

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №4 «Запросы на выборку и модификацию данных.
Представления. Работа с индексами»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Бунос М.В.

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

1. Запросы к базе данных	3
2. Представления.....	6
3. Кастом запросы.....	7
4. Индексы.....	8
Вывод.....	11

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.

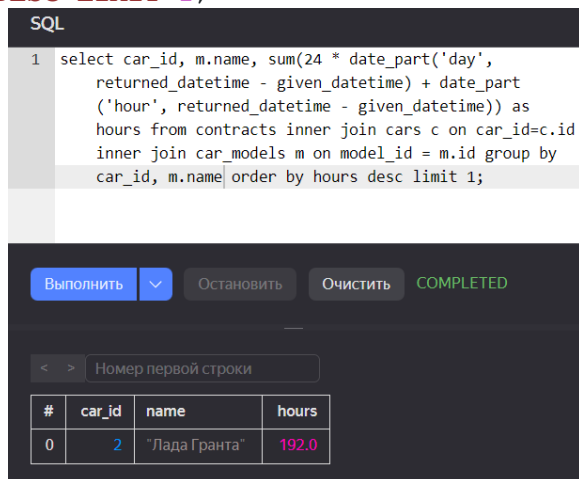
Практическое задание:

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 базе данных

- Какой автомобиль находился в прокате максимальное количество часов?

```
SELECT car_id, m.name, SUM(24 * date_part('day', returned_datetime -  
given_datetime) + date_part('hour', returned_datetime - given_datetime))  
AS hours FROM contracts INNER JOIN cars c ON car_id=c.id INNER JOIN  
car_models m ON model_id = m.id GROUP BY car_id, m.name ORDER BY hours  
DESC LIMIT 1;
```



The screenshot shows a SQL query execution interface. At the top, the SQL query is displayed in a text area. Below the query, there are buttons for 'Выполнить' (Execute), 'Остановить' (Stop), and 'Очистить' (Clear), along with a 'COMPLETED' status indicator. Below the buttons, there is a table with the results of the query. The table has four columns: '#', 'car_id', 'name', and 'hours'. The first row shows the result for the car with the highest rental hours.

#	car_id	name	hours
0	2	"Лада Гранта"	192.0

- Автомобили какой марки чаще всего брались в прокат?

```
SELECT car_id, m.name, COUNT(car_id) AS cnt FROM contracts INNER JOIN cars  
c ON car_id=c.id INNER JOIN car_models m ON model_id = m.id GROUP BY  
car_id, m.name ORDER BY cnt DESC LIMIT 1;
```

SQL

```
1 select car_id, m.name, count(car_id) as cnt from
   contracts inner join cars c on car_id=c.id inner join
   car_models m on model_id = m.id group by car_id, m
   .name order by cnt desc limit 1;
```

Выполнить Остановить Очистить COMPLETED

< > Номер первой строки

#	car_id	name	cnt
0	1	"Лада Веста"	1

- Определить убытки от простоя автомобилей за текущий день.

```
WITH current_prices AS (SELECT price, model_id FROM rent_prices WHERE
DATE_TRUNC('day', CURRENT_TIMESTAMP) < end_datetime)
SELECT car_id, m.name, p.price*(24 - SUM(24 * date_part('day',
returned_datetime - GREATEST(DATE_TRUNC('day', CURRENT_TIMESTAMP),
given_datetime)) + date_part('hour', returned_datetime -
GREATEST(given_datetime, DATE_TRUNC('day', CURRENT_TIMESTAMP))))) AS
end_price FROM contracts INNER JOIN cars c ON car_id=c.id INNER JOIN
car_models m ON c.model_id = m.id INNER JOIN current_prices p ON
c.model_id = p.model_id WHERE returned_datetime > DATE_TRUNC('day',
CURRENT_TIMESTAMP) GROUP BY car_id, m.name, p.price;
```

