

Лабораторный практикум

МДК 11.01 Технология разработки
и защиты баз данных

SQL

Куропаткина О.П.

ЛАБОРАТОРНАЯ РАБОТА №12

ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

Как уже было рассмотрено ранее, хранимые процедуры состоят из нескольких инструкций и имеют от нуля до нескольких входных параметров, но обычно не возвращают никаких параметров. В отличие от хранимых процедур, **функции всегда возвращают одно значение.**

Определяемая пользователем функция представляет собой подпрограмму, которая **принимает параметры, выполняет действия, такие как сложные вычисления, а затем возвращает результат этих действий в виде значения.**

```
1  /*Пользовательские функции возвращаемые скалярное значение (не табличное)*/
2  USE ForEducationSQL
3  GO
4
5  CREATE FUNCTION Hello() -- создать функцию
6  RETURNS nvarchar(30)    -- объявляем тип возвращаемого значения
7  AS
8  BEGIN --начало тела ф-ции
9  DECLARE @MyVar nvarchar(20) = 'Hello World!';
10 RETURN @MyVar; --возвращаемое значение ф-ции
11 END; -- конец тела ф-ции
12 GO
13
14 PRINT dbo.Hello();
```

144 %
Сообщения
Hello World!

Предложение RETURNS определяет тип данных значения, возвращаемого ОПФ. Это может быть почти любой стандартный тип данных, поддерживаемый системой баз данных, включая тип данных TABLE. Единственным типом данных, который нельзя указывать, является тип данных TIMESTAMP.

Определяемые пользователем функции могут быть либо скалярными, либо табличными. Скалярные функции возвращают атомарное (скалярное) значение. Это означает, что в предложении RETURNS скалярной функции указывается один из стандартных типов данных. Функция является табличной, если предложение RETURNS возвращает набор строк.

Блок BEGIN/END, содержит реализацию функции. Последней инструкцией блока должна быть инструкция RETURN с аргументом.

(Значением аргумента является возвращаемое функцией значение.) Внутри блока BEGIN/END разрешаются только следующие инструкции:

- инструкции присвоения, такие как SET;
- инструкции для управления ходом выполнения, такие как WHILE и IF;
- инструкции DECLARE, объявляющие локальные переменные;
- инструкции SELECT, содержащие списки столбцов выборки с выражениями, значения которых присваиваются переменным, являющимися локальными для данной функции;
- инструкции INSERT, UPDATE и DELETE, которые изменяют переменные с типом данных TABLE, являющиеся локальными для данной функции.

Вызов определяемой пользователем функции

Определенную пользователем функцию можно вызывать с помощью инструкций Transact-SQL, таких как SELECT, INSERT, UPDATE или DELETE. Вызов функции осуществляется, указывая ее имя с парой круглых скобок в конце, в которых можно задать один или несколько аргументов. Аргументы — это значения или выражения, которые передаются входным параметрам, определяемым сразу же после имени функции. При вызове функции, когда для ее параметров не определены значения по умолчанию, для всех этих параметров необходимо предоставить аргументы в том же самом порядке, в каком эти параметры определены в инструкции CREATE FUNCTION.

Нужно явно указывать схему при вызове пользовательской функции (dbo.).

Для примера создадим следующую таблицу:

```
19 CREATE TABLE TestTable
20 (
21     id int identity not null,
22     name varchar(25) not null,
23     CDate smalldatetime not null
24 )
25 GO
```

Далее при помощи цикла заполним тестовыми данными ранее созданную таблицу:

```
29 DECLARE @MyVar int =1;
30 DECLARE @MyVcVar varchar(10);
31
32 WHILE @MyVar < 20
33 BEGIN
34     SET @MyVcVar = 'Test ' + CAST(@MyVar as varchar);
35     -- заносим данные в таблицу
36     INSERT TestTable
37     ( name, CDate )
38     VALUES (@MyVcVar, DATEADD(MI, @MyVar, '01.01.2021 10:00:00'));
39
40     SET @MyVar = @MyVar + 1;
41 END
42 GO
```

144 %

Результаты Сообщения

	id	name	CDate
1	20	Test 1	2021-01-01 10:01:00
2	21	Test 2	2021-01-01 10:02:00
3	22	Test 3	2021-01-01 10:03:00
4	23	Test 4	2021-01-01 10:04:00
5	24	Test 5	2021-01-01 10:05:00
6	25	Test 6	2021-01-01 10:06:00
7	26	Test 7	2021-01-01 10:07:00
8	27	Test 8	2021-01-01 10:08:00
9	28	Test 9	2021-01-01 10:09:00

Попытаемся осуществить выборку данных из таблицы по заданной дате.

```
44 /* данная выборка будет пустая, так как поле CDate
45    содержит значение типа datetime (возвращает не только
46    дату но и текущее время до 3-х сотых секунды)*/
47 SELECT * FROM TestTable
48 WHERE CDate = '01.01.2021';
49 GO
```

144 %

Результаты Сообщения

	id	name	CDate
--	----	------	-------

Создадим пользовательскую функцию, которая будет приводить входной аргумент к типу данных date.

```
52 CREATE FUNCTION DateOnly (@Date datetime) -- создаем функцию, вводя ее имя и аргумент
53 RETURNS date --объявляем возвращаемый тип
54 AS
55 BEGIN --тело функции
56     RETURN CAST(@Date as date); -- приводим к типу date возвращаемое значение
57 END
58 GO
```

Применим созданную функцию в запросе.

