

КЛАССЫ COMMAND И DATAREADER

Технология *ADO.NET* предоставляет в распоряжение разработчика целый ряд классов, предназначенных для извлечения и обработки данных. Классы *Command* и *DataReader* позволяют выполнить SQL-оператор любого типа и прочитать результат.

Основные объекты провайдера данных: объект Command

Продолжим работу с windows-проектом. Добавьте в форму еще одну командную кнопку, которая будет выполнять подключение к базе данных и выполнять операции с объектом *Command*.

Объект *Command* предоставляет методы, которые можно использовать для выполнения команды, в зависимости от того, нужно ли извлечь полный результирующий набор, получить единственное значение или просто выполнить отличную от запроса команду. Рассмотрим использование этих методов в windows-приложении.

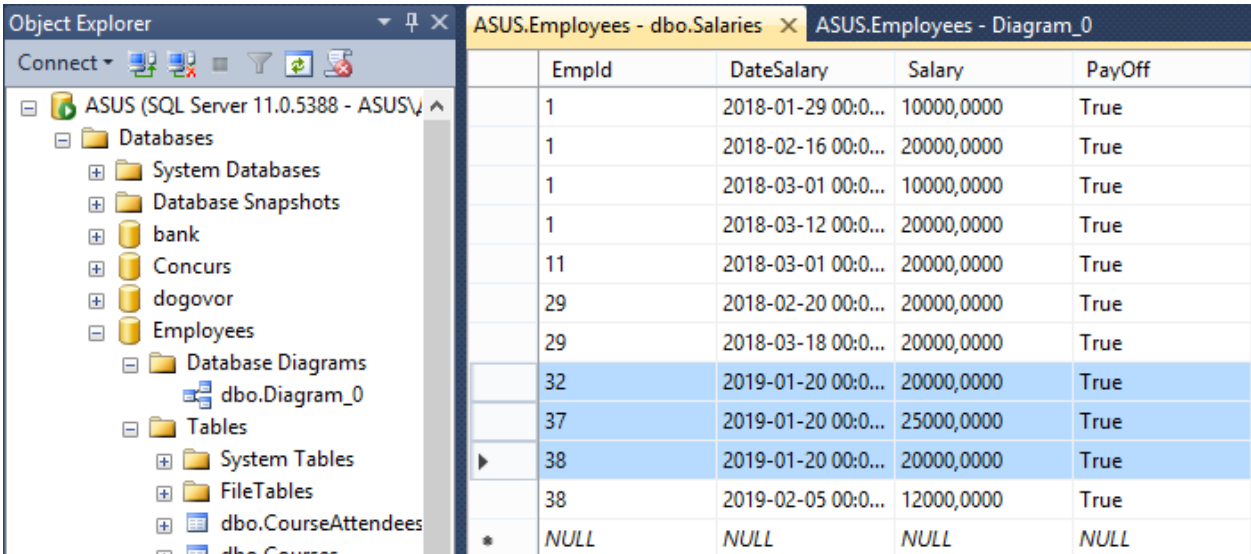
Метод *ExecuteScalar()* выполняет команду SQL и возвращает первое поле первой найденной записи. В каких случаях используется этот метод?

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.

Разработаем в win-форме следующий пример команды: требуется определить количество сотрудников, которым была выплачена заработная плата 20 января 2019 года. В базе данных *Employees* необходимо выполнить следующую команду:

```
select count(EmpId) from Salaries where Salaries.DateSalary='2019-01-20'
```

Если для выполнения этой команды использовать метод *ExecuteScalar()*, то результатом будет единственное возвращаемое командой значение *count(FIO)*. На рисунке 1 показана таблица, к которой будет выполнен запрос. На рисунках 2 и 3 приведены, соответственно, интерфейсы формы и веб-страницы, на рисунках 4 и 5 представлены программные коды кнопок *ExecuteScalar*.



