**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»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

«Реализация БД в рамках СУБД»

**Выполнили:**

Бардышев Артём Антонович,

студент группы N3346

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Салихов Максим Русланович,

преподаватель, ФБИТ

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Содержание

[Введение 3](#_Toc210324068)

[1 Создание базы данных 4](#_Toc210324069)

[1.1 Создаем базу 4](#_Toc210324070)

[1.2 Подключаемся к базе 4](#_Toc210324071)

[2 Создание таблиц 5](#_Toc210324072)

[3 Индексация (ускорение JOIN и поиска) 10](#_Toc210324073)

[4 Заполнение тестовыми данными 11](#_Toc210324074)

[4.1 Customers 11](#_Toc210324075)

[4.2 Category 11](#_Toc210324076)

[4.3 Product 12](#_Toc210324077)

[4.4 Sku 12](#_Toc210324078)

[4.5 Price 13](#_Toc210324079)

[4.6 Store 13](#_Toc210324080)

[4.7 Inventory 14](#_Toc210324081)

[4.8 Orders 14](#_Toc210324082)

[4.9 OrderItem 15](#_Toc210324083)

[4.10 Payment 15](#_Toc210324084)

[5 Пример модификации структуры 16](#_Toc210324085)

[6 Представления 17](#_Toc210324086)

[6.1 View 1 – Мои заказы (для покупателей) 17](#_Toc210324087)

[6.2 View 2 – Наличие и цена (витрина SKU) 17](#_Toc210324088)

[6.3 View 3 – Потребность в пополнении 18](#_Toc210324089)

[6.4 View 4 – Продажи по SKU 18](#_Toc210324090)

[Заключение 20](#_Toc210324091)

[Список использованных источников 21](#_Toc210324092)

Введение

Цель работы – Получение практических навыков по созданию, наполнению и администрированию реляционных баз данных в системе управления базами данных PostgreSQL, включая создание таблиц, настройку индексов и связей, внесение данных, а также построение представлений для различных пользователей.

**Ход выполнения работы:**

1. **Создание базы данных.**  
   В PostgreSQL создана база данных apple\_store.
2. **Создание таблиц.**  
   Были созданы таблицы: Customer, Category, Product, Sku, Price, Store, Inventory, Order, OrderItem, Payment.  
   Для каждой таблицы определены первичные ключи, внешние ключи и атрибуты.
3. **Индексация.**  
   Для ускорения выполнения запросов добавлены индексы на атрибуты, используемые в связях и фильтрации.
4. **Заполнение данными.**  
   Таблицы наполнены тестовыми данными (не менее 7 записей для основных таблиц).
5. **Модификация структуры.**  
   С помощью ALTER TABLE в таблицу Customer добавлен новый атрибут BonusPoints.
6. **Создание представлений.**  
   Созданы представления для пользователей:
   * «Мои заказы» (агрегированные данные по заказам и платежам покупателя).
   * «Наличие и цена» (витрина SKU с остатками и действующей ценой).
   * «Потребность в пополнении» (анализ складских остатков).

«Продажи по SKU» (агрегация по продажам за период).

# Создание базы данных

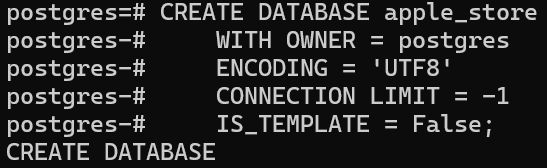
## Создаем базу

CREATE DATABASE apple\_store

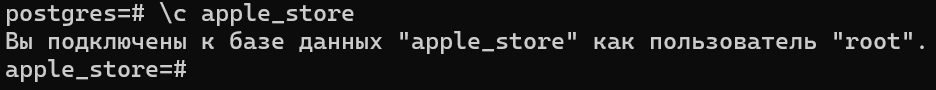
WITH OWNER = postgres

ENCODING = 'UTF8'

CONNECTION LIMIT = -1

 IS\_TEMPLATE = False;

## Подключаемся к базе

\c apple\_store

# Создание таблиц

CREATE TABLE Customer (

CustomerID SERIAL PRIMARY KEY,

FullName VARCHAR(150),

Phone VARCHAR(30),

Email VARCHAR(150),

RegistrationDate DATE

);

CREATE TABLE Category (

CategoryID SERIAL PRIMARY KEY,

CategoryName VARCHAR(100),

ParentCategoryID INT REFERENCES Category(CategoryID)

);

CREATE TABLE Product (

ProductID SERIAL PRIMARY KEY,

ProductName VARCHAR(150),

CategoryID INT REFERENCES Category(CategoryID),

Description TEXT

);

