

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

Отчет

по лабораторной работе №4 «Запросы на выборку и модификацию данных.
Представления. Работа с индексами»

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

Автор: Бархатова Н.А.

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.

ИТМО

Санкт-Петербург 2023

Оглавление

| | |
|---|----|
| Цель работы | 3 |
| Практическое задание | 3 |
| Схема базы данных: | 3 |
| Выполнение..... | 3 |
| Запросы к базе данных | 3 |
| 1. Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц..... | 4 |
| 2. Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа)..... | 4 |
| 3. Подсчитать, сколько ингредиентов содержит каждое блюдо..... | 5 |
| 4. Вывести название блюда, содержащее максимальное число ингредиентов. | 5 |
| 5. Какой повар может приготовить максимальное число видов блюд? | 7 |
| 6. Сколько закреплено столов за каждым из официантов? | 8 |
| 7. Какой из ингредиентов используется в максимальном количестве блюд? | 9 |
| Представления..... | 10 |
| 1. Для расчета стоимости ингредиентов для заданного блюда..... | 10 |
| 2. Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату. | 11 |
| Запросы на модификацию данных | 14 |
| 1. INSERT (добавление новых данных в таблицу order) | 14 |
| 2. UPDATE (изменение данных в таблице order) | 15 |
| 3. DELETE (удаление данных из таблицы order) | 18 |
| Создание индексов | 20 |
| 1. Простой индекс..... | 20 |
| 2. Составной индекс | 21 |
| Вывод..... | 23 |

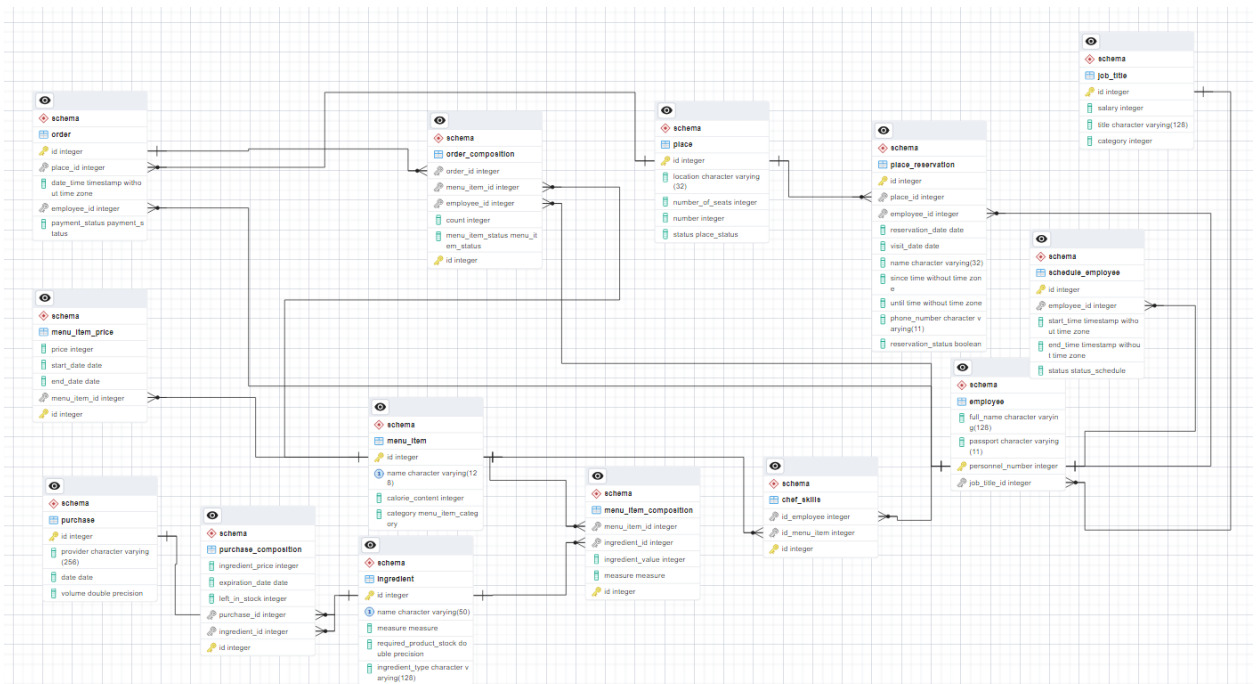
Цель работы

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.

