

Лабораторная работа 6. Пользовательские функции

Цель работы: научиться создавать пользовательские функции в среде SQL Server Management Studio.

Теоретические сведения

В среде SQL Server Management Studio все пользовательские функции находятся в папке «Функции», расположенной в папке «Программирование» в обозревателе объектов (рис.6.1).

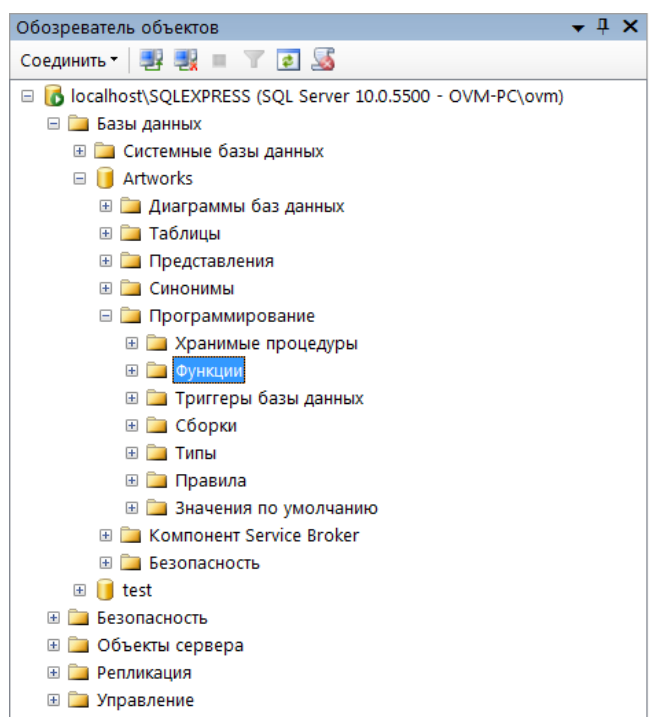


Рис.6.1

Начнём с создания скалярных пользовательских функций. Для создания новой скалярной пользовательской функции в обозревателе объектов щёлкните правой кнопкой мыши по папке «Функции» и в появившемся меню выберите пункт «Создать → Скалярная функция...». Появится окно новой скалярной пользовательской функции (рис.6.2).

