Министерство науки и высшего образования Российской Федерации

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

Отчет

по лабораторной работе «Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL»

по дисциплине «Проектирование и реализация баз данных»

Автор: Коротин А.М.

Факультет: ИКТ

Группа: К32391

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

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.

Вариант 14

Задание 2. Создать запросы:

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

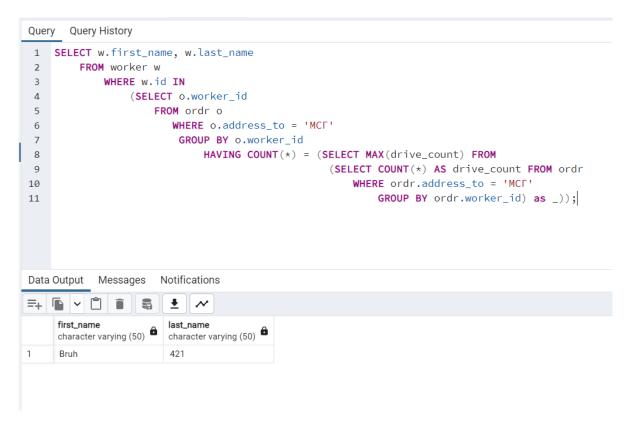
Задание 3. Создать представление:

- содержащее сведения о незанятых на данный момент водителях;
- зарплата всех водителей за вчерашний день.

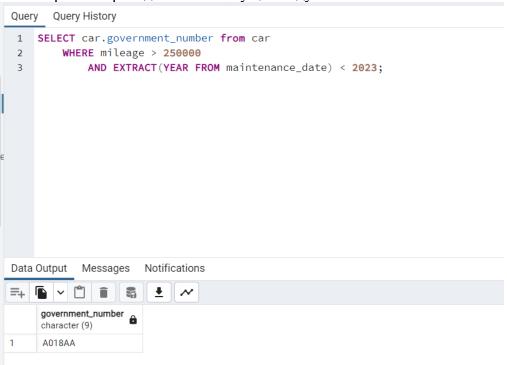
Ход работы:

Задание 2:

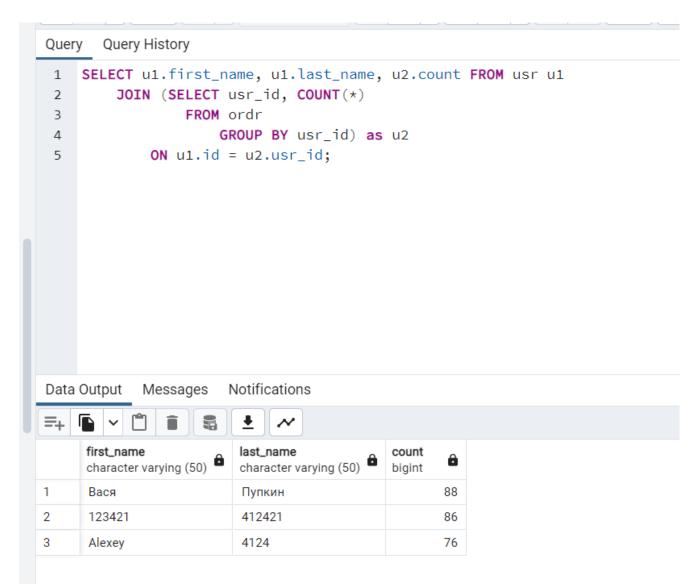
1. Вывести данные о водителе, который чаще всех доставляет пассажиров на заданную улицу.



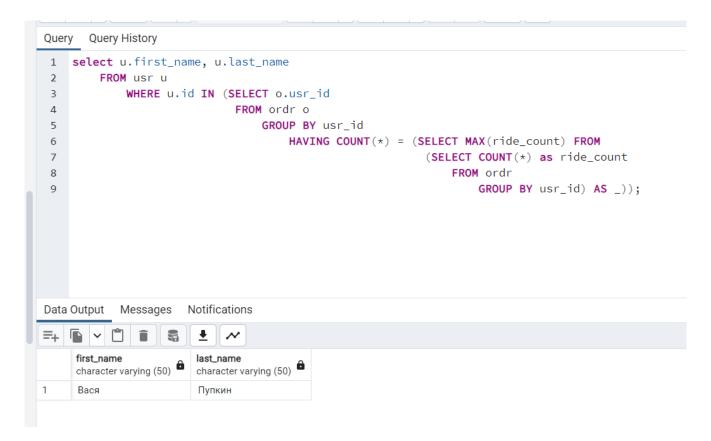
2. Вывести данные об автомобилях, которые имеют пробег более 250 тысяч. километров и которые не проходили ТО в текущем году.



