package com.div.com;

import java.util.Scanner;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import java.sql.PreparedStatement;

public class connectdb {

static void addstock() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

Scanner myObj = new Scanner(System.in);

try {

System.out.println("Enter PROD ID\n");

String prod = myObj.nextLine();

System.out.println("Enter Product name\n");

String prod\_name = myObj.nextLine();

System.out.println("Enter product price\n");

int prod\_price = myObj.nextInt();

System.out.println("You entered product id as"+ prod +"\n "+ "product name as"+ prod\_name+"\n " +" and product price as" + prod\_price);

/\*dd a product (p100, cd, 5) in Product and (p100, d2, 50) in Stock. \*/

String sql18 = "INSERT INTO stock (PROD,DEP,QUANTITY) VALUES (?,?,?);";

PreparedStatement stmt = connection.prepareStatement(sql18);

stmt.setString(1,prod);

stmt.setString(2,prod\_name);

stmt.setInt(3,prod\_price);

stmt.executeUpdate();

stmt.close();

connection.close();

System.out.println("added");

} finally {

myObj.close();

}

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void viewproduct() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

System.out.println("Connection OK");

Statement stmt = connection.createStatement();

String sql1 = "select \* from product";

stmt.executeUpdate(sql1);

stmt.close();

connection.close();

System.out.println("sql1");

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void deletevalues() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

System.out.println("Connection OK");

Statement stmt = connection.createStatement();

String sql1 = "DELETE FROM product WHERE prod = 'P1';";

stmt.executeUpdate(sql1);

stmt.close();

connection.close();

System.out.println("product P1 has been deleted");

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void createtables() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

System.out.println("Connection OK");

Statement stmt = connection.createStatement();

String sql1 = "CREATE TABLE IF NOT EXISTS PRODUCT" +

"(PROD CHAR(50) PRIMARY KEY NOT NULL," +

" NAME TEXT NOT NULL, " +

" PRICE INT NOT NULL)";

String sql2 = "CREATE TABLE IF NOT EXISTS DEPOT " +

"(DEP CHAR(50) PRIMARY KEY NOT NULL," +

" ADDR TEXT NOT NULL, " +

" VOLUME INT NOT NULL)";

String sql3 = "CREATE TABLE IF NOT EXISTS STOCK " +

"(PROD CHAR(50) PRIMARY KEY NOT NULL,"+

"DEP CHAR(50),"+

" QUANTITY INT NOT NULL)";

stmt.executeUpdate(sql1);

stmt.executeUpdate(sql2);

stmt.executeUpdate(sql3);

stmt.close();

connection.close();

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void addproduct() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

Scanner myObj = new Scanner(System.in);

try {

System.out.println("Enter PROD ID\n");

String prod = myObj.nextLine();

System.out.println("Enter Product name\n");

String prod\_name = myObj.nextLine();

System.out.println("Enter product price\n");

int prod\_price = myObj.nextInt();

System.out.println("You entered product id as"+ prod +"\n "+ "product name as"+ prod\_name+"\n " +" and product price as" + prod\_price);

/\*dd a product (p100, cd, 5) in Product and (p100, d2, 50) in Stock. \*/

String sql18 = "INSERT INTO PRODUCT (PROD,NAME,PRICE) VALUES (?,?,?);";

String sql19 = "INSERT INTO STOCK (PROD,DEP,QUANTITY) VALUES (?,'D1','9000');";

PreparedStatement stmt = connection.prepareStatement(sql18);

PreparedStatement stmt1 = connection.prepareStatement(sql19);

stmt.setString(1,prod);

stmt.setString(2,prod\_name);

stmt.setInt(3,prod\_price);

stmt.executeUpdate();

stmt.close();

stmt1.setString(1,prod);

stmt1.executeUpdate();

stmt1.close();

connection.close();

System.out.println("added");

} finally {

myObj.close();

}

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void adddepot() {

Connection connection = null;

String host="localhost";

String port="5432";

String db\_name="posts";

String username="postgres";

String password="new\_password";

try {

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

connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name+"", ""+username+"", ""+password+"");

if (connection != null) {

Scanner appy = new Scanner(System.in);

try {

System.out.println("Enter depot id\n");

String prod = appy.nextLine();

System.out.println("Enter depot addres\n");

String prod\_name = appy.nextLine();

System.out.println("Enter product volume\n");

int prod\_price = appy.nextInt();

System.out.println("You entered product id as"+ prod +"\n "+ "product name as"+ prod\_name+"\n " +" and product price as" + prod\_price);

/\*dd a product (p100, cd, 5) in Product and (p100, d2, 50) in Stock. \*/

String sql18 = "INSERT INTO DEPOT (DEP,ADDR,VOLUME) VALUES (?,?,?);";

String sql19 = "INSERT INTO STOCK (PROD,DEP,QUANTITY) VALUES (?,?,?);";

String A = "P1";

String B = prod;

int C = 9000;

PreparedStatement stmt = connection.prepareStatement(sql18);

PreparedStatement stmt1 = connection.prepareStatement(sql19);

stmt.setString(1,prod);

stmt.setString(2,prod\_name);

stmt.setInt(3,prod\_price);

stmt.executeUpdate();

stmt.close();

stmt1.setString(1,A);

stmt1.setString(2,B);

stmt1.setInt(3,C);

stmt1.executeUpdate();

stmt1.close();

connection.close();

System.out.println("added");

} finally {

appy.close();

}

} else {

System.out.println("Connection Failed");

}

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

createtables();

/\*addvalues("huhuhuh","jiojoijoijoi",9000);

deletevalues();\*/

int i = 0;

while( i < 6 ) {

System.out.println("ENTER\n"

+ "1 To add product\n"

+ "2 To Add depot\n"

+ "3 To view product\n"

+ "4 To view stock\n"

+ "5 To vie depot\n"

+ "6 to add stock\n");

Scanner myObj1 = new Scanner(System.in);

try {

int userName = myObj1.nextInt();

System.out.println(("you entered: " + userName));

if(userName == 1) {

addproduct();

}

if(userName == 2) {

adddepot();

}

if(userName == 3) {

//viewproduct();

System.out.println("products ..........");

}

if(userName == 4) {

System.out.println("View stock............");

}

if(userName == 5) {

System.out.println("View Depo..........t");

}

if(userName == 6) {

addstock();

}

if(userName > 6) {

System.out.println("Invalid Input");

}

} finally {

myObj1.close();

}

}

}

}