

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Базы данных»
ТЕМА: РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ В СУБД MySQL

Студент гр. 0303

Калмак Д.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2022

Цель работы.

Необходимо развернуть MySQL локально, создать базу данных, написать запросы для создания таблиц, заполнить тестовыми данными, написать запросы к БД.

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.

- 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)?

Необходимо развернуть MySQL локально:

- Написать запросы для создания таблиц из предыдущей лабораторной работы
- Заполнить 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 предыдущей лабораторной работы
- Исходный код выложить на www.db-fiddle.com для проверки работоспособности
- Исходный код в виде .sql файла запустить в виде PR в репо

Выполнение работы.

Основой для создания базы данных является диаграмма базы данных.
(см. рис. 1)

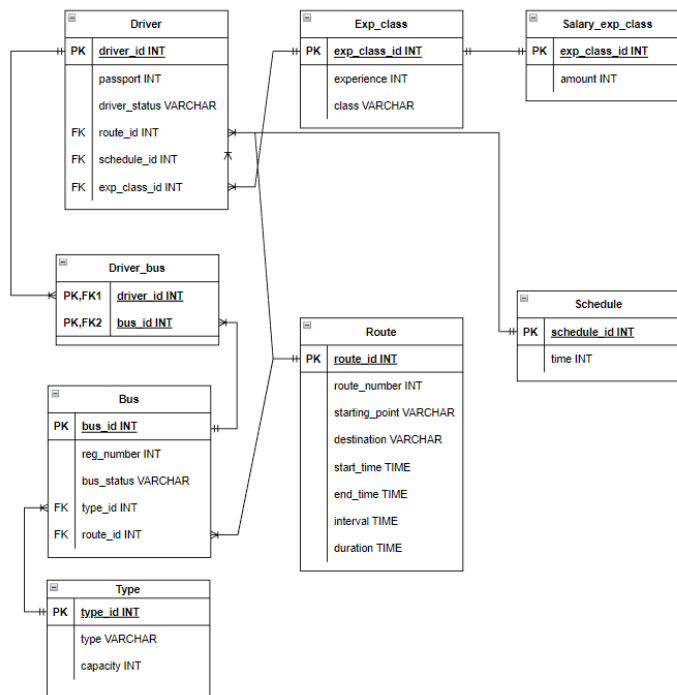


Рисунок 1 – Диаграмма базы данных

Запросы для создания таблиц и заполнения их данными представлены в файле schema.sql. Результаты выполнения запросов представлены на рис. 2-9.

type			
WHERE			
	type_id	type	capacity
1	1	Маленький	41
2	2	Средний	107
3	3	Большой	149
4	4	Особо большой	162
5	5	Крупный	169

Рисунок 2 – Таблица типов автобусов

	schedule_id	time1	time2
1	1	05:00:00	09:00:00
2	2	09:00:00	13:00:00
3	3	13:00:00	17:00:00
4	4	17:00:00	21:00:00
5	5	21:00:00	01:00:00

Рисунок 3 – Таблица графика работы

	exp_class_id	experience	class
1	1	1	Новичок
2	2	3	Опытный
3	3	5	Опытный
4	4	6	Практичный
5	5	7	Мастер своего дела

Рисунок 4 – Таблица стажа и класса

	exp_class_id	amount
1	1	35000
2	2	50000
3	3	70000
4	4	90000
5	5	100000

Рисунок 5 – Таблица окладов

	route_id	route_number	starting_point	destination	start_time	end_time	interval_time	duration
1	1	219	Udelnaya	Pionerskaya	05:31:00	00:31:00	00:15:00	00:10:00
2	2	127	Kolomyagi	Artseulovskaya_Alley	05:21:00	00:31:00	00:07:00	00:37:00
3	3	171	JSC_UEC-Klimov	Shavrova	06:06:00	00:10:00	00:12:00	00:40:00
4	4	235	Staraya Derevnaya Metro Station	Zemsky Lane	05:31:00	00:01:00	00:14:00	00:52:00
5	5	285	Friendly Avenue	Rusakovskaya Street	06:01:00	00:01:00	00:08:00	00:15:00

Рисунок 6 – Таблица маршрутов

	bus_id	reg_number	bus_status	type_id	route_id
1	1	111	Работает	1	1
2	2	222	Работает	2	2
3	3	333	Работает	2	3
4	4	444	Работает	3	4
5	5	445	Работает	1	5