Схема базы данных:



Выполнение

Запросы к базе данных

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

```
SELECT
    e.full_name AS waiter_name,
    SUM (mip.price) AS total_order_amount
FROM
    schema.order o
JOIN
    schema.employee e ON o.employee_id = e.personnel_number
JOIN
    schema.order_composition oc ON o.id = oc.order_id
JOIN
    schema.menu_item_price mip ON oc.menu_item_id = mip.menu_item_id
WHERE
    o.date_time >= date_trunc('month', CURRENT_DATE) - interval '1 month'
    AND o.date_time < date_trunc('month', CURRENT_DATE) AND
    o.payment_status = 'Paid'
GROUP BY
    e.full_name
HAVING
    SUM (mip.price) = (SELECT MAX(total_order_amount) FROM (SELECT
e.full_name AS waiter_name, SUM (mip.price) AS total_order_amount
                        FROM schema.order o
                        JOIN schema.employee e ON o.employee_id =
e.personnel_number
                        JOIN schema.order_composition oc ON o.id =
oc.order_id
                        JOIN schema.menu_item_price mip ON oc.menu_item_id
= mip.menu_item_id
                        WHERE o.date_time >= date_trunc('month',
CURRENT_DATE) - interval '1 month'
                        AND o.date_time < date_trunc('month',
CURRENT_DATE) AND o.payment_status = 'Paid'
                        GROUP BY e.full_name) AS subquery)
ORDER BY
    total_order_amount DESC;
```

| | waiter_name | | total_order_amount | | |
|---|--------------------------------|--|--------------------|--|--|
| | character varying (128) | | bigint | | |
| 1 | Николаева Татьяна Владимировна | | 43205 | | |

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

```
SELECT
    e.full_name AS waiter_name,
    SUM (mip.price * 0.05) AS total_bonus
FROM
    schema.order o
JOIN
```

```

    schema.employee e ON o.employee_id = e.personnel_number
JOIN
    schema.order_composition oc ON o.id = oc.order_id
JOIN
    schema.menu_item_price mip ON oc.menu_item_id = mip.menu_item_id
WHERE
    DATE_TRUNC ('day', o.date_time) >= (CURRENT_DATE - interval '39 day')
AND o.payment_status = 'Paid'
GROUP BY
    e.personnel_number
ORDER BY
    total_bonus DESC;

```

Data Output Сообщения Notifications

| | waiter_name character varying (128) | total_bonus numeric |
|---|--|------------------------|
| 1 | Николаева Татьяна Владимировна | 2160.25 |
| 2 | Смирнова Ольга Ивановна | 2085.60 |
| 3 | Козлова-Коровкина Анна Николаевна | 1421.20 |

3. Подсчитать, сколько ингредиентов содержит каждое блюдо.

```

SELECT
    mic.menu_item_id,
    mi.name AS menu_item_name,
    COUNT(mic.ingredient_id) AS num_ingredients
FROM
    schema.menu_item_composition mic
JOIN
    schema.menu_item mi ON mic.menu_item_id = mi.id
GROUP BY
    mic.menu_item_id, mi.name
ORDER BY
    mic.menu_item_id;

```

```

1 SELECT
2     mic.menu_item_id,
3     mi.name AS menu_item_name,
4     COUNT(mic.ingredient_id) AS num_ingredients
5 FROM
6     schema.menu_item_composition mic
7 JOIN
8     schema.menu_item mi ON mic.menu_item_id = mi.id
9 GROUP BY
10    mic.menu_item_id, mi.name
11 ORDER BY
12    mic.menu_item_id;
13

```

| | menu_item_id integer | menu_item_name character varying (128) | num_ingredients bigint |
|----|-------------------------|---|---------------------------|
| 1 | 1 | Картофельное пюре с говяжьим стейком | 5 |
| 2 | 2 | Жареный картофель с капустой и свиной | 5 |
| 3 | 3 | Морковный суп с курицей | 4 |
| 4 | 4 | Салат с креветками и оливковым маслом | 3 |
| 5 | 5 | Лосось с овощами на гриле | 5 |
| 6 | 6 | Гречка с сыром и орехами | 3 |
| 7 | 7 | Рис с овощами и курицей по-тайски | 6 |
| 8 | 8 | Банановый десерт с медом и орехами | 3 |
| 9 | 9 | Пшенная каша с яблоками и корицей | 3 |
| 10 | 10 | Овсяный завтрак с фруктами и сметаной | 5 |
| 11 | 11 | Сельдь под шубой | 4 |
| 12 | 12 | Куриные котлеты с овощами на гриле | 5 |
| 13 | 13 | Паста с креветками в сливочном соусе | 3 |
| 14 | 14 | Салат из свежих овощей с оливковым маслом | 3 |
| 15 | 15 | Фруктовый салат с йогуртом и орехами | 3 |

4. Вывести название блюда, содержащее максимальное число ингредиентов.

```

SELECT
    mi.name AS menu_item_name,

```



```

1 SELECT
2     mi.name AS menu_item_name,
3     COUNT(mic.ingredient_id) AS num_ingredients
4 FROM
5     schema.menu_item_composition mic
6 JOIN
7     schema.menu_item mi ON mic.menu_item_id = mi.id
8 GROUP BY
9     mi.name
10 HAVING
11     COUNT(mic.ingredient_id) = (SELECT MAX(num_ingredients) FROM (SELECT mi.name AS menu_item_name, COUNT(mic.ingredient_id) AS num_ingredients
12 FROM schema.menu_item_composition mic
13 JOIN schema.menu_item mi ON mic.menu_item_id = mi.id
14 GROUP BY mi.id, mi.name) AS subquery)
15 ORDER BY
16     num_ingredients DESC;
17

