

**NEAR EAST UNIVERSITY**



**Faculty of Engineering**

**Department of Computer Engineering**

**STUDENTS REGISTRATION SYSTEM USING  
DELPHI PROGRAMMING**

**Graduation Project  
COM-400**

**Student: Ismael Zaky Elhamss (20011006)**

**Supervisor: Assoc.Prof.Dr.Rahib Abiyev**

**Nicosia- 2005**

**NEAR EAST UNIVERSITY**



**Faculty of Engineering**

**Department of Computer Engineering**

**STUDENTS REGISTRATION SYSTEM USING  
DELPHI PROGRAMMING**

**Graduation Project  
COM-400**

**Student: Ismael Zaky Elhamss (20011006)**

**Supervisor: Assoc.Prof.Dr.Rahib Abiyev**

**Nicosia- 2005**

## ACKNOWLEDGEMENT

*"I would like and foremost to thank Allah whom its accomplishment would not have been possible.*

*I would like to thank my father and my mother for giving me the chance to complete my academic study, supporting me and giving me the opportunity to achieve my goal in life. Especially I would to thank brothers Mohammed and Ahmed for supporting me.*

*I would like be honored to direct my thank to my supervisor Assoc.prof. Dr: Rahib abiyev for being so co-operative and for his advices during me preparation to the graduation project.*

*I would also like to thank all of my friends they were always available for my assistance throughout this project."*

## ABSTRACT

This program is designed for register new students in university and it's designed for regular students. This program is based on BORLAND DELPHI 7 programming language. All the criterions of this program are taken according upon the request of the Near East University Registration Office of Faculty of Engineering and all the features of this software can be adjusted according to the desire of the Faculty Staff students.

All the screens that will appear in the usage of are program will be illustrated in the coming chapters in details.

The Registration, Updating, Searching, Records, Report, and Help will be briefly discussed in the third chapter. This program is designed to help Near East University students and staff to do easy and sufficient registration for the courses.

## TABLE OF CONTENTS

ACKNOWLEDGMENT .....	i
ABSTRACT .....	ii
TABLE OF CONTENTS .....	iii
INTRODUCTION.....	v
CHAPTER 1 .....	1
INTEGRATED DEVELOPMENT ENVIRONMENT (IDE).....	1
1.1 INTRODUCTION .....	1
1.2 The Delphi IDE .....	1
1.3 A Quick Look at the Delphi IDE .....	2
1.4 The Delphi Workspace .....	3
1.5 The Delphi Main Menu and Toolbar. ....	3
1.6 Using the Component Palette.....	5
1.7 Placing Multiple Copies of a Component .....	5
1.8 About Delphi Forms .....	6
1.8.1 Main Window Forms.....	6
1.8.2 Creating the Main Window Form.....	6
1.8.3 Dialog Box Forms.....	8
1.8.4 Creating a Dialog Form.....	9
1.9 A Multiple-Form Application .....	9
1.9.1 Adding Units.....	10
1.9.2 Some Key Properties for Forms .....	10
1.10 The Object Inspector .....	11
1.10.1 The Component Selector .....	11
1.10.2 The Properties Page.....	12
1.10.3 The Events Page .....	13
1.11 Code Templates.....	13
1.12. Writing Code for the Window Menu .....	14
CHAPTER 2 .....	15
(DATABASE ACCESS) .....	15
2.1. Microsoft Access Description.....	15
2.2. Starting Microsoft Access.....	16
2.3. Creating New, and Opening Existing Databases .....	17
2.4. Create a database using the Database Wizard.....	18
2.5. Create a database without using the Database Wizard.....	18
2.6. Tables .....	19
2.7. Create a Table from scratch in Design view.....	20



2.8. Primary Key .....	22
2.9. Switching Views .....	22
2.10. Entering Data .....	23
2.11. Manipulating Data .....	23
2.12. Advanced Table Features w/Microsoft Access.....	24
2.12.1. Assigning a field a specific set of characters.....	24
2.12.2. Formatting a field to look a specific way (HINT) .....	24
2.13. Relationships .....	25
 CHAPTER 3.....	28
TIME TABLE DESIGN .....	28
3.1. Structure of Time Table Design: .....	28
3.2. Define Relationships Between Forms: .....	28
3.3. Define Relationships Between Tables: .....	29
3.4. Delphi database components: .....	30
3.5. Layout of the Application: .....	30
3.5.1. Main menu screen:.....	30
3.5.2. Add New Employee Screen: .....	31
3.5.3. Update Employee Record Screen: .....	33
3.5.4. Add New Course Screen:.....	34
3.5.5. Update Courses Screen: .....	34
3.5.6. Add New Room Screen:.....	35
3.5.7. Time Table Screen: .....	37
 CONCLUSION .....	40
 REFERENCES.....	41
APPENDIX.....	42

## INTRODUCTION

Many daily activities are done in daily Near East University .most important part of these activities is the registration of students for being regular and entering the studying area.

For Register these students and make them regular and entering the studying field I decided to do my project to make a Registration program using PORLAND DELPHI 7, In this language there is special procedure called Database Desktop to create some tables that used in the project.

In the project there are approximately fifteen forms which were used to complete the program. The main menu of the program consists of buttons which is connected to other form. the first form is used for register the students by entering all their information, second for entering the records of the students in all departments, third for updating the information for each student with his Grades and GPA in all semesters.

Each member has been assigned a special form in this program. These forms are updated consistently depending on the kind of registraion that had been made, forth for searching students information, courses, and grades, the fifth one for previewing report for students grades and GPA, viewing the whole courses for both Computer engineering department and Electrical and Electronic Engineering

Department

It's the main idea about my project and it will be briefly discussed later in chapter three, and the codes that are used are listed in the appendix.

For the help of usage of this program Flowcharts were added to this report to clarify the usage, these flowcharts are based on the way that how we will use the program, every screen were assembled by flowchart diagram its briefly discussed in the second chapter.

## **CHAPTER ONE**

### **DATABASE**

#### **1.1. Database Programming**

Delphi's support for database applications is one of the key features of the programming environment. Many programmers spend most of their time writing data-access code, which needs to be the most robust portion of a database application. This chapter provides an overview of Delphi's extensive support for database programming. What you will find here is a discussion of the theory of database design. I am assuming that you already know the fundamentals of database design and have already designed the structure of a database. I will not look into database-specific problems; my goal is to help you understand how Delphi supports database access. I will begin with an explanation of the alternatives Delphi offers in terms of data access, and then I will provide an overview of the database components that I have used in my program. This chapter includes an overview of the TDataSet class, an in-depth analysis of the TField components, and the use of data-aware controls. The following chapters will provide information on more advanced database programming topics, such as client/server programming, the use of dbGo, dbExpress, and InterBase Express

#### **1.2. Database Structure:**

First thing as we know Delphi's support for database applications is one of the key features of the programming environment. Many programmers spend most of their time writing data-access code, which needs to be the most robust portion of a database application. You can create very complex database applications, starting from a blank form or one generated by Delphi's Database form wizard. On a computer, permanent data-including database data is always stored in files. There are several techniques you can use to accomplish this storage. Delphi can use both approaches; or more precisely, it uses a custom approach that works well with both underlying structures. You always refer to a database with its name or an alias, which is a sort of a nickname of a database,



but this reference can be to a database file or to a directory containing files with tables. The approach used by Delphi depends on the database format you are using:

- Paradox and dBASE tables define databases as directories and each table as a separate file or actually multiple files if you include indexes.
- Access, interBase, and most SQL server use a single huge file containing the entire database, with all tables and indexes.

### **1.3. Delphi Database Components**

Delphi includes a number of components related to database. The data access page of the component Palette contains components used to interact to databases. Most of them are non visual components, since they encapsulate database connection, tables, queries, and similar elements. Fortunately, Delphi also provides a number of predefined components you can use to view and edit database data. In the data control page, there are visual components used to view and edit the data in a form. These controls are called data-aware controls. To access a database in Delphi, you generally need a data source, identified by the data source component. The data source component, however, does not indicate the data directly; it refers to a data set component. This can be a table, the result of a query, the result of stored procedure, the data fetched from a remote server, or some other custom data set. As soon as you have placed a table or query component on the form, you can use the dataset property of the data source component to refer to it. For this property, the object inspector lists available data sets of the current form, or of other forms connected with the current one (using the File > Used Form command).

### **1.4. Working with SQL**

Create basic SQL statements that selected data from existing tables. The most fundamental form of the SQL statement is the `SELECTED_FROM` clause. This clause is used to select one or more columns from a table and display the results of in a result set, or view.

Optional clauses that you can add to the `select_Form` clause:

- The WHERE clause: Used to limit the rows in the result set using logical comparisons (for example, WHERE Table1. Name = ' JOUN') and to link two tables in a single, no updatable, view (for example, WHERE Table1. Name=Table2.Name).
- The ORDER BY clause: Used to control the order in which the result set is displayed (for example, ORDER BY Name ASC)
- The GROUP BY clause: Used to create a subtotal result set based on a break column (for example, GROUP BY Name).
  - The HAVING clause: Used only with the GROUP BY clause, the HAVING clause acts as a WHERE clause for the GROUP BY subtotal clause (for example, GROUP BY Name HAVING SUM(Grades Total ) > 65).
- The INNER JOIN clause: Used to join two tables together into a single, updatable result set. The INNER JOIN results rows that have a corresponding match in both tables.
- The LEFT JOIN and RIGHT JOIN: Used to join two tables into a single, updatable set.

The LEFT JOIN includes all records from the first (Left hand) table and all rows from the second table that have a corresponding match the RIGHT JOIN works in reverse.
- The UNION clause: Used to combine two or more complete SQL queries into a single result set(for example, SELECT \* FROM Table1 UNION SELECT \* FROM Table2 ).
- The TRANSFORM\_PIVOT clause: Used to create across- tab query as a result set (for example, TRANSFORM SUM (Credit Value) FROM Credit Table GROUP BY Grades PIVOT Credit).

Additional SQL keywords that you can use to control the contents of the result set:

- BETWEEN \_ AND

- DISTINCT and DISTINCTROW
- AS
- TOP n and TOP n PERCENT
- AVG, COUNT, MAX, MIN, and SUM

### **1.5. Building Database Application:**

Delphi database applications do not have direct access to the data sources that they reference. Delphi interfaces with the Borland Database Engine (BDE), which does have direct access to a number of data sources, including dBASE , Paradox , ASCII , FoxPro, and Access tables. The BDE can also interface with Borland's SQL links, a tool that allows access to a number of local and remote SQL servers. The fact Delphi applications generally don't access data directly but use the BDE basically means that you will need to install the BDE along with your applications on your clients' computers. This is not difficult, since Delphi includes the "Lite" version of an installation program (Install Shield) that can be used to prepare installation disks for the BDE, along with your own application. The BDE files are required your Delphi database applications won't work without them but you can distribute them freely. Delphi now includes a ClientDataset component you can use to access data from an OLE server running on a different computer.

### **1.6. Handling Database Error:**

Another important element of database programming is handling database error in custom ways. Of course, you can let Delphi show an exception message each time a database error occurs, but you might want to try to connect the errors or simply show more details. There are basically three approaches you can use to handle database errors:

- You can wrap a try-except block around risky database operations, such as a call to the open method of a query or the post method of a data set. This is not possible when the operation is generated by the interaction a data aware control.



- You can install a handler for the OnException event of the global application object.
- You can handle specific events of the data sets related to errors, as OnPostError, OnEditError, OnDeleteError, and OnUpdateError.

While most of the exception classes in Delphi simply deliver an error message with database exceptions you see a list of error, showing local BDE error codes and also the native error codes of the SQL server you are connected with.

Besides this error-handling code, the program has a table and a query, along with the error related event handlers. As already mentioned you can install an event handler related to specific errors of a dataset. The three events OnPostError, OnDeleteError, and OnEditError have the same structure. Their handlers receive as parameters the dataset, the error itself, and an action you can request from the system.

### **1.7. Update Databases with SQL:**

To add, delete, and edit data within tables using DML (Data Manipulation Language) SQL keywords by using DML statements you can quickly create test data for tables and load default values into startup tables. DML statements-such as Append queries, Make Table queries, and Delete queries can outperform equivalent Delphi code versions of the same operations.

Managing data within the tables using the following DML keywords:

- The INSERT INTO statement can be used to add new rows to the table using the VALUES clause.
- You can create an Append query by using the INSERT INTO\_FROM syntax to copy data from one table to another. You can also copy data from one database to another using the IN clause on an INSERT INTO\_FROM statement.
- You can create new tables by coping the structure and some of the data using the SELECT INTO statement. This statement can incorporate WHERE, ORDER BY, GROUP BY, and HAVING clauses to limit the scope of the data used to populate the new table you create.
  - You can use DELETE FROM clause to remove one or more records from an existing table. You can even create customized views of the database using the JOIN clause, and remove only records that are the result of a JOIN statement



### 1.8. The Client Dataset Component:

Finally, there is a component derived from `TDataSet` that has a peculiar behavior and can be combined with other data-access components. The `ClientDataSet` component, in fact, is a dataset accessing data kept in memory. The in-memory data can be totally temporary (lost as you exit the program), saved to a local file as a snapshot, and imported by another dataset using a Provider component. This last situation is certainly the most common: You can hook a `ClientDataSet` to any other local dataset, or use Borland's multitier support (discussed in *Accessing a Database: BDE, dbExpress, and other alternatives* "Multitier Database Applications with DataSnap") to retrieve data from a dataset hosted by a different application, possibly running on a separate computer. The `ClientDataSet` component becomes particularly useful if the data-access components you are using provide limited or no caching. This is particularly true of the new dbExpress engine, but can equally help you when using the BDE or other native components.