CREATE TABLE Sku (

SkuID SERIAL PRIMARY KEY,

ProductID INT REFERENCES Product(ProductID),

Color VARCHAR(50),

Memory VARCHAR(50),

ModelCode VARCHAR(50)

);

CREATE TABLE Price (

PriceID SERIAL PRIMARY KEY,

SkuID INT REFERENCES Sku(SkuID),

Price DECIMAL(10,2),

StartDate DATE,

EndDate DATE

);

CREATE TABLE Store (

StoreID SERIAL PRIMARY KEY,

StoreName VARCHAR(100),

Type VARCHAR(30),

Address VARCHAR(200)

);

CREATE TABLE Inventory (

InventoryID SERIAL PRIMARY KEY,

StoreID INT REFERENCES Store(StoreID),

SkuID INT REFERENCES Sku(SkuID),

QtyOnHand INT,

QtyReserved INT,

UpdatedAt DATE

);

CREATE TABLE "Order" (

OrderID SERIAL PRIMARY KEY,

CustomerID INT REFERENCES Customer(CustomerID),

OrderDate DATE,

Status VARCHAR(30),

StoreID INT REFERENCES Store(StoreID)

);

CREATE TABLE OrderItem (

OrderItemID SERIAL PRIMARY KEY,

OrderID INT REFERENCES "Order"(OrderID),

SkuID INT REFERENCES Sku(SkuID),

Qty INT,

UnitPriceLocked DECIMAL(10,2)

);

CREATE TABLE Payment (

PaymentID SERIAL PRIMARY KEY,

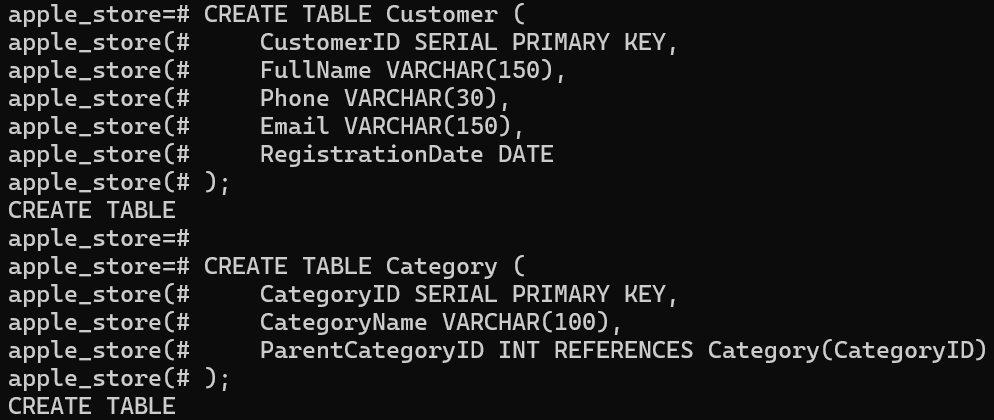
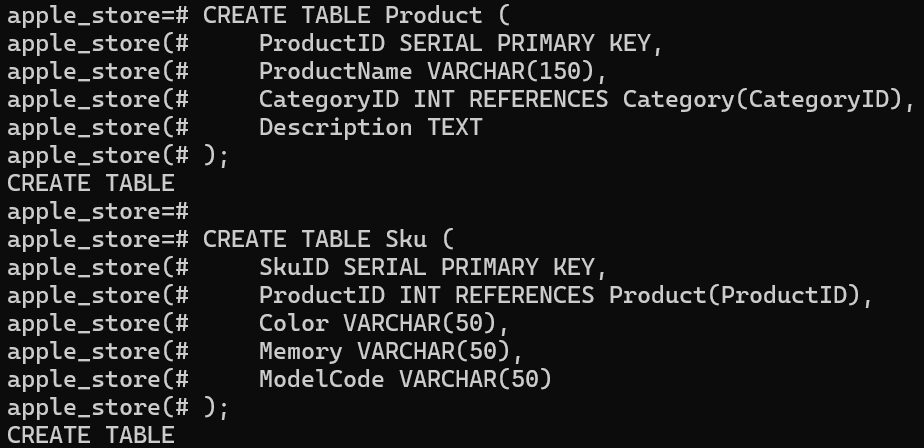
OrderID INT REFERENCES "Order"(OrderID),

