

**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: Кузьмина Марина Леонидовна

Факультет: ИКТ

Группа: K32421

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## СОДЕРЖАНИЕ

1 Цель работы и практическое задание.....	3
2 Выполнение	
2.1 Наименование БД.....	3
2.2 Схема логической модели базы данных, сгенерированная в Generate ERD.....	4
2.3 Выполнение запросов.....	5
2.4 Выполнение представлений.....	9
2.4 Выполнение запросов на модификацию данных.....	9
2.4.1 INSERT.....	9
2.4.2 UPDATE.....	12
2.4.3 DELETE.....	13
2.5 Создание индексов.....	13
3 Выводы.....	17

## **1 Цель работы и практическое задание**

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

Практическое задание:

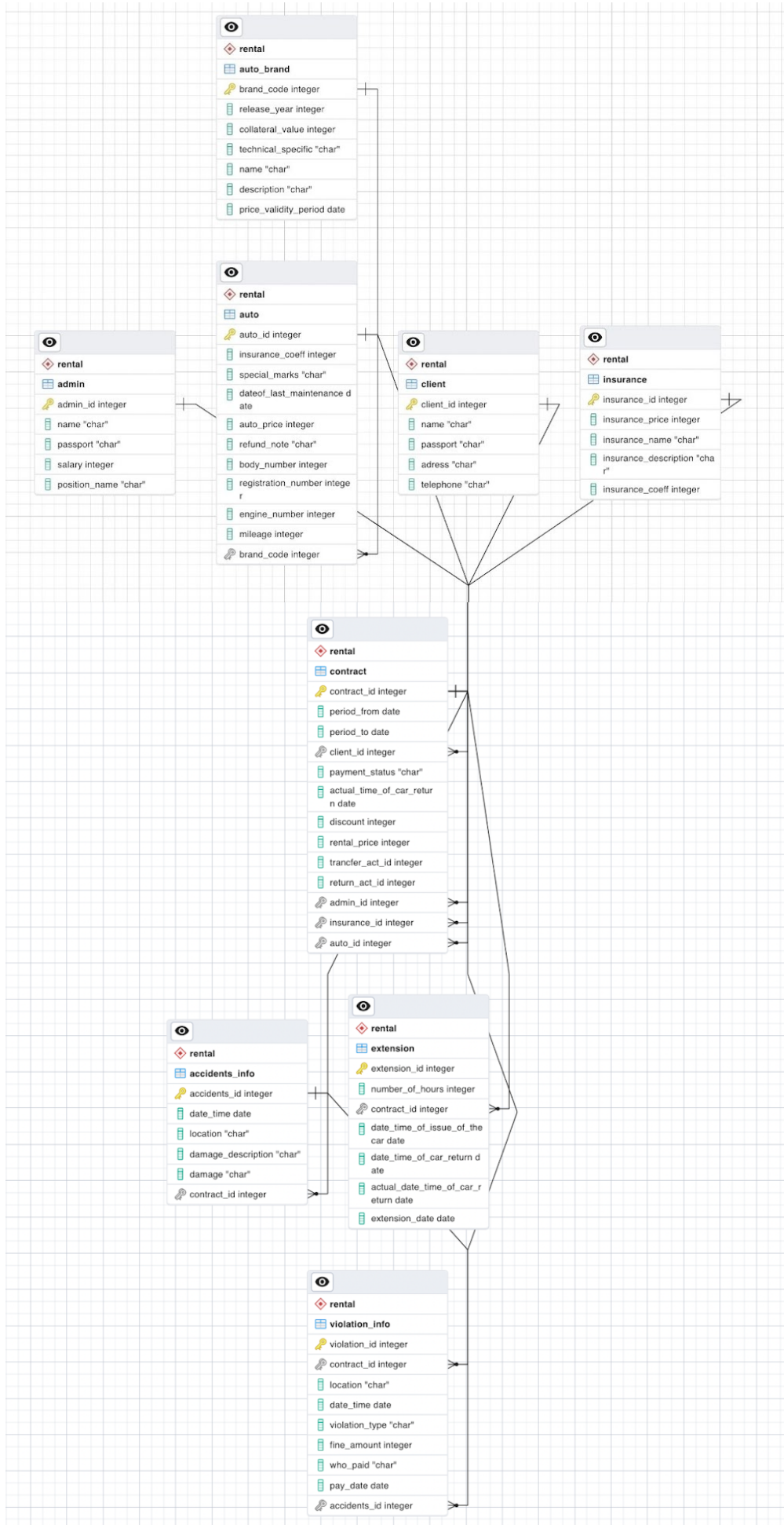
1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
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.

