

## ТЕМА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ С ДОСТУПОМ К SQL БАЗАМ ДАННЫХ

В настоящее время технология **ADO.NET** предоставляет в распоряжение разработчика целый ряд классов, предназначенных для извлечения и обработки данных. Применение классов **ADO.NET** становится важным условием при создании современных управляемых данными приложений.

### ЛАБОРАТОРНАЯ РАБОТА 3. ТЕХНОЛОГИЯ СОЕДИНЕНИЯ С УДАЛЕННЫМ ИСТОЧНИКОМ ДАННЫХ. ОСНОВНЫЕ ОБЪЕКТЫ ПРОВАЙДЕРА ДАННЫХ: CONNECTION, COMMAND

#### Использование инструментов Visual Studio для создания базы данных

Среда Visual Studio .NET содержит удобный инструмент для работы с базами данных. Чтобы создать новую базу данных выполните следующие действия:

- Запустите интегрированную среду разработки Visual Studio .NET.
- В среде Visual Studio откройте окно **Обозреватель Серверов** (Server Explorer) командой меню **Вид - Обозреватель Серверов** (View – Server Explorer).

Для создания новой базы данных щелкните правой кнопкой мыши на объекте Подключиться к серверу (DataConnection). В контекстном меню выберите команду Новая база данных (New Database) рис.1.

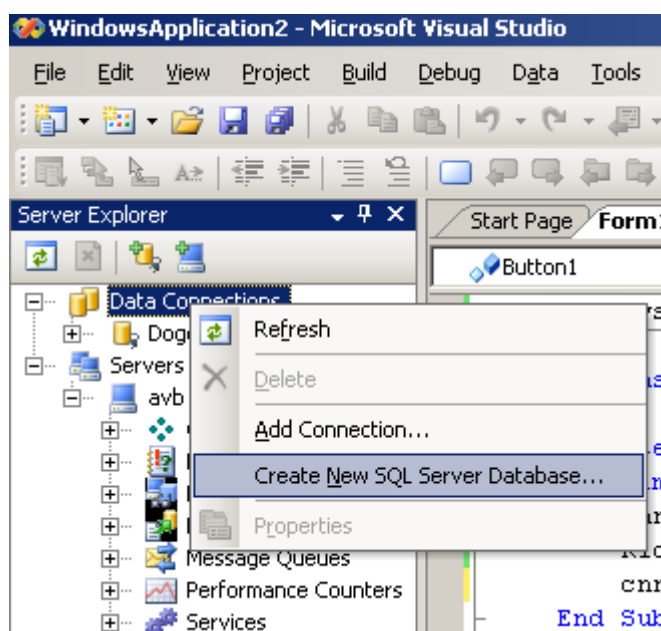


Рис.1. Создание базы данных в среде Visual Studio .NET

На экране появится диалоговое окно **Create Database** (создать базу данных рис. 2). Укажите имя сервера, введите имя новой базы данных и щелкните ОК.

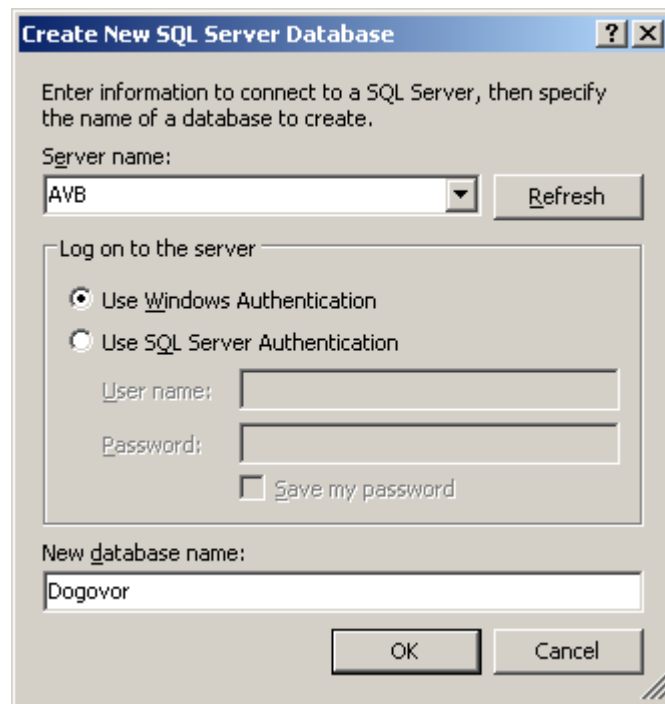


Рис.2. Подключение к SQL Server

После этого в окне Server Explorer появится новая база данных (рис.3). В нашем случае Dogovor.

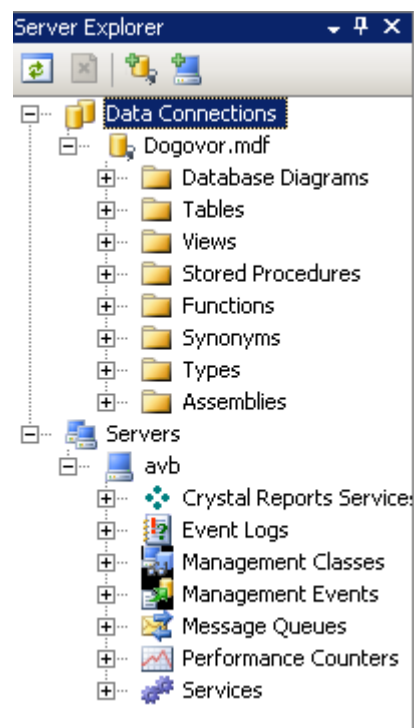


Рис.3. Окно обозревателя серверов Visual Studio

Для создания таблицы базы данных щелкните правой кнопкой мыши на узле Таблицы, а затем из контекстного меню выберите команду Создать таблицу. После ввода названия поля, следует указать его тип и размер.

