

Laboratory 1

4 labs to present

L1: receive: Lab 1 (S1/S2)

Present: Lab 3 (S5/S6)

L2: receive: Lab 3 (S5/S6)

Present: Lab 4 (S7/S8)

L3: receive: Lab 4 (S7/S8)

Present: Lab 5 (S9/S10)

L4: receive: Lab 5 (S9/S10)

Present: Lab 6 (S11/S12)

Practical Exam: Lab 7 (S13/S14)

Connection between SQL SERVER – VISUAL STUDIO (C#) – through a WINDOWS FORMS APPLICATION / WINDOWS FORMS APP (.NET FRAMEWORK)

We consider the following database

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'Lab1C'. The 'Client' table has columns: Cid (PK, int, not null), Name (varchar(50), null), Gender (varchar(50), null), and Dob (date, null). The 'Product' table has columns: Pid (PK, int, not null), PName (varchar(50), null), Quantity (int, null), Price (float, null), and Cid (FK, int, not null). A relationship line connects the 'Cid' column of the 'Product' table to the 'Cid' column of the 'Client' table. On the right, the 'Query Results' window shows the output of a query: 'select * from Client' and 'select * from Product'. The results are displayed in two tables. The first table, 'Client', has columns: Cid, Name, Gender, and Dob. The second table, 'Product', has columns: Pid, PName, Quantity, Price, and Cid. A status bar at the bottom indicates 'Query executed successfully.'

Cid	Name	Gender	Dob
1	Name	Gender	1900-01-01
2	Client 2	f	2000-02-03
3	Client 3	f	1980-05-16
4	Client 4	m	1896-11-08
5	Client 5	m	1999-06-14
6	alin	m	1989-12-11
7	Maria55	female	2010-12-12
8	1019	6	2010-12-12

Pid	PName	Quantity	Price	Cid	
1	4	lemon	5	5	2
2	6	chocolate	5	3.5	4
3	7	ice cream	2	8	3
4	15	Miere	2	23	2
5	16	Cartofi	420	10	3
6	17	Flori	3	10	3
7	18	tiramisu	1	12	3

Client- parent table

Product – child table

1. Display all parent records
`SELECT * FROM Client`
2. Selecting a parent record, it must display its child records
`SELECT * FROM Product WHERE Cid=...` (the one that is selected from the list of the clients)
3. Selecting a child record, it must allow to remove or update the data of the child
`DELETE FROM Product WHERE Pid=...` (the one that is selected)

