

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО»**

**Факультет инфокоммуникационных технологий**

**Дисциплина:**

**«Проектирование и реализация баз данных»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2  
«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,  
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL»**

**Выполнил:**

студент группы К32392

Байков Иван

---

(подпись)

**Проверил(а):**

Говорова Марина Михайловна

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург  
2023 г.

**Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**Практическое задание:**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и посмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

**Индивидуальное практическое задание:**

База данных “Служба заказа такси”

Запросы на выборку:

1. Вывести данные о водителе, который чаще всех доставляет пассажиров на заданную улицу.
2. Вывести данные об автомобилях, которые имеют пробег более 250 тысяч километров и которые не проходили ТО в текущем году.
3. Сколько раз каждый пассажир воспользовался услугами таксопарка?
4. Вывести данные пассажира, который воспользовался услугами таксопарка максимальное число раз.
5. Вывести данные о водителе, который ездит на самом дорогом автомобиле.
6. Вывести данные пассажира, который всегда ездит с одним и тем же водителем.
7. Какие автомобили имеют пробег больше среднего пробега для своей марки.

Представления:

1. одержащее сведения о незанятых на данный момент водителях;
2. зарплата всех водителей за вчерашний день.

**Схема базы данных:**

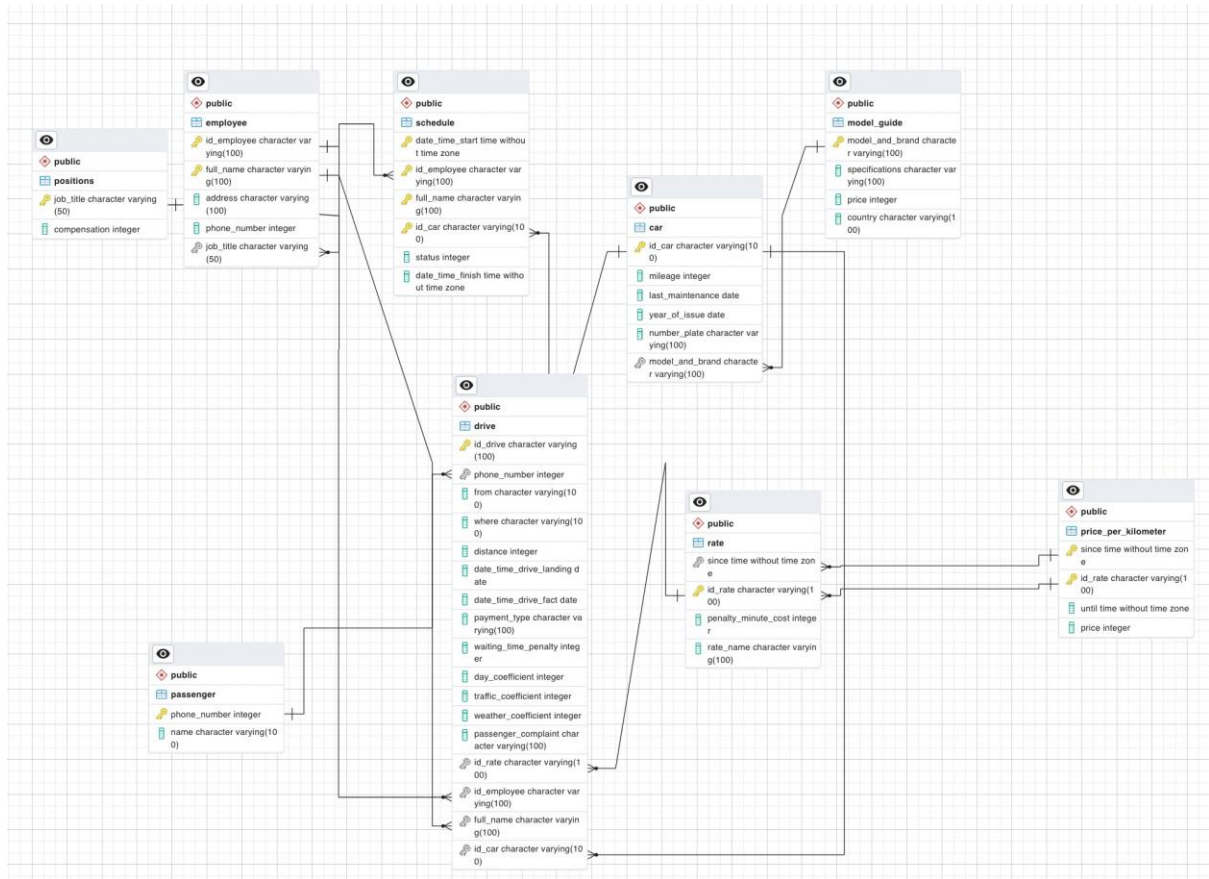


Рис. 1 - Схема базы данных

## Выполнение

### Запросы на выборку:

1. Вывести данные о водителе, который чаще всех доставляет пассажиров на заданную улицу.

```

SELECT  e.id_employee,  e.full_name,e.address,  e.phone_number,  COUNT(*) as
delivery_count
FROM employee e
JOIN drive d ON e.id_employee = d.id_employee
WHERE e.job_title = 'Водитель' AND d.where LIKE '%Рубинштейна%'
GROUP BY e.id_employee, e.full_name
ORDER BY delivery_count DESC
LIMIT 1;
  
