Object Oriented Programming

Lab Manual 06

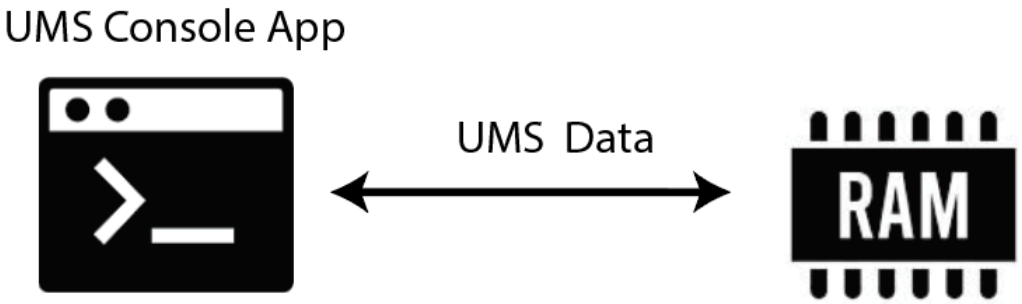
Introduction

After a week of rigorous coding, Welcome back!

This lab Agenda.

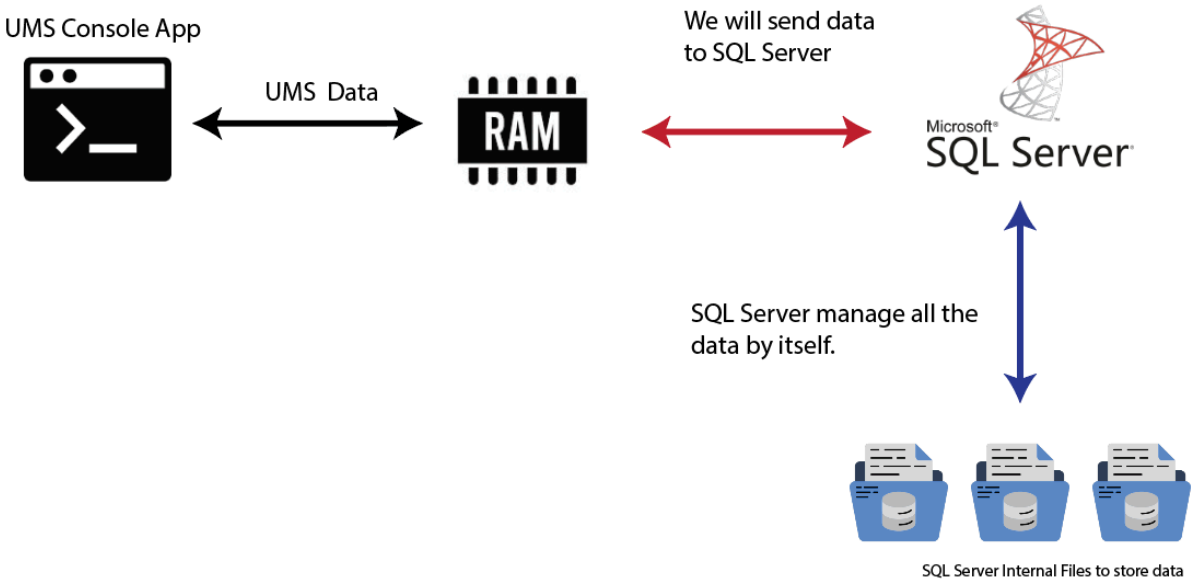
In this lab, we're doing some exciting work. Before, we stored data in our computer's memory and worked with it. Later, we tried saving data in text files (.txt), which was okay but slow and messy, especially with lots of data and tasks. So, now we're using a Database System to keep things organized and speedy.
What's the plan? We connect our code to a special database server and send our data there using SQL commands. The server takes care of everything. This is a common method used by many websites and systems to handle their data.

We have started with this.



We did this.

