

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по лабораторной работе № 4
«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ
ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В
POSTGRESQL»
по дисциплине:
«Проектирование и реализация баз данных»

Выполнила:
студентка II курса ИКТ
группы К3240
Никифорова Кюнна́й Васильевна

Проверила:
Говорова 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 1X, 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:

- I. Наименование БД: «Служба заказа такси»
- II. Схема логической модели базы данных, сгенерированная в Generate ERD:

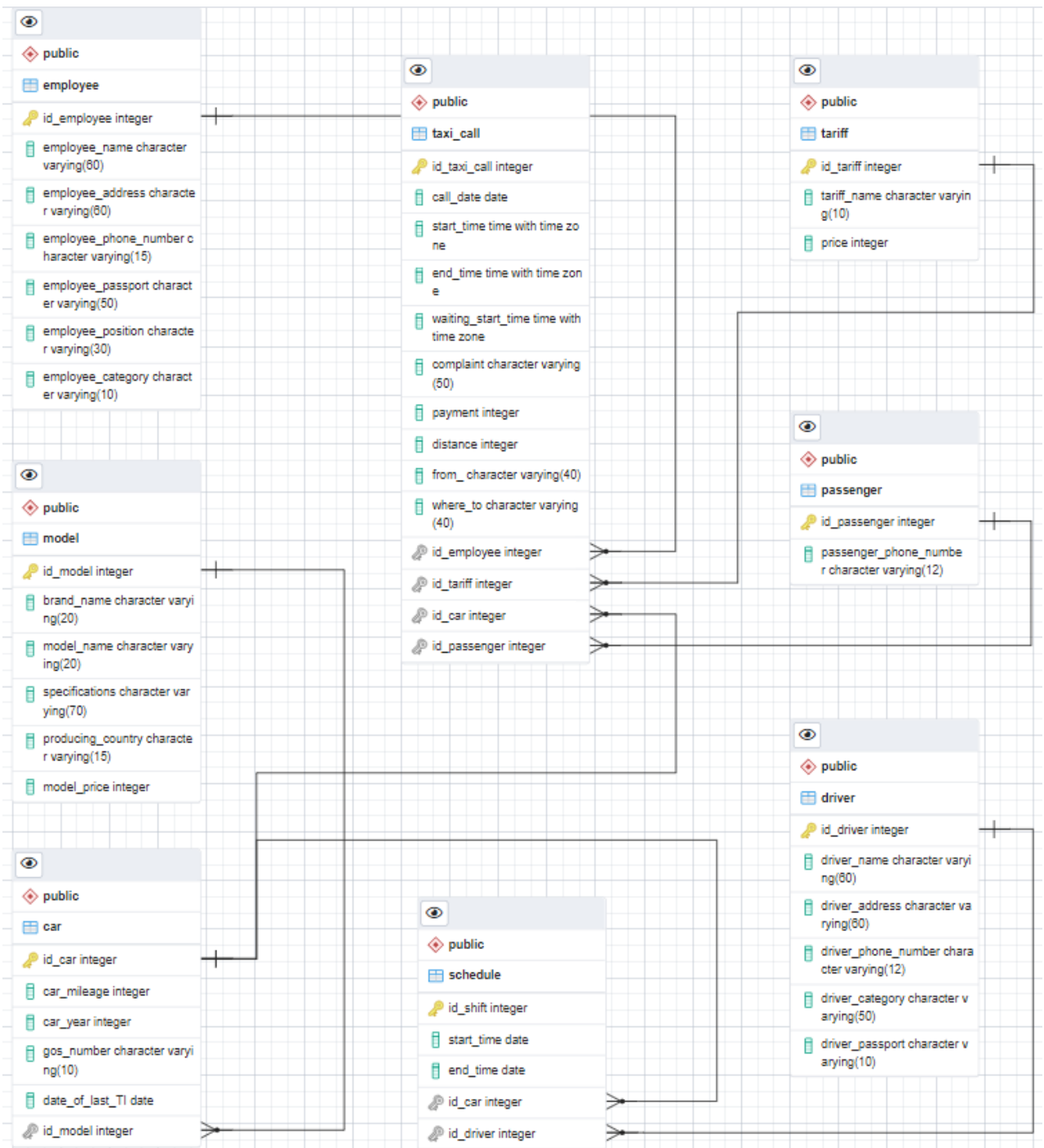


Рис. 1 - Схема логической модели базы данных

III. Запросы к базе данных

- Вывести данные о водителе, который чаще всех доставляет пассажиров на заданную улицу (Невский проспект)

