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: Зотов М. Д.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

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). 5
3. Подсчитать, сколько ингредиентов содержит каждое блюдо. 6
4. Вывести название блюда, содержащее максимальное число ингредиентов. 7
5. Вывести ингредиенты, которые закупаются, но не используются в блюдах 8
6. Сколько закреплено столов за каждым из официантов? 9
7. Какой из ингредиентов используется в максимальном количестве блюд? 10
Представления 11
1. Для расчета стоимости ингредиентов для заданного блюда 11
2. Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату. 13
Запросы на модификацию данных 15
1. INSERT 15
2. UPDATE 17
3. DELETE 19
Создание индексов 21
1. Простой индекс 21
2. Составной индекс 23
Вывод 26

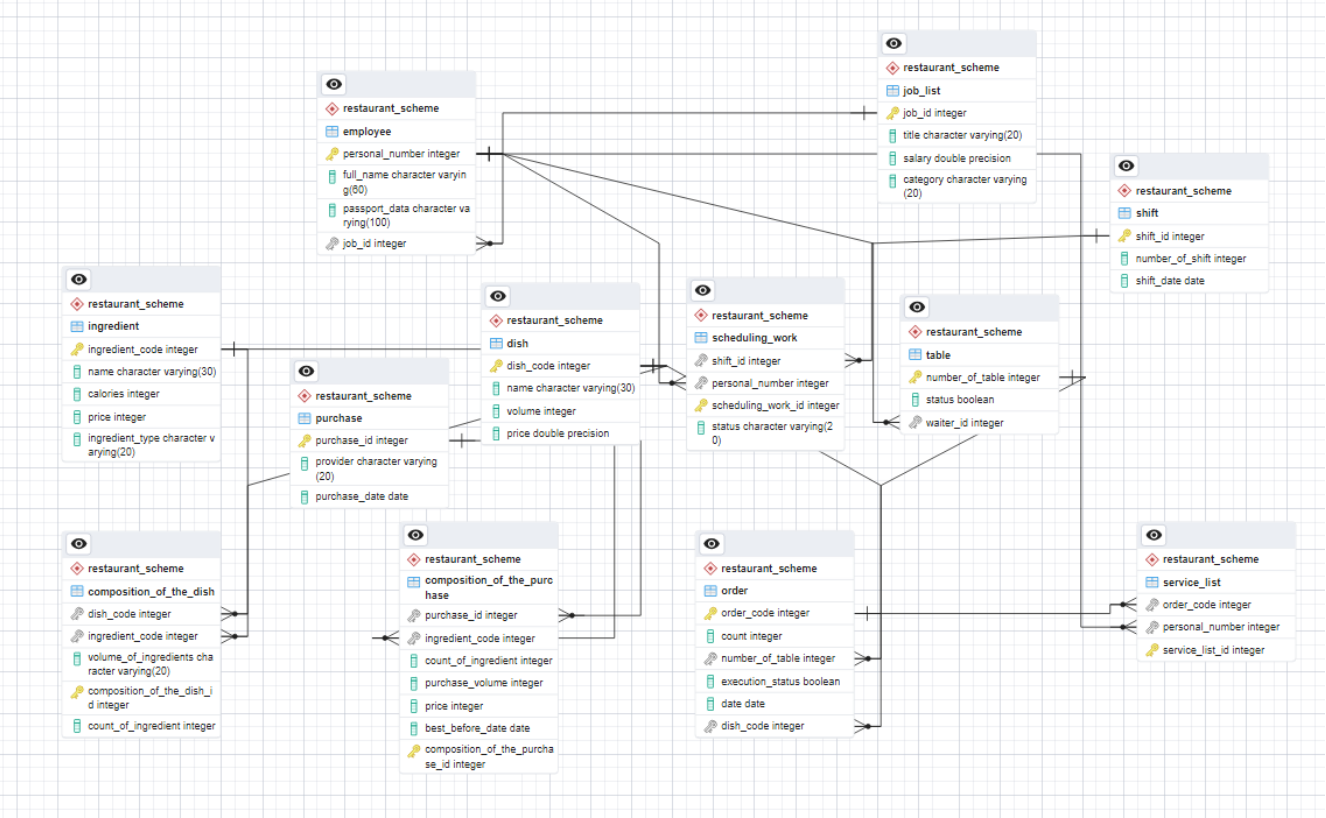
Цель работы

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:



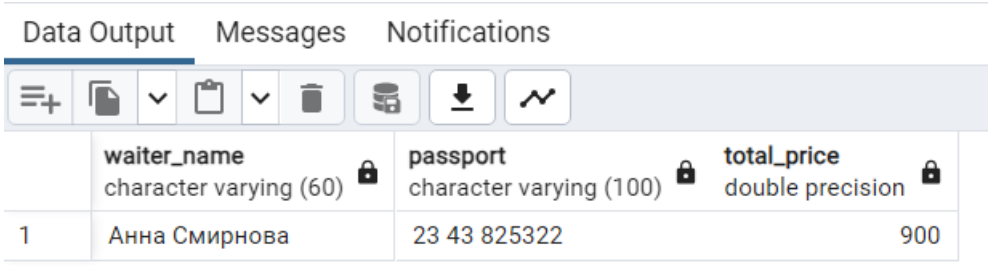
Выполнение

Запросы к базе данных

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

```
SELECT
  full_name AS waiter_name,
  passport_data AS passport,
  SUM(d.price) AS total_price
FROM restaurant_scheme.employee e
JOIN restaurant_scheme.table ta ON ta.waiter_id = e.personal_number
JOIN restaurant_scheme.order o ON o.number_of_table = ta.number_of_table
JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
WHERE
  o.date BETWEEN DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '1 month'
  AND DATE_TRUNC('month', CURRENT_DATE) AND o.execution_status='true'
GROUP BY e.personal_number, e.passport_data
HAVING SUM(d.price) = (
            SELECT SUM(d2.price)
    FROM restaurant_scheme.employee e2
    JOIN restaurant_scheme.table ta2 ON ta2.waiter_id = e2.personal_number
    JOIN restaurant_scheme.order o2 ON o2.number_of_table = ta2.number_of_table
    JOIN restaurant_scheme.dish d2 ON d2.dish_code = o2.dish_code
    WHERE
                    o2.date BETWEEN DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '1 month'
                    AND DATE_TRUNC('month', CURRENT_DATE)
              AND o2.execution_status = 'true'
    GROUP BY e2.personal_number
    ORDER BY SUM(d2.price) DESC
    LIMIT 1
  )
ORDER BY total_price DESC;
```

```sql
SELECT
  full_name AS waiter_name,
  passport_data AS passport,
  SUM(d.price) AS total_price
FROM restaurant_scheme.employee e
JOIN restaurant_scheme.table ta ON ta.waiter_id = e.personal_number
JOIN restaurant_scheme.order o ON o.number_of_table = ta.number_of_table
JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
WHERE
  o.date BETWEEN DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '1 month'
  AND DATE_TRUNC('month', CURRENT_DATE) AND o.execution_status='true'
GROUP BY e.personal_number, e.passport_data
HAVING SUM(d.price) = (
    SELECT SUM(d2.price)
    FROM restaurant_scheme.employee e2
    JOIN restaurant_scheme.table ta2 ON ta2.waiter_id = e2.personal_number
    JOIN restaurant_scheme.order o2 ON o2.number_of_table = ta2.number_of_table
    JOIN restaurant_scheme.dish d2 ON d2.dish_code = o2.dish_code
    WHERE
        o2.date BETWEEN DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '1 month'
        AND DATE_TRUNC('month', CURRENT_DATE)
        AND o2.execution_status = 'true'
    GROUP BY e2.personal_number
    ORDER BY SUM(d2.price) DESC
    LIMIT 1
  )
ORDER BY total_price DESC;
```



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

```
SELECT
  e.full_name AS full_name,
  SUM(d.price * 0.05) AS bonus
FROM restaurant_scheme.employee e
JOIN restaurant_scheme.table ta ON ta.waiter_id = e.personal_number
JOIN restaurant_scheme.order o ON o.number_of_table = ta.number_of_table
JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
WHERE
  DATE_TRUNC('day', o.date) >= (CURRENT_DATE - interval '10 day')
GROUP BY e.personal_number;
```

Query Query History

```sql
SELECT
  e.full_name AS full_name,
  SUM(d.price * 0.05) AS bonus
FROM restaurant_scheme.employee e
JOIN restaurant_scheme.table ta ON ta.waiter_id = e.personal_number
JOIN restaurant_scheme.order o ON o.number_of_table = ta.number_of_table
JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
WHERE
  DATE_TRUNC('day', o.date) >= (CURRENT_DATE - interval '10 day')
GROUP BY e.personal_number;
```