EmpId	DateSalary	Salary	PayOff
1	2018-01-29 00:00:00	10000,0000	True
1	2018-02-16 00:00:00	20000,0000	True
1	2018-03-01 00:00:00	10000,0000	True
1	2018-03-12 00:00:00	20000,0000	True
11	2018-03-01 00:00:00	20000,0000	True
29	2018-02-20 00:00:00	20000,0000	True
29	2018-03-18 00:00:00	20000,0000	True
32	2019-01-20 00:00:00	20000,0000	True
37	2019-01-20 00:00:00	25000,0000	True
38	2019-01-20 00:00:00	20000,0000	True
38	2019-02-05 00:00:00	12000,0000	True
*	NULL	NULL	NULL

Рисунок 1 – Записи в таблице Salaries

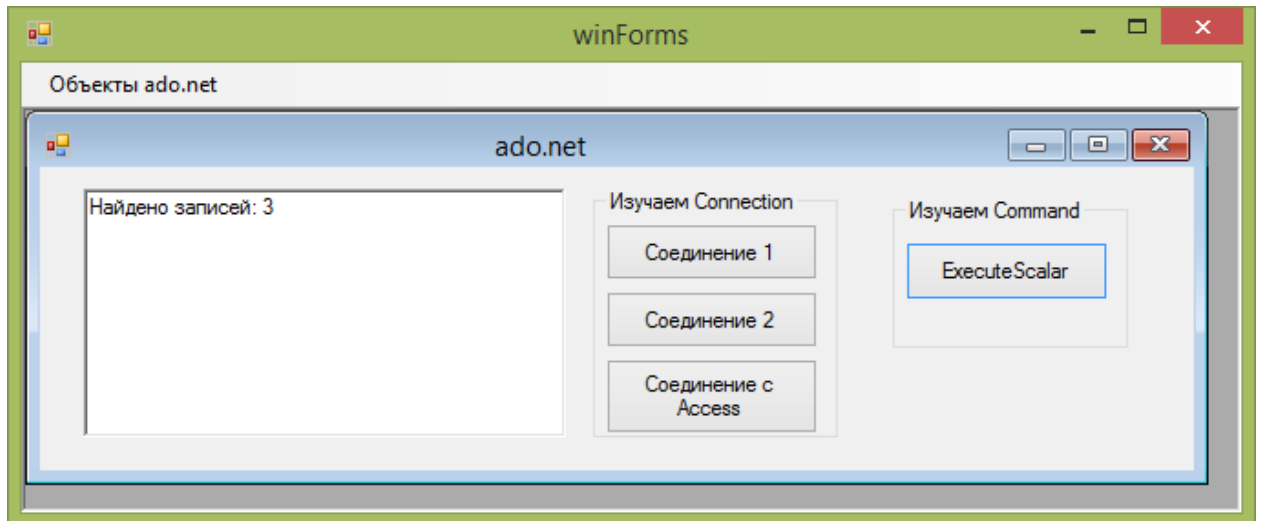


Рисунок 2 – Результат выполнения команды ExecuteScalar() в win-форме

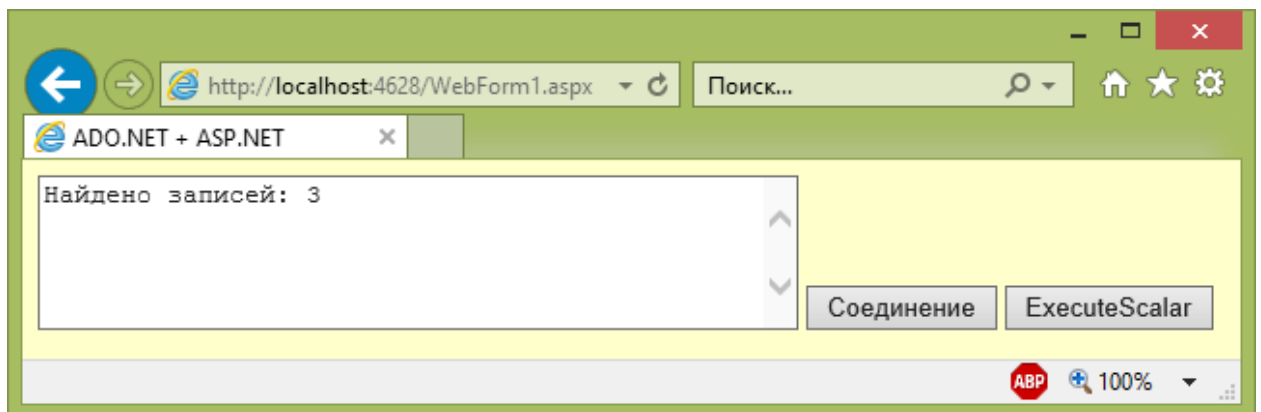


Рисунок 3 - Результат выполнения команды ExecuteScalar() в web-форме

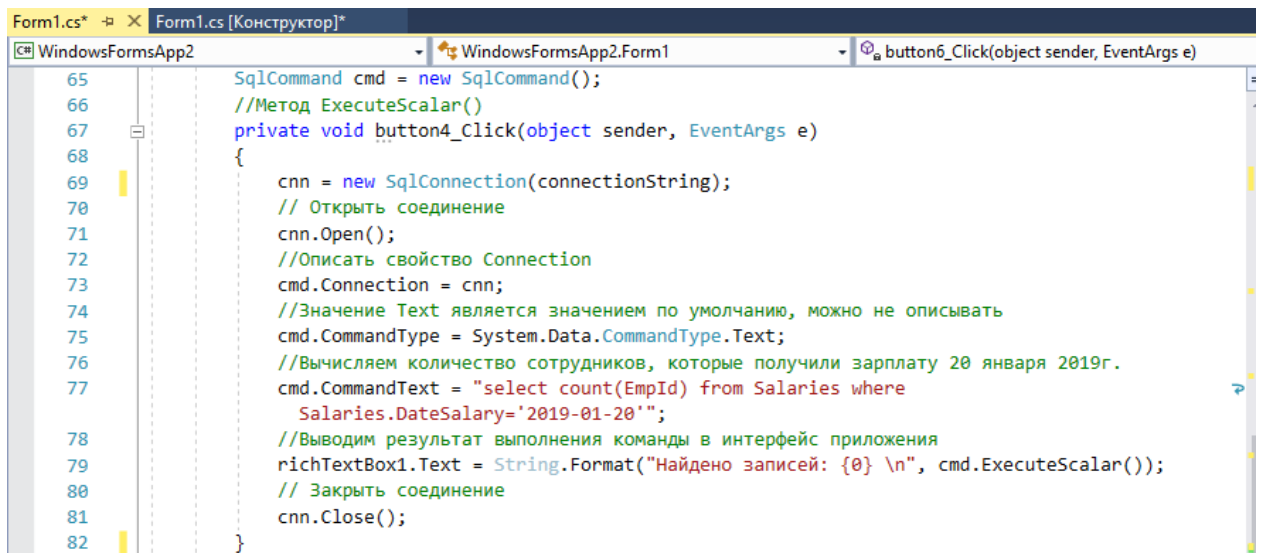


Рисунок 4 – Программный код кнопки в приложении WindowsForms

```

32 //Команда ExecuteScalar()
33 protected void Button2_Click(object sender, EventArgs e)
34 {
35     SqlConnection cnn = new SqlConnection(cnnStr);
36     //Используем перегруженную версию конструктора класса SqlCommand
37     SqlCommand cmd = new SqlCommand("select count(EmpId) from Salaries where
38         Salaries.DateSalary='2019-01-20'", cnn);
