Banco de Dados

Stored Procedure

Profa.: Márcia Sampaio Lima

EST - UEA

Stored Procedure

- São rotinas definidas no banco de dados, identificadas por um nome pelo qual podem ser invocadas.

- Uma Stored Procedure pode executar uma série de instruções, receber parâmetros e retornar valores.

Stored Procedure

- Forma de transferir parte do processamento direto sobre os dados para o banco de dados.
- Considerando que as máquinas servidoras geralmente têm configurações de hardware mais robustas em relação às máquinas clientes.
- Como executar várias ações no banco de dados a partir de uma única instrução?
  - Stores Procedures

Stored Procedure

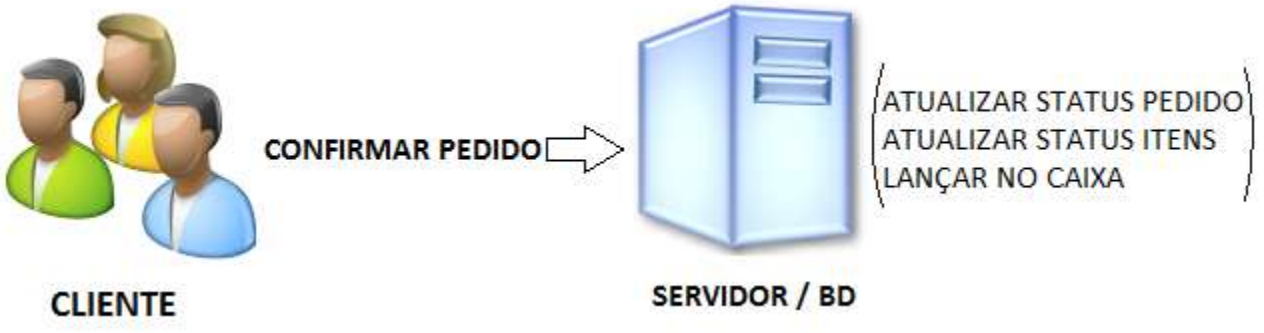
- Exemplo do minimundo: comercio
  - O cliente faz um pedido, no qual são inseridos itens;
  - O pedido (bem como os itens) permanece com status “PENDENTE” até ser confirmado;
  - O operador confirma o pedido, registrando a venda.

Stored Procedure

- Exemplo do minimundo: comercio
  - Ao confirmar a venda é preciso ter uma rotina de confirmação do pedido:
    - Atualizar o status do pedido;
    - Atualizar o status dos itens do pedido;
    - Gerar a Nota Fiscal.
  - Devem ser executadas pelo menos três instruções de atualização e/ou inserção.

Stored Procedure

- Por outro lado, poderíamos agrupar essas três instruções no corpo de um procedimento e chamá-lo a partir da aplicação uma única vez.
- As ações de update/insert/delete, a partir daí, ficariam por conta do servidor.



Stored Procedure

- Vantagens:
  - Simplificação da execução de instruções **SQL** pela aplicação;
  - Transferência de parte da responsabilidade de processamento para o servidor.
  - Facilidade na manutenção, reduzindo a quantidade de alterações na aplicação.

Stored Procedure

- Pontos negativos:
  - Necessidade de maior conhecimento da sintaxe do banco de dados para escrita de rotinas em SQL;
  - As rotinas ficam mais facilmente acessíveis. Alguém que tenha acesso ao banco poderá visualizar e alterar o código.

Stored Procedure

- **Criação de stored procedures no MySQL**

```
DELIMITER $$

CREATE PROCEDURE nome_procedimento (parâmetros)
BEGIN
        /*CORPO DO PROCEDIMENTO*/
END $$

DELIMITER ;
```

Stored Procedure

- Onde:
  - “nome_procedimento” : nome que identificará o procedimento armazenado. Segue as mesmas regras para definição de variáveis, não podendo iniciar com número ou caracteres especiais (exceto o underline “_”).
  - “parâmetros”: são opcionais.

