Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО» (Университет ИТМО)

Факультет инфокомунникационных технологий

Образовательная программа Мобильные и сетевые технологии

Направление подготовки(специальность) 09.03.03 Прикладная

Информатика

## ОТЧЕТ

**по дисциплине «Проектирование и реализация баз данных»** на тему: Запросы на выборку и модификацию данных,представления и индексы в PostgreSQL

Обучающийся: Шляхов Д.О. К32402

Преподаватель: Говорова М. М.

Дата 17.04.2023

**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.

## Индивидуальное задание:

## Вариант 14. БД «Служба заказа такси»

Описание предметной области: Система должна фиксировать все вызовы такси и распределять их между водителями.

Каждому водителю ежедневно начисляется заработная плата в зависимости от количества вызовов и их тарифа (50% от заработанной им суммы). Автомобили могут быть собственностью компании или таксиста.

Заказ принимает дежурный администратор и передает его водителю. В заказе фиксируется тип оплаты – наличными или онлайн.

В системе необходимо хранить график работы водителей.

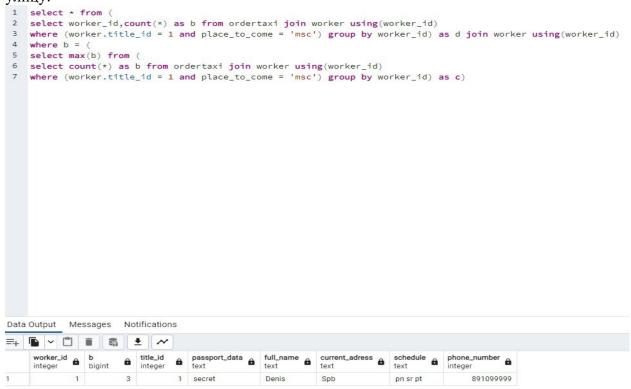
Ежедневно действуют базовые тарифы на тип предоставляемых авто, но в зависимости от времени суток и ситуации на дорогах, цена может корректироваться.

БД должна содержать следующий минимальный набор сведений: Код сотрудника. ФИО сотрудника. Адрес сотрудника. № телефона сотрудника. Паспортные данные сотрудника. Должность сотрудника. Категория сотрудника. Наименование модели и марки автомобиля. Технические характеристики. Стран-производитель. Стоимость. Код тарифа. Наименование тарифа. Цена за километр. Код автомобиля. Госномер автомобиля. Год выпуска. Пробег. Дата последнего ТО. Дата вызова. Время посадки пассажира. Время высадки пассажира. Номер телефона пассажира. Откуда. Куда. Расстояние. Штраф за время ожидания (в минутах). Оплата (онлайн (при заказе) или наличными). Рекламация клиента на вызов.

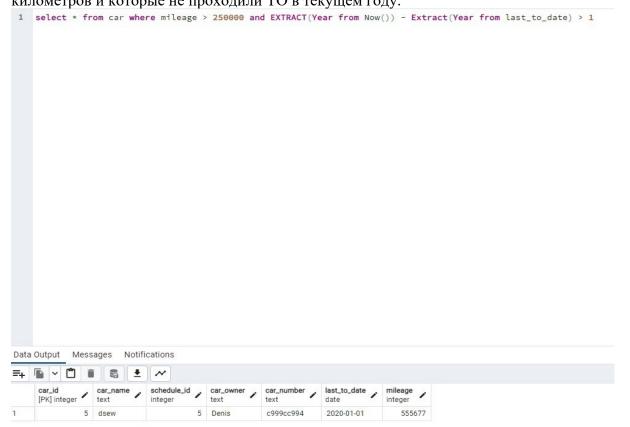
#### Выполнение:

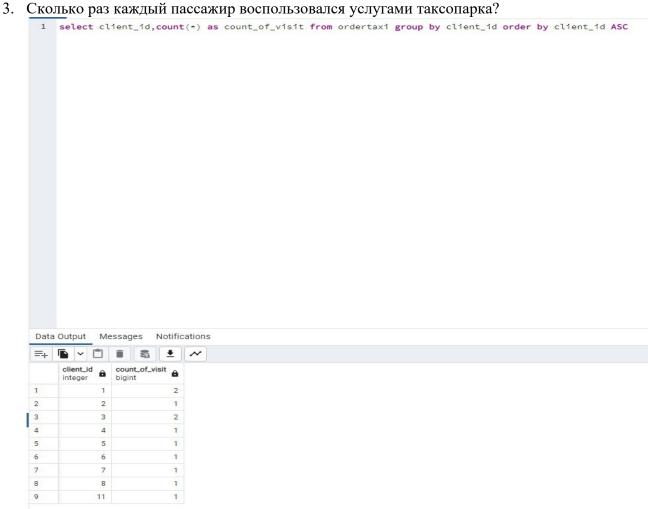
## Запросы

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

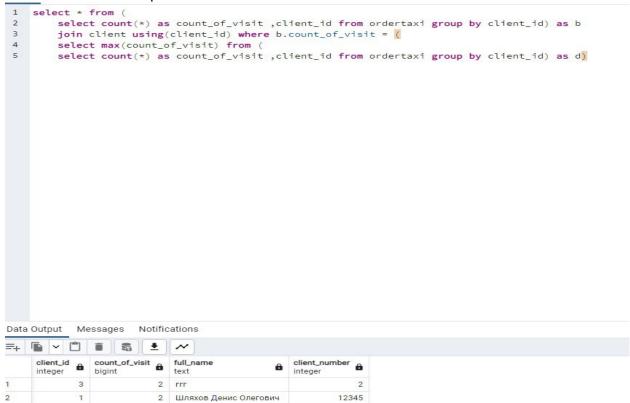


2. Вывести данные об автомобилях, которые имеют пробег более 250 тысяч. километров и которые не проходили ТО в текущем году.





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



# 5. Вывести данные о водителе, который ездит на самом дорогом автомобиле. 1 select worker\_id,title\_id,passport\_data,full\_name,current\_adress,schedule,phone\_number 2 from define\_car join car using(car\_name) join schedule using(schedule\_id) join worker using(worker\_id) join job\_tittle using(title\_id) 3 where car\_price = (select max(car\_price) from define\_car) and title\_id = 1 Data Output Messages Notifications worker\_id a title\_id a passport\_data a text text a full\_name a current\_adress a schedule b phone\_number a 1 asad 3 spb 891099999 asdasdzcd pn sr pt 6. Вывести данные пассажира, который всегда ездит с одним и тем же водителем. 1 select \* from (select client\_id,count(worker\_id) as workers\_count from ordertaxi group by client\_id) as c join client using(client\_id) where workers\_count = 1 Data Output Messages Notifications =+ 6 - 0 1 8 ± ~ client\_id bigint full\_name client\_number a text

11

4

2

1 cccp

1 888

1 спб

1 ппп

1 точка

1 py

1 ленинград

23

3

55

1

66 77 

# Представления

содержащее сведения о незанятых на данный момент водителях; 1 Create or REPLACE VIEW free\_drivers as 2 select \* from worker right join (select DISTINCT worker\_id) from ordertaxi where (NOW() > time\_to\_come)) as d using (worker\_id) Data Output Messages Notifications Query returned successfully in 32 msec. SELECT \* FROM public.free\_drivers 2 Data Output Messages Notifications =+ \* N · ~ 🖺 55 worker\_id full\_name schedule phone\_number current\_adress title id passport\_data text integer integer text integer text text 1 4 4 ded zxxa spb 891099999 pn sr pt 2 6 5 dfg ffg spb pn sr pt 891099999 3 2 2 fff 891099999 asdasd spb pn sr pt 7 4 1 891099999 wer qwe spb pn sr pt 5 3 asad asdasdzcd spb pn sr pt 891099999 1 Denis 891099999 6 1 Spb pn sr pt secret 7 5 5 ddsd spb pn sr pt 891099999

• зарплата всех водителей за вчерашний день.

```
1 Create or REPLACE VIEW latest_salary as
2 select worker_id,sum(ordertaxi.cost) as salary from ordertaxi where EXTRACT(DAY from (now() - time_to_come)) = 1 group by worker_id
Data Output Messages Notifications
CREATE VIEW
Query returned successfully in 30 msec.
   1
        SELECT * FROM public.latest_salary
   2
                                   Notifications
 Data Output
                    Messages
 =+
        ₽ ∨ 🖺
                             5
                         worker_id
                          salary
                          numeric 🔓
         integer
 1
                      5
                                25000
 2
                      7
                                 7000
```

