

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий

Лабораторная работа №2
«Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»
по дисциплине «Проектирование и
реализация баз данных»

Выполнил:
студент II курса ИКТ
группы К3241
Конев А.В.

Проверил:
Говорова М.М.

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.

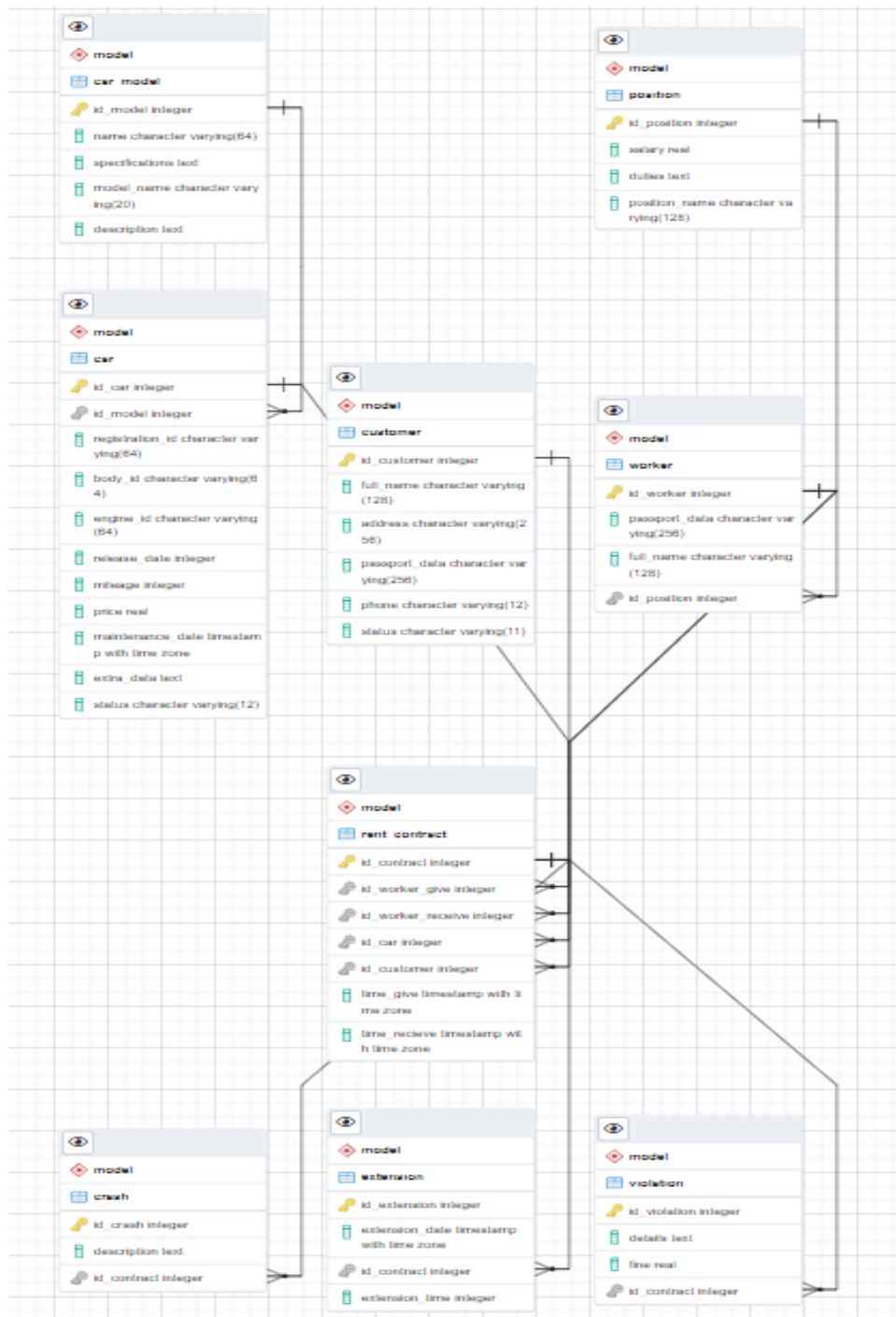
Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

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:

1. Схема логической модели базы данных



2. Запросы и представления на выборку данных к базе данных

Запросы

1 - Какой автомобиль находился в прокате максимальное количество часов?

