Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 4

по теме: Запросы на выборку данных к БД PostgreSQL. Представления в PostgreSQL

по дисциплине: Проектирование и реализация баз данных

Специальность: 09.03.03 Мобильные и сетевые технологии

Проверил: Говорова М.М.
_____
Дата: «29» мая 2021г.
Оценка _____

Выполнил:
студент группы К3240
Костылев Иван

Санкт-Петербург 2021 г

Цель работы

Овладение практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL

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.
4. Просмотреть историю запросов.

Ход работы

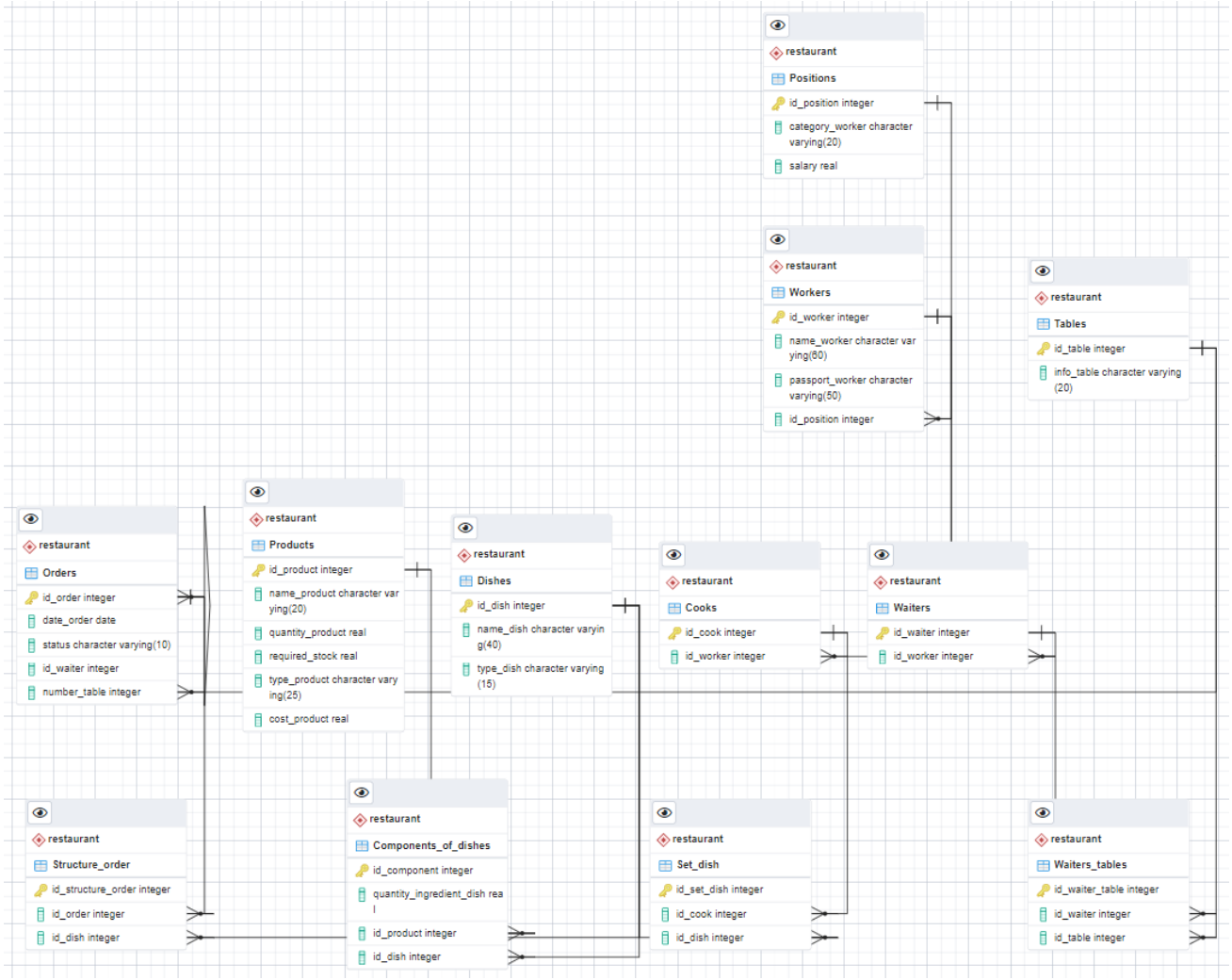
Вариант 13. БД «Ресторан»

Описание предметной области: Сотрудники ресторана – повара и официанты. За каждым официантом закреплены определенные столы. Каждый повар готовит определенный набор 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: 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. Поставщик. Наименование блюда. Код блюда. Объем ингредиента. Номер стола. Дата заказа. Код заказа. Количество. Название блюда. Ингредиенты, входящие в блюдо. Тип ингредиента.

