

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Инженерно-исследовательский факультет

Лабораторная работа №4

**«Запросы на выборку и модификацию данных, представления и
индексы в PostgreSQL»**

Работу выполнил:

Чуб Илья Евгеньевич

Группа К3240 ФИКТ

Проверила:

Горова Марина Михайловна

Санкт-Петербург

2022

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.

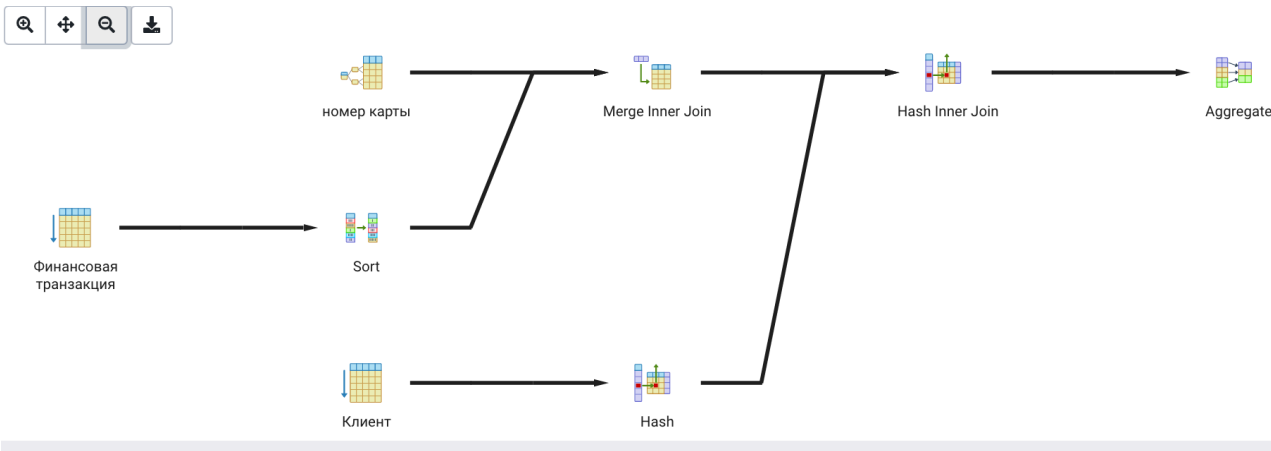
Практическое задание:

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.

Выполнение работы:

- 1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
- 1. Сколько раз заправлял автомобиль каждый из клиентов за заданный период.

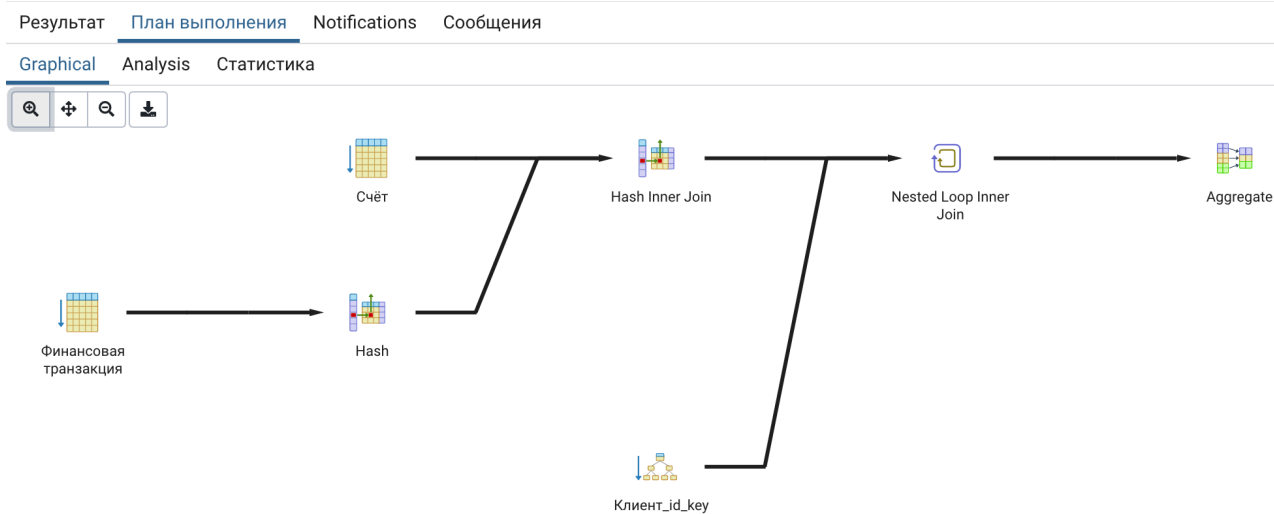
```
1 select
2     client.id as "ID клиента",
3     client."фιο" as "ФИО",
4     count(client.id) as "количество заправок"
5
6 from "Финансовая транзакция" financial_transaction
7
8 join "Счёт" financial_account
9 on financial_account."номер карты" = financial_transaction."номер карты"
10
11 join "Клиент" client
12 on client.id = financial_account."id клиента"
13
14 group by client.id
```



