

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 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)?

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.

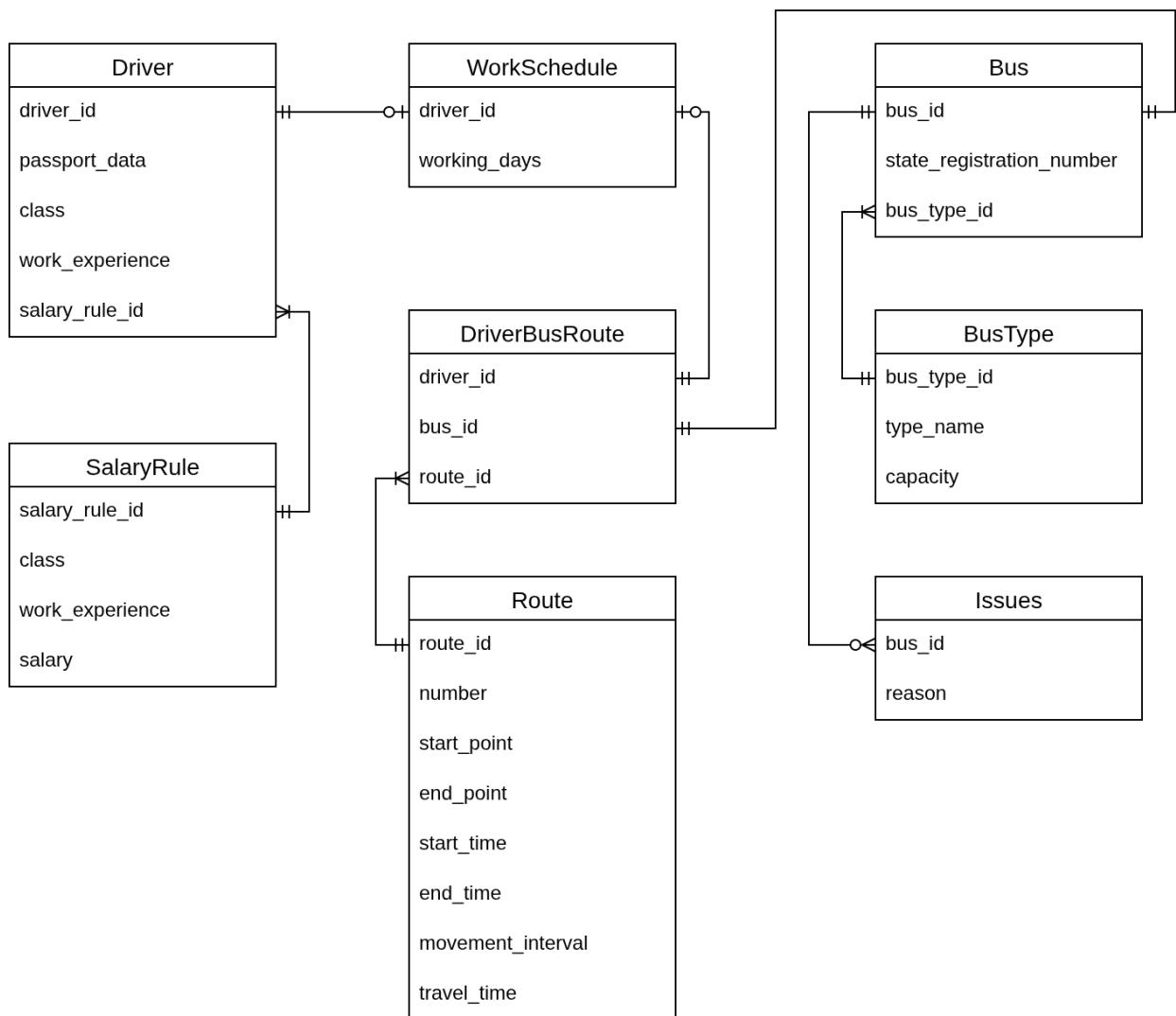


Рисунок № 1 – Entity-Relation модель

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:

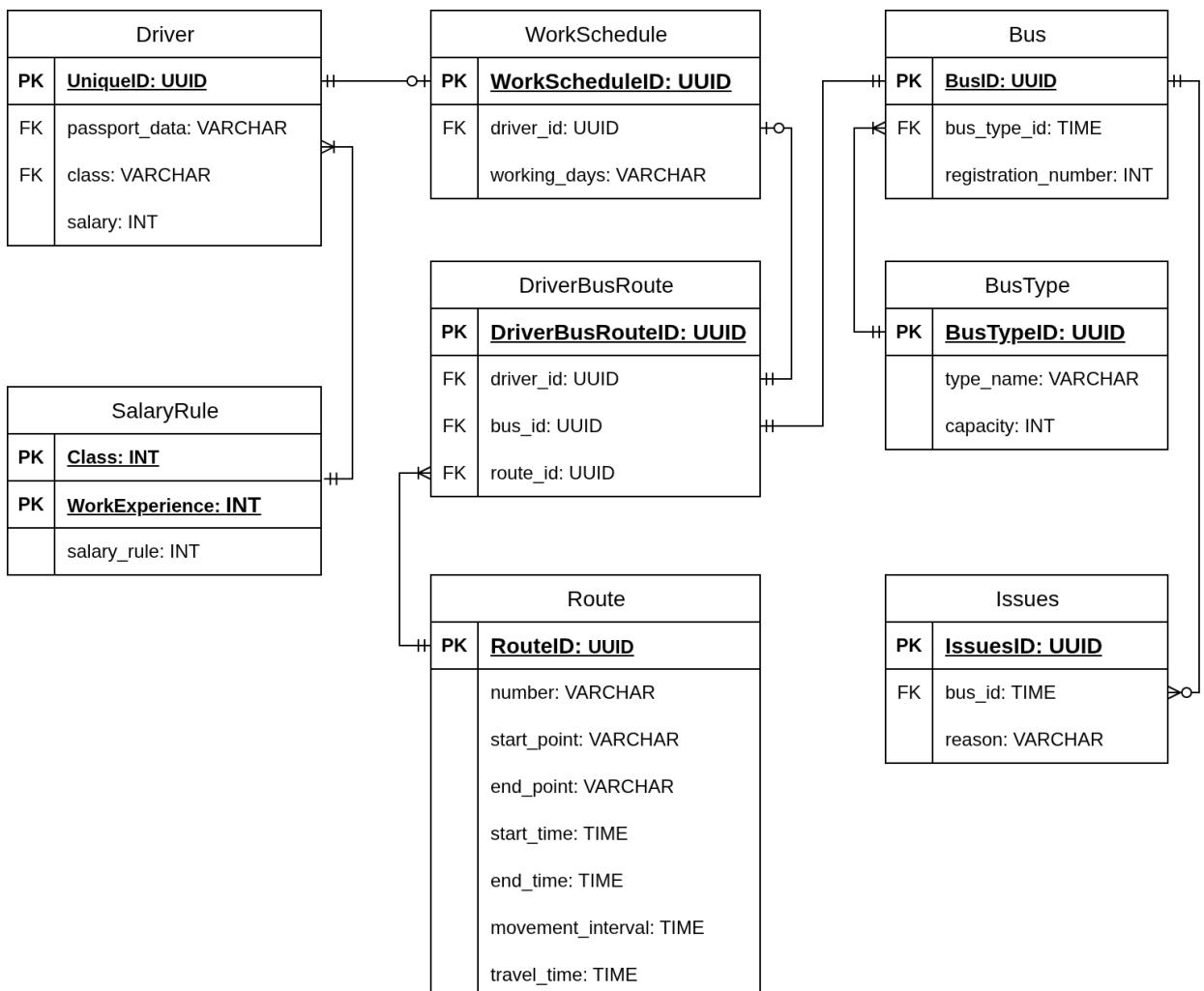


Рисунок № 2 – Структура БД

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 → passport_data, class, salary
2. Таблица: WorkSchedule Первичный ключ: составной (driver_id, working_days) Атрибуты: отсутствуют дополнительные
Функциональные зависимости: driver_id, working_days → (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 → (все атрибуты являются частью ключа)
4. Таблица: BusType Первичный ключ: type_name Атрибуты: capacity
Функциональные зависимости: type_name → capacity
5. Таблица: SalaryRule Первичный ключ: составной (Class, WorkExperience) Атрибуты: salary_rule Функциональные зависимости: Class, WorkExperience → 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 → 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 → 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>