Query Editor История запросов

```
1 SELECT driver.id_driver, driver.driver_name, driver.driver_address, driver.driver_phone_number,
2 driver.driver_category, driver.driver_passport
3 FROM driver, schedule, taxi_call
4 WHERE schedule.id_car = taxi_call.id_car
5       AND driver.id_driver = schedule.id_driver
6       AND taxi_call.where_to LIKE 'Nevsky Avenue%'
7 GROUP BY driver.id_driver
8 HAVING COUNT(taxi_call.where_to) = (SELECT COUNT(taxi_call.where_to) AS count_s
9 FROM taxi_call, schedule
10 WHERE schedule.id_car = taxi_call.id_car
11       AND taxi_call.where_to LIKE 'Nevsky Avenue%'
12 GROUP BY schedule.id_driver
13 ORDER BY count_s DESC
14 LIMIT 1)
```

Результат План выполнения Сообщения Notifications

	id_driver [PK] integer	driver_name character varying (60)	driver_address character varying (60)	driver_phone_number character varying (12)	driver_category character varying (50)	driver_passport character varying (10)	
1	5	Kalinin Pavel	Lomonosova 22	+79022042768	B	9810897652	

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

Query Editor История запросов

```
1 SELECT car.id_car, car.car_mileage, car.gos_number, car."date_of_last_TI", model.brand_name, model.model_name
2 FROM car, model
3 WHERE car.id_model = model.id_model
4       AND car.car_mileage > 250000
5       AND date_part('year', "date_of_last_TI") < date_part('year', CURRENT_DATE)
```

Результат План выполнения Сообщения Notifications

	id_car integer	car_mileage integer	gos_number character varying (10)	date_of_last_TI date	brand_name character varying (20)	model_name character varying (20)	
1	1	300322	k981as	2021-09-10	Ford	Falcon	
2	4	261134	l396kc	2021-11-30	BMW	4 Series	
3	7	256931	t785cp	2021-12-26	Chevrolet	Camaro	
4	8	297455	k677ae	2021-08-08	Ford	Falcon	

- Сколько раз каждый пассажир воспользовался услугами таксопарка?

Query Editor

История запросов

1

SELECT id_passenger, COUNT(id_passenger) as count_call

2

FROM taxi_call

3

GROUP BY id_passenger

Результат

План выполнения

Сообщения

Notifications

	<div>id_passenger</div> <div>integer</div>	<div>count_call</div> <div>bigint</div>	
1	15	1	
2	14	2	
3	13	2	
4	16	3	

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

1	SELECT passenger.id_passenger, passenger.passenger_phone_number		
2	FROM passenger, taxi_call		
3	WHERE passenger.id_passenger = taxi_call.id_passenger		
4	GROUP BY passenger.id_passenger		
5	HAVING COUNT(passenger.id_passenger) = (SELECT COUNT(id_passenger)		
6	FROM taxi_call		
7	GROUP BY id_passenger		
8	ORDER BY COUNT(id_passenger) DESC		
9	LIMIT 1)		

Результат		План выполнения		Сообщения		Notifications	
	id_passenger [PK] integer		passenger_phone_number character varying (12)				
1	16		+79869338925				

- Вывести данные о водителе, который ездит на самом дорогом автомобиле.

Query Editor

История запросов

```
1 SELECT driver.id_driver, driver.driver_name, driver.driver_address, driver.driver_phone_number,
2 driver.driver_category, driver.driver_passport
3 FROM car, model, driver, schedule
4 WHERE driver.id_driver = schedule.id_driver
5        AND schedule.id_car = car.id_car
6        AND car.id_model = model.id_model
7        AND model.model_price = (SELECT MAX(model_price) FROM model)
```

Результат

План выполнения

Сообщения

Notifications

	id_driver [PK] integer	driver_name character varying (60)	driver_address character varying (60)	driver_phone_number character varying (12)	driver_category character varying (50)	driver_passport character varying (10)	
1	6	Balandina Alina	Belarusskaya 6	+79168816475	C	9808278270	

- Вывести данные пассажира, который всегда ездит с одним и тем же водителем.

