

Banco de Dados: Stored Procedures

Procedimentos Armazenados

Prof. Márcio Funes



Plano de aula

Procedimentos Armazenados

Parâmetros

Variáveis

Estruturas de controle

Exercícios

Introdução

O contexto

O gerenciamento de bancos de dados, de maneira geral, possui processos que podem ser automatizados.

Por exemplo:

- Inserir e atualizar registros
- Fazer consultas parametrizadas
- Backups
- Implementar regras de negócio
- Validar dados

O contexto

Uma **Stored Procedure** é um rotina armazenada no banco de dados.

Podem conter comandos SQL, além de algoritmos com variáveis, loops e estruturas de decisão.

Uma procedure é semelhante a um método na linguagem Java ou Python.

Podem ainda ser usadas para restringir acesso a grupos de usuários, aumentando com isso a segurança do banco de dados.

Com isso, é possível agilizar processos na aplicação.

Stored Procedures

Stored Procedures

Veja abaixo a sintaxe mínima para criação de uma Stored Procedure no **MYSQL**

```
create procedure nome_da_procedure()  
begin  
    -- Comandos SQL  
end
```

Stored Procedures

Veja abaixo a sintaxe mínima para criação de uma Stored Procedure no **SQL SERVER**.

```
CREATE PROCEDURE nome_da_procedure AS  
-- comandos sql
```

Stored Procedures

MySQL

```
create procedure nome_da_procedure()  
begin  
    -- Comandos SQL  
end
```

SQL SERVER

```
CREATE PROCEDURE nome_da_procedure AS  
-- comandos sql
```


Stored Procedures

Para criar uma procedure que simplesmente seleciona dados no MYSQL:

```
create procedure relProdutos()  
begin  
    select produto.*, marca.nome from  
        produto left join marca on fk_marca = marca.id;  
end
```

Stored Procedures

Para criar uma procedure que simplesmente seleciona dados no SQL SERVER:

```
CREATE PROCEDURE seleciona_produto AS  
SELECT * FROM produtos
```

Stored Procedures

Para esse exemplo considere as seguintes tabelas:

```
create database Empresa;  
use Empresa;
```

```
create table marcas (  
    marca_id      int          auto_increment    primary key,  
    nome_marca    varchar(50)  not null,  
    origem        varchar(50)  
);
```

```
create table produtos (  
    prod_id       int          auto_increment    primary key,  
    nome_prod     varchar(50)  not null,  
    qtd_estoque   int          not null          default 0,  
    estoque_mim   int          not null          default 0,  
    data_fabricacao timestamp          default now(),  
    perecivel     boolean,  
    valor         decimal(10,2),  
    marca_id      int          references marcas(marca_id)  
);
```

Stored Procedures

Uma procedure é executada por meio do comando **call** no MySQL:

```
50 • call relProdutos();
```

```
51
```

```
52
```



	id	nome	preco_custo	preco_venda	lucro	fk_marca	nome
▶	1	Lápis	5.00	8.00	3.00	NULL	NULL
	2	Borracha	2.00	4.00	2.00	2	Labra
	3	caneta	3.00	5.00	2.00	3	Coimbra
	4	Caderno	12.00	32.00	20.00	4	Ajax

Stored Procedures

Uma procedure é executada por meio do comando **exec** no SQL SERVER:

```
CREATE PROCEDURE seleciona_produto AS  
SELECT * FROM produtos  
GO;
```

```
EXEC seleciona_produto;
```

134 %

Resultados Mensagens

	prod_id	nome_prod	qtd_estoque	estoque_mim	data_fabricacao	perecivel	valor	marca_id
1	1	lapis	4502	100	2016-03-03	0	2.50	1
2	2	lapis	8800	100	2015-05-05	0	14.00	2
3	3	borracha	2907	100	2013-07-08	0	4.20	1
4	4	borracha	5408	100	2015-08-02	0	2.00	2
5	5	caderno	7004	100	2016-03-04	0	22.50	1
6	6	caneta	8030	100	2013-02-04	0	11.00	1

Parâmetros

Stored Procedures

Uma procedure também pode receber parâmetros:

```
CREATE PROCEDURE estoqueBaixo  
@estoque int  
AS  
    SELECT produtos.*, marcas.nome_marca  
    FROM produtos LEFT JOIN marcas  
    ON marcas.marca_id = produtos.prod_id  
    WHERE qtd_estoque < @estoque  
  
EXEC estoqueBaixo 100;
```

Stored Procedures

Uma procedure também pode receber parâmetros:

```
create procedure relProdutos(marca_id integer)
begin
    select produto.*, marca.nome from
        produto left join marca on fk_marca = marca.id
    where marca.id = marca_id;
end
```

```
call relProdutos(2);
```


Stored Procedures

É possível ainda criar variáveis para gerar dados:

```
DECLARE @estoqueMinimo int  
SET @estoqueMinimo = 100;  
EXEC estoqueBaixo @estoqueMinimo;
```

Stored Procedures - Exemplo

01 – Crie um Procedimento Armazenado (SP) que toda vez que uma produto for cadastrado no banco de dados **produtos**, automaticamente o lucro deve ser calculado e inserido em outra tabela chamada **Lucro**.

```

] CREATE PROCEDURE novoLapis
    @nomeProduto varchar(50),
    @qtdEstoque int,
    @data_fabricacao date,
    @perecivel bit,
    @valor decimal(10,2)
AS
]   DECLARE @estoqueMin AS int,
        @lucro AS decimal(10,2),
        @minhaMarca AS int;
]   SET @estoqueMin = 50;
]   SET @lucro = @valor + 10;
]   SET @minhaMarca = (SELECT marcas.marca_id
                        FROM marcas
                        WHERE marcas.nome_marca = 'Faber Castel');
]   INSERT INTO produtos
VALUES (@nomeProduto,
        @qtdEstoque,
        @estoqueMin,
        @data_fabricacao,
        @perecivel,
        @valor,
        @minhaMarca);

]   INSERT INTO Lucro
VALUES (@nomeProduto,
        @lucro);

