## ХРАНИМЫЕ ПРОЦЕДУРЫ

Хранимая процедура представляет собой откомпилированный заранее набор инструкций SQL, хранимый под указанным именем и обрабатываемый как единое целое. Правильное использование хранимых процедур позволяет увеличить быстродействие приложения.

Хранимые процедуры существуют независимо от таблиц или каких-либо других объектов базы данных. Они вызываются клиентской программой, другой хранимой процедурой или триггером. Разработчик может управлять правами доступа к хранимой процедуре, разрешая или запрещая ее выполнение.

### Синтаксис параметризованных команд

**Параметризованная команда** — это команда, которая использует символы-заполнители в тексте команды SQL. Заполнитель указывает место для динамически применяемых значений, которые затем пересылаются через коллекцию **Parameters** объекту **Command**.

Вернемся к задаче из Практикума 3: требуется определить количество сотрудников, которым была выплачена заработная плата 20 января 2019 года. Решение было следующим:

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

Очевидно, что передача параметру фактического значения в тексте SQL-команды – не самый удачный способ. Эффективнее использовать именованный параметр, описание которого можно добавить в специализированную коллекцию *Parameters*. На рисунке 1 показано как будет изменен программный код приложения.

```
C# WindowsFormsApp2

→ 

¶

WindowsFormsApp2.Form1

     66
                    //Метод ExecuteScalar()
                    private void button4_Click(object sender, EventArgs e)
    68
    69
                        cnn = new SqlConnection(connectionString);
    70
                        // Открыть соединение
    71
                        cnn.Open();
     72
                       //Описать свойство Connection
    73
                        cmd.Connection = cnn;
     74
                       //Значение Text является значением по умолчанию, можно не описывать
    75
                        cmd.CommandType = System.Data.CommandType.Text;
     76
                        //Вычисляем количество сотрудников, которые получили зарплату в конкретную дату
                        cmd.CommandText = "select count(EmpId) from Salaries where Salaries.DateSalary=@dateZ";
    77
    78
                        //Добавляем параметр в коллекцию Parameters
                        cmd.Parameters.AddWithValue("@dateZ", dateTimePicker1.Value.Date);
    79
    80
                        //Выводим результат выполнения команды в интерфейс приложения
                        richTextBox1.Text = String.Format("Найдено записей: {0} \n", cmd.ExecuteScalar());
    81
    82
                        // Закрыть соединение
    83
                        cnn.Close();
    84
```

Рисунок 1 – Параметризованная команда

Для ввода конкретного значения даты потребуется добавить в интерфейс приложения элемент, например, dateTimePicker для win-формы или Calendar для web-формы, как показано на рисунках 2 и 3 соответственно.

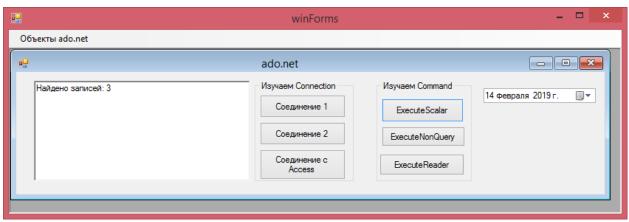


Рисунок 2 – Изменение в интерфейсе приложения

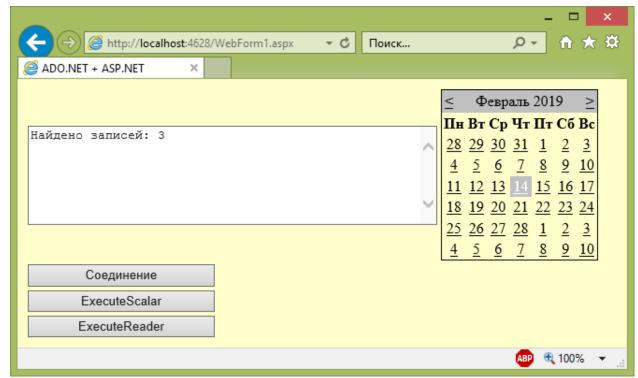


Рисунок 3 – Веб-интерфейс с добавленным элементом Calendar

# Объект Command и вызов хранимых процедур

Хранимые процедуры являются хорошо инкапсулированными блоками логики, которые могут принимать данные (через входные параметры) и возвращать данные (через результирующие наборы и выходные параметры).

При описании свойств объекта Command для **CommandType**. **Text** используется значение **CommandType**. **StoredProcedure**, а имя хранимой процедуры присваивается свойству **CommandText**.

Для создания хранимой процедуры средствами Visual Studio .NET выберите на вкладке Обозреватель серверов папку Хранимые процедуры и добавьте новую процедуру. Для создания хранимой процедуры можно использовать Management Studio MS SQL Server как показано на рисунке 4.

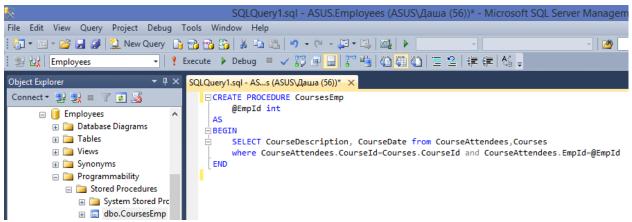


Рисунок 4 – Создание хранимой процедуры в менеджере сервера MS SQL Server

В любом случае среда разработки предложит шаблон для ввода текста процедуры (рисунок 4).

В шаблоне следует изменить имя процедуры, описать, если это необходимо, входные и выходные параметры, после оператора AS ввести код процедуры (рисунок 4).