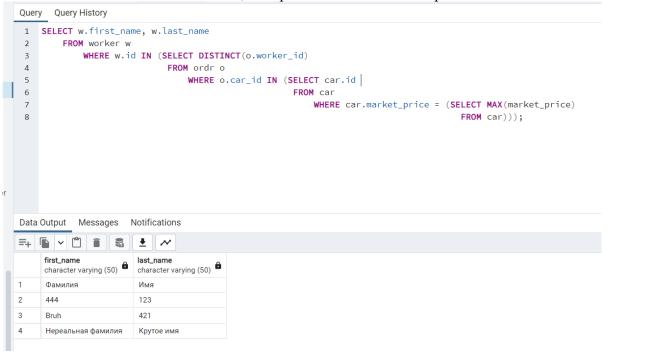
3. Сколько раз каждый пассажир воспользовался услугами таксопарка?



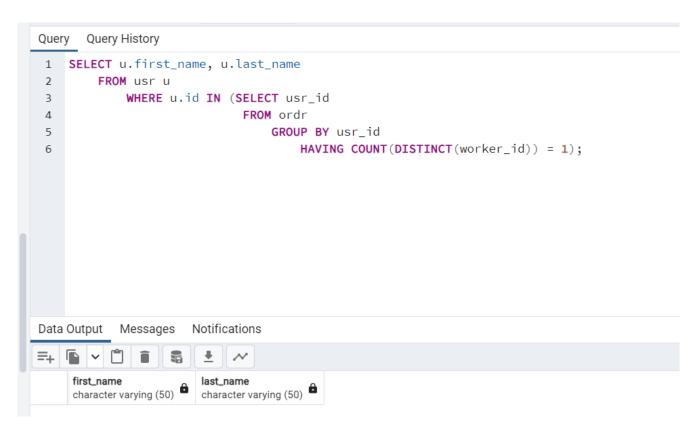
4. Вывести данные пассажира, который воспользовался услугами таксопарка максимальное число раз.



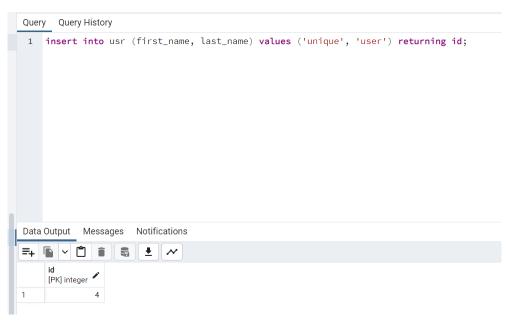
5. Вывести данные о водителе, который ездит на самом дорогом автомобиле.



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



Таких пользователей нет в связи со спецификой генерации данных. Добавим тестовые данные для проверки запроса:

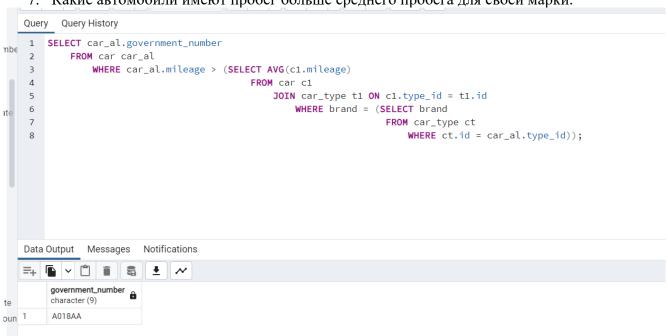




Как можем видеть, запрос работает исправно

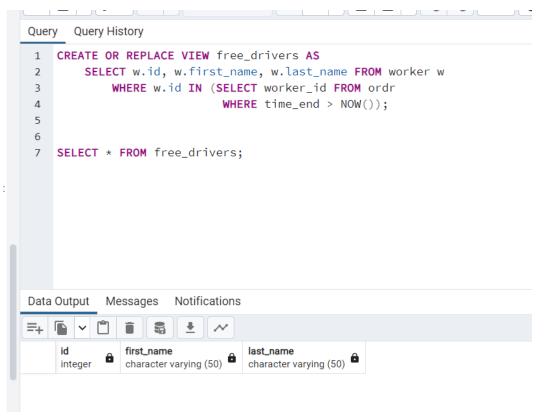


7. Какие автомобили имеют пробег больше среднего пробега для своей марки.

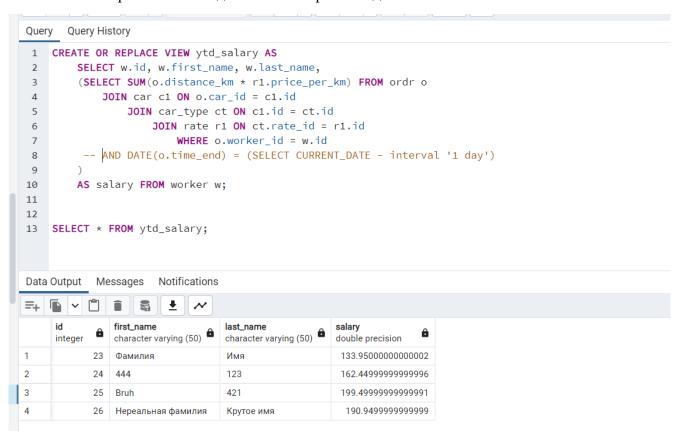


Задание 3 Создать представление:

1. содержащее сведения о незанятых на данный момент водителях



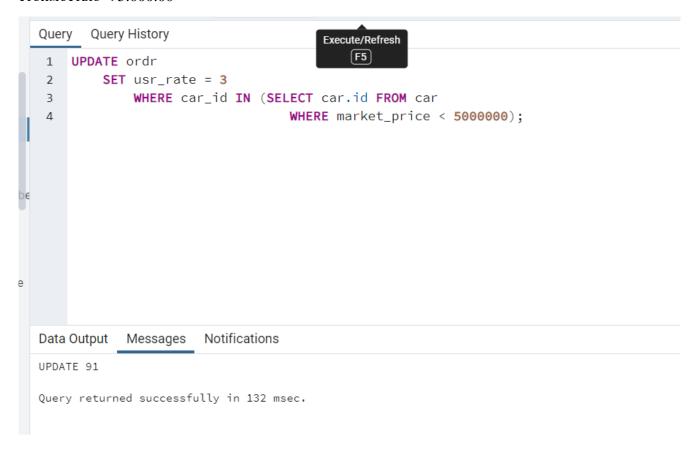
2. зарплата всех водителей за вчерашний день



Условие на вчерашний день опущено, потому что у всех водителей 0 зарплаты в этот день в связи со спецификой генерации данных

2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

1) Поставить пользовательскую оценку 3 для заказов, которые были исполнены на машинах, стоимостью < 5.000.00



2) Удалить все заказы, исполненные на Renault Logan

```
Query Query History
   DELETE FROM ordr
        WHERE car_id IN (SELECT car.id
2
                             FROM car
3
                                  WHERE car.type_id IN (SELECT ct.id
4
                                                           FROM car_type ct
5
                                                                WHERE ct.model = 'Logan' AND
6
                                                                      ct.brand = 'Renault'));
Data Output
           Messages Notifications
DELETE 160
Query returned successfully in 132 msec.
```

