

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет

по лабораторной работе «Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»
по дисциплине «**Проектирование и реализация баз данных**»
Вариант 13. БД «Ресторан»

Автор: Нестеренко Ю. А.

Факультет: ИКТ

Группа: К32422

Преподаватель: Говорова М. М.

Дата: 26.03.2023



Санкт-Петербург 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Описание предметной области	4
2 Схема логической модели базы данных, сгенерированная в Generate ERD	5
3 Запросы к базе данных	6
4 Представления.....	12
5 Запросы на модификацию данных	14
6 Создание индексов.....	22
ВЫВОДЫ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

ВВЕДЕНИЕ

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.

1 Описание предметной области

Вариант 13. БД «Ресторан»

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.

2 Схема логической модели базы данных, сгенерированная в Generate ERD

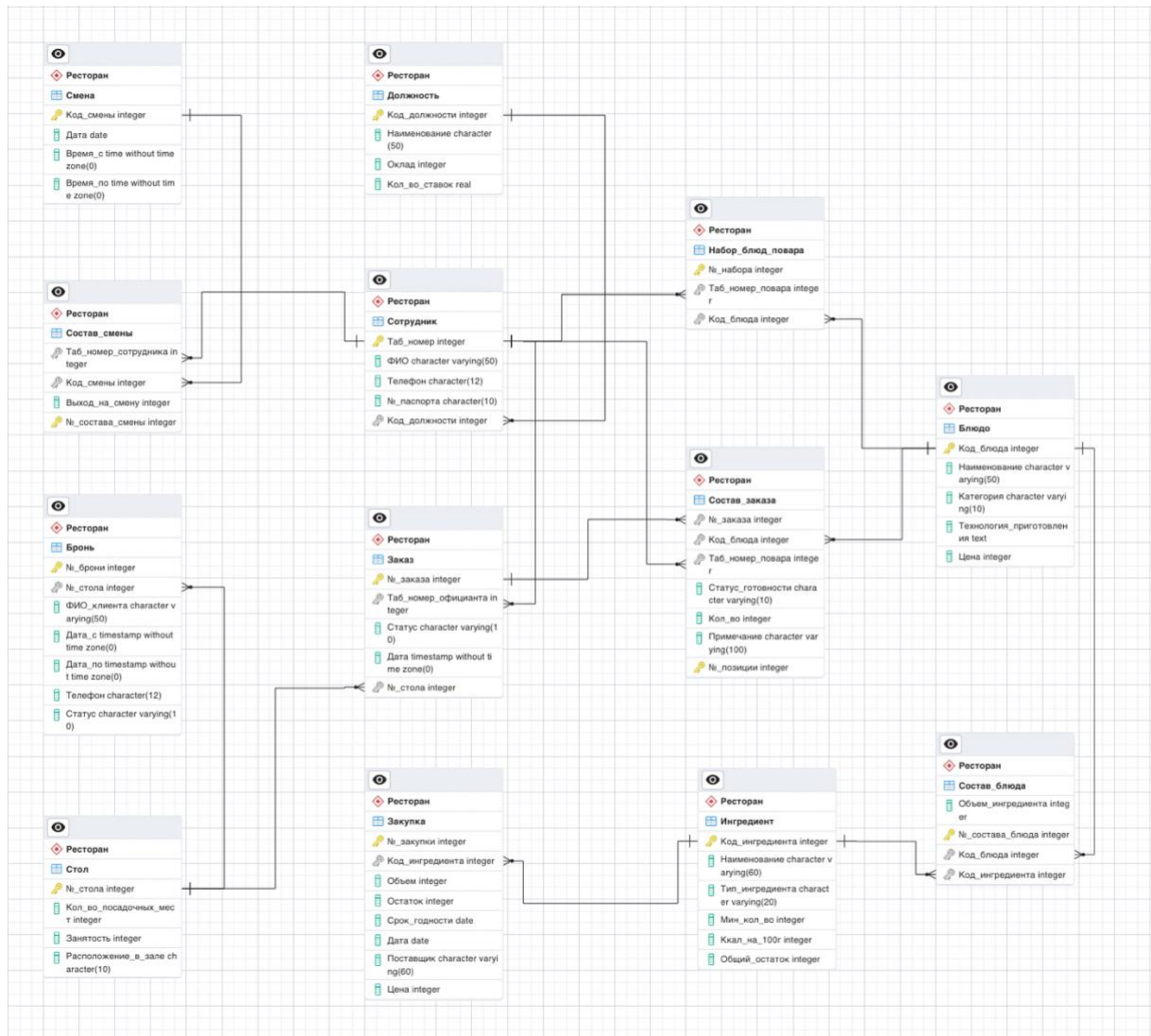


Рисунок 1 – Схема ЛМ БД в Generate ERD

3 Запросы к базе данных

1. Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.

```
SELECT * FROM "Ресторан"."Сотрудник" WHERE Таб_номер IN
(
    SELECT "Таб_номер_официанта" FROM "Ресторан"."Заказ" WHERE
    "Дата" BETWEEN NOW() - interval'1 month' AND NOW()
    GROUP BY "Таб_номер_официанта" HAVING
        SUM((SELECT SUM("Кол_во" * (SELECT "Цена" FROM
"Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" =
"Состав_заказа"."Код_блюда"))
FROM "Ресторан"."Состав_заказа" WHERE
"Состав_заказа"."№_заказа" = "Заказ"."№_заказа")) =
        (SELECT MAX(summary) FROM (SELECT SUM((SELECT
SUM("Кол_во" * (SELECT "Цена" FROM "Ресторан"."Блюдо"
WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда"))
FROM "Ресторан"."Состав_заказа" WHERE
"Состав_заказа"."№_заказа" = "Заказ"."№_заказа")) summary
FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() -
interval'1 month' AND NOW()
        GROUP BY "Таб_номер_официанта") AS sum_table)
)
```

```
17 SELECT * FROM "Ресторан"."Сотрудник" WHERE Таб_номер IN
18 (
19     SELECT "Таб_номер_официанта" FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() - interval'1 month' AND NOW()
20     GROUP BY "Таб_номер_официанта" HAVING
21         SUM((SELECT SUM("Кол_во" * (SELECT "Цена" FROM "Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда")))
22         FROM "Ресторан"."Состав_заказа" WHERE "Состав_заказа"."№_заказа" = "Заказ"."№_заказа")) =
23         (SELECT MAX(summary) FROM (SELECT SUM((SELECT SUM("Кол_во" * (SELECT "Цена" FROM "Ресторан"."Блюдо"
24             WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда")))
25             FROM "Ресторан"."Состав_заказа" WHERE "Состав_заказа"."№_заказа" = "Заказ"."№_заказа")) summary
26             FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() - interval'1 month' AND NOW()
27             GROUP BY "Таб_номер_официанта") AS sum_table)
28 )
29 
```

Data Output

Messages

Notifications

	Таб_номер [PK] integer	ФИО character varying (50)	Телефон character (12)	№_паспорта character (10)	Код_должности integer
1	667843	Зюсин Константин Никитич	+78443657238	4306403023	204

