

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

О Т Ч Е Т

по лабораторной работе

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

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Бармина Диана Андреевна

Факультет: ИКТ

Группа: K32421

Преподаватель: Говорова Марина Михайловна

Дата сдачи: 14.04.23



Санкт-Петербург 2023

СОДЕРЖАНИЕ

1 Описание работы	3
2 Схема базы данных	4
3 Выполнение запросов	5
3.1 Запрос №1	5
3.2 Запрос №2	6
3.3 Запрос №3	7
3.4 Запрос №4	8
3.5 Запрос №5	9
3.6 Запрос №6	10
3.7 Запрос №7	11
4 Представления	13
4.1 Представление №1	13
4.2 Представление №2	14
5 Запросы на модификацию данных	15
5.1 Update с подзапросом	15
5.2 Delete с подзапросом	16
5.3 Insert с подзапросом	18
6 Запросы с индексами	20
6.1 Запрос с индексами 1	20
6.2 Запрос с индексами 2	22
ЗАКЛЮЧЕНИЕ	25

1 Описание работы

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 по варианту «Ресторан».

2 Схема базы данных

Схема была создана с использованием генератора ERD-схемы в pgadmin.

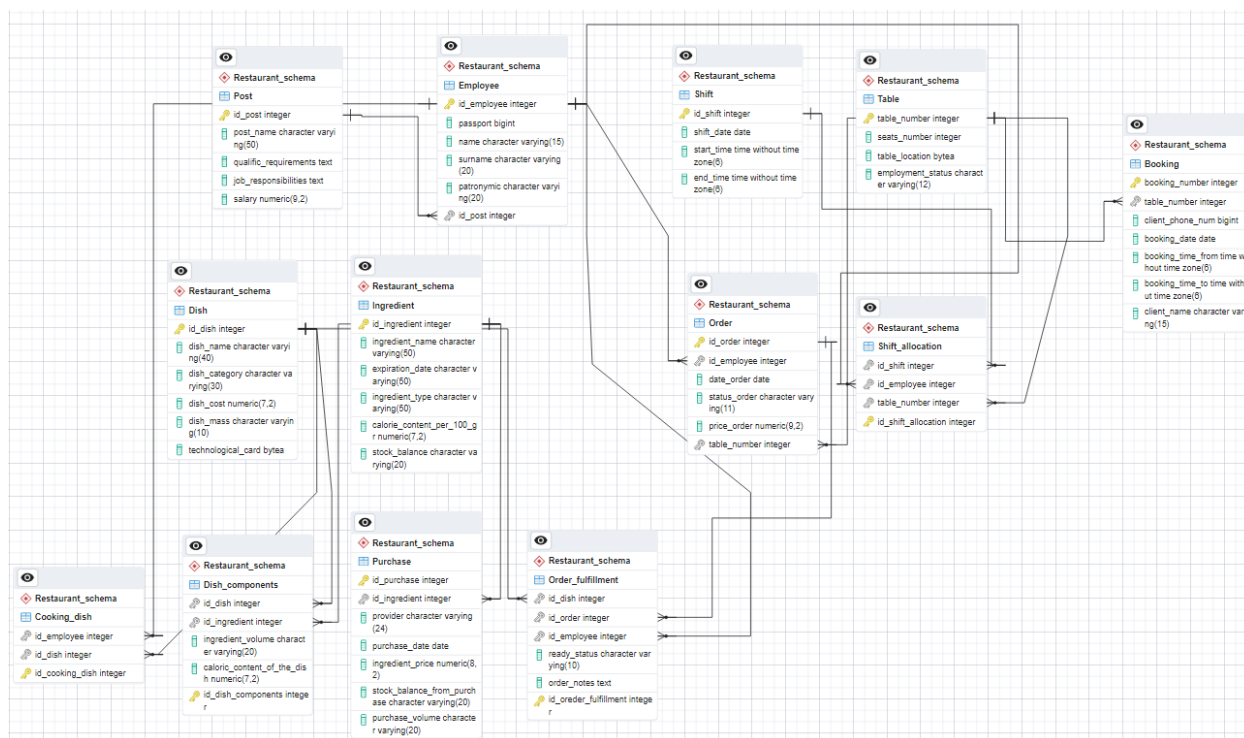


Рисунок 1 – Схема базы данных «Ресторан»

3 Выполнение запросов

3.1 Запрос №1

Запрос №1: вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.

Код SQL:

```
SELECT "Restaurant_schema"."Employee".surname,  
       "Restaurant_schema"."Employee".name,  
       "Restaurant_schema"."Employee".patronymic,  
       table_3.id_employee,  
       table_3.sum  
FROM "Restaurant_schema"."Employee",  
     (SELECT id_employee,  
            sum(price_order)  
      FROM "Restaurant_schema"."Order"  
     WHERE id_employee =  
           (SELECT id_employee  
            FROM  
              (SELECT id_employee,  
                     sum(price_order)  
                FROM "Restaurant_schema"."Order"  
               WHERE date_order < date(now())  
                     AND date_order >= (date(now()) - 30)  
               GROUP BY id_employee) AS table_1  
             WHERE table_1.sum =  
                   (SELECT max(table_2.sum)  
                    FROM  
                      (SELECT id_employee,  
                             sum(price_order)  
                        FROM "Restaurant_schema"."Order"  
                       WHERE date_order < date(now())  
                             AND date_order >= (date(now()) - 30)  
                       GROUP BY id_employee) AS table_2))  
             GROUP BY id_employee) AS table_3  
     WHERE "Restaurant_schema"."Employee".id_employee =  
table_3.id_employee
```

Выполнение запроса:

```

select "Restaurant_schema"."Employee".surname,
       "Restaurant_schema"."Employee".name,
       "Restaurant_schema"."Employee".patronymic,
       table_3.id_employee, table_3.sum
from "Restaurant_schema"."Employee", (select id_employee, sum(price_order)
from "Restaurant_schema"."Order"
where id_employee = (select id_employee from
                     (select id_employee, sum(price_order)
                      from "Restaurant_schema"."Order"
                      where date_order < date(now()) and date_order >= (date(now()
                      group by id_employee) as table_1
                     where table_1.sum = (select max(table_2.sum)
                      from (select id_employee, sum(price_order)
                          from "Restaurant_schema"."Order"
                          where date_order < date(now()) and date_order >= (date(now()
                          group by id_employee) as table_2))
                     group by id_employee) as table_3
where "Restaurant_schema"."Employee".id_employee = table_3.id_employee

```

Data Output Messages Notifications					
	surname character varying (20)	name character varying (15)	patronymic character varying (20)	id_employee integer	sum numeric
1	Усова	Наталья	Алексеевна	4	10300.00

Рисунок 2 – Выполнение запроса №1

3.2 Запрос №2

Запрос №2: рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа).