1. **Название создаваемой БД:** Restaurant

2. Схема инфологической модели



3. Скрипты запросов

3.1.Вывести данные официанта, принявшего максимальное число заказов

```
select workers.id_worker, workers.name_worker, workers.passport_worker
    from restaurant."Workers" as workers,
    (select waiters_ord.id_waiter from
        (select Waiters.id_waiter, count(Orders.id_waiter) as count_ord
         from restaurant."Orders" as Orders join restaurant."Waiters" as Waiters using(id_waiter)
         group by Waiters.id_waiter) as waiters_ord
    where
        waiters_ord.count_ord >= all(select count(Orders.id_waiter) as count_ord
         from restaurant."Orders" as Orders join restaurant."Waiters" as Waiters using(id_waiter)
         group by Waiters.id_waiter)
    group by waiters_ord.id_waiter) as max_waiter,
    restaurant."Waiters" as waiters
where (workers.id_worker = waiters.id_worker) AND (waiters.id_waiter = max_waiter.id_waiter)
```

Результат:

	id_worker [PK] integer	name_worker character varying (60)	passport_worker character varying (50)
1	2	Ivanov Sergey Aristarkhovich	57** ****43

3.2.Подсчитать, сколько ингредиентов содержит каждое блюдо.

```
SELECT d.id_dish, d.name_dish, d.type_dish, count(comp.id_component)
        FROM restaurant."Dishes" AS d,
        restaurant."Components_of_dishes" AS comp
        WHERE (d.id_dish = comp.id_dish)
GROUP BY d.id_dish
ORDER BY id_dish;
```

Результат:

	id_dish [PK] integer	name_dish character varying (40)	type_dish character varying (15)	count bigint
1	2	яблочный пирог	dessert	3
2	3	борщ	first course	4
3	4	каша	first course	2
4	5	картошка жареная	second course	2
5	6	стейк	second course	1
6	7	курица с овощами	second course	3

3.3.Вывести название блюда, содержащее максимальное число ингредиентов.

```
1.  SELECT d1.id_dish, d1.name_dish, MAX(d1.num_comp) as num_comp FROM
2.          (SELECT d.id_dish, d.name_dish, d.type_dish, count(comp.id_component) AS num_comp
3.                  FROM restaurant."Dishes" AS d,
4.                  restaurant."Components_of_dishes" AS comp
5.                  WHERE (d.id_dish = comp.id_dish)
6.          GROUP BY d.id_dish) AS d1
7.          WHERE NOT EXISTS
8.          (SELECT * FROM (
9.                  SELECT d.id_dish, d.name_dish, d.type_dish, count(comp.id_component) AS num_comp
10.                 FROM restaurant."Dishes" AS d,
11.                 restaurant."Components_of_dishes" AS comp
12.                 WHERE (d.id_dish = comp.id_dish)
13.         GROUP BY d.id_dish) AS d2
14.         WHERE (d1.num_comp < d2.num_comp ))
```

	id_dish [PK] integer	name_dish character varying (40)	num_comp bigint
1	3	борщ	4

3.4.Какой повар может приготовить максимальное число видов блюд?

```
1.  SELECT cooks.id_cook, cooks.name_worker, cooks.passport_worker, c1.type_count FROM
2.          (SELECT id_cook, count(distinct type_dish) AS type_count
3.                  FROM restaurant."Set_dish" JOIN restaurant."Dishes" USING (id_dish)
4.                  GROUP BY id_cook) AS c1,
5.          (SELECT cooks.id_worker, cooks.id_cook, wrk.name_worker, wrk.passport_worker
6.                  FROM restaurant."Cooks" as cooks,
7.                  restaurant."Workers" as wrk
8.                  WHERE (cooks.id_worker = wrk.id_worker)) AS cooks
9.          WHERE NOT EXISTS
10.                 (SELECT * FROM
11.                         (SELECT id_cook, count(distinct type_dish) AS type_count
12.                         FROM restaurant."Set_dish" JOIN restaurant."Dishes" USING (id_dish)
13.                                 GROUP BY id_cook) AS c2
14.                 WHERE c1.type_count < c2.type_count) AND (cooks.id_cook = c1.id_cook);
```