Total rows: 1 of 1

Query complete 00:00:00.091

Рисунок 2 – Выполнение запроса SELECT 1

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

```

SELECT "Таб_номер_официанта", 0.05 * SUM(
    (SELECT SUM("Кол_во" * (SELECT "Цена" FROM "Ресторан"."Блюдо"
    WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда")))
    FROM "Ресторан"."Состав_заказа" WHERE "Состав_заказа"."№_заказа"
    = "Заказ"."№_заказа")) "Премия"
FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() - interval'10 days'
AND NOW() GROUP BY "Таб_номер_официанта"

```

```

18 SELECT "Таб_номер_официанта", 0.05 * SUM(
19     (SELECT SUM("Кол_во" * (SELECT "Цена" FROM "Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда")))
20     FROM "Ресторан"."Состав_заказа" WHERE "Состав_заказа"."№_заказа" = "Заказ"."№_заказа")) "Премия"
21 FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() - interval'10 days' AND NOW() GROUP BY "Таб_номер_официанта"
22
23

```

Data Output

	Таб_номер_официанта integer	Премия numeric
1	289731	86.50
2	564738	86.50
3	576482	20.00
4	667843	359.00
5	673762	65.50

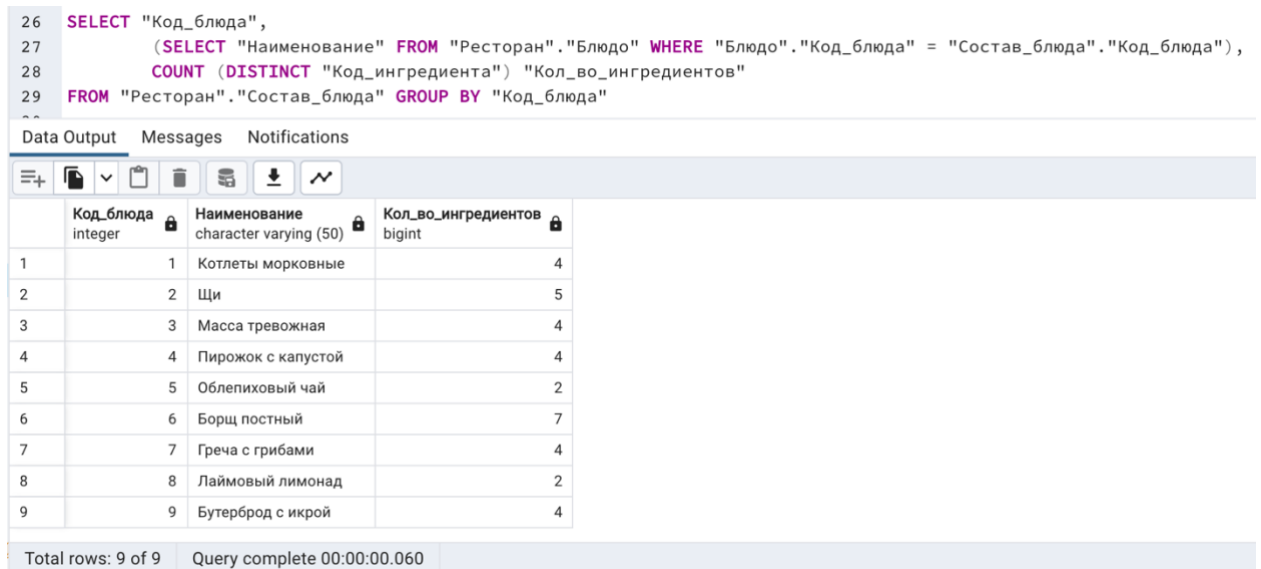
Total rows: 5 of 5

Query complete 00:00:00.053

Рисунок 3 – Выполнение запроса SELECT 2

3. Подсчитать, сколько ингредиентов содержит каждое блюдо.

```
SELECT "Код_блюда",  
      (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE  
"Блюдо"."Код_блюда" = "Состав_блюда"."Код_блюда"),  
      COUNT (DISTINCT "Код_ингредиента") "Кол_во_ингредиентов"  
FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда"
```



The screenshot shows a database query execution interface. At the top, the SQL query is displayed with line numbers 26 to 29. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 9 rows and 4 columns. The columns are 'Код_блюда' (integer), 'Наименование' (character varying (50)), 'Кол_во_ингредиентов' (bigint), and an unnamed column. The data is as follows:

	Код_блюда integer	Наименование character varying (50)	Кол_во_ингредиентов bigint
1	1	Котлеты морковные	4
2	2	Щи	5
3	3	Масса тревожная	4
4	4	Пирожок с капустой	4
5	5	Облепиховый чай	2
6	6	Борщ постный	7
7	7	Греча с грибами	4
8	8	Лаймовый лимонад	2
9	9	Бутерброд с икрой	4

At the bottom, it says 'Total rows: 9 of 9' and 'Query complete 00:00:00.060'.

Рисунок 4 – Выполнение запроса SELECT 3

4. Вывести название блюда, содержащее максимальное число ингредиентов.

```
SELECT (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE  
"Блюдо"."Код_блюда" = "Состав_блюда"."Код_блюда")  
FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда"  
HAVING COUNT (DISTINCT "Код_ингредиента") =  
      (SELECT MAX("Кол_во_ингредиентов") FROM (SELECT COUNT  
(DISTINCT "Код_ингредиента") "Кол_во_ингредиентов"  
FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда") AS  
max_ingredients)
```


34	SELECT (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" = "Состав_блюда"."Код_блюда")
35	FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда"
36	HAVING COUNT (DISTINCT "Код_ингредиента") =
37	(SELECT MAX("Кол_во_ингредиентов") FROM (SELECT COUNT (DISTINCT "Код_ингредиента") "Кол_во_ингредиентов"
38	FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда") AS max_ingredients)

Data Output	Messages	Notifications
<div> <div>Наименование</div> <div>character varying (50)</div> </div>		
1	Борщ постный	

Total rows: 1 of 1	Query complete 00:00:00.045
--------------------	-----------------------------

Рисунок 5 – Выполнение запроса SELECT 4

5. Какой повар может приготовить максимальное число видов блюд?

