"Profitablity Analysis"

-- Overall Profitability Summary

```
SELECT
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(AVG(Profit), 2) AS Avg_Profit_Per_Order
FROM
    orders;
```

-- Profit by Category and Sub-Category

```
SELECT
    Category,
    Sub_Category,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(AVG(Profit), 2) AS Avg_Profit
FROM
    orders
GROUP BY
    Category, Sub_Category
ORDER BY
    Total_Profit ASC;
```

-- Profit by Region and State

```
SELECT
```

```
    Region,
    State,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Region, State
ORDER BY
    Total_Profit ASC;
```

-- Products with High Sales but Negative Profit

```
SELECT
    Product_Name,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Product_Name
HAVING
    SUM(Profit) < 0 AND SUM(Sales) > 500
ORDER BY
    Total_Profit ASC;
```

-- Average Discount vs Profit by Sub-Category

```
SELECT
```

```
    Sub_Category,
    ROUND(AVG(Discount), 2) AS Avg_Discount,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Sub_Category
ORDER BY
    Avg_Discount DESC;
```

-- Profit by Segment

```
SELECT
    Segment,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(AVG(Profit), 2) AS Avg_Profit_Per_Order
FROM
    orders
GROUP BY
    Segment
ORDER BY
    Total_Profit DESC;
```

-- Top 5 Most Profitable Products

```
SELECT
    Product_Name,
```

```
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Product_Name
ORDER BY
    Total_Profit DESC
LIMIT 5;
```

-- Bottom 5 Least Profitable Products

```
SELECT
    Product_Name,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Product_Name
ORDER BY
    Total_Profit ASC
LIMIT 5;
```

-- Total number of unique customers

```
SELECT
    COUNT(DISTINCT Customer_ID) AS Total_Customers
FROM
```

```
    orders;

-- Profit per customer

SELECT
    Customer_ID,
    Customer_Name,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    COUNT(DISTINCT Order_ID) AS Total_Orders
FROM
    orders
GROUP BY
    Customer_ID, Customer_Name
ORDER BY
    Total_Profit DESC;

-- Top 10 profitable customers

SELECT
    Customer_ID,
    Customer_Name,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Customer_ID, Customer_Name
ORDER BY
```

```
    Total_Profit DESC
LIMIT 10;
```

-- Customers who buy a lot but are not profitable

```
SELECT
    Customer_ID,
    Customer_Name,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Customer_ID, Customer_Name
HAVING
    SUM(Profit) < 0 AND SUM(Sales) > 1000
ORDER BY
    Total_Profit ASC;
```

-- Average sales and profit per customer segment

```
SELECT
    Segment,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    COUNT(DISTINCT Customer_ID) AS Total_Customers,
    ROUND(SUM(Sales)/COUNT(DISTINCT Customer_ID), 2) AS Avg_Sales_Per_Customer,
    ROUND(SUM(Profit)/COUNT(DISTINCT Customer_ID), 2) AS Avg_Profit_Per_Customer
```

```
FROM
    orders
GROUP BY
    Segment;
```

"Customer Analysis"

-- Number of unique customers by state

```
SELECT
   State,
   COUNT(DISTINCT Customer_ID) AS Customer_Count
FROM
   orders
GROUP BY
   State
ORDER BY
   Customer_Count DESC;
```

-- Top 10 customers by number of orders

```
SELECT
   Customer_ID,
   Customer_Name,
   COUNT(DISTINCT Order_ID) AS Orders_Count
FROM
   orders
GROUP BY
   Customer_ID, Customer_Name
ORDER BY
   Orders_Count DESC
LIMIT 10;
```

"Discount-Impact Analysis"

-- Grouping data by discount level (rounded to nearest 0.1)

```
SELECT
   ROUND(Discount, 1) AS Discount_Level,
   COUNT(*) AS Order_Count,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit,
   ROUND(AVG(Profit), 2) AS Avg_Profit_Per_Order
FROM
   orders
GROUP BY
   ROUND(Discount, 1)
ORDER BY
   Discount_Level;
```

-- Compare orders with and without discounts

```
SELECT
   CASE
      WHEN Discount = 0 THEN 'No Discount'
      ELSE 'Discount Applied'
   END AS Discount_Status,
   COUNT(*) AS Order_Count,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit,
```

```
    ROUND(AVG(Profit), 2) AS Avg_Profit_Per_Order
FROM
    orders
GROUP BY
    Discount_Status;
```

