

Stored Procedures

Stored procedures são funções muito úteis para o dia-dia de DBAs e desenvolvedores. Stored Procedure, que traduzido significa Procedimento Armazenado, é um conjunto de comandos em **SQL** que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

Um Stored Procedure pode reduzir o tráfego na rede, melhorar a performance de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc.

Há cinco tipos de procedures básicos que podemos criar:

- Procedimentos Locais - São criados a partir de um banco de dados do próprio usuário;
- Procedimentos Temporários - Existem dois tipos de procedimentos temporários: Locais, que devem começar com # e Globais, que devem começar com ##;
- Procedimentos de Sistema - Armazenados no banco de dados padrão do SQL Server (Master), podemos indentifica-los com as siglas sp, que se origina de stored procedure. Tais procedures executam as tarefas administrativas e podem ser executadas a partir de qualquer banco de dados.
- Procedimentos Remotos - Podemos usar Queries Distribuídas para tais procedures. São utilizadas apenas para compatibilidade.
- Procedimentos Estendidos - Diferente dos procedimentos já citados, este tipo de procedimento recebe a extensão .dll e são executadas fora do SGBD SQL Server. São identificadas com o prefixo xp.

Quando utilizar procedures

- Quando temos várias aplicações escritas em diferentes linguagens, ou rodam em plataformas diferentes, porém executam a mesma função.
- Quando damos prioridade à consistência e segurança.

Os bancos (Itaú, Bradesco, Real, etc), por exemplo, em geral, utilizam stored procedures para todas as operações em comum. Os procedimentos podem assegurar que as operações sejam registradas de forma correta e segura.

Por que é mais seguro?

Seguindo a linha de raciocínio dos bancos, **utilizando stored procedures outras aplicações e usuários não conseguiriam nenhum tipo de acesso às tabelas do banco de dados** de forma direta.

Eles poderiam apenas executar os stored procedures, que rodam ações específicas e determinadas pelos DBAs e desenvolvedores.

Criando a Tabela Estados

```
CREATE TABLE uf (id INT NOT NULL PRIMARY KEY , estado VARCHAR(02) NULL , pais VARCHAR(45) NULL );
```

Inserindo Registros

```
INSERT INTO uf (id,estado,pais) VALUES (1, 'SP', 'Brasil');
INSERT INTO uf (id,estado,pais) VALUES (2, 'MG', 'Brasil');
INSERT INTO uf (id, estado, pais) VALUES (3, 'RR', 'Brasil');
INSERT INTO uf (id,estado, pais) VALUES (4, 'AA', 'USA');
INSERT INTO uf (id, estado, pais) VALUES (5, 'OC', 'USA');
```

Criando uma Stored Procedure

```
DELIMITER $$
CREATE PROCEDURE listar_estado (IN _id VARCHAR(10))
BEGIN
    IF(_id IS NULL) THEN

        SELECT * FROM uf;

    ELSE

        SELECT * FROM uf where pais = _id;

    END IF;
END
```

Testando Stored Procedure

- 1) CALL listar_estado(null);
- 2) CALL listar_estado("Brasil");

Executando através do PHP

Lista de Países

<select name="responsavel" id="responsavel">

<?php

\$host="localhost";

\$bancodedados="meu";

\$usuario_banco="root";

\$senha_banco="2635968";

\$conn;

\$db;

\$result;

/* conectando ao banco */

\$conn=mysql_connect(\$host,\$usuario_banco,\$senha_banco);

\$db=mysql_select_db(\$bancodedados,\$conn);

\$result = mysql_query("CALL listar_estado('Brasil');", \$conn);

while (\$linha = mysql_fetch_array(\$result))

{

 echo "<option value=" . \$linha['id'] . ">" . \$linha['estado'] . "</option>";

}

?>

</select>

Exercício II

DELIMITER \$\$

CREATE PROCEDURE selecionar_alunos(IN quantidade INT)

BEGIN

SELECT * FROM notas LIMIT quantidade;

END \$\$

DELIMITER

Exercício III - Propor

DELIMITER \$\$

CREATE PROCEDURE dataaniversario (IN _id VARCHAR(10))

BEGIN

SELECT * FROM agenda where datanasc = _id;

END

Testando

CALL dataaniversario('01/10/2012');

Exercício IV – Left Join , max , min