

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

ОТЧЕТ

по лабораторной работе № 4

Проектирование и реализация баз данных

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ.
ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ

Студент:

Группа № 436209

А.А. Цырульников

Преподаватель:

Преподаватель практики

М.М. Говорова

Санкт-Петербург 2025

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ

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 лабораторной работы №2, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть 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.

Вариант 14. Индивидуальное задание, часть 2:

Создать запросы

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.

Создать представление

- Содержащее сведения о незанятых на данный момент водителях;
- Зарплата всех водителей за вчерашний день.

Создать хранимые процедуры

- Для вывода данных о пассажирах, которые заказывали такси в заданном, как параметр, временном интервале.
- Вывести сведения о том, куда был доставлен пассажир по заданному номеру телефона пассажира.
- Для вычисления суммарного дохода таксопарка за истекший месяц.

Создать необходимые триггеры

1 Ход работы:

Ниже на рисунке ?? представлена схема базы данных, которая была создана в ходе выполнения лабораторной работы №3.

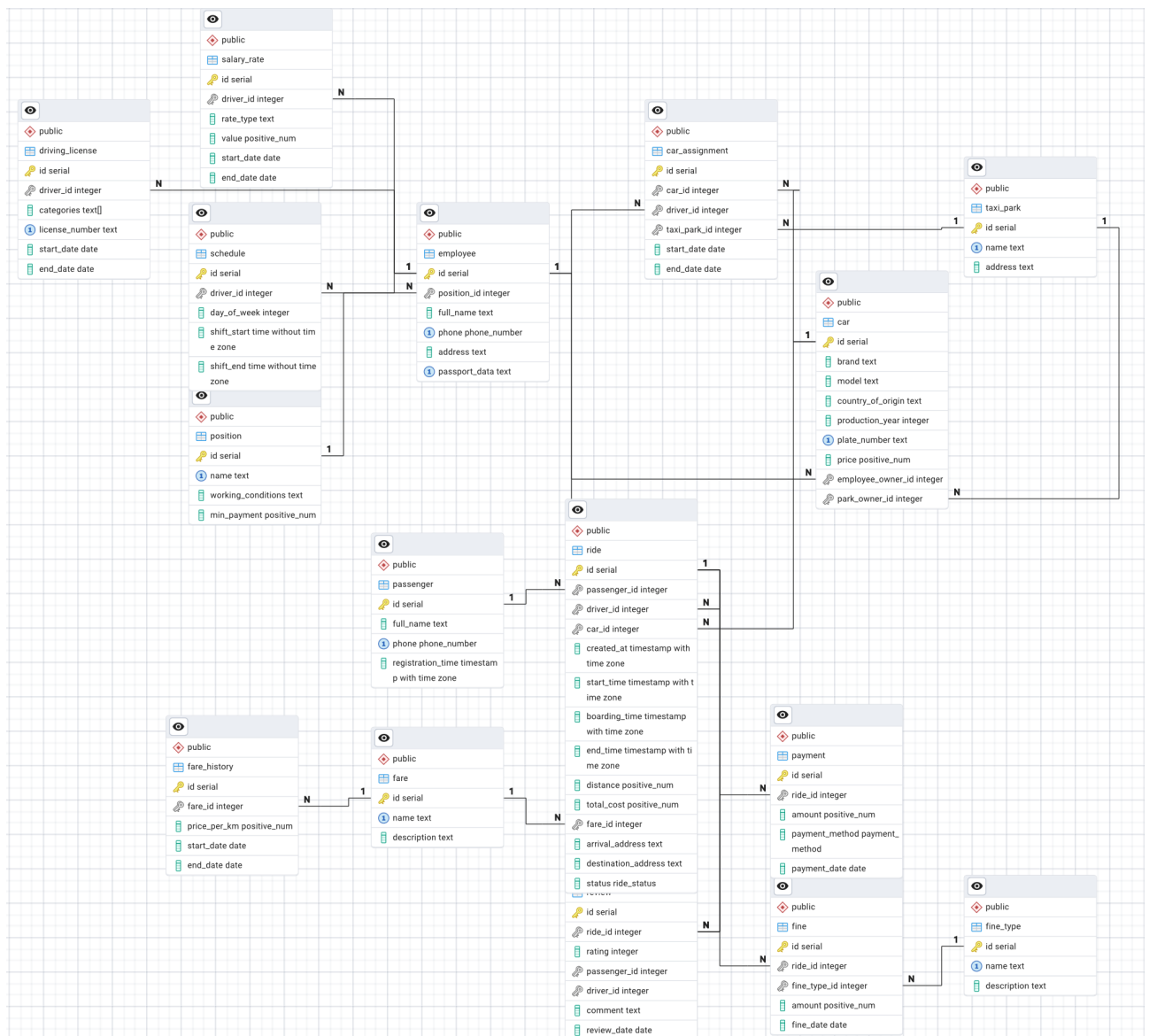


Рисунок 1.1 — Схема базы данных

1.1 Запросы на выборку данных

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

```

1  SELECT
2  e.id,
3  e.full_name,
4  e.phone,
5  COUNT(*) AS deliveries_count
6  FROM taxi_service.ride AS r
7  JOIN taxi_service.employee AS e
8  ON e.id = r.driver_id
9  WHERE r.destination_address ILIKE '%' || 'street_name
10         ' || '%',
11  GROUP BY e.id, e.full_name, e.phone
12  ORDER BY deliveries_count DESC
13  LIMIT 1;

```

	id [PK] integer	full_name text	phone text	deliveries_count bigint
1	6	Исаева Елена Владимировна	+79008889900	1