	Результат	План выполнения	Notifications	Сообщения
	ID клиента integer	ФИО text		количество заправок bigint
1	4	Суханов Александр Васильевич		1
2	2	Ильин Филипп Назарович		1
3	3	Иванов Адам Дамирович		3
4	5	Заякина Анна Генадьевна		1
5	1	Костин Артём Георгиевич		2

2. Кто из клиентов не приобретал топливо в июле текущего года?

```
1 select distinct client.id
2 from "Финансовая транзакция" financial_transaction
3
4 join "Счёт" account
5 on account."номер карты" = financial_transaction."номер карты"
6
7 join "Клиент" client
8 on client.id = account."id клиента"
9
10 where financial_transaction."дата и время" not between '2022-07-01' and '2022-08-01'
11
```




Результат

План выполнения

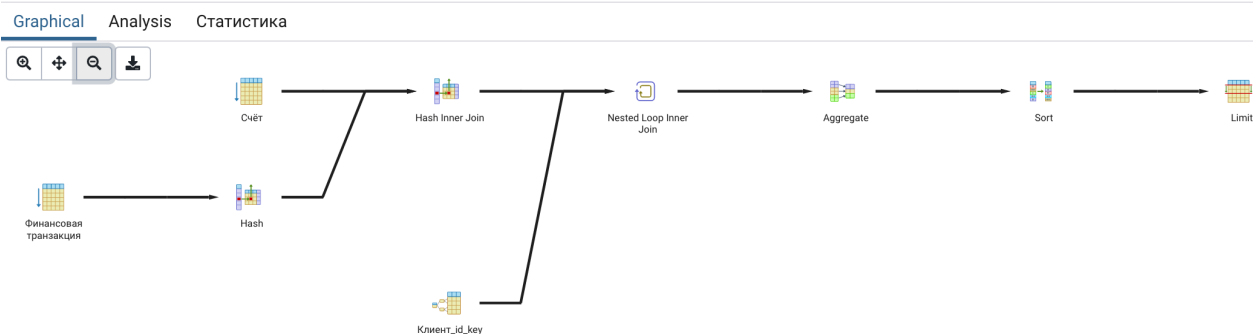
Notifications

Сообщения

	<div>id</div> <div>[PK] integer </div>	
1	4	
2	3	
3	1	
4	5	
5	2	

3. Найти клиента, купившего наибольший объем топлива по всей сети.

```
1 select
2     client.id as "ID клиента",
3     client."фio" as "ФИО клиента",
4     sum(finansial_transaction."количество топлива") as "сумма объёмов купленного топлива"
5
6 from "Финансовая транзакция" finansial_transaction
7
8 join "Счёт" account
9 on account."номер карты" = finansial_transaction."номер карты"
10
11 join "Клиент" client
12 on client.id = account."id клиента"
13
14 group by client.id
15 order by "сумма объёмов купленного топлива" desc
16 limit 1
```



Результат План выполнения Notifications Сообщения

	ID клиента integer	ФИО клиента text	сумма объёмов купленного топлива real
1	3	Иванов Адам Дамирович	3022

4. Вывести данные клиента, купившего топлива на наибольшую сумму в заданный день.