Код SQL:

```

SELECT "Restaurant_schema"."Employee".surname,
       "Restaurant_schema"."Employee".name,
       "Restaurant_schema"."Employee".patronymic,
       table_3.id_employee,
       table_3.sum*0.05 AS premia
FROM "Restaurant_schema"."Employee",
     (SELECT id_employee,
            sum(price_order)
      FROM "Restaurant_schema"."Order"
      WHERE date_order <= date(now())
            AND date_order > (date(now()) - 10)
      GROUP BY id_employee) AS table_3
WHERE "Restaurant_schema"."Employee".id_employee = table_3.id_employee

```

Выполнение запроса:

<pre> select "Restaurant_schema"."Employee".surname, "Restaurant_schema"."Employee".name, "Restaurant_schema"."Employee".patronymic, table_3.id_employee, table_3.sum*0.05 as premia from "Restaurant_schema"."Employee", (select id_employee, sum(price_order) from "Restaurant_schema"."Order" where date_order <= date(now()) and date_order > (date(now()) - 10) group by id_employee) as table_3 where "Restaurant_schema"."Employee".id_employee = table_3.id_employee </pre>																													
<div> <div>Data Output Messages Notifications</div> <div> <div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div> <table> <tr> <th></th><th>surname character varying (20)</th><th>name character varying (15)</th><th>patronymic character varying (20)</th><th>id_employee integer</th><th>premia numeric</th></tr> <tr> <td>1</td><td>Макмутова</td><td>Олеся</td><td>Максимовна</td><td>3</td><td>312.5000</td></tr> <tr> <td>2</td><td>Усова</td><td>Наталья</td><td>Алексеевна</td><td>4</td><td>274.0000</td></tr> <tr> <td>3</td><td>Велесова</td><td>Анастасия</td><td>Андреевна</td><td>5</td><td>355.0000</td></tr> </table> </div> </div>							surname character varying (20)	name character varying (15)	patronymic character varying (20)	id_employee integer	premia numeric	1	Макмутова	Олеся	Максимовна	3	312.5000	2	Усова	Наталья	Алексеевна	4	274.0000	3	Велесова	Анастасия	Андреевна	5	355.0000
	surname character varying (20)	name character varying (15)	patronymic character varying (20)	id_employee integer	premia numeric																								
1	Макмутова	Олеся	Максимовна	3	312.5000																								
2	Усова	Наталья	Алексеевна	4	274.0000																								
3	Велесова	Анастасия	Андреевна	5	355.0000																								

Рисунок 3 – Выполнение запроса №2

3.3 Запрос №3

Запрос №3: подсчитать, сколько ингредиентов содержит каждое блюдо.

Код SQL:

```

SELECT id_dish,
       (SELECT dish_name
        FROM "Restaurant_schema"."Dish"
        WHERE "Restaurant_schema"."Dish".id_dish
              =
              "Restaurant_schema"."Dish_components".id_dish), count(distinct id_ingredient)
FROM "Restaurant_schema"."Dish_components"
GROUP BY id_dish

```

Выполнение запроса:

```

select id_dish,
(select dish_name from "Restaurant_schema"."Dish"
where "Restaurant_schema"."Dish".id_dish = "Restaurant_schema"."Dish_components".id_dish),
count(distinct id_ingredient)
from "Restaurant_schema"."Dish_components"
group by id_dish

```

	id_dish integer	dish_name character varying (40)	count bigint
1	1	Солянка	1
2	2	Яблочный пирог	2
3	4	Груша в вине	2
4	5	Паста с морепродуктами	1
5	6	Куриная грудка с пюре	3

Рисунок 4 – Выполнение запроса №3

3.4 Запрос №4

Запрос №4: вывести название блюда, содержащее максимальное число ингредиентов.

Код SQL:

```

SELECT "Restaurant_schema"."Dish".dish_name,
       table_3.count
FROM "Restaurant_schema"."Dish",
     (SELECT id_dish,
            count(id_ingredient)
     FROM "Restaurant_schema"."Dish_components"
     WHERE id_dish =
           (SELECT id_dish
            FROM
              (SELECT id_dish,
                     count(id_ingredient)
                  FROM "Restaurant_schema"."Dish_components"
                  GROUP BY id_dish) AS table_1
            WHERE table_1.count =
                  (SELECT max(table_2.count)
                   FROM
                     (SELECT id_dish,
                            count(id_ingredient)
                         FROM "Restaurant_schema"."Dish_components"
                         GROUP BY id_dish) AS table_2))
            GROUP BY id_dish) AS table_3
     WHERE "Restaurant_schema"."Dish".id_dish = table_3.id_dish

```


Выполнение запроса:

```
select "Restaurant_schema"."Dish".dish_name, table_3.count
from "Restaurant_schema"."Dish", (select id_dish, count(id_ingredient)
from "Restaurant_schema"."Dish_components"
where id_dish = (select id_dish from
                (select id_dish, count(id_ingredient)
                 from "Restaurant_schema"."Dish_components"
                 group by id_dish) as table_1
                where table_1.count = (select max(table_2.count)
                                     from (select id_dish, count(id_ingredient)
                                             from "Restaurant_schema"."Dish_components"
                                             group by id_dish) as table_2))
group by id_dish) as table_3
where "Restaurant_schema"."Dish".id_dish = table_3.id_dish
```

Data Output	Messages	Explain	×	Notifications
	dish_name character varying (40)	count bigint		
1	Куриная грудка с пюре	3		

Рисунок 5 – Выполнение запроса №4

3.5 Запрос №5

Запрос №5: Какой повар может приготовить максимальное число видов блюд?