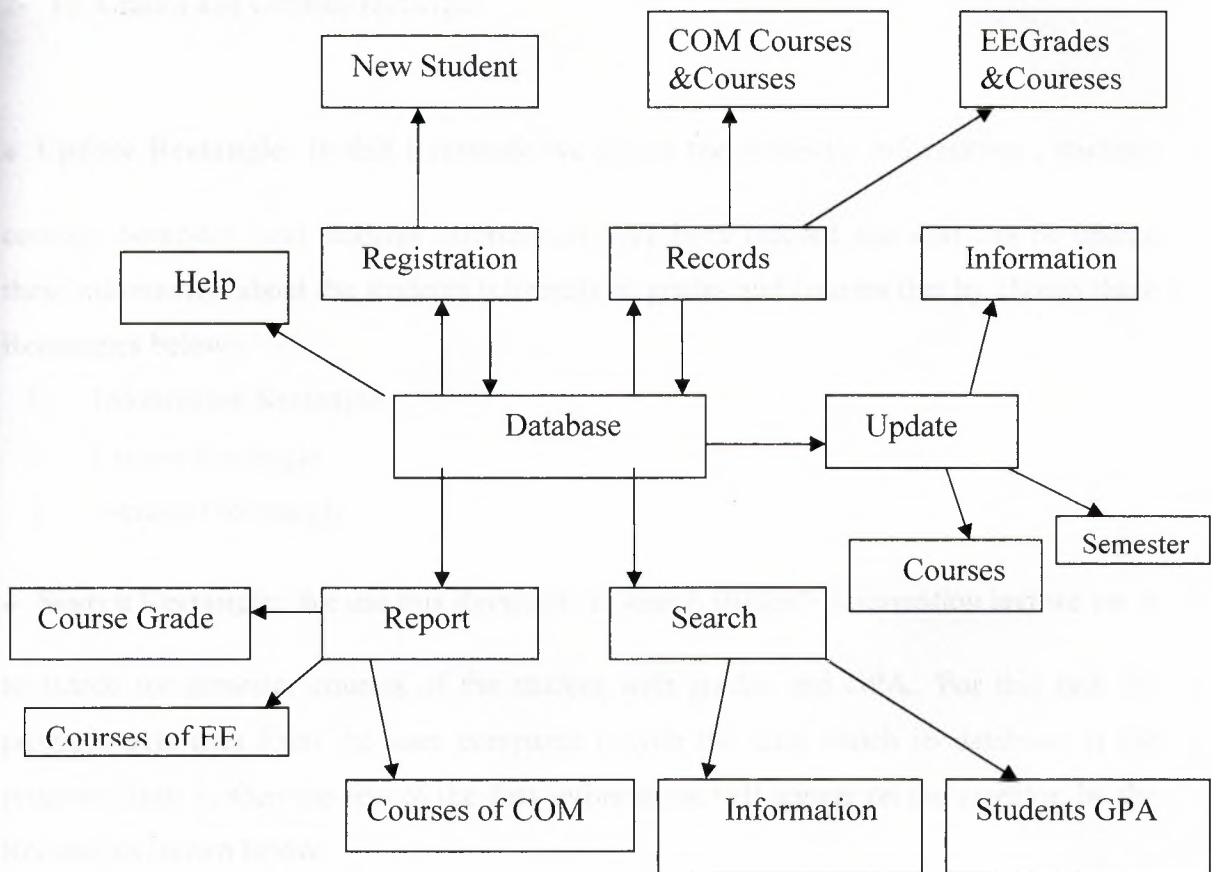
On the other hand, ADO already provides most of the services of the `ClientDataSet` component and using these two at the same time can be useful only in limited situations.

## CHAPTE 2

### FLOWCHART FOR THE PROGRAM OF REGISTRATION STUDETNS

#### 2.1. Main Program Description:

The Main Program internal design is as follows:



**Figure 2.1** Structure of the program

### 2.1.1. Explanation of the Internal Design:

- **Registration Rectangle:** for registration here we can enter student information by going to New Student.

- **Records Rectangle :** its used to enter both Grades and Courses of students ,and its accomplished by the above two rectangles

1- COM Grades and Courses rectangle.

2- EE Grades and Courses rectangle.

- **Update Rectangle:** In this Rectangle we shows the student's information , students courses, Semester and students information have been entered and also can be update these information about the students information, grades and courses that by chosen these Rectangles below:

1- Information Rectangle

2- Course Rectangle

3- Semester Rectangle

- **Search Rectangle:** We use this Rectangle to search student's information and we use it to search for semester courses of the student with grades and GPA.. For this task the program gets data from the user compares it with the data which in database, if the program finds it, then the rest of the data information will appear on the monitor, by the Rectangles shown below:

1- Information Rectangle

2- Student GPA Rectangle

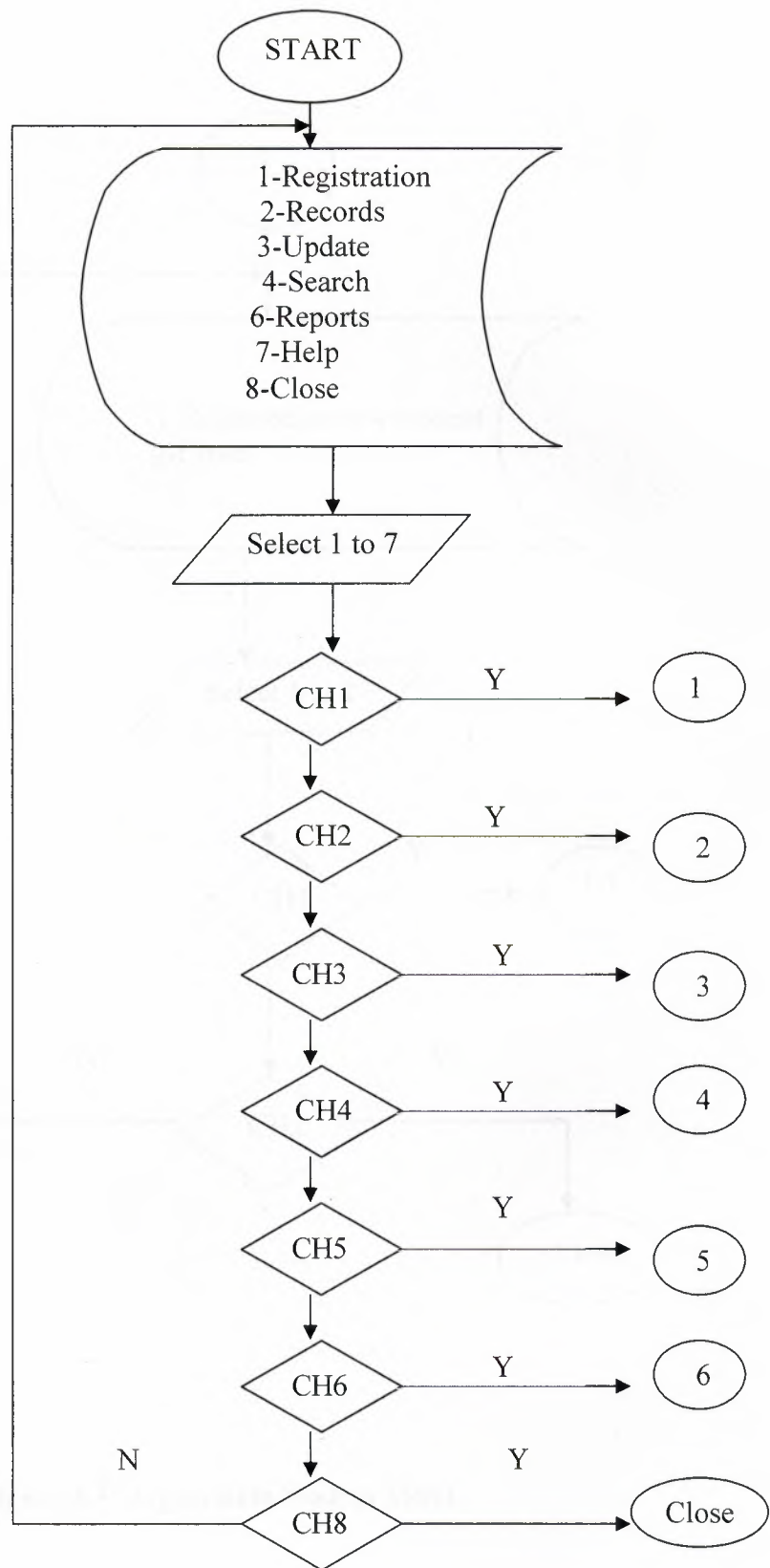
- **Report Rectangle:** In this Rectangle the student's information and students courses with the grades and GPA in the any selected semester is printed, and also we can print

the whole student courses with grades and CGPA and courses of students in the selected semester, these done by chosen the Rectangles shown below:

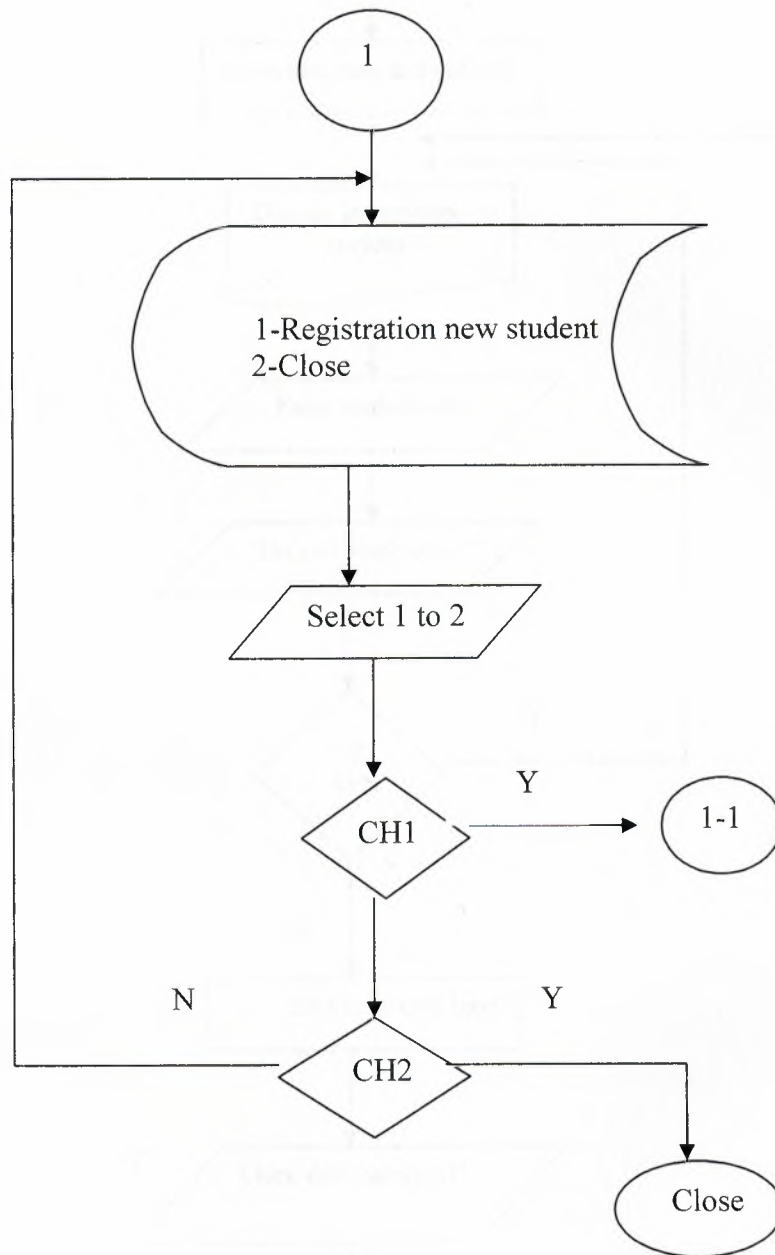
- 1- Semester Courses and Grades for students.
- 2- Whole Course of COM students
- 3- Whole Course of EE students.

• **Help Rectangle:** This Rectangle is assigned to give help and some information about the programmer and help files to show how to use the program

