Microsoft SQL Server 2016 (T-SQL)

Aula 16

Professor Anderson Henrique

Assuntos tratados nessa aula:

01 – Stored Procedures – Criação e Execução

02 – Stored Procedures – Alteração e Parâmetros de Entrada

03 – Stored Procedures – Parâmetros de Entrada e INSERT

04 – Stored Procedures – Parâmetros de Saída e RETURN

Professor Anderson Henrique

01 – Stored Procedures – Criação e Execução

São lotes (batches) de declarações SQL que podem ser executadas como uma subrotina

Permitem centralizar a lógica de acesso aos dados em um único local, facilitando a manutenção e otimização de código

também é possível ajustar permissões de acesso aos usuários, definindo quem pode ou não executá-las

Criar um Procedimento Armazenado

CREATE PROCEDURE nome_procedimento (@Parâmetro Tipo_Dados)
AS
Bloco de códigos

Exemplo 1

```
CREATE PROCEDURE teste
AS
SELECT ‘LogusTI Treinamentos e Cursos’ AS Nome
```

Para executar:

```
EXEC(UTE) teste
```

Obs.: Se o procedimento armazenado for o primeiro comando de um batch, não é necessário usar a palavra EXEC

Exemplo 2

```
CREATE PROCEDURE p_LivroValor
AS
SELECT Nome_Livro, Preco_Livro
FROM tbl_Livro
```

Para executar:

```
EXEC(UTE) p_LivroValor
```

Visualizar conteúdo de um SP

Use o procedimento armazenado **sp_helptext** para extrair o conteúdo de texto de uma stored procedure:

EXEC sp_helptext nome_procedimento

Exemplo:
EXEC sp_helptext p_LivroValor

Criar um Procedimento Armazenado (Criptografado)

```
CREATE PROCEDURE nome_procedimento (@Parâmetro Tipo_Dados)
WITH ENCRYPTION
AS
Bloco de códigos

CREATE PROCEDURE p_LivroISBN
WITH ENCRYPTION
AS
SELECT Nome_Livro, ISBN FROM tbl_Livro
```

Tente visualizar seu conteúdo com sp_helptext

02 – Stored Procedures – Alteração e Parâmetros de Entrada

Para modificar um Stored Procedure podemos utilizar o comando ALTER PROCEDURE

ALTER PROCEDURE nome_procedimento
Bloco de código da sp

Parâmetros de Entrada

```
ALTER PROCEDURE teste (@par1 AS int)
AS
SELECT @par1
```

Executar passando um parâmetro:
EXEC teste 22 (22 é o valor do parâmetro passado)

Outro exemplo com parâmetro de entrada

```
ALTER PROCEDURE p_LivroValor (@ID SMALLINT)
AS
SELECT Nome_Livro As Título, Preco_Livro AS Preço
FROM tbl_Livro
WHERE ID_Livro = @ID
```

Executar passando um parâmetro:

EXEC p_LivroValor 4

Múltiplos Parâmetros de Entrada

```
ALTER PROCEDURE teste (@par1 AS int, @par2 AS varchar(20))
AS
SELECT @par1
SELECT @par2
```

Executar passando um parâmetro:

```
EXEC teste 22, ‘Laranja’ (por posição)
EXEC teste @par1 = 25, @par2 = ‘Abacate’ (por nome)
```

03 – Stored Procedures –Parâmetros de Entrada e INSERT

```
ALTER PROCEDURE p_LivroValor (@ID SMALLINT, @Preco MONEY)
AS
SELECT Nome_Livro AS Título, Preco_Livro AS Preço
FROM tbl_Livro
WHERE ID_Livro > @ID AND Preco_Livro > @Preco
```

Executar:

EXEC p_LivroValor @ID = 3, @Preco = 60

Outro exemplo - parâmetros

Desejo fornecer o ID e a quantidade de um título adquirido, e saber o valor total pago pelos livros

```
ALTER PROCEDURE p_LivroValor(@Quantidade SMALLINT, @ID SMALLINT)
AS
SELECT Nome_Livro AS Título, Preco_Livro * @Quantidade AS Preço
FROM tbl_Livro
WHERE ID_Livro = @ID
```

Executar

```
EXEC p_LivroValor @ID = 3, @Quantidade = 10
```

TECH
soluções em ti

Exemplo – Inserção de Dados

CREATE PROCEDURE p_insere_editora(@nome VARCHAR(50))
AS
INSERT INTO tbl_Editora(Nome_Editora)
VALUES(@nome)

Execução e verificação:
EXEC p_insere_editora @nome = 'Apress' ('Alpha Books', 'Companhia das Letras', 'Ediouro')
SELECT * FROM tbl_Editora

TECH
soluções em ti

04 – Stored Procedures –Parâmetros de Saída e RETURN

Parâmetros com valor padrão

```
CREATE PROCEDURE p_teste_valor_padrão(@param1 INT, @param2 VARCHAR(20) = 'Valor Padrão!')
AS
SELECT 'Valor do parâmetro 1: ' + CAST(@param1 AS VARCHAR)
SELECT 'Valor do parâmetro 2: ' + @param2
```

Execução:

```
EXEC p_teste_valor_padrão 30
EXEC p_teste_valor_padrão @param1 = 40, @param2 = 'Valor Modificado'
```

Parâmetros de saída

Os parâmetros de saída habilitam um procedimento armazenado a retornar dados ao procedimento chamador

Usamos a palavra-chave OUTPUT quando o procedimento é criado, e também quando é chamado

No procedimento armazenado, o procedimento de saída aparece como uma variável local; No procedimento chamador, uma variável deve ser criada para receber o parâmetro de saída

Parâmetros de saída

```
ALTER PROCEDURE teste (@par1 AS INT OUTPUT)
AS
SELECT @par1 * 2
RETURN
```

Executar passando um parâmetro:

```
DECLARE @valor AS INT = 15
EXEC teste @valor OUTPUT
PRINT @valor
```

Comando RETURN

O comando RETURN termina incondicionalmente o procedimento e retorna um valor inteiro ao chamador

Pode ser usado para retornar status de sucesso ou falha de procedimento

```
ALTER PROCEDURE p_LivroValor (@Quantidade SMALLINT,
@Cod SMALLINT = -10, @ID SMALLINT)
AS
-- SET NOCOUNT ON
IF @ID >= 1
BEGIN
SELECT Nome_Livro As Título, Preco_Livro *@Quantidade AS Preço
FROM tbl_Livro
WHERE ID_Livro = @ID
RETURN 1
END
ELSE
RETURN @Cod
```

Executando o procedimento:

```
DECLARE @Codigo INT
EXEC @Codigo = p_LivroValor @ID = 2, @Quantidade = 5
PRINT @Codigo

EXEC @Codigo = p_LivroValor @ID = 15, @Quantidade = 5
PRINT @Codigo
```

Obs.: SET NOCOUNT ON (**Evita retornar a quantidade de linhas afetadas**)

Dúvidas?

Professor Anderson Henrique

Para a próxima aula

01 – Funções – Função Escalar

02 – Funções – Valor de Tabela Embutida

03 – Funções – Valor de Tabela com Várias Instruções

Professor Anderson Henrique