Data Output Messages Notifications

	full_name character varying (60)	bonus double precision
1	Анна Смирнова	45
2	Игорь Сапрыкин	32.5

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
```

```
SELECT
  d.name AS dish_name,
  COUNT(cd.composition_of_the_dish_id) AS count_of_ingredients
FROM restaurant_scheme.dish d
JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
WHERE cd.dish_code=d.dish_code
GROUP BY d.dish_code;
```

Query | Query History

```
1  SELECT
2    d.name AS dish_name,
3    COUNT(cd.composition_of_the_dish_id) AS count_of_ingredients
4  FROM restaurant_scheme.dish d
5  JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
6  WHERE cd.dish_code=d.dish_code
7  GROUP BY d.dish_code;
```

Data Output | Messages | Notifications

	dish_name character varying (30)	count_of_ingredients bigint
1	Тирамису	6
2	Лосось с терияки	7
3	Суши сэт "Дракон"	13

4. Вывести название блюда, содержащее максимальное число ингредиентов.

```
SELECT
  d.name AS dish_name,
  COUNT(cd.composition_of_the_dish_id) AS count_of_ingredients
FROM restaurant_scheme.dish d
JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
WHERE cd.dish_code=d.dish_code
GROUP BY d.dish_code
```

```
HAVING COUNT(cd.composition_of_the_dish_id) = (
        SELECT COUNT(cd2.composition_of_the_dish_id) AS "c"
        FROM restaurant_scheme.composition_of_the_dish cd2
        GROUP BY cd2.dish_code
        ORDER BY "c" DESC
        LIMIT 1
);
```

Query Query History

```
1  SELECT
2      d.name AS dish_name,
3      COUNT(cd.composition_of_the_dish_id) AS count_of_ingredients
4  FROM restaurant_scheme.dish d
5  JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
6  WHERE cd.dish_code=d.dish_code
7  GROUP BY d.dish_code
8  HAVING COUNT(cd.composition_of_the_dish_id) = (
9      SELECT COUNT(cd2.composition_of_the_dish_id) AS "c"
10     FROM restaurant_scheme.composition_of_the_dish cd2
11     GROUP BY cd2.dish_code
12     ORDER BY "c" DESC
13     LIMIT 1
14 );
```

Data Output Messages Notifications

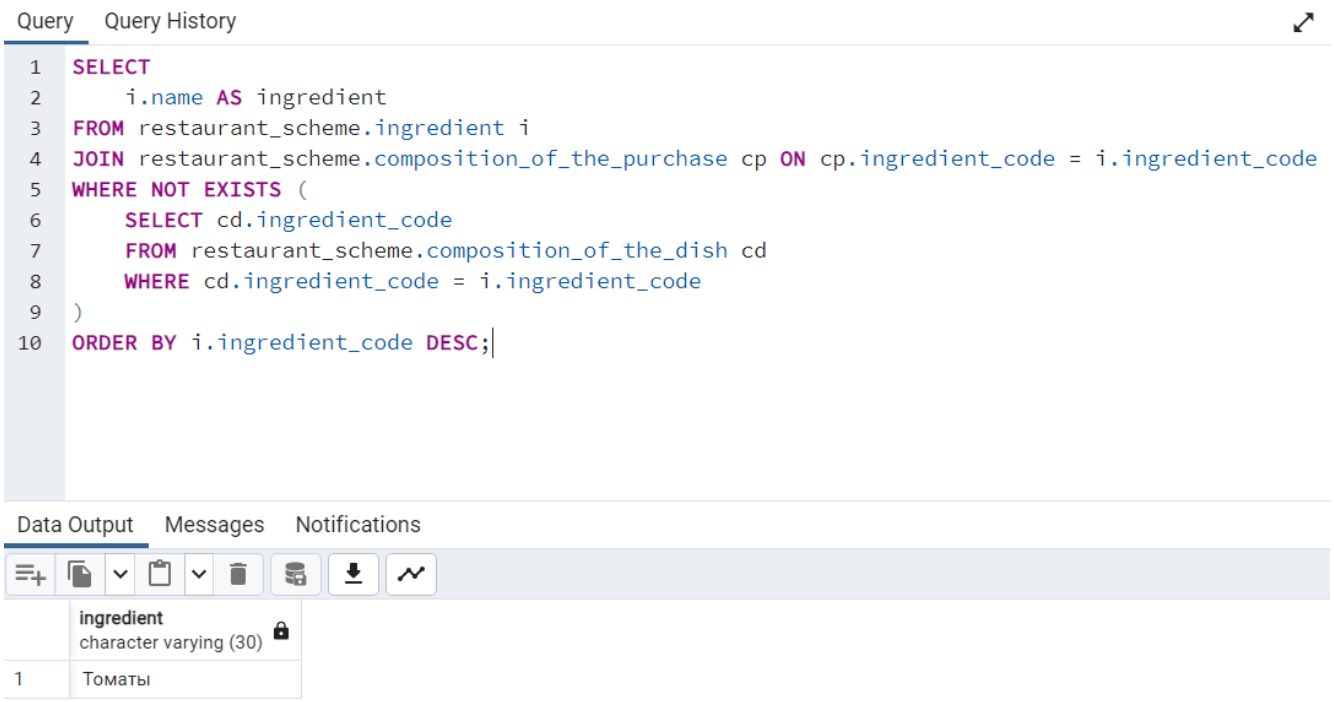
	dish_name character varying (30)	count_of_ingredients bigint
1	Суши сэт "Дракон"	13

5. Вывести ингредиенты, которые закупаются, но не используются в блюдах.

