

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет инфокоммуникационных технологий

Дисциплина:

«Проектирование и реализация баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

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

Выполнил:

студент группы К32402

Пономарев Константин Витальевич

(подпись)

Проверил(а):

Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург 2023 г.

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4. **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.

Вариант 13. БД «Ресторан»

Описание предметной области: Необходимо создать систему для обслуживания заказов клиентов в ресторане.

Сотрудники ресторана – повара и официанты.

За каждым официантом закреплены определенные столы за смену. Клиенты могут бронировать столы заранее.

Каждый повар может готовить определенный набор блюд.

Официант принимает заказ от стола и передает его на кухню. Шеф-повар распределяет блюда для приготовления между поварами. В одном заказе может быть несколько одинаковых или разных блюд.

Запас продуктов на складе не должен быть ниже заданного значения.

Цена заказа складывается из стоимости ингредиентов и наценки, которая составляет 40% стоимости ингредиентов.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Паспортные данные сотрудника. Категория сотрудника. Должность сотрудника. Оклад сотрудника. Наименование ингредиента. Код ингредиента. Дата закупки. Объем закупки. Количество продукта на складе. Необходимый запас продукта. Срок годности. Цена ингредиента. Калорийность (на 100г продукта). Поставщик. Наименование блюда. Код блюда. Объем ингредиента. Номер стола. Дата заказа. Код заказа.

Количество. Название блюда. Ингредиенты, входящие в блюдо. Тип ингредиента.

Задание 1.1 (ЛР 1 БД). Выполните инфологическое моделирование базы данных системы. (Ограничения задать самостоятельно.)

Задание 1.2. Создайте логическую модель БД, используя ИЛМ (задание 1.1). Используйте необходимые средства поддержки целостности данных в СУБД.

Задание 2. Создать запросы:

- Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.
- Рассчитать премию каждого официанта за последний месяц (5% от стоимости каждого заказа).
- Подсчитать, сколько ингредиентов содержит каждое блюдо.
- Вывести название блюда, содержащее максимальное число ингредиентов.
- Какой повар может приготовить максимальное число видов блюд? Сколько закреплено столов за каждым из официантов?
- Какой из ингредиентов используется в максимальном количестве блюд?

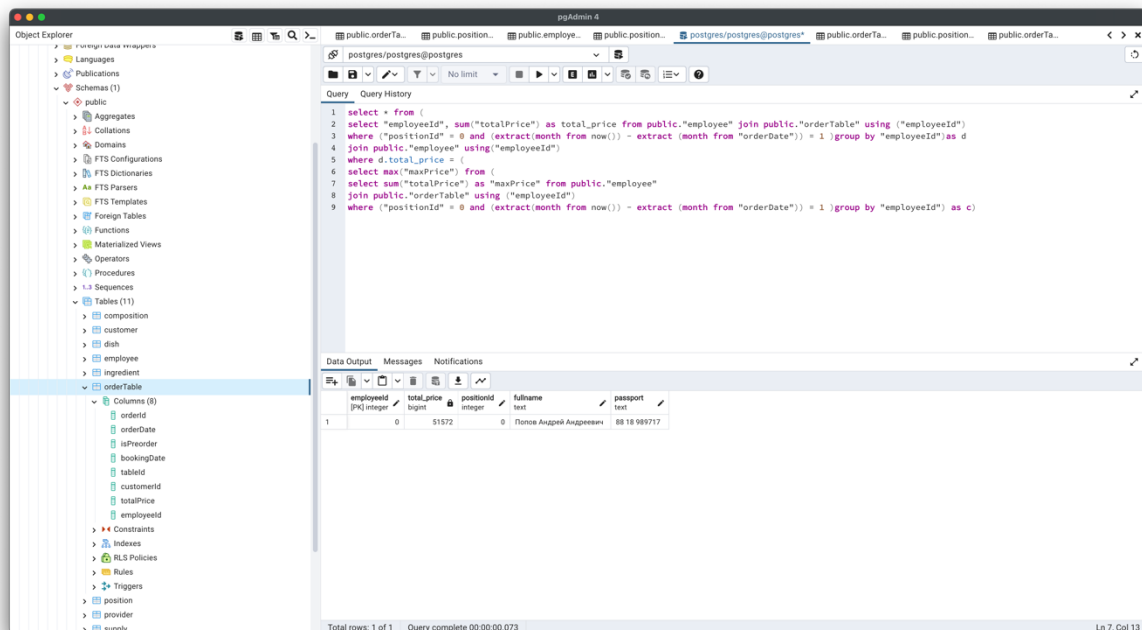
Задание 3. Создать представление:

- для расчета стоимости ингредиентов для заданного блюда;
- количество приготовленных блюд по каждому блюду за определенную дату.

Выполнение заданий:

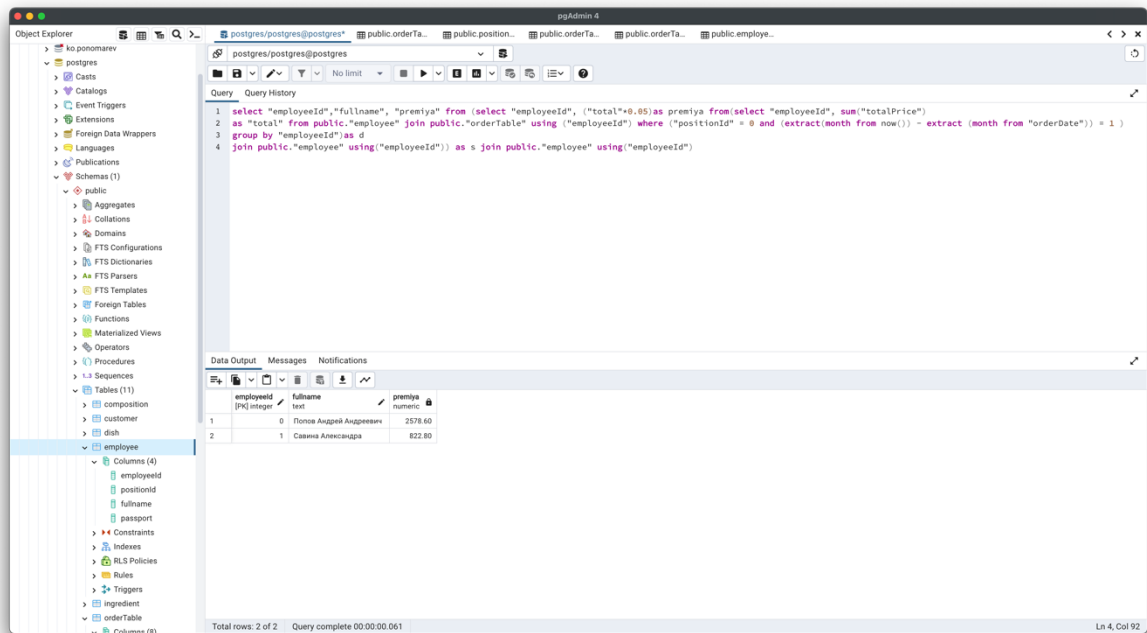
Запросы:

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



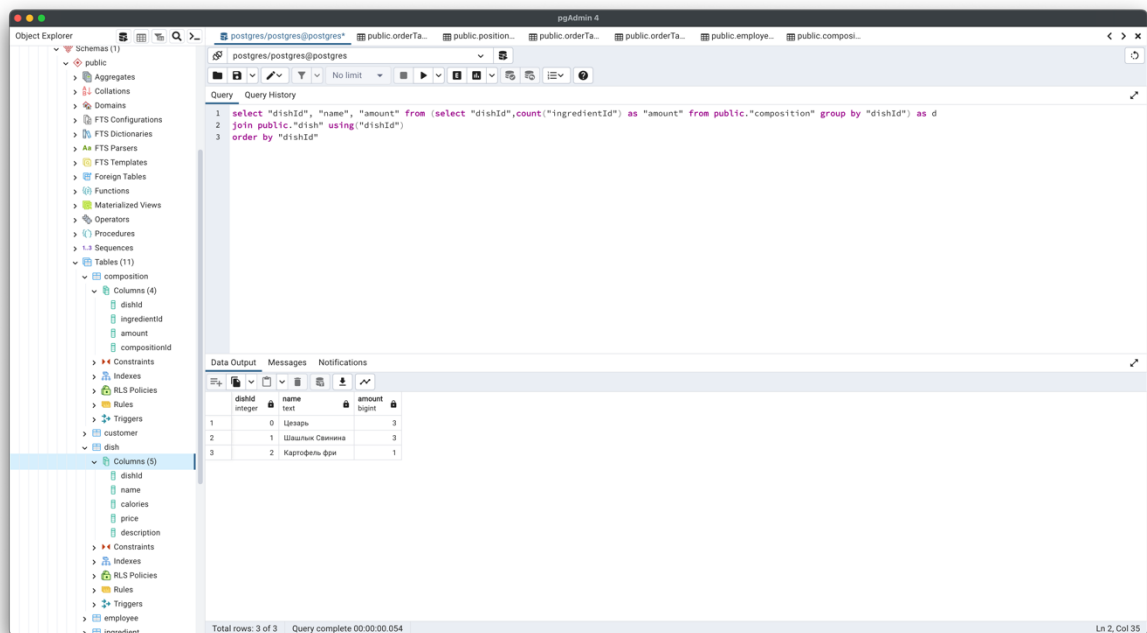
```
select * from (  
select "employeeId", sum("totalPrice") as total_price from public."employee"  
join public."orderTable" using ("employeeId")  
where ("positionId" = 0 and (extract(month from now()) - extract (month from  
"orderDate"))) = 1 )group by "employeeId")as d  
join public."employee" using("employeeId")  
where d.total_price = (  
select max("maxPrice") from (  
select sum("totalPrice") as "maxPrice" from public."employee"  
join public."orderTable" using ("employeeId")  
where ("positionId" = 0 and (extract(month from now()) - extract (month from  
"orderDate"))) = 1 )group by "employeeId") as c)
```

Рассчитать премию каждого официанта за последний месяц (5% от стоимости каждого заказа).



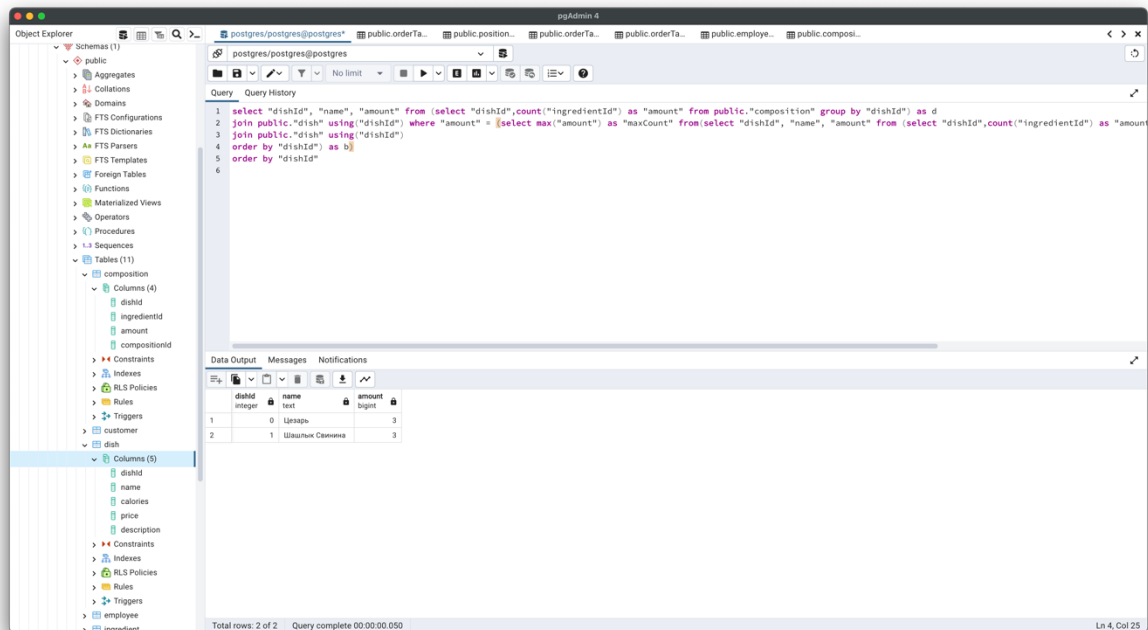
```
select "employeeId","fullname", "premiya" from (select "employeeId",
("total"*0.05)as premiya from(select "employeeId", sum("totalPrice")
as "total" from public."employee" join public."orderTable" using
("employeeId") where ("positionId" = 0 and (extract(month from now()) -
extract (month from "orderDate"))) = 1 )
group by "employeeId")as d
join public."employee" using("employeeId")) as s join public."employee"
using("employeeId")
```

Подсчитать, сколько ингредиентов содержит каждое блюдо.



