

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

Факультет **инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки(специальность) **09.03.03 Прикладная**  
**Информатика**

**О Т Ч Е Т**  
**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

Санкт-Петербург 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.

```
1 select * from (  
2 select worker_id,count(*) as b from ordertaxi join worker using(worker_id)  
3 where (worker.title_id = 1 and place_to_come = 'msc') group by worker_id) as d join worker using(worker_id)  
4 where b = (  
5 select max(b) from (  
6 select count(*) as b from ordertaxi join worker using(worker_id)  
7 where (worker.title_id = 1 and place_to_come = 'msc') group by worker_id) as c)
```

Data Output Messages Notifications



	worker_id integer	b bigint	title_id integer	passport_data text	full_name text	current_adress text	schedule text	phone_number integer
1	1	3	1	secret	Denis	Spb	pn sr pt	891099999

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

```
1 select * from car where mileage > 250000 and EXTRACT(Year from Now()) - Extract(Year from last_to_date) > 1
```

Data Output Messages Notifications



	car_id [PK] integer	car_name text	schedule_id integer	car_owner text	car_number text	last_to_date date	mileage integer
1	5	dsew	5	Denis	c999cc994	2020-01-01	555677

3. Сколько раз каждый пассажир воспользовался услугами таксопарка?

```
1 select client_id,count(*) as count_of_visit from ordertaxi group by client_id order by client_id ASC
```

Data Output Messages Notifications



	client_id integer	count_of_visit bigint
1	1	2
2	2	1
3	3	2
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	11	1

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

```
1 select * from (  
2   select count(*) as count_of_visit ,client_id from ordertaxi group by client_id) as b  
3 join client using(client_id) where b.count_of_visit = (  
4   select max(count_of_visit) from (  
5     select count(*) as count_of_visit ,client_id from ordertaxi group by client_id) as d)
```

Data Output Messages Notifications



	client_id integer	count_of_visit bigint	full_name text	client_number integer
1	3	2	rrr	2
2	1	2	Шляхов Денис Олегович	12345

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 integer	title_id integer	passport_data text	full_name text	current_adress text	schedule text	phone_number integer
1	3	1	asad	asdasdzcd	spb	pn sr pt	891099999

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



	client_id integer	workers_count bigint	full_name text	client_number integer
1	11	1	осср	23
2	5	1	ленинград	4
3	4	1	ввв	3
4	6	1	спб	55
5	2	1	ппп	1
6	7	1	точка	66
7	8	1	ру	77

7. Какие автомобили имеют пробег больше среднего пробега для своей марки.

```
1 select * from car join (select car_name,avg(mileage) as avg_mileage from car group by car_name) as b using(car_name) where mileage > avg_mileage
```

Data Output Messages Notifications



	car_name text	car_id [PK] integer	schedule_id integer	car_owner text	car_number text	last_to_date date	mileage integer	avg_mileage numeric
1	krutaya	7	1	Denis	c999cc996	2020-05-01	123445	86722.5000000000000

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

CREATE VIEW

Query returned successfully in 32 msec.

```
1 SELECT * FROM public.free_drivers
2
```

Data Output Messages Notifications

	worker_id integer	title_id integer	passport_data text	full_name text	current_adress text	schedule text	phone_number integer
1	4	4	ded	zxxa	spb	pn sr pt	891099999
2	6	5	dfg	ffg	spb	pn sr pt	891099999
3	2	2	fff	asdasd	spb	pn sr pt	891099999
4	7	1	wer	qwe	spb	pn sr pt	891099999
5	3	1	asad	asdasdzcd	spb	pn sr pt	891099999
6	1	1	secret	Denis	Spb	pn sr pt	891099999
7	5	5	zaz	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(ordertax1.cost) as salary from ordertax1 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
```

Data Output Messages Notifications



	worker_id integer	salary numeric
1	5	25000
2	7	7000



# Модификация данных

## Добавление нового работника

До:

Data OutputMessagesNotifications

	worker_id [PK] integer	title_id integer	passport_data text	full_name text	current_adress text	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

```
1 INSERT INTO public.worker(  
2   worker_id, title_id, passport_data, full_name, current_adress, schedule, phone_number)  
3   VALUES (  
4     (select (max(worker_id) + 1) from worker),  
5     (select min(title_id) from job_title),  
6     333333, 'Porosenok', 'Yagni', 'today', 974441);
```

Data OutputMessagesNotifications

INSERT 0 1







Query returned successfully in 31 msec.

После:

	worker_id [PK] integer	title_id integer	passport_data text	full_name text	current_adress text	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
8	8	1	333333	Porosenok	Yagni	today	974441

Увеличить цену поездки если самая дорогая машина в парке.

До:

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

