

Tarefa 24 - Stored Procedures – MySQL

Prof. Dr. Aparecido Freitas

1. Introdução

Para a execução desta atividade iremos considerar um Banco de Dados chamado **produtodb**, com as seguintes definições de tabelas:

CREATE TABLE Fabricante (

```
idFabricante INT(11) NOT NULL,  
Nome VARCHAR(60) NOT NULL,  
PRIMARY KEY (idFabricante)
```

);

CREATE TABLE Categoria (

```
idCategoria INT(11) NOT NULL,  
Descricao VARCHAR(60) NOT NULL,  
PRIMARY KEY (idCategoria)
```

);

CREATE TABLE Produto (

```
idProduto INT(11) NOT NULL,  
Descricao VARCHAR(45) NOT NULL,  
idCategoria INT(11) NULL DEFAULT NULL,  
idFabricante INT(11) NOT NULL,  
PRIMARY KEY (idProduto),  
INDEX fk_Produto_Categoria_idx (idCategoria ASC),  
INDEX fk_Produto_Fabricante1_idx (idFabricante ASC),  
CONSTRAINT fk_Produto_Categoria FOREIGN KEY (idCategoria) REFERENCES Categoria  
(idCategoria),  
CONSTRAINT fk_Produto_Fabricante1 FOREIGN KEY (idFabricante) REFERENCES Fabricante  
(idFabricante)
```

);

CREATE TABLE Filial (

```
idFilial INT(11) NOT NULL,  
idFabricante INT(11) NOT NULL,  
Nome VARCHAR(45) NOT NULL,  
Contato VARCHAR(45) NULL DEFAULT NULL,  
PRIMARY KEY (idFilial, idFabricante),  
INDEX fk_Filial_Fabricante1_idx (idFabricante ASC),  
CONSTRAINT fk_Filial_Fabricante1 FOREIGN KEY (idFabricante) REFERENCES Fabricante  
(idFabricante)
```

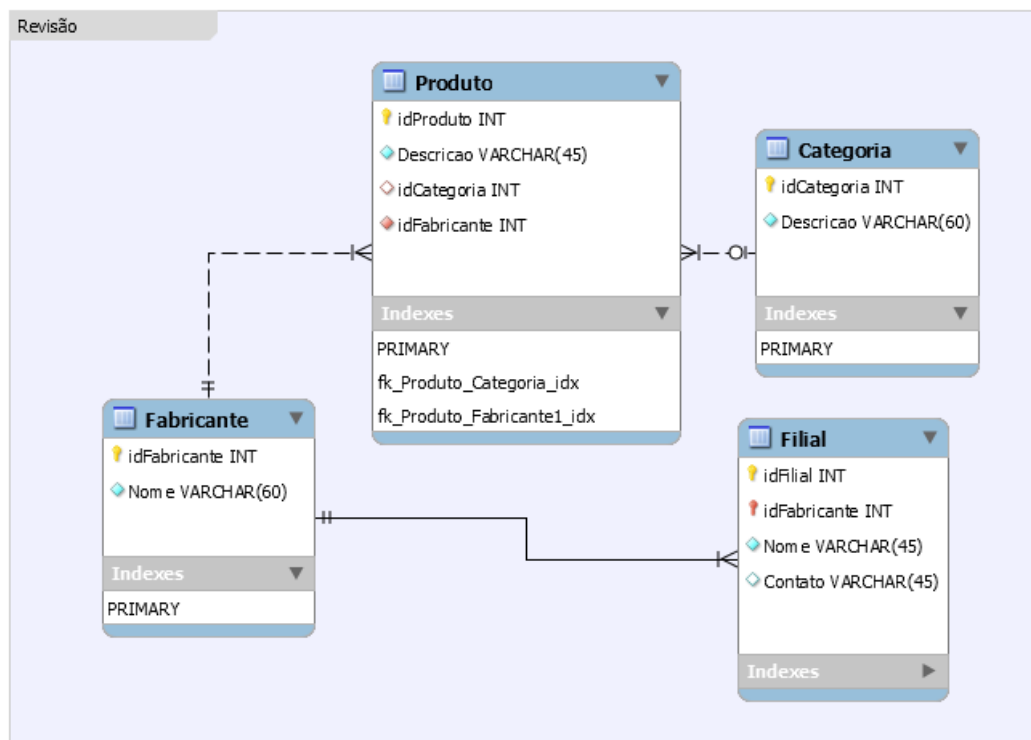
);

```

INSERT INTO fabricante (idFabricante, Nome) VALUES ('1', 'Nestlé');
INSERT INTO fabricante (idFabricante, Nome) VALUES ('2', 'Parmalat');
INSERT INTO fabricante (idFabricante, Nome) VALUES ('3', 'Kelloggs');
INSERT INTO categoria (idCategoria, Descricao) VALUES ('1', 'Leite');
INSERT INTO categoria (idCategoria, Descricao) VALUES ('2', 'Cereais Matinais');
INSERT INTO categoria (idCategoria, Descricao) VALUES ('3', 'Achocolatado');
INSERT INTO produto (idProduto, Descricao, idCategoria, idFabricante) VALUES (1, 'Leite Integral', '1', '1');
INSERT INTO produto (idProduto, Descricao, idCategoria, idFabricante) VALUES (2, 'Nescau', '3', '1');
INSERT INTO produto (idProduto, Descricao, idCategoria, idFabricante) VALUES (3, 'Sucrilhos', '2', '3');