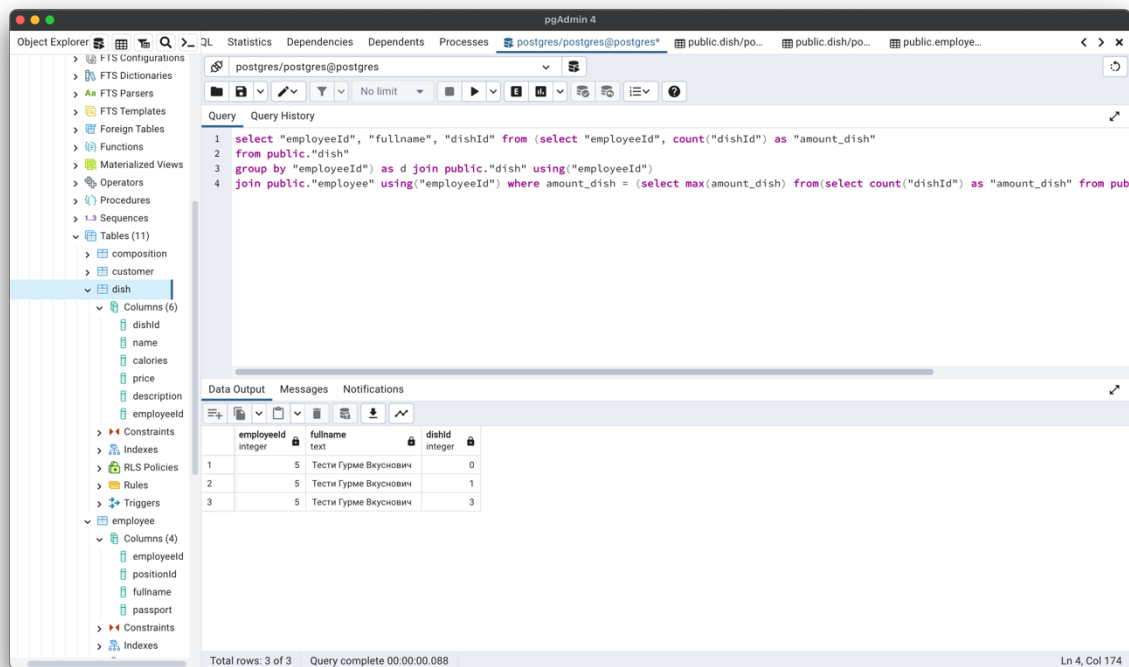
```
select "dishId", "name", "amount" from (select "dishId", count ("ingredientId") as
"amount" from public."composition" group by "dishId") as d join public."dish"
using("dishId") order by "dishId"
```

Вывести название блюда, содержащее максимальное число ингредиентов.



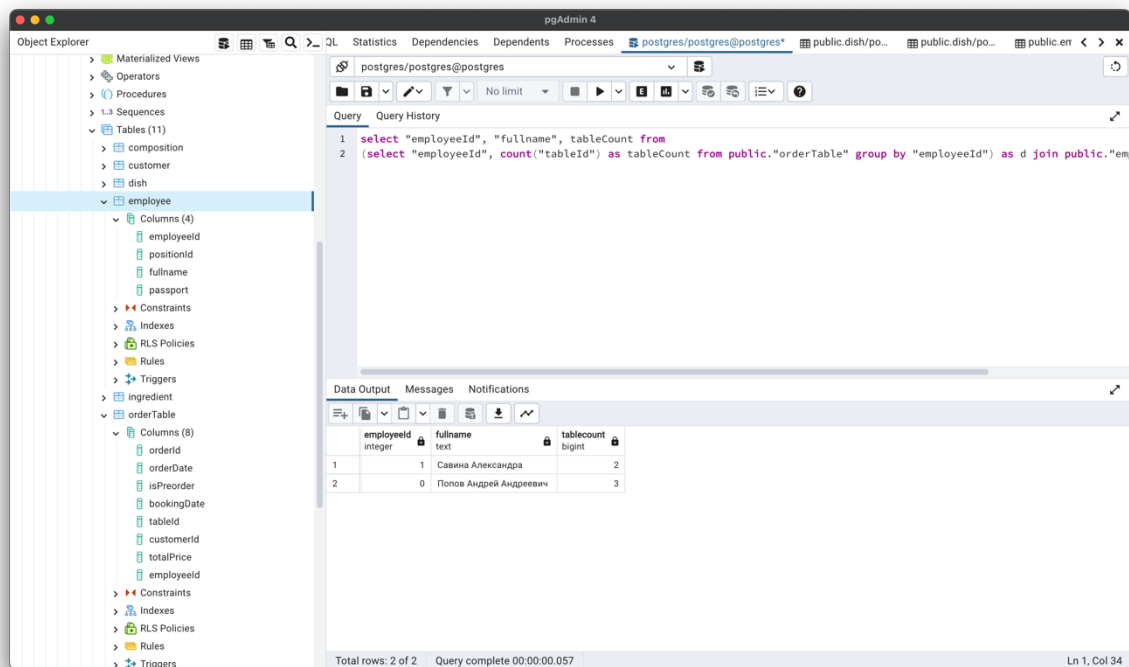
```
select "dishId", "name", "amount" from (select "dishId",count("ingredientId") as
"amount" from public."composition" group by "dishId") as d
join public."dish" using("dishId") where "amount" = (select max("amount") as
"maxCount" from(select "dishId", "name", "amount" from (select
"dishId",count("ingredientId") as "amount" from public."composition" group by
"dishId") as d
join public."dish" using("dishId")
order by "dishId") as b)
order by "dishId"
```

Какой повар может приготовить максимальное число видов блюд?



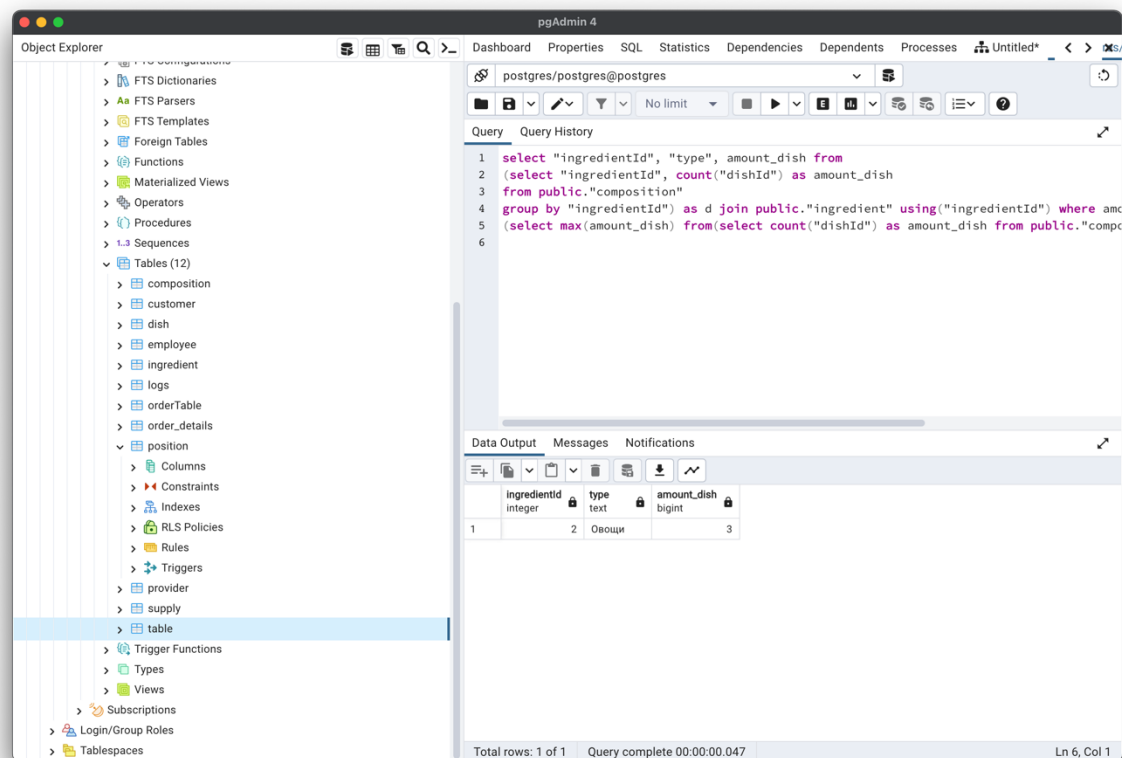
```
select "employeeId", "fullname", "dishId", from (select "employeeId",
count("dishId") as "amount_dish"
from public."dish"
group by "employeeId") as d join public."dish" using("employeeId")
join public."employee" using("employeeId") where amount_dish = (select
max(amount_dish) from(select count("dishId") as "amount_dish" from
public."dish" group by "employeeId") as f)
```


