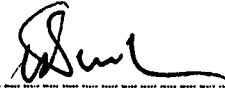


**ESTABLISHMENT OF A DATABASE
MANAGEMENT SYSTEM FOR BILKENT UNIVERSITY
DORMITORY MANAGEMENT**

**A THESIS
SUBMITTED TO THE DEPARTMENT OF MANAGEMENT
AND THE INSTITUTE OF BUSINESS ADMINISTRATION
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF BUSINESS ADMINISTRATION**

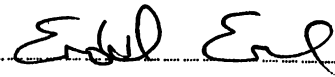
**BY
KAHRAMAN GÜNAYDIN
SEPTEMBER, 1990**

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.



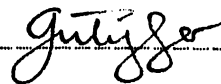
Prof. Dr. Umit Berkman

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.



Assist. Prof. Erdal Erel

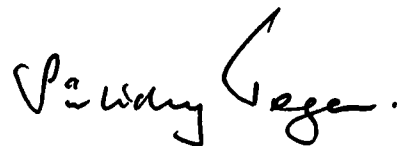
I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.



Assist. Prof. Guliz Ger

Approved for the Graduate School of Business Administration.

Prof. Dr. Subidey Togan



ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. Dr. Umit Berkman for his patient supervision.

I appreciate the humanly guidance, continued encouragement and suggestions that Assist. Prof. Erdal Erel And Assist Prof. Guliz Ger provided me throughout this thesis study.

ABSTRACT

ESTABLISHMENT OF A DBMS FOR BILKENT UNIVERSITY DORMITORY MANAGEMENT

By

Kahraman Gunaydin
Supervisor: Prof. Dr. Umit Berkman

The purpose of this thesis is to establish a Database Management System for Bilkent University Dormitory management. The aim of the program is to use information more easier and quicker than the present manual system. Activities such as, entering editing, deleting and retrieving data with this program are more quicker and easier. Sorting through large pieces of information is available and more efficient with this program.

Key words: Database Management System, Bilkent University Dormitory management, Information, Manual System.

ÖZET

BİLKENT ÜNİVERSİTESİ YURT YÖNETİMİ İÇİN BİR VERİTABANI İŞLETİM SİSTEMİ OLUSTURULMASI

Kahraman Günaydın
Tez Yöneticisi: Prof. Dr. Umit BERKMAN

Bu çalışma, Bilkent Üniversitesi yurt yönetimine yeni bir veritabanı işletim sistemi inşa etmek amacıyla yapıldı. Buradaki hedef bilgilerin daha kolay ve daha cabuk kullanılmasıydı. Daha önceki elle dosyalama sistemine nazaran, bu yeni oluşturulan programla, veri girme, değiştirme, silme ve geri çağırma daha cabuk ve kolay yapılır hale geldi. Östelik bu programla verilerin sıralandırılması ve ayıklanması daha hızlı ve güvenilir oldu.

Anahtar kelimeler: Veritabanı İşletim Sistemi, Bilkent Üniversitesi Yurt Yönetimi, Bilgi, Elle Dosyalama Sistemi.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	ii
ABSTRACT	iii
OZET	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	ix
LIST OF TABLES	ix
1. INTRODUCTION	1
1.1 INTRODUCTION TO THE SUBJECT	1
1.2 PROBLEM DEFINITION	1
1.3 METHODOLOGY	3
1.4 OUTLINE OF THE THESIS	4
2. LITERATURE SURVEY ON DBMS	6
2.1 DEFINITION OF A DBMS	6
2.2 DATABASE MANAGEMENT OBJECTIVES	7
2.3 DEVELOPMENT OF DBMS	8
2.4 COMPONENTS OF A DBMS	10
2.4.1 Database	10
2.4.2 Database Definition	10
2.4.3 Data Manipulation Language	11
2.4.4 Database Procedures	11
2.5 TYPES OF DATABASES	12
2.6 SOME FUNDAMENTAL CONCEPTS	12
2.6.1 File	12
2.6.2 Record	13

2.6.3 Field	13
2.6.4 Index File	13
2.6.5 Key Field	13
2.6.6 Normalization	14
3. HOW TO USE THE PROGRAM	15
3.1 INTRODUCTION	15
3.2 THE MENUS	15
3.2.1 Main Menu (Anatablo)	15
3.2.2 Data Entry (Veri Girme)	18
3.2.2.1 Alternative 1	18
3.2.2.2 Alternative 2	18
3.2.2.3 Alternative 3	19
3.2.2.4 Alternative 4	19
3.2.2.5 Alternative 5	22
3.2.3 Changing Data (Veri Degistirme)	24
3.2.4 Deleting Data (Veri Silme)	24
3.2.5 Retrieving Data (Veri Tarama)	24
3.2.5.1 Alternatives Of Retrieving Data (Tarama Secenekleri)	25
3.2.6 Some Statistics (Bazi Istatistikler)	29
4. CONCLUSION	30
REFERENCES	32
APPENDICES	
A. Internal Structure Of The Program	33
B. Anatablo.prg Module	41
C. Vergir.prg Module	43
D. Verdeg.prg Module	46

E. Versil.prg Module	50
F. Vertar.prg Module	59
G. Istatistik.prg`Module	68
H. Application Example	71
I. Turkish Version of Chapter 3 (Kullanım kilavuzu)	85

LIST OF FIGURES

FIGURE	Page
1. Network of Main Menu	16
2. Network of Retrieving Data Menu	17

LIST OF TABLES

TABLE	Page
1. Structure of The Yurt.dbf File	19
2. Structure of The Kimlik.dbf File	20
3. Structure of The Nufus.dbf File	21
4. Structure of The Egitim.dbf File	22
5. Structure of The Aile.dbf File	23
6. Structure of The Yurt1b1.1b1 File	26
7. Structure of The Kimlik1b1.1b1 File	26
8. Structure of The Nufus1b1.1b1 File	26
9. Structure of The Egitim1b1.1b1 File	27
10. Structure of The Aile1b1.1b1 File	27

CHAPTER 1 - INTRODUCTION

1.1 INTRODUCTION TO THE SUBJECT

The purpose of this thesis is to prepare a Database Management system (DBMS) which will manage the data of Bilkent's student dormitories.

According to Sharman (1987) " A database is a large organized store of information which is available when you need it. The purpose of a database is to capture information and make it available. A database system should be truly general-purpose software package which is capable of processing any logical structure and manipulating the data within it. This software is correctly known as a DBMS."

1.2 PROBLEM DEFINITION

We know that most organizations will have some form of database, either manual (paper based) system or a computerized database system.

A paper based system (which is currently used in Bilkent University Dormitories for student registration) can be organized in many ways i.e. different folders in filing cabinet, or it could be collection of cards in a box sorted in the alphabetical order. There are many disadvantages with a paper based filing system, some of the disadvantages are as follows.

- i) Records can be lost or replaced in the wrong order

by the user.

- ii) They can be very bulky, lots boxes or cabinets,
- iii) It is difficult to manipulate, display or analyse the data,
- iv) If there are many records, to retrieve them can be very time consuming.

The above are the just a few of the disadvantages of a paper based filing system. All of these problems exist in the Bilkent University Dormitories current registration system.

Any form of filing system should be able to fullfil at least two roles in particular.(5)

- i) To keep track of what has happened in the past.
- ii) To use the information to assist current activities and also be able to plan future activities.

If we look at these two points, it is the second point that lends itself to a computer based system. A paper based filing system, is incomparable to a computer database system; especially in the areas of sorting through large pieces of information, and then showing that information in a useful and meaningful format.

After several discussions with the dormitory management, it is determined that the neccessary requirements for the new DBMS are stated as follows:

1. Input of Data
2. Editing Data
3. Deleting Data
4. Retrieval of Data
 - a) According to their departments
 - b) According to their first names
 - c) According to their last names
 - d) According to their dormitory registration number
 - e) According to their entry date to the dormitory

And neccessary files that should be kept for each student are stated as follows:

1. Dormitory file (yurt.dbf)
2. ID file (kimlik.dbf)
3. Identity book file (nufus.dbf)
4. Education file (egitim.dbf)
5. Family file (aile.dbf)

1.3 METHODOLOGY

Relational database model is one of the most suitable models that we can use for the above DBMS, due to its simplicity in operation and flexibilitiy. As Database Management System software, dBASE III plus pocket program and its language will be used.

dBASE III was first introduced in 1984 and was designed for 16-bit microprocessors. The program was written in C and designed to support menus, which made its use much simpler for the non-programmer.(7)

dBASE III Plus was then introduced in late 1985. Two of the main differences between dBASE III and dBASE III Plus are that the latter has networking capabilities and a far superior menu system.

Both programs have the following system requirements:

- A minimum of 256K of memory
- Require MS-DOS or PC-DOS versions 2.00 or later

Twin 360K double sided floppy disk drives or one 360K floppy disk drive plus a hard-disk drive.

In this thesis, we will follow the following designing steps.

1. Description of the problem: This is given in the problem definition section.

2. Determining the universal relation attributes: After several discussions with manager of the dormitories, some attributes and restrictions on them were deemed to be prime candidates for the universal relation.

3. Designing normalized files.

4. Adding indexes.

1.4 OUTLINE OF THE THESIS

There are four chapters in this thesis. First chapter is the introduction. In this chapter, introduction to the subject, problem definition and the methodology is given.

The second chapter is devoted to the literature survey. In this chapter, following headings can be found: definition of a DBMS, database management objectives, development of DBMS, components of a DBMS, types of databases, and some fundamental concepts.

How to use the program is the subject of the third chapter. In this chapter, we will find explanations of how to use the programs such as, how menus come on the screen, which menus perform which jobs and which inputs are needed, which outputs are obtained.

The fourth chapter is devoted to conclusion.

CHAPTER 2 - LITERATURE SURVEY ON DBMS

2.1 DEFINITION OF A DBMS

A database may be defined as a unified collection of data that is to be shared by all authorized personnel in an enterprise.(10) The enterprise could be a company, a department in a company, a bank, or a dormitory as in this thesis. The purpose of the database is to store all of the data of interest to the enterprise in one location, so that redundant data storage within the enterprise is eliminated.

Whether the database being developed is to be stored in a mainframe computer or in a micro computer, the function of the database management system (DBMS) is the same. The DBMS is the software - hardware package that makes the database easily accesible to the users. The database manager provides the software tools required to create, load, query, and update data in the database. The database manager also handles all I/O and memory management related to the database. In short, a well designed DBMS will provide software which makes it easy for user to communicate with the database.