```

Empregaremos o seguinte modelo de dados:



Uma variável local somente será válida durante a execução de uma **Stored Procedure**, sendo que, após o término da execução de tais procedimentos, esta variável é destruída da memória, juntamente com seu respectivo valor.

Uma variável é um objeto de dados nomeado, cujo valor pode mudar durante a execução do procedimento armazenado.

Para declarar uma variável dentro de um procedimento armazenado, você usa a instrução DECLARE da seguinte forma:

```
DECLARE varNome VARCHAR(45) DEFAULT 'Ricardo';
```

```
DECLARE varSaldo, varSequencia INT DEFAULT 0;
```

O valor de uma variável local pode ser alterado através da instrução SET:

```
DELIMITER $$
CREATE PROCEDURE sp_DeclaracaoVariaveisLocais()
BEGIN
    DECLARE varNome      VARCHAR(45) DEFAULT 'Ricardo';
    DECLARE varSobrenome VARCHAR(45);

    SET varSobrenome = ' Resende';

    SELECT CONCAT(varNome, varSobreNome);

END$$
DELIMITER ;
```

```
CALL sp_DeclaracaoVariaveisLocais();
```

Ou através da instrução into, nos casos em que o valor é proveniente do resultado de um select.

```
DELIMITER $$
CREATE PROCEDURE sp_AtribuicaoValor()
BEGIN
    DECLARE varQuant INT;
    SELECT COUNT(*) INTO varQuant FROM PRODUTO;
    SELECT CONCAT('O total de produtos é igual a ', varQuant);
END$$
DELIMITER ;
```

Após a execução da Procedure, será exibido o resultado:

```
mysql> call sp_atribuicaovalor();
+-----+
| concat('Total de produtos: ', varQuant) |
+-----+
| Total de produtos: 3                     |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

mysql>
```

É possível atribuir “N” valores em uma única instrução INTO.

```
DELIMITER $$
CREATE procedure sp_MultiplasAtribuicoes()
BEGIN
    DECLARE varCodProduto INT;
    DECLARE varDesProduto VARCHAR(45) DEFAULT '';

    SELECT p.idProduto, p.Descricao INTO varCodProduto, varDesProduto
    FROM produto p
    ORDER BY p.idProduto DESC
    LIMIT 1;

    SELECT CONCAT('O último produto cadastrado foi o ', varDesProduto,
                  ' com código ', varCodProduto, '.');
END$$
DELIMITER ;
```

Crie uma Stored Procedure chamada **sp_PraticandoVariaveisLocais**.

Ao ser invocada essa procedure deve, utilizando o conceito de Variável Local exibir o seguinte retorno:

"O [PRODUTO] não possui categoria vinculada"

Onde **[PRODUTO]** representa o nome de um produto que não possui uma categoria vinculada.

Resposta: