <> ’N’
GOTO NUM
? ’----------------------------------------------------------’
DISPLAY OFF ’ RECORD # ’+STR(NUM,3),’  MODEL CODE : ’+CODE
DISP OFF ’ NAME : ’+NAME
DISP OFF ’ COMPANY : ’+CMY
DISP OFF ’ SALES REPRESENTATIVE : ’+SERV
DISP OFF ’ LOCATION : ’+LOC,’  TEL. ’+TEL
? ’----------------------------------------------------------’
STORE NUM+1 TO NUM
IF NUM > X
?
’  ******** END OF FILE *******
STORE ’N’ TO CONT
WAIT
ENDIF
ENDDO
ENDIF
IF SELECT = '3'
*( Change information or delete a record )*
ERASE
STORE ' ' TO CONT
DO WHILE !(CONT) <> 'N'
ERASE
@ 3,0
? ' **** EDIT or DELETE A EQUIPMENT RECORD **** ' 
? INPUT ' ** ENTER RECORD NUMBER --- ' TO X
GOTO X
* ( pull record out ) *
@ 8,5 SAY 'MODEL CODE ' GET CODE
@ 8,40 SAY ' NAME ' GET NAME
@ 10,5 SAY 'EQUIPMENT COMPANY ' GET CMY
@ 11,5 SAY 'SERVICE REPRESENTATIVE ' GET SERV
@ 12,5 SAY 'LOCATION ' GET LOC
@ 14,5 SAY 'TEL. NO. ' GET TEL
?
ACCEPT ' ** EDIT or DELETE --( E or D) ' TO CH
STORE !(CH) TO CH
IF CH='E'
* change data **********
READ
ENDIF
* delete record ********
IF CH='D'
ACCEPT ' ** ARE YOU SURE ? (Y/N) ' TO SURE
IF !(SURE) = 'Y'
DELETE
? ' **** do not interrupt **** ' 
pack
ENDIF
ENDIF
?
ACCEPT ' * DO YOU WANT ANOTHER RECORD (Y/N)? ' TO CONT
ENDO
ENDIF
IF SELECT='1'
?
? ' * EQUIPMENT IN DEPARTMENT *'
? '---------------------------------' 
?
STORE ' ' TO SEE
DO WHILE SEE <> 'Q'
ERASE
APPEND BLANK
*5,5 SAY '**** INFORMATION OF EQUIPMENT *****',
*7,0 SAY ' EQUIPMENT NAME ' GET NAME
*7,37 SAY ' CODE ' GET CODE
*9,0 SAY ' EQUIPMENT COMPANY ' GET CMY
*11,0 SAY ' SERVICE REPRESENTATIVE ' GET SERV
*13,0 SAY ' EQUIPMENT LOCATION ' GET LOC
*15,0 SAY ' SERVICE PHONE NUMBER ' GET TEL PICT '###-####'
READ
? ' Q - Quit '
? ' <RETURN> TO CONTINUE'
?
WAIT TO SEE
ERASE
ENDDO
ENDIF
ENDDO
ENDDO
* STA.PRG STATISTICAL SUMMARY
SET TALK OFF
ERASE
DO WHILE T
ERASE
@ 5,0 SAY 'COMMAND NUMBER'
@ 7,3 SAY '0 - MAIN MENU'
@ 8,3 SAY '1 - STUDENT ON FINANCIAL AID'
@ 9,3 SAY '2 - GRADUATES'
@ 10,3 SAY '3 - STATISTICAL SUMMARY '
STORE ' ' TO NOIN
@ 12,1 SAY '* ENTER A NUMBER YOU WANT ' GET NOIN
READ
IF NOIN='0'
  RETURN
ENDIF
IF NOIN='1'
  USE B:FINCE
 **********************
STORE ' ' TO ANS
ERASE
@ 4,4 SAY 'COMMAND NUMBER'
@ 7,5 SAY ' 1 - ENTER DATA '
@ 9,5 SAY ' 2 - LOOK FILE LISTING '
@ 11,5 SAY ' 3 - EDIT or DELETE A RECORD '
@ 15,4 SAY '* ENTER A NUMBER --- ' GET ANS
READ
IF ANS='2'
  *( browse data file ) ********
  LIST OFF
  WAIT
ENDIF
IF ANS='3'
  *( delete or change information of a record ) *
  DO B:STAFIN
ENDIF
IF ANS = '1'
ERASE
STORE ' ' TO CONT
STORE ' ' TO SEMNO
STORE ' ' TO TYR
?
STORE ' 1 - FALL,  2 - SPRING,  3 - SUMMER ' TO P
ERASE
*( enter semester and year ) *****
@ 5,3 SAY P
@ 8,3 SAY ' * SEMESTER ( CODE ) ' GET SEMNO
@ 8,35 SAY ' * YEAR ' GET TYR PICT '#####'
READ
IF SEMNO='1'

STORE 'FALL' TO TSEMTR
ENDIF
IF SEMNO='2'
  STORE 'SPRING' TO TSEMTR
ENDIF
IF SEMNO='3'
  STORE 'SUMMER' TO TSEMTR
ENDIF
?
STORE 'RA-RESEARCH ASSISTANTSHIP, TA-TEACHING ASSISTANT';
  'SHIP TS-TUITION SCHOLARSHIP' TO D
DO WHILE CONT <> 'Q'
  APPEND BLANK
  REPLACE YR WITH TYR
  REPLACE SEMTR WITH TSEMTR
  @ 10,5 SAY 'STUDENT NAME' GET FNAME
  @ 11,12 SAY 'SSNO' GET FSSNO PICT '####-####'
  @ 12,5 SAY 'AID TYPE' GET TYPE
  @ 12,18 SAY '(RA, TA, TS)'
  @ 13,5 SAY 'GRANT NO.' GET GRANTNO
  @ 14,5 SAY 'BUDGET NO.' GET BUDGNTNO
  @ 15,5 SAY 'TOT AMOUNT' GET AMOUNT
  @ 17,5 SAY 'SEMESTER: '+SEMTR
  @ 17,27 SAY 'YEAR: '+YR
  @ 20,1 SAY D
  READ
  STORE ' ' TO REINP
  @ 22,1 SAY '* ANOTHER RECORD?(Y/N)' GET REINP
  READ
  IF !(REINP)='N'
    STORE 'Q' TO CONT
  ENDFI
  ERASE
  ENDDO
ENDIF
***** end of financial aid *****
IF NOIN='2'
  ***** for graduates file *****
  USE BRSTUREC
  ERASE
  ACCEPT '*' FOR A CERTAIN SEMESTER OR WHOLE(C/W)' TO CW
  IF !(CW)='C'
    * enter semester and year *
    DO B:SEMTYR
  ENDFI
  ACCEPT '*' DO YOU WANT TO PRINT?' TO PNT
  ?
  IF !(PNT)='Y'
? ' ***** SET TOP OF PAGE ***** ?
WAIT
SET PRINT ON
ENDIF
?
DISP OFF ' **** GRADUATES LIST ****
DISP OFF '__________________________
IF !(CW) = 'C'
ELSE
LIST OFF SSNO, NAME, G:SEM, G:YR FOR G:SEM # ' ' ;
         . AND. G:YR # ' ' ;
ENDIF
SET PRINT OFF
WAIT
ENDIF