SQL

```
returned_datetime - GREATEST(given_datetime,
DATE_TRUNC('day', CURRENT_TIMESTAMP))))) AS
end_price FROM contracts INNER JOIN cars c ON
car_id=c.id INNER JOIN car_models m ON c.model_id
= m.id INNER JOIN current_prices p ON c.model_id =
p.model_id where returned_datetime > DATE_TRUNC
('day', CURRENT_TIMESTAMP) GROUP BY car_id, m.name
, p.price;
```

Выполнить Остановить Очистить COMPLETED

< > Номер первой строки

#	car_id	name	end_price
0	2	"Лада Гранта"	132000.0
1	1	"Лада Веста"	160000.0
2	3	"Kia Rio"	117000.0

- Вывести данные автомобиля, имеющего максимальный пробег.

```
SELECT * FROM cars ORDER BY mileage DESC LIMIT 1;
```

SQL

```
1 select * from cars order by mileage desc limit 1;
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	id	registration_number	mileage	issue_year	engine_number
0	3	"P333PP77"	15000	2020	"EN67890"

- Какой автомобиль суммарно находился в прокате дольше всех.

SELECT car_id, m.name, returned_datetime - given_datetime **AS** time_in_rent
FROM contracts **INNER JOIN** cars c **ON** car_id=c.id **INNER JOIN** car_models m **ON**
model_id = m.id **ORDER BY** time in rent **DESC LIMIT 1**;

SQL

```
1 SELECT car_id, m.name, returned_datetime - given_datetime  
as time_in_rent FROM contracts INNER JOIN cars c ON  
car_id=c.id INNER JOIN car_models m ON model_id = m  
.id ORDER BY time_in_rent DESC LIMIT 1;
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	car_id	name	time_in_rent
0	1	"Лада Веста"	"0 years 0 mons 182 days 22 hours 0 mins 0.00 secs"

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

SELECT cm.name, **COUNT**(c.id) **FROM** cars c **INNER JOIN** car_models cm **ON** cm.id
= model_id **GROUP BY** model_id, cm.name;

SQL

```
1 select cm.name, count(c.id) from cars c inner join  
car_models cm on cm.id = model_id group by model_id,  
cm.name;
```

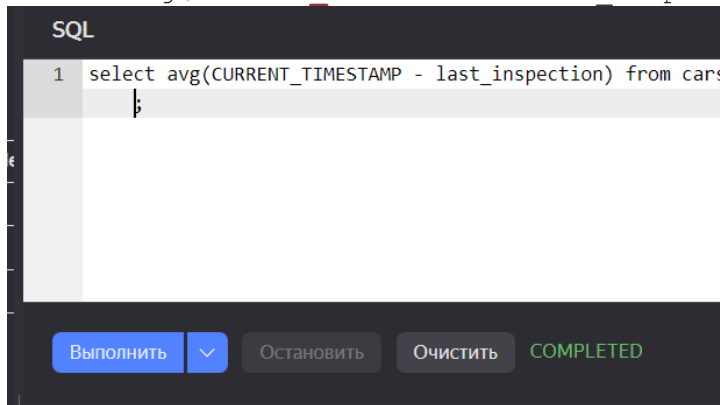
Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	name	count
0	"Kia Rio"	1
1	"Лада Веста"	1
2	"Лада Гранта"	1

- Определить средний “возраст” автомобилей компании.