SELECT "Таб_номер_повара",

(SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE

"Сотрудник"."Таб_номер" = "Набор_блюд_повара"."Таб_номер_повара")

FROM "Ресторан"."Набор_блюд_повара" GROUP BY "Таб_номер_повара"

HAVING COUNT (DISTINCT "Код_блюда") =

(SELECT MAX("Кол_во_блюд") FROM (SELECT COUNT (DISTINCT

"Код_блюда") "Кол_во_блюд"

FROM "Ресторан"."Набор_блюд_повара" GROUP BY

"Таб_номер_повара") AS max_dishes)

43	SELECT "Таб_номер_повара",
44	(SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE "Сотрудник"."Таб_номер" = "Набор_блюд_повара"."Таб_номер_повара")
45	FROM "Ресторан"."Набор_блюд_повара" GROUP BY "Таб_номер_повара"
46	HAVING COUNT (DISTINCT "Код_блюда") =
47	(SELECT MAX("Кол_во_блюд") FROM (SELECT COUNT (DISTINCT "Код_блюда") "Кол_во_блюд"
48	FROM "Ресторан"."Набор_блюд_повара" GROUP BY "Таб_номер_повара") AS max_dishes)

Data Output	Messages	Notifications
<div> <div>Таб_номер_повара</div> <div>integer</div> </div> <div> <div>ФИО</div> <div>character varying (50)</div> </div>		
1	443111	Жирков Леонид Дмитриевич
2	565647	Мясоедов Илья Петрович

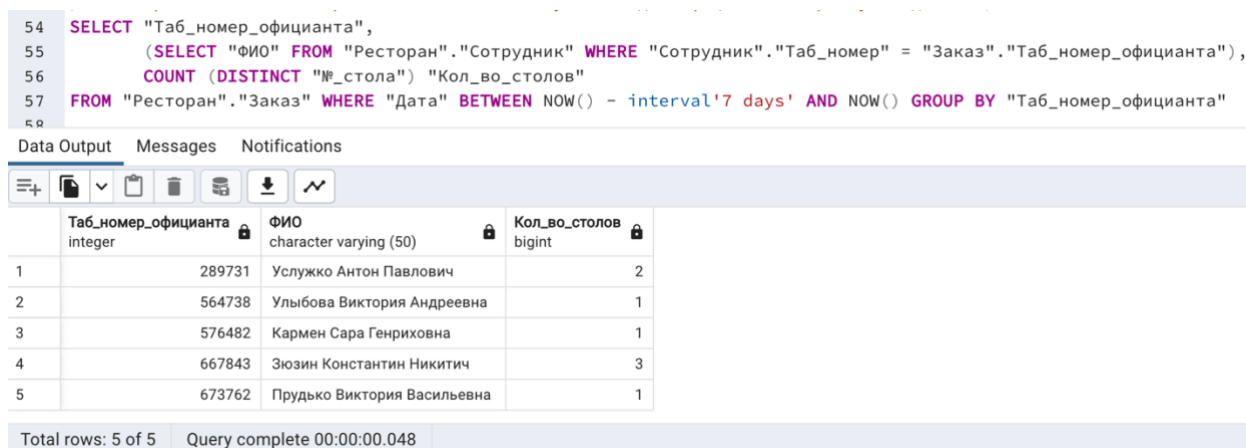
Total rows: 2 of 2	Query complete 00:00:00.156
--------------------	-----------------------------

Рисунок 6 – Выполнение запроса SELECT 5

6. Сколько закреплено столов за каждым из официантов?

Так как при работе над прошлой лабораторной работой было получено указание избавиться от закрепления столов за официантами при реализации БД, то предлагается альтернативный запрос: «сколько различных столов обслужил каждый официант за прошлую неделю?»

```
SELECT "Таб_номер_официанта",  
       (SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE  
        "Сотрудник"."Таб_номер" = "Заказ"."Таб_номер_официанта"),  
       COUNT (DISTINCT "№_стола") "Кол_во_столов"  
FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN NOW() - interval'7 days'  
AND NOW() GROUP BY "Таб_номер_официанта"
```



The screenshot shows a database query execution interface. At the top, the SQL query is displayed with line numbers 54 to 57. Below the query, there is a toolbar with icons for various actions. The main area displays the results of the query in a table with 5 rows and 3 columns. The columns are labeled 'Таб_номер_официанта' (integer), 'ФИО' (character varying (50)), and 'Кол_во_столов' (bigint). The data rows show the following information:

	Таб_номер_официанта integer	ФИО character varying (50)	Кол_во_столов bigint
1	289731	Услужко Антон Павлович	2
2	564738	Улыбова Виктория Андреевна	1
3	576482	Кармен Сара Генриховна	1
4	667843	Зюзин Константин Никитич	3
5	673762	Прудько Виктория Васильевна	1

At the bottom of the interface, it shows 'Total rows: 5 of 5' and 'Query complete 00:00:00.048'.

Рисунок 7 – Выполнение запроса SELECT 6

7. Какой из ингредиентов используется в максимальном количестве блюд?

```
SELECT "Код_ингредиента",  
       (SELECT "Наименование" FROM "Ресторан"."Ингредиент" WHERE  
        "Ингредиент"."Код_ингредиента" = "Состав_блюда"."Код_ингредиента")  
FROM "Ресторан"."Состав_блюда" GROUP BY "Код_ингредиента" HAVING  
COUNT (DISTINCT "Код_блюда") =  
       (SELECT MAX("Кол_во_блюд") FROM (SELECT COUNT(DISTINCT  
        "Код_блюда") "Кол_во_блюд"
```

FROM "Ресторан"."Состав_блюда" GROUP BY "Код_ингредиента") AS
max_dishes)

```

62 SELECT "Код_ингредиента",
63       (SELECT "Наименование" FROM "Ресторан"."Ингредиент" WHERE "Ингредиент"."Код_ингредиента" = "Состав_блюда"."Код_ингредиента")
64 FROM "Ресторан"."Состав_блюда" GROUP BY "Код_ингредиента" HAVING COUNT (DISTINCT "Код_блюда") =
65      (SELECT MAX("Кол_во_блюдо") FROM (SELECT COUNT(DISTINCT "Код_блюда") "Кол_во_блюдо"
66      FROM "Ресторан"."Состав_блюда" GROUP BY "Код_ингредиента") AS max_dishes)
67

```

Data Output Messages Notifications

Код_ингредиента integer	Наименование character varying (60)
1	101 Масло сливочное 82,5%

Total rows: 1 of 1 Query complete 00:00:00.096

Рисунок 8 – Выполнение запроса SELECT 7

4 Представления

1. Создать представление для расчета стоимости ингредиентов для заданного блюда.

```
CREATE VIEW "Стоимость_ингредиентов_блюда" AS
SELECT "Код_блюда", "Наименование",
       (SELECT ROUND(SUM("Объем_ингредиента" * (SELECT "Цена"::float
/      "Объем"::float      FROM      "Ресторан"."Закупка"      WHERE
"Закупка"."Код_ингредиента" = "Состав_блюда"."Код_ингредиента" AND
"Остаток" >= "Состав_блюда"."Объем_ингредиента" AND "Дата" <=
"Срок_годности" ORDER BY "Дата" LIMIT 1))::numeric, 2) AS
"Стоимость_ингредиентов"
FROM "Ресторан"."Состав_блюда"
WHERE "Состав_блюда"."Код_блюда" = "Блюдо"."Код_блюда"
GROUP BY "Код_блюда")
FROM "Ресторан"."Блюдо"
```

Код_блюда integer	Наименование character varying (50)	Стоимость_ингредиентов numeric
1	4 Пирожок с капустой	59.72
2	5 Облепиховый чай	29.50
3	6 Борщ постный	83.50
4	7 Греча с грибами	100.93
5	9 Бутерброд с икрой	682.66
6	3 Масса тревожная	291.11
7	1 Котлеты морковные	58.10
8	2 Щи	76.96
9	8 Лаймовый лимонад	10.04

