

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

«Реализация БД в рамках СУБД»

**Выполнил:**

Студент гр. N3347 Нгуен Тхе Вьет



(подпись)

**Проверил:**

Салихов Максим Русланович, аспирант ФБИТ

(отметка о выполнении)

(подпись)

Санкт-Петербург

2024 г.

## СОДЕРЖАНИЕ

Содержание .....	2
ВВЕДЕНИЕ .....	3
ХОД РАБОТЫ .....	4
1.1    Выбор СУБД .....	4
1.2    Создание БД .....	4
1.2.1    Код для создания всех таблиц; .....	6
1.2.2    Код для внесения данных в созданные таблицы .....	6
1.2.3    Код одной SQL-команды для модифицирования структуры таблицы .....	8
1.3    Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.....	8
1.4    Установить взаимосвязи между таблицами.....	8
1.5    Создать представления, составленные в пункте 5 лабораторной 1.....	8
1.5.1    Представление для клиентов .....	8
1.5.2    Представление для сотрудников.....	9
1.5.3    Представление для менеджера.....	10
1.6    Дополнительно. Тестовых запросов к вашей БД .....	11
1.6.1    Получить все заказы с информацией о клиентах и сотрудниках. ....	11
1.6.2    Перечислить продукты в конкретной категории.....	12
1.6.3    Найти заказы, размещенные конкретным клиентом.....	12
1.6.4    Найти the Most Popular Product .....	12
1.6.5    Расчет общего дохода за определенный период .....	13

## ВВЕДЕНИЕ

**Цель работы:** Получение навыков по работе с современными системами управления базами данных

### Задание

1. Выбрать систему управления базами данных (СУБД), которая будет использована в рамках лабораторной работы. Кратко обосновать свой выбор.
2. Создать БД в выбранной в вами СУБД на основе итоговой разработанной схемы отношений из ЛР 1. Заполните созданную вами БД информацией, сгенерируйте как минимум 7-8 кортежей с данными для каждой из ваших основных таблиц. В отчете по лабораторной работе укажите следующий SQL-код (написанный вами или сгенерированный средствами администрирования СУБД):
  - код для создания всех таблиц;
  - код для внесения данных в созданные таблицы;
  - код хотя бы одной SQL-команды для модифицирования структуры таблицы;
3. Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.
4. Установить взаимосвязи между таблицами.
5. Дополнительно. Тестовых запросов к вашей БД
6. Создать представления, составленные в пункте 5 лабораторной 1.

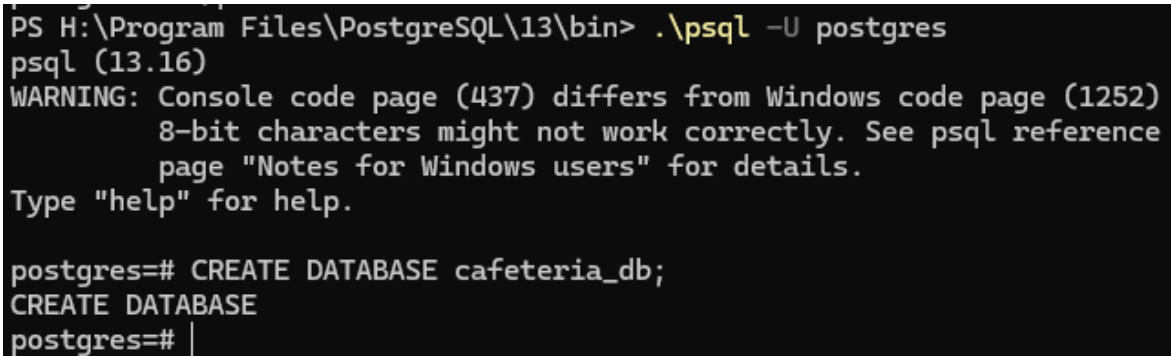
## ХОД РАБОТЫ

### 1.1 Выбор СУБД

В качестве инструментария для создания БД использована СУБД postgresSQL, а также графический интерфейс pgAdmin4.

