

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курский государственный университет»

Кафедра программного обеспечения и администрирования  
информационных систем

Отчёт  
по лабораторной работе № 6

” Управляемый провайдер SQL. Работа с базами данных с использованием  
управляемого провайдера SQL ”

по дисциплине  
“Программирование на C#”

Выполнил:

студент группы 413

Мусонда Салиму

Проверил:

ст.пр. кафедры ПОиАИС

Ураева Е.Е.

Курск, 2020

**Цель:** с использованием управляемого провайдера SQL реализовать механизмы установления соединения с базой данных MS SQL, построения команд SQL, заполнения данными объекта DataSet.

### **Задание**

Реализовать в программе установление соединения с исходной базой данных MS SQL, реализовать и выполнить не менее трёх команд SQL (удаление, обновление и вставка) по работе с исходной базой данных, реализовать заполнение данными исходной базы данных объекта DataSet.

**Примечание:** для выполнения задания необходимо создать базу данных, включающую как минимум 3-4 таблицы и 2 отношения, представленные в базе данных Microsoft Access, с использованием которой выполнялись лабораторной работы 3, 4.

### **Текст программы**

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

namespace lab4_csharp
{
    public partial class Form1 : Form
    {
        DataTable t1,t2,t3;
        DataSet s1;
        SqlDataAdapter da,da1,da2;
        SqlCommand com;
        DataRow r1;
        bool dg2 = false;
        DataRelation depwork,stuuni;
        public SqlConnection con;
        String cmd = "select * from gorada";
        String cmd1 = "select * from Students";
        String cmd2 = "select * from Unis";
        SqlCommandBuilder bc;

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

        string conner= @"Server=DESKTOP-KCDFND4\SQLEXPRESS; initial catalog = cities;
integrated security = true";
        con = new SqlConnection(conner);
        s1 = new DataSet();
        t3 = new DataTable();

    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {

        //textBox5.Text = comboBox1.SelectedValue.ToString();
    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void dataGridView1_RowStateChanged(object sender,
DataGridViewRowStateChangedEventArgs e)
    {

    }

    private void dataGridView1_SelectionChanged(object sender, EventArgs e)
    {
        //s1.Tables[1].DefaultView.RowFilter = "CityID = " +
dataGridView1.CurrentRow.Cells["CityID"].Value;
        //dataGridView2.DataSource = s1.Tables[1];
    }

    private void button2_Click(object sender, EventArgs e)
    {
        s1.WriteXml(@"C:\Users\CHATW\Documents\xml\tables.xml");
        s1.Clear();
        s1.Dispose();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        s1.ReadXml(@"C:\Users\CHATW\Documents\xml\tables.xml");
        dataGridView1.DataSource = s1.Tables[0];
        dataGridView2.DataSource = s1.Tables[1];
    }

```

```

        dataGridView3.DataSource = s1.Tables[2];

    }

    private void dataGridView3_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void dataGridView3_SelectionChanged(object sender, EventArgs e)
    {

    }

    private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void button7_Click(object sender, EventArgs e)
    {
        string comm = "select max(StudID) + 1 from Students";
        com = new SqlCommand(comm, con);
        con.Open();
        string id = com.ExecuteScalar().ToString();
        con.Close();
        textBox2.Text = id;

    }

    private void button8_Click(object sender, EventArgs e)
    {
        string id = textBox2.Text;
        string fname = textBox3.Text;
        string lname = textBox4.Text;
        string cid = textBox5.Text;
        string uid = textBox6.Text;

        string comm1 = string.Format("set identity_insert Students on insert into
Students (StudID,FName,LName,CityID,UniID) values ({0}','{1}','{2}',{3},{4})", id, fname,
lname, cid, uid);
        com = new SqlCommand(comm1, con);
        con.Open();
        com.ExecuteNonQuery();
        con .Close();
        DataRow n1= s1.Tables[1].NewRow();
        n1["StudID"] = id;
        n1["FName"] = fname;
        n1["LName"] = lname;
        n1["CityID"] = cid;
        n1["UniID"] = uid;
        s1.Tables[1].Rows.Add(n1);

        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();
        textBox6.Clear();

    }

```

```

private void button6_Click(object sender, EventArgs e)
{
    string comm = string.Format("update Students set FName = '{0}', LName = '{1}'
, CityID = {2}, UniID = {3} where StudID = {4}", textBox3.Text, textBox4.Text,
textBox5.Text, textBox6.Text, textBox2.Text);
    com = new SqlCommand(comm, con);
    con.Open();
    com.ExecuteNonQuery();
    con.Close();

    DataRow[] cr1 = s1.Tables[1].Select("StudID = " + textBox1.Text);
    cr1[0][0] = textBox2.Text;
    cr1[0][1] = textBox3.Text;
    cr1[0][2] = textBox4.Text;
    cr1[0][3] = textBox5.Text;
    cr1[0][4] = textBox6.Text;