Результат:

	id_cook integer	name_worker character varying (60)	passport_worker character varying (50)	type_count bigint
1	6	Uhazhorov Kirill Gennadievich	47** ****09	3

3.5. Сколько закреплено столов за каждым из официантов?

```
1.  SELECT wrk.id_worker, wrk.name_worker, wrk.passport_worker, wtr_cnt.tables_count FROM
2.          (SELECT wtr.id_waiter, count(distinct wtr_tbl.id_table) AS tables_count
3.                  FROM restaurant."Waiters" AS wtr,
4.                  restaurant."Waiters_tables" AS wtr_tbl
5.                  WHERE (wtr.id_waiter = wtr_tbl.id_waiter)
6.          GROUP BY wtr.id_waiter) AS wtr_cnt,
7.          (restaurant."Waiters"  JOIN restaurant."Workers" USING (id_worker)) AS wrk
8.          WHERE wrk.id_waiter = wtr_cnt.id_waiter
9.  ORDER BY tables_count DESC;
```

Результат:

	id_worker integer	name_worker character varying (60)	passport_worker character varying (50)	tables_count bigint
1	6	Zhdanov Sergey Yurievich	09** ****32	5
2	2	Ivanov Sergey Aristarkhovich	57** ****43	4
3	7	Vaseva Regina Andreevna	47** ****67	3
4	8	Name Name Anonnim	**** ******	3

3.6. Какой из ингредиентов используется в максимальном количестве блюд?

```
1.  SELECT cnt1.id_product, cnt1.name_product, cnt1.dishes_cnt
2.          FROM (SELECT prod.id_product, prod.name_product, count(comp.id_dish) AS dishes_cnt
3.                  FROM restaurant."Components_of_dishes" AS comp,
4.                  restaurant."Products" AS prod
5.                  WHERE (prod.id_product = comp.id_product)
6.          GROUP BY prod.id_product) AS cnt1
7.          WHERE NOT EXISTS (SELECT *
8.                  FROM (SELECT prod.id_product, prod.name_product, count(comp.id_dish) AS dishes_cnt
9.                              FROM restaurant."Components_of_dishes" AS comp,
10.                                 restaurant."Products" AS prod
11.                                 WHERE (prod.id_product = comp.id_product)
12.                             GROUP BY prod.id_product) AS cnt2
13.                 WHERE (cnt1.dishes_cnt < cnt2.dishes_cnt));
```

Результат:

	id_product [PK] integer	name_product character varying (20)	dishes_cnt bigint
1	8	картофель	3
2	9	лук	3
3	6	курица	3

3.7.Вывести данные официанта, принявшего заказы на максимальную сумму.

```
SELECT data1.id_worker, data1.name_worker, data1.passport_worker, data1.worker_sum FROM
        (SELECT workers.id_worker, workers.name_worker, workers.passport_worker,
waiters_work.sum AS worker_sum FROM
                        (SELECT waiters.id_waiter, SUM(costs_ord.sum_cost_dish) FROM
                                (SELECT struct_ord.id_order, SUM(costs.cost_dish) AS
sum_cost_dish FROM
                                        (SELECT comp.id_dish, dishes.name_dish,
SUM(comp.quantity_ingredient_dish*prod.cost_product)*1.4 AS cost_dish
                                                FROM restaurant."Components_of_dishes" AS comp,
                                                        restaurant."Products" AS prod,
                                                        restaurant."Dishes" AS dishes
                                                WHERE comp.id_dish = dishes.id_dish AND
comp.id_product = prod.id_product
                                                GROUP BY comp.id_dish, dishes.name_dish) AS
costs,
                                        restaurant."Structure_order" AS struct_ord
                                        WHERE struct_ord.id_dish = costs.id_dish
                                GROUP BY struct_ord.id_order) AS costs_ord,
                                restaurant."Waiters" AS waiters,
                                restaurant."Orders" AS ord
                                WHERE (costs_ord.id_order = ord.id_order AND waiters.id_waiter =
ord.id_waiter)
                        GROUP BY waiters.id_waiter) AS waiters_work,
                        (restaurant."Workers" JOIN restaurant."Waiters" USING(id_worker)) AS
workers
                        WHERE workers.id_waiter = waiters_work.id_waiter) AS data1
WHERE NOT EXISTS (SELECT * FROM
          (SELECT workers.id_worker, workers.name_worker, workers.passport_worker,
waiters_work.sum AS worker_sum FROM
                        (SELECT waiters.id_waiter, SUM(costs_ord.sum_cost_dish) FROM
                                (SELECT struct_ord.id_order, SUM(costs.cost_dish) AS
sum_cost_dish FROM
                                        (SELECT comp.id_dish, dishes.name_dish,
SUM(comp.quantity_ingredient_dish*prod.cost_product)*1.4 AS cost_dish
                                                FROM restaurant."Components_of_dishes" AS comp,
                                                        restaurant."Products" AS prod,
                                                        restaurant."Dishes" AS dishes
                                                WHERE comp.id_dish = dishes.id_dish AND
comp.id_product = prod.id_product
                                                GROUP BY comp.id_dish, dishes.name_dish) AS
costs,
                                        restaurant."Structure_order" AS struct_ord
                                        WHERE struct_ord.id_dish = costs.id_dish
                                GROUP BY struct_ord.id_order) AS costs_ord,
                                restaurant."Waiters" AS waiters,
                                restaurant."Orders" AS ord
```

```
35.                             WHERE (costs_ord.id_order = ord.id_order AND waiters.id_waiter =
    ord.id_waiter)
36.                       GROUP BY waiters.id_waiter) AS waiters_work,
37.                       (restaurant."Workers" JOIN restaurant."Waiters" USING(id_worker)) AS
    workers
38.                       WHERE workers.id_waiter = waiters_work.id_waiter) AS data2
39.          WHERE data2.worker_sum > data1.worker_sum)
```

