

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 2 курса ИКТ

группы

К32422

Бабичев

Леонид

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 11. БД «Автомастерская»

Описание предметной области: Сеть автомастерских осуществляет ремонт автомашин, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы. Заработная плата мастеров составляет 50% стоимости работы.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Должность. Разряд мастера. Специализация. Адрес автомастерской. Дата заказа. Гос. Номер автомобиля. Марка. Мощность автомобиля. Год выпуска. Цвет автомобиля. Дата принятия в ремонт. Плановая дата окончания ремонта. Фактическая дата окончания ремонта. Вид ремонта. Стоимость вида ремонта. Название детали. Цена детали. Марка и модель автомобиля. Страна производителя. Госномер автомобиля. ФИО владельца. Номер телефона владельца. E-mail владельца.

Схема базы данных (из ЛР 1):

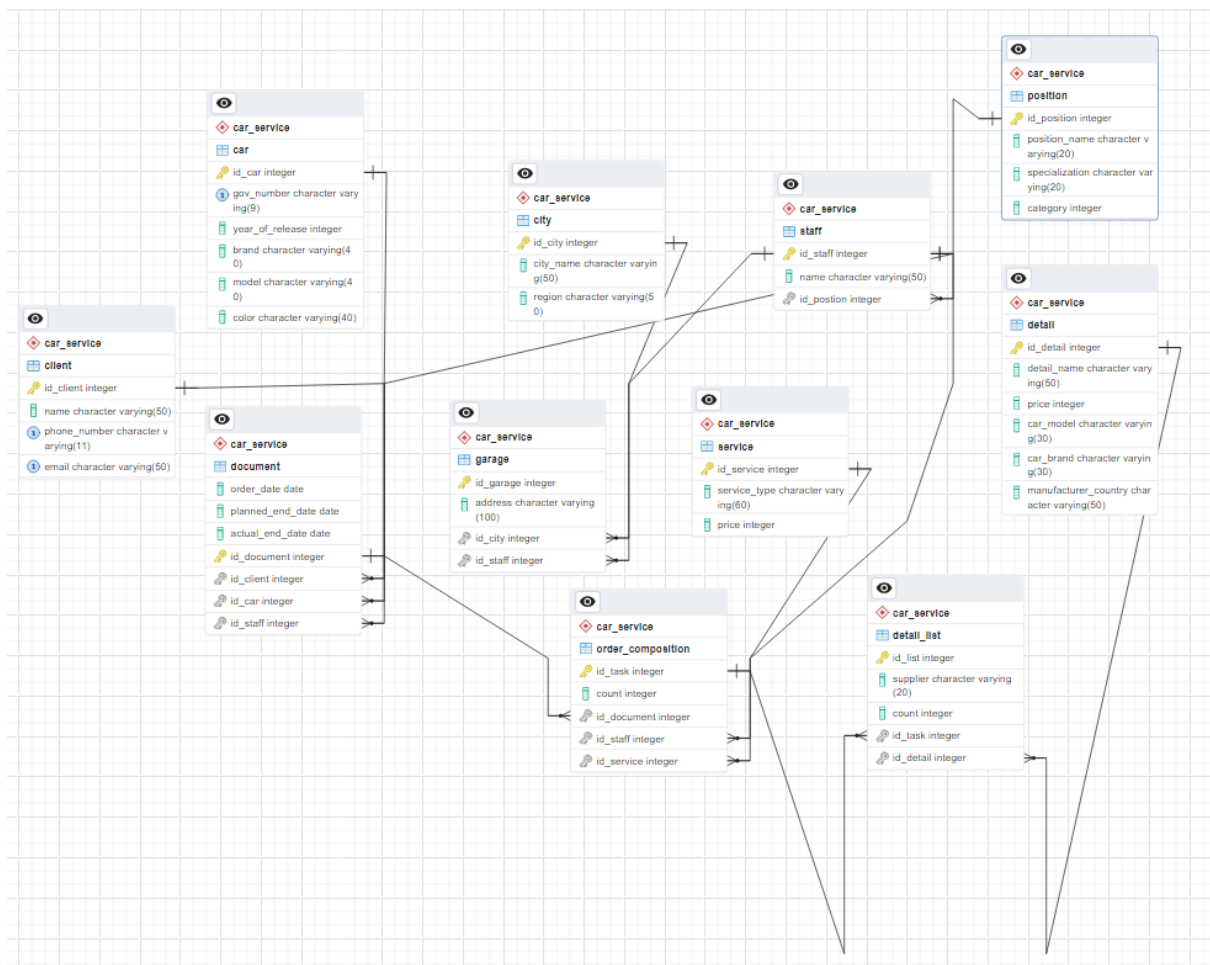


Рисунок 1. Схема базы данных.