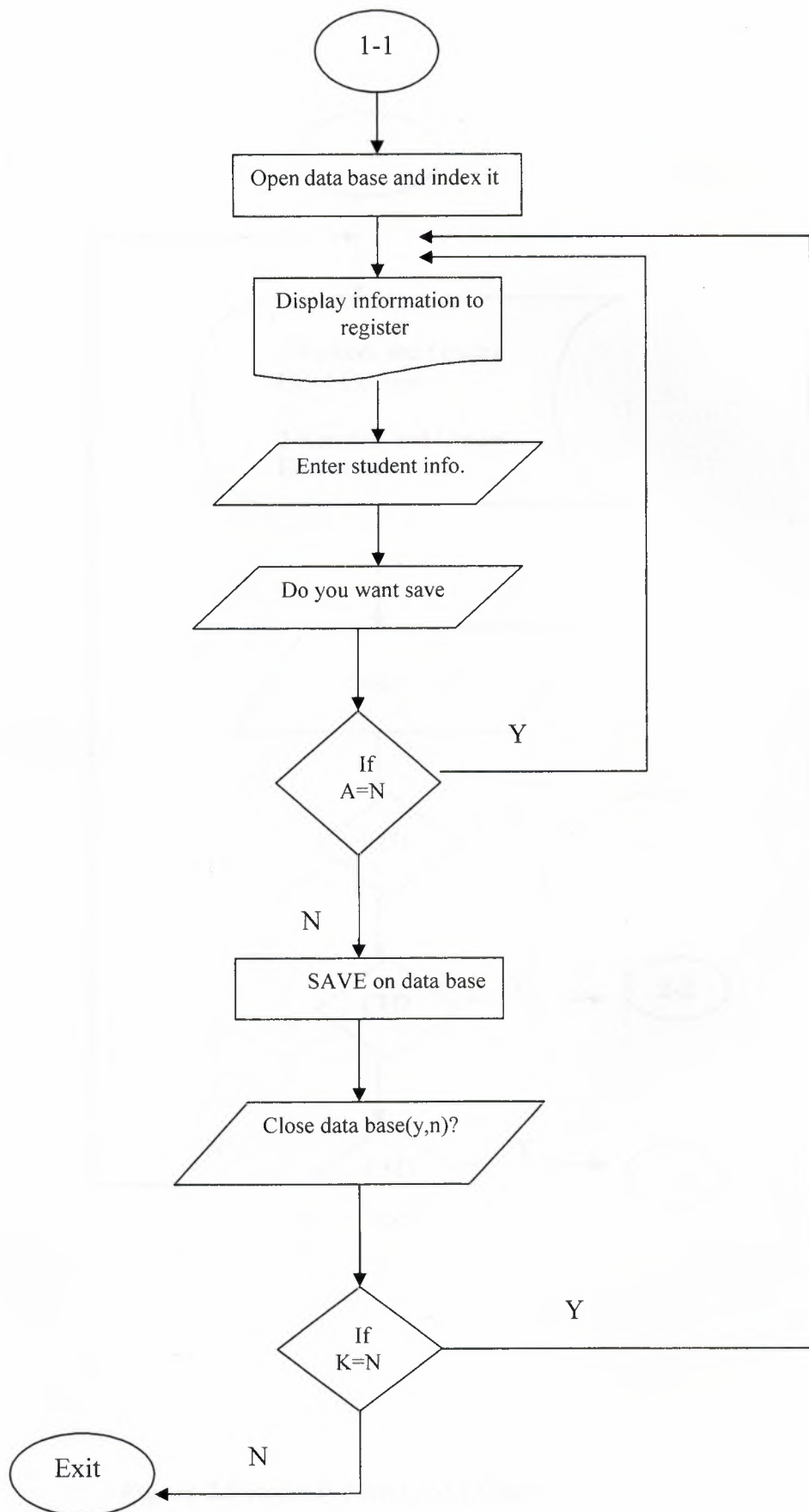




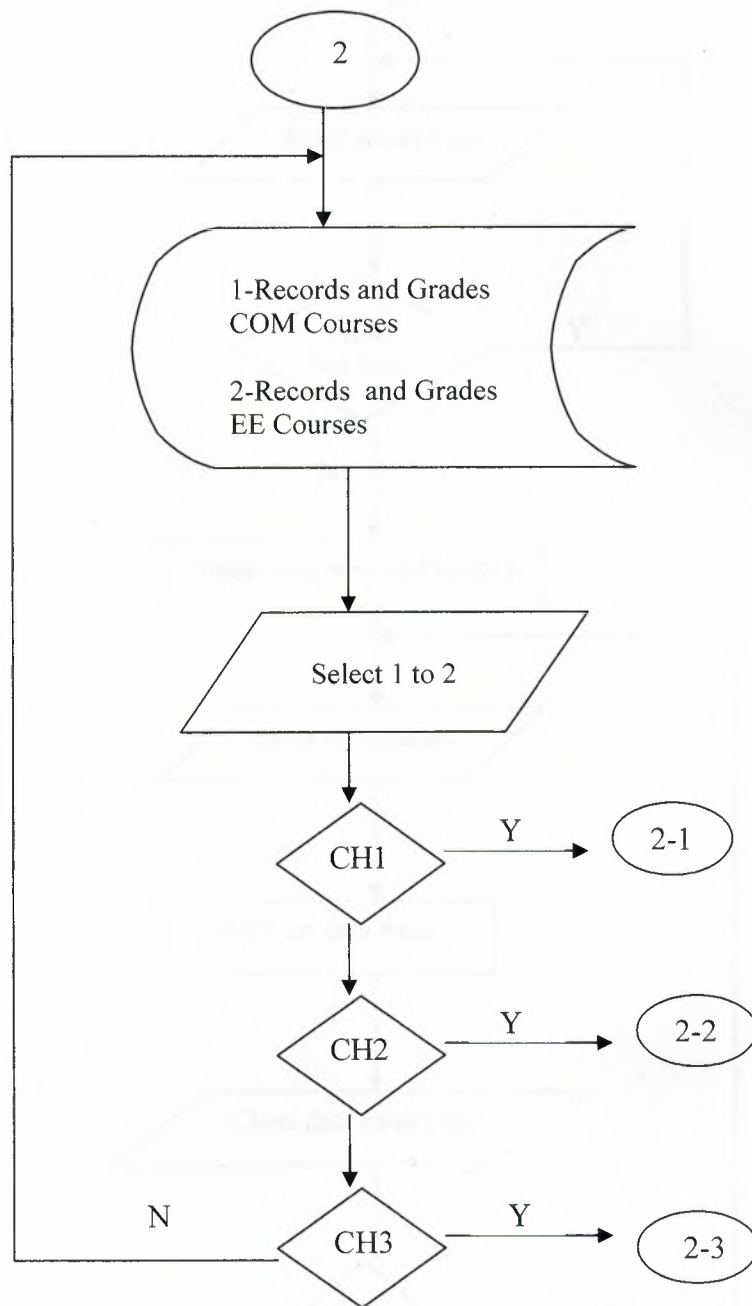
**Figure 2.2** Main menu Flow Chart



**Figure 2.3** Registration Student Menu

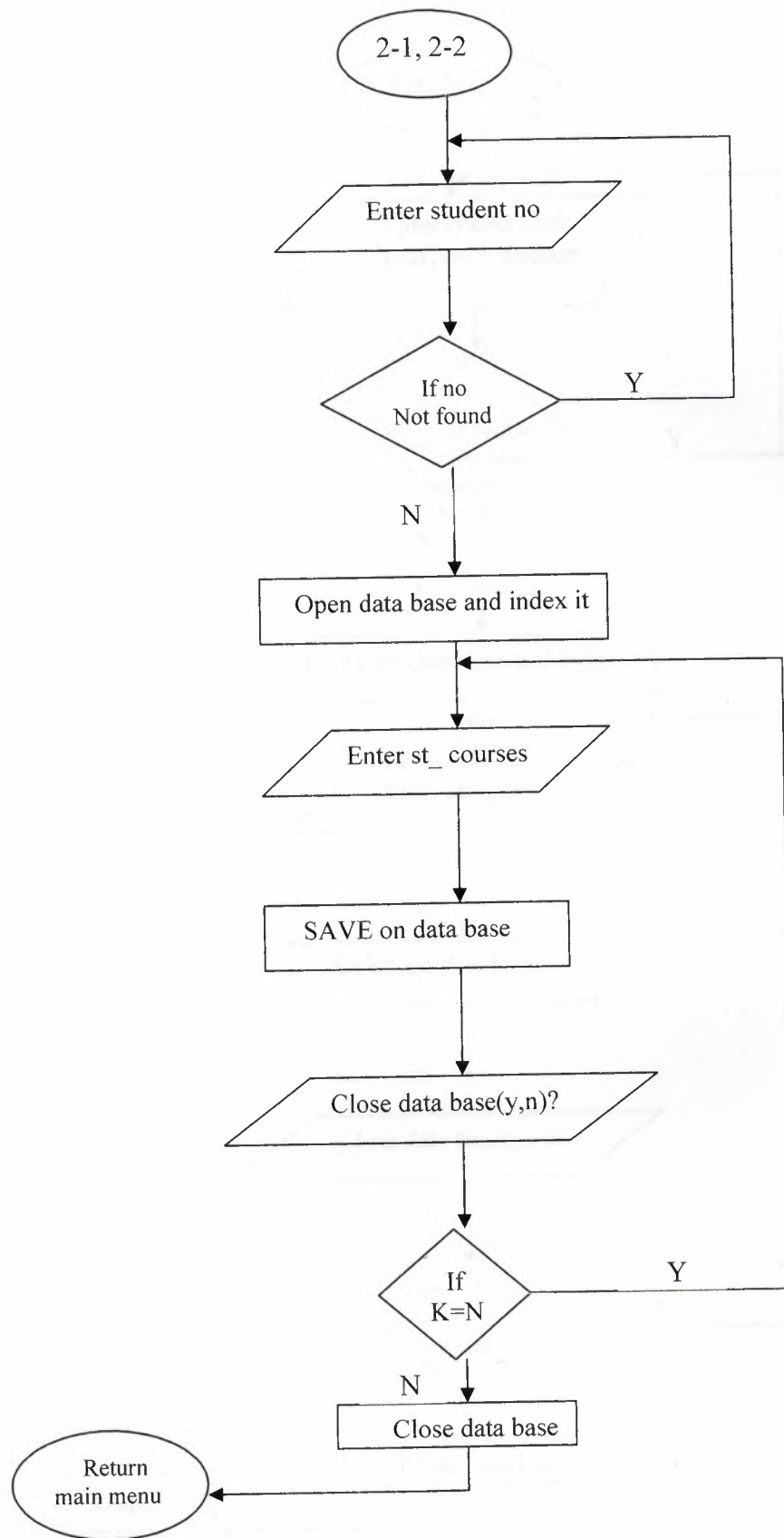


**Figure 2.4** Add New Student Flow Chart

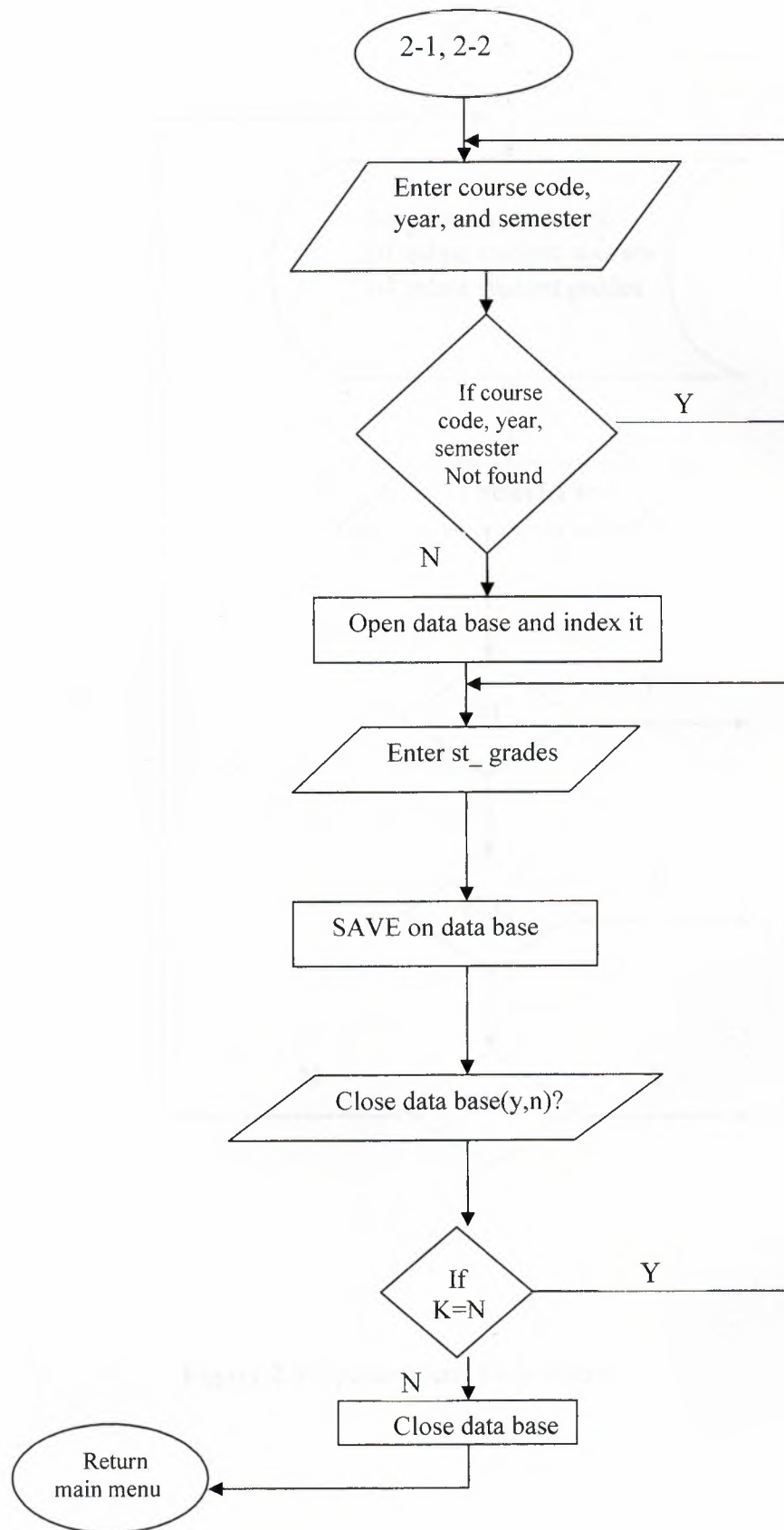


**Figure 2.5** Records Menu Flow Chart

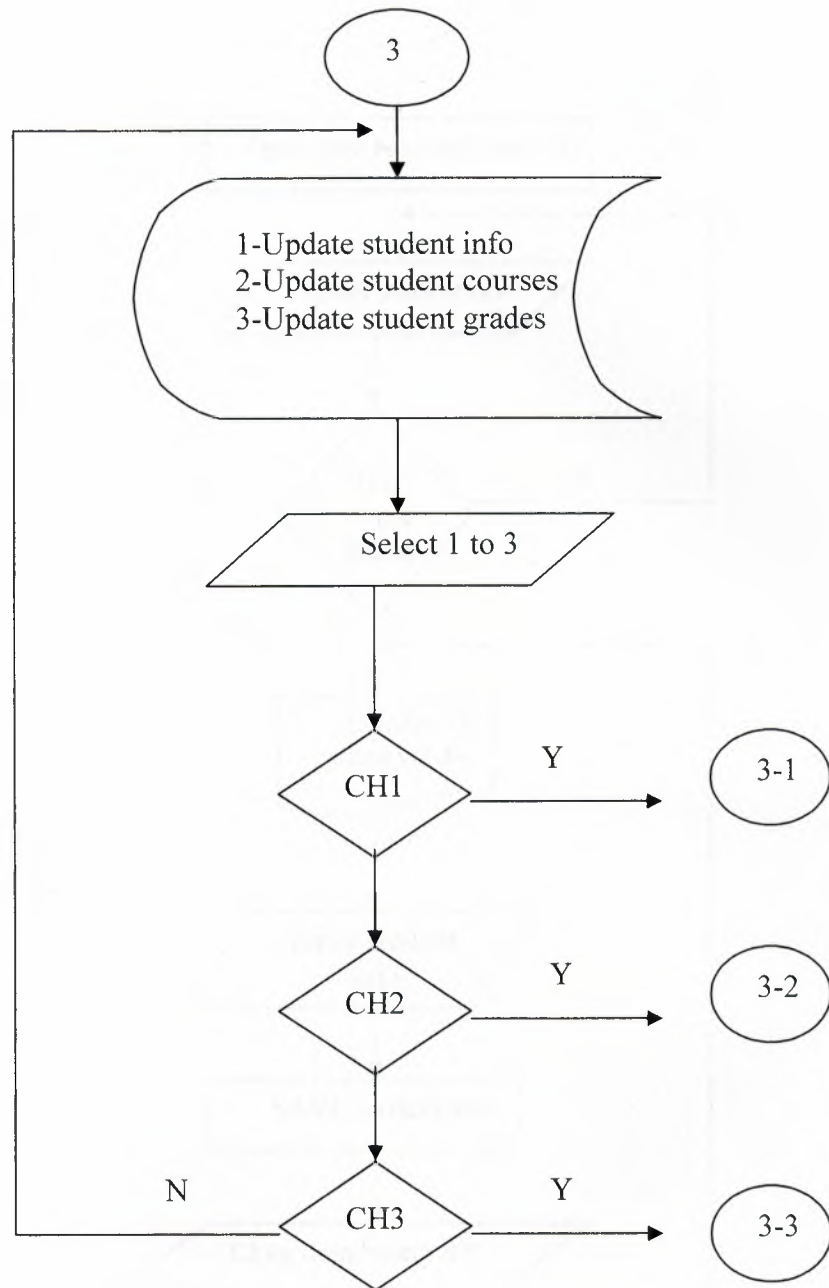




