

Функции определяемые пользователем

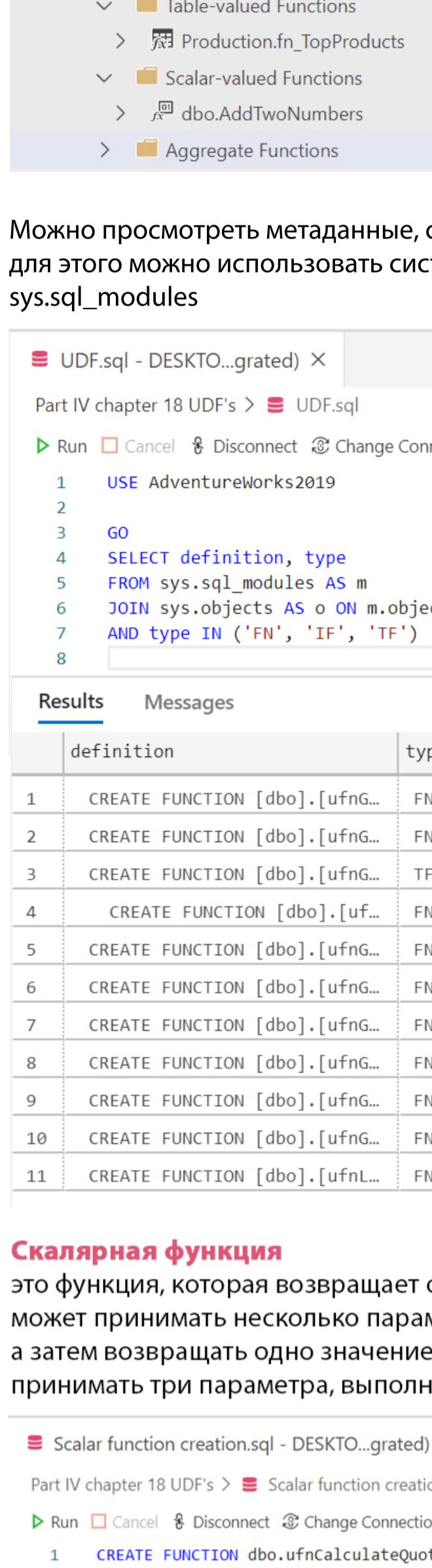
User Defined Functions (UDF)

UDF редко используются на практике и в продакшене, в основном из-за проблем с производительностью

Пользовательские функции могут принимать параметры

Пользовательские функции бывают трех различных типов

1. **Скалярные функции**, возвращающие одно значение
2. Встроенные **таблично-значные функции**, аналогичные представлениям (view)
3. **Многозначные табличные функции**, которые строят результатирующий набор с помощью кода



В Azure Data Studio во вкладке Connections все UDF хранятся в папке Functions

Здесь все функции разделены на скалярные, табличные и агрегирующие

Можно просмотреть метаданные, связанные с UDF's

для этого можно использовать системное представление sys.sql_modules

definition	type
CREATE FUNCTION [dbo].[ufnG...	FN
CREATE FUNCTION [dbo].[ufnG...	FN
CREATE FUNCTION [dbo].[ufnG...	TF
CREATE FUNCTION [dbo].[ufnG...	FN
CREATE FUNCTION [dbo].[ufnL...	FN

Скалярная функция

это функция, которая возвращает одно определенное значение. Функция может принимать несколько параметров, выполнять вычисления, а затем возвращать одно значение. Например, скалярная функция может принимать три параметра, выполнять вычисление и возвращать ответ

```
1 CREATE FUNCTION dbo.ufnCalculateQuotient -- создаём функцию
2 (
3     @Numerator numeric(5, 2), -- число 5 чисел до запятой, 2 после
4     @Denominator numeric(5, 2) = 1.0 -- у делителя есть значение по умолчанию
5 )
6 RETURNS numeric(5,2) -- функция будет возвращать число
7 AS
8 BEGIN
9     RETURN @Numerator/@Denominator; -- функция будет делить числа
10 END
11
12 GO
13 SELECT dbo.ufnCalculateQuotient(12.1, 7.45), -- делим числа передаваемые функции
14      dbo.ufnCalculateQuotient (7.0, DEFAULT) -- делим 7 на значение по умолчанию
15
```