Выполнение:

1. Запросы к базе данных. Выполнить запросы согласно индивидуальному заданию, часть 2. В отчете привести формулировку запроса, команду, скриншот выполнения запроса;

- Выбрать фамилию того механика, который чаще всех работает с автомобилями марки "Субару".

Query
Query History

```

1 SELECT *
2 FROM
3     (SELECT name,
4           count(*),
5           dense_rank() over(
6               ORDER BY count(*) DESC) rnk
7     FROM car_service.document d
8     JOIN car_service.car c ON d.id_car=c.id_car
9     JOIN car_service.staff s ON d.id_staff=s.id_staff
10    WHERE brand = 'subaru'
11    GROUP BY name) x
12 WHERE rnk = 1

```

Data Output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	name character varying (50) 🔒	count bigint 🔒	rnk bigint 🔒
1	Илья Волков	2	1
2	Олег Терехно	2	1

- Определить тех владельцев автомобилей, которых всегда обслуживает один и тот же механик. Вывести фамилии механика и его постоянного клиента.

Query
Query History

```

1 SELECT c.name,
2        s.name
3 FROM car_service.document d
4 JOIN car_service.client c ON d.id_client=c.id_client
5 JOIN car_service.staff s ON d.id_staff=s.id_staff
6 WHERE c.name in
7     (SELECT c.name
8     FROM car_service.document d
9     JOIN car_service.client c ON d.id_client=c.id_client
10    JOIN car_service.staff s ON d.id_staff=s.id_staff
11    GROUP BY c.name
12    HAVING count(DISTINCT s.name) = 1)

```

Data Output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	name character varying (50) 🔒	name character varying (50) 🔒
1	Егор Крид	Илья Волков

- Вывести фамилии механиков, которые не выполняли работы в срок и количество дней просрочки выполнения заказа.

Query









Query History

```
1 SELECT id_document,
2         name,
3         abs(planned_end_date - actual_end_date)
4 FROM car_service.document d
5 JOIN car_service.staff s ON d.id_staff=s.id_staff
6 WHERE planned_end_date < actual_end_date
```

Data Output

Messages

Notifications



	id_document integer	name character varying (50)	abs integer
1	102	Олег Терехно	1
2	104	Олег Терехно	2
3	105	Илья Волков	364
4	106	Максим Иванов	8
5	110	Илья Волков	10
6	101	Илья Волков	3

- Вывести данные механика, который выполнял все виды ремонта за прошедшую неделю.

<pre>1 WITH cte AS 2 (SELECT oc.id_staff 3 FROM car_service.order_composition oc 4 JOIN car_service.document d ON d.id_document=oc.id_document 5 JOIN car_service.service s ON s.id_service=oc.id_service 6 WHERE actual_end_date BETWEEN '2023-04-17' AND '2023-04-23' 7 GROUP BY oc.id_staff 8 HAVING count(DISTINCT service_type) = 3) 9 SELECT s.id_staff, 10 name 11 FROM cte 12 LEFT JOIN car_service.staff s ON cte.id_staff=s.id_staff</pre>			
Data Output		Messages	Notifications
<div></div>			
	id_staff [PK] integer	name character varying (50)	
1	1	Максим Иванов	

```
1 WITH cte AS
2   (SELECT oc.id_staff
3    FROM car_service.order_composition oc
4    JOIN car_service.document d ON d.id_document=oc.id_document
5    JOIN car_service.service s ON s.id_service=oc.id_service
6    WHERE actual_end_date >= date_trunc('week', current_date - interval '1 week')
7    AND actual_end_date < date_trunc('week', current_date)
8    GROUP BY oc.id_staff
9    HAVING count(DISTINCT service_type) = 3)
10 SELECT s.id_staff,
11        name
12 FROM cte
13 LEFT JOIN car_service.staff s ON cte.id_staff=s.id_staff
```

Сколько заработал каждый мастер за прошедший месяц?