```
SELECT
        i.name AS ingredient
FROM restaurant_scheme.ingredient i
JOIN restaurant_scheme.composition_of_the_purchase cp ON cp.ingredient_code = i.ingredient_code
WHERE NOT EXISTS (
    SELECT cd.ingredient_code
    FROM restaurant_scheme.composition_of_the_dish cd
    WHERE cd.ingredient_code = i.ingredient_code
```

)

ORDER BY i.ingredient_code DESC;



6. Сколько закреплено столов за каждым из официантов?

SELECT

e.full_name AS waiter_name,

COUNT(tb.number_of_table) AS "count"

FROM restaurant_scheme.table tb

JOIN restaurant_scheme.employee e ON e.personal_number = tb.waiter_id

GROUP BY e.personal_number;

Query Query History

```
1  SELECT
2    e.full_name AS waiter_name,
3    COUNT(tb.number_of_table) AS "count"
4  FROM restaurant_scheme.table tb
5  JOIN restaurant_scheme.employee e ON e.personal_number = tb.waiter_id
6  GROUP BY e.personal_number;
```

Data Output Messages Notifications

	waiter_name character varying (60)	count bigint
1	Михаил Петров	4
2	Анна Смирнова	4
3	Игорь Сапрыкин	4

7. Какой из ингредиентов используется в максимальном количестве блюд?

```
SELECT
   i.name AS ingredient,
   COUNT(d.dish_code) AS "count"
FROM restaurant_scheme.ingredient i
JOIN restaurant_scheme.composition_of_the_dish cd ON cd.ingredient_code = i.ingredient_code
JOIN restaurant_scheme.dish d ON d.dish_code = cd.dish_code
GROUP BY i.ingredient_code
HAVING COUNT(d.dish_code) = (
        SELECT COUNT(cd2.dish_code) AS "c"
        FROM restaurant_scheme.composition_of_the_dish cd2
        GROUP BY cd2.ingredient_code
        ORDER BY "c" DESC
```

LIMIT 1

);

Query Query History

```
1  SELECT
2      i.name AS ingredient,
3      COUNT(d.dish_code) AS "count"
4  FROM restaurant_scheme.ingredient i
5  JOIN restaurant_scheme.composition_of_the_dish cd ON cd.ingredient_code = i.ingredient_code
6  JOIN restaurant_scheme.dish d ON d.dish_code = cd.dish_code
7  GROUP BY i.ingredient_code
8  HAVING COUNT(d.dish_code) = (
9      SELECT COUNT(cd2.dish_code) AS "c"
10     FROM restaurant_scheme.composition_of_the_dish cd2
11     GROUP BY cd2.ingredient_code
12     ORDER BY "c" DESC
13     LIMIT 1
14 );
```

Data Output Messages Notifications

	ingredient character varying (30)	count bigint
1	Сахар	3

Представления

1. **Для расчета стоимости ингредиентов блюда**