```

Data Output Сообщения Notifications

| | menu_item_name character varying (128) | num_ingredients bigint |
|---|---|---------------------------|
| 1 | Рис с овощами и курицей по-тайски | 6 |

```

1 SELECT
2     mi.name AS menu_item_name,
3     COUNT(mic.ingredient_id) AS num_ingredients
4 FROM
5     schema.menu_item_composition mic
6 JOIN
7     schema.menu_item mi ON mic.menu_item_id = mi.id
8 GROUP BY
9     mi.name
10 HAVING
11     COUNT(mic.ingredient_id) = (SELECT MAX(num_ingredients) FROM (SELECT mi.name AS menu_item_name, COUNT(mic.ingredient_id) AS num_ingredients
12 FROM schema.menu_item_composition mic
13 JOIN schema.menu_item mi ON mic.menu_item_id = mi.id
14 GROUP BY mi.name) AS subquery)
15 ORDER BY
16     num_ingredients DESC;
17

```

Data Output Сообщения Notifications

| | menu_item_name character varying (128) | num_ingredients bigint |
|--|---|---------------------------|
| | Рис с овощами и курицей по-тайски | 6 |

5. Какой повар может приготовить максимальное число видов блюд?

```

SELECT e.full_name, COUNT(cs.id_menu_item) as dish_count
FROM schema.chef_skills cs
JOIN schema.employee e ON e.personnel_number = cs.id_employee
GROUP BY e.personnel_number, e.full_name
HAVING COUNT(cs.id_menu_item) = (
    SELECT MAX(dish_count)
    FROM (
        SELECT COUNT(id_menu_item) as dish_count
        FROM schema.chef_skills
        GROUP BY id_employee
    ) max_dish_count
);

```

```

1 SELECT e.full_name, COUNT(cs.id_menu_item) as dish_count
2 FROM schema.chef_skills cs
3 JOIN schema.employee e ON e.personnel_number = cs.id_employee
4 GROUP BY e.personnel_number, e.full_name
5 HAVING COUNT(cs.id_menu_item) = (
6     SELECT MAX(dish_count)
7     FROM (
8         SELECT COUNT(id_menu_item) as dish_count
9         FROM schema.chef_skills
10        GROUP BY id_employee
11     ) max_dish_count
12 );
13

```

Data Output Сообщения Notifications

| | full_name character varying (128) | dish_count bigint |
|---|--------------------------------------|----------------------|
| 1 | Иванова Мария Петровна | 4 |

6. Сколько закреплено столов за каждым из официантов?

```

SELECT
    e.full_name AS waiter_name,
    COUNT(DISTINCT o.place_id) AS num_tables_assigned
FROM
    schema.order o
JOIN
    schema.employee e ON o.employee_id = e.personnel_number
GROUP BY
    e.personnel_number, e.full_name
ORDER BY
    num_tables_assigned DESC;

```



```

1  SELECT
2      e.full_name AS waiter_name,
3      COUNT(DISTINCT o.place_id) AS num_tables_assigned
4  FROM
5      schema.order o
6  JOIN
7      schema.employee e ON o.employee_id = e.personnel_number
8  GROUP BY
9      e.personnel_number, e.full_name
10 ORDER BY
11     num_tables_assigned DESC;

```

Data Output Сообщения Notifications



| | waiter_name character varying (128) | num_tables_assigned bigint |
|---|--|-------------------------------|
| 1 | Козлова-Коровкина Анна Николаевна | 5 |
| 2 | Николаева Татьяна Владимировна | 5 |
| 3 | Смирнова Ольга Ивановна | 4 |

7. Какой из ингредиентов используется в максимальном количестве блюд?

```

SELECT
    i.name,
    COUNT(DISTINCT mic.menu_item_id) AS num_dishes
FROM
    schema.menu_item_composition mic
JOIN
    schema.ingredient i ON i.id = mic.ingredient_id
GROUP BY
    i.id, i.name
HAVING
    COUNT(DISTINCT mic.menu_item_id) = (SELECT MAX(num_dishes) FROM (SELECT
i.name, COUNT(DISTINCT mic.menu_item_id) AS num_dishes
                                FROM schema.menu_item_composition mic
                                JOIN schema.ingredient i ON i.id = mic.ingredient_id
                                GROUP BY i.id, i.name) AS subquery)
ORDER BY
    num_dishes DESC;

```

```

1 SELECT
2     i.name,
3     COUNT(DISTINCT mic.menu_item_id) AS num_dishes
4 FROM
5     schema.menu_item_composition mic
6 JOIN
7     schema.ingredient i ON i.id = mic.ingredient_id
8 GROUP BY
9     i.id, i.name
10 HAVING
11     COUNT(DISTINCT mic.menu_item_id) = (SELECT MAX(num_dishes) FROM (SELECT i.name, COUNT(DISTINCT mic.menu_item_id) AS num_dishes
12 FROM schema.menu_item_composition mic
13 JOIN schema.ingredient i ON i.id = mic.ingredient_id
14 GROUP BY i.id, i.name) AS subquery)
15 ORDER BY
16     num_dishes DESC;