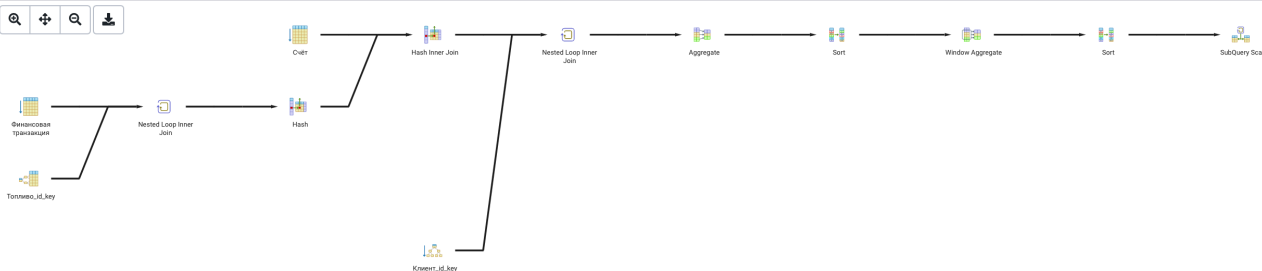
Query Editor

```
1 with pre_result as (  
2     select  
3         client.id,  
4         sum(financial_transaction."количество топлива"),  
5         fuel."вид",  
6         row_number() over(partition by fuel."вид" order by fuel."вид" desc, sum(financial_transaction."количество топлива") desc) as row_number  
7  
8     from "Финансовая транзакция" financial_transaction  
9  
10    join "Счёт" account  
11    on account."номер карты" = financial_transaction."номер карты"  
12  
13    join "Клиент" client  
14    on client.id = account."id клиента"  
15  
16    join "Топливо" fuel  
17    on fuel.id = financial_transaction."id топлива"  
18  
19    where financial_transaction."дата и время" between '2022-05-10 00:00:00' and '2022-06-10 00:00:00'  
20  
21    group by client.id, fuel."вид"  
22    order by sum(financial_transaction."количество топлива") desc  
23 )  
24  
25 select *  
26 from pre_result  
27 where row_number = 1  
28
```

План выполнен < >

План выполнения

Graphical Analysis Статистика

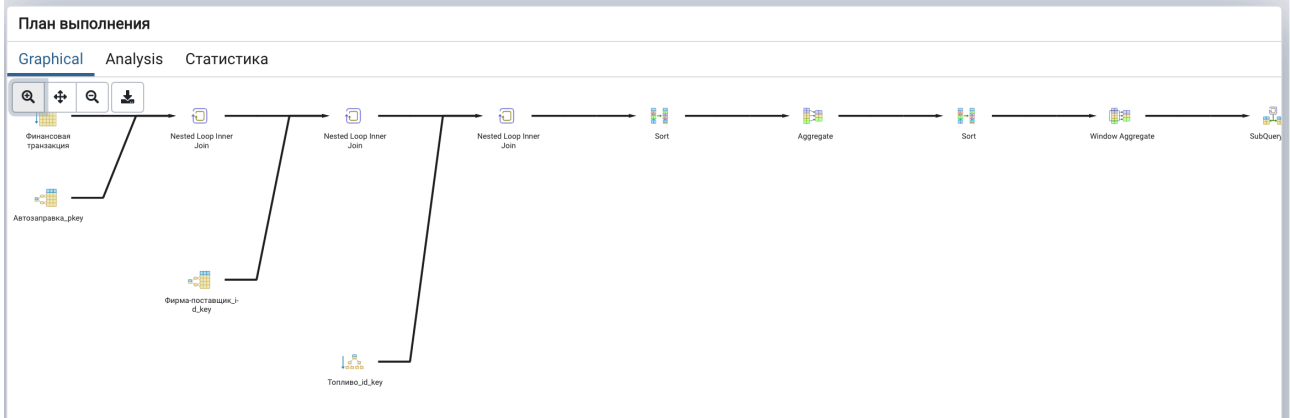


Результат

	id integer	sum real	вид "вид топлива"	row_number bigint
1	3	3022	электричество	1
2	1	62	бензин	1