The text files are not recommended when we have many operations and different types of data to store. We use the Database Servers, and we just provide data to the Server so its Server responsibility to store the data and keep it available to the application with best way.

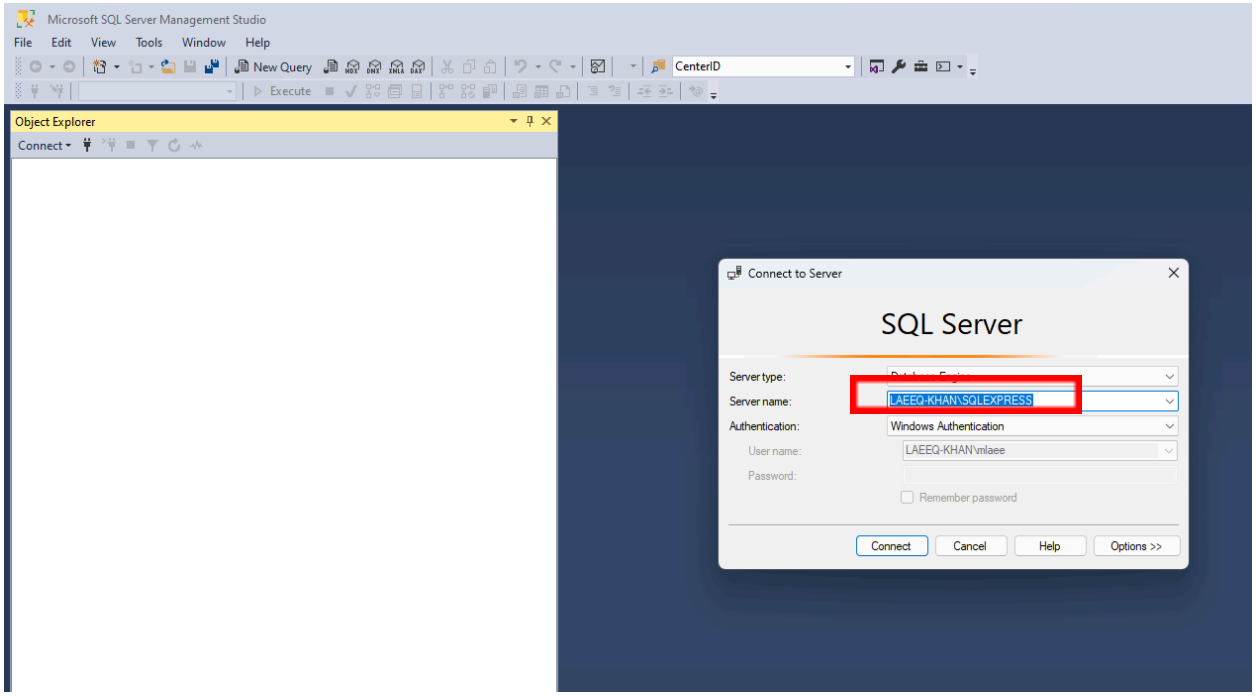


Let's do it.

Step 1: Make sure you have installed SQL Database Server and SQL Management Studio 19.