Total rows: 9 of 9 Query complete 00:00:00.052

Рисунок 9 – Просмотр содержимого VIEW 1

2. Создать представление для всех поваров количество приготовленных блюд по каждому блюду за определенную дату.

```
CREATE VIEW "Блюда_повара_за_дату" AS
SELECT "Дата"::date, "Таб_номер_повара",
       (SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE
        "Состав_заказа"."Таб_номер_повара" = "Сотрудник"."Таб_номер"),
       "Код_блюда",
       (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE
        "Состав_заказа"."Код_блюда" = "Блюдо"."Код_блюда"),
       SUM("Кол_во") "Кол_во"
FROM "Ресторан"."Заказ" JOIN "Ресторан"."Состав_заказа" ON
"Заказ"."№_заказа" = "Состав_заказа"."№_заказа"
GROUP BY "Дата"::date, "Таб_номер_повара", "Код_блюда"
ORDER BY "Дата" DESC
```

```

16 CREATE VIEW "Блюда_повара_за_дату" AS
17 SELECT "Дата"::date, "Таб_номер_повара",
18      (SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE "Состав_заказа"."Таб_номер_повара" = "Сотрудник"."Таб_номер"),
19      "Код_блюда",
20      (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Состав_заказа"."Код_блюда" = "Блюдо"."Код_блюда"),
21      SUM("Кол_во") "Кол_во"
22 FROM "Ресторан"."Заказ" JOIN "Ресторан"."Состав_заказа" ON
23 "Заказ"."№_заказа" = "Состав_заказа"."№_заказа"
24 GROUP BY "Дата"::date, "Таб_номер_повара", "Код_блюда"
25 ORDER BY "Дата" DESC
26
27
28 SELECT * FROM "Блюда_повара_за_дату"

```

	Дата date	Таб_номер_повара integer	ФИО character varying (50)	Код_блюда integer	Наименование character varying (50)	Кол_во bigint
1	2023-03-15	435276	Плюшкин Юрий Алексеевич	8	Лаймовый лимонад	3
2	2023-03-15	443111	Жирков Леонид Дмитриевич	4	Пирожок с капустой	5
3	2023-03-15	454432	Маслова Ирина Юрьевна	2	Щи	2
4	2023-03-15	454432	Маслова Ирина Юрьевна	9	Бутерброд с икрой	2
5	2023-03-15	563321	Ракова Елена Владимировна	1	Котлеты морковные	3
6	2023-03-15	563321	Ракова Елена Владимировна	9	Бутерброд с икрой	2
7	2023-03-15	565647	Мясоедов Илья Петрович	1	Котлеты морковные	1
8	2023-03-15	565647	Мясоедов Илья Петрович	7	Греча с грибами	1
9	2023-03-15	871237	Перцев Павел Степанович	5	Облепиховый чай	3
10	2023-03-14	443111	Жирков Леонид Дмитриевич	1	Котлеты морковные	1
11	2023-03-14	443111	Жирков Леонид Дмитриевич	5	Облепиховый чай	1
12	2023-03-14	454432	Маслова Ирина Юрьевна	9	Бутерброд с икрой	4
13	2023-03-14	565647	Мясоедов Илья Петрович	3	Масса тревожная	3
14	2023-03-14	565647	Мясоедов Илья Петрович	8	Лаймовый лимонад	4
15	2023-03-13	563321	Ракова Елена Владимировна	4	Пирожок с капустой	1

Total rows: 22 of 22 Query complete 00:00:00.053

Рисунок 10 – Просмотр содержимого VIEW 2

5 Запросы на модификацию данных

1. INSERT

Назначить официантов на последнюю доступную смену. Нужны все "отдохнувшие" официанты, то есть кто не работал хотя бы 1 день перед заданной сменой.

```
INSERT INTO "Ресторан"."Состав_смены"
SELECT      "Таб_номер",      (SELECT      MAX("Код_смены")      FROM
"Ресторан"."Смена"), 0 FROM "Ресторан"."Сотрудник"
WHERE  "Код_должности" IN (SELECT "Код_должности" FROM
"Ресторан"."Должность" WHERE "Наименование" = 'Официант') AND
NOT EXISTS(SELECT * FROM "Ресторан"."Состав_смены" WHERE
"Таб_номер_сотрудника" = "Сотрудник"."Таб_номер" AND
"Выход_на_смену" = 1 AND "Код_смены" IN
(SELECT "Код_смены" FROM "Ресторан"."Смена"
WHERE "Дата" + interval'1 day' >=
(SELECT "Дата" FROM "Ресторан"."Смена"
WHERE "Код_смены" = (SELECT
MAX("Код_смены") FROM "Ресторан"."Смена")
)
)
)
```

1	SELECT * FROM "Ресторан"."Состав_смены"			
2	ORDER BY "№_состава_смены" ASC			

Data Output	Messages	Notifications
-------------	----------	---------------

--	--	--	--	--	--	--	--

	Таб_номер_сотрудника integer	Код_смены integer	Выход_на_смену integer	№_состава_смены [PK] integer
5	564738	1	1	5
6	673762	5	1	6
7	673762	7	1	7
8	564738	4	1	8
9	564738	7	1	9
10	564738	2	1	10
11	667843	2	1	11
12	564738	2	1	12
13	667843	3	1	13
14	667843	4	1	14
15	576482	2	1	15
16	564738	3	1	16
17	576482	7	1	17
18	576482	5	1	18
19	667843	5	1	19
20	673762	4	0	20
21	289731	3	1	21
22	673762	3	0	22
23	289731	3	1	23
24	667843	7	1	24
25	289731	6	1	25
26	673762	2	0	26
27	667843	6	1	27

Total rows: 27 of 27	Query complete 00:00:00.113
----------------------	-----------------------------

Рисунок 11 – Просмотр содержимого БД до INSERT

```

1 SELECT * FROM "Ресторан"."Состав_смены"
2 ORDER BY "№_состава_смены" ASC

```

Data Output Messages Notifications

	Таб_номер_сотрудника integer	Код_смены integer	Выход_на_смену integer	№_состава_смены [PK] integer
7	673762	7	1	7
8	564738	4	1	8
9	564738	7	1	9
10	564738	2	1	10
11	667843	2	1	11
12	564738	2	1	12
13	667843	3	1	13
14	667843	4	1	14
15	576482	2	1	15
16	564738	3	1	16
17	576482	7	1	17
18	576482	5	1	18
19	667843	5	1	19
20	673762	4	0	20
21	289731	3	1	21
22	673762	3	0	22
23	289731	3	1	23
24	667843	7	1	24
25	289731	6	1	25
26	673762	2	0	26
27	667843	6	1	27
28	289731	8	0	28
29	781212	8	0	29

Total rows: 29 of 29

Query complete 00:00:00.091

Рисунок 12 – Просмотр содержимого БД после INSERT

2. UPDATE

Увеличить оклад каждой должности по коэффициенту:

1) Официантам - коэффициент равен 1/100000 суммы всех заказов за прошедший месяц.

2) Поварам - коэффициент равен 1/500 суммарного кол-ва приготовленных блюд за прошедший месяц.