2.2 DATABASE MANAGEMENT OBJECTIVES

The basic goal of the database management is to allow an organization to improve its use and control of its data resources. This goal can be divided into four distinct objectives.(9) :

1. Ease of use: The first objective is to make the DBMS easy to use. If the system is hard to use, people will need more training time and will be more prone to mistakes. As a result, fewer people will be able to use the system directly.

2. Evolvability: Evolvability is, in a sense, long term ease of use. It should be easy to modify database and applications that use it to meet new or changing requirements. This is an important objective, because of the high cost of the program maintenance.

3. Data integrity: improved data integrity is the third DBMS objective.

i) Validity: involves ensuring that only acceptable data values are entered into the database.

ii) Backup and recovery: the DBMS must be able to correct and restore the database when there has been a hardware or software failure.

4. Data security: The fourth DBMS objective is improved data security. The database is a valuable organizational resource that must be protected from unauthorized access and use.

2.3 DEVELOPMENT OF DBMS

DBMS's can be classified in several ways for instance, most early DBMSs can be classified according to type of user for whom they were designed. Although this classification has more historic than current importance, it should be discussed. Because some of the problems that arose due to this classification still plague DBMS developers and users.

According to Olin H. Bray (3) , in the late 1960's a data management problem arose, first within the intelligence community, but soon within many other types of organizations as well. End users, that is, non programming users needed to be able to ask a variety of unanticipated or ad hoc questions of their database and to receive a relatively fast response. "Relatively fast" in these cases meant seconds or minutes, as opposed to days, weeks, or months it would have taken for a program to be written for each of these unanticipated queries. To meet this need, a variety of "self - contained" DBMSs were developed.

The data stored in the database were defined to the simple query/update language for the end user. The query/update language was quick and easy to learn and allowed the end user to answer most of his questions himself, without the need of a specific program. These systems were called self contained database management systems. Certain types of data could be put up on this

type of database and essentially removed from the day to day concern of the data processing department. These early self contained DBMSs were designed solely for the end user and therefore had no programming language interface for the programmer. While these self contained systems were evolving in a different data processing environment.

As application systems and their data requirements became more complex, analysts and programmers were forced to spend more time and effort on routine data management functions, rather than on applications. A second type of DBMS evolved to meet this need : the hosted DBMS. Again, the data structure was defined and a set of data management routines was developed to do most of the routine data management operations for the programmer. The data manipulation language for these systems was not self contained, but was embedded or hosted within one or more programming languages.

Information management system (IMS) and its successors evolved to solve this type of problem data management within the traditional data processing environment.

2.4 COMPONENTS OF A DBMS

2.4.1 Database

These are the data that describe the various entities, attributes, relationships of interest. An entity type refers to a specific kind of object, such as customers or students. Attributes are characteristics of these entities. Each type entity has a specific set of characteristics or attributes. A student for example, has a number, a name, an address, etc. All these attributes describe a student, but some of them (e.g. their student number) are identifiers that uniquely specify a particular student. Most entity types have only one identifier, but it may be made up of several attributes. Relationships indicate the ways in which various entity types are linked.

2.4.2 Database Definition

The second component of a DBMS is the database definition. Most DBMS's today have two level architecture for their database definition. These two levels are called the schema and the subschema. The schema provides a complete description of the entire database as it is stored. This definition includes the physical characteristics of the data, such as the format, storage location and the access paths to the data. The schema also defines the logical structure of

the database.

The subschema describes that part of the database used by a particular application. It describes the database as the application expects it, rather than as it is actually stored. The subschema is similar to the schema except that it contains only logical characteristics of the database. Since the user should be completely isolated from the physical aspects of the database, the subschema does not need to define any of these characteristics. The subschema defines entity types, their relationships, their attributes and the format in which the application expects the data.

2.4.3 Data Manipulation Language

Data manipulation language is the third component of a DBMS. It must allow the user to do four basic operations:

1. Retrieve data from database
2. Modify data that already exist in the database
3. Add new data to the database
4. Delete data that already exist in the database

2.4.4 Database Procedures

The procedures actually perform the various database management functions. Examples of these procedures include determining whether a user has authorization to access the data, physically locating

the data in the database, and mapping between the schema and the subschema.

2.5 TYPES OF DATABASES

There are three major types of databases : Hierarchic, Network and Relational. Most database systems developed recently are relational.(6) And we will also use relational database organization in this thesis, because of its higher convenience and flexibility relative to others. A relational database consists of a collection of normalized files with no links between them. These files are known as tables or relations. Records in the files are known as rows, and fields in the file are known as columns. The tables need not be sorted. All the necessary data retrieval functions are done by database manager using a few special operations, including one which merges files together.

2.6 SOME FUNDAMENTAL CONCEPTS

2.6.1 File

A file is typically a collection of records of the same type. (e.g. family (aile) file used in this thesis)

2.6.2 Record

A record describes an object together with the details of that object. Thus, a record in a file might describe names of the father, mother, brothers/sisters and their occupations, education levels and addresses.

2.6.3 Field

A field is a single unit of information within a record. In the example above, name of the father is a field.

2.6.4 Index File

Index file is used in conjunction with a main data file. It has short records which contain just the key data from the main file. The index is kept in sorted order while the main file is unsorted. When a new record is added to the data file, it can simply be placed at the end. The index is updated at the same time to maintain the correct record. (e.g. in the thesis BOLINDX is an index file and KIMLIK file is its main file.)

2.6.5 Key Field

A key is a string of characters that is related to one record in the database. For easy reference, a key is based on some important element (a field) in the

record. For instance, the keys that are mostly used in this thesis are the names and the surnames of the students.

2.6.6 Normalization

Normalization is a methodology for the design of the relational databases. This is a bottom-up methodology. The design proceeds as follows. First, a poor relational schema is designed directly from the requirements. Then the schema is refined in steps by eliminating certain aspects of redundancy (and thus potential inconsistency and update anomalies.) (Sharman (10), Date (4), Jackson (6), Ullman (11), Smith and Brnes (1), Zaniola and Melkonoff (12)). As an example of the normalization process, the one which is used in this thesis will be given. Consider the following situation. The record of each student should contain information about their dormitories, education, family, ID, and identity books. Redundant storage of these can be seen easily. If one of them is changed, instead of updating one of them in the relation (because only one value is actually changed), we should update all of them. Normalized form of the above relation can be the separated files of dormitory (yurt file), ID (kimlik file), education (egitim file), family (aile file), and identity book (nufus file).

CHAPTER 3 - HOW TO USE THE PROGRAM

3.1 INTRODUCTION

In this chapter, you will find explanations of how to use the program such as how menus come on the screen and which menus perform which jobs. A turkish version of this chapter is given in Appendix I.

The internal structure details of each program files can be seen in Appendix A. Also the program files are given in Appendices B, C, D, E, F, G.

3.2 THE MENUS

There are twelve menus which can be called successively in the program, and each menu provides some alternatives to select. The network of the menus can be seen in figures 1 & 2. Selecting an alternative can be achieved by typing the number that corresponds to that alternative in the menu. In each menu the last alternative which is exit (cikis) helps us to turn back to the upper menu or exit the program in the main menu (anatablo).

3.2.1 Main Menu (Anatablo)

To start the program and reach this menu, the user should use dBASE III Plus pocket program and give the command "DO ANATABLO". Then, a menu will appear on the screen, which the user can choice one of the six

MAIN MENU ANA TABLO

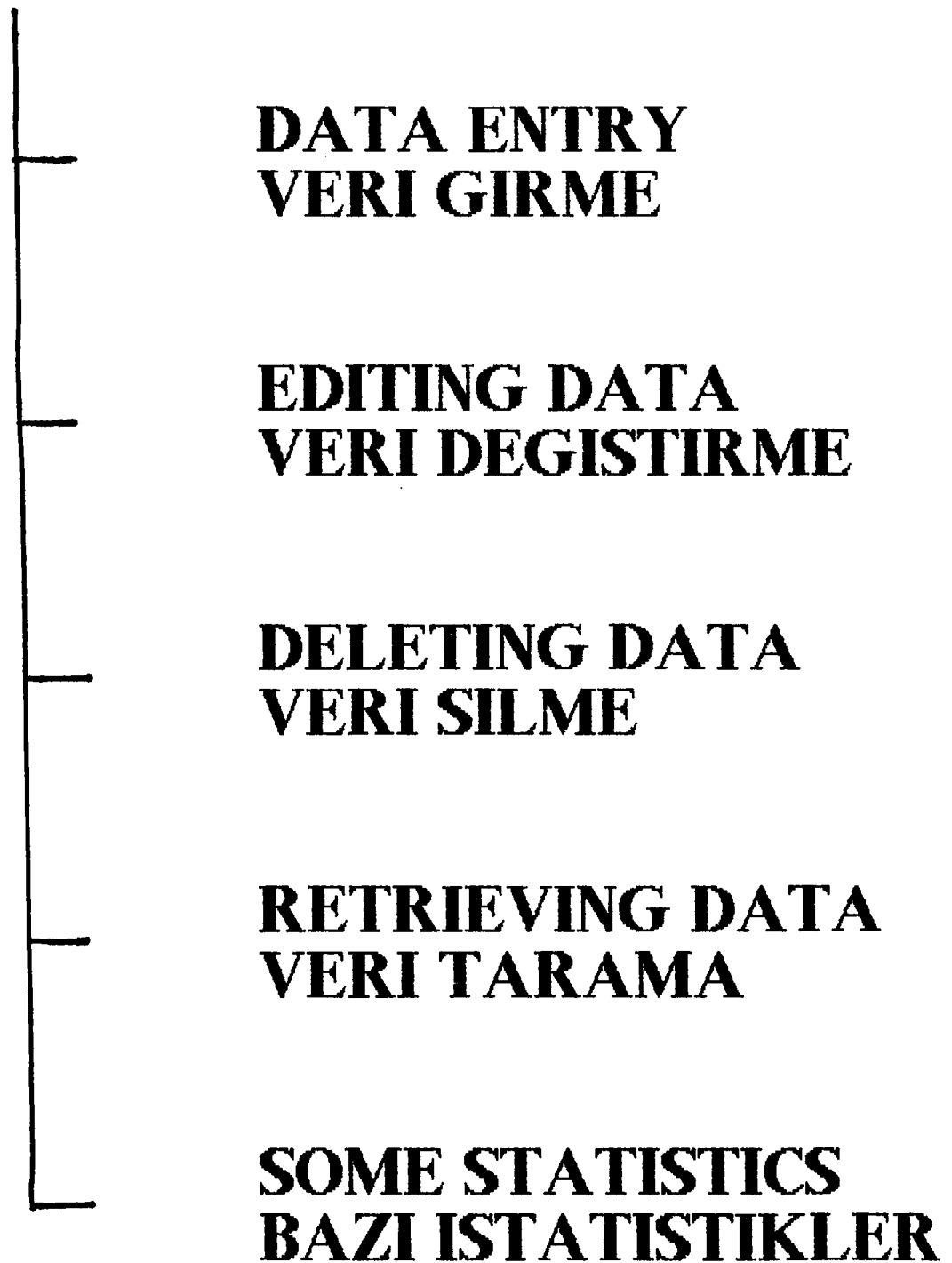


figure 1.

