

Задание. Разработать приложение Java с подключением базы данных.

1. Разработайте приложение, которое автоматизирует одну из функций учета посещаемости студентов вуза.

Для работы с JDBC выполнить следующие шаги:

Создать объект Connection для соединения с базой данных.

Создать объект Statement под Connection для выполнения SQL-команд.

Написать и выполнить SQL-запрос через объекты Statement и Connection.

Обработать результат запроса.

Закрыть объекты Statement и Connection для освобождения ресурсов.

Операции JDBC выполняются с использованием объектов Connection, Statement и ResultSet из пакета java.sql.

Код программы:

```
public class Lab4_task1_sem2_114M {
    public static void main(String[] args) throws SQLException {

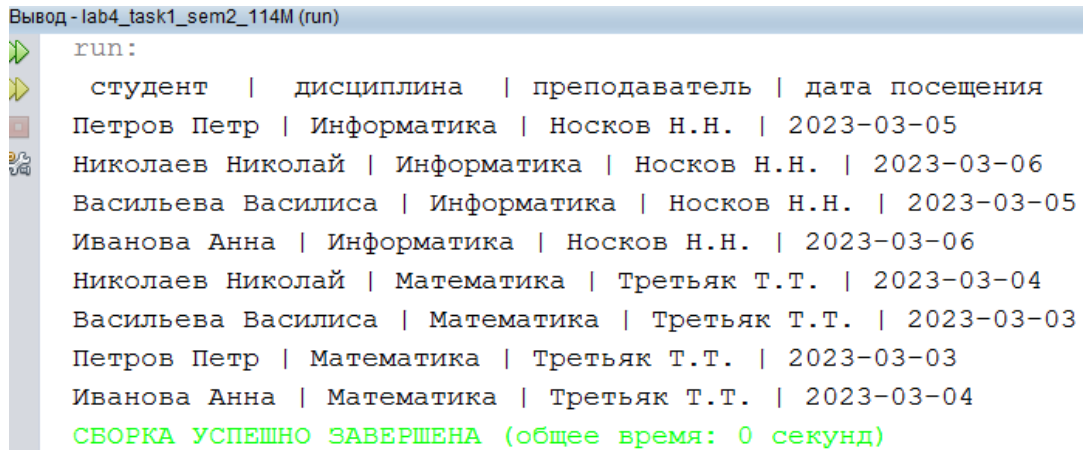
        //данные для соединения с БД
        String url =
            "jdbc:postgresql://localhost:5432/basejava";
        String user = "postgres";
        String password = "qazxsw";
        //Выделение Connection-объекта для подключения к серверу
        //базы данных
        //Операции JDBC выполняются с помощью объектов
        Connection, Statement и ResultSet
        try (Connection dbConnection =
            DriverManager.getConnection(url, user, password);
            //Выделение Statement-объекта под Connection,
            //созданный ранее для выполнения SQL-команды.
            Statement st = dbConnection.createStatement();
            //запрос sql на получение данных из таблицы
            //посещаемости
            ResultSet rs = st.executeQuery("SELECT s_name,
            d_name,t_name, a_date FROM attendance,students,
            discipline,teachers \n"
                + "WHERE a_discipline = d_id and
            a_student= s_id and t_discipline=d_id order by d_name")) {
            //вывод полученных данных из таблицы посещаемости
            System.out.println(" студент | дисциплина |
            преподаватель | дата посещения");
            while (rs.next()) {
```

```

        System.out.println(rs.getString(1) + " | " +
rs.getString(2) + " | " + rs.getString(3) + " | " +
rs.getDate(4));
    }
} catch (SQLException e) {
    System.out.println(e.getMessage());
}
}
}