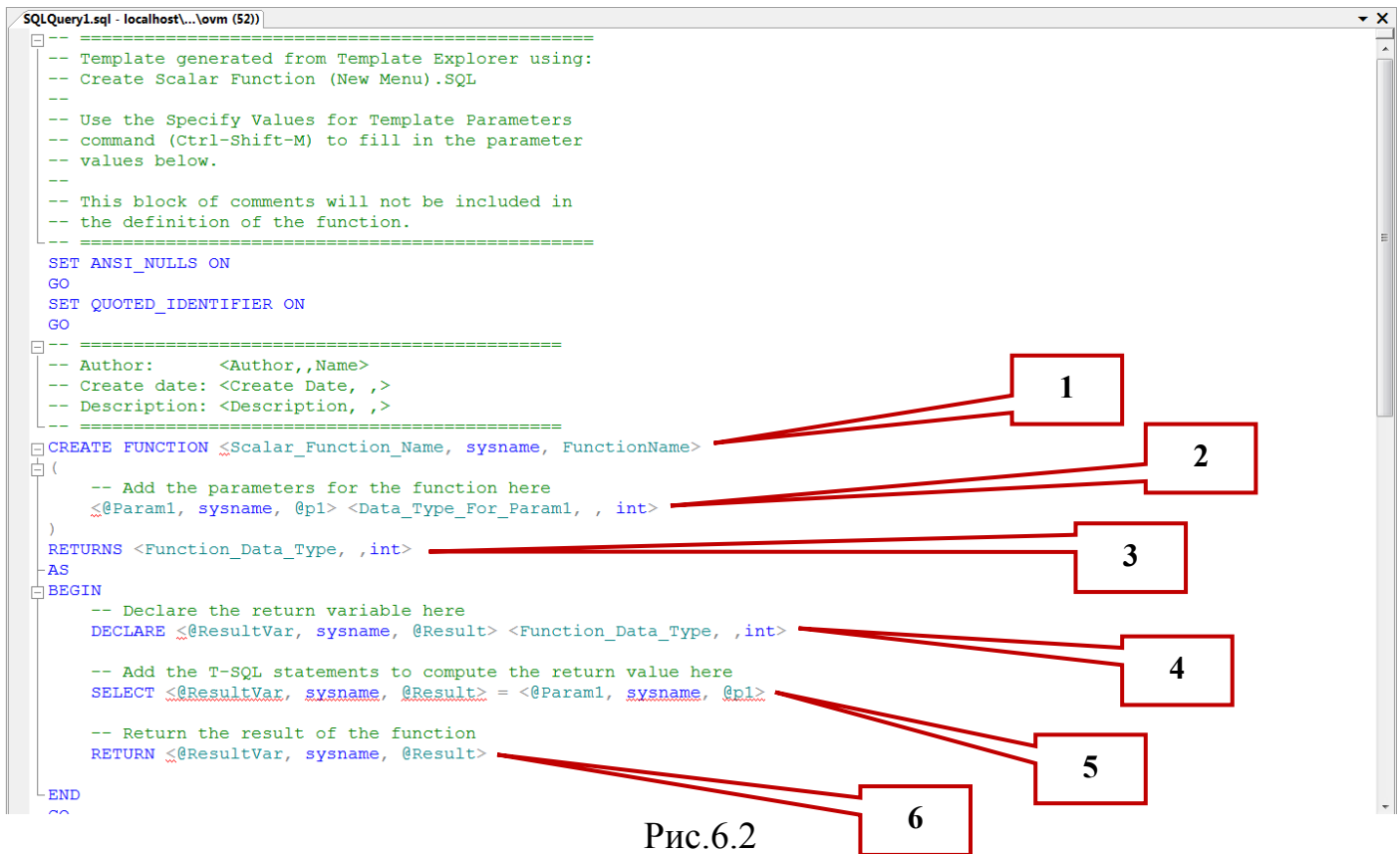


Рис.6.2

Синтаксис скалярной пользовательской функции похож на синтаксис хранимой процедуры. Но есть и существенные отличия (рис.6.2):

1. Область определения имени функции (Scalar_Function_Name);
2. Параметры, передаваемые в функцию (@param1). Определение параметров аналогично определению параметров в хранимой процедуре;
3. Тип данных значения возвращаемого функцией;
4. Область объявления переменных, используемых внутри функции.

Объявление переменных имеет следующий синтаксис:

```
DECLARE @<Имя переменной> <Тип данных>
```

5. Тело самой пользовательской функции, содержит команды языка T-SQL;
6. Команда RETURN возвращающая результат выполнения функции. Имеет следующий синтаксис:

```
RETURN @<Имя переменной с результатом>
```

Переменная должна быть того же типа данных, который был указан в пункте 3.

Создадим скалярную пользовательскую функцию, вычисляющую среднее трёх величин. В окне новой пользовательской функции наберите код представленный на рис. 6.3.

```
SQLQuery2.sql - localhost\... (52)*
-- Create Scalar Function (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the function.
=====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
=====
-- Author:      <Author,,Name>
-- Create date: <Create Date, ,>
-- Description: <Description, ,>
=====
CREATE FUNCTION FuncMeanValue
(
    -- Add the parameters for the function here
    @Value1 int, @Value2 int, @Value3 int
)
RETURNS real
AS
BEGIN
    -- Declare the return variable here
    DECLARE @Result real

    -- Add the T-SQL statements to compute the return value here
    SELECT @Result = (@Value1 + @Value2 + @Value3) / 3

    -- Return the result of the function
    RETURN @Result
END
GO
```

Рис.6.3

Рассмотрим более подробно код данной скалярной пользовательской функции:

1. CREATE FUNCTION FuncMeanValue определяет имя создаваемой функции как FuncMeanValue;
2. @Value1, @Value2, @Value3 определяют три параметра процедуры Value1, Value2 и Value3. Данным параметрам можно присвоить целые числа (тип данных int);
3. RETURNS real показывает, что функция возвращает вещественные числа (тип данных real);
4. DECLARE @Result real – объявляется переменная @Result для хранения результата работы функции вещественного типа (тип данных real);
5. SELECT @Result = (@Value1 + @Value2 + @Value3) / 3 вычисляет среднее и помещает результат в переменную @Result;
6. RETURN @Result возвращает значение переменной @Result.

Для создания функции, выполним вышеописанный код, нажав кнопку «Выполнить» на панели инструментов. В нижней части окна с кодом появится сообщение «Выполнение команд успешно завершено.».

Для проверки работы созданной скалярной пользовательской функции необходимо создать новый пустой запрос, нажав на кнопку «Создать запрос» на панели инструментов. В появившемся окне с пустым запросом наберите команду SELECT dbo.FuncMeanValue (3, 5, 4) и нажмите кнопку «Выполнить» на панели инструментов (рис.6.4).