RETRIEVING DATA FROM TARAMA

PRINT
PRINTER'A YOLLA

DISPLAY/PRINT
ALL
HEPSINI GOR/BAS

DISPLAY/PRINT
ACCORDING TO
DEPARTMENTS
BOLUMLERE
GORE GOR/BAS

CHOOSE WHICH FILE
DO YOU WANT TO
DISPLAY/PRINT
GORMEK/BASMAK
ISTEDIGINIZ DOSYAYI
SECINIZ

DISPLAY ON THE
SCREEN
EKRANDA GOR

Figure 2.

alternatives. These alternatives are :

1. Veri Girme (Data Entry)
2. Veri Degistirme (Changing Data)
3. Veri Silme (Deleting Data)
4. Veri Tarama (Retrieving Data)
5. Bazi Istatistikler (Some Statistics)
6. Cikis (Exit)

Anatablo.prg Module is given in Appendix B

3.2.2 Data Entry (Veri Girme)

When you choose the first alternative in the main menu, you will face with a new menu (Vergir.menu) that asks you which files that you want to enter data. These alternatives are :

1. Yurt Bilgileri (Dormitory Data)
2. Kimlik Bilgileri (ID Data)
3. Nufus Bilgileri (Identity Book Data)
4. Egitim Durumu (Education Data)
5. Aile Bilgileri (Family Data)

Vergir.prg Module is given in Appendix C

3.2.2.1 Alternative 1

The program will use Yurt.dbf file and let you to enter new data to the file. The format of the file is given below in Table 1.

3.2.2.2 Alternative 2

Here, the program will use Kimlik.dbf file, the structure of kimlik file is given below in Table 2.

TABLE 1

STRUCTURE OF THE YURT.DBF FILE

FIELD	FIELD NAME	TYPE	WIDTH
1	ISIM	CHARACTER	30
2	SOYAD	CHARACTER	15
3	YURD	CHARACTER	8
4	GIRIS	DATE	8
5	BLOK	NUMERIC	2
6	ODA	NUMERIC	3
7	YATAK	NUMERIC	1
8	TELEFON	NUMERIC	4
9	YURTNO	NUMERIC	8
10	SICIL	MEMO	10
11	XXXXXXXXXX	CHARACTER	10

XXXXXXXXXX: Stands for any additional information.

3.2.2.3 Alternative 3

Used file is Nufus.dbf and the structure of it, is given below in Table 3.

3.2.2.4 Alternative 4

Egitim.dbf file is used in this alternative. The structure of it is shown below in Table 4.

TABLE 2
STRUCTURE OF THE KIMLIK.DBF FILE

FIELD	FIELD NAME	TYPE	WIDTH
1	ISIM	CHARACTER	30
2	SOYAD	CHARACTER	15
3	DOGUM YERI	CHARACTER	15
4	DOGUM TARIHI	DATE	8
5	UYRUGU	CHARACTER	8
6	CINSIYETI	CHARACTER	1
7	FAKULTE	CHARACTER	5
8	BOLUM	CHARACTER	15
9	SINIF	NUMERIC	1
10	GIRIS TARIHI	DATE	8
11	XXXXXXXXXX	CHARACTER	10

TABLE 3
STRUCTURE OF THE NUFUS.DBF FILE

FIELD	FIELD NAME	TYPE	WIDTH
1	ISIM	CHARACTER	30
2	SOYAD	CHARACTER	15
3	IL	CHARACTER	12
4	ILCE	CHARACTER	15
5	BUCAK	CHARACTER	15
6	MAHALLE	CHARACTER	15
7	HANE	CHARACTER	6
8	CILT	CHARACTER	7
9	SAYFA	NUMERIC	3
10	MEDENI HALI	CHARACTER	5
11	DIN	CHARACTER	3
12	PASAPORT NO	NUMERIC	8
13	IKATES NO	NUMERIC	8
14	XXXXXXXXXX	CHARACTER	10

TABLE 4
STRUCTURE OF THE EGITIM.DBF FILE

FIELD	FIELD NAME	TYPE	WIDTH	DECIMAL
1	ISIM	CHARACTER	30	
2	SOYAD	CHARACTER	15	
3	FAKULTE	CHARACTER	6	
4	BOLUM	CHARACTER	15	
5	SINIF	NUMERIC	1	
6	LISE	CHARACTER	15	
7	KOLU	CHARACTER	10	
8	DERECESI	NUMERIC	3	1
9	BURS	CHARACTER	15	
10	YABANCI DIL	CHARACTER	20	
11	XXXXXXXXX	CHARACTER	10	

3.2.2.5 Alternative 5

In this alternative, the program will use Aile.dbf file. And the structure of it is given below in Table 5.

TABLE 5
STRUCTURE OF THE AILE.DBF FILE

FIELD	FIELD NAME	TYPE	WIDTH
1	ADI	CHARACTER	30
2	SOYAD	CHARACTER	15
3	ANKARA ADRESI	CHARACTER	40
4	VELI ADI	CHARACTER	15
5	VELI SOYADI	CHARACTER	15
6	ISI	CHARACTER	15
7	IS ADRESI	CHARACTER	40
8	IKAMET ADRESI	CHARACTER	40
9	BABA ADI	CHARACTER	15
10	BABANIN ISI	CHARACTER	15
11	BABANIN ADRESI	CHARACTER	40
12	ANNE ADI	CHARACTER	15
13	ANNENIN ISI	CHARACTER	15
14	ANNENIN ADRESI	CHARACTER	40
15	KARDES ADI 1	CHARACTER	15
16	KARDES ADI 2	CHARACTER	15
17	KARDES ADI 3	CHARACTER	15
18	ACIL ADRES	CHARACTER	40
19	ACIL TELEFON	NUMERIC	9
20	XXXXXXXXXX	CHARACTER	10

3.2.3 Changing Data (Veri Degistirme)

The second alternative in the main menu helps you to change data that is stored before, but changed later on. This menu (Verdeg.menu) asks you to choose files that you want to make changes. These files are the same with the files in data entry menu.

Verdeg.prg module is given in Appendix D

3.2.4 Deleting Data (Veri Silme)

This alternative, in the main menu, helps you to delete some old or duplicated data in the files. There are seven alternatives in this menu (Versil.menu). First five alternatives ask you to which file do you want to delete. These alternatives can be used in deleting duplicated data. the sixth alternative helps you to delete all files of a given student. This alternative can be used when you want to delete data about an old student. seventh alternative is to exit to main menu.

Versil.prg Module is given in Appendix E.

3.2.5 Retrieving Data (Veri Tarama)

The fourth alternative in the main menu helps us to retrieve data in different ways. First menu (Vertar.menu) that we will be faced with will serve us two alternative :

- 1.Ekranda Gor (Display on the Screen)

2.Printer'a Yolla (Print)

These alternatives as can be understood from their names will help you to see information on the screen or in the printout form. Whatever you choose from these alternatives, you will face with the same menu next.

Vertar.prq Module is given in Appendix E

3.2.5.1 Alternatives of Retrieving Data (Tarama secenekleri)

This menu (VT1.menu) will show up, when you choose 1 or 2 in the previous menu. There are two alternatives again :

1.Tek Tek Gor/Bas (Display/Print One by One)

2.Hepsini Gor/Bas (Display/Print All)

In displaying and printing, the program will use Yurt.lbl, Kimlik.lbl, Nufus.lbl, Egitim.lbl, Aile.lbl files. The format of each file is given below in Tables 6,7,8,9,10.

When you choose first alternative, the program will ask you to enter name and last name of the student and then provides you with five different alternatives which are the previously explained files (yurt, kimlik, nufus, egitim, aile). You can see any of these or all of them about the student that you gave his/her name and surname.

TABLE 6
STRUCTURE OF THE YURTLBL.LBL FILE

LABEL CONTENTS:

1:	ISIM,SOYAD
2:	YURD,YURT NO
3:	GIRIS
4:	BLOK,ODA,YATAK
5:	TELEFON

TABLE 7
STRUCTURE OF THE KIMLIKLBL.LBL FILE

LABEL CONTENTS:

1:	ISIM,SOYAD
2:	DOGUM YERI, DOGUM TARIHI
3:	UYRUGU, CINSIYETI
4:	FAKULTE,BOLUM,SINIF
5:	GIRIS TARIHI

TABLE 8
STRUCTURE OF THE NUFUSLBL.LBL FILE

LABEL CONTENTS:

1:	ISIM,SOYAD
2:	IL,ILCE
3:	BUCAK,MAHALLE
4:	HANE,CILT,SAYFA
5:	MEDENI HALI, DINI
6:	PASAPORT NO, IKATES NO

TABLE 9
STRUCTURE OF THE EGITIMLBL.LBL FILE

LABEL CONTENTS:

1:	ISIM,SOYAD
2:	FAKULTE,BOLUM,SINIF
3:	LISE,KOLU
4:	DERECESI
5:	BURS
6:	YABANCI DIL

TABLE 10
STRUCTURE OF THE AILELBL.LBL FILE

LABEL CONTENTS:

1:	ADI,SOYAD
2:	ANKARA ADRESI
3:	VELI ADI,VELI SOYADI
4:	VELI ISI, IS ADRESI
5:	IKAMETGAH ADRESI
6:	BABA ADI, BABANIN ISI
7:	BABANIN ADRESI
8:	ANNE ADI,ANNENIN ISI
9:	ANNENIN ADRESI
10:	KARDES ADLARI
11:	ACIL ADRES,ACIL TELEFON

-

When you choose the second alternative, you will be supplied with five criteria which are determined by the help of discussion sessions with the dormitory management. These are :

- 1.Display or Print (from now on, we will use D/P notation for Display/Print) according to the departments.