Query Query History

```
1 WITH cte AS
2   (SELECT oc.id_staff,
3          sum(price) AS total
4   FROM car_service.order_composition oc
5   JOIN car_service.document d ON d.id_document=oc.id_document
6   JOIN car_service.service s ON s.id_service=oc.id_service
7   WHERE actual_end_date BETWEEN '2023-01-01' AND '2023-04-30'
8   GROUP BY oc.id_staff)
9 SELECT s.id_staff,
10        name,
11        total
12 FROM cte
13 LEFT JOIN car_service.staff s ON cte.id_staff=s.id_staff
```

Data Output Messages Notifications

	id_staff [PK] integer	name character varying (50)	total bigint
1	1	Максим Иванов	10500
2	2	Илья Волков	1000

Query Query History

```
1 WITH cte AS
2   (SELECT oc.id_staff,
3          sum(price) AS total
4   FROM car_service.order_composition oc
5   JOIN car_service.document d ON d.id_document=oc.id_document
6   JOIN car_service.service s ON s.id_service=oc.id_service
7   WHERE actual_end_date >= date_trunc('week', current_date - interval '1 week')
8         AND actual_end_date < date_trunc('week', current_date)
9   GROUP BY oc.id_staff)
10 SELECT s.id_staff,
11        name,
12        total
13 FROM cte
14 LEFT JOIN car_service.staff s ON cte.id_staff=s.id_staff
```

Вывести данные владельцев автомобилей, которые обращались в ремонт больше одного раза.

```
1 SELECT name
2 FROM car_service.document d
3 JOIN car_service.client s ON s.id_client=d.id_client
4 GROUP BY name
5 HAVING count(*)>1
```

Data Output		Messages	Notifications
	name character varying (50)		
1	Диана Бабичева		
2	Леонид Бабичев		
3	Максим Демидов		

- За каждый день просрочки выполнения заказа механику назначается штраф в размере 5%. Рассчитать штраф каждого механика за прошедший месяц.

```
1 WITH cte AS
2 (SELECT *
3 FROM car_service.document d
4 JOIN car_service.staff s ON d.id_staff=s.id_staff
5 JOIN car_service.order_composition oc ON d.id_document=oc.id_document
6 JOIN car_service.service sr ON sr.id_service=oc.id_service)
7 SELECT name,
8 (abs(planned_end_date-actual_end_date)*0.05*price) AS fine
9 FROM cte
10 WHERE planned_end_date < actual_end_date
11 AND actual_end_date BETWEEN '2023-04-17' AND '2023-04-23'
```

Data Output		Messages	Notifications
	name character varying (50)	fine numeric	
1	Илья Волков	150.00	

Query Query History

Execute/Refresh F5

```

1 WITH cte AS
2   (SELECT *
3    FROM car_service.document d
4    JOIN car_service.staff s ON d.id_staff=s.id_staff
5    JOIN car_service.order_composition oc ON d.id_document=oc.id_document
6    JOIN car_service.service sr ON sr.id_service=oc.id_service)
7   SELECT name,
8          (abs(planned_end_date-actual_end_date)*0.05*price) AS fine
9   FROM cte
10  WHERE planned_end_date < actual_end_date
11         AND actual_end_date >= date_trunc('week', current_date - interval '1 week')
12         AND actual_end_date < date_trunc('week', current_date)

```

Data Output Messages Notifications

≡+

📄

📋

🗑️

🔍

⬇️

📶

name

character varying (50)

fine

numeric

Представления. Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3. В отчете привести формулировку запроса, команду, скриншот выполнения запроса и просмотр содержимого представления;

- для заказчиков (фамилию механика и модель автомобиля, которую он ремонтирует чаще всего)

```

1 CREATE VIEW pop_car AS (SELECT *
2   FROM
3     (SELECT name,
4            brand,
5            count(*),
6            dense_rank() over(
7                                ORDER BY count(*) DESC) rnk
8   FROM car_service.document d
9   JOIN car_service.car c ON d.id_car=c.id_car
10  JOIN car_service.staff s ON d.id_staff=s.id_staff
11  GROUP BY name,
12           brand) x
13 WHERE rnk = 1)
14

```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 125 msec.

```
1 SELECT * FROM pop_car
```

Data Output Messages Notifications



	name character varying (50)	brand character varying (40)	count bigint	rnk bigint
1	Илья Волков	bmw	2	1
2	Илья Волков	subaru	2	1
3	Максим Иванов	bmw	2	1
4	Олег Терехно	subaru	2	1

- для менеджеров (рассчитать премию все механикам, которые за прошедший месяц все свои заказы выполнили своевременно - 10% от зарплаты).

