Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

ОТЧЕТ

по лабораторной работе № 4

Название: **Оптимизация процессов в PostgresSQL**

Дисциплина: Базы данных

Студент ИУ6-33Б (Группа) 19.11.2023 (Подпись, дата) В.К. Залыгин (И.О. Фамилия)

Преподаватель (Подпись, дата) М.А. Скворцова (И.О. Фамилия)

Москва, 2023

Цель

Сформировать у студента понимание методов упрощения работы аналитика с БД.

Ход работы

Часть 1. Функции

```
create or replace function compute_sum_module(modu integer, lim integer)
returns integer as
$$
declare
    curs1 cursor for select payments.id, payments.price from payments;
    id uuid;
    price integer;
    res integer;
begin
    res = 0;
    while lim > 0 loop
        fetch curs1 into id, price;
        if id % modu = 0 then
            res = res + price;
        end if;
        lim = lim - 1;
    end loop;
    return res;
exception when others
then
    return -1;
end
$$
language 'plpgsql'
```

Рисунок 1. Функция вычисления суммы стоимостей услуг с ограничением по количеству, id которых имеет определенную кратность.

Часть 2. Анализ и оптимизация запросов

Будем исследовать и оптимизировать работы 2 запросов.

```
-- Какие квартиры имеют номер < 300
select *
from apartaments
where cast(full_address->>'apartament_number' as integer) < 300;

--Вывести все платежи по квартирам, которые находятся на Льва Толстого, где цена платежа больше 3000
select *
from payments
join apartaments on apartaments.id = payments.apartament_id
where apartaments.full_address->>'street' = 'улица Льва Толстого' and payments.price > 3000;
```

Рисунок 2. Запросы для оптимизации

```
explain
select *
from apartaments
where cast(full_address->>'apartament_number' as integer) < 300;

explain (ANALYZE)
select *
from apartaments
where cast(full_address->>'apartament_number' as integer) < 300;

QUERY PLAN                                                                |
--------------------------------------------------------------------------+
Seq Scan on apartaments  (cost=0.00..43700.00 rows=333333 width=158)      |
  Filter: (((full_address ->> 'apartament_number'::text))::integer < 300)|

QUERY PLAN                                                                                                                                |
------------------------------------------------------------------------------------------------------------------------------------------+
Seq Scan on apartaments  (cost=0.00..43700.00 rows=333333 width=158) (actual time=0.062..327.899 rows=299709 loops=1)|
  Filter: (((full_address ->> 'apartament_number'::text))::integer < 300)                                                                 |
  Rows Removed by Filter: 700292                                                                                                          |
Planning Time: 0.040 ms                                                                                                                   |
Execution Time: 339.606 ms                                                                                                                |
```

Рисунок 3. План и выполнение запроса 1 до создания индекса

```
create index on apartaments (full_address);

explain
select *
from apartaments
where cast(full_address->>'apartament_number' as integer) < 300;

explain (ANALYZE)
select *
from apartaments
where cast(full_address->>'apartament_number' as integer) < 300;

QUERY PLAN                                                                |
--------------------------------------------------------------------------+
Seq Scan on apartaments  (cost=0.00..43700.02 rows=333334 width=158)      |
  Filter: (((full_address ->> 'apartament_number'::text))::integer < 300)|

QUERY PLAN                                                                                                                                |
------------------------------------------------------------------------------------------------------------------------------------------+
Seq Scan on apartaments  (cost=0.00..43700.02 rows=333334 width=158) (actual time=0.031..324.752 rows=299709 loops=1)|
  Filter: (((full_address ->> 'apartament_number'::text))::integer < 300)                                                                 |
  Rows Removed by Filter: 700292                                                                                                          |
Planning Time: 0.043 ms                                                                                                                   |
Execution Time: 336.084 ms                                                                                                                |
```