```

Рис. 2 - SELECT №1

	id_employee [PK] character varying (100)	full_name [PK] character varying (100)	address character varying (100)	phone_number integer	delivery_count bigint
1	1	Романов Алексей Игоревич	Новочеркасская 11	89964541	1

Рис. 3 - Результат SELECT №1

2. Вывести данные об автомобилях, которые имеют пробег более 250 тысяч. километров и которые не проходили ТО в текущем году.

```
SELECT *
FROM car
WHERE mileage > 250000
      AND (EXTRACT(YEAR FROM last_maintenance) < EXTRACT(YEAR FROM CURRENT_DATE)
      OR last_maintenance IS NULL);
```

Рис. 4 - SELECT №2

	id_car [PK] character varying (100)	mileage integer	last_maintenance date	year_of_issue date	number_plate character varying (100)	model_and_brand character varying (100)
1	102	400000	2021-02-02	2020-03-14	at888x	Bmw

Рис. 5 - Результат SELECT №2

3. Сколько раз каждый пассажир воспользовался услугами таксопарка?

```
SELECT p.phone_number, p.name, COUNT(*) AS trip_count
FROM passenger p
JOIN drive d ON p.phone_number = d.phone_number
GROUP BY p.phone_number, p.name;
```

Рис. 6 - SELECT №3

	phone_number [PK] integer	name character varying (100)	trip_count bigint
1	1234567	Иван Иванов	4
2	89497854	Иван	1

Рис. 7 - Результат SELECT №3

4. Вывести данные пассажира, который воспользовался услугами таксопарка максимальное число раз.

```
SELECT p.phone_number, p.name, COUNT(*) AS trip_count
FROM passenger p
```

```
JOIN drive d ON p.phone_number = d.phone_number
GROUP BY p.phone_number, p.name
ORDER BY trip_count DESC
LIMIT 1;
```

Рис. 8 - SELECT №4

	phone_number [PK] integer	name character varying (100)	trip_count bigint
1	1234567	Иван Иванов	4

Рис. 9 - Результат SELECT №4

5. Вывести данные о водителе, который ездит на самом дорогом автомобиле.

```
SELECT e.id_employee, e.full_name
FROM employee e
JOIN schedule s ON e.id_employee = s.id_employee
JOIN car c ON s.id_car = c.id_car
JOIN model_guide m ON c.model_and_brand = m.model_and_brand
WHERE e.job_title = 'Водитель'
AND m.price = (SELECT MAX(price) FROM model_guide);
```

Рис. 10 - SELECT №5

	id_employee [PK] character varying (100)	full_name [PK] character varying (100)
1	2	Василенко Антон Игоревич

Рис. 11 - Результат SELECT №5

6. Вывести данные пассажира, который всегда ездит с одним и тем же водителем.

```
SELECT p.phone_number, p.name, e.full_name AS driver_name
FROM passenger p
JOIN drive d ON p.phone_number = d.phone_number
JOIN employee e ON d.id_employee = e.id_employee
GROUP BY p.phone_number, p.name, e.full_name
HAVING COUNT(DISTINCT d.id_employee) = 1;
```

Рис. 12 - SELECT №6

	phone_number integer	name character varying (100)	driver_name character varying (100)
1	1234567	Иван Иванов	Романов Алексей Игоревич
2	89497854	Иван	Романов Алексей Игоревич

Рис. 13 - Результат SELECT №6

7. Какие автомобили имеют пробег больше среднего пробега для своей марки.

```
SELECT c.id_car, c.mileage, c.last_maintenance, c.year_of_issue, c.number_plate,
c.model_and_brand
FROM car c
JOIN (
    SELECT model_and_brand, AVG(mileage) AS avg_mileage
    FROM car
    GROUP BY model_and_brand
) AS avg ON c.model_and_brand = avg.model_and_brand
WHERE c.mileage > avg.avg_mileage;
```

Рис. 14 - SELECT №7

	id_car [PK] character varying (100)	mileage integer	last_maintenance date	year_of_issue date	number_plate character varying (100)	model_and_brand character varying (100)
1	102	400000	2021-02-02	2020-03-14	at888x	Bmw

Рис. 15 - Результат SELECT №7

### Представления:

1.Содержащее сведения о незанятых на данный момент водителях;

```
CREATE VIEW unoccupied_drivers AS
SELECT e.id_employee, e.full_name
FROM employee e
WHERE e.job_title = 'Водитель' AND e.id_employee NOT IN (
    SELECT d.id_employee
    FROM drive d
    WHERE current_timestamp BETWEEN d.date_time_drive_landing AND
d.date_time_drive_fact
);
```