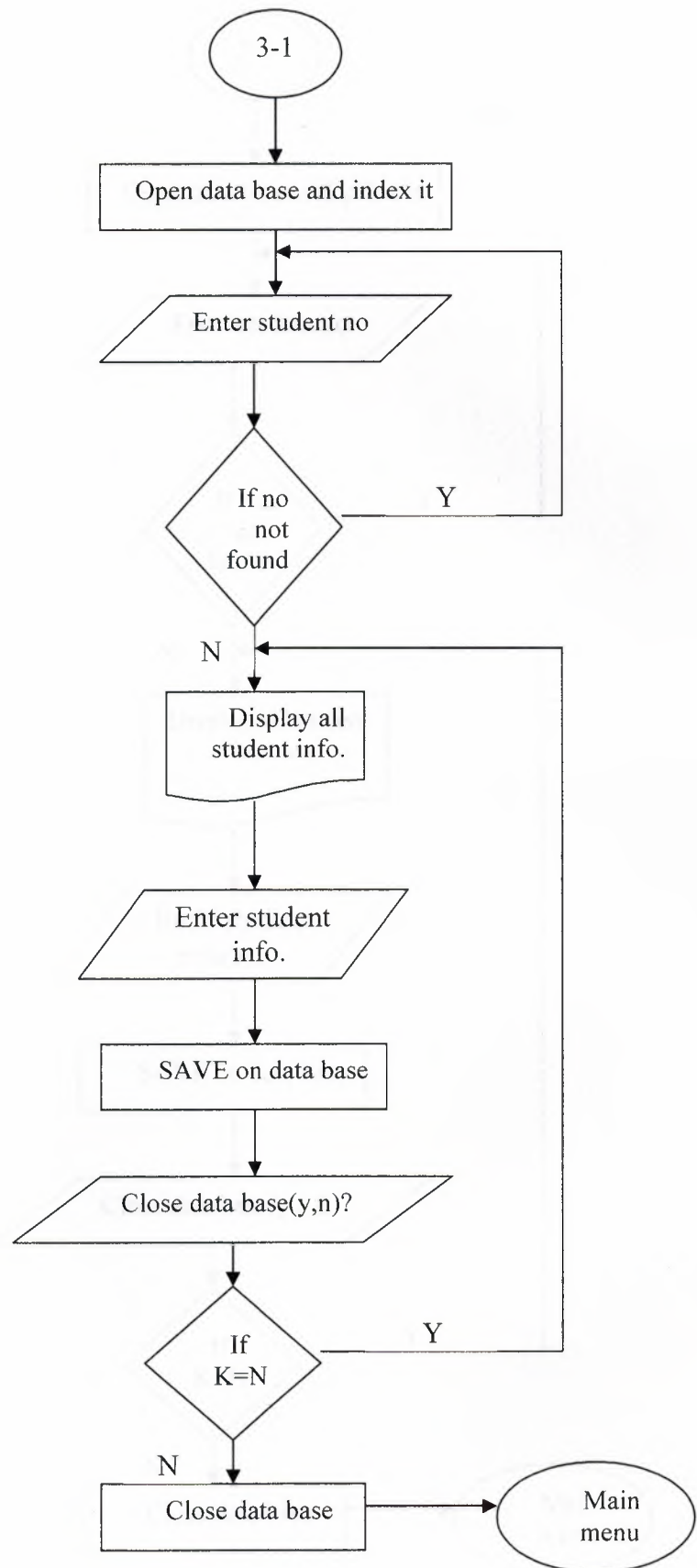
**Figure 2.6** Records Courses Flow Chart



**Figure 2.7** Records Grades Flow Chart

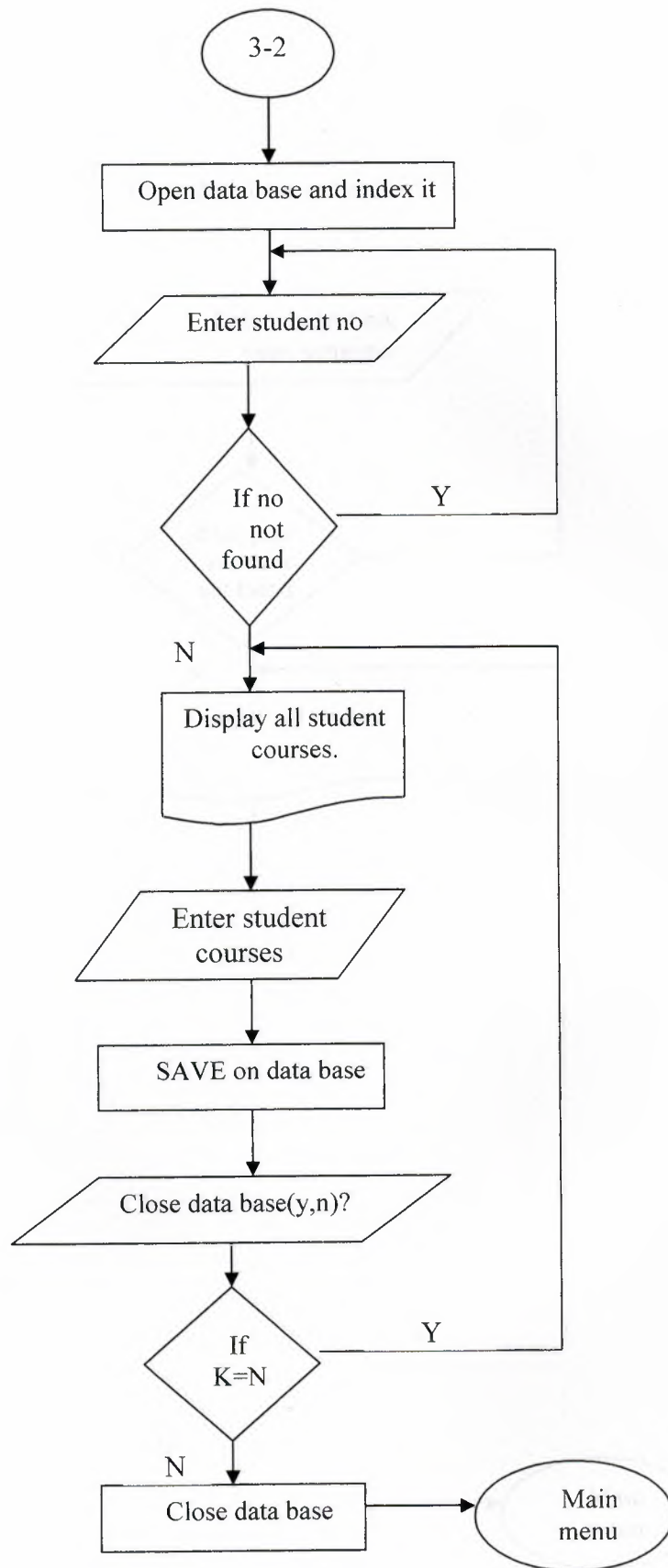


**Figure 2.8** Update Menu Flow Chart



**Figure 2.9** Update Student Info.





**Figure 2.10** Update student Courses

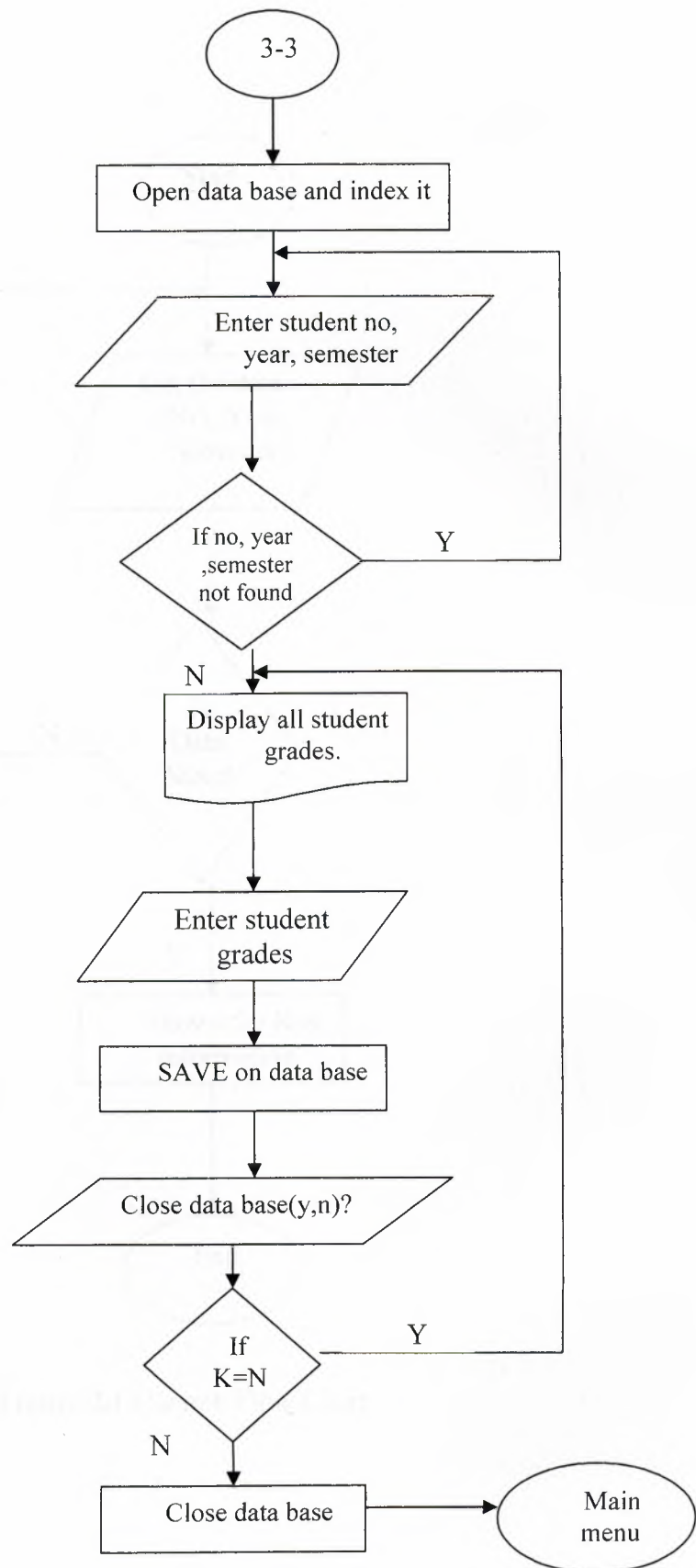
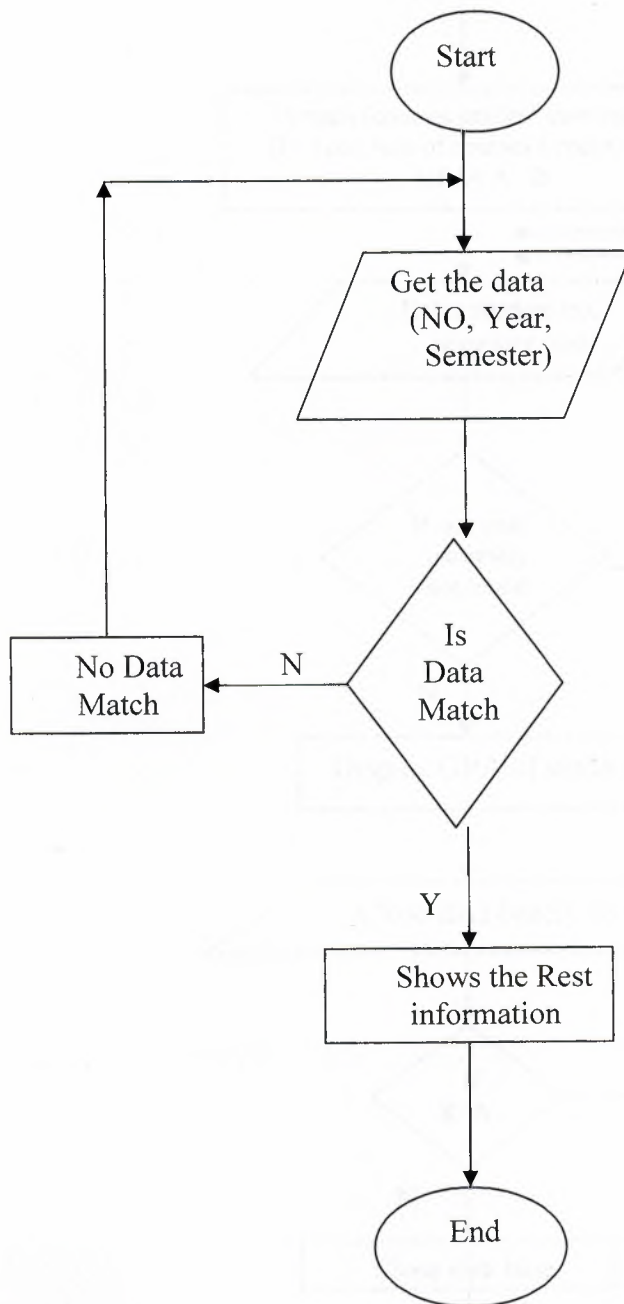
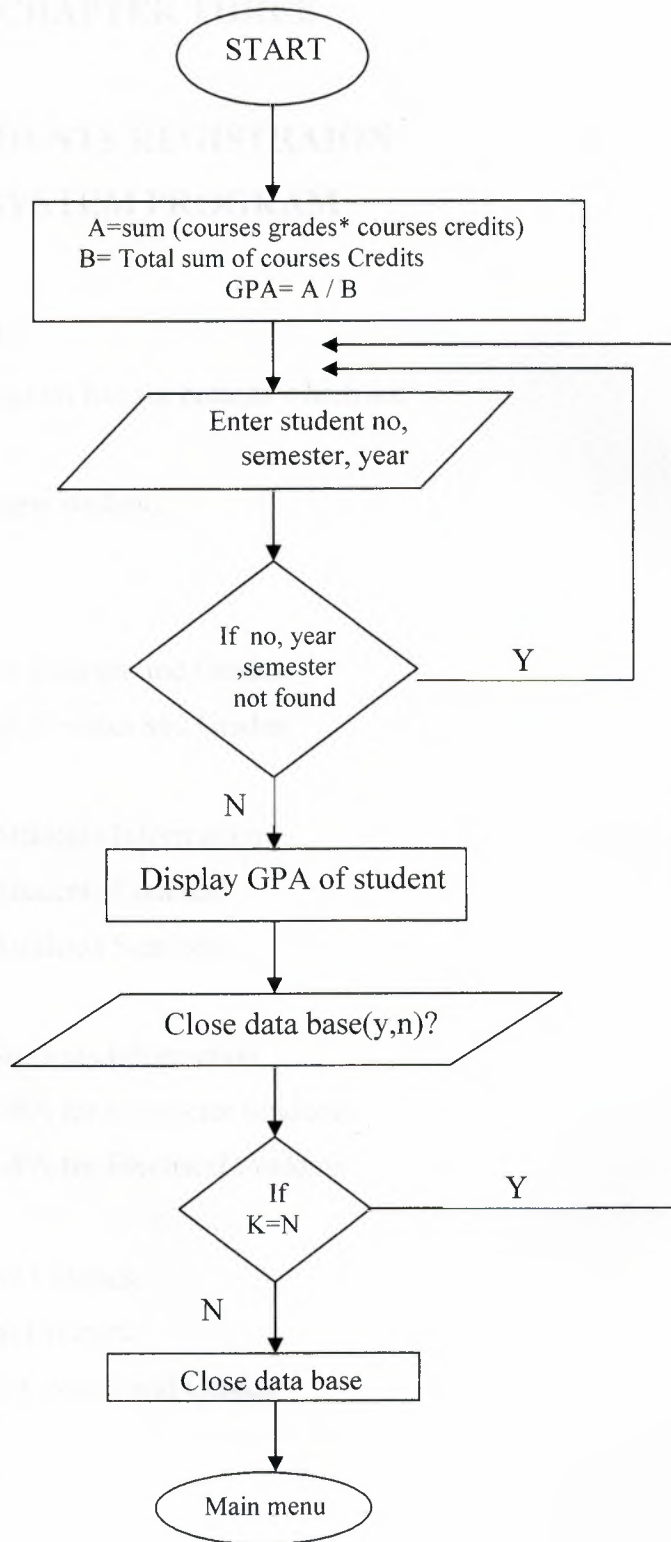