```
CREATE VIEW restaurant_scheme.ingredient_cost_calculation AS
SELECT
  d.name AS dish,
  SUM(i.price * cd.count_of_ingredient / 1000) AS total_price
FROM restaurant_scheme.dish d
JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
JOIN restaurant_scheme.ingredient i ON i.ingredient_code = cd.ingredient_code
GROUP BY d.dish_code;
```

Query | Query History

```sql
CREATE VIEW restaurant_scheme.ingredient_cost_calculation AS
SELECT
  d.name AS dish,
  SUM(i.price * cd.count_of_ingredient / 1000) AS total_price
FROM restaurant_scheme.dish d
JOIN restaurant_scheme.composition_of_the_dish cd ON cd.dish_code = d.dish_code
JOIN restaurant_scheme.ingredient i ON i.ingredient_code = cd.ingredient_code
GROUP BY d.dish_code;
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 70 msec.
```

Query | Query History

```sql
SELECT * FROM restaurant_scheme.ingredient_cost_calculation;
```

Data Output | Messages | Notifications

	dish character varying (30)	total_price bigint
1	Тирамису	44
2	Лосось с терияки	105
3	Суши сэт "Дракон"	377

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

```
CREATE VIEW restaurant_scheme.number_of_dishes_cooked AS
SELECT
  e.full_name AS full_name,
  d.name AS dish,
  COUNT(d.dish_code) * o.count AS "count",
  o.date AS date
FROM restaurant_scheme.employee e
JOIN restaurant_scheme.service_list sl ON sl.personal_number = e.personal_number
JOIN restaurant_scheme.order o ON o.order_code = sl.order_code
JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
JOIN restaurant_scheme.job_list jl ON jl.job_id = e.job_id
WHERE jl.title = 'Повар' OR jl.title = 'Шеф-повар'
GROUP BY e.personal_number, dish, o.count, date
ORDER BY e.personal_number;
```

Query | Query History

```
1  CREATE VIEW restaurant_scheme.number_of_dishes_cooked AS
2  SELECT
3    e.full_name AS full_name,
4    d.name AS dish,
5    COUNT(d.dish_code) * o.count AS "count",
6    o.date AS date
7  FROM restaurant_scheme.employee e
8  JOIN restaurant_scheme.service_list sl ON sl.personal_number = e.personal_number
9  JOIN restaurant_scheme.order o ON o.order_code = sl.order_code
10 JOIN restaurant_scheme.dish d ON d.dish_code = o.dish_code
11 JOIN restaurant_scheme.job_list jl ON jl.job_id = e.job_id
12 WHERE jl.title = 'Повар' OR jl.title = 'Шеф-повар'
13 GROUP BY e.personal_number, dish, o.count, date
14 ORDER BY e.personal_number;
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 77 msec.
```

Query Query History

```
1  SELECT * FROM restaurant_scheme.number_of_dishes_cooked;
```

Data Output Messages Notifications

	full_name character varying (60)	dish character varying (30)	count bigint	date date
1	Алексей Васнецов	Тирамису	2	2024-03-08
2	Иван Андреев	Суши сэт "Дракон"	1	2024-03-08
3	София Тимофеева	Лосось с терияки	2	2024-03-08

Запросы на модификацию данных

1. **INSERT**

Query | Query History

```
SELECT * FROM restaurant_scheme.composition_of_the_dish
ORDER BY dish_code ASC;
```

Data Output | Messages | Notifications

	dish_code integer	ingredient_code integer	volume_of_ingredients character varying (20)	composition_of_the_dish_id [PK] integer	count_of_ingredient integer
1	6	24	г	7	300
2	6	5	г	12	30
3	6	10	г	11	200
4	6	4	г	10	70
5	6	13	г	9	50
6	6	27	г	8	100
7	11	4	г	25	60
8	11	7	г	19	480

Total rows: 25 of 25 | Query complete 00:00:00.122

Query | Query History

```
INSERT INTO restaurant_scheme.composition_of_the_dish
(dish_code, ingredient_code, volume_of_ingredients, count_of_ingredient)
VALUES (
  (SELECT dish_code FROM restaurant_scheme.dish WHERE dish.name = 'Лосось с терияки'),
  (SELECT ingredient_code FROM restaurant_scheme.ingredient WHERE ingredient.name = 'Лимон'),
  'г',
  30
);
```

Query | Query History

