**Trabalho Prático – Etapas 3 e 4**

Daniel Gonçalves - 12011BCC011

Mateus Rocha Resende - 11921BCC027

Vitor Barbosa Lemes Fernandes - 11921BCC035

**Etapa 3 - Acesso via Linguagem Procedural no Servidor**

**i. Implementar um exemplo de integridade de dados ou alguma funcionalidade da aplicação que execute no servidor e seja acionada por meio de um gatilho. A função deve utilizar alguma forma de iteração (laço de repetição ou chamada recursiva).**

Segue abaixo tentativa de implementar um gatilho que retorna quantos pré-requisitos possui uma matéria em que um aluno se matriculou:

CREATE OR REPLACE FUNCTION fazprerequisito () RETURNS TRIGGER AS

$$

DECLARE

mycomand TEXT;

BEGIN

RETURN (WITH RECURSIVE tPrereq (nivel,disc,pre\_req) AS

(

SELECT 1, root.disc, root.pre\_req

FROM requisitos root

WHERE root.disc = NEW.disc\_sigla

UNION ALL

SELECT nivel+1, child.disc, child.pre\_req

FROM tPrereq parent, requisitos child

WHERE parent.pre\_req = child.disc

) SELECT DISTINCT COUNT(pre\_req) FROM tPrereq);

END

$$

LANGUAGE plpgsql;

CREATE TRIGGER tfazprerequisito BEFORE INSERT OR UPDATE ON matricula FOR EACH ROW EXECUTE PROCEDURE fazprerequisito();

**Etapa 4 - Acesso via Linguagem Procedural no Cliente**

1. **Implementar uma interface para Inclusão; Alteração e Exclusão em uma ou mais tabelas do BD;**
2. **Implementar uma interface para apresentar o resultado de uma consulta com junção de duas ou mais tabelas do BD;**
3. **Implementar uma transação que altere mais de uma tabela do BD rodando no cliente;**

O programa foi desenvolvido em Java. Abaixo segue o código fonte:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Principal {

static Scanner teclado = new Scanner (System.in);

static Connection conn = null;

static Statement stmt = null;

static ResultSet rs = null;

private static void editarProf() {

String opcao = "-1";

try{

System.out.println("COMANDOS: ");

System.out.println("[1] Adicionar um professor");

System.out.println("[2] Remover um professor");

System.out.println("[3] Alterar um professor");

opcao = teclado.nextLine();

switch(opcao){

case "1":

System.out.print("Id do professor: ");

int idprof = teclado.nextInt();

teclado.nextLine();

System.out.print("Nome do professor: ");

String nomeprof = teclado.nextLine();

System.out.print("Nascimento do professor (DD-MM-AAAA): ");

String nascprof = teclado.nextLine();

System.out.print("Salario do professor: ");

Float salario = teclado.nextFloat();

teclado.nextLine();

System.out.print("Sigla da faculdade: ");

String sigla = teclado.nextLine();

String query = "INSERT INTO professor VALUES ("+idprof+", '"+nomeprof+"', '"+nascprof+"', "+salario+", '"+sigla+"');";

int r = stmt.executeUpdate(query);

if(r>0){

System.out.println("Adicionado com sucesso!");

}else{

System.out.println("Erro!");

}

break;

case "2":

System.out.print("Id do professor que deseja remover: ");

idprof = teclado.nextInt();

teclado.nextLine();

query = "DELETE FROM professor WHERE id\_prof = "+ idprof +";";

r = stmt.executeUpdate(query);

if(r>0){

System.out.println("Removido com sucesso!");

}else{

System.out.println("Não existe um professor com o id informado!");

}

break;

case "3":

System.out.println("Id do professor que deseja alterar: ");

idprof = teclado.nextInt();

teclado.nextLine();

System.out.print("Novo nome: ");

nomeprof = teclado.nextLine();

System.out.print("Nova data de nascimento (DD-MM-AAAA): ");

nascprof = teclado.nextLine();

System.out.print("Novo salario: ");

salario = teclado.nextFloat();

teclado.nextLine();

System.out.print("Nova sigla da faculdade: ");

sigla = teclado.nextLine();

query = "UPDATE professor SET prof\_nome = '"+nomeprof+"', nascimento\_prof = '"+nascprof+"', salario = '"+salario+"', facul\_sigla= '"+sigla+"' WHERE id\_prof = '"+idprof+"';";

r = stmt.executeUpdate(query);

if(r>0){

System.out.println("Atualizado com sucesso!");

}else{

System.out.println("Erro!");

}

break;

default:

break;

}

}catch(SQLException e){

System.out.println("Erro de sintaxe?");

}

}

private static void consulta(){

String query = "SELECT prof\_nome, facul\_sigla FROM professor NATURAL JOIN faculdade;";

try{

ResultSet r = stmt.executeQuery(query);

System.out.println("'prof\_nome', 'facul\_sigla'");

while(r.next()){

String prof\_nome = r.getString("prof\_nome");

String sigla = r.getString("facul\_sigla");

System.out.println("'"+prof\_nome+"', '"+sigla+"'");

}

}catch(SQLException e){

System.out.println("ERRO!");

}

}

private static void alteraNota(){

try{

System.out.print("Id do aluno que deseja alterar a nota: ");

int idaluno = teclado.nextInt();

teclado.nextLine();

System.out.print("Id da turma do aluno: ");

int idturma = teclado.nextInt();

teclado.nextLine();

System.out.print("Sigla da disciplina: ");

String sigladisc = teclado.nextLine();

System.out.print("Nova nota do aluno: ");

float nNota = teclado.nextFloat();

teclado.nextLine();

System.out.print("Novo CRA do aluno: ");

float nCRA = teclado.nextFloat();

teclado.nextLine();

String query = "rollback;\n" +

"BEGIN TRANSACTION;\n" +

"UPDATE matricula SET nota = '"+nNota+"' WHERE id\_alun = '"+idaluno+"' AND id\_turma = '"+idturma+"' AND disc\_sigla = '"+sigladisc+"';\n" +

"UPDATE aluno SET cra = '"+nCRA+"' WHERE id\_alun = '"+idaluno+"';\n" +

"END TRANSACTION;";

stmt.executeUpdate(query);

}catch(SQLException e){

System.out.println("ERRO!");

}

}

public static void main(String[] args) {

String url = "jdbc:postgresql://200.131.206.13:5432/vitorlemes";

System.out.print("User: ");

String user = teclado.nextLine();

System.out.print("Senha: ");

String password = teclado.nextLine();

String query = "";

try {

Class.forName("org.postgresql.Driver");

} catch (ClassNotFoundException ex) {

Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);

}

try{

conn = DriverManager.getConnection(url, user, password);

stmt = conn.createStatement();

stmt.execute("SET search\_path TO universidade");

String opcao = "-1";

while(!"0".equals(opcao)){

System.out.println("\nCOMANDOS: ");

System.out.println("[1] Editar professores");

System.out.println("[2] Consultar relação de professores e faculdades");

System.out.println("[3] Alterar a nota de um aluno");

System.out.println("[0] Encerrar programa");

opcao = teclado.nextLine();

switch(opcao){

case "1":

editarProf();

break;

case "2":

consulta();

break;

case "3":

alteraNota();

break;

default:

break;

}

}

}catch(SQLException e){

System.out.println(e.getMessage());

}

}

}