Сколько закреплено столов за каждым из официантов?



```
select "employeeId", "fullname", tableCount from
(select "employeeId", count("tableId") as tableCount from public."orderTable"
group by "employeeId") as d join public."employee" using("employeeId")
```

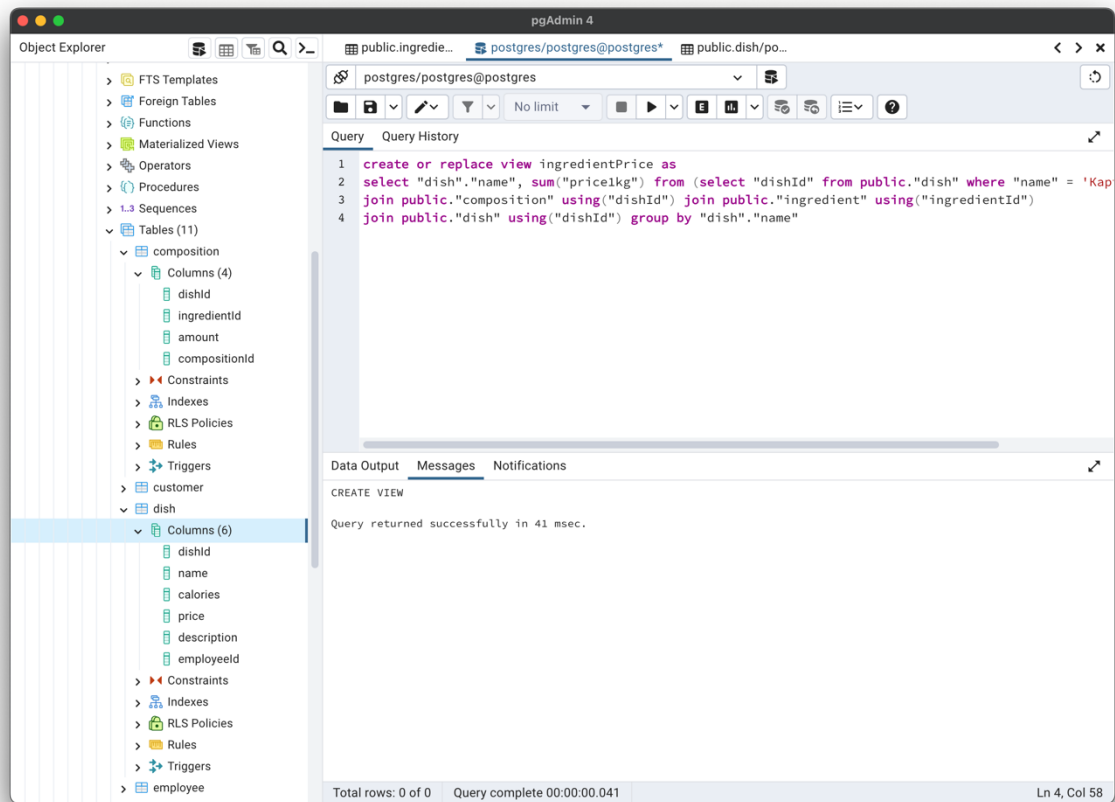
Какой из ингредиентов используется в максимальном количестве блюд?