39     // Открыть соединение
40     cnn.Open();
41     //Выводим результат выполнения команды в интерфейс приложения
42     TextBox1.Text = String.Format("Найдено записей: {0} \n", cmd.ExecuteScalar());
43     // Закрывать соединение
44     cnn.Close();
45 }

```

Рисунок 5 – Программный код кнопки в приложении WebForm

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.

Продолжаем работу с проектами. Задача – добавить описание нового курса для повышения квалификации сотрудников в таблицу Courses. На рисунке 6 приведен программный код команды Insert.

```

84 //Метод ExecuteNonQuery()
85 private void button5_Click(object sender, EventArgs e)
86 {
87     cnn = new SqlConnection(connectionString);
88     cnn.Open();
89     cmd.Connection = cnn;
90     //Вставляем новую строку в таблицу Courses
91     cmd.CommandText = "insert into Courses values(1, 'Изучение ADO.NET', '2019-03-05', '+7
92         (921)678-90-89',1)";
93     //Выполнить команду и показать результат
94     richTextBox1.Text = String.Format("Обработано записей: {0} \n", cmd.ExecuteNonQuery());
95     cnn.Close();
96 }

```

Рисунок 6 – Команда Insert для таблицы Courses

На рисунке 7 демонстрируется вставленная строка в таблице.