Рисунок 7 – Таблица автобусов

	driver_id	passport	driver_status	route_id	schedule_id	exp_class_id
1	1	4000356789	Здоров	3	1	2
2	2	3000867321	Здоров	5	1	1
3	3	4011789009	Здоров	1	3	1
4	4	4005328742	Здоров	2	3	2
5	5	4014938049	Здоров	4	2	3

Рисунок 8 – Таблица водителей

	driver_id	bus_id
1	1	3
2	2	5
3	3	1
4	4	2
5	5	4

Рисунок 9 – Таблица водителей и автобусов

Запросы к БД, отвечающие на вопросы из задания, представлены в файле query.sql. Результаты выполнения запросов представлены на рис. 10-19.

```
# Список водителей, работающих на 219 маршруте с указанием графика их работы
SELECT driver.driver_id AS Номер_водителя, passport AS Паспорт_водителя, time1 AS Начало_смены, time2 AS Конец_смены
FROM driver
JOIN driver_bus db on driver.driver_id = db.driver_id
JOIN schedule USING(schedule_id)
JOIN route USING(route_id)
WHERE route_number = 219;
```

№	Номер_водителя	Паспорт_водителя	Начало_смены	Конец_смены
1	3	4011789009	13:00:00	17:00:00

Рисунок 10 – Список водителей, работающих на 219 маршруте с указанием графика их работы

```
# Какие автобусы обслуживают 235 маршрут
SELECT bus_id AS Номер_автобуса_в_системе, reg_number AS Номер_государственной_регистрации_автобуса
FROM bus
JOIN route USING(route_id)
WHERE route_number = 235;
```

№	Номер_автобуса_в_системе	Номер_государственной_регистрации_автобуса
1	4	444

Рисунок 11 – Какие автобусы обслуживают 235 маршрут

```
# Какие маршруты начинаются в пункте JSC_UEC-Klimov
SELECT route_number AS Номер_маршрута
FROM route
WHERE starting_point LIKE 'JSC_UEC-Klimov';
```

№	Номер_маршрута
1	171

Рисунок 12 – Какие маршруты начинаются в пункте JSC_UEC-Klimov

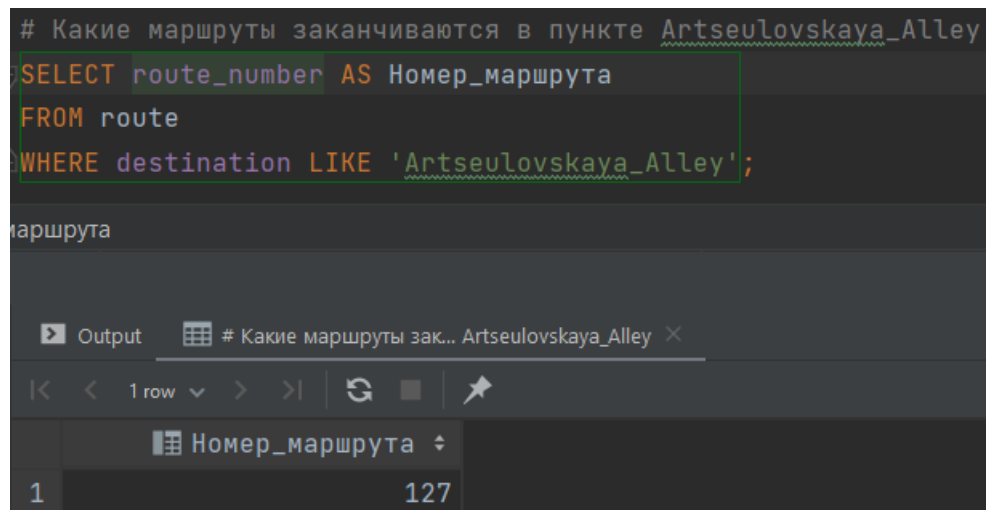


Рисунок 13 – Какие маршруты заканчиваются в пункте Artseulovskaya_Alley

The screenshot shows a SQL query in a dark-themed editor:

```
# Когда начинается и заканчивается движение автобусов на каждом маршруте
SELECT route_number AS Номер_маршрута, start_time AS Начало_движения_автобусов, end_time AS Конец_движения_автобусов
FROM route;
```

Below the query, the output is displayed in a table with the following data:

Номер_маршрута	Начало_движения_автобусов	Конец_движения_автобусов
219	05:31:00	00:31:00
127	05:21:00	00:31:00
171	06:06:00	00:10:00
235	05:31:00	00:01:00
285	06:01:00	00:01:00

