Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №3.2 «Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными»

по дисциплине «Проектирование и реализация баз данных»

Автор: Корниенко М.Ю.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы	3
Практическое задание	
Вариант 12. БД «Прокат автомобилей»	3
Рисунок 1 — Схема логической модели базы данных	4
Листинг дампа	5
Вывод	4

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

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 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.

1. Запросы к БД.

Какой автомобиль находился в прокате максимальное количество часов? Листинг:

```
WITH TotalHours AS (
    SELECT
         r.car id,
         m.brand AS model brand,
         SUM(EXTRACT(EPOCH FROM (r.end date - r.start date)) / 3600) AS total hours
    FROM contracts r
    INNER JOIN cars c ON r.car id = c.id
    INNER JOIN models m ON c.model id = m.id
    GROUP BY r.car id, m.brand
SELECT th.car id, th.model brand, th.total hours
FROM TotalHours th
WHERE th.total hours = (
    SELECT MAX (total hours)
    FROM TotalHours
);
        Query Query History
         1 WITH TotalHours AS (
              SELECT
         3
                  r.car_id,
         4
                  m.brand AS model_brand,
                  SUM(EXTRACT(EPOCH FROM (r.end_date - r.start_date)) / 3600) AS total_hours
              FROM contracts r
         6
               INNER JOIN cars c ON r.car_id = c.id
              INNER JOIN models m ON c.model_id = m.id
         8
        9
              GROUP BY r.car_id, m.brand
        10 )
        11 SELECT th.car_id, th.model_brand, th.total_hours
        12 FROM TotalHours th
        13 WHERE th.total_hours = (
        14
              SELECT MAX(total_hours)
               FROM TotalHours
        15
        16 );
        17
        Data Output Messages Notifications
        异 ┗ ∨ 🖺 ∨ 盲 曷 🕹 ~
                    model_brand
                                    total_hours
                 character varying (100)
            integer
                                    numeric
                  1 Lada
                                    8736.00000000000000000
```

Автомобили какой марки чаще всего брались в прокат?

```
FROM RentalCounts rc
WHERE rc.rental count = (
     SELECT MAX (rental count)
     FROM RentalCounts
        Query Query History
            WITH RentalCounts AS (
         2
                SELECT
                    m.brand AS model_brand,
         3
                    COUNT(c.id) AS rental_count
               FROM contracts ct
         6
               INNER JOIN cars c ON ct.car_id = c.id
         7
               INNER JOIN models m ON c.model_id = m.id
         8
                GROUP BY m. brand
         9 )
        10 SELECT rc.model_brand, rc.rental_count
        11
            FROM RentalCounts rc
            WHERE rc.rental_count = (
        13
                SELECT MAX(rental_count)
        14
                FROM RentalCounts
            );
        15
        16
        Data Output Messages Notifications
        =+ 🖺 🗸 📋 🗸 📋
                             a ± ~
                               rental_count
             model_brand
             character varying (100)
                               bigint
        1
             BMW
        2
             Kia
        3
                                         1
             Lada
        4
                                         1
             Mercedes
        5
             Toyota
```

);

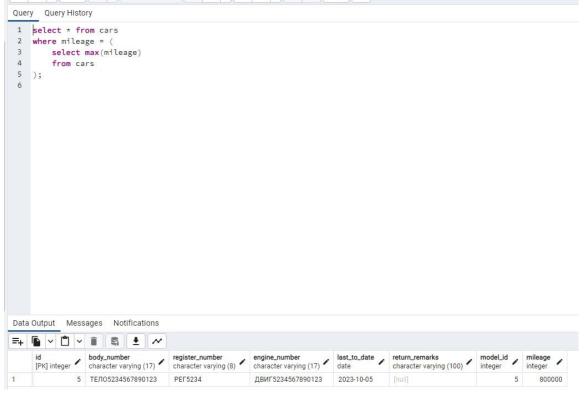
Определить убытки от простоя автомобилей за вчерашний день. Листинг:

```
WITH DowntimePeriods AS (
    SELECT
        car.id AS car id,
        models.id AS model id,
        models.price AS price,
        GREATEST (contracts.end date, NOW() - INTERVAL '1 day') AS end date,
            LEAD(contracts.start date) OVER (PARTITION BY car.id ORDER BY
contracts.start date),
            NOW()
        ) AS next_start_date
    FROM cars car
    LEFT JOIN contracts ON car.id = contracts.car id
    LEFT JOIN models ON car.model id = models.id
DowntimeCalculations AS (
    SELECT
        dp.car_id,
        dp.model id,
```

```
Query Query History
1 WITH DowntimePeriods AS (
2
       SELECT
3
           car.id AS car_id,
           models.id AS model_id,
 4
 5
           models.price AS price,
 6
           GREATEST(contracts.end_date, NOW() - INTERVAL '1 day') AS end_date,
 7
           COALESCE (
 8
               LEAD(contracts.start_date) OVER (PARTITION BY car.id ORDER BY contracts.start_date),
9
               NOW()
10
           ) AS next_start_date
11
       FROM cars car
12
        LEFT JOIN contracts ON car.id = contracts.car_id
13
        LEFT JOIN models ON car.model_id = models.id
14 ),
15 DowntimeCalculations AS (
16
       SELECT
17
           dp.car_id,
18
           dp.model_id,
19
           dp.price,
20
           EXTRACT(EPOCH FROM (dp.next_start_date - dp.end_date)) / 3600 AS downtime_hours
21
        FROM DowntimePeriods dp
22
        WHERE dp.next_start_date > dp.end_date
23 )
24 SELECT
        SUM(dp.price * dp.downtime_hours) AS total_downtime_cost
26 FROM DowntimeCalculations dp;
Data Output Messages Notifications
    total_downtime_cost
    numeric
1
```

Вывести данные автомобиля, имеющего максимальный пробег.

```
select * from cars
where mileage = (
    select max(mileage)
    from cars
);
```



Какой автомобиль суммарно находился в прокате дольше всех.

```
select cars.*
from contracts
join cars on cars.id = contracts.car_id
where (end_date - start_date) = (
    select max(end_date - start_date)
    from contracts
);
```



Определить, каким количеством автомобилей каждой марки и модели владеет компания.

Листинг:

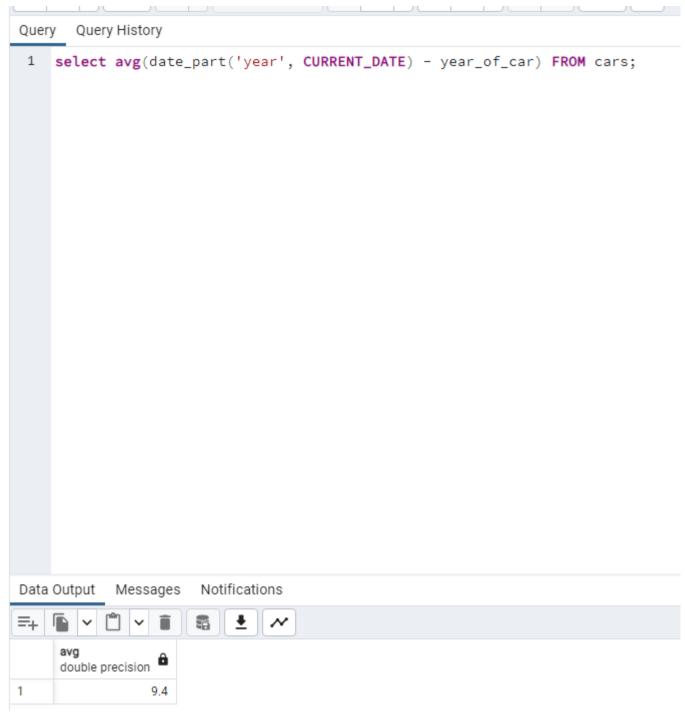
```
select models.brand as "brand", count(*) AS "number"
join models on cars.model id = models.id
group by models.brand;
       Query Query History
        1 select models.brand as "brand", count(*) AS "number"
           join models on cars.model_id = models.id
           group by models.brand;
        Data Output Messages Notifications
                                *
            brand
                               number
            character varying (100)
                                     0
       1
            Mercedes
                                      1
       2
                                      1
            Lada
       3
            Kia
                                      1
       4
            BMW
                                      1
```

Определить средний "возраст" автомобилей компании.

Листинг:

Toyota

```
select avg(date part('year', CURRENT DATE) - year of car) FROM cars;
```

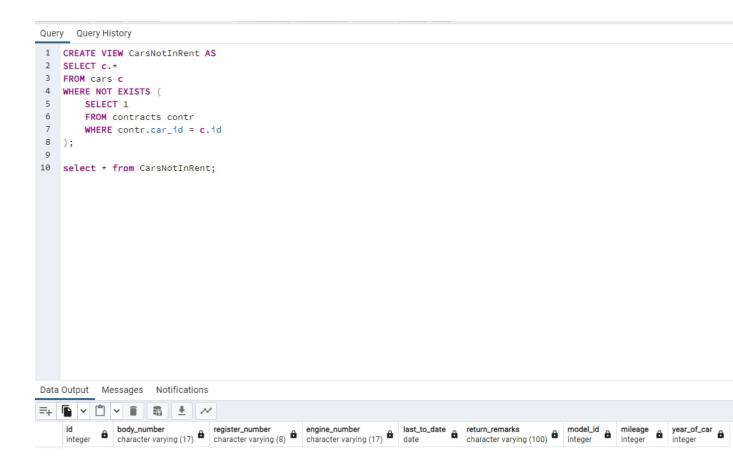


2. Создание представлений:

Какой автомобиль ни разу не был в прокате? Листинг:

```
CREATE VIEW CarsNotInRent AS
SELECT C *
```

```
SELECT c.*
FROM cars c
WHERE NOT EXISTS (
         SELECT 1
         FROM contracts contr
         WHERE contr.car_id = c.id
);
```



Вывести данные клиентов, не вернувших автомобиль вовремя.

```
CREATE VIEW ExpiredClients AS

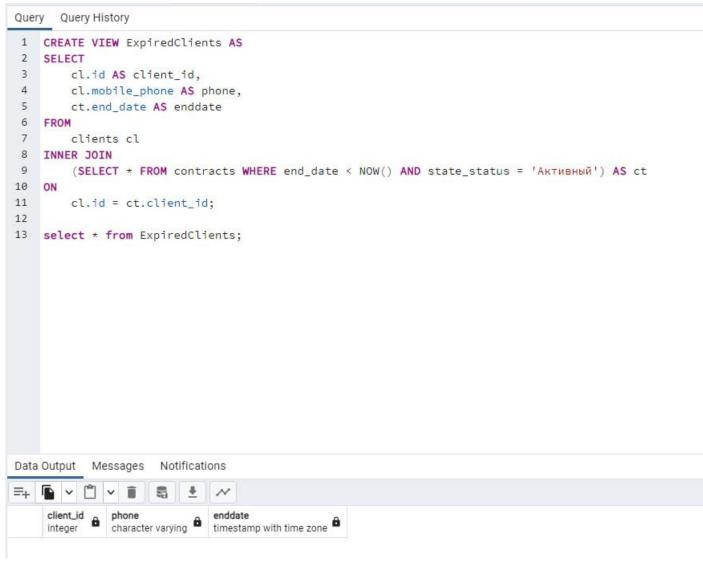
SELECT

    cl.id AS client_id,
    cl.mobile_phone AS phone,
    ct.end_date AS enddate

FROM
    clients cl

INNER JOIN
    (SELECT * FROM contracts WHERE end_date < NOW() AND state_status = 'Активный') AS

ct
ON
    cl.id = ct.client_id;
```



з. Запросы на модификацию данных:

Добавление работника на должность с наибольшим количеством работников Листинг:

```
INSERT INTO workers (id, name, passport_data, position_id)

SELECT 6, 'Виталий Витальев', 12345678, (

SELECT p.id

FROM positions p

JOIN workers w ON p.id = w.position_id

GROUP BY p.id

ORDER BY COUNT(w.id) DESC

LIMIT 1

);
```

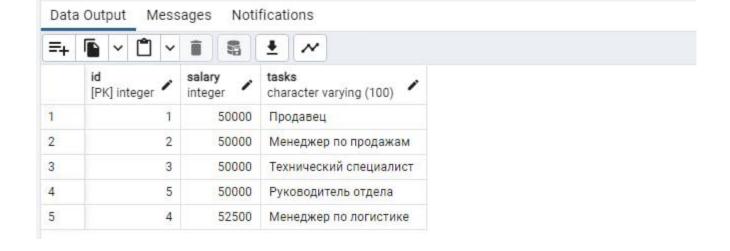
```
Query
        Query History
 1
     INSERT INTO workers (id, name, passport_data, position_id)
 2
     SELECT 6, 'Виталий Витальев', 12345678, (
 3
         SELECT p.id
 4
         FROM positions p
 5
         JOIN workers w ON p.id = w.position_id
 6
         GROUP BY p.id
 7
         ORDER BY COUNT(w.id) DESC
 8
         LIMIT 1
 9
     );
10
11
     select * from workers;
Data Output
             Messages
                         Notifications
=+
                                        passport_data
                                                            position_id
     [PK] integer
                   character varying (50)
                                        character varying (12)
                                                            integer
                   Мария Сергеева
                                        123456789012
1
                                                                      1
2
                2
                   Петр Алексеев
                                        234567890123
                                                                      2
3
                   Светлана Васильева
                                        345678901234
                                                                      3
                3
4
                   Андрей Петров
                                        456789012345
                                                                      4
5
                   Екатерина Иванова
                                        567890123456
                                                                      5
                5
6
                   Виталий Витальев
                                        12345678
                                                                      5
```

Повышение зарплаты для должности, на которой суммарно получают меньше всего

```
UPDATE positions
SET salary = salary * 1.05
WHERE id = (
```

```
SELECT p.id
FROM positions p
LEFT JOIN workers w ON p.id = w.position_id
GROUP BY p.id
ORDER BY SUM(p.salary) ASC
LIMIT 1
);
```

```
Query Query History
    UPDATE positions
1
2
    SET salary = salary * 1.05
    WHERE id = (
3
4
        SELECT p.id
 5
        FROM positions p
6
        LEFT JOIN workers w ON p.id = w.position_id
7
        GROUP BY p.id
8
        ORDER BY SUM(p.salary) ASC
9
        LIMIT 1
10
    );
11
12
    select * from positions;
```



Удаление неактивных контрактов с конца которых прошло более 1 года

```
SELECT cl.id
      FROM clients cl
      LEFT JOIN contracts ct ON cl.id = ct.client id
      GROUP BY cl.id
      HAVING MAX(ct.end date) < NOW() - INTERVAL '1 year' AND MAX(ct.state status) !=</pre>
'Активный'
);
         Query Query History
            DELETE FROM contracts
          2
             WHERE client_id IN (
                 SELECT cl.id
          3
          4
                 FROM clients cl
          5
                 LEFT JOIN contracts ct ON cl.id = ct.client_id
                 GROUP BY cl.id
          6
          7
                 HAVING MAX(ct.end_date) < NOW() - INTERVAL '1 year' AND MAX(ct.state_status) != 'Активный'
          8
             );
         10 select * from contracts;
         Data Output Messages Notifications
         client_id /
                                                                                                                            worker_id ,
                                                                                                                                      total_cost /
                          start_date
                                               end_date
                                                                    state_status
                                                                                      paid_status
              [PK] integer
                                                                    character varying (50)
                         timestamp with time zone
                                               timestamp with time zone
                                                                                      character varying (50)
                                                                                                        integer
                                                                                                                  integer
                                                                                                                            integer
                                                                                                                                      integer
                       1 2023-01-01 00:00:00+03
                                               2023-12-31 00:00:00+03
                                                                                      Оплачено
                                                                    Активный
         2
                       2 2023-01-02 00:00:00+03
                                               2023-12-30 00:00:00+03
                                                                                                               2
                                                                                                                         2
                                                                                                                                   2
                                                                    Завершён
                                                                                      Не оплачено
         3
                       3 2023-01-03 00:00:00+03
                                               2023-12-29 00:00:00+03
                                                                    Активный
                                                                                      Оплачено
                                                                                                               3
                                                                                                                         3
                                                                                                                                   3
         4
                       4 2023-01-04 00:00:00+03
                                               2023-12-28 00:00:00+03
                                                                    Завершён
                                                                                      Оплачено
                                                                                                               4
                                                                                                                         4
                                                                                                                                   4
```

5

Создание индексов

5 2023-01-05 00:00:00+03

Листинг:

DELETE FROM contracts
WHERE client id IN (

```
CREATE INDEX idx_cars_model_id ON cars(model_id);
CREATE INDEX idx_contracts_id_end_date ON contracts(id, end_date);
CREATE INDEX idx_workers_name ON workers(name);
CREATE INDEX idx workers name position id ON workers(name, position id);
```

2023-12-27 00:00:00+03

Сравнение времени работы с индексами и без:

1. Синдексами

```
Query
       Query History
 2
         SELECT
 3
             car.id AS car_id,
 4
             models.id AS model_id,
 5
             models.price AS price,
              \label{eq:greatest} \textbf{GREATEST}(\texttt{contracts.end\_date}, \ \texttt{NOW}() \ - \ \texttt{INTERVAL} \ '1 \ \texttt{day'}) \ \textbf{AS} \ \texttt{end\_date},
 6
 7
              COALESCE (
 8
                  LEAD(contracts.start_date) OVER (PARTITION BY car.id ORDER BY contracts.start_date),
9
                  NOW()
10
             ) AS next_start_date
         FROM cars car
11
12
         LEFT JOIN contracts ON car.id = contracts.car_id
13
         LEFT JOIN models ON car.model_id = models.id
14
    ),
    DowntimeCalculations AS (
15
16
         SELECT
17
             dp.car_id,
18
             dp.model_id,
19
             dp.price,
20
              EXTRACT(EPOCH FROM (dp.next_start_date - dp.end_date)) / 3600 AS downtime_hours
21
         FROM DowntimePeriods dp
22
         WHERE dp.next_start_date > dp.end_date
23 )
24 SELECT
25
         SUM(dp.price * dp.downtime_hours) AS total_downtime_cost
26 FROM DowntimeCalculations dp;
27
Data Output Messages Notifications
     total_downtime_cost
     numeric
Total rows: 1 of 1 Query complete 00:00:00.051
```

2. Без индексов

```
Query History
Query
 2
        SELECT
 3
            car.id AS car_id,
 4
            models.id AS model_id,
 5
            models.price AS price,
            GREATEST(contracts.end_date, NOW() - INTERVAL '1 day') AS end_date,
 6
 7
            COALESCE (
 8
                LEAD(contracts.start_date) OVER (PARTITION BY car.id ORDER BY contracts.start_date)
 9
                NOW()
10
            ) AS next_start_date
11
        FROM cars car
12
        LEFT JOIN contracts ON car.id = contracts.car_id
13
        LEFT JOIN models ON car.model_id = models.id
14
   DowntimeCalculations AS (
15
16
        SELECT
17
            dp.car_id,
18
            dp.model_id,
            dp.price,
19
20
            EXTRACT(EPOCH FROM (dp.next_start_date - dp.end_date)) / 3600 AS downtime_hours
21
        FROM DowntimePeriods dp
22
        WHERE dp.next_start_date > dp.end_date
23
24
    SELECT
25
        SUM(dp.price * dp.downtime_hours) AS total_downtime_cost
26
    FROM DowntimeCalculations dp;
27
Data Output Messages Notifications
=+ | • | • | • | • |
                     8
                          <u>*</u> /~
     total_downtime_cost
     numeric
1
Total rows: 1 of 1 Query complete 00:00:00.110
```

Вывод

Во время лабораторной работы я научился использовать разнообразные SQL-запросы для взаимодействия с базой данных, а также узнал, как создавать представления и индексы. Кроме того, я провёл сравнение времени выполнения SELECT-запросов с использованием индексов и без них. Как оказалось, наличие индексов сокращает время выполнения запросов.