Для вызова хранимой процедуры с параметрами необходимо:

- описать свойства объекта Command: CommandType и CommandText,
- добавить имена параметров в коллекцию Parameters объекта Command,
- указать где взять фактические значения параметров,
- выбрать и запустить подходящий метод объекта *Command*.

При наличии входного параметра, код вызова хранимой процедуры изменится, как показано на рисунке 5.

```
private void button7_Click(object sender, EventArgs e)
                                   134
Подключения даннью
                                   135
  asus.Employees.dbo
                                   136
                                                        //Запуск хранимой процедуры с входным параметром
     Tafimmer
                                   137
                                                        cnn = new SqlConnection(connectionString);
     Представления
                                                        SqlCommand cmd - new SqlCommand("CoursesEmp",cnn);
                                   138
       Хранимые процедуры
                                   139
                                                        //Значение StoredProcedure для типа команды
      CoursesEmp
                                                        cmd.CommandType = CommandType.StoredProcedure;
          ⊕ @Empld
                                                       //Добавление входного параметра с указанием где взять его значения
                                   141
       CoursesInsert
                                   142
                                                        cmd.Parameters.AddWithValue("@EmpId", Convert.ToInt32(textBox1.Text));
       Dep_Del
                                   143
                                                        cmn.Open();
       DepartmentEmployees
                                   144
                                                        //Выполняем команду и получаем объект DataDeader
       FioHead
                                                       SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                                   145
       InsertSal
                                                        //Создаем объект DataTable для преобразования
                                   146
       MaxidCourses
                                   147
                                                        DataTable datatable = new DataTable():
       SPOutPar
                                   148
                                                        datatable.Load(reader);
       SPwithPar
                                   149
                                                        // Вывести в интерфейсе
    Функции
                                                        dataGridView1.DataSource - null;
                                   150
    Синонимы
                                                        dataGridView1.DataSource = datatable;
                                   151
```

Рисунок 5 – Описание команды для вызова процедуры с входным параметром

В интерфейсе формы надо организовать ввод значения входного параметра и отображение результата выполнения хранимой процедуры, например, как на рисунке 6.

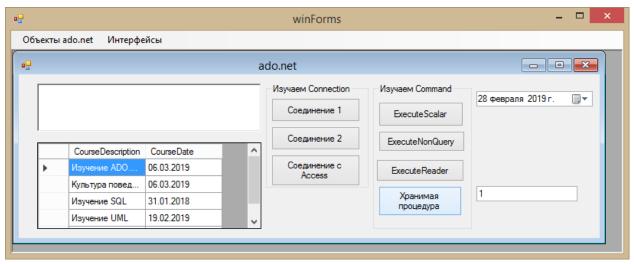


Рисунок 6 – Ввод значения входного параметра в win-форме

Особенности использования в хранимых процедурах выходных параметров можно увидеть на рисунках 7, 8 и 9. На рисунке 7 показан текст хранимой процедуры с входным и выходным параметром: вычисляется количество сотрудников, которые прошли повышение квалификации в указанную дату. На рисунке 8 показан способ описания объекта *Command* для процедуры с выходным параметром. На рисунке 9 – интерфейс windows-формы при запуске процедуры.

```
□CREATE PROCEDURE SPOutPar

□DateZ date,
□countFio int OUTPUT

AS
□ BEGIN
□ SELECT @countFio=count(EmpId)
FROM Courses, CourseAttendees
WHERE CourseAttendees.CourseId=CourseS.CourseId AND CourseDate=@DateZ
END
```

Рисунок 7 – Текст хранимой процедуры с входными и выходными параметрами

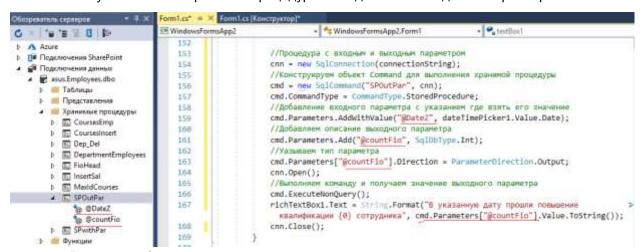


Рисунок 8 – Запуск хранимой процедуры с входными и выходными параметрами

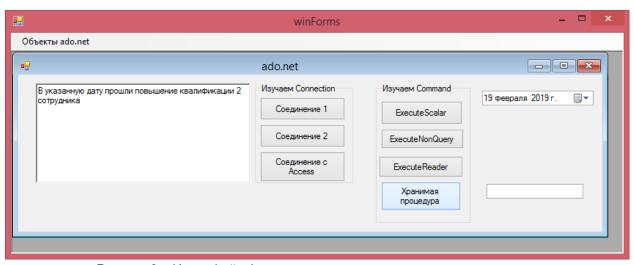


Рисунок 9 – Интерфейс формы при запуске процедуры с параметрами

# Задания для самостоятельной работы

- 1) Напишите и примените параметризованную команду для выборки данных, соответствующим указанным значениям. Продемонстрируйте ввод значений параметров команды из windows-интерфейса <u>или</u> web-интерфейса.
- 2) Напишите несколько хранимых процедур с входными и выходными параметрами по примерам, представленным в практикуме.

Материалы для знакомства с синтаксисом и примеры хранимых процедур посмотрите на сайте Microsoft:

https://docs.microsoft.com/ru-ru/sql/relational-databases/stored-procedures/create-a-stored-procedure

Для ввода значений входных параметров и просмотра значений выходных параметров используйте элементы интерфейса windows-форм <u>или</u> web-форм: разработка хранимых процедур никак не связана с типом интерфейса, а вызов процедуры и передача в нее параметров практически не имеют отличий в windows или web интерфейсах.