Результат:

	id_worker [PK] integer	name_worker character varying (60)	passport_worker character varying (50)	worker_sum double precision
1	2	Ivanov Sergey Aristarkhovich	57** ****43	1640.8

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

```
1.  SELECT workers.id_worker, workers.name_worker, workers.passport_worker, waiters_work.sum *
    0.05 AS prem_sum FROM
2.          (SELECT waiters.id_waiter, waiters.id_worker, SUM(costs_ord.sum_cost_dish) FROM
3.                  (SELECT struct_ord.id_order, SUM(costs.cost_dish) AS sum_cost_dish FROM
4.                          (SELECT comp.id_dish, dishes.name_dish,
    SUM(comp.quantity_ingredient_dish*prod.cost_product)*1.4 AS cost_dish
5.                                  FROM restaurant."Components_of_dishes" AS comp,
6.                                          restaurant."Products" AS prod,
7.                                          restaurant."Dishes" AS dishes
8.                                  WHERE comp.id_dish = dishes.id_dish AND comp.id_product =
    prod.id_product
9.                                  GROUP BY comp.id_dish, dishes.name_dish) AS costs,
10.                         restaurant."Structure_order" AS struct_ord
11.                         WHERE struct_ord.id_dish = costs.id_dish
12.                 GROUP BY struct_ord.id_order) AS costs_ord,
13.                 restaurant."Waiters" AS waiters,
14.                 restaurant."Orders" AS ord
15.                 WHERE (costs_ord.id_order = ord.id_order AND waiters.id_waiter = ord.id_waiter
    AND
16.                             (date_order + INTEGER '10' >= CURRENT_DATE))
17.         GROUP BY waiters.id_waiter) AS waiters_work,
18.         (restaurant."Workers" JOIN restaurant."Waiters" USING(id_worker)) AS workers
19.         WHERE workers.id_waiter = waiters_work.id_waiter;
```

	id_worker [PK] integer	name_worker character varying (60)	passport_worker character varying (50)	prem_sum double precision
1	2	Ivanov Sergey Aristarkhovich	57** ****43	69.16
2	6	Zhdanov Sergey Yurievich	09** ****32	28.49
3	8	Name Name Anonnim	**** ******	34.86

4. Запросы на модификацию данных

4.1. Добавить 3 новых заказа, взяв за основу первые 3 (дату взять сегодняшнюю, а статус задать «принят»)

Содержание таблицы restaurant.”Orders”:

	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	1	2021-04-20	ready	1	4
2	2	2021-04-20	ready	3	2
3	3	2021-04-20	ready	2	2
4	4	2021-05-26	ready	4	3
5	5	2021-05-26	ready	1	1
6	6	2021-05-27	paid	1	6
7	7	2021-05-27	ready	2	3
8	8	2021-05-27	accepted	1	2
9	9	2021-05-27	ready	2	7

