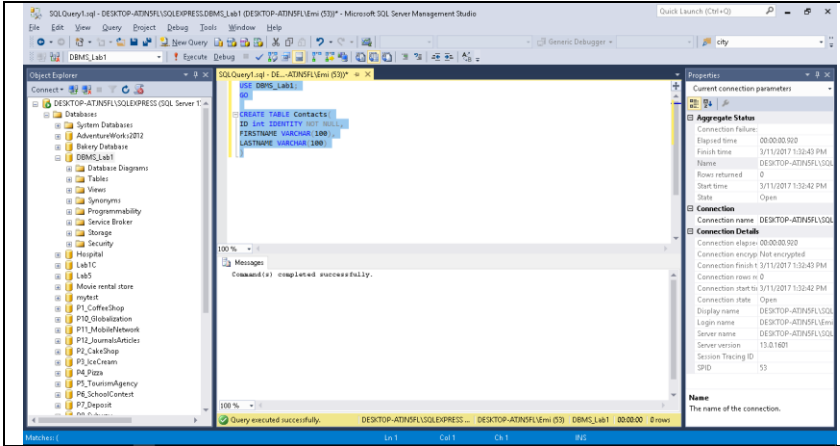


Application Lab1

SQL SERVER

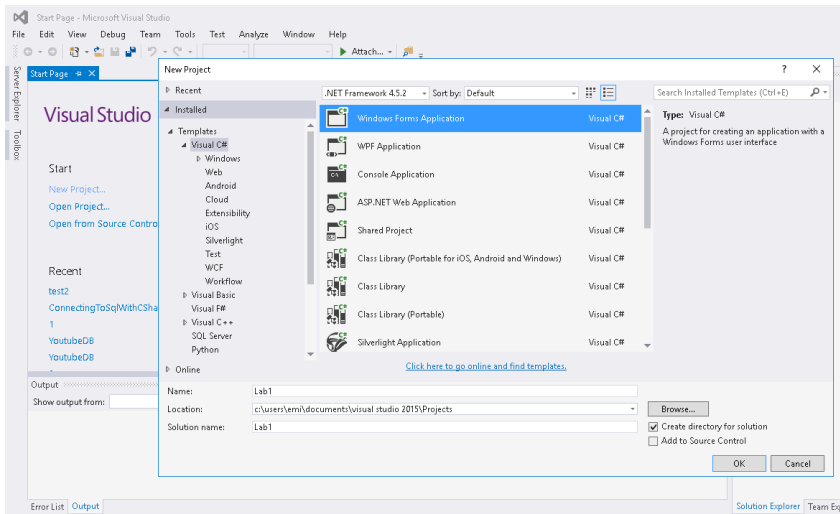


Use dbms_lab1;
Go

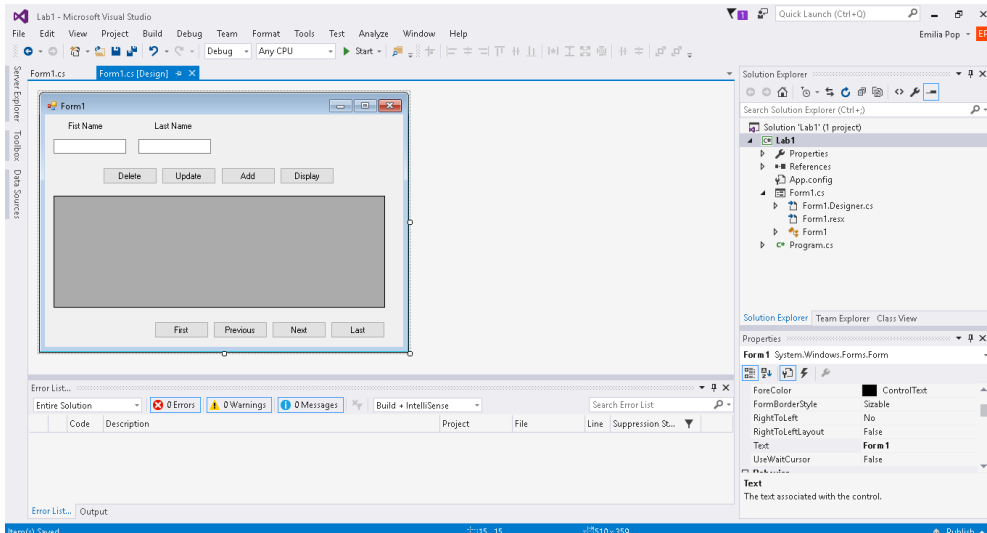
```
Create table Contacts(  
  Id int identity not null,  
  Firstname varchar(100),  
  Lastname varchar(100)  
)
```

