**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Базы данных»**

Тема: Реализация базы данных с использованием ORM

|  |  |  |
| --- | --- | --- |
| Студент гр. 1303 |  | Кузнецов Н.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**Цель работы.**

Создание базы данных с использованием Object-Relational Mapping (ORM).

**Текст задания**

Вариант 6

* Описать в виде моделей Sequelize таблицы из 1-й лабораторной работы
* Написать скрипт заполнения тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
* Написать запросы к БД, отвечающие на вопросы из 1-й лабораторной работы с использованием ORM. Вывести результаты в консоль (или иной человек-читабельный вывод)
* Запушить в репозиторий исходный код проекта, соблюсти .gitignore, убрать исходную базу из проекта (или иные нагенерированные данные бд если они есть).
* Описать процесс запуска: команды, зависимости
* В отчёте описать цель, текст задания в соответствии с вариантом, выбранную ORM, инструкцию по запуску, скриншоты (код) моделей ORM, скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ), ссылку на PR в приложении, вывод

**Выполнение работы**

Для установки необходимых библиотек использовались следующие команды:

npm install sequelize

npm install pg

В файле index.js происходит настройка соединения с базой данных:

const sequelize = new Sequelize('university', 'postgres', '1', {  
 host: 'localhost',  
 dialect: 'postgres',  
 define: {  
 allowNull: false,  
 },  
});

Здесь создается объект sequelize, который представляет собой соединение с базой данных. В конструкторе Sequelize передаются следующие параметры:

* university: Название базы данных.
* postgres: Имя пользователя базы данных.
* 1: Пароль пользователя базы данных.
* { host: 'localhost', dialect: 'postgres' }: Дополнительные опции, указывающие хост (localhost) и используемый диалект (PostgreSQL).

Реализованные модели:

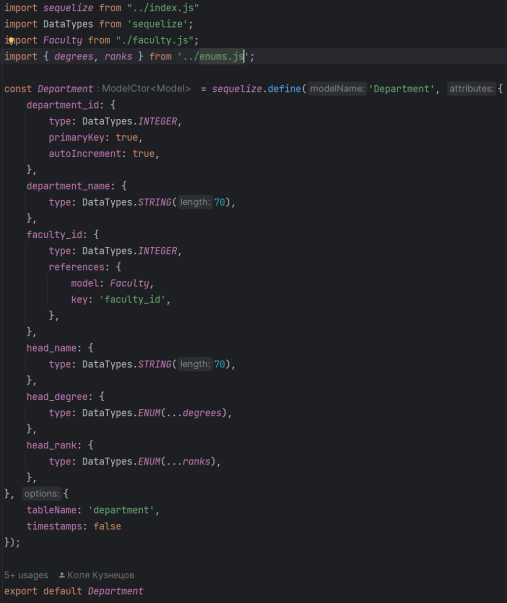


Рисунок 1. – Структура созданной БД.

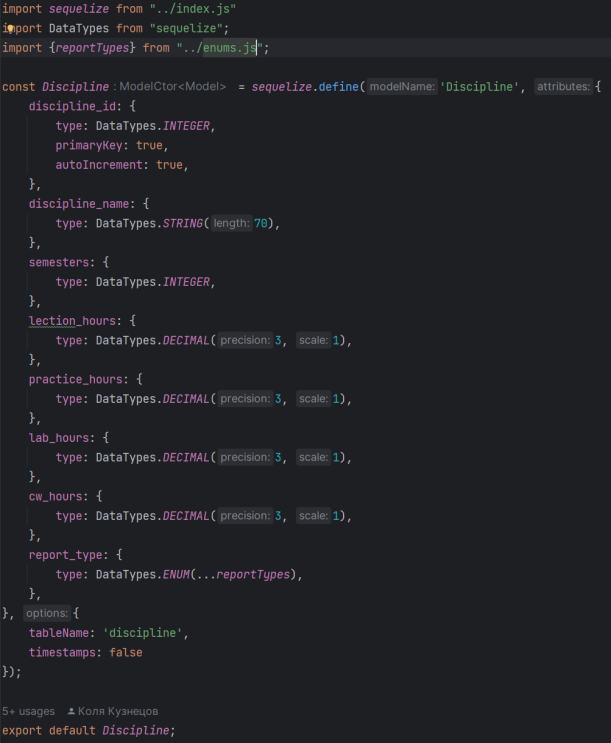


Рисунок 2. – Структура созданной БД.