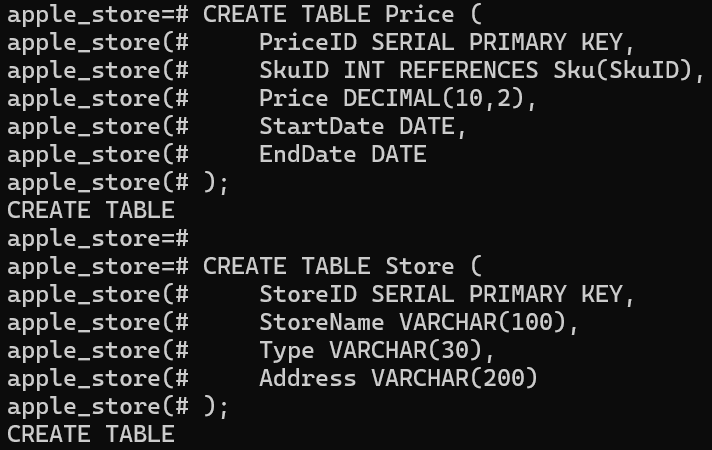
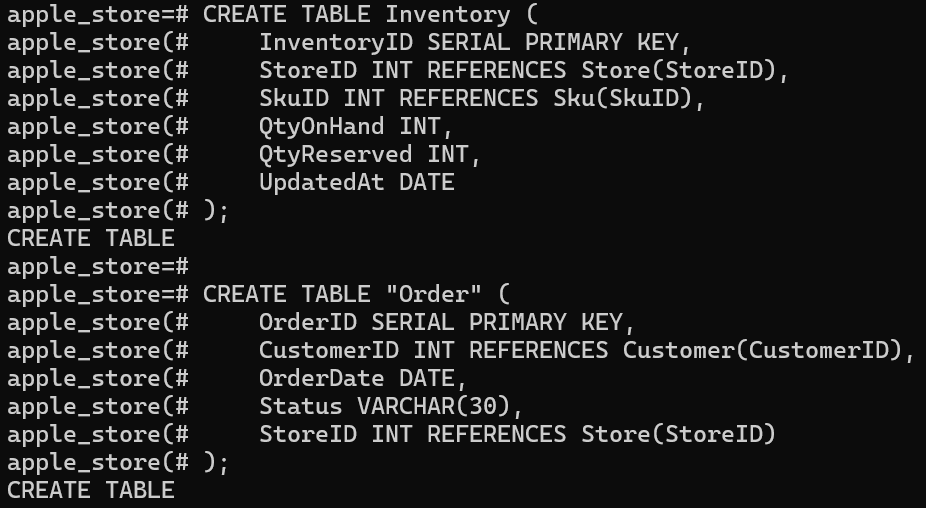
Amount DECIMAL(10,2),

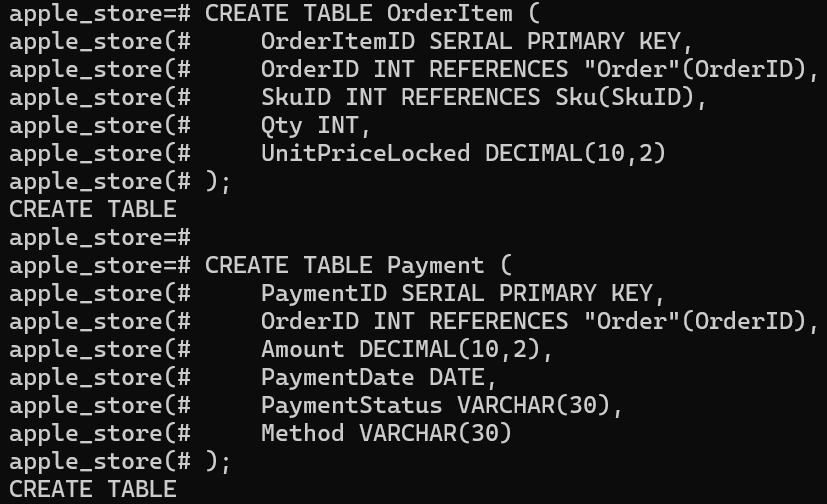
PaymentDate DATE,

PaymentStatus VARCHAR(30),

Method VARCHAR(30)

);

Продолжение

Продолжение

# Индексация (ускорение JOIN и поиска)

CREATE INDEX idx\_product\_category ON Product(CategoryID);

CREATE INDEX idx\_sku\_product ON Sku(ProductID);

CREATE INDEX idx\_price\_sku ON Price(SkuID);

