College of Computer Science and Technology
Zhejiang University of Technology

The Task for Composite Experiments on Database System

Name: Mollick Shahriar Hossain

Student ID: L201826100121

Major: Computer Science and Technology

Course:Composite Experiment on Database Systems

Teacher: Duanyang Liu

Date: 2021/07/23

Index

1.Objective:

The objective of this course is learning to solve the practical problem using the knowledge on database system. From this process, I can exercise how to design the database for a practical case, how to create the database including tables, views, primary keys etc, how to use SQL commands to complete querying tasks and other tasks, and how to use a programming language to communicate with the database. This experiment will train the skills of database designing, SQL commands and database programming.

2.Project Description and Requirement:

There is an administrator who manages student-learning information of the college. The administrator needs to deal with much querying about students' information. Of course, it is very boring. So he wants to design a database to help him to complete the work efficiently and conveniently. According to his daily works, he lists his requirement as follows:

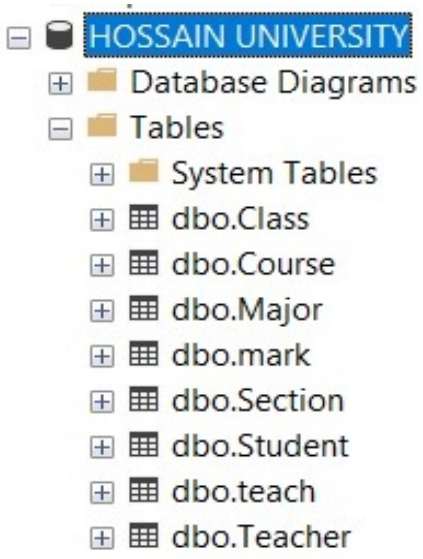
1. The college has some different majors, for example, computer science, software engineering, digital media, network engineering, etc. Each major has several classes and each class has about 30 students.
2. The information about a student involves student ID, name, sex, age, hometown, province, total credits, etc.
3. The information about the course involves course ID, course name, major, credits, etc. One course often has several sections, and one section is taught by one teacher in a semester. The information about the section involves course ID, section ID, semester, year, teaching location, etc.
4. The information about the teacher involves teacher ID, name, sex, age, title, contact phone, office, email, etc.
5. The information about a student's marks involves student ID, course ID, semester, course marks, etc.
6. Note that: One major has more than one class. One course in a semester needs more than one teacher. And one teacher can take more than one course in a semester.

3.Project Design with DDL sentences

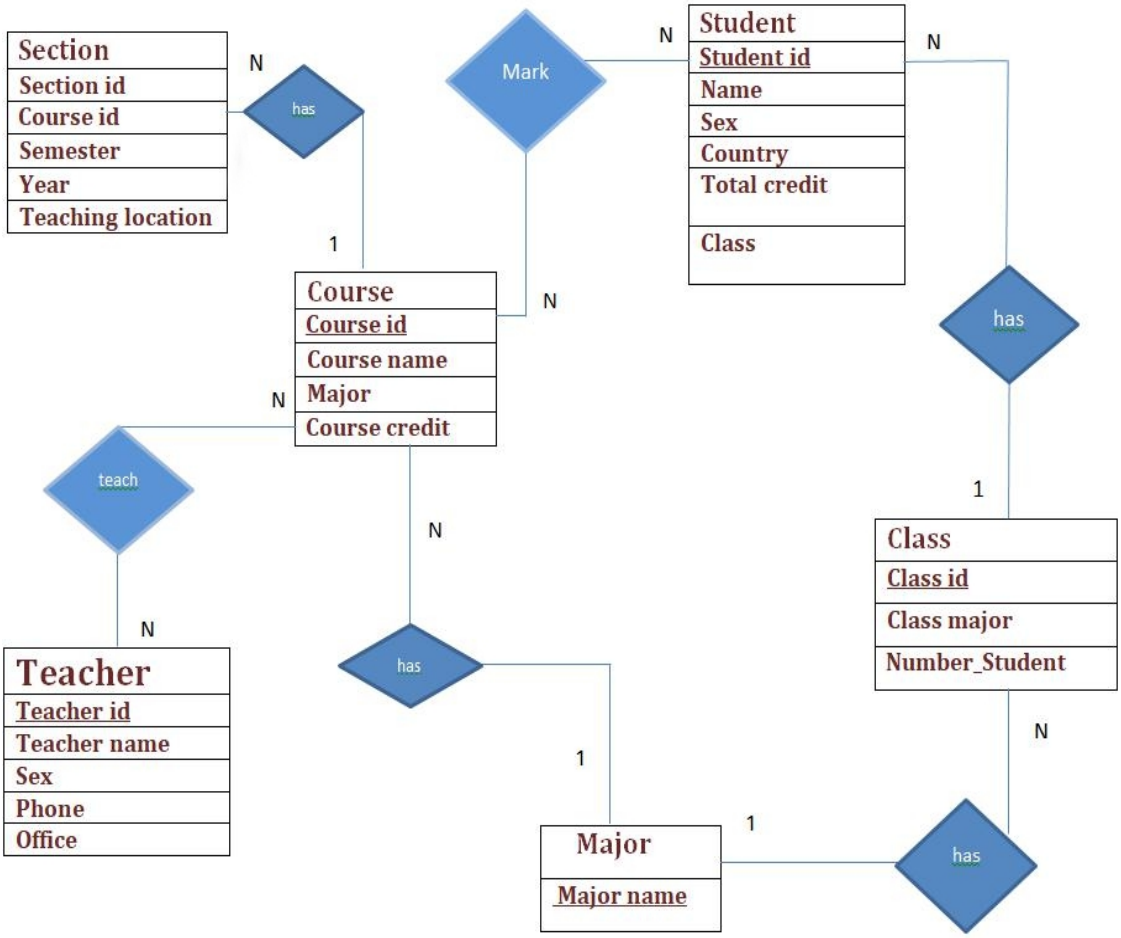
☞ DDL sentences For HOSSAIN UNIVERSITY database:

```
create table Major([Major name] varchar(30) primary key);

create table Course([Course id] varchar(30) primary key,[Course name]varchar(30),major varchar(30),[Course credit] int,
foreign key(major) references Major([Major name]));

create table Class([Class id] varchar(30) primary key,[Class major] varchar(30),Number_Student varchar(30),
foreign key([Class major]) references Major([Major name]));

create table Student([Student id] varchar(30) primary key,[Name] varchar(30),Sex char,country varchar(30),[Total credit] float,
Class varchar(30),foreign key(Class) references Class([Class id]));

create table Section([Section id] varchar(30),[Course id] varchar(30),Semester char,[Year] int,[Teaching location] varchar(30),
foreign key([Course id]) references Course([Course id]));

create table Teacher([Teacher id] varchar(30) primary key,[Teacher name] varchar(30),sex char, Phone varchar(30),
Office varchar(30));

create table teach([Teacher id] varchar(30),[Course id] varchar(30)
foreign key([Course id]) references Course([Course id]),
foreign key([Teacher id]) references Teacher([Teacher id]));

create table mark([Course id] varchar(30),[Student id] varchar(30),[course marks] float, Semester char
foreign key([Course id]) references Course([Course id]),
foreign key([Student id]) references Student([Student id]));
```

```
0 %
Messages
Commands completed successfully.

Completion time: 2021-07-13T07:02:50.1120590+06:00
```



