



Stored Procedure

Stored Procedure

Criação e Execução no SQL Server



***Stored Procedure* - Procedimento Armazenado**

Definição

São lotes (*batches*) de declarações SQL que podem ser executados como uma subrotina.

Permitem centralizar a lógica de acesso aos dados em 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 'Fabrício' 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_TituloAno
```

```
AS
```

```
SELECT titulo, ano
```

```
FROM Livro
```

Para executar:

```
EXEC(UTE) p_TituloAno
```



Visualizar conteúdo de SP

Use o procedimento armazenado `sp_helptext` para extrair o conteúdo de texto de um *stored procedure*?

```
EXEC sp_helptext nome_procedimento
```

Ex:

```
EXEC sp_helptext p_LivroValor
```



Criptografar *Stored Procedure*

```
CREATE PROCEDURE p_LivroISBN
```

```
WITH ENCRYPTION
```

```
AS
```

```
SELECT titulo, isbn
```

```
FROM Livro
```

Agora tente visualizar seu
conteúdo com sp_helptext:
Exec sp_helptext p_LivroISBN



Modificar Stored Procedure

ALTER PROCEDURE

ALTER PROCEDURE nome_procedimento

bloco de código da sp



Parâmetro de Entrada

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

```
AS
```

```
SELECT @par1
```

Executar passando um parametro?

EXEC teste 22 – 22 é o valor do parâmetro passado



Exemplo 2 Parâmetro de Entrada acessando banco

```
ALTER PROCEDURE p_TituloAno
```

```
(@ANO INT)
```

```
AS
```

```
SELECT titulo AS 'Livro', ano AS 'Ano Publicação'
```

```
FROM Livro
```

```
WHERE ano = @ANO
```

Executar passando um parâmetro

EXEC teste 22 – 22 é o valor do parâmetro passado



Exemplo 3 Múltiplos parâmetros de entrada

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

```
AS
```

```
BEGIN
```

```
SELECT @par1
```

```
SELECT @par2
```

```
END
```

Executar passando um parâmetro:

```
EXEC teste 22, 'Vermelho' –por posição
```

```
EXEC teste @par1 = 25, @par2 = 'Laranja' – por nome
```



Exemplo 4 Múltiplos parâmetros de entrada

```
ALTER PROCEDURE p_TituloAno (@ANO INT, @TITULO varchar(100))
```

```
AS
```

```
SELECT titulo AS 'Livro', ano AS 'Ano Publicacao'
```

```
FROM Livro
```

```
WHERE ano > @ANO and titulo like %@titulo%
```

Executar passando um parâmetro:

```
EXEC p_TituloAno @ANO= 2000, @TITULO = Potter —por posição
```

```
EXEC teste @par1 = 25, @par2 = 'Laranja' – por nome
```



Exemplo 5 Múltiplos parâmetros de entrada

```
CREATE PROCEDURE p_LivroValor(@Quantidade SMALLINT, @ID  
SMALLINT)
```

```
AS
```

```
SELECT titulo as 'Livro', valor*quantidade AS 'Valor Total'
```

```
FROM Livro
```

```
WHERE id = @ID
```

Executar

```
EXEC p_LivroValor @ID = 5 @Quantidade = 10
```



Exemplo 6 Inserção de dados

```
CREATE PROCEDURE p_inserir_editora (@nome VARCHAR(50))
```

```
AS
```

```
INSERT INTO Editora (nome)
```

```
VALUES (@nome)
```

Execução e verificação

```
EXEC p_inserir_editora @nome = 'Editora Exemplo'
```

```
Select * from editora
```



Exemplo 7 Parâmetro com valor padrão

```
CREATE PROCEDURE p_teste_valor_padrao (
```

```
@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_padrao 30
```

```
EXEC p_teste_valor_padrao @param1 = 40, @param2='Valor Modificado'
```




Parâmetros de Saída

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

Usamos a palavra-chave `OUTPUT` quando o procedimento é criado, 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.



Usando o RETURN

```
ALTER PROCEDURE p_LivroValor (@Qtd SMALLINT, @Cod SMALLINT = -10, @ID SMALLINT)
AS
  -SET NOCOUNT ON
  IF @ID >= 100
    BEGIN
      SELECT titulo AS Livro, valor*@Qtd AS Total
      FROM Livro
      WHERE id = @ID
      RETURN 1
    END
  ELSE
    RETURN @Cod
Executar:
DECLARE @Codigo INT
EXEC @Codigo = p_LivroValor @ID = 5, @Qtd = 10
PRINT @Codigo
```



Prática

1. Criar uma procedure para pesquisar um livro por um trecho de título, esta procedure deve exibir os dados do **Livro** contendo **autor** e sua **editora**.
2. Crie uma procedure para cadastrar no banco um livro contendo todas as suas informações e realizando um incremento no seu número de ID (coluna criada nos exemplo da aula) e validação de dados (se este livro já não está cadastrado).

Dica = use a função **max** para retornar o maior ID salvo .