Код SQL:

```
SELECT "Restaurant_schema"."Employee".surname,
       "Restaurant_schema"."Employee".name,
       "Restaurant_schema"."Employee".patronymic,
       table_3.id_employee,
       table_3.count
FROM "Restaurant_schema"."Employee",
     (SELECT id_employee,
            count(id_dish)
      FROM "Restaurant_schema"."Cooking_dish"
     WHERE id_employee =
           (SELECT id_employee
            FROM
              (SELECT id_employee,
                     count(id_dish)
                  FROM "Restaurant_schema"."Cooking_dish"
                 GROUP BY id_employee) AS table_1
            WHERE table_1.count =
                  (SELECT max(table_2.count)
                   FROM
                     (SELECT id_employee,
                            count(id_dish)
                      FROM "Restaurant_schema"."Cooking_dish"
```

```

GROUP BY id_employee) AS table_2))
GROUP BY id_employee) AS table_3
WHERE "Restaurant_schema"."Employee".id_employee =
table_3.id_employee

```

Выполнение запроса:

```

select "Restaurant_schema"."Employee".surname,
       "Restaurant_schema"."Employee".name,
       "Restaurant_schema"."Employee".patronymic,
       table_3.id_employee, table_3.count
from "Restaurant_schema"."Employee", (select id_employee, count(id_dish)
from "Restaurant_schema"."Cooking_dish"
where id_employee = (select id_employee from
                     (select id_employee, count(id_dish)
                      from "Restaurant_schema"."Cooking_dish"
                      group by id_employee) as table_1
                     where table_1.count = (select max(table_2.count)
from (select id_employee, count(id_dish)
from "Restaurant_schema"."Cooking_dish"
group by id_employee) as table_2))
group by id_employee) as table_3
where "Restaurant_schema"."Employee".id_employee = table_3.id_employee

```

Data Output Messages Explain × Notifications					
	surname character varying (20)	name character varying (15)	patronymic character varying (20)	id_employee integer	count bigint
1	Ромашкин	Алексей	Сергеевич	2	4

Рисунок 6 – Выполнение запроса №5

3.6 Запрос №6

Запрос №6: Сколько закреплено столов за каждым из официантов?

Код SQL:

```

SELECT id_employee,
       (SELECT surname
        FROM "Restaurant_schema"."Employee"
        WHERE "Restaurant_schema"."Employee".id_employee =
"Restaurant_schema"."Shift_allocation".id_employee), count(table_number)
FROM "Restaurant_schema"."Shift_allocation"
GROUP BY id_employee

```

Выполнение запроса:

```
select id_employee,
(select surname from "Restaurant_schema"."Employee"
 where "Restaurant_schema"."Employee".id_employee = "Restaurant_schema"."Shift_allocation".id_employee),
count(table_number)
from "Restaurant_schema"."Shift_allocation"
group by id_employee
```

Data Output Messages Explain x Notifications			
	id_employee integer	surname character varying (20)	count bigint
1	1	Сергеев	2
2	3	Макмутова	1
3	4	Усова	2

Рисунок 7 – Выполнение запроса №6

3.7 Запрос №7

Запрос №7: какой из ингредиентов используется в максимальном количестве блюд?

Код SQL:

```
SELECT id_ingredient,
      (SELECT ingredient_name
       FROM "Restaurant_schema"."Ingredient"
       WHERE "Restaurant_schema"."Ingredient".id_ingredient =
"Restaurant_schema"."Dish_components".id_ingredient), count(id_dish) AS count_dish
FROM "Restaurant_schema"."Dish_components"
WHERE id_ingredient =
      (SELECT id_ingredient
       FROM
        (SELECT id_ingredient,
                 count(id_dish)
        FROM "Restaurant_schema"."Dish_components"
        GROUP BY id_ingredient) AS table_1
        WHERE table_1.count =
          (SELECT max(table_1.count)
           FROM
            (SELECT id_ingredient,
                     count(id_dish)
            FROM "Restaurant_schema"."Dish_components"
            GROUP BY id_ingredient) AS table_1))
        GROUP BY id_ingredient
```

Выполнение запроса:

```

select id_ingredient,
(select ingredient_name from "Restaurant_schema"."Ingredient"
where "Restaurant_schema"."Ingredient".id_ingredient = "Restaurant_schema"."Dish_components".id_ingredient),
count(id_dish) as count_dish
from "Restaurant_schema"."Dish_components"
where id_ingredient = (select id_ingredient from
(select id_ingredient, count(id_dish)
from "Restaurant_schema"."Dish_components"
group by id_ingredient) as table_1
where table_1.count = (select max(table_1.count) from (select id_ingredient, count(id_dish)
from "Restaurant_schema"."Dish_components"
group by id_ingredient) as table_1)) group by id_ingredient

```

Data Output Messages Explain × Notifications

	id_ingredient integer	ingredient_name character varying (50)	count_dish bigint
1	5	Молоко	3

Рисунок 8 – Выполнение запроса №7

4 Представления

4.1 Представление №1

Представление №1: для расчета стоимости ингредиентов для заданного блюда.

Код SQL:

```
SELECT id_dish,
       dish_name,

       (SELECT
round(sum("Restaurant_schema"."Dish_components".ingredient_volume::numeric *
        (SELECT (ingredient_price / purchase_volume::numeric)
        FROM "Restaurant_schema"."Purchase",
        (SELECT min(purchase_date)
        FROM "Restaurant_schema"."Purchase"
        WHERE "Restaurant_schema"."Purchase".id_ingredient =
"Restaurant_schema"."Dish_components".id_ingredient) AS table1
        WHERE "Restaurant_schema"."Purchase".id_ingredient =
"Restaurant_schema"."Dish_components".id_ingredient
        AND "Restaurant_schema"."Purchase".purchase_date =
table1.min)), 2)
FROM "Restaurant_schema"."Dish_components"
WHERE "Restaurant_schema"."Dish_components".id_dish =
"Restaurant_schema"."Dish".id_dish
GROUP BY id_dish)
FROM "Restaurant_schema"."Dish";

SELECT * FROM dish_price_ing
```