☞E-R Diagram for the HOSSAIN UNIVERSITY database:



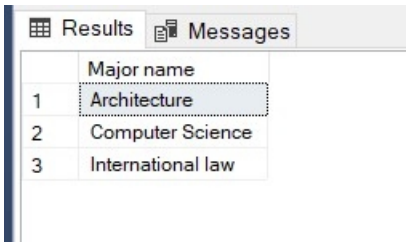
3.1 Implantation and explanation:

In this project I created a university data base which is named **(HOSSAIN UNIVERSITY)** the data base have 8 tables, 6 tables can be clearly seen from the E-R diagram (Class,Student, Teacher, Major, Course, Section) while the remaining 2 tables (mark, teach) are created from N-N relationship between 2 tables, I ignored the total participant for some tables in the diagram for the sake of simplicity. Of course after creating the tables, data had been inserted to them using the easiest way however I am quite familiar with using the "insert into table "command, but given the number of tables and data that have to be filled, I filled the tables directly using the table options in the SQL management studio, but the rest of the project will be finished using only SQL commands.

3.2Tables content:

❑Major

The university has 3 majors which are shown below:

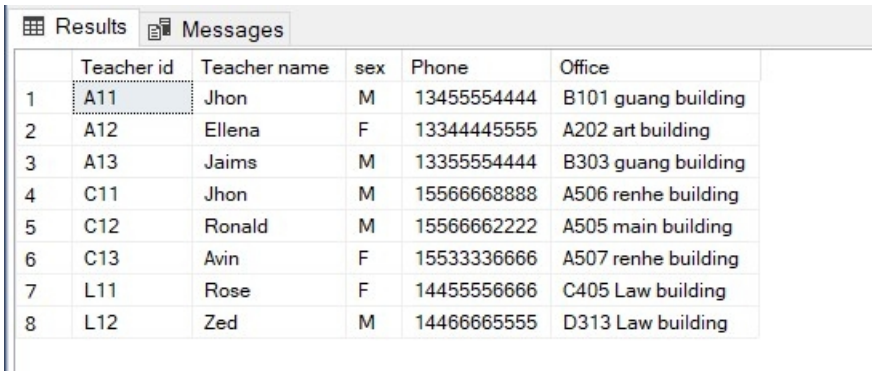


Results Messages

	Major name
1	Architecture
2	Computer Science
3	International law

❑Teacher

The teacher info shown below:



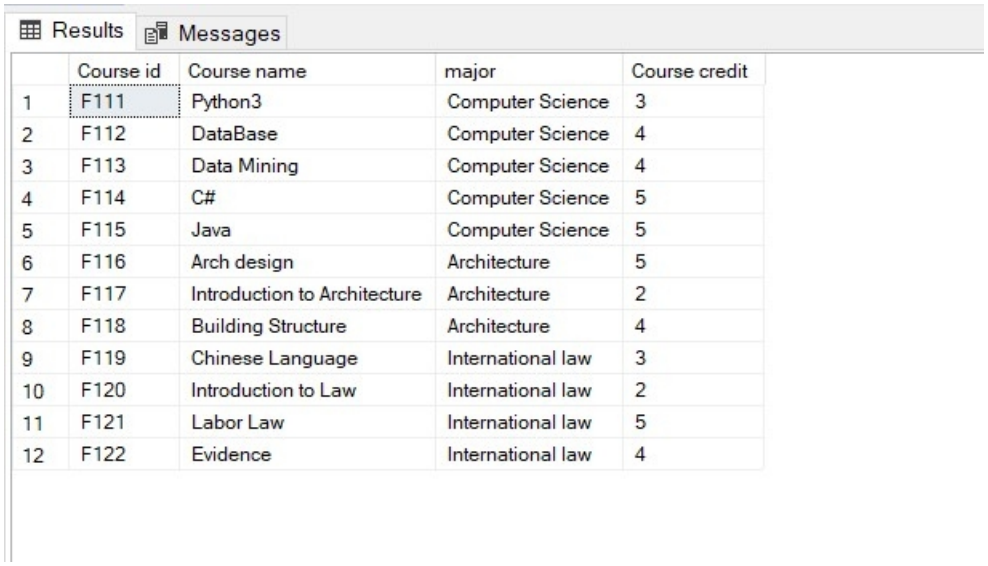
Results Messages

	Teacher id	Teacher name	sex	Phone	Office
1	A11	Jhon	M	13455554444	B101 guang building
2	A12	Ellena	F	13344445555	A202 art building
3	A13	Jaims	M	13355554444	B303 guang building
4	C11	Jhon	M	15566668888	A506 renhe building
5	C12	Ronald	M	15566662222	A505 main building
6	C13	Avin	F	15533336666	A507 renhe building
7	L11	Rose	F	14455556666	C405 Law building
8	L12	Zed	M	14466665555	D313 Law building

❑Course

I didn't put so many courses just tried to keep it simple,

The courses info is below:



Results Messages

	Course id	Course name	major	Course credit
1	F111	Python3	Computer Science	3
2	F112	DataBase	Computer Science	4
3	F113	Data Mining	Computer Science	4
4	F114	C#	Computer Science	5
5	F115	Java	Computer Science	5
6	F116	Arch design	Architecture	5
7	F117	Introduction to Architecture	Architecture	2
8	F118	Building Structure	Architecture	4
9	F119	Chinese Language	International law	3
10	F120	Introduction to Law	International law	2
11	F121	Labor Law	International law	5
12	F122	Evidence	International law	4

