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

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 4. ОСНОВНЫЕ ОБЪЕКТЫ ПРОВАЙДЕРА ДАННЫХ: ОБЪЕКТ COMMAND

Задание 1. Основные объекты провайдера данных: объект Command

SqlCommand – класс представляет инструкцию Transact-SQL или хранимую процедуру, выполняемую над базой данных SQL Server. Класс имеет набор свойств, например:

- ***Connection*** - получает или задает объект SqlConnection, используемый данным экземпляром класса SqlCommand.
- ***CommandText*** - возвращает или задает инструкцию Transact-SQL, имя таблицы или хранимую процедуру, выполняемую для источника данных.
- ***CommandType*** - возвращает или задает значение, указывающее, как будет интерпретироваться свойство CommandText.

В качестве примера SQL-команды просмотрим содержание поля postavshik в таблице dogovor (рис.1).

```
SqlCommand cmd = new SqlCommand();
//кнопка Command
private void button2_Click(object sender, EventArgs e)
{
    cmd.Connection = cnn;
    //Значение Text является значение по умолчанию - можно не описывать
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select postavshik from dogovor order by dogovor_date DESC";
    cnn.Open();
    richTextBox1.Text = Convert.ToString(cmd.ExecuteScalar());
    cnn.Close();
}
```

Рис.1. Использование объекта SqlCommand

Результат выполнения команды показан на рис.2. Метод ExecuteScalar() возвращает первое поле первой найденной записи.

dogovor: Запрос(nataliapc.dogovor) | Form1.cs [Конструктор] | Form6.cs

	do...	tovar	pocupatel	postavshik	dogovor_date
	1	винты	ХозМаг №1	ООО "Склад винтов"	05.11.2013 0:00:...
	2	винты	ХозМаг №1	ООО "Склад винтов"	12.04.2013 0:00:...
	3	болты	ХозМаг№2	ООО "Склад болтов"	20.09.2013 0:00:...
	4	болты	Стройтовары	ООО "Склад болтов"	01.12.2013 0:00:...

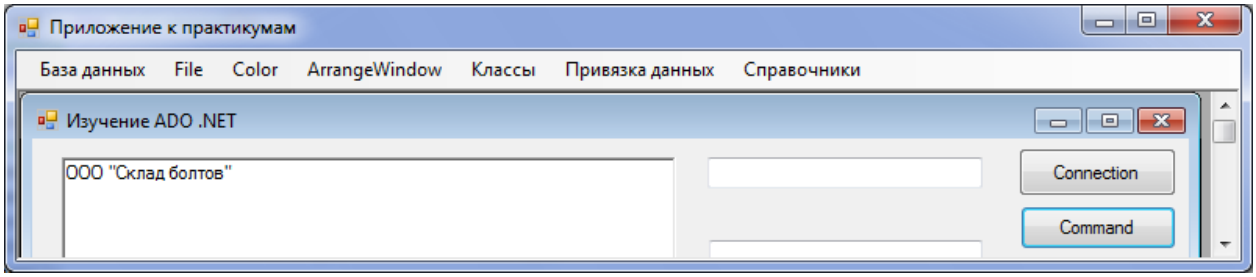


Рис. 2. Выполнение SQL-команды

Задание для самостоятельной работы

1. Измените сортировку найденных записей с DESC на ASC и проверьте результат выполнения команды Select.
2. Класс SqlCommand имеет несколько перегруженных версий конструктора. Самостоятельно используйте в своем приложении следующие версии конструктора класса:
   - SqlCommand(string) - инициализирует новый экземпляр класса SqlCommand с текстом запроса.
   - SqlCommand(string, SqlConnection) - инициализирует новый экземпляр класса SqlCommand текстом запроса и подключением SqlConnection.

Задание 2. Выполнение команд: метод ExecuteScalar()

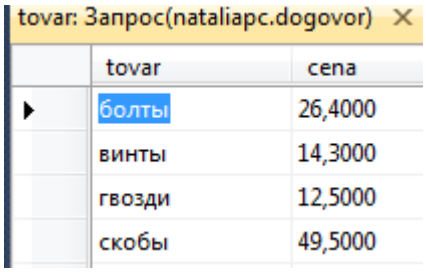
Метод ***ExecuteScalar***() – выполняет команду SQL и возвращает первое поле первой записи. В каких случаях используется этот метод?

Иногда требуется выполнить команду, которая возвращает скалярное значение, то есть только одно значение. Типичными примерами являются команды SQL для вычисления суммы, среднего значения или общего количества записей (Sum, Avg, Count, Min, Max). Другими примерами являются справочные таблицы для подстановки одного значения или команды, возвращающие логические значения. Метод ***ExecuteScalar***() выполняет заданную команду и возвращает значение только первого поля первой записи, остальные записи игнорирует.