Query Editor Query History

```
1 SELECT c1.id_car, date_trunc('hours', sum(r1.time_recieve - r1.time_give)) as max_time_rent
2 FROM model.car c1 JOIN model.rent_contract r1 ON c1.id_car = r1.id_car
3 GROUP BY c1.id_car
4 HAVING date_trunc('hours', sum(r1.time_recieve - r1.time_give))
5 >= ALL(SELECT date_trunc('hours', sum(r.time_recieve - r.time_give))
6        FROM model.car c JOIN model.rent_contract r ON c.id_car = r.id_car
7        GROUP BY c.id_car);
```

Data Output Explain Messages Notifications

	id_car [PK] integer	max_time_rent interval	
1	4	1 day 02:00:00	

2 - Автомобили какой марки чаще всего брались в прокат?

```
1 SELECT m.name, COUNT(*) FROM
2 (model.car_model m JOIN model.car c ON m.id_model = c.id_model)
3 JOIN model.rent_contract r ON r.id_car = c.id_car
4 GROUP BY m.name
5 HAVING COUNT(*) >= ALL(
6     SELECT COUNT(*) FROM (model.car_model m1 JOIN model.car c1 ON c1.id_model = m1.id_model)
7     JOIN model.rent_contract r1 ON c1.id_car = r1.id_car
8     GROUP BY m1.name);
```

Data Output Explain Messages Notifications

	name character varying (64)	count bigint	
1	chevrolet	2	
2	honda	2	

3 - Определить убытки от простоя автомобилей за вчерашний день.

```
1 SELECT c.id_car, c.price *
2 (24 - COALESCE(date_part('hours', SUM(r.time_recieve - r.time_give)), 0)) AS loss
3 FROM model.car c LEFT JOIN
4 (SELECT * FROM model.rent_contract r1
5  WHERE r1.time_give > 'yesterday'
6  AND r1.time_recieve < 'today') AS r
7 ON c.id_car = r.id_car
8 GROUP BY c.id_car;
```

Data Output Explain Messages Notifications

	id_car [PK] integer	loss double precision	
1	1	1365.6000366210938	
2	3	136029.59765625	
3	5	818.3999633789062	
4	4	11157.599853515625	
5	2	1610.3999633789062	

4 - Вывести данные автомобиля, имеющего максимальный пробег.

```

1 SELECT * FROM model.car c1
2 WHERE NOT EXISTS (
3     SELECT * FROM model.car c2
4     WHERE c2.mileage > c1.mileage)

```

	id_car [PK] integer	id_model integer	registration_id character varying (64)	body_id character varying (64)	engine_id character varying (64)	release_date integer	mileage integer	price real	maintenance_date timestamp with time zone	extra_c text
1		5	5 13131	353	7575	1976	67435	34.1	2015-04-23 16:09:56.819+00	2018-0

5 - Какой автомобиль суммарно находился в прокате дольше всех?

```

1 SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent
2 FROM model.car c1 JOIN model.rent_contract r1 ON c1.id_car = r1.id_car
3 GROUP BY c1.id_car
4 HAVING SUM(r1.time_recieve - r1.time_give)
5 >= ALL(SELECT SUM(r.time_recieve - r.time_give)
6 FROM model.car c JOIN model.rent_contract r ON r.id_car = c.id_car
7 GROUP BY c.id_car);

```

	id_car [PK] integer	max_time_rent interval
1	4	1 day 02:00:10.805

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

```

1 SELECT COUNT(*), m.name
2 FROM model.car c JOIN model.car_model m
3 ON c.id_model = m.id_model
4 GROUP BY m.name;
5

```

	count bigint	name character varying (64)
1	1	toyota
2	1	chevrolet
3	2	honda
4	1	dodge

7 - Определить средний “возраст” автомобилей компании.

```

1 SELECT AVG(date_part('year', CURRENT_DATE) - c.release_date) AS avg_age
2 FROM model.car c;
3

```

	avg_age double precision
1	50.2

Представления

1 - Какой автомобиль ни разу не был в прокате?