❑Section

Results Messages

	Section id	Course id	Semester	Year	Teaching location
1	1.1	F111	1	2018	guang building
2	2.3	F112	1	2018	computer building
3	5.2	F113	2	2018	renhe building
4	1.2	F114	1	2020	guang building
5	4.3	F115	2	2019	computer building
6	3.3	F116	1	2018	art building
7	1.1	F117	2	2020	art buidling
8	2.5	F118	2	2019	art building
9	2.9	F119	1	2019	guang building
10	3.7	F120	2	2018	law building
11	1.2	F121	1	2019	law building
12	3.3	F122	1	2020	law building

❑Class

Considered that the class is the group of people in the

Same major, there are 4 or 3 people in each class an each major have only one class

Results Messages

	Class id	Class major	Number_Student
1	C1	Computer Science	4
2	C2	International law	3
3	C3	Architecture	4

❑Student

Results Messages

	Student id	Name	Sex	country	Total credit	Class
1	L201726630117	Shonon	M	Bangladesh	4	C1
2	L201726630118	Jack	M	India	5	C1
3	L201726630119	Bob	M	Japan	3.5	C1
4	L201726630120	April	F	Italy	2	C1
5	L201727750111	Sofia	F	Zimbabwe	4	C2
6	L201727750112	Danny	M	Syria	5	C2
7	L201727750113	Lily	F	Germany	3	C2
8	L201728830111	Mike	M	Tanzania	2.5	C3
9	L201728830112	Kane	M	England	4.6	C3
10	L201728830113	Sera	F	Canada	4.8	C3
11	L201728830114	Alro	M	America	3.9	C3

[Note: I randomly filled the Total credits when I created student table but the total credit values can always be corrected using the right calculation query]

❑ Teach

Results | Messages

	Teacher id	Course id
1	A11	F116
2	A12	F117
3	A13	F118
4	L11	F119
5	L12	F120
6	C11	F111
7	C12	F112
8	C12	F115
9	C13	F113
10	C13	F114
11	C11	F115
12	L11	F121
13	L12	F122

❑Mark

Results | Messages

	Course id	Student id	course marks	Semester
1	F111	L201726630117	95	1
2	F112	L201726630117	97	1
3	F113	L201726630118	80	2
4	F115	L201726630119	75	2
5	F116	L201728830111	89	1
6	F117	L201728830111	96	2
7	F118	L201728830112	66	2
8	F119	L201727750111	77	1
9	F120	L201727750111	60	2
10	F121	L201727750112	77	1
11	F122	L201727750113	65	1
12	F111	L201726630119	88	1
13	F112	L201726630120	74	1
14	F113	L201726630120	80	2
15	F114	L201726630120	66	1
16	F114	L201726630118	95	1
17	F116	L201728830113	97	1
18	F117	L201728830113	70	2
19	F118	L201728830113	80	2
20	F119	L201727750113	66	1
21	F120	L201727750112	90	2
22	F121	L201727750111	95	1
23	F122	L201727750112	70	1

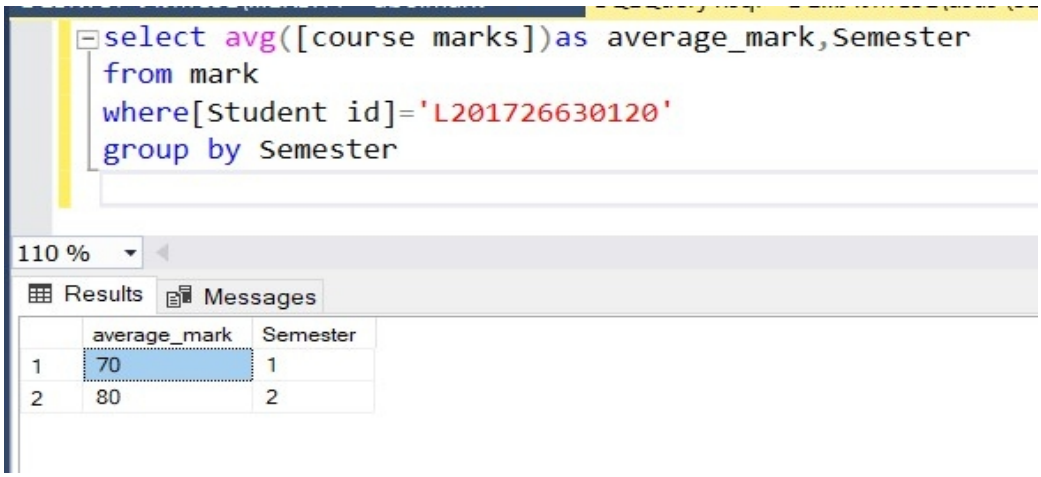
4.Project Realization(Functions and querying)

❑To list one student's marks of all courses grouping by semester

☞SQL QUERY:

```
select avg([course marks])as average_mark,Semester
from mark
where[Student id]='L201726630120'
group by Semester
```

➥Screenshot:



❑To list all students' marks about some course in the ascending or decreasing order. And each querying only lists the marks of students in one class

☞SQL QUERY:

```
select[Course id],[Student id],[course marks]
from mark
where[Course id]='F121'
order by [course marks] desc

select[Course id],[Student id],[course marks]
from mark
where [Course id]='F121'
order by[course marks] asc
```

➥Screenshot:

```
select[Course id],[Student id],[course marks]
from mark
where[Course id]='F121'
order by [course marks] desc
```

110 %

Results | Messages

	Course id	Student id	course marks
1	F121	L201727750111	95
2	F121	L201727750112	77

```
select[Course id],[Student id],[course marks]
from mark
where [Course id]='F121'
order by[course marks] asc
```

110 %

Results | Messages

	Course id	Student id	course marks
1	F121	L201727750112	77
2	F121	L201727750111	95

❑To compute the average marks of each course and the average marks can be grouped by class

☞SQL QUERY:

```
select distinct student.Class,avg(mark.[course marks]) as average
from mark,Student
where mark.[Student id]=Student.[Student id]
group by Student.class
```

➥Screenshot:

```
select distinct student.Class,avg(mark.[course marks]) as average
from mark,Student
where mark.[Student id]=Student.[Student id]
group by Student.class
```

