МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

**ЛАБОРАТОРНАЯ РАБОТА №9**

По дисциплине «СПП»

Выполнил

Студент ФЭИС

3- го курса, группы ПО-5

Брич М.Н.

Проверил

Крощенко А. А.

2022

**Цель работы:** приобрести практические навыки разработки баз данных и начальной интеграции БД с кодом Java с помощью JDBC.

**Вариант:** 2

База данных «Расписание занятий на факультете».

**Задание:**

Реализовать базу данных из не менее 5 таблиц на заданную тематику. При реализации продумать типизацию полей и внешние ключи в таблицах. Визуализировать разработанную БД с помощью схемы, на которой отображены все таблицы и связи между ними (пример, схема на рис. 1). На языке Java с использованием JDBC реализовать подключение к БД и выполнить основные типы запросов, продемонстрировать результаты преподавателю и включить тексты составленных запросов в отчет. Основные типы запросов – 1. На выборку/на выборку с упорядочиванием (SELECT); 2. На добавление (INSERT INTO); 3. На удаление (DELETE FROM); 4. На модификацию (UPDATE). Базу данные можно реализовать в любой СУБД (MySQL, PostgreSQL, SQLite и др.)

**Код программы:**

**Код создания таблиц в MySQL**

USE `spp\_lab9`;

CREATE TABLE `calendar` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`SemesterID` int(11) NOT NULL,

`WeekDay` int(11) NOT NULL,

`LessonID` int(11) NOT NULL,

`LessonTime` varchar(50) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

INSERT INTO `calendar` (`ID`, `SemesterID`, `WeekDay`, `LessonID`, `LessonTime`) VALUES

(1, 6, 4, 1, '12:00 - 13:20'),

(2, 6, 4, 2, '13:40 - 15:10'),

(3, 6, 1, 1, '12:00 - 13:20');

CREATE TABLE `groups` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`GroupName` varchar(50) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=19 DEFAULT CHARSET=latin1;

INSERT INTO `groups` (`ID`, `GroupName`) VALUES

(1, 'PO-5'),

(2, 'PO-4');

CREATE TABLE `lecturers` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`FirstName` varchar(50) NOT NULL,

`LastName` varchar(50) NOT NULL,

`Patronymic` varchar(50) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

INSERT INTO `lecturers` (`ID`, `FirstName`, `LastName`, `Patronymic`) VALUES

(1, 'Ivan', 'Gladkiu', 'Ivanovich'),

(2, 'Vladimir', 'Lenin', 'Ilich'),

(3, 'Vladimir', 'Putin', 'Vladimirovich');

CREATE TABLE `subjects` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`SubjectName` varchar(50) NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

INSERT INTO `subjects` (`ID`, `SubjectName`) VALUES

(1, 'Math'),

(2, 'SPP'),

(3, 'Physics');

CREATE TABLE `timetable` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`GroupID` int(11) DEFAULT NULL,

`SubjectID` int(11) DEFAULT NULL,

`LecturerID` int(11) DEFAULT NULL,

`WeekDay` int(11) DEFAULT NULL,

`LessonID` int(11) DEFAULT NULL,

PRIMARY KEY (`ID`),

KEY `GroupID` (`GroupID`),

KEY `LecturerID` (`LecturerID`),

KEY `SubjectID` (`SubjectID`),

KEY `WeekDay` (`WeekDay`),

KEY `timetable\_ibfk\_4` (`LessonID`),

CONSTRAINT `timetable\_ibfk\_1` FOREIGN KEY (`GroupID`) REFERENCES `groups` (`ID`),

CONSTRAINT `timetable\_ibfk\_2` FOREIGN KEY (`LecturerID`) REFERENCES `lecturers` (`ID`),

CONSTRAINT `timetable\_ibfk\_3` FOREIGN KEY (`SubjectID`) REFERENCES `subjects` (`ID`),

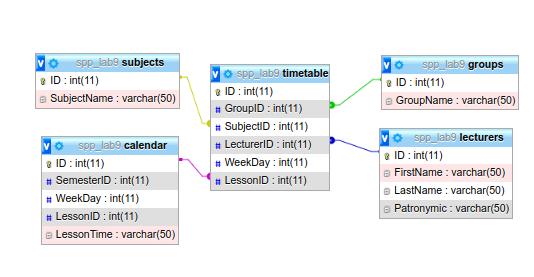
CONSTRAINT `timetable\_ibfk\_4` FOREIGN KEY (`LessonID`) REFERENCES `calendar` (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

INSERT INTO `timetable` (`ID`, `GroupID`, `SubjectID`, `LecturerID`, `WeekDay`, `LessonID`) VALUES

(2, 1, 1, 1, 4, 1),

(3, 2, 3, 3, 4, 2);



**Код HelloApplication.java**

package com.example.lab9;

import dnl.utils.text.table.TextTable;

import javax.swing.event.TableModelListener;

import javax.swing.table.TableModel;

import java.sql.\*;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class HelloApplication {

public static void main(String[] args) {

List<String> GroupName = new ArrayList<>();

List<String> SubjectName = new ArrayList<>();

List<String> FirstName = new ArrayList<>();

List<String> LastName = new ArrayList<>();

List<String> LessonTime = new ArrayList<>();

List<String> LessonID = new ArrayList<>();

List<List<String>> timetable = new ArrayList<>();