3) Остальным - увеличить на 5%.

```
UPDATE "Ресторан"."Должность" SET "Оклад" = CASE
WHEN "Наименование" = 'Официант'
THEN "Оклад"*(SELECT 1 + SUM((SELECT SUM("Кол_во" * (SELECT
"Цена" FROM "Ресторан"."Блюдо"
WHERE "Блюдо"."Код_блюда" = "Состав_заказа"."Код_блюда"))
FROM "Ресторан"."Состав_заказа"
WHERE "Состав_заказа"."№_заказа" =
"Заказ"."№_заказа")) / 100000
FROM "Ресторан"."Заказ" WHERE "Дата" BETWEEN
NOW() - interval'1 month' AND NOW())
WHEN LOWER("Наименование") LIKE '%повар%' OR
LOWER("Наименование") LIKE '%шеф%'
THEN "Оклад"*(SELECT 1 + (SUM("Кол_во") / 500) FROM
"Блюда_повара_за_дату" WHERE "Дата" BETWEEN NOW() - interval'1
month' AND NOW())
ELSE "Оклад"*1.05
END
```

```

180 SELECT * FROM "Ресторан"."Должность"
181 ORDER BY "Код_должности" ASC

```

Data Output Messages Notifications

	Код_должности [PK] integer	Наименование character (50)	Оклад integer	Кол_во_ставок real
1	101	Шеф-повар	350000	1
2	102	Су-шеф	200000	1
3	103	Повар 2 категории	130000	1
4	104	Повар 1 категории	96000	1
5	105	Повар 1 категории	48000	0.5
6	106	Повар-стажер	35000	0.5
7	201	Администратор зала	112000	1
8	202	Хостес	84000	1
9	203	Бармен	79000	1
10	204	Официант	60000	1
11	205	Официант	30000	0.5

Total rows: 11 of 11 Query complete 00:00:00.065

Рисунок 13 – Просмотр содержимого БД до UPDATE

```

180 SELECT * FROM "Ресторан"."Должность"
181 ORDER BY "Код_должности" ASC

```

Data Output Messages Notifications

	Код_должности [PK] integer	Наименование character (50)	Оклад integer	Кол_во_ставок real
1	101	Шеф-повар	384300	1
2	102	Су-шеф	219600	1
3	103	Повар 2 категории	136500	1
4	104	Повар 1 категории	100800	1
5	105	Повар 1 категории	50400	0.5
6	106	Повар-стажер	36750	0.5
7	201	Администратор зала	117600	1
8	202	Хостес	88200	1
9	203	Бармен	82950	1
10	204	Официант	67410	1
11	205	Официант	33705	0.5

Total rows: 11 of 11 Query complete 00:00:00.039

Рисунок 14 – Просмотр содержимого БД после UPDATE

3. DELETE

Для каждого повара удалить из списка блюд, которые он умеет готовить, те, которые он ни разу не готовил за последние полгода. Но это не касается всех напитков и блюд, в которых меньше трех ингредиентов.

```
DELETE FROM "Ресторан"."Набор_блюд_повара" WHERE
    NOT EXISTS (SELECT * FROM "Ресторан"."Состав_заказа"
        WHERE "Состав_заказа"."Таб_номер_повара" =
"Набор_блюд_повара"."Таб_номер_повара"
        AND "Состав_заказа"."Код_блюда" =
"Набор_блюд_повара"."Код_блюда"
        AND (SELECT "Дата" FROM "Ресторан"."Заказ"
            WHERE "Заказ"."№_заказа" = "Состав_заказа"."№_заказа")
            BETWEEN NOW() - interval'6 months' AND NOW())
    AND (SELECT "Категория" FROM "Ресторан"."Блюдо"
        WHERE "Блюдо"."Код_блюда" =
"Набор_блюд_повара"."Код_блюда") != 'Напиток'
        AND (SELECT COUNT("Код_ингредиента") FROM
"Ресторан"."Состав_блюда"
            WHERE "Состав_блюда"."Код_блюда" =
"Набор_блюд_повара"."Код_блюда") > 2
```

```

1 SELECT * FROM "Ресторан"."Набор_блюдов_повара"
2 ORDER BY "№_набора" ASC

```

Data Output Messages Notifications

	№_набора [PK] integer	Таб_номер_повара integer	Код_блюда integer
1	1	443111	1
2	2	443111	2
3	3	443111	3
4	4	443111	4
5	5	443111	5
6	6	443111	6
7	7	443111	7
8	8	443111	8
9	9	443111	9
10	10	565647	1
11	11	565647	2
12	12	565647	3
13	13	565647	4
14	14	565647	5
15	15	565647	6
16	16	565647	7
17	17	565647	8
18	18	565647	9
19	19	454432	1
20	20	454432	2
21	21	454432	4
22	22	454432	6
Total rows: 43 of 43		Query complete 00:00:00.553	

Рисунок 15 – Просмотр содержимого БД до DELETE

```

1 SELECT * FROM "Ресторан"."Набор_блюдо_повара"
2 ORDER BY "№_набора" ASC

```

Data Output Messages Notifications

	№_набора [PK] integer	Таб_номер_повара integer	Код_блюда integer
1	1	443111	1
2	4	443111	4
3	5	443111	5
4	8	443111	8
5	10	565647	1
6	12	565647	3
7	14	565647	5
8	16	565647	7
9	17	565647	8
10	20	454432	2
11	24	454432	9
12	26	764321	3
13	27	764321	5
14	28	764321	6
15	29	764321	8
16	32	763565	5
17	33	763565	8
18	34	563321	1
19	35	563321	4
20	36	563321	7
21	37	563321	9
22	39	135276	5
Total rows: 26 of 26		Query complete 00:00:00.067	

Рисунок 16 – Просмотр содержимого БД после DELETE

6 Создание индексов

1. Простой индекс

Сравнение выполнения без индекса и с простым индексом для запроса SELECT №3.

Без индекса:

```
169 SELECT "Код_блюда",
170       (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" = "Состав_блюда"."Код_блюда"),
171       COUNT (DISTINCT "Код_ингредиента") "Кол_во_ингредиентов"
172 FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда"
173
```

	Код_блюда integer	Наименование character varying (50)	Кол_во_ингредиентов bigint
1	1	Котлеты морковные	4
2	2	Щи	5
3	3	Масса тревожная	4
4	4	Пирожок с капустой	4
5	5	Облепиховый чай	2
6	6	Борщ постный	7
7	7	Греча с грибами	4
8	8	Лаймовый лимонад	2
9	9	Бутерброд с икрой	4