Выполнение :

```
create view dish_price_ing as
select id_dish, dish_name,
(select round(sum("Restaurant_schema"."Dish_components".ingredient_volume::numeric *
(select (ingredient_price / purchase_volume::numeric)
from "Restaurant_schema"."Purchase", (select min(purchase_date) from "Restaurant_schema"."Purchase"
where "Restaurant_schema"."Purchase".id_ingredient =
"Restaurant_schema"."Dish_components".id_ingredient) as table1
where "Restaurant_schema"."Purchase".id_ingredient =
"Restaurant_schema"."Dish_components".id_ingredient
and "Restaurant_schema"."Purchase".purchase_date = table1.min)), 2)
from "Restaurant_schema"."Dish_components"
where "Restaurant_schema"."Dish_components".id_dish =
"Restaurant_schema"."Dish".id_dish
group by id_dish)
from "Restaurant_schema"."Dish";

select * from dish_price_ing
```

	id_dish integer	dish_name character varying (40)	round numeric
1	1	Солянка	6.82
2	2	Яблочный пирог	3.39
3	4	Груша в вине	7.61
4	5	Паста с морепродуктами	6.82
5	6	Куриная грудка с пюре	3.40

Рисунок 7 – Выполнение представления №1

4.1 Представление №2

Представление №2: для всех поваров количество приготовленных блюд по каждому блюду за определенную дату.

Код SQL:

```
create view count_dishes_for_date_employee_cook AS
SELECT date_order,
       "Restaurant_schema"."Order_fulfillment".id_employee,
       (SELECT surname FROM "Restaurant_schema"."Employee"
        WHERE "Restaurant_schema"."Order_fulfillment".id_employee =
"Restaurant_schema"."Employee".id_employee), "Restaurant_schema"."Order_fulfillment".id_dish,
       (SELECT dish_name
        FROM "Restaurant_schema"."Dish"
        WHERE "Restaurant_schema"."Order_fulfillment".id_dish =
"Restaurant_schema"."Dish".id_dish),
       (SELECT table1.count
        FROM
            (SELECT id_dish, id_employee, date_order, count(id_dish)
             FROM "Restaurant_schema"."Order_fulfillment"
             GROUP BY id_dish, id_employee, date_order) AS table1
        WHERE table1.id_dish = "Restaurant_schema"."Order_fulfillment".id_dish
          AND table1.id_employee =
"Restaurant_schema"."Order_fulfillment".id_employee
          AND table1.date_order =
"Restaurant_schema"."Order".date_order)
FROM "Restaurant_schema"."Order"
JOIN "Restaurant_schema"."Order_fulfillment"
ON "Restaurant_schema"."Order".id_order =
"Restaurant_schema"."Order_fulfillment".id_order
GROUP BY date_order,
       "Restaurant_schema"."Order_fulfillment".id_employee,
       "Restaurant_schema"."Order_fulfillment".id_dish;
SELECT * FROM count_dishes_for_date_employee_cook
```

Выполнение:

date_order	id_employee	surname	id_dish	dish_name	count
2023-03-19	1	Сергеев	4	Груша в вине	1
2023-03-19	1	Сергеев	6	Куриная грудка с пюре	2
2023-03-19	9	Муратова	5	Паста с морепродуктами	1
2023-04-08	1	Сергеев	6	Куриная грудка с пюре	2
2023-04-08	2	Ромашкин	1	Солянка	1
2023-04-08	2	Ромашкин	2	Яблочный пирог	1
2023-04-08	6	Боянь	2	Яблочный пирог	1
2023-04-08	8	Кувашов	2	Яблочный пирог	1

Рисунок 7 – Выполнение представления №2

5 Запросы на модификацию данных

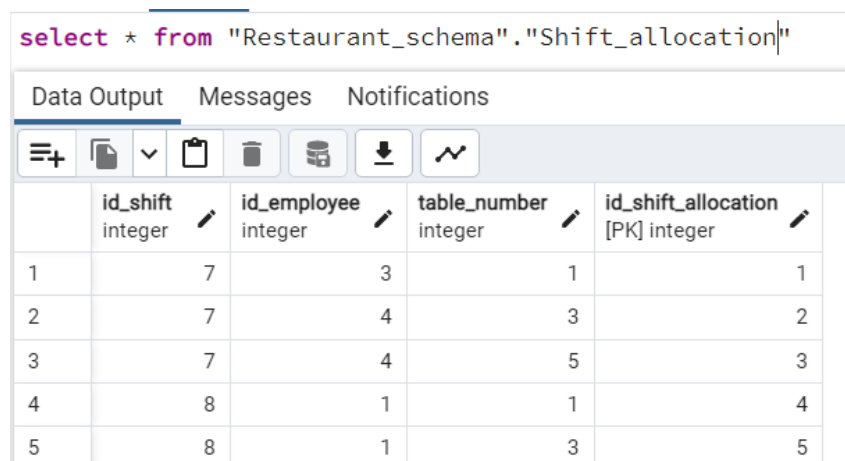
5.1 Update с подзапросом

Запрос: изменить данные в таблице «Распределение по сменам», а именно номер стола на 8, где номер сотрудника есть в списке сотрудников, принявших всего заказов на минимальную сумму за все время.

Код SQL:

```
UPDATE "Restaurant_schema"."Shift_allocation" SET table_number = 8
WHERE id_employee IN
    (SELECT id_employee
    FROM "Restaurant_schema"."Order"
    WHERE id_employee =
        (SELECT id_employee
        FROM
            (SELECT id_employee,
            sum(price_order)
            FROM "Restaurant_schema"."Order"
            GROUP BY id_employee) AS table_1
            WHERE table_1.sum =
                (SELECT min(table_2.sum)
                FROM
                    (SELECT id_employee,
                    sum(price_order)
                    FROM "Restaurant_schema"."Order"
                    GROUP BY id_employee) AS table_2))
            GROUP BY id_employee)
```

Наполнение таблицы до вставки данных:



select * from "Restaurant_schema"."Shift_allocation"				
	id_shift integer	id_employee integer	table_number integer	id_shift_allocation [PK] integer
1	7	3	1	1
2	7	4	3	2
3	7	4	5	3
4	8	1	1	4
5	8	1	3	5

Рисунок 8 – Выполнение запроса №7

Наполнение таблицы после вставки данных:

select * from "Restaurant_schema"."Shift_allocation"				
Data Output Messages Notifications				
	id_shift integer	id_employee integer	table_number integer	id_shift_allocation [PK] integer
1	7	4	3	2
2	7	4	5	3
3	8	1	1	4
4	8	1	3	5
5	7	3	8	1

Рисунок 8 – Выполнение запроса №7

5.2 Delete с подзапросом

Запрос: удалить из таблицы «Заказ» записи, номер стола которых есть в списке столов с наименьшей суммой цены заказов за все время.