```

Data Output Сообщения Notifications

| | name character varying (50) | num_dishes bigint |
|---|--------------------------------|----------------------|
| 1 | Соль | 7 |

Представления

1. Для расчета стоимости ингредиентов для заданного блюда

```
CREATE VIEW schema.view_menu_item_cost AS
```

```
SELECT
```

```
    mic.menu_item_id,
```

```
    mi.name AS menu_item_name,
```

```
    SUM (pc.ingredient_price * mic.ingredient_value) AS total_cost
```

```
FROM
```

```
    schema.menu_item_composition mic
```

```
JOIN
```

```
    schema.menu_item mi ON mic.menu_item_id = mi.id
```

```
JOIN
```

```
    schema.purchase_composition pc ON mic.ingredient_id = pc.ingredient_id
```

```
GROUP BY
```

```
    mic.menu_item_id, mi.name;
```

```

1 CREATE VIEW schema.view_menu_item_cost AS
2 SELECT
3     mic.menu_item_id,
4     mi.name AS menu_item_name,
5     SUM(pc.ingredient_price * mic.ingredient_value) AS total_cost
6 FROM
7     schema.menu_item_composition mic
8 JOIN
9     schema.menu_item mi ON mic.menu_item_id = mi.id
10 JOIN
11     schema.purchase_composition pc ON mic.ingredient_id = pc.ingredient_id
12 GROUP BY
13     mic.menu_item_id, mi.name;
14

```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 392 msec.

```

1 SELECT * FROM schema.view_menu_item_cost;

```

Data Output Сообщения Notifications

| | menu_item_id integer | menu_item_name character varying (128) | total_cost double precision |
|----|-------------------------|---|--------------------------------|
| 1 | 8 | Банановый десерт с медом и орехами | 40.349999999999994 |
| 2 | 11 | Сельдь под шубой | 165.19 |
| 3 | 1 | Картофельное пюре с говяжьим стейком | 300.38 |
| 4 | 14 | Салат из свежих овощей с оливковым маслом | 37.5 |
| 5 | 4 | Салат с креветками и оливковым маслом | 240.06 |
| 6 | 5 | Лосось с овощами на гриле | 765.34 |
| 7 | 7 | Рис с овощами и курицей по-тайски | 265.29 |
| 8 | 3 | Морковный суп с курицей | 231.65 |
| 9 | 15 | Фруктовый салат с йогуртом и орехами | 173 |
| 10 | 10 | Овсяный завтрак с фруктами и сметаной | 129 |
| 11 | 12 | Куриные котлеты с овощами на гриле | 265.29 |
| 12 | 9 | Пшениная каша с яблоками и корицей | 160.04 |
| 13 | 13 | Паста с креветками в сливочном соусе | 420.17999999999995 |
| 14 | 2 | Жареный картофель с капустой и свиной | 425.34 |
| 15 | 6 | Гречка с сыром и орехами | 50.1 |

- Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату.

CREATE VIEW schema.view_chef_dishes_count AS

SELECT

```

e.full_name AS chef_name,

mi.name AS menu_item_name,

COUNT(o.id) AS dishes_count

FROM

schema.order o

JOIN

schema.order_composition oc ON o.id = oc.order_id

JOIN

schema.employee e ON oc.employee_id = e.personnel_number

JOIN

schema.menu_item mi ON oc.menu_item_id = mi.id

WHERE

DATE_TRUNC('day', o.date_time) = '2023-11-01'::date

GROUP BY

e.full_name, mi.name;

```

```

1 CREATE VIEW schema.view_chef_dishes_count AS
2 SELECT
3     e.full_name AS chef_name,
4     mi.name AS menu_item_name,
5     COUNT(o.id) AS dishes_count
6 FROM
7     schema.order o
8 JOIN
9     schema.order_composition oc ON o.id = oc.order_id
10 JOIN
11     schema.employee e ON oc.employee_id = e.personnel_number
12 JOIN
13     schema.menu_item mi ON oc.menu_item_id = mi.id
14 WHERE
15     DATE_TRUNC('day', o.date_time) = '2023-11-01'::date
16 GROUP BY
17     e.full_name, mi.name;

```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 294 msec.

1

SELECT * FROM schema.view_chef_dishes_count;

2

Data Output

Сообщения

Notifications

chef_name

character varying (128)

menu_item_name

character varying (128)