110 %

Results | Messages

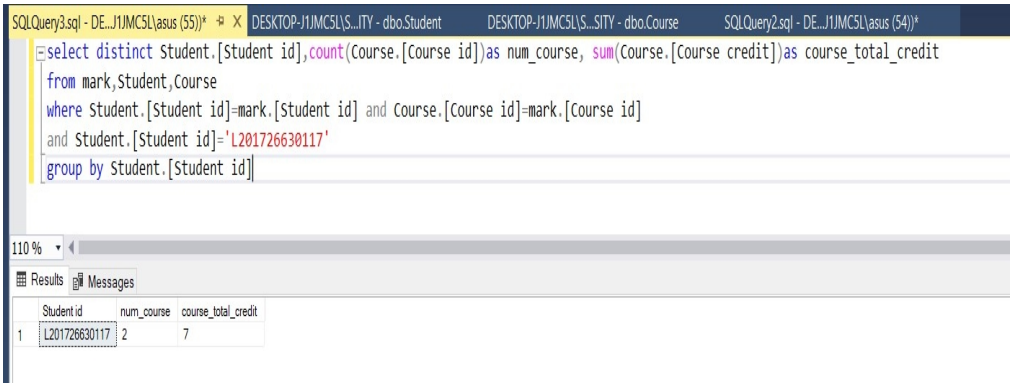
	Class	average
1	C1	83.3333333333333
2	C2	75
3	C3	83

❑**To count the number of courses that one student has learned, and compute his total credits**

☞SQL QUERY:

```
select distinct Student.[Student id],count(Course.[Course id])as num_course, sum(Course.[Course credit])as course_total_credit
from mark,Student,Course
where Student.[Student id]=mark.[Student id] and Course.[Course id]=mark.[Course id]
and Student.[Student id]='L201726630117'
group by Student.[Student id]
```

➥Screenshot:

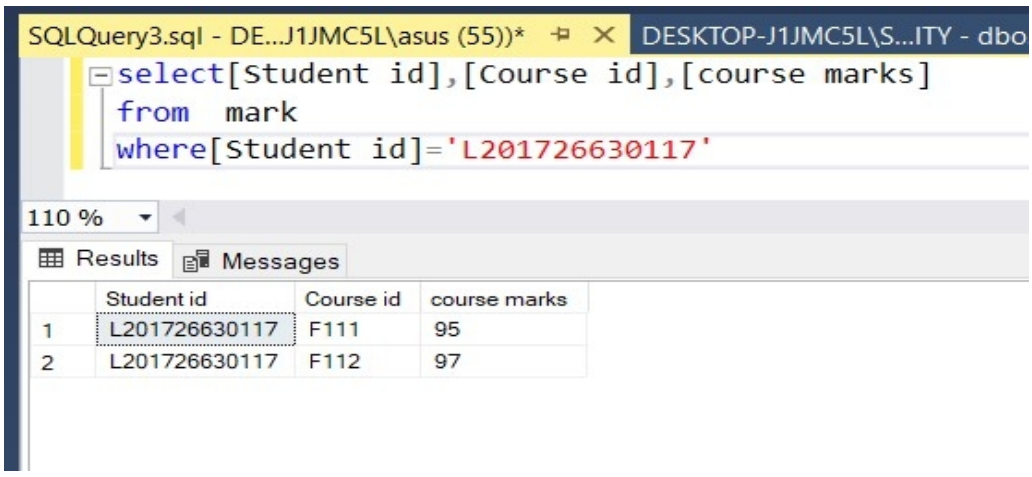


❑**To query marks of one student about some course**

☞SQL QUERY:

```
select[Student id],[Course id],[course marks]
from  mark
where[Student id]='L201726630117'
```

➥Screenshot:

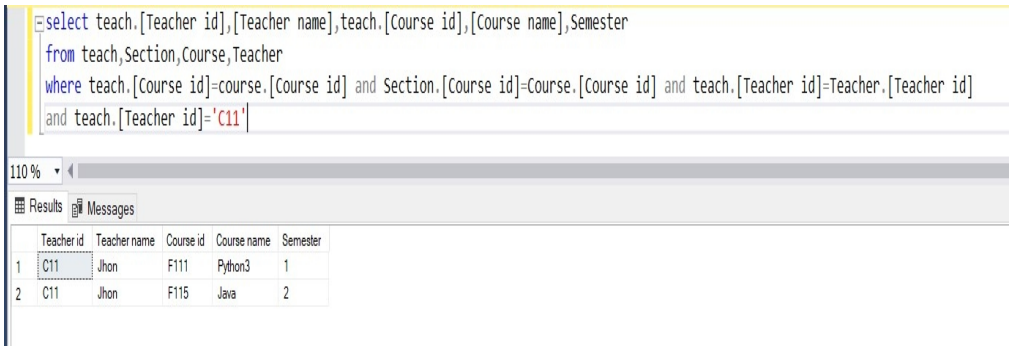


❑**To query courses that are taught by some teacher in one semester**

☞SQL QUERY:

```
select teach.[Teacher id],[Teacher name],teach.[Course id],[Course name],Semester
from teach,Section,Course,Teacher
where teach.[Course id]=course.[Course id] and Section.[Course id]=Course.[Course id] and teach.[Teacher id]=Teacher.[Teacher id]
and teach.[Teacher id]='C11'
```

➥Screenshot:

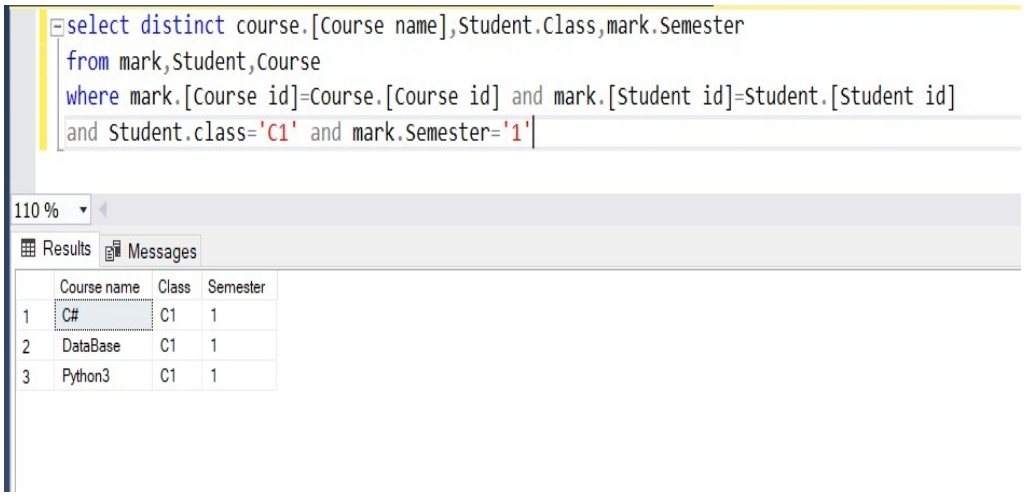


❑**To query courses that are taken by some class in one semester**

☞SQL QUERY:

```
select distinct course.[Course name],Student.Class,mark.Semester
from mark,Student,Course
where mark.[Course id]=Course.[Course id] and mark.[Student id]=Student.[Student id]
and Student.class='C1' and mark.Semester='1'
```