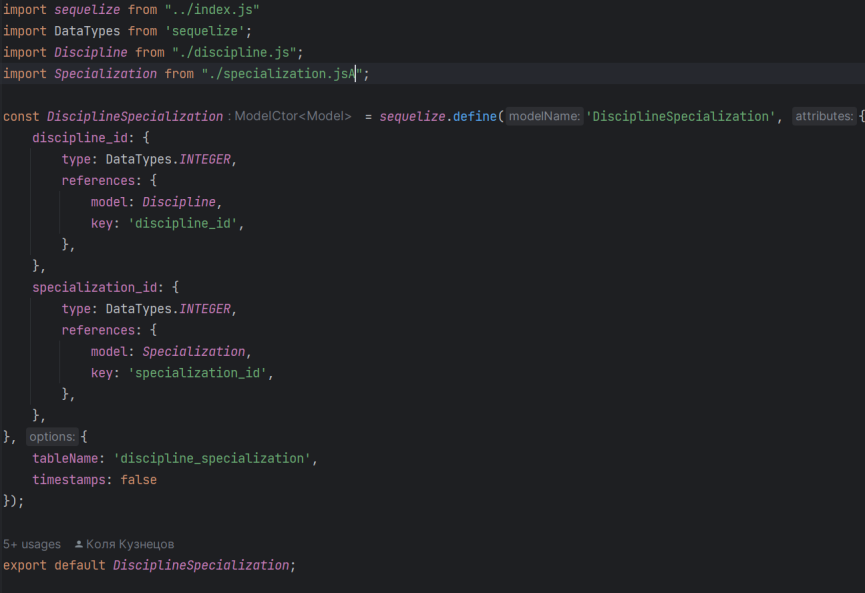


Рисунок 3. – Структура созданной БД.



Рисунок 4. – Структура созданной БД.

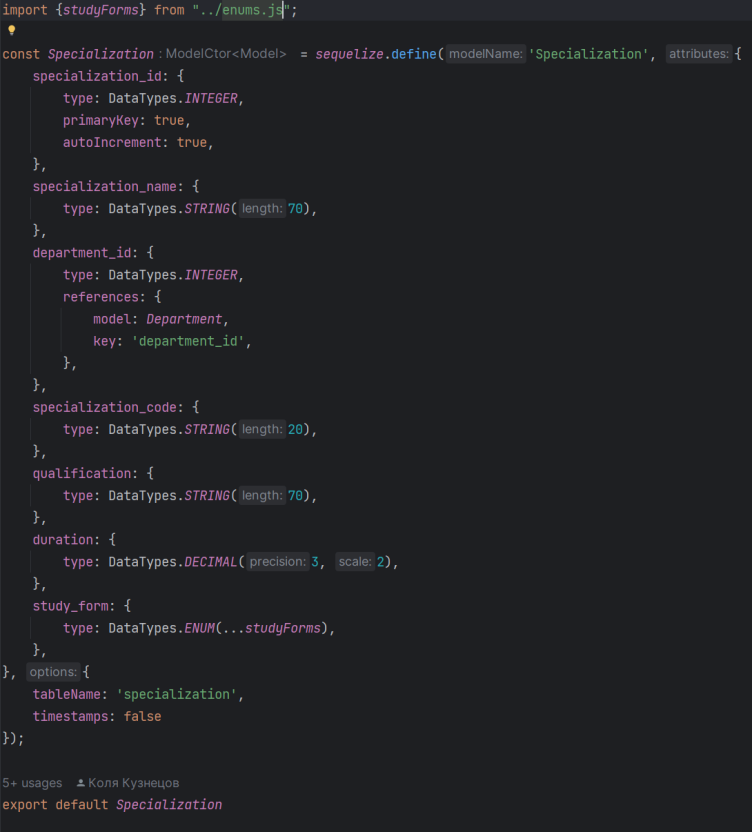


Рисунок 5. – Структура созданной БД.