```
1 UPDATE public.ordertaxi
2 SET cost = cost * 1.5
3 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_title using(title_id)
4 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

	order_id [PK] integer	client_id integer	place_to_come text	time_to_come date	cost numeric	pay_type text	travel_start date	distance numeric	travel_start_calc date	penalty numeric	day_status_coef numeric	coeff numeric	additional_cost numeric	tariff_id integer	worker_id integer	car_id integer	
1		1	1 Spb	2021-01-02	300	nal	2021-01-01	5	2021-01-01	[null]		2	2	111	1	1	1
2		2	2 Spb	2021-01-02	66644	nal	2021-01-01	111	2021-01-01	[null]		2	2	[null]	4	7	1
3		3	3 Spb	2021-01-02	55555	nal	2021-01-01	2	2021-01-01	[null]		2	2	[null]	3	6	2
4		4	4 msc	2021-01-02	34534	nal	2021-01-01	23	2021-01-01	[null]		2	2	[null]	2	5	1
5		5	5 msc	2021-01-02	567343	nal	2021-01-01	551	2021-01-01	[null]		2	2	[null]	4	4	4
6		6	6 msc	2021-01-02	346684.5	nal	2021-01-01	1123	2021-01-01	[null]		2	2	[null]	3	3	3
7		7	7 msc	2021-01-02	3434554	nal	2021-01-01	545	2021-01-01	[null]		2	2	[null]	2	2	5
8		8	8 Spb	2021-01-02	150.0	nal	2021-01-01	111	2021-01-01	[null]		2	2	[null]	1	3	2
9		11	11 msc	2021-01-02	12356	nal	2021-01-01	345	2021-01-01	[null]		2	2	[null]	3	1	5
10		12	1 msc	2021-01-02	234566	nal	2021-01-01	345	2021-01-01	[null]		2	2	[null]	2	1	6
11		13	3 msc	2021-01-02	123456	nal	2021-01-01	234	2021-01-01	[null]		2	2	[null]	1	1	7
12		14	7 msc	2023-05-07	7000	nal	2023-05-06	12341	2023-05-06	[null]		2	2	[null]	1	7	4
13		15	3 Spb	2023-05-07	10000	nal	2023-05-06	12341	2023-05-06	[null]		2	2	[null]	1	5	5
14		16	2 Spb	2023-05-07	15000	nal	2023-05-06	12333	2023-05-06	[null]		2	2	[null]	1	5	5

Удалить заказы клиентов у которых на счету меньше 10 рублей

До:

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

```
1 DELETE FROM public.ordertaxi
2 WHERE client_id IN((select client_id from client where client_number < 10));|
```

Data Output Messages Notifications

DELETE 7  
Query returned successfully in 19 msec.

После:

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

```
1 select * from define_car where car_name = 'krutaya'
```

	car_name [PK] text	car_class text	car_country text	year_of_release date	car_price numeric	technical_char text
1	krutaya	A	Spa	2020-01-01	1000000000	good

Data Output Messages Explain × Notifications

Graphical Analysis Statistics

  
define\_car

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

```
1 CREATE UNIQUE INDEX car_name_idx ON define_car (car_name);
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 34 msec.

Data Output Messages Explain × Notifications

Graphical Analysis Statistics



define\_car

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

Graphical Analysis Statistics

#	Node	Rows	
		Actual	Loops
1.	→ Seq Scan on define_car as define_car (rows=1 loops=1) Filter (car_name = 'trdaya;text') Rows Removed by Filter: 5	1	1


2 запрос :

```
1 select * from define_car where car_price > 100000 and year_of_release <= '2020-01-01'
```

Data Output Messages Explain × Notifications						
	car_name [PK] text	car_class text	car_country text	year_of_release date	car_price numeric	technical_char text
1	Wolkswagen polo	A	Germany	2020-01-01	2000000	good
2	zcvbn	F	Spa	2020-01-01	2223456	good
3	dsew	D	Spa	2020-01-01	1111111	good
4	qwerty	C	Spa	2020-01-01	1234353461	good
5	as	B	Spa	2020-01-01	12314123123	good
6	krutaya	A	Spa	2020-01-01	1000000000	good

Data Output Messages Explain × Notifications

Graphical Analysis Statistics

  
define\_car

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 Messages Notifications

CREATE INDEX

Query returned successfully in 28 msec.

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

#	Node	Rows		Loops
		Actual	Estimated	
1	→ Seq Scan on define_car as define_car (rows=1) Filter: (car_price <= 10000000000) AND (year_of_release <= 2000-01-01 'date') Rows Removed by Filter: 0	1	1	1

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.