## **2 Выполнение работы**

### **2.1 Наименование БД**

БД «Прокат автомобилей»

## 2.2 Схема логической модели базы данных, сгенерированная в Generate ERD



## 2.3 Выполнение запросов

- Какой автомобиль находился в прокате максимальное количество часов?

```
SELECT auto.registration_number, SUM(extension.number_of_hours)
AS total_hours
FROM rental.auto
JOIN rental.contract ON auto.auto_id = contract.auto_id
JOIN rental.extension ON contract.contract_id = extension.contract_id
GROUP BY auto.registration_number
ORDER BY total_hours DESC
LIMIT 1;
```

```
SELECT auto.registration_number, SUM(extension.number_of_hours) AS total_hours
FROM rental.auto
JOIN rental.contract ON auto.auto_id = contract.auto_id
JOIN rental.extension ON contract.contract_id = extension.contract_id
GROUP BY auto.registration_number
ORDER BY total_hours DESC
LIMIT 1;
```

Data Output	Messages	Notifications
	<b>registration_number</b> integer	<b>total_hours</b> bigint
1	1	20

- Автомобили какой марки чаще всего брались в прокат?

```
SELECT ab.name AS brand_name, COUNT(*) AS rental_count
FROM rental.contract c
JOIN rental.auto a ON c.auto_id = a.auto_id
JOIN rental.auto_brand ab ON a.brand_code = ab.brand_code
GROUP BY ab.name
ORDER BY rental_count DESC;
```

```

SELECT ab.name AS brand_name, COUNT(*) AS rental_count
FROM rental.contract c
JOIN rental.auto a ON c.auto_id = a.auto_id
JOIN rental.auto_brand ab ON a.brand_code = ab.brand_code
GROUP BY ab.name
ORDER BY rental_count DESC

```

Data Output			Messages	Notifications
	<b>brand_name</b> character varying (20)	<b>rental_count</b> bigint		
1	Volvo	3		

- Определить убытки от простоя автомобилей за вчерашний день.

```

SELECT SUM(auto_price) AS losses

```

```

FROM rental.auto

```

```

WHERE auto_id IN (

```

```

    SELECT auto_id

```

```

    FROM rental.contract AS c

```

```

    WHERE c.period_from <= NOW() - INTERVAL '1 DAY' AND
(c.actual_time_of_car_return IS NULL OR c.actual_time_of_car_return
>= NOW())

```

```

);

```

```

SELECT SUM(auto_price) AS losses
FROM rental.auto
WHERE auto_id IN (
    SELECT auto_id
    FROM rental.contract AS c
    WHERE c.period_from <= NOW() - INTERVAL '1 DAY' AND (c.actual_time_of_car_return IS NULL OR c.actual_time_of_car_return >= NOW())
);

```

Data Output			Messages	Notifications
	<b>losses</b> bigint			
1	[null]			

- Вывести данные автомобиля, имеющего максимальный пробег.

```
SELECT a.registration_number, b.name, a.mileage
```

```
FROM rental.auto AS a
```

```
JOIN rental.auto_brand AS b ON a.brand_code = b.brand_code
```

```
WHERE a.mileage = (SELECT MAX(mileage) FROM rental.auto);
```

```
SELECT a.registration_number, b.name, a.mileage
FROM rental.auto AS a
JOIN rental.auto_brand AS b ON a.brand_code = b.brand_code
WHERE a.mileage = (SELECT MAX(mileage) FROM rental.auto);
```

Data Output Messages Notifications			
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	registration_number integer	name character varying (20)	mileage integer
1	2	Porsche	4839

