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)

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

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).

1 СОЗДАНИЕ БАЗЫ ДАННЫХ

1.1 Создаем базу

```
CREATE DATABASE apple_store
    WITH OWNER = postgres
    ENCODING = 'UTF8'
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
```

```
postgres=# CREATE DATABASE apple_store
postgres-#     WITH OWNER = postgres
postgres-#     ENCODING = 'UTF8'
postgres-#     CONNECTION LIMIT = -1
postgres-#     IS_TEMPLATE = False;
CREATE DATABASE
```

1.2 Подключаемся к базе

```
\c apple_store
```

```
postgres=# \c apple_store
Вы подключены к базе данных "apple_store" как пользователь "root".
apple_store=#
```

2 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)
);
```

```
apple_store=# CREATE TABLE Customer (
apple_store(#     CustomerID SERIAL PRIMARY KEY,
apple_store(#     FullName VARCHAR(150),
apple_store(#     Phone VARCHAR(30),
apple_store(#     Email VARCHAR(150),
apple_store(#     RegistrationDate DATE
apple_store(# );
CREATE TABLE
apple_store=#
apple_store=# CREATE TABLE Category (
apple_store(#     CategoryID SERIAL PRIMARY KEY,
apple_store(#     CategoryName VARCHAR(100),
apple_store(#     ParentCategoryID INT REFERENCES Category(CategoryID)
apple_store(# );
CREATE TABLE
apple_store=# CREATE TABLE Product (
apple_store(#     ProductID SERIAL PRIMARY KEY,
apple_store(#     ProductName VARCHAR(150),
apple_store(#     CategoryID INT REFERENCES Category(CategoryID),
apple_store(#     Description TEXT
apple_store(# );
CREATE TABLE
apple_store=#
apple_store=# CREATE TABLE Sku (
apple_store(#     SkuID SERIAL PRIMARY KEY,
apple_store(#     ProductID INT REFERENCES Product(ProductID),
apple_store(#     Color VARCHAR(50),
apple_store(#     Memory VARCHAR(50),
apple_store(#     ModelCode VARCHAR(50)
apple_store(# );
CREATE TABLE
```

Продолжение

```
apple_store=# CREATE TABLE Price (
apple_store(#     PriceID SERIAL PRIMARY KEY,
apple_store(#     SkuID INT REFERENCES Sku(SkuID),
apple_store(#     Price DECIMAL(10,2),
apple_store(#     StartDate DATE,
apple_store(#     EndDate DATE
apple_store(# );
CREATE TABLE
apple_store=#
apple_store=# CREATE TABLE Store (
apple_store(#     StoreID SERIAL PRIMARY KEY,
apple_store(#     StoreName VARCHAR(100),
apple_store(#     Type VARCHAR(30),
apple_store(#     Address VARCHAR(200)
apple_store(# );
CREATE TABLE
```

```
apple_store=# CREATE TABLE Inventory (
apple_store(#     InventoryID SERIAL PRIMARY KEY,
apple_store(#     StoreID INT REFERENCES Store(StoreID),
apple_store(#     SkuID INT REFERENCES Sku(SkuID),
apple_store(#     QtyOnHand INT,
apple_store(#     QtyReserved INT,
apple_store(#     UpdatedAt DATE
apple_store(# );
CREATE TABLE
apple_store=#
apple_store=# CREATE TABLE "Order" (
apple_store(#     OrderID SERIAL PRIMARY KEY,
apple_store(#     CustomerID INT REFERENCES Customer(CustomerID),
apple_store(#     OrderDate DATE,
apple_store(#     Status VARCHAR(30),
apple_store(#     StoreID INT REFERENCES Store(StoreID)
apple_store(# );
CREATE TABLE
```

Продолжение