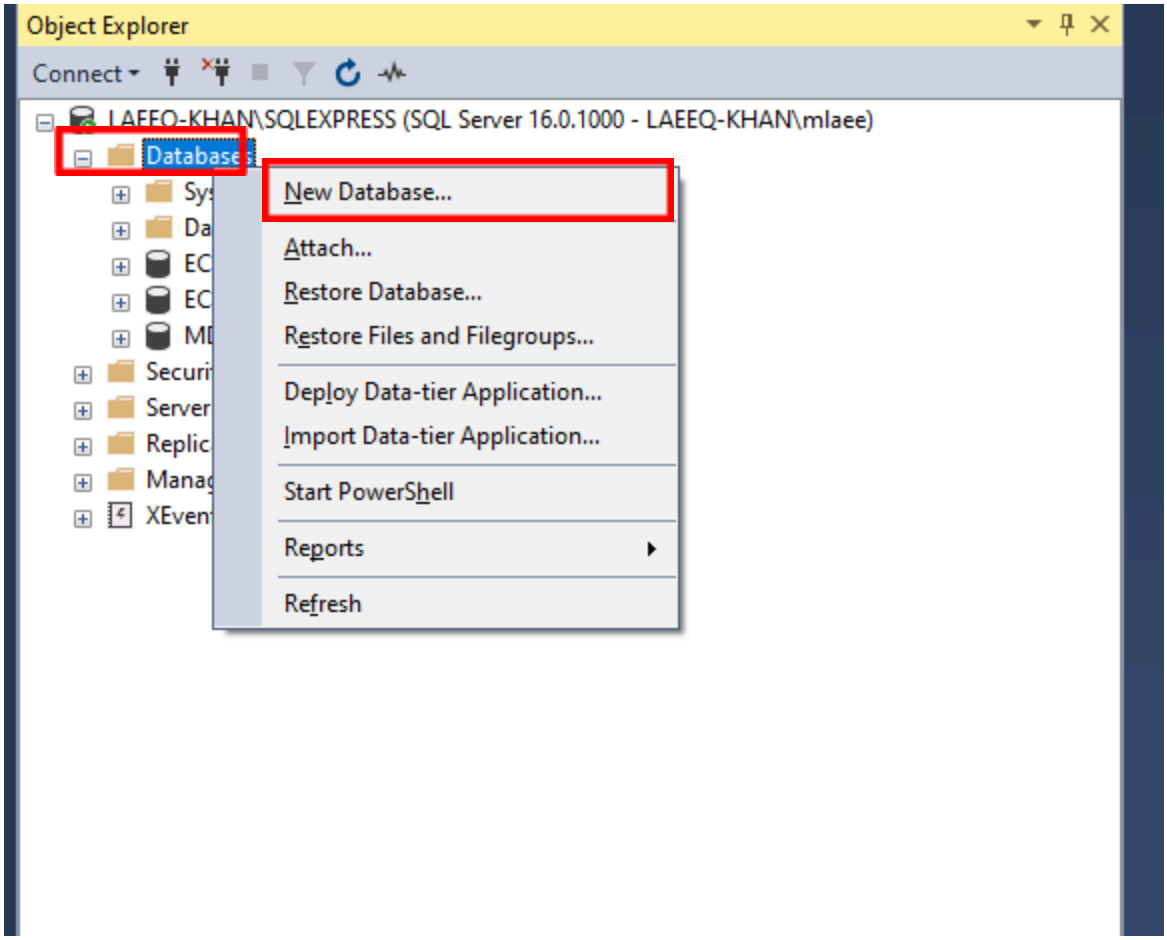
Step 2: Get the connection string. If you don't know the connection string following procedure.

