**The laboratory work 12**

**Introduction**

The acronym CRUD is made up of four words. Those four words are taken from the first letters of the SQL command.

C refers to Create. This means to generate a new request or to add something.

R refers to Read. This means to display all records.

U refers to Update. This means to change the existing records.

D refers to Delete. This means to delete existing records.

**Step 1**

Create a database from MSSQL with the name of the School.

create database school;

use school

create table teacher(id int, Name varchar(50), Address varchar(50),salary int);

select \* from teacher;

SQL

**Table**



**Step 2**

We create a Windows Application project in C#. Now the project is ready for use.



**How to Use the C# Windows Application Project**

The left-hand side has a toolbox for inserting elements into the Windows projects.

Select an element from the toolbox and drop it on the form page.

Each element has properties to change other properties according to the requirements.

Double click any element to redirect its element function.

**Step 3**

Design a front-end part.

Insert four labels from the toolbox and change the name of labels from the label property text.

Select Label from Toolbox and drop on the Form1 Page.



Then, change the label name from Label property Text and write a name as ID, Name, Address, Salary.



**Step 4**

Insert five Text boxes from the toolbox and change the name of Text from the input property name.

Select Textbox from Toolbox and drop on the Form1 Page.



Then, change the name from the textbox property name. We have written the name.



**Step 5**

Insert six buttons from the toolbox and change the button's name from the button property text and name.

Please select the button from Toolbox and drop it on the Form1 Page.



Then, change the name of the button, as well as the text of the button, from the button property to Insert, Update, Delete, Show All, Find, or Exit.



**Step 6**

Insert data grid view from the toolbox.

Select data grid view from Toolbox and drop on the Form1 Page.



**Step 7**

Now the design is complete.



**Step 8**

Here we start to apply CRUD Operations. To double click on the From1 page and create a SqlConnection and SqlCommand object and in a connection object, we store a connection string. The cmd object we store a SQL command and establish the SQL connection.

SqlConnection conn;

SqlCommand cmd;

private void Form1\_Load(object sender, EventArgs e) {

 conn = new SqlConnection(@ "Data Source=LAPTOP-2F8CO37S;Initial Catalog=school;Integrated Security=True");

 cmd = new SqlCommand();

 cmd.Connection = conn;

}

C#

**Step 9**

**Insert**

Double-click on the Insert button. Then, write a code under the function to create query variable and in-store the SQL insert command. Then we define SQL CommandText property equal to query, open the connection, and call to ExecuteNonQuery function, then close the connection. Here we make another function Clear data for after submitting details to clear the text boxes.

private void btninsert\_Click(object sender, EventArgs e) {

 string query = $ "insert into teacher values('{txtid.Text.ToString()}','{txtname.Text}','{txtaddress.Text}','{txtsalary.Text.ToString()}')";

 cmd.CommandText = query;

 conn.Open();

 cmd.ExecuteNonQuery();

 cleardata();

 conn.Close();

 displaydata();

}

private void cleardata() {

 txtid.Clear();

 txtname.Clear();

 txtaddress.Clear();

 txtsalary.Clear();

}

C#

**Step 10**

**Update**

Double-click on the Update button. Then we write a code under the function. To open the connection and define the SQL command type, we use command type text and then pass the SQL update command into command text. Then we call to ExecuteNonQuery function.

private void btnupdate\_Click(object sender, EventArgs e) {

 conn.Open();

 SqlCommand cmd = conn.CreateCommand();

 cmd.CommandType = CommandType.Text;

 cmd.CommandText = "update teacher set name='" + txtname.Text + "',Address='" + txtaddress.Text + "',salary='" + txtsalary.Text.ToString() + "' where id='" + txtid.Text.ToString() + "' ";

 cmd.ExecuteNonQuery();

 conn.Close();

 displaydata();

 cleardata();

}

C#

**Step 11**

**Read**

Double-click the Show All button. Here, write a display data function for users to display data into the data grid view. In the display function, open the connection, then create a command - it stores the SqlCommand variable - then define command type. Here we use command type text in a commandText. We pass the SQL Select \* from teacher command, and call the ExecuteNonQuery function. Then, create DataTable and SqlDataAdapter objects, and in a SqlDataAdapter object. We give the command object and then call a data table function. Fill and pass the data table object as a parameter then call the DataSouce function from the data grid view then close the connection.

private void btnshow\_Click(object sender, EventArgs e) {

 displaydata();

}

private void displaydata() {

 conn.Open();

 SqlCommand cmd = conn.CreateCommand();

 cmd.CommandType = CommandType.Text;

 cmd.CommandText = "select \* from teacher";

 cmd.ExecuteNonQuery();

 DataTable dt = new DataTable();

 SqlDataAdapter da = new SqlDataAdapter(cmd);

 da.Fill(dt);

 dataGridView1.DataSource = dt;

 conn.Close();

}

C#

**Step 12**

**Delete**

Double-click the Delete button. Then we write a code under the function to create a query variable, and that variable we write a SQL delete command. This makes the query equal to Command text. Then, open the connection and call the ExecuteNonQuery function. Then make dataGridView1.DataSource equal to query, and close the connection.

private void btndelete\_Click(object sender, EventArgs e) {

 string query = $ "delete teacher where id='{txtid.Text.ToString()}'";

 cmd.CommandText = query;

 conn.Open();

 cmd.ExecuteNonQuery();

 dataGridView1.DataSource = query;

 cleardata();

 conn.Close();

 displaydata();

}

C#

**Step 13**

**Find**

Double-click the Find button. Then write a code under the function to open the connection. Create a command. Then, write a command to type in command text. We pass the select command, then call the ExecuteNonQuery function. Then, create DataTable and SqlDataAdapter object and call the DataAdapter fill function. Define all text values for display and make sure that dataGridView1.DataSource is equal to the data table variable. Lastly, close the connection.

private void btnfind\_Click(object sender, EventArgs e) {

 conn.Open();

 SqlCommand cmd = conn.CreateCommand();

 cmd.CommandType = CommandType.Text;

 cmd.CommandText = "select \* from teacher where id='" + txtsearch.Text + "'";

 cmd.ExecuteNonQuery();

 DataTable dt = new DataTable();

 SqlDataAdapter da = new SqlDataAdapter(cmd);

 da.Fill(dt);

 txtname.Text = dt.ToString();

 txtaddress.Text = dt.ToString();

 txtsalary.Text = dt.ToString();

 dataGridView1.DataSource = dt;

 conn.Close();

}

C#

**Step 14**

Double click the exit button and write the method Application. Exit for the exit application.

private void btnsave\_exit\_Click(object sender, EventArgs e) {

 Application.Exit();

}

C#

**Output**



**Conclusion**

The Windows application is the easiest method for creating a Windows application with uses of the text box, button, label, image, checkbox, grid, and more.

It is simple and time-saving.