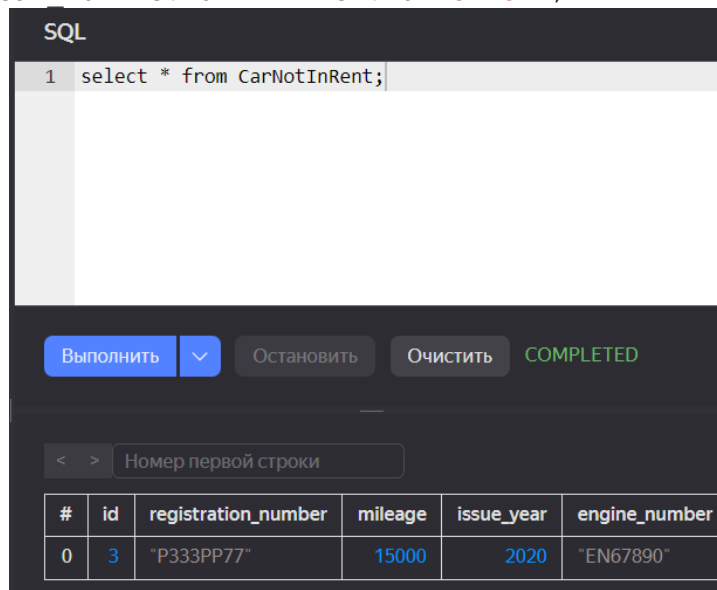
```
SELECT avg(CURRENT_TIMESTAMP - last_inspection) FROM cars;
```



2. Представления

- Какой автомобиль ни разу не был в прокате?

```
CREATE VIEW CarNotInRent AS SELECT c.* FROM cars c LEFT JOIN contracts cn ON cn.car_id = c.id WHERE cn.id IS NULL;
```



- Вывести данные клиентов, не вернувших автомобиль вовремя.

```
CREATE VIEW expiredClients AS SELECT cl.* FROM contracts c INNER JOIN clients cl ON cl.id=c.client_id WHERE returned_datetime < CURRENT_TIMESTAMP AND common_status = 1;
```

SQL

```
1
2 select * from expiredClients;
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	id	mobile_phone	name	discount	adress	passport
0	3	"+79371234569"	"Сергей Сергеев"	0	"Казань, Кремль 3"	"4703"

3. Кастом запросы

- Добавить новую тачку по имени

```
INSERT INTO cars (id, registration_number, mileage, issue_year,
engine_number, body_number, price, last_inspection, special_marks,
return_mark, model_id) VALUES (4, 'X228EP98', 15000, 2020, 'EN66666',
'BN77777', 1300000, '2023-01-01', NULL, 1, (SELECT id FROM car_models
WHERE name='Лада Веста'));
```

SQL

```
1 insert into cars (id, registration_number, mileage,
issue_year, engine_number, body_number, price,
last_inspection, special_marks, return_mark, model_id
) values (4, 'X228EP98', 15000, 2020, 'EN66666',
'BN77777', 1300000, '2023-01-01', null, 1, (select id
from car_models where name='Лада Веста'));
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	result
0	1

- Добавить умничка в имя где работник больше всего засветился в контрактах

```
UPDATE employees SET name = concat(name, ' (умничка)') WHERE id = (SELECT
employee_id FROM contracts GROUP BY employee_id ORDER BY COUNT(id) DESC
LIMIT 1);
```

SQL

```
1 update employees set name = concat(name, ' (умничка)')
   where id = (select employee_id from contracts group
               by employee_id order by count(id) desc limit 1);
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	result
0	1

- Уволить сотрудника, который вообще не принимал участие в контрактах

DELETE FROM employees **WHERE** id **NOT IN** (**SELECT** employee_id **FROM** contracts **GROUP BY** employee_id **ORDER BY** COUNT(id));

SQL

