**CURSO T-SQL – ALURA:**

**SQL Server: T-SQL com Microsoft SQL Server 2017**

**<https://cursos.alura.com.br/course/tsql-com-sql-server-2017/task/41175>**

**04. STORED PROCEDURES**

Nesta aula, aprendemos:

* A manipular *Stored Procedures*;
* As SPs de cálculos e escalares;
* Que existe SPs do sistema que podem ser usadas;
* SPs com entradas de dados escalares e tabelas;
* Como obter uma variável vinda de uma SP;
* A alterar e excluir SPs.

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Crie uma nova consulta e inclua o código abaixo:

CREATE PROCEDURE BuscaPorEntidades @ENTIDADE AS VARCHAR(10)

AS

BEGIN

IF @ENTIDADE = 'CLIENTES'

SELECT [CPF] AS IDENTIFICADOR, [NOME] AS DESCRITOR,

[BAIRRO] AS BAIRRO FROM [TABELA DE CLIENTES]

ELSE IF @ENTIDADE = 'VENDEDORES'

SELECT [MATRICULA] AS IDENTIFICADOR, [NOME] AS DESCRITOR,

[BAIRRO] AS BAIRRO FROM [TABELA DE VENDEDORES]

END

Esta SP irá selecionar os clientes ou os vendedores, dependendo do parâmetro passado para ela.

2) Se você executar:

EXEC BuscaPorEntidades @ENTIDADE = 'CLIENTES'

Você irá visualizar a lista de clientes.

3) Se você executar:

EXEC BuscaPorEntidades @ENTIDADE = 'VENDEDORES'

Você verá a lista de vendedores.

4) Crie uma nova consulta e inclua o código abaixo:

CREATE PROCEDURE CalculaIdade

AS

BEGIN

UPDATE [TABELA DE CLIENTES] SET IDADE =

DATEDIFF(YEAR, [DATA DE NASCIMENTO], GETDATE())

END

Esta rotina irá calcular a idade de todos os clientes baseado na data de nascimento e no dia de hoje retornado pelo computador.

5) Insira um novo cliente:

INSERT INTO [TABELA DE CLIENTES] (

CPF, [NOME], [ENDERECO 1], BAIRRO, CIDADE,

ESTADO, CEP, [DATA DE NASCIMENTO], SEXO,

[LIMITE DE CREDITO], [VOLUME DE COMPRA], [PRIMEIRA COMPRA]

) VALUES (

'123123123', 'JOÃO MACHADO', 'RUA PROJETADA A',

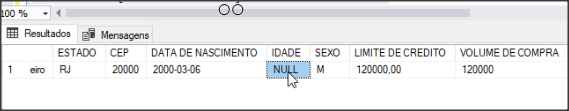
'MADUREIRA', 'Rio de Janeiro', 'RJ', '20000',

'20000306', 'M', 120000, 120000, 120000

)

6) Se você observar os dados deste novo cliente na tabela, terá:

SELECT \* FROM [TABELA DE CLIENTES] WHERE CPF = '123123123'



7) Agora, execute a SP:

EXEC CalculaIdade

8) Executando a consulta novamente, você verá que a idade foi atualizada:

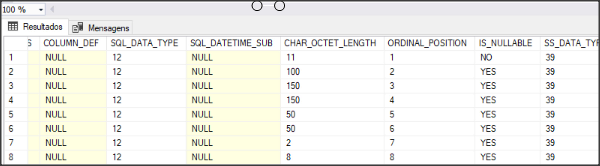


9) Crie uma nova consulta e inclua o código abaixo:

EXEC sp\_columns @table\_name = 'TABELA DE CLIENTES',

@table\_owner = 'dbo'

Executando, você terá:



Esta SP do sistema mostra as informações dos campos da tabela de clientes.

10) Agora digite e execute o comando abaixo:

EXEC sp\_tables @table\_name = 'TAB%',

@table\_owner = 'dbo',

@table\_qualifier = '\*\*`SUCOS\_VENDAS`\*\*'

Aqui você tem informações sobre as tabelas que começam com **TAB**.