Рисунок 4. План и выполнение запроса 1 после создания индекса

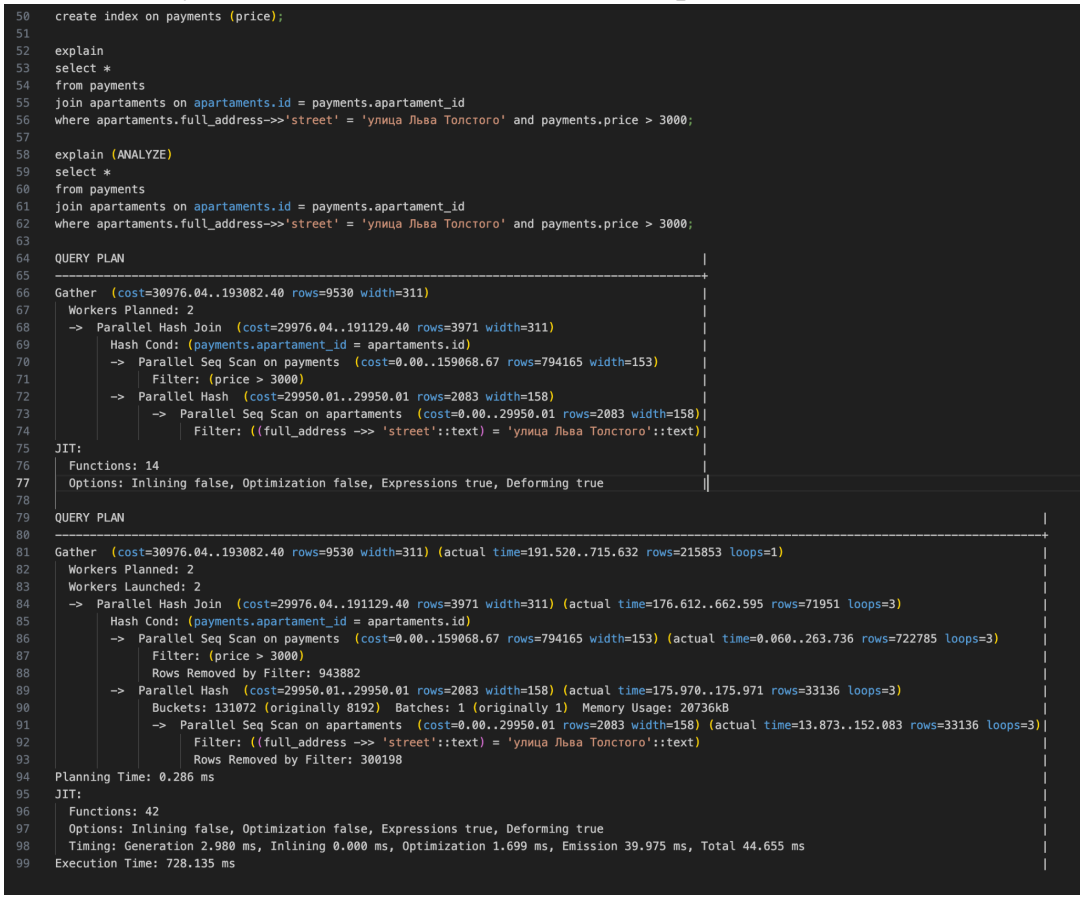
```
explain
select *
from payments
join apartaments on apartaments.id = payments.apartament_id
where apartaments.full_address->>'street' = 'улица Льва Толстого' and payments.price > 3000;

explain (ANALYZE)
select *
from payments
join apartaments on apartaments.id = payments.apartament_id
where apartaments.full_address->>'street' = 'улица Льва Толстого' and payments.price > 3000;
```

```
QUERY PLAN                                                                                                  |
------------------------------------------------------------------------------------------------------------+
Gather  (cost=30976.04..190505.86 rows=8704 width=311)                                                       |
  Workers Planned: 2                                                                                        |
  ->  Parallel Hash Join  (cost=29976.04..188635.46 rows=3627 width=311)                                    |
        Hash Cond: (payments.apartament_id = apartaments.id)                                                |
        ->  Parallel Seq Scan on payments  (cost=0.00..156755.51 rows=725294 width=153)                     |
              Filter: (price > 3000)                                                                        |
        ->  Parallel Hash  (cost=29950.01..29950.01 rows=2083 width=158)                                    |
              ->  Parallel Seq Scan on apartaments  (cost=0.00..29950.01 rows=2083 width=158)               |
                    Filter: ((full_address ->> 'street'::text) = 'улица Льва Толстого'::text)               |
JIT:                                                                                                        |
  Functions: 14                                                                                             |
  Options: Inlining false, Optimization false, Expressions true, Deforming true                             |
```

```
QUERY PLAN                                                                                                                                                                  |
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
Gather  (cost=30976.04..190505.86 rows=8704 width=311) (actual time=243.628..1135.171 rows=215853 loops=1)                                                                  |
  Workers Planned: 2                                                                                                                                                        |
  Workers Launched: 2                                                                                                                                                       |
  ->  Parallel Hash Join  (cost=29976.04..188635.46 rows=3627 width=311) (actual time=209.860..1044.985 rows=71951 loops=3)                                                |
        Hash Cond: (payments.apartament_id = apartaments.id)                                                                                                                |
        ->  Parallel Seq Scan on payments  (cost=0.00..156755.51 rows=725294 width=153) (actual time=0.563..627.313 rows=722785 loops=3)                                    |
              Filter: (price > 3000)                                                                                                                                        |
              Rows Removed by Filter: 943882                                                                                                                                |
        ->  Parallel Hash  (cost=29950.01..29950.01 rows=2083 width=158) (actual time=208.639..208.639 rows=33136 loops=3)                                                  |
              Buckets: 131072 (originally 8192)  Batches: 1 (originally 1)  Memory Usage: 20736kB                                                                          |
              ->  Parallel Seq Scan on apartaments  (cost=0.00..29950.01 rows=2083 width=158) (actual time=25.005..176.026 rows=33136 loops=3)                              |
                    Filter: ((full_address ->> 'street'::text) = 'улица Льва Толстого'::text)                                                                               |
                    Rows Removed by Filter: 300198                                                                                                                          |
Planning Time: 0.235 ms                                                                                                                                                     |
JIT:                                                                                                                                                                        |
  Functions: 42                                                                                                                                                             |
  Options: Inlining false, Optimization false, Expressions true, Deforming true                                                                                             |
  Timing: Generation 3.057 ms, Inlining 0.000 ms, Optimization 11.432 ms, Emission 63.633 ms, Total 78.122 ms                                                               |
Execution Time: 1147.079 ms                                                                                                                                                 |
```

