

# PostgreSQL Superstore Analysis Report

## 1. Import Data into PostgreSQL

```
-- Create database
CREATE DATABASE superstore;

-- Connect
\c superstore;

-- Create table
CREATE TABLE superstore_data (
    order_id          VARCHAR(20),
    order_date        DATE,
    ship_date         DATE,
    ship_mode         VARCHAR(50),
    customer_id       VARCHAR(20),
    customer_name     VARCHAR(100),
    segment          VARCHAR(50),
    country           VARCHAR(50),
    city             VARCHAR(100),
    state            VARCHAR(100),
    postal_code       INT,
    region           VARCHAR(50),
    product_id       VARCHAR(20),
    category         VARCHAR(50),
    sub_category     VARCHAR(50),
    product_name     VARCHAR(255),
    sales            NUMERIC(10,2),
    quantity         INT,
    discount         NUMERIC(5,2),
    profit           NUMERIC(10,2)
);

-- Import CSV
COPY superstore_data
FROM 'C:/path/to/Superstore_Cleaned.csv'
DELIMITER ','
CSV HEADER;
```

## 2. Basic Queries (SELECT, WHERE, ORDER BY, GROUP BY)

```
-- Select specific columns
SELECT order_id, order_date, customer_name, sales
FROM superstore_data
LIMIT 10;

-- WHERE condition: Get orders from California
SELECT order_id, state, sales
FROM superstore_data
WHERE state = 'California';

-- ORDER BY: Top 5 highest sales orders
```

```

SELECT order_id, customer_name, sales
FROM superstore_data
ORDER BY sales DESC
LIMIT 5;

-- GROUP BY: Total sales by region
SELECT region, SUM(sales) AS total_sales
FROM superstore_data
GROUP BY region
ORDER BY total_sales DESC;

```

### 3. JOINS (INNER, LEFT, RIGHT)

```

-- INNER JOIN: Only matching regions
SELECT s.order_id, s.region, r.region_name, s.sales
FROM superstore_data s
INNER JOIN region_lookup r
ON s.region = r.region_code
LIMIT 10;

-- LEFT JOIN: All orders, even if no match in region_lookup
SELECT s.order_id, s.region, r.region_name
FROM superstore_data s
LEFT JOIN region_lookup r
ON s.region = r.region_code;

-- RIGHT JOIN: All regions, even if no orders
SELECT s.order_id, s.region, r.region_name
FROM superstore_data s
RIGHT JOIN region_lookup r
ON s.region = r.region_code;

```

### 4. Subqueries

```

-- Find customers with sales above average
SELECT customer_name, SUM(sales) AS total_sales
FROM superstore_data
GROUP BY customer_name
HAVING SUM(sales) > (
    SELECT AVG(sales) FROM superstore_data
);

-- Find top 5 profitable products
SELECT product_name, profit
FROM superstore_data
WHERE profit IN (
    SELECT profit
    FROM superstore_data
    ORDER BY profit DESC
    LIMIT 5
);

```

### 5. Aggregate Functions

```

-- Total sales
SELECT SUM(sales) AS total_sales FROM superstore_data;

-- Average discount
SELECT AVG(discount) AS avg_discount FROM superstore_data;

-- Count of unique customers
SELECT COUNT(DISTINCT customer_id) AS total_customers FROM superstore_data;

-- Maximum & Minimum profit
SELECT MAX(profit) AS max_profit, MIN(profit) AS min_profit
FROM superstore_data;

-- Sales & profit per category
SELECT category, SUM(sales) AS total_sales, SUM(profit) AS total_profit
FROM superstore_data
GROUP BY category
ORDER BY total_sales DESC;

```

## 6. Create Views for Analysis

```

-- View: Sales summary by region
CREATE VIEW region_sales AS
SELECT region, SUM(sales) AS total_sales, SUM(profit) AS total_profit
FROM superstore_data
GROUP BY region;

-- View: Top customers
CREATE VIEW top_customers AS
SELECT customer_name, SUM(sales) AS total_sales
FROM superstore_data
GROUP BY customer_name
ORDER BY total_sales DESC
LIMIT 10;

-- View: Profitability by category
CREATE VIEW category_profit AS
SELECT category, SUM(profit) AS total_profit, AVG(discount) AS avg_discount
FROM superstore_data
GROUP BY category;

```

## 7. Query Optimization with Indexes

```

-- Index on customer_id for faster customer lookups
CREATE INDEX idx_customer_id ON superstore_data(customer_id);

-- Index on order_date for faster time-based queries
CREATE INDEX idx_order_date ON superstore_data(order_date);

-- Index on region for faster grouping
CREATE INDEX idx_region ON superstore_data(region);

```