

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО»**

**Факультет инфокоммуникационных технологий**

**Дисциплина:**

**«Проектирование и реализация баз данных»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**«Запросы на выборку и модификацию данных,  
представления и индексы в PostgreSQL»**

**Выполнил:**

студент группы K33391

Черкес Артур Викторович

---

(подпись)

**Проверил(а):**

Говорова Марина Михайловна

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург 2023

г.

**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, 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.

### **Вариант 13. БД «Ресторан»**

Описание предметной области: Необходимо создать систему для обслуживания заказов клиентов в ресторане.

Сотрудники ресторана – повара и официанты.

За каждым официантом закреплены определенные столы за смену. Клиенты могут бронировать столы заранее.

Каждый повар может готовить определенный набор блюд.

Официант принимает заказ от стола и передает его на кухню. Шеф-повар распределяет блюда для приготовления между поварами. В одном заказе может быть несколько одинаковых или разных блюд.

Запас продуктов на складе не должен быть ниже заданного значения.

Цена заказа складывается из стоимости ингредиентов и наценки, которая составляет 40% стоимости ингредиентов.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Паспортные данные сотрудника. Категория сотрудника. Должность сотрудника. Оклад сотрудника. Наименование ингредиента. Код ингредиента. Дата закупки. Объем закупки. Количество продукта на складе. Необходимый запас продукта. Срок годности. Цена ингредиента. Калорийность (на 100г продукта). Поставщик. Наименование

блюда. Код блюда. Объем ингредиента. Номер стола. Дата заказа. Код заказа. Количество. Название блюда. Ингредиенты, входящие в блюдо. Тип ингредиента.

### Выполнение:

### Запросы:

- Посчитать кол-во ингредиентов в блюде.

```
1 SELECT d.name AS dish_name, COUNT(c.ingredients_id) AS ingredient_count
2 FROM restaurant.dishes d
3 LEFT JOIN restaurant.composion c ON d.id = c.dishes_id
4 GROUP BY d.name
5 ORDER BY d.name;
```

Data Output Messages Notifications

	dish_name character varying (60)	ingredient_count bigint
1	Блюдо 1	1
2	Блюдо 2	1
3	Блюдо 3	3

```
SELECT    d.name    AS    dish_name,    COUNT(c.ingredients_id)    AS
ingredient_count
FROM restaurant.dishes d
LEFT JOIN restaurant.composion c ON d.id = c.dishes_id
GROUP BY d.name
ORDER BY d.name;
```

Узнать, какой повар готовит больше всего блюд

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

WITH ChefDishCounts AS (

SELECT

s.full\_name AS cook,

COUNT(\*) AS d\_amount

FROM

restaurant.staff AS s

JOIN

restaurant.skills AS sk ON s.personnel\_id = sk.stuff\_key

GROUP BY

s.full\_name

)

SELECT

cook,

d\_amount

FROM

ChefDishCounts

WHERE

d\_amount = (

SELECT

MAX(d\_amount)

FROM

ChefDishCounts

);

Data Output

Messages

Notifications

+

📄

▼

📄

▼

🗑️

📄

📄

📄

📄

📄

cook

character varying (80)

d\_amount

bigint

1

Мария Сидорова

2

Какой ингредиент содержится в максимальном количестве блюд

Query

Query History

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

SELECT

i.name AS ingredient\_name,

COUNT(c.ingredients\_id) AS dish\_count

FROM

restaurant.ingredients AS i

JOIN

restaurant.composion AS c

ON i.id = c.ingredients\_id

GROUP BY

i.name

)

SELECT

ingredient\_name,

dish\_count

FROM

IngredientDishCounts

WHERE

dish\_count = (

SELECT

MAX(dish\_count)

FROM

IngredientDishCounts

);

Data Output

Messages

Notifications

+

📄

▼

📄

▼

🗑️

📄

📄

📄

📄

📄

ingredient\_name

character varying (60)

dish\_count

bigint

1

Оленина

3

Количество блюд, заказанных в какой-то день:

```
1 SELECT
2     d.name AS dish_name,
3     COUNT(*) AS dish_count
4 FROM
5     restaurant.orders AS o
6 JOIN
7     restaurant.orders_precessings AS op
8     ON o.id = op.orders_id
9 JOIN
10    restaurant.dishes AS d
11    ON op.dishes_id = d.id
12 WHERE
13     o.date = '2023-10-02'
14 GROUP BY
15     d.name;
```