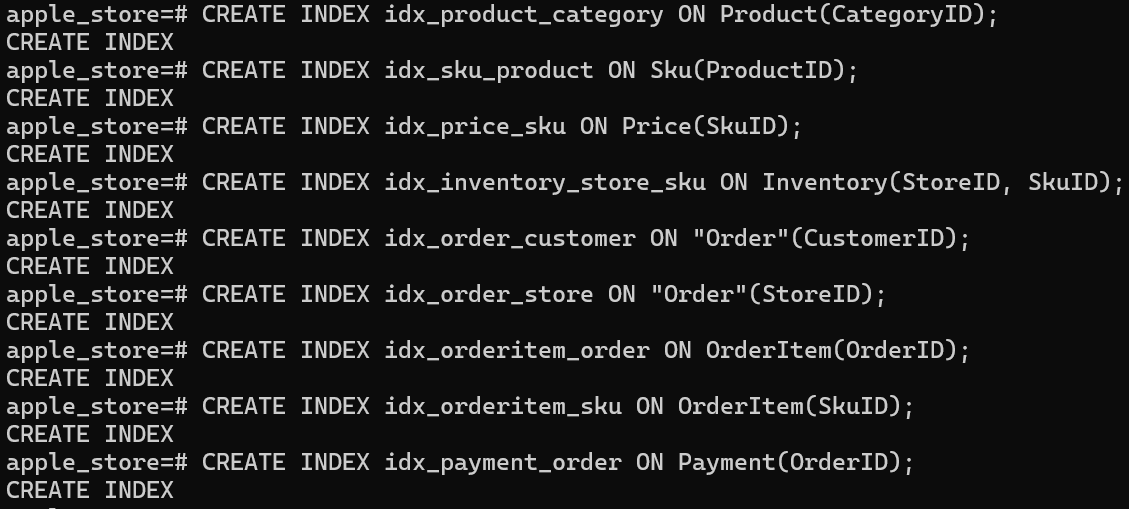
CREATE INDEX idx\_inventory\_store\_sku ON Inventory(StoreID, SkuID);

CREATE INDEX idx\_order\_customer ON "Order"(CustomerID);

CREATE INDEX idx\_order\_store ON "Order"(StoreID);

CREATE INDEX idx\_orderitem\_order ON OrderItem(OrderID);

CREATE INDEX idx\_orderitem\_sku ON OrderItem(SkuID);

CREATE INDEX idx\_payment\_order ON Payment(OrderID);

# Заполнение тестовыми данными

## Customers

INSERT INTO Customer (FullName, Phone, Email, RegistrationDate) VALUES

('Иванов Иван', '+79991234567', 'ivanov@mail.ru', '2024-01-12'),

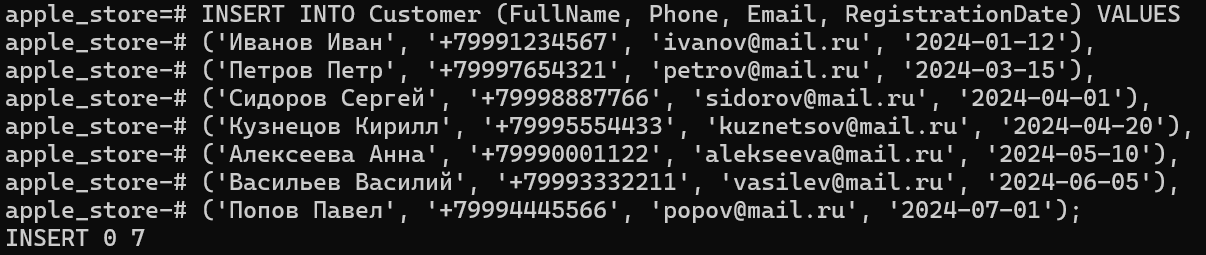
('Петров Петр', '+79997654321', 'petrov@mail.ru', '2024-03-15'),

('Сидоров Сергей', '+79998887766', 'sidorov@mail.ru', '2024-04-01'),

('Кузнецов Кирилл', '+79995554433', 'kuznetsov@mail.ru', '2024-04-20'),

('Алексеева Анна', '+79990001122', 'alekseeva@mail.ru', '2024-05-10'),

('Васильев Василий', '+79993332211', 'vasilev@mail.ru', '2024-06-05'),

('Попов Павел', '+79994445566', 'popov@mail.ru', '2024-07-01');

## Category

INSERT INTO Category (CategoryName, ParentCategoryID) VALUES

('iPhone', NULL),

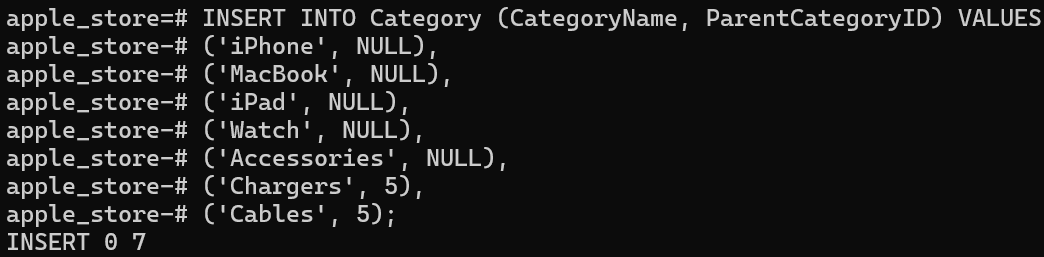
('MacBook', NULL),

('iPad', NULL),

('Watch', NULL),

('Accessories', NULL),

('Chargers', 5),

('Cables', 5);