IF NOIN = '3'
**** statistical summary list ****
STORE "G:SEM = ' ' . AND. G:YR = ' ' " TO L
ERASE
USE B:STUREC
COUNT TO X1 FOR 'M' $(I(SEX)) . AND. &L
COUNT TO X2 FOR 'F' $(I(SEX)) . AND. &L
COUNT TO X3 FOR 'D' $(I(NATION)) . AND. &L
COUNT TO X4 FOR 'F' $(I(NATION)) . AND. &L
COUNT TO X5 FOR 'CSI $(I(MAJOR)) . AND. &L
COUNT TO X6 FOR 'MAHI $(I(MAJOR)) . AND. &L
COUNT TO X7 FOR 'APM $(I(MAJOR)) . AND. &L
?
? ' SUMMARY LIST'
? '================================================================================'
@ 4,3 SAY 'MALE: ' + STR(X1,4)
@ 4,30 SAY 'FEMALE: ' + STR(X2,4)
@ 6,3 SAY 'DOMESTIC: ' + STR(X3,4)
@ 6,30 SAY 'FOREIGN: ' + STR(X4,4)
@ 8,3 SAY 'MAJOR: '
@ 9,5 SAY 'CS: ' + STR(X5,3)
@ 9,25 SAY 'MATH: ' + STR(X6,3)
@ 9,50 SAY 'APM: ' + STR(X7,3)
COUNT TO X
@ 12,3 SAY ' # TOTAL: ' + STR(X,4)
?
? ' # ENTER ANY KEY ' ?
WAIT
USE
USE B:FINCE
COUNT TO X1 FOR TRIM(TYPE) = 'RA'
COUNT TO X2 FOR TRIM(TYPE) = 'TA'
COUNT TO X3 FOR TRIM(TYPE) = 'TW'
ERASE
? ' STUDENT ON FINANCIAL AID ' '======================================
@5,5 SAY ' RESEARCH ASSISTANTSHIP ' +STR(X1,4)
@7,5 SAY ' TEACHING ASSISTANTSHIP ' +STR(X2,4)
@9,5 SAY ' TUITION SCHOLARSHIP ' +STR(X3,4)
@12,5 SAY '# TOTAL : '+STR(X1+X2+X3,4)
?
' # ENTER ANY KEY '
WAIT
USE
USE B:STUREC
STORE "G:SEM <> ' ' " TO D
?
COUNT TO X1 FOR TRIM(MAJOR)= 'CS' .AND. &D
COUNT TO X2 FOR TRIM(MAJOR)= 'MATH' .AND. &D
COUNT TO X3 FOR TRIM(MAJOR)= 'APM' .AND. &D
ERASE
@3,0
? ' IF YOU WANT THE GRADUATES FOR A PARTICULAR SEMESTER'
' PLEASE GO BACK TO MAIN MENU, AND ENTER NUMBER "6" '
STORE 1 TO X
DO WHILE X < 25
STORE X+1 TO X
ENDDO
? ' GRADUATES SUMMARY ' ' GLASSLERSUMMARY ' '======================================
@5,5 SAY ' COMPUTER SCI. ' +STR(X1,4)
@7,5 SAY ' MATHEMATICS ' +STR(X2,4)
@9,5 SAY ' APPLIED MATH. ' +STR(X3,4)
@12,4 SAY '# TOTAL : '+STR(X1+X2+X3,3)
?
WAIT
ENDDIF
ENDDO **********LAST
NOTE FISEE.PRG
* THIS PROGRAM IS TO SEE THE CONTENTS OF BUDGET FILE
*
SET TALK OFF
ERASE
USE B:FISCMG
COUNT TO X
STORE 1 TO CT
DO WHILE T
USE B:FISCMG
IF CT > X
@ 22,3 SAY '********** END OF FILE **********'
STORE 1 TO ZO
DO WHILE ZO < 20
STORE ZO+1 TO ZO
ENDDO
RELEASE ZO
RETURN
ENDIF
GOTO CT
ERASE
STORE STR(CT,3) TO I
@ 5,5 SAY ' ** THIS IS RECORD &I '
@ 7,0 SAY 'GRAND TITLE' GET G:TIT
@ 7,40 SAY 'NUMBER ' GET G:NO
@ 9,0 SAY 'BUDGET CODE ' GET B:CODE
@ 9,40 SAY 'AMOUNT ' GET AMT
@ 11,0 SAY 'FUNDING SOURCE ' GET FD:SOUCE
@ 11,40 SAY 'DIRECTOR ' GET PRODIR
@ 12,0 SAY 'AUG# ' GET AUHG
STORE B:CODE TO TNO
**** TNO will be stored to SUB-OBJECT file ****
USE
USE B:SUBPRO
***** search record *****
LOCATE FOR TRIM(SNO)=TRIM(TNO)
IF TRIM(SNO)=TRIM(TNO)
***** record found *****
@ 14,0 SAY '-------- SUB BUDGET :
@ 16,2 SAY ' CODE TITLE '
               ' AMOUNT'
@ 17,2 SAY '=====================================
               '=====================================

@ 18,0
LIST OFF ALL S:CODE, S:TIT, S:AMT FOR TRIM(SNO)=TRIM(TNO)
ELSE
@ 16,0 SAY '-------- NO SUB BUDGET IN THIS OBJECT BUDGET'
***** go to next object *****
STORE 1 TO ZO
DO WHILE ZO < 15
STORE \textit{Z0}+1 \text{ TO } \textit{ZO}
ENDDO
ENDIF
STORE ' ' \text{ TO } \textit{NEX}
ACCEPT ' ' \text{ * DO YOU WANT NEXT RECORD?(Y/N) ' } \text{ TO } \textit{NEX}
STORE !(\textit{NEX}) \text{ TO } \textit{NEX}
IF \textit{NEX}= 'N'
\text{ RETURN}
ELSE
STORE CT+1 \text{ TO } \textit{CT}
ENDIF
ENDDO
NOTE SEMTYR THIS PROGRAM IS TO ENTER THE SEMESTER WHICH
* THE USER WANTS
*
SET TALK OFF
ERASE
@ 5,3 SAY '* 1 - FALL 2 - SPRING 3 - SUMMER * '
STORE ' ' TO TSEM
STORE ' ' TO TYR
@ 7,3 SAY '* ENTER A SEMESTER NUMBER ' GET TSEM
@ 7,38 SAY 'YEAR ' GET TYR
READ
IF TSEM='1'
   STORE 'FALL ' TO TSEMR
ENDIF
IF TSEM='2'
   STORE 'SPRING ' TO TSEMR
ENDIF
IF TSEM='3'
   STORE 'SUMMER ' TO TSEMR
ENDIF
RELEASE TSEM
NOTE STASUB1 STATISTICAL SUMMARY LIST FOR STUDENTS
* ENROLLED IN CURRENT SEMESTER
SET TALK OFF
COUNT TO X1 FOR !(TRIM(SEX))='M'
COUNT TO X2 FOR !(TRIM(SEX))='F'
COUNT TO X3 FOR !(TRIM(NATION))='D'
COUNT TO X4 FOR !(TRIM(NATION))='F'
COUNT TO X5 FOR !(TRIM(MAJOR))='CS'
COUNT TO X6 FOR !(TRIM(MAJOR))='MATH'
COUNT TO X7 FOR !(TRIM(MAJOR))='APM'
SET PRINT ON
? ' STUDENT SUMMARY LIST (CURRENT SEMESTER) ' 
? ' ==------------------------------------------------=='
? 
DISP OFF ' MALE : '+STR(X1,4), 'FEMALE : '+STR(X2,4)
DISP OFF ' DOMESTIC : '+STR(X3,4), ' FOREIGN : '
+STR(X4,4)
DISP OFF ' MAJOR: ' 
DISP OFF ' CS : '+STR(X5,3), ' MATH : '+STR(X6,3); 
' APM : '+STR(X7,3)
DISP OFF ' * TOTAL : '+STR(X5+X6+X7,4) 
?