```
1  SELECT * FROM restaurant_scheme.composition_of_the_dish
2  ORDER BY dish_code ASC;
```

Data Output | Messages | Notifications

	dish_code integer	ingredient_code integer	volume_of_ingredients character varying (20)	composition_of_the_dish_id [PK] integer	count_of_ingredient integer
1	6	27	г	8	100
2	6	18	г	27	30
3	6	5	г	12	30
4	6	10	г	11	200
5	6	4	г	10	70
6	6	13	г	9	50
7	6	24	г	7	300
8	11	28	г	14	95

Total rows: 26 of 26 | Query complete 00:00:00.117

2. UPDATE (изменение данных в таблице order)

Query | Query History

```
1  SELECT * FROM restaurant_scheme.scheduling_work
2  ORDER BY status ASC;
```

Data Output | Messages | Notifications

	shift_id integer	personal_number integer	scheduling_work_id [PK] integer	status character varying (20)
1	7	10	2	Отпуск
2	3	8	8	Отпуск
3	11	11	9	Отпуск
4	9	13	11	Отпуск
5	12	12	10	Отпуск
6	5	4	4	Отпуск
7	2	5	5	Отпуск
8	10	14	12	Отпуск

Total rows: 12 of 12 | Query complete 00:00:00.144

Query | Query History

```
1  UPDATE restaurant_scheme.scheduling_work
2  SET status = 'Вышел'
3  WHERE personal_number IN (SELECT personal_number FROM restaurant_scheme.employee
4                WHERE full_name = 'Иван Андреев');
```

Query | Query History

```
1  SELECT * FROM restaurant_scheme.scheduling_work
2  ORDER BY status ASC;
```

Data Output | Messages | Notifications

	shift_id integer	personal_number integer	scheduling_work_id [PK] integer	status character varying (20)
1	1	3	3	Вышел
2	3	8	8	Отпуск
3	11	11	9	Отпуск
4	9	13	11	Отпуск
5	12	12	10	Отпуск
6	5	4	4	Отпуск
7	7	10	2	Отпуск
8	10	14	12	Отпуск

Total rows: 12 of 12 | Query complete 00:00:00.114

3. DELETE

Query | Query History

```
1  SELECT * FROM restaurant_scheme.composition_of_the_dish
2  ORDER BY dish_code ASC;
```

Data Output | Messages | Notifications

	dish_code integer	ingredient_code integer	volume_of_ingredients character varying (20)	composition_of_the_dish_id [PK] integer	count_of_ingredient integer
1	6	13	г	9	50
2	6	3	кг	29	1
3	6	18	г	27	30
4	6	5	г	12	30
5	6	10	г	11	200
6	6	4	г	10	70
7	6	24	г	7	300
8	6	27	г	8	100

Total rows: 27 of 27 | Query complete 00:00:00.137

Query | Query History

```
1  DELETE FROM restaurant_scheme.composition_of_the_dish
2  WHERE ingredient_code IN (
3    SELECT ingredient_code FROM restaurant_scheme.ingredient
4    WHERE ingredient.name = 'Мука'
5  );
```

Query | Query History

```
1  SELECT * FROM restaurant_scheme.composition_of_the_dish
2  ORDER BY dish_code ASC;
```

Data Output | Messages | Notifications

	dish_code integer	ingredient_code integer	volume_of_ingredients character varying (20)	composition_of_the_dish_id [PK] integer	count_of_ingredient integer
1	6	27	г	8	100
2	6	18	г	27	30
3	6	5	г	12	30
4	6	10	г	11	200
5	6	4	г	10	70
6	6	13	г	9	50
7	6	24	г	7	300
8	11	28	г	14	95

Total rows: 26 of 26 | Query complete 00:00:00.108

Создание индексов

1. **Простой индекс**

До создания индекса:

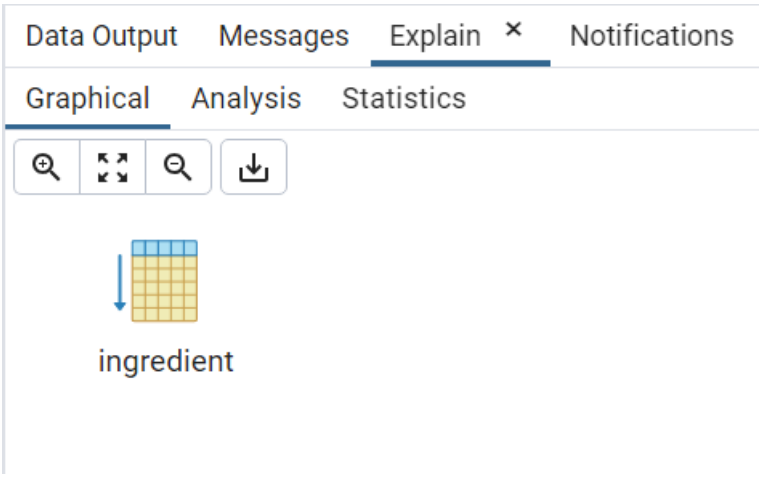
Query Query History