5. Какое топливо пользуется наибольшим спросом в прошедшем году на АЗС конкретного поставщика?

```
Query Editor
1  with pre_result as (
2      select
3          sum(financial_transaction."количество топлива"),
4          company."название",
5          financial_transaction."id топлива",
6          row_number() over(
7              partition by company."название"
8              order by financial_transaction."id топлива" desc, sum(financial_transaction."количество топлива") desc
9          ) as row_number
10
11     from "Финансовая транзакция" financial_transaction
12
13     join "Автозаправка" gas_station
14     on financial_transaction."id автозаправки" = gas_station.id
15
16     join "Фирма-поставщик" company
17     on gas_station."id поставщика" = company.id
18
19     join "Топливо" fuel
20     on fuel.id = financial_transaction."id топлива"
21
22     where financial_transaction."дата и время" between '2022-01-01 00:00:00' and '2022-12-31 00:00:00'
23     group by company.id, financial_transaction."id топлива"
24 )
25
26 select *
27 from pre_result
28 where row_number = 1
29
```



Результат

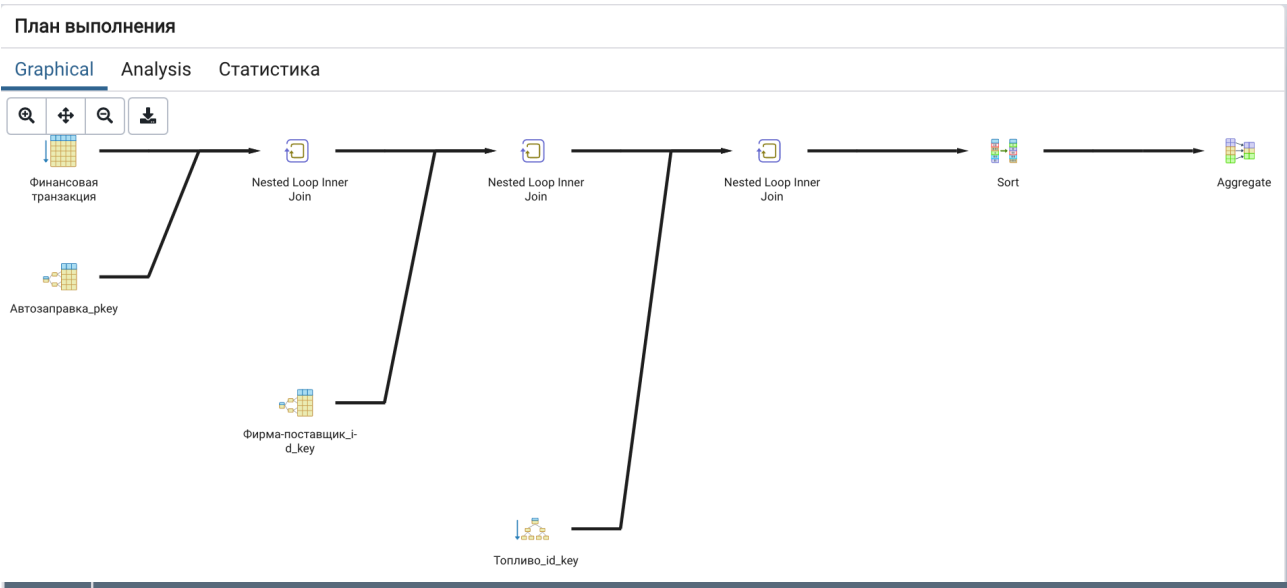
	sum real	название text	id топлива integer	row_number bigint
1	29.5	BP	2	1
2	25	Neste	2	1
3	62	Shell	3	1
4	5022	Ленэнерго	5	1

6. Сколько топлива каждого вида было продано за прошедший месяц по каждому поставщику на каждой АЗС.

Автозаправки/postgres@localhost

Query Editor

```
1 select
2     sum(financial_transaction."количество топлива") as "количество топлива",
3     company."название" as "фирма-поставщик",
4     gas_station.id as "id автозаправки",
5     financial_transaction."id топлива"
6
7 from "Финансовая транзакция" financial_transaction
8
9 join "Автозаправка" gas_station
10 on financial_transaction."id автозаправки" = gas_station.id
11
12 join "Фирма-поставщик" company
13 on gas_station."id поставщика" = company.id
14
15 join "Топливо" fuel
16 on fuel.id = financial_transaction."id топлива"
17
18 where financial_transaction."дата и время" between '2022-05-01 00:00:00' and '2022-05-31 00:00:00'
19 group by financial_transaction."id топлива", gas_station.id, company."название"
20
21 order by company."название"
22
```



Результат

	количество топлива real	фирма-поставщик text	id автозаправки integer	id топлива integer
1	29.5	BP	15	2
2	25	Neste	4	2
3	33	Shell	2	2
4	30	Shell	1	3
5	32	Shell	8	3
6	2022	Ленэнерго	5	5
7	3000	Ленэнерго	13	5

7. Какая из заправок продала топлива на наибольшую сумму по всем автозаправкам за последний год?