```
apple_store=# CREATE TABLE OrderItem (
apple_store(#     OrderItemID SERIAL PRIMARY KEY,
apple_store(#     OrderID INT REFERENCES "Order"(OrderID),
apple_store(#     SkuID INT REFERENCES Sku(SkuID),
apple_store(#     Qty INT,
apple_store(#     UnitPriceLocked DECIMAL(10,2)
apple_store(# );
CREATE TABLE
apple_store=#
apple_store=# CREATE TABLE Payment (
apple_store(#     PaymentID SERIAL PRIMARY KEY,
apple_store(#     OrderID INT REFERENCES "Order"(OrderID),
apple_store(#     Amount DECIMAL(10,2),
apple_store(#     PaymentDate DATE,
apple_store(#     PaymentStatus VARCHAR(30),
apple_store(#     Method VARCHAR(30)
apple_store(# );
CREATE TABLE
```

3 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);

```
apple_store=# CREATE INDEX idx_product_category ON Product(CategoryID);
CREATE INDEX
apple_store=# CREATE INDEX idx_sku_product ON Sku(ProductID);
CREATE INDEX
apple_store=# CREATE INDEX idx_price_sku ON Price(SkuID);
CREATE INDEX
apple_store=# CREATE INDEX idx_inventory_store_sku ON Inventory(StoreID, SkuID);
CREATE INDEX
apple_store=# CREATE INDEX idx_order_customer ON "Order"(CustomerID);
CREATE INDEX
apple_store=# CREATE INDEX idx_order_store ON "Order"(StoreID);
CREATE INDEX
apple_store=# CREATE INDEX idx_orderitem_order ON OrderItem(OrderID);
CREATE INDEX
apple_store=# CREATE INDEX idx_orderitem_sku ON OrderItem(SkuID);
CREATE INDEX
apple_store=# CREATE INDEX idx_payment_order ON Payment(OrderID);
CREATE INDEX
```

4 ЗАПОЛНЕНИЕ ТЕСТОВЫМИ ДАННЫМИ

4.1 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');

```
apple_store=# INSERT INTO Customer (FullName, Phone, Email, RegistrationDate) VALUES
apple_store-# ('Иванов Иван', '+79991234567', 'ivanov@mail.ru', '2024-01-12'),
apple_store-# ('Петров Петр', '+79997654321', 'petrov@mail.ru', '2024-03-15'),
apple_store-# ('Сидоров Сергей', '+79998887766', 'sidorov@mail.ru', '2024-04-01'),
apple_store-# ('Кузнецов Кирилл', '+79995554433', 'kuznetsov@mail.ru', '2024-04-20'),
apple_store-# ('Алексеева Анна', '+79990001122', 'alekseeva@mail.ru', '2024-05-10'),
apple_store-# ('Васильев Василий', '+79993332211', 'vasilev@mail.ru', '2024-06-05'),
apple_store-# ('Попов Павел', '+79994445566', 'popov@mail.ru', '2024-07-01');
INSERT 0 7
```

4.2 Category

INSERT INTO Category (CategoryName, ParentCategoryID) VALUES

('iPhone', NULL),

('MacBook', NULL),

('iPad', NULL),

('Watch', NULL),

('Accessories', NULL),

('Chargers', 5),

('Cables', 5);

```
apple_store=# INSERT INTO Category (CategoryName, ParentCategoryID) VALUES
apple_store-# ('iPhone', NULL),
apple_store-# ('MacBook', NULL),
apple_store-# ('iPad', NULL),
apple_store-# ('Watch', NULL),
apple_store-# ('Accessories', NULL),
apple_store-# ('Chargers', 5),
apple_store-# ('Cables', 5);
INSERT 0 7
```

4.3 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');

```
apple_store=# INSERT INTO Product (ProductName, CategoryID, Description) VALUES
apple_store-# ('iPhone 15 Pro', 1, 'Apple flagship smartphone'),
apple_store-# ('iPhone 15', 1, 'Standard iPhone'),
apple_store-# ('MacBook Air M2', 2, 'Light laptop'),
apple_store-# ('MacBook Pro M3', 2, 'Professional laptop'),
apple_store-# ('iPad Pro', 3, 'High-end tablet'),
apple_store-# ('Apple Watch Ultra', 4, 'Smart watch'),
apple_store-# ('AirPods Pro', 5, 'Wireless headphones');
INSERT 0 7
```

4.4 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');

