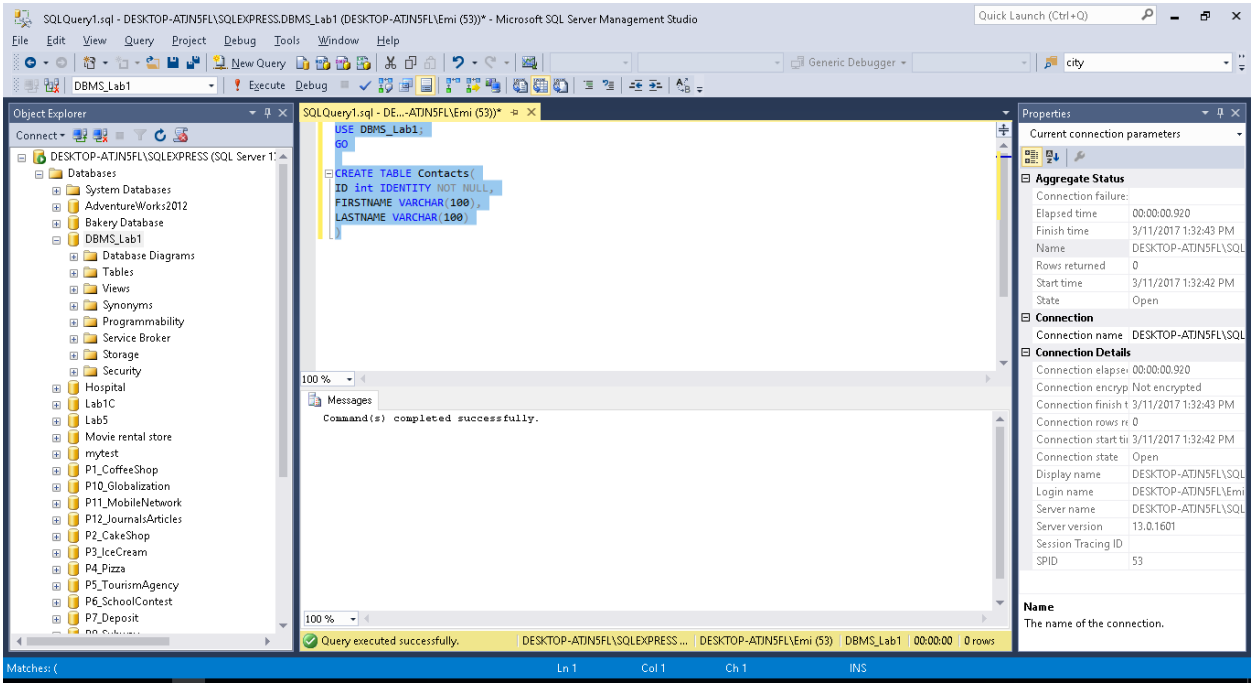
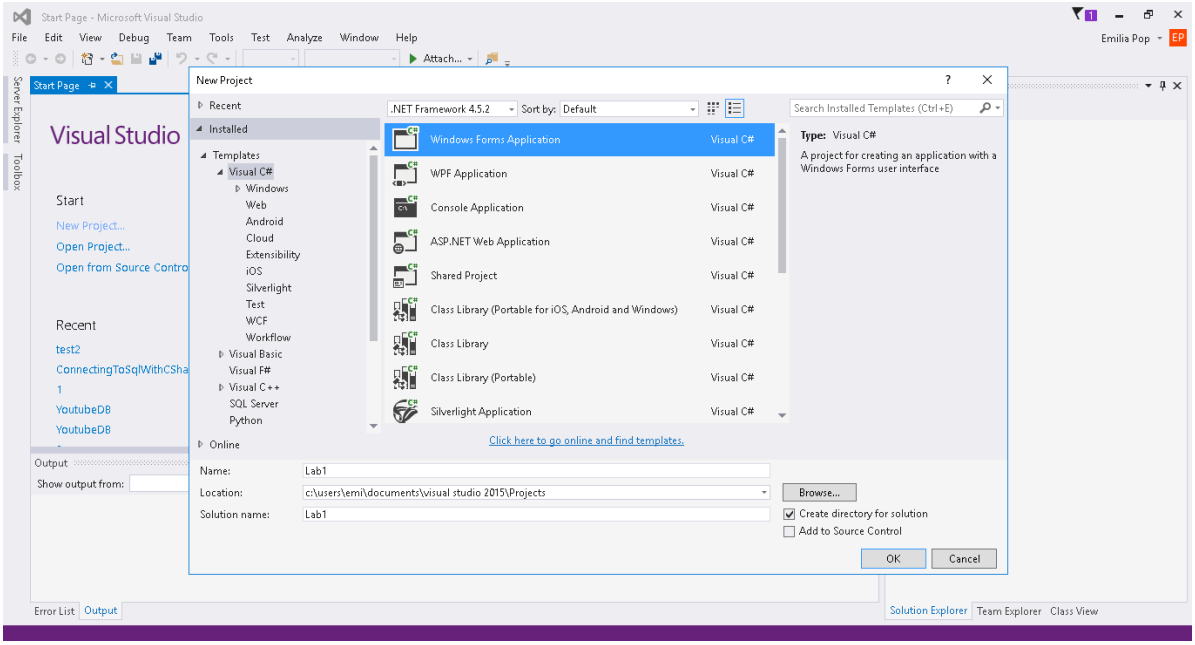
Application Lab1