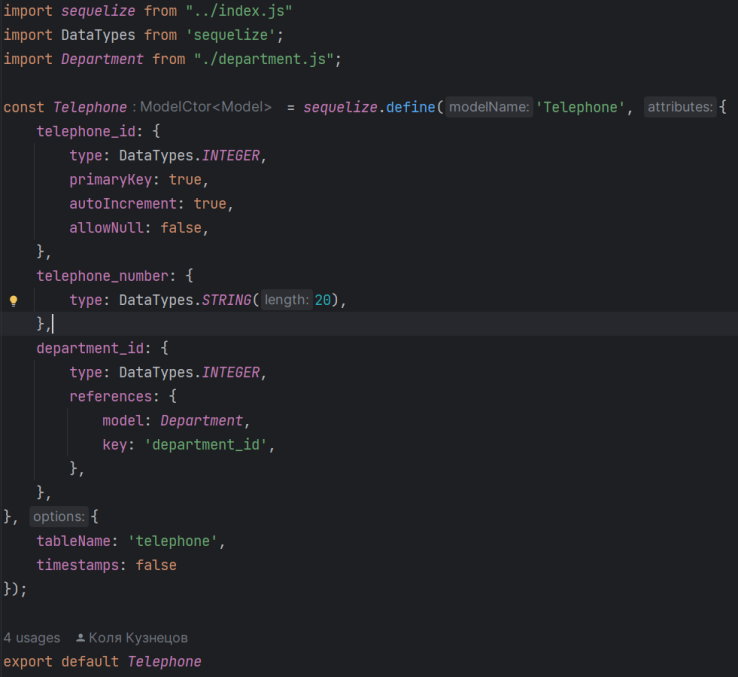


Рисунок 6. – Структура созданной БД.

Выполним запросы к БД предложенные вариантом:

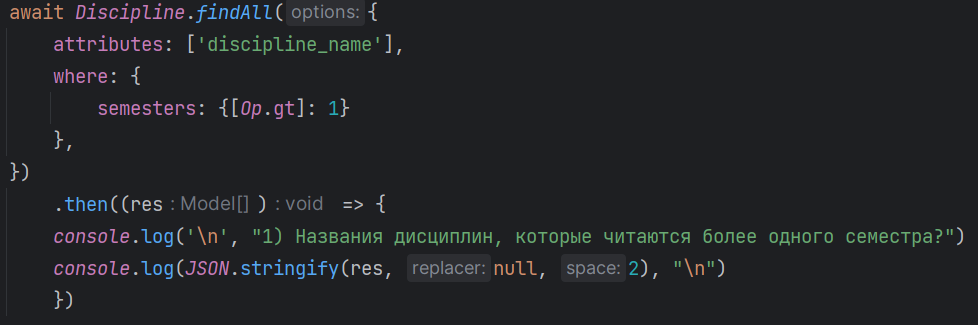


Рисунок 7. – Запрос 1.

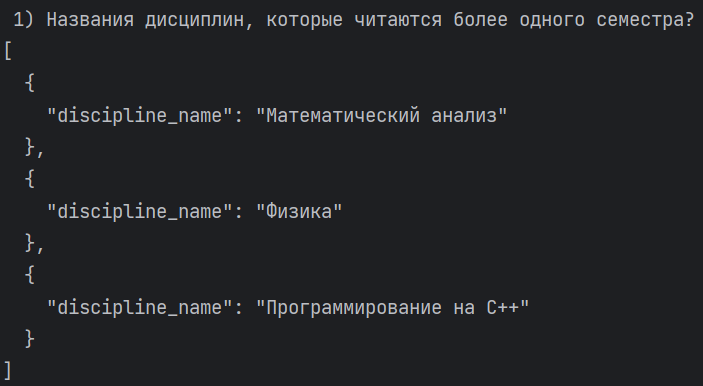


Рисунок 7. – Результат запроса 1.

