МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

“Харківський авіаційний інститут”

Факультет СУЛА

Кафедра математичного моделювання та штучного інтелекту

**Лабораторна робота №3**

з Організація баз даних

на тему: «ADO.NET: connected layer Tran&Storage.»

Виконав: студент 3 курсу групи №335a

напряму підготовки(спеціальності):

122 Комп’ютерні науки

(шифр і назва напряму підготовки(спеціальності))

Гринюк М.О.

(прізвище й ініціали студента)

Прийняв: ст. викл., к.т.н. Коробчинський К.П.

(посада, науковий ступінь, прізвище й ініціали)

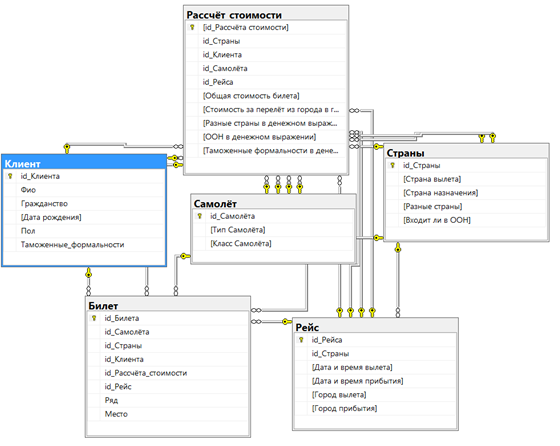
Харків – 2020

**Цель работы:** Изучить принципы построения пользовательского интерфейса, рассмотреть вопросы использования основных серверных элементов управления, применяемых для создания Web-приложений, рассмотреть вопросы организации ввода данных от пользователя.

**Ход работы:**

1. Пройти тест;
2. Добавить логическую модель предметной обрасти;
3. Создать добавить к своему проекту класс логирования;
4. Создать проект C#, создать **connected layer**и реализовать действия:
   1. Хранимые процедуры;
   2. Транзакции;
5. **Зайдите на сайт MSDN.**Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на занятии, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. *Сохраните ссылки и дайте им короткое описание;*
6. Для каждого пункта показать программный код и пример работы (скрин);
7. Оформить отчёт и сохранить на севере STM;
8. Уточнить сколько пар выполняется работа и когда необходимо её сдать;
9. Знать ответы на контрольные вопросы.

**Логическая модель БД**



**Выполнение работы**

1. Создать добавить к своему проекту класс логирования;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using NLog;

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

NLog.Logger logger = LogManager.GetCurrentClassLogger();

logger.Trace("trace message");

logger.Debug("debug message");

logger.Info("info message");

logger.Warn("warn message");

logger.Error("error message");

logger.Fatal("fatal message");

string conStr = @"Data Source=COMPUTER2\MSSQLSERVER02;

Initial Catalog=Agency;

Integrated Security=True";

using (SqlConnection conn = new SqlConnection(conStr))

{

try

{

conn.StateChange += connection\_StateChange;

conn.Open();

MessageBox.Show("Connection Open");

conn.Close();

MessageBox.Show("Connection closed");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

static void connection\_StateChange(object sender, System.Data.StateChangeEventArgs e)

{

SqlConnection connection = sender as SqlConnection;