Рисунок 5. План и выполнение запроса 2 до индекса



```
create index on payments (price);

explain
select *
from payments
join apartaments on apartaments.id = payments.apartament_id
where apartaments.full_address->>'street' = 'улица Льва Толстого' and payments.price > 3000;

explain (ANALYZE)
select *
from payments
join apartaments on apartaments.id = payments.apartament_id
where apartaments.full_address->>'street' = 'улица Льва Толстого' and payments.price > 3000;
```

```
QUERY PLAN                                                                                                  |
------------------------------------------------------------------------------------------------------------+
Gather  (cost=30976.04..193082.40 rows=9530 width=311)                                                       |
  Workers Planned: 2                                                                                        |
  ->  Parallel Hash Join  (cost=29976.04..191129.40 rows=3971 width=311)                                    |
        Hash Cond: (payments.apartament_id = apartaments.id)                                                |
        ->  Parallel Seq Scan on payments  (cost=0.00..159068.67 rows=794165 width=153)                     |
              Filter: (price > 3000)                                                                        |
        ->  Parallel Hash  (cost=29950.01..29950.01 rows=2083 width=158)                                    |
              ->  Parallel Seq Scan on apartaments  (cost=0.00..29950.01 rows=2083 width=158)               |
                    Filter: ((full_address ->> 'street'::text) = 'улица Льва Толстого'::text)               |
JIT:                                                                                                        |
  Functions: 14                                                                                             |
  Options: Inlining false, Optimization false, Expressions true, Deforming true                             |
```

```
QUERY PLAN                                                                                                                                                                  |
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
Gather  (cost=30976.04..193082.40 rows=9530 width=311) (actual time=191.520..715.632 rows=215853 loops=1)                                                                  |
  Workers Planned: 2                                                                                                                                                        |
  Workers Launched: 2                                                                                                                                                       |
  ->  Parallel Hash Join  (cost=29976.04..191129.40 rows=3971 width=311) (actual time=176.612..662.595 rows=71951 loops=3)                                                |
        Hash Cond: (payments.apartament_id = apartaments.id)                                                                                                                |
        ->  Parallel Seq Scan on payments  (cost=0.00..159068.67 rows=794165 width=153) (actual time=0.060..263.736 rows=722785 loops=3)                                    |
              Filter: (price > 3000)                                                                                                                                        |
              Rows Removed by Filter: 943882                                                                                                                                |
        ->  Parallel Hash  (cost=29950.01..29950.01 rows=2083 width=158) (actual time=175.970..175.971 rows=33136 loops=3)                                                  |
              Buckets: 131072 (originally 8192)  Batches: 1 (originally 1)  Memory Usage: 20736kB                                                                          |
              ->  Parallel Seq Scan on apartaments  (cost=0.00..29950.01 rows=2083 width=158) (actual time=13.873..152.083 rows=33136 loops=3)                              |
                    Filter: ((full_address ->> 'street'::text) = 'улица Льва Толстого'::text)                                                                               |
                    Rows Removed by Filter: 300198                                                                                                                          |
Planning Time: 0.286 ms                                                                                                                                                     |
JIT:                                                                                                                                                                        |
  Functions: 42                                                                                                                                                             |
  Options: Inlining false, Optimization false, Expressions true, Deforming true                                                                                             |
  Timing: Generation 2.980 ms, Inlining 0.000 ms, Optimization 1.699 ms, Emission 39.975 ms, Total 44.655 ms                                                                |
Execution Time: 728.135 ms                                                                                                                                                  |
```

