



доц. д-р Цветанка Георгиева-Трифенова

# ДЕФИНИРАНИ ОТ ПОТРЕБИТЕЛЯ ФУНКЦИИ



# СЪДЪРЖАНИЕ

---

- ✗ Скаларни функции
- ✗ Функции, връщащи таблица
- ✗ Многоструктурни функции, връщащи таблица

# ДЕФИНИРАНИ ОТ ПОТРЕБИТЕЛЯ ФУНКЦИИ

- ✖ Типове дефинирани от потребителя функции в зависимост от типа на върнатата от тях стойност:
  - + скаларни функции (*scalar functions*);
  - + функции, връщащи таблица (*inline table-valued functions*);
  - + многоструктурни функции, връщащи таблица (*multistatement table-valued functions*).



# СКАЛАРНИ ФУНКЦИИ

- ✗ Връщат една стойност от:
  - + скаларен тип данни (например *integer*, *varchar*, *char*, *money*, *datetime*, *bit* и т.н.);
  - + дефиниран от потребителя тип данни, ако е базиран на скаларен тип данни.

# СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ

```
CREATE FUNCTION [owner_name.]function_name
( [ { @parameter_name [AS]
  scalar_parameter_data_type
    [ = default ] } [ , ... n ] ] )
RETURNS scalar_return_data_type
[ WITH [ENCRYPTION] [[,] SCHEMABINDING] ]
[AS]
BEGIN
    function_body
    RETURN scalar_expression
END
```

# СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ (2)

- ✗ ENCRYPTION – задаване на криптиране на колоната в системната таблица, съдържаща текста на конструкциите, включени в дефиницията на функцията;
- ✗ SCHEMABINDING – задаване на свързване с обектите на базата от данни, към които функцията прави обръщение, т.е. те не могат да бъдат променяни или изтривани;
- ✗ изпълняване на скаларните функции:  
*[database\_name.] owner\_name.function\_name([argument \_expr] [, ...])*
- ✗ извикването се извършва при съставяне на скаларни изрази, включително за дефиниране на изчислими колони и CHECK ограничения.



# СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ – ПРИМЕР

```
CREATE FUNCTION fnNeedToReorder
( @nReorderLevel int, @nUnitsInStock int,
  @bDiscontinued bit )
  RETURNS varchar(3)
AS
BEGIN
  DECLARE @sReturnValue varchar(3)

  IF @bDiscontinued = 1
    SET @sReturnValue = 'No'
  ELSE
    IF @nUnitsInStock <= @nReorderLevel
      SET @sReturnValue = 'Yes'
    ELSE
      SET @sReturnValue = 'No'
  RETURN @sReturnValue
END
```

## СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ – ПРИМЕР (2)

### ✖ Изпълняване на функцията

```
SELECT ProductID, ProductName,  
       Stock, ReorderLevel,  
       Discontinued,  
       dbo.fnNeedToReorder(ReorderLevel, Stock,  
       Discontinued)  
       AS sNeedToReorder  
FROM Products
```

	ProductID	ProductName	Stock	ReorderLevel	Discontinued	sNeedToReorder
1	1	ябълки	120.000	10.000	0	No
2	2	портокали	90.000	10.000	0	No
3	3	домати	100.000	15.000	0	No
4	4	картофи	55.000	20.000	0	No
5	5	чипс	20.000	5.000	1	No
6	6	царевичен снакс	24.000	5.000	0	No
7	7	макарони	10.000	12.000	0	Yes
8	8	еклери	52.000	5.000	0	No
9	9	шоколад	231.000	10.000	0	No



# СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ – ПРИМЕР (3)

```
CREATE FUNCTION fnFormatTelephoneNumber
```

```
(@sPhone char(10))
```

```
RETURNS varchar(15)
```

```
AS
```

```
BEGIN
```

```
DECLARE @sPhoneFormat varchar(15)
```

```
IF LEN(@sPhone) < 10
```

```
    SET @sPhoneFormat = @sPhone
```

```
ELSE SET @sPhoneFormat =
```

```
    '(' + LEFT(@sPhone, 3) + ')' +
```

```
    SUBSTRING(@sPhone, 4, 3) +
```

```
    '-' + RIGHT(@sPhone, 4)
```

```
RETURN @sPhoneFormat
```

```
END
```

## СЪЗДАВАНЕ НА СКАЛАРНИ ФУНКЦИИ – ПРИМЕР (4)

✗ Резултатът от следното извикване на функцията:

```
SELECT dbo.fnFormatTelephoneNumber  
      ('1234448888') AS FormatTelephoneNumber
```

има вида:

```
FormatTelephoneNumber  
-----  
(123) 444-8888
```

# ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА

- ✗ Дефинира се с една SELECT конструкция, определяща таблицата, която функцията трябва да върне като резултат и не може да съдържа други T-SQL конструкции:

```
CREATE FUNCTION [owner_name.]function_name
    ( [ { @parameter_name [AS]
      scalar_parameter_data_type
        [ = default ] } [ ,... n ] ] )
RETURNS TABLE
[ WITH [ENCRYPTION] [ [, ] SCHEMABINDING] ]
[AS]
RETURN [ ( ] select_statement [ ) ]
```



# ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА (2)

- ✗ Функции, връщащи таблица, могат да бъдат извиквани, като се включи само името на функцията:

```
[database_name.] [owner_name.] function_name  
([argument_expr] [, ...])
```

# ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА – ПРИМЕР

```
CREATE FUNCTION fnGetEmployeesByCity
```

```
(@sCity varchar(30))
```

```
    RETURNS table
```

```
AS
```

```
RETURN
```

```
(    SELECT EmployeeID, FirstName, LastName,
```

```
        Address
```

```
    FROM Employees
```

```
    WHERE City = @sCity    )
```