Query Editor Query History

```
1 CREATE OR REPLACE VIEW model.not_rented_once
2 AS SELECT * FROM model.car c
3 WHERE NOT EXISTS (SELECT r.id_contract FROM model.rent_contract r
4                     WHERE c.id_car = r.id_car);
5
```

Data Output Explain Messages Notifications

CREATE VIEW

Query returned successfully in 90 msec.

1 SELECT * FROM model.not_rented_once

Data Output Explain Messages Notifications

id_car	id_model	registration_id	body_id	engine_id	release_date	mileage	price	maintenance_date	extra_data
integer	integer	character varying (64)	character varying (64)	character varying (64)	integer	integer	real	timestamp with time zone	text

2 - Вывести данные клиентов, не вернувших автомобиль вовремя.

```
1 CREATE OR REPLACE VIEW model.banned_list
2 AS SELECT c.id_customer, c.full_name, c.address, c.passport_data, c.phone, c.status
3 FROM model.customer c
4 WHERE c.status = 'banned';
5
```

Data Output Explain Messages Notifications

CREATE VIEW

Query returned successfully in 104 msec.

1 SELECT * FROM model.banned_list

2

Data Output Explain Messages Notifications

id_customer	full_name	address	passport_data	phone	status
integer	character varying (128)	character varying (256)	character varying (256)	character varying (12)	charac
1	995 Шелыгин Давид Давидович	Россия, г. Северск, Спортивная ул., д. 16 кв.182	4429 404530 710-553 ОВД России по г. Северск 11.09.2016	+74953415243	banned

3. Запросы на модификацию данных

1 - Увеличить стоимость аренды всех автомобилей марки honda на 230 руб.

До

```
1 SELECT c.id_car, c.price
2 FROM model.car c JOIN model.car_model m
3 ON c.id_model = m.id_model
4 WHERE m.name = 'honda'
```

Data Output Explain Messages Notifications

id_car	price
[PK] integer	real
1	56.9
2	5667.9

Запрос

```
1 UPDATE model.car
2 SET price = price + 230.0
3 WHERE id_car IN(SELECT c.id_car
4 FROM model.car c JOIN model.car_model m
5 ON c.id_model = m.id_model
6 WHERE m.name = 'honda')
7
```

Data Output Explain Messages Notifications

UPDATE 2

Query returned successfully in 78 msec.

После

```
1 SELECT c.id_car, c.price
2 FROM model.car c JOIN model.car_model m
3 ON c.id_model = m.id_model
4 WHERE m.name = 'honda'
5
```

Data Output Explain Messages Notifications

	id_car [PK] integer	price real
1	1	286.9
2	3	5897.9

2 - Удалить модель машины, которая списана

До

```
1 SELECT * FROM model.car_model
2
```

Data Output Explain Messages Notifications

	id_model [PK] integer	name character varying (64)	specifications text	modelName character varying (20)	description text
1	2	honda	some text	honda civic	some text
2	3	chevrolet	some text	chevrolet camaro	some text
3	4	honda	bad tires	honda accord	trash
4	5	toyota	some japanese text	toyota hilix	some text
5	1	dodge	some text	dodge ram	some text
6	6	test brand	some notes	test brand k	decommissio...

Запрос

```
1 DELETE FROM model.car_model
2 WHERE id_model IN(
3 SELECT m.id_model
4 FROM model.car_model m
5 WHERE description = 'decommissioned')
```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 62 msec.