- Какой автомобиль суммарно находился в прокате дольше всех.

```
SELECT
```

```
    a.registration_number,
```

```
    SUM(EXTRACT(HOUR FROM AGE(c.actual_time_of_car_return,
c.period_from))) AS total_hours_rented
```

```
FROM
```

```
    rental.contract AS c
```

```
    JOIN rental.auto AS a ON c.auto_id = a.auto_id
```

```
GROUP BY
```

```
    a.registration_number
```

```
ORDER BY
```

```
    total_hours_rented DESC
```

```
LIMIT 1;
```

```

SELECT
  a.registration_number,
  SUM(EXTRACT(HOUR FROM AGE(c.actual_time_of_car_return, c.period_from))) AS total_hours_rented
FROM
  rental.contract AS c
  JOIN rental.auto AS a ON c.auto_id = a.auto_id
GROUP BY
  a.registration_number
ORDER BY
  total_hours_rented DESC
LIMIT 1;

```

Data Output	Messages	Notifications						
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> <table> <thead> <tr> <th></th><th>registration_number integer</th><th>total_hours_rented numeric</th></tr> </thead> <tbody> <tr> <td>1</td><td>1</td><td>24</td></tr> </tbody> </table>		registration_number integer	total_hours_rented numeric	1	1	24		
	registration_number integer	total_hours_rented numeric						
1	1	24						

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

```

SELECT ab.name AS brand_name, a.brand_code AS model_name,
COUNT(*) AS car_count

```

```

FROM rental.auto AS a

```

```

JOIN rental.auto_brand AS ab

```

```

ON a.brand_code = ab.brand_code

```

```

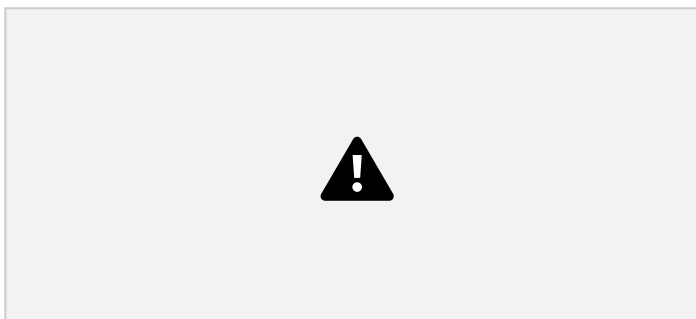
GROUP BY ab.name, a.brand_code;

```

```

SELECT ab.name AS brand_name, a.brand_code AS model_name, COUNT(*) AS car_count
FROM rental.auto AS a
JOIN rental.auto_brand AS ab
ON a.brand_code = ab.brand_code
GROUP BY ab.name, a.brand_code;

```





- Определить средний “возраст” автомобилей компании.

```
SELECT AVG(EXTRACT(YEAR FROM NOW()) - release_year) AS
average_age
```

```
FROM rental.auto_brand;
```

```
SELECT AVG(EXTRACT(YEAR FROM NOW()) - release_year) AS average_age
FROM rental.auto_brand;
```

Data Output		Messages
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> </div>		
	average_age numeric	🔒
1	13.6000000000000000	

## 2.4 Выполнение представлений

- Какой автомобиль ни разу не был в прокате?

```
CREATE VIEW rental.never_rented_cars AS
```

```
SELECT *
```

```
FROM rental.auto a
```

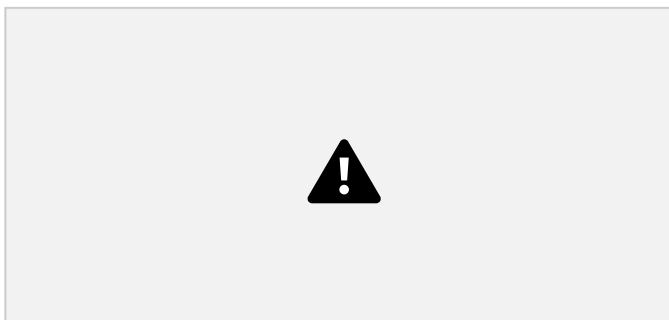
```
WHERE NOT EXISTS (
```

```
  SELECT 1
```

```
  FROM rental.contract c
```

```
  WHERE c.auto_id = a.auto_id
```

```
);
```





- Вывести данные клиентов, не вернувших автомобиль вовремя.

```
CREATE VIEW rental.late_returns AS
```

```
SELECT c.client_id, c.name, c.passport, c.adress, c.telephone
```

```
FROM rental.client c
```

```
JOIN rental.contract co ON c.client_id = co.client_id
```

```
WHERE co.actual_time_of_car_return > co.period_to;
```

```
CREATE VIEW rental.late_returns AS
SELECT c.client_id, c.name, c.passport, c.adress, c.telephone
FROM rental.client c
JOIN rental.contract co ON c.client_id = co.client_id
WHERE co.actual_time_of_car_return > co.period_to;
```

Data Output   **Messages**   Notifications

CREATE VIEW

Query returned successfully in 60 msec.