# Модификация данных

# Добавление нового работника

# До:

	worker_id [PK] integer	title_id integer	passport_data /	full_name /	current_adress /	schedule text	phone_number integer				
1	1	1	secret	Denis	Spb	pn sr pt	891099999				
2	2	2	fff	asdasd	spb	pn sr pt	891099999				
3	3	1	asad	asdasdzcd	spb	pn sr pt	891099999				
4	4	4	ded	zxxa	spb	pn sr pt	891099999				
5	5	5	zaz	ddsd	spb	pn sr pt	891099999				
6	6	5	dfg	ffg	spb	pn sr pt	891099999				
7	7	1	wer	qwe	spb	pn sr pt	891099999				

```
INSERT INTO public.worker(

worker_id, title_id, passport_data, full_name, current_adress, schedule, phone_number)

VALUES (

select (max(worker_id) + 1) from worker),

(select min(title_id) from job_tittle),

3333333, 'Porosenok', 'Yagni', 'today', 974441);

Data Output Messages Notifications

INSERT 0 1

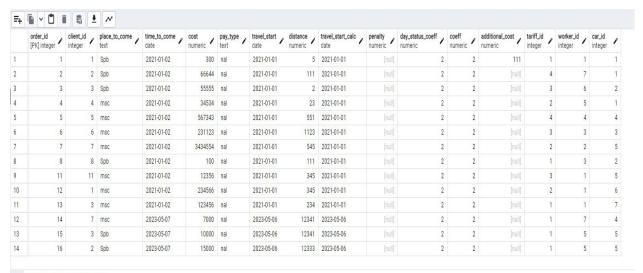
Query returned successfully in 31 msec.
```

## После:

	worker_id .	title_id .	passport_data ,	full_name .	current_adress .	schedule .	phone_number .	
	[PK] integer	integer	text	text	text	text	integer	
1	1	1 1		Denis	Spb	pn sr pt	891099999	
2	2	2	fff	asdasd	spb	pn sr pt	891099999	
3	3	1	asad	asdasdzcd	spb	pn sr pt	891099999	
4	4	4	ded	zxxa	spb	pn sr pt	891099999	
5	5	5	zaz	ddsd	spb	pn sr pt	891099999	
6	6	5	dfg	ffg	spb	pn sr pt	891099999	
7	7	1	wer	qwe	spb	pn sr pt	891099999	
8	8	1	333333	Porosenok	Yagni	today	974441	

#### Увеличить цену поездки если самая дорогая машина в парке.

#### До:



1 UPDATE public.ordertaxi

WHERE worker\_id = (select worker\_id from define\_car join car using(car\_name) join schedule using(schedule\_id) join worker using(worker\_id) join job\_tittle using(title\_id)

where car\_price = (select max(car\_price) from define\_car) and title\_id = 1);

Data Output Messages Notifications

UPDATE 2

Query returned successfully in 35 msec.

#### После:

Data Output Messages Notifications □ □ □ ■ ■ セル time\_to\_come / travel\_start / additional\_cost travel\_start\_calc / pay\_type / day\_status\_coeff order\_id client\_id cost distance penalty coeff tariff\_id worker\_id numeric / [PK] integer numeric 🖍 integer date text date numeric date numeric numeric numeric integer integer integer 2021-01-02 5 2021-01-01 Spb 300 nal 2021-01-01 111 2 2021-01-02 2 Spb 66644 nal 2021-01-01 111 2021-01-01 3 3 Spb 2021-01-02 55555 nal 2021-01-01 2 2021-01-01 2 4 msc 2021-01-02 34534 nal 2021-01-01 23 2021-01-01 5 2021-01-02 567343 nal 2021-01-01 551 2021-01-01 4 2021-01-02 346684.5 nal 2021-01-01 6 msc 1123 2021-01-01 7 msc 2021-01-02 3434554 nal 2021-01-01 545 2021-01-01 5 8 2021-01-02 150.0 nal 2021-01-01 111 2021-01-01 2 8 Spb 9 11 2021-01-02 345 2021-01-01 5 11 msc 12356 nal 2021-01-01 2 2 10 12 msc 2021-01-02 234566 nal 2021-01-01 345 2021-01-01 2 6 11 13 2021-01-02 123456 nal 2021-01-01 234 2021-01-01 2023-05-06 12 14 2023-05-07 7000 nal 2023-05-06 4 7 msc 13 15 2023-05-07 10000 nal 2023-05-06 5 5 3 Spb 12341 2023-05-06 2 14 16 2 Spb 2023-05-07 15000 nal 2023-05-06 12333 2023-05-06 2 5 5