После

```
1 SELECT * FROM model.car_model
```

	id_model [PK] integer	name character varying (64)	specifications text	model_name character varying (20)	description text
1	2	honda	some text	honda civic	some text
2	3	chevrolet	some text	chevrolet camaro	some text
3	4	honda	bad tires	honda accord	trash
4	5	toyota	some japanese text	toyota hilix	some text
5	1	dodge	some text	dodge ram	some text

3 - Нанять нового работника на позицию giver До

```
1 SELECT * FROM model.worker
```

	id_worker [PK] integer	passport_data character varying (256)	full_name character varying (128)	id_position integer
1	100	4522 499855 870-600 ОУФМС России по г. Москва 22.06.2018 Россия, г. Москва, Мирная ул., д. 13 кв.65	Константинов Касьян Владленович	123
2	101	4723 734357 740-841 Отделом УФМС России по г. Уфа 05.05.2017 Россия, г. Уфа, Космонавтов ул., д. 8 кв.120	Бобылёв Петр Русланович	234
3	102	4675 517566 690-379 Отделом УФМС России по г. Дербент 19.12.2014 Россия, г. Дербент, Октябрьская ул., д. 10 кв.104	Доронин Герман Геннадьевич	345
4	103	4828 427226 950-913 Отделением УФМС России по г. Арзамас 13.08.2018 Россия, г. Арзамас, Луговая ул., д. 17 кв.114	Фёдоров Исаак Артемович	456

Запрос

```
1 INSERT INTO model.worker(id_worker, passport_data, full_name, id_position)
2 VALUES(
3     111,
4     'sometestdata',
5     'sometestname',
6     (SELECT id_position FROM model.position WHERE position_name = 'giver')
7 )
```

Data Output	Explain	Messages	Notifications
INSERT 0 1			
Query returned successfully in 84 msec.			

После

```
1 SELECT * FROM model.worker
```

	id_worker [PK] integer	passport_data character varying (256)	full_name character varying (128)	id_position integer
1	100	4522 499855 870-600 ОУФМС России по г. Москва 22.06.2018 Россия, г. Москва, Мирная ул., д. 13 кв.65	Константинов Касьян Владленович	123
2	101	4723 734357 740-841 Отделом УФМС России по г. Уфа 05.05.2017 Россия, г. Уфа, Космонавтов ул., д. 8 кв.120	Бобылёв Петр Русланович	234
3	102	4675 517566 690-379 Отделом УФМС России по г. Дербент 19.12.2014 Россия, г. Дербент, Октябрьская ул., д. 10 кв.104	Доронин Герман Геннадьевич	345
4	103	4828 427226 950-913 Отделением УФМС России по г. Арзамас 13.08.2018 Россия, г. Арзамас, Луговая ул., д. 17 кв.114	Фёдоров Исаак Артемович	456
5	111	sometestdata	sometestname	123

4. История запросов

▶ SELECT * FROM model.banned_list
20:51:49
▶ CREATE OR REPLACE VIEW model.banned_list AS SELECT c.id_customer, c.full_name, c.address...
20:50:52
▶ CREATE OR REPLACE VIEW model.banned_list AS SELECT cus.id_customer, cus.full_name, cus.a...
20:50:37
▶ CREATE OR REPLACE VIEW model.banned_clients AS SELECT cus.id_customer, cus.full_name, cu...
20:49:56
▶ CREATE OR REPLACE VIEW model.not_rented_once AS SELECT * FROM model.car c WHERE NOT EXIS...
20:32:39
▶ SELECT AVG(date_part('year', CURRENT_DATE) - c.release_date) AS avg_age FROM model.car c;
20:14:25
▶ SELECT COUNT(*), m.name FROM model.car c JOIN model.car_model m ON c.id_model = m.id_mod...
20:07:16
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:03:20
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:03:15
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:02:58
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:01:51
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:00:32
▶ SELECT c1.id_car, SUM(r1.time_recieve - r1.time_give) AS max_time_rent FROM model.car c1...
20:00:03
▶ SELECT c.id_car, date_part('hours', SUM(r.time_recieve - r.time_give)) AS time_in_rent F...
19:50:40
▶ SELECT c.id_car, date_part('hours', SUM(r.time_recieve - r.time_give)) AS time_in_rent F...
19:50:09
▶ SELECT * FROM model.car c1 WHERE NOT EXISTS (SELECT * FROM model.car c2 WHERE c2.mileag...
19:41:06
▶ SELECT c.id_car, c.price * (24 - COALESCE(date_part('hours', SUM(r.time_recieve - r.time...
19:27:34