Figure 2.11 Update Student Grades



**Figure 2.13** Search Flow Chart



**Figure 2.14** GPA Calculation Flow Chart

## **CHAPTER THREE**

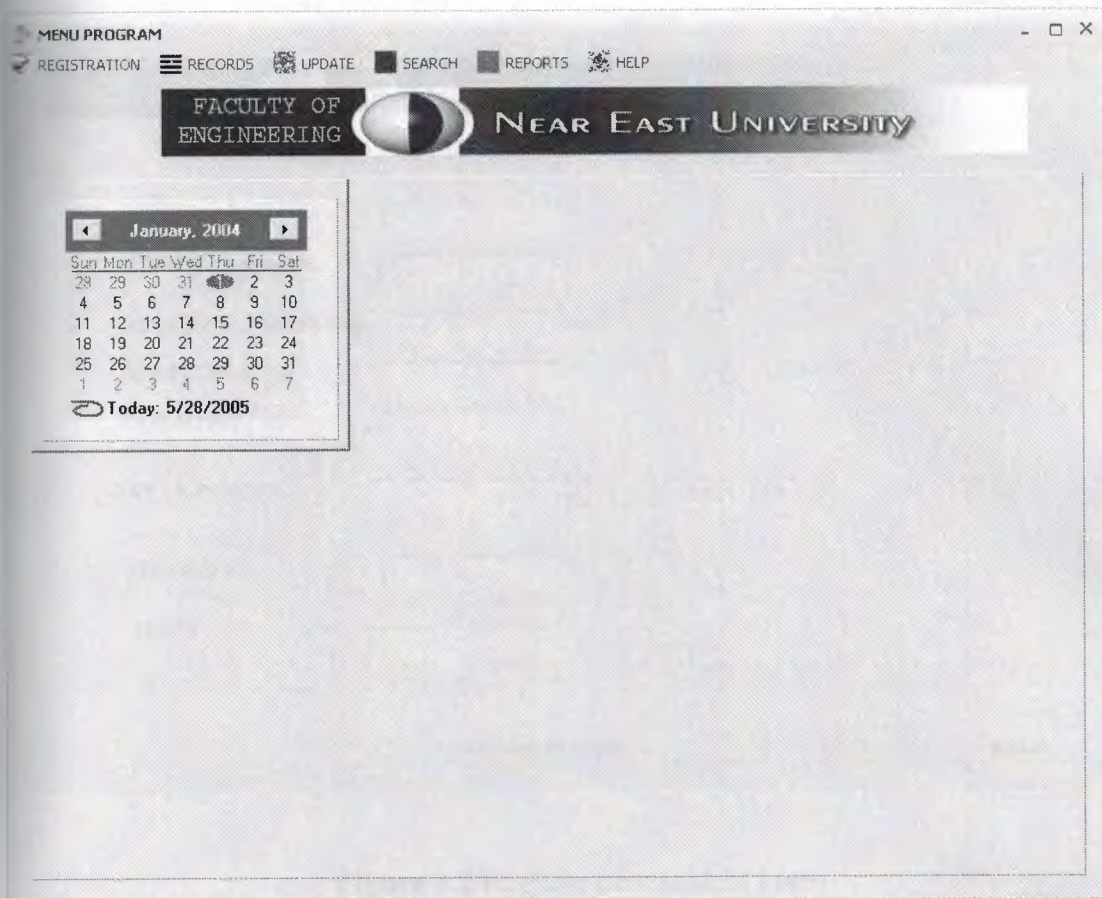
### **STUDENTS REGISTRAION SYSTEM PROGRAM**

#### **3.1. Main Menu Form:**

The Main Menu of the program has six buttons which are:

1. Registration:
  - 1) Register new student.
  - 2) Close.
2. Records:
  - 1) Computer Courses and Grades.
  - 2) Electrical Courses and Grades.
3. Update
  - 1) Update Students Information.
  - 2) Update Students Courses.
  - 3) Update Students Semester.
4. Search:
  - 1) Search Students Information.
  - 2) Search GPA for Computer Students.
  - 3) Search GPA for Electrical Students.
5. Report:
  - 1) Computer Courses.
  - 2) Electrical Courses.
  - 3) Semester Courses and Grades.
6. Help:
  - 1) Help





**Figure 3.1** Main Menu Form

### 3.2. Students Information Form:

This Form allows us to enter information such as: Student Number, student name, student surname, student birthplace, country, Department, student birthday, date.

The uses of elements used in this form:

- 1- Accept entries Button: This button used to keep and save the information of students which entered in the DBedit components to a file.
- 2- DBEDIT: here we used eight Dbedit and its used to send any thing we right in to the table.
- 3- Back button: its connected to the main form when we click it, it will send us to the main menu form.

REGISTER NEW STUDENT

FACULTY OF ENGINEERING

NEAR EAST UNIVERSITY

REGISTER NEW STUDENT

STUDENT NO

STUDNET NAME

STUDENT SURNAME

BIRTH PLACE

COUNTRY

DEPARTMENT

COMPUTER ENG

BIRTHDAY

DATE

ACCEPT ENTRIES

BACK

Figure 3.2 Register new student Form

### 3.3. Update Information Form:

We use this form when we want to update or correct some of student information. It consists of two buttons and nine DBedit components and Back button.

- 1- Search Button: This button used to search the students information and its search's by using database files.
- 2- Accept Entries Button: this button is used for save the current information and or any update happens in these fields or keep as same as before.
- 3- Delete Entries Button: this button is used to delete the entries fields.
- 4- Back button: it's connected to the main form when we click it, it will send us to the main menu form.



Form9

FACULTY OF ENGINEERING

NEAR EAST UNIVERSITY

SEARCH STUDENT INFORMATION

STUDENTE NO

STUDENT NO

STUDNET NAME

STUDENT SURNAME

BIRTH PLACE

COUNTRY

DEPARTMENT

BIRTHDAY

DATE

**Figure 3.3** Update Student Info. Form

### 3.4. Registration Courses for Computer students Form:

This Form allows us to enter courses of computer Engineering students and it consists of the following:

- 1- Search : the usage of this button is to search the student information to insure that he is register already if yes the name and surname of the student will appear in the fields, if not nothing will happen..
- 2- Dblookupcombobox component: this component is to get the data from table and sent or set the data to another table and this component consists the whole courses of the computer department.
- 3- Accept Entries: it's used to save the changes and courses that we register for the student.
- 4- Eight DBedit components: these are used to enter data.

- 5- Back button: its connected to the main form when we click it ,it will send us to the main menu form.

Form3

FACULTY OF ENGINEERING NEAR EAST UNIVERSITY

COMPUTER STUDENTS COURSES AND GRADES

COMPUTER COURSES | COMPUTER GRADES

STUDNET NO  Search

ACADEMIC YEAR  SEMESTER

STUDENT NAME  STUDNET SURNAME

COURSE CODES  COURSE NAME

CREDIT  DEPARTMENT

STUDENT NO

ACCEPT ENTRIES

BACK

**Figure 3.4** Records COM Courses Form

### 3.5. Registration Courses for Electrical and Electronic students Form:

This Form allows us to enter courses of Electrical and Electronic Engineering students and it consists of the following:

- 1- Search : the usage of this button is to search the student information to insure that he is register already if yes the name and surname of the student will appear in the fields, if not nothing will happen..
- 2- Dblookupcombobox component: this component is to get the data from table and sent or set the data to another table and this component consists the whole courses of the computer department.



- 3- Accept Entries: it's used to save the changes and courses that we register for the student.
- 4- Eight DBedit components: these are used to enter data.
- 5- Back button: its connected to the main form when we click it ,it will send us to the main menu form.

Form6

FACULTY OF ENGINEERING NEAR EAST UNIVERSITY

**ELECTRICAL STUDENTS COURSES AND GRADES**

ELECTRICAL COURSES ELECTRICAL GRADES

STUDNET NO  Search

ACADEMIC YEAR  SEMESTER

STUDENT NAME  STUDNET SURNAME

COURSE CODES  COURSE NAME

CREDIT  DEPARTMENT

STUDENT NO

ACCEPT ENTRIES

BACK

**Figure 3.5** Records EE Courses Form

### 3.6. Searching Information Form:

This form we use it to get the whole information about students, and its description is as follows:

- 1- Search Button: this button is used to get the information student by entering the number of student in the edit component as you see below, if the number of student in the file which keeps it then the information will be appeared.

- 2- DBgrid component: this component capable to display a whole table and the whole information.
- 3- Back button: its connected to the main form when we click it ,it will send us to the main menu form

SEARCH ALL STUDENTS INFORMATION

FACULTY OF ENGINEERING NEAR EAST UNIVERSITY

STUDENT INFORMATION FORM

STUDENT NO  Search

St_no	STUDENT_NAME	STUDENT_SURNAME	BIRTHDAY	BIRTHPLACE	DATE	DEPARTI ▲
-------	--------------	-----------------	----------	------------	------	-----------

< BACK

**Figure 3.6** Search Students Informaiton. Form

### 3.7. Update Semester Courses Form:

This Form allows us to update the student courses in the any semester we will be chosen and it consists of:

- 1- Combobox Component: this component used to select a single value from specified set. It has three values of the semester name (fall, spring, summer).
- 2- Search Button: after choosing semester and enter the number of student, we press this button to get the data from the file which save it on it and we use Query property to search by using student number, year, and semester.



- 3- DBgrid component: this component capable to display a whole table at once and you can edit the grid's contents.
- 4- Navigator Component: it consists eight buttons First, Last, Previous Next, Delete, Refresh, Edit, and Post.
- 5- Back button: its connected to the main form when we click it ,it will send us to the main menu form

UPDATE SEMESTER COURSES

FACULTY OF ENGINEERING NEAR EAST UNIVERSITY

UPATE SEMESTER COURSES

SEMESTER YEAR ST\_NO SEARCH

2005

Course_code	Course_name	Grades

< << >> + - > >>

< BACK

Figure 3.7 Update Semester courses Form

### 3.8. Update whole Courses of student Form:

This Form for updating the student courses, it consists of:

- 1- Search button: the task of this button is to get the student courses from the file that save it on it, after we write the student number in the edit component as we see bellow.
- 2- DBgrid component: we talk about this component in the previous Form, and of course whole data of student are appearing in this component.