WAIT
NOTE * THIS IS A PRINT MODULE *

SET TALK OFF
ERASE
DO WHILE T
ERASE
?
* WHICH FILE OF FOLLOWING YOU WANT ? *
?

0. EXIT
1. STUDENT FILE (CURRENT ENROLLED)
2. GRADUATES FILE
3. STUDENTS ON FINANCIAL AID
4. TEXTBOOKS FILE
5. EQUIPMENT FILE
6. FACULTY/STAFF FILE
7. STATISTICAL SUMMARY
8. COURSE SCHEDULE
9. CLASS LISTING
A. OBJECT BUDGET FILE
B. SUB-OBJECT BUDGET FILE
C. BUDGET ACCOUNT TRANSACTIONS
D. STUDENT GRADE REPORT
?

ACCEPT * ENTER A NUMBER YOU WANT --- TO NUM
? CHR(7)
IF NUM='0'
    RETURN
ENDIF
IF !(NUM)='D'
    DO B:STUPRT
ENDIF
IF NUM='1'
    USE B:STUREC
    COPY TO B:STUTEMP FOR G:SEM = ' ' .AND. G:YR = ' ' 
    USE B:STUTEMP
    IF EOF
        ? ' ********* NO RECORD IN FILE ***********
        WAIT
    ENDIF
    ? ' ********* DO NOT INTERRUPT ******
? CHR(7)
INDEX ON SSNO TO B:STUTEMP
USE B:STUTEMP INDEX B:STUTEMP
COPY TO B:STEMP
USE
DELE FILE B:STUTEMP
USE B:STEMP
STORE 1 TO X
COUNT TO LAST
GOTO TOP
DO WHILE X <= LAST
  SET PRINT OFF
  ? ' ** Ready to print (set top of page) ** ' -
  wait
  SET PRINT ON
  ? ' __________________ STUDENT RECORDS LISTING ____________?
STORE 1 TO REC
? '---------------------------------------------';
 '---------------------------------------------'
DO WHILE REC <= 5
GOTO X
DISP OFF 'S.S NO.: '+SSNO, ' Name: '+TRIM(NAME);
  ' TEL.: '+TEL
DISP OFF 'Address: '+TRIM(ADDR)+', '+CITY,STATE;
  ZIP:CODE
DISP OFF 'Birth date:, birthat, ' Sex: ', sex;
  ' Nationality: ', NATION
disp off 'Major: '+MAJOR, ' Advisor: '+ADVOR
? '---------------------------------------------';
 '---------------------------------------------'
STORE X+1 TO X
STORE REC+1 TO REC
IF X > LAST
  WAIT
  DO B:STASUB1
  ? ** CALCULATE THE STATISTICAL SUMMARY **
  ? STORE 100 TO REC
  SET PRINT OFF
  ? '________________________ END OF FILE __________________________
  WAIT
ENDIF
ENDDO
ENDDO
ENDIF
IF NUM='2'
USE B:STUREC
ERASE
ACCEPT '* C - A CERTAIN SEMESTER W - WHOLE LIST ' TO CW
IF I(CW)='C'
  DO B:SEMTYR
  COPY TO B:GRADU FOR G:SEM = TSEM .AND. G:YR = TYR
ELSE
  COPY TO B:GRADU FOR G:SEM # ' .AND. G:YR # ' ""
ENDIF
USE B:GRADU
COUNT TO LAST
STORE 1 TO REC
STORE 1 TO X
DO WHILE REC <= LAST
SET PRINT OFF
? 
? ' ** Ready to print (set to top of page) ? ----'
wait
set print on
?
? '----------------------------- GRADUATES LISTING -----------------------------('

? '-----------------------------------'

? ' ---------------------------------------'
do while X <= 5
GOTO REC
disp off 'S.S.NO. :',SSNO,' Name :',trim(NAME);
' TEL. : '+TEL
DISP OFF 'Address : '+trim(ADDR),' '+CITY+STATE,ZIP:CODE
DISP OFF 'Major : '+MAJOR,' Date of graduation : '
+G:SEM,G:YR
DISP OFF 'Thesis title :'+THESIS
disp off 'Approved by :'+appon,' Date :',THDATE
? '---------------------------------------'

store x+1 TO X
STORE REC+1 TO REC
IF REC > LAST
WAIT
?
*** CALCULATE THE STATISTICAL SUMMARY ***
DO B:STASUB3
STORE 100 TO X
SET PRINT OFF
? 
END OF FILE

ENDIF
ENDDO
ENDDO
ENDIF **********=2 GRADUATES FILE
IF NUM='3'
USE B:FINCE
SET PRINT ON
?
? ' ** STUDENT ON FINANCIAL AID LISTING **'

? ' S.S.NO | NAME | AID TYPE |
' BUDGET NO. | AMOUNT |
LIST ALL FSSNO, ''FNAME, ''TYPE, BUDGT: NO, ''AMOUNT, ''SemTR, YR OFF