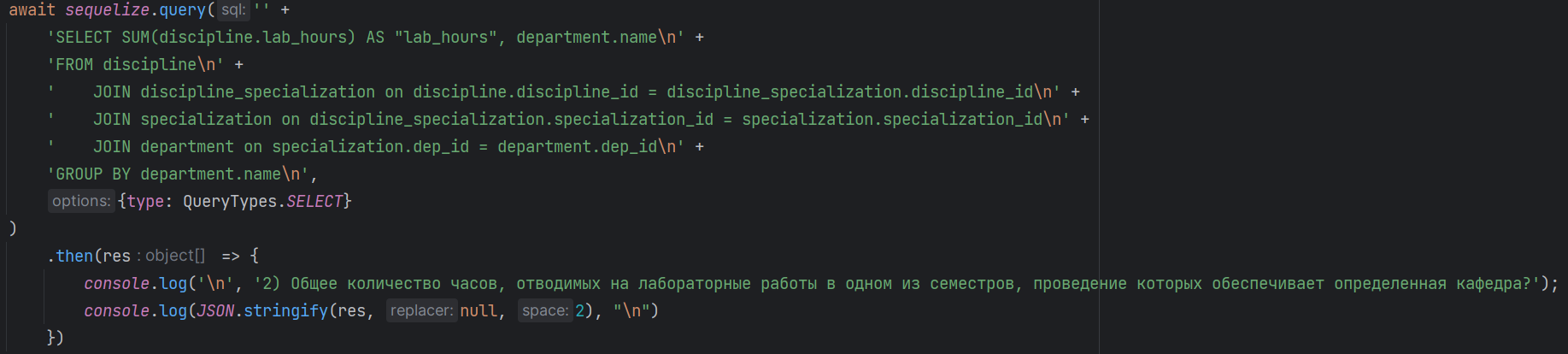


Рисунок 8 – Запрос 2.

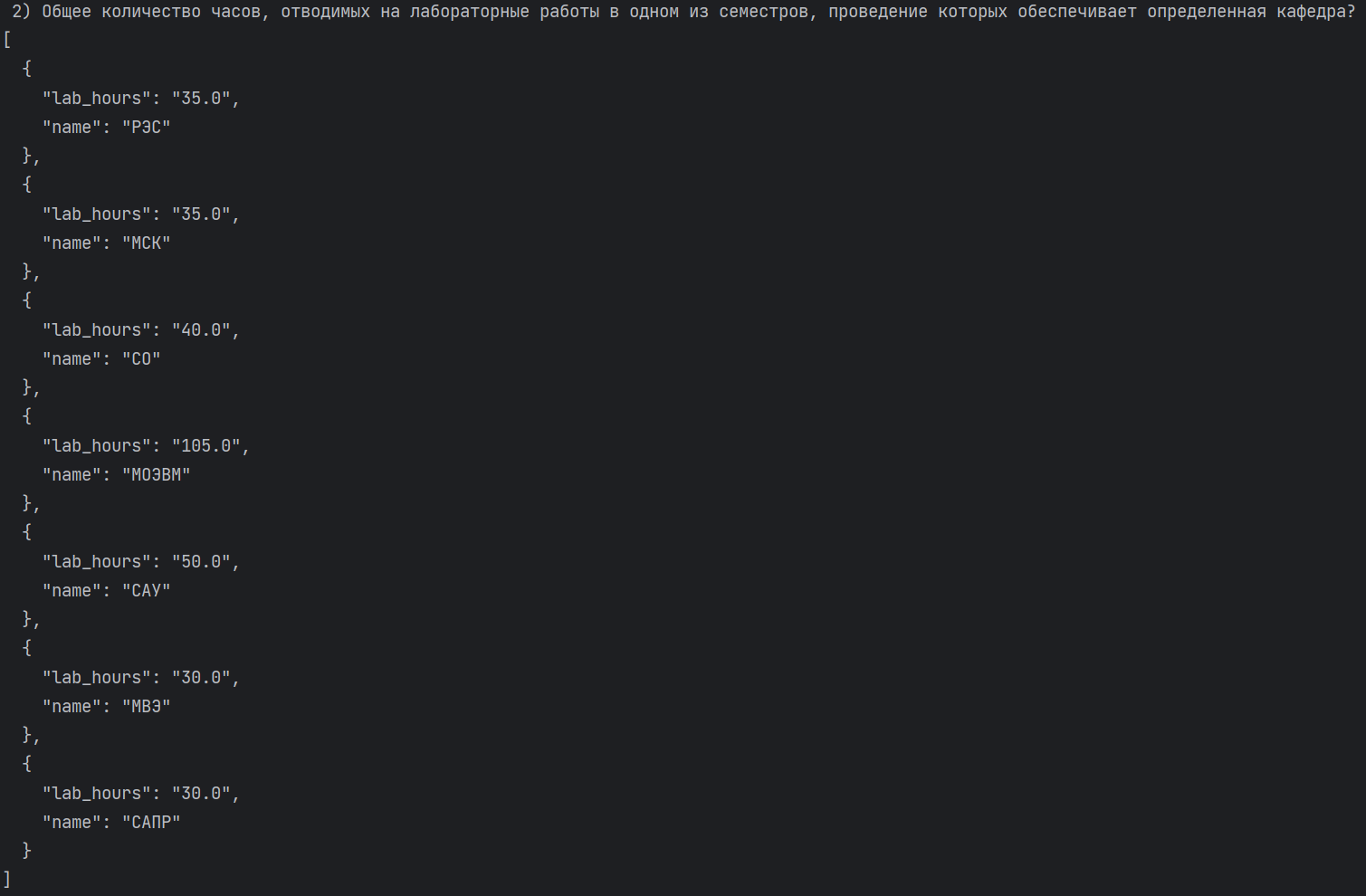


Рисунок 8 – Результат запроса 2.