- 3- Navigator component: this component has some task buttons, these buttons are First, Previous, Last, Next, and Delete unnecessary data.
- 4- Back button: its connected to the main form when we click it ,it will send us to the main menu form

[illegible]

**Figure 3.8**Update Whole Courses Form

### 3.9. Record Grades Form:

This Form for the computer department that records the grades of students, it consist three buttons, lookupcombobox, and combobox which are:

- 1- Lookupcombobox component: this component is getting the data from table and set these data to another table and here the data is courses of computer department.
- 2- Combobox component: this component used to select one of the values which it consist and here there are three value for semester name (fall, spring, summer).
- 3- Search Component: this button has only one task to get the data from the file, which saves the data on it, and here the data is the students' courses. By using Query component and using three information course codes, year, and semester.



- 4- Save button: this button is used to keep or save the data, after records the grades of the students to the suitable file.
- 5- DBgrid component: in this component there is operation that when we record the grade of student in the field which consist of the grades since we must double click to the next field in the dbgrid that called (Grt), to compute the value of the grades. For example if the grade is BB that means the value of this grade is 3.0, BA means 3.5 and so on.
- 6- Back button: its connected to the main form when we click it ,it will send us to the main menu form .

St_no	First_name	Surname	Grades	Gr	Gredit	Grit
-------	------------	---------	--------	----	--------	------

**Figure 3.9** Records COM Grades Form

### 3.10. GPA Form for Computer Department:

In this form or Form we calculate the GPA for any semester of the students, that by using some components:

- 1- Search button: this button has tow tasks one for searching by used student number, year, and name of semester by using combobox list, second for calculating the GPA of the student and it will appear in the edit component as you see in the Form bellow:
- 2- Back button: its connected to the main form when we click it ,it will send us to the main menu form



[illegible]

**Figure 3.10** GPA of COM Student Form

### 3.11. GPA Form for Electrical and Electronic Department:

In this form we calculate the GPA for any semester of the students by using some components:

- 1- Search button: this button has two tasks one for searching by used student number, year, and name of semester by using combo box list, second for calculate the GPA of the student and it will appear in the edit component .
- 2- Back button: its connected to the main form when we click it ,it will send us to the main menu form.

[illegible]

**Figure 3.11** GPA of EE Students Form

### 3.12. Grades of Student in a Semester Form:

In this form or Form there are the grades of student in one semester, we get these information by using search button and there is printer preview button.

1. Search button: the usage of this button to looking for the student information by chose the number of student and year and name of semester.
2. Printer preview button: the usage of this button is to show us the printer page after get the grades of student
3. Back button: its connected to the main form when we click it ,it will send us to the main menu form.



Course_code	Course_name	Grades
com101	Introduction to Computers	AA
ENG 101	English I	CB

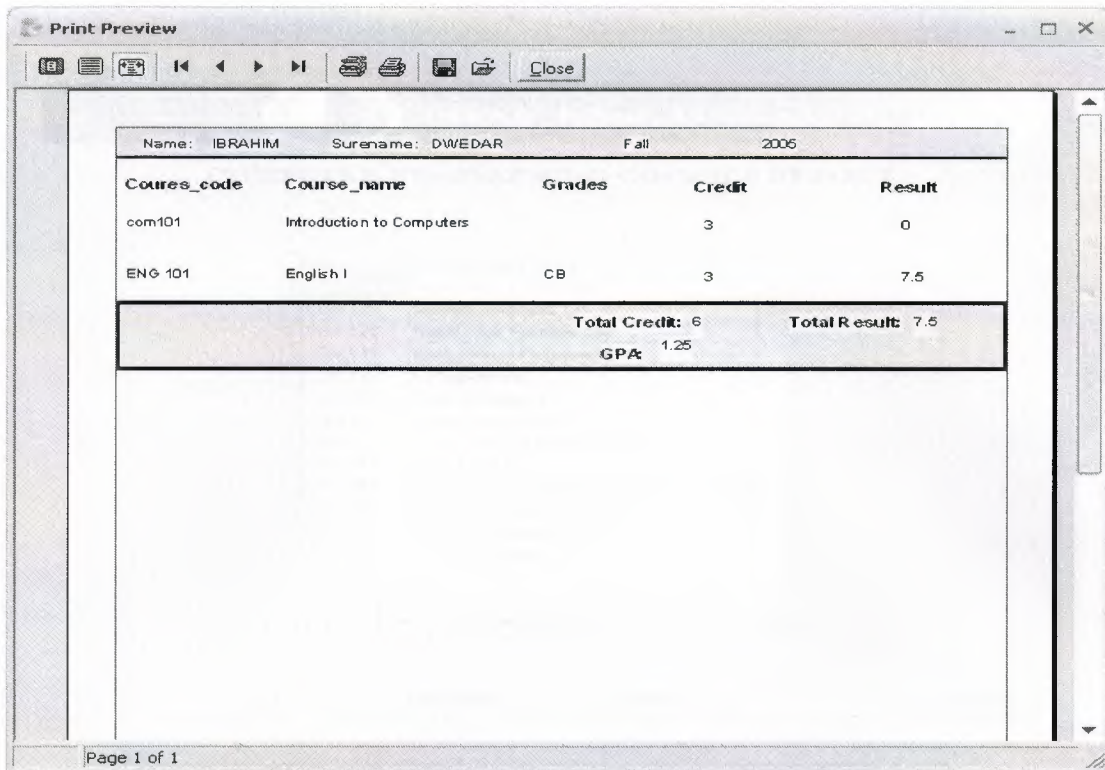
**Figure 3.12** Report Form for student Courses and Grades

### 3.12.1. Preview of Student Courses and Grades:

Here is the preview of the courses and grades of student, its showing the name of student, surname, semester and the year.

In it we can see the courses that this student is registered in and it gives the GPA of the courses of the student that he registered.

This preview it's supported by DELPHI PROGRAM and there is save the results and save it as quick report file, and open existing report using button, its just opening the QRP files which is related to Delphi.



Name: IBRAHIM		Surname: DWEDAR		Fall	2005
Courses_code	Course_name	Grades	Credit	Result	
com101	Introduction to Computers		3	0	
ENG 101	English I	CB	3	7.5	
			Total Credit: 6	Total Result: 7.5	
			GPA: 1.25		

Figure 3.13 Report for student Courses and Grades

### 3.13. Computer Engineering Courses Report:

This form is used for updating and printing the whole courses of Computer engineering students, DBnavigator is used and three buttons:

1. Preview Button: These buttons is used for showing and preview the whole courses of computer engineering courses.
2. Print Button: This button is used for directly printing the whole courses of computer engineering students without going to preview window.
3. Back button: it's connected to the main form when we click it, it will send us to the main menu form.



Course_code	Course_name
COM 101	Introduction to Computers
COM 121	Discrete Structures
COM 122	Digital Logic Fundamentals
COM 131	Introduction to Programming
COM 132	C Programming
COM 200	Summer Training I
COM 211	Digital Logic Systems
COM 226	Object Oriented Programming
COM 241	Data Structures
COM 242	Introduction to Database Management system
COM 252	Computer Architecture
COM 300	Summer Training II
COM 301	Microprocessors

**Figure 3.14** Update Computer Engineering Courses Report

### 3.13.1. Preview of Computer Engineering Courses Report:

Here is the preview of the whole courses of Computer engineering department.

In it we can see the courses that this student is registered in and it gives the GPA of the courses of the student that he registered.

This preview it's supported by DELPHI PROGRAM and there is save the results and save it as quick report file, and open existing report using button, its just opening the QRP files which is related to Delphi.



## REFERENCES

### BOOKS:

- 1- Borland Delphi in 21 days/
- 2- Jeff Duntemman "Delphi Programming Explorer"
- 3- Marco Cantu" Mastering Delphi "

### WEBSITES

[www.sybex.com](http://www.sybex.com)

[www.marcocantu.com](http://www.marcocantu.com)

[www.kdtool.com](http://www.kdtool.com)

[www.delphi.com](http://www.delphi.com)

**APPENDIX****1. MAIN MENU:**

unit REG;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, jpeg, ExtCtrls, ComCtrls;

type

TForm1 = class(TForm)

MainMenu1: TMainMenu;

REGISTRATION1: TMenuItem;

REGISTERNEWSTUDENT1: TMenuItem;

CLOSE1: TMenuItem;

RECORDS1: TMenuItem;

COMPUTERSTUDENTS1: TMenuItem;

ELECTRICALSTUDENTS1: TMenuItem;

EECOURCES1: TMenuItem;

EEGRADES1: TMenuItem;

EESTUDENTS3: TMenuItem;

EEST1: TMenuItem;

EEGRADES2: TMenuItem;

EECOURSES1: TMenuItem;

UPDATE1: TMenuItem;

STUDENTSINFORAMATION1: TMenuItem;

STUDENTINFORMATION1: TMenuItem;

STUDENTCOURCES1: TMenuItem;

STUDENTSEMESTER1: TMenuItem;

SEARCH1: TMenuItem;

STUDENTINFORMATION2: TMenuItem;

STUDENTGPA1: TMenuItem;

COMSTUDENTS1: TMenuItem;

EESTUDENTS1: TMenuItem;

REPORTS1: TMenuItem;

COURSES1: TMenuItem;

COMSTUDENTS2: TMenuItem;

EESTUDENTS2: TMenuItem;

SEMISTERCOURSE1: TMenuItem;

HELP1: TMenuItem;

HELP2: TMenuItem;

Image1: TImage;

Panel1: TPanel;

TabControl1: TTabControl;

Panel6: TPanel;

MonthCalendar1: TMonthCalendar;

procedure REGISTERNEWSTUDENT1Click(Sender: TObject);

procedure CLOSE1Click(Sender: TObject);

```

procedure COMPUTERSTUDENTS1Click(Sender: TObject);
procedure EECOURCES1Click(Sender: TObject);
procedure EEGRADES1Click(Sender: TObject);
procedure EESTUDENTS3Click(Sender: TObject);
procedure EEGRADES2Click(Sender: TObject);
procedure EECOURSES1Click(Sender: TObject);
procedure STUDENTINFORMATION1Click(Sender: TObject);
procedure STUDENTCOURCES1Click(Sender: TObject);
procedure STUDENTSEMESTER1Click(Sender: TObject);
procedure STUDENTINFORMATION2Click(Sender: TObject);
procedure COMSTUDENTS1Click(Sender: TObject);
procedure EESTUDENTS1Click(Sender: TObject);
procedure UNDERGRADUATEPROGRAM1Click(Sender: TObject);
procedure SEMISTERCOURSE1Click(Sender: TObject);
procedure COMSTUDENTS2Click(Sender: TObject);
procedure EESTUDENTS2Click(Sender: TObject);
procedure MonthCalendar1Click(Sender: TObject);
procedure Panel6Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

uses REG2, REG3, REG6, REG4, REG5, REG7, REG8, REG9, REG10, REG11,
  REG12,
  REG13, REG14, REG15, REG18, REG17;

{$R *.dfm}

procedure TForm1.REGISTERNEWSTUDENT1Click(Sender: TObject);
begin
  Application.CreateForm(TForm2, Form2);
  FORM2.SHOW;
  FORM1.Hide;
end;

procedure TForm1.CLOSE1Click(Sender: TObject);
begin
  CLOSE;
end;

procedure TForm1.COMPUTERSTUDENTS1Click(Sender: TObject);
begin

```

```
FORM3.SHOW;  
FORM2.HIDE;  
FORM1.Hide;  
end;  
  
procedure TForm1.EECOURCES1Click(Sender: TObject);  
begin  
FORM4.SHOW;  
FORM1.HIDE;  
end;  
  
procedure TForm1.EEGRADES1Click(Sender: TObject);  
begin  
FORM5.SHOW;  
FORM1.Hide;  
end;  
  
procedure TForm1.EESTUDENTS3Click(Sender: TObject);  
begin  
FORM6.Show;  
FORM1.HIDE;  
end;  
  
procedure TForm1.EEGRADES2Click(Sender: TObject);  
begin  
FORM7.SHOW;  
FORM1.Hide;  
end;  
  
procedure TForm1.EECOURSES1Click(Sender: TObject);  
begin  
FORM8.SHOW;  
FORM1.Hide;  
end;  
  
procedure TForm1.STUDENTINFORMATION1Click(Sender: TObject);  
begin  
FORM9.SHOW;  
FORM1.HIDE;  
end;  
  
procedure TForm1.STUDENTCOURCES1Click(Sender: TObject);  
begin  
FORM10.SHOW;  
FORM1.Hide;  
end;  
  
procedure TForm1.STUDENTSEMESTER1Click(Sender: TObject);  
begin  
FORM11.SHOW;
```



```
FORM1.Hide;  
end;
```

```
procedure TForm1.STUDENTINFORMATION2Click(Sender: TObject);  
begin  
FORM12.SHOW;  
FORM1.Hide;  
end;
```

```
procedure TForm1.COMSTUDENTS1Click(Sender: TObject);  
begin  
FORM13.SHOW;  
FORM1.Hide;  
end;
```

```
procedure TForm1.EESTUDENTS1Click(Sender: TObject);  
begin  
FORM14.SHOW;  
FORM1.Hide;  
end;
```

```
procedure TForm1.UNDERGRADUATEPROGRAM1Click(Sender: TObject);  
begin  
FORM15.SHOW;  
FORM1.Hide;  
end;
```

```
procedure TForm1.SEMISTERCOURSE1Click(Sender: TObject);  
begin  
form18.show;  
form1.Hide;  
end;
```

```
procedure TForm1.COMSTUDENTS2Click(Sender: TObject);  
begin  
FORM17.SHOW;  
FORM1.Hide;  
  
end;
```

```
procedure TForm1.EESTUDENTS2Click(Sender: TObject);  
begin  
FORM15.Show;  
FORM1.Hide;  
end;
```

```
procedure TForm1.MonthCalendar1Click(Sender: TObject);  
begin  
  
end;
```

```

procedure TForm1.Panel6Click(Sender: TObject);
begin

end;

end.