Запрос вставки

```
1.  INSERT INTO restaurant."Orders"(
2.          date_order, STATUS, id_waiter, number_table)
3.          (SELECT now() AS date_order, 'accepted' AS STATUS, id_waiter, number_table
4.                  FROM restaurant."Orders" WHERE id_order <= 3);
```

Содержание таблицы restaurant.”Orders” после вставки:

	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	1	2021-04-20	ready	1	4
2	2	2021-04-20	ready	3	2
3	3	2021-04-20	ready	2	2
4	4	2021-05-26	ready	4	3
5	5	2021-05-26	ready	1	1
6	6	2021-05-27	paid	1	6
7	7	2021-05-27	ready	2	3
8	8	2021-05-27	accepted	1	2
9	9	2021-05-27	ready	2	7
10	10	2021-05-29	accepted	1	4
11	11	2021-05-29	accepted	3	2
12	12	2021-05-29	accepted	2	2

4.2. Предположим, прошло время, наступил следующий день. Тогда нам нужно изменить статус у всех заказов «accepted» на «canceled» (заказ был принят, но так и не был оплачен).

Содержание таблицы restaurant.”Orders” перед изменением:

	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	1	2021-04-20	ready	1	4
2	2	2021-04-20	ready	3	2
3	3	2021-04-20	ready	2	2
4	4	2021-05-26	ready	4	3
5	5	2021-05-26	ready	1	1
6	6	2021-05-27	paid	1	6
7	7	2021-05-27	ready	2	3
8	9	2021-05-27	ready	2	7
9	8	2021-05-27	accepted	1	2
10	10	2021-05-29	accepted	1	4
11	11	2021-05-29	accepted	3	2
12	12	2021-05-29	accepted	2	2

```
1. UPDATE restaurant."Orders"
2.         SET STATUS = 'canceled'
3.         WHERE STATUS = 'accepted' AND date_order < CURRENT_DATE;
```

	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	1	2021-04-20	ready	1	4
2	2	2021-04-20	ready	3	2
3	3	2021-04-20	ready	2	2
4	4	2021-05-26	ready	4	3
5	5	2021-05-26	ready	1	1
6	6	2021-05-27	paid	1	6
7	7	2021-05-27	ready	2	3
8	8	2021-05-27	canceled	1	2
9	9	2021-05-27	ready	2	7
10	10	2021-05-29	accepted	1	4
11	11	2021-05-29	accepted	3	2
12	12	2021-05-29	accepted	2	2

4.3. Владелец ресторана чрезмерно экономный человек, поэтому требует регулярно избавляться от старых данных в таблице заказов. Нужно удалить всю информацию о заказах, которые были ранее, чем 30 дней назад.

Таблица Orders перед удалением

	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	1	2021-04-20	ready	1	4
2	2	2021-04-20	ready	3	2
3	3	2021-04-20	ready	2	2
4	4	2021-05-26	ready	4	3
5	5	2021-05-26	ready	1	1
6	6	2021-05-27	paid	1	6
7	7	2021-05-27	ready	2	3
8	8	2021-05-27	canceled	1	2
9	9	2021-05-27	ready	2	7
10	10	2021-05-29	accepted	1	4
11	11	2021-05-29	accepted	3	2
12	12	2021-05-29	accepted	2	2

Таблица Structure_order перед удалением

	id_structure_order [PK] integer	id_order integer	id_dish integer
1	26	1	2
2	27	1	5
3	28	2	3
4	29	3	4
5	30	3	7
6	31	3	5
7	32	3	5
8	33	4	2
9	34	4	6
10	35	4	7
11	36	5	2
12	37	5	3
13	38	5	4
14	39	5	5
15	40	5	6
16	41	5	7
17	42	6	2
18	43	6	7
19	44	7	4
20	45	7	5
21	46	8	3
22	47	8	4
23	48	9	5
24	49	9	6
25	50	9	7

Скрипт удаления данных:

```
DELETE FROM restaurant."Structure_order" AS struct
        WHERE struct.id_structure_order IN (SELECT struct.id_structure_order
                                            FROM restaurant."Orders" AS ord,
                                            restaurant."Structure_order" AS struct
                 WHERE ord.date_order < CURRENT_DATE - INTEGER '30'
                AND struct.id_order = ord.id_order);

DELETE FROM restaurant."Orders"
        WHERE id_order IN (
                SELECT ord.id_order FROM restaurant."Orders" AS ord
                WHERE ord.date_order < CURRENT_DATE - INTEGER '30');
```