Query Editor История запросов

```
1 SELECT passenger.id_passenger, passenger.passenger_phone_number
2 FROM passenger, taxi_call, schedule
3 WHERE schedule.id_car = taxi_call.id_car
4       AND passenger.id_passenger = taxi_call.id_passenger
5 GROUP BY passenger.id_passenger
6 HAVING COUNT(DISTINCT schedule.id_driver) = 1
```

Результат План выполнения Сообщения Notifications

	id_passenger [PK] integer	passenger_phone_number character varying (12)	
1	13	+79342045795	
2	14	+79045536872	
3	15	+79867438927	

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

Query Editor История запросов

```
1 SELECT car.id_car, car.car_mileage, t_mileage.id_model, t_mileage.mean_mileage FROM car,
2 (SELECT id_model, round(AVG(car.car_mileage)) as mean_mileage
3 FROM car
4 GROUP BY id_model) as t_mileage
5 WHERE car.id_model = t_mileage.id_model
6       AND car.car_mileage >= t_mileage.mean_mileage
```

Результат План выполнения Сообщения Notifications

	id_car [PK] integer	car_mileage integer	id_model integer	mean_mileage numeric	
1	1	300322	1	207557	
2	3	120432	2	77213	
3	4	261134	3	146905	
4	5	104569	2	77213	
5	7	256931	4	130643	
6	8	297455	1	207557	

IV. Представления

- содержащее сведения о незанятых на данный момент водителях

Query Editor История запросов

```
1 CREATE VIEW public.free_driver AS
2     SELECT driver.id_driver, driver.driver_name, driver.driver_address, driver.driver_phone_number,
3           driver.driver_category, driver.driver_passport
4     FROM driver, schedule
5     WHERE driver.id_driver = schedule.id_driver
6           AND schedule.end_time IS NULL
7           AND schedule.start_time = CURRENT_DATE
```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 35 msec.

Query Editor История запросов

```
1 SELECT * FROM public.free_driver
```

Результат План выполнения Сообщения Notifications

	id_driver integer	driver_name character varying (60)	driver_address character varying (60)	driver_phone_number character varying (12)	driver_category character varying (50)	driver_passport character varying (10)	
1	7	Balabanov Viktor	Nevsky Avenue 40	+79847229743	B	9814504853	

- зарплата всех водителей за вчерашний день

Query Editor История запросов

```
1 CREATE VIEW public.yesterday_salary AS
2     SELECT SUM(tariff.price * taxi_call.distance)
3     FROM taxi_call, tariff
4     WHERE tariff.id_tariff = taxi_call.id_tariff
5           AND taxi_call.call_date = CURRENT_DATE - 1
```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 36 msec.

Query Editor История запросов

```
1 SELECT * FROM public.yesterday_salary
```

Результат План выполнения Сообщения Notifications

	sum bigint	
1	1672000	

V. Запросы на модификацию данных

- INSERT

Запрос: *Добавить в базу данных новый автомобиль*

Query Editor История запросов

```
1 INSERT INTO public.car (id_car, car_mileage, car_year, gos_number, "date_of_last_TI", id_model)
2 OVERRIDING SYSTEM VALUE VALUES (
3     (SELECT MAX(id_car) + 1 FROM public.car), 43638, 2020, 'x394ej', '2022-01-26',
4     (SELECT id_model FROM public.model WHERE model_name = 'Falcon')
5 )
```

Результат План выполнения Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 59 msec.

До:

Результат План выполнения Сообщения Notifications

	id_car [PK] integer	car_mileage integer	car_year integer	gos_number character varying (10)	date_of_last_TI date	id_model integer
1	1	300322	2018	k981as	2021-09-10	1
2	2	4355	2021	c165cc	2021-12-26	4
3	3	120432	2019	w580ee	2022-01-08	2
4	4	261134	2018	l396kc	2021-11-30	3
5	5	104569	2020	m266qw	2022-01-18	2
6	6	32675	2020	w222eq	2022-02-10	3
7	7	256931	2019	t785cp	2021-12-26	4
8	8	297455	2018	k677ae	2021-08-08	1
9	9	6639	2021	m896kr	2021-11-30	2
10	10	24894	2019	e927qw	2022-01-18	1

После:

Результат План выполнения Сообщения Notifications

	id_car [PK] integer	car_mileage integer	car_year integer	gos_number character varying (10)	date_of_last_TI date	id_model integer
1	1	300322	2018	k981as	2021-09-10	1
2	2	4355	2021	c165cc	2021-12-26	4
3	3	120432	2019	w580ee	2022-01-08	2
4	4	261134	2018	l396kc	2021-11-30	3
5	5	104569	2020	m266qw	2022-01-18	2
6	6	32675	2020	w222eq	2022-02-10	3
7	7	256931	2019	t785cp	2021-12-26	4
8	8	297455	2018	k677ae	2021-08-08	1
9	9	6639	2021	m896kr	2021-11-30	2
10	10	24894	2019	e927qw	2022-01-18	1
11	11	43638	2020	x394ej	2022-01-26	1

- UPDATE

Запрос: *Повысить цену тарифа на 60 рублей у тарифов 'night' и 'rush hour'*

Query Editor

История запросов

1

UPDATE public.tariff

2

SET price = price + 60

3

WHERE tariff_name IN (

4

SELECT tariff_name

5

FROM tariff

6

WHERE tariff_name = 'night' or tariff_name = 'rush hour'

7

)

Результат

План выполнения

Сообщения

Notifications

UPDATE 2

Запрос завершён успешно, время выполнения: 80 msec.