-- Check if some categories are more sensitive to discounting

```
SELECT
    Category,
    ROUND(AVG(Discount), 2) AS Avg_Discount,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(SUM(Profit)/SUM(Sales)*100, 2) AS Profit_Margin_Percent
FROM
    orders
GROUP BY
    Category
ORDER BY
    Profit_Margin_Percent ASC;
```

-- Orders with high discounts and negative profit

```
SELECT
    Order_ID,
    Customer_Name,
    Product_Name,
    Discount,
```

```
    Sales,
    Profit
FROM
    orders
WHERE
    Discount >= 0.3 AND Profit < 0
ORDER BY
    Discount DESC;
```

-- Which sub-categories lose the most due to discounts?

```
SELECT
    Sub_Category,
    ROUND(AVG(Discount), 2) AS Avg_Discount,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(SUM(Profit)/SUM(Sales)*100, 2) AS Profit_Margin_Percent
FROM
    orders
GROUP BY
    Sub_Category
ORDER BY
    Profit_Margin_Percent ASC;
```

"Sales Analysis"

-- Monthly Sales Trend

```
SELECT
   DATE_FORMAT(STR_TO_DATE(Order_Date, '%m/%d/%Y'), '%Y-%m') AS Month,
   ROUND(SUM(Sales), 2) AS Total_Sales
FROM
   orders
GROUP BY
   Month
ORDER BY
   Month;
```

-- Top 10 Products by Sales

```
SELECT
   Product_Name,
   ROUND(SUM(Sales), 2) AS Total_Sales
FROM
   orders
GROUP BY
   Product_Name
ORDER BY
   Total_Sales DESC
LIMIT 10;
```

-- Sales and Profit by Category and Sub-Category

```
SELECT
   Category,
   Sub_Category,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
   orders
GROUP BY
   Category, Sub_Category
ORDER BY
   Total_Profit DESC;
```

-- Sales and Profit by Region and State

```
SELECT
   Region,
   State,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
   orders
GROUP BY
   Region, State
ORDER BY
   Total_Profit DESC;
```

-- Sales and Profit by Segment

```
SELECT
   Segment,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
   orders
GROUP BY
   Segment
ORDER BY
   Total_Sales DESC;
```

-- Sales and Profit by Ship Mode

```
SELECT
   Ship_Mode,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit,
   COUNT(*) AS Orders_Count
FROM
   orders
GROUP BY
   Ship_Mode
ORDER BY
   Total_Sales DESC;
```

-- Average Discount vs Profit by Sub-Category

```
SELECT
   Sub_Category,
   ROUND(AVG(Discount), 2) AS Avg_Discount,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
   orders
GROUP BY
   Sub_Category
ORDER BY
   Avg_Discount DESC;
```

-- Top 10 Profitable Products

```
SELECT
   Product_Name,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
   orders
GROUP BY
   Product_Name
ORDER BY
   Total_Profit DESC
LIMIT 10;
```

-- Bottom 10 Products by Profit

```
SELECT
```

```
    Product_Name,
    ROUND(SUM(Profit), 2) AS Total_Profit
FROM
    orders
GROUP BY
    Product_Name
ORDER BY
    Total_Profit ASC
LIMIT 10;
```

"Shipping Analysis"

-- Analyze total sales, profit, and average delivery time by shipping mode

```
SELECT
   Ship_Mode,
   COUNT(*) AS Order_Count,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit,
   ROUND(AVG(DATEDIFF(Ship_Date, Order_Date)), 2) AS Avg_Delivery_Days
FROM
   orders
GROUP BY
   Ship_Mode
ORDER BY
   Total_Profit DESC;
```

-- Find out which regions prefer which shipping modes and how profitable each is

```
SELECT
   Region,
   Ship_Mode,
   COUNT(*) AS Order_Count,
   ROUND(SUM(Sales), 2) AS Total_Sales,
   ROUND(SUM(Profit), 2) AS Total_Profit
FROM
```

```
    orders
GROUP BY
    Region, Ship_Mode
ORDER BY
    Region, Total_Profit DESC;
```

-- Check orders that took more than 5 days to deliver

```
SELECT
    Order_ID,
    Customer_Name,
    Ship_Mode,
    Order_Date,
    Ship_Date,
    DATEDIFF(Ship_Date, Order_Date) AS Delivery_Days,
    Sales,
    Profit
FROM
    orders
WHERE
    DATEDIFF(Ship_Date, Order_Date) > 5
ORDER BY
    Delivery_Days DESC;
```