11) Crie uma nova consulta e inclua o código abaixo:

CREATE PROCEDURE BuscaNotasCliente

@CPF AS VARCHAR(12),

@DATA\_INICIAL AS DATETIME = '20160101',

@DATA\_FINAL AS DATETIME = '20161231'

AS

BEGIN

SELECT \* FROM [NOTAS FISCAIS] WHERE CPF = @CPF

AND [DATA] >= @DATA\_INICIAL AND [DATA] <= @DATA\_FINAL

END

Aqui você tem dois parâmetros (**DATA\_INICIAL** e **DATA\_FINAL**) que possuem valores padrões.

12) Se você executar:

EXEC BuscaNotasCliente @CPF = '19290992743'

Os valores das outras duas variáveis que são parâmetros da SP estarão com os valores padrões.

13) Se você executar:

EXEC BuscaNotasCliente @CPF = '19290992743',

@DATA\_INICIAL = '20161201'

Somente a variável **@DATA\_FINAL** será executada com o parâmetro padrão.

14) Se você executar:

EXEC BuscaNotasCliente @CPF = '19290992743',

@DATA\_FINAL = '20160131'

Você terá o valor padrão agora para a **DATA\_INICIAL**.

15) Se você executar:

EXEC BuscaNotasCliente @DATA\_FINAL = '20160131',

@CPF = '19290992743', @DATA\_INICIAL = '20160105'

Todas as variáveis terão valores respeitando as declarações.

16) Você pode omitir o nome da variável na chamada da SP, as variáveis serão usadas na ordem de suas declarações:

EXEC BuscaNotasCliente '19290992743'

As variáveis **DATA\_INICIAL** e **DATA\_FINAL** ficarão com os valores padrões.

17) Usando:

EXEC BuscaNotasCliente '19290992743', '20161201'

Apenas **DATA\_FINAL** ficará com o valor padrão.

18) E finalmente:

EXEC BuscaNotasCliente '19290992743', DEFAULT, '20160131'

Com o valor de **DATA\_INICIAL** sendo o padrão.

19) Crie uma nova consulta e inclua o código abaixo:

DECLARE @Lista AS ListaClientes

INSERT INTO @Lista (CPF)

VALUES ('8502682733'), ('8719655770'), ('9283760794')

Acima foi criada uma variável do tipo tabela, com uma lista de código de clientes.

20) Com esta variável, você pode obter o faturamento destes clientes:

SELECT A.[CPF], A.[NOME],

SUM(C.[QUANTIDADE] \* C.[PREÇO]) AS FATURAMENTO

FROM [TABELA DE CLIENTES] A INNER JOIN [NOTAS FISCAIS] B

ON A.CPF = B.CPF AND YEAR(B.[DATA]) = 2016

INNER JOIN [ITENS NOTAS FISCAIS] C ON B.NUMERO = C.NUMERO

INNER JOIN @Lista D ON A.CPF = D.CPF

GROUP BY A.[CPF], A.[NOME]

21) Agora você pode passar, como parâmetro, a tabela com a lista de CPFs para uma SP que irá calcular o faturamento:

CREATE PROCEDURE FaturamentoClientes2016

@LISTA AS ListaClientes READONLY

AS

SELECT A.[CPF], A.[NOME],

SUM(C.[QUANTIDADE] \* C.[PREÇO]) AS FATURAMENTO

FROM [TABELA DE CLIENTES] A INNER JOIN [NOTAS FISCAIS] B

ON A.CPF = B.CPF AND YEAR(B.[DATA]) = 2016

INNER JOIN [ITENS NOTAS FISCAIS] C ON B.NUMERO = C.NUMERO

INNER JOIN @LISTA D ON A.CPF = D.CPF

GROUP BY A.[CPF], A.[NOME]

22) Usando agora esta SP para obter o faturamento, você terá:

DECLARE @Lista AS ListaClientes

INSERT INTO @Lista (CPF)

VALUES ('5840119709'), ('7771579779'), ('9283760794')