До:

Результат				План выполнения	Сообщения	Notifications
	id_tariff [PK] integer		tariff_name character varying (10)		price integer	
1		1	morning		100	
2		2	day		80	
3		3	night		120	
4		4	rush hour		140	

После:

Результат				План выполнения	Сообщения	Notifications
	id_tariff [PK] integer		tariff_name character varying (10)		price integer	
1		1	morning		100	
2		2	day		80	
3		3	night		180	
4		4	rush hour		200	

- DELETE

Запрос: Удалить из базы данных вызовы такси, которые были совершены более, чем 3 месяца назад

Query Editor

История запросов

1

DELETE FROM public.taxi_call

2

WHERE id_taxi_call IN (

3

SELECT id_taxi_call

4

FROM taxi_call

5

WHERE CURRENT_DATE - INTERVAL '3 MONTH' > call_date

6

)

Результат

План выполнения

Сообщения

Notifications

DELETE 3

Запрос завершён успешно, время выполнения: 51 msec.

До:

	id_taxi_call [PK] integer	call_date date	start_time time with time zone	end_time time with time zone	waiting_start_time time with time zone	complaint character varying (50)	payment integer	distance integer
1	1	2021-11-10	12:25:05+03:00	12:48:32+03:00	12:23:00+03:00		316	5600
2	2	2021-12-28	01:42:11+03:00	02:09:30+03:00	01:40:00+03:00	some text	608	4900
3	3	2022-01-02	19:20:01+03:00	20:35:09+03:00	19:17:00+03:00		544	6600
4	4	2022-01-19	15:55:05+03:00	16:18:32+03:00	15:52:00+03:00	some text	268	3600
5	5	2022-02-25	04:12:11+03:00	04:39:30+03:00	04:10:00+03:00	some text	408	4500
6	6	2022-03-27	09:16:01+03:00	10:03:09+03:00	09:12:00+03:00		585	8600
7	7	2022-04-02	08:42:23+03:00	09:04:30+03:00	08:40:00+03:00	some text	234	5300
8	8	2022-04-02	23:07:01+03:00	23:33:09+03:00	23:03:00+03:00		485	7400

После:

	id_taxi_call [PK] integer	call_date date	start_time time with time zone	end_time time with time zone	waiting_start_time time with time zone	complaint character varying (50)	payment integer	distance integer
1	4	2022-01-19	15:55:05+03:00	16:18:32+03:00	15:52:00+03:00	some text	268	3600
2	5	2022-02-25	04:12:11+03:00	04:39:30+03:00	04:10:00+03:00	some text	408	4500
3	6	2022-03-27	09:16:01+03:00	10:03:09+03:00	09:12:00+03:00		585	8600
4	7	2022-04-02	08:42:23+03:00	09:04:30+03:00	08:40:00+03:00	some text	234	5300
5	8	2022-04-02	23:07:01+03:00	23:33:09+03:00	23:03:00+03:00		485	7400

VI. Создание индексов

- Без индексов (Запрос №3)

– План выполнения запроса

Query Editor История запросов

```
1 EXPLAIN SELECT id_passenger, COUNT(id_passenger) as count_call
2 FROM taxi_call
3 GROUP BY id_passenger
```

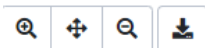
Результат План выполнения Сообщения Notifications

QUERY PLAN		text	
1	HashAggregate (cost=12.85..14.75 rows=190 width=12)		
2	[...] Group Key: id_passenger		
3	[...] -> Seq Scan on taxi_call (cost=0.00..11.90 rows=190 width=4)		

– Графическое выполнение запроса

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика



public.taxi_call

Aggregate

– Время выполнения запроса

Результат План выполнения Сообщения Notifications

Запрос выполнен успешно. Общее время выполнения: 57 мсек.
обработано строк: 3.