List<String> timetableName = Arrays.asList("GroupName", "SubjectName", "FirstName", "LastName", "LessonTime", "LessonID");

int count = 1;

String connectionUrl = "jdbc:mysql://localhost:3306/spp\_lab9";

try (Connection con = DriverManager.getConnection(connectionUrl, "root", "root")) {

if (con.isValid(30)) {

System.out.println("Success connection");

Statement stmt = con.createStatement();

System.out.println("\n SQL query: SELECT \* FROM timetable");

String selectThursday = "SELECT t.ID, g.GroupName, sub.SubjectName, l.FirstName, l.LastName, c.LessonTime, c.LessonID " +

"FROM timetable t " +

" INNER JOIN groups g ON t.GroupID = g.ID " +

" INNER JOIN subjects sub ON t.SubjectID = sub.ID " +

" INNER JOIN lecturers l ON t.LecturerID = l.ID " +

" INNER JOIN calendar c ON t.LessonID = c.ID " +

"WHERE t.WeekDay = 4 " +

" AND g.GroupName = 'PO-5' " +

"ORDER BY t.LessonID";

ResultSet rs = stmt.executeQuery(selectThursday);

while (rs.next()) {

GroupName.add(rs.getString("GroupName"));

SubjectName.add(rs.getString("SubjectName"));

FirstName.add(rs.getString("FirstName"));

LastName.add(rs.getString("LastName"));

LessonTime.add(rs.getString("LessonTime"));

LessonID.add(rs.getString("LessonID"));

timetable.add(GroupName);

timetable.add(SubjectName);

timetable.add(FirstName);

timetable.add(LastName);

timetable.add(LessonTime);

timetable.add(LessonID);

}

TableModel tm = new TableModel() {

@Override

public int getRowCount() {

return timetable.get(0).size();

}

@Override

public int getColumnCount() {

return timetable.size();

}

@Override

public String getColumnName(int columnIndex) {

return timetableName.get(columnIndex);

}

@Override

public Class<?> getColumnClass(int columnIndex) {

return timetable.getClass();

}

@Override

public boolean isCellEditable(int rowIndex, int columnIndex) {

return timetable.get(columnIndex) != null && timetable.get(columnIndex).get(rowIndex) != null;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

return timetable.get(columnIndex).get(rowIndex);

}

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

}

@Override

public void addTableModelListener(TableModelListener l) {

}

@Override

public void removeTableModelListener(TableModelListener l) {

}

};

TextTable tt = new TextTable(tm);

tt.printTable();

System.out.println("\n SQL query: INSERT INTO groups (`GroupName`) VALUES ('TEST')");

String addingTestGroup = "INSERT INTO groups (`GroupName`) VALUES ('TEST') ";

stmt.execute(addingTestGroup);

String readingTestGroup = "SELECT \* FROM groups";

ResultSet testGroupFirst = stmt.executeQuery(readingTestGroup);

System.out.println("\n Table: groups");

while (testGroupFirst.next()) {

System.out.println(count++ + ". " + testGroupFirst.getString("GroupName"));

}

System.out.println("\n SQL query: UPDATE groups SET GroupName='NORD' WHERE GroupName='TEST'");

String updatingTestGroup = "UPDATE groups SET GroupName='NORD' WHERE GroupName='TEST'";

stmt.execute(updatingTestGroup);

ResultSet testGroupSecond = stmt.executeQuery(readingTestGroup);

System.out.println("\n Table: groups");

count = 1;

while (testGroupSecond.next()) {

System.out.println(count++ + ". " + testGroupSecond.getString("GroupName"));

}

System.out.println("\n SQL query: DELETE FROM groups WHERE GroupName='NORD'");

String deletingTestGroup = "DELETE FROM groups WHERE GroupName='NORD'";

stmt.execute(deletingTestGroup);

ResultSet testGroupThird = stmt.executeQuery(readingTestGroup);

System.out.println("\n Table: groups");

count = 1;

while (testGroupThird.next()) {

System.out.println(count++ + ". " + testGroupThird.getString("GroupName"));

}

} else {

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

}

} catch (SQLException e) {

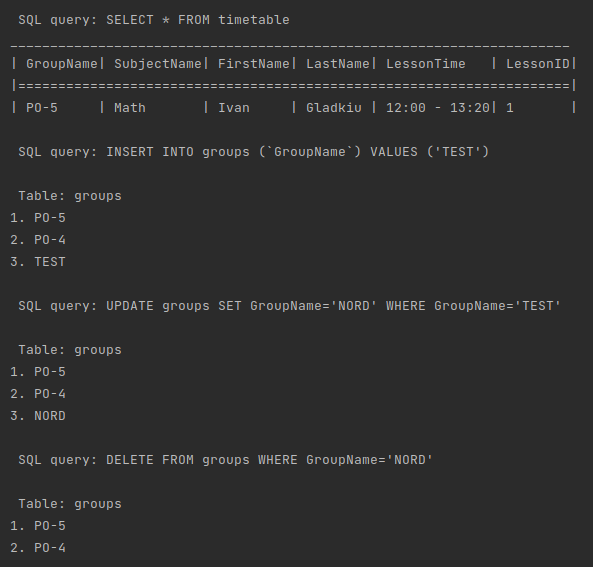
e.printStackTrace();

}

}

}

**Результаты работы:**



**Выводы:** в ходе выполнения лабораторной работы были получены практические навыки разработки баз данных и начальной интеграции БД с кодом Java с помощью JDBC.