Results Messages

(No column name)	(No column name)
1	1.62

Scalar function creation.sql - DESKTO...grated) X

```
16 USE AdventureWorks2019
17 GO
18 CREATE FUNCTION dbo.ufnGetOrderTotalByProduct(@ProductID int) -- создаём функцию
19 RETURNS int -- возвращающее число
20 AS
21 BEGIN
22     DECLARE @OrderTotal int;
23     SELECT @OrderTotal = sum(sod.OrderQty)
24     FROM Production.Product p
25     JOIN Sales.SalesOrderDetail sod
26     ON p.ProductID = sod.ProductID
27     WHERE p.ProductID = @ProductID
28     GROUP BY p.ProductID;
29     RETURN @OrderTotal; -- общее число заказов
30 END
31
32 GO
33 SELECT p.Name, dbo.ufnGetOrderTotalByProduct(p.ProductID) as OrderTotal -- результатом получим
34 FROM Production.Product p -- общее число заказов
35 ORDER BY OrderTotal DESC; -- для каждой компании
```

Results Messages

Name	OrderTotal
1 AWC Logo Cap	8311
2 Water Bottle - 30 oz.	6815
3 Sport-100 Helmet, Blue	6743
4 Long-Sleeve Logo Jersey, L	6592
5 Sport-100 Helmet, Black	6532
6 Sport-100 Helmet, Red	6266
7 Classic Vest, S	4247

Table-valued functions

такие функции не имеют синтаксиса начала и конца. Вместо этого оператор SELECT возвращается в виде виртуальной таблицы

```
1 CREATE FUNCTION dbo.ufnGetOrderTotalByProductCategory(@ProductCategoryID int) -- создаём таблично значимую функцию
2 RETURNS TABLE -- которая будет возвращать таблицу
3 AS
4 RETURN
5 (
6     SELECT p.ProductID, p.Name, sum(sod.OrderQty) as TotalOrders
7     FROM Production.Product p
8     JOIN Sales.SalesOrderDetail sod
9     ON p.ProductID = sod.ProductID
10    JOIN Production.ProductSubcategory s
11    ON p.ProductSubcategoryID = s.ProductSubcategoryID
12    JOIN Production.ProductCategory c
13    ON s.ProductCategoryID = c.ProductCategoryID
14    WHERE c.ProductCategoryID = @ProductCategoryID
15    GROUP BY p.ProductID, p.Name
16 )
17
18 GO
19 SELECT ProductID, Name, TotalOrders -- вызываем функцию с параметром id 1
20 ORDER BY TotalOrders DESC
```

Results Messages

ProductID	Name	TotalOrders
1	Mountain-200 Black, 38	2977
2	Mountain-200 Black, 42	2664
3	Mountain-200 Silver, 38	2394
4	Road-650 Black, 52	2270
5	Road-650 Red, 44	2254
6	Road-650 Red, 60	2232
7	Mountain-200 Silver, 46	2216
8	Mountain-200 Black, 46	2111

Using parameters.sql - DESKTO...grated) X

```
1 CREATE FUNCTION dbo.ufnProductOrderTotals (@ProductID nvarchar(100)) -- создаём функцию
2 RETURNS TABLE -- возвращающее значение
3 AS
4 RETURN
5 (
6     SELECT p.ProductID, p.Name, sum(sod.OrderQty) as TotalOrders
7     FROM Production.Product p
8     CROSS APPLY Sales.SalesOrderDetail sod
9     ON p.ProductID = sod.ProductID
10    WHERE p.ProductID = @ProductID
11    GROUP BY p.ProductID, p.Name
12 )
13
14 GO
15 SELECT t.Name, t.TotalOrders -- CROSS APPLY применяет
16 FROM Production.Product p
17 CROSS APPLY dbo.ufnProductOrderTotals(p.ProductID) t -- функцию к каждой строке
18 ORDER BY t.TotalOrders DESC
```

Results Messages

Name	TotalOrders
1 AWC Logo Cap	8311
2 Water Bottle - 30 oz.	6815
3 Sport-100 Helmet, Blue	6743
4 Long-Sleeve Logo Jersey, L	6592
5 Sport-100 Helmet, Black	6532
6 Sport-100 Helmet, Red	6266
7 Classic Vest, S	4247
8 Patch Kit/8 Patches	3865