



Universidade Federal de Sergipe
Departamento de Sistemas de Informação
Banco de Dados II

Funções definidas pelo Usuário

André Vinicius R. P. Nascimento
andreviniciusnascimento@gmail.com



Conteúdo

- Introdução
- Classificação
- Funções Escalares
- Funções do tipo Table-Valued
- Funções In-Line
- Verificando Informações



Introdução

- A partir da versão 2000 do SqlServer, a linguagem Transact-SQL trouxe a possibilidade de criação de funções definidas pelo usuário.
- Antes da versão 2000, as únicas funções possíveis eram as predefinidas.



Introdução

- Uma função definida pelo usuário no Sql Server é semelhante às funções existentes em outras linguagens de programação.
- São rotinas que aceitam parâmetros, executam uma ação e retornam o resultado da ação como um valor.



Classificação

- A linguagem Transact SQL possui os seguintes tipos de funções definidas pelo usuário:
 - Scalar (Funções Escalares)
 - Table-Valued (Funções que retornam tabelas)
 - In-Line (Funções em Linha)



Funções Escalares

- São funções que retornam um único valor do tipo definido como retorno da função.
- Sintaxe:

```
CREATE FUNCTION <ESQUEMA>.<NOME_FUNCAO>  
    (<PARAMETROS>)  
RETURNS <TIPO DO RETORNO>  
AS  
BEGIN  
    <CORPO DA FUNÇÃO>  
    RETURN <RETORNO>  
END
```



Funções Escalares

- Exemplo:

```
CREATE FUNCTION SP_TRIM(@STR VARCHAR(200))  
RETURNS VARCHAR(200)  
AS  
BEGIN  
    RETURN LTRIM(RTRIM(@STR))  
END
```



Funções Escalares

- Os parênteses na declaração são obrigatórios mesmo que a função não apresente parâmetros:

```
CREATE FUNCTION SP_FATOR_DE_AJUSTE()  
RETURNS NUMERIC(10,2)  
AS  
BEGIN  
    RETURN 12.56  
END
```




Funções Escalares

- Executando uma Função:

```
SELECT dbo.SP_TRIM(' JOSE  ')
```

```
SELECT MATRICULA, dbo.SP_TRIM(NOME)  
FROM TB_CLIENTE
```

```
UPDATE TB_CLIENTE  
SET NOME = dbo.SP_TRIM(NOME)
```



Funções Escalares

- Uma função também pode ser executada utilizando o Comando EXEC.
- Nesses casos, a passagem de parâmetros é semelhante à passagem em procedimentos.

```
DECLARE @NOME VARCHAR(200)
```

```
EXEC @NOME = dbo.SP_TRIM ' JOSE      '
```

```
PRINT @NOME
```



Funções Escalares

- Alterando uma Função:

```
ALTER FUNCTION SP_TRIM(@STR VARCHAR(200))  
RETURNS VARCHAR(200)  
AS  
BEGIN  
    SET @STR = REPLACE(@STR, '-', ' ' )  
    RETURN LTRIM(RTRIM(@STR))  
END
```



Funções Escalares

- Removendo uma Função:

```
DROP FUNCTION <NOME DA FUNÇÃO>
```

```
DROP FUNCTION SP_TRIM
```



Funções Escalares

- Observações:
 - Uma função não pode apresentar parâmetros de saída.
 - Uma função não pode modificar o estado de um banco de dados.



Funções Escalares

- A seguinte função não pode ser criada:

```
CREATE FUNCTION SP_TRIM2 (@STR VARCHAR(200) , @STR1  
    VARCHAR(200) OUTPUT)  
RETURNS VARCHAR(200)  
AS  
BEGIN  
    RETURN LTRIM(RTRIM(@STR))  
END
```



Funções Escalares

- A seguinte função não pode ser criada:

```
CREATE FUNCTION SP_ALTERA_SALARIO()  
RETURNS VARCHAR(10)  
AS  
BEGIN  
    UPDATE TB_CLIENTE  
    SET SALARIO = SALARIO *1.2  
    RETURN 'OK'  
END
```



Funções (Table-Valued)

- São funções cujo retorno é do tipo tabela
- Sintaxe:

```
CREATE FUNCTION <ESQUEMA>.<NOME_FUNCAO>
    (<PARAMETROS>)
RETURNS @VARIABEL_RETORNO TABLE <DEFINIÇÃO DA
TABELA>
AS
BEGIN
    <CORPO DA FUNÇÃO>
    RETURN
END
```




Funções (Table-Valued)

```
CREATE FUNCTION dbo.SP_FUNCIONARIOS_SALARIO  
    (@SALARIO NUMERIC(10,2))  
RETURNS @FUNCIONARIOS TABLE  
    (MATRICULA INT NOT NULL PRIMARY KEY,  
     NOME VARCHAR(40) NOT NULL,  
     SALARIO NUMERIC(10,2)  
    )  
AS  
BEGIN  
    INSERT INTO @FUNCIONARIOS  
    SELECT MATRICULA, NM_FUNCIONARIO, SALARIO  
    FROM TB_FUNCIONARIO  
    WHERE SALARIO >= @SALARIO AND  
           SALARIO <= @SALARIO * 1.4  
    RETURN  
END
```



Funções (Table-Valued)

- Observações:
 - As função do tipo Table-Valued podem apresentar qualquer número de comandos. Por esse motivo, esse tipo de função também é conhecido como Multi-Statement Table-Valued Functions.
 - O comando RETURN não pode apresentar parâmetros.



Funções (In-Line)

- Funções In-Line representam um tipo especial de funções Table-Valued.
- Seu principal objetivo é implementar um conceito conhecido como Visões Parametrizadas.
- As funções In-Line não possuem corpo, apenas um comando sql.



Funções (In-Line)

- Sintaxe:

```
CREATE FUNCTION <ESQUEMA>.<NOME_FUNCAO>  
    (<PARAMETROS>)  
RETURNS TABLE  
AS  
RETURN  
(  
    <COMANDO SELECT>  
)
```



Funções (In-Line)

- Exemplo:

```
CREATE FUNCTION dbo.SP_FUNCIONARIOS_SEXO (@SEXO
    VARCHAR(1))
RETURNS TABLE
AS
RETURN
(
    SELECT MATRICULA, NM_FUNCIONARIO, SALARIO
    FROM TB_FUNCIONARIO
    WHERE SEXO = @SEXO
)
```



Verificando Informações

- As seguintes stored procedures podem ser utilizadas para obter informações sobre funções:

```
sp_help 'dbo.SP_TRIM'
```

```
sp_help 'dbo.SP_FUNCIONARIOS_SALARIO'
```

```
sp_helptext 'dbo.SP_FUNCIONARIOS_SEXO'
```

```
sp_stored_procedures 'SP_TRIM'
```