```

Вывод:



```

Вывод - lab4_task1_sem2_114M (run)
run:
студент | дисциплина | преподаватель | дата посещения
Петров Петр | Информатика | Носков Н.Н. | 2023-03-05
Николаев Николай | Информатика | Носков Н.Н. | 2023-03-06
Васильева Василиса | Информатика | Носков Н.Н. | 2023-03-05
Иванова Анна | Информатика | Носков Н.Н. | 2023-03-06
Николаев Николай | Математика | Третьяк Т.Т. | 2023-03-04
Васильева Василиса | Математика | Третьяк Т.Т. | 2023-03-03
Петров Петр | Математика | Третьяк Т.Т. | 2023-03-03
Иванова Анна | Математика | Третьяк Т.Т. | 2023-03-04
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

2. Разработайте программу на языке Java без использования пользовательского интерфейса на основе сценариев:

Сведения об объектах должны храниться в таблицах базы данных, чтение и редактирование данных осуществляется при помощи запросов SQL.

Определите характеристики объектов. Обязательной характеристикой объекта является его уникальный идентификатор. Уникальность идентификаторов при выполнении операций добавления и редактирования объектов должна обеспечиваться средствами СУБД или средствами разрабатываемой программы.

Программа должна выполнять следующие операции с данными: а) добавление нового объекта; б) изменение параметров существующего объекта; в) удаление объекта; г) поиск объектов по заданным критериям и вывод информации об объектах.

Код программы:

```

public class Lab4_task2_sem2_114M {
    //данные для соединения с БД
    static String connection =
"jdbc:postgresql://localhost:5432/basejava";
    static String user = "postgres";
    static String password = "qazxsw";
    public static void main(String[] args) {

```

```

        try { //вызов метода вывода данных из таблицы
успеваемости
            SelectTable();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        //метод соединение с БД
        private static Connection getDBConnection() {
            Connection dbConnection = null;
            try { //JDBC драйвер обеспечивает соединение с базой
данных
                dbConnection =
DriverManager.getConnection(connection, user, password);
                return dbConnection;
            } catch (SQLException e) {
                System.out.println(e.getMessage());
            }
            return dbConnection;
        }
        private static void SelectTable() throws SQLException {
            Connection dbConnection = null; //соединение
            Statement statement = null; //создание запросов
            //sql запрос получения всех данных из табл.
            String selectTableSQL = "SELECT a_id, s_name,
д_name,t_name, a_date FROM attendance,students,
discipline,teachers \n"
                + "WHERE a_discipline = d_id and
a_student= s_id and t_discipline=d_id order by a_id";
            //sql запрос добавления объекта
            String insertTableSQL = "INSERT INTO attendance (a_id,
a_student, a_discipline,a_date) VALUES (5, 3, 1, '1.04.2023')";
            //sql запрос изменения объекта
            String updateTableSQL = "UPDATE attendance SET a_student
= 4 WHERE a_id = 2";
            //sql запрос удаления объекта
            String deleteTableSQL = "DELETE FROM attendance WHERE
a_id=1";
            //sql запрос поиска по параметру
            String searchTableSQL = "SELECT s_name, d_name, a_date
FROM attendance,students,discipline \n"
                + "WHERE a_discipline = d_id and
a_student= s_id and d_name='Математика'";
            try {
                dbConnection = getDBConnection(); //метод соединения
с БД
                //создание объекта для отправки инструкций SQL в
базу данных
                statement = dbConnection.createStatement();
                //метод для выполнения команд SQL добавления
                statement.executeUpdate(insertTableSQL);
                //метод для выполнения команд SQL изменения
                statement.executeUpdate(updateTableSQL);
            }
        }
    }
}

```

```

        //метод для выполнения команд SQL удаления
        statement.executeUpdate(deleteTableSQL);
        //результаты запроса выполнения команды SELECT
        ResultSet rs =
statement.executeQuery(selectTableSQL);
        //вывод полученных данных из таблицы посещаемости
        System.out.println("№| студент | дисциплина |
преподаватель | дата посещения");
        while (rs.next()) {
            System.out.println(rs.getInt(1)+ " | " +
rs.getString(2) + " | " + rs.getString(3) + " | " +
rs.getString(4) + " | " + rs.getDate(5));
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
    dbConnection.close();
}
}

```

Вывод:

Изначально в таблице успеваемости были данные о четырех студентах

Вывод - Lab4_task2_sem2_114M (run)