- 2.D/P according to their names.

- 3.D/P according to their surnames.

- 4.D/P according to their dormitory registration number.

- 5.D/P according to their entry date to the dormitory.

When first criterion is chosen, the program will use Kimlik.dbf file and another menu will show up in the screen which provides you with two alternatives.

- 1.Tek Tek Bolumleri Gor/Bas (D/P One Department)

- 2.Butun Bolumleri Gor/Bas (D/P All Departments)

In the first alternative, the user gives the name of the department and program shows you all students of the given department.

In the second alternative, all departments will show up with alphabetical order. The program will use Kimlik.lbl in displaying and printing the records.

Second, third, fourth and fifth criteria in the previous menu will use Yurt.dbf file and Yurt.lbl in displaying/printing.

Within this program, computer will always ask

the following question :

INDEX FILE IS ALREADY EXIST, OVERWRITE IT (Y) (N)

Y should be entered each time in order to continue.

3.2.6. Some Statistics (Bazi Istatistikler)

When we choose this alternative in the main menu, we will be faced with four alternatives. All these alternatives give us the number of students of specified characteristics. These characteristics are:

1. Male students,
2. Female students,
3. Foreign students,
4. Number of students, according to their classes.

.

CHAPTER IV - CONCLUSION

This thesis provides a new and computerized solution for student registration of Bilkent University Dormitories. Main idea in the thesis is the establishment of DBMS for Bilkent University Dormitory management. We have used dBASE III Plus for the establishment of a DBMS which is easy to use, easy to modify, easy to meet new or changing requirements and which has improved data integrity and security.

Five files are used for each student, and there are ten menus that makes the program easy to use. Yurt, Kimlik, Nufus, Egitim, and Aile files and their structures are determined after the discussions with the dormitory management.

Some of the advantages of this program are:

- 1) Records will not be lost or replaced in the wrong order by the user,
- 2) It will be easy to manipulate, display, or analyse the data,
- 3) Retrieving records will be quicker then before,
- 4) It is easy to modify program, when it is needed.

The main disadvantage of the program is the limits of the dBASE III Plus. Some of the limits are:

- 8 characters in a date field,
- 1 character in a logical field,

- 19 characters in a numeric field,
- 4000 characters in a memo field

10 open files at one time.

Most important limitation of this program is the capacity of the memory device that will be used. Some computations on the number of complete records that can be stored by different memory devices are given below.

A single floppy disk : $360.000/1026=350$ records.

360.000 is the capacity of the floppy disk.

1026 is the total of field widths in each file for each student.

A 1.2 MB Disk drive : $1.200.000/1026=1.170$ records.

An optional 20 MB Hard disk : $20.000.000/1026=19.500$ records

And also if capacity is available, this program can store up to $1.000.000.000/1026=974.658$ records. But in practice, dBASE III Plus shouldn't be used to manage more than 10.000 records in a single database. Because it takes too long to retrieve information.

An application example of the program is given in Appendix H.

REFERENCES

1. Barnes, G.M. and Smith, P.D., Files And Database, Addison-Wesley Publishing Company, Massachusetts, (1987).
2. Bowers, D.S., From Data To Database, Van Nostrand Reinhold(UK) Co. Ltd., Molly Millars Lane, Wokingham, Berkshire, England, (1987).
3. Bray, O.H., Distributed Database Management System, Lexington Books, Massachusetts, (1982).
4. Date, C.J., An Introduction To Database Systems, 3rd Ed, Addison-Wesley, Reading, Massachusetts, (1981).
5. De Pace, M., The DBASE Programming Language, Collins Books, 8 Grafton Street, London, (1986).
6. Jackson, G.A., Relational Database Design With Microcomputer Applications, Prentice Hall, Englewood Cliffs, New Jersey, (1988).
7. Lima, T., Mastering dBASE III Plus In Less Than A Day, Prentice Hall, Englewood Cliffs, New Jersey, (1986).
8. Martin, J., An End Users Guide To Database, Prentice Hall, Englewood Cliffs, New Jersey, (1981).
9. Rishe, N., Database Design Fundamentals, Prentice Hall, Englewood Cliffs, New Jersey, (1988).
10. Sharman, G., Introduction To The Database On Microcomputers, Addison-Wesley Publishing Company, Wokingham, England, (1987).
11. Ullman, J.D., Principles Of Database Systems, Computer Science Press, Washingt D.C., (1979).
12. Zaniolo, C., and Melkanoff, M.A., "On The Design Of Relational Database Design Schemata", ACM Transactions On Database Systems, Vol.6, No.1, (1981), pp.1-47.

APPENDIX A

.

INTERNAL STRUCTURE OF THE PROGRAM

1. INTRODUCTION

This Appendix explains the internal structure details of each program files (*.prg files). The program files can be seen in Appendix A,B,C,D,E. While going through this chapter, we will follow the same sequence with the third chapter.

2. ANATABLO.PRG

Here, we will see the commands that are used in the Anatablo.prg module. These are : clear, clear all, set talk off, set echo off, store, do while, do ase, @say/get, @row1,col1 to row2,col2 double, read, return.

2.1 Clear

Clears the screen of any previously displayed data and positions the cursor at the top left corner of the screen.

2.2 Clear All

Closes all database files including the one currently in use.

2.3 Set Talk Off

Used in programs to prevent the display of dBASE

messages such as the current record number.

2.4 Set Echo Off

Used not to display program lines.

2.5 Store

Creates memory variable, here, this command assigns the type characteristics to a variable in the sense that the created variable will be of the same type as the expression stored to it.

2.6 Do While

Enables you to perform a DO loop. This means that the commands between the DO WHILE line and the ENDDO line will be executed repeatedly while the condition specified in the command evaluates as true. If the condition is evaluated as false, execution will pass immediately to the command following the ENDDO line.

2.7 Do Case

This command enables you to select a single course of action from a number of alternatives. The command is frequently used in processing menus where the user is able to select one menu option at a time.

2.8 @ Say/Get

@row,col say/get specify a particular position on the screen for either output or input and to apply formatting rules to the display and input areas.

2.9 @Row1,Col1 to Row2,Col2 Double

Draws a rectangle with specified coordinates.

2.10 Read

This command is used to activate a Get area and to transfer data from the Get area.

2.11 Return

Hands execution back to the calling module.

3. VERGIR.PRG

Additional commands that are used under this program are:

3.1 Use

Opens a database file for use. The filename may be supplied without an extension.

3.2 Append

Result in addition of a record or records

physically to the end of the .dbf file currently in use.

4. VERDEG.PRG

Additional commands are:

4.1 Public

This command is used to prevent the release of specified memory variables when the program module in which they are created comes to an end. They do not exist outside the program module that created them, unless it is in a nested program module. (i.e. one that was called from within the module which created the variable. Foundveri is a variable specified in the program).

4.2 Accept

Used as an alternative to the @Get command to ask the user for character information. The prompt is an optional character string which is valuable in explaining to the user what is being asked.

4.3 Edit

Present the current specified record and let you to make changes on it.

5. VERSIL.PRG

Additional commands are:

5.1 Delete

Used to remove records from the database file. It is , however, only the first step in the process since it does not itself physically delete records from the database. It simply marks them for future deletion.

5.2 Pack

Physical deletion of records is carried out by the Pack command.

5.3 Wait

This command displays a message on the screen and then suspends execution of the program until the user presses a key, before it allows the program to continue.

5.4 Do

Executes a separate program module in a .prg file.

6. VERTAR.PRG

6.1 If

Executes the commands between the If line and its

paired Endif, if the condition evaluates as true, otherwise control passes to the command following the paired Endif.

6.2 Else

Provide the means of executing one set of commands within the If/Endif.

6.3 Label

This command is intended to help you to produce name and adress labels from a dBASE database file. As a prerequisite it is necessary to have created an .lbl file which contains the definition of the labels to be produced by the label command. (e.g. LABEL NEXT 1 FORM YURT.LBL will display the next record in the form of previously created .lbl file.)

6.4 Index

Creates a seperate index file using the filename specified in the command togetherwith the filename extension .ndx.

The purpose of index file is twofold. It provides a means of sequencing the database information without actually disturbing the database itself and it enables you to find a particular record within the seconds.

7. ISTATISTIK.PRG

7.1 Count

This command provides a count or tally either of the total number of records in the database file currently in use, or only of those records which meet the specified conditions.

APPENDIX B


```

TYPE ANATABLO.PRG
CLEAR
CLEAR ALL
TALK OFF
ECHO OFF
KAHRAHAN
IT
DORE " " TO CHOICE
WHILE .T.
  CLEAR
  @2,15 TO 18,60 DOUBLE
  @3,22 SAY '          ANA TABLO'
  @7,17 SAY '<1>'          VERI GIRME'
  @8,17 SAY '<2>'          VERI DEGISTIRME'
  @9,17 SAY '<3>'          VERI SILME'
  @10,17 SAY '<4>'         VERI TARAMA'
  @11,17 SAY '<5>'         BAZI ISTATISTIKLER'
  @12,17 SAY '<6>'         CIKIS'
  @14,17 SAY 'SECENEGINIZI GIRINIZ.' GET CHOICE
  READ
  DO CASE
    CASE CHOICE = "1"
      DO VERGIR
    CASE CHOICE = "2"
      DO VERDEG
    CASE CHOICE = "3"
      DO VS
    CASE CHOICE = "4"
      DO VERTAR
    CASE CHOICE = "5"
      DO ISTATISTIK
    CASE CHOICE = "6"
      RETURN
    OTHERWISE
      LOOP
  ENDCASE
DORE " " TO CHOICE
DDO

```

APPENDIX C

```

TYPE VERGIR.PRG
T TALK OFF
T ECHO OFF
DORE " " TO CHOICE
WHILE .T.
  CLEAR
  @2,15 TO 18,60 DOUBLE
  @3,22 SAY '      VERI GIRILECEK DOSYALAR'
  @7,17 SAY ' <1>          YURT BILGILERI'
  @8,17 SAY ' <2>          KIMLIK BILGILERI'
  @9,17 SAY ' <3>          NUFUS BILGILERI'
  @10,17 SAY ' <4>         EGITIM DURUMU'
  @11,17 SAY ' <5>         AILE BILGILERI'
  @12,17 SAY ' <6>         CIKIS'
  @14,17 SAY '      SECENEGINIZI GIRINIZ.' GET CHOICE
  READ
  DO CASE
    CASE CHOICE = "1"
      DO VBA
    CASE CHOICE = "2"
      DO VBB
    CASE CHOICE = "3"
      DO VGC
    CASE CHOICE = "4"
      DO VGD
    CASE CHOICE = "5"
      DO VGE
    CASE CHOICE = "6"
      RETURN
    OTHERWISE
      LOOP
  ENDCASE
DORE " " TO CHOICE
DDO