}

private void button9_Click(object sender, EventArgs e)
{
    try
    {
        DataRow[] cr1 = s1.Tables[1].Select("StudID = " + textBox1.Text);
        s1.Tables[1].Rows.Remove(cr1[0]);
        s1.Tables[1].AcceptChanges();
        com = new SqlCommand("delete from Students where StudID = " +
textBox1.Text, con);
        con.Open();
        com.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("удален");

    }catch(Exception t)
    {
        MessageBox.Show(t.ToString());
    }
}

private void button5_Click(object sender, EventArgs e)
{
    DataRow[] cr1 = s1.Tables[1].Select("StudID = " + textBox1.Text);
    textBox2.Text = textBox1.Text;
    textBox3.Text = cr1[0][1].ToString();
    textBox4.Text = cr1[0][2].ToString();
    textBox5.Text = cr1[0][3].ToString();
    textBox6.Text = cr1[0][4].ToString();

}

private void dataGridView2_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{

```

```

    }

    private void button4_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        DataRow[] cr1 = s1.Tables[1].Select("StudID = " + textBox1.Text);
        if (cr1.Length < 1) { MessageBox.Show("HET TAKOGO CTYДЕНТА"); return; }

        else
        {
            foreach (DataRow cr in cr1)
            {
                listBox1.Items.Add("Names = " + cr["FName"].ToString() + "\t" +
                cr["LName"].ToString());
                listBox1.Items.Add("Student ID = " + cr["StudID"].ToString());
                listBox1.Items.Add("\n");
                foreach (DataRow cr2 in ((DataRow)cr).GetParentRows(depwork))
                {
                    listBox1.Items.Add(cr2["CityName"].ToString());

                    foreach (DataRow cr3 in ((DataRow)cr).GetParentRows(stuuni))
                    {
                        listBox1.Items.Add(cr3["UName"].ToString() + "\t" +
                        cr3["UAdd"].ToString());
                    }
                    listBox1.Items.Add("\n");
                }
            }
        }

    }

    private void dataGridView2_CellValueChanged(object sender,
    DataGridViewCellEventArgs e)
    {
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            da = new SqlDataAdapter(cmd, con);
            da.Fill(s1, "GORADA");
            da1 = new SqlDataAdapter(cmd1, con);
            da1.Fill(s1, "STUDENTS");
            da2 = new SqlDataAdapter(cmd2, con);
            da2.Fill(s1, "UNIVERSITIES");
            bc = new SqlCommandBuilder(da);
            DataColumn pk = s1.Tables[0].Columns[0];
            DataColumn fk = s1.Tables[1].Columns[3];
            DataColumn pk1 = s1.Tables[2].Columns[0];
            DataColumn fk1 = s1.Tables[1].Columns[4];
            bc = new SqlCommandBuilder(da);

            depwork = new DataRelation("depwork", pk, fk);
            stuuni = new DataRelation("stuuni", pk1, fk1);
            s1.Relations.Add(depwork);
            s1.Relations.Add(stuuni);
        }
    }

```

```

dataGridView1.DataSource = s1.Tables[0];
dataGridView2.DataSource = s1.Tables[1];
dataGridView3.DataSource = s1.Tables[2];

```