Напишем для нашей базы данных Dogovor запрос на поиск товара с максимальной стоимостью:

```
cmd.CommandText = "select tovar from tovar order by cena DESC";
```

На рис. 3 представлен результат поиска записей в таблице базы данных и содержание самой таблицы.



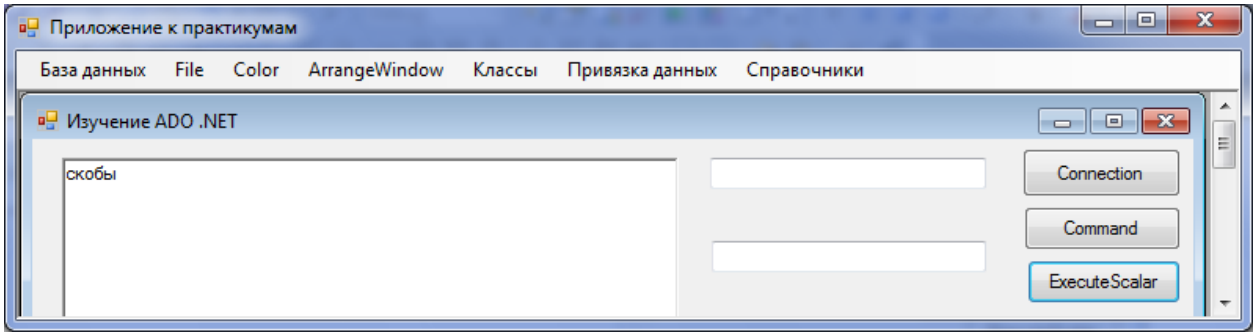


Рис. 3. Результат поиска записей методом ExecuteScalar()

Задание для самостоятельной работы

1. Протестируйте использование команды Insert Into, измените SQL-команду и просмотрите содержание таблицы:

```
cmd.CommandText = "insert into tovar values ('гвозди',12.5); select max(cena) from
tovar";
```

2. Протестируйте использование команды Update: напишите SQL-команду, которая изменяет некоторое поле таблицы, и вычисляет количество измененных записей. Просмотрите содержание таблицы.
3. Протестируйте использование команды Delete самостоятельно, измените SQL-команду и просмотрите содержание измененной таблицы.

Задание 3. Выполнение команд: метод ExecuteNonQuery()

Метод ***ExecuteNonQuery()*** - выполняет инструкцию Transact-SQL для установленного соединения и возвращает количество задействованных в инструкции строк.

При выполнении команд UPDATE, INSERT и DELETE метод ExecuteNonQuery() возвращают количество строк, которые были обработаны с их помощью. Для всех прочих типов команд возвращаемым значением является -1. В случае отката также возвращается значение -1.

Задание для самостоятельной работы

Добавьте в форму текстовое окно для ввода значения.

С помощью метода ExecuteNonQuery() организуйте использование команд Insert, Delete обязательно с входными параметрами и анализом количества вставленных или удаленных записей.

С помощью метода ExecuteNonQuery() опишите создание и удаление таблицы. Формальный пример команды Create Table приведен ниже. Команда удаления таблицы Drop Table.

CREATE TABLE MyTable (col1 INT PRIMARY KEY, col2 VarChar(25));

DROP TABLE MyTable;

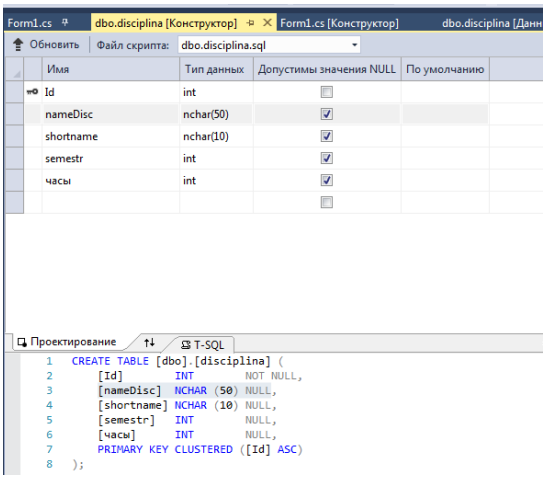
Содержание отчета

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. Интерфейс приложения (форма).