# Удалить заказы клиентов у которых на счету меньше 10 рублей

# До:

	order_id [PK] integer	client_id integer		ext	time_to_come /	cost numeric /	pay_type /	travel_start / distance /	travel_start_calc /	penalty numeric /	day_status_coeff numeric	coeff numeric	additional_cost numeric	tariff_id /	worker_id integer	car_id integer
1	1		1 8	Spb	2021-01-02	300	nal	2021-01-01 5	2021-01-01		2	2	111	1	1	1
2	2		2 5	Spb	2021-01-02	66644	nal	2021-01-01 111	2021-01-01		2	2		4	7	1
3	3		3 5	Spb	2021-01-02	55555	nal	2021-01-01 2	2021-01-01		2	2		3	6	2
4	4		4 г	msc	2021-01-02	34534	nal	2021-01-01 23	2021-01-01		2	2		2	5	1
5	5		5 r	msc	2021-01-02	567343	nal	2021-01-01 551	2021-01-01		2	2		4	4	4
6	6		6 r	nsc	2021-01-02	231123	nal	2021-01-01 1123	2021-01-01		2	2		3	3	3
7	7		7 r	msc	2021-01-02	3434554	nal	2021-01-01 545	2021-01-01		2	2		2	2	
8	8		8 8	Spb	2021-01-02	100	nal	2021-01-01 111	2021-01-01		2	2		1	3	2
9	11		11 r	msc	2021-01-02	12356	nal	2021-01-01 345	2021-01-01		2	2		3	1	5
10	12		1 r	msc	2021-01-02	234566	nal	2021-01-01 345	2021-01-01		2	2		2	1	6
11	13		3 r	msc	2021-01-02	123456	nal	2021-01-01 234	2021-01-01		2	2		1	1	7
12	14		7 r	msc	2023-05-07	7000	nal	2023-05-06 12341	2023-05-06		2	2		1	7	2
13	15		3 8	Spb	2023-05-07	10000	nal	2023-05-06 12341	2023-05-06		2	2		1	5	5
14	16		2 8	Spb	2023-05-07	15000	nal	2023-05-06 12333	2023-05-06		2	2		1	5	

DELETE FROM public.ordertaxi
WHERE client\_id IN((select client\_id from client where client\_number < 10));

Data Output Messages Notifications

Query returned successfully in 19 msec.

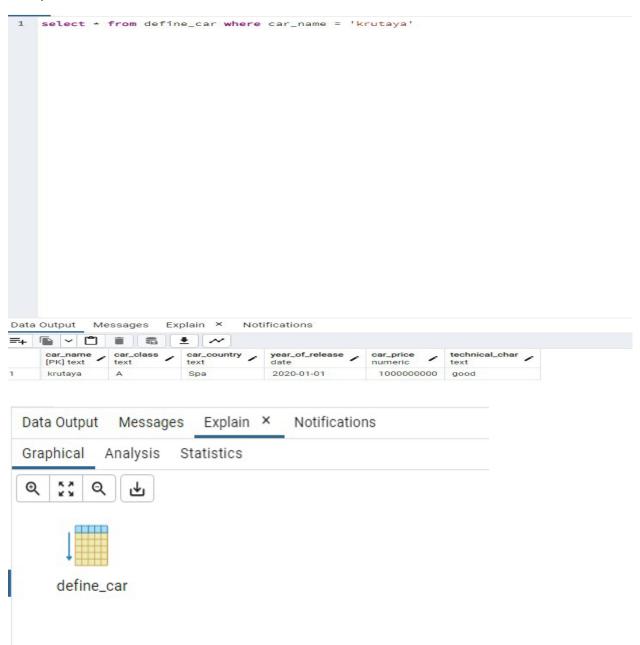
## После:

	order_id [PK] integer	client_id /	place_to_come /	time_to_come /	cost numeric /	pay_type / text	date distance numeric	travel_start_calc /	penalty numeric /	day_status_coeff / numeric	coeff numeric /	additional_cost /	tariff_id /	worker_id /	car_id integer	1
1	1	1	Spb	2021-01-02	300	nal	2021-01-01	2021-01-01		2	2	111	1	1		1
2	6	6	msc	2021-01-02	346684.5	nal	2021-01-01 1123	2021-01-01		2	2		3	3		3
3	7	7	msc	2021-01-02	3434554	nal	2021-01-01 545	2021-01-01		2	2		2	2		5
4	8	8	Spb	2021-01-02	150.0	nal	2021-01-01 111	2021-01-01		2	2	[nulf]	1	3		2
5	11	11	msc	2021-01-02	12356	nal	2021-01-01 345	2021-01-01		2	2		3	1		5
6	12	1	msc	2021-01-02	234566	nal	2021-01-01 345	2021-01-01		2	2		2	1		6
7	14	7	msc	2023-05-07	7000	nal	2023-05-06 12341	2023-05-06		2	2		1	7		4