## Product

INSERT INTO Product (ProductName, CategoryID, Description) VALUES

('iPhone 15 Pro', 1, 'Apple flagship smartphone'),

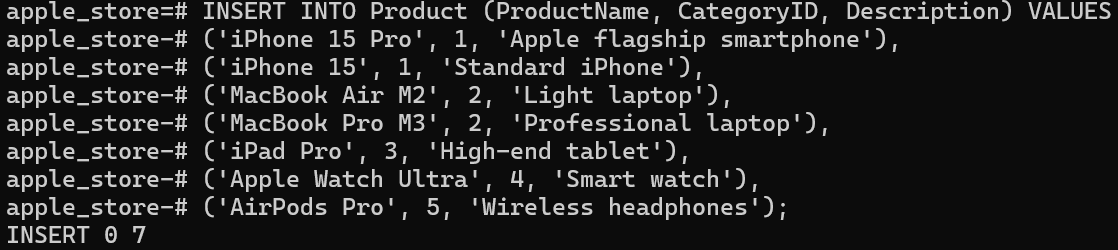
('iPhone 15', 1, 'Standard iPhone'),

('MacBook Air M2', 2, 'Light laptop'),

('MacBook Pro M3', 2, 'Professional laptop'),

('iPad Pro', 3, 'High-end tablet'),

('Apple Watch Ultra', 4, 'Smart watch'),

('AirPods Pro', 5, 'Wireless headphones');

## Sku

INSERT INTO Sku (ProductID, Color, Memory, ModelCode) VALUES

(1, 'Black', '256GB', 'IP15P-B-256'),

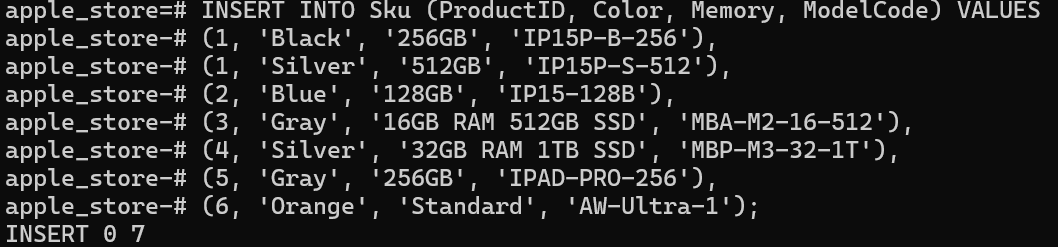
(1, 'Silver', '512GB', 'IP15P-S-512'),

(2, 'Blue', '128GB', 'IP15-128B'),

(3, 'Gray', '16GB RAM 512GB SSD', 'MBA-M2-16-512'),

(4, 'Silver', '32GB RAM 1TB SSD', 'MBP-M3-32-1T'),

(5, 'Gray', '256GB', 'IPAD-PRO-256'),

(6, 'Orange', 'Standard', 'AW-Ultra-1');

## Price

INSERT INTO Price (SkuID, Price, StartDate, EndDate) VALUES

(1, 120000, '2024-01-01', '2024-12-31'),

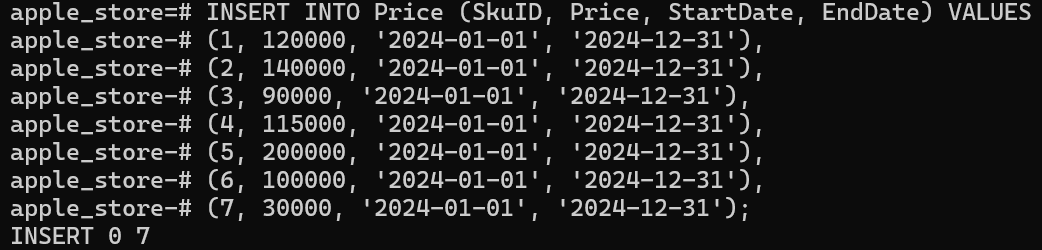
(2, 140000, '2024-01-01', '2024-12-31'),

(3, 90000, '2024-01-01', '2024-12-31'),

(4, 115000, '2024-01-01', '2024-12-31'),

(5, 200000, '2024-01-01', '2024-12-31'),

(6, 100000, '2024-01-01', '2024-12-31'),

(7, 30000, '2024-01-01', '2024-12-31');

