

Function

(Função)

Function

- Uma função contém um bloco de comandos SQL responsável por executar um procedimento e retornar um valor ou uma série de valores.
- As funções possuem semelhanças com as stored procedures, já que ambas consistem em uma maneira simplificada de realizar consultas complexas.
- Ao utilizar funções, devemos considerar que o retorno obtido de uma função pode ser um valor único ou dados de uma tabela.

Create Function

Objetivo: Criar uma função

Sintaxe

CREATE FUNCTION <nome_função>

([<@parâmetro><tipo>[=default]], [<@parâmetro><tipo>[=default]])

RETURNS <tipo>

[**AS**]

BEGIN

<comandos_SQL>

RETURN(<valor>)

END

Em que:

Nome_função: nome da função a ser criada

Parâmetro: parâmetro a ser declarado na função. Uma função pode ter até 2100 parâmetros. Deve começar com @.

Tipo: tipo de dados do parâmetro.

Default: é um valor padrão para o parâmetro. Se um valor default for definido, a função pode ser executada sem especificar um valor para esse parâmetro.

Comandos_SQL: Comandos SQL que serão executados na função.

Valor: valor a ser retornado pela função.

Exemplo:

```
Create Function fc_RetornaAluno(@Aluno smallint)
```

```
RETURNS Nchar(40)
```

```
AS
```

```
BEGIN
```

```
DECLARE @Retorno Nchar(40) SET @Retorno=(SELECT nomeAluno + ' *** ' + rgAluno FROM tbAluno WHERE  
codAluno=@Aluno)
```

```
RETURN @Retorno
```

```
END
```

Função que retorna nome e rg do aluno concatenados

Para invocar a função,
execute:

```
SELECT codAluno  
      ,nomeAluno  
      ,MinhaFuncao=dbo.fc_RetornaAluno(codAluno)  
FROM tbAluno
```

Ou

```
select MinhaFuncao=dbo.fc_RetornaAluno(1)
```

Resultados		Mensagens
MinhaFuncao		
1	Paulo Santos *** 82.292.122-0	

Resultados		Mensagens	
	codAluno	nomeAluno	MinhaFuncao
1	1	Paulo Santos	Paulo Santos *** 82.292.122-0
2	2	Maria da Gloria	Maria da Gloria *** 45.233.123-0
3	3	Perla Nogueira Silva	Perla Nogueira Silva *** 23.533.211-9
4	4	Gilson Barros Silva	Gilson Barros Silva *** 34.221.111-x
5	5	Mariana Barbosa Santos	Mariana Barbosa Santos *** 54.222.122-9
6	6	ERNESTO	ERNESTO *** 2
7	7	Felipe	Felipe *** 2
8	8	Cristiane	Cristiane *** 2
9	9	Pietro	Pietro *** 2
10	10	Breno	Breno *** 286395459
11	11	Marlene	Marlene *** 82341239

Alter Function

Objetivo: Alterar uma função

Sintaxe

ALTER FUNCTION <nome_função>

([<@parâmetro><tipo>[=default]], [<@parâmetro><tipo>[=default]])

RETURNS <tipo>

[**AS**]

BEGIN

<comandos_SQL>

RETURN(<valor>)

END

Drop Function

Objetivo: Remover uma função

Sintaxe

DROP FUNCTION <nome_função>

Exemplo:

Drop Function fc_RetornaAluno

Function Exemplo: Informada uma data como parâmetro, informa o dia da semana:

```
create function  
fc_diaSemana(@Data date)  
    returns varchar(40) as  
begin
```

```
Declare @DiaSemana varchar  
(40)
```

```
Declare @Dia int
```

```
set @Dia= DatePart(dw,@Data)
```

```
if @Dia=1
```

```
begin
```

```
set @DiaSemana ='DOMINGO'
```

```
end
```

```
if @Dia=2
```

```
begin
```

```
Set @DiaSemana ='SEGUNDA-FEIRA'
```

```
end
```

```
if @Dia=3
```

```
begin
```

```
set @DiaSemana ='TERÇA-FEIRA'
```

```
end
```

```
if @Dia=4
```

```
Set @DiaSemana ='QUARTA-FEIRA'
```

```
if @Dia=5
```

```
begin
```

```
set @DiaSemana ='QUINTA-FEIRA'
```

```
end
```

```
if @Dia=6
```

```
begin
```

```
set @DiaSemana ='SEXTA-FEIRA'
```

```
end
```

```
if @Dia=7
```

```
begin
```

```
set @DiaSemana ='SÁBADO'
```

```
end
```

```
RETURN @DiaSemana
```

```
END
```

```
drop function dbo.fc_diaSemana
```

```
select datamatricula as 'Data da  
matricula',
```

```
DiaDaSemana=dbo.fc_diaSemana(dataMa  
tricula)
```

```
from tbMatricula
```

