МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Отчет по лабораторной работе № 3

по дисциплине

Информационные технологии в системах управления

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Миндров А.Е.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Меженин М.Д.

(подпись) (фамилия, и.,о.)

17-АС

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2020

**Цель:**  создание базы данных и всех таблиц.

**Листинг:**

**Main.java**

package CargoTransportation;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.util.Scanner;  
import java.util.StringTokenizer;  
  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String Server = "jdbc:mysql://localhost:3306/";  
 String Username = "root";  
 String Password = "root";  
 String pgDbName = "sys";  
 String myDbName = "CargoTransportation";  
 String URL = Server + pgDbName + "?useUnicode=true&serverTimezone=UTC";  
  
  
 System.*out*.println("--------Connecting to database----------");  
 Connection connection = null;  
 try {  
 connection = DriverManager.*getConnection*(URL,Username,Password);  
 } catch (SQLException ex) {  
 System.*out*.println(ex);  
 }  
 if (connection != null) {  
 System.*out*.println("Connection established!");  
 } else {  
 System.*out*.println("Connection failed!");  
 }  
  
 *DispHelp*();  
  
  
  
 boolean cont = true;  
 String inp = " ";  
 Scanner input = new Scanner(System.*in*);  
 while(cont){  
 System.*out*.printf(">");  
 inp = input.nextLine();  
 StringTokenizer cmd\_line = new StringTokenizer(inp);  
 int num\_of\_tokens = cmd\_line.countTokens();  
 if (num\_of\_tokens==0)  
 break;  
 String cmd = cmd\_line.nextToken().toLowerCase();  
  
 if(cmd.equals("q")) cont = false;  
 *//* else if(cmd.equals("cdb"))  
 {  
  
 System.*out*.println("Creating "+myDbName+" database. Please wait ...");  
  
 URL = Server + myDbName + "?useUnicode=true&serverTimezone=UTC";  
 String command = "CREATE DATABASE " + myDbName;  
 PreparedStatement preparedStatement = connection.prepareStatement(command);  
 try{  
 preparedStatement.executeUpdate();  
 System.*out*.println("Database "+myDbName+" has been created!");  
  
 try {  
 connection = DriverManager.*getConnection*(URL, Username, Password);  
 System.*out*.println("------ " + myDbName +" database connection established ------");  
  
  
 PreparedStatement prepSt = null;  
 System.*out*.println("Creating \"Cargo\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Cargo ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "type varchar (30),"  
 + "cost float (2),"  
 + "volume float (2),"  
 + "weight float (2),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Vehicles\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Vehicles ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "license\_plate varchar(9),"  
 + "model varchar(30),"  
 + "fuel\_consumption float (1),"  
 + "carrying float (2),"  
 + "wagon\_volume float (2),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Drivers\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Drivers ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "full\_name varchar (50),"  
 + "license varchar(11),"  
 + "phone\_number varchar(15),"  
 + "vehicle\_id integer,"  
 + "FOREIGN KEY (vehicle\_id) REFERENCES Vehicles (id),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Clients\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Clients ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "full\_name varchar(50),"  
 + "phone\_number varchar (15),"  
 + "address varchar (50),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Managers\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Managers ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "full\_name varchar (50),"  
 + "phone\_number varchar (15),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Routes\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Routes ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "start\_point varchar (50),"  
 + "end\_point varchar(50),"  
 + "distance float (2),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
  
 System.*out*.println("Creating \"Orders\" table. Please wait ...");  
 prepSt = connection.prepareStatement("CREATE TABLE Orders ("  
 + "id integer NOT NULL AUTO\_INCREMENT,"  
 + "manager\_id integer,"  
 + "client\_id integer,"  
 + "route\_id integer,"  
 + "driver\_id integer,"  
 + "cargo\_id integer,"  
 + "FOREIGN KEY (manager\_id) REFERENCES Managers (id),"  
 + "FOREIGN KEY (client\_id) REFERENCES Clients (id),"  
 + "FOREIGN KEY (route\_id) REFERENCES Routes (id),"  
 + "FOREIGN KEY (driver\_id) REFERENCES Drivers (id),"  
 + "FOREIGN KEY (cargo\_id) REFERENCES Cargo (id),"  
 + "order\_date varchar (10),"  
 + "delivery\_date varchar (10),"  
 + "PRIMARY KEY (id));");  
 prepSt.executeUpdate();  
  
 System.*out*.println("Done ...");  
  
  
 } catch (SQLException ex) {  
 System.*out*.println("- ? -" + ex);  
 }  
  
 }  
 catch(SQLException e)  
 {  
 System.*out*.println("Error: " +e);  
 }  
 }  
 else if(cmd.equals("help"))  
 {  
 *DispHelp*();  
 }  
 else  
 {  
 System.*out*.println("Unknown command!");  
 *DispHelp*();  
 }  
 }  
 }  
  
 private static void DispHelp()  
 {  
 System.*out*.println("q - quit");  
 System.*out*.println("cdb - create database");  
 System.*out*.println("help - for help");  
 }  
}

**Результат работы:**