IF EOF
WAIT
?
*** CALCULATE THE STATISTICAL SUMMARY ***
DO B:STASUB2
ENDIF
SET PRINT OFF
WAIT
ENDIF
IF NUM='4'
USE B:TEXTB
COUNT TO LAST
STORE 1 TO X
STORE 1 TO REC
DO WHILE REC <= LAST
?
? '******** Ready to PRINT ? ( Set top of page ) ********
wait
STORE 1 TO X
SET PRINT ON
?
? '*** LISTING OF TEXTBOOKS *** '
? '=================================================='
DO WHILE X <= 10
GOTO REC
DISP OFF 'BOOK:', B:NAME
DISP OFF 'COURSE USING: ' + B:COUR;
   SEMESTER USING : '+B:TIME
DISP OFF 'PUBLISHER : '+PUBSHER
DISP OFF 'SEMESTER USING : '+B:TIME
? '=================================================='
STORE X+1 TO X
STORE REC+1 TO REC
IF REC > LAST
STORE 100 TO X
SET PRINT OFF
?
'******** END OF FILE ********
WAIT
ENDIF
ENDDO
ENDIF
******** =4 TEXTBOOK FILE
IF NUM='5'
USE B:EQUIP
STORE 1 TO X
COUNT TO LAST
GOTO TOP
? ' **** READY TO PRINT ? (SET TO TOP OF PAGE ) --'
WAIT
SET PRINT ON
DO WHILE X < LAST
? ' **** LISTING OF EQUIPMENT ****
? '----------------------------------------';
'---------'
STORE 1 TO REC
DO WHILE 1 REC <= 10
GOTO X
DISP OFF 'Model name :',NAME, ' Model number : ' +CODE
DISP OFF 'Company : '+CMY,' Service : '+SERV
DISP OFF 'Location :',loc, ' Tel. : '+TEL
? '----------------------------------------';
'---------'
STORE X+1 to X
STORE REC+1 TO REC
IF X > LAST
SET PRINT OFF
? ' **** END OF FILE ****
WAIT
STORE 100 TO REC
ENDIF
ENDDO
ENDIF ************=5 EQUIPMENT FILE
IF NUM='6'
USE B:FACTY
COUNT TO LAST
STORE 1 TO X
? ' READY TO PRINT ?'
WAIT
? CHR(7)
SET PRINT ON
DO WHILE X <= LAST
STORE 1 TO REC
? ' FACULTY/STAFF LISTING ***********
? '----------------------------------------';
'---------'
DO WHILE REC <= 10
GOTO REC
DISP OFF 'NAME : '+NAME, ' S.S.NO. :',SSNO
DISP OFF 'ADDRESS : '+TRIM(FADDRS)+ ' '; FSTATE+FZIP
DISP OFF 'TITLE : '+FTIT, ' Tel. : '+FTEL
? '----------------------------------------';
STORE REC+1 TO REC
STORE X+1 TO X
IF X > LAST
   STORE 100 TO REC
   SET PRINT OFF
   ? ' END OF FILE
   WAIT
ENDIF
ENDDO
ENDDO
ENDDO
ENDIF

IF NUM='7'
SET PRINT ON
USE B:STUREC
COPY TO B:STUTEM FOR G:SEM = ' ' 
USE B:STUTEM
DO B:STASUB1
USE
USE B:STUREC
COPY TO B:GRADU FOR G:SEM # ' ' 
DO B:STASUB3
USE
USE B:FINCE
DO B:STASUB2
USE
WAIT
SET PRINT OFF
ENDIF