```

TYPE VBA.PRG
T TALK OFF
T ECHO OFF
E YURT
PEND
AD
TURN

TYPE VBB.PRG
T TALK OFF
T ECHO OFF
E KIMLIK
PEND
AD
TURN

TYPE VBC.PRG
T TALK OFF
T ECHO OFF
E NUFUS
PEND
AD
TURN

TYPE VBD.PRG
T TALK OFF
T ECHO OFF
E EGITIM
PEND
AD
TURN

TYPE VGE.PRG
T TALK OFF
T ECHO OFF
E AILE
PEND
AD
TURN

APPENDIX D

```

YPE VERDEG.PRG
  TALK OFF
  ECHO OFF
  RE " " TO CHOICE
  WHILE .T.
    CLEAR
    @2,15 TO 18,60
    @3,22 SAY '   VERI DEGISTIRILECEK DOSYALAR'
    @7,17 SAY ' <1>           YURT BILGILERI'
    @8,17 SAY ' <2>           KIMLIK BILGILERI'
    @9,17 SAY ' <3>           NUFUS BILGILERI'
    @10,17 SAY ' <4>          EGITIM DURUMU'
    @11,17 SAY ' <5>          AILE BILGILRI'
    @12,17 SAY ' <6>          CIKIS'
    @14,17 SAY ' SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
      CASE CHOICE = "1"
        DO VDA
      CASE CHOICE = "2"
        DO VDB
      CASE CHOICE = "3"
        DO VDC
      CASE CHOICE = "4"
        DO VDD
      CASE CHOICE = "5"
        DO VDE
      CASE CHOICE = "6"
        RETURN
      OTHERWISE
        LOOP
    ENDCASE
  JRE " " TO CHOICE
  DDQ

```

```

TYPE VDA.PRG
EAR
EAR ALL
UNDVERI=.T.
T TALK OFF
T ECHO OFF
E YURT
  CLEAR
  ?
  ?
  ?
ACCEPT " ISMI GIRINIZ : " TO NAME
ACCEPT " SOYAD GIRINIZ : " TO SIRNAME
EDIT ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  IF EOF ()
    FOUNDVERI=.F.
  RETURN
DIF
TURN

```

```

TYPE VDB.PRG
EAR ALL
BLIC FOUNDVERI
T TALK OFF
T ECHO OFF
UNDVERI = .T.
E KIMLIK
  CLEAR
  ?
  ?
  ?
ACCEPT " ISMI GIRINIZ:" TO NAME
ACCEPT " SOYADI GIRINIZ:" TO SIRNAME
EDIT ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  IF EOF()
    FOUNDVERI= .F.
  RETURN
NDIF
TURN

```

```

TYPE VDC.PRG
EAR ALL
UBLIC FOUNDVERI
ET TALK OFF
ET ECHO OFF
UNDVERI=.T.
SE NUFUS
  CLEAR
  ?
  ?
  ?
ACCEPT" ISMI GIRINIZ : " TO NAME
ACCEPT" SOYADI GIRINIZ : " TO SIRNAME
EDIT ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  IF EOF()
    FOUNDVERI=.F.
  RETURN
NDIF
TURN

```

```

. TYPE VDD.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE EGITIM
  CLEAR
  ?
  ?
  ?
  ACCEPT " ISMI GIRINIZ:" TO NAME
  ACCEPT " SOYADI GIRINIZ:" TO SIRNAME
  EDIT ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  IF EOF()
    FOUNDVERI=.F.
    RETURN
  ENDIF
RETURN

```

```

. TYPE VDE.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE AILE
  CLEAR
  ?
  ?
  ?
  ACCEPT " ISMI GIRINIZ:" TO NAME
  ACCEPT " SOYADI GIRINIZ:" TO SIRNAME
  EDIT ALL FOR ADI=NAME .AND. SOYAD=SIRNAME
  IF EOF()
    FOUNDVERI=.F.
    RETURN
  ENDIF
RETURN

```


APPENDIX E

```

. TYPE VERSIL.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '        SILINICEK DOSYAYI SECINIZ'
    @7,17 SAY ' <1>                YURT BILGILERI'
    @8,17 SAY ' <2>                KIMLIK BILGILERI'
    @9,17 SAY ' <3>                NUFUS BILGILERI'
    @10,17 SAY ' <4>               EGITIM DURUMU'
    @11,17 SAY ' <5>               AILE BILGILERI'
    @12,17 SAY ' <6>               HEPSINI SIL'
    @13,17 SAY ' <7>               CIKIS'
    @15,17 SAY ' SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO VSA
        CASE CHOICE = "2"
            DO VSB
        CASE CHOICE = "3"
            DO VSC
        CASE CHOICE = "4"
            DO VSD
        CASE CHOICE = "5"
            DO VSE
        CASE CHOICE = "6"
            DO VSF
        CASE CHOICE = "7"
            RETURN
    ENDCASE
    STORE " " TO CHOICE
ENDDO
RETURN

```

```

. TYPE VSA.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE YURT
    CLEAR
    ?
    ?
    ?
    ACCEPT" ISMI GIRINIZ: " TO NAME
    ACCEPT" SOYADI GIRINIZ: " TO SIRNAME
    DELETE ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
    PACK
    WAIT" BU KAYIT SILINMISTIR."
    IF EOF()
        FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

```

. TYPE VSB.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE KIMLIK
    CLEAR
    ?
    ?
    ?
    ACCEPT" ISMI GIRINIZ:" TO NAME
    ACCEPT" SOYADI GIRINIZ:" TO SIRNAME
    DELETE ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
    PACK
    WAIT" BU KAYIT SILINMISTIR."
    IF EOF()
        FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

```

. TYPE VSD.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI =.T.
USE NUFUS
  CLEAR
  ?
  ?
  ?
  ACCEPT" ISMI GIRINIZ:" TO NAME
  ACCEPT" SOYADI GIRINIZ:" TO SIRNAME
  DELETE ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  PACK
  WAIT" BU KAYIT SILINMISTIR."
  IF EOF()
    FOUNDVERI= .F.
  RETURN
ENDIF
RETURN

```

```

. TYPE VSD.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE EGITIM
  CLEAR
  ?
  ?
  ?
  ACCEPT" ISMI GIRINIZ:" TO NAME
  ACCEPT" SOYADI GIRINIZ:" TO SIRNAME
  DELETE ALL FOR ISIM=NAME .AND. SOYAD=SIRNAME
  PACK
  WAIT" BU KAYIT SILINMISTIR."
  IF EOF()
    FOUNDVERI=.F.
  RETURN
ENDIF
RETURN

```

```

. TYPE VSE.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
USE AILE
  CLEAR
  ?
  ?
  ?
  ACCEPT"ISMI GIRINIZ:" TO NAME
  ACCEPT"SOYADI GIRINIZ:" TO SIRNAME
  DELETE ALL FOR ADI=NAME .AND. SOYAD=SIRNAME
  PACK
  WAIT" BU KAYIT SILINMISTIR."
  IF EOF()
    FOUNDVERI=.F.
  RETURN
ENDIF
RETURN

```

```

. TYPE VSF.PRG
PUBLIC FOUNDVERI
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI=.T.
  CLEAR
  ?
  ?
  ?
  ACCEPT" ISMI GIRINIZ:" TO NAME
  ACCEPT" SOYADINI GIRINIZ:" TO SIRNAME
  *****
  *      CALL SUBROUTINE VS1      *
  *****
DO VS1
  *****
  *      CALL SUBROUTINE VS2      *
  *****
DO VS2
  *****
  *      CALL SUBROUTINE VS3      *
  *****
DO VS3
  *****
  *      CALL SUBROUTINE VS4      *
  *****
DO VS4
  *****
  *      CALL SUBROUTINE VS5      *
  *****
DO VS5
  USE
RETURN

```

```

. TYPE VS1.PRG
SET TALK OFF
SET ECHO OFF
DO VSZ
IF .NOT. FOUNDVERI
RETURN
ELSE
DELETE
PACK
@ 5.0 SAY " YURT KAYITLARI SILINMISTIR."
STORE 1 TO N
DO WHILE N<=40
    STORE N+1 TO N
ENDDO N
RELEASE N
USE
RETURN
ENDIF
RETURN

```

```

. TYPE VS2.PRG
SET TALK OFF
SET ECHO OFF
DO VSY
IF .NOT. FOUNDVERI
RETURN
ELSE
DELETE
PACK
@ 7.0 SAY "KIMLIK KAYITLARI SILINMISTIR."
STORE 1 TO N
DO WHILE N<=40
    STORE N+1 TO N
ENDDO N
RELEASE N
USE
ENDIF
RETURN
RETURN

```

```

. TYPE VS3.PRG
SET TALK OFF
SET ECHO OFF
DO VSV
IF .NOT. FOUNDVER1
RETURN
ELSE
DELETE
*PACK
@ 9.0 SAY "NUFUS KAYITLARI SILINMISTIR."
STORE 1 TO N
DO WHILE N<=40
    STORE N+1 TO N
ENDDO N
RELEASE N
USE
RETURN
ENDIF
RETURN

```

```

. TYPE VS4.PRG
SET TALK OFF
SET ECHO OFF
DO VSV
IF .NOT. FOUNDVER1
RETURN
ELSE
DELETE
PACK
@ 11.0 SAY " EBITIM KAYITLARI SILINMISTIR."
STORE 1 TO N
DO WHILE N<=40
    STORE N+1 TO N
ENDDO N
RELEASE N
USE
RETURN
ENDIF
RETURN

```

```

. TYPE VS5.PRG
SET TALK OFF
SET ECHO OFF
DO VSR
IF .NOT. FOUNDVER1
RETURN
ELSE
DELETE
PACK
@ 13,0 SAY " AILE KAYITLARI SILIRMIŞTİR."
STORE 1 TO N
DO WHILE NK=40
    STORE N+1 TO N
ENDDO N
RELEASE N
USE
RETURN
ENDIF
RETURN

```

```

. TYPE VS2.PRG
USE YURT
CLEAR
LOCATE FOR ISIM=NAME .AND. SOYAD=SIRNAME
IF EOF()
    FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