MessageBox.Show //вывод информации о соединении и его состоянии

(

"Connection to" + Environment.NewLine +

"Data Source: " + connection.DataSource + Environment.NewLine +

"Database: " + connection.Database + Environment.NewLine +

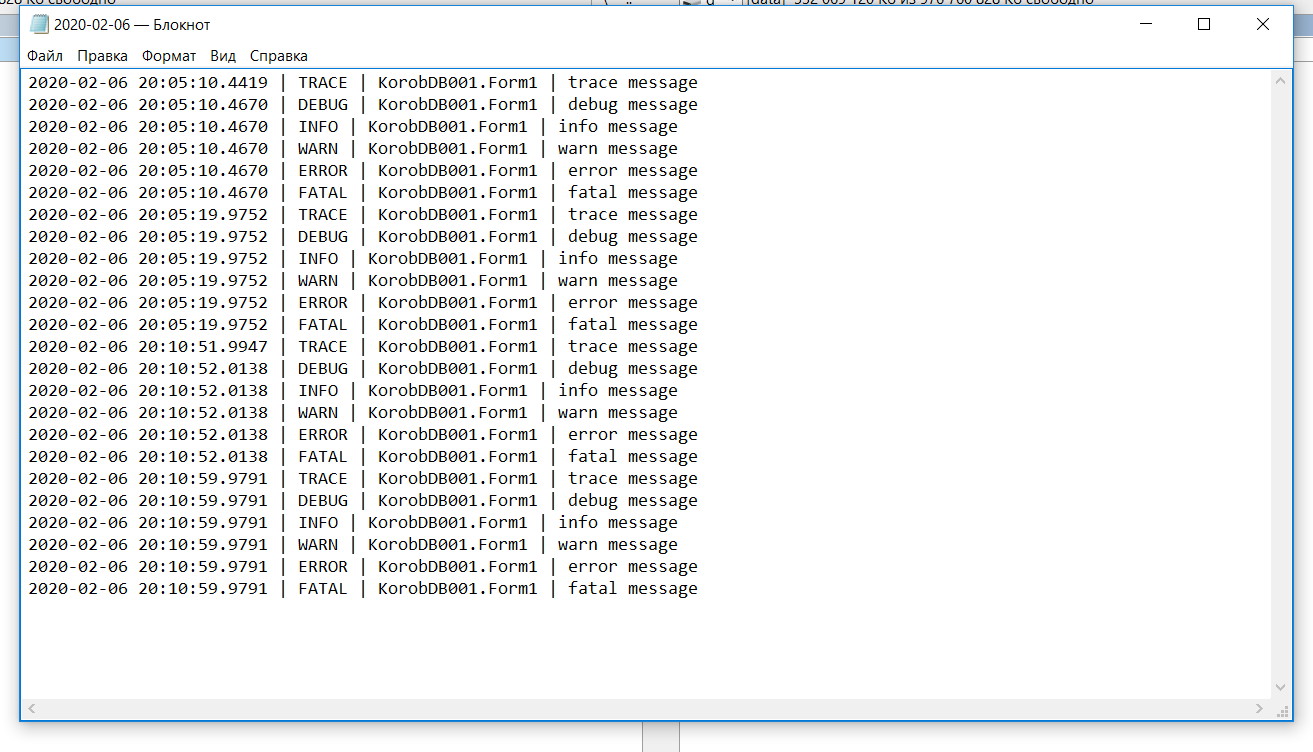
"State: " + connection.State

);

}

}

**Скриншоты работы:**



1. Создать проект C#, создать **connected layer**и реализовать действия:
   1. Хранимые процедуры;

2.1 Хранимые процедуры;

Хранимые процедуры являются еще одной формой выполнения запросов к базе данных. Но по сравнению с ранее рассмотренными запросами, которые посылаются из приложения базе данных, хранимые процедуры определяются на сервере и предоставляют большую производительность и являются более безопасными.

Объект SqlCommand имеет встроенную поддержку хранимых процедур. В частности у него определено свойство CommandType, которое в качестве значения принимает значение из перечисления System.Data.CommandType. И значение System.Data.CommandType.StoredProcedure как раз указывает, что будет использоваться хранимая процедура.Для выполнения команды нам потребуется sql-выражение и объект подключения:

Процедура: изменить параметр надбавки выбранного клиента в таблице.

public partial class Form1 : Form

{

DataSet ds;

SqlDataAdapter adapter;

string connectionString = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;

string strcom = "Select Фио,Гражданство,Дата\_Рождения,Пол,Таможенные\_формальности from dbo.Клиент";

private void button2\_Click(object sender, EventArgs e)

{

string Fio = dataGridView1.CurrentCell.Value.ToString();

byte fromalities = 0;

if (checkBox1.Checked)

{

fromalities = 1;

}

else

{

fromalities = 0;

}

string sqlExpression = "modife";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

// указываем, что команда представляет хранимую процедуру

command.CommandType = System.Data.CommandType.StoredProcedure;

// параметр для ввода имени

SqlParameter nadbavkaparam = new SqlParameter

{

ParameterName = "@Nadbavka",

Value = fromalities

};

// добавляем параметр

command.Parameters.Add(nadbavkaparam);

// параметр для ввода возраста

SqlParameter fioparam = new SqlParameter

{

ParameterName = "@FIO",

Value = Fio

};

command.Parameters.Add(fioparam);

var result = command.ExecuteScalar();

