

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных с использованием ORM**

Студент гр. 3341

Мальцев К.Л.

Преподаватель

Заславский М.М.

Санкт-Петербург

2025

### **Цель работы.**

Целью лабораторной работы является освоение ORM библиотеки Sequelize для Node.js: развертывание, установка зависимостей, заполнение БД данными и написание запросов к БД из 1-й лабораторной работы.

### **Задание.**

Вариант 9 Пусть требуется создать программную систему, предназначенную для диспетчера автобусного парка. Такая система должна обеспечивать хранение сведений о водителях, о маршрутах и характеристиках автобусов. Каждый водитель характеризуется паспортными данными, классом, стажем работы и окладом, причем оклад зависит от класса и стажа работы. Маршрут автобуса характеризуется номером маршрута, названием начального и конечного пункта движения, временем начала и конца движения, интервалом движения и протяженностью в минутах (время движения от кольца до кольца). Характеристиками автобуса являются: номер государственной регистрации автобуса, его тип и вместимость, причем вместимость автобуса зависит от его типа. Каждый водитель закреплен за отдельным автобусом и работает на определенном маршруте, но в случае поломки своего автобуса или болезни другого водителя может пересесть на другую машину. В базе должен храниться график работы водителей. Необходимо предусмотреть возможность корректировки БД в случаях поступления на работу нового водителя, списания старого автобуса, введения нового маршрута или изменения старого и т.п. Диспетчеру автопарка могут потребоваться следующие сведения:

Список водителей, работающих на определенном маршруте с указанием графика их работы?

Какие автобусы обслуживают данный маршрут?

Какие маршруты начинаются или заканчиваются в пункте с заданным названием?

Когда начинается и заканчивается движение автобусов на каждом маршруте?

Какова протяженность определенного маршрута?

Какова общая протяженность маршрутов, обслуживаемых автопарком?

Какие автобусы не вышли на линию, и по какой причине (неисправность, отсутствие водителя)?

В данной лабораторной работе рекомендуется использовать Sequelize (Node.js). В случае, если не знаете как с ним работать, то можно воспользоваться примером из репозитория.

Вы можете использовать другой ORM по вашему выбору по согласованию с преподавателем, принимающим у вас практики.

Необходимо развернуть Sequelize на своем ПК и выполнить следующие задачи:

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 человеко-читабельный вывод)

Запустить 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 отчете 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.**

В 1-й лабораторной работе на основе текста задания была построена структура БД (рис.1).