### 1.2 Создание БД

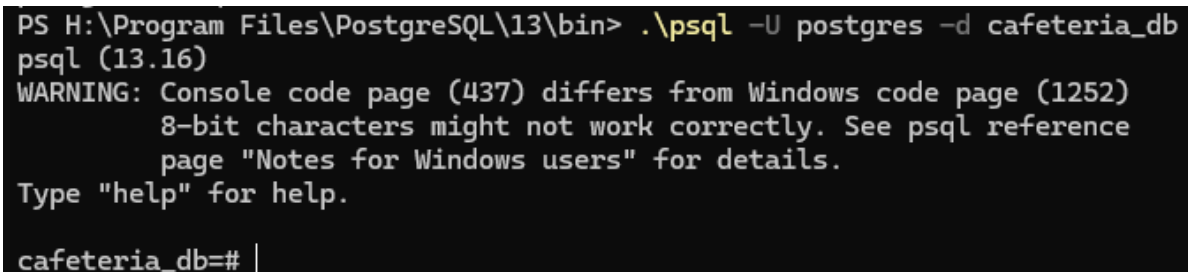
```
CREATE DATABASE cafeteria_db;
```



```
PS H:\Program Files\PostgreSQL\13\bin> .\psql -U postgres
psql (13.16)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# CREATE DATABASE cafeteria_db;
CREATE DATABASE
postgres=# |
```

Рисунок 1 – создать базу данных



```
PS H:\Program Files\PostgreSQL\13\bin> .\psql -U postgres -d cafeteria_db
psql (13.16)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

cafeteria_db=# |
```

