#### Министерство образования и науки Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №2

по дисциплине   
«Распределённые программные системы»

Тема: «Работа с реляционной СУБД средствами JDBC API»

Выполнил:

Ст.гр.ПРИ-116

Чернова Д.В.

Принял:

Проскурина Г.В.

Владимир, 2018 г.

ЦЕЛЬ РАБОТЫ.

Изучить средства JDBC API для работы с реляционными СУБД и получить навыки их практического использования.

ХОД РАБОТЫ.

В результате выполнения лабораторной работы было создано две таблицы, связанные друг с другом, а также несколько классов для обеспечения работы с таблицами. Операции включают в себя создание базы данных, создание таблиц, заполнение базы данных и основные операции с данными.

Ниже приведены исходные коды и скриншоты, показывающие работоспособность программы.

Класс ConnectionClass.java  
**package** ru.shvartz.lab2;  
**package** ru.shvartz.lab2;  
  
**import** java.sql.\*;  
  
  
**public class** ConnectionClass {  
  
 **public static void** main(String[] args) **throws** SQLException {  
 *workingSQL*();  
 }  
  
 **public static void** workingSQL() **throws** SQLException {  
 Connection connection = *getDBConnection*();  
 **try** {  
 sqlCommands commands = **new** sqlCommands();  
 sqlTasks.*doCommand*(connection, commands.getDropDatabaseString());  
 System.***out***.println(**"database was dropped"**);  
 sqlTasks.*doCommand*(connection, commands.getCreateDataBaseString());  
 System.***out***.println(**"database was created"**);  
 sqlTasks.*doCommand*(connection, commands.getUseDatabase());  
 sqlTasks.*doCommand*(connection, commands.getCreateCourse());  
 sqlTasks.*doCommand*(connection, commands.getHack());  
 sqlTasks.*doCommand*(connection, commands.getCreateUser());  
 sqlTasks.*doCommand*(connection, commands.getAlterTable());  
 sqlTasks.*doCommand*(connection, commands.getDeleteAllFromUser());  
 sqlTasks.*doCommand*(connection, commands.getDeleteAllFromCourse());  
 sqlTasks.*doCommand*(connection, commands.getInsertTableUser());  
 sqlTasks.*doCommand*(connection, commands.getInsertTableCourse());  
  
 sqlTasks.*select*(connection,**"user"**);  
  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 } **finally** {  
 connection.close();  
 }  
 }  
  
 **public static** Connection getDBConnection() {  
 ConnectionModel connectionModel = **new** ConnectionModel(**"localhost"**, 3306, **"lab2"**, **"root"**, **"AkwcEcsE"**);  
 Connection connection = **null**;  
 **try** {  
 Class.*forName*(connectionModel.getDriverName());  
 } **catch** (ClassNotFoundException e) {  
 System.***out***.println(e.getMessage());  
 }  
 **try** {  
 connection = DriverManager.*getConnection*(connectionModel.getUrl(), connectionModel.getUser(), connectionModel.getPassword());  
 **return** connection;  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 }  
 **return** connection;  
 }  
  
}

Класс sqlTasks.java

**package** ru.shvartz.lab2;  
  
**import** java.sql.\*;  
  
**public class** sqlTasks {  
  
 **public static void** doCommand(Connection connection, String command) {  
 **try** {  
 connection.prepareStatement(command).execute();  
 } **catch** (SQLException e) {  
 System.***out***.println(e.getMessage());  
 }  
 }  
  
 **public static void** select(Connection connection, String tableName) **throws** SQLException{  
 **try** {  
 PreparedStatement preparedStatement = connection.prepareStatement((**"select \* from "**).concat(tableName + **";"**));  
 ResultSet res = preparedStatement.executeQuery();  
  
 **while** (res.next()) {  
 String id = res.getString(**"userId"**);  
 String name = res.getString(**"username"**);  
 System.out.println(**"id: "** + id);  
 System.out.println(**"name: "** + name);  
 }  
 } **catch** (SQLException e) {  
 System.out.println(e.getMessage());  
 }  
 }  
}