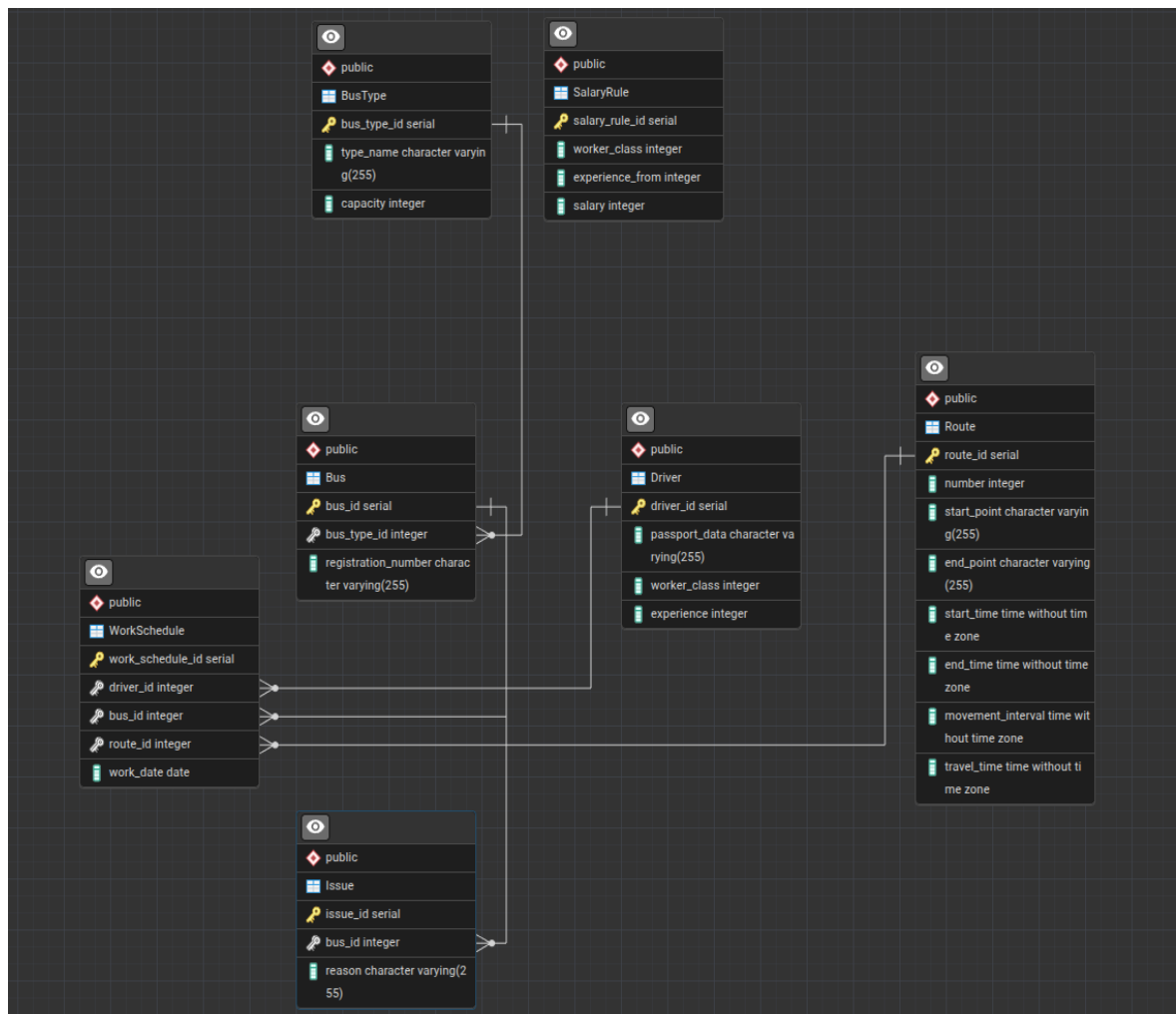


Рисунок 1 – Структура БД

Данная лабораторная работа была выполнена с использованием Sequelize.

Для удобства используется docker контейнер, где предустанавливаются нужные зависимости. Для работы можно подключиться к контейнеру по ssh.

Описание структуры:

- schema.js - описание моделей,
- seed.js - заполнение данными
- query.js - запросы

```

8
9  const BusType = sequelize.define(
10    'BusType',
11    {
12      bus_type_id: {
13        type: DataTypes.INTEGER,
14        primaryKey: true,
15        autoIncrement: true
16      },
17      type_name: {
18        type: DataTypes.STRING,
19        allowNull: false
20      },
21      capacity: {
22        type: DataTypes.INTEGER,
23        allowNull: false
24      }
25    },
26    {
27      freezeTableName: true,
28      timestamps: false
29    }
30  );

```

Рисунок 3 – Пример модели (СКРИНШОТ)

Для запуска:

```

docker compose up -d --build
ssh root@localhost -p 2222 # pass: root
node seed.js # заполнение модели
node query.js # запросы

```

Разработанный программный код см. в **приложении А**.

## **Выводы.**

В результате выполнения лабораторной работы была освоена ORM библиотека Sequelize для Node.js. Sequelize была развернута, были установлены зависимости, в виде моделей описаны таблицы из 1-й лабораторной работы, написан скрипт заполнения тестовыми данными и запросы к БД, отвечающие на вопросы из 1-й лабораторной работы с использованием ORM.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Ссылка на Pull-request с результатами выполненной лабораторной работы:

<https://github.com/moevm/sql-2025-3341/pull/28>