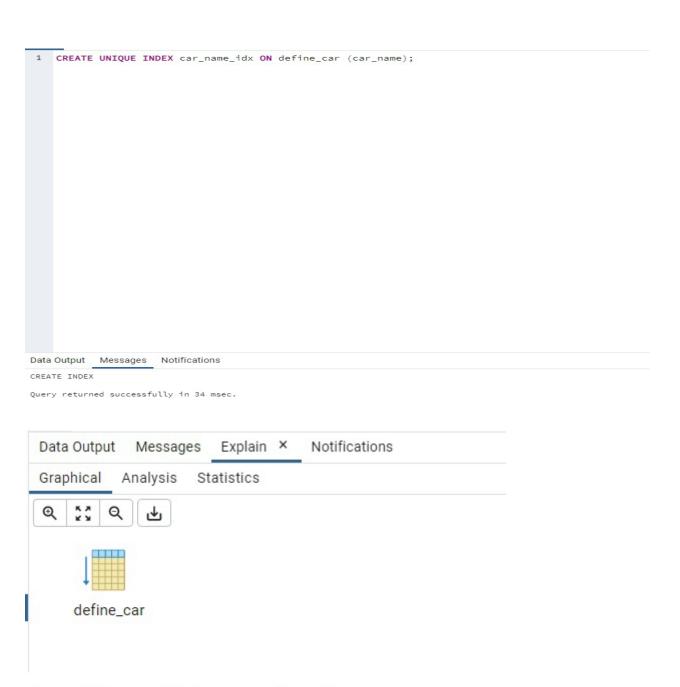
#### Создание индексов

Здесь я решил показать, что использование индексов не всегда является лучшим решением, так как отладчик пути решает как ему быстрее выполнить запрос.

#### 1 запрос:



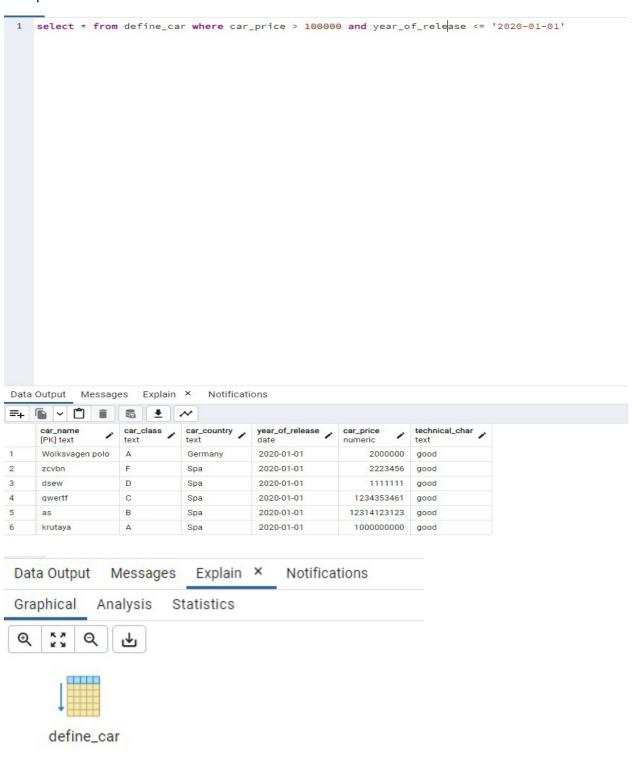
Successfully run. Total query runtime: 32 msec. 1 rows affected.



Successfully run. Total query runtime: 32 msec. 1 rows affected.



#### 2 запрос:



Successfully run. Total query runtime: 31 msec. 1 rows affected.

1	CREATE	INDEX	car_idx	ON define_car	(car_price,	year_of_release);		
Data	Output	Messa	ges Notif	fications				
CREA	TE INDEX							
Quer	y return	ed succe	essfully i	in 28 msec.				
			-					
Su	ccessf	ullv	run. To	otal query n	untime: 32	msec.		
	rows a			, .				
-								
	Node						Rows Actual	Loops
	→ Seq Scan of 1. Filter ((car.	on define_car as define .price > '100000'::numer	_car (rows=d loops=1) ic) AND (year_of_release >= 2000	(0-01-01 indate))			-	6 3

индексы помогают сократить время сложного запроса, но на примере простых запросов мы видим что планировщик считает что лучше просканировать обычным способом

# ЗАКЛЮЧЕНИЕ

SQL запросы позволяют изменять, добавлять или удалять данные, а также составлять различные выборки, подсчитывать числовые характеристики. Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись примерно столько же. Это связано с небольшим количеством данных в таблице.