```
1 delete from employees where id not in (select employee_id
    from contracts group by employee_id order by count(id
    ));
```

Выполнить ▼ Остановить Очистить COMPLETED

< > Номер первой строки

#	result
0	1

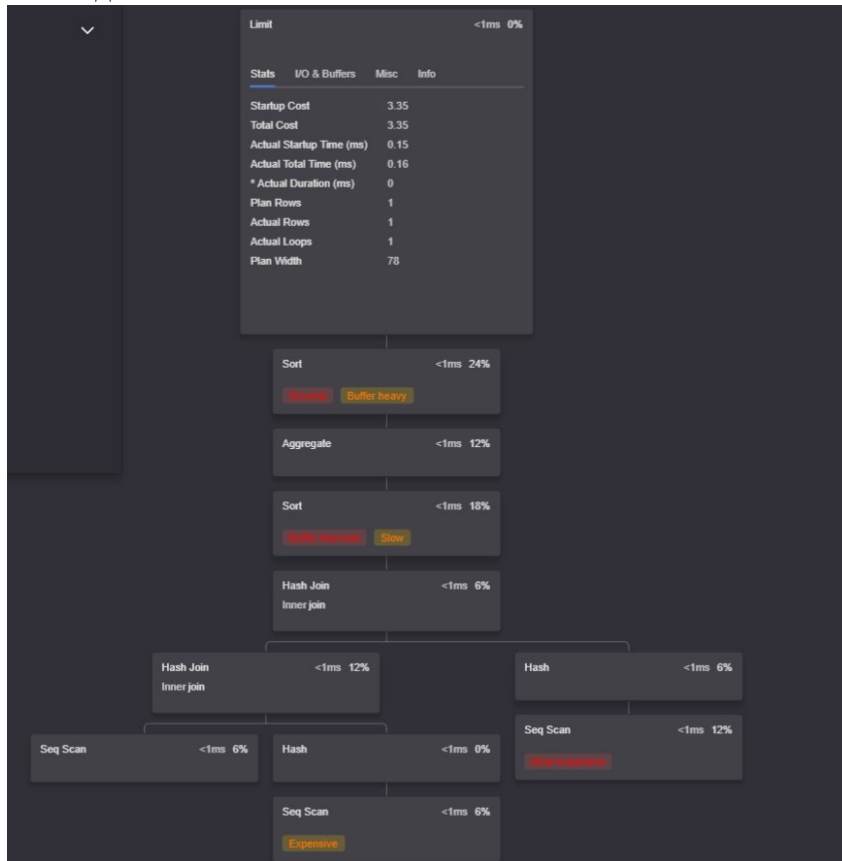
4. Индексы

Создадим индекс на даты в контракте и на айди машины:

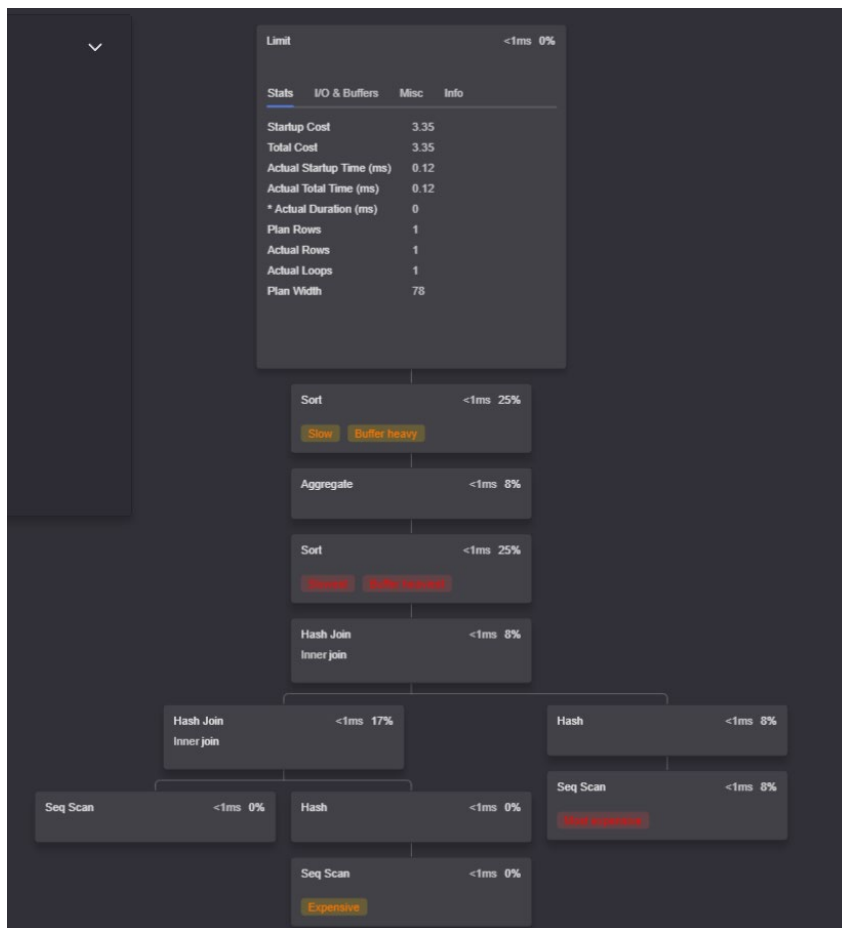
```
CREATE INDEX idx_car_id ON contracts (car_id);  
CREATE INDEX idx_dates ON contracts (returned_datetime , given_datetime);
```