dishes_count

bigint

1

Григорьева Мария Сергеевна

Овсяный завтрак с фруктами и сметаной

8

2

Григорьева Мария Сергеевна

Пшенная каша с яблоками и корицей

3

3

Григорьева Мария Сергеевна

Сельдь под шубой

1

4

Дмитриева Ольга Игоревна

Куриные котлеты с овощами на гриле

5

5

Дмитриева Ольга Игоревна

Паста с креветками в сливочном соусе

8

6

Дмитриева Ольга Игоревна

Салат из свежих овощей с оливковым маслом

3

7

Иванова Мария Петровна

Гречка с сыром и орехами

3

8

Иванова Мария Петровна

Лосось с овощами на гриле

2

9

Иванова Мария Петровна

Рис с овощами и курицей по-тайски

4

10

Иванова Мария Петровна

Салат с креветками и оливковым маслом

2

11

Кузнецова Анна Сергеевна

Жареный картофель с капустой и свининой

6

12

Кузнецова Анна Сергеевна

Картофельное пюре с говяжьим стейком

8

13

Кузнецова Анна Сергеевна

Морковный суп с курицей

3

14

Петрова Екатерина Александровна

Банановый десерт с медом и орехами

1

15

Петрова Екатерина Александровна

Фруктовый салат с йогуртом и орехами

2

Запросы на модификацию данных

1. INSERT (добавление новых данных в таблицу order)

1SELECT * FROM schema.order

Data Output

Сообщения

Notifications

| | id [PK] integer | place_id integer | date_time timestamp without time zone | employee_id integer | payment_status character varying (32) |
|----|--------------------|---------------------|--|------------------------|--|
| 1 | 98 | 1 | 2023-11-01 12:30:36 | 14 | Paid |
| 2 | 99 | 1 | 2023-11-01 14:30:12 | 14 | Paid |
| 3 | 100 | 1 | 2023-11-01 16:30:00 | 14 | Paid |
| 4 | 101 | 1 | 2023-11-01 17:30:50 | 14 | Paid |
| 5 | 102 | 1 | 2023-11-01 18:09:34 | 14 | Paid |
| 6 | 103 | 1 | 2023-11-01 21:30:00 | 14 | Cancelled |
| 7 | 104 | 5 | 2023-11-01 14:50:03 | 14 | Paid |
| 8 | 105 | 5 | 2023-11-01 18:10:38 | 14 | Cancelled |
| 9 | 106 | 5 | 2023-11-01 21:30:49 | 14 | Paid |
| 10 | 107 | 2 | 2023-11-01 12:32:00 | 16 | Paid |
| 11 | 108 | 2 | 2023-11-01 15:50:00 | 16 | Cancelled |
| 12 | 109 | 2 | 2023-11-01 20:30:03 | 16 | Paid |
| 13 | 110 | 2 | 2023-11-01 22:30:00 | 16 | Paid |
| 14 | 111 | 4 | 2023-11-01 13:19:54 | 16 | Paid |
| 15 | 112 | 4 | 2023-11-01 18:58:45 | 16 | Paid |
| 16 | 113 | 4 | 2023-11-01 21:59:01 | 16 | Paid |
| 17 | 114 | 3 | 2023-11-01 12:09:04 | 16 | Paid |
| 18 | 115 | 3 | 2023-11-01 14:28:13 | 16 | Paid |
| 19 | 116 | 3 | 2023-11-01 17:59:54 | 16 | Paid |
| 20 | 117 | 3 | 2023-11-01 20:58:06 | 16 | Paid |
| 21 | 118 | 2 | 2023-11-02 13:30:36 | 15 | Paid |

| | |
|--|--|
| 1 | INSERT INTO schema.order (place_id, date_time, employee_id) |
| 2 | VALUES (|
| 3 | (SELECT id FROM schema.place WHERE location = 'На террасе'), |
| 4 | '2023-12-01 13:30:00', |
| 5 | (SELECT personnel_number FROM schema.employee WHERE full_name = 'Смирнова Ольга Ивановна') |
| 6 |); |
| 7 | |
| 8 | |
| Data Output Сообщения Notifications | |
| INSERT 0 1 | |
| Запрос завершён успешно, время выполнения: 122 мсек. | |

Scratch Pad Запрос История запросов

1 SELECT * FROM schema.order
2

Data Output Сообщения Notifications

| | id [PK] integer | place_id integer | date_time timestamp without time zone | employee_id integer | payment_status character varying (32) |
|----|--------------------|---------------------|--|------------------------|--|
| 1 | 194 | 5 | 2023-12-01 13:30:00 | 14 | Unpaid |
| 2 | 98 | 1 | 2023-11-01 12:30:36 | 14 | Paid |
| 3 | 99 | 1 | 2023-11-01 14:30:12 | 14 | Paid |
| 4 | 100 | 1 | 2023-11-01 16:30:00 | 14 | Paid |
| 5 | 101 | 1 | 2023-11-01 17:30:50 | 14 | Paid |
| 6 | 102 | 1 | 2023-11-01 18:09:34 | 14 | Paid |
| 7 | 103 | 1 | 2023-11-01 21:30:00 | 14 | Cancelled |
| 8 | 104 | 5 | 2023-11-01 14:50:03 | 14 | Paid |
| 9 | 105 | 5 | 2023-11-01 18:10:38 | 14 | Cancelled |
| 10 | 106 | 5 | 2023-11-01 21:30:49 | 14 | Paid |
| 11 | 107 | 2 | 2023-11-01 12:32:00 | 16 | Paid |
| 12 | 108 | 2 | 2023-11-01 15:50:00 | 16 | Cancelled |
| 13 | 109 | 2 | 2023-11-01 20:30:03 | 16 | Paid |
| 14 | 110 | 2 | 2023-11-01 22:30:00 | 16 | Paid |
| 15 | 111 | 4 | 2023-11-01 13:19:54 | 16 | Paid |
| 16 | 112 | 4 | 2023-11-01 18:58:45 | 16 | Paid |
| 17 | 113 | 4 | 2023-11-01 21:59:01 | 16 | Paid |
| 18 | 114 | 3 | 2023-11-01 12:09:04 | 16 | Paid |
| 19 | 115 | 3 | 2023-11-01 14:28:13 | 16 | Paid |
| 20 | 116 | 3 | 2023-11-01 17:59:54 | 16 | Paid |
| 21 | 117 | 3 | 2023-11-01 22:59:06 | 16 | Paid |