Select * from Contacts

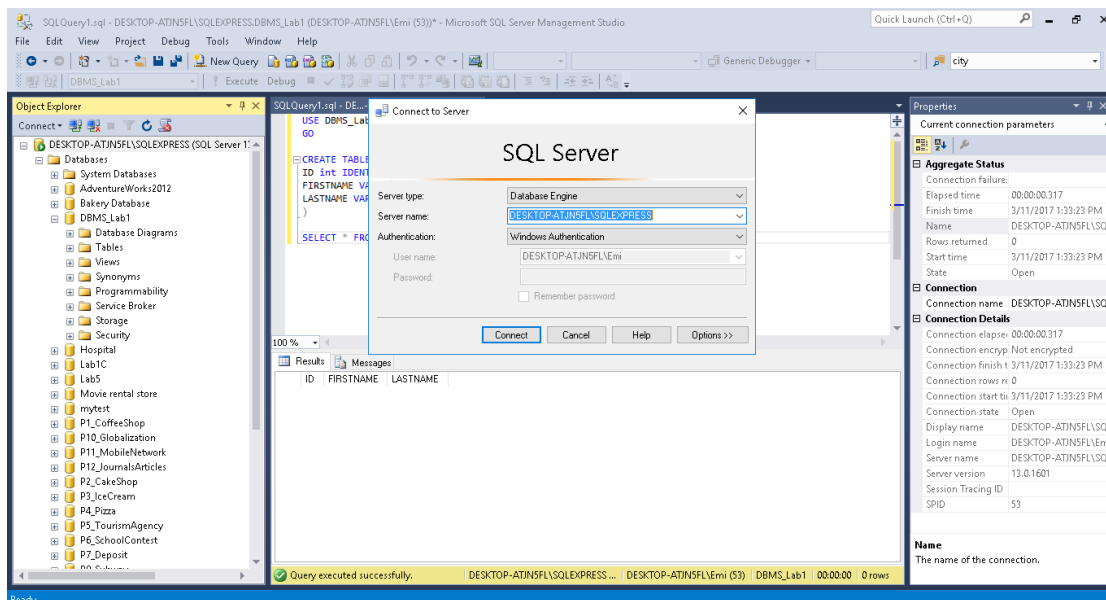
VISUAL STUDIO: WINDOWS FORMS APPLICATION / WINDOWS FORMS APP(.NET FRAMEWORK)



The form that will be implemented is:



Include in code the package: `using System.Data.SqlClient;`
Take the ServerName



1. Display all the Contacts – also appear automatically in TextBox (the first record is selected)

Form1

Fist Name Last Name

Dana Pop

Delete Update Add Display

ID	FIRSTNAME	LASTNAME
5	Dana	Pop
7	Andrei	Muresan
8	Dan	Pop
9	Cristian	Mihai
1010	Tudor	A
1011	Dana	Pop

Record 0 of 5 First Previous Next Last

```
SqlConnection cs = new SqlConnection("Data
Source=DESKTOP-
ATJN5FL\\SQLEXPRESS;Initial
Catalog=DBMS_Lab1;Integrated Security=True");
SqlDataAdapter da = new SqlDataAdapter();
DataSet ds = new DataSet();
BindingSource bs = new BindingSource();
```

```
private void btnDisplay_Click(object sender, EventArgs e)
{
    da.SelectCommand = new SqlCommand("SELECT * FROM Contacts", cs);
    ds.Clear();
    da.Fill(ds);

    dataGridView.DataSource = ds.Tables[0];
    bs.DataSource = ds.Tables[0];

    txtFirstName.DataBindings.Add("Text", bs, "FirstName");
    txtLastName.DataBindings.Add("Text", bs, "LastName");
    // Conection between textbox and the record from the Binding Source
    //last parameter is the name of the field of the table.
    // always will be inserted the first line in the TextBox-es
    // to move to next rows or others, can be use the methods MoveFirst(), MoveNext(), MovePrevious(),
    MoveLast()
    // there is no need to populate the Text Box-es for each method, because it works automatically
}
```

2. Add a new Contact

ID	FIRSTNAME	LASTNAME
1	Andrei	Muresan
2	Emilia	Muresan
3	Ana	Maria
4	Andrei	Muresan
5	Dana	Pop

```
private void btnAdd_Click(object sender,
EventArgs e)
{
    da.InsertCommand = new
SqlCommand("INSERT INTO Contacts VALUES (@f,
@l)", cs);
    da.InsertCommand.Parameters.Add("@f",
SqlDbType.VarChar).Value = txtFirstName.Text;
    da.InsertCommand.Parameters.Add("@l",
SqlDbType.VarChar).Value = txtLastName.Text;

    cs.Open();
    da.InsertCommand.ExecuteNonQuery();
    cs.Close();
}
```

3. Go to the next record (or previous, or last, or first).

ID	FIRSTNAME	LASTNAME
1	Andrei	Muresan
2	Emilia	Muresan
3	Ana	Maria

ID	FIRSTNAME	LASTNAME
1	Andrei	Muresan
2	Emilia	Muresan
3	Ana	Maria
4	Andrei	Muresan