Рисунок 6. План и выполнение запроса 2 после добавления индекса

Вывод

Было сформировано понимание методов упрощения работы аналитика с БД.

Вопросы

Объяснить, как работают написанные запросы:

SQL-запросы представляют собой команды для работы с базами данных. Процесс выполнения запроса включает анализ запроса, создание оптимального плана выполнения, оптимизацию и фактическое выполнение операций.

2. В чем отличие первичного ключа и уникального индекса:

Первичный ключ уникально идентифицирует каждую запись в таблице и предотвращает наличие дубликатов. Уникальный индекс также гарантирует уникальность значений, но не обязательно является первичным ключом.

3. В каких случаях имеет смысл создавать индексы? Какие колонки следует включать в индекс и почему:

Индексы полезны для ускорения выполнения запросов, особенно при поиске, сортировке и соединении данных. Колонки для индексации выбираются на основе частоты использования в условиях запросов и их уникальности.

4. Рассказать о проблеме фрагментации индексов. Как бороться с фрагментацией:

Фрагментация индексов происходит, когда данные в индексе становятся неупорядоченными. Борьба с фрагментацией включает регулярную переорганизацию или перестройку индексов.

5. В чем разница между Index Scan и Index Seek:

Index Scan означает, что весь индекс просматривается для поиска данных, в то время как Index Seek использует структуру индекса для непосредственного поиска конкретных значений.

6. На что влияет порядок сортировки (ASC\DESC) при создании индекса?

Продемонстрировать это:

Порядок сортировки влияет на то, как данные хранятся в индексе. Например, при поиске в убывающем порядке индекс с DESC может быть эффективнее.

7. Продемонстрировать полезность индекса по выражению:

Индексы по выражению позволяют создавать индексы на вычисляемых столбцах, что может быть полезным для оптимизации запросов с использованием выражений.

8. Продемонстрировать полезность частичного индекса:

Частичные индексы могут содержать только часть данных в столбце, что полезно, если вы часто используете запросы с определенным условием