// если нам не надо возвращать id

//var result = command.ExecuteNonQuery();

if (fromalities == 1)

{

MessageBox.Show($"{fioparam.Value} now has had nadbavku");

}

else

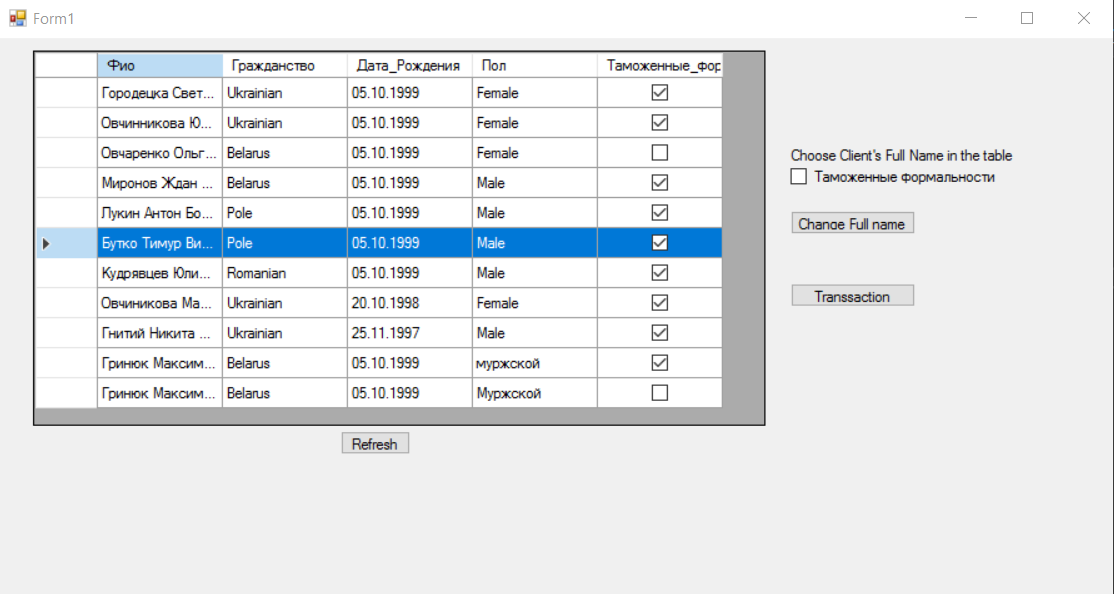
{

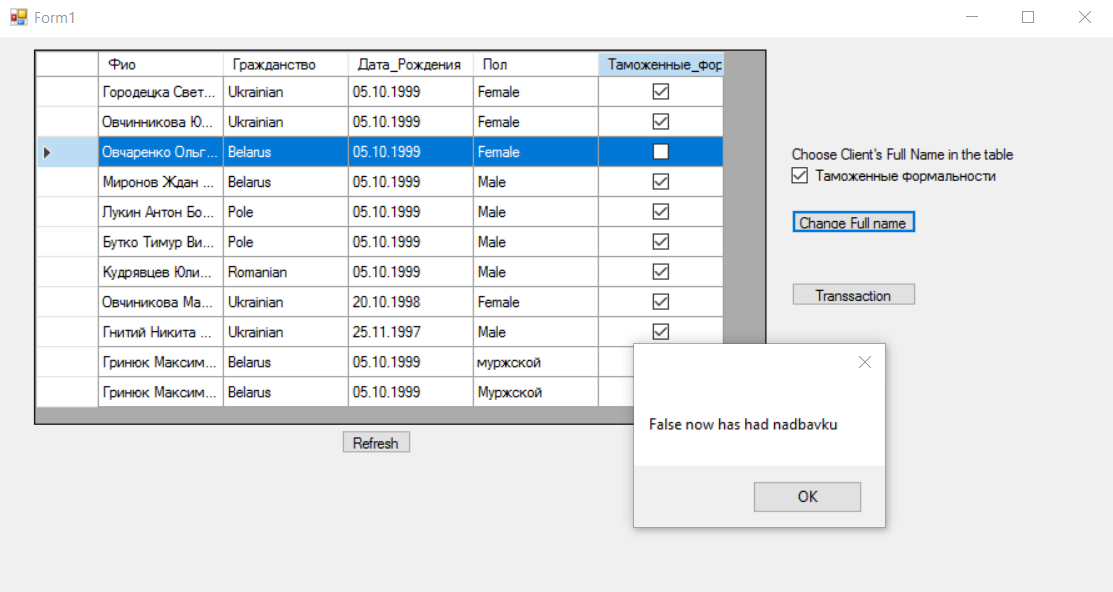
MessageBox.Show($"{fioparam.Value} now hasn't had nadbavku");