5. Индексы и EXPLAIN

1. Запросы без индексов

Запрос №6

```
1 SELECT COUNT(*), m.name
2 FROM model.car c JOIN model.car_model m
3 ON c.id_model = m.id_model
4 GROUP BY m.name;
```

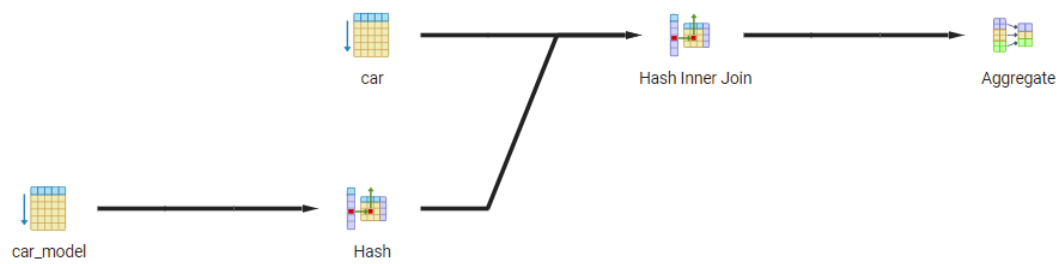
Время выполнения запроса

Successfully run. Total query runtime: 65 msec.
1 rows affected.

План запроса

	QUERY PLAN	
	text	
1	HashAggregate (cost=2.21..2.26 rows=5 width=154)	
2	[...] Group Key: m.name	
3	[...] -> Hash Join (cost=1.11..2.19 rows=5 width=146)	
4	[...] Hash Cond: (c.id_model = m.id_model)	
5	[...] -> Seq Scan on car c (cost=0.00..1.05 rows=5 width=4)	
6	[...] -> Hash (cost=1.05..1.05 rows=5 width=150)	
7	[...] -> Seq Scan on car_model m (cost=0.00..1.05 rows=5 width=150)	

Графическое представление



Запрос №4

```

1 EXPLAIN SELECT * FROM model.car c1
2 WHERE NOT EXISTS (
3     SELECT * FROM model.car c2
4     WHERE c2.mileage > c1.mileage)
  
```

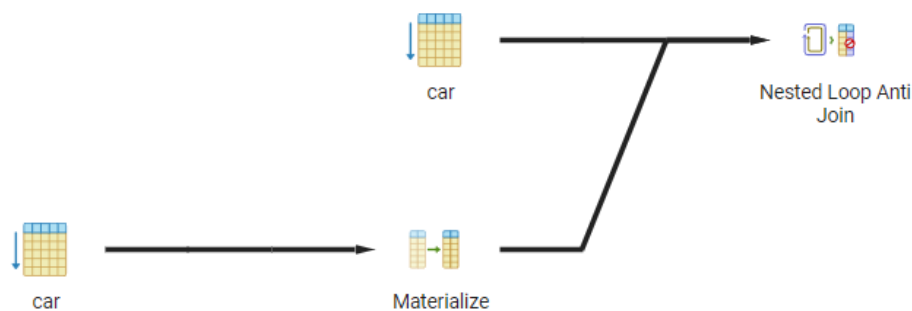
Время работы запроса

Successfully run. Total query runtime: 93 msec.
1 rows affected.

