Revisão
Stored Procedures

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Estrutura da stored procedure

1. **DELIMITER:** Esta cláusula é escrita seguida por uma combinação de símbolos que não serão usados dentro da SP.
2. **CREATE PROCEDURE:** Este comando é escrito seguido pelo nome que identificará a SP.
3. **BEGIN:** Esta cláusula é usada para indicar o início do código SQL.
4. **Bloco de Instruções SQL**.
5. **END:** Esta cláusula é escrita seguida pela combinação de símbolos definidos em DELIMITER e é usada para indicar o final do código SQL.

SQL

```sql
DELIMITER $$
CREATE PROCEDURE sp_nome_procedimento()
BEGIN
    -- Bloco de Instruções SQL;
END $$
```

O que é um parâmetro?

- Os parâmetros são variáveis para onde os dados são enviados e recebidos de programas clientes.
- Eles são definidos na cláusula CREATE.
- As SPs podem ter um, mais ou nenhum parâmetro de entrada e também podem ter um, mais ou nenhum parâmetro de saída.
- Existem 3 tipos de parâmetros:

Parâmetro	Tipo	Função
IN	Entrada	Recebe dados
OUT	Saída	Retorna datos
INOUT	Entrada-Saída	Recebe e retorna dados

- Uma SP pode ter parâmetros. Os parâmetros representam como uma SP pode receber valores, retornar valores ou ambos.
- Existem parâmetros de entrada (IN), saída (OUT) e entrada / saída (INOUT).

Exemplo:

SQL
```sql
DELIMITER $$
CREATE PROCEDURE sp_produtos(IN filtro_categoria VARCHAR(15))
BEGIN
    SELECT ProdutoNome, PrecoUnitario FROM produtos p
    JOIN categorias C ON p.CategoriaID = c.CategoriaID
    WHERE CategoriaNome = filtro_categoria
   END $$
```

SQL
```sql
CALL sp_produtos('Frutos do mar');
```

- Uma SP pode ter parâmetros. Os parâmetros representam como uma SP pode receber valores, retornar valores ou ambos.
- Existem parâmetros de entrada (IN), saída (OUT) e entrada / saída (INOUT).

Exemplo:

SQL
```
CREATE PROCEDURE sp_produtos(IN filtro_categoria VARCHAR(15), in nomeproduto
varchar(100))
BEGIN
    SELECT ProdutoNome, PrecoUnitario FROM produtos p
    JOIN categorias C ON p.CategoriaID = c.CategoriaID
    WHERE CategoriaNome = filtro_categoria
      and ProdutoNome = nomeproduto
END $$
```

SQL
```
CALL sp_produtos('Frutos do mar', ‘descricao produto’);
```

- Dentro de uma SP, é permitido declarar e atribuir valores a uma variável usando SET ou em uma instrução SELECT usando INTO.
- Fora da SP, usamos as variáveis acrescentando o símbolo @.

Exemplo:

SQL

```
DELIMITER $$
CREATE PROCEDURE sp_quaantidade_produtos(IN filtro_categoria VARCHAR(15), OUT
quantidade INT)
BEGIN
    SELECT count(*)  INTO quantidade
      FROM produtos p
    JOIN categorias C ON p.CategoriaID = c.CategoriaID
    WHERE CategoriaNome = filtro_categoria;
END $$
```

SQL

```
CALL sp_qaantidade_produtos('frutos do mar', @quantidade-frutos_do_mar);
SELECT @quanttidade_frutos_do_maar;
```

Declaração do parâmetro INOUT

É o mesmo parâmetro usado para entrada e saída de dados. Você pode receber valores e retornar os resultados na mesma variável.

Sintaxe:

```sql
CREATE PROCEDURE sp_nome_procedimento(INOUT param1 TIPO_DE_DADO, INOUT param2 TIPO_DE_DADO);
```

Exemplo:

```sql
DELIMITER $$
CREATE PROCEDURE sp_nome_procedimento(INOUT aumento FLOAT)
BEGIN
     SET aumento = aumento + 25700.50;
END $$
```

Execução:

```sql
SET @salario = 2000.00; -- Declaração e atribuição de variável (dado)
CALL sp_nome_procedimento(@salario); -- Execução e envío de dado (2000.00)
SELECT @salario; -- Exibe o resultado
```

Vantagens da stored procedure

- **Grande velocidade de resposta:** tudo é processado no servidor.
- **Maior segurança:** Limita e impoede o acesso direto às tabelas onde os dados são armazenados, evitando a manipulação direta por aplicativos clientes.
- **Independência:** Todo o código está dentro do banco de dados e não depende de arquivos externos.
- **Reutilização do código:** a necessidade de reescrever um conjunto de instruções é eliminada.
- **Manutenção mais fácil:** Menor custo de modificação quando as regras de negócios mudam.

Desvantagens da stored procedure

- **Modificação difícil:** se a modificação for necessária, sua definição deve ser totalmente substituída. Em bancos de dados muito complexos, a modificação pode afetar outras partes do software que direta ou indiretamente se referem a ele.
- **Aumento do uso de memória:** se usarmos muitos procedimentos armazenados, o uso da memória de cada conexão que usa esses procedimentos aumentará substancialmente.
- **Restrito para lógica de negócios complexa:** Na realidade, as construções de procedimento armazenado não são projetadas para desenvolver lógica de negócios complexa e flexível.


DigitalHouse>
Coding School