Класс sqlCommands.java

**package** ru.shvartz.lab2;  
  
**public class** sqlCommands {  
 **private** String **dropDatabaseString** = **"DROP DATABASE lab2;"**;  
 **private** String **createDataBaseString** = **"CREATE DATABASE lab2;"**;  
  
 **private** String **useDatabase** = **"use lab2;"**;  
 **private** String **hack** = **"set foreign\_key\_checks=0;"**;  
  
  
 **private** String **hackBack** = **"set foreign\_key\_checks=1;"**;  
  
 **public** String getHack() {  
 **return hack**;  
 }  
  
 **private** String **createUser** = **"CREATE TABLE USER("** + **"userId int NOT NULL, "** + **"username VARCHAR(20) NOT NULL, "** + **"password VARCHAR(20) NOT NULL, "** + **"email VARCHAR(20) NOT NULL, "** + **"courseID int NOT NULL, "** + **"createdDate date not null,"** + **"PRIMARY KEY (userId) "** + **")"**;  
 **private** String **createCourse** = **"CREATE TABLE COURSE("** + **"courseId int NOT NULL, "** + **"courseName VARCHAR(20) NOT NULL, "** + **"courseDescription VARCHAR(20) NOT NULL, "** + **"PRIMARY KEY (courseId) "** + **")"**;  
  
 **private** String **insertTableUser** = **"INSERT INTO USER VALUES"** + **"('1','usname','pass', 'mail','1', '12.08.2018'),"** +  
 **"('2','usname','pass', 'mail','1', '12.08.2018')"**;  
  
  
 **private** String **insertTableCourse** = **"INSERT INTO COURSE VALUES"** +  
 **"('1','name1','description1'),"** +  
 **"('2','name2','description2')"**;  
  
 **private** String **deleteFromTable** = **"delete from user where userId = 1"**;  
 **private** String **selectFromUser** = **"select \* from user"**;  
 **private** String **selectFromCourse** = **"select \* from course"**;  
 **private** String **updateUser** = **"UPDATE USER SET username = 'michail'"**;  
  
 **private** String **deleteAllFromUser** = **"delete from user;"**;  
 **private** String **deleteAllFromCourse** = **"delete from course;"**;  
 **private** String **alterTable** = **"alter table user add foreign key (courseID) references course(courseId);"**;  
  
 **public** String getAlterTable() {  
 **return alterTable**;  
 }  
  
 **public** String getDeleteAllFromUser() {  
 **return deleteAllFromUser**;  
 }  
  
 **public** String getDeleteAllFromCourse() {  
 **return deleteAllFromCourse**;  
 }  
  
 **public** String getDropDatabaseString() {  
 **return dropDatabaseString**;  
 }  
  
 **public** String getCreateDataBaseString() {  
 **return createDataBaseString**;  
 }  
  
 **public** String getUseDatabase() {  
 **return useDatabase**;  
 }  
  
 **public** String getCreateUser() {  
 **return createUser**;  
 }  
  
 **public** String getCreateCourse() {  
 **return createCourse**;  
 }  
  
 **public** String getInsertTableUser() {  
 **return insertTableUser**;  
 }  
  
 **public** String getDeleteFromTable() {  
 **return deleteFromTable**;  
 }  
  
 **public** String getSelectFromUser() {  
 **return selectFromUser**;  
 }  
  
 **public** String getUpdateUser() {  
 **return updateUser**;  
 }  
 **public** String getInsertTableCourse() {  
 **return insertTableCourse**;  
 }  
  
 **public** String getSelectFromCourse() {  
 **return selectFromCourse**;  
 }  
  
 **public** String getHackBack() {  
 **return hackBack**;  
 }  
}

Класс ConnectionModel.java

**package** ru.shvartz.lab2;  
  
**public class** ConnectionModel {  
 **private** String **host**;  
 **private int port**;  
 **private** String **database**;  
 **private** String **user**;  
 **private** String **password**;  
 **private** String **url**;  
 **private** String **driverName**;  
  
  
 **public** ConnectionModel(String host, **int** port, String database, String user, String password) {  
 **this**.**host** = host;  
 **this**.**port** = port;  
 **this**.**database** = database;  
 **this**.**user** = user;  
 **this**.**password** = password;  
 **this**.**url** = **"jdbc:mysql://"** + host + **":"** + port + **"/"** + database + **"?serverTimezone=UTC"**;  
 **this**.**driverName** = **"com.mysql.cj.jdbc.Driver"**;  
 }  
  
 **public** String getUser() {  
 **return user**;  
 }  
 **public** String getPassword() {  
 **return password**;  
 }  
 **public** String getUrl() {  
 **return url**;  
 }  
 **public** String getDriverName() {  
 **return driverName**;  
 }  
}

Результат работы программы.