Рисунок 1.2 — Вывод данных о водителе, который чаще всех доставляет пассажиров на заданную улицу.

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

```

1  SELECT
2  c.id,
3  c.brand,
4  c.model,
5  c.plate_number,
6  cm.mileage
7  FROM taxi_service.car c

```

```

8      JOIN taxi_service.car_mileage cm ON c.id = cm.
        car_id
9      LEFT JOIN (
10     SELECT car_id
11     FROM taxi_service.car_service
12     WHERE EXTRACT(YEAR FROM service_date) = EXTRACT(
        YEAR FROM CURRENT_DATE)
13     ) cs ON c.id = cs.car_id
14     WHERE cm.mileage > 250000 AND cs.car_id IS NULL;

```

	id integer	brand text	model text	plate_number text	mileage integer
1	3	Kia	Rio	C789YP78	278900
2	6	Ford	Focus	O987TP78	258400

Рисунок 1.3 — Вывод данных об автомобилях, которые имеют пробег более 250 тысяч километров и которые не проходили ТО в текущем году.

1.1.3 Сколько раз каждый пассажир воспользовался услугами таксопарка?

```

1      SELECT
2      p.id,
3      p.full_name,
4      p.phone,
5      COUNT(r.id) AS ride_count
6      FROM taxi_service.passenger AS p
7      JOIN taxi_service.ride AS r
8      ON p.id = r.passenger_id
9      GROUP BY p.id, p.full_name, p.phone
10     ORDER BY ride_count DESC

```





	id [PK] integer 	full_name text 	phone text 	ride_count bigint 
1	11	Калинин Андрей Юрьевич	+79111234567	7
2	4	Смирнова Анна Владимировна	+79004445566	6
3	10	Королёва Евгения Александровна	+79005551234	6
4	2	Кузнецова Ольга Николаевна	+79002223344	4
5	14	Ящерова Ирина Михайловна	+79444567890	4
6	9	Романова Светлана Петровна	+79004447788	4
7	1	Петров Петр Петрович	+79001112233	4
8	8	Киселёв Олег Николаевич	+79003336677	3
9	6	Воробьёв Иван Сергеевич	+79001114455	3
10	13	Михайлов Василий Петрович	+79333456789	3
11	5	Новиков Алексей Николаевич	+79005556677	3
12	3	Морозов Дмитрий Сергеевич	+79003334455	2
13	12	Левкина Екатерина Дмитриевна	+79222345678	2
14	7	Бабушкина Татьяна Юрьевна	+79002225566	2

Рисунок 1.4 — Вывод данных о количестве поездок каждого пассажира.