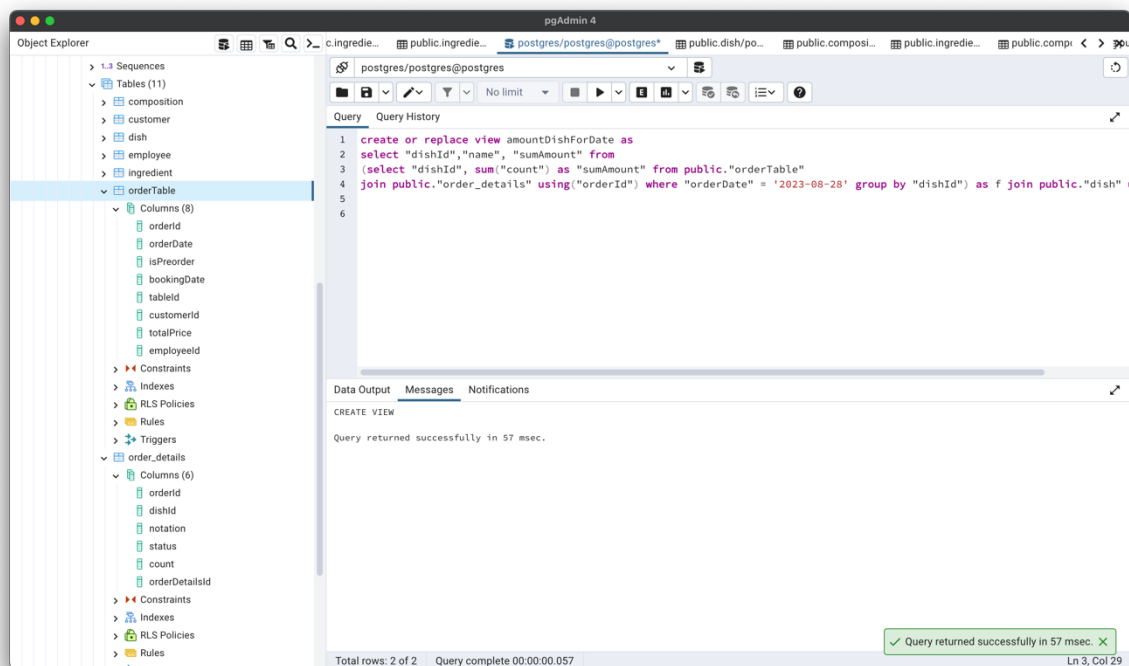
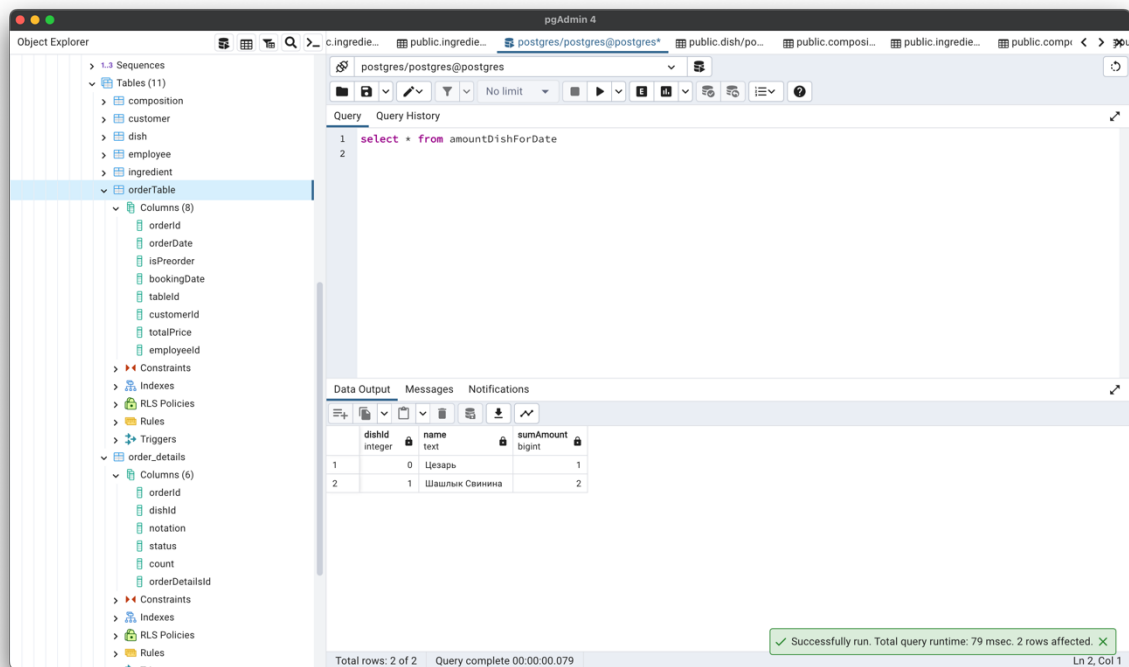
```
select "ingredientId", "type", amount_dish from
(select "ingredientId", count("dishId") as amount_dish
from public."composition"
group by "ingredientId") as d join public."ingredient" using("ingredientId")
where amount_dish =
(select max(amount_dish) from(select count("dishId") as amount_dish from
public."composition" group by "ingredientId") as f)
```

Представления:

для расчета стоимости ингредиентов для заданного блюда;

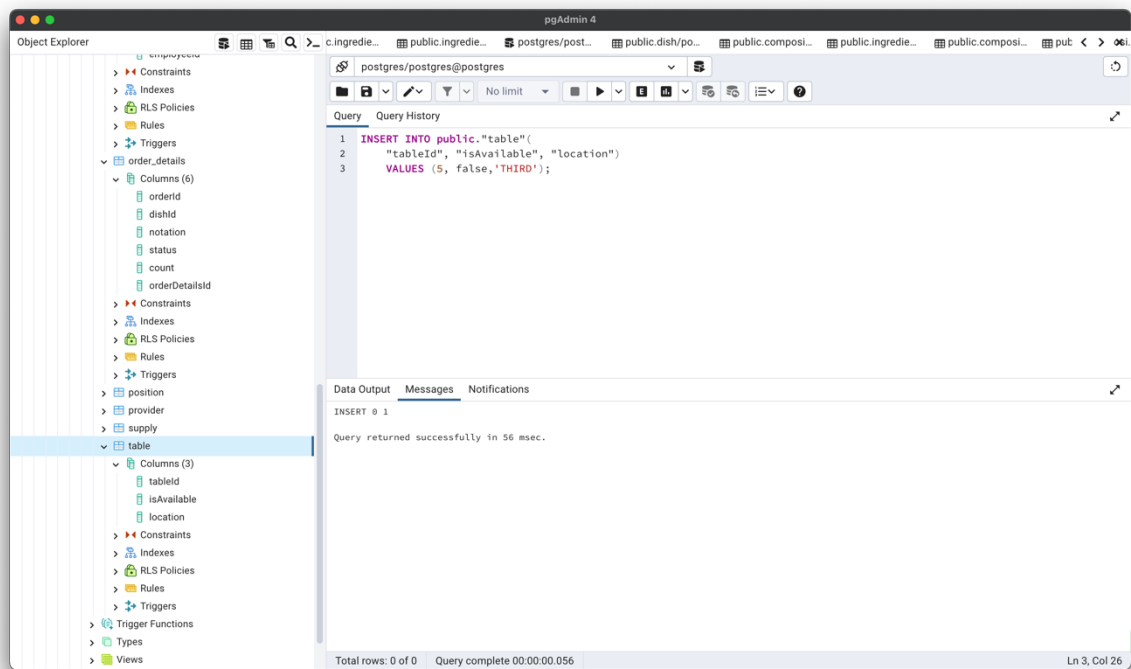


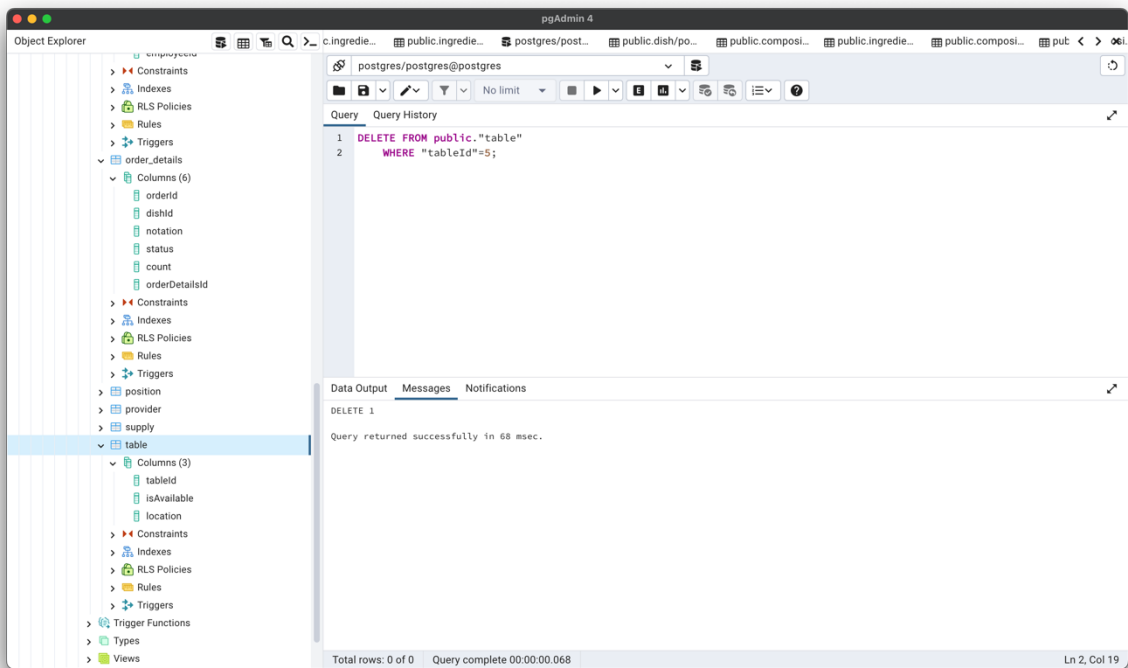
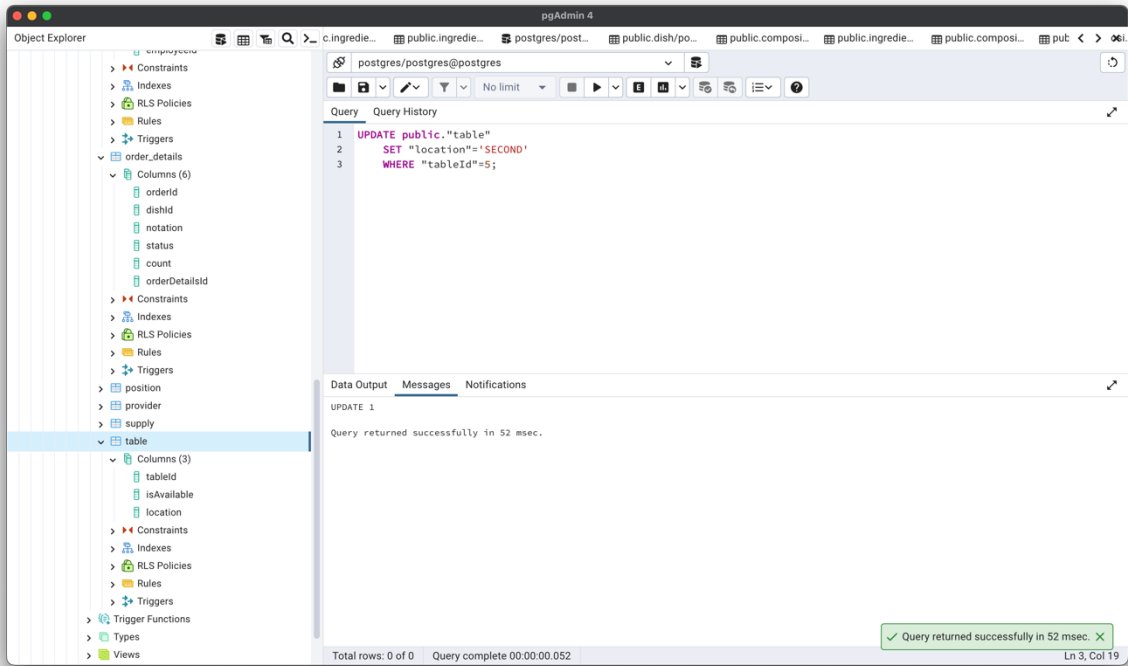
количество приготовленных блюд по каждому блюду за определенную дату.



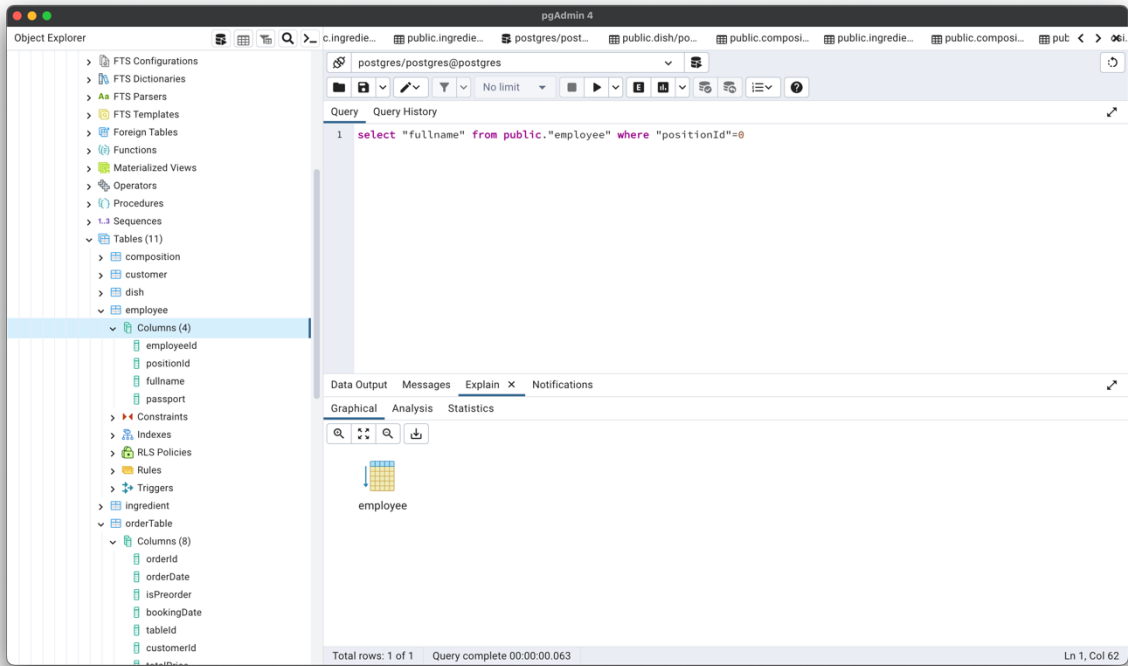
На скриншоте не видно, но using("dishId")