Query

Query history









```
1 CREATE VIEW staff_bonus AS with cte as (SELECT *
2 FROM car_service.document d
3 JOIN car_service.staff s ON d.id_staff=s.id_staff
4 JOIN car_service.order_composition oc ON d.id_document=oc.id_document
5 JOIN car_service.service sr ON sr.id_service=oc.id_service)
6
7 select name, sum(price)*1.1 from cte
8 where planned_end_date > actual_end_date and actual_end_date BETWEEN '2023-04-17' AND '
9 group by name
```



```
1 SELECT * FROM staff_bonus
2
```

Data Output

Messages

Notifications



	name character varying (50) 	?column? numeric 
1	Максим Иванов	2750

- Запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;
 - INSERT с использованием подзапросов;

```

1 select * from car_service.detail
2
3 INSERT INTO car_service.detail (id_detail, detail_name, price, car_model, car_brand, ma
4 SELECT 5, 'Руль', 5000, 'X5', 'BMW', 'Германия'
5 WHERE NOT EXISTS (
6     SELECT 1 FROM car_service.detail WHERE id_detail = 5
7 );

```

Data Output Messages Notifications

	id_detail [PK] integer	detail_name character varying (50)	price integer	car_model character varying (30)	car_brand character varying (30)	manufacturer_country character varying (50)
1	4	Фара	15000	V90 Cross Country	Volvo	Швеция
2	3	Тормозная колодка	50000	Valkyrie	Aston Martin	Великобритания
3	1	Аккумулятор	31500	Impreza Sport	Subaru	Япония
4	5	Руль	5000	X5	BMW	Германия

– UPDATE с использованием подзапросов;

```

1 UPDATE car_service.document
2 SET id_staff = 2
3 WHERE id_document not in
4     (SELECT id_document
5      FROM car_service.order_composition
6      WHERE COUNT>1)
7
8

```

Data Output Messages Notifications

UPDATE 11

Query returned successfully in 379 msec.

– DELETE с использованием подзапросов;

```

1 DELETE FROM car_service.document
2 WHERE id_client IN (
3     SELECT c.id_client
4     FROM car_service.client c
5     JOIN car_service.document d ON c.id_client = d.id_client
6     WHERE EXTRACT('month' FROM d.order_date) = 4
7     AND d.id_car IN (
8         SELECT id_car
9         FROM car_service.car
10        WHERE year_of_release > 2014
11    )
12 );
13
14

```

4. План запросов

Без индекса План запроса без индекса. Результат представлен на рисунке

```
1 EXPLAIN SELECT c.name,
2             c.email
3 FROM car_service.document d
4 JOIN car_service.client c ON d.id_client=c.id_client
5 JOIN car_service.staff s ON d.id_staff=s.id_staff
6 WHERE c.name in
7       (SELECT c.name
8        FROM car_service.document d
9         JOIN car_service.client c ON d.id_client=c.id_client
10        JOIN car_service.staff s ON d.id_staff=s.id_staff
11         GROUP BY c.name
12        HAVING count(DISTINCT s.name) = 1)
```

Data Output Messages Explain × Notifications

QUERY PLAN

1	Nested Loop (cost=19.79..28.62 rows=1 width=236)
2	-> Nested Loop (cost=19.64..21.76 rows=1 width=240)
3	Join Filter: (c.id_client = d.id_client)
4	-> Hash Join (cost=19.64..20.69 rows=1 width=240)
5	Hash Cond: ((c.name)::text = (c_1.name)::text)
6	-> Seq Scan on client c (cost=0.00..1.03 rows=3 width=240)
7	-> Hash (cost=19.63..19.63 rows=1 width=118)
8	-> GroupAggregate (cost=19.56..19.62 rows=1 width=118)
9	Group Key: c_1.name

Total rows: 24 of 24 Query complete 00:00:00.071 Ln 12, Col 40

✓ Successfully run. Total query runtime: 71 msec. 24 rows affected. ✕

Создание мульти индекса

```
1 CREATE INDEX multiple ON car_service.client(name,email)
```