Data Output		Messages		Notifications					
<div><div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div></div></div>									
	client_id integer		name "char" (1)		passport "char" (1)		adress "char" (1)		telephone "char" (1)

## 2.4 Выполнение запросов на модификацию данных

### 2.4.1 INSERT

Добавить в таблицу “contract” новую строку, где id админа = 1113, а клиента = 11111

```
INSERT INTO rental.contract VALUES(2, '2021-05-20', '2021-05-25',  
(SELECT client_id FROM rental.client WHERE client_id = 11111), 'Payed',  
'2021-05-25',1, 20.000, 2, 2, (SELECT admin_id FROM rental.admin WHERE  
admin_id = 1113), 1, 2)
```

```
INSERT INTO rental.contract VALUES(2, '2021-05-20', '2021-05-25',  
(SELECT client_id FROM rental.client WHERE client_id = 11111),  
'Payed', '2021-05-25',1, 20.000, 2, 2,  
(SELECT admin_id FROM rental.admin WHERE admin_id = 1113), 1, 2)
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 167 msec.

До:

	contract_id	period_from	period_to	client_id	payment_status	actual_time_of_car_return	discount	rental_price	transfer_act_id	return_act_id	admin_id
	integer	date	date	integer	"char" (1)	date	integer	integer	integer	integer	integer
1	1	2021-01-01	2021-01-05	11111	Payed	2021-01-05	1	10000	1	1	11111

После:

	contract_id	period_from	period_to	client_id	payment_status	actual_time_of_car_return	discount	rental_price	transfer_act_id	return_act_id	admin_id
	integer	date	date	integer	"char" (1)	date	integer	integer	integer	integer	integer
1	1	2021-01-01	2021-01-05	11111	Payed	2021-01-05	1	10000	1	1	11111
2	2	2021-05-20	2021-05-25	11111	Payed	2021-05-25	1	20000	2	2	11113

## 2.4.2 UPDATE

Обновить фактическое время возврата авто после продления на 2021-11-11, по номеру контракта, по id админа.

```
UPDATE rental.extension SET "actual_date_time_of_car_return" =  
'2021-11-11' WHERE "contract_id" IN (SELECT contract.contract_id FROM  
rental.contract WHERE admin_id = 11111)
```

```
UPDATE rental.extension SET "actual_date_time_of_car_return" = '2021-11-11' WHERE "contract_id" IN  
(SELECT contract.contract_id FROM rental.contract WHERE admin_id = 11111)
```



До:

Data Output Messages Notifications							
	extension_id [PK] integer	number_of_hours integer	contract_id integer	date_time_of_issue_of_the_car date	date_time_of_car_return date	actual_date_time_of_car_return date	extension_date date
1	1	20	1	2021-01-05	2022-11-10	2021-11-10	2021-01-05

После:

Data Output Messages Notifications							
	extension_id [PK] integer	number_of_hours integer	contract_id integer	date_time_of_issue_of_the_car date	date_time_of_car_return date	actual_date_time_of_car_return date	extension_date date
1	1	20	1	2021-01-05	2022-11-10	2021-11-11	2021-01-05

## 2.4.3 DELETE

Удалить из таблицы contract строку с контрактом, который заключал админ с id = 1113.