SQL SERVER

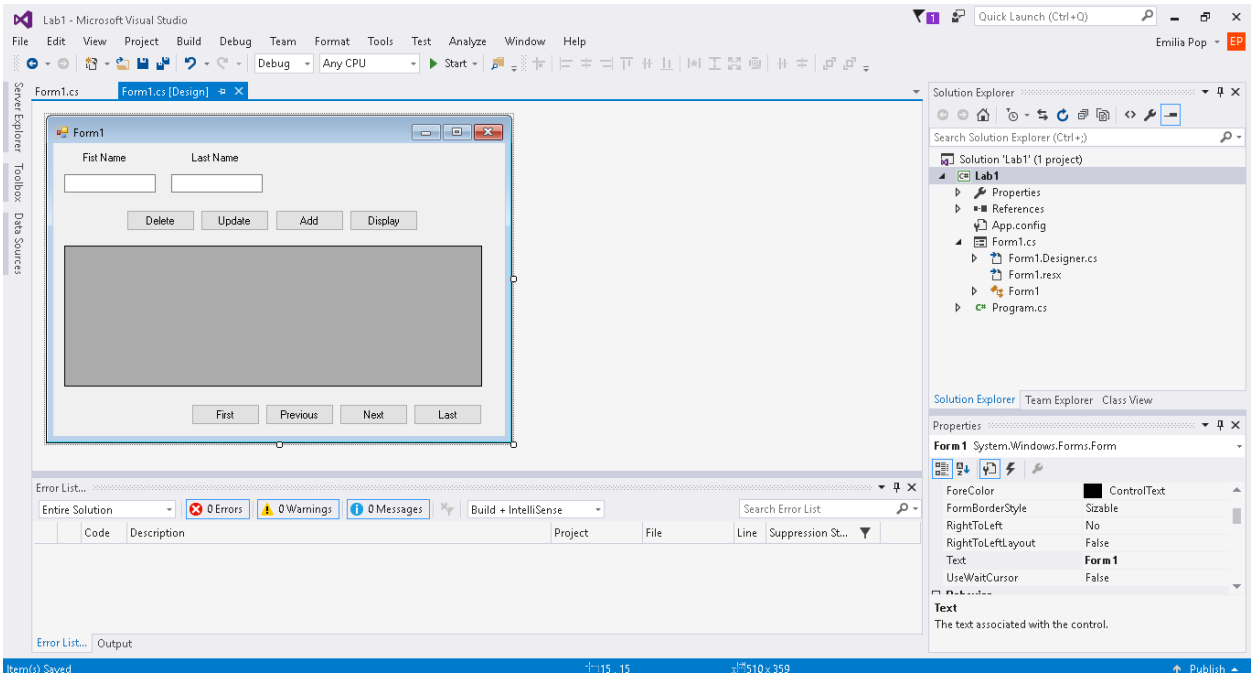


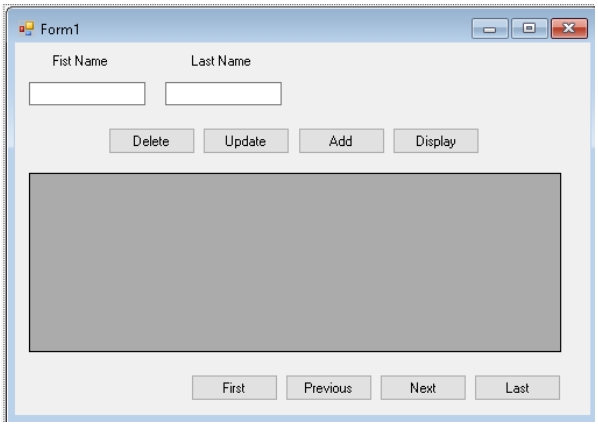

```
Use dbms_lab1;
Go

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