Total rows: 9 of 9 Query complete 00:00:00.073

Рисунок 17 – Выполнение запроса SELECT №3 без индекса

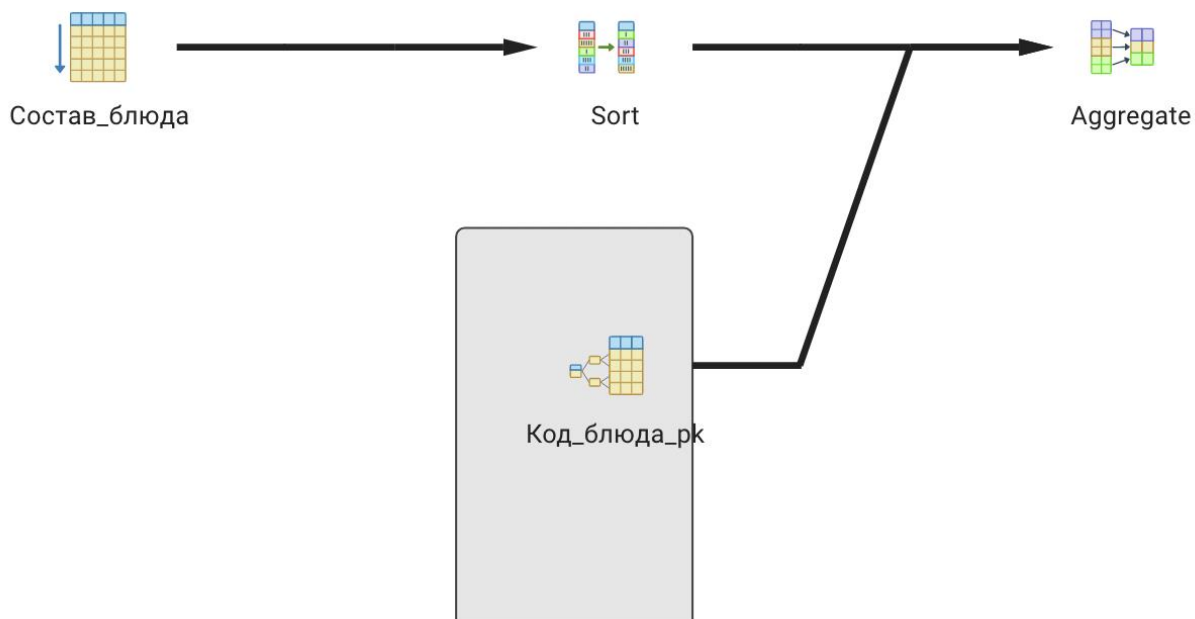


Рисунок 18 – Схема запроса SELECT №3 без индекса

	QUERY PLAN	
	text	
1	GroupAggregate (cost=2.29..296.86 rows=36 width=130)	
2	Group Key: "Состав_блюда"."Код_блюда"	
3	-> Sort (cost=2.29..2.38 rows=36 width=8)	
4	Sort Key: "Состав_блюда"."Код_блюда"	
5	-> Seq Scan on "Состав_блюда" (cost=0.00..1.36 rows=36 width=8)	
6	SubPlan 1	
7	-> Index Scan using "Код_блюда_pk" on "Блюдо" (cost=0.15..8.17 rows=1 width=118)	
8	Index Cond: ("Код_блюда" = "Состав_блюда"."Код_блюда")	

Рисунок 19 – План запроса SELECT №3 без индекса

С индексом:

CREATE INDEX "idx_состав_код_блюда" ON
"Ресторан"."Состав_блюда"("Код_блюда")

```

169 SELECT "Код_блюда",
170       (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Блюдо"."Код_блюда" = "Состав_блюда"."Код_блюда"),
171       COUNT (DISTINCT "Код_ингредиента") "Кол_во_ингредиентов"
172 FROM "Ресторан"."Состав_блюда" GROUP BY "Код_блюда"

```

Data Output			
	Код_блюда integer	Наименование character varying (50)	Кол_во_ингредиентов bigint
1	1	Котлеты морковные	4
2	2	Щи	5
3	3	Масса тревожная	4
4	4	Пирожок с капустой	4
5	5	Облепиховый чай	2
6	6	Борщ постный	7
7	7	Греча с грибами	4
8	8	Лаймовый лимонад	2
9	9	Бутерброд с икрой	4

Total rows: 9 of 9 Query complete 00:00:00.050

Рисунок 20 – Выполнение запроса SELECT №3 с индексом

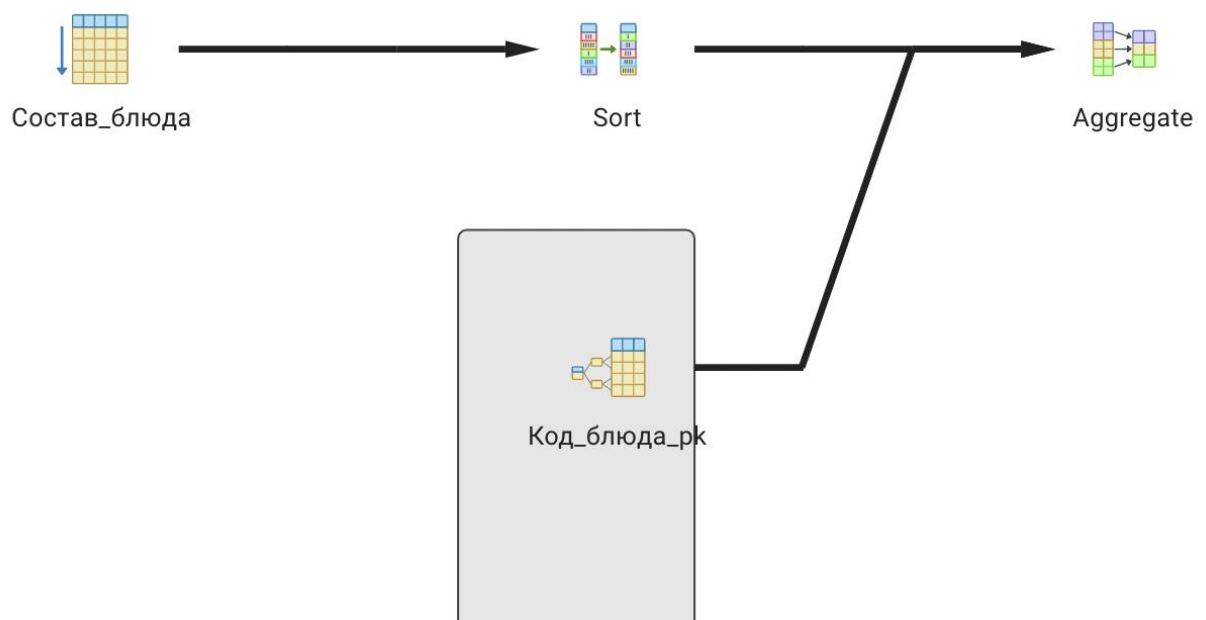


Рисунок 21 – Схема запроса SELECT №3 с индексом

	QUERY PLAN	
	text	
1	GroupAggregate (cost=2.29..296.86 rows=36 width=130)	
2	Group Key: "Состав_блюда"."Код_блюда"	
3	-> Sort (cost=2.29..2.38 rows=36 width=8)	
4	Sort Key: "Состав_блюда"."Код_блюда"	
5	-> Seq Scan on "Состав_блюда" (cost=0.00..1.36 rows=36 width=8)	
6	SubPlan 1	
7	-> Index Scan using "Код_блюда_pk" on "Блюдо" (cost=0.15..8.17 rows=1 width=118)	
8	Index Cond: ("Код_блюда" = "Состав_блюда"."Код_блюда")	

Рисунок 22 – План запроса SELECT №3 с индексом

План выполнения запроса без индекса и с индексом не изменился, однако наблюдается незначительное уменьшение времени выполнения запроса при использовании индекса.