Рисунок 9 – Запрос 3.

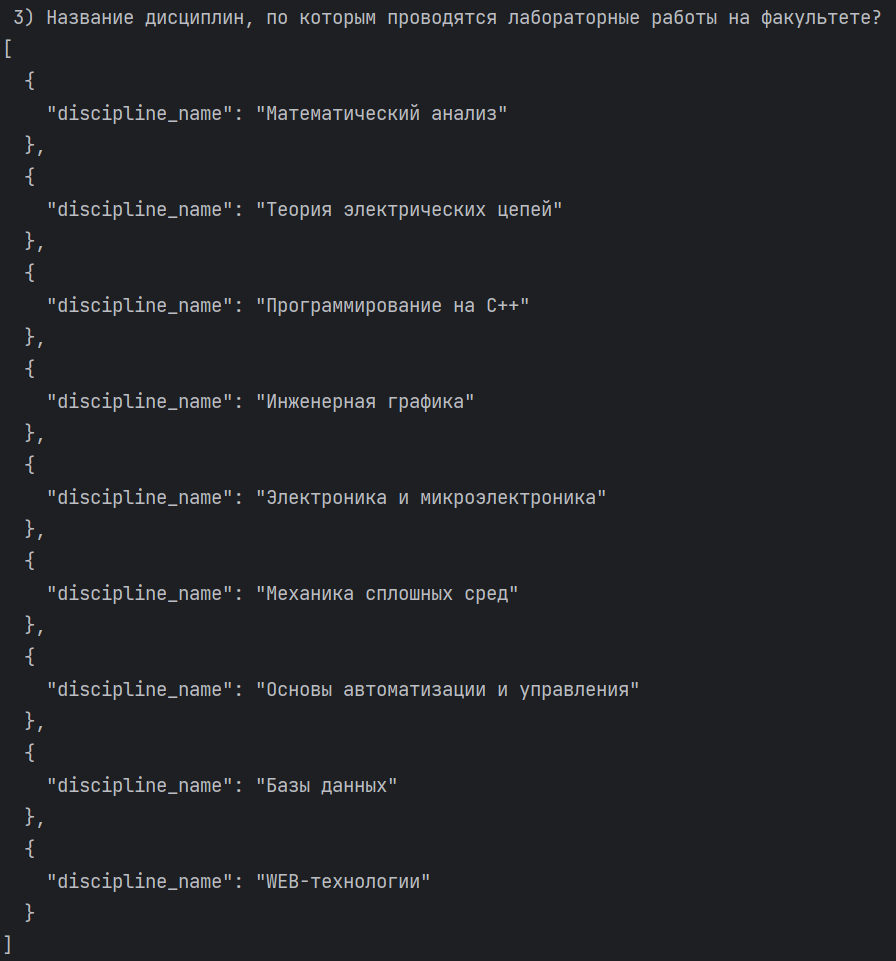


Рисунок 9 – Результат запроса 3.



Рисунок 10 – Запрос 4.