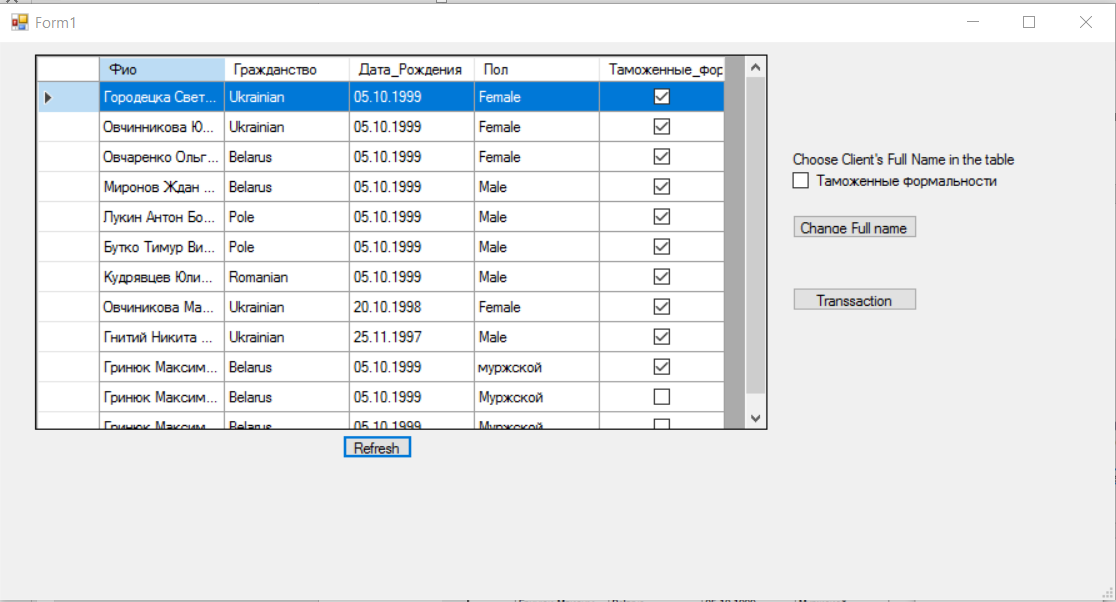
}

}

]}







2.2 Транзакция

Транзакции позволяют выполнять ряд операций в виде одного целостного пакета. И если хотя бы одна из этих операций завершится неудачно, то произойдет откат выполнения остальных операций.

Для начала транзакции надо вызвать метод BeginTransaction() объекта SqlConnection:

Транзакция: 2 вставка нового клиента и обновление данных.

private void button3\_Click(object sender, EventArgs e)

{

string fio = string.Empty;

string grazdanstvo = string.Empty;

DateTime dataofbirth = new DateTime();

string gender = string.Empty;

string connectionString = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlTransaction transaction = conn.BeginTransaction();

SqlCommand command = conn.CreateCommand();

command.Transaction = transaction;

try

{

// Включение команд в транзакцию

command.CommandText = "INSERT INTO Клиент(Фио, Гражданство, Дата\_Рождения, Пол) " +

"VALUES('Гринюк Максим Александровичч', 'Украинец','1999-10-05','Муржской')";

command.ExecuteNonQuery();

command.CommandText = "Update Клиент SET Гражданство = 'Belarus' WHERE Гражданство = 'Украинец'";

command.ExecuteNonQuery();

transaction.Commit();

MessageBox.Show("Data was added");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

// При возникновении любой ошибки выполняется откат транзакции.

transaction.Rollback();