```
apple_store=# INSERT INTO Sku (ProductID, Color, Memory, ModelCode) VALUES
apple_store-# (1, 'Black', '256GB', 'IP15P-B-256'),
apple_store-# (1, 'Silver', '512GB', 'IP15P-S-512'),
apple_store-# (2, 'Blue', '128GB', 'IP15-128B'),
apple_store-# (3, 'Gray', '16GB RAM 512GB SSD', 'MBA-M2-16-512'),
apple_store-# (4, 'Silver', '32GB RAM 1TB SSD', 'MBP-M3-32-1T'),
apple_store-# (5, 'Gray', '256GB', 'IPAD-PRO-256'),
apple_store-# (6, 'Orange', 'Standard', 'AW-Ultra-1');
INSERT 0 7
```

4.5 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');

```
apple_store=# INSERT INTO Price (SkuID, Price, StartDate, EndDate) VALUES
apple_store-# (1, 120000, '2024-01-01', '2024-12-31'),
apple_store-# (2, 140000, '2024-01-01', '2024-12-31'),
apple_store-# (3, 90000, '2024-01-01', '2024-12-31'),
apple_store-# (4, 115000, '2024-01-01', '2024-12-31'),
apple_store-# (5, 200000, '2024-01-01', '2024-12-31'),
apple_store-# (6, 100000, '2024-01-01', '2024-12-31'),
apple_store-# (7, 30000, '2024-01-01', '2024-12-31');
INSERT 0 7
```

4.6 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');

```
apple_store=# INSERT INTO Store (StoreName, Type, Address) VALUES
apple_store-# ('Apple Store Невский', 'retail', 'СПб, Невский 100'),
apple_store-# ('Apple Store Ладожский', 'retail', 'СПб, Ладожская пл. 1'),
apple_store-# ('Склад СПб-1', 'warehouse', 'СПб, Индустриальный 15'),
apple_store-# ('Склад Москва-1', 'warehouse', 'Москва, Каширское ш. 22'),
apple_store-# ('Apple Store Москва Центр', 'retail', 'Москва, Арбат 25'),
apple_store-# ('Склад Казань', 'warehouse', 'Казань, Кремль 3'),
apple_store-# ('Apple Store Казань', 'retail', 'Казань, Баумана 5');
INSERT 0 7
```

4.7 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');

```
apple_store=# INSERT INTO Inventory (StoreID, SkuID, QtyOnHand, QtyReserved, UpdatedAt) VALUES
apple_store-# (1, 1, 10, 2, '2024-05-01'),
apple_store-# (1, 2, 5, 1, '2024-05-01'),
apple_store-# (2, 3, 7, 0, '2024-05-02'),
apple_store-# (3, 4, 20, 5, '2024-05-02'),
apple_store-# (4, 5, 15, 3, '2024-05-03'),
apple_store-# (5, 6, 8, 1, '2024-05-03'),
apple_store-# (7, 7, 12, 4, '2024-05-04');
INSERT 0 7
```

4.8 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);

```
apple_store=# INSERT INTO "Order" (CustomerID, OrderDate, Status, StoreID) VALUES
apple_store-# (1, '2024-05-10', 'completed', 1),
apple_store-# (2, '2024-05-11', 'processing', 2),
apple_store-# (3, '2024-05-12', 'cancelled', 1),
apple_store-# (4, '2024-05-13', 'completed', 5),
apple_store-# (5, '2024-05-14', 'completed', 5),
apple_store-# (6, '2024-05-15', 'processing', 7),
apple_store-# (7, '2024-05-16', 'completed', 1);
INSERT 0 7
```

4.9 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);

```
apple_store=# INSERT INTO OrderItem (OrderID, SkuID, Qty, UnitPriceLocked) VALUES
apple_store-# (1, 1, 1, 120000),
apple_store-# (1, 3, 1, 90000),
apple_store-# (2, 2, 1, 140000),
apple_store-# (3, 7, 2, 30000),
apple_store-# (4, 5, 1, 200000),
apple_store-# (5, 6, 1, 100000),
apple_store-# (6, 4, 1, 115000);
INSERT 0 7
```

4.10 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');