Select * from Contacts
```

VISUAL STUDIO

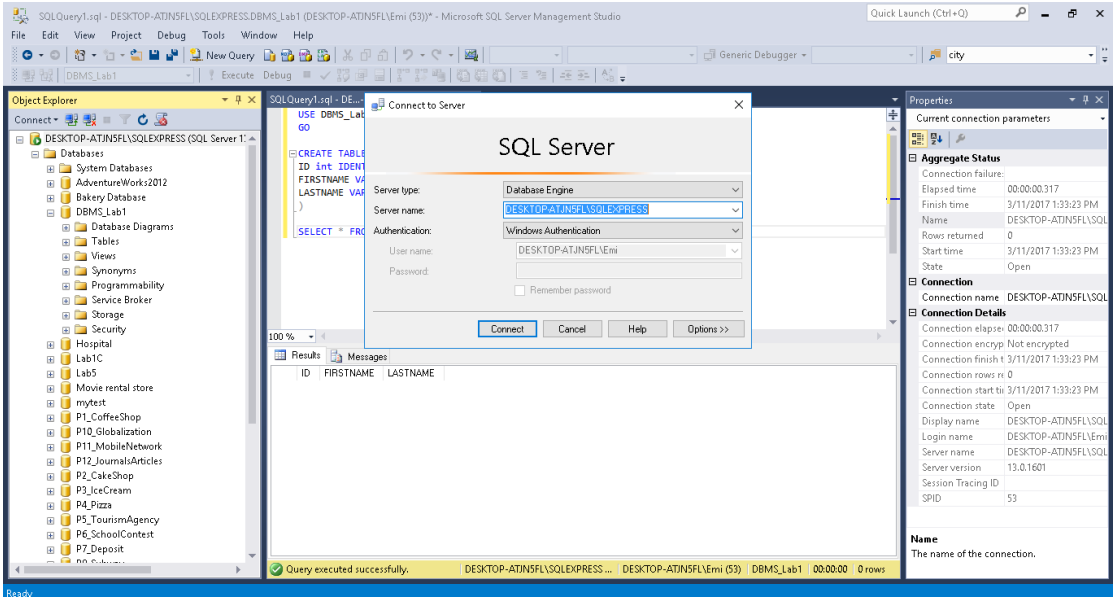


The form that will be implemented is:

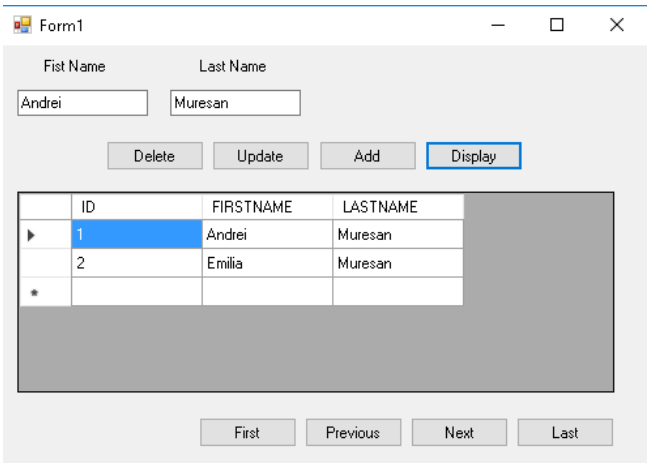


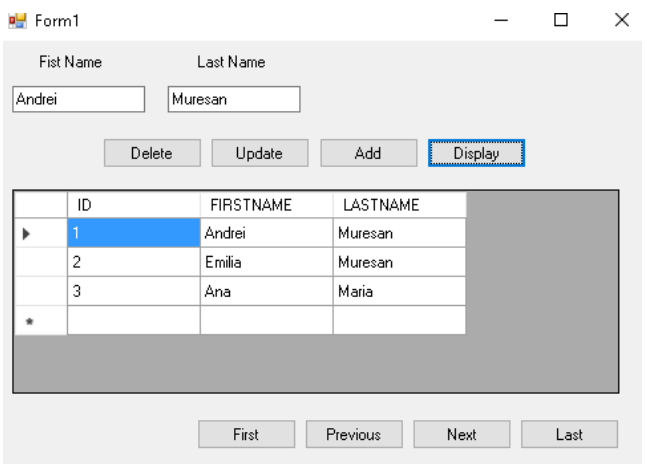


Include in code the package: using System.Data.SqlClient;
Take the ServerName



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

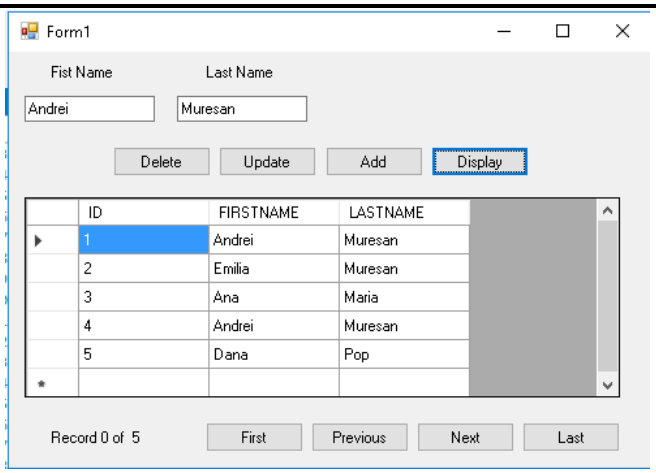





```
SqlConnection cs = new SqlConnection("Data Source=DESKTOP-ATJN5FL\\SQLEXPRESS;Initial Catalog=DBMS_Lab1;Integrated Security=True");
SqlDataAdapter da = new SqlDataAdapter();
DataSet ds = new DataSet();
BindingSource bs = new BindingSource();
```

```
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 texbox and the record from the Binding Source
        //last parameter is the name of the field of the table.
 }
```

