

Проект

Состав команды:

1. Мельников Андрей Александрович (ИАД-14)
2. Родина Ольга Николаевна (ИАД-14)
3. Старостин Павел Владимирович (ИАД-17)
4. Турсунов Данил Вячеславович (ИАД-16)
5. Шиморин Матвей Алексеевич (ИАД-17)

Задача:

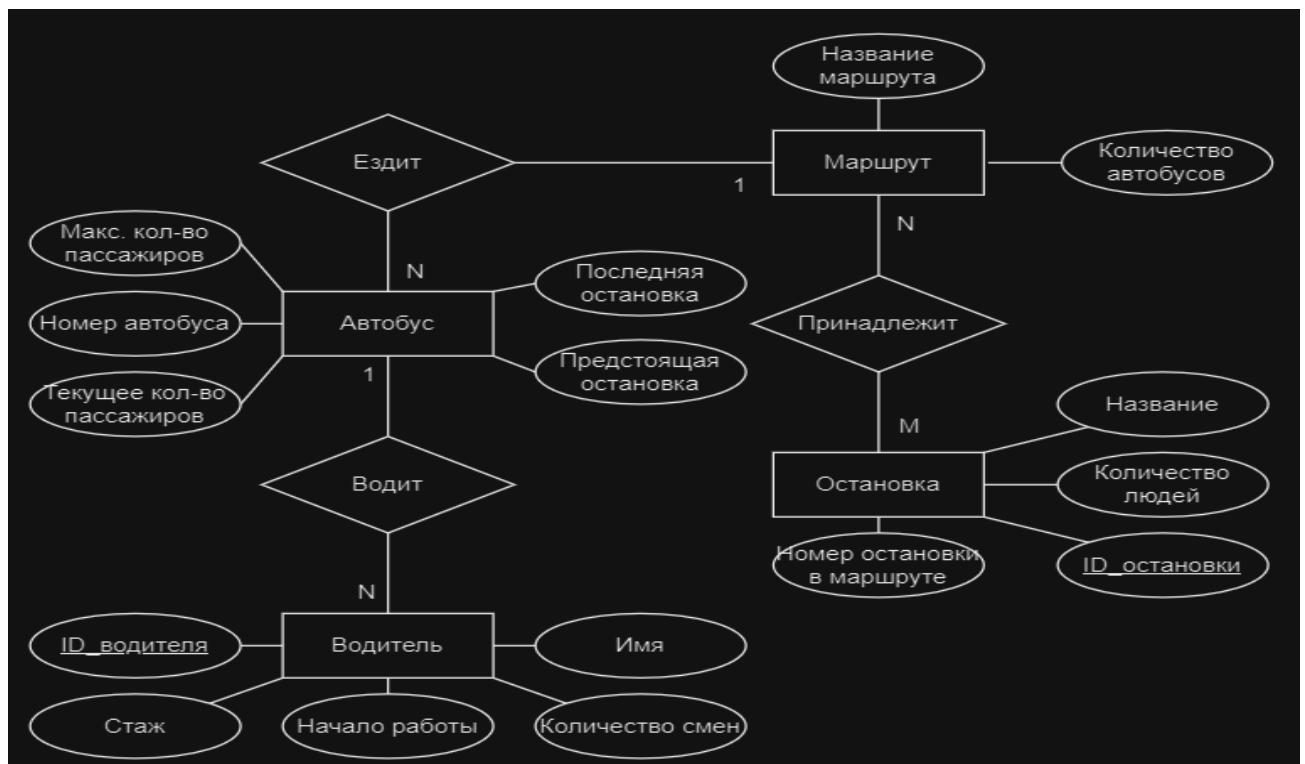
По городу (примерная численность населения 500 тысяч человек) ездят различные маршруты. Каждый из маршрутов состоит из определенного набора остановок и обслуживается несколькими автобусами. Для каждой остановки указаны номера маршрутов, которые на ней останавливаются, количество людей, ожидающих свой автобус на ней, а также ее название. За каждым автобусом закреплено несколько водителей, также у каждого автобуса есть номер, количество пассажиров в данный момент, максимальное количество пассажиров, предыдущая и следующая остановки. У водителей есть ФИО, стаж работы, дата устройства на работу, номер маршрута, по которому он работает, количество смен за месяц.

Требования и ограничения:

Количество людей на остановках и в автобусах не может быть null и отрицательным, если людей нет – ставим 0. Очередность остановок в маршруте неотрицательная и начинается с нуля. Стаж работы водителя не может быть null и отрицательным, если его нет – ставим 0. Один и тот же водитель не может ездить на разных маршрутах и автобусах. Так же, как и один и тот же автобус не может ездить по разным маршрутам.

Нефункциональные требования: выделено достаточно средств на производительную систему.