Таблицы Orders и Structure_order после удаления

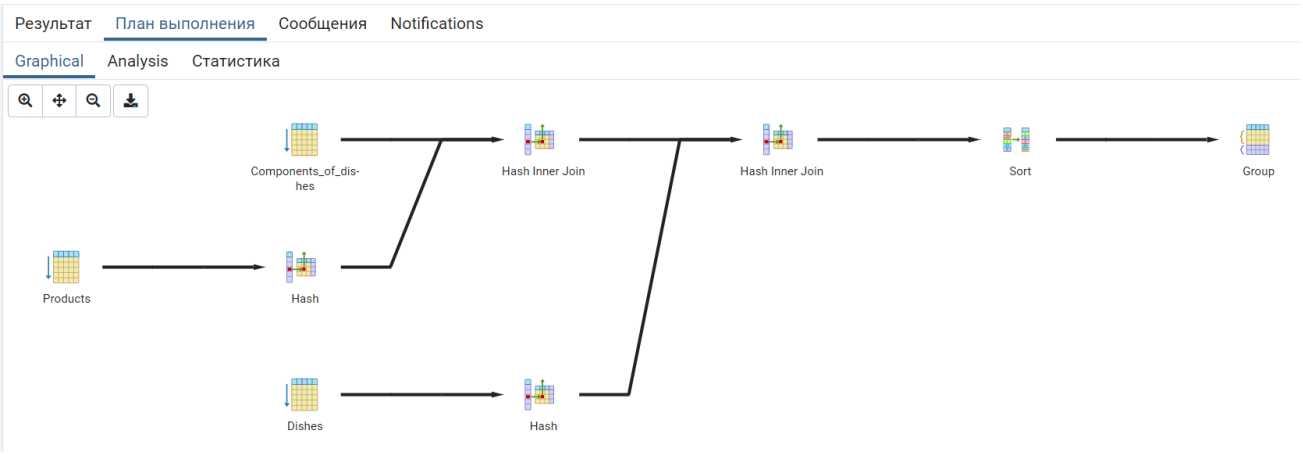
	id_order [PK] integer	date_order date	status character varying (10)	id_waiter integer	number_table integer
1	4	2021-05-26	ready	4	3
2	5	2021-05-26	ready	1	1
3	6	2021-05-27	paid	1	6
4	7	2021-05-27	ready	2	3
5	8	2021-05-27	canceled	1	2
6	9	2021-05-27	ready	2	7
7	10	2021-05-29	accepted	1	4
8	11	2021-05-29	accepted	3	2
9	12	2021-05-29	accepted	2	2

	id_structure_order [PK] integer	id_order integer	id_dish integer
1	33	4	2
2	34	4	6
3	35	4	7
4	36	5	2
5	37	5	3
6	38	5	4
7	39	5	5
8	40	5	6
9	41	5	7
10	42	6	2
11	43	6	7
12	44	7	4
13	45	7	5
14	46	8	3
15	47	8	4
16	48	9	5
17	49	9	6
18	50	9	7

5. **Создать представления:**

5.1.для расчета стоимости ингредиентов для заданного блюда;

```
1.  SELECT  dishes.name_dish, prod.name_product,
2.          comp.quantity_ingredient_dish, prod.cost_product,
    (comp.quantity_ingredient_dish*prod.cost_product) AS cost_component
3.          FROM restaurant."Components_of_dishes" AS comp,
4.                  restaurant."Products" AS prod,
5.                  restaurant."Dishes" AS dishes
6.          WHERE comp.id_dish = dishes.id_dish AND comp.id_product = prod.id_product
7.          GROUP BY dishes.name_dish, prod.name_product,
8.          comp.quantity_ingredient_dish,
    prod.cost_product,(comp.quantity_ingredient_dish*prod.cost_product)
```



5.2. количество приготовленных блюд по каждому блюду за определенную дату.

```
1.  SELECT DISTINCT  ord.date_order, d.name_dish, COUNT(d.id_dish)
2.  FROM restaurant."Dishes" AS d,
3.  (restaurant."Orders" JOIN restaurant."Structure_order" USING(id_order)) AS ord
4.  WHERE ord.id_dish = d.id_dish
5.  GROUP BY d.id_dish, ord.date_order
6.  ORDER BY ord.date_order;
```