➥Screenshot:



4.1 Other Possible Functions and querying that I can image .

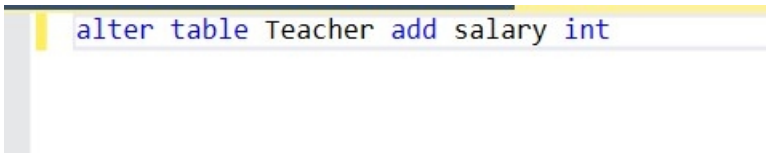
(This functions and querying also frequently use at work field.)

❑For create new attribute in a table

SQL CODE:

alter table Teacher add salary int

Screenshot:



	Teacher id	Teacher na...	sex	Phone	Office	salary
▶	A11	Jhon	M	13455554444	B101 guang...	NULL
	A12	Ellena	F	13344445555	A202 art bui...	NULL
	A13	Jaims	M	13355554444	B303 guang...	NULL
	C11	Jhon	M	15566668888	A506 renhe ...	NULL
	C12	Ronald	M	15566662222	A505 main ...	NULL
	C13	Avin	F	15533336666	A507 renhe ...	NULL
	L11	Rose	F	14455556666	C405 Law b...	NULL
	L12	Zed	M	14466665555	D313 Law b...	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Note:[I added new a attribute name salary in Teacher table]

❑Now I am showing “like” operator and two special characters (percent(%) and underscore(_)) in where clauses

SQL CODE:

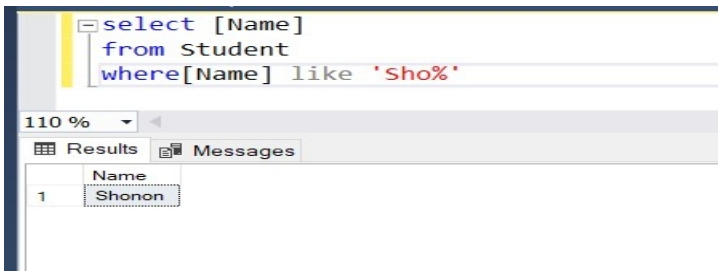
➥Use: %

select [Name]

from Student

where[Name] like 'Sho%'

Screenshot:



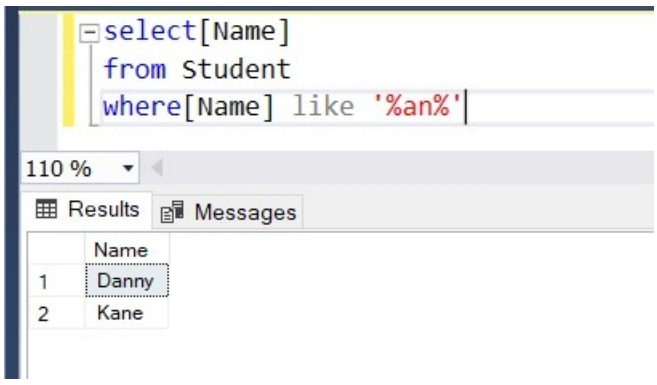
Note:[it matched any string after [‘Sho’]

select[Name]

from Student

where[Name] like '%an%'

Screenshot:



Note:[>it matched any string that contains 'an']

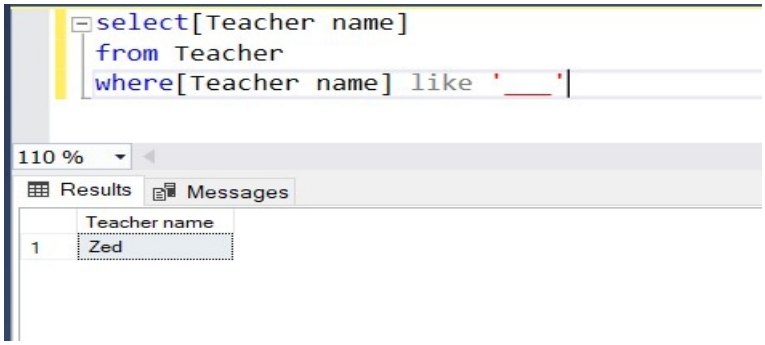
➥Use:(_)

select[Teacher name]

from Teacher

where[Teacher name] like '___'

Screenshot:

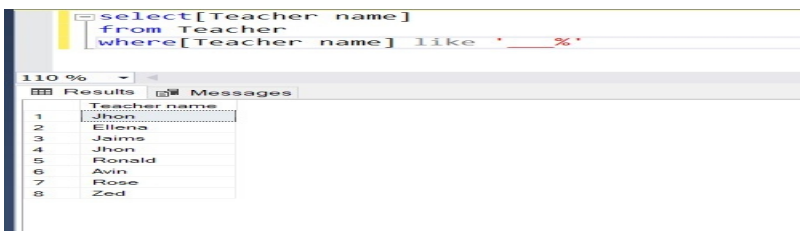


Note:[it exactly matched the number of underscores.For this result underscores used 3]

select[Teacher name]

from Teacher

where[Teacher name] like '___%'



Note:[it matched at least 3 characters. ==>3)

❑ **For this functions I can also change any table's content I am showing a example**

Before update data table content was from mark table.

