## ЛАБОРАТОРНАЯ РАБОТА № 7-8. Разработка пользовательского интерфейса для работы с базами данных

В первую очередь нужно иметь подключение к базе данных postrgres в с#. Для того чтобы иметь возможность работать с базами данных PostgreSQL в нашем случае будем использовать Npgsql — .Net Data Provider for Postgresql скачиваем, добавляем Npgsql.dll в свой проект и копируем в папку с скомпилированным проектом необходимые библиотеки (чтобы работало): Npgsql.dll и Mono.Security.dll.

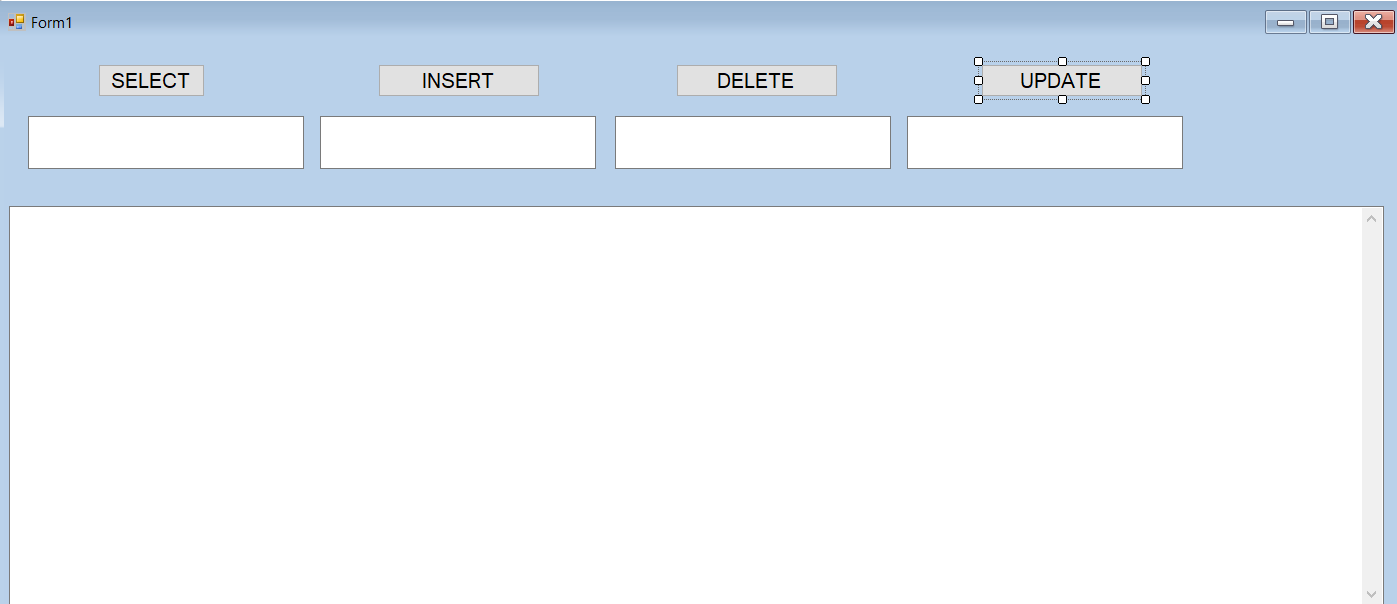
Начнём с подключения к базе:

string connstring = "Server=127.0.0.1; Port=5432; User Id=postgres; Password=pavel; Database=magazin;";

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

Далее займемся формой (нашем builder), выглядеть он будет так:



Для каждой кнопки запроса создадим методы их обрабатывающие.

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

{

string insert;

insert = textBox2.Text;

PostGreSQL pgTest = new PostGreSQL();

dataItems = pgTest.PostgreSQLtest2(insert);

tbDataItems.Clear();

for (int i = 0; i < dataItems.Count; i++)

{

tbDataItems.Text += dataItems[i];

tbDataItems.ScrollToCaret();

}

}

public List<string> PostgreSQLtest2(string insert)

{

try

{

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

NpgsqlCommand command = new NpgsqlCommand(insert, connection);

int count = command.ExecuteNonQuery();

if (count == 1)

dataItems.Add("Запись добавлена");

else

dataItems.Add("Не удается вставить новую запись");

connection.Close();

return dataItems;

}

catch (Exception msg)

{

MessageBox.Show(msg.ToString());

throw;

}

}

Выше был описан пример метода добавления в базу данных. При нажатии на кнопку INSERT, происходит считывание из поля под insert, после чего оно передается в метод, который после подключения к бд в postgres передает ей запрос и выполняет его. После всего выводиться запись добавлена или произошла ошибка при добавлении.