	CourseId	CourseDescription	CourseDate	telefon	очные
1	1	Изучение ADO.NET	2019-03-05 00:0...	+7(921)678-90-89	True

Рисунок 7 - Изменение таблицы базы данных

Метод *ExecuteReader()* выполняет запрос SELECT и возвращает объект *DataReader*.

Класс *DataReader* позволяет читать данные, возвращенные командой SELECT, по одной строке, в однонаправленном, доступном только для чтения потоке. Иногда это называют курсором. Использование *DataReader* представляет наиболее быстрый способ доступа к данным.

Одним из основных методов *DataReader* является метод *Read()*. Метод перемещает курсор строки на следующую строку в потоке. Метод *Read()* возвращает *true*, если существует следующая строка для чтения, или *false*, если прочитана последняя строка в наборе.

Решим задачу просмотра списка сотрудников с указанием даты рождения. Из таблицы Emp по команде SELECT можно получить массив данных и прочитать его в приложении. На рисунках 8 и 9 показан результат запроса к таблице. На рисунке 10 приведен программный код для реализации.

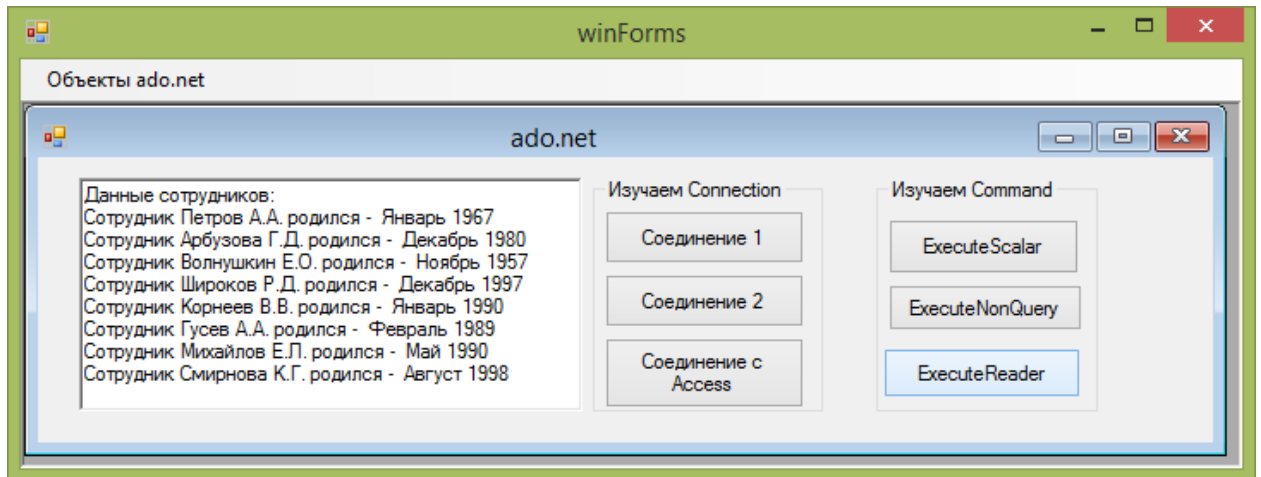


Рисунок 8 – Чтение набора данных из объекта DataReader

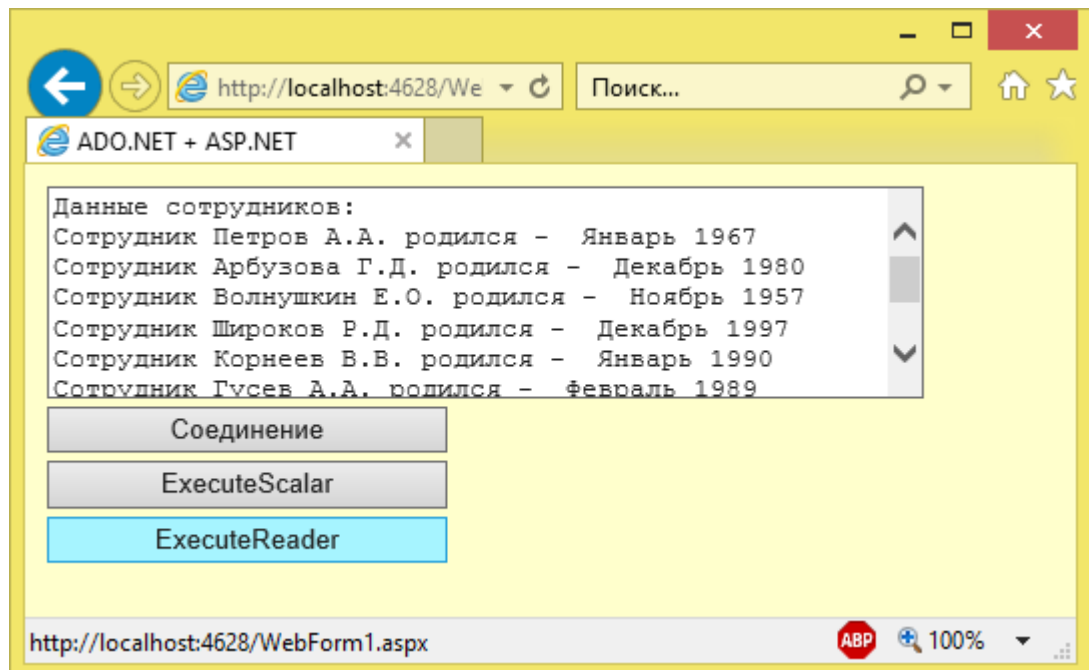


Рисунок 9 – Чтение DataReader для web-страницы

Обратите внимание на интерфейс страницы на рисунке 9: Для удобства размещения кнопок и текстового окна на странице была использована таблица. На рисунке 11 показана web-страница в режиме визуального конструирования. Внизу окна видны три вкладки: *Конструктор*, *Разделить*, *Исходный код*. С помощью этих переключателей можно видеть одновременно и теги разметки, и визуальные элементы одновременно (Разделить) или только теги (Исходный код), или только элементы (Конструктор). Займитесь разработкой вашей веб-страницы.