## Store

INSERT INTO Store (StoreName, Type, Address) VALUES

('Apple Store Невский', 'retail', 'СПб, Невский 100'),

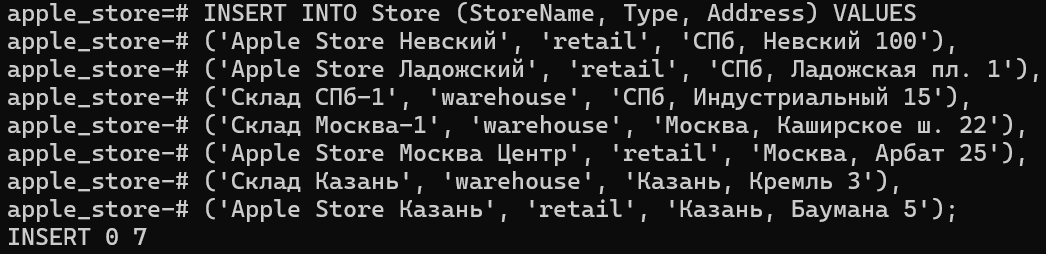
('Apple Store Ладожский', 'retail', 'СПб, Ладожская пл. 1'),

('Склад СПб-1', 'warehouse', 'СПб, Индустриальный 15'),

('Склад Москва-1', 'warehouse', 'Москва, Каширское ш. 22'),

('Apple Store Москва Центр', 'retail', 'Москва, Арбат 25'),

('Склад Казань', 'warehouse', 'Казань, Кремль 3'),

('Apple Store Казань', 'retail', 'Казань, Баумана 5');

## Inventory

INSERT INTO Inventory (StoreID, SkuID, QtyOnHand, QtyReserved, UpdatedAt) VALUES

(1, 1, 10, 2, '2024-05-01'),

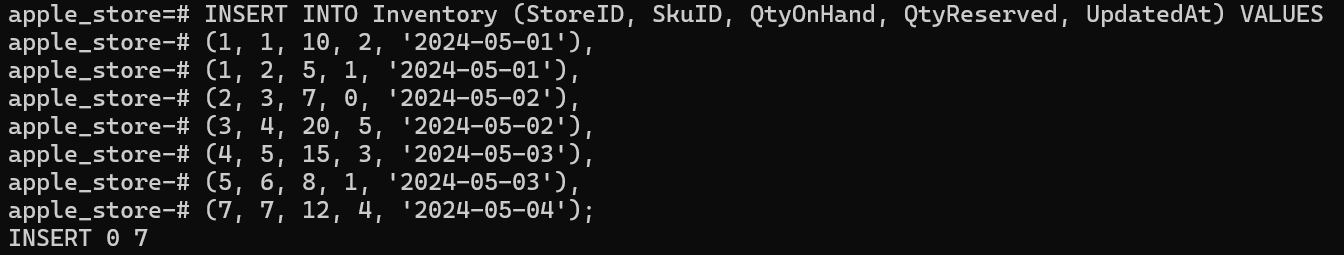
(1, 2, 5, 1, '2024-05-01'),

(2, 3, 7, 0, '2024-05-02'),

(3, 4, 20, 5, '2024-05-02'),

(4, 5, 15, 3, '2024-05-03'),

(5, 6, 8, 1, '2024-05-03'),

(7, 7, 12, 4, '2024-05-04');

## Orders

INSERT INTO "Order" (CustomerID, OrderDate, Status, StoreID) VALUES

(1, '2024-05-10', 'completed', 1),

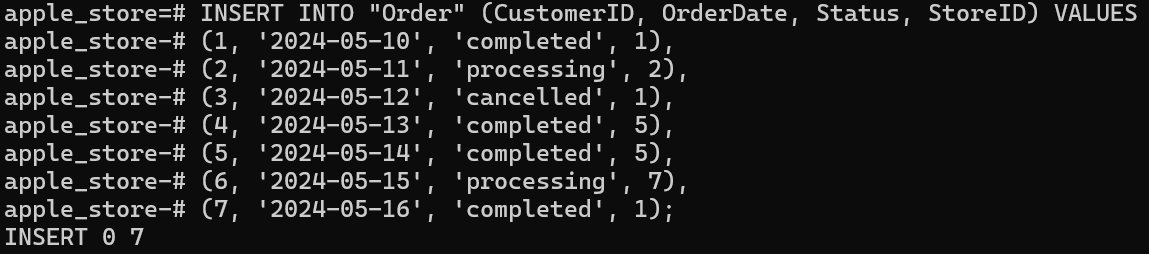
(2, '2024-05-11', 'processing', 2),

(3, '2024-05-12', 'cancelled', 1),

(4, '2024-05-13', 'completed', 5),

(5, '2024-05-14', 'completed', 5),

(6, '2024-05-15', 'processing', 7),

(7, '2024-05-16', 'completed', 1);