Автозаправки/postgres@localhost

Query Editor

```
1 select
2     sum(financial_transaction."количество топлива"*companys_fuel."цена за единицу объема") as "сумма",
3     gas_station.id,
4     gas_station."адрес",
5     company."название" as "фирма-поставщик"
6
7 from "Финансовая транзакция" financial_transaction
8
9 join "Автозаправка" gas_station
10 on financial_transaction."id автозаправки" = gas_station.id
11
12 join "Фирма-поставщик" company
13 on gas_station."id поставщика" = company.id
14
15 join "Топливо поставщика" companys_fuel
16 on company.id = companys_fuel."id поставщика"
17
18 join "Топливо" fuel
19 on fuel.id = financial_transaction."id топлива"
20
21 where financial_transaction."дата и время" between '2022-01-01 00:00:00' and '2023-01-01 00:00:00'
22 group by gas_station.id, "фирма-поставщик"
23
24 order by "сумма" desc
25 limit 1
26
```

План выполнения

План выполнения

Graphical Analysis Статистика

```
graph LR
    FT[Финансовая транзакция] -- "Nested Loop Inner Join" --> GS[Автозаправка, pkey]
    GS -- "Nested Loop Inner Join" --> C[Фирма-поставщик, id_key]
    C -- "Nested Loop Inner Join" --> CF[Топливо поставщика, id_key]
    CF -- "Nested Loop Inner Join" --> F[Топливо, id_key]
    F --> AG[Aggregate]
    AG --> S[Sort]
    S --> L[Limit]
```

Результат

	сумма real	id integer	адрес text	фирма-поставщик text
1	4988.4497	15	ул. Карбышева, 5	BP

8. Создать представление, содержащее сведения обо всех АЗС и всех видах топлива, которые они продают.

Query Editor

```
1 create view all_gas_stations as
2
3 select
4     gas_station.id as "номер автозаправки",
5     company."название" as "фирма-поставщик",
6     companys_fuel."id топлива" as "номер топлива"
7
8 from "Автозаправка" gas_station
9
10 join "Фирма-поставщик" company
11 on gas_station."id поставщика" = company.id
12
13 join "Топливо поставщика" companys_fuel
14 on company.id = companys_fuel."id поставщика"
15
```

Сообщения

CREATE VIEW

Запрос завершён успешно, время выполнения: 64 msec.

Результат

	номер автозаправки integer	фирма-поставщик text	номер топлива integer
1	1	Shell	1
2	2	Shell	1
3	3	BP	4
4	3	BP	2
5	3	BP	3
6	5	Ленэнерго	5
7	6	Shell	1
8	7	Shell	1
9	8	Shell	1
10	12	Ленэнерго	5
11	13	Ленэнерго	5
12	14	Ленэнерго	5

9. Самая прибыльная АЗС за истекший месяц для каждого производителя.

Query Editor

```
1 create view best_gas_station as
2 with pre_result as (
3     select
4         sum(financial_transaction."количество топлива"*companys_fuel."цена за единицу объема") as "сумма",
5         gas_station.id,
6         gas_station."адрес",
7         company."название" as "фирма-поставщик",
8         row_number() over(
9             partition by company."название"
10            order by gas_station.id desc, sum(financial_transaction."количество топлива"*companys_fuel."цена за единицу объема") desc
11        ) as row_number
12
13    from "Финансовая транзакция" financial_transaction
14
15    join "Автозаправка" gas_station
16    on financial_transaction."id автозаправки" = gas_station.id
17
18    join "Фирма-поставщик" company
19    on gas_station."id поставщика" = company.id
20
21    join "Топливо поставщика" companys_fuel
22    on company.id = companys_fuel."id поставщика"
23
24    join "Топливо" fuel
25    on fuel.id = financial_transaction."id топлива"
26
27    where financial_transaction."дата и время" between '2022-01-01 00:00:00' and '2023-01-01 00:00:00'
28    group by gas_station.id, "фирма-поставщик"
29
30    order by "сумма" desc
31 )
32
33 select *
34 from pre_result
35 where row_number = 1
```

План выполнения

Use
Explain/Explain

CREATE VIEW

Запрос завершён успешно, время выполнения: 36 msec.