-- Create delivery speed buckets and analyze their profitability

```
SELECT
    CASE
```

```
        WHEN DATEDIFF(Ship_Date, Order_Date) <= 2 THEN 'Fast (≤2 Days)'
        WHEN DATEDIFF(Ship_Date, Order_Date) BETWEEN 3 AND 5 THEN 'Moderate (3–5 Days)'
        ELSE 'Slow (>5 Days)'
    END AS Delivery_Speed,
    COUNT(*) AS Order_Count,
    ROUND(SUM(Sales), 2) AS Total_Sales,
    ROUND(SUM(Profit), 2) AS Total_Profit,
    ROUND(AVG(Profit), 2) AS Avg_Profit_Per_Order
FROM
    orders
GROUP BY
    Delivery_Speed
ORDER BY
    Delivery_Speed;
```

-- Orders with high discounts and longer delivery — potential churn zone

```
SELECT
    Order_ID,
    Customer_Name,
    Ship_Mode,
    Discount,
    DATEDIFF(Ship_Date, Order_Date) AS Delivery_Days,
    Sales,
    Profit
FROM
    orders
```

```
WHERE
    Discount >= 0.3 AND DATEDIFF(Ship_Date, Order_Date) > 5 AND Profit < 0
ORDER BY
    Delivery_Days DESC;
```

"Superstore_EDA"

-- Creating database

```
CREATE DATABASE superstore_db;

USE superstore_db;


CREATE TABLE orders (
    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(50),
    State VARCHAR(50),
    Postal_Code VARCHAR(20),
    Region VARCHAR(50),
    Product_ID VARCHAR(20),
    Category VARCHAR(50),
    Sub_Category VARCHAR(50),
    Product_Name VARCHAR(150),
    Sales DECIMAL(10,2),
    Quantity INT,
    Discount DECIMAL(4,2),
    Profit DECIMAL(10,2)
);
```

```
-- Step 1: Preview the data
SELECT * FROM orders;

-- Step 2: Count total number of rows
SELECT COUNT(*) AS total_orders FROM orders;

-- Step 3: Check for duplicate Order IDs
SELECT Order_ID, COUNT(*) AS count
FROM orders
GROUP BY Order_ID
HAVING COUNT(*) > 1;

-- to inspect columns
SELECT * FROM orders LIMIT 1;

ALTER TABLE orders
rename column `row id` to row_id,
  RENAME COLUMN `Order ID` TO Order_ID,
  RENAME COLUMN `Order Date` TO Order_Date,
  RENAME COLUMN `Ship Date` TO Ship_Date,
  RENAME COLUMN `Ship Mode` TO Ship_Mode,
  RENAME COLUMN `Customer ID` TO Customer_ID,
  RENAME COLUMN `Customer Name` TO Customer_Name,
  RENAME COLUMN `Postal Code` TO Postal_Code,
  RENAME COLUMN `Product ID` TO Product_ID,
  RENAME COLUMN `Sub-Category` TO Sub_Category,
```

```
 RENAME COLUMN `Product Name` TO Product_Name;


SELECT *
FROM orders
WHERE Order_ID IN (
 SELECT Order_ID
 FROM orders
 GROUP BY Order_ID
 HAVING COUNT(*) > 1
);
```

-- to check real duplicate data

```
select* from orders
group by row_id, order_id, order_date, ship_date, ship_mode, customer_id,
customer_name, segment,
country, city, state, postal_code, region, product_id, category, sub_category,
product_name, sales, quantity, discount, profit
HAVING COUNT(*) > 1;
```

-- How many orders are in the dataset

```
SELECT COUNT(DISTINCT Order_ID) AS total_orders
FROM orders;
```

-- What is the total sales and total profit?

```
SELECT
   ROUND(SUM(Sales), 2) AS total_sales,
   ROUND(SUM(Profit), 2) AS total_profit
FROM orders;
```

-- Top 5 most profitable products

```
SELECT
   Product_Name,
   ROUND(SUM(Profit), 2) AS total_profit
FROM orders
GROUP BY Product_Name
ORDER BY total_profit DESC
LIMIT 5;
```

-- Which regions are losing profit?

```
SELECT
   Region,
   ROUND(SUM(Profit), 2) AS total_profit
FROM orders
GROUP BY Region
ORDER BY total_profit ASC;
```