```
SELECT car_id, m.name, SUM(24 * date_part('day', returned_datetime -  
given_datetime) + date_part('hour', returned_datetime - given_datetime))  
AS hours FROM contracts INNER JOIN cars c ON car_id=c.id INNER JOIN  
car_models m ON model_id = m.id GROUP BY car_id, m.name ORDER BY hours  
DESC LIMIT 1;
```

Без индекса:

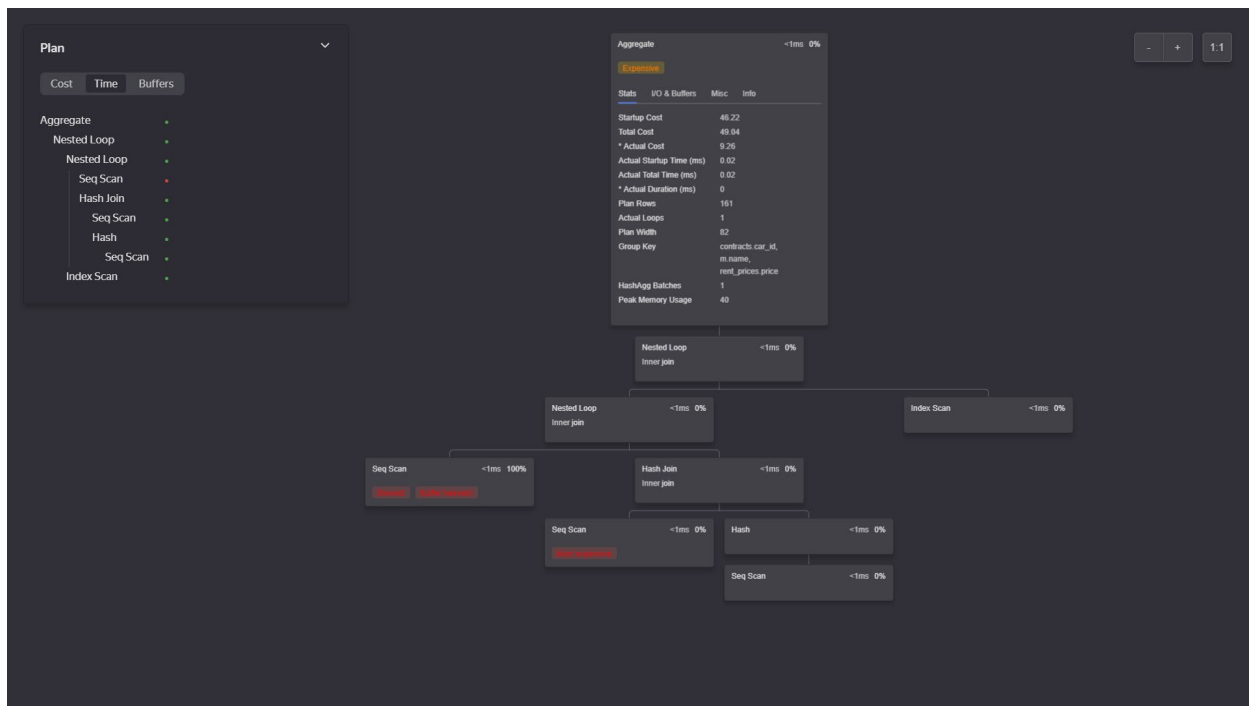


С индексом:

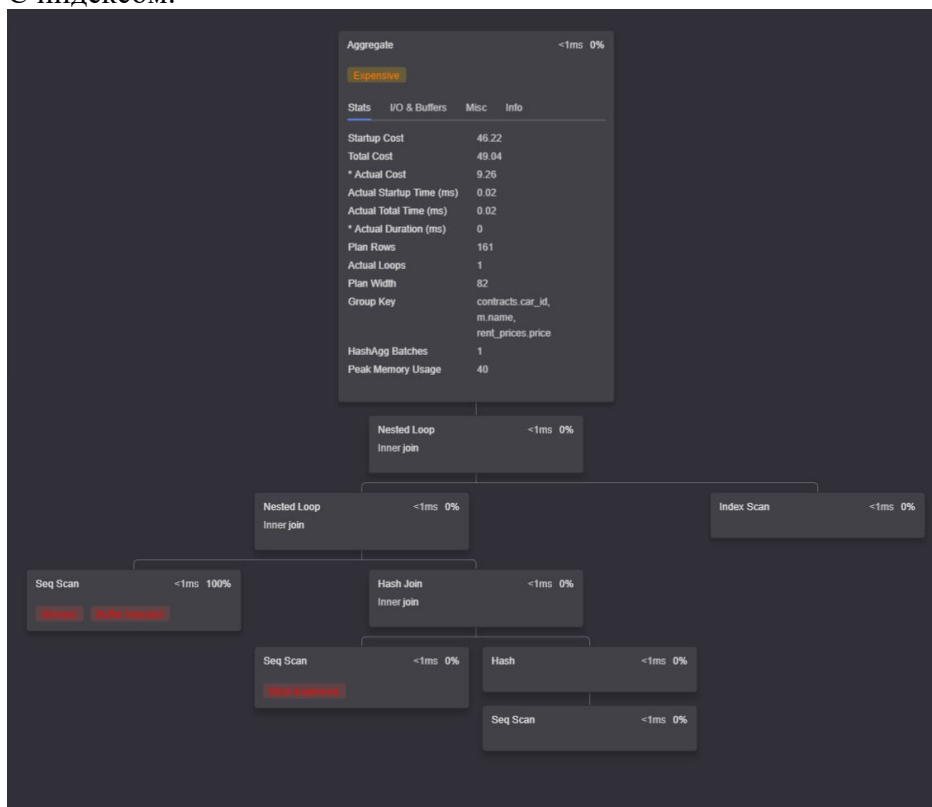


```
WITH current_prices AS (SELECT price, model_id FROM rent_prices WHERE
DATE_TRUNC('day', CURRENT_TIMESTAMP) < end_datetime)
SELECT car_id, m.name, p.price*(24 - SUM(24 * date_part('day',
returned_datetime - GREATEST(DATE_TRUNC('day', CURRENT_TIMESTAMP),
given_datetime)) + date_part('hour', returned_datetime -
GREATEST(given_datetime, DATE_TRUNC('day', CURRENT_TIMESTAMP))))) AS
end_price FROM contracts INNER JOIN cars c ON car_id=c.id INNER JOIN
car_models m ON c.model_id = m.id INNER JOIN current_prices p ON
c.model_id = p.model_id WHERE returned_datetime > DATE_TRUNC('day',
CURRENT_TIMESTAMP) GROUP BY car_id, m.name, p.price;
```

Без индекса:



С индексом:



Вывод

В ходе лабораторной работы я освоил практические навыки по выполнению select, insert, delete и update запросов. Также, научился делать представления и индексы. В ходе анализа в первом запросе индексы дали буст в ~40%, во втором запросе практически не дали.