Open the SQL Server Management Studio

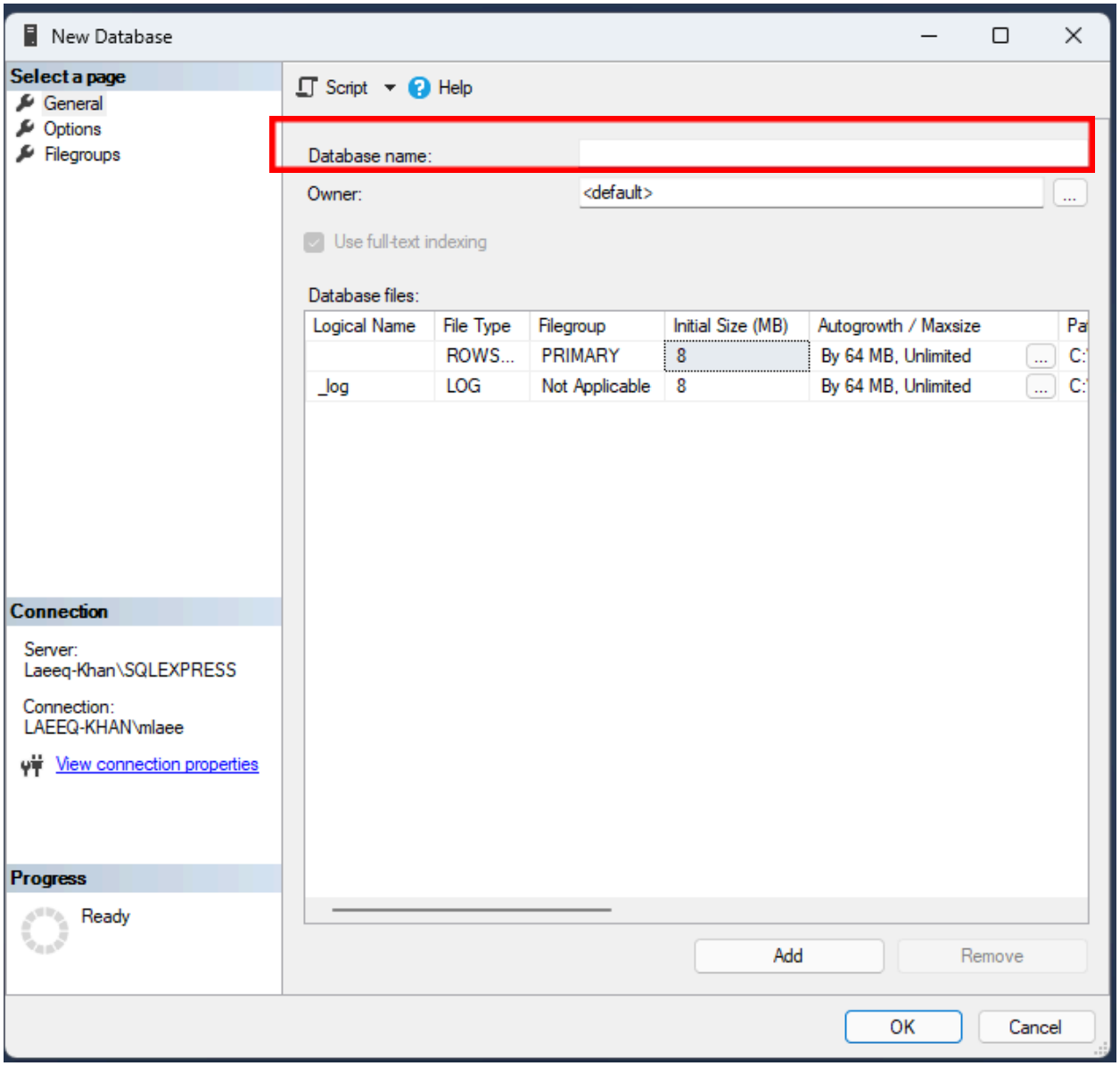


Copy the Server Name into the file and click on the **Connect** Button.

