



Instructor:

CLO

- Mr. Nazeef Ul Haq (Lab)

Learning Objectives:

- CRUD operations using C# language with SQL server

Helping Material:

SQL CRUD Operations Queries Syntax

- Insert Data into Table:
`INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);`
- Update Data into Table:
`UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;`
- Retrieve Data from Table:
`SELECT column1, column2, ...
FROM table_name;`
- Delete Data from a Table:
`DELETE FROM table_name WHERE condition;`

Connection of SQL using C#

C# code to insert data in a table while clicking on a button is given below.

```
var con = Configuration.GetInstance().getConnection();  
SqlCommand cmd = new SqlCommand("Insert into Table_1 values (@ID, @Name,  
@Department)", con);  
cmd.Parameters.AddWithValue("@ID", int.Parse(textBox1.Text));  
cmd.Parameters.AddWithValue("@Name", textBox2.Text);  
cmd.Parameters.AddWithValue("@Department", textBox3.Text);  
cmd.ExecuteNonQuery();  
MessageBox.Show("Successfully saved");
```

C# code to show data in a grid is given below.

```
var con = Configuration.GetInstance().getConnection();  
SqlCommand cmd = new SqlCommand("Select * from Table_1", con);  
SqlDataAdapter da = new SqlDataAdapter(cmd);  
DataTable dt = new DataTable();  
da.Fill(dt);  
dataGridView1.DataSource = dt;
```

Lab Tasks

1. Create a new database named Lab2
2. Create table with following schema
Student(RegistrationNumber:String, Name:String , Department:String , Session:Number, CGPA:Number, Address:String)
3. Use the student table to create the following CRUD operations
 - a. Add a Student
 - b. Edit Student
 - c. List of Students
 - d. Search Student
 - e. Delete Student
4. UI components for the fields should be appropriate. Change the Edit Use case to optimize such that all the fields should not be updated in table, you need to detect, which field has been changed on frontend, then update only those fields which have been changed. E.g. for editing of student, we change registration number, then database should only update only registration number.

Home Task 1 (Section C)

1. Use the following schema to create a database named Lab2_Home.
2. Create following tables
 - a. Student(RegistrationNumber:String, Name:String , Department:String , Session:Number, Address:String)
 - b. Course(Name:String, Code:String)
 - c. Enrollments(StudentRegNo:String, CourseName:String)
 - d. Attendance(StudentRegNo:String, CourseName:String, TimeStamp:datetime,Status:Boolean)
3. Use the CRUD operations on student from Lab task.
4. Create the CRUD operation for courses.
5. Create a form to register a student and unregister from the course.
6. Create a page to mark and edit attendance for a particular course.
7. Each form/page should be responsive.

Home Task 2 (Section B)

1. Use the following schema to create a database named Lab2_Home.
2. Create following tables
 - a. Student(RegistrationNumber:String, Name:String , Department:String , Session:Number, Address:String)
 - b. Course(Course_ID:String, Course_Name:String, Student_Name:String, Teacher_Name:String, Semester:String)
 - c. Enrollments(StudentRegNo:String, CourseName:String)
 - d. Attendance(StudentRegNo:String, CourseName:String, TimeStamp:datetime,Status:Boolean)
3. Use the CRUD operations on student from Lab task.
4. Create the CRUD operation for courses.
5. Create a form to register a student and unregister from the course. It should work for every field e.g. if you want to change only teacher name, semester, course name, or student name etc. It should also show specific field of data whenever required along with all data.

6. Each form/page should be responsive.

Home Task 3 (Section A)

1. Use the following schema to create a database named Lab2_Home.
2. Create following tables
 - a. Student(RegistrationNumber:String, Name:String , Department:String , Session:Number, Address:String)
 - b. Course(Course_ID:String, Course_Name:String, Student_Name:String, Teacher_Name:String, Semester:String)
 - c. Result(Student_ID:String, Student_Name:String, Course_Name:String, Marks:Int, Grade:String, Section:String, Semester:int, Session:Int)
3. Use the CRUD operations on student from Lab task.
4. Create the CRUD operation for courses.
5. Create the CRUD operation for courses.
6. Create a form to enter a marks of student and enter a course. It should work for every field e.g. if you want to change only teacher name, semester, course name, or student name etc. It should also show specific field of data whenever required along with all data. It should also work if you want to update marks of any specific student.
7. Each form/page should be responsive.

What to Submit:

You are required to submit the following files.

- Dbscript.sql
- .cs files form visual studio project only
- No other files other than should be submitted