Рисунок 14 – Когда начинается и заканчивается движение автобусов на каждом маршруте

The screenshot shows a SQL query in a dark-themed editor:

```
# Какова протяженность 127 маршрута
SELECT route_number AS Номер_маршрута, duration AS Протяженность_маршрута
FROM route
WHERE route_number = 127;
```

Below the query, the output is displayed in a table with the following data:

Номер_маршрута	Протяженность_маршрута
127	00:37:00

Рисунок 15 – Какова протяженность 127 маршрута

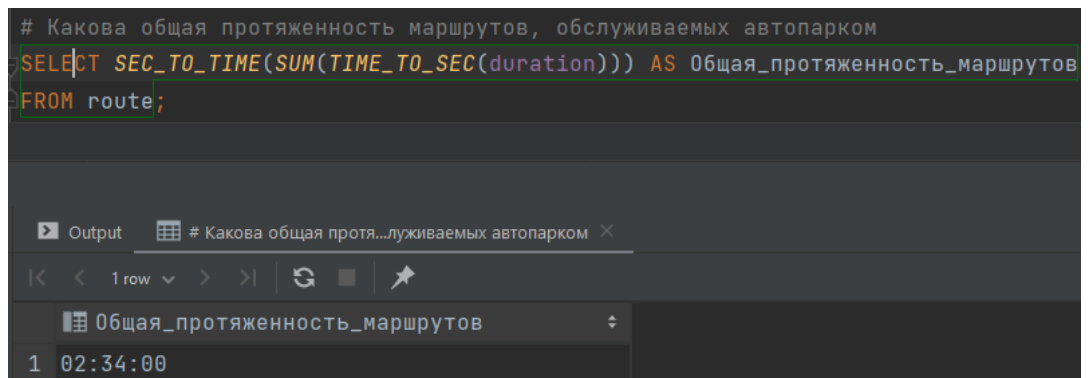


Рисунок 16 – Какова общая протяженность маршрутов, обслуживаемых автопарком

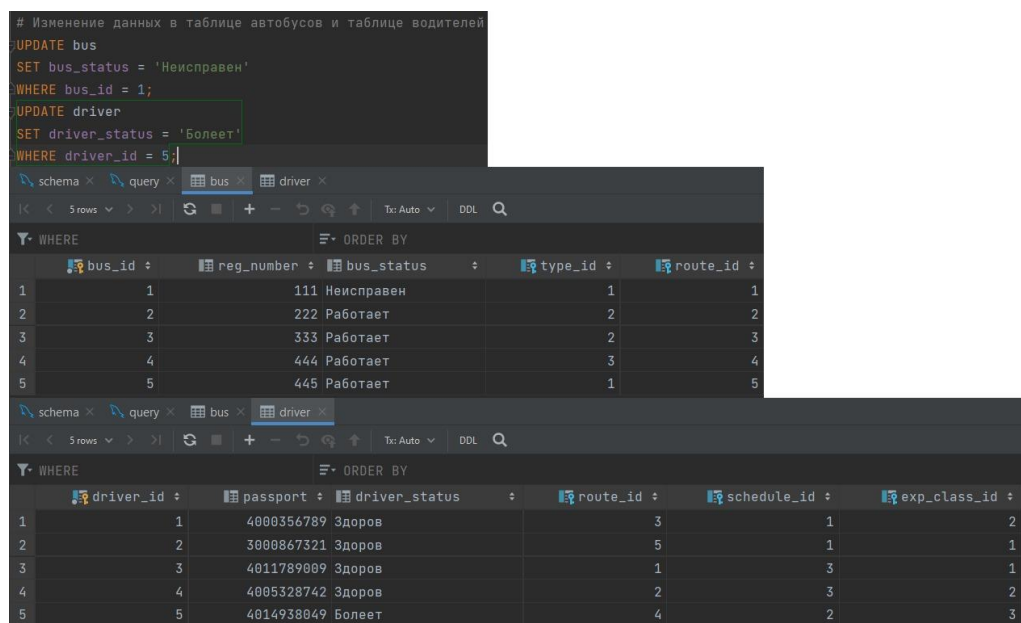