ER-модель:



Функциональные зависимости и нормализация:

ID_водителя -> (Стаж, начало работы, количество смен, имя, номер автобуса, текущее количество пассажиров, максимальное количество пассажиров, название маршрута, следующая остановка, конечная остановка, название маршрута, количество автобусов)

номер_автобуса -> (текущее количество пассажиров, максимальное количество пассажиров, название маршрута, следующая остановка, конечная остановка, название маршрута, количество автобусов)

название_маршрута -> (количество автобусов)

ID_остановки -> (количество людей на остановке, номер остановки)

Какие могут быть **проблемы** из-за **недонормализованности**:

- 1) Дублирование информации (название маршрута повторяется 3 раза в первой записи, и 2 раза во второй)
- 2) Неоднозначность (количество автобусов связано с названием маршрута, а также с номером автобусов и с ID водителя)
- 3) Отсутствие первичных ключей

Применив алгоритм синтеза к этим функциональным зависимостям получаем следующие таблицы (обозн. primary key и foreign key):

Водитель(ID водителя, имя, стаж, начало работы, количество смен, номер автобуса)

Автобус(Номер автобуса, макс кол-во пассажиров, текущее кол-во пассажиров, последняя остановка, предстоящая остановка, название маршрута)

Маршрут(Название маршрута, количество автобусов)

Принадлежит(Название маршрута, ID остановки)

Остановка (ID остановки, название остановки, количество людей, номер остановки в маршруте)

Данные таблицы находятся в **3 нормальной форме**.

Создание таблиц в SQL:

```
CREATE TABLE ROUT(  
    name varchar primary key, --Имя маршрута(например, Т83)  
    bus_count integer not null,  
    check (bus_count >= 0)  
);  
  
CREATE TABLE STOP(  
    ID integer primary key, --ID остановки  
    name varchar not null, --Имя остановки (например, Львовская)  
    cur_people_num integer default 0, --Количество людей на остановке  
    num_in_rout integer not null, --Очередность остановки в маршруте  
    check (cur_people_num >= 0 and num_in_rout >= 0)  
);  
  
CREATE TABLE BUS(  
    state_number varchar primary key, --Гос. номер  
    max_passenger_num int, --Максимальное количество пассажиров в автобусе  
    cur_passenger_num int default 0, --Текущее количество пассажиров в автобусе  
    next_station integer references STOP(ID) on delete set null on update cascade,  
    --Текущая и предыдущая остановки, при удалении остановки из таблицы stop значения автоматически становятся  
    null. При изменении значения автоматически изменяются  
    last_station integer references STOP(ID) on delete set null on update cascade,  
    rout_name varchar references ROUT(name) on delete set null on update cascade, --Маршрут, к которому  
    прикреплён автобус  
    check (max_passenger_num >= 0 AND cur_passenger_num >= 0)
```

```

);
CREATE TABLE DRIVER(
  ID integer primary key, --ID водителя
  experience integer, --Стаж
  num_of_shifts integer, --Количество смен
  name varchar not null, --Имя водителя
  hiring_date timestamp not null, --Дата принятия на работу
  bus_number varchar references BUS(state_number) on delete set null on update cascade, --К какому автобусу
  --прикреплён водитель. При удалении автобуса ставится null, при изменении - изменяется
  check (experience>=0 and num_of_shifts>=0)
);
CREATE TABLE STOP_IN_ROUT(
  stop_ID integer references STOP(ID), --ID остановки
  rout_name varchar references ROUT(name) ON DELETE CASCADE ON UPDATE CASCADE --ID маршрута к
  --которому остановка принадлежит
);

```

SQL запросы:

1) Найти имена водителей, работающих на маршрутах, на которых ездят больше 5 автобусов.

```

SELECT d.name FROM DRIVER d
JOIN BUS b ON b.state_number = d.bus_number
JOIN ROUT r ON r.name = b.rout_name
WHERE r.bus_count > 5;

```

2) Найти водителя с самым большим стажем, который проезжает остановку "Ашхабадская".

```

SELECT DRIVER.* FROM DRIVER d
JOIN BUS b ON d.bus_number = b.state_number
JOIN STOP_IN_ROUT sr ON b.rout_name = sr.rout_name
JOIN STOP s ON sr.stop_ID = s.ID
WHERE s.name = 'Ашхабадская'
ORDER BY d.experience DESC
LIMIT 1;

```

3) Для каждого автобуса найти средний стаж водителей, работающих на нем. Вывести, к какому маршруту принадлежит каждый автобус.

```

SELECT b.state_number, avg(d.experience) OVER (partition BY b.state_number), r.name FROM DRIVER d
INNER JOIN BUS b ON b.state_number = d.bus_number
INNER JOIN ROUT r ON r.name = b.rout_name
ORDER BY r.name;

```

4) Для каждого маршрута найти общее количество пассажиров.

```

SELECT BUS.rout_name, sum(BUS.cur_passenger_num) FROM BUS
GROUP BY BUS.rout_name;

```