}

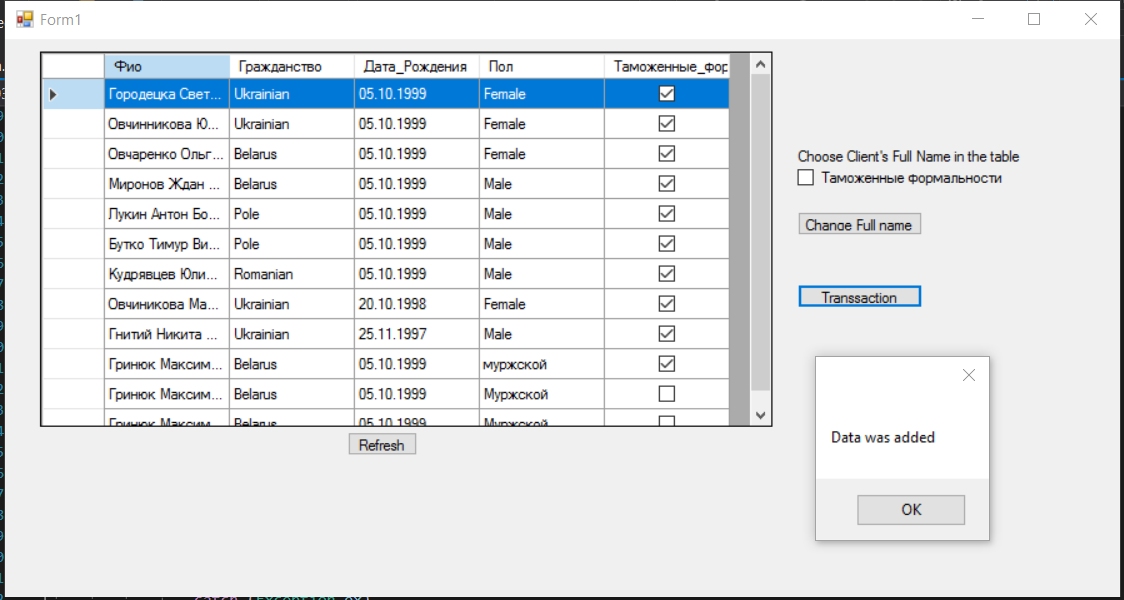
finally

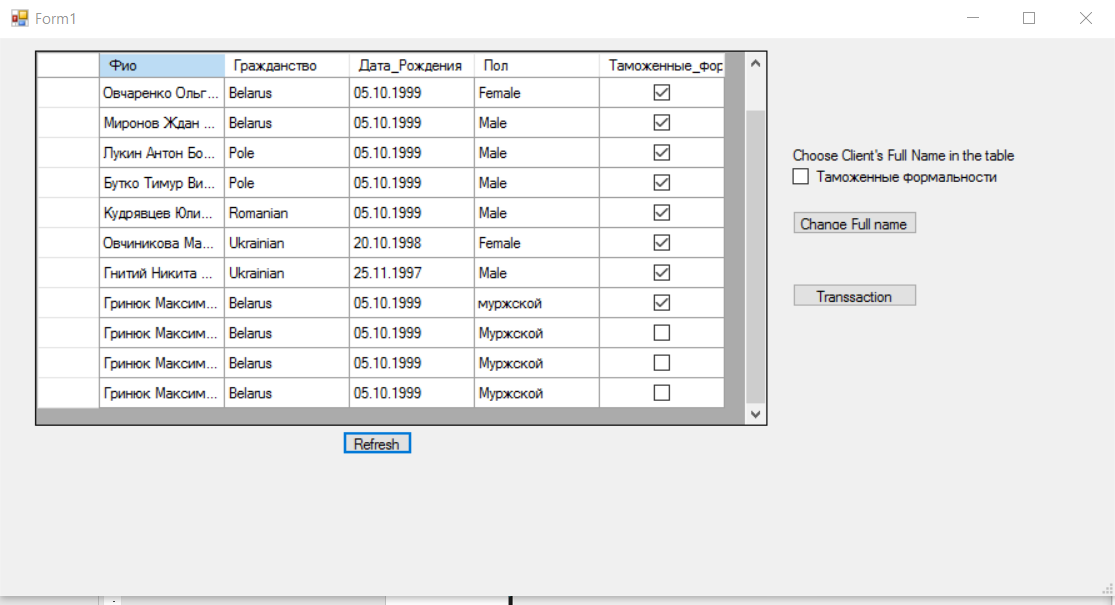
{

conn.Close();

}

}





**Вывод:** Изучил принципы построения пользовательского интерфейса, рассмотрел вопросы использования основных серверных элементов управления, применяемых для создания Web-приложений, рассмотрел вопросы организации ввода данных от пользователя.