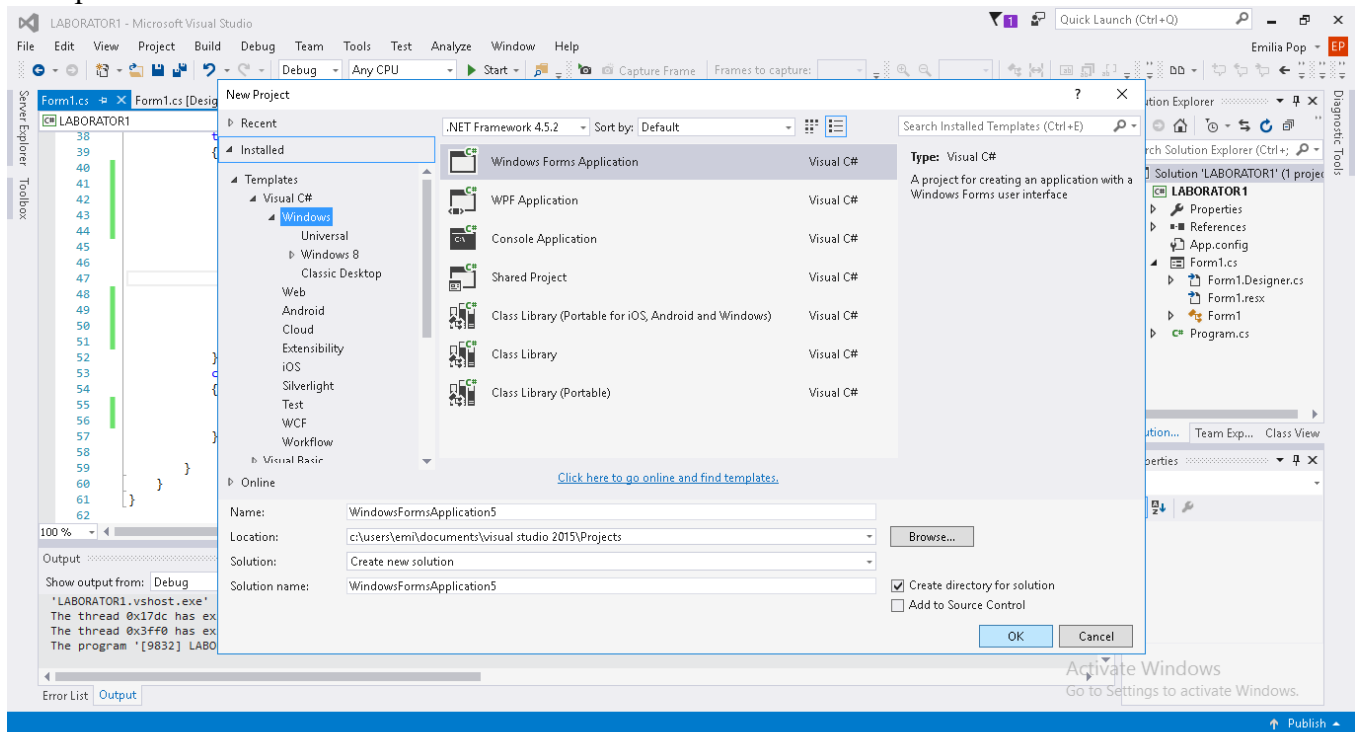
UPDATE Product SET PName=@p, Quantity=@q, Price=@pr, Cid=...(preferable a list of of clients from where to choose) WHERE Pid = ... (the one that is selected)

4. Selecting a parent record, it must allow to add a new child record
 INSERT INTO Product(PName, Quantity, Price, Cid) VALUES (@pname, @q, @p, ...)

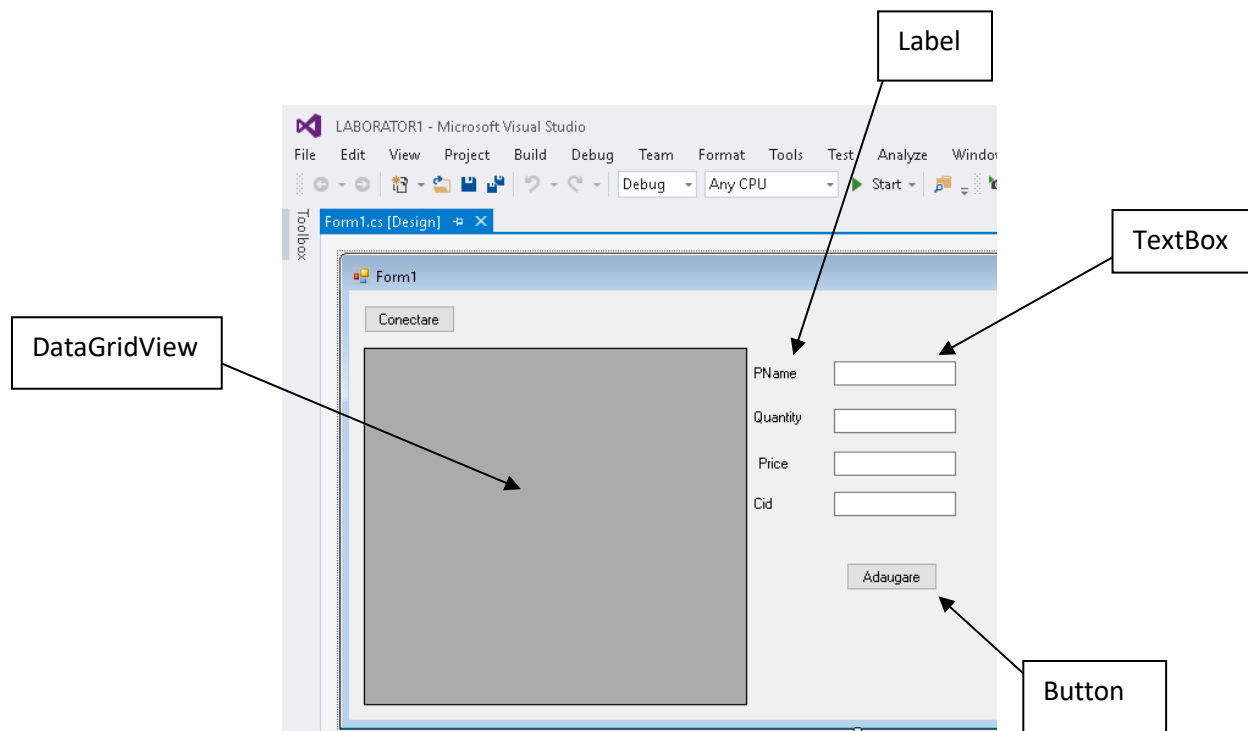
The values that have to be introduced can be taken from TextBox-es.