Рисунок 17 – Изменение данных в таблице автобусов и таблице водителей

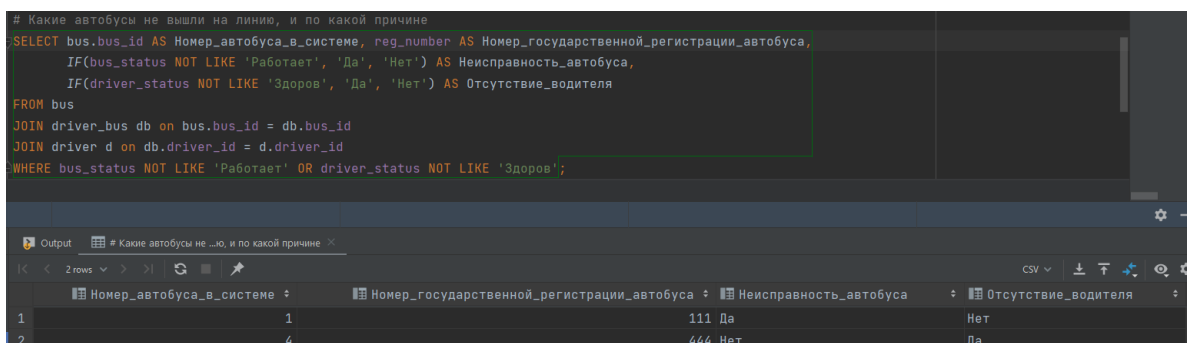


Рисунок 18 – Какие автобусы не вышли на линию, и по какой причине

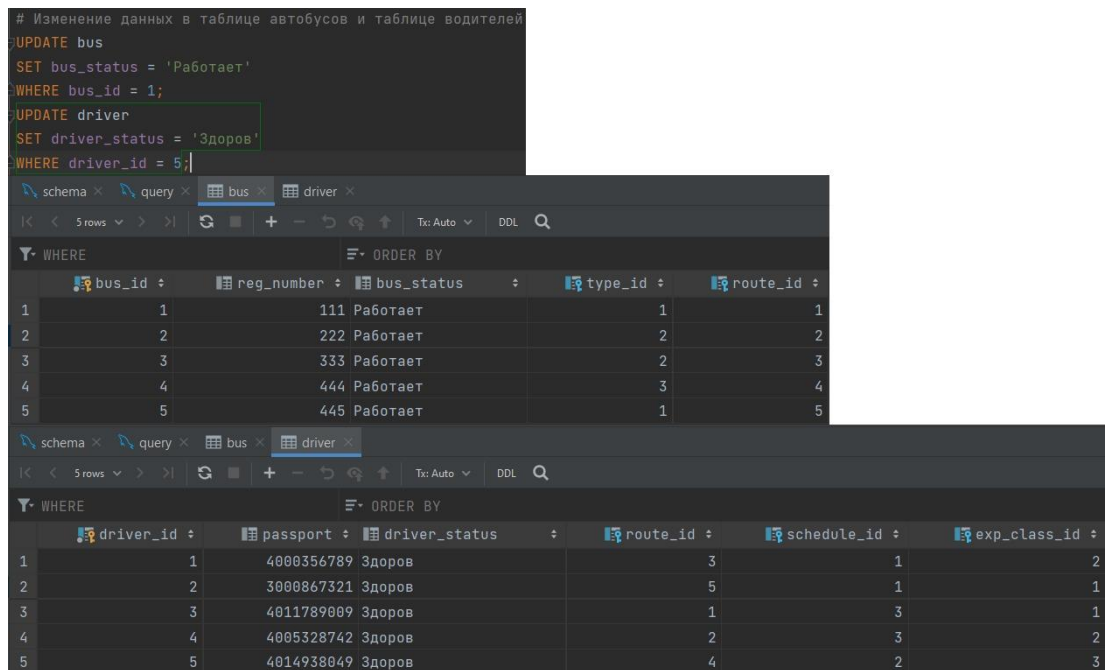


Рисунок 19 – Изменение данных в таблице автобусов и таблице водителей

Ссылка на пулл-реквест, ссылка на db-fiddle, исходный код в виде .sql файлов представлены в Приложении А.

Выводы.

Таким образом, был развернут MySQL локально, создана база данных на основе диаграммы базы данных, а именно написаны запросы для создания таблиц, таблицы заполнены тестовыми данными. Также были написаны запросы к БД, отвечающие на вопросы задания.