Рисунок 2 – получить доступ к базе данных

Для начала создадим сущности в соответствии со схемой отношений на рисунке 1

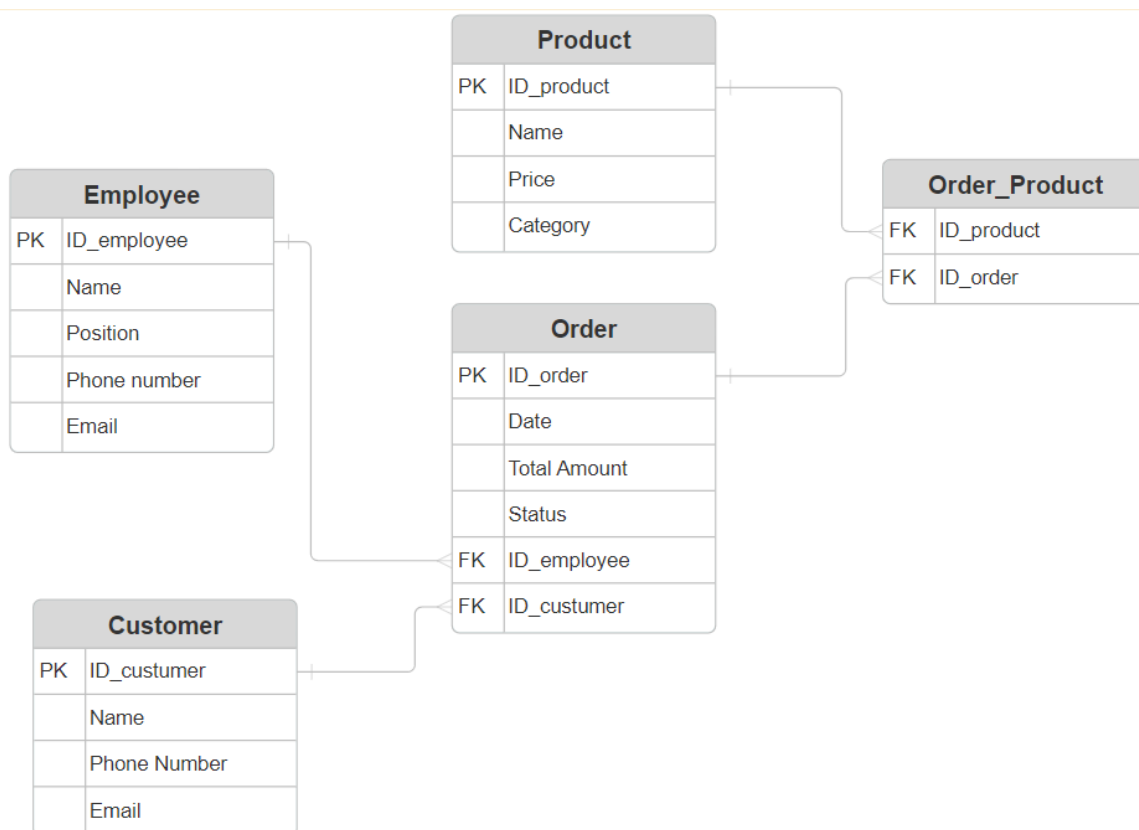


Рисунок 3 – Схема предварительных отношений БД кафетерия

### 1.2.1 Код для создания всех таблиц;

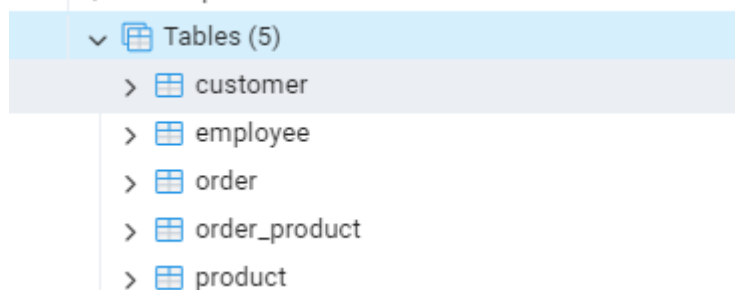
```
-- Creating Employee table
CREATE TABLE Employee (
    ID_employee SERIAL PRIMARY KEY,
    Name VARCHAR(255),
    Position VARCHAR(255),
    Phone_Number VARCHAR(20),
    Email VARCHAR(255)
);

-- Creating Product table
CREATE TABLE Product (
    ID_product SERIAL PRIMARY KEY,
    Name VARCHAR(255),
    Price DECIMAL(10,2),
    Category VARCHAR(255)
);

-- Creating Customer Table
CREATE TABLE Customer (
    Id_customer SERIAL PRIMARY KEY,
    Name VARCHAR(255),
    Phone_Number VARCHAR(20),
    Email VARCHAR(255)
);

-- Creating Order table
CREATE TABLE "order" (
    ID_order SERIAL PRIMARY KEY,
    "Date" DATE,
    Total_Amount DECIMAL(10,2),
    Status VARCHAR(255),
    Customer_ID INT,
    Employee_ID INT
);

-- Creating Order_Product table
CREATE TABLE Order_Product (
    Order_ID INT,
    Product_ID INT,
    PRIMARY KEY (Order_ID, Product_ID)
);
```