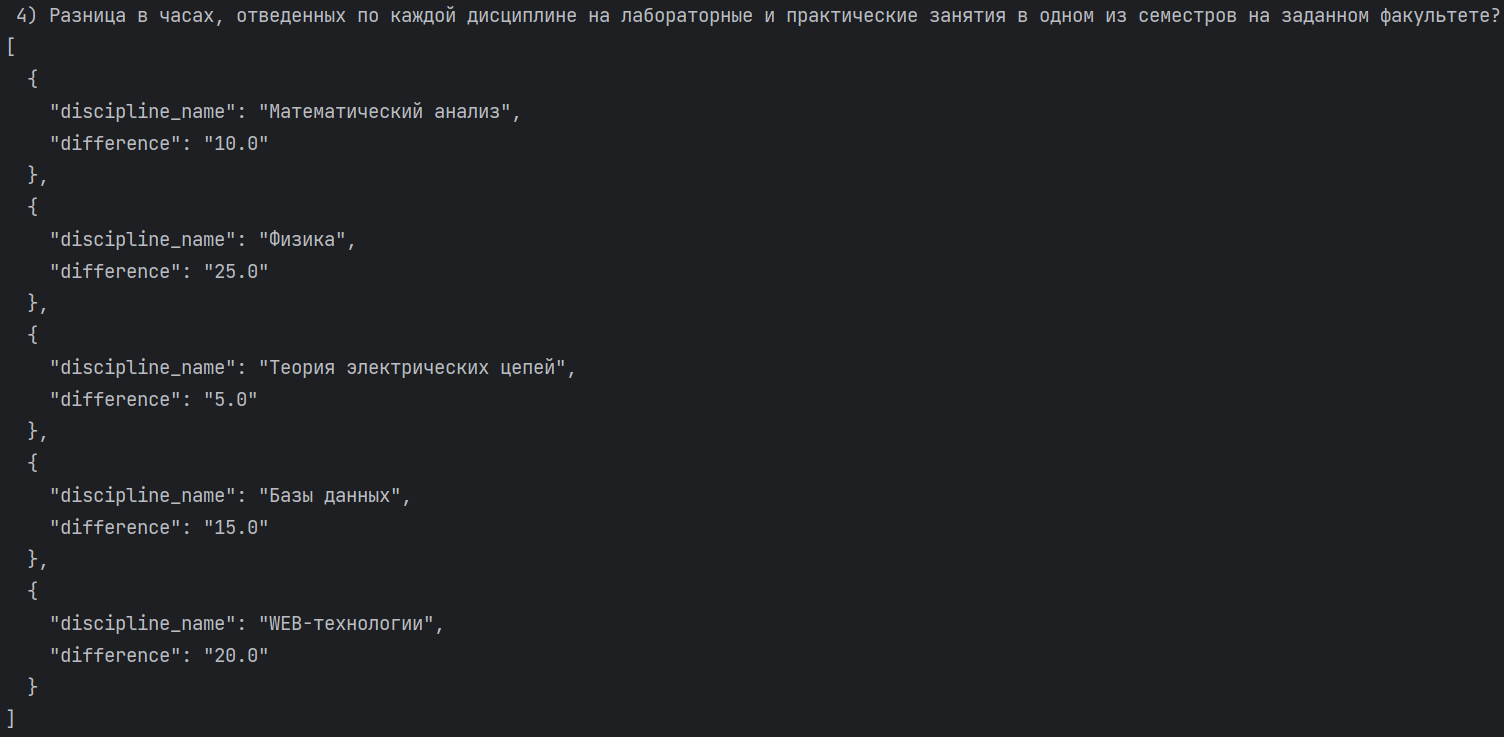


Рисунок 10 – Результат запроса 4.



Рисунок 11 – Запрос 5.

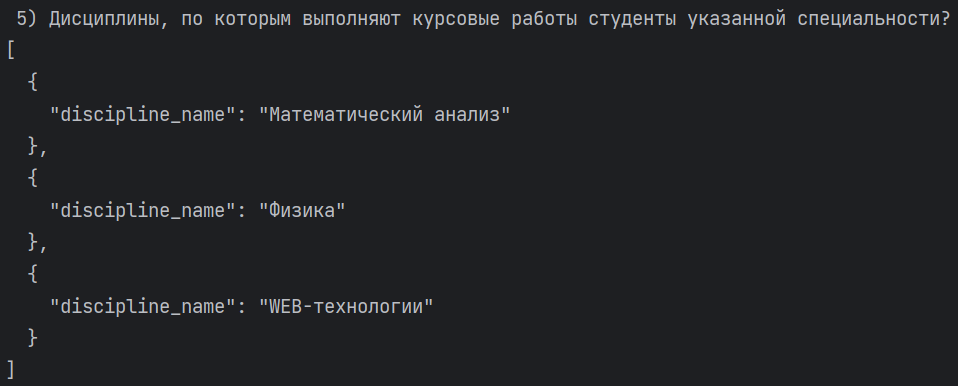


Рисунок 12 – Результат запроса 5.



Рисунок 13 – Запрос 6.