```
DELETE FROM rental.contract WHERE admin_id IN (SELECT  
admin.admin_id FROM rental.admin WHERE admin_id = 1113)
```

```
DELETE FROM rental.contract WHERE admin_id IN (SELECT admin.admin_id FROM rental.admin WHERE admin_id = 1113)
```

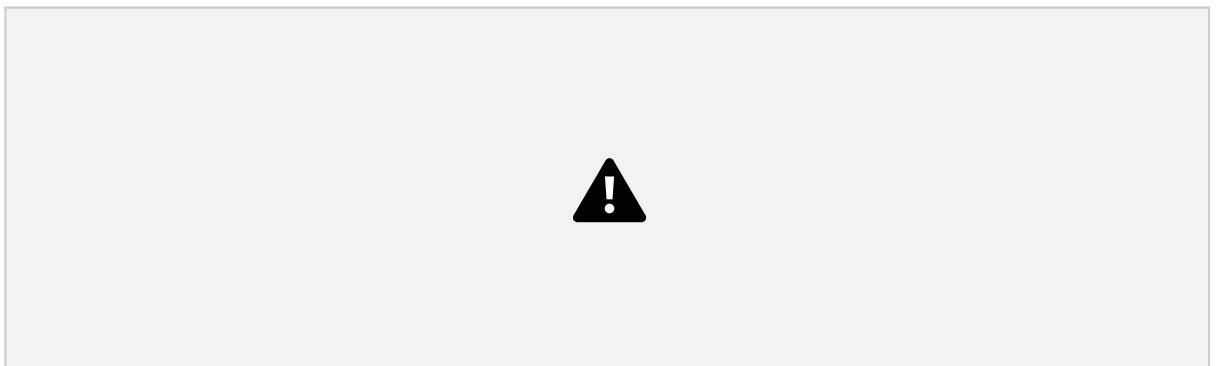


После:

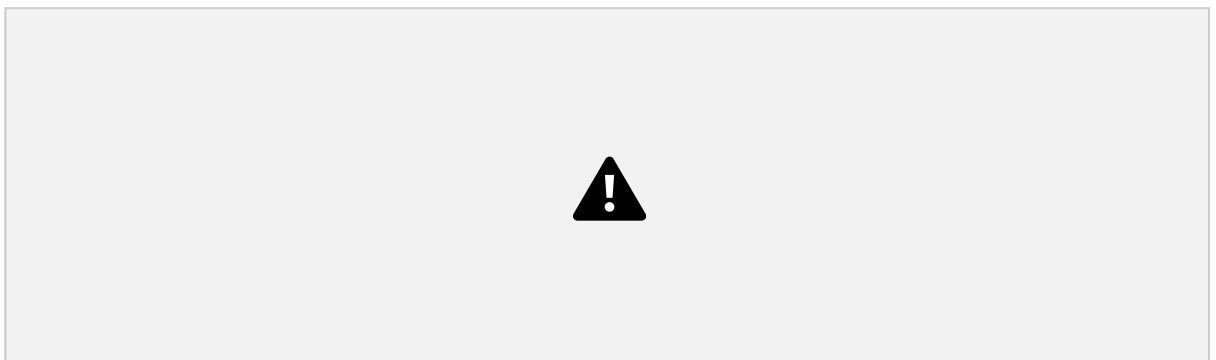
Data Output Messages Notifications											
	tract_id integer	period_from date	period_to date	client_id integer	payment_status "char" (1)	actual_time_of_car_return date	discount integer	rental_price integer	transfer_act_id integer	return_act_id integer	admin_id integer
1	1	2021-01-01	2021-01-05	11111	P	2021-01-05	1	10000	1	1	11111

## 2.5 Создание индексов

### 1) Запрос без индекса



### 2) Создание индексов



Data Output   **Messages**   Notifications

---

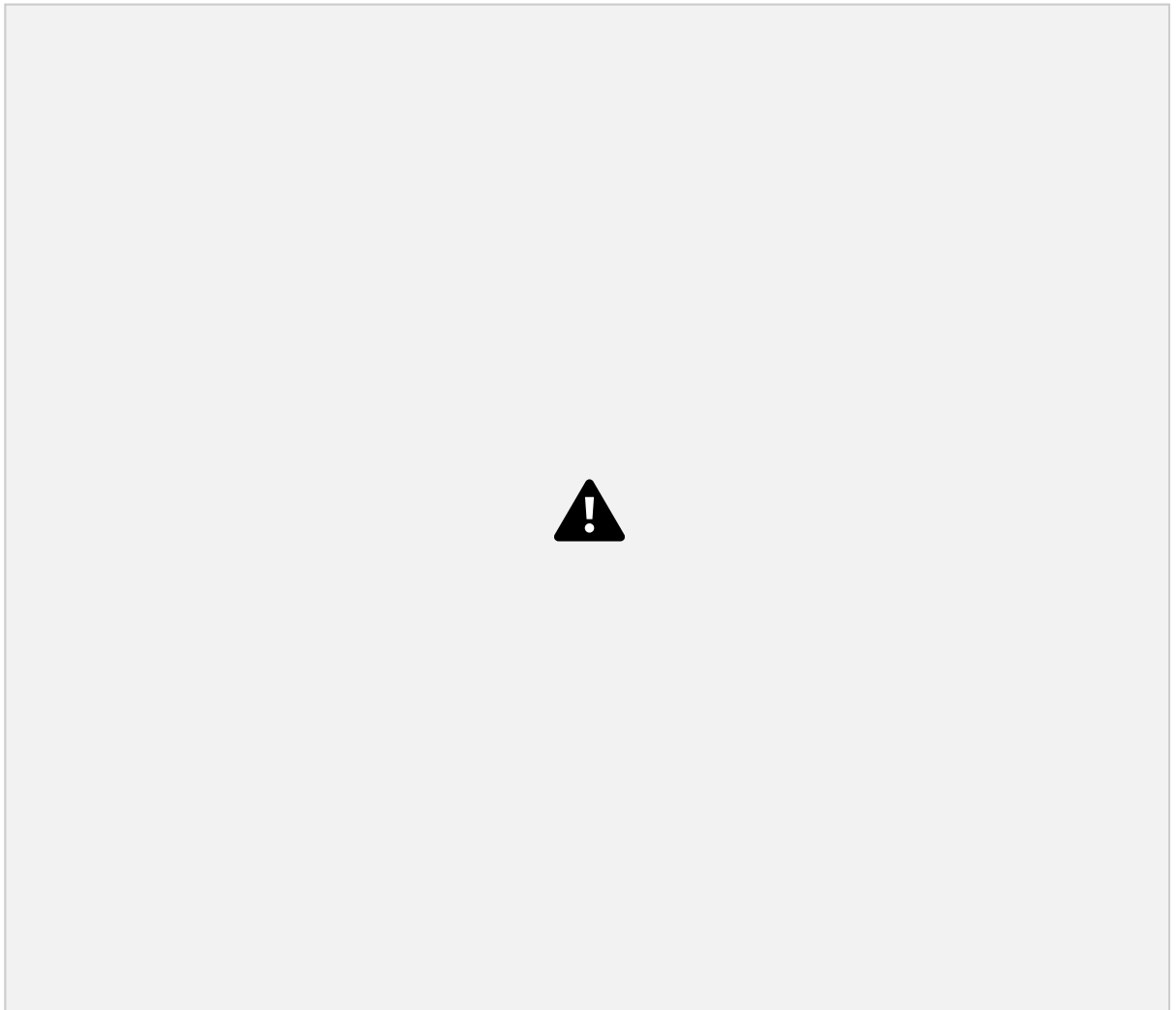
CREATE INDEX

Query returned successfully in 70 msec.

### 3) Запрос с индексом



#### 4) План



#### 4) Разница во времени выполнения

Query complete 00:00:00.075

Query complete 00:00:00.059

## 5) Удаление индексов

```
DROP INDEX idx_contract_auto;  
DROP INDEX idx_contract_client;  
DROP INDEX idx_contract_admin;  
DROP INDEX idx_contract_insurance;  
DROP INDEX idx_accidents_contract;  
DROP INDEX idx_client_passport;  
DROP INDEX idx_auto_brand;  
DROP INDEX idx_violation_contract;  
DROP INDEX idx_extension_contract
```



### **3 Выводы**

В рамках лабораторной работы были созданы запросы и представления на выборку данных к базе данных PostgreSQL согласно индивидуальному заданию, часть 2 и 3. Были созданы 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Были изучены графические представления запросов. Были созданы простой и составной индексы для двух произвольных запросов.

Таким образом, за выполнение данной лабораторной работы удалось познакомиться с представлениями и индексами и успешно их реализовать.

Индексы при больших запросах позволили значительно выиграть время выполнения, план запроса остался тем же.