- Без индексов (Запрос №6)

- План выполнения запроса

Query Editor История запросов

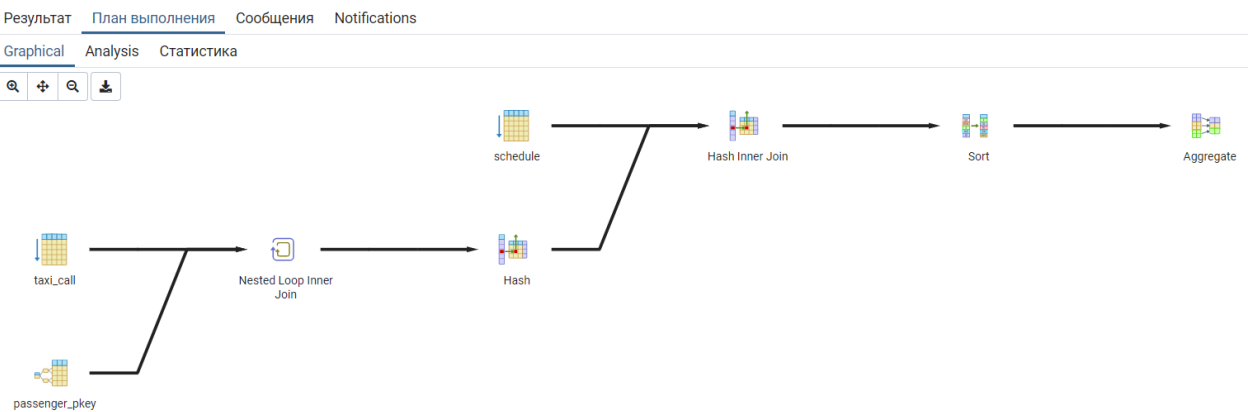
```
1 EXPLAIN SELECT passenger.id_passenger, passenger.passenger_phone_number
2 FROM passenger, taxi_call, schedule
3 WHERE schedule.id_car = taxi_call.id_car
4       AND passenger.id_passenger = taxi_call.id_passenger
5 GROUP BY passenger.id_passenger
6 HAVING COUNT (DISTINCT schedule.id_driver) = 1
```

Результат План выполнения Сообщения Notifications

QUERY PLAN
text

1	GroupAggregate (cost=60.90..61.74 rows=1 width=46)
2	[...] Group Key: passenger.id_passenger
3	[...] Filter: (count(DISTINCT schedule.id_driver) = 1)
4	[...] -> Sort (cost=60.90..61.00 rows=42 width=50)
5	[...] Sort Key: passenger.id_passenger
6	[...] -> Hash Join (cost=25.97..59.77 rows=42 width=50)
7	[...] Hash Cond: (schedule.id_car = taxi_call.id_car)
8	[...] -> Seq Scan on schedule (cost=0.00..27.00 rows=1700 width=8)
9	[...] -> Hash (cost=25.91..25.91 rows=5 width=50)
10	[...] -> Nested Loop (cost=0.15..25.91 rows=5 width=50)
11	[...] -> Seq Scan on taxi_call (cost=0.00..1.05 rows=5 width=8)
12	[...] -> Index Scan using passenger_pkey on passenger (cost=0.15..4.97 rows=1 width=46)
13	[...] Index Cond: (id_passenger = taxi_call.id_passenger)

- Графическое выполнение запроса



- Время выполнения запроса

Результат План выполнения Сообщения Notifications

Запрос выполнен успешно. Общее время выполнения: 46 мсек.
обработано строк: 13.

- **С индексами (Запрос №3)**

- Создание индекса и план выполнения запроса

[Query Editor](#) [История запросов](#)

```
1 CREATE INDEX index_count_call on public.taxi_call(id_passenger)
```

- Время выполнения запроса

[Результат](#) [План выполнения](#) [Сообщения](#) [Notifications](#)

CREATE INDEX

Запрос завершён успешно, время выполнения: 54 msec.

- **С индексами (Запрос №6)**

- Создание индекса и план выполнения запроса

[Query Editor](#) [История запросов](#)

```
1 CREATE UNIQUE INDEX index_passenger ON public.passenger(id_passenger, passenger_phone_number)
```

- Время выполнения запроса

[Результат](#) [План выполнения](#) [Сообщения](#) [Notifications](#)

CREATE INDEX

Запрос завершён успешно, время выполнения: 39 msec.

- **Удаление индексов**

Query Editor История запросов

```
1 DROP INDEX index_count_call
```

Результат План выполнения Сообщения Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 70 msec.

Query Editor История запросов

```
1 DROP INDEX index_passenger
```

Результат План выполнения Сообщения Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 46 msec.

С индексами время выполнения вышло немного меньше, чем без индексов, но загвоздка в том, что индексы у меня создаются, а план выполнения не показывается, может это из-за того, что данных не достаточно.

Выводы:

В результате выполненной работы:

- созданы запросы и представления;
- составлены запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов;
- созданы простой и составной индексы для двух запросов и сравнены время выполнения запросов без индексов и с индексами.