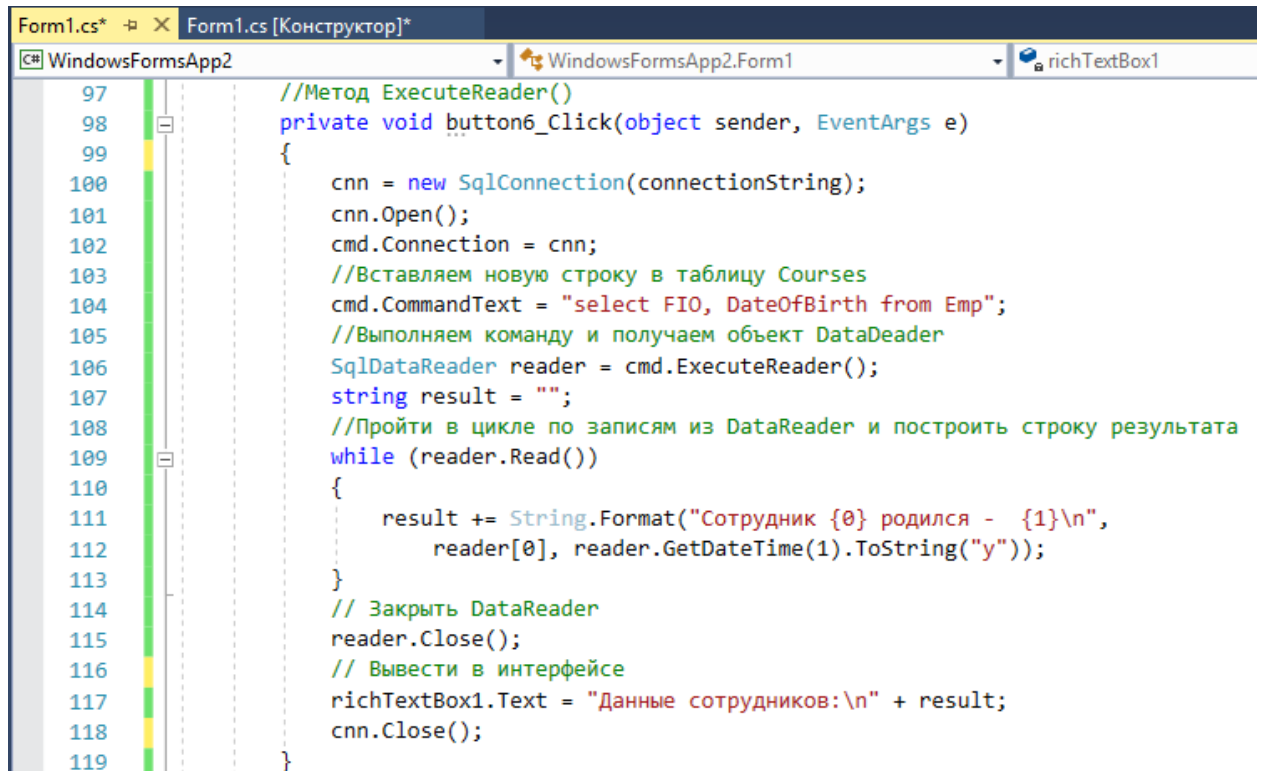


Рисунок 10 – Код win-формы для чтения небуферизуемого набора данных с помощью DataReader

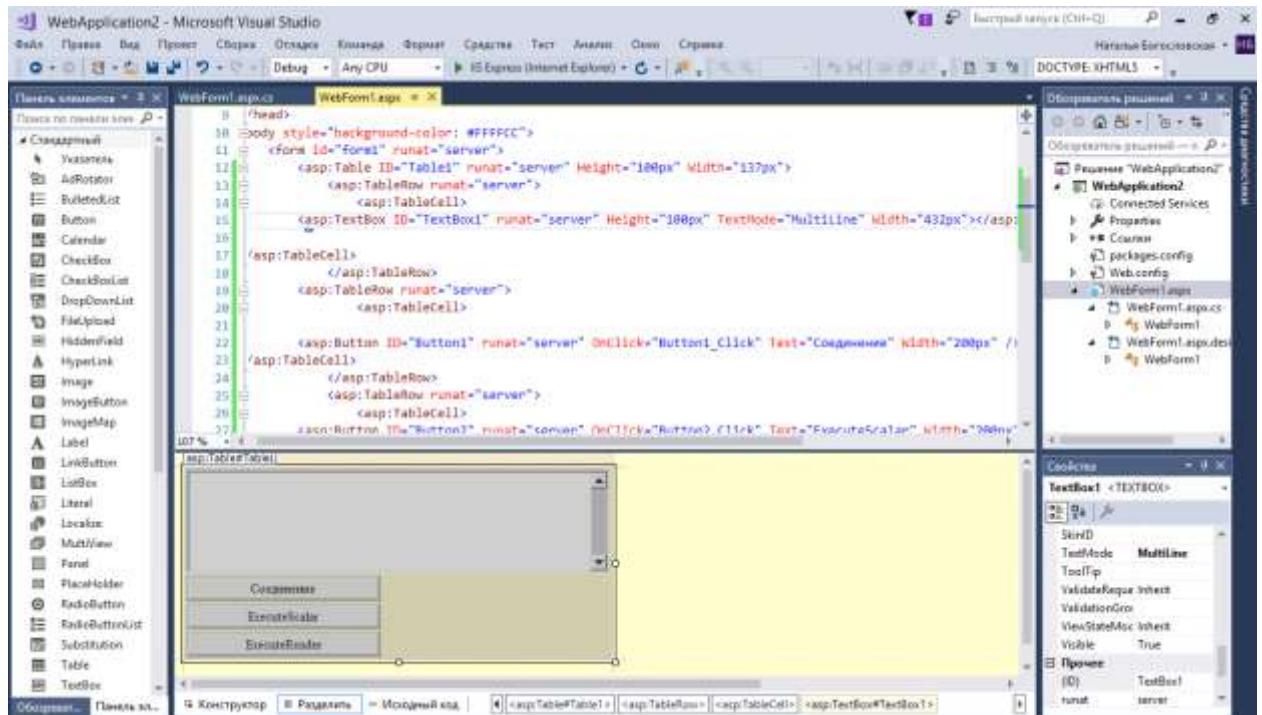


Рисунок 11 – Разработка интерфейса веб-формы в режиме конструирования

Команда SQL может быть набором команд, например, такой случай рассмотрен на рисунке 12: сначала составляется список сотрудников с указанием даты рождения, а затем – список сотрудников с названием отделов, в которых они работают.