Data Output Messages Notifications



	ingredient_name character varying (60)	dish_count bigint
1	Оленина	3

Какой повар приготовил больше всего блюд за смену

QueryQuery History

1WITH CookDishCounts AS (  
2SELECT  
3d.staff\_personnel\_id AS cook\_id,  
4s.full\_name AS cook\_name,  
5COUNT(\*) AS d\_amount  
6FROM  
7restaurant.dishes AS d  
8JOIN  
9restaurant.staff AS s  
10ON d.staff\_personnel\_id = s.personnel\_id  
11JOIN  
12restaurant.orders\_precessings AS op  
13ON op.dishes\_id = d.id  
14JOIN  
15restaurant.orders AS o  
16ON o.id = op.orders\_id  
17JOIN  
18restaurant.work\_schedule AS ws  
19ON ws.staff\_personnel\_id = d.staff\_personnel\_id  
20WHERE  
21ws.work\_status = 'Рабочий'  
22GROUP BY  
23d.staff\_personnel\_id, s.full\_name  
24)

25SELECT  
26cook\_id,  
27cook\_name,  
28d\_amount  
29FROM  
30CookDishCounts  
31WHERE  
32d\_amount = (SELECT MAX(d\_amount) FROM CookDishCounts);  
33

Data OutputMessagesNotifications

cook\_id

bigint

cook\_name

character varying (80)

d\_amount

bigint

1	1	Иван Иванов	1
2	2	Петр Петров	1

Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа).

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

SELECT

s.full\_name AS waiter,

SUM(d.price \* op.dishes\_amount \* 0.05) AS bonus

FROM

restaurant.staff AS s

JOIN

restaurant.orders AS o

ON s.personnel\_id = o.staff\_personnel\_id

JOIN

restaurant.orders\_precessings AS op

ON o.id = op.orders\_id

JOIN

restaurant.dishes AS d

ON op.dishes\_id = d.id

WHERE

o.date >= CURRENT\_DATE - INTERVAL '10 days'

GROUP BY

s.full\_name;

Data Output

Messages

Notifications

waiter

character varying (80)

bonus

numeric

1

Мария Сидорова

55.00

Сколько закреплено столов за каждым из официантов?

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

SELECT

s.full\_name AS waiter,

COUNT(t.id) AS t\_amount

FROM

restaurant.tables AS t

INNER JOIN

restaurant.staff AS s

ON t.staff\_personnel\_id = s.personnel\_id

INNER JOIN

restaurant.positions AS p

ON s.positions\_id = p.id

WHERE

p.name = 'Официант'

GROUP BY

s.full\_name;

Data Output

Messages

Notifications

waiter

character varying (80)

t\_amount

bigint

1

Мария Сидорова

2

2

Новый Официант

1

### Представления:

**Список ингредиентов с калорийностью более 150 калорий на 100 грамм.**

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

CREATE VIEW high\_calorie\_ingredients AS

SELECT

id,

product\_type,

name,

calories\_100g

FROM

restaurant.ingredients

WHERE

calories\_100g > 150;

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 43 msec.

Query

Query History

1

2

3

SELECT \* FROM high\_calorie\_ingredients;

Data Output

Messages

Notifications

+

📄

▼

📄

▼

🗑️

📄

📄

📄

📄

id

bigint

🔒

product\_type

character varying (30)

🔒

name

character varying (60)

🔒

calories\_100g

bigint

🔒

1

5

Рыба

Окунь

200



Представление с информацией о сотрудниках и их графике работы:

```
CREATE VIEW staff_work_schedule AS
SELECT
    s.full_name AS staff_name,
    ws.work_status AS work_status,
    ss.beginning AS shift_start,
    ss.ending AS shift_end
FROM
    staff s
JOIN
    work_schedule ws ON s.personnel_id = ws.staff_personnel_id
JOIN
    shifts_schedule ss ON ws.shift_id = ss.id;
```

Messages  
Query returned successfully in 47 msec.

Query Query History

1 SELECT \* FROM staff\_work\_schedule;

2 |

Data Output Messages Notifications

	staff_name character varying (80)	work_status character varying (20)	shift_start timestamp without time zone	shift_end timestamp without time zone
1	Иван Иванов	Рабочий	2023-10-02 00:00:00	2023-10-02 11:59:59
2	Петр Петров	Рабочий	2023-10-02 12:00:00	2023-10-02 23:59:59
3	Мария Сидорова	Отсутствует	2023-10-02 12:00:00	2023-10-02 23:59:59

Модификация данных

```
1 UPDATE restaurant.ingredients AS i
2 SET calories_100g = (
3     SELECT AVG(calories_100g)
4     FROM restaurant.ingredients
5     WHERE product_type = i.product_type
6 )
7 WHERE id = 4;
```