```

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

```

```

}
}
}

```

## Тестирование

Тестирование заполнения таблиц представлено на рисунке 1.

The screenshot shows a Windows application window titled "Form1" with a menu bar containing "Manage" and "netcoreapp3.1". The main area contains three data grids and a control panel.

**Grid 1: Cities**

CityID	CityName
1	MOSCOW
2	KURSK
3	BELGOROD
4	TULA
5	OREL
6	PETROZAVODSK

**Grid 2: Students**

StudID	FName	LName	CityID	UniID
100	ALINANI	SILWENGA	11	4
101	MUSONDA	SALIMU	2	1
102	ELIAH	MUSETKA	2	1
103	PATRICK	KUMWFNDA	1	4

**Grid 3: Universities**

UniID	UName	UAdd
1	Kursk State	radishova 26
2	rudn	yugo zapadnaya
3	belgorod state	pabyedi 85
4	kazan state	pabyedi 20
5	Tula state	lenina 35

**Control Panel:**

- Buttons: FILL, SAVE, RETRIEVE, UPDATE, ADD, DELETE, DONE, SAVE.
- Input fields: STUDENT ID, FIRST NAME, LAST NAME, CITY ID, UNI ID.
- Search section: ID, SEARCH button.

Рисунок 1 - Тестирование заполнения таблиц

Тестирование запроса на вставку представлено на рисунке 2.

Form1

	CityID	CityName
▶	1	MOSCOW
	2	KURSK
	3	BELGOROD
	4	TULA
	5	OREL
	6	PETROZAVODSK

	StudID	FName	LName	CityID	UnilD
	115	BRAVO	BARIDO	1	2
	116	JOYCE	MEYER	2	1
*					

	UnilD	UName	UAdd
▶	1	Kursk State	radishova 26
	2	rudn	yugo zapadnaya
	3	belgorod state	pabyedi 85
	4	kazan state	pabyedi 20
	5	Tula state	lenina 35

STUDENT ID:   
FIRST NAME:   
LAST NAME:   
CITY ID:   
UNI ID:

ID

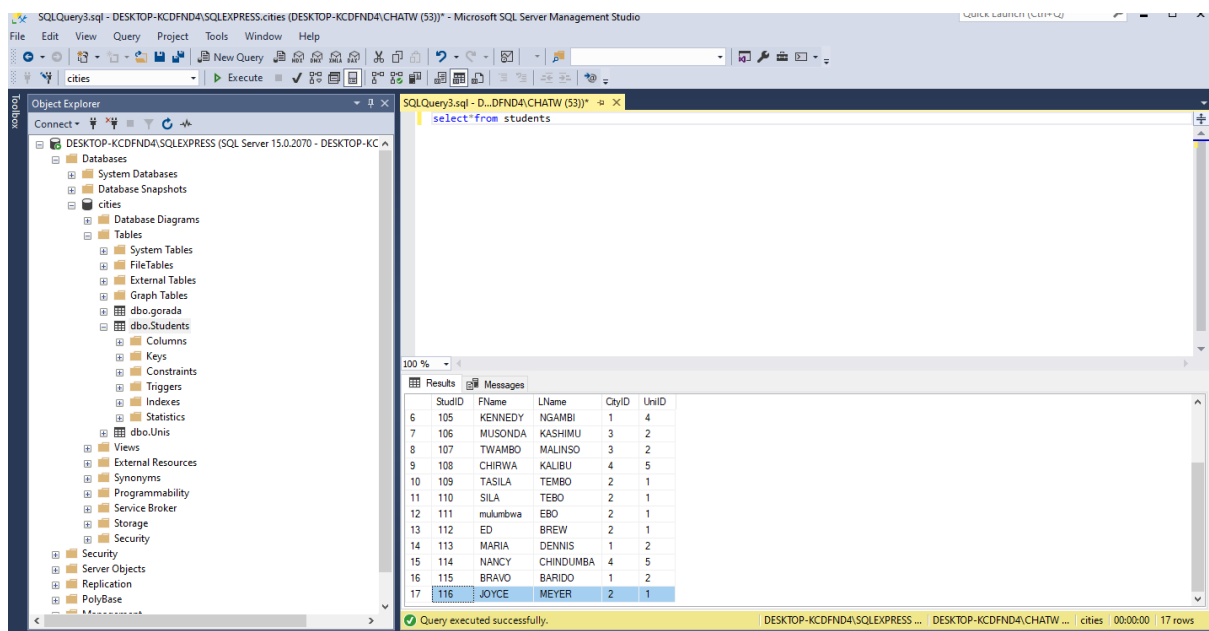


Рисунок 2 – Тестирование запроса на вставку

Тестирование запроса на удаление представлено на рисунке 3.



The application window 'Form1' displays the following data tables:

	CityID	CityName
1	MOSCOW	
2	KURSK	
3	BELGOROD	
4	TULA	
5	OREL	
6	PETROZAVODSK	

StudID	FName	LName	CityID	UnilD
100	ALINANI	SILWENGA	11	4
101	MUSONDA	SALIMU	2	1
102	ELIAH	MUSETKA	2	1
103	PATRICK	KIJMWFNDA	1	4

UnilD	UName	UAdd
1	Kursk State	radishova 26
2	rudn	yugo zapadnaya
3	belgorod state	pabyedi 85
4	kazan state	pabyedi 20
5	Tula state	lenina 35

Buttons: FILL, SAVE, RETRIEVE, UPDATE, ADD, DELETE, DONE, SAVE.

Search section: STUDENT ID (116), FIRST NAME (JOYCE), LAST NAME (MEYER), CITY ID (2), UNI ID (1). ID: 116, SEARCH.

Modal dialog: удален, OK.

Рисунок 3 – Тестирование запроса на удаление

Тестирование запроса на обновление представлено на рисунке 4.

The application window 'Form1' displays the following data tables:

	CityID	CityName
1	MOSCOW	
2	KURSK	
3	BELGOROD	
4	TULA	
5	OREL	
6	PETROZAVODSK	

StudID	FName	LName	CityID	UnilD
100	ALINANI	SILWENGA	2	1
101	MUSONDA	SALIMU	2	1
102	ELIAH	MUSETKA	2	1
103	PATRICK	KIJMWFNDA	1	4

UnilD	UName	UAdd
1	Kursk State	radishova 26
2	rudn	yugo zapadnaya
3	belgorod state	pabyedi 85
4	kazan state	pabyedi 20
5	Tula state	lenina 35

Buttons: FILL, SAVE, RETRIEVE, UPDATE, ADD, DELETE, DONE, SAVE.

Search section: STUDENT ID (100), FIRST NAME (ALINANI), LAST NAME (SILWENGA), CITY ID (2), UNI ID (1). ID: 100, SEARCH.

Рисунок 4 – Тестирование запроса на обновление