Частное учреждение образования

«Колледж бизнеса и права»

Лабораторная работа № 12

Учащийся Е. А. Чужавко

группа: Т-992

2022

**Задание. Вариант 2.**

Реализовать БД по вариантам. Заполнить с помощью программы.

Накладные – НомДок, ДатаПост, Поставщик, НоменкНомМатериала, ТабНомМОЛ, Колич

Материалы – НоменклНомМатериала, НаименованиеМатериала, ЕдиницаИзмерения, Цена

МатОтЛица – ТабНомМОЛ, ФамилияМОЛ

Поставщики – НомПоставщика, НаименПоставщика

package com.company;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws ClassNotFoundException, SQLException {

Class.forName("com.mysql.cj.jdbc.Driver");

Scanner in = new Scanner(System.in);

int i = 10;

while(i!=5){

System.out.println("Введите пункт из меню: \n1 - создать таблицы\n2 - заполнить таблицы\n3 - вывести значения из таблиц\n4 - удалить таблицы\n5 - выход");

i = in.nextInt();

switch (i){

case 1:

createTables();

break;

case 2:

fillTables();

break;

case 3:

printTables();

break;

case 4:

deleteTables();

break;

}

}

}

public static void createTables() throws SQLException {

SqlCon.statement.executeUpdate("CREATE TABLE IF NOT EXISTS Provider (" +

"Id\_Provider int auto\_increment primary key," +

"ProvidersName varchar(64) not null)");

SqlCon.statement.executeUpdate("CREATE TABLE IF NOT EXISTS MatOtLica (" +

"TabNomMOL int auto\_increment primary key," +

"SurnameMOL varchar(64) not null)");

SqlCon.statement.executeUpdate("CREATE TABLE IF NOT EXISTS Materials (" +

"NomenklNomMat int auto\_increment primary key," +

"NameMat varchar(64) not null," +

"EdIzm varchar(24) not null," +

"Price float not null)");

SqlCon.statement.executeUpdate("CREATE TABLE IF NOT EXISTS Nakladnie (" +

"NomDok int auto\_increment primary key," +

"DataPost date not null," +

"Postavshik int not null," +

"NomenklNomMat int not null," +

"TabNomMOL int not null," +

"FOREIGN KEY(Postavshik) REFERENCES Provider(Id\_Provider)," +

"FOREIGN KEY(NomenklNomMat) REFERENCES Materials(NomenklNomMat)," +

"FOREIGN KEY(TabNomMOL) REFERENCES MatOtLica(TabNomMOL)," +

"Kolichestvo int not null)");

System.out.println("Выполнено");

}

public static void fillTables() throws SQLException {

SqlCon.statement.executeUpdate("INSERT INTO Provider (ProvidersName) value ('Gazprom')");

SqlCon.statement.executeUpdate("INSERT INTO Provider (ProvidersName) value ('Lykoil')");

SqlCon.statement.executeUpdate("INSERT INTO MatOtLica (SurnameMOL) value ('Popugai E A')");

SqlCon.statement.executeUpdate("INSERT INTO MatOtLica (SurnameMOL) value ('Boldysh S K')");

SqlCon.statement.executeUpdate("INSERT INTO Materials (NameMat,EdIzm,Price) value ('Benzin','litri',2)");

SqlCon.statement.executeUpdate("INSERT INTO Materials (NameMat,EdIzm,Price) value ('Gaz','kubi',5)");

SqlCon.statement.executeUpdate("INSERT INTO Nakladnie (DataPost,Postavshik,NomenklNomMat, TabNomMOL,Kolichestvo) value ('2016-05-05',1,1,1,10)");

SqlCon.statement.executeUpdate("INSERT INTO Nakladnie (DataPost,Postavshik,NomenklNomMat, TabNomMOL,Kolichestvo) value ('2017-06-06',2,2,2,20)");

System.out.println("Выполнено");

}

public static void printTables() throws SQLException{

System.out.println("ПОСТАВЩИКИ:");

ResultSet resultSet=SqlCon.statement.executeQuery("SELECT \* FROM Provider");

while(resultSet.next()){

System.out.println(resultSet.getString(1)+" "+resultSet.getString(2));

}

System.out.println("\n\nМОЛ:");

ResultSet resultSet1=SqlCon.statement.executeQuery("SELECT \* FROM MatOtLica");

while(resultSet1.next()){

System.out.println(resultSet1.getString(1)+" "+resultSet1.getString(2));

}

System.out.println("\n\nМАТЕРИАЛЫ:");

ResultSet resultSet2=SqlCon.statement.executeQuery("SELECT \* FROM Materials");

while(resultSet2.next()){

System.out.println(resultSet2.getString(1)+" "+resultSet2.getString(2)+" "+resultSet2.getString(3)+" "+resultSet2.getString(4));

}

System.out.println("\n\nНакладные:");

ResultSet resultSet3=SqlCon.statement.executeQuery("SELECT \* FROM Nakladnie");

while(resultSet3.next()){

System.out.println(resultSet3.getString(1)+" "+resultSet3.getString(2)+" "+resultSet3.getString(3)+" "+resultSet3.getString(4)+" "+resultSet3.getString(5)+" "+resultSet3.getString(6));

}

System.out.println("Выполнено");

}

public static void deleteTables() throws SQLException {

SqlCon.statement.executeUpdate("DROP TABLE Nakladnie");

SqlCon.statement.executeUpdate("DROP TABLE Provider");

SqlCon.statement.executeUpdate("DROP TABLE MatOtLica");

SqlCon.statement.executeUpdate("DROP TABLE Materials");

System.out.println("Выполнено");

}

}

package com.company;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class SqlCon {

public static final String NAME\_USER="Egor";

public static final String PASSWORD="qazplmwsxokn134";

public static final String URL="http://194.87.210.23/";

public static Connection connection;

public static Statement statement;

static {

try{

connection= DriverManager.getConnection(URL,NAME\_USER,PASSWORD);

}catch (SQLException e){

e.printStackTrace();

throw new RuntimeException();

}

}

static {

try {

statement=connection.createStatement();

}catch (SQLException e){

e.printStackTrace();

throw new RuntimeException();

}

}

}