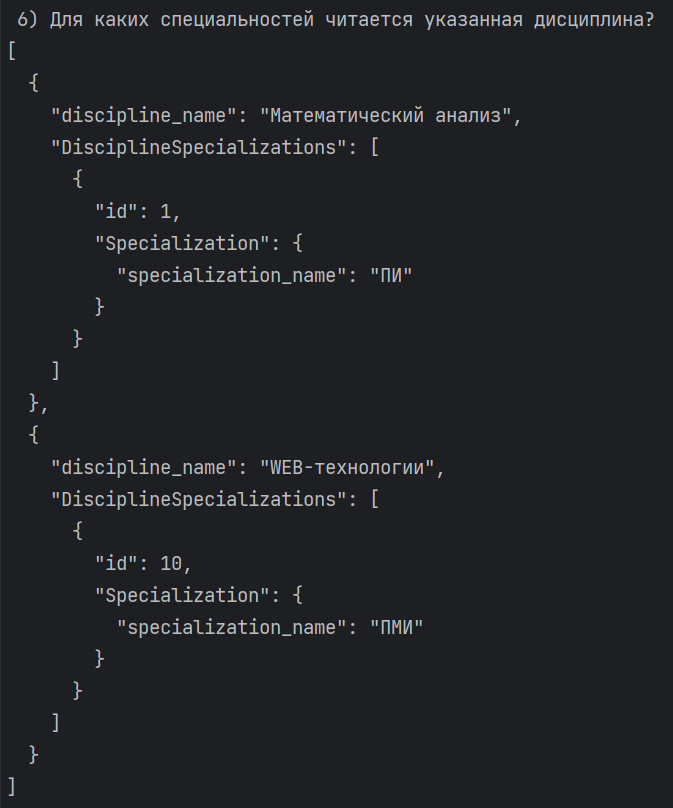


Рисунок 14 – Результат запроса 6.

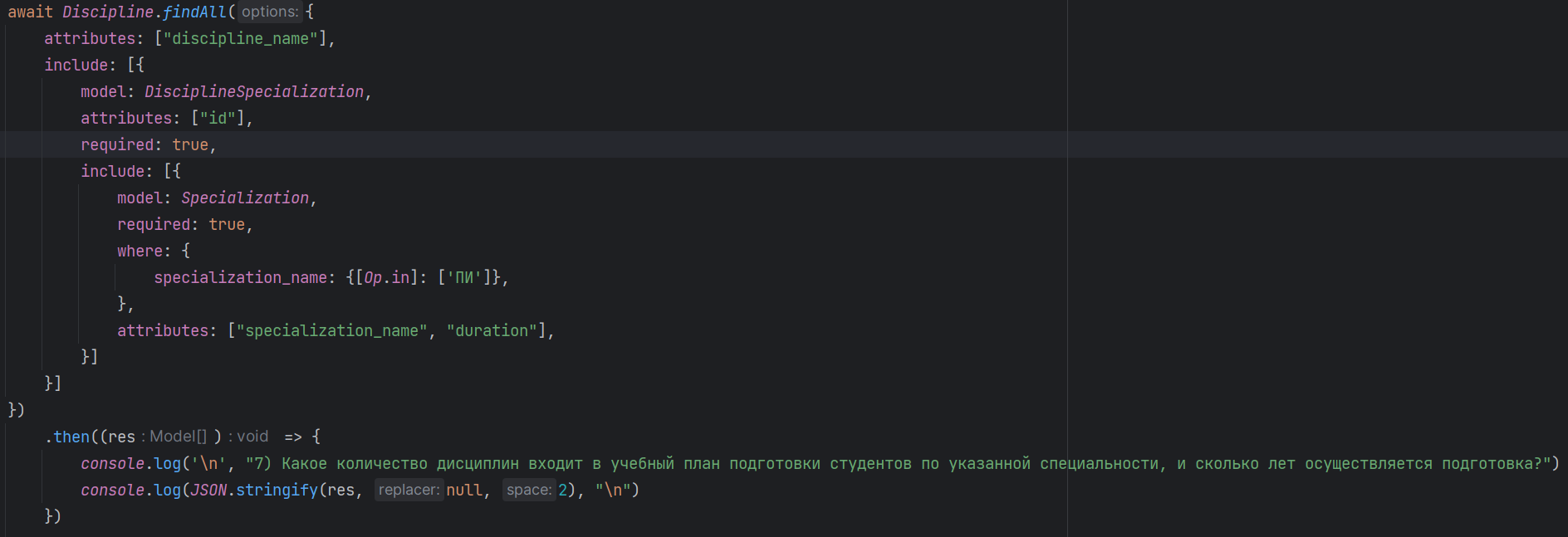


Рисунок 15 – Запрос 7.