2. UPDATE (изменение данных в таблице order)

Scratch Pad Запрос История запросов

1 UPDATE schema.order
2 SET payment_status = 'Cancelled'
3 WHERE employee_id IN (SELECT personnel_number FROM schema.employee WHERE full_name LIKE '%Ольга%');

Data Output Сообщения Notifications






UPDATE 35

Запрос завершён успешно, время выполнения: 490 мсек.

```
1 SELECT * FROM schema.order
```

Data Output Сообщения Notifications

| | id [PK] integer | place_id integer | date_time timestamp without time zone | employee_id integer | payment_status character varying (32) |
|----|--------------------|---------------------|--|------------------------|--|
| 1 | 107 | 2 | 2023-11-01 12:32:00 | 16 | Paid |
| 2 | 108 | 2 | 2023-11-01 15:50:00 | 16 | Paid |
| 3 | 109 | 2 | 2023-11-01 20:30:03 | 16 | Paid |
| 4 | 110 | 2 | 2023-11-01 22:30:00 | 16 | Paid |
| 5 | 111 | 4 | 2023-11-01 13:19:54 | 16 | Paid |
| 6 | 112 | 4 | 2023-11-01 18:58:45 | 16 | Paid |
| 7 | 113 | 4 | 2023-11-01 21:59:01 | 16 | Paid |
| 8 | 114 | 3 | 2023-11-01 12:09:04 | 16 | Paid |
| 9 | 115 | 3 | 2023-11-01 14:28:13 | 16 | Paid |
| 10 | 116 | 3 | 2023-11-01 17:59:54 | 16 | Paid |
| 11 | 117 | 3 | 2023-11-01 20:58:06 | 16 | Paid |
| 12 | 118 | 2 | 2023-11-02 13:30:36 | 15 | Paid |
| 13 | 119 | 2 | 2023-11-02 14:30:12 | 15 | Paid |
| 14 | 120 | 2 | 2023-11-02 16:30:00 | 15 | Paid |
| 15 | 121 | 2 | 2023-11-02 17:30:50 | 15 | Paid |
| 16 | 122 | 2 | 2023-11-02 18:09:34 | 15 | Paid |
| 17 | 123 | 4 | 2023-11-02 21:30:00 | 15 | Paid |
| 18 | 124 | 4 | 2023-11-02 14:50:03 | 15 | Paid |
| 19 | 125 | 4 | 2023-11-02 18:10:38 | 15 | Paid |
| 20 | 126 | 4 | 2023-11-02 21:30:49 | 15 | Paid |

| | id [PK] integer  | place_id integer  | date_time timestamp without time zone  | employee_id integer  | payment_status character varying (32)  |
|----|--|---|--|--|--|
| 29 | 135 | 5 | 2023-11-02 12:39:54 | 16 | Paid |
| 30 | 136 | 3 | 2023-11-02 18:49:54 | 16 | Paid |
| 31 | 98 | 1 | 2023-11-01 12:30:36 | 14 | Cancelled |
| 32 | 99 | 1 | 2023-11-01 14:30:12 | 14 | Cancelled |
| 33 | 100 | 1 | 2023-11-01 16:30:00 | 14 | Cancelled |
| 34 | 101 | 1 | 2023-11-01 17:30:50 | 14 | Cancelled |
| 35 | 102 | 1 | 2023-11-01 18:09:34 | 14 | Cancelled |
| 36 | 103 | 1 | 2023-11-01 21:30:00 | 14 | Cancelled |
| 37 | 104 | 5 | 2023-11-01 14:50:03 | 14 | Cancelled |
| 38 | 105 | 5 | 2023-11-01 18:10:38 | 14 | Cancelled |
| 39 | 106 | 5 | 2023-11-01 21:30:49 | 14 | Cancelled |
| 40 | 185 | 1 | 2023-11-05 13:30:36 | 14 | Cancelled |
| 41 | 186 | 1 | 2023-11-05 14:33:12 | 14 | Cancelled |
| 42 | 187 | 1 | 2023-11-05 20:30:07 | 14 | Cancelled |
| 43 | 188 | 5 | 2023-11-05 12:30:50 | 14 | Cancelled |
| 44 | 189 | 5 | 2023-11-05 14:09:34 | 14 | Cancelled |
| 45 | 190 | 5 | 2023-11-05 16:03:00 | 14 | Cancelled |
| 46 | 191 | 5 | 2023-11-05 20:09:03 | 14 | Cancelled |
| 47 | 192 | 5 | 2023-11-05 21:10:38 | 14 | Cancelled |
| 48 | 138 | 1 | 2023-11-03 13:10:36 | 14 | Cancelled |