SQL SERVER ----- VISUAL STUDIO (C#)
 (tables, ...) (DataAdapter) (DataSet)

Example:



SELECT - use the method Fill()
 INSERT, UPDATE, DELETE – use the method ExecuteNonQuery()



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

```

```

namespace LABORATOR1
{
    public partial class Form1 : Form
    {

```

```

        SqlConnection cs = new SqlConnection("Data Source=DESKTOP-
        ATJN5FL\\SQLEXPRESS;Initial Catalog=Lab1C;Integrated Security=True");
        SqlDataAdapter da = new SqlDataAdapter();
        DataSet ds = new DataSet();

```

```

        public Form1()
        {
            InitializeComponent();
        }

```

```

        private void button1_Click(object sender, EventArgs e)
        {
            da.SelectCommand = new SqlCommand("SELECT * FROM Product", cs);
            ds.Clear();
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }

```

```

    }

    private void button2_Click(object sender, EventArgs e)
    {
        try
        {
            da.InsertCommand = new SqlCommand("INSERT INTO Product (PName, Quantity, Price,
Cid) VALUES(@p,@q, @pr, @c)", cs);
            da.InsertCommand.Parameters.Add("@p", SqlDbType.VarChar).Value = textBox1.Text;
            da.InsertCommand.Parameters.Add("@q", SqlDbType.Int).Value =
Int32.Parse(textBox2.Text);
            da.InsertCommand.Parameters.Add("@pr", SqlDbType.Int).Value =
Int32.Parse(textBox3.Text);
            da.InsertCommand.Parameters.Add("c", SqlDbType.Int).Value = Int32.Parse(textBox4.Text);
            cs.Open();
            da.InsertCommand.ExecuteNonQuery();
            MessageBox.Show("Inserted Succesfull to the Database");
            cs.Close();
            // already inserted - appear in the list
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
            cs.Close();
        }
    }
}
}
}

```

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a button labeled "Conectare" at the top left. Below it is a data grid with the following data:

Pid	PName	Quantity
2	cheese	3
3	orange	4
4	lemon	5
5	grapes red	12
6	chocolate	5
7	ice cream	2
8	flower	5
9	orange	1

To the right of the grid, there are four input fields labeled "PName", "Quantity", "Price", and "Cid". Below these fields is a button labeled "Adaugare".

Form1

Conectare

Pid	PName	Quantity
5	grapes red	12
6	chocolate	5
7	ice cream	2
8	flower	5
9	orange	1
11	a	1
12	t	2
*		

PName

tiramisu

Quantity

4

Price

12

Cid

3

Adaugare

Price, Cid)

1.Text;

(textBox2.Te

e(textBox3.T

textBox4.Text

Inserted Succesfull to the Database

OK

Form1

Conectare

Pid	PName	Quantity
6	chocolate	5
7	ice cream	2
8	flower	5
9	orange	1
11	a	1
12	t	2
13	tiramisu	4
*		

PName

tiramisu

Quantity

4

Price

12

Cid

3

Adaugare