Рис. 16 - VIEW №1

2.Зарплата всех водителей за вчерашний день.

```
CREATE VIEW driver_earnings AS
```

```

SELECT d.id_drive, d.id_employee, e.full_name, d.distance * p.price AS earnings
FROM drive d
JOIN rate r ON d.id_rate = r.id_rate
JOIN price_per_kilometer p ON r.id_rate = p.id_rate
JOIN employee e ON d.id_employee = e.id_employee
WHERE d.date_time_drive_landing = (CURRENT_DATE - INTERVAL '1 day');

```

Рис. 17 - VIEW №1

### INSERT:

Добавить в расписание незанятого сотрудника с незанятой машиной, у которой наивысший пробег

```

INSERT INTO schedule (date_time_start, id_employee, full_name, id_car, status,
date_time_finish)
SELECT '09:00:00', e.id_employee, e.full_name, c.id_car, 1, '21:00:00'
FROM employee e
JOIN car c ON e.job_title = 'Водитель' AND c.id_car NOT IN (SELECT id_car FROM
schedule)
WHERE e.id_employee NOT IN (SELECT id_employee FROM schedule)
ORDER BY e.id_employee DESC, c.mileage DESC
LIMIT 1;

```

Рис. 18 - INSERT

	date_time_start [PK] time without time zone	id_employee [PK] character varying (100)	full_name [PK] character varying (100)	id_car [PK] character varying (100)	status integer	date_time_finish time without time zone
1	09:00:00	1	Романов Алексей Игоревич	101	[null]	21:00:00
2	09:00:00	2	Василенко Антон Игоревич	102	1	21:00:00
3	09:00:00	9	Миронова Екатерина	106	1	21:00:00
4	09:00:00	8	Соколов Иван	103	1	21:00:00
5	09:00:00	7	Иванов Иван	107	1	21:00:00
6	09:00:00	4	Смирнов Сергей	105	1	21:00:00

Рис. 19 - Результат INSERT

### UPDATE:

Уменьшить сегодняшнюю смену водителю, у которого вчера было минимум 2 поездки

```

UPDATE schedule
SET date_time_finish = date_time_finish - INTERVAL '1 hour'
WHERE id_employee IN (
    SELECT id_employee

```

```

FROM drive
WHERE date(date_time_drive_fact) = current_date - interval '1 day'
GROUP BY id_employee
HAVING COUNT(*) >= 2
);

```

Рис. 20 - UPDATE

	date_time_start [PK] time without time zone	id_employee [PK] character varying (100)	full_name [PK] character varying (100)	id_car [PK] character varying (100)	status integer	date_time_finish time without time zone
1	09:00:00	2	Василенко Антон Игоревич	102	1	21:00:00
2	09:00:00	9	Миронова Екатерина	106	1	21:00:00
3	09:00:00	8	Соколов Иван	103	1	21:00:00
4	09:00:00	7	Иванов Иван	107	1	21:00:00
5	09:00:00	4	Смирнов Сергей	105	1	21:00:00
6	09:00:00	1	Романов Алексей Игоревич	101	[null]	20:00:00

Рис. 21 - Результат UPDATE

### DELETE:

Удаляет всех модели из справочника, которых нет в наличии сервиса.

```

DELETE FROM model_guide
WHERE model_and_brand NOT IN (
    SELECT DISTINCT model_and_brand
    FROM car
);

```

Рис. 22 - DELETE

### Простой индекс:

```

SELECT COUNT(*)
FROM drive
WHERE DATE(date_time_drive_fact) <= '2023-12-31';

CREATE INDEX drive_date_index ON drive (date_time_drive_fact);

```

Рис. 23 - простой INDEX

Проведя серию замеров из 20 измерений, получилось сократить среднее время выполнения с 101мс до 60мс.

### Составной индекс:

```

SELECT d.id_drive, d.id_employee, e.full_name, d.distance * p.price AS earnings
FROM drive d
JOIN rate r ON d.id_rate = r.id_rate
JOIN price_per_kilometer p ON r.id_rate = p.id_rate
JOIN employee e ON d.id_employee = e.id_employee

```



```
WHERE d.date_time_drive_landing = (CURRENT_DATE - INTERVAL '1 day');  
  
CREATE INDEX drive_earnings_index ON drive (date_time_drive_landing, id_employee);
```

Рис. 24 - составной INDEX

Проведя серию замеров из 20 измерений, получилось сократить среднее время выполнения с 63мс до 55мс.

**Выводы:**

В процессе выполнения лабораторной работы получилось ознакомиться с составлением INSERT, UPDATE, DELETE запросов. Также удалось ознакомиться с графическим представлением запросов. Были составлены простые и составные индексы, что позволило наглядно увидеть уменьшение кол-ва этапов при выполнении запроса, а также уменьшение времени выполнения.