На рисунке 13 демонстрируется последовательное чтение полученных массивов данных.


```
//Метод ExecuteReader()
private void button6_Click(object sender, EventArgs e)
{
    cnn = new SqlConnection(connectionString);
    cnn.Open();
    cmd.Connection = cnn;
    //Вставляем новую строку в таблицу Courses
    cmd.CommandText = "select FIO, DateOfBirth from Emp; select FIO, Description from Emp, Dep where Emp.DepId=Dep.DepId;";
    //Выполняем команду и получаем объект DataReader
    SqlDataReader reader = cmd.ExecuteReader();
    string result = "";
    //Пройти в цикле по записям из DataReader и построить строку результата
    while (reader.Read())
    {
        result += String.Format("Сотрудник {0} родился - {1}\n",
            reader[0], reader.GetDateTime(1).ToString("y"));
    }
    result += new string('*', 67);
    //Чтение следующего набора данных
    reader.NextResult();
    while (reader.Read())
    {
        result += String.Format("Сотрудник {0} работает в - {1}\n",
            reader[0], reader[1]);
    }
    // Закрыть DataReader
    reader.Close();
    // Вывести в интерфейсе
    richTextBox1.Text = "Данные сотрудников:\n" + result;
    cnn.Close();
}
```

Рисунок 12 – Чтение нескольких наборов данных методом `NextResult()` объекта `DataReader`

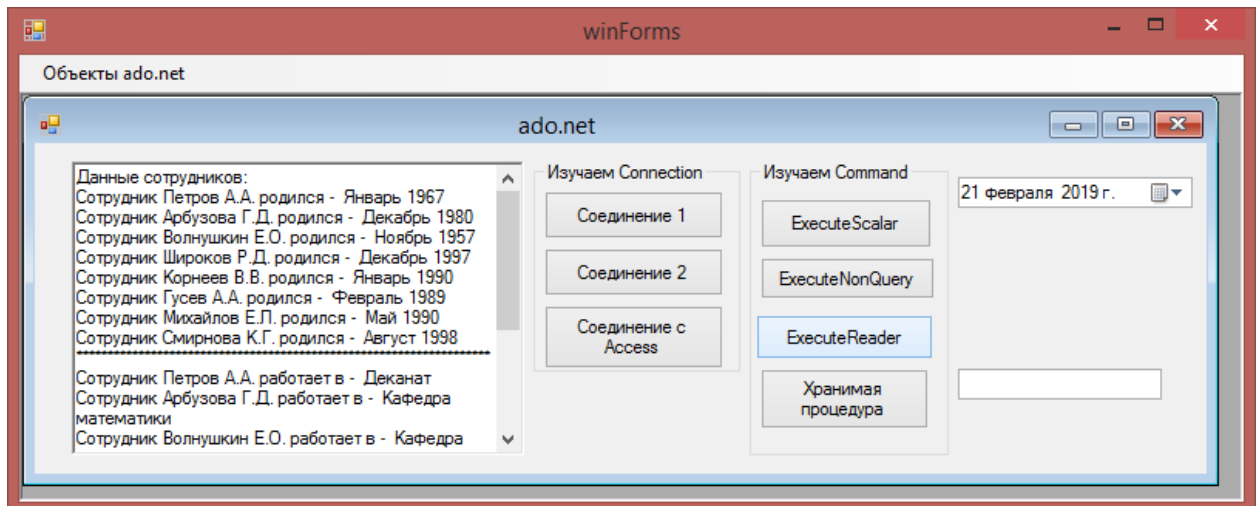


Рисунок 13 – Результат чтения двух наборов данных

Задание для самостоятельной работы

Для методов *ExecuteScalar()*, *ExecuteNonQuery()*, *ExecuteReader()* разработать команды SQL и применить их в windows- и web-проектах.

Формы и страницы должны иметь разработанный интерфейс.