Код SQL:

```
delete
FROM "Restaurant_schema"."Order"
WHERE table_number IN
    (SELECT table_1.table_number
    FROM
        (SELECT table_number,
            sum(price_order)
        FROM "Restaurant_schema"."Order"
        GROUP BY table_number) AS table_1
    WHERE table_1.sum =
        (SELECT min(table_2.sum)
        FROM
            (SELECT table_number,
                sum(price_order)
            FROM "Restaurant_schema"."Order"
            GROUP BY table_number) AS table_2) )
```

Наполнение таблицы до удаления данных:


```
select * from "Restaurant_schema"."Order"
```

	id_order [PK] integer	id_employee integer	date_order date	status_order character varying (11)	price_order numeric (9,2)	table_number integer
1	3	5	2023-03-19	accepted	3880.00	3
2	5	4	2023-03-19	paid	1950.00	1
3	21	5	2023-04-08	paid	2850.00	3
4	22	5	2023-04-08	accepted	1200.00	5
5	23	5	2023-04-08	paid	1400.00	4
6	24	5	2023-04-08	paid	1650.00	6
7	26	4	2023-04-08	paid	2700.00	4
8	27	4	2023-04-08	paid	1880.00	1
9	2	4	2023-04-08	accepted	1950.00	1
10	4	4	2023-04-08	in progress	1950.00	1
11	6	4	2023-04-08	accepted	1950.00	1
12	7	4	2023-04-08	accepted	2500.00	2
13	18	3	2023-04-08	in progress	2500.00	7
14	19	3	2023-04-08	accepted	2000.00	6
15	20	3	2023-04-08	in progress	1750.00	8
16	25	4	2023-04-08	in progress	900.00	2

Рисунок 8 – Наполнение таблицы до удаления

Наполнение таблицы после удаления данных:

```
select * from "Restaurant_schema"."Order"
```

	id_order [PK] integer	id_employee integer	date_order date	status_order character varying (11)	price_order numeric (9,2)	table_number integer
1	3	5	2023-03-19	accepted	3880.00	3
2	5	4	2023-03-19	paid	1950.00	1
3	21	5	2023-04-08	paid	2850.00	3
4	23	5	2023-04-08	paid	1400.00	4
5	24	5	2023-04-08	paid	1650.00	6
6	26	4	2023-04-08	paid	2700.00	4
7	27	4	2023-04-08	paid	1880.00	1
8	2	4	2023-04-08	accepted	1950.00	1
9	4	4	2023-04-08	in progress	1950.00	1
10	6	4	2023-04-08	accepted	1950.00	1
11	7	4	2023-04-08	accepted	2500.00	2
12	18	3	2023-04-08	in progress	2500.00	7
13	19	3	2023-04-08	accepted	2000.00	6
14	20	3	2023-04-08	in progress	1750.00	8
15	25	4	2023-04-08	in progress	900.00	2

Рисунок 8 – Наполнение таблицы после удаления

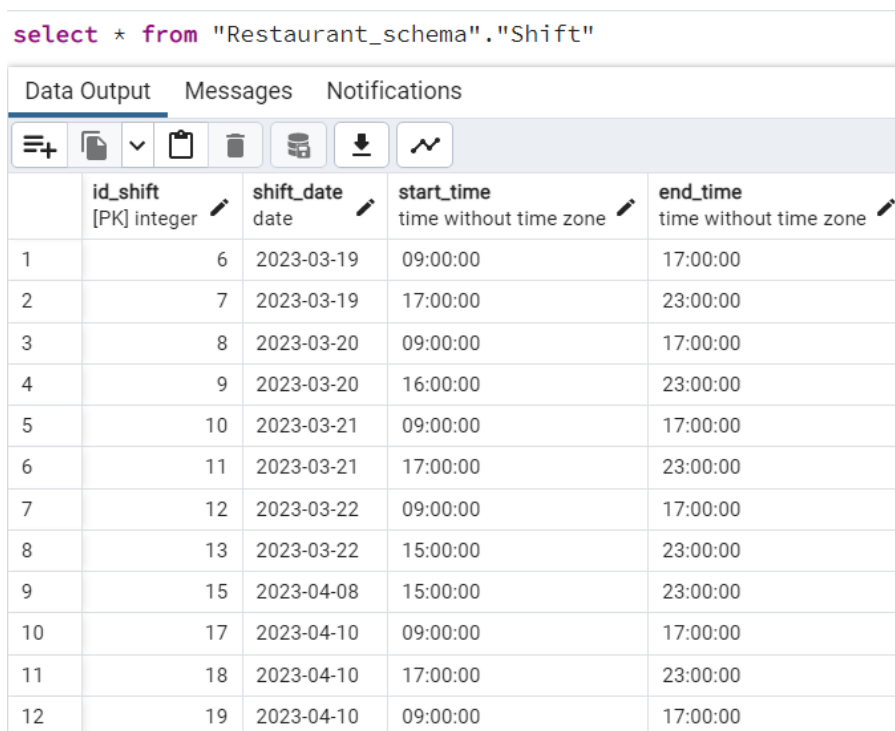
5.3 Insert с подзапросом

Запрос: вставить данные смены, в которой принимали участие сотрудники с номерами 1 или 6 или 5 и обслуживали столы 1 или 2 или 5 или 8, изменив дату на сегодняшнюю.