ПРИЛОЖЕНИЕ А

Ссылка на пулл реквест:

<https://github.com/moevm/sql-2022-0303/pull/32>

Ссылка на db-fiddle:

<https://www.db-fiddle.com/f/chilUXwBKM6u5HtqEut4lT/1#&togetherjs=cjEpbCAz80>

Исходный код

Файл schema.sql

```
# создание базы данных
CREATE DATABASE bus_park_dispatcher;
USE bus_park_dispatcher;

# тип автобуса
CREATE TABLE type (
    type_id INT PRIMARY KEY AUTO_INCREMENT,
    type VARCHAR(100),
    capacity INT
);

# заполнение таблицы типов автобуса данными
INSERT INTO type(type, capacity)
VALUES ('Маленький', 41),
       ('Средний', 107),
       ('Большой', 149),
       ('Особо большой', 162),
       ('Крупный', 169);

# график работы
CREATE TABLE schedule (
    schedule_id INT PRIMARY KEY AUTO_INCREMENT,
    time1 TIME,
    time2 TIME
);

# заполнение таблицы графика работы данными
INSERT INTO schedule(time1, time2)
VALUES ('05:00:00', '09:00:00'),
       ('09:00:00', '13:00:00');
```

```
    ('13:00:00', '17:00:00'),  
    ('17:00:00', '21:00:00'),  
    ('21:00:00', '01:00:00');
```

стаж и класс

```
CREATE TABLE exp_class (  
    exp_class_id INT PRIMARY KEY AUTO_INCREMENT,  
    experience INT,  
    class VARCHAR(100)  
);
```

заполнение таблицы стажа и класса данными

```
INSERT INTO exp_class(experience, class)  
VALUES (1, 'Новичок'),  
       (3, 'Опытный'),  
       (5, 'Опытный'),  
       (6, 'Практичный'),  
       (7, 'Мастер своего дела');
```

оклады

```
CREATE TABLE salary_exp_class (  
    exp_class_id INT PRIMARY KEY AUTO_INCREMENT,  
    amount INT,  
    FOREIGN KEY (exp_class_id) REFERENCES exp_class (exp_class_id) ON DELETE  
CASCADE  
);
```

заполнение таблицы окладов данными

```
INSERT INTO salary_exp_class(amount)  
VALUES (35000),  
       (50000),  
       (70000),  
       (90000),  
       (100000);
```

маршруты

```
CREATE TABLE route (  
    route_id INT PRIMARY KEY AUTO_INCREMENT,  
    route_number INT,  
    starting_point VARCHAR(100),  
    destination VARCHAR(100),  
    start_time TIME,  
    end_time TIME,  
    interval_time TIME,  
    duration TIME
```

```
);
# заполнение таблицы маршрутов данными
INSERT INTO route(route_number, starting_point, destination, start_time,
end_time, interval_time, duration)
VALUES (219, 'Udelnaya', 'Pionerskaya', '05:31:00', '00:31:00', '00:15:00',
'00:10:00'),
      (127, 'Kolomyagi', 'Artseulovskaya_Alley', '05:21:00', '00:31:00',
'00:07:00', '00:37:00'),
      (171, 'JSC_UEC-Klimov', 'Shavrova', '06:06:00', '00:10:00', '00:12:00',
'00:40:00'),
      (235, 'Staraya Derevnnya Metro Station', 'Zemsky Lane', '05:31:00',
'00:01:00', '00:14:00', '00:52:00'),
      (285, 'Friendly Avenue', 'Rusakovskaya Street', '06:01:00', '00:01:00',
'00:08:00', '00:15:00');
```

автобусы

```
CREATE TABLE bus (
    bus_id INT PRIMARY KEY AUTO_INCREMENT,
    reg_number INT,
    bus_status VARCHAR(100),
    type_id INT,
    route_id INT,
    FOREIGN KEY (type_id) REFERENCES type (type_id) ON DELETE SET NULL,
    FOREIGN KEY (route_id) REFERENCES route (route_id) ON DELETE SET NULL
);
```

заполнение таблицы автобусов данными

```
INSERT INTO bus(reg_number, bus_status, type_id, route_id)
VALUES (111, 'Работает', 1, 1),
      (222, 'Работает', 2, 2),
      (333, 'Работает', 2, 3),
      (444, 'Работает', 3, 4),
      (445, 'Работает', 1, 5);
```