Исходный код

namespace PgSql

{

public partial class Form1 : Form

{

List<string> dataItems = new List<string>();

public Form1()

{

InitializeComponent();

}

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

{

string select;

select = textBox1.Text;

PostGreSQL pgTest = new PostGreSQL();

dataItems = pgTest.PostgreSQLtest1(select);

tbDataItems.Clear();

for (int i = 0; i < dataItems.Count; i++)

{

tbDataItems.Text += dataItems[i];

tbDataItems.ScrollToCaret();

}

}

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

{

string insert;

insert = textBox2.Text;

PostGreSQL pgTest = new PostGreSQL();

dataItems = pgTest.PostgreSQLtest2(insert);

tbDataItems.Clear();

for (int i = 0; i < dataItems.Count; i++)

{

tbDataItems.Text += dataItems[i];

tbDataItems.ScrollToCaret();

}

}

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

{

string delete;

delete = textBox3.Text;

PostGreSQL pgTest = new PostGreSQL();

dataItems = pgTest.PostgreSQLtest3(delete);

tbDataItems.Clear();

for (int i = 0; i < dataItems.Count; i++)

{

tbDataItems.Text += dataItems[i];

tbDataItems.ScrollToCaret();

}

}

private void button4\_Click(object sender, EventArgs e)

{

string update;

update = textBox4.Text;

PostGreSQL pgTest = new PostGreSQL();

dataItems = pgTest.PostgreSQLtest4(update);

tbDataItems.Clear();

for (int i = 0; i < dataItems.Count; i++)

{

tbDataItems.Text += dataItems[i];

tbDataItems.ScrollToCaret();

}

}

private void tbDataItems\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

}

}

namespace PgSql

{

public class PostGreSQL

{

List<string> dataItems = new List<string>();

public void PostgreSQL()

{

}

string connstring = "Server=127.0.0.1; Port=5432; User Id=postgres; Password=pavel; Database=magazin;";

public List<string> PostgreSQLtest1(string select)

{

try

{

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

NpgsqlCommand command = new NpgsqlCommand(select, connection);

NpgsqlDataReader dataReader = command.ExecuteReader();

dataItems.Add("id\_klient" + "| |" + "fio\_klient" + "| |" + "name\_klient" + "| |" + "adress" + "| |" + "nomer" + "\r\n");

for (int i=0; dataReader.Read(); i++)

{

dataItems.Add(dataReader[0].ToString() + "| |" + dataReader[1].ToString() + "| |" + dataReader[2].ToString() + "| |" + dataReader[3].ToString() + "| |" + dataReader[4].ToString() + "\r\n");

}

connection.Close();

return dataItems;

}

catch (Exception msg)

{

MessageBox.Show(msg.ToString());

throw;

}

}

public List<string> PostgreSQLtest2(string insert)

{

try

{

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

NpgsqlCommand command = new NpgsqlCommand(insert, connection);

int count = command.ExecuteNonQuery();

if (count == 1)

dataItems.Add("Запись добавлена");

else

dataItems.Add("Не удается вставить новую запись");

connection.Close();

return dataItems;

}

catch (Exception msg)

{

MessageBox.Show(msg.ToString());

throw;

}

}

public List<string> PostgreSQLtest3(string delete)

{

try

{

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

NpgsqlCommand command = new NpgsqlCommand(delete, connection);

int count = command.ExecuteNonQuery();

if (count == 1)

dataItems.Add("Запись удалена");

else

dataItems.Add("Не удается удалить запись");

connection.Close();

return dataItems;

}

catch (Exception msg)

{

MessageBox.Show(msg.ToString());

throw;

}

}

public List<string> PostgreSQLtest4(string update)

{

try

{

NpgsqlConnection connection = new NpgsqlConnection(connstring);

connection.Open();

NpgsqlCommand command = new NpgsqlCommand(update, connection);

int count = command.ExecuteNonQuery();

if (count == 1)

dataItems.Add("Запись изменена");

else

dataItems.Add("Не удается изменить запись");

connection.Close();

return dataItems;

}

catch (Exception msg)

{

MessageBox.Show(msg.ToString());

throw;

}

}

}

}