Код SQL:

```
insert into "Restaurant_schema"."Shift" (shift_date, start_time, end_time)
SELECT '2023-04-10', start_time, end_time
FROM "Restaurant_schema"."Shift"
WHERE id_shift IN
      (SELECT id_shift
       FROM "Restaurant_schema"."Shift_allocation"
       WHERE id_employee IN (1, 6, 5) and table_number in (1, 2, 5, 8))
```

Наполнение таблицы до вставки данных:



	id_shift [PK] integer	shift_date date	start_time time without time zone	end_time time without time zone
1	6	2023-03-19	09:00:00	17:00:00
2	7	2023-03-19	17:00:00	23:00:00
3	8	2023-03-20	09:00:00	17:00:00
4	9	2023-03-20	16:00:00	23:00:00
5	10	2023-03-21	09:00:00	17:00:00
6	11	2023-03-21	17:00:00	23:00:00
7	12	2023-03-22	09:00:00	17:00:00
8	13	2023-03-22	15:00:00	23:00:00
9	15	2023-04-08	15:00:00	23:00:00
10	17	2023-04-10	09:00:00	17:00:00
11	18	2023-04-10	17:00:00	23:00:00
12	19	2023-04-10	09:00:00	17:00:00

Рисунок 8 – Наполнение таблицы до вставки

Наполнение таблицы после вставки данных:

1 `select * from "Restaurant_schema"."Shift"`

Data Output Messages Notifications					
	id_shift [PK] integer	shift_date date	start_time time without time zone	end_time time without time zone	
1	6	2023-03-19	09:00:00	17:00:00	
2	7	2023-03-19	17:00:00	23:00:00	
3	8	2023-03-20	09:00:00	17:00:00	
4	9	2023-03-20	16:00:00	23:00:00	
5	10	2023-03-21	09:00:00	17:00:00	
6	11	2023-03-21	17:00:00	23:00:00	
7	12	2023-03-22	09:00:00	17:00:00	
8	13	2023-03-22	15:00:00	23:00:00	
9	15	2023-04-08	15:00:00	23:00:00	
10	17	2023-04-10	09:00:00	17:00:00	
11	18	2023-04-10	17:00:00	23:00:00	
12	19	2023-04-10	09:00:00	17:00:00	
13	20	2023-04-10	09:00:00	17:00:00	

Рисунок 8 – Наполнение таблицы после вставки

6 Запросы с индексами

6.1 Запрос с индексами 1

Запрос без индексов: запрос №

Код:

```
SELECT id_dish,  
       (SELECT dish_name  
        FROM "Restaurant_schema"."Dish"  
        WHERE "Restaurant_schema"."Dish".id_dish  
              "Restaurant_schema"."Dish_components".id_dish), count(distinct id_ingredient)  
FROM "Restaurant_schema"."Dish_components"  
GROUP BY id_dish
```

Планы запроса без индексов:

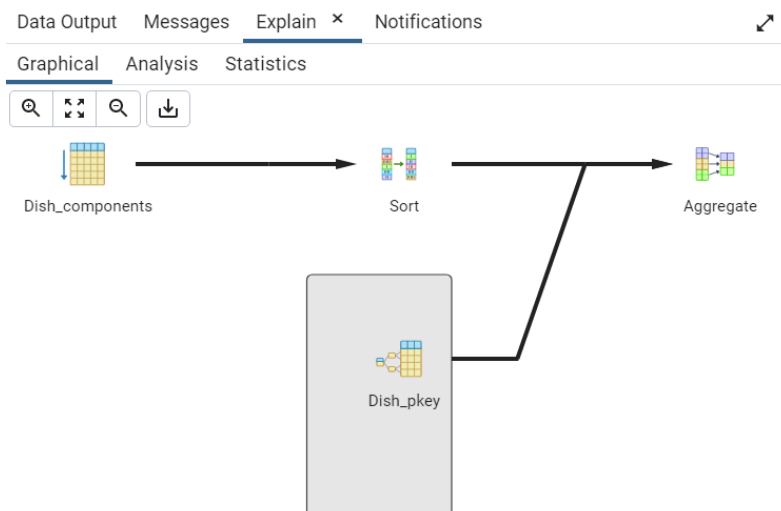


Рисунок 10 – Графический план запроса

QUERY PLAN		text	
1	GroupAggregate (cost=1.14..50.23 rows=6 width=110)		
2	Group Key: "Dish_components".id_dish		
3	-> Sort (cost=1.14..1.15 rows=6 width=8)		
4	Sort Key: "Dish_components".id_dish		
5	-> Seq Scan on "Dish_components" (cost=0.00..1.06 rows=6 width=8)		
6	SubPlan 1		
7	-> Index Scan using "Dish_pkey" on "Dish" (cost=0.15..8.17 rows=1 width=98)		
8	Index Cond: (id_dish = "Dish_components".id_dish)		

Рисунок 11 – Аналитический план запроса

Total rows: 8 of 8	Query complete 00:00:00.045
--------------------	-----------------------------

Рисунок 12 – Время запроса без индексов

Добавление индексов: CREATE index "index_id_dish" ON "Restaurant_schema"."Dish_components"(id_dish)

Планы запроса с индексами:

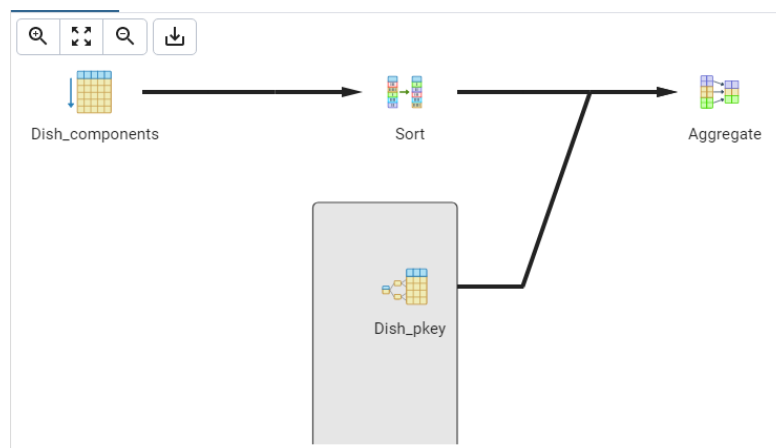


Рисунок 13 – Графический план запросов с индексами

Data Output	Messages	Explain	×	Notifications
<div> <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> <div> <div>QUERY PLAN</div> <div>text</div> <div>🔒</div> </div> </div>				
1	GroupAggregate (cost=1.23..74.88 rows=9 width=110)			
2	Group Key: "Dish_components".id_dish			
3	-> Sort (cost=1.23..1.26 rows=9 width=8)			
4	Sort Key: "Dish_components".id_dish			
5	-> Seq Scan on "Dish_components" (cost=0.00..1.09 rows=9 width=8)			
6	SubPlan 1			
7	-> Index Scan using "Dish_pkey" on "Dish" (cost=0.15..8.17 rows=1 width=98)			
8	Index Cond: (id_dish = "Dish_components".id_dish)			