```
apple_store=# INSERT INTO Payment (OrderID, Amount, PaymentDate, PaymentStatus, Method) VALUES
apple_store-# (1, 210000, '2024-05-10', 'success', 'Card'),
apple_store-# (2, 140000, '2024-05-11', 'pending', 'ApplePay'),
apple_store-# (3, 60000, '2024-05-12', 'refunded', 'Card'),
apple_store-# (4, 200000, '2024-05-13', 'success', 'Card'),
apple_store-# (5, 100000, '2024-05-14', 'success', 'Cash'),
apple_store-# (6, 115000, '2024-05-15', 'pending', 'Card'),
apple_store-# (7, 120000, '2024-05-16', 'success', 'ApplePay');
INSERT 0 7
```

5 ПРИМЕР МОДИФИКАЦИИ СТРУКТУРЫ

ALTER TABLE Customer ADD COLUMN BonusPoints INT DEFAULT 0;

```
apple_store=# ALTER TABLE Customer ADD COLUMN BonusPoints INT DEFAULT 0;
ALTER TABLE
```

6 ПРЕДСТАВЛЕНИЯ

6.1 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;
```

```
apple_store=# CREATE VIEW MyOrders AS
apple_store-# SELECT o.OrderID, o.OrderDate, o.Status,
apple_store-#        SUM(oi.Qty * oi.UnitPriceLocked) AS TotalSum,
apple_store-#        COUNT(oi.OrderItemID) AS ItemsCount,
apple_store-#        MAX(p.PaymentStatus) AS LastPaymentStatus
apple_store-# FROM "Order" o
apple_store-# JOIN OrderItem oi ON o.OrderID = oi.OrderID
apple_store-# LEFT JOIN Payment p ON o.OrderID = p.OrderID
apple_store-# GROUP BY o.OrderID, o.OrderDate, o.Status;
CREATE VIEW
```

6.2 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;
```

```
apple_store=# CREATE VIEW StockAndPrice AS
apple_store-# SELECT pr.ProductName, s.Color, s.Memory, s.ModelCode,
apple_store-#        p.Price, (i.QtyOnHand - i.QtyReserved) AS Available, st.StoreName
apple_store-# FROM Product pr
apple_store-# JOIN Sku s ON pr.ProductID = s.ProductID
apple_store-# JOIN Price p ON s.SkuID = p.SkuID
apple_store-# JOIN Inventory i ON s.SkuID = i.SkuID
apple_store-# JOIN Store st ON i.StoreID = st.StoreID
apple_store-# WHERE CURRENT_DATE BETWEEN p.StartDate AND p.EndDate;
CREATE VIEW
```

6.3 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;
```

```
apple_store=# CREATE VIEW Replenishment AS
apple_store-# SELECT st.StoreName, s.ModelCode,
apple_store-#        (i.QtyOnHand - i.QtyReserved) AS Available,
apple_store-#        CASE WHEN (i.QtyOnHand - i.QtyReserved) < 5
apple_store-#             THEN 'Нужно пополнить' ELSE 'Достаточно' END AS Status
apple_store-# FROM Inventory i
apple_store-# JOIN Store st ON i.StoreID = st.StoreID
apple_store-# JOIN Sku s ON i.SkuID = s.SkuID;
CREATE VIEW
```

6.4 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;
```

```
apple_store=# CREATE VIEW SalesBySku AS
apple_store-# SELECT s.ModelCode, s.Color, s.Memory,
apple_store-#        SUM(oi.Qty) AS TotalQty,
apple_store-#        SUM(oi.Qty * oi.UnitPriceLocked) AS Revenue,
apple_store-#        AVG(oi.UnitPriceLocked) AS AvgPrice
apple_store-# FROM OrderItem oi
apple_store-# JOIN Sku s ON oi.SkuID = s.SkuID
apple_store-# JOIN "Order" o ON oi.OrderID = o.OrderID
apple_store-# WHERE o.OrderDate BETWEEN '2024-01-01' AND '2024-12-31'
apple_store-# GROUP BY s.ModelCode, s.Color, s.Memory;
CREATE VIEW
```

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/.