Внесите несколько записей в таблицу.

Теперь можно приступить к разработке программного кода приложения и тестированию базы данных.

## Основные объекты провайдера данных: объект Connection

Объект **Connection** служит для установления соединения с заданным источником данных. Соединение можно настроить, редактируя нужным образом значения параметров строки подключения - свойство **ConnectionString**.

Для подключения к базе данных необходимо использовать свойство **ConnectionString** и метод **Open()**.

Объект **Connection** имеет методы **Open()** и **Close()**, которые открывают/закрывают подключение к базе данных со значениями свойств, определяемыми объектом **ConnectionString**.

Объект **Connection** имеет набор свойств, например, **ServerVersion**, **State**, которые отображают состояние объекта во время последней сетевой операции, выполненной при подключении.

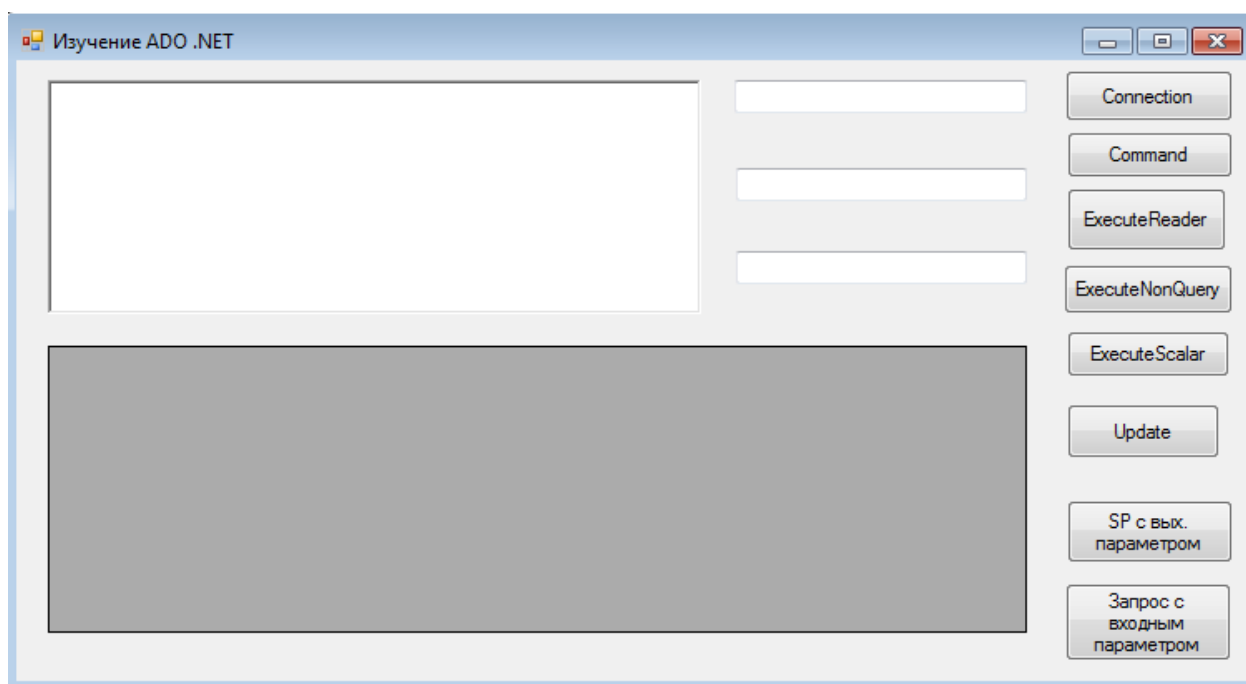


Рис.5. Пример интерфейса формы для изучения объектов ADO .NET

## Листинг программного кода приложения с подключением к базе данных

```
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 WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

        SqlConnection cnn = new SqlConnection(@"Data Source=gs-24-05-
n06\sqlexpress;Initial Catalog= namedb; Integrated Security = True");

        private void button1_Click(object sender, EventArgs e)
        {
            cnn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "select name from pocupatel";
            cmd.Connection = cnn;
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            { listBox1.Items.Add(reader[0].ToString()); }
            cnn.Close();
        }
    }
}

```

### Задание для самостоятельной работы

1. Создайте одну из таблиц вашей базы данных средствами Visual Studio.
2. Заполните таблицу записями.
3. В Visual Studio на языке C# создайте новый проект как приложение Windows Forms.
4. Разместите на форме элементы listBox (для отображения данных) и button (кнопка).
5. Перейдите в окно ввода программного кода
6. Добавьте необходимую ссылку на пространство имен: **using System.Data.SqlClient;**
7. Выполните подключение к базе данных из формы вашего приложения и отобразите один из столбцов в элементе listBox (см. пример кода).

### Содержание отчета

1. Титульный лист
2. Названия использованных элементов (имя SQL-сервера, имя базы данных, имя таблицы, названия столбцов таблицы).
3. Копия экрана с формой, на которой видны данные из таблицы.
4. Программный код