2. Add a new Contact

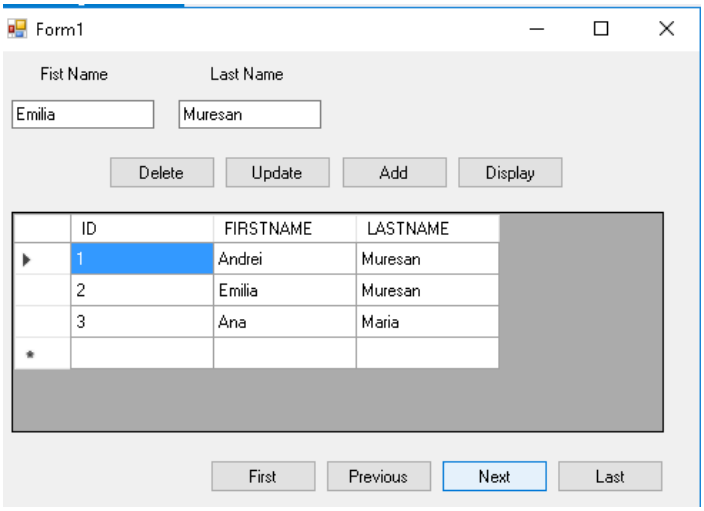


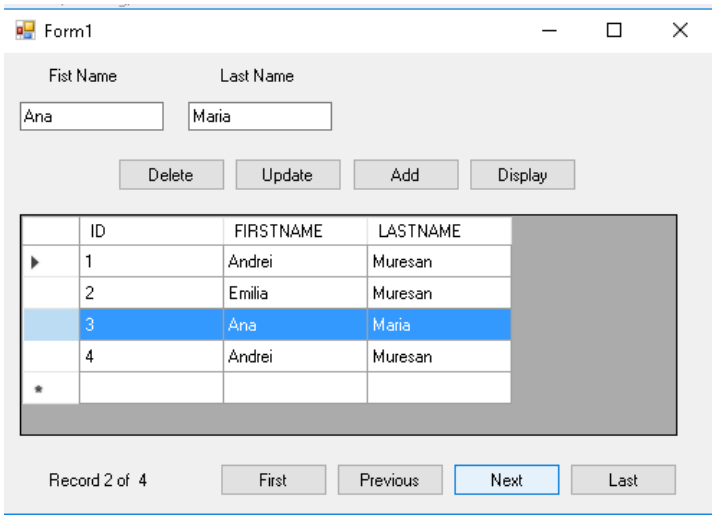
```
    private void btnAdd_Click(object sender, EventArgs e)
    {
        da.InsertCommand = new SqlCommand("INSERT INTO Contacts VALUES (@f, @l)", cs);
```

```
    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).



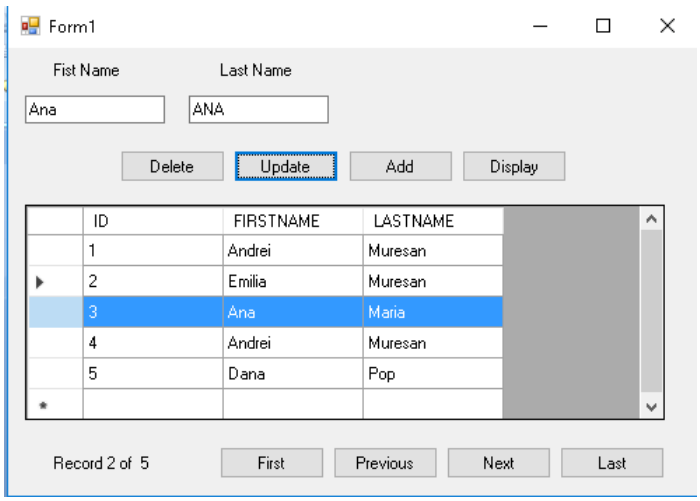


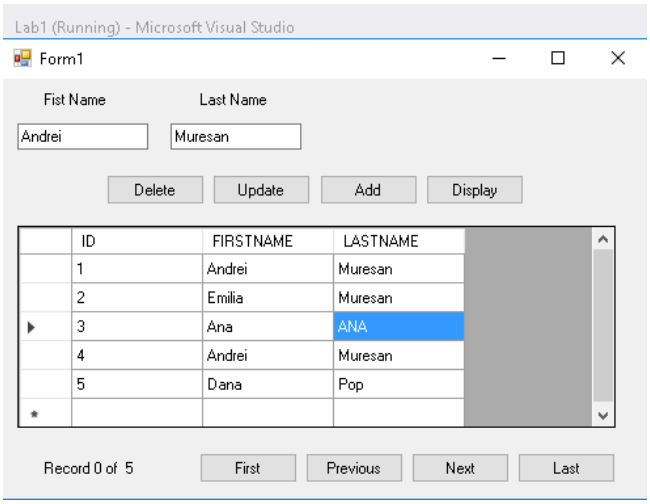
Mark the selection inside the dataGridView and give also a message in a label – take the position of each record when you navigate.

BindingSource permit the navigation through the DataGridView and 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();
```