IF NUM='8'
USE B:CORSCH
STORE 1 TO X
COUNT TO LAST
GOTO TOP
? ' READY TO PRINT ? ( SET TO TOP OF PAGE )
WAIT
SET PRINT ON
DO WHILE X <= LAST
STORE 1 TO REC
? ' COURSE SCHEDULE
? ' SEMESTER : '+SEMTR,YEAR
? ' Hr. Rm. Time ';
? ' Instr ';
DO WHILE REC <= 20
DISP OFF ' CNO, $(CTIT,1,15),CHUR,RM,TIME,DAY;
TRIM(CINSTR)
STORE X+1 TO X
STORE REC+1 TO REC
IF X > LAST
  SET PRINT OFF
?
? '    ***** EOF    *****  ',
STORE 100 TO REC
WAIT
ENDIF
ENDDO
ENDDO
ENDIF
IF NUM='9'
USE B:CORLIS
STORE ' ' TO ANS
DO WHILE ANS # 'N'
ERASE
STORE ' ' TO KEY
DO B:SEMTYR
@ 13,0 SAY ' *** ENTER THE COURSE YOU WANT ';
GET KEY PICT 'XXX-XXXX'
READ
LOCATE FOR CNO=KEY .AND. SEMTR=TSEM .AND. YR=TYR
IF CNO=KEY .AND. SEMTR=TSEM .AND. YR=TYR
USE B:CORLIS
COPY TO B:KEY FOR CNO=KEY .AND. SEMTR=TSEM .AND. YR=TYR
USE
USE B:KEY
? 'READY TO PRINT ? (SET TO TOP OF PAGE ) -- ';
WAIT
SET PRINT ON
DISP OFF '    ***** CALSS LISTING    *****;
DISP OFF ' SEMESTER :,SEMTR,YR ,'; COURSE :,CNO
? '----------------------------------------'
? ' STUDENT S.S.NO     NAME     GRADE     POINT'
? '----------------------------------------'
LIST OFF ALL SSSNO, ' +SNAME, GRADE, ' ',POINT
?
COUNT TO TOT
DISP OFF '* TOTAL ='+STR(TOT,3)
?
ELSE
? ' *** NOT IN FILE, CHECK YOUR INPUT *** ';
ENDIF
ACCEPT '* WANT ANOTHER COURSE CLASS LISTING (Y/N)?' TO ANS
STORE !(ANS) TO ANS
ENDDO
SET PRINT OFF
ENDIF ********** =191 CALSS LISTING FILE
IF NUM='A'
USE B:FISCMG
STORE 1 TO X
COUNT TO LAST
? " ** READY TO PRINT ? (SET TO TOP OF PAGE) "
WAIT
SET PRINT ON
DO WHILE X <= LAST
STORE 1 TO REC
? " ** OBJECT BUDGET FILE **"?
? '---------------';
'---------------'
DO WHILE REC <= 4
GOTO X
DISP OFF 'OBJECT BUDGET CODE : '+B:CODE
DISP OFF 'GRANT TITLE : '+G:TIT+' NUMBER : '+G:NO
DISP OFF 'SOURCE ':,FD:SOUCE, ' PROJECT DIRECTOR : ';+PRODIR
DISP OFF 'AU HG# : ',AUHG, ' AMOUNT : $';+STR(AMT,8,2)
? '---------------';
'---------------'
STORE X+1 TO X
STORE REC+1 TO REC
IF X > LAST
STORE 100 TO REC
SET PRINT OFF
?
? ; ********** EOF **********
WAIT
ENDIF
ENDDO
ENDDO
ENDIF ********** ="A" OBJECT BUDGET FILE
IF NUM='B'
USE B:SUBPRO
STORE 1 TO X
COUNT TO LAST
GOTO TOP
GOTO TOP
SET PRINT ON
DO WHILE X<= LAST
DISP OFF ' *** SUB-OBJECT BUDGET FILE ***';
'-------------'
? ' SUB-OBJECT CODE BUDGET TITLE ';
'-------------'
STORE 1 TO REC
DO WHILE REC <= 40
GOTO X
DISP OFF ' ,$(S:CODE,1,15),', 'S:TIT, STR(S:AMT,9,2)
STORE X+1 TO X
STORE REC+1 TO REC
IF X > LAST
  STORE 100 TO REC
  SET PRINT OFF
ENDIF
? '********** EOF **********'
WAIT
ENDDO
ENDDO
ENDIF
IF NUM='C'
USE B:TRANSFIL
IF EOF
  RETURN
ENDIF
STORE 1 TO X
COUNT TO LAST
GOTO TOP
? '*** READY TO PRINT ? (SET TO TOP OF PAGE) ---'
WAIT
SET PRINT ON
DO WHILE X <= LAST
  '*** TRANSACTION LISTING ***'
? '----------------------------------';
'-------------'
STORE 1 TO REC
DO WHILE REC <= 9
  GOTO X
  DISP OFF ' Check No. '+CHKNO,' Amount :$', AMT;
    '+TIME
  DISP OFF ' To / From :'+TO:WHO
  DISP OFF ' Memo :'+PURPOSE
  IF $(PURPOSE1,1,10) # '
    DISP OFF ' PURPOSE1
ENDIF
? '----------------------------------';
'-------------'
STORE X+1 TO X
STORE REC+1 TO REC
IF X > LAST
  SET PRINT OFF
  STORE 100 TO REC
ENDIF
? '********** EOF **********'
WAIT
ENDIF
ENDDO
ENDDO
ENDIF
SET PRINT OFF
ENDDO **************************** LAST ENDDO
NOTE FIS.PRG
* THIS PROGRAM IS TO SET UP BUDGET FILE AND MAKE TRANSACTIONS
ERASE
SET TALK OFF
DO WHILE T
  ERASE
  @ 4,5 SAY 'COMMAND NUMBER'
  @ 6,7 SAY '0 - MAIN MENU'
  @ 7,7 SAY '1 - ENTER INFORMATION'
  @ 8,7 SAY '2 - SEE FILE CONTENTS'
  @ 9,7 SAY '3 - ADD / SUBTRACT TRANSACTION'
  @ 10,7 SAY '4 - TRANSACTION RECORDS'
STORE ' ' TO ANSWER
  @ 13,5 SAY 'ENTER A NUMBER' GET ANSWER
READ
  IF ANSWER = '0'
    RETURN
  ENDIF
  IF ANSWER = '2'
    DO B:FISEE
  ENDIF
  IF ANSWER = '3'
    DO B:FIS3
  ENDIF
  IF ANSWER = '4'
    USE B:TRANSFILE
    ? ' CHK # ISSUE DATE PAY-TO '
    ' AMOUNT'
    ? ' -------------------------------'
    ' ' DISP OFF ALL ' NO.'+TRIM(CHKNO),'' ' TIME;'' ' TO:WHO, AMT
    ?
    WAIT
  ENDIF
  IF ANSWER = '1'
    STORE ' ' TO A
    USE B:FISCMG
    APPEND BLANK
    DO WHILE A <> 'Q'
      ERASE
      @ 3,3 SAY ' **** MAKE SURE YOU INPUT IS CORRECT ****
      @ 7,3 SAY 'GRANT TITLE' GET G:TIT
      @ 7,44 SAY 'GRANT NO' GET G:NO
      @ 9,3 SAY 'BUDGET CODE' GET B:CODE
      @ 11,3 SAY 'FOUNDING SOURCE' GET FD:SOUCE
      @ 12,3 SAY 'PROJECT DIRECTOR' GET PRODIR
      @ 14,3 SAY 'AU HG NO' GET AUHG
      @ 14,30 SAY 'AMOUNT' GET AMT
      READ
STORE ' ' TO TW
@ 16,3 SAY '*** IS ALL INFORMATION CORRECT?(Y/N)' GET TW
READ
IF !(TW) = 'N'
  LOOP
ENDIF
STORE ' ' TO TW
DO WHILE TW = ' ' 
@ 16,2 SAY ' * DOES THIS OBJECT HAVE SUB-OBJECT?'; 
  '(Y/N)' GET TW
READ
ENDDO
IF !(TW) = 'Y'
  *** OPEN SUB-OBJECT FILE ***
  STORE B:CODE TO TNO
  USE B:SUBPRO
  APPEND BLANK
  STORE ' ' TO TS
  DO WHILE TS <> 'Q'
    REPLACE SNO WITH TNO
    @ 18,5 SAY ' OBJECT CODE ' + SNO
    @ 19,5 SAY ' SUB-OBJECT CODE ' GET S:CODE
    @ 20,5 SAY ' BUDGET TITLE ' GET S:TIT
    @ 21,5 SAY ' AMOUNT ' GET S:AMT
    READ
    @ 23,5 SAY ' * MORE SUB-OBJECT BUDGET?(Y/N)' GET TS
    READ
    IF !(TS) = 'N'
      STORE 'Q' TO TS
    ELSE
      APPEND BLANK
    ENDIF
  ENDDO
ENDIF
ERASE
*
@ 5,5 SAY '****** BACK TO MAIN OBJECT PROCEDURE ****
@ 8,5 SAY ' * DO YOU WANT ANOTHER OBJECT BUDGET';
  'RECORD?(Y/N)' GET A
READ
STORE !(A) TO A
IF A = 'N'
  ** BACK TO MENU ****
  STORE 'Q' TO A
ELSE
  USE
  USE B:FISCMG
  APPEND BLANK
ENDIF
ENDDO
ENDIF
ENDDO
NOTE FIS3.PRG
* MAKE TRANSACTIONS TO A BUDGET ACCOUNT *
* SET TALK OFF
ERASE
STORE " " TO FINDM
STORE " " TO FINDS
DO WHILE T
*
@ 2,3 SAY "******** TO ADD / SUBTRACT ********"
STORE " " TO KEY
@ 5,3 SAY "* ENTER BUDGET ACCOUNT CODE YOU';
'WANT( 0 TO EXIT) ' GET KEY
READ
STORE I(TRIM(KEY)) TO KEY
IF KEY='0'
  RETURN
ENDIF
USE B:FISCMG
LOCATE FOR I(TRIM(B:CODE))=KEY
IF I(TRIM(B:CODE))=KEY
  STORE 'Y' TO FINDM
  STORE # TO MNUM
@ 8,3 SAY "** FIND IN OBJECT BUDGET FILE ***
  STORE 1 TO X1
  DO WHILE X1 < 20
    STORE X1+1 TO X1
  ENDDO
  RELEASE X1
ELSE
USE
USE B:SUBPRO
LOCATE FOR I(TRIM(S:CODE))=KEY
IF I(TRIM(S:CODE))=KEY
  STORE 'Y' TO FINDS
@ 8,3 SAY "** FIND IN SUB BUDGET FILE ***
  STORE # TO SNUM
  STORE SNO TO TKEY
USE
USE B:FISCMG
LOCATE FOR I(TRIM(B:CODE)) = TRIM(TKEY)
IF I(TRIM(B:CODE))=TRIM(TKEY)
  STORE # TO MNUM
ENDIF
ELSE
  @ 8,3 SAY "*** CAN NOT FIND, NO THIS BUDGET CODE ***
  STORE 1 TO ZO
  DO WHILE ZO < 20
    STORE ZO+1 TO ZO
  ENDDO
LOOP
ENDIF
ENDIF
ERASE
USE B:TRANSFILE
STORE ' ' TO TRANS
STORE 0 TO T:OUT
APPEND BLANK
DO WHILE TRANS <> 'Q'
ERASE
@ 3,4 SAY '*** ENTER TRANSACTION AMOUNT ***'
REPLACE CODE WITH KEY
@ 5,0 SAY 'Budget code: ' +KEY
@ 7,0 SAY 'Check No. ' GET CHKNO
@ 7,30 SAY 'Pay to or Receive from ' GET TO:WHO
@ 9,0 SAY 'Check Amount ' GET AMT
@ 9,40 SAY 'Date ' GET TIME PICT '#/#/#/84'
@ 10,4 SAY '(Note: 20.00 or -20.00)'
@ 11,0 SAY 'What use ' GET PURPOSE
@ 12,10 GET PURPOSE1
READ
STORE AMT TO D:OUT
IF FINDS = 'Y'
USE
USE B:SUBPRO
GOTO SNUM
STORE S:AMT TO T:AMT
@ 12,0 SAY 'SUB OBJECT CODE ' GET S:CODE
@ 12,40 SAY 'BUDGET TITLE ' GET S:TIT
@ 13,0 SAY 'AMOUNT ' GET S:AMT
REPLACE S:AMT WITH T:AMT+D:OUT
WAIT
@ 13,0 SAY 'AMOUNT ' GET S:AMT
USE
ENDIF
IF FINDS = 'Y' .OR. FINDM = 'Y'
USE B:FISCMG
GOTO MNUM
STORE AMT TO T:AMT
@ 15,0 SAY 'OBJECT BUDGET ' GET B:CODE
@ 15,40 SAY 'GRANT NO ' GET G:NO
@ 16,0 SAY 'DIRECTOR ' GET PRODIR
@ 16,40 SAY 'AMOUNT ' GET AMT
REPLACE AMT WITH T:AMT+D:OUT
WAIT
@ 16,40 SAY 'AMOUNT' GET AMT
USE
ENDIF
?
ACCEPT ' * MORE CHECKS FOR THIS ACCOUNT?(Y/N) ' TO ANS
STORE 1(ANS) TO ANS
IF ANS='N'
    STORE 'Q' TO TRANS
ELSE
    USE B:TRANSFILE
    APPEND BLANK
ENDIF
ENDDO