Результат

	сумма real	id integer	адрес text	фирма-поставщик text	row_number bigint
1	4988.4497	15	ул. Карбышева, 5	BP	1
2	4890	13	наб. Обводного канала, 24, Санкт-Петербург	Ленэнерго	1
3	1532.8	8	просп. Маршала Блюхера, 2	Shell	1

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

1. Удалить неиспользуемые карты

```
1 delete
2 from "Карта" card
3 where card."номер" not in (
4     select "номер карты"
5     from "Счёт"
6 )
```

public.Карта/Автозаправки/postg

Результат

	номер [PK] integer	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	
18	18	
19	19	
20	20	

Результат

	номер [PK] integer	
1	1	
2	2	
3	5	
4	10	
5	11	
6	12	
7	15	

2. Добавить пользователю 100 рублей на счет, если id_фирмы = 1

Автозаправки/postgres@localhost ▾

Query Editor

```

1 update
2 "Счёт"
3 SET "остаток средств"="остаток средств"+100
4 where "Счёт".id_фирмы = 1

```

	номер [PK] integer	id клиента integer	номер карты integer	остаток средств real	id_фирмы integer
1	1	2	15	1432.22	4
2	2542	1	12	533.01	4
3	22421	4	10	436.22	4
4	22422	5	11	1000	2
5	103020	2	2	10402	1
6	123123	3	5	943	4
7	10101010	1	1	1000	1

Результат

	номер [PK] integer	id клиента integer	номер карты integer	остаток средств real	id_фирмы integer
1	1	2	15	1432.22	4
2	2542	1	12	533.01	4
3	22421	4	10	436.22	4
4	22422	5	11	1000	2
5	103020	2	2	10502	1
6	123123	3	5	943	4
7	10101010	1	1	1100	1

3. Добавить тестовую фирму

```
1 insert into "Фирма-поставщик"
2 values (
3     1 + (
4         select id
5         from "Фирма-поставщик"
6         order by id desc
7         limit 1
8     ),
9     'Mock',
10    'Mock',
11    'Mock'
12 )
13
14
```

Результат

	<div>id</div> <div>[PK] integer</div>	<div>юридический адрес</div> <div>text</div>	<div>название</div> <div>text</div>	<div>телефон</div> <div>text</div>
1	1	125196, Москва Город, ул. Лесная, д. 9	Shell	+77172555444
2	2	115230, Г.Москва, вн.тер.г. Муниципальный Округ Нагорный, проезд Электролитный, д. 3, стр. 32	BP	88002001070
3	3	196158, Санкт-петербург Город, ш. Пулковское, д.32-а	Neste	+78127030610
4	4	197227, Г.Санкт-Петербург, вн.тер.г. Муниципальный Округ Озеро Долгое, ул Гаккелевская, д. 21, литера А	Ленэнерго	88002200220

Результат

	<div>id</div> <div>[PK] integer</div>	<div>юридический адрес</div> <div>text</div>	<div>название</div> <div>text</div>	<div>телефон</div> <div>text</div>
1	1	125196, Москва Город, ул. Лесная, д. 9	Shell	+77172555444
2	2	115230, Г.Москва, вн.тер.г. Муниципальный Округ Нагорный, проезд Электролитный, д. 3, стр. 32	BP	88002001070
3	3	196158, Санкт-петербург Город, ш. Пулковское, д.32-а	Neste	+78127030610
4	4	197227, Г.Санкт-Петербург, вн.тер.г. Муниципальный Округ Озеро Долгое, ул Гаккелевская, д. 21, литера А	Ленэнерго	88002200220
5	5	Mock	Mock	Mock