Рисунок 14 – Аналитический план запроса с индексами

Total rows: 8 of 8	Query complete 00:00:00.075
--------------------	-----------------------------

Рисунок 15 – Время запросов с индексами

6.2 Запрос с индексами 2

Запрос: создание запроса для представления №2.

Код:

```
SELECT date_order, "Restaurant_schema"."Order_fulfillment".id_employee,  
      (SELECT surname  
       FROM "Restaurant_schema"."Employee"  
       WHERE "Restaurant_schema"."Order_fulfillment".id_employee =  
"Restaurant_schema"."Employee".id_employee), "Restaurant_schema"."Order_fulfillment".id_dish,  
      (SELECT dish_name  
       FROM "Restaurant_schema"."Dish"  
       WHERE "Restaurant_schema"."Order_fulfillment".id_dish =  
"Restaurant_schema"."Dish".id_dish),  
      (SELECT table1.count  
       FROM  
        (SELECT id_dish, id_employee, date_order, count(id_dish)  
         FROM "Restaurant_schema"."Order_fulfillment"  
         GROUP BY id_dish, id_employee, date_order) AS table1  
        WHERE table1.id_dish = "Restaurant_schema"."Order_fulfillment".id_dish  
        AND table1.id_employee = "Restaurant_schema"."Order_fulfillment".id_employee  
        AND table1.date_order = "Restaurant_schema"."Order".date_order)  
FROM "Restaurant_schema"."Order" JOIN "Restaurant_schema"."Order_fulfillment"  
ON "Restaurant_schema"."Order".id_order = "Restaurant_schema"."Order_fulfillment".id_order  
GROUP BY date_order,  
         "Restaurant_schema"."Order_fulfillment".id_employee,  
         "Restaurant_schema"."Order_fulfillment".id_dish;
```

Планы запроса без индексов:

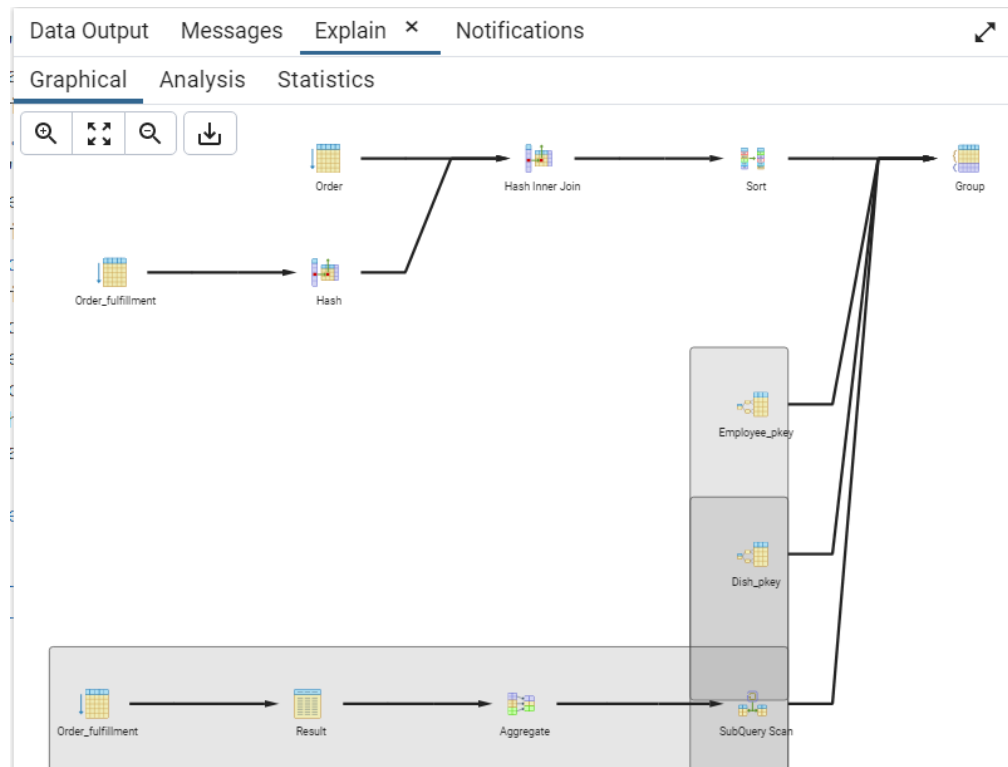


Рисунок 10 – Графический план запроса

	QUERY PLAN
	text
1	Group (cost=22.85..180.42 rows=9 width=176)
2	Group Key: "Order".date_order, "Order_fulfillment".id_employee, "Order_fulfillment".id_dish
3	-> Sort (cost=22.85..22.87 rows=9 width=12)
4	Sort Key: "Order".date_order, "Order_fulfillment".id_employee, "Order_fulfillment".id_dish
5	-> Hash Join (cost=1.20..22.71 rows=9 width=12)
6	Hash Cond: ("Order".id_order = "Order_fulfillment".id_order)
7	-> Seq Scan on "Order" (cost=0.00..18.30 rows=830 width=8)
8	-> Hash (cost=1.09..1.09 rows=9 width=12)
9	-> Seq Scan on "Order_fulfillment" (cost=0.00..1.09 rows=9 width=12)
10	SubPlan 1
11	-> Index Scan using "Employee_pkey" on "Employee" (cost=0.15..8.17 rows=1 width=58)
12	Index Cond: (id_employee = "Order_fulfillment".id_employee)
13	SubPlan 2
14	-> Index Scan using "Dish_pkey" on "Dish" (cost=0.15..8.17 rows=1 width=98)
15	Index Cond: (id_dish = "Order_fulfillment".id_dish)
16	SubPlan 3
17	-> Subquery Scan on table1 (cost=0.00..1.17 rows=1 width=8)
18	-> GroupAggregate (cost=0.00..1.16 rows=1 width=20)
19	Group Key: "Order_fulfillment_1".id_dish, "Order_fulfillment_1".id_employee, "Order".date_order
20	-> Result (cost=0.00..1.14 rows=1 width=12)
21	One-Time Filter: ("Order".date_order = "Order".date_order)
22	-> Seq Scan on "Order_fulfillment" "Order_fulfillment_1" (cost=0.00..1.14 rows=1 width=8)
23	Filter: ((id_dish = "Order_fulfillment".id_dish) AND (id_employee = "Order_fulfillment".id_employee))

Рисунок 11 – Аналитический план запроса

Total rows: 9 of 9

Query complete 00:00:00.108

Рисунок 12 – Время запроса без индексов

Добавление индексов: CREATE INDEX id_employee_dish on "Restaurant_schema"."Order_fulfillment"(id_employee, id_dish)

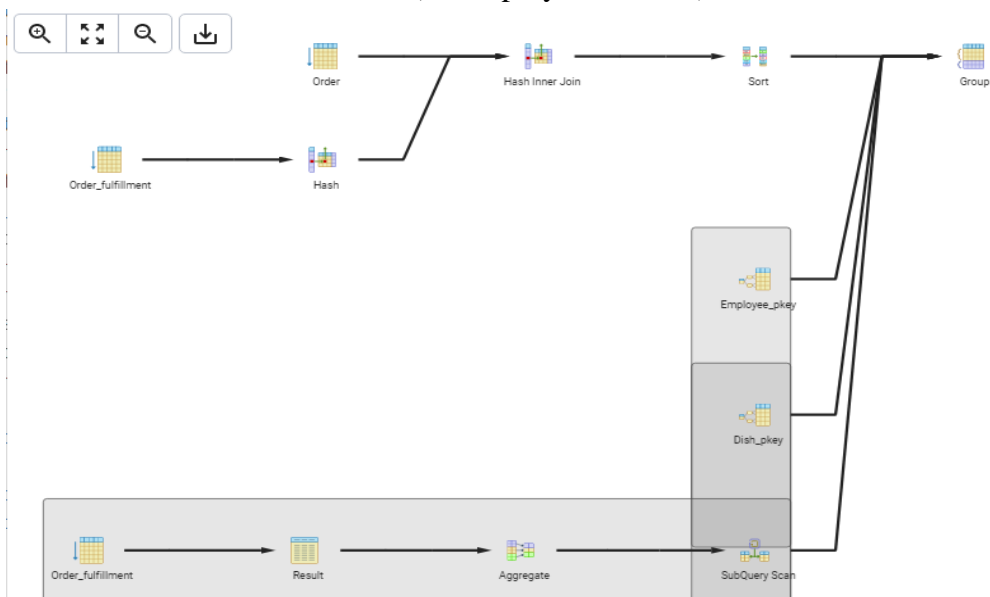


Рисунок 13 – Графический план запросов с индексами

QUERY PLAN	
text	
1	Group (cost=22.85..180.42 rows=9 width=176)
2	Group Key: "Order".date_order, "Order_fulfillment".id_employee, "Order_fulfillment".id_dish
3	-> Sort (cost=22.85..22.87 rows=9 width=12)
4	Sort Key: "Order".date_order, "Order_fulfillment".id_employee, "Order_fulfillment".id_dish
5	-> Hash Join (cost=1.20..22.71 rows=9 width=12)
6	Hash Cond: ("Order".id_order = "Order_fulfillment".id_order)
7	-> Seq Scan on "Order" (cost=0.00..18.30 rows=830 width=8)
8	-> Hash (cost=1.09..1.09 rows=9 width=12)
9	-> Seq Scan on "Order_fulfillment" (cost=0.00..1.09 rows=9 width=12)
10	SubPlan 1
11	-> Index Scan using "Employee_pkey" on "Employee" (cost=0.15..8.17 rows=1 width=58)
12	Index Cond: (id_employee = "Order_fulfillment".id_employee)
13	SubPlan 2
14	-> Index Scan using "Dish_pkey" on "Dish" (cost=0.15..8.17 rows=1 width=98)
15	Index Cond: (id_dish = "Order_fulfillment".id_dish)
16	SubPlan 3
17	-> Subquery Scan on table1 (cost=0.00..1.17 rows=1 width=8)
18	-> GroupAggregate (cost=0.00..1.16 rows=1 width=20)
19	Group Key: "Order_fulfillment_1".id_dish, "Order_fulfillment_1".id_employee, "Order".date_order
20	-> Result (cost=0.00..1.14 rows=1 width=12)
21	One-Time Filter: ("Order".date_order = "Order".date_order)
22	-> Seq Scan on "Order_fulfillment" "Order_fulfillment_1" (cost=0.00..1.14 rows=1 width=8)
23	Filter: ((id_dish = "Order_fulfillment".id_dish) AND (id_employee = "Order_fulfillment".id_employee))

Рисунок 14 – Аналитический план запроса с индексами

Total rows: 9 of 9	Query complete 00:00:00.067
--------------------	-----------------------------

Рисунок 15 – Время запросов с индексами

ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы были созданы запросы и представления на выборку данных к базе данных PostgreSQL согласно индивидуальному заданию, часть 2 и 3. Были созданы 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Были изучены графические представления запросов. Были созданы простой и составной индексы для двух произвольных запросов.

Таким образом, за выполнение данной лабораторной работы удалось познакомиться с представлениями и индексами и успешно их реализовать. Также были отработаны навыки выполнения запросов на индивидуальных заданиях. Успешно были реализованы различные модификации данных с подзапросами.

Индексы при больших запросах позволили значительно выиграть время выполнения, план запроса остался тем же.