2. Составной индекс

Сравнение выполнения без индекса и с составным индексом (и одним простым) для запроса SELECT из представления VIEW №2.

Без индекса:

```

182 SELECT "Дата"::date, "Таб_номер_повара",
183       (SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE "Состав_заказа"."Таб_номер_повара" = "Сотрудник"."Таб_номер"),
184       "Код_блюда",
185       (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Состав_заказа"."Код_блюда" = "Блюдо"."Код_блюда"),
186       SUM("Кол_во") "Кол_во"
187 FROM "Ресторан"."Заказ" JOIN "Ресторан"."Состав_заказа" ON
188 "Заказ"."№_заказа" = "Состав_заказа"."№_заказа"
189 GROUP BY "Дата"::date, "Таб_номер_повара", "Код_блюда"
190 ORDER BY "Дата" DESC

```

	Дата date	Таб_номер_повара integer	ФИО character varying (50)	Код_блюда integer	Наименование character varying (50)	Кол_во bigint
1	2023-03-15	435276	Плюшкин Юрий Алексеевич	8	Лаймовый лимонад	3
2	2023-03-15	443111	Жирков Леонид Дмитриевич	4	Пирожок с капустой	5
3	2023-03-15	454432	Маслова Ирина Юрьевна	2	Щи	2
4	2023-03-15	454432	Маслова Ирина Юрьевна	9	Бутерброд с икрой	2

Total rows: 22 of 22 Query complete 00:00:00.105

Рисунок 23 – Выполнение запроса SELECT из VIEW №2 без индекса

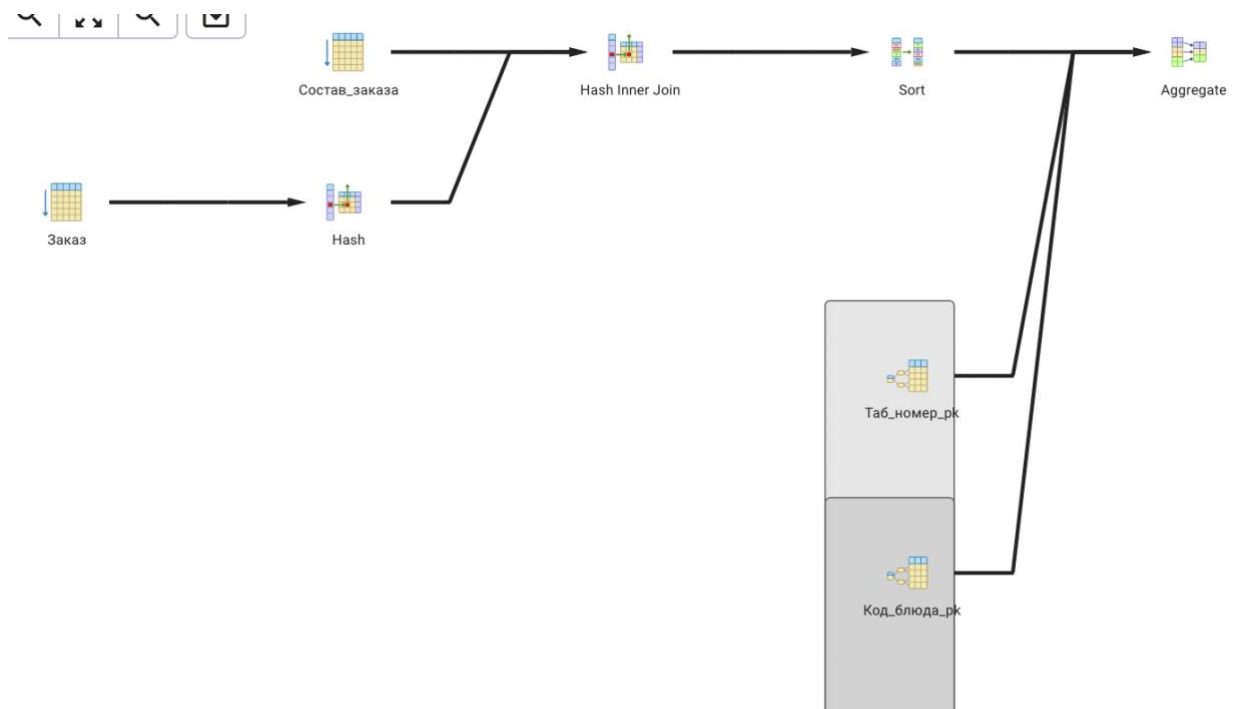


Рисунок 24 – Схема запроса SELECT из VIEW №2 без индекса


	QUERY PLAN	
	text	
1	GroupAggregate (cost=3.08..379.25 rows=23 width=256)	
2	Group Key: (("Заказ"."Дата")::date), "Состав_заказа"."Таб_номер_повара", "Состав_заказа"."Код_блюда"	
3	-> Sort (cost=3.08..3.14 rows=23 width=16)	
4	Sort Key: (("Заказ"."Дата")::date) DESC, "Состав_заказа"."Таб_номер_повара", "Состав_заказа"."Код_блюда"	
5	-> Hash Join (cost=1.20..2.56 rows=23 width=16)	
6	Hash Cond: ("Состав_заказа"."№_заказа" = "Заказ"."№_заказа")	
7	-> Seq Scan on "Состав_заказа" (cost=0.00..1.23 rows=23 width=16)	
8	-> Hash (cost=1.09..1.09 rows=9 width=12)	
9	-> Seq Scan on "Заказ" (cost=0.00..1.09 rows=9 width=12)	
10	SubPlan 1	
11	-> Index Scan using "Таб_номер_pk" on "Сотрудник" (cost=0.15..8.17 rows=1 width=118)	
12	Index Cond: ("Таб_номер" = "Состав_заказа"."Таб_номер_повара")	
13	SubPlan 2	
14	-> Index Scan using "Код_блюда_pk" on "Блюда" (cost=0.15..8.17 rows=1 width=118)	
15	Index Cond: ("Код_блюда" = "Состав_заказа"."Код_блюда")	

Рисунок 25 – План запроса SELECT из VIEW №2 без индекса

С индексом:

```
CREATE INDEX "idx_состав_таб_код" ON
"Ресторан"."Состав_заказа"("Таб_номер_повара", "Код_блюда")
CREATE INDEX "idx_заказ_дата" ON "Ресторан"."Заказ"("Дата")
```

```

182 SELECT "Дата"::date, "Таб_номер_повара",
183       (SELECT "ФИО" FROM "Ресторан"."Сотрудник" WHERE "Состав_заказа"."Таб_номер_повара" = "Сотрудник"."Таб_номер"),
184       "Код_блюда",
185       (SELECT "Наименование" FROM "Ресторан"."Блюдо" WHERE "Состав_заказа"."Код_блюда" = "Блюдо"."Код_блюда"),
186       SUM("Кол_во") "Кол_во"
187 FROM "Ресторан"."Заказ" JOIN "Ресторан"."Состав_заказа" ON
188 "Заказ"."№_заказа" = "Состав_заказа"."№_заказа"
189 GROUP BY "Дата"::date, "Таб_номер_повара", "Код_блюда"
190 ORDER BY "Дата" DESC
191
192
193 CREATE INDEX "idx_состав_таб_код" ON "Ресторан"."Состав_заказа" ("Таб_номер_повара", "Код_блюда")
194