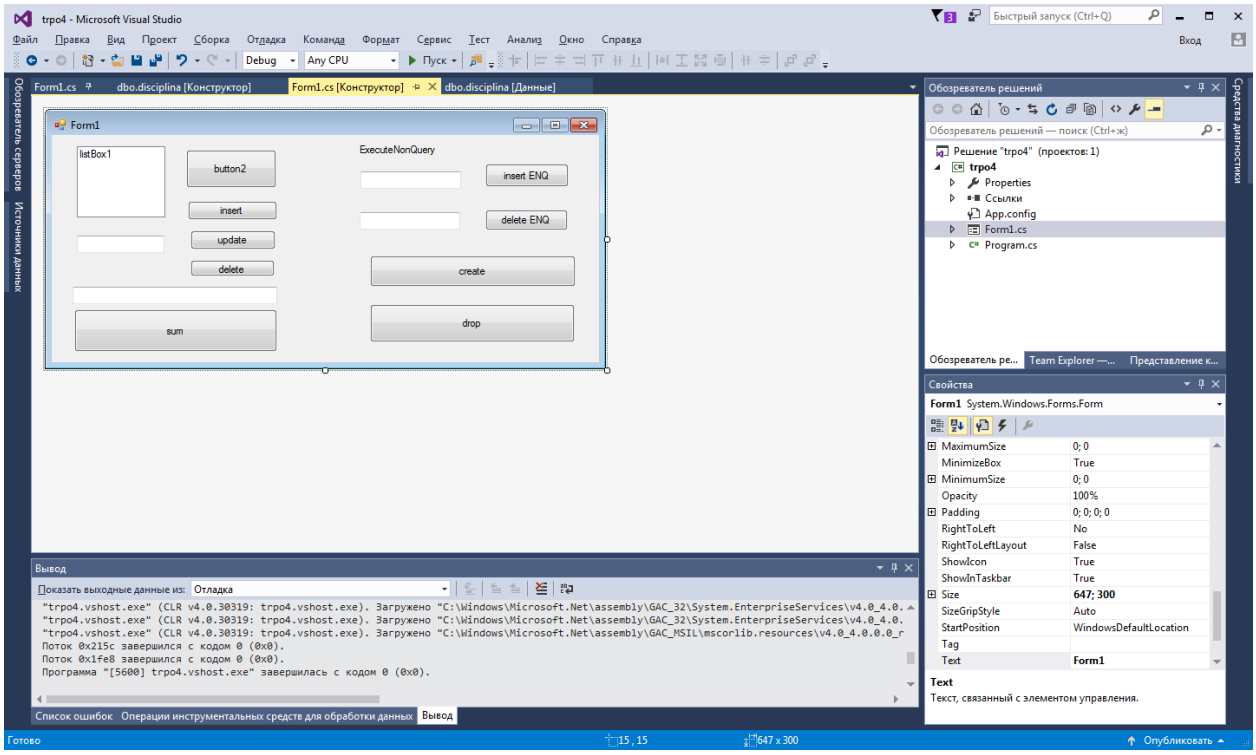
5. Программный код

Пример выполнения заданий

Поля таблицы



Форма приложения



Листинг

```
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 trpo4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        SqlConnection cnn = new SqlConnection(@"Data Source=rais12-;Initial Catalog=
newdb; Integrated Security = True");
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        SqlCommand cmd = new SqlCommand();
        private void button2_Click(object sender, EventArgs e)
        {
            cnn.Open();

            cmd.CommandText = "select nameDisc from  disciplina";
            cmd.Connection = cnn;
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            { listBox1.Items.Add(reader[0].ToString()); }

            cnn.Close();
            cnn.Open();  cmd.Connection = cnn;
            cmd.CommandText = "select semestr from  disciplina order by semestr DESC";
            textBox1.Text = "semestr #  "+ Convert.ToString(cmd.ExecuteScalar());
            cnn.Close();

        }

        private void button1_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "select sum([часы]) from  disciplina";
            textBox1.Text = "Всего часов =  " + Convert.ToString(cmd.ExecuteScalar());
            cnn.Close();

        }

        private void button3_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "insert into  disciplina values (10,'Объектно-
ориентированное програм','ООП',3, 70)";
            cmd.ExecuteScalar();
            cnn.Close();

        }

        private void button4_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "update  disciplina set semestr = 10 where [часы] > 50";
```

```
            cmd.ExecuteScalar();
            cnn.Close();

        }

        private void button5_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "delete from  disciplina where [часы] = 70";
            cmd.ExecuteScalar();
            cnn.Close();

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void button6_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "insert into  disciplina values (10,'Объектно-
ориентированное програм','ООП'," + textBox3.Text+ ", 70)";
            textBox3.Text = "Изменено =  " + Convert.ToString(cmd.ExecuteNonQuery());
            cnn.Close();

        }

        private void button7_Click(object sender, EventArgs e)
        {   cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "delete from  disciplina where semestr = " + textBox4.Text;
            textBox4.Text = "Удалено "+ Convert.ToString(cmd.ExecuteNonQuery());
            cnn.Close();

        }

        private void button8_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
        cmd.CommandText = "create table day (number int primary key, event nchar(30)) " ;
            cmd.ExecuteNonQuery();
            cnn.Close();

        }

        private void button9_Click(object sender, EventArgs e)
        {
            cnn.Open(); cmd.Connection = cnn;
            cmd.CommandText = "drop table day";
            cmd.ExecuteNonQuery();
            cnn.Close();

        }
    }
}
```