Exemplo de function pra validar cpf

```
select bdEstoque.dbo.CPF_VALIDO(cpfCliente) from tbCliente
```

```
use bdEstoque
```

```
go
```

```
CREATE FUNCTION CPF_VALIDO(@CPF VARCHAR(11))
```

```
RETURNS CHAR(1)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @INDICE INT,
```

```
            @SOMA INT,
```

```
            @DIG1 INT,
```

```
            @DIG2 INT,
```

```
            @CPF_TEMP VARCHAR(11),
```

```
            @DIGITOS_IGUAIS CHAR(1),
```

```
            @RESULTADO CHAR(1)
```

```
SET @RESULTADO = 'N'
```

```
/*
Verificando se os digitos são iguais
A Principio CPF com todos o números iguais são Inválidos
apesar de validar o Calculo do digito verificado
EX: O CPF 00000000000 é inválido, mas pelo calculo
Validaria
*/

SET @CPF_TEMP = SUBSTRING(@CPF,1,1)

SET @INDICE = 1
SET @DIGITOS_IGUAIS = 'S'

WHILE (@INDICE <= 11)
BEGIN
    IF SUBSTRING(@CPF,@INDICE,1) <> @CPF_TEMP
        SET @DIGITOS_IGUAIS = 'N'
    SET @INDICE = @INDICE + 1
END;

--Caso os digitos não sejam todos iguais Começo o calculo do digitos
IF @DIGITOS_IGUAIS = 'N'
BEGIN
    --Cálculo do 1º dígito
    SET @SOMA = 0
    SET @INDICE = 1
    WHILE (@INDICE <= 9)
    BEGIN
        SET @Soma = @Soma + CONVERT(INT,SUBSTRING(@CPF,@INDICE,1)) * (11 - @INDICE);
        SET @INDICE = @INDICE + 1
    END

    SET @DIG1 = 11 - (@SOMA % 11)

    IF @DIG1 > 9
        SET @DIG1 = 0;

    -- Cálculo do 2º dígito }
    SET @SOMA = 0
    SET @INDICE = 1
    WHILE (@INDICE <= 10)
    BEGIN
        SET @Soma = @Soma + CONVERT(INT,SUBSTRING(@CPF,@INDICE,1)) * (12 - @INDICE);
        SET @INDICE = @INDICE + 1
    END

    SET @DIG2 = 11 - (@SOMA % 11)

    IF @DIG2 > 9
        SET @DIG2 = 0;
```

```
-- Validando
IF (@DIG1 = SUBSTRING(@CPF,LEN(@CPF)-1,1)) AND (@DIG2 =
SUBSTRING(@CPF,LEN(@CPF),1))
    SET @RESULTADO = 'S'
ELSE
    SET @RESULTADO = 'N'
END
RETURN @RESULTADO
END
```

Exercícios usando bdEscola

1. Crie uma função que informada uma data da matrícula , retorne o dia da semana.
2. Crie uma função que de acordo com a carga horária do curso exiba curso rápido ou curso extenso. (Rápido menos de 1000 horas).
3. Crie uma função que de acordo com o valor do curso exiba curso caro ou curso barato. (Curso caro acima de 400).
4. Criar uma função que informada a data da matrícula converta-a no formato dd/mm/aaaa.

bdEstoque

- Criar uma função que retorne o dia de semana da venda (no formato segunda, terça, etc) ao lado do código da venda, valor total da venda e sua data
- Criar uma função que receba o código do cliente e retorne o total de vendas que o cliente já realizou
- Criar uma função que receba o código de um vendedor e o mês e informe o total de vendas do vendedor no mês informado
- Criar uma função que usando o bdEstoque diga se o cpf do cliente é ou não válido