```sql
-- HOW TO CREATE A DATABASE?
-- STEP1:

-- Create Database
CREATE DATABASE SuperstoreDummy;
USE SuperstoreDummy;

-- Customers Table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100),
    Segment VARCHAR(50),
    Region VARCHAR(50)
);

-- Products Table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Category VARCHAR(50),
    SubCategory VARCHAR(50),
    UnitPrice DECIMAL(10,2)
);

-- Orders Table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    CustomerID INT,
    ProductID INT,
    Quantity INT,
    Discount DECIMAL(4,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

--STEP2:

-- Insert Data
INSERT INTO Customers (CustomerID, CustomerName, Segment, Region) VALUES
(1, 'Alice Smith', 'Consumer', 'East'),
(2, 'Bob Johnson', 'Corporate', 'West'),
(3, 'Charlie Brown', 'Home Office', 'Central'),
(4, 'Diana Ross', 'Consumer', 'South'),
(5, 'Ethan Hunt', 'Corporate', 'East');

INSERT INTO Products (ProductID, ProductName, Category, SubCategory, UnitPrice)
  VALUES
(101, 'Stapler', 'Office Supplies', 'Binders', 5.99),
```

```
(102, 'Printer', 'Technology', 'Machines', 199.99),
(103, 'Desk Chair', 'Furniture', 'Chairs', 89.99),
(104, 'Monitor', 'Technology', 'Displays', 149.99),
(105, 'Notebook', 'Office Supplies', 'Paper', 2.99);

INSERT INTO Orders (OrderID, OrderDate, CustomerID, ProductID, Quantity, Discount)
   VALUES
(1001, '2024-01-10', 1, 101, 10, 0.1),
(1002, '2024-01-15', 2, 102, 1, 0),
(1003, '2024-02-01', 3, 103, 2, 0.2),
(1004, '2024-02-20', 4, 104, 1, 0.15),
(1005, '2024-03-05', 5, 105, 25, 0),
(1006, '2024-03-12', 1, 104, 1, 0),
(1007, '2024-03-25', 2, 105, 50, 0.05);

-- STEP3:
-- EXPLORATORY DATA ANALYSIS

-- Calculate Sales and Profit
SELECT
    o.OrderID,
    c.CustomerName,
    p.ProductName,
    o.Quantity,
    o.Discount,
    ROUND(o.Quantity * p.UnitPrice * (1 - o.Discount), 2) AS Sales,
    ROUND(o.Quantity * p.UnitPrice * (1 - o.Discount) * 0.2, 2) AS Profit,
    o.OrderDate
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID;

-- Check Table
SELECT*
FROM Orders;

-- Sales by Product Category
SELECT
    p.Category,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS TotalSales
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY p.Category
ORDER BY TotalSales DESC;

-- Best-selling Products by Revenue
SELECT TOP 2
    p.ProductName,
```

```sql
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS Revenue
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY Revenue DESC;

-- Profits by Product Category
SELECT
    p.Category,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount) * 0.2) AS Profit
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY  p.Category
ORDER BY Profit DESC;

-- Total Sales over time
SELECT
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS TotalSales
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID;

--  Total Quantity sold Over time
SELECT
    SUM(o.Quantity ) AS TotalQuantity
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID

-- which months or quarters had the highest  and lowest sales?
SELECT TOP 1
    FORMAT(o.OrderDate, 'yyyy-MM') AS Month,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS Sales
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY FORMAT(o.OrderDate, 'yyyy-MM')
ORDER BY Sales DESC; -- Highest Sales Month

SELECT TOP 1
    FORMAT(o.OrderDate, 'yyyy-MM') AS Month,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS Sales
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY FORMAT(o.OrderDate, 'yyyy-MM')
ORDER BY Sales ASC; -- Lowest Sales Month
```

```sql
-- Average Order Value by Month
SELECT
    MONTH(o.OrderDate) AS Month,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS MonthlyRevenue,
    COUNT(DISTINCT(OrderID)) AS TotalOrders,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount))/COUNT(DISTINCT(OrderID)) AS
      AOV
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY MONTH(o.OrderDate)
ORDER BY MONTH(o.OrderDate) DESC;

-- Top Customers by Revenue
SELECT TOP 2
    c.CustomerName,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS Revenue
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY c.CustomerName
ORDER BY Revenue DESC;

-- Average purchase frequency by Customer
SELECT
    COUNT(DISTINCT(o.OrderID))/ COUNT(DISTINCT(c.CustomerID)) AS
      AveragePurchaseFrequency
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID;

-- which customer segment spends most
SELECT TOP 1
    c.Segment,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS TotalSpend
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY c.Segment
ORDER BY TotalSpend DESC;

-- Which Regions generates the most Sales
SELECT Top 2
    c.Region,
    SUM(o.Quantity * p.UnitPrice * (1 - o.Discount)) AS Sales
    FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
```

```
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY c.Region
ORDER BY Sales DESC;
```