STORE ' ' TO ANS
ACCEPT ' * DO YOU WANT ANOTHER ACCOUNT?(Y/N) ' TO ANS
IF 1(ANS)='N'
    RETURN
ELSE
    USE
    USE B:TRANSFILE
    APPEND BLANK
ENDIF
ERASE
ENDDO
NOTE DELECOUR.PRG
* THIS PROGRAM IS TO DELETE A COURSE *
SET TALK OFF
ERASE
USE B: CORSCH
DO WHILE T
ERASE
STORE ' ' TO TEMPCOURSE
@ 4,5 SAY '* ENTER COURSE WANT TO DELETE *'
@ 6,5 SAY '* COURSE NUMBER ( MCS-000, 0 EXIT )';
GET TEMPCOURSE PICT 'XXX-XXXX'
READ
IF TEMPCOURSE='O' -
   RETURN
ENDIF
LOCATE FOR CNO=TEMPCOURSE
IF CNO=TEMPCOURSE
GOTO #
?
ACCEPT ' > COURSE FIND IN FILE, REALLY WANT';
'TO DELETE?(Y/N)' TO ANSW
STORE I(ANSW) TO ANSW
IF ANSW='Y'
DELETE
PACK
ENDIF
ELSE
?
'***** NOT IN FILE, CHECK YOUR INPUT ******'
STORE 1 TO X
DO WHILE X < 20
   STORE X+1 TO X
ENDDO
LOOP
ENDIF
ERASE
ENDDO
* COUR.PRG This program is to set up a course schedule, class listing, and student grade recording

SET TALK OFF
DO WHILE T
  ERASE
  @ 5,5 SAY ' Command number '
  @ 7,8 SAY ' 0 - Main menu '
  @ 8,8 SAY ' 1 - Course scheduling '
  @ 9,8 SAY ' 2 - Add/Drop, class listing, grade updating '
  @ 10,8 SAY ' 3 - Delete course from course schedule'
  STORE ' ' TO NOIN
  @ 13,5 SAY '* Enter a number ---' GET NOIN
  READ
  IF NOIN='0'
    RETURN
  ENDIF
  IF NOIN='2'
    DO B:CORLST
  ENDIF
  IF NOIN='3'
    DO B:DELELCOURSE
  ENDIF
  IF NOIN='1'
    ERASE
    @ 3,3 SAY ' 1 - Enter course data '
    @ 5,3 SAY ' 2 - Schedule list '
    STORE ' ' TO ANS
    @ 7,2 SAY '* Enter a number ' GET ANS
    READ
    IF ANS = '2'
      USE B:CORSCH
      DISP OFF '\', 'SEMTR+', ',YEAR
      ? '-----------------------------------------';
      '-----------------------------------------'
      STORE 1 TO RECNO
      COUNT TO X
      STORE ' ' TO CNT
      DO WHILE RECN <= X .AND. CNT <= 'N'
      DISP OFF '\', 'CNO', ' ', 'CHUR', ' ', 'TIME', ' ', 'DAY,' 'RM', ',', 'CINSTR
      STORE RECN+1 TO RECN
      IF RECN > X
        STORE 'N' TO CNT
      ELSE
        GOTO RECN
      ENDIF
      ? '-----------------------------------------';
      '-----------------------------------------'
    ENDDO
WAIT
ENDIF

IF ANS='1'
ERASE
USE CORSCH
STORE ' ' TO ANSO
@ 5,5 SAY "ENTER <Y> TO CLEAR DATA FILE';
    ' FOR A NEW SCHEDULE ' GET ANSO
READ
IF I(ANSO)='Y'
    DELETE ALL
PACK
ENDIF
STORE ' 1 - FALL, 2 - SPRING, 3';
    ' SUMMER ' TO P
ERASE
STORE ' ' TO SEM
STORE ' ' TO YR
@ 5,0 SAY P
@ 7,2 SAY 'ENTER SEMESTER (1,2,3)' GET SEM
@ 8,2 SAY 'YEAR' GET YR
READ
IF SEM='1'
    STORE 'FALL' TO SEMT
ENDIF
IF SEM='2'
    STORE 'SPRING' TO SEMT
ENDIF
IF SEM='3'
    STORE 'SUMMER' TO SEM
ENDIF
STORE ' ' TO CONW
DO WHILE CONW # 'N'
    ERASE
APPEND BLANK
REPLACE SEMTR WITH SEMT,YEAR WITH YR
@ 10,2 SAY ' 'SEM'T
@ 10,15 SAY YR
@ 11,2 SAY 'COURSE NO ' GET CNO PICT 'XXX-XXXX'
@ 11,35 SAY 'HOUR ' GET CHUR
@ 13,2 SAY 'TIME ' GET TIME
@ 13,30 SAY 'DAY ' GET DAY
@ 13,45 SAY 'ROOM ' GET RM
@ 15,2 SAY 'INSTRUCTOR ' GET CINSTR
READ
@ 17,2 SAY '* ANOTHER COURSE TO INPUT ?(Y/N)' GET CONW
READ
ENDDO
ENDIF
ENDIF
ENDDO
* THIS PROGRAM IS FOR ENTERRING STUDENTS GRADE
SET TALK OFF
*
STORE ' ' TO COLIST
DO B:SEMTYR
DO WHILE COLIST <> 'N'
ERASE
USE B:CORLIS
@ 5,5 SAY '* *** ENTER GRADE *** '*
COUNT TO X
GOTO TOP
STORE ' ' TO TCO
@ 7,5 SAY '* ENTER COURSE NO. (O EXIT) :';
GET TCO PICT 'XXX-XXXX'
READ
IF TCO='O -'
RETURN
ENDIF
STORE 0 TO J
DO WHILE J < X
ERASE
STORE J+1 TO J
GOTO J
IF CNO=TCO .AND. SEMTR=SEMTR .AND. YR=TYR
@ 8,5 SAY '---- ENTER GRADE ';
@ 11,8 SAY 'SSNO : ' + SSSNO
@ 11,40 SAY 'NAME : ' + SNAME
@ 13,8 SAY 'COURSE : ' + CNAME
@ 15,8 SAY 'GRADE : ' GET GRADE
READ
DO CASE
CASE GRADE = 'A'
STORE 4.0 TO P
CASE GRADE = 'B'
STORE 3.0 TO P
CASE GRADE = 'C'
STORE 2.0 TO P
CASE GRADE = 'D'
STORE 1.0 TO P
CASE GRADE = 'F'
STORE 0.0 TO P
OTHERWISE
STORE 0.0 TO P
ENDCASE
REPLACE POINT WITH P
ENDIF
IF J=X
? ' * END OF FILE *** '*
ENDIF
EDDO
?  
ACCEPT '* MORE TO INPUT? (Y/N) ' TO COLIST  
STORE !(COLIST) TO COLIST  
ENDDO
* STAFIN.PRG STATISTICAL SUMMARY *