```
1  SELECT name FROM restaurant_scheme.ingredient
2  WHERE ingredient_type = 'Сухие продукты';
```

Data Output Messages Explain Notifications

	name character varying (30)
1	Сахар
2	Мука
3	Рис
4	Соль поваренная
5	Кофе
6	Какао порошок
7	Нори

```
Successfully run. Total query runtime: 130 msec.
7 rows affected.
```


Data Output
Messages
Explain
Notifications
Graphical
Analysis
Statistics
ingredient

Создадим индекс:

Query Query History

```
1  CREATE INDEX find_ingredient_with_type
2  ON restaurant_scheme.ingredient (ingredient_type);
```

Data Output Messages Explain Notifications

```
CREATE INDEX

Query returned successfully in 1 secs 132 msec.
```

После создания индекса:

```
Successfully run. Total query runtime: 98 msec.
7 rows affected.
```

2. **Составной индекс**

До создания индекса:

Query | Query History

```sql
SELECT
    e.personal_number AS "id",
    e.full_name AS waiter_name,
    COUNT(tb.number_of_table) AS "count"
FROM restaurant_scheme.table tb
JOIN restaurant_scheme.employee e ON e.personal_number = tb.waiter_id
GROUP BY "id";
```

Data Output | Messages | Explain | Notifications

	id integer	waiter_name character varying (60)	count bigint
1	5	Михаил Петров	4
2	4	Анна Смирнова	4
3	12	Игорь Сапрыкин	4

```
Successfully run. Total query runtime: 132 msec.
3 rows affected.
```

Создадим индекс:

Query Query History

```
1  CREATE INDEX ind2
2  ON restaurant_scheme.table (number_of_table, waiter_id);
```

Data Output Messages Explain Notifications

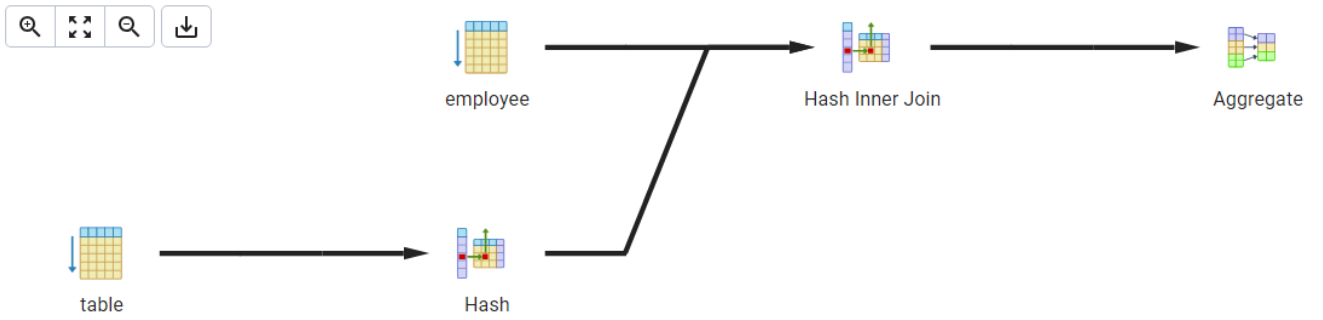
```
CREATE INDEX

Query returned successfully in 72 msec.
```

После создания индекса:

```
Successfully run. Total query runtime: 98 msec.
3 rows affected.
```

Data Output Messages Explain Notifications

Graphical Analysis Statistics



Вывод

В ходе выполнения данной лабораторной работы я научился писать sql запросы на выборку данных, создал представления, создал запросы на модификацию данных, а также разобрался в простых и составных индексах.