```

. TYPE VSY.PRG
USE KIMLIK
FOUNDVERI=.T.
CLEAR
LOCATE FOR ISIM=NAME .AND. SOYAD=SIRNAME
IF EOF()
    FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

```

. TYPE VSV.PRG
USE NUFUS
FOUNDVERI=.T.
CLEAR
LOCATE FOR ISIM=NAME .AND. SOYAD=SIRNAME
IF EOF()
    FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```



```

. TYPE VSU.PRG
USE EGITIM
    FOUNDVERI=.T.
    CLEAR
    LOCATE FOR ISIM=NAME .AND. SOYAD=SIRNAME
    IF EOF()
        FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

```

. TYPE VSR.PRG
USE AILE
    FOUNDVERI=.T.
    CLEAR
    LOCATE FOR ADI=NAME .AND. SOYAD=SIRNAME
    IF EOF()
        FOUNDVERI=.F.
    RETURN
ENDIF
RETURN

```

APPENDIX F

```

. TYPE VERTAR.PRG
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '          VERI TARAMA'
    @7,17 SAY ' <1>          EKRANDA GOR'
    @8,17 SAY ' <2>          PRINTER'A YOLLA'
    @9,17 SAY ' <3>          CIKIS'
    @12,17 SAY ' SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO VT1
        CASE CHOICE = "2"
            DO VT2
        CASE CHOICE = "3"
            RETURN
    ENDCASE
    STORE " " TO CHOICE
ENDDO

```

```

. TYPE VT1.PRG
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '          TARAMA SECENEKLERI'
    @7,17 SAY ' <1>          TEK TEK GOR'
    @8,17 SAY ' <2>          HEPSINI GOR'
    @9,17 SAY ' <3>          CIKIS'
    @12,17 SAY ' SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO VTG
        CASE CHOICE = "2"
            DO VHG
        CASE CHOICE = "3"
            RETURN
    ENDCASE
STORE " " TO CHOICE
ENDDO

```

```

. TYPE VT2.PRG
SET TALK OFF
SET ECHO OFF
SET PRINT ON
@20,1 SAY " PRINTERIN ACIK VE ONLINE (READY) OLDUGUNU KONTROL EDINIZ."
WAIT
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '          PRINT SECENEKLERI'
    @7,17 SAY ' <1>          TEK TEK BAS'
    @8,17 SAY ' <2>          HEPSINI BAS'
    @9,17 SAY ' <3>          CIKIS'
    @16,17 SAY ' SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO VTG
        CASE CHOICE = "2"
            DO VHG
        CASE CHOICE = "3"
            SET PRINT OFF
            RETURN
    ENDCASE
STORE " " TO CHOICE
ENDDO

```

```

. TYPE VTG.PRG
PUBLIC FOUNDVERI
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI = .T.
  CLEAR
  ?
  ?
  ?
  ACCEPT " İSMİ GIRİNİZ:" TO NAME
  ACCEPT " SOYADINI GIRİNİZ:" TO SIRNAME
  DO VTT
RETURN

```

```

. TYPE VTT.PRG
  STORE " " TO CHOICE
  DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,18 SAY 'GORMEK/BASMAK İSTEDİĞİNİZ DOSYAYI SECİNİZ'
    @7,17 SAY ' <1>          YURT BİLGİLERİ'
    @8,17 SAY ' <2>          KİMLİK BİLGİLERİ'
    @9,17 SAY ' <3>          NUFUS BİLGİLERİ'
    @10,17 SAY <4>          EĞİTİM DURUMU'
    @11,17 SAY <5>          AİLE BİLGİLERİ'
    @12,17 SAY ' <6>          ÇIKIŞ'
    @16,17 SAY 'SECENEGİNİZİ GIRİNİZ.' GET CHOICE
    READ
    DO CASE
      CASE CHOICE = "1"
        DO VSZ ,
        DO VVV
      CASE CHOICE = "2"
        DO VSY
        DO VV1
      CASE CHOICE = "3"
        DO VSV
        DO VV2
      CASE CHOICE = "4"
        DO VSU
        DO VV3
      CASE CHOICE = "5"
        DO VSR
        DO VV4
      CASE CHOICE = "6"
        RETURN
      OTHERWISE
        LOOP
    ENDCASE
  ENDDO
RETURN

```

```
. TYPE VVV.PRG
IF .NOT. FOUNDVERI
RETURN
ELSE
LABEL NEXT 1 FORM YUNLEBL
WAIT
USE
RETURN
ENDIF
RETURN
```

```
. TYPE VV1.PRG
IF .NOT. FOUNDVERI
RETURN
ELSE
LABEL NEXT 1 FORM KIMLIKLBL
WAIT
USE
RETURN
ENDIF
RETURN
```

```
. TYPE VV2.PRG
IF .NOT. FOUNDVERI
RETURN
ELSE
LABEL NEXT 1 FORM NUFUSLBL
WAIT
USE
RETURN
ENDIF
RETURN
```

```
. TYPE VV3.PRG
IF .NOT. FOUNDVERI
RETURN
ELSE
LABEL NEXT 1 FORM EGIT1MLBL
WAIT
USE
RETURN
ENDIF
RETURN
```

```

. TYPE VV4.PRG
IF .NOT. FOUNDVERI
RETURN
ELSE
LABEL NEXT 1 FORM AILELBL
WAIT
USE
RETURN
ENDIF
RETURN

```

```

. TYPE VH6.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '      HEPSINI GOR/BAS'
    @7,17 SAY '    <1>      BOLUMLERINE GORE'
    @8,17 SAY '    <2>      ISIM SIRASINA GORE'
    @9,17 SAY '    <3>      SOYADI SIRASINA GORE'
    @10,17 SAY '  <4>      YURT NUMARALARINA GORE'
    @11,17 SAY '  <5>      GIRIS TARIHINE GORE'
    @12,17 SAY '  <6>      CIKIS'
    @16,17 SAY 'SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO HB6
        CASE CHOICE = "2"
            DO H16
        CASE CHOICE = "3"
            DO H36
        CASE CHOICE = "4"
            DO HY6
        CASE CHOICE = "5"
            DO H66
        CASE CHOICE = "6"
            RETURN
        OTHERWISE
            LOOP
    ENDCASE
    STORE " " TO CHOICE
ENDDO

```

```

. TYPE HBB.PRG
CLEAR
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
  CLEAR
  @2,15 TO 18,60 DOUBLE
  @3,22 SAY "      BOLUMLERE GORE GOR"
  @7,17 SAY "<1>      TEK TEK BOLUMLERI GOR"
  @8,17 SAY "<2>      BUTUN BOLUMLERI GOR"
  @9,17 SAY "<3>      CIKIS"
  @16,17 SAY "  SECENEGINIZI GIRINIZ." GET CHOICE
  READ
  DO CASE
    CASE CHOICE = "1"
      DO HBT
    CASE CHOICE = "2"
      DO HBB
    CASE CHOICE = "3"
      RETURN
  ENDCASE
STORE " " TO CHOICE

. TYPE HBT.PRG
CLEAR ALL
PUBLIC FOUNDVERI
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE KIMLIK
  CLEAR
  ?
  ?
  ?
  ACCEPT " GORMEK ISTEDIGINIZ BOLUMU GIRINIZ:" TO DEPARTMENT
  COUNT TO NUM FOR BOLUM = DEPARTMENT
  ? DEPARTMENT, " : BU BOLUMDEKI OGRENCI SAYISI = ", NUM
  WAIT
  GO TOP
  DO WHILE ANSWER = "E"
    LABEL NEXT 3 FORM KIMLIKLBL FOR BOLUM=DEPARTMENT
    ACCEPT " DEVAMINI GOR (E) EVET / (H) HAYIR " TO ANSWER
    IF EOF()
      FOUNDVERI= .F.
      RETURN
    ENDIF
    SKIP
  ENDDO
RETURN

```



```

. TYPE HBB.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE KIMLIK
INDEX ON BOLUM TO BOLINDX
DO WHILE ANSWER = "E"
    LABEL NEXT 3 FORM KIMLIKLBL
    ACCEPT "DEVAMINI GOR (E) EVET / (H) HAYIR " TO ANSWER
    IF EOF()
        FOUNDVERI= .F.
        RETURN
    ENDIF
    SKIP
ENDDO
WAIT

```

```

. TYPE HIG.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE YURT
INDEX ON ISIM TO ISINDX
DO WHILE ANSWER = "E"
    LABEL NEXT 3 FORM YURTLBL
    ACCEPT "DEVAMINI GOR (E) EVET / (H) HAYIR " TO ANSWER
    IF EOF()
        FOUNDVERI= .F.
        RETURN
    ENDIF
    SKIP
ENDDO
WAIT

```

```

. TYPE HSG.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE YURT
INDEX ON SOYAD TO SOYINDX
DO WHILE ANSWER = "E"
    LABEL NEXT 3 FORM YURTLBL
    ACCEPT "DEVAMINI GOR (E) EVET / (H) HAYIR " TO ANSWER
    IF EOF()
        FOUNDVERI= .F.
        RETURN
    ENDIF
    SKIP
ENDDO
WAIT

```

```

. TYPE HYG.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE YURT
INDEX ON YURTNO TO YURINDX
DO WHILE ANSWER = "E"
    LABEL NEXT 3 FROM YURTLBL
    ACCEPT "DEVAMINI GOR (E) EVET / (H) HAYIR" TO ANSWER
    IF EOF()
        FOUNDVERI= .F.
        RETURN
    ENDIF
    SKIP
ENDDO
WAIT

```

```

. TYPE HGG.PRG
CLEAR ALL
SET TALK OFF
SET ECHO OFF
FOUNDVERI= .T.
STORE "E" TO ANSWER
USE YURT
INDEX ON GIRIS TO GIRISINDX
DO WHILE ANSWER = "E"
    LABEL NEXT 3 FROM YURTLBL
    ACCEPT "DEVAMINI GOR (E) EVET / (H) HAYIR" TO ANSWER
    IF EOF()
        FOUNDVERI= .F.
        RETURN
    ENDIF
    SKIP
ENDDO
WAIT