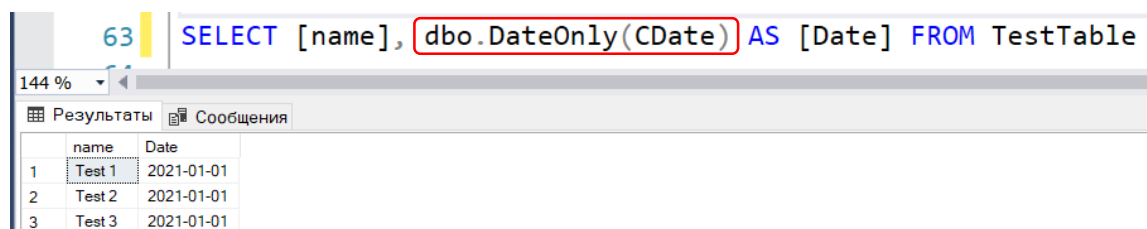
```
60 SELECT * FROM TestTable -- данная выборка вернет 19 записей(все за сегодняшний день
61 WHERE dbo.DateOnly(CDate) = dbo.DateOnly('01.01.2021'); --нужно явно указывать схему при вызове функции
```

144 %

Результаты Сообщения

	id	name	CDate
1	20	Test 1	2021-01-01 10:01:00
2	21	Test 2	2021-01-01 10:02:00
3	22	Test 3	2021-01-01 10:03:00
4	23	Test 4	2021-01-01 10:04:00
5	24	Test 5	2021-01-01 10:05:00

Скалярную пользовательскую функцию можно использовать как в условной части запроса (WHERE), так и в поисковой (SELECT).



САМОСТОЯТЕЛЬНО:

Обратите внимание, что наименования столбцов должны быть на английском языке в стиле CamelCase. Пользовательские функции должны быть именованы по шаблону: UDF_Name (например, UDP_ServiceList)

1. Создайте пользовательскую функцию, которая в качестве входных параметров будет принимать дату начала предоставления услуги и продолжительность услуги. Возвращать – дату окончания предоставления услуги.
2. Создайте пользовательскую функцию, которая будет возвращать таблицу, содержащую следующую информацию: Название услуги, ФИО и номер телефона клиента (шаблон: Иванов Иван Иванович / тел. +7(777)777-77-77), дата начала предоставления услуги, дата окончания предоставления услуги (рассчитать при помощи вызова ранее созданной пользовательской функции), стоимость услуги. В качестве входных параметров функция должна принимать диапазон дат (например, за период времени от 01.01.2019 до 01.06.2019)
3. Создайте пользовательскую функцию, которая будет возвращать среднюю стоимость товара.
4. Создайте пользовательскую функцию, которая будет возвращать таблицу, содержащую следующую информацию: Наименование товара, Наименование производителя, Цена товара, Отклонение от средней стоимости товара (рассчитать при помощи вызова ранее созданной пользовательской функции). В результирующий набор данных следует выводить только актуальные товары.

Пример:

	Title	Name	Cost	DeviationOfAVGCost
1	Super Minds. Level 4. Workbook with Online Resources	Cambridge	1210.00	6.61
2	Super Minds. Workbook 2 with Online Resources	Cambridge	1200.00	-3.39
3	Eyes Open. Level 3. Student's Book	Cambridge	1350.00	146.61
4	Advanced Grammar in Use. Book without Answers	Cambridge	1850.00	646.61
5	Английский на каждый день	ACT	737.00	-466.39

Задание на дополнительную оценку (закрепляем пройденное)

Создайте столбец PersonalDiscount типа данных decimal(3,2) в таблице Client. Данное поле нам понадобится для формирования персональной скидки клиента в зависимости от общей суммы от оказанных услуг данному клиенту. Присвойте данному столбцу значение по умолчанию = 0. После, создайте хранимую процедуру, которая в зависимости от общей суммы от оказанных услуг клиенту будет обновлять столбец PersonalDiscount по следующим условиям: если сумма в размере от 0 до 5000 рублей включительно – скидка 5% (0,05), от 5000 до 10000 включительно – скидка 10% (0,1), свыше 10000 – скидка 15% (0,15).