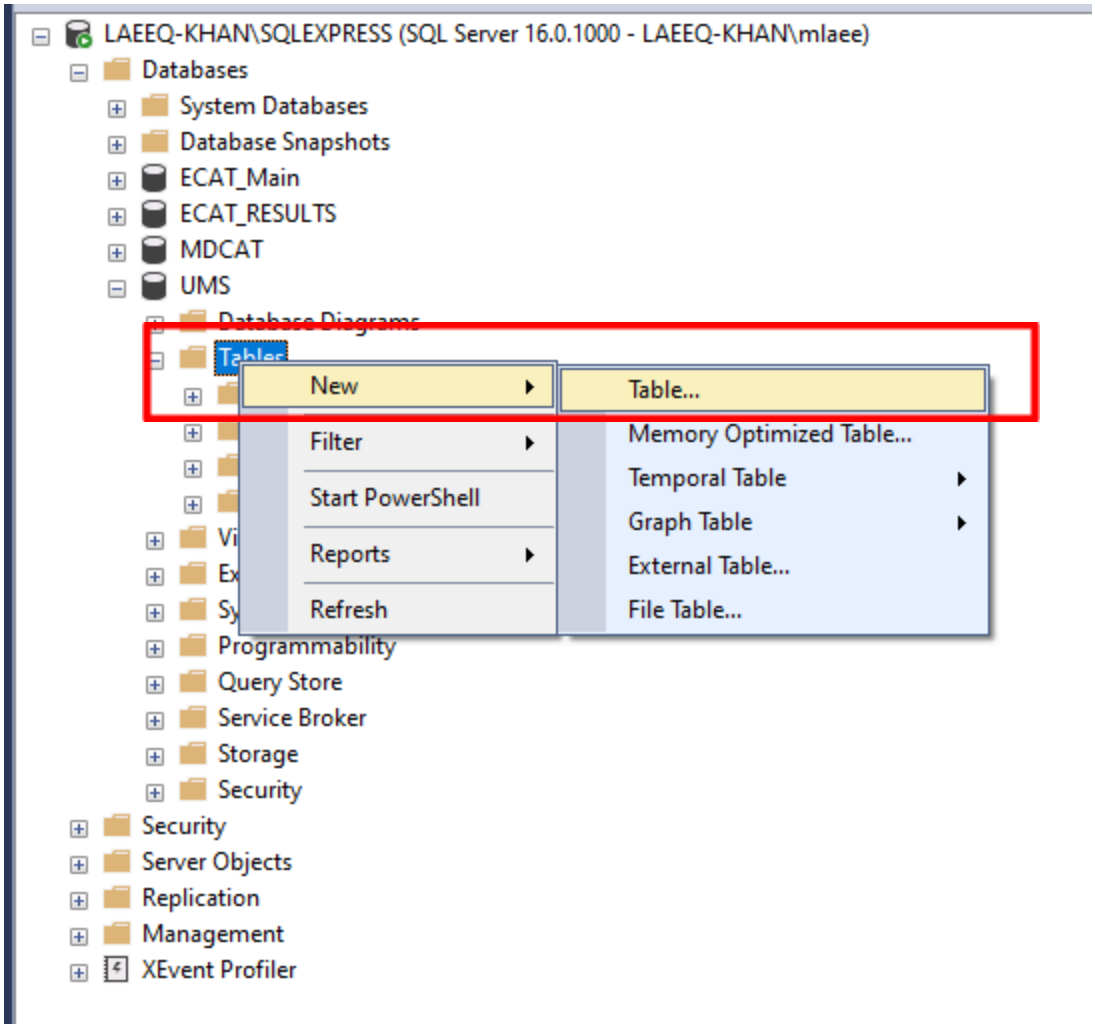
Create your database by clicking on Databases option in the **Object Explorer** and then choose **New Database** option from the drop down.