3. DELETE (удаление данных из таблицы order)

```
1 SELECT * FROM schema.schedule_employee
```

Data Output Сообщения Notifications

| | | | | | |
|--|---------------------------------------|---|--|--|--|
| <div><div><div><div>≡</div><div>+</div></div><div><div>📄</div><div>▼</div></div><div><div>📄</div><div>▼</div></div><div><div>🗑️</div></div><div><div>🗄️</div></div><div><div>⬇️</div></div><div><div>📈</div></div></div></div> | | | | | |
| | <div>id</div> <div>[PK] integer</div> | <div>employee_id</div> <div>integer</div> | <div>start_time</div> <div>timestamp without time zone</div> | <div>end_time</div> <div>timestamp without time zone</div> | <div>status</div> <div>status_schedule</div> |
| 13 | 53 | 13 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 14 | 54 | 13 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 15 | 55 | 13 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Больничный |
| 16 | 56 | 14 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 17 | 57 | 14 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 18 | 58 | 14 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Вышел |
| 19 | 59 | 15 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 20 | 60 | 15 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 21 | 61 | 16 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 22 | 62 | 16 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 23 | 63 | 16 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 24 | 64 | 16 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | В отпуске |
| 25 | 65 | 16 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | В отпуске |
| 26 | 66 | 17 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 27 | 67 | 17 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 28 | 68 | 17 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 29 | 69 | 17 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 30 | 70 | 17 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Вышел |
| 31 | 71 | 18 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 32 | 72 | 18 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| Total rows: 40 of 40 | | Query complete 00:00:00.154 | | | |

```
1 DELETE FROM schema.schedule_employee
2 WHERE employee_id IN (SELECT personnel_number FROM schema.employee WHERE full_name LIKE '%Владимировна%');
```

Data Output Сообщения Notifications

DELETE 5

Запрос завершён успешно, время выполнения: 609 msec.

```
1 SELECT * FROM schema.schedule_employee
```

Data Output Сообщения Notifications

| | id [PK] integer | employee_id integer | start_time timestamp without time zone | end_time timestamp without time zone | status status_schedule |
|---|--------------------|------------------------|---|---|---------------------------|
| 17 | 57 | 14 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 18 | 58 | 14 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Вышел |
| 19 | 59 | 15 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 20 | 60 | 15 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 21 | 66 | 17 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 22 | 67 | 17 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 23 | 68 | 17 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 24 | 69 | 17 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 25 | 70 | 17 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Вышел |
| 26 | 71 | 18 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| 27 | 72 | 18 | 2023-11-02 12:00:00 | 2023-11-02 23:00:00 | Вышел |
| 28 | 73 | 18 | 2023-11-03 12:00:00 | 2023-11-03 23:00:00 | Вышел |
| 29 | 74 | 18 | 2023-11-04 12:00:00 | 2023-11-04 23:00:00 | Вышел |
| 30 | 75 | 18 | 2023-11-05 12:00:00 | 2023-11-05 23:00:00 | Вышел |
| 31 | 76 | 19 | 2023-11-01 12:00:00 | 2023-11-01 23:00:00 | Вышел |
| Total rows: 35 of 35 Query complete 00:00:00.122 | | | | | |

✓ Запрос выполнен успешно. Общее время

```
1 SELECT * FROM schema.order|
2
3
4
```

Data Output Сообщения Notifications

| | id [PK] integer | place_id integer | date_time timestamp without time zone | employee_id integer | payment_status character varying (32) |
|----|--------------------|---------------------|--|------------------------|--|
| 1 | 98 | 1 | 2023-11-01 12:30:36 | 14 | Paid |
| 2 | 99 | 1 | 2023-11-01 14:30:12 | 14 | Paid |
| 3 | 100 | 1 | 2023-11-01 16:30:00 | 14 | Paid |
| 4 | 101 | 1 | 2023-11-01 17:30:50 | 14 | Paid |
| 5 | 102 | 1 | 2023-11-01 18:09:34 | 14 | Paid |
| 6 | 103 | 1 | 2023-11-01 21:30:00 | 14 | Cancelled |
| 7 | 104 | 5 | 2023-11-01 14:50:03 | 14 | Paid |
| 8 | 105 | 5 | 2023-11-01 18:10:38 | 14 | Cancelled |
| 9 | 106 | 5 | 2023-11-01 21:30:49 | 14 | Paid |
| 10 | 107 | 2 | 2023-11-01 12:32:00 | 16 | Paid |
| 11 | 108 | 2 | 2023-11-01 15:50:00 | 16 | Cancelled |
| 12 | 109 | 2 | 2023-11-01 20:30:03 | 16 | Paid |
| 13 | 110 | 2 | 2023-11-01 22:30:00 | 16 | Paid |
| 14 | 111 | 4 | 2023-11-01 13:19:54 | 16 | Paid |
| 15 | 112 | 4 | 2023-11-01 18:58:45 | 16 | Paid |
| 16 | 113 | 4 | 2023-11-01 21:59:01 | 16 | Paid |
| 17 | 114 | 3 | 2023-11-01 12:09:04 | | |
| 18 | 115 | 3 | 2023-11-01 14:28:13 | | |