```

APPENDIX G

```

. TYPE ISTATISTIK.PRG
SET TALK OFF
SET ECHO OFF
STORE " " TO CHOICE
DO WHILE .T.
    CLEAR
    @2,15 TO 18,60 DOUBLE
    @3,22 SAY '      ISTATISTIK SECENEKLERI'
    @7,17 SAY  <1>          ERKEK OGRENCI SAYISI'
    @8,17 SAY  <2>          KIZ OGRENCI SAYISI'
    @9,17 SAY  <3>          YABANCI OGRENCI SAYISI'
    @10,17 SAY <4> SINIFLARINA GORE OGRENCI DAGILIMI'
    @11,17 SAY <5>          CIKIS'
    @13,17 SAY '      SECENEGINIZI GIRINIZ.' GET CHOICE
    READ
    DO CASE
        CASE CHOICE = "1"
            DO VISA
        CASE CHOICE = "2"
            DO VISB
        CASE CHOICE = "3"
            DO VISC
        CASE CHOICE = "4"
            DO VISD
        CASE CHOICE = "5"
            RETURN
        OTHERWISE
            LOOP
    ENDCASE
ENDDO

```

```

. TYPE VISA.PRG
SET TALK OFF
SET ECHO OFF
CLEAR
USE KIMLIK
STORE "E" TO CINS
COUNT TO CIN FOR CINSIYETI = CINS
? CINS," : TOPLAM ERKEK OGRENCI SAYISI = ",CIN
WAIT
RETURN

```

```

. TYPE VISD.PRG
SET TALK OFF
SET ECHO OFF
CLEAR
USE KIMLIK
STORE "K" TO CINS
COUNT TO CIN FOR CINSIYETI = CINS
? CINS," : TOPLAM KIZ OGRENCI SAYISI = ",CIN
WAIT
RETURN

```

```

. TYPE VISD.PRG
SET TALK OFF
SET ECHO OFF
CLEAR
USE KIMLIK
STORE "T.C." TO UYRUK
COUNT TO UYRU FOR UYRUGU <> UYRUK
? " TOPLAM YABANCI OGRENCI SAYISI = ",UYRU
WAIT
RETURN

```

```

. TYPE VISD.PRG
SET TALK OFF
SET ECHO OFF
CLEAR
STORE " " TO CLASS
USE KIMLIK
@2,15 TO 18,60 DOUBLE
@3,22 SAY ' SINIF SECENEKLERI'
@7,17 SAY ' <H> TOPLAM HAZIRLIK OGRENCILERI'
@8,17 SAY ' <1> TOPLAM BIRINCI SINIF OGRENCILERI'
@9,17 SAY ' <2> TOPLAM IKINCI SINIF OGRENCILERI'
@10,17 SAY ' <3> TOPLAM UCUNCU SINIF OGRENCILERI'
@11,17 SAY ' <4> TOPLAM DORDUNCU SINIF OGRENCILERI'
@12,17 SAY ' <5> TOPLAM MASTER OGRENCILERI'
@14,17 SAY 'HANGI SINIFI GORMEK ISTIYORSANIZ GIRINIZ.';
GET CLASS
READ
COUNT TO SIN FOR SINIF = CLASS
? CLASS," : BU SINIFDAKI TOPLAM OGRENCI SAYISI = ",SIN
WAIT
RETURN

```

APPENDIX H


```

ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?
3          VERI DEGISTIRILECEK DOSYALAR          3
3
3
3
3
3 <1>          YURT BILGILERI          3
3 <2>          KIMLIK BILGILERI        3
3 <3>          NUFUS BILGILERI        3
3 <4>          EGITIM DURUMU          3
3 <5>          AILE BILGILRI          3
3 <6>          CIKIS                  3
3
3 SECENEGINIZI GIRINIZ.                3
3
3
3
@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY

```

This screen will be seen, when the second alternative in the main menu is chosen.

```

MI GIRINIZ:KAAN
YADI GIRINIZ:GUVEN

```

If we choose second alternative in the menu which is above, the computer will ask us that, what is the name and last name of the student whose record will be changed.

When third alternative in the main menu, the computer will ask you that "THIS ALTERNATIVE WILL DELETE THE RECORDS. DO YOU WANT TO GO ON WITH THIS ALTERNATIVE ? Y/N E/H

BU SECENEK KAYITLARI SILECEKTIR. DEVAM ETMEK ISTIYORMUSUNUZ? (E) EYET / (H) HAYIR

```

#####
:
:
:
: SECENEGİNİZİ GİRİNİZ. 3
:
: <6> CIKIS
: <5> BAZI İSTATİSTİKLER
: <4> VERİ TARAMA
: <3> VERİ SİLME
: <2> VERİ DEĞİSTİRME
: <1> VERİ GİRME
:
:
:
: ANA TABLO
#####

```

This is the record of Kaan Guven in the kimik file. We can change any information on it.

```

#####
Record No. 10
# CURSOR <--> # UP DOWN # DELETE # Insert Mode: Ins #
# Char: Del # Exit/Save: ~End #
# Word: Home End # Page: PgUp PgDn # Field: ~Y # Abort: Esc #
# Help: F1 # Record: ~U # Memo: ~Home #
#####
İSİM KAAN
SOYAD GUVEN
DOĞUMYERİ İZMİR
DOĞUM TARİHİ 11/24/71
T.C.
SINIVYETİ E
AKULTE MEBF
KOLUM FİZİK
SINIF 2
TARİH 09/05/88
XXXXXXXXXX

```

```

#####
#                                     #
#           SILINICEK DOSYAYI SECINIZ           #
#                                     #
#                                     #
#                                     #
#                                     #
#   <1>                                YURT BILGILERI   #
#   <2>                                KIMLIK BILGILERI  #
#   <3>                                NUFUS BILGILERI   #
#   <4>                                EGITIM DURUMU     #
#   <5>                                AILE BILGILERI    #
#   <6>                                HEPSINI SIL      #
#   <7>                                CIKIS             #
#                                     #
#                                     #
#   SECENEGINIZI GIRINIZ..                   #
#                                     #
#                                     #
#####

```

When you say Yes (Evet) to the above question, this screen will show up.

ISMI GIRINIZ:SAMI
SOYADI GIRINIZ:AYYORGUN
BU KAYIT SILINMISTIR.

When we choose one of the above alternatives, the computer will ask us the name and the last name of the student whose record or records will be deleted.

This screen will show up, when we choose fourth alternative in the main menu.

When we choose first alternative which is Ekranda Gor (Display on the screen), this menu will be seen on the screen.

SMI GIRINIZ:KAAN
DYADINI GIRINIZ:GUVEN

If Tek Tek Gor alternative is chosen, the computer will ask the name and the last name of the student.

```

#####
# GORMEK/BASMAK ISTEDIGINIZ DOSYAYI SECINIZ #
#                                             #
#                                             #
#                                             #
# <1>          YURT BILGILERI                #
# <2>          KIMLIK BILGILERI              #
# <3>          NUFUS BILGILERI              #
# <4>          EGITIM DURUMU                #
# <5>          AILE BILGILERI               #
# <6>          CIKIS                        #
#                                             #
#                                             #
#                                             #
# SECENEGINIZI GIRINIZ.                   #
#                                             #
#####

```

After giving the name and the last name of the student, this menu will show up. We can choose one of the above files and see the related record.

○

This is an example where the aile file is chosen and the record of Kaan Guven is seen.

This is the screen that we will be faced with when we choose Hepsini Gor (Display all) alternative.

This is the screen of Bolumlerine Gore alternative in the previous menu.

When you choose Tek Tek Bolumleri Gor alternative in the above, the computer will directly ask you the name of the department that you want to see. And give you the number of students belonging to that department.

RUN SAVASCI
MANYA 11/27/72
C. E
BF ELEKTRIK MUH. 1
/09/89

EVAMINI GOR (E) EVET / (H) HAYIR E
EVAMINI GOR (E) EVET / (H) HAYIR E
ILA ABDULKADIROGLU
NYA 08/01/69
C. E
BF ELEKTRIK MUH. 2
/ /

EVAMINI GOR (E) EVET / (H) HAYIR E
PARSLAN ARSLAN
KARA 05/01/72
C. E
BF ELEKTRIK MUH. 2
/07/89

EVAMINI GOR (E) EVET / (H) HAYIR

These are the students of Elektrik Muh. Department.


```

#####
#                SINIF SECENEKLERI                #
#                                                    #
#                                                    #
#                                                    #
#   <H>          TOPLAM HAZIRLIK OGRENCILERI      #
#   <1>          TOPLAM BIRINCI SINIF OGRENCILERI #
#   <2>          TOPLAM IKINCI SINIF OGRENCILERI  #
#   <3>          TOPLAM UCUNCU SINIF OGRENCILERI  #
#   <4>          TOPLAM DORDUNCU SINIF OGRENCILERI#
#   <5>          TOPLAM MASTER OGRENCILERI       #
#                                                    #
#   HANGI SINIFI GORMEK ISTIYORSANIZ GIRINIZ.    #
#                                                    #
#                                                    #
#                                                    #
#####

```

When fourth alternative is chosen, this menu will show up and ask us which class do we want to see.

[illegible]

```

: BU SINIFDAKI TOPLAM OGRENCI SAYISI = 13
Press any key to continue...

```

Here is the number of students of the second class.

APPENDIX I

1. GİRİŞ

Bu kılavuzda menülerin ekrana hangi sırayla geldiğini ve hangi işleri yaptığını göreceğiz.

2. MENÜLER

Bu program birbirini takip eden oniki menüden oluşmuştur. Bu menülerin şemasını Şekil 1 & 2 de görebilirsiniz. Menülerdeki herhangi bir seçeneği seçmek için karşısındaki rakamı girmek yeterli olacaktır. Bütün menülerde son seçenek Çıkış seçeneğidir. Çıkışla bir önceki menüye dönülür.

2.1 ANATABLO

Programı başlatmak için "DO ANATABLO" komutunu vermemiz şarttır. Bu komuttan sonra ilk menü olan Anatablo ekranda gözükecektir. Bu menüde seçebileceğiniz altı değişik seçenek vardır. Bu seçenekler:

1. Veri Girme
2. Veri Değiştirme
3. Veri Silme
4. Veri Tarama
5. Bazı İstatistikler
6. Çıkış

2.2 Veri Girme

Anatablodaki ilk seçenek bize veri girmede yardımcı olacaktır. Bu seçenek seçildiği zaman yeni bir menü gözükecek ve bizden girilecek dosyayı seçmemizi

isteyecektir. Aşağıdakiler veri girilebilecek dosyalardır.

1. Yurt Bilgileri
2. Kimlik Bilgileri
3. Nüfus Bilgileri
4. Eğitim Durumu
5. Aile Bilgileri

2.2.1 Seçenek 1

Yukarıdaki birinci seçenek içeriği Tablo 1 de izah edilen Yurt dosyasını kullanacak ve bu içerik doğrultusunda veri girmemize izin verecektir.