	Course id	Student id	course mar...	Semester
▶	F111	L201726630...	95	1
	F112	L201726630...	97	1
	F113	L201726630...	80	2
	F115	L201726630...	75	2
	F116	L201728830...	89	1
	F117	L201728830...	96	2
	F118	L201728830...	66	2
	F119	L201727750...	77	1
	F120	L201727750...	60	2
	F121	L201727750...	77	1
	F122	L201727750...	65	1
	F111	L201726630...	88	1
	F112	L201726630...	74	1
	F113	L201726630...	80	2
	F114	L201726630...	66	1
	F114	L201726630...	95	1
	F116	L201728830...	97	1
	F117	L201728830...	70	2
	F118	L201728830...	80	2
	F119	L201727750...	66	1
	F120	L201727750...	90	2
	F121	L201727750...	95	1
	F122	L201727750...	70	1
*	NULL	NULL	NULL	NULL

➥After using update

update mark

set [course marks]=100

where [course marks] =97

Screenshot:

	Course id	Student id	course mar...	Semester
▶	F111	L201726630...	95	1
	F112	L201726630...	100	1
	F113	L201726630...	80	2
	F115	L201726630...	75	2
	F116	L201728830...	89	1
	F117	L201728830...	96	2
	F118	L201728830...	66	2
	F119	L201727750...	77	1
	F120	L201727750...	60	2
	F121	L201727750...	77	1
	F122	L201727750...	65	1
	F111	L201726630...	88	1
	F112	L201726630...	74	1
	F113	L201726630...	80	2
	F114	L201726630...	66	1
	F114	L201726630...	95	1
	F116	L201728830...	100	1
	F117	L201728830...	70	2
	F118	L201728830...	80	2
	F119	L201727750...	66	1
	F120	L201727750...	90	2
	F121	L201727750...	95	1
	F122	L201727750...	70	1
*	NULL	NULL	NULL	NULL

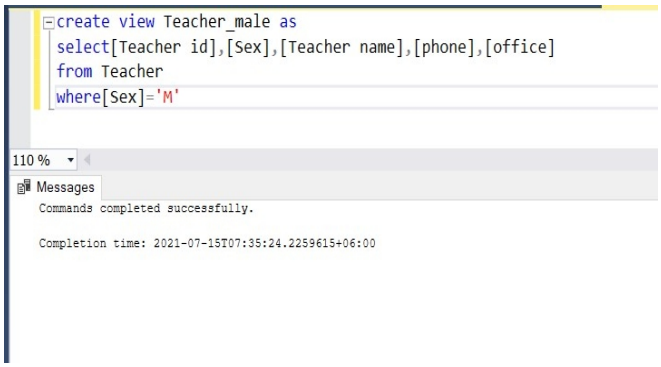
Note:[mark change 97 to 100 from course mark attribute]

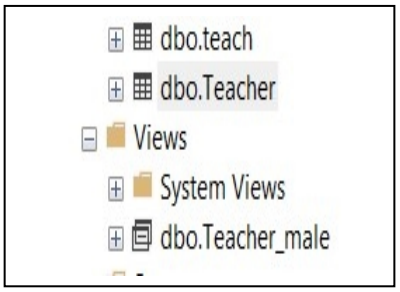
❑**Using view:A view provides a mechanism to hide certain data from the view of certain users.**

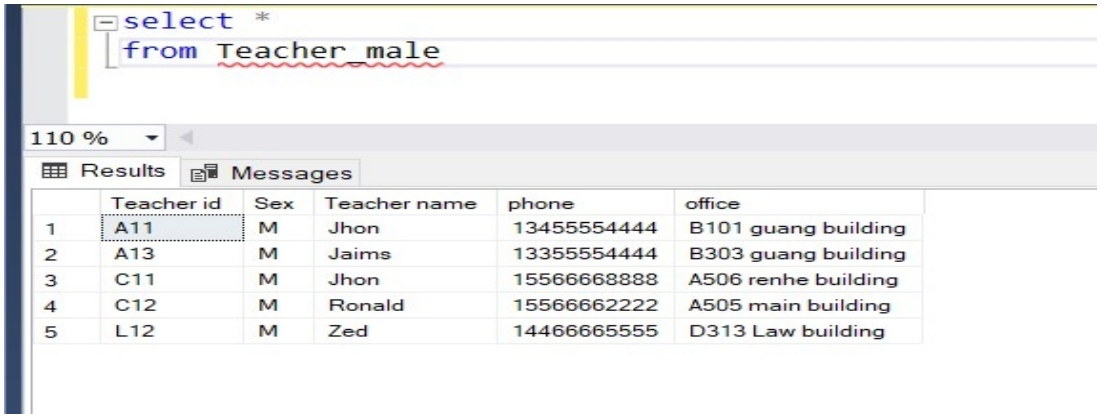
SQL CODE:

```
create view Teacher_male as
select[Teacher id],[Sex],[Teacher name],[phone],[office]
from Teacher
where[Sex]='M'
```

Screenshot:







	Teacher id	Sex	Teacher name	phone	office
1	A11	M	Jhon	13455554444	B101 guang building
2	A13	M	Jaims	13355554444	B303 guang building
3	C11	M	Jhon	15566668888	A506 renhe building
4	C12	M	Ronald	15566662222	A505 main building
5	L12	M	Zed	14466665555	D313 Law building

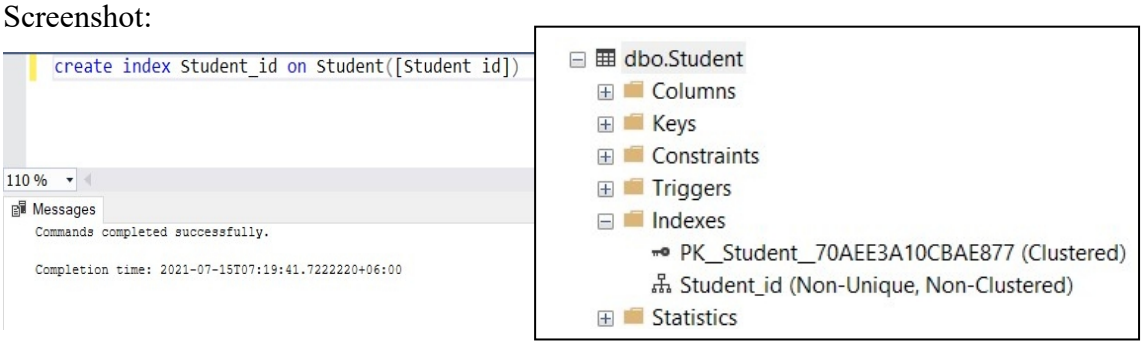
Note:[Once a view is defined, the view name can be used to refer to the virtual relation that the view generates.View definition is not the same as creating a new relation by evaluating the query expression Rather, a view definition causes the saving of an expression; the expression is substituted into queries using the view.View also allow insert new data]