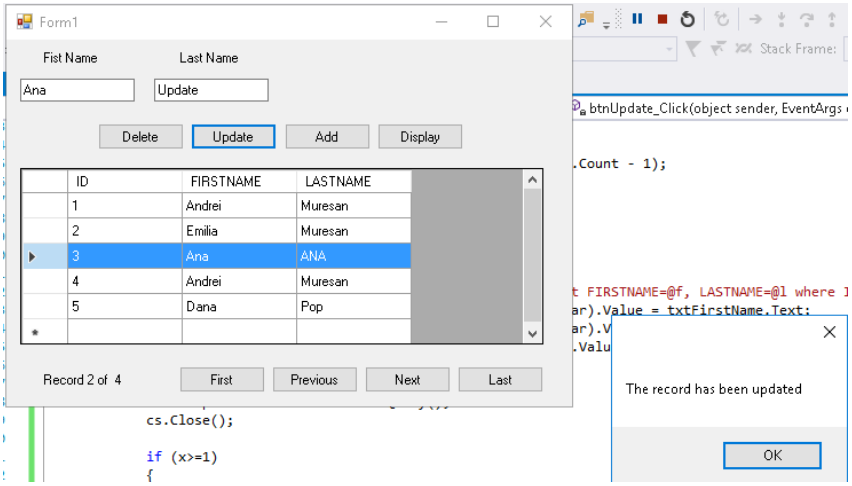
```
    records();
}

private void btnNext_Click(object sender, EventArgs e)
{
    bs.MoveNext();
    dataGridViewUpdate();
    records();
}

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





---

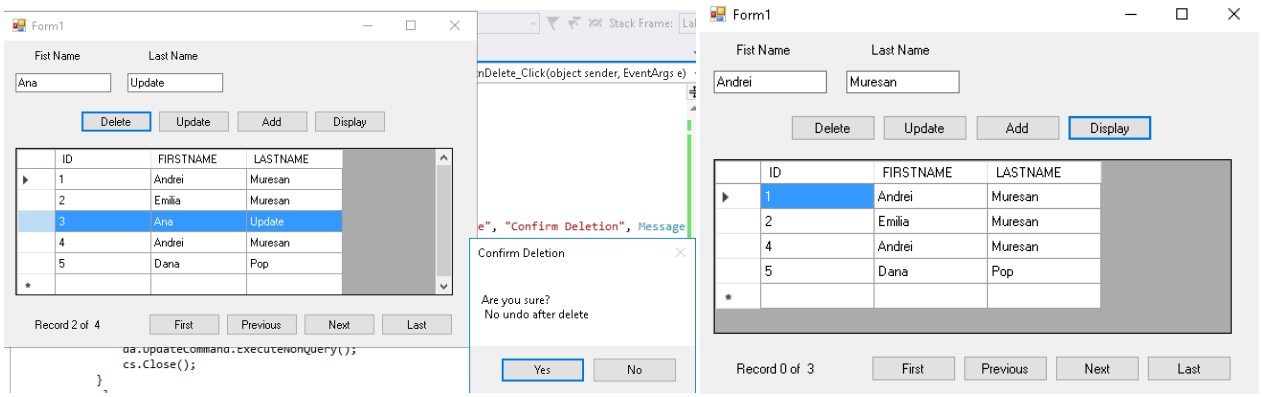
4. Update – by taking the ID from the DataGridView




```
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




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