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

```

1  SELECT
2      p.id,
3      p.full_name,
4      p.phone,
5      COUNT(r.id) AS ride_count
6  FROM taxi_service.passenger AS p
7  JOIN taxi_service.ride AS r
8  ON p.id = r.passenger_id
9      GROUP BY p.id, p.full_name, p.phone
10     ORDER BY ride_count DESC

```



```
11      LIMIT 1;
```

	id [PK] integer	full_name text	phone text	ride_count bigint
1	11	Калинин Андрей Юрьевич	+79111234567	7

Рисунок 1.5 — Вывод данных пассажира, который воспользовался услугами таксопарка максимальное число раз.

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

```
1      SELECT
2      e.id,
3      e.full_name,
4      e.phone,
5      e.address,
6      c.brand,
7      c.model,
8      c.price
9  FROM taxi_service.employee e
10 JOIN taxi_service.car_assignment ca ON e.id = ca.
    driver_id
11 JOIN taxi_service.car c ON ca.car_id = c.id
12 WHERE c.price = (SELECT MAX(price) FROM taxi_service.
    car)
13 LIMIT 1;
```

	id integer	full_name text	phone text	address text	brand text	model text	price numeric (10,2)
1	9	Николаев Сергей Иванович	+79012345678	ул. Гоголя, д. 5	Toyota	RAV4	1800000.00

Рисунок 1.6 — Вывод данных о водителе, который ездит на самом дорогом автомобиле.

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

```
1  SELECT
2    p.id,
3    p.full_name,
4    p.phone
5  FROM taxi_service.passenger p
6  WHERE p.id IN (
7    SELECT passenger_id
8    FROM taxi_service.ride
9    GROUP BY passenger_id
10   HAVING COUNT(DISTINCT driver_id) = 1
11   AND COUNT(*) > 1
12 );
```

	id [PK] integer	full_name text	phone text
1	10	Королёва Евгения Александровна	+79005551234
2	11	Сидоров Виктор Алексеевич	+79001234567

Рисунок 1.7 — Вывод данных пассажира, который всегда ездит с одним и тем же водителем.

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

```
1  SELECT
2    c.id,
3    c.brand,
4    c.model,
5    c.plate_number,
```

```

6      cm.mileage
7  FROM taxi_service.car c
8  JOIN taxi_service.car_mileage cm ON c.id = cm.car_id
9  JOIN (
10     SELECT
11         c.brand,
12         AVG(cm.mileage) AS avg_mileage
13     FROM taxi_service.car c
14     JOIN taxi_service.car_mileage cm ON c.id = cm.
15         car_id
16     GROUP BY c.brand
17 ) avg_by_brand ON c.brand = avg_by_brand.brand
18 WHERE cm.mileage > avg_by_brand.avg_mileage
ORDER BY c.brand, cm.mileage DESC;

```

	id integer	brand text	model text	plate_number text	mileage integer
1	1	Hyundai	Solaris	A123BC77	220000
2	3	Kia	Rio	C789YP78	350000
3	14	Toyota	RAV4	X888PT78	300000
4	2	Toyota	Camry	B456EK77	250000
5	5	Volkswagen	Polo	M6540P77	250000

Рисунок 1.8 — Вывод данных о автомобилях, которые имеют пробег больше среднего пробега для своей марки.

1.2 Запросы на модификацию данных

1.2.1 UPDATE: Обновление цен на тарифы

```

1  UPDATE fare_history

```

```
2 SET price_per_km = price_per_km * 1.1
3 WHERE end_date > CURRENT_DATE;
```

1.2.2 INSERT: Добавление новых автомобилей в парк

```
1 INSERT INTO car (brand, model, country_of_origin,
2 production_year, plate_number, price,
3 employee_owner_id, park_owner_id)
4 SELECT
5 c.brand,
6 c.model,
7 c.country_of_origin,
8 c.production_year,
9 c.plate_number,
10 c.price,
11 c.employee_owner_id,
12 c.park_owner_id
FROM taxi_service.car
WHERE id IN (SELECT id FROM transferred_cars);
```

1.2.3 DELETE: Удаление штрафов для поездок с максимальным рейтингом

```
1 DELETE FROM fine
2 WHERE ride_id IN (
3 SELECT r.id
4 FROM ride r
5 JOIN review rv ON r.id = rv.ride_id
6 WHERE rv.rating = 5
7 );
```