2.2.2 Seçenek 2

İkinci seçenek, Tablo 2 de gösterilen Kimlik dosyasını kullanacaktır.

2.2.3 Seçenek 3

Burada Nüfus dosyası kullanılacaktır. Nüfus dosyasının içeriği Tablo 3 de verilmiştir.

2.2.4 Seçenek 4

Eğitim dosyası dördüncü seçeneğin kullanacağı dosyadır. Bu dosyanın içeriği Tablo 4 dedir.

2.2.5 Seçenek 5

Bu seçenekte program Aile dosyasını kullanacak ve Tablo 5 te belirtilen içerik doğrultusunda veri girilmesine izin verecektir.

2.3 Veri Değiştirme

Anatablodaki ikinci seçenek, Veri değiştirmemize

yardım edecektir. Gene veri dırmede olduğu gibi. bize veri değıştirilecek dosyayı soracaktır. Buradaki tek farklılık bize verisinin dedistirileceđi kisinin sorulacak olmasıdır.

2.4 Veri Silme

Anatablodaki üçüncü secenek, veri silmemize yardımcı olacaktır. Veri silmek bir dana düzeltilemeyeceğinden, program ilk başta size emin olup olmadığınızı soracaktır. Bu soruya Evet cevabı verirseniz işleme devam edecek ve veri silinecek kişinin ismini, soyadını ve hangi dosyanın silineceğini bize soracaktır.

2.5 Veri Tarama

Anatablonun bize sunduđu dördüncü secenek veri taramamıza yardımcı olacaktır. Bu secenek seçildiğinde karşımıza çıkacak ilk menü bize iki secenek sunacaktır. Bunlar:

1. Ekranda Gör
2. Printer'a Yolla

İsimlerindenede anlaşılabilineceđi gibi, bu seçenekler, bizim ya verileri sadece ekranda görmemizi yada aynı zamanda basmamızı sağlamaktadır. Bu seçeneklerin hangisini secersek seçelim karşılaşacağımız ikinci menü aynı olacaktır.

2.5.1 Tarama Seçenekleri

Bu menü yukarda sözü edilen ikinci menüdür. Ve bize yine iki seçenek sunar. Bu seçenekler:

1. Tek Tek Gör\Bas
2. Hepsini Gör\Bas

Ekranda görme ve basmak için, program Tablo 6, 7, 8, 9, ve 10 da gösterilen formatları kullanacaktır.

Yukarıdaki ilk secenek secildiğinde program bize görmek\basmak istediğimiz öğrencinin adını ve soyadını soracaktır. Bunları girdikten sonra bize hangi dosyayı görmek yada basmak istediğimizi soracak ve bu dosyayı görmemizi \ basmamızı sağlayacaktır.

İkinci secenek bizeg beş kriteriya sunacaktır. Bu kriteriyalar şunlardır:

1. Bölümlerine Göre Gör\Bas
2. İsimlerine Göre Gör\Bas
3. Soyadlarına Göre Gör\Bas
4. Yurt Numaralarına Göre Gör\Bas
5. Giriş Tarihlerine Göre Gör\Bas

İlk kriteriya seçildiğinde, program kimlik dosyasını kullanacak ve yeni bir menü sunacaktır. Bu menüde yine iki secenek vardır. Bunları:

1. Tek Tek Bölümleri Gör\Bas
2. Bütün Bölümleri Gör\Bas

İlk secenekte bölüm adı girilir ve bu bölümde okuyan tüm öğrenciler görülebilir.

İkinci secenek bütün bölümleri alfabetik sırada gösterir\basar.

İkinci, üçüncü, dördüncü, ve beşinci kriteriyalar Yurt dosyasını kullanırlar.

Bütün bu kriteryalarda, bilgisayar bize aşağıdaki soruyu İngilizce olarak soracaktır.

INDEX FILE IS ALREADY EXIST, OVERWRITE IT (Y) (N)

Bu soruya daima Y cevabı girilmelidir.

2.6 Bazı İstatistikler

Anatablodaki 5inci seçenek bize yeni bir menü sunacaktır. Bu menüde bazı istatistik bilgileri bulunabilir. Bunlar:

1. Erkek Öğrencilerin Sayısı
2. Kız Öğrencilerin Sayısı
3. Yabancı Öğrencilerin Sayısı
4. Öğrencilerin Sınıflara Göre Dağılımı

İlk üç seçenekte, kriteryalara uyan öğrenci sayısı direk olarak ekranda görülecektir.

Dördüncü seçenek bize hangi sınıfın öğrencilerini görmek istediğimizi sorduktan sonra öğrenci sayısını verecektir.

TABLO 1

YURT DOSYASININ YAPISI

BÖLÜM	BÖLÜM ADI	TİP	GENİŞLİK
1	İSİM	KARAKTER	30
2	SOYAD	KARAKTER	15
3	YURD	KARAKTER	8
4	GİRİŞ	TARİH	8
5	BLOK	RAKAM	2
6	ODA	RAKAM	3
7	YATAK	RAKAM	1
8	TELEFON	RAKAM	4
9	YURTNO	RAKAM	8
10	SİCİL	KARAKTER	70
11	XXXXXXXX	KARAKTER	10

XXXXXXXX: Ek olarak verilmiş olan bu bölüm, sonradan ortaya çıkabilecek bir kriteriya için ayrılmıştır.

TABLO 2

KİMLİK DOSYASININ YAPISI

BÖLÜM	BÖLÜM ADI	TİP	GENİŞLİK
1	İSİM	KARAKTER	30
2	SOYAD	KARAKTER	15
3	DOĞUM YERİ	KARAKTER	15
4	DOĞUM TARİHİ	TARİH	8
5	UYRUGU	KARAKTER	8
6	CİNSİYETİ	KARAKTER	1
7	FAKULTE	KARAKTER	5
8	BÖLÜM	KARAKTER	15
9	SINIF	RAKAM	1
10	GİRİŞ TARİHİ	TARİH	8
11	XXXXXXXX	KARAKTER	10

TABLO 3

NUFUS DOSYASININ YAPISI

BÖLÜM	BÖLÜM ADI	TIP	GENİŞLİK
1	İSİM	KARAKTER	30
2	SOYAD	KARAKTER	15
3	İL	KARAKTER	12
4	İLCE	KARAKTER	15
5	BUCAK	KARAKTER	15
6	MAHALLE	KARAKTER	15
7	HANE	KARAKTER	6
8	CİLT	KARAKTER	7
9	SAYFA	RAKAM	3
10	MEDENİ HALİ	KARAKTER	5
11	DİNİ	KARAKTER	3
12	PASAPORT NO	RAKAM	8
13	İKATESNO	RAKAM	8
14	XXXXXXXXX	KARAKTER	10

TABLO 4

EGİTİM DOSYASININ YAPISI

BÖLÜM	BÖLÜM ADI	TİP	GENİŞLİK
1	İSİM	KARAKTER	30
2	SOYAD	KARAKTER	15
3	FAKULTE	KARAKTER	6
4	BÖLÜM	KARAKTER	15
5	SINIF	RAKAM	1
6	LİSE	KARAKTER	15
7	KOLU	KARAKTER	10
8	DERECESİ	RAKAM	3
9	BURS	KARAKTER	15
10	YABANCI DİL	KARAKTER	20
11	XXXXXXXX	KARAKTER	10

TABLO 5

AİLE DOSYASININ YAPISI

BÖLÜM -----	BÖLÜM ADI	TİP	GENİŞLİK
1	ADI	KARAKTER	30
2	SOYAD	KARAKTER	15
3	ANKARA ADRESİ	KARAKTER	40
4	VELİ ADI	KARAKTER	15
5	VELİ SOYADI	KARAKTER	15
6	İŞİ	KARAKTER	15
7	İŞ ADRESİ	KARAKTER	40
8	İKAMET ADRESİ	KARAKTER	40
9	BABA ADI	KARAKTER	15
10	BABANIN İŞİ	KARAKTER	15
11	BABANIN ADRESİ	KARAKTER	40
12	ANNE ADI	KARAKTER	15
13	ANNENİN İŞİ	KARAKTER	15
14	ANNENİN ADRESİ	KARAKTER	40
15	KARDEŞ ADI 1	KARAKTER	15
16	KARDEŞ ADI 2	KARAKTER	15
17	KARDEŞ ADI 3	KARAKTER	15
18	ACİL ADRES	KARAKTER	40
19	ACİL TELEFON	RAKAM	9
20	XXXXXXXXXX	KARAKTER	10

TABLO 6

YURLBL DOSYASININ FORMATI

SATIR ICERIGI

1: ISIM, SOYAD
 2: YURT, YURT NO
 3: GİRİŞ
 4: BLOK, ODA, YATAK
 5: TELEFON
 6: XXXXXXXXX

TABLO 7

KIMLIKLBL DOSYASININ FORMATI

SATIR ICERIGI

1: ISIM, SOYAD
 2: DOĞUM YERİ, DOĞUM TARİHİ
 3: UYRUGU, CİNSİYETİ
 4: FAKÜLTE, BÖLÜM, SINIF
 5: GİRİŞ TARİHİ
 6: XXXXXXXXX

TABLO 8

NUFUSLBL DOSYASININ FORMATI

SATIR ICERIGI

1:	ISIM, SOYAD
2:	IL, ILCE
3:	BUCAK, MAHALLE
4:	HANE, CILT, SAYFA
5:	MEDENI HALI, DINI
6:	PASAPORT NO, IKATES NO
7:	XXXXXXXX

TABLO 9

EGITIMLBL DOSYASININ FORMATI

SATIR ICERIGI

1:	ISIM, SOYAD
2:	FAKOLTE, BOLUM, SINIF
3:	LISE, KULU
4:	DERECESI
5:	BURS
6:	YABANCI DIL
7:	XXXXXXXXXX

TABLO 10

AİLELBL DÜSYASININ FORMATI

SATIR İCERİĞİ

1:	ADI, SOYAD
2:	ANKARA ADRESİ
3:	VELİ ADI, VELİ SOYADI
4:	VELİ İSİ, İS ADRESİ
5:	İKAMEİGAH ADRESİ
6:	BABA ADI, BABANIN İSİ
7:	BABANIN ADRESİ
8:	ANNE ADI, ANNENİN İSİ
9:	ANNENİN ADRESİ
10:	KARDES ADLARI
11:	ACİL ADRES, ACİL TELEFON
12:	XXXXXXXXXX