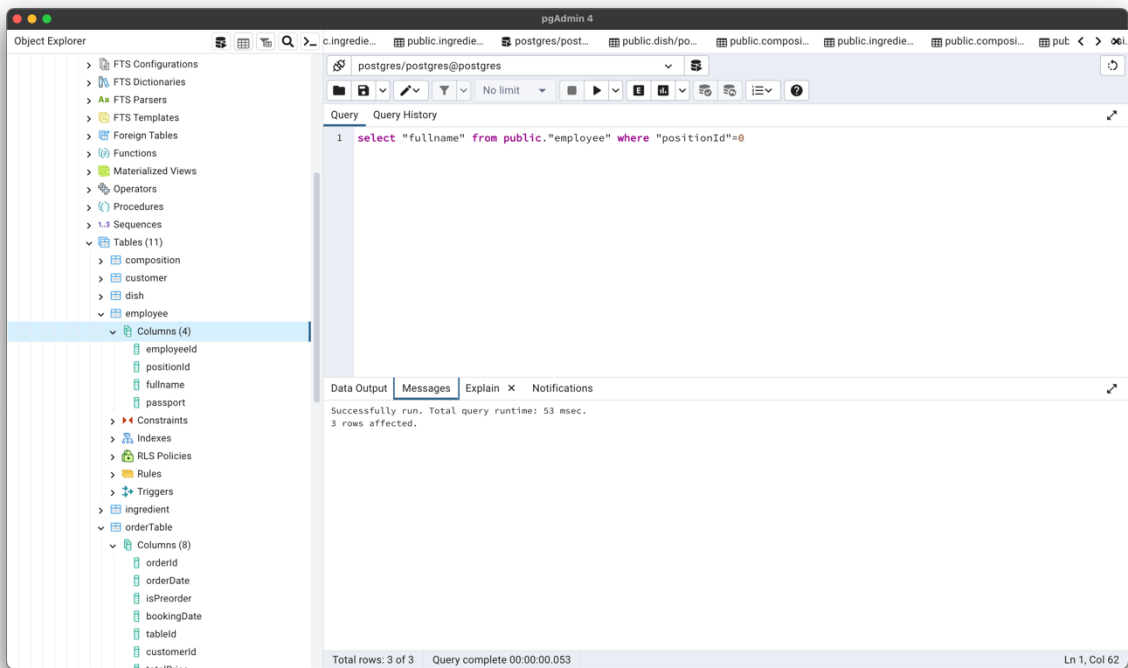
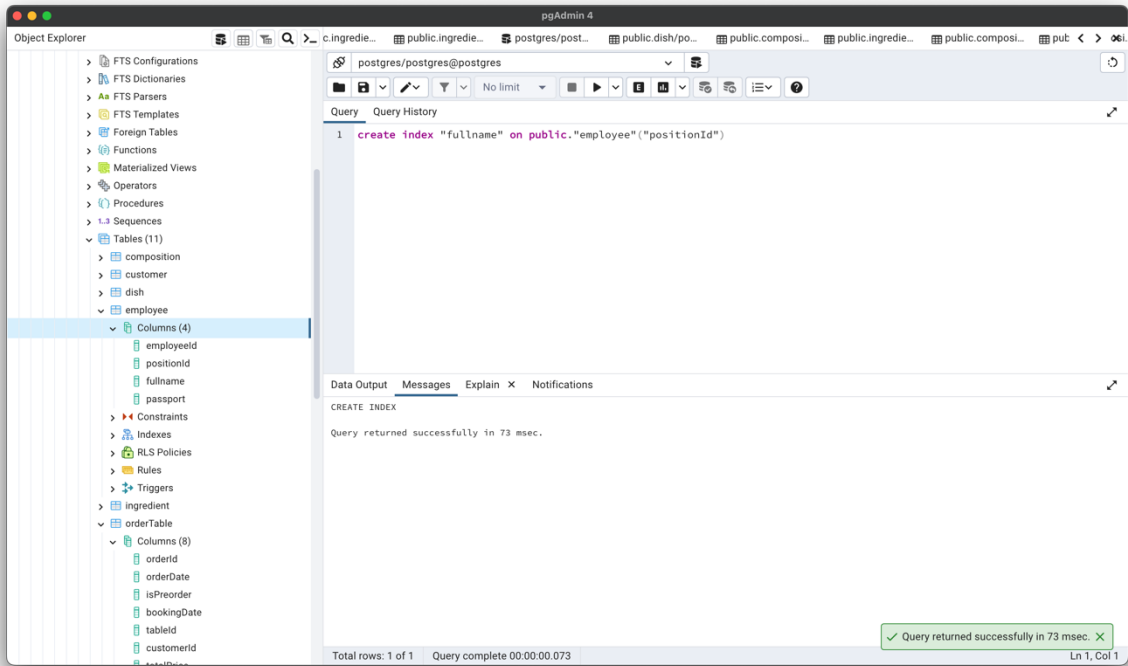
Модификация данных

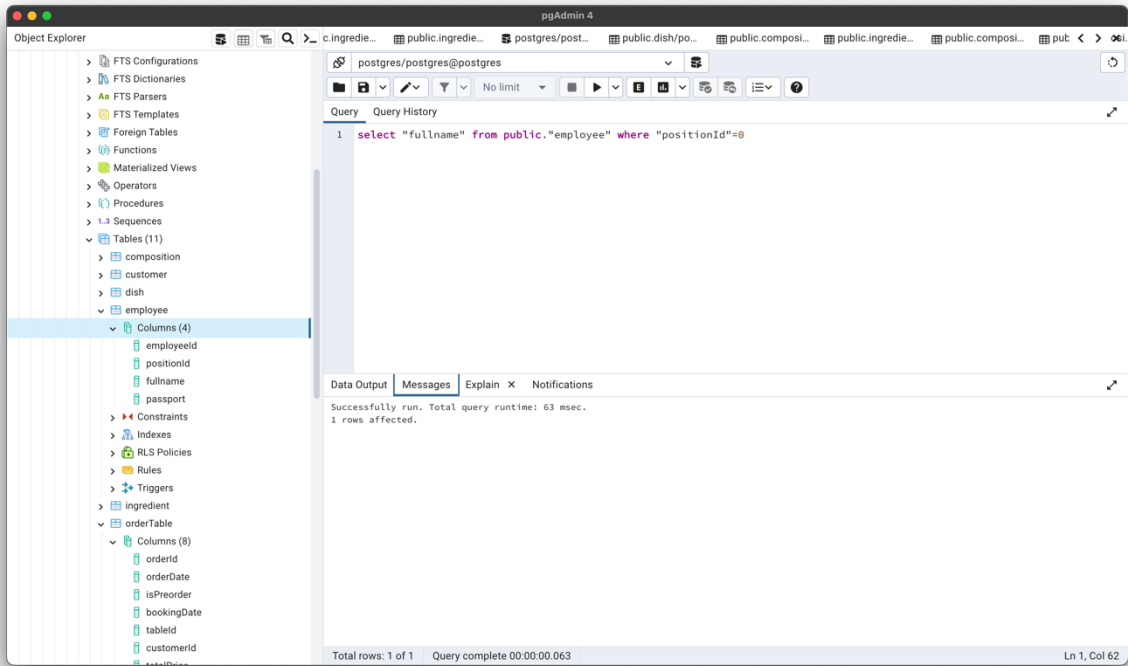




Создание индексов







pgAdmin 4

Object Explorer

- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - composition
 - customer
 - dish
 - employee
 - Columns (4)
 - employeeid
 - positionid
 - fullname
 - passport
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - ingredient
 - orderTable
 - Columns (8)
 - orderid
 - orderDate
 - isPreOrder
 - bookingDate
 - tableid
 - customerid
 - tableid
 - tableid

Query

```
1 -- select "employeeId", "fullname", tableCount from
2 -- (select "employeeId", count("tableId") as tableCount from public."orderTable" group by "employeeId") as d join public."employee"
3
4 create unique index "totalTables" on public."employee"("employeeId", "fullname")
```

Execute/Refresh

Data Output Messages Explain X Notifications

CREATE INDEX

Query returned successfully in 50 msec.

Query complete 00:00:00.050

Ln 4, Col 75

pgAdmin 4

Object Explorer

- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - composition
 - customer
 - dish
 - employee
 - Columns (4)
 - employeeid
 - positionid
 - fullname
 - passport
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - ingredient
 - orderTable
 - Columns (8)
 - orderid
 - orderDate
 - isPreOrder
 - bookingDate
 - tableid
 - customerid
 - tableid
 - tableid

Query

```
1 select "employeeId", "fullname", tableCount from
2 (select "employeeId", count("tableId") as tableCount from public."orderTable" group by "employeeId") as d join public."employee"
```

Execute/Refresh

Data Output Messages Explain X Notifications

Graphical Analysis Statistics

orderTable

Aggregate

Hash Inner Join

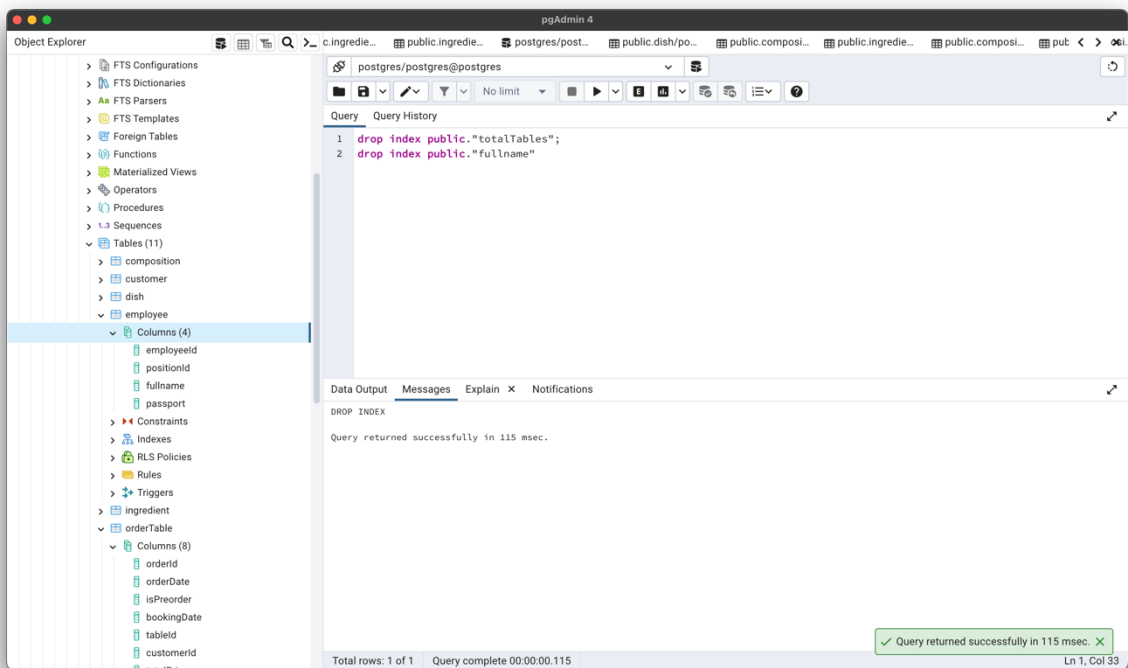
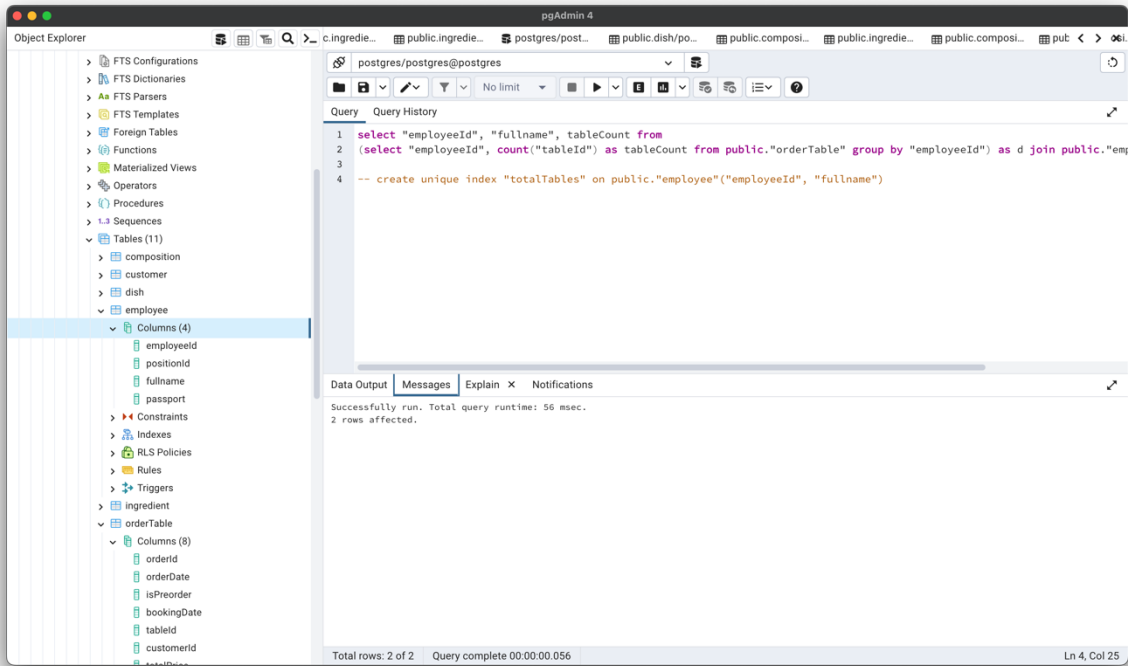
employee

Hash

Successfully run. Total query runtime: 44 msec. 1 rows affected.

Query complete 00:00:00.044

Ln 2, Col 149



Выводы

SQL запросы позволяют изменять, добавлять или удалять данные, а также составлять различные выборки, подсчитывать числовые характеристики. Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись примерно столько же. Это связано с небольшим количеством данных в таблице. Также можно сделать вывод, что индексы помогают сократить время сложного запроса, но на примере простых запросов видно, что планировщик считает что лучше просканировать обычным способом.