Data Output Messages Notifications

UPDATE 1

Query Query History

```
1 UPDATE restaurant.orders
2 SET date = '2023-10-05', status = 'Изменен', payment_state = 'Не оплачен'
3 WHERE id = (SELECT id FROM restaurant.orders WHERE payment_state = 'Не оплачен' LIMIT 1);
```

Data Output Messages Notifications

UPDATE 1

Удаление заказа, еду для которого приготовил повар с самой высокой ставкой

[Copy](#)[Copy to Query Editor](#)

```
DELETE FROM restaurant.orders_precessings
WHERE orders_id IN (
    SELECT id
    FROM restaurant.orders
    WHERE staff_personnel_id = (
        SELECT personnel_id
        FROM restaurant.staff
        WHERE positions_id = (
            SELECT positions_id
            FROM restaurant.positions
            ORDER BY working_rates DESC
        LIMIT 1)
    LIMIT 1)
LIMIT 1));
```

#### Messages

Query returned successfully in 67 msec.

# Создание индексов

Без индекса:

QueryQuery History

12

SELECT \* FROM restaurant.dishes WHERE name = 'Блюдо 2';

Data OutputMessagesExplain xNotifications

	id [PK] bigint	staff_personnel_id bigint	recipe character varying (32000)	name character varying (60)	type character varying (30)
1	11	2	Рецепт блюда 2	Блюдо 2	Второе

CopyCopy to Query Editor

SELECT \* FROM restaurant.dishes WHERE name = 'Блюдо 2';

## Messages

Successfully run. Total query runtime: 60 msec. 1 rows affected.

Query

Query History

1

SELECT \* FROM restaurant.dishes WHERE name = 'Блюдо 2';

2

Data Output

Messages

Explain 

×

Notifications

Graphical

Analysis

Statistics

dishes

Query

Query History

1

SELECT \* FROM restaurant.dishes WHERE name = 'Блюдо 2';

2

|

Data Output

Messages

Explain 

×

Notifications

Graphical

Analysis

Statistics

#	Node
1.	→ Seq Scan on dishes as dishes Filter: ((name)::text = 'Блюдо 2')::text

Создание индекса:

Copy Copy to Query Editor

```
CREATE INDEX idx_dishes_name ON restaurant.dishes (name);
```

Messages

Query returned successfully in 39 msec.

Поиск с индексом:

```
SELECT *
FROM restaurant.dishes
WHERE name = 'Блюдо 2';
```

Messages

Successfully run. Total query runtime: 53 msec. 1 rows affected.

Оказался быстрее на 7 мс

Без индекса:

Query Query History

```
1 SELECT
2   d.staff_personnel_id AS повар_id,
3   s.full_name AS повар_имя,
4   COUNT(*) AS количество_блюд
5 FROM
6   restaurant.dishes AS d
7 JOIN
8   restaurant.staff AS s
9   ON d.staff_personnel_id = s.personnel_id
10 JOIN
11   restaurant.orders_precessings AS op
12   ON op.dishes_id = d.id
13 JOIN
14   restaurant.orders AS o
15   ON o.id = op.orders_id
```

Data Output Messages Explain x Notifications

	повар_id bigint	повар_имя character varying (80)	количество_блюд bigint
1	1	Иван Иванов	1



Query Query History

1

CREATE INDEX idx\_optimized\_join

2

ON restaurant.orders\_precessings (dishes\_id, orders\_id);

Data Output Messages Explain × Notifications

CREATE INDEX

Query returned successfully in 47 msec.

Поиск и индексом:

Query Query History

6

restaurant.dishes AS d

7

JOIN

8

restaurant.staff AS s

9

ON d.staff\_personnel\_id = s.personnel\_id

10

JOIN

11

restaurant.orders\_precessings AS op

12

ON op.dishes\_id = d.id

13

JOIN

14

restaurant.orders AS o

15

ON o.id = op.orders\_id

16

JOIN

17

restaurant.work\_schedule AS ws

18

ON ws.staff\_personnel\_id = d.staff\_personnel\_id

19

WHERE

20

ws.work\_status = 'Рабочий'

21

GROUP BY

22

d.staff\_personnel\_id, s.full\_name

23

ORDER BY

24

количество\_блюд DESC

25

LIMIT 1;

Data Output Messages Explain × Notifications

Successfully run. Total query runtime: 45 msec.  
1 rows affected.

Быстрее на 6 мс



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

### **Выводы**

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.