Stored Procedure

- **Procedimentos com Parâmetros**
  - Para que um procedimento receba parâmetros, é necessário seguir certa sintaxe (dentro dos parênteses),

```
(MODO nome TIPO, MODO nome TIPO, MODO nome TIPO)
```

Stored Procedure

- **Procedimentos com Parâmetros**

```
(MODO nome TIPO, MODO nome TIPO, MODO nome TIPO)
```

  - “nome”: dos parâmetros.
  - “TIPO” : int, varchar, decimal
  - “MODO”: indica a forma como o parâmetro será tratado no procedimento, se será apenas um dado de entrada, apenas de saída ou se terá ambas as funções. Os valores possíveis:

Stored Procedure

- Onde:
  - IN: indica que o parâmetro é apenas para entrada/recebimento de dados, não podendo ser usado para retorno;
  - OUT: usado para parâmetros de saída. Para esse tipo não pode ser informado um valor direto (como ‘teste’, 1 ou 2.3), deve ser passada uma variável “por referência”;
  - INOUT: este tipo de parâmetro pode ser usado para os dois fins (entrada e saída de dados).

Stored Procedure

- **Chamando uma Stored Procedure**
  - Através da palavra reservada CALL:
  - CALL nome_procedimento(parâmetros);

Stored Procedure

- **Exemplo 1**: Recebimento de Parâmetros
  - Objetivo: fazer um select na tabela Livros, limitando a quantidade de registros pela quantidade recebida como parâmetro

```
DELIMITER $$

CREATE PROCEDURE Selecionar_Livros(IN quantidade INT)
BEGIN
    SELECT * FROM livros
     LIMIT quantidade;
END $$

DELIMITER ;
```

Stored Procedure

- Como executá-la?
  - Aba Rotinas --> Seleciona a Procedure --> Executar

Stored Procedure

- Exemplo: recebimento e retorno de parâmetro de saída.
  - Objetivo: retornar a quantidade de registros da tabela Livros, passando esse valor para a variável de saída “quantidade”.
  - Para isso foi utilizada a palavra reservada INTO.

```
DELIMITER $$

CREATE PROCEDURE Verificar_Quantidade_Produtos(OUT
quantidade INT)
    BEGIN
        SELECT COUNT(*) INTO quantidade FROM Livros;
   END $$

DELIMITER ;
```

Stored Procedure

- **Como chamá-la?**
  - **A**ba Rotinas --> Seleciona a Procedure --> Executar

  - Uso do símbolo (@) seguido do nome da variável que receberá o valor de saída.
  - CALL Verificar_Quantidade_Produtos(@total);
  - SELECT @total;

Stored Procedure

- Exemplo 3:
  - Stored Procedure que recebe uma variável e a altera, definindo-a como o seu próprio valor elevado à segunda potência.

```
DELIMITER $$

CREATE PROCEDURE Elevar_Ao_Quadrado(INOUT numero
INT)
BEGIN
    SET numero = numero * numero;
END $$

DELIMITER ;
```

Stored Procedure

- Os procedimentos são armazenados:
  - tabela ROUTINES
  - Do BD INFORMATION_SCHEMA, que é o dicionário de dados do MySQL.

- Listar todos os stored routines:

```
SELECT * FROM INFORMATION_SCHEMA.ROUTINES;
```

Stored Procedure

```
DELIMITER $$

CREATE PROCEDURE biblioteca_livroInsert(v_cod INT,
v_nome VARCHAR(60), v_ano INT, v_qtde INT)
BEGIN
 IF ((v_nome != '') && (v_ano > 2000))
  THEN
    INSERT INTO livros VALUES (v_cod,v_nome, v_ano,
v_qtde);
   ELSE
     SELECT 'NOME e ANO devem ser fornecidos para o
cadastro!' AS Msg;
  END IF;
END $$

DELIMITER ;
```