3. Изучить графическое представление запросов и просмотреть историю запросов

История запросов

Show queries generated internally by pgAdmin? ☒ Да

Today - 16.06.2022

▶ insert into "Фирма-поставщик" values (1 + (select id from "Фир... 15:08:34

▶ insert into "Фирма-поставщик" values (1 + (select id from "Фир... 15:07:59

▶ insert into "Фирма-поставщик" values (1 + (select id from "Фир... 15:07:42

▶ insert into "Фирма-поставщик" values (1 + (select id from "Фир... 15:07:27

▶ insert into "Фирма-поставщик" values (1 + 10, -- select id --... 15:05:53

▶ insert into "Фирма-поставщик" values (1 + 10, -- select id --... 15:05:38

▶ insert into "Фирма-поставщик" values (1 + 10 -- select id -- ... 15:05:23

▶ insert into "Фирма-поставщик" values (1 + select id from "Фир... 15:04:54

▶ insert into "Фирма-поставщик" company values (1 + select id f... 15:04:45

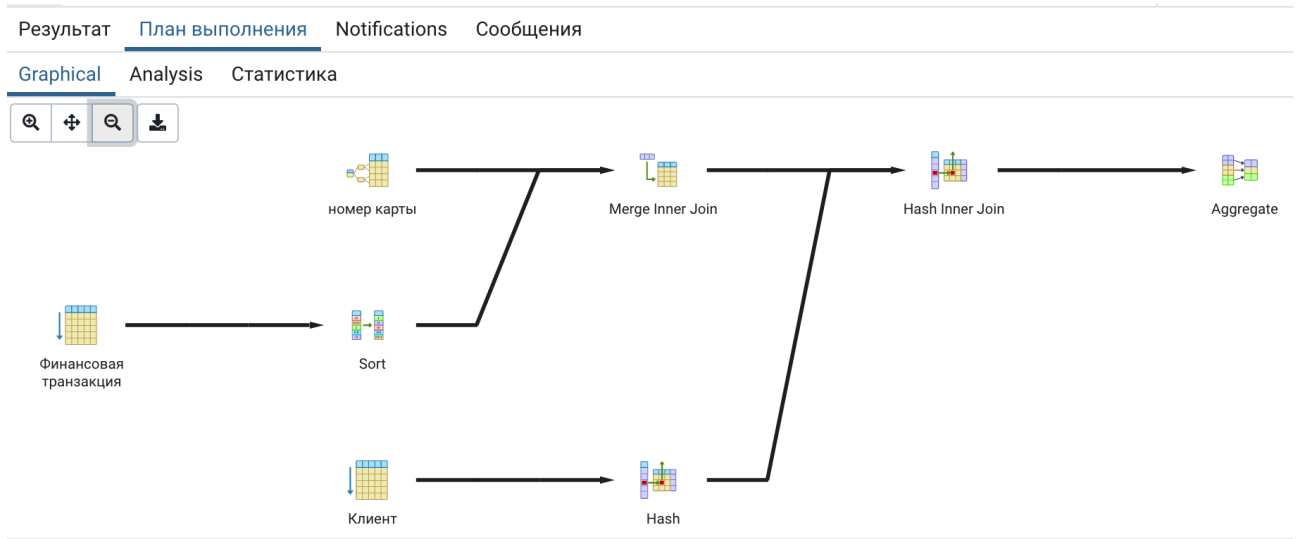
▶ -- create view best_gas_station as with pre_result as (select... 14:07:44

▶ create view best_gas_station as with pre_result as (select su... 14:07:36

16.06.2022 15:08:34 1 31 msec
Дата Rows Affected Продолжительность

Copy Copy to Query Editor

```
insert into "Фирма-поставщик"
values (
  1 + (
    select id
    from "Фирма-поставщик"
    order by id desc
    limit 1
  ),
  'Mock',
  'Mock',
  'Mock'
)
```



4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами.

Запрос: Сколько раз заправлял автомобиль каждый из клиентов за заданный период.

Запрос без индекса

Запрос с простым индексом

```
1 select "номер карты" from "Финансовая транзакция"
2
```

✓ Запрос выполнен успешно. Общее время выполнения: 39 msec. обработано строк: 9.

CREATE INDEX "номер карты" ON "Финансовая транзакция" ("номер карты")

✓ Запрос выполнен успешно. Общее время выполнения: 40 msec. обработано строк: 9.

Запрос с составным индексом

create index number_id_idx on "Финансовая транзакция"("номер карты", "id топлива")

Запрос завершён успешно, время выполнения: 45 msec.

Вывод:

В ходе выполнения работы удалось овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.