Создание индексов

1. Простой индекс

До создания индекса:

```
1 SELECT name FROM schema.ingredient WHERE ingredient_type = 'Овощи'
```

Data Output Сообщения Notifications

| | name character varying (50)  |
|---|--|
| 1 | Картофель |
| 2 | Капуста |
| 3 | Морковь |
| 4 | Лук |

```
1 SELECT name FROM schema.ingredient WHERE ingredient_type = 'Овощи'
2
3
```

Data Output Сообщения Notifications

Запрос выполнен успешно. Общее время выполнения: 113 msec.
обработано строк: 4.

Создадим индекс:

```
1 CREATE INDEX idx_ingredient_type_search
2 ON schema.ingredient (ingredient_type);
3
```

Data Output Сообщения Notifications

CREATE INDEX

Запрос завершён успешно, время выполнения: 136 msec.

После создания индекса:

```
1 SELECT name from schema.ingredient WHERE ingredient_type = 'Овощи'
```

Data Output Сообщения Notifications

Запрос выполнен успешно. Общее время выполнения: 105 msec.
обработано строк: 4.

2. Составной индекс

До создания индекса:

```

1 SELECT
2     mic.menu_item_id,
3     mi.name AS menu_item_name,
4     COUNT(mic.ingredient_id) AS num_ingredients
5 FROM
6     schema.menu_item_composition mic
7 JOIN
8     schema.menu_item mi ON mic.menu_item_id = mi.id
9 GROUP BY
10    mic.menu_item_id, mi.name
11 ORDER BY
12    mic.menu_item_id;

```

Data Output **Сообщения** План выполнения × Graph Visualiser × Notifications

Запрос выполнен успешно. Общее время выполнения: 211 мсек.

Создадим индекс:

```

1 CREATE INDEX idx_menu_item_composition
2 ON schema.menu_item_composition (menu_item_id, ingredient_id);
3

```

Data Output **Сообщения** План выполнения × Graph Visualiser × Notifications

CREATE INDEX

Запрос завершён успешно, время выполнения: 107 мсек.

После создания индекса:

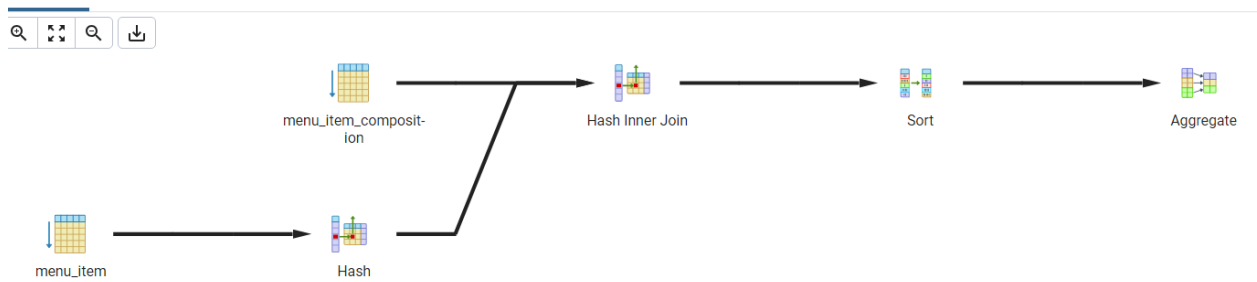
```

1 SELECT
2     mic.menu_item_id,
3     mi.name AS menu_item_name,
4     COUNT(mic.ingredient_id) AS num_ingredients
5 FROM
6     schema.menu_item_composition mic
7 JOIN
8     schema.menu_item mi ON mic.menu_item_id = mi.id
9 GROUP BY
10    mic.menu_item_id, mi.name
11 ORDER BY
12    mic.menu_item_id;
13

```

Data Output **Сообщения** План выполнения × Graph Visualiser × Notifications

Запрос выполнен успешно. Общее время выполнения: 143 мсек.



Вывод

В ходе выполнения данной лабораторной работы я успешно взаимодействовала с базой данных PostgreSQL, создавая разнообразные запросы на выборку данных. Особое внимание уделялось работе с подзапросами, что позволило применять их в запросах на запись, удаление и обновление данных. Это открывает новые возможности для эффективного управления информацией в базе данных, сокращая необходимость в дополнительных запросах. В процессе работы над запросами на модификацию данных я убедилась в гибкости подхода с использованием подзапросов, что позволяет осуществлять сложные манипуляции с данными, учитывая их взаимосвязи и контекст. Я выяснила, что применение индексов ускоряет операции чтения запросов. Я провели сравнение времени выполнения запросов до и после создания индексов, выявив повышение производительности. Важно отметить, что при этом следует балансировать между эффективностью чтения и возможными издержками при изменении данных в базе.