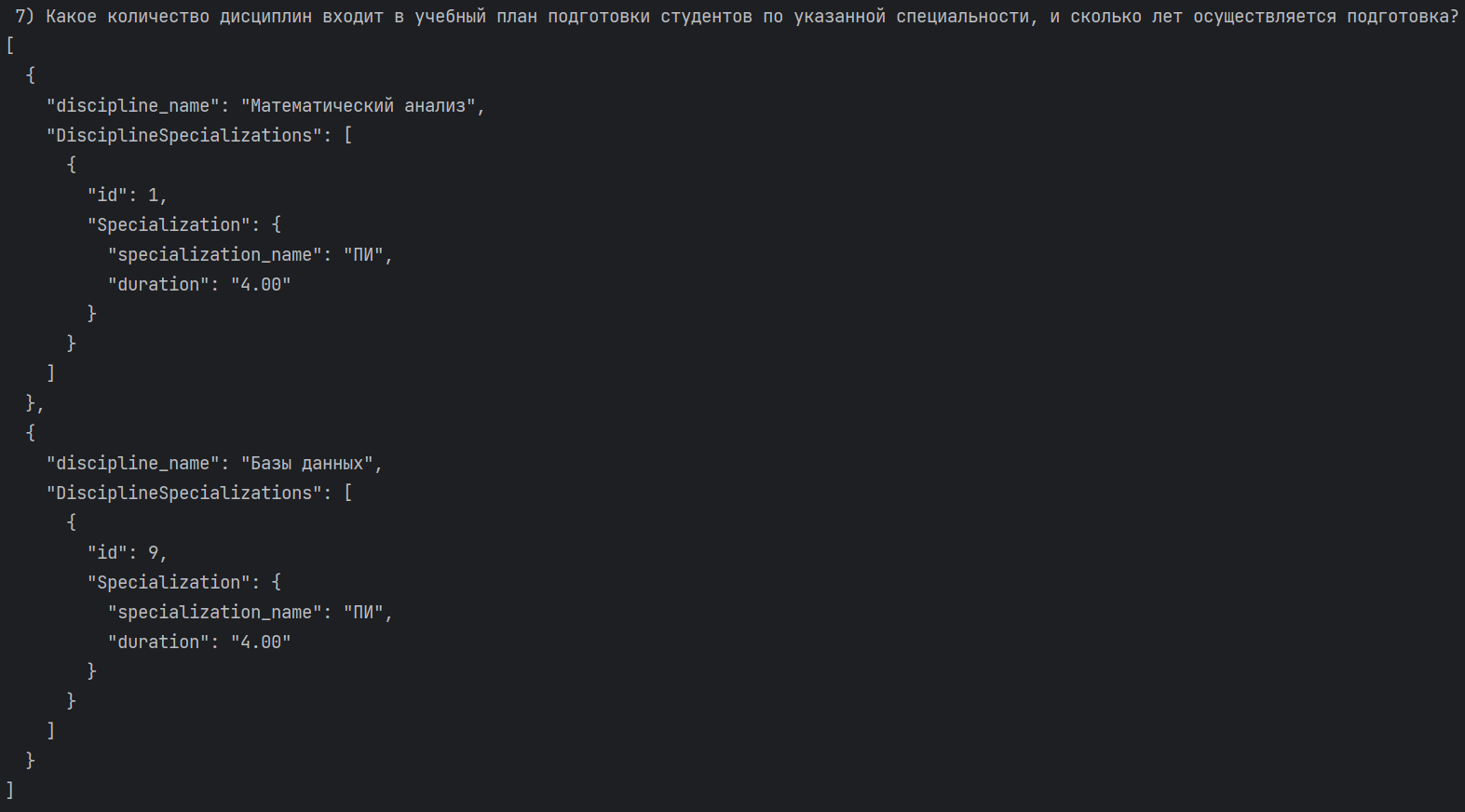


Рисунок 16 – Результат запроса 7.

**Выводы.**

В данной лабораторной работе освоена работа с ORM для Node.js -Sequelize.

**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303>/pull/42