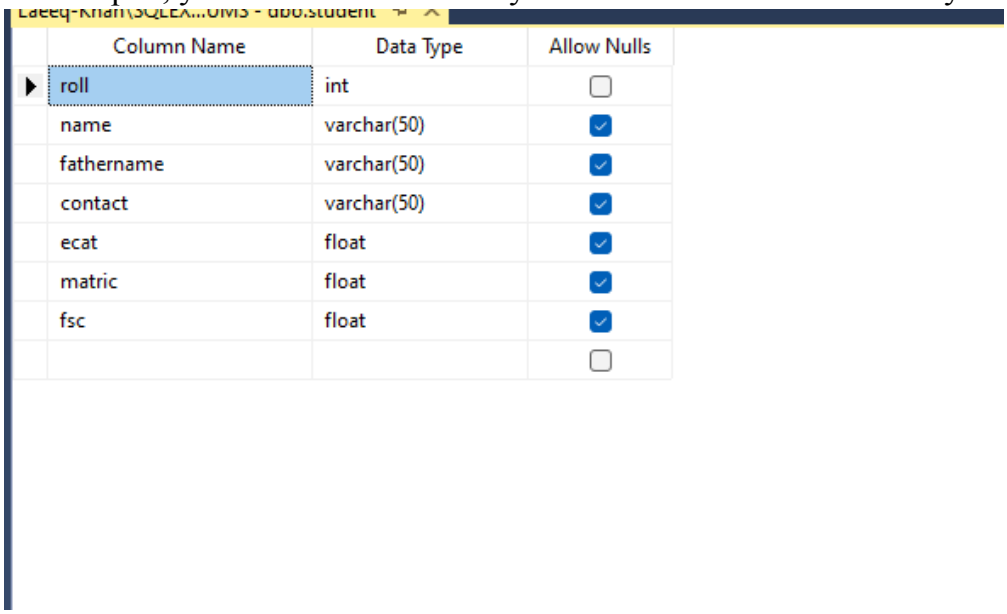
Write your database name like UMS or whatever you want.



After creating your database, create the tables in the database by clicking on the table and then you can choose **New > Table** option from the drop down lists.



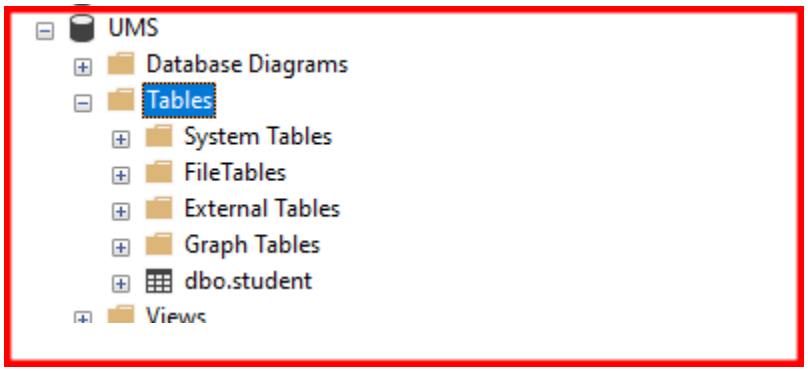
If will open, you a new window and you can create the columns for your table



Column Name	Data Type	Allow Nulls
roll	int	☐
name	varchar(50)	☑
fathername	varchar(50)	☑
contact	varchar(50)	☑
ecat	float	☑
matric	float	☑
fsc	float	☑
		☐

After this Save your table by using **Control + S** command

I have saved that as student table.



Let's build the connection string.
Server=**Your Server Name**;Database=**Your Database Name**;Trusted_Connection=True;

My Server Name is LAEEQ-KHAN\SQLEXPRESS and Database name is UMS.
My connection string will look like this
Server=LAEEQ-KHAN\SQLEXPRESS;Database=UMS;Trusted_Connection=True;
If you have Password base authentication on your SQL Server, you can build your connection string like this
Server=**myServerAddress**;Database=**myDatabase**;UserId=**myUsername**;Password=**my Password**;

Writing the Code
Create your business layer classes (You have to use your OLD UMS Project that you converted into BL, DL and UI layer

My student class code will look like this. All data members are private and a parameterized constructor and getter and setters for the all data members.

```
1 reference
internal class Student
{
    private string Roll;
    private string Name;
    private string FatherName;
    private string Contact;
    private float Ecat;
    private float Matric;
    private float FSc;

    0 references
    public Student(string roll, string name, string fatherName, string contact, float ecat, float matric, float fSc)
    {
        Roll=roll;
        Name=name;
        FatherName=fatherName;
        Contact=contact;
        Ecat=ecat;
        Matric=matric;
        FSc=fSc;
    }

    0 references
    public void SetRoll(string roll)
    {
        Roll=roll;
    }
    0 references
    public string GetRoll()
    {
        return Roll;
    }

}
```

