

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Базы данных»**  
**Тема: Проектирование ER-модели и структуры БД**

Студент гр. 3341

Мальцев К.Л.

Преподаватель

Заславский М.М.

Санкт-Петербург

2025

## 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 пересест 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)?

## Выполнение работы

### 1. ER модель

Модель отношения сущностей (Entity Relation) была реализована с помощью графического редактора draw.io. Сущности составлялись в соответствии с заданием и сведениям, которые могут потребоваться.

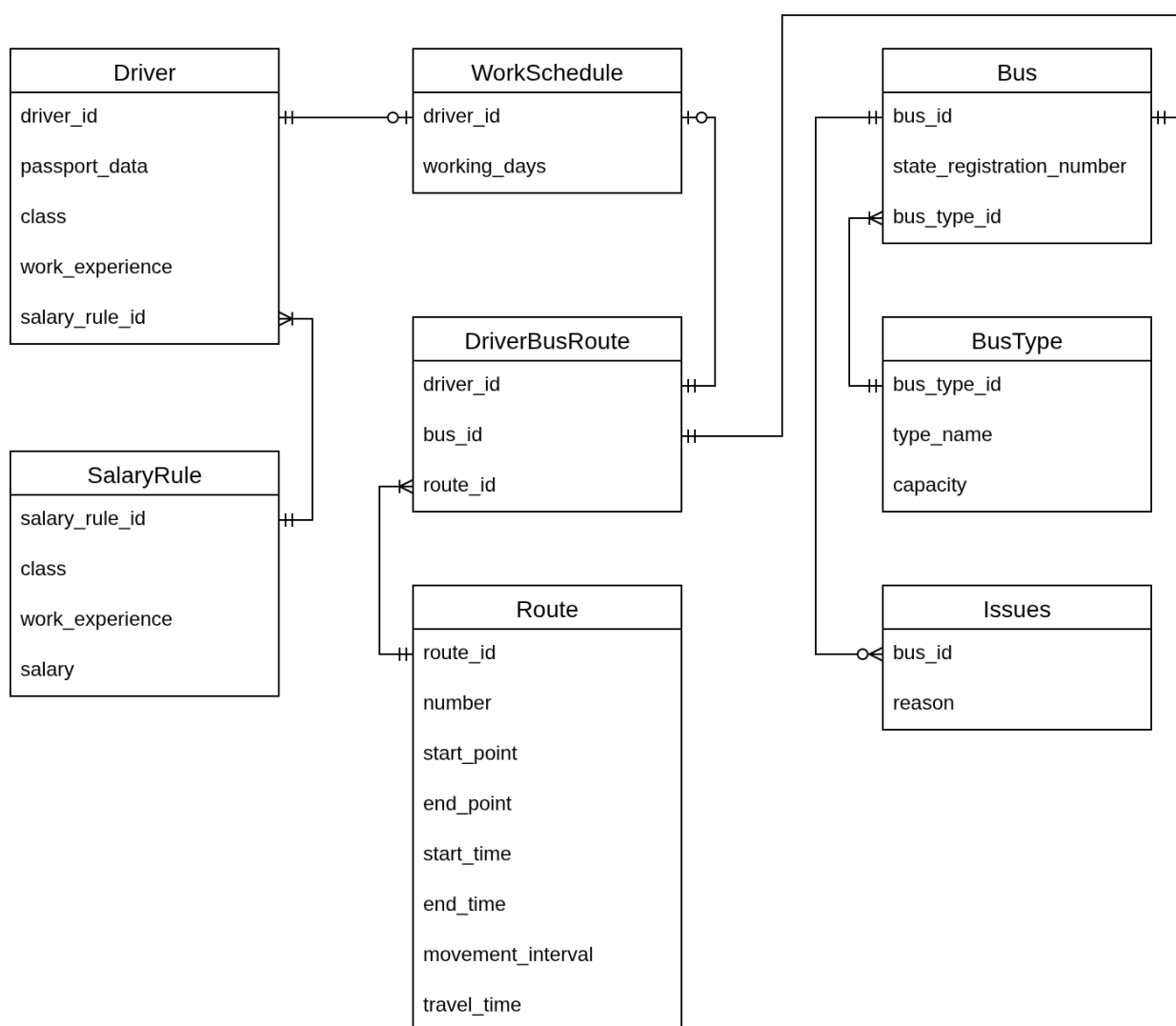


Рисунок № 1 – Entity-Relation модель

Описание сущностей:

#### 1. Driver (Водитель)

- driver\_id — идентификатор водителя
- passport\_data — паспортные данные

- class — класс водителя
  - work\_experience — стаж работы
  - salary\_rule\_id — ссылка на правило расчета зарплаты
2. SalaryRule (Правило расчета зарплаты)
- salary\_rule\_id — идентификатор правила
  - class — класс водителя
  - work\_experience — стаж работы
  - salary — размер оклада
3. WorkSchedule (График работы)
- driver\_id — идентификатор водителя
  - working\_days — рабочие дни
4. DriverBusRoute (Назначение водителя на маршрут)
- driver\_id — идентификатор водителя
  - bus\_id — идентификатор автобуса
  - route\_id — идентификатор маршрута
5. Route (Маршрут)
- route\_id — идентификатор маршрута
  - number — номер маршрута
  - start\_point — начальная остановка
  - end\_point — конечная остановка
  - start\_time — время начала работы
  - end\_time — время окончания работы
  - movement\_interval — интервал движения
  - travel\_time — время в пути
6. Bus (Автобус)
- bus\_id — идентификатор автобуса
  - state\_registration\_number — государственный номер
  - bus\_type\_id — тип автобуса
7. BusType (Тип автобуса)
- bus\_type\_id — идентификатор типа
  - type\_name — название типа

- capacity — вместимость

#### 8. Issues (Неисправности)

- bus\_id — идентификатор автобуса
- reason — причина неисправности

### Построение структуры БД

На основе реализованной ER-модели (Рисунок № 1) была спроектирована реляционная модель базы данных:

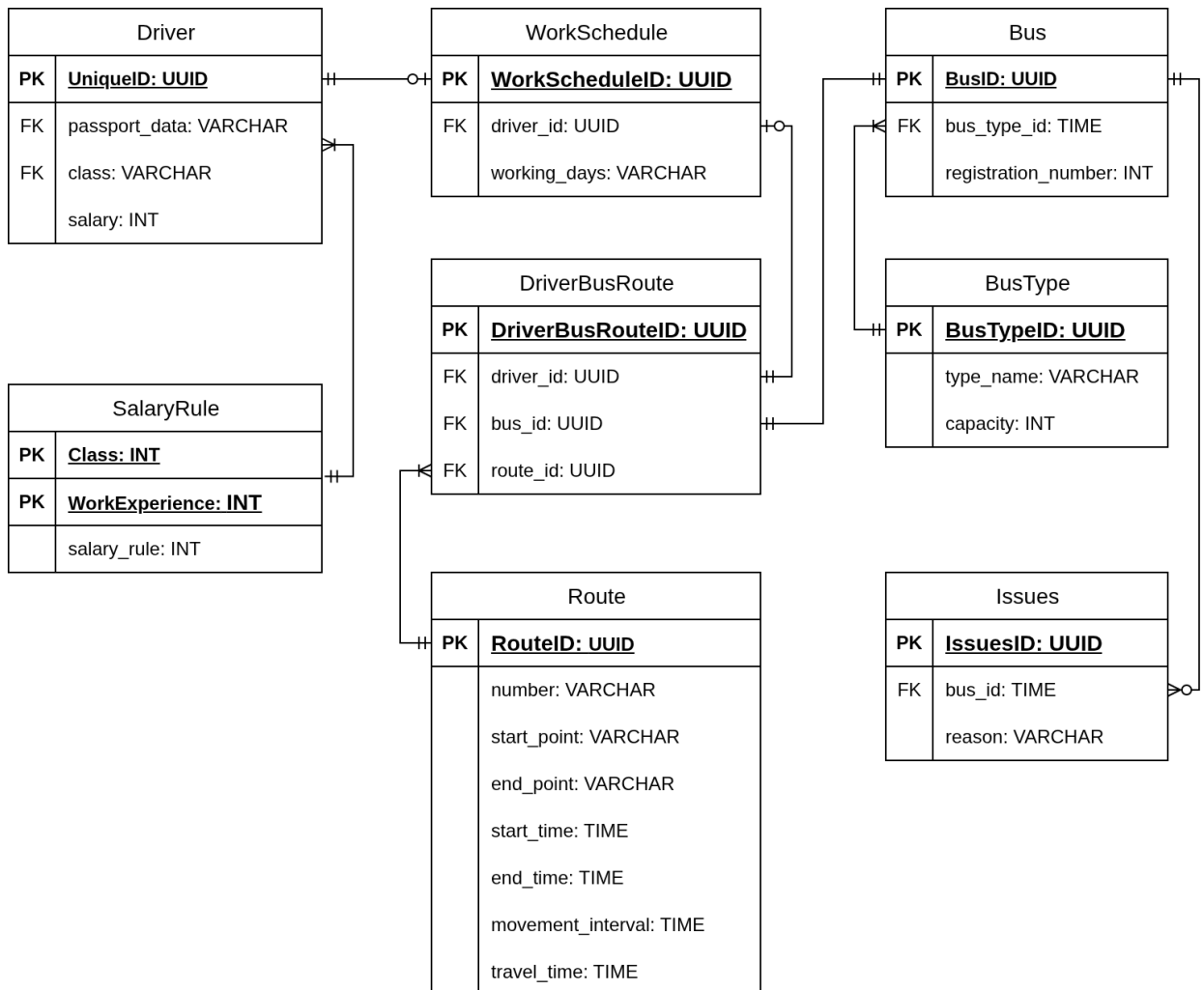


Рисунок № 2 – Структура БД

Для всех таблиц в качестве первичного ключа используется суррогатный ключ *id* типа *UUID*. Добавлены каждому полю — тип данных, показаны первичные (PK) и внешние ключи (FK) соответственно.

### Проверка соответствия НФБК:

1. Таблица: Driver Первичный ключ: UniqueID (UUID)  
Атрибуты: passport\_data, class, salary  
Функциональные зависимости: UniqueID  $\rightarrow$  passport\_data, class, salary
2. Таблица: WorkSchedule Первичный ключ: составной (driver\_id, working\_days) Атрибуты: отсутствуют дополнительные  
Функциональные зависимости: driver\_id, working\_days  $\rightarrow$  (все атрибуты являются частью ключа)
3. Таблица: DriverBusRoute Первичный ключ: составной (driver\_id, bus\_id, route\_id) Атрибуты: отсутствуют дополнительные Функциональные зависимости: driver\_id, bus\_id, route\_id  $\rightarrow$  (все атрибуты являются частью ключа)
4. Таблица: BusType Первичный ключ: type\_name Атрибуты: capacity  
Функциональные зависимости: type\_name  $\rightarrow$  capacity
5. Таблица: SalaryRule Первичный ключ: составной (Class, WorkExperience) Атрибуты: salary\_rule Функциональные зависимости: Class, WorkExperience  $\rightarrow$  salary\_rule Вывод: Соответствует НФБК.
6. Таблица: Route Первичный ключ: number Атрибуты: start\_point, end\_point, start\_time, end\_time, movement\_interval, travel\_time  
Функциональные зависимости: number  $\rightarrow$  start\_point, end\_point, start\_time, end\_time, movement\_interval, travel\_time
7. Таблица: Issues Первичный ключ: bus\_id Атрибуты: reason  
Функциональные зависимости: bus\_id  $\rightarrow$  reason

Все сущности соответствуют НФБК.

## Вывод

В ходе проектирования была разработана ER-модель и реляционная структура БД. Проведенный анализ показал, что все таблицы соответствуют Нормальной Форме Бойса-Кодда (НФБК), поскольку каждая таблица имеет один или несколько потенциальных ключей, и все выявленные функциональные зависимости исходят от этих ключей.

**ПРИЛОЖЕНИЕ А**  
**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Pull Request: <https://github.com/moevm/sql-2025-3341/pull/12>