## OrderItem

INSERT INTO OrderItem (OrderID, SkuID, Qty, UnitPriceLocked) VALUES

(1, 1, 1, 120000),

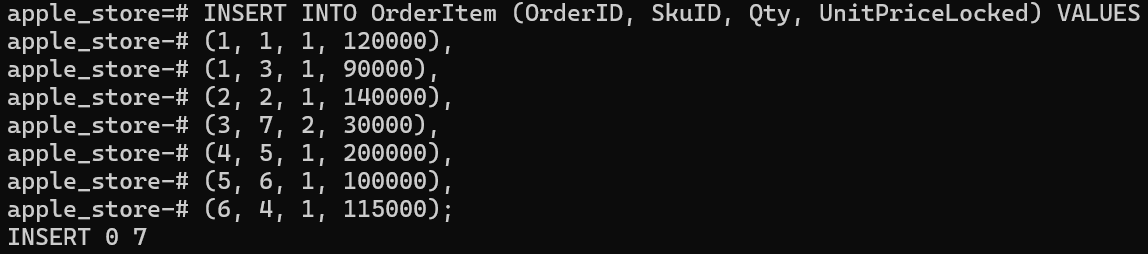
(1, 3, 1, 90000),

(2, 2, 1, 140000),

(3, 7, 2, 30000),

(4, 5, 1, 200000),

(5, 6, 1, 100000),

(6, 4, 1, 115000);

## Payment

INSERT INTO Payment (OrderID, Amount, PaymentDate, PaymentStatus, Method) VALUES

(1, 210000, '2024-05-10', 'success', 'Card'),

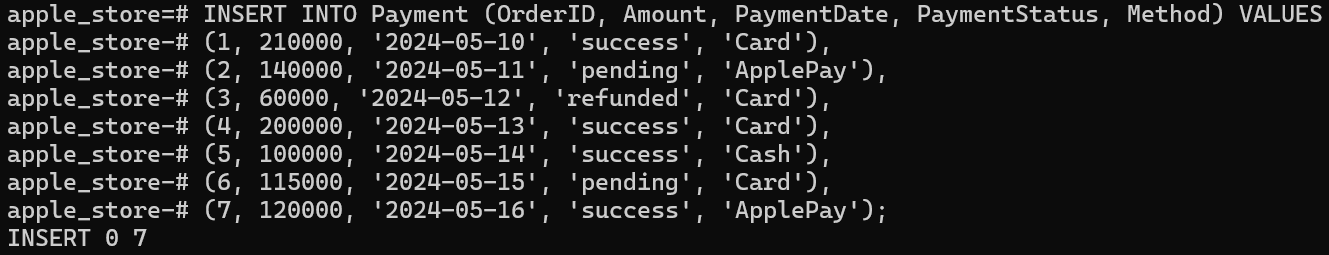
(2, 140000, '2024-05-11', 'pending', 'ApplePay'),

(3, 60000, '2024-05-12', 'refunded', 'Card'),

(4, 200000, '2024-05-13', 'success', 'Card'),

(5, 100000, '2024-05-14', 'success', 'Cash'),

(6, 115000, '2024-05-15', 'pending', 'Card'),

(7, 120000, '2024-05-16', 'success', 'ApplePay');

# Пример модификации структуры

ALTER TABLE Customer ADD COLUMN BonusPoints INT DEFAULT 0;

# Представления

## View 1 – Мои заказы (для покупателей)

CREATE VIEW MyOrders AS

SELECT o.OrderID, o.OrderDate, o.Status,

SUM(oi.Qty \* oi.UnitPriceLocked) AS TotalSum,

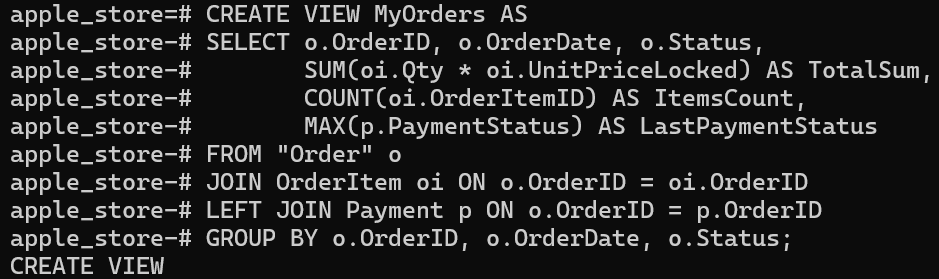
COUNT(oi.OrderItemID) AS ItemsCount,

MAX(p.PaymentStatus) AS LastPaymentStatus

FROM "Order" o

JOIN OrderItem oi ON o.OrderID = oi.OrderID

LEFT JOIN Payment p ON o.OrderID = p.OrderID

GROUP BY o.OrderID, o.OrderDate, o.Status;