SET TALK OFF
COUNT TO LAST
GOTO TOP
STORE ' ' TO CONT
DO WHILE CONT <> 'Q'
  INPUT ' * ENTER RECORD NUMBER WANT TO EDIT' ; 
  or DELETE ' TO NUM
  IF NUM > LAST
    ? ' *** RECORD OUT OF RANGE ***'
    ACCEPT ' *** WANT CONTINUE ? (Y/N) ' TO ACC
    IF ACC = 'N'
      RETURN
    ELSE
      LOOP
   .ENDIF
  ENDIF
  GOTO NUM
  ACCEPT ' * EDIT or DELETE, ( E or D ) ' TO CHAR
  ? CHR(7)
  IF CHAR = 'D'
    ACCEPT ' * ARE YOU SURE (Y/N) ' TO SURE
    IF ACC = 'Y'
      DELETE
      ? ' ** DO NOT INTERRUPT ! **'
      PACK
   .ENDIF
  ENDIF
  IF CHAR = 'E'
    ERASE
    @ 5,5 SAY ' * EDIT A RECORD *** '
    @ 7,5 SAY 'NAME ' GET FNAME
    @ 7,40 SAY 'SSNO ' GET FSSNO
    @ 9,5 SAY 'TYPE ' GET TYPE
    @ 10,5 SAY 'GRANT TITLE ' GET GRANT: NO
    @ 11,5 SAY 'BUDGET NO ' GET BUDGT: NO
    @ 13,5 SAY 'AMOUNT ' GET AMOUNT
    @ 15,5 SAY 'SEMESTER ' GET SEMTR
    @ 15,25 SAY 'YEAR ' GET YR
    READ
 .ENDIF
  ACCEPT ' * DO YOU WANT ANOTHER RECORD? (Y/N) ' TO MORE
  ? CHR(7)
  IF MORE = 'N'
    STORE 'Q' TO CONT
  ELSE
    LOOP
 .ENDIF
ENDDO
NOTE - STUPRT.PRG THIS PROGRAM IS TO PRINT STUDENT
* TRANSCRIPT
*
DO WHILE T
ERASE
STORE ' ' TO T:SSNO
@ 5,5 SAY ' ENTER STUDENT ID NO, 0 TO EXIT ';
GET T:SSNO PICT '###-##-####'
READ
IF $(T:SSNO,1,1)='0' .AND. $(T:SSNO,2,10)=' ' RETURN
ENDIF
IF LEN(TRIM(T:SSNO)) < 11
?
? ' ********** BAD INPUT, INPUT AGAIN **********
STORE 1 TO B
DO WHILE B < 20
STORE B+1 TO B
ENDDO
LOOP
ENDIF
********** ID INPUT IS RIGHT **********
STORE ' ' TO T:NAME
@ 7,5 SAY ' STUDENT NAME ' GET T:NAME
READ
USE B:STUREC
LOCATE FOR SSNO=T:SSNO
IF SSNO=T:SSNO
?
* STUDENT IS IN FILE *
ELSE
?
** STUDENT NOT IN FILE, CHECK YOUR INPUT **
STORE 1 TO X
DO WHILE X < 20
STORE X+1 TO X
ENDDO
loop
endif
USE B:CORLIS
STORE ' 1 - CURRENT SEMESTER 2 - WHOLE RECORDS' TO L
@ 4,2 SAY ' ' + L
STORE ' ' TO SEC
@ 8,3 SAY ' ENTER A NUMBER' GET SEC
READ
IF SEC= '1'
STORE ' 1 - FALL, 2 - SPRING, 3 - SUMMER ' TO L1
@ 6, 2 SAY ' ' + L1
STORE ' ' TO SE
STORE ' ' TO YEAR
STORE ' ' TO SE
STORE ' ' TO YEAR
@ 9,3 SAY ' ' GET YEAR
@ 9,20 SAY ' ' SEMESTER CODE ' GET SE
READ
STORE ' ' TO SEM
IF SE = '1'
    STORE 'FALL ' TO SEM
ENDIF
IF SE = '2'
    STORE 'SPRING ' TO SEM
ENDIF
IF SE = '3'
    STORE 'SUMMER ' TO SEM
ENDIF
COUNT TO X FOR !(TRIM(SEMTR)) = TRIM(SEM);
    .AND. YR = YEAR .AND. SSSNO = T:SSNO;
    .AND. I(GRADE) <> 'I' .AND. I(GRADE) <> 'W';
    .AND. GRADE <> ''
SUM POINT TO SUMT FOR !(TRIM(SEMTR)) = TRIM(SEM);
    .AND. YR = YEAR .AND. SSSNO = T:SSNO
STORE SUMT/X TO GPA
ENDIF
IF SEC = '2'
    COUNT TO X FOR SSSNO = T:SSNO .AND. I(GRADE) # 'I';
    .AND. I(GRADE) # 'W' .AND. I(GRADE) # '
SUM POINT TO SUMT FOR SSSNO = T:SSNO
STORE SUMT/X TO GPA
ENDIF
STORE 'YR = YEAR .AND. SEMTR = SEM' TO STING
SET PRINT ON
?
? ' SEMESTER S.S.NO COURSE GRADE'
? ' =================================================='
? IF SEC = '1'
DISP ', SEMTR, YR, SSSNO, ' ', GRADE FOR SSSNO = T:SSNO;
    .AND. &STING OFF
ENDIF
IF SEC = '2'
DISP ', SEMTR, YR, SSSNO, ' ', GRADE FOR SSSNO = T:SSNO OFF
ENDIF
?
DISP OFF ' *------ GPA = ', STR(GPA, 4, 2)
SET PRINT OFF
?
ACCEPT ' * WANT ANOTHER RECORD (Y/N) ? ' TO SEE
STORE ! (SEE) TO SEE
ENDDO
<table>
<thead>
<tr>
<th>S.S.NO.</th>
<th>Name</th>
<th>TEL.</th>
<th>Address</th>
<th>Birth date</th>
<th>Sex</th>
<th>Nationality</th>
<th>Major</th>
<th>Advisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>000-00-0001</td>
<td>CHEN, LIEN-BON</td>
<td>892-9484</td>
<td>244 14TH ST. ATLANTA GA</td>
<td>01-10-54</td>
<td>MA</td>
<td>FO</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>000-00-1222</td>
<td>CHIENGCHAICHAN, LADDAPORN</td>
<td>892-0484</td>
<td>244 14TH ST, NE ATLANTA GA 30309</td>
<td>09-10-51</td>
<td>FE</td>
<td>FO</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>234-56-0987</td>
<td>J.C. Jackson</td>
<td>998-2766</td>
<td>1000, Acon st. Atlanta GA 30336</td>
<td>09-02-50</td>
<td>M</td>
<td>D</td>
<td>MATH</td>
<td></td>
</tr>
<tr>
<td>252-23-7872</td>
<td>BONNER, PHYLLIS L.</td>
<td>758-4491</td>
<td>1991 DELOWE DR #F-6 ATLANTA GA 30311</td>
<td>09-26-60</td>
<td>FE</td>
<td>FO</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>254-35-5078</td>
<td>JOSEPH WAN</td>
<td>352-0752</td>
<td>2234 Bolton Dr., #3</td>
<td>10-21-1953</td>
<td>M</td>
<td>F</td>
<td>CS</td>
<td>Dr. Martin</td>
</tr>
<tr>
<td>255-80-5443</td>
<td>BROWN, FRANCES L.</td>
<td>691-5506</td>
<td>1256 BROOKSIDE COURT MABLETON GA</td>
<td>01-16-52</td>
<td>FE</td>
<td>DO</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CS</th>
<th>MATH</th>
<th>APM</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>S.S.NO.</td>
<td>Name</td>
<td>TEL.</td>
<td>Address</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------</td>
<td>--------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>491-74-5077</td>
<td>WALLACE, SHARON M.</td>
<td>874-6899</td>
<td>960 GREENWOOD AVENUE APT. 5</td>
</tr>
<tr>
<td>254-35-5078</td>
<td>JOSEPH Y. WAN</td>
<td>352-0752</td>
<td>2234, BOLTON DR. #3</td>
</tr>
</tbody>
</table>