❑**Use of index.Here I created index to a column.The use of the indexing is that if I have a database with lots of data it makes the search much faster and easier.**

SQL CODE:

create index Student_id on Student([Student id]

Screenshot:



❑**I created a table with some with build in data_types and inserted a value to it :**

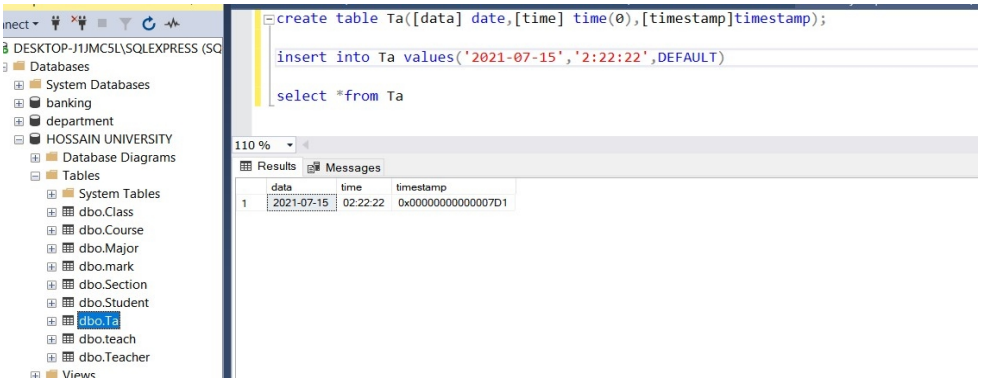
SQL CODE:

create table Ta([data] date,[time] time(0),[timestamp]timestamp);

insert into Ta values('2021-07-15','2:22:22',DEFAULT)

select *from Ta

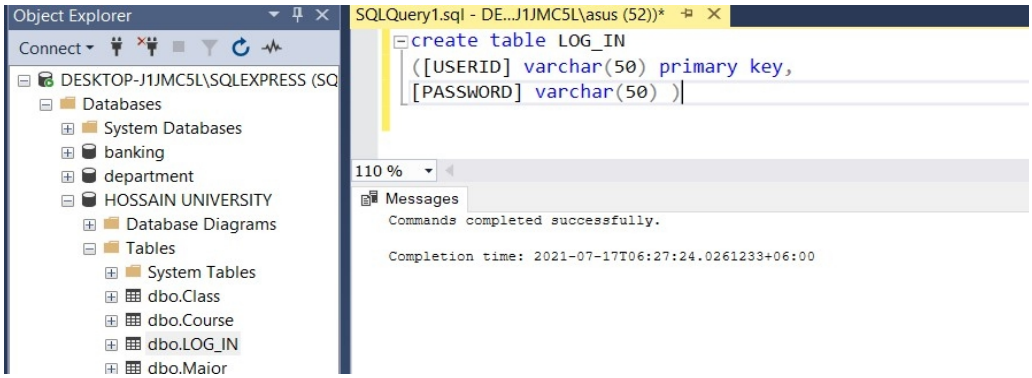
Screenshot:

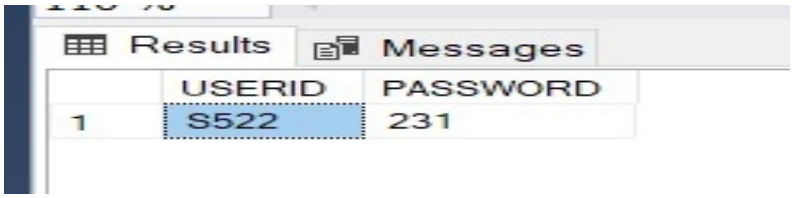


5.Designing GUI for Database with C#

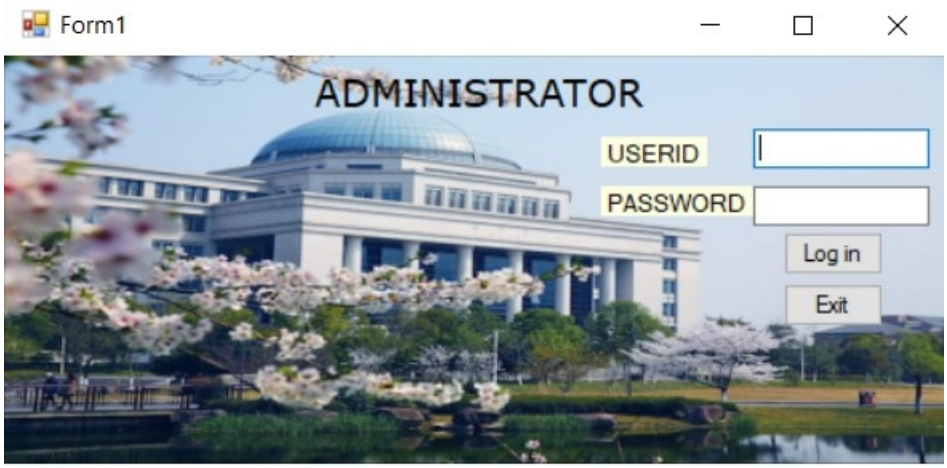
I used the C# programming language to design the GUI and create the connection to the HOSSAIN UNIVERSITY database. I considered just the administrator login to the system,of course it would had been better if added students login and granted different access permissions between them and the administrator but again I wanted to make it simple. since I'm solving the administrators' problem,Anyway to do that I had to add a LOG_IN table that contains the user id in my case it's only "the administrator id" and the password.

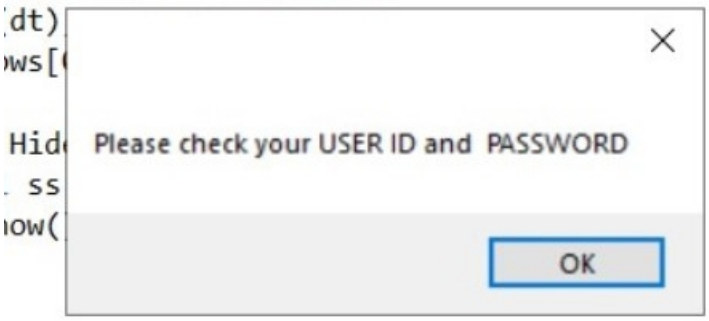
❑SQL CODE WITH OUTPUT:





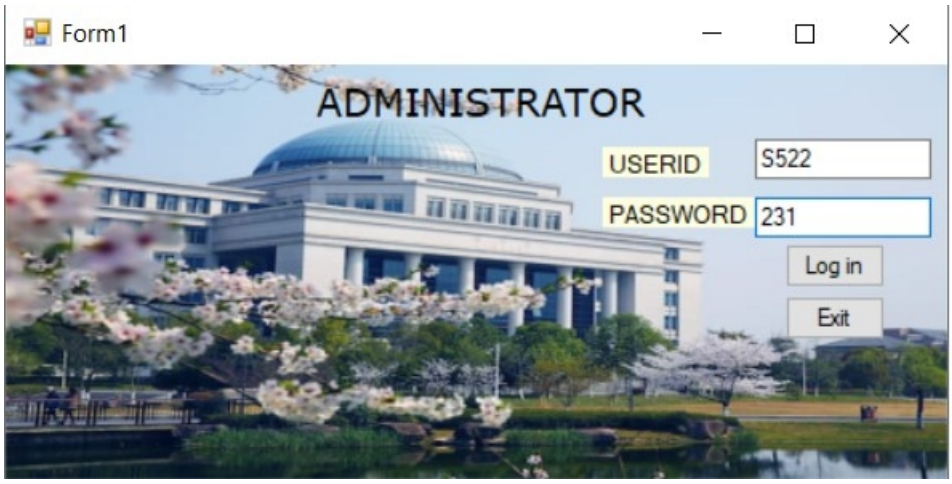
☞So, when open the app and following window appears:

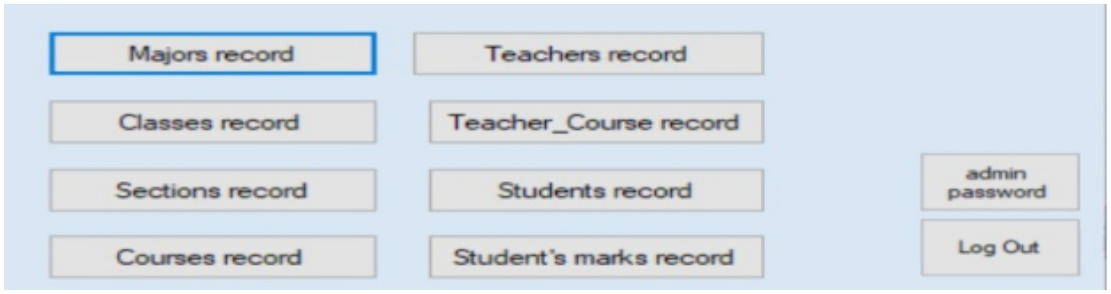




➥Note:[If the user enter the wrong password this message will appear]

If the user id and password are correct the window that accesses the university database will open:





➥ Note:[The admin can view each table in the data base on the grid view by clicking the buttons he/she can log out and that would make him/her return to the log in window from there he can exit or log in again]

❑Some pictures of my C# programming code:

```
private void button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-J1JMC5L\SQLEXPRESS;Initial Catalog=HOSSAIN UNIVERSITY;Integrated Security=True");

    SqlDataAdapter sda = new SqlDataAdapter("select * from LOG_IN where [USERID] = 'S522' and [PASSWORD]='231'", con);
    DataTable dt = new DataTable();
    sda.Fill(dt);
    if (dt.Rows[0][0].ToString() == "1")
    {
        this.Hide();
        Form1 ss = new Form1();
        ss.Show();
    }
    else
    {
        MessageBox.Show("Please check your USER ID and  PASSWORD");
    }
```

```
string connectionString = @"Data Source=DESKTOP-J1JMC5L\SQLEXPRESS;Initial Catalog=HOSSAIN UNIVERSITY;Integrated Security=True";
SqlDataAdapter sda = new SqlDataAdapter();
SqlCommandBuilder scb1 = new SqlCommandBuilder();
DataTable dt = new DataTable();
DataGridView dgv1 = new DataGridView();

private void bclass_Click(object sender, EventArgs e)
{
    using (SqlConnection sqlcon = new SqlConnection(connectionString))
    {

        sda = new SqlDataAdapter("SELECT *FROM Class", sqlcon);
        dt = new DataTable();
        sda.Fill(dt);
        DataGridView dgv1 = new DataGridView();
        dgv1.DataSource = dt;
    }


}
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
public Form1()
{
    InitializeComponent();
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

    private void label2_Click(object sender, EventArgs e)
    {

    }
    public string conString = "Data Source=DESKTOP-J1JMC5L\\SQLEXPRESS;Initial Catalog=HOSSAIN UNIVERSITY;Integrated Security=True";
    private void button1_Click(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection(conString);
        con.Open();
        if(con.State==System.Data.ConnectionState.Open)
        {
            string q = "insert into Log in(USERID,PASSWORD)values('" + txtUSERID.Text.ToString() + "','"+txtPASSWORD.Text.ToString()+"')";
            SqlCommand cmd = new SqlCommand(q, con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Log in Successfuly...!");
        }
    }
}
}
```

6.Project Conclusion:

As I am at the end of this report, it's time to express my thoughts and about my work experience on what I learned from this topic and how I deal with it.In this project I learnt how to implement what I studied during this semester to a real world problem. Creating tables having relationships and drawing E-R diagram to represent this relationships,SQL sentence,how to explain etc.First when I started doing the project I think the most part that I struggled with was the design of the E_R diagram because there are number of different ways to design it when I started thinking about it Following personal logic could result in a diagram, following the exact requirement, even if some doesn't make sense, could also result in a diagram, and trying to merge personal logic with the requirement will give another different diagram, and of course designing a bad diagram could mess up everything, so drawing the design was kind of confusing but in a way it's also very interesting it took me almost two days to finally decide to choose what diagram I wanted to go with, then implementing the database and the queries was much simpler but filling the tables took more time than the implantation of the database and the queries . now I understand why the administrator was bored and why I don't wish to do his job in the future **however after finishing The requirements for additional marks I decided that I want to try making the GUI for the database so it I ended up making it super simple but learning how to make it was very fun. but I didn't 100% complete it .** The admin can view all the tables but can't modify them which is most of the point of having admin user . I wanted the admin to be able to directly modify the tables from the grid view but I kept getting error and didn't manage to solve it so I just went with just what I already did .Finally, I think that the databases are very powerful tools to solve a lot of our real life problems and make it easy for us to deal with a lot of important tasks and are essential for programs to understand and above everything it's fun to design and create our own databases .