1.3 Представления

1.3.1 Представление 1: Незанятые водители

```
1  CREATE OR REPLACE VIEW free_drivers AS
2  SELECT
3      e.id,
4      e.full_name ,
5      e.phone ,
6      e.address ,
7      p.name AS position_name ,
8      dl.categories ,
9      dl.license_number
10 FROM
11     employee e
12 JOIN
13     position p ON e.position_id = p.id
14 LEFT JOIN
15     driving_license dl ON e.id = dl.driver_id
16 WHERE
17     p.name = '          ',
18     AND e.id NOT IN (
19         SELECT DISTINCT driver_id
20         FROM ride
21         WHERE status = 'in_progress'
22     )
23     AND e.id NOT IN (
24         SELECT DISTINCT driver_id
25         FROM car_assignment
26         WHERE CURRENT_DATE BETWEEN start_date AND
27             end_date
28     );
```

	id integer	full_name text	phone text	address text	position_name text	categories text[]	license_number text
1	4	Смирнов Иван Петрович	+79006667788	ул. Ломоносова, д. 3	Водитель	{B,C}	BZ1999334
2	5	Ковалев Павел Дмитриевич	+79007778899	ул. Гагарина, д. 4	Водитель	{B,C}	BZ1151334
3	6	Исаева Елена Владимировна	+79008889900	пр. Ленина, д. 22	Водитель	{B,C}	BZ1188334
4	10	Андреева Ольга Викторовна	+79023456789	пр. Невский, д. 120	Водитель	{B,C}	FG5432109
5	11	Соколов Дмитрий Андреевич	+79034567890	ул. Дзержинского, д. 8	Водитель	{B,D}	HJ2468013

Рисунок 1.9 — Представление 1: Незанятые водители

1.3.2 Представление 2: Зарплата всех водителей за вчерашний день

```

1      CREATE OR REPLACE VIEW driver_salary_yesterday AS
2  SELECT
3      e.id,
4      e.full_name,
5      CASE
6          WHEN sr.rate_type = '
              ' THEN sr.value / 30
7          WHEN sr.rate_type = '
              ' THEN
8              (SELECT COALESCE(SUM(EXTRACT(EPOCH FROM (
                  end_time - start_time)) / 3600), 0) * sr
9                  .value
10                 FROM ride
11                 WHERE driver_id = e.id
12                     AND DATE(end_time) = CURRENT_DATE -
13                         INTERVAL '1 day'
14                     AND status = 'completed')
15          WHEN sr.rate_type = '
              ' THEN
16              (SELECT COALESCE(SUM(distance), 0) * sr.
                  value
17                 FROM ride
18                 WHERE driver_id = e.id

```

```

17         AND DATE(end_time) = CURRENT_DATE -
18             INTERVAL '1 day'
19         AND status = 'completed')
20     ELSE 0
21 END AS salary
22 FROM
23     employee e
24 JOIN
25     position p ON e.position_id = p.id
26 LEFT JOIN
27     salary_rate sr ON e.id = sr.driver_id AND
28     CURRENT_DATE - INTERVAL '1 day' BETWEEN sr.
    start_date AND sr.end_date
WHERE
    p.name = '          ';

```

	id [PK] integer	full_name text	salary double precision
1	1	Иванов Иван Иванович	2000
2	2	Петрова Мария Сергеевна	1466.666666666667
3	3	Сидоров Алексей Олегович	468
4	4	Смирнов Иван Петрович	0
5	5	Ковалев Павел Дмитриевич	0
6	6	Исаева Елена Владимировна	0
7	9	Николаев Сергей Иванович	0
8	10	Андреева Ольга Викторовна	0
9	11	Соколов Дмитрий Андреевич	0

Рисунок 1.10 — Представление 2: Зарплата всех водителей за вчерашний день

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были освоены навыки создания сложных SQL-запросов в PostgreSQL для получения данных из базы таксопарка. Выполнено создание представлений для упрощения доступа к часто используемой информации, например, о незанятых водителях. Разработаны запросы с группировкой и подзапросами для анализа данных о поездках, водителях и автомобилях. Освоена оптимизация запросов с помощью индексов, что позволяет ускорить выборку данных. Также изучена модификация данных (добавление, обновление, удаление) с использованием подзапросов.