```

run:
№| студент | дисциплина | преподаватель | дата посещения
1 | Петров Петр | Математика | Третьяк Т.Т. | 2023-03-03
2 | Васильева Василиса | Информатика | Носков Н.Н. | 2023-03-05
3 | Николаев Николай | Программирование | Морозова М.М. | 2023-04-03
4 | Иванова Анна | Электроника | Котова К.К. | 2023-04-05
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

После добавления 5й записи, изменения 2й записи, а также удаления 1й записи, таблица успеваемости стала выглядеть следующим образом

Вывод - Lab4_task2_sem2_114M (run)

```

run:
№| студент | дисциплина | преподаватель | дата посещения
2 | Иванова Анна | Информатика | Носков Н.Н. | 2023-03-05
3 | Николаев Николай | Программирование | Морозова М.М. | 2023-04-03
4 | Иванова Анна | Электроника | Котова К.К. | 2023-04-05
5 | Николаев Николай | Математика | Третьяк Т.Т. | 2023-04-01
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

Поиск по параметру предмет= “математика”

Вывод - Lab4_task2_sem2_114M (run)

```

Математику посещают:
студент | дисциплина | дата посещения
Николаев Николай | Математика | 2023-04-01
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

3. В установленной СУБД создайте новую базу данных с таблицами для хранения объектов в соответствии с заданной предметной областью.

Основные условия для выполнения задания:

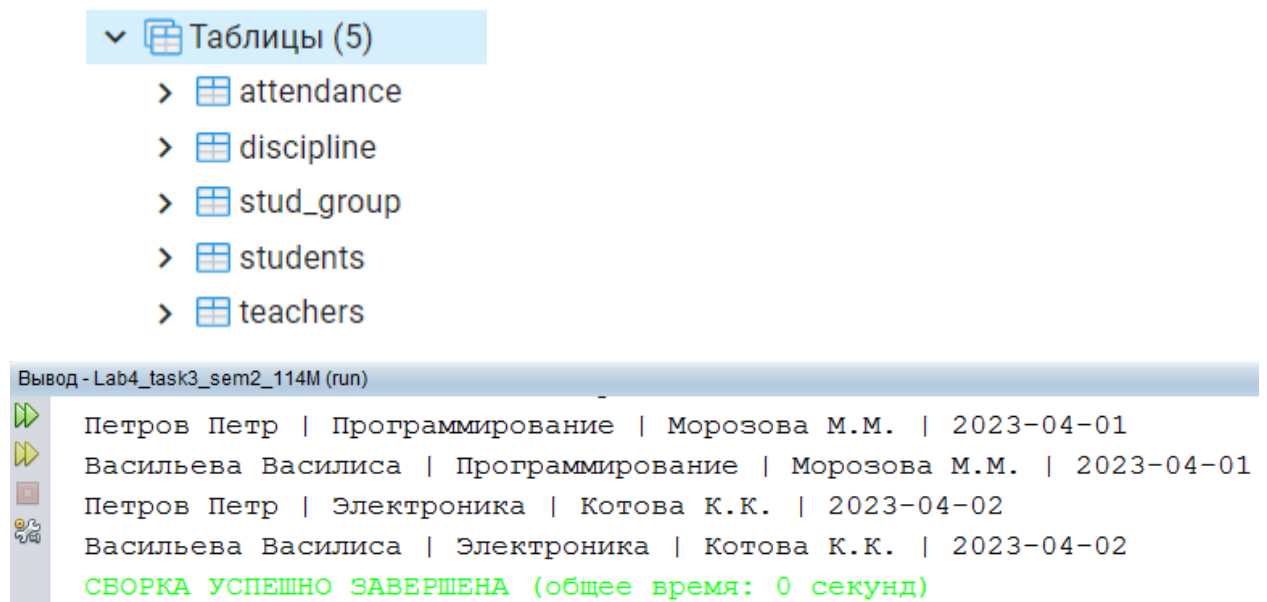
Количество правильно спроектированных реляционных таблиц должно составлять не менее 5. При выполнении задания необходимо продемонстрировать использование PreparedStatement для работы с базой данных.