Mark the selection inside the dataGridView and give also a message in a label (the position of each record when you navigate through it). So, **BindingSource** allows the navigation through the DataGridView and can take the positions.

```
private void dataGridViewUpdate()
{
    dataGridView.ClearSelection();
    dataGridView.Rows[bs.Position].Selected = true;
    records();
}
```

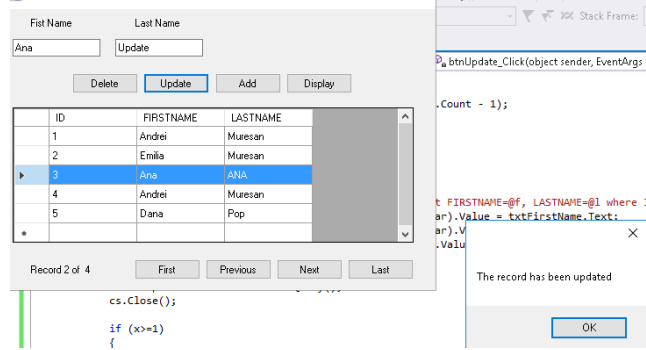
```
private void records()
{
    label3.Text = "Record " + bs.Position + " of "
+ (bs.Count - 1);
}
```

```
private void btnFirst_Click(object sender, EventArgs e)
{
    bs.MoveFirst();
    dataGridViewUpdate();
    records();
}
```

```
private void btnPrevious_Click(object sender,
EventArgs e)
{
    bs.MovePrevious();
    dataGridViewUpdate();
    records();
}
```

<pre>private void btnNext_Click(object sender, EventArgs e) { bs.MoveNext(); dataGridViewUpdate(); records(); }</pre>	<pre>private void btnLast_Click(object sender, EventArgs e) { bs.MoveLast(); dataGridViewUpdate(); records(); }</pre>

4. Update – by taking the ID from the DataGridView

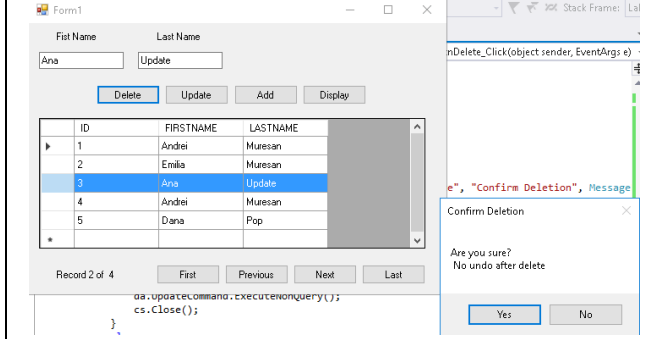


The screenshot shows a form with input fields for 'First Name' (Ana) and 'Last Name' (Update). Below are buttons for 'Delete', 'Update', 'Add', and 'Display'. A DataGridView contains 5 records. The 'Update' button is highlighted. A message box is displayed with the text 'The record has been updated'.

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    int x;
    da.UpdateCommand = new SqlCommand("Update
Contacts set FIRSTNAME=@f, LASTNAME=@l where
ID=@id", cs);
    da.UpdateCommand.Parameters.Add("@f",
SqlDbType.VarChar).Value = txtFirstName.Text;
    da.UpdateCommand.Parameters.Add("@l",
SqlDbType.VarChar).Value = txtLastName.Text;
    da.UpdateCommand.Parameters.Add("@id",
SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];

    cs.Open();
    x= da.UpdateCommand.ExecuteNonQuery();
    cs.Close();
    if (x>=1)
    {
        MessageBox.Show("The record has been updated");
    }
}
```

5. Delete



The screenshot shows the same form as before, but the 'Delete' button is highlighted. A confirmation dialog box is shown with the text 'Are you sure? No undo after delete' and 'Confirm Deletion'.

```
private void btnDelete_Click(object sender, EventArgs e)
{
    DialogResult dr;
    dr = MessageBox.Show("Are you sure?\n No undo after
delete", "Confirm Deletion", MessageBoxButtons.YesNo);
    if (dr == DialogResult.Yes)
    {
        da.DeleteCommand = new SqlCommand("Delete
from Contacts where ID=@id", cs);
        da.DeleteCommand.Parameters.Add("@id",
SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];

        cs.Open();
        da.DeleteCommand.ExecuteNonQuery();
        cs.Close();

        ds.Clear();
    }
}
```

Form1

Fist Name

Last Name

Andrei

Muresan

Delete

Update

Add

Display

	ID	FIRSTNAME	LASTNAME
▶	1	Andrei	Muresan
	2	Emilia	Muresan
	4	Andrei	Muresan
	5	Dana	Pop
*			

Record 0 of 3

First

Previous

Next

Last

```
da.Fill(ds);
}
else
{
    MessageBox.Show("Deletion Aborded");
}
}
```