**COMPUTER SCI.** 2  
**MATHEMATICS** 0  
**APPLIED MATH.** 0  
**TOTAL :** 2
*** LISTING OF TEXTBOOKS ***

================================================================================
BOOK: DATA BASE ORGANIZATION
COURSE USING: MCS-570    SEMESTER USING : 1984/FALL
PUBLISHER :
SEMESTER USING : 1984/FALL

================================================================================
BOOK: PROGRAMMING LANGUAGE DESIGN
COURSE USING: MCS-550    SEMESTER USING : 1984/FALL
PUBLISHER :
SEMESTER USING : 1984/FALL

================================================================================
<table>
<thead>
<tr>
<th>NAME</th>
<th>S.S.NO.:</th>
<th>ADDRESS</th>
<th>TITLE</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARTIN, BENJAMIN J.</td>
<td>258-66-0650</td>
<td>337 LYNHURST DR.</td>
<td>CHAIRMAN, PROF.</td>
<td>691-1383</td>
</tr>
<tr>
<td>WARSI, NAZIR A.</td>
<td>111-11-1111</td>
<td>1234 NELBY DRIVE</td>
<td>PROFESSOR</td>
<td>555-1213</td>
</tr>
<tr>
<td>Mr. John Kao</td>
<td>254-38-9056</td>
<td>2022, Bolton Dr., #5</td>
<td>Ass. Professor</td>
<td>352-3738</td>
</tr>
</tbody>
</table>
** OBJECT BUDGET FILE **

<table>
<thead>
<tr>
<th>OBJECT BUDGET CODE</th>
<th>GRANT TITLE</th>
<th>SOURCE</th>
<th>PROJECT DIRECTOR</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12-472-101-000</td>
<td>dynamic programming</td>
<td>NASA</td>
<td>WARSİ</td>
<td>$26577.00</td>
</tr>
<tr>
<td>1-12-472-202-000</td>
<td>STELLAR MOTIONS</td>
<td>NASA</td>
<td>MARTINAU</td>
<td>$44800.00</td>
</tr>
<tr>
<td>T-100</td>
<td>BUSINESS ICC</td>
<td>TV-CABLE</td>
<td>TOM</td>
<td>$19699.30</td>
</tr>
</tbody>
</table>
### SUB-OBJECT BUDGET FILE

<table>
<thead>
<tr>
<th>SUB-OBJECT CODE</th>
<th>BUDGET TITLE</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>DYNAMIC PROGRAMMING</td>
<td>0.00</td>
</tr>
<tr>
<td>706</td>
<td>DYNAMIC PROGRAMMING</td>
<td>20.00</td>
</tr>
<tr>
<td>304</td>
<td>DYNAMIC PROGRAMMING</td>
<td>15140.00</td>
</tr>
<tr>
<td>404</td>
<td>STELLAR MOTIONS</td>
<td>456.00</td>
</tr>
<tr>
<td>906</td>
<td>STELLAR MOTIONS</td>
<td>350.00</td>
</tr>
<tr>
<td>896</td>
<td>STELLAR MOTIONS</td>
<td>345.00</td>
</tr>
<tr>
<td>T-100-01</td>
<td>AUDITOR EXPENSE</td>
<td>2999.30</td>
</tr>
<tr>
<td>T-100-02</td>
<td>FIELD GAS</td>
<td>700.00</td>
</tr>
<tr>
<td>Check No.</td>
<td>Amount</td>
<td>Date</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>101</td>
<td>-100.00</td>
<td>09/18/84</td>
</tr>
<tr>
<td></td>
<td>To / From: Book store</td>
<td>Memo: expense</td>
</tr>
<tr>
<td>102</td>
<td>2000.00</td>
<td>02/02/84</td>
</tr>
<tr>
<td></td>
<td>To / From: NASA</td>
<td>Memo: Research</td>
</tr>
<tr>
<td>105</td>
<td>-5000.00</td>
<td>11/14/84</td>
</tr>
<tr>
<td></td>
<td>To / From: Dr. Martin</td>
<td>Memo: Research expense</td>
</tr>
<tr>
<td>107</td>
<td>-200.00</td>
<td>11/09/84</td>
</tr>
<tr>
<td></td>
<td>To / From: Dr. Warsi</td>
<td>Memo: Office supply</td>
</tr>
<tr>
<td>204</td>
<td>500.00</td>
<td>09/27/84</td>
</tr>
<tr>
<td></td>
<td>To / From: JARY</td>
<td>Memo: PAYROLL</td>
</tr>
<tr>
<td>205</td>
<td>-500.70</td>
<td>09/28/84</td>
</tr>
<tr>
<td></td>
<td>To / From: JOE</td>
<td>Memo: PAYROLL</td>
</tr>
<tr>
<td>201</td>
<td>-300.00</td>
<td>11/02/84</td>
</tr>
<tr>
<td></td>
<td>To / From: Mr. bk</td>
<td>Memo: Power</td>
</tr>
</tbody>
</table>