Код программы:

```
public class Lab4_task3_sem2_114M {
    public static void main(String[] args) {
        //данные для соединения с БД
        String url =
            "jdbc:postgresql://localhost:5432/basejava";
        String user = "postgres";
        String password = "qazxsw";
        //запрос на получение данных из таблицы
        String selectTableSQL = "SELECT s_name, d_name, t_name,
a_date FROM attendance, students, discipline, teachers \n"
            + "WHERE a_discipline = d_id and
a_student= s_id and t_discipline=d_id order by d_name";
        //Выделение Connection-объекта для подключения к серверу
        базы данных
        //Операции JDBC выполняются с помощью объектов
        Connection, PreparedStatement и ResultSet
        try (Connection dbConnection =
            DriverManager.getConnection(url, user, password);
            //Использования PreparedStatement для
            формирования запроса
            PreparedStatement pst =
            dbConnection.prepareStatement(selectTableSQL);
            ResultSet rs = pst.executeQuery()) {
            //вывод полученных данных из таблицы посещаемости
            System.out.println(" студент | дисциплина |
преподаватель | дата посещения");
            while (rs.next()) {
                System.out.println(rs.getString(1) + " | " +
rs.getString(2) + " | " + rs.getString(3) + " | " +
rs.getDate(4));
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Вывод:

Созданные таблицы в PostgreSQL



4-5. Установите соединение с базой данных из Java-приложения. Управление соединением осуществляется через класс `java.sql.Connection`. Для установки соединения используется метод `getConnection` класса `java.sql.DriverManager`. Метод принимает на вход параметры соединения и в случае успеха возвращает объект класса `Connection`. В случае возникновения ошибки метод `getConnection` генерирует исключение `SQLException`.

Создайте запросы SQL через класс `java.sql.Statement`. Объект класса `Statement` создается в рамках заданного соединения при помощи метода `createStatement` класса `Connection`:

```
Statement s = con.createStatement();
```

Код программы:

```
public class Lab4_task4_sem2_114M {
    //данные для соединения с БД
    static String connection =
"jdbc:postgresql://localhost:5432/basejava";
    static String user = "postgres";
    static String password = "qazxsw";
    public static void main(String[] args) {
        try { //вызов метода вывода данных из таблицы
успеваемости
            SelectTable();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
    //метод соединение с БД
    private static Connection getDBConnection() {
        Connection dbConnection = null;
```

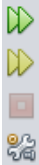
```

        try { //JDBC драйвер обеспечивает соединение с базой
данных
            dbConnection =
DriverManager.getConnection(connection, user, password);
            return dbConnection;
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        return dbConnection;
    }
    private static void SelectTable() throws SQLException {
        Connection dbConnection = null; //соединение
        Statement statement = null; //создание запросов
        //sql запрос получения всех данных из табл.
        String selectTableSQL = "SELECT s_name, d_name, t_name,
a_date FROM attendance, students, discipline, teachers \n"
            + "WHERE a_discipline = d_id and
a_student= s_id and t_discipline=d_id order by d_name";
        try {
            dbConnection = getDBConnection(); //метод соединения
с БД
            //создание объекта для отправки инструкций SQL в
базу данных
            statement = dbConnection.createStatement();
            //результаты запроса выполнения команды SELECT
            ResultSet rs =
statement.executeQuery(selectTableSQL);
            //вывод полученных данных из таблицы посещаемости
            System.out.println(" студент | дисциплина |
преподаватель | дата посещения");
            while (rs.next()) {
                System.out.println(rs.getString(1) + " | " +
rs.getString(2) + " | " + rs.getString(3) + " | " +
rs.getDate(4));
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        dbConnection.close();
    }
}

```

Вывод:

Вывод - Lab4_task4_sem2_114M (run)



run:

студент	дисциплина	преподаватель	дата посещения
Петров Петр	Информатика	Носков Н.Н.	2023-03-05
Николаев Николай	Информатика	Носков Н.Н.	2023-03-06
Васильева Василиса	Информатика	Носков Н.Н.	2023-03-05
Иванова Анна	Информатика	Носков Н.Н.	2023-03-06
Петров Петр	Математика	Третьяк Т.Т.	2023-03-03
Васильева Василиса	Математика	Третьяк Т.Т.	2023-03-03