Let’s configure your connection string in your Visual Studio to be used inside the code.
Student DL Layer Code.

```
2 references
internal class SudentDL
{
    public static string connectionString = "Server=LAEEQ-KHAN\\SQLEXPRESS;Database=UMS;Trusted_Connection=True;";

    1 reference
    public static Student CreateStudent(Student student)
    {
        string query = string.Format("INSERT INTO Students (Roll, Name, FatherName, Contact, Ecat, Matric, Fsc) " +
                           "VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}')",
                           student.GetRoll(), student.GetName(), student.GetFatherName(),
                           student.GetContact(), student.GetEcat(), student.GetMatric(), student.GetFsc());

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            SqlCommand command = new SqlCommand(query, connection);

            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected > 0) return student;

        }
        return null;
    }

    1 reference
    public static List<Student> GetAllStudents()
    {
        List<Student> students = new List<Student>();
        string query = "SELECT * FROM Students";

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            SqlCommand command = new SqlCommand(query, connection);

            connection.Open();
            SqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                Student student = new Student(
                    Convert.ToString(reader["Roll"]),    // or reader.GetString(1)
                    Convert.ToString(reader["Name"]),    // or reader.GetString(2)
                    Convert.ToString(reader["FatherName"]), // or reader.GetString(3)
                    Convert.ToString(reader["Contact"]), // or reader.GetString(4)
                    Convert.ToSingle(reader["Ecat"]),    // or reader.GetString(5)
                    Convert.ToSingle(reader["Matric"]), // or reader.GetString(6)
                    Convert.ToSingle(reader["Fsc"])     // or reader.GetString(7)
                );
                students.Add(student);
            }
        }

        return students;
    }
```

```
0 references
public static Student ReadStudent(int roll)
{
    string query = string.Format("SELECT * FROM Students WHERE Roll = '{0}'", roll);

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(query, connection);

        connection.Open();
        SqlDataReader reader = command.ExecuteReader();

        if (reader.Read())
        {
            return new Student(
                Convert.ToString(reader["Roll"]),  // or reader.GetString(1)
                Convert.ToString(reader["Name"]),   //so on
                Convert.ToString(reader["FatherName"]),
                Convert.ToString(reader["Contact"]),
                Convert.ToSingle(reader["Ecat"]),
                Convert.ToSingle(reader["Matric"]),
                Convert.ToSingle(reader["Fsc"])
            );
        }
        else
        {
            return null;
        }
    }
}
```

```
0 references
public static Student UpdateStudent(Student student)
{
    string query = string.Format("UPDATE Students SET Name = '{0}', FatherName = '{1}', Contact = '{2}', " +
                        "Ecat = '{3}', Matric = '{4}', Fsc = '{5}' WHERE Roll = '{6}'",
                        student.GetName(), student.GetFatherName(), student.GetContact(),
                        student.GetEcat(), student.GetMatric(), student.GetFsc(), student.GetRoll());


    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(query, connection);

        connection.Open();
        int rowsAffected =  command.ExecuteNonQuery();
        if(rowsAffected > 0) return student;
    }

    return null;
}
```

```
0 references
public static bool DeleteStudent(int roll)
{
    string query = string.Format("DELETE FROM Students WHERE Roll = '{0}'", roll);

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(query, connection);

        connection.Open();
        int rowsAffected = command.ExecuteNonQuery();
        if (rowsAffected > 0) return true;
        else return false;
    }
}
```

Driver code

```
namespace Labwork
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {

            //Driver code, you can write your menu code here

            Student student = new Student("1", "Ali", "Khan", "03055466154", 320, 500, 600);
            SudentDL.CreateStudent(student);


            List<Student> studentList = SudentDL.GetAllStudents();



        }
    }
}
```

Complete your UMS with all features and write all your database functions in the DL layers.