## View 2 – Наличие и цена (витрина SKU)

CREATE VIEW StockAndPrice AS

SELECT pr.ProductName, s.Color, s.Memory, s.ModelCode,

p.Price, (i.QtyOnHand - i.QtyReserved) AS Available, st.StoreName

FROM Product pr

JOIN Sku s ON pr.ProductID = s.ProductID

JOIN Price p ON s.SkuID = p.SkuID

JOIN Inventory i ON s.SkuID = i.SkuID

JOIN Store st ON i.StoreID = st.StoreID

WHERE CURRENT\_DATE BETWEEN p.StartDate AND p.EndDate;

## View 3 – Потребность в пополнении

CREATE VIEW Replenishment AS

SELECT st.StoreName, s.ModelCode,

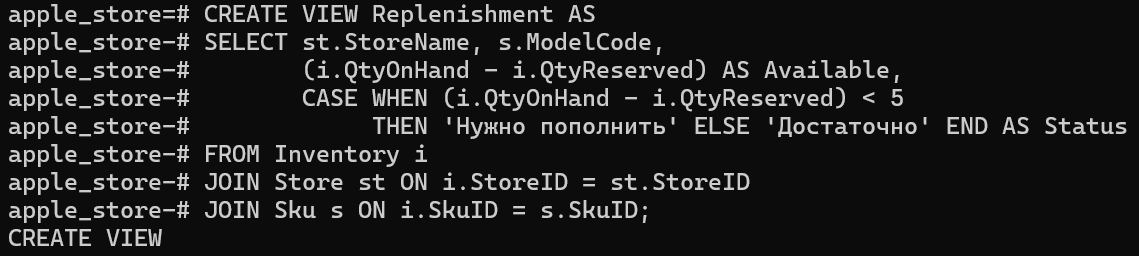
(i.QtyOnHand - i.QtyReserved) AS Available,

CASE WHEN (i.QtyOnHand - i.QtyReserved) < 5

THEN 'Нужно пополнить' ELSE 'Достаточно' END AS Status

FROM Inventory i

JOIN Store st ON i.StoreID = st.StoreID

JOIN Sku s ON i.SkuID = s.SkuID;

## View 4 – Продажи по SKU

CREATE VIEW SalesBySku AS

SELECT s.ModelCode, s.Color, s.Memory,

SUM(oi.Qty) AS TotalQty,

SUM(oi.Qty \* oi.UnitPriceLocked) AS Revenue,

AVG(oi.UnitPriceLocked) AS AvgPrice

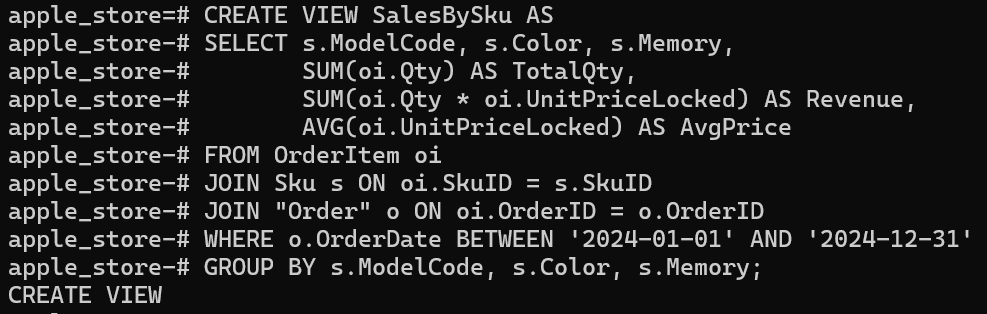
FROM OrderItem oi

JOIN Sku s ON oi.SkuID = s.SkuID

JOIN "Order" o ON oi.OrderID = o.OrderID

WHERE o.OrderDate BETWEEN '2024-01-01' AND '2024-12-31'

GROUP BY s.ModelCode, s.Color, s.Memory;



Заключение

В ходе лабораторной работы была реализована база данных «Магазин Apple» в СУБД PostgreSQL. Созданы таблицы в соответствии с инфологической моделью из ЛР №1, заданы связи и индексы. Таблицы наполнены тестовыми данными, выполнено изменение структуры и построены представления для различных пользователей.  
Полученные навыки позволяют уверенно работать с реляционными БД на уровне создания структуры, наполнения данными, оптимизации запросов и организации удобных представлений для конечных пользователей.

Список использованных источников

1. Новиков Б. А., Горшкова Е. А., Графеева Н. Г. Основы технологий баз данных. – 2-е изд. – М.: ДМК Пресс, 2020. – 582 с.
2. Хомоненко А. Д. (ред.). Базы данных. – 6-е изд., доп. – СПб.: КОРОНА-Век, 2009. – 736 с.
3. Документация PostgreSQL: <https://www.postgresql.org/docs/>.