```

Data Output Messages Notifications

	Дата date	Таб_номер_повара integer	ФИО character varying (50)	Код_блюда integer	Наименование character varying (50)	Кол_во bigint
1	2023-03-15	435276	Плюшкин Юрий Алексеевич	8	Лаймовый лимонад	3
2	2023-03-15	443111	Жирков Леонид Дмитриевич	4	Пирожок с капустой	5
3	2023-03-15	454432	Маслова Ирина Юрьевна	2	Щи	2
4	2023-03-15	454432	Маслова Ирина Юрьевна	9	Бутерброд с икрой	2
5	2023-03-15	563321	Ракова Елена Владимировна	1	Котлеты морковные	3
6	2023-03-15	563321	Ракова Елена Владимировна	9	Бутерброд с икрой	2
7	2023-03-15	565647	Мясоедов Илья Петрович	1	Котлеты морковные	1
8	2023-03-15	565647	Мясоедов Илья Петрович	7	Греча с грибами	1
9	2023-03-15	871237	Перцев Павел Степанович	5	Облепиховый чай	3
10	2023-03-14	443111	Жирков Леонид Дмитриевич	1	Котлеты морковные	1

Total rows: 22 of 22 Query complete 00:00:00.039

Рисунок 23 – Выполнение запроса SELECT из VIEW №2 с индексом

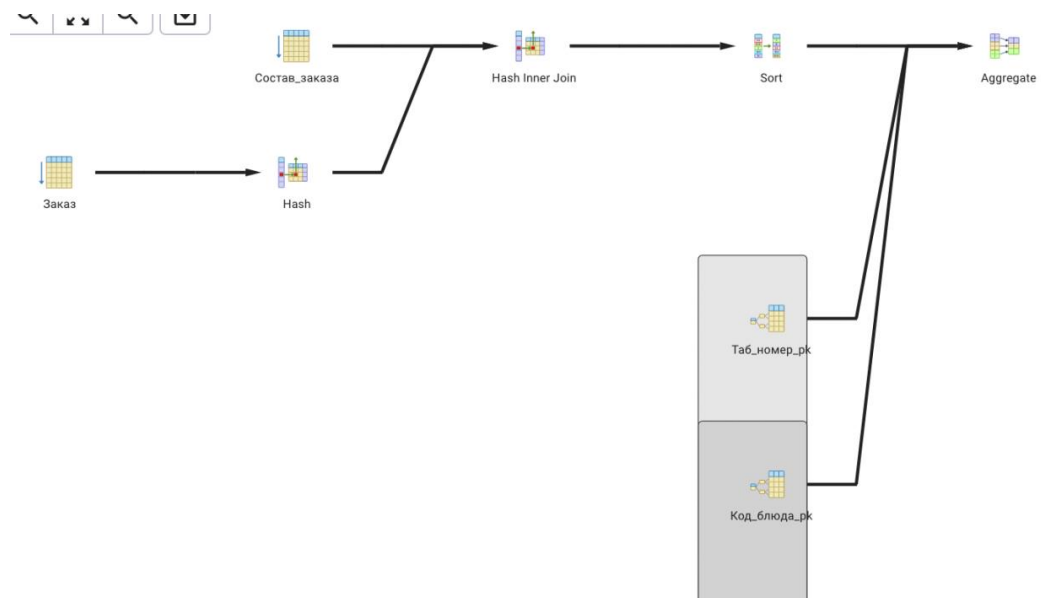


Рисунок 24 – Схема запроса SELECT из VIEW №2 с индексом


	QUERY PLAN	
	text	
1	GroupAggregate (cost=3.08..379.25 rows=23 width=256)	
2	Group Key: (("Заказ"."Дата")::date), "Состав_заказа"."Таб_номер_повара", "Состав_заказа"."Код_блюда"	
3	-> Sort (cost=3.08..3.14 rows=23 width=16)	
4	Sort Key: (("Заказ"."Дата")::date) DESC, "Состав_заказа"."Таб_номер_повара", "Состав_заказа"."Код_блюда"	
5	-> Hash Join (cost=1.20..2.56 rows=23 width=16)	
6	Hash Cond: ("Состав_заказа"."№_заказа" = "Заказ"."№_заказа")	
7	-> Seq Scan on "Состав_заказа" (cost=0.00..1.23 rows=23 width=16)	
8	-> Hash (cost=1.09..1.09 rows=9 width=12)	
9	-> Seq Scan on "Заказ" (cost=0.00..1.09 rows=9 width=12)	
10	SubPlan 1	
11	-> Index Scan using "Таб_номер_pk" on "Сотрудник" (cost=0.15..8.17 rows=1 width=118)	
12	Index Cond: ("Таб_номер" = "Состав_заказа"."Таб_номер_повара")	
13	SubPlan 2	
14	-> Index Scan using "Код_блюда_pk" on "Блюда" (cost=0.15..8.17 rows=1 width=118)	
15	Index Cond: ("Код_блюда" = "Состав_заказа"."Код_блюда")	

Рисунок 25 – План запроса SELECT из VIEW №2 с индексом

Аналогично предыдущему индексу план выполнения запроса не изменился. Тем не менее, время выполнения запроса при использовании индекса заметно уменьшилось.

ВЫВОДЫ

В рамках данной лабораторной работы получены практические навыки создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов. Реализованы 7 запросов на выборку данных с командой SELECT и 2 представления с командой VIEW согласно индивидуальному заданию - варианту 13 БД «Ресторан». Составлены 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Изучены графические представления запросов и просмотрена история запросов. Также были созданы простой и составной индексы для двух запросов. При сравнении времени выполнения и планов запросов без индексов и с индексами, выявлено, что для данных запросов план выполнения не изменился, то есть планировщик запросов счел использование индексов по первичному ключу более эффективным. Однако наблюдалось незначительное увеличение производительности, которое вероятно было случайным, так как явно его можно увидеть только при большой наполненности базы данными (большем количестве записей). Дополнительно получен опыт использования программы pgAdmin 4.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лекция 2.2.2 Индексы в реляционной БД. 2023. (дата обращения 19.03.2023)
2. Документация PostgreSQL [Электронный ресурс] // Официальный сайт PostgreSQL. 1996–2023. URL: <https://www.postgresql.org/docs/13/index.html> (дата обращения: 15.03.2023).
3. Документация pgAdmin 4 PostgreSQL [Электронный ресурс] // Официальный сайт pgAdmin. URL: <https://www.pgadmin.org/docs/pgadmin4/latest/> (дата обращения 15.03.2023)
4. 2.2.1.8 SELECT SQL. Представления. Модификация данных. 2023. (дата обращения 17.03.2023)