```

## 2. REGISTER NEW STUDENT:

```

unit REG2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, DB, DBTables, StdCtrls, Mask, jpeg, ExtCtrls, Menus;

type
  TForm2 = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit8: TDBEdit;
    DBEdit9: TDBEdit;
    Panel2: TPanel;
    Label10: TLabel;
    Button1: TButton;
    Button2: TButton;
    Table1: TTable;
    DataSource1: TDataSource;
    DBEdit7: TDBEdit;
    DBLookupComboBox1: TDBLookupComboBox;
    Table2: TTable;
    DataSource2: TDataSource;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure DBLookupComboBox1Click(Sender: TObject);

```

```

3 procedure FormCreate(Sender: TObject);

private
4 { Private declarations }
public
5 { Public declarations }
end;

var
6 Form2: TForm2;

implementation

uses REG;

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
table1.Post;
table1.Insert;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
FORM2.Hide;
FORM1.SHOW;
end;

procedure TForm2.DBLookupComboBox1Click(Sender: TObject);
begin
if dblookupcombobox1.Text<>" then
dbedit7.Text:=table2['DEPARTMENT'];

end;

procedure TForm2.FormCreate(Sender: TObject);
begin
table1.Append;
end;

end.

```

**3. RECORDS OF COURSES FOR COMPUTER ENGINEERING STUDENTS:**

unit REG4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, DBCtrls, Mask, jpeg, ExtCtrls;

type

```
TForm4 = class(TForm)
  Panel1: TPanel;
  Image1: TImage;
  GroupBox1: TGroupBox;
  Label5: TLabel;
  Label8: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label7: TLabel;
  Label4: TLabel;
  Label13: TLabel;
  Label6: TLabel;
  Label9: TLabel;
  DBEdit7: TDBEdit;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  DBEdit4: TDBEdit;
  DBLookupComboBox1: TDBLookupComboBox;
  DBEdit3: TDBEdit;
  DBEdit5: TDBEdit;
  Button2: TButton;
  DBEdit6: TDBEdit;
  DBEdit8: TDBEdit;
  Panel3: TPanel;
  Label10: TLabel;
  Edit1: TEdit;
  Button3: TButton;
  Button1: TButton;
  Panel2: TPanel;
  Label1: TLabel;
  Table4: TTable;
  DataSource1: TDataSource;
  DataSource3: TDataSource;
  Table2: TTable;
  DataSource4: TDataSource;
  Table3: TTable;
  Table1: TTable;
  DataSource2: TDataSource;
  procedure Button3Click(Sender: TObject);
```



```

procedure DBLookupComboBox1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;

implementation

uses REG;

{$R *.dfm}

procedure TForm4.Button3Click(Sender: TObject);
begin
  table3.filtered:=false;
  table3.filter:='St_no='+edit1.text;
  table3.filtered:=true;
  table1.insert;
  if dbedit7.Text<>" then
    dbedit3.Text:=table3['st_no'];
  if dbedit7.Text<>" then
    dbedit1.Text:=table3['student_name'];
  if dbedit7.Text<>" then
    dbedit2.Text:=table3['student_surname'];
  dbedit6.text:=table4.Fields[1].Text;
  dbedit8.Text:=table4.Fields[0].Text;
  //if form1.Edit1.Text<>" then
  //dbedit6.Text:=form1.Edit1.Text;
end;

procedure TForm4.DBLookupComboBox1Click(Sender: TObject);
begin
  if dblookupcombobox1.Text<>" then
    dbedit4.Text:=table2['course_name'];
  if dblookupcombobox1.Text<>" then
    dbedit5.Text:=table2['gredit'];
end;

procedure TForm4.Button2Click(Sender: TObject);
begin
  table1.Post;
  table1.Insert;
  if dbedit7.Text<>" then

```

```

dbedit3.Text:=table3['st_no'];
if dbedit7.Text<>" then
dbedit1.Text:=table3['STUDENT_NAME'];
if dbedit7.Text<>" then
dbedit2.Text:=table3['STUDENT_SURNAME'];
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;
end;

procedure TForm4.Button1Click(Sender: TObject);
begin
FORM4.Hide;
FORM1.SHOW;
end;

procedure TForm4.FormCreate(Sender: TObject);
begin
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;
end;

end.

```

#### 4. GRADE RECORDS FOR COMPUTER ENGINEERING STUDENTS:

```

unit REG5;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBTables, DB, DBCtrls, StdCtrls, Grids, DBGrids, ExtCtrls, jpeg;

type
  TForm5 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    DBLookupComboBox1: TDBLookupComboBox;
    Button2: TButton;
    Table1: TTable;
    DataSource2: TDataSource;
    Table2: TTable;
    DataSource3: TDataSource;
    DataSource1: TDataSource;

```

```

Query1: TQuery;
Table3: TTable;
Button3: TButton;
Panel2: TPanel;
Image1: TImage;
Panel3: TPanel;
Label4: TLabel;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form5: TForm5;

implementation

uses REG;

{$R *.dfm}

procedure TForm5.Button1Click(Sender: TObject);
VAR str: string;
begin
  str:='select*from grade where';

  if DBLookupComboBox1.Text <> " then
    str:=str+' course_code="'+ DBLookupComboBox1.Text+" and '";
  if edit2.Text <> " then
    str:=str+' academic_year="'+ edit2.Text+" and '";
  if combobox1.Text <> " then
    str:=str+' semester="'+ combobox1.Text+" '";
  query1.Close;
  query1.sql.Clear;
  query1.sql.add(str);
  query1.Open;
  dbgrid1.Visible:=true;
end;

procedure TForm5.Button2Click(Sender: TObject);
begin
  query1.Edit;
  query1.Post;
  edit2.text:=table3.Fields[1].Text;

```





```

end;

procedure TForm5.Button3Click(Sender: TObject);
begin
  FORM5.Hide;
  FORM1.SHOW;
end;

procedure TForm5.FormCreate(Sender: TObject);
begin
  TABLE1.Append;
  TABLE2.Append;
  TABLE3.Append;
end;

end.

```

## **5. RECORDS OF COURSES FOR ELECTRICAL ENGINEERING STUDENTS:**

```

unit REG7;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls, DBCtrls, Mask, jpeg, ExtCtrls;

type
  TForm7 = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    GroupBox1: TGroupBox;
    Label5: TLabel;
    Label8: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label7: TLabel;
    Label4: TLabel;
    Label13: TLabel;
    Label6: TLabel;
    Label9: TLabel;
    DBEdit7: TDBEdit;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit4: TDBEdit;
    DBLookupComboBox1: TDBLookupComboBox;
    DBEdit3: TDBEdit;
    DBEdit5: TDBEdit;
    Button2: TButton;

```

```

DBEdit6: TDBEdit;
DBEdit8: TDBEdit;
Panel3: TPanel;
Label10: TLabel;
Edit1: TEdit;
Button3: TButton;
Button1: TButton;
Panel2: TPanel;
Label1: TLabel;
Table1: TTable;
DataSource1: TDataSource;
DataSource2: TDataSource;
Table2: TTable;
Table3: TTable;
DataSource3: TDataSource;
Table4: TTable;
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure DBLookupComboBox1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form7: TForm7;

implementation

uses REG;

{$R *.dfm}

procedure TForm7.Button3Click(Sender: TObject);
begin
  table3.filtered:=false;
  table3.filter:='st_no='+edit1.text;
  table3.filtered:=true;
  table1.insert;
  if dbedit7.Text<>" then
    dbedit3.Text:=table3['st_no'];
  if dbedit7.Text<>" then
    dbedit1.Text:=table3['STUDENT_NAME'];
  if dbedit7.Text<>" then
    dbedit2.Text:=table3['STUDENT_SURNAME'];
  dbedit6.text:=table4.Fields[1].Text;
  dbedit8.Text:=table4.Fields[0].Text;

```

```
//DBRadioGroup1.Value:=table4.Fields[0].Text;
//if form1.Edit1.Text<>" then
//dbedit6.Text:=form1.Edit1.Text;
end;
```

```
procedure TForm7.Button2Click(Sender: TObject);
begin
table1.Post;
table1.Insert;
if dbedit7.Text<>" then
dbedit3.Text:=table3['st_no'];
if dbedit7.Text<>" then
dbedit1.Text:=table3['STUDENT_NAME'];
if dbedit7.Text<>" then
dbedit2.Text:=table3['STUDENT_SURNAME'];
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;
//DBRadioGroup1.Value:=table4.Fields[0].Text;
//if form1.Edit1.Text<>" then
//dbedit6.Text:=form1.Edit1.Text;
end;
```

```
procedure TForm7.Button1Click(Sender: TObject);
begin
FORM7.Hide;
FORM1.SHOW;
end;
```

```
procedure TForm7.DBLookupComboBox1Click(Sender: TObject);
begin
if dblookupcombobox1.text<>" then
dbedit4.Text:=table2['course_name'];
if dblookupcombobox1.text<>" then
dbedit5.Text:=table2['Credits'];
end;
```

```
procedure TForm7.FormCreate(Sender: TObject);
begin
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;
//DBRadioGroup1.Value:=table4.Fields[0].Text;
TABLE1.Append;
TABLE2.APPEND;
```

```
TABLE4.Append;
```

```
end;
```

```
end.
```



**6. GRADE RECORDS FOR ELECTRICAL ENGINEERING STUDENTS:**

unit REG8;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DBTables, DB, StdCtrls, DBCtrls, Grids, DBGrids, ExtCtrls, jpeg;

type

```

TForm8 = class(TForm)
  Panel1: TPanel;
  Button1: TButton;
  DBGrid1: TDBGrid;
  DBLookupComboBox2: TDBLookupComboBox;
  Edit1: TEdit;
  ComboBox2: TComboBox;
  Button2: TButton;
  Table1: TTable;
  DataSource1: TDataSource;
  DataSource2: TDataSource;
  Table2: TTable;
  Query1: TQuery;
  DataSource3: TDataSource;
  Table3: TTable;
  Button3: TButton;
  Panel2: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Panel3: TPanel;
  Label4: TLabel;
  Image1: TImage;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

var

Form8: TForm8;

implementation

uses REG;

```

{$R *.dfm}

procedure TForm8.Button1Click(Sender: TObject);
VAR
str: string;

begin

str:='select * from grade where';

if DBLookupComboBox2.Text <> " then
str:=str+' course_code="'+ DBLookupComboBox2.Text+'" and ';
if edit1.Text <> " then
str:=str+' academic_year="'+ edit1.Text+'" and';
if combobox2.Text <> " then
str:=str+' semester="'+ combobox2.Text+'" ';

query1.Close;
query1.sql.Clear;
query1.sql.add(str);
query1.Open;
dbgrid1.Visible:=true;
end;

procedure TForm8.Button2Click(Sender: TObject);
begin
query1.Edit;
query1.Post;
end;

procedure TForm8.Button3Click(Sender: TObject);
begin
FORM8.Hide;
FORM1.SHOW;
end;

procedure TForm8.FormCreate(Sender: TObject);
begin
edit1.Text:=table3.Fields[1].Text;
end;

end.

```

## 7. SEARCH STUDENT INFORMATION:

```
unit REG9;
```

```
interface
```

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, Mask, DBCtrls, ExtCtrls, jpeg;

type

```
TForm9 = class(TForm)
  Label3: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label2: TLabel;
  Panel1: TPanel;
  Label4: TLabel;
  Button1: TButton;
  Edit1: TEdit;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  DBEdit4: TDBEdit;
  DBEdit5: TDBEdit;
  DBEdit6: TDBEdit;
  DBEdit8: TDBEdit;
  DBEdit9: TDBEdit;
  DBEdit3: TDBEdit;
  Button2: TButton;
  Button3: TButton;
  Table1: TTable;
  DataSource1: TDataSource;
  Button4: TButton;
  Panel3: TPanel;
  Image1: TImage;
  Panel4: TPanel;
  Label1: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  Label17: TLabel;
  Label18: TLabel;
  procedure Button3Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button4Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
```



```

public
  { Public declarations }
end;

var
  Form9: TForm9;

implementation

uses REG;

{$R *.dfm}

procedure TForm9.Button3Click(Sender: TObject);
begin
  if messagedlg('Are you sure you want to delete the current reecords?',
    mtconfirmation,[mbyes,mbno],0)=idyesthen
    table1.delete;
end;

procedure TForm9.Button2Click(Sender: TObject);
begin
  table1.Post;
  table1.insert;
end;

procedure TForm9.Button1Click(Sender: TObject);
begin
  table1.Filtered:=false;
  table1.Filter:='st_no='+edit1.Text;
  table1.Filtered:=true;
  table1.Edit;
  table1.Post;
  table1.Refresh;
end;

procedure TForm9.Button4Click(Sender: TObject);
begin
  FORM9.Hide;
  FORM1.SHOW;
end;

procedure TForm9.FormCreate(Sender: TObject);
begin
  table1.Append;
end;

end.

```

**8. UPDATE STUDENT COURSES:**

unit REG10;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, Menus, DBTables, DBCtrls, Grids, DBGrids, StdCtrls, ExtCtrls,  
jpeg;

type

TForm10 = class(TForm)  
  Panel1: TPanel;  
  Label1: TLabel;  
  Button1: TButton;  
  Edit1: TEdit;  
  DBGrid1: TDBGrid;  
  DBNavigator1: TDBNavigator;  
  Table1: TTable;  
  DataSource3: TDataSource;  
  Table2: TTable;  
  Button2: TButton;  
  Panel3: TPanel;  
  Image1: TImage;  
  Panel2: TPanel;  
  Label2: TLabel;  
  procedure Button1Click(Sender: TObject);  
  procedure Button2Click(Sender: TObject);  
  procedure FormCreate(Sender: TObject);

private

  { Private declarations }

public

  { Public declarations }

end;

var

  Form10: TForm10;

implementation

uses REG;

{ \$R \*.dfm }

procedure TForm10.Button1Click(Sender: TObject);

begin

  table2.Filtered:=false;

  table2.filter:='st\_no='+edit1.Text;

  table2.Filtered:=true;

