

Функции, определенные пользователем (UDF)

Функции, определенные пользователем — подпрограммы, которые принимают параметры, выполняют действие, например, сложные вычисления, и возвращают результат этого действия в виде значения. Возвращаемое значение может быть либо единичным скалярным значением, либо результирующим набором (таблицей).

В языках программирования обычно существуют два типа подпрограмм:

- хранимые процедуры
- функции, определенные пользователем (**User Defined Functions, UDF**)

Хранимые процедуры содержат несколько операторов, могут иметь ноль или несколько входных параметров, но обычно не возвращают выходных параметров. В противоположность этому функции всегда имеют одно возвращаемое значение. Однако в функциях нельзя использовать временные таблицы и операции изменения данных в физических таблицах. Вместо временных таблиц, в функциях используются табличные переменные.

Функции, определенные пользователем, могут быть скалярными или табличными. Скалярная функция возвращает скалярное значение (число). Это означает, что в предложении RETURNS скалярной функции вы задаете один из стандартных типов данных. Функции являются табличными, если предложение RETURNS возвращает набор строк.

Определяемые пользователем функции предоставляют следующие возможности:

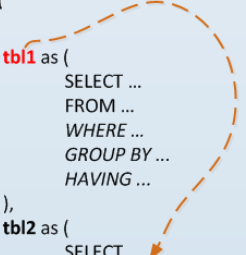
- Делают возможным использование модульного программирования;
- Позволяют ускорить выполнение;
- Позволяют уменьшить сетевой трафик.

Виды табличных функций

- **Табличная функция** — функция, в которой задается структура результирующей таблицы. В теле функции выполняется запрос, который наполняет эту результирующую таблицу данными. После чего, возвращается эта таблица.

- **Inline-табличная функция** — данный вид функции содержит комбинированный запрос (SELECT), который возвращает результирующий набор данных (таблицу). Обычно строится в формате WITH ... SELECT ...

- **Скалярная функция** — Функция, которая возвращает 1 значение. Значение может иметь различный формат данных.

Структура табличной функции	Структура Inline табличной функции	Структура скалярной функции
<pre> CREATE FUNCTION <Имя> (<Параметры>) RETURNS @tableVariable TABLE (<Список полей с типом данных>) AS BEGIN Блок объявлений (Declare statements) Код процедуры RETURN; END; GO </pre>	<pre> CREATE FUNCTION <Имя> (<Параметры>) RETURNS TABLE AS RETURN (WITH tbl1 as (SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ...), tbl2 as (SELECT ... FROM tbl1 WHERE ...), ... tblN as (SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ...) SELECT ... FROM tbl1, tbl2, ... , tblN); GO </pre> 	<pre> CREATE FUNCTION <Имя> (<Параметры>) RETURNS INT AS BEGIN Блок объявлений (Declare statements) Код процедуры RETURN <VALUE>; END; GO </pre>

Пример. Создание пользовательской функции:

```

CREATE FUNCTION udf_Product(@num1 INT, @num2 INT)
RETURNS INT AS
BEGIN
    DECLARE @Product INT;
    SET @Product = ISNULL(@num1,0) * ISNULL(@num2,0);
    RETURN @Product;
END;

```

Пример. Вызов пользовательской функции:

```

SELECT udf_Product(2,10)

```