EXEC FaturamentoClientes2016 @LISTA = @Lista

23) Crie uma nova consulta e inclua o código abaixo:

CREATE PROCEDURE RetornaValores

@CPF AS VARCHAR(12),

@ANO AS INT,

@NUM\_NOTAS AS INT OUTPUT,

@FATURAMENTO AS FLOAT OUTPUT

AS

BEGIN

SELECT @NUM\_NOTAS = COUNT(\*)

FROM [NOTAS FISCAIS] WHERE CPF = @CPF

AND YEAR([DATA]) = @ANO

SELECT @FATURAMENTO = SUM(QUANTIDADE \* [PREÇO])

FROM [ITENS NOTAS FISCAIS] B

INNER JOIN [NOTAS FISCAIS] A ON A.NUMERO = B.NUMERO

WHERE CPF = @CPF

AND YEAR([DATA]) = @ANO

END

Neste código, os parâmetros **@CPF** e **@ANO** são passados como valor e **@NUM\_NOTAS** e **@FATURAMENTO** como referência.

24) Logo, você pode visualizar pelo script que chama a SP, os valores modificados nas variáveis passadas como referência:

DECLARE @NUMERO\_NOTAS INT, @FATURAMENTO FLOAT

DECLARE @CPF VARCHAR(12), @ANO INT

SET @CPF = '19290992743'

SET @ANO = 2017

EXEC RetornaValores @CPF = @CPF,

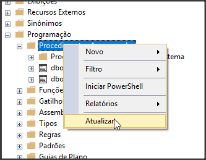
@ANO = @ANO,

@NUM\_NOTAS = @NUMERO\_NOTAS OUTPUT,

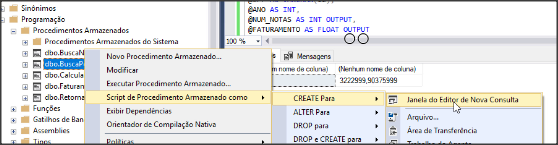
@FATURAMENTO = @FATURAMENTO OUTPUT

SELECT @NUMERO\_NOTAS, @FATURAMENTO

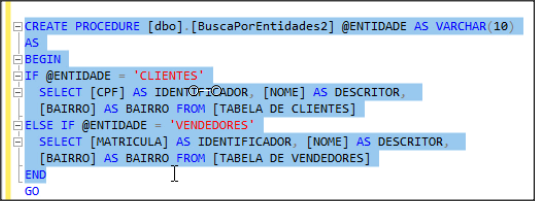
25) Clique com o botão da direita do mouse sobre **Procedimentos Armazenados** e clique em **Atualizar**:



26) Selecione a SP **BuscaPorEntidades** e clique com o botão da direita do mouse sobre ela, e escolha **Script de Procedimento Armazenado como --> CREATE Para --> Janela do Editor da Nova Consulta**:



27) Mude o nome da SP para **BuscaPorEntidades2**. Execute para criação de uma nova SP:



28) Para executá-la, basta digitar:

EXEC BuscaPorEntidades2 'CLIENTES'

29) O comando para alterar a SP é mostrado abaixo:

ALTER PROCEDURE [dbo].[BuscaPorEntidades2] @ENTIDADE AS VARCHAR(10)

AS

BEGIN

IF @ENTIDADE = 'CLIENTES'

SELECT [CPF] AS IDENTIFICADOR, [NOME] AS DESCRITOR

FROM [TABELA DE CLIENTES]

ELSE IF @ENTIDADE = 'VENDEDORES'

SELECT [MATRICULA] AS IDENTIFICADOR, [NOME] AS DESCRITOR

FROM [TABELA DE VENDEDORES]

END

30) Se você executar novamente a chamada, verá outro resultado.

31) Você pode também apagar a SP através do comando **DROP**. Para executar o **DROP**, execute-o desde a SP exista:

IF OBJECT\_ID('BuscaPorEntidades2', 'P') IS NOT NULL

DROP PROCEDURE BuscaPorEntidades2