### 1.2.2 Код для внесения данных в созданные таблицы

```
-- Sample data for Employee table
INSERT INTO Employee (Name, Position, Phone_Number, Email) VALUES
('Alice Johnson', 'Waiter', '555-123-4567', 'alice.johnson@example.com'),
('Bob Williams', 'Cook', '555-987-6543', 'bob.williams@example.com'),
```

```

('Charlie Brown', 'Manager', '555-567-8901', 'charlie.brown@example.com'),
('Diana Davis', 'Waiter', '555-234-5678', 'diana.davis@example.com'),
('Eve Edwards', 'Cook', '555-876-5432', 'eve.edwards@example.com'),
('Frank Foster', 'Cashier', '555-345-6789', 'frank.foster@example.com'),
('Grace Garcia', 'Waiter', '555-765-4321', 'grace.garcia@example.com'),
('Henry Harris', 'Cook', '555-456-7890', 'henry.harris@example.com');

-- Sample data for Product table
INSERT INTO Product (Name, Price, Category) VALUES
('Coffee', 2.50, 'Beverage'),
('Tea', 2.00, 'Beverage'),
('Pizza', 12.00, 'Food'),
('Burger', 8.50, 'Food'),
('Salad', 7.00, 'Food'),
('Cake', 5.00, 'Dessert'),
('Ice Cream', 4.00, 'Dessert'),
('Soda', 1.50, 'Beverage');

-- Sample data for Customer table
INSERT INTO Customer (Name, Phone_Number, Email)
VALUES
('Ivan Ivanov', '555-111-2222', 'ian.ivanov@example.com'),
('Jane Jones', '555-333-4444', 'jane.jones@example.com'),
('Kevin Kim', '555-555-6666', 'kevin.kim@example.com'),
('Laura Lee', '555-777-8888', 'laura.lee@example.com'),
('Mike Miller', '555-999-0000', 'mike.miller@example.com'),
('Nancy Nelson', '555-222-3333', 'nancy.nelson@example.com'),
('Oscar Olsen', '555-444-5555', 'oscar.olsen@example.com'),
('Patricia Perez', '555-666-7777', 'patricia.perez@example.com');

-- Sample data for Order table
INSERT INTO "order" ("Date", Total_Amount, Status, Customer_ID, Employee_ID)
VALUES
('2024-03-01', 14.50, 'Completed', 1, 1),
('2024-03-01', 10.50, 'Completed', 2, 2),
('2024-03-03', 7.00, 'Completed', 3, 3),
('2024-03-04', 7.50, 'Completed', 4, 4),
('2024-03-05', 19.00, 'Completed', 5, 5),
('2024-03-05', 10.00, 'Completed', 6, 6),
('2024-03-07', 11.00, 'Completed', 7, 7),
('2024-03-08', 12.00, 'Completed', 8, 8),
('2024-03-09', 19.00, 'Open', 3, 1),
('2024-03-10', 2.50, 'Open', 6, 4),
('2024-03-11', 8.50, 'Cancelled', 8, 2),
('2024-03-12', 4.00, 'Open', 1, 7),
('2024-03-12', 12.00, 'Open', 5, 8);

-- Sample data for Order_Product table
INSERT INTO Order_Product (Order_ID, Product_ID)
VALUES
(1, 1),
(1, 3),
(2, 2),
(2, 4),
(3, 5),
(4, 1),
(4, 6),
(5, 3),
(5, 7),
(6, 4),
(6, 8),
(7, 1),
(7, 5),
(8, 3),

```

```
(9, 2),  
(9, 5),  
(10, 1),  
(11, 4),  
(12, 7),  
(13, 3);
```

### **1.2.3 Код одной SQL-команды для модифицирования структуры таблицы**

```
-- Adding a 'Hire Date' column to the Employee table  
ALTER TABLE Employee  
ADD "Hire Date" DATE;  
-- Removing the 'Hire Date' column from the Employee table  
ALTER TABLE Employee  
DROP COLUMN "Hire Date";
```

### **1.3 Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.**

```
-- Index on Customer_ID in the Order table  
CREATE INDEX idx_order_customer_id ON "order" (Customer_ID);  
  
-- Index on Name and Category in the Product table  
CREATE INDEX idx_product_name_category ON product (Name, Category);
```

### **1.4 Установить взаимосвязи между таблицами**

```
-- Add foreign keys to order_product table  
ALTER TABLE Order_Product  
ADD CONSTRAINT FK_Order_Product_Order  
    FOREIGN KEY (Order_ID) REFERENCES "order"(ID_order),  
ADD CONSTRAINT FK_Order_Product_Product  
    FOREIGN KEY (Product_ID) REFERENCES Product(ID_product);  
  
-- Add foreign keys to order table  
ALTER TABLE "order"  
ADD CONSTRAINT FK_Order_Customer  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Id_customer),  
ADD CONSTRAINT FK_Order_Employee  
    FOREIGN KEY (Employee_ID) REFERENCES Employee(ID_employee);
```