водители

```
CREATE TABLE driver (
    driver_id INT PRIMARY KEY AUTO_INCREMENT,
    passport INT UNSIGNED,
    driver_status VARCHAR(100),
    route_id INT,
    schedule_id INT,
    exp_class_id INT,
    FOREIGN KEY (route_id) REFERENCES route (route_id) ON DELETE SET NULL,
```

```

        FOREIGN KEY (schedule_id) REFERENCES schedule (schedule_id) ON DELETE SET
NULL,
        FOREIGN KEY (exp_class_id) REFERENCES exp_class (exp_class_id) ON DELETE
SET NULL
    );
# заполнение таблицы водителей данными
INSERT INTO driver(passport, driver_status, route_id, schedule_id, exp_class_id)
VALUES (4000356789, 'Здоров', 3, 1, 2),
       (3000867321, 'Здоров', 5, 1, 1),
       (4011789009, 'Здоров', 1, 3, 1),
       (4005328742, 'Здоров', 2, 3, 2),
       (4014938049, 'Здоров', 4, 2, 3);

# водители и автобусы
CREATE TABLE driver_bus (
    driver_id INT,
    bus_id INT,
    FOREIGN KEY (driver_id) REFERENCES driver (driver_id) ON DELETE SET NULL,
    FOREIGN KEY (bus_id) REFERENCES bus (bus_id) ON DELETE SET NULL
);
# заполнение таблицы водителей и автобусов данными
INSERT INTO driver_bus(driver_id, bus_id)
VALUES (1, 3),
       (2, 5),
       (3, 1),
       (4, 2),
       (5, 4);

```

Файл query.sql

```

# Список водителей, работающих на 219 маршруте с указанием графика их работы
SELECT driver.driver_id AS Номер_водителя, passport AS Паспорт_водителя, time1
AS Начало_смены, time2 AS Конец_смены
FROM driver
JOIN driver_bus db on driver.driver_id = db.driver_id
JOIN schedule USING(schedule_id)
JOIN route USING(route_id)
WHERE route_number = 219;

# Какие автобусы обслуживают 235 маршрут
SELECT      bus_id      AS      Номер_автобуса_в_системе,      reg_number      AS
Номер_государственной_регистрации_автобуса
FROM bus

```

```
JOIN route USING(route_id)
```

```
WHERE route_number = 235;
```

```
# Какие маршруты начинаются в пункте JSC_UEC-Klimov
```

```
SELECT route_number AS Номер_маршрута
```

```
FROM route
```

```
WHERE starting_point LIKE 'JSC_UEC-Klimov';
```

```
# Какие маршруты заканчиваются в пункте Artseulovskaya_Alley
```

```
SELECT route_number AS Номер_маршрута
```

```
FROM route
```

```
WHERE destination LIKE 'Artseulovskaya_Alley';
```

```
# Когда начинается и заканчивается движение автобусов на каждом маршруте
```

```
SELECT route_number AS Номер_маршрута, start_time AS Начало_движения_автобусов,
```

```
end_time AS Конец_движения_автобусов
```

```
FROM route;
```

```
# Какова протяженность 127 маршрута
```

```
SELECT route_number AS Номер_маршрута, duration AS Протяженность_маршрута
```

```
FROM route
```

```
WHERE route_number = 127;
```

```
# Какова общая протяженность маршрутов, обслуживаемых автопарком
```

```
SELECT SEC_TO_TIME(SUM(TIME_TO_SEC(duration)))
```

```
AS
```

```
Общая_протяженность_маршрутов
```

```
FROM route;
```

```
# Изменение данных в таблице автобусов и таблице водителей
```

```
UPDATE bus
```

```
SET bus_status = 'Неисправен'
```

```
WHERE bus_id = 1;
```

```
UPDATE driver
```

```
SET driver_status = 'Болеет'
```

```
WHERE driver_id = 5;
```

```
# Какие автобусы не вышли на линию, и по какой причине
```

```
SELECT bus.bus_id AS Номер_автобуса_в_системе, reg_number AS
```

```
Номер_государственной_регистрации_автобуса,
```

```
IF(bus_status NOT LIKE 'Работает', 'Да', 'Нет') AS
```

```
Неисправность_автобуса,
```

```
IF(driver_status NOT LIKE 'Здоров', 'Да', 'Нет') AS Отсутствие_водителя
```

```
FROM bus
```

```
JOIN driver_bus db on bus.bus_id = db.bus_id
```

```
JOIN driver d on db.driver_id = d.driver_id  
WHERE bus_status NOT LIKE 'Работает' OR driver_status NOT LIKE 'Здоров';
```

```
# Изменение данных в таблице автобусов и таблице водителей
```

```
UPDATE bus
```

```
SET bus_status = 'Работает'
```

```
WHERE bus_id = 1;
```

```
UPDATE driver
```

```
SET driver_status = 'Здоров'
```

```
WHERE driver_id = 5;
```