Результат с индексом

```

1  EXPLAIN SELECT c.name,
2      c.email
3  FROM car_service.document d
4  JOIN car_service.client c ON d.id_client=c.id_client
5  JOIN car_service.staff s ON d.id_staff=s.id_staff
6  WHERE c.name in
7      (SELECT c.name
8      FROM car_service.document d
9      JOIN car_service.client c ON d.id_client=c.id_client
10     JOIN car_service.staff s ON d.id_staff=s.id_staff
11     GROUP BY c.name
12     HAVING count(DISTINCT s.name) = 1)

```

Data Output Messages Explain × Notifications

QUERY PLAN text

1	Nested Loop (cost=19.79..28.62 rows=1 width=236)
2	-> Nested Loop (cost=19.64..21.76 rows=1 width=240)
3	Join Filter: (c.id_client = d.id_client)
4	-> Hash Join (cost=19.64..20.69 rows=1 width=240)
5	Hash Cond: ((c.name)::text = (c_1.name)::text)
6	-> Seq Scan on client c (cost=0.00..1.03 rows=3 width=240)
7	-> Hash (cost=19.63..19.63 rows=1 width=118)
8	-> GroupAggregate (cost=19.56..19.62 rows=1 width=118)
9	Group Key: c_1.name

✓ Successfully run. Total query runtime: 75 msec. 24 rows affected. ✕

Total rows: 24 of 24 Query complete 00:00:00.075 Ln 12, Col 40

Без индекса

Query Query History Execute/Refresh (F5) Scratch Pad ×

```

1  EXPLAIN SELECT name
2  FROM car_service.document d
3  JOIN car_service.client s ON s.id_client=d.id_client
4  GROUP BY name
5  HAVING count(*)>1

```

Data Output Messages Explain × Notifications

QUERY PLAN text

1	HashAggregate (cost=2.13..2.17 rows=1 width=118)
2	Group Key: s.name
3	Filter: (count(*) > 1)
4	-> Hash Join (cost=1.07..2.12 rows=3 width=118)
5	Hash Cond: (d.id_client = s.id_client)
6	-> Seq Scan on document d (cost=0.00..1.03 rows=3 width=4)
7	-> Hash (cost=1.03..1.03 rows=3 width=122)
8	-> Seq Scan on client s (cost=0.00..1.03 rows=3 width=122)

✓ Successfully run. Total query runtime: 124 msec. 8 rows affected. ✕

Total rows: 8 of 8 Query complete 00:00:00.124 Ln 5, Col 18

Создание индекса

```

1  CREATE INDEX name ON car_service.client(name)

```

С индексом

The screenshot shows a database query interface. At the top, there's a 'Query' tab with a text area containing an SQL query. To the right of the query is an 'Execute/Refresh' button with an 'F5' icon. Below the query is a 'Data Output' tab, which is currently displaying the 'EXPLAIN' plan of the query. The plan is a table with 8 rows. To the right of the plan is a 'Scratch Pad' tab. At the bottom, there's a status bar showing 'Total rows: 8 of 8', 'Query complete 00:00:00.092', and a green success message: 'Successfully run. Total query runtime: 92 msec. 8 rows affected.'

```
1 EXPLAIN SELECT name
2 FROM car_service.document d
3 JOIN car_service.client s ON s.id_client=d.id_client
4 GROUP BY name
5 HAVING count(*)>1
```

	QUERY PLAN
1	HashAggregate (cost=2.13..2.17 rows=1 width=118)
2	Group Key: s.name
3	Filter: (count(*) > 1)
4	-> Hash Join (cost=1.07..2.12 rows=3 width=118)
5	Hash Cond: (d.id_client = s.id_client)
6	-> Seq Scan on document d (cost=0.00..1.03 rows=3 width=4)
7	-> Hash (cost=1.03..1.03 rows=3 width=122)
8	-> Seq Scan on client s (cost=0.00..1.03 rows=3 width=122)

✓ Successfully run. Total query runtime: 92 msec. 8 rows affected. ✕

Total rows: 8 of 8 Query complete 00:00:00.092 Ln 5, Col 18

Выводы

Во время этой лабораторной работы я:

1. Научился добавлять, изменять и удалять данные в таблицах.
2. Научился писать различные запросы.
3. Научился создавать представления.
4. Научился создавать индексы.
5. Научился просматривать план запроса и графическую модель объяснения.
6. Сделал вывод о том, что выполнение запросов с индексами и без них практически не отличаются по времени. Это связано с небольшим количеством данных.