**Функция** – это объект БД, представляющий собой поименованный код T-SQL. Для создания, удаления и изменения функций надо использовать операторы CREATE, DROP и ALTER соответственно.

Отличие функций от хранимых процедур в ограничениях, накладываемых на код функции, в форме представления результата работы, а также в способе вызова. В функции не допускается применение DDL-операторов, DML-операторов, изменяющих БД (INSERT, DELETE, UPDATE), конструкций TRY/CATCH, а также использование транзакций.

Результатом выполнения функции является возвращаемое к точке вызова значение. Если функция возвращает единственное значение (число, строка, дата, время и пр.), то она называется *скалярной*. Функция, возвращающая таблицу, называется *табличной*. В зависимости от структуры кода, различают *встроенные* функции и *многооператорные* табличные функции.

**\*\*Функция подсчета количества товаров, заказанных фирмой:\*\***

```sql

CREATE FUNCTION COUNT\_Zakazy

(@firmName NVARCHAR(50))

RETURNS INT

AS

BEGIN

DECLARE @count INT;

SELECT @count = COUNT(\*)

FROM Заказы Z

JOIN Заказчики C ON Z.Заказчик = C.Наименование\_фирмы

WHERE C.Наименование\_фирмы = @firmName;

RETURN @count;

END;

```

**\*\*Пример использования функции:\*\***

```sql

SELECT Наименование\_фирмы, dbo.COUNT\_Zakazy(Наименование\_фирмы) AS 'Количество заказов'

FROM Заказчики;

```

**\*\*Скалярная функция FZakazy:\*\***

```sql

CREATE FUNCTION FZakazy

(@tz CHAR(20))

RETURNS VARCHAR(300)

AS

BEGIN

DECLARE @tv CHAR(20);

DECLARE @t VARCHAR(300) = 'Заказанные товары: ';

DECLARE ZkTovar CURSOR LOCAL FOR

SELECT Наименование\_товара FROM Заказы WHERE Заказчик = @tz;

OPEN ZkTovar;

FETCH ZkTovar INTO @tv;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @t = @t + ', ' + RTRIM(@tv);

FETCH ZkTovar INTO @tv;

END;

CLOSE ZkTovar;

RETURN @t;

END;

```

**\*\*Пример использования скалярной функции FZakazy:\*\***

```sql

SELECT Наименование\_фирмы, dbo.FZakazy(Наименование\_фирмы) AS 'Заказанные товары'

FROM Заказчики;

```

**\*\*Табличная функция FTovCena:\*\***

```sql

CREATE FUNCTION FTovCena

(@tovar NVARCHAR(50), @price REAL)

RETURNS TABLE

AS

RETURN

(

SELECT

Товары.Наименование AS 'Товар',

Товары.Себестоимость AS 'Исходная цена',

Заказы.Цена\_продажи AS 'Цена продажи'

FROM

Товары

LEFT JOIN Заказы ON Товары.Наименование = Заказы.Наименование\_товара

WHERE

(@tovar IS NULL OR Товары.Наименование = @tovar)

AND (@price IS NULL OR Заказы.Цена\_продажи = @price)

);

```

**\*\*Примеры использования табличной функции FTovCena:\*\***

```sql

-- Все товары и цены

SELECT \* FROM dbo.FTovCena(NULL, NULL);

-- Товары с именем 'Стул' и все цены

SELECT \* FROM dbo.FTovCena('Стул', NULL);

-- Все товары и цены продажи, которые равны 400

SELECT \* FROM dbo.FTovCena(NULL, 400);

-- Товары с именем 'Шкаф' и цены продажи, которые равны 340

SELECT \* FROM dbo.FTovCena('Шкаф', 340);

```

**\*\*Скалярная функция FKolTov:\*\***

```sql

CREATE FUNCTION FKolTov

(@orderer CHAR(20))

RETURNS TABLE

AS

RETURN

(

SELECT

Заказы.Наименование\_товара AS 'Товар',

COUNT(Заказы.Наименование\_товара) AS 'Количество заказов'

FROM

Заказы

WHERE

Заказы.Заказчик = @orderer

GROUP BY

Заказы.Наименование\_товара

);

```

**\*\*Примеры использования скалярной функции FKolTov:\*\***

```sql

-- Общее количество заказов для всех товаров

SELECT \* FROM dbo.FKolTov(NULL);

-- Количество заказов для конкретного заказчика

SELECT \* FROM dbo.FKolTov('Луч');

```