На рис.1 показан вывод в консоль.

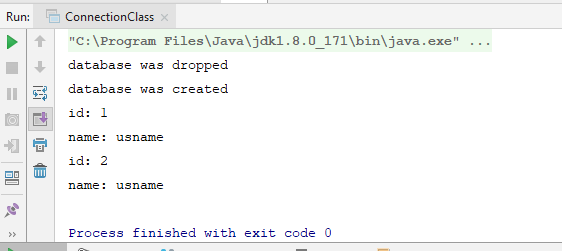


Рисунок . Вывод в консоль

На рис.2 показана созданная база данных.

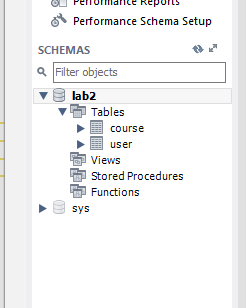


Рисунок . Созданная база данных

На рис.3 показана заполненная таблица course.

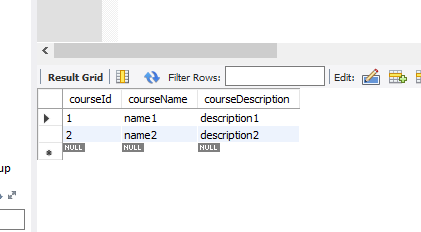


Рисунок . Таблица course

На рис.4 показана заполненная таблица user.

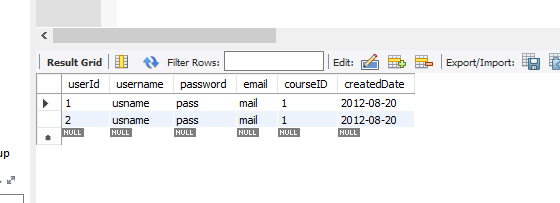


Рисунок . Таблица user

Структура созданных таблиц представлена на рис.5.

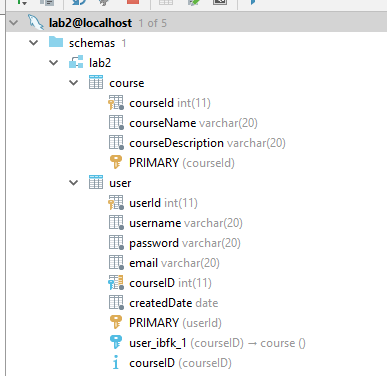


Рисунок . Структура созданных таблиц

Диаграмма, построенная средствами Intellij IDEA, представлена на рис.6.

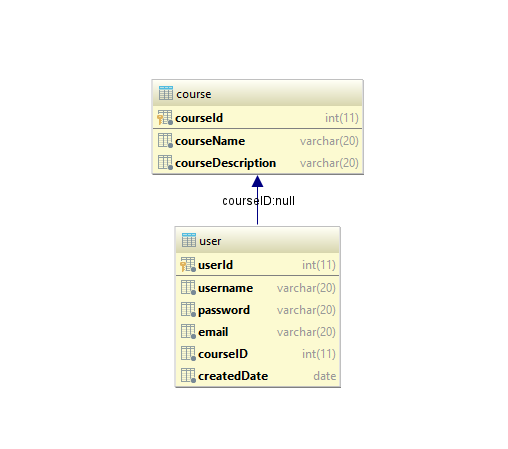


Рисунок . Диаграмма

ВЫВОДЫ.

В ходе выполнения лабораторной работы были изучены средства JDBC API для работы с реляционными СУБД и получены навыки их практического использования.