⍰ 1. Aggregations and Summaries

```
-- 1. Total sales
SELECT SUM(Sales) AS Total_Sales FROM Simple_superstor;

-- 2. Average profit per order
SELECT AVG(Profit) AS Avg_Profit FROM Simple_superstor;

-- 3. Total quantity sold by category
SELECT Category, SUM(Quantity) AS Total_Quantity
FROM Simple_superstor
GROUP BY Category;

-- 4. Average discount by sub-category
SELECT `Sub-Category`, AVG(Discount) AS Avg_Discount
FROM Simple_superstor
GROUP BY `Sub-Category`;

-- 5. Maximum and minimum profit
SELECT MAX(Profit) AS Max_Profit, MIN(Profit) AS Min_Profit FROM
Simple_superstor;
```

□ 2. Filtering with WHERE and HAVING

```
-- 6. Orders with profit > 1000
SELECT * FROM Simple_superstor
WHERE Profit > 1000;

-- 7. High-discount orders
SELECT * FROM Simple_superstor
WHERE Discount > 0.5;

-- 8. Orders from California
SELECT * FROM Simple_superstor
WHERE State = 'California';

-- 9. Customers who ordered more than 10 items
SELECT * FROM Simple_superstor
WHERE Quantity > 10;

-- 10. Grouped and filtered using HAVING
SELECT Category, SUM(Profit) AS Total_Profit
FROM Simple