3) Пользователя с наиболее встречающимся именем в заказах.

```
Query Query History

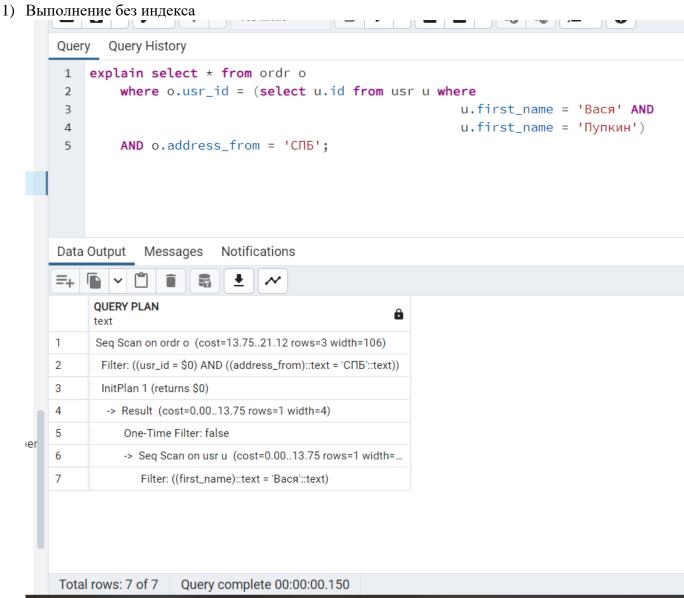
1 INSERT INTO usr (first_name, last_name)
2 SELECT u.first_name, '-' from usr u
3 JOIN ordr o ON u.id = o.usr_id
4 GROUP BY u.first_name
5 ORDER BY COUNT(*) DESC
6 LIMIT 1;

Data Output Messages Notifications

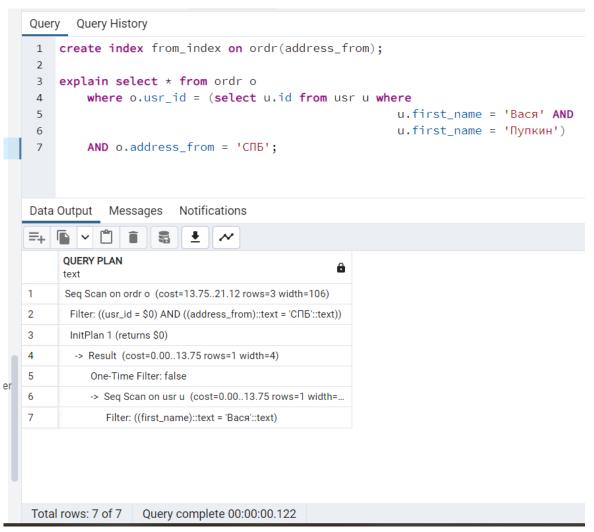
INSERT 0 1

Query returned successfully in 124 msec.
```

3. Создать простые и составные индексы и сравнить время выполнения



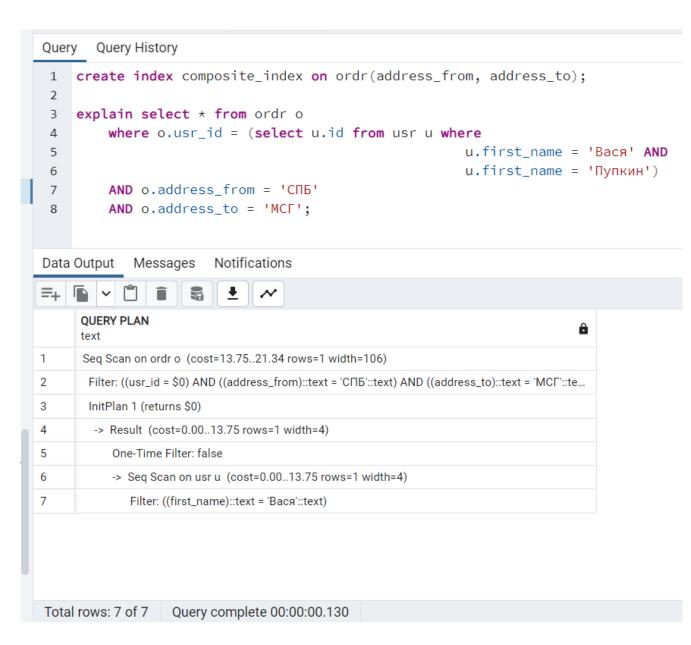
2) Простой индекс



3) Составной индекс

```
Query
        Query History
     explain select * from ordr o
 1
          where o.usr_id = (select u.id from usr u where
 2
                                                                   u.first_name = 'Вася' AND
 3
 4
                                                                   u.first_name = 'Пупкин')
 5
          AND o.address_from = 'CΠ5'
 6
          AND o.address_to = 'MCF';
Data Output Messages
                          Notifications
=+ 6 ~ 1 1
      QUERY PLAN
                                                                                     ô
      text
1
      Seq Scan on ordr o (cost=13.75..21.34 rows=1 width=106)
2
       Filter: ((usr_id = $0) AND ((address_from)::text = 'CПБ'::text) AND ((address_to)::text = 'MCГ'::te...
3
       InitPlan 1 (returns $0)
        -> Result (cost=0.00..13.75 rows=1 width=4)
4
5
           One-Time Filter: false
6
           -> Seq Scan on usr u (cost=0.00..13.75 rows=1 width=4)
7
              Filter: ((first_name)::text = 'Bacs'::text)
```

Total rows: 7 of 7 Query complete 00:00:00.122



Вывод

В ходе данной лабораторной работы мной были изучены сложные вложенные SQL запросы, а также индексы.

Последние не повлияли на скорость выполнения запросов из-за того, что количество строк в таблицах (порядка 200) не настолько больше, чтобы заметно затруднять поиск.