### **1.5 Создать представления, составленные в пункте 5 лабораторной 1.**

#### **Представления для базы данных кафетерия**

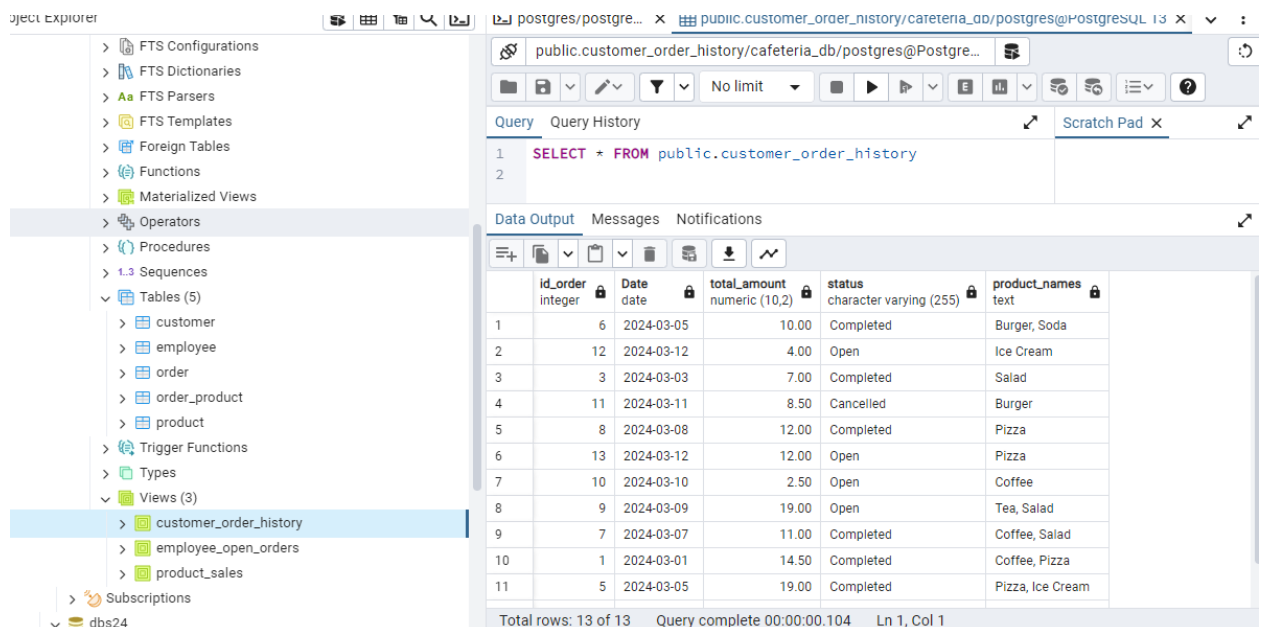
##### **1.5.1 Представление для клиентов**

- Это представление позволяет клиентам видеть свою историю заказов, включая идентификатор заказа, дату, общую сумму, статус и названия заказанных продуктов.



**Объяснение :**

```
CREATE VIEW customer_order_history AS
SELECT
    o.id_order,
    o."Date",
    o.total_amount,
    o.status,
    STRING_AGG(p.name, ', ') AS product_names
FROM
    "order" o
JOIN
    order_product op ON o.id_order = op.order_id
JOIN
    product p ON op.product_id = p.id_product
JOIN
    customer c ON o.customer_id = c.id_customer
GROUP BY
    o.id_order, o."Date", o.total_amount, o.status;
```



The screenshot displays a PostgreSQL database interface. On the left, the 'Object Explorer' shows the database structure, with 'customer\_order\_history' selected under the 'Views' folder. The main window shows the query 'SELECT \* FROM public.customer\_order\_history' and its results in a table with 13 rows. The table columns are: id\_order (integer), Date (date), totalAmount (numeric (10,2)), status (character varying (255)), and product\_names (text).

id_order	Date	totalAmount	status	product_names
1	2024-03-05	10.00	Completed	Burger, Soda
2	2024-03-12	4.00	Open	Ice Cream
3	2024-03-03	7.00	Completed	Salad
4	2024-03-11	8.50	Cancelled	Burger
5	2024-03-08	12.00	Completed	Pizza
6	2024-03-12	12.00	Open	Pizza
7	2024-03-10	2.50	Open	Coffee
8	2024-03-09	19.00	Open	Tea, Salad
9	2024-03-07	11.00	Completed	Coffee, Salad
10	2024-03-01	14.50	Completed	Coffee, Pizza
11	2024-03-05	19.00	Completed	Pizza, Ice Cream

Total rows: 13 of 13 Query complete 00:00:00.104 Ln 1, Col 1

Рисунок 4 – Представление для клиентов

### 1.5.2 Представление для сотрудников

- Это представление показывает сотрудникам список открытых заказов, включая идентификатор заказа, имя клиента, название продукта и назначенного сотрудника.