✗ Функцията се извиква по следния начин:

```
SELECT * FROM fnGetEmployeesByCity('Велико Търново')
```

✗ Резултатът от функция от този вид може да бъде съединяван с други таблици:

```
SELECT *
```

```
FROM fnGetEmployeesByCity('Велико Търново') е
```

```
INNER JOIN Sales s ON e.EmployeeID = s.EmployeeID
```

# ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА – ПРИМЕР (2)

```
CREATE FUNCTION fnGetEmployeesByCity
```

```
(@sCity varchar(30))
```

```
RETURNS table
```

```
AS
```

```
RETURN
```

```
( SELECT EmployeeID, FirstName, LastName,
```

```
Address
```

```
FROM Employees
```

```
WHERE City = @sCity
```

	EmployeeID	FirstName	LastName	Address
1	1	Стоян	Георгиев	ул. Г. Козарев, 4
2	2	Георги	Хростов	ул. Г. Козарев, 12
3	4	Стела	Миланова	ул. Княз Борис, 19
4	5	Атанас	Лазаров	ул. Хр. Ботев, 8

✗ Функцията се извиква по следния начин:

```
SELECT * FROM fnGetEmp1
```

✗ Резултатът от функция от този вид може да бъде съединяван с други

	EmployeeID	FirstName	LastName	Address	SaleID	CustomerID	EmployeeID	SaleDate	TotalForSale	Discount
1	1	Стоян	Георгиев	ул. Г. Козарев, 4	1	1	1	2010-01-30 09:08:00.000	8.75	0.00
2	1	Стоян	Георгиев	ул. Г. Козарев, 4	2	3	1	2010-01-30 10:08:00.000	42.00	0.15
3	4	Стела	Миланова	ул. Княз Борис, 19	4	2	4	2010-01-31 13:08:00.000	10.10	0.00
4	2	Георги	Хростов	ул. Г. Козарев, 12	5	1	2	2009-10-06 00:00:00.000	0.00	0.00
5	2	Георги	Хростов	ул. Г. Козарев, 12	6	1	2	2009-10-06 00:00:00.000	0.00	0.00



# МНОГОСТРУКТУРНИ ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА

- ✗ изрично се дефинира структурата на таблицата, която се връща като резултат;
- ✗ задават се имената на колоните и типа на данните им в RETURNS;
- ✗ могат да се включат множество и разнообразни T-SQL конструкции.

# МНОГОСТРУКТУРНИ ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА (2)

```
CREATE FUNCTION [owner_name.]function_name
    ( [ { @parameter_name [AS]
      scalar_parameter_data_type
        [ = default ] } [,... n ] ] )
RETURNS @return_variable TABLE
    ( { column_definition | table_constraint }
      [,... n ] )
    [ WITH [ENCRYPTION] [ [, ] SCHEMABINDING ] ]
[AS]
BEGIN
    function_body
    RETURN
END
```

# МНОГОСТРУКТУРНИ ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА

## – ПРИМЕР

```
CREATE FUNCTION fnGetEmployeesByCity2
(@sCity varchar(30))
RETURNS @tblMyEmployees table
(
    FirstName varchar(20),
    LastName varchar(40),
    Address varchar(120)
)

AS
BEGIN
    INSERT INTO @tblMyEmployees
        (FirstName, LastName, Address)
    SELECT FirstName, LastName, Address
    FROM Employees
    WHERE City = @sCity
    ORDER BY LastName

RETURN

END
```



# МНОГОСТРУКТУРНИ ФУНКЦИИ, ВРЪЩАЩИ ТАБЛИЦА – ПРИМЕР (2)

- ✗ Функцията може да бъде извикана по следния начин:

```
SELECT * FROM  
    fnGetEmployeesByCity2 ( 'Велико Търново' )
```

	FirstName	LastName	Address
1	Стоян	Георгиев	ул. Г. Козарев, 4
2	Атанас	Лазаров	ул. Хр. Ботев, 8
3	Стела	Миланова	ул. Княз Борис, 19
4	Георги	Хростов	ул. Г. Козарев, 12

```
ALTER FUNCTION fnGetEmployeesByCity2
(@sCity varchar(30))
    RETURNS @tblMyEmployees table
    (
        FirstName varchar(20),
        LastName varchar(40),
        Address varchar(120) )
AS
BEGIN
    INSERT INTO @tblMyEmployees
        (FirstName, LastName, Address)
    SELECT FirstName, LastName, Address
    FROM Employees
    WHERE City = @sCity
    ORDER BY LastName

    IF NOT EXISTS (SELECT * FROM @tblMyEmployees)
        INSERT INTO @tblMyEmployees (Address)
        VALUES ('Не са намерени служители в
                избрания град.')

    RETURN
END
```

# ЗАДАЧИ

- ✗ 1. Да се напише скаларна функция, която да връща максималното продадено количество от продукт с определено наименование. Ако няма продажби от продукта с подаденото наименование, функцията да връща 0.
- ✗ 2. Да се напише функция (*inline*), която връща продуктите от дадена категория.
- ✗ 3. Да се напише функция, връщаща името на продукт, продадено количество, доставчика на първите 5 най-продавани продукта през определена година.
- ✗ 4. Да се напише многоструктурна функция, която по даден идентификатор на служител връща таблица, съдържаща всички служители, пряко или непряко подчинени на дадения служител.





Цветанка Георгиева-Трифорова, 2017

Някои права запазени.

Презентацията е достъпна под лиценз Creative Commons,

Признание-Некомерсиално-Без производни,

<https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>