```

end;

procedure TForm10.Button2Click(Sender: TObject);
begin
  FORM10.Hide;
  FORM1.SHOW;
end;

procedure TForm10.FormCreate(Sender: TObject);
begin
  TABLE2.Append;
  TABLE1.Append;
end;

end.

```

## 9. UPATE SEMESTER COURSES:

```

unit REG11;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBTables, DB, Menus, Grids, DBGrids, StdCtrls, DBCtrls, ExtCtrls,
  jpeg;

type
  TForm11 = class(TForm)
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    DBNavigator1: TDBNavigator;
    button3: TButton;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Query1: TQuery;
    DataSource2: TDataSource;
    Table1: TTable;
    DataSource3: TDataSource;
    Table2: TTable;
    Table3: TTable;
    Query2: TQuery;
    Button1: TButton;
    Panel2: TPanel;
    Image1: TImage;

```

```

Panel3: TPanel;
Label1: TLabel;
procedure button3Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form11: TForm11;

implementation

uses REG;

{$R *.dfm}

procedure TForm11.button3Click(Sender: TObject);
begin
  table1.Filtered:=false;
  table1.Filter:=' st_no='+edit1.Text+' and semester="'+combobox1.text+'" and
  academic_year="'+edit2.text+'";
  table1.Filtered:=true;
end;

procedure TForm11.FormCreate(Sender: TObject);
begin
  edit2.text:=table3.Fields[1].Text;
end;

procedure TForm11.Button1Click(Sender: TObject);
begin
  FORM11.Hide;
  FORM1.SHOW;
end;

end.

```

## 10. STUDENT INFORMATION FORM:

```
unit REG12;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls, ExtCtrls, Grids, DBGrids, jpeg;
```



type

```
TForm12 = class(TForm)
  DBGrid1: TDBGrid;
  Panel1: TPanel;
  Label1: TLabel;
  Button1: TButton;
  Edit1: TEdit;
  Table1: TTable;
  DataSource1: TDataSource;
  Button2: TButton;
  Panel2: TPanel;
  Panel3: TPanel;
  Image1: TImage;
  Panel4: TPanel;
  Label2: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form12: TForm12;
```

implementation

uses REG;

```
{ $R *.dfm }
```

```
procedure TForm12.Button1Click(Sender: TObject);
begin
  table1.Filtered:=false;
  table1.Filter:='st_no='+edit1.Text;
  table1.Filtered:=true;
end;
```

```
procedure TForm12.Button2Click(Sender: TObject);
begin
  FORM1.SHOW;
  FORM12.Hide;
end;
```

```
procedure TForm12.FormCreate(Sender: TObject);
begin
```

end;

end.

## 11. GPA FOR COMPUTER STUDENTS:

unit REG13;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DBTables, DB, Mask, DBCtrls, StdCtrls, Grids, DBGrids, ExtCtrls,  
jpeg;

type

```
TForm13 = class(TForm)
  Panel1: TPanel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Button1: TButton;
  Edit1: TEdit;
  DBGrid1: TDBGrid;
  Edit2: TEdit;
  ComboBox1: TComboBox;
  DBNavigator1: TDBNavigator;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  Edit3: TEdit;
  DataSource1: TDataSource;
  Query1: TQuery;
  DataSource2: TDataSource;
  Table1: TTable;
  DataSource3: TDataSource;
  Table2: TTable;
  Query2: TQuery;
  DataSource4: TDataSource;
  Query3: TQuery;
  DataSource5: TDataSource;
  Table3: TTable;
  Button2: TButton;
  Panel2: TPanel;
  Panel3: TPanel;
  Label7: TLabel;
  Image1: TImage;
  Label6: TLabel;
  Label5: TLabel;
  Label1: TLabel;
  procedure Button1Click(Sender: TObject);
```

```

    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form13: TForm13;

implementation

uses REG;

{$R *.dfm}

procedure TForm13.Button1Click(Sender: TObject);
VAR
sw,sr,st,se: string;
s,d,f:double;
begin

sw:='select * from grade where';

if edit1.Text <> " then
sw:=sw+' st_no="'+ edit1.Text+" and '";
if edit2.Text <> " then
sw:=sw+' academic_year="'+ edit2.Text+" and '";
if combobox1.Text <> " then
sw:=sw+' semester="'+ combobox1.Text+" '";
query1.Close;
query1.sql.Clear;
query1.sql.add(sw);
query1.Open;
sr:='select sum(grt) from grade where';
if edit1.Text <> " then
sr:=sr+' st_no="'+ edit1.Text+" and '";
if edit2.Text <> " then
sr:=sr+' academic_year="'+ edit2.Text+" and '";
if combobox1.Text <> " then
sr:=sr+' semester="'+ combobox1.Text+" '";

query2.Close;
query2.sql.Clear;
query2.sql.add(sr);
query2.Open;
st:='select sum(gredit) from grade where';
if edit1.Text <> " then
st:=st+' st_no="'+ edit1.Text+" and '";

```

```

if edit2.Text <> "" then
st:=st+' academic_year="'+ edit2.Text+'" and';
if combobox1.Text <> "" then
st:=st+' semester="'+ combobox1.Text+'" ';
query3.Close;
query3.sql.Clear;
query3.sql.add(st);
query3.Open;
//dbgrid1.Visible:=true;
s:=strtoint(dbedit1.Text);
d:=strtoint(dbedit2.Text);
f:=s/d;
str(f:3:2,se);
edit3.Text:=se;

end;

procedure TForm13.Button2Click(Sender: TObject);
begin
FORM13.Hide;
FORM1.SHOW;
end;

procedure TForm13.FormCreate(Sender: TObject);
begin

end;

end.

```

## **12. GPA FOR ELECRITICAL STUDENTS:**

```
unit REG14;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DBTables, DB, Menus, Mask, DBCtrls, StdCtrls, Grids, DBGrids,
ExtCtrls, jpeg;
```

```
type
```

```
TForm14 = class(TForm)
```

```
Panel2: TPanel;
```

```
Image1: TImage;
```

```
Label6: TLabel;
```

```
Label5: TLabel;
```

```
Label1: TLabel;
```

```
Panel3: TPanel;
```

```
Label7: TLabel;
```

```
Panel1: TPanel;
```

```
Label2: TLabel;
```



```

Label3: TLabel;
Label4: TLabel;
Button1: TButton;
Edit1: TEdit;
DBGrid1: TDBGrid;
Edit2: TEdit;
ComboBox1: TComboBox;
DBNavigator1: TDBNavigator;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
Edit3: TEdit;
Button2: TButton;
DataSource1: TDataSource;
Query1: TQuery;
DataSource2: TDataSource;
Table1: TTable;
DataSource3: TDataSource;
Table2: TTable;
Query2: TQuery;
DataSource4: TDataSource;
Query3: TQuery;
DataSource5: TDataSource;
Table3: TTable;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form14: TForm14;

implementation

uses REG;

{$R *.dfm}

procedure TForm14.Button1Click(Sender: TObject);
VAR
  sw,sr,st,se: string;
  s,d,f:double;
begin
  sw:='select * from grade where';
  if edit1.Text <> '' then

```

```

sw:=sw+' st_no="'+ edit1.Text+" and '
if edit2.Text <> "" then
sw:=sw+' academic_year="'+ edit2.Text+" and '
if combobox1.Text <> "" then
sw:=sw+' semester="'+ combobox1.Text+" '
query1.Close;
query1.sql.Clear;
query1.sql.add(sw);
query1.Open;
dbgrid1.Visible:=true;
sr:='select sum(grt) from grade where';
if edit1.Text <> "" then
sr:=sr+' st_no="'+ edit1.Text+" and '
if edit2.Text <> "" then
sr:=sr+' academic_year="'+ edit2.Text+" and '
if combobox1.Text <> "" then
sr:=sr+' semester="'+ combobox1.Text+" '

query2.Close;
query2.sql.Clear;
query2.sql.add(sr);
query2.Open;
st:='select sum(gredit) from grade where';
if edit1.Text <> "" then
st:=st+' st_no="'+ edit1.Text+" and '
if edit2.Text <> "" then
st:=st+' academic_year="'+ edit2.Text+" and '
if combobox1.Text <> "" then
st:=st+' semester="'+ combobox1.Text+" '

query3.Close;
query3.sql.Clear;
query3.sql.add(st);
query3.Open;
s:=strtofloat(dbedit2.Text);
d:=strtoint(dbedit1.Text);
f:=s/d;
str(f:3:2,se);
edit3.Text:=se;

end;

procedure TForm14.Button2Click(Sender: TObject);
begin
FORM14.Hide;
FORM1.SHOW;
end;

procedure TForm14.FormCreate(Sender: TObject);
begin

```

end;

end.

### 13. ELECTRICAL ENGINEERING COURSES REPORT:

unit REG15;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, ExtCtrls, QuickRpt, StdCtrls, jpeg, DBCtrls,  
Grids, DBGrids, QRCtrl;

type

```
TForm15 = class(TForm)
  Panel1: TPanel;
  Image1: TImage;
  DBGrid1: TDBGrid;
  DBNavigator1: TDBNavigator;
  Button2: TButton;
  Button1: TButton;
  Button3: TButton;
  QuickRep1: TQuickRep;
  Button4: TButton;
  QRLabel3: TQRLabel;
  QRGroup2: TQRGroup;
  QRLabel1: TQRLabel;
  QRLabel2: TQRLabel;
  QRLabel4: TQRLabel;
  QRLabel5: TQRLabel;
  QRBand1: TQRBand;
  QRDBText1: TQRDBText;
  QRDBText2: TQRDBText;
  QRDBText3: TQRDBText;
  Table1: TTable;
  DataSource1: TDataSource;
  Panel2: TPanel;
  Label10: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form15: TForm15;

implementation

uses REG;

{$R *.dfm}

procedure TForm15.Button1Click(Sender: TObject);
begin
  FORM1.SHOW;
  FORM15.HIDE;
end;

procedure TForm15.Button2Click(Sender: TObject);
begin
  FORM15.QuickRep1.Preview;
end;

procedure TForm15.Button3Click(Sender: TObject);
begin
  FORM15.QuickRep1.Print;
end;

procedure TForm15.FormCreate(Sender: TObject);
begin

end;

end.
```

#### **14. COMPUTER ENGINEERING COURSES REPORT:**

```
unit REG17;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, DBTables, jpeg, ExtCtrls,
  QuickRpt, DBCtrls, QRCtrls;

type
  TForm17 = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    Panel2: TPanel;
    Label10: TLabel;
    Table1: TTable;
```



```

DataSource1: TDataSource;
DBGrid1: TDBGrid;
Button1: TButton;
Button2: TButton;
DBNavigator1: TDBNavigator;
Table2: TTable;
QuickRep1: TQuickRep;
QRBand1: TQRBand;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRGroup2: TQRGroup;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRDBText3: TQRDBText;
Button3: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form17: TForm17;

implementation

uses REG;

{$R *.dfm}

procedure TForm17.Button1Click(Sender: TObject);
begin
  FORM17.Hide;
  FORM1.SHOW;
end;

procedure TForm17.Button2Click(Sender: TObject);
begin
  FORM17.QuickRep1.Preview;
end;

procedure TForm17.Button3Click(Sender: TObject);
begin

```

```
FORM17.QuickRep1.PRINT;
end;
```

```
procedure TForm17.FormCreate(Sender: TObject);
begin
```

```
end;
```

```
end.
```

## **15. REPORT FOR STUDENTS COURESES AND GRADES:**

```
unit REG18;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DBTables, DB, Menus, QRCtrls, QuickRpt, StdCtrls, Grids,
DBGrids, ExtCtrls, jpeg;
```

```
type
```

```
TForm18 = class(TForm)
  Panel1: TPanel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Button1: TButton;
  Edit1: TEdit;
  DBGrid1: TDBGrid;
  Edit2: TEdit;
  ComboBox1: TComboBox;
  Button2: TButton;
  QuickRep1: TQuickRep;
  QRGroup1: TQRGroup;
  QRDBText4: TQRDBText;
  QRDBText5: TQRDBText;
  QRDBText7: TQRDBText;
  QRDBText8: TQRDBText;
  QRLabel9: TQRLabel;
  QRLabel10: TQRLabel;
  QRBand1: TQRBand;
  QRDBText1: TQRDBText;
  QRDBText2: TQRDBText;
  QRDBText3: TQRDBText;
  QRDBText6: TQRDBText;
  QRExpr1: TQRExpr;
  QRGroup2: TQRGroup;
  QRLabel1: TQRLabel;
  QRLabel2: TQRLabel;
  QRLabel3: TQRLabel;
  QRLabel4: TQRLabel;
```

```

QRLabel5: TQRLabel;
QRBand2: TQRBand;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
QRBand3: TQRBand;
QRExpr3: TQRExpr;
QRLabel7: TQRLabel;
QRExpr2: TQRExpr;
QRLabel6: TQRLabel;
QRExpr4: TQRExpr;
QRLabel8: TQRLabel;
DataSource1: TDataSource;
Query1: TQuery;
DataSource2: TDataSource;
Table1: TTable;
DataSource3: TDataSource;
Table2: TTable;
stquery: TQuery;
Table3: TTable;
Panel2: TPanel;
Button4: TButton;
Image1: TImage;
Panel3: TPanel;
Label1: TLabel;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form18: TForm18;

implementation

uses REG;

{$R *.dfm}

procedure TForm18.Button2Click(Sender: TObject);
begin
  form18.QuickRep1.Preview;
  form18.Visible:=false;
end;

procedure TForm18.Button1Click(Sender: TObject);

```



```
VAR
str: string;

begin

str:='select*from grade where';

if edit1.Text <> " " then
str:=str+' st_no="'+ edit1.Text+" and '";
if edit2.Text <> " " then
str:=str+' academic_year="'+ edit2.Text+" and '";
if combobox1.Text <> " " then
str:=str+' semester="'+ combobox1.Text+" '";
query1.Close;
query1.sql.Clear;
query1.sql.add(str);
query1.Open;
  dbgrid1.Visible:=true;
end;

procedure TForm18.Button4Click(Sender: TObject);
begin
FORM18.Hide;
FORM1.SHOW;
end;

procedure TForm18.FormCreate(Sender: TObject);
begin

end;

end.
```