**Объяснение :**

```
CREATE VIEW employee_open_orders AS
SELECT
    o.id_order,
    c.Name AS customer_name,
    p.Name AS product_name,
    e.Name AS employee_name
FROM
    "order" o
JOIN
```

```

        order_product op ON o.id_order = op.order_id
JOIN
        product p ON op.product_id = p.id_product
JOIN
        customer c ON o.customer_id = c.id_customer
JOIN
        employee e ON o.employee_id = e.id_employee
WHERE
        o.status = 'Open';

```

The screenshot shows a PostgreSQL IDE interface. On the left is the 'Object Explorer' pane with a tree view of database objects. The 'Tables (5)' folder is expanded, and 'employee\_open\_orders' is selected. The main pane shows a query window with the following SQL query:

```
SELECT * FROM public.employee_open_orders
```

Below the query window is the 'Data Output' pane, which displays the results of the query in a table format. The table has five columns: 'id\_order' (integer), 'customer\_name' (character varying (255)), 'product\_name' (character varying (255)), and 'employee\_name' (character varying (255)). The results show five rows of data.

id_order	customer_name	product_name	employee_name
1	9 Kevin Kim	Tea	Alice Johnson
2	9 Kevin Kim	Salad	Alice Johnson
3	10 Nancy Nelson	Coffee	Diana Davis
4	12 Ivan Ivanov	Ice Cream	Grace Garcia
5	13 Mike Miller	Pizza	Henry Harris

At the bottom of the Data Output pane, it says 'Total rows: 5 of 5 Query complete 00:00:00.111 Ln 1, Col 1'.

Рисунок 5 – Представление для сотрудников

### 1.5.3 Представление для менеджера

- Это представление предоставляет информацию о продажах продуктов, отображая название каждого продукта и его общую выручку.

#### Объяснение:

```

CREATE VIEW product_sales AS
SELECT
    p.Name AS product_name,
    SUM(p.Price) AS total_revenue
FROM
    "order" o
JOIN
    order_product op ON o.id_order = op.order_id
JOIN
    product p ON op.product_id = p.id_product
WHERE
    o.status = 'Completed'
GROUP BY
    p.Name;