```

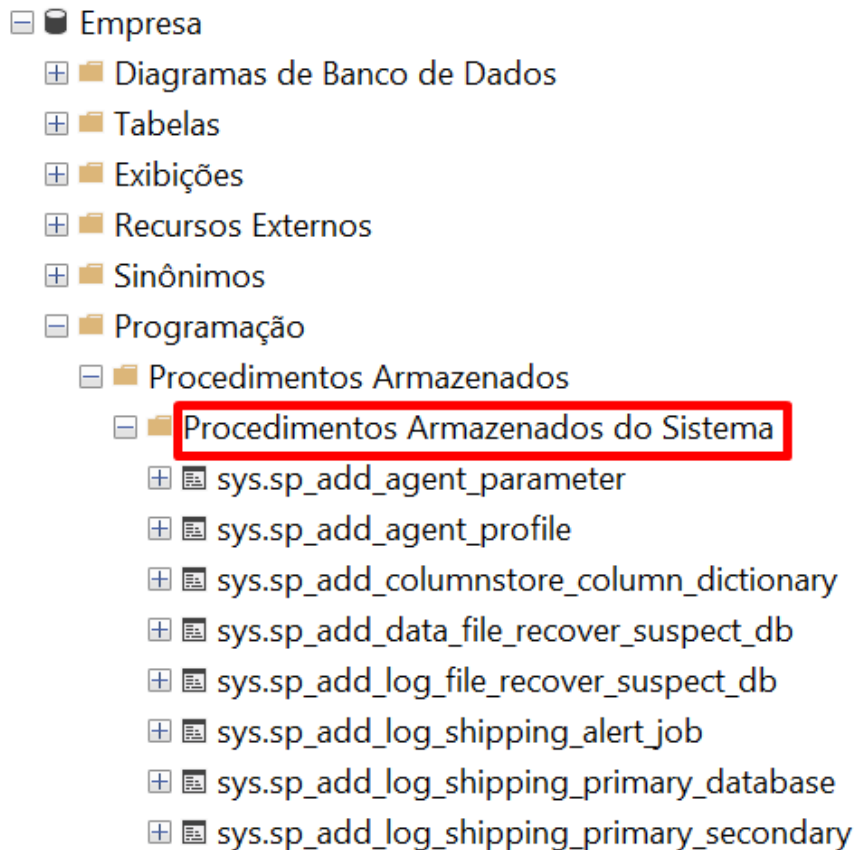
Stored Procedures - Exemplo

```
CREATE TABLE Lucro (  
    lucro_id          int          identity          primary key,  
    nome_produto      varchar(50)  not null,  
    lucro_produto     decimal(10,2)  
);
```

```
EXEC novoLapis 'Lápis X', 10, '2016-03-03', 0, 10.5;
```

```
SELECT * FROM produtos WHERE produtos.nome_prod LIKE 'Lápis X';  
SELECT * FROM Lucro;
```

O SQL SERVER possui diversas SP prontas que podemos utilizar.



Exemplos

Renomear uma SP existente

```
EXEC sp_rename 'novoLapis', 'sp_novoLapis';
```

Ver a estrutura de um SP.

```
EXEC sp_helptext sp_novoLapis;
```

Stored Procedures Internas

```
EXEC sp_monitor;  
EXEC sp_server_info;  
EXEC sp_helpdb;  
EXEC sp_datatype_info;  
EXEC sp_helplanguage;  
EXEC sp_helpindex Produtos;  
EXEC sp_helpconstraint Produtos;  
EXEC sp_depends Produtos;  
EXEC sp_who;  
EXEC sp_who2;  
EXEC sp_tables;  
EXEC sp_databases;  
EXEC sp_rename 'Produtos', 'Produtos1';
```

Para saber mais...

<https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html>

<https://medium.com/@peter.lafferty/mysql-stored-procedures-101-6b4fe230967>

<https://www.a2hosting.com/kb/developer-corner/mysql/mysql-stored-functions-and-procedures>

<https://www.w3resource.com/mysql/mysql-procedure.php>

<https://www.paladion.net/blogs/are-stored-procedures-safe-against-sql-injection>

Exercícios

Exercícios

Utilizando o script Empresa.sql, crie as **Storage Procedures** abaixo:

1. Crie uma SP que exibe o preço médio dos produtos.
2. Crie uma SP que ao passar uma marca como parâmetro retorna todos os produtos daquela marca.
3. Crie uma SP que receba dois valores (um menor e outro maior) como parâmetro e retorne todos os produtos com a quantidade dentro do intervalo dos dois valores fornecidos como parâmetros.
4. Crie uma SP onde após um novo registro na tabela produto_fornecedor for criado, ele exibe o nome do produto e o nome do fornecedor que acabou de ser registrado.
5. Crie uma SP que receba como parâmetro o nome de um fornecedor e insira automaticamente o nome do fornecedor e um e-mail no formato nome_fornecedor@nome_fornecedor.com.br na tabela fornecedores.