**WEEK 3: Exploratory Data Analysis (EDA) with SQL**

**Description:**  
This document contains intermediate-to-advanced SQL queries executed on the Superstore dataset.  
It includes time-based analysis, segment insights, product performance, and advanced CASE-based classification.

**Tools Used:** MySQL  
**Dataset:** Cleaned Superstore Data

**Task 1: Time-Based Sales Analysis:**

1. Total sales and profit per year:

**MySQL query** - Select

YEAR(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS order\_year,

sum(sales) as total\_sales,

sum(profit) as total\_profit

From sample\_superstore

group by order\_year;

1. Sales trend per month:

**MySQL query** - SELECT

YEAR(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS Date\_as\_Year,

MONTH(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS MonthNum,

SUM(sales) AS total\_sales

from sample\_superstore

group by Date\_as\_Year, MonthNum

order by Date\_as\_Year, MonthNum asc;

1. Which month has the highest average profit?

**MySQL query** - SELECT

YEAR(STR\_TO\_DATE(order\_date, '%m/%d/%y')) AS Years,

MONTHNAME(STR\_TO\_DATE(order\_date, '%m/%d/%y')) AS months,

AVG(profit) AS avg\_profit

FROM sample\_superstore

GROUP BY Years , months

ORDER BY avg\_profit DESC

LIMIT 1;

1. Compare order volume by weekday (Mon–Sun)

**MySQL query** - SELECT

DAYNAME(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS Daynames,

DAYOFWEEK(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS DayNum,

COUNT(\*) AS total\_order

FROM

sample\_superstore

GROUP BY Daynames , DayNum

ORDER BY DayNum ASC;

**Task 2: Regional & Customer Segment Insights**

1. Total sales by region and segment

**MySQL Query** - SELECT

Region, Segment, SUM(sales) AS total\_sales

FROM

sample\_superstore

GROUP BY Region, Segment

ORDER BY total\_sales DESC

1. Region with the highest average discount

**MySQL Query** - SELECT

Region, AVG(Discount) AS Avg\_Discount

FROM

sample\_superstore

GROUP BY Region

ORDER BY Avg\_Discount;

3. Segment with the highest return on sales (Profit ÷ Sales)

**MySQL Query** - Select Segment, SUM(Profit) / SUM(Sales) as Return\_on\_Sales

From sample\_superstore

Group by Segment

**Task 3: Product Performance Analysis**

1. Top 10 sub-categories by sales

**MySQL Query** - SELECT

sub\_category, SUM (sales) AS total\_sales

FROM

sample\_superstore

GROUP BY sub\_category

ORDER BY total\_sales DESC

LIMIT 10;

1. Which category has the highest profit margin? (Profit/Sales)?

**MySQL Query** - SELECT

Category, SUM (Profit)/SUM(Sales) AS Profit\_margin

FROM

sample\_superstore

GROUP BY Category

ORDER BY Profit\_margin

1. Count of Unique Products per Category

**MySQL Query** - SELECT

Category, COUNT(DISTINCT product\_name) AS total\_count

FROM

sample\_superstore

GROUP BY Category

ORDER BY total\_count

1. Products with Negative Profit in 10+ Orders

**MySQL Query** - SELECT product\_name, SUM(profit) AS total\_profit

FROM sample\_superstore

WHERE profit < 0

GROUP BY product\_name

HAVING COUNT(\*) >= 10

**Task 4: Advanced SQL Practice**

1. Use CASE WHEN to Classify Order Value

**MySQL Query** - SELECT order\_id, sales,

CASE

WHEN sales > 500 THEN 'High'

WHEN sales BETWEEN 100 AND 500 THEN 'Medium'

ELSE 'Low'

END AS order\_value

FROM sample\_superstore;

1. Monthly Summary for the Latest Year

**MySQL Query** - SELECT

MONTHNAME(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) AS months,

SUM(sales) AS total\_sales,

SUM(profit) AS total\_profit,

AVG(Discount) AS avg\_discount,

COUNT(order\_id) AS total\_order

FROM

sample\_superstore

WHERE

YEAR(STR\_TO\_DATE(order\_date, '%m/%d/%Y')) = 2017

GROUP BY months

ORDER BY FIELD(months, 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December');

1. Top Performing States by Profit Margin

**MySQL Query** - Select State,sum(profit)/sum(sales) as profit\_margin

From sample\_superstore

group by state

order by profit\_margin desc

limit 10;

1. Flag Orders with High Discount + Low Profit

**MySQL Query** - Select order\_id, Discount, Profit,

CASE

WHEN Discount > 0.3 AND Profit < 0 THEN 'Risk'

ELSE 'Safe'

END AS Risk\_Label

From sample\_superstore