План запроса

	QUERY PLAN
	text
1	Nested Loop Anti Join (cost=0.00..2.39 rows=3 width=540)
2	[...] Join Filter: (c2.mileage > c1.mileage)
3	[...] -> Seq Scan on car c1 (cost=0.00..1.05 rows=5 width=540)
4	[...] -> Materialize (cost=0.00..1.07 rows=5 width=4)
5	[...] -> Seq Scan on car c2 (cost=0.00..1.05 rows=5 width=4)

Графическое представление



Создание простого индекса для второго запроса (запрос №4)

```
1 CREATE INDEX index1 ON model.car(id_car);
```

CREATE INDEX

Query returned successfully in 92 msec.

Проверка созданного индекса

car

> Columns (11)

> Constraints

✓ Indexes (5)

car_body_id_uindex

car_engine_id_uindex

car_id_model_uindex

car_registration_id_uindex

index1

План запроса

	QUERY PLAN	
	text	
1	Nested Loop Anti Join (cost=0.00..2.39 rows=3 width=540)	
2	[...] Join Filter: (c2.mileage > c1.mileage)	
3	[...] -> Seq Scan on car c1 (cost=0.00..1.05 rows=5 width=540)	
4	[...] -> Materialize (cost=0.00..1.07 rows=5 width=4)	
5	[...] -> Seq Scan on car c2 (cost=0.00..1.05 rows=5 width=4)	

Время выполнения

Successfully run. Total query runtime: 114 msec.
5 rows affected.

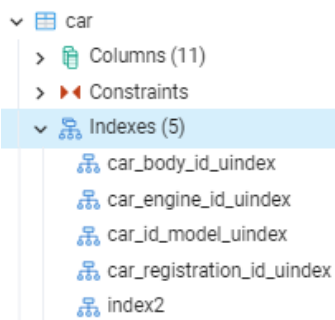
Создание составного индекса для второго запроса (запрос №6)

```
1 CREATE INDEX index2 ON model.car(id_car, id_model)
```

CREATE INDEX

Query returned successfully in 66 msec.

Проверка созданного индекса



План запроса

QUERY PLAN		text	
1	HashAggregate	(cost=2.21..2.26 rows=5 width=154)	
2	[...]	Group Key: m.name	
3	[...] -> Hash Join	(cost=1.11..2.19 rows=5 width=146)	
4	[...] Hash Cond:	(c.id_model = m.id_model)	
5	[...] -> Seq Scan on car c	(cost=0.00..1.05 rows=5 width=4)	
6	[...] -> Hash	(cost=1.05..1.05 rows=5 width=150)	
7	[...] -> Seq Scan on car_model m	(cost=0.00..1.05 rows=5 width=150)	

Время выполнения

```
Successfully run. Total query runtime: 85 msec.  
7 rows affected.
```

Удаление индексов

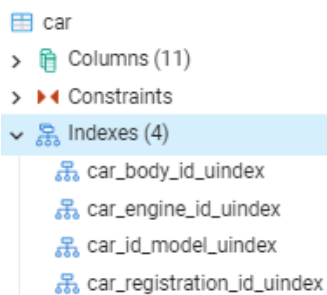
Удаление первого индекса

```
1 DROP INDEX model.index1
```

DROP INDEX

Query returned successfully in 164 msec.

Проверка удаления



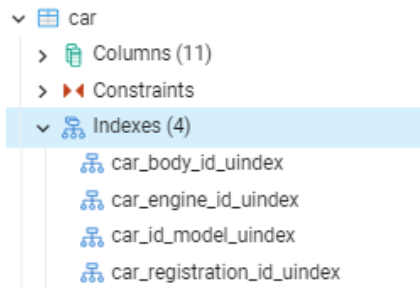
Удаление второго индекса

```
1 DROP INDEX model.index2
```

```
DROP INDEX
```

```
Query returned successfully in 62 msec.
```

Проверка удаления



Вывод: с помощью SQL запросов были показаны, изменены, удалены и добавлены данные в базу. Время выполнения запросов с использованием индексов стало дольше.