```

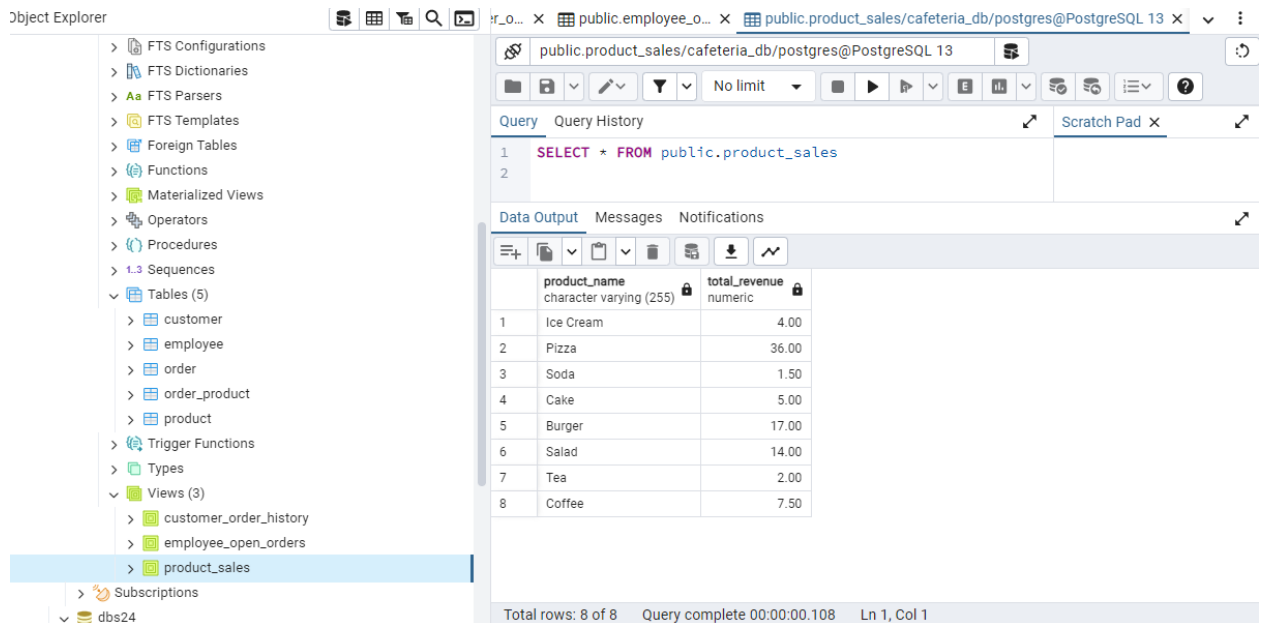


Рисунок 6 – Представление для менеджер

## 1.6 Дополнительно. Тестовых запросов к вашей БД

### 1.6.1 Получить все заказы с информацией о клиентах и сотрудниках.

```
SELECT
    o.ID_order,
    o."Date",
    o.Total_Amount,
    o.Status,
    c.Name AS CustomerName,
    e.Name AS EmployeeName
FROM
    "order" o
JOIN
    Customer c ON o.Customer_ID = c.Id_customer
JOIN
    Employee e ON o.Employee_ID = e.ID_employee;
```

Этот запрос извлекает все заказы из таблицы "order" и объединяет их с таблицами Customer и Employee, чтобы отобразить имя клиента и сотрудника, который обработал заказ.

id_order	Date	total_amount	status	customername	employeename
12	2024-03-12	4.00	Open	Ivan Ivanov	Grace Garcia
1	2024-03-01	14.50	Completed	Ivan Ivanov	Alice Johnson
2	2024-03-01	10.50	Completed	Jane Jones	Bob Williams
3	2024-03-03	7.00	Completed	Kevin Kim	Charlie Brown
9	2024-03-09	19.00	Open	Kevin Kim	Alice Johnson
4	2024-03-04	7.50	Completed	Laura Lee	Diana Davis
13	2024-03-12	12.00	Open	Mike Miller	Henry Harris
5	2024-03-05	19.00	Completed	Mike Miller	Eve Edwards
6	2024-03-05	10.00	Completed	Nancy Nelson	Frank Foster
10	2024-03-10	2.50	Open	Nancy Nelson	Diana Davis
7	2024-03-07	11.00	Completed	Oscar Olsen	Grace Garcia
8	2024-03-08	12.00	Completed	Patricia Perez	Henry Harris
11	2024-03-11	8.50	Cancelled	Patricia Perez	Bob Williams
(13 rows)					

### 1.6.2 Перечислить продукты в конкретной категории.

```
SELECT
    p.Name,
    p.Price
FROM
    Product p
WHERE
    p.Category = 'Food';
```

--Этот запрос выбирает названия и цены всех продуктов, принадлежащих к категории 'Food'.

name	price
Pizza	12.00
Burger	8.50
Salad	7.00
(3 rows)	

### 1.6.3 Найти заказы, размещенные конкретным клиентом.

```
SELECT
    o.ID_order,
    o."Date",
    o.Total_Amount,
    o.Status
FROM
    "order" o
JOIN
    Customer c ON o.Customer_ID = c.Id_customer
WHERE
    c.Name = 'Ivan Ivanov';
```

--Этот запрос извлекает сведения о заказах, размещенных клиентом по имени «Ivan Ivanov»

id_order	Date	total_amount	status
1	2024-03-01	14.50	Completed
12	2024-03-12	4.00	Open
(2 rows)			

### 1.6.4 Найти the Most Popular Product

```
SELECT
    p.Name,
    COUNT(op.Product_ID) AS OrderCount
FROM
    Order_Product op
JOIN
    Product p ON op.Product_ID = p.ID_product
GROUP BY
    p.Name
ORDER BY
    OrderCount DESC
```

```
LIMIT 1;
```

name	ordercount
Pizza	4

(1 row)

### 1.6.5 Расчет общего дохода за определенный период

```
SELECT
    SUM(o.Total_Amount) AS TotalRevenue
FROM
    "order" o
WHERE
    o."Date" BETWEEN '2024-03-01' AND '2024-03-12';
```

totalrevenue
137.50