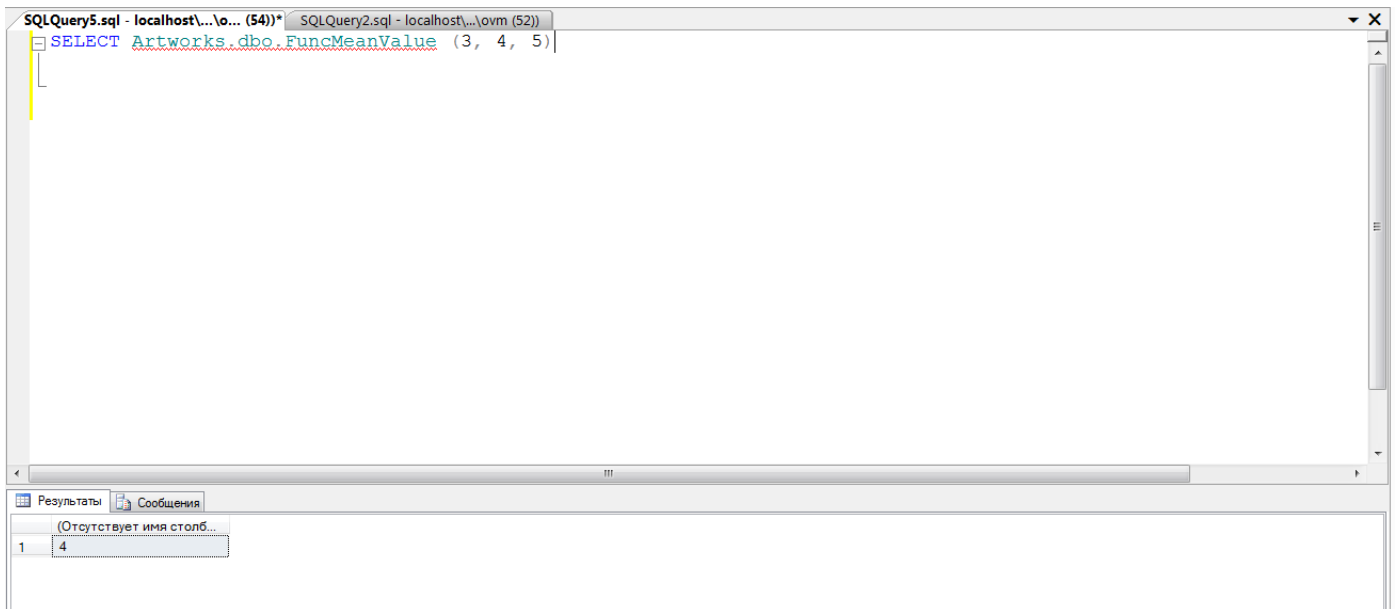


Рис.6.4

В нижней части окна с кодом появится результат выполнения новой скалярной пользовательской функции.

Теперь перейдём к созданию табличных пользовательских функций. Для создания табличной пользовательской функции в обозревателе объектов щёлкните правой кнопкой мыши по папке «Функции» и в появившемся меню выберите пункт «Создать → Встроенная функция, возвращающая табличное значение...». Появится окно новой табличной пользовательской функции (рис.6.5).

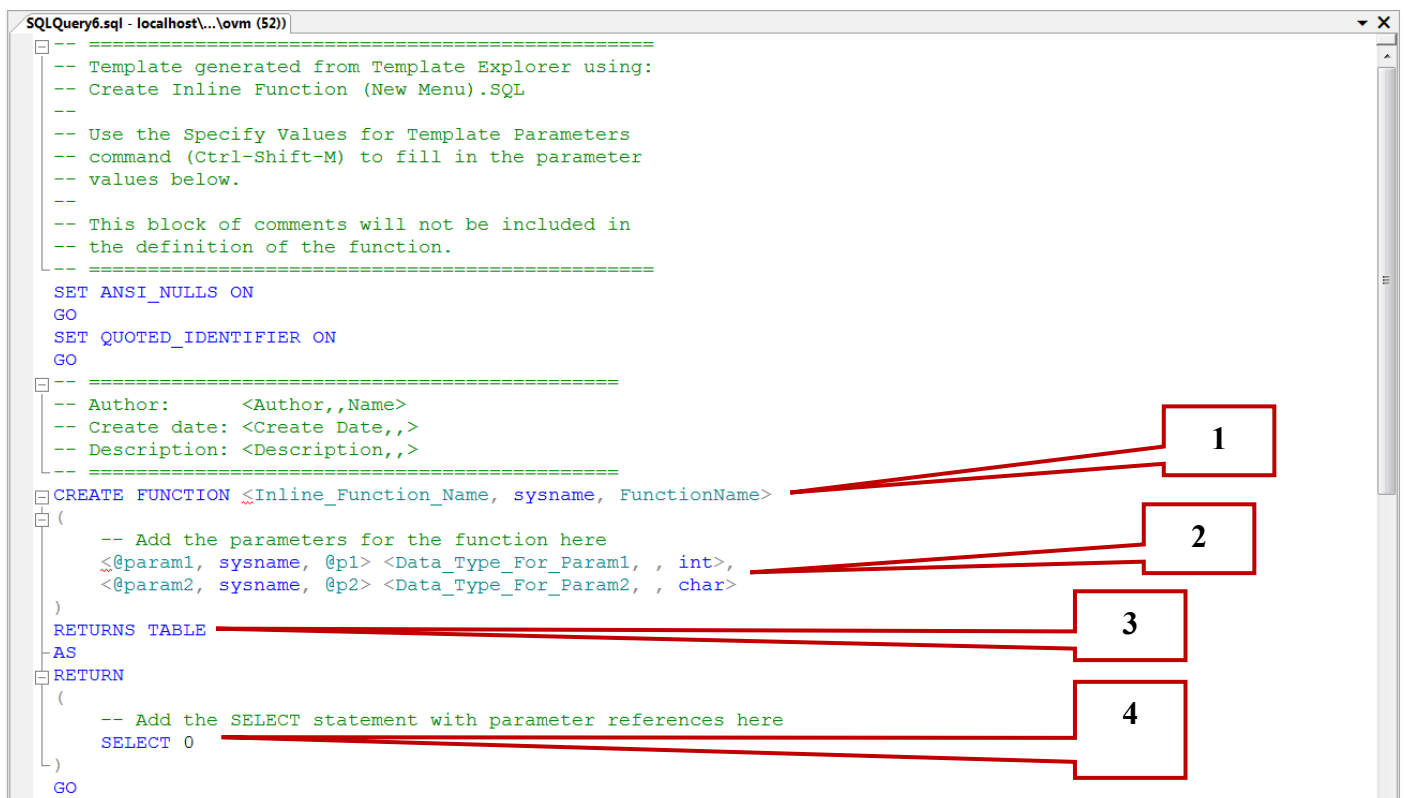


Рис.6.5

Табличная пользовательская функция состоит из следующих разделов:

1. Область определения имени функции (`Inline_Function_Name`);

2. Параметры, передаваемые в функцию (@param1, @param2);
3. RETURNS TABLE показывает, что функция возвращает таблицу;
4. Тело самой пользовательской функции состоит из команды SELECT языка T-SQL.

Рассмотрим создание табличной пользовательской функции, выбирающей информацию о художниках по стране их проживания. В окне новой пользовательской функции (рис. 6.5) наберем следующий код (рис. 6.6):

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER FUNCTION [GetArtists]
(
    -- Add the parameters for the function here
    @Country varchar(25)
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT  Authors.AuthorLastName, Authors.AuthorFirstName, Authors.AuthorFatherName,
            Authors.AuthorBirthdate, Authors.AuthorDeathdate
    FROM    Countries INNER JOIN
            Authors ON Countries.CountryCode = Authors.CountryCode
    WHERE   (Countries.CountryName = @Country)
)

```

Рис.6.6

Из кода функции видно, что она принимает один параметр типа varchar (название страны) и реализуется запросом:

```

SELECT  Authors.AuthorLastName, Authors.AuthorFirstName,
Authors.AuthorFatherName, Authors.AuthorBirthdate,
Authors.AuthorDeathdate
FROM Countries
INNER JOIN Authors
ON Countries.CountryCode = Authors.CountryCode
WHERE (Countries.CountryName = @Country)

```

Из таблицы «Authors» выбираются фамилии, имена, отчества, даты рождения и смерти художников, при этом таблица «Authors» соединяется (INNER JOIN) с таблицей «Countries» по столбцу «CountryCode». Условием соединения является равенство содержимого поля «CountryCode» передаваемому в функцию параметру.

Результат выполнения этой функции приведен на рис. 6.7.

AuthorLastName	AuthorFirstName	AuthorFatherName	AuthorBirthdate	AuthorDeathdate
1 Айвазовский	Иван	Константинович	1817-07-17	1900-04-19
2 Шишкин	Иван	Иванович	1832-01-25	1898-03-08
3 Репин	Илья	Ефимович	1844-08-05	1930-09-29
4 Коровин	Константин	Алексеевич	1861-12-05	1939-09-11
5 Шагал	Марк	Захарович	1887-07-06	1985-03-28
6 Рерих	Николай	Константинович	1874-10-09	1947-12-13
7 Полenov	Василий	Дмитриевич	1844-06-01	1927-07-18
8 Кандинский	Василий	Васильевич	1866-12-16	1944-12-13

Рис. 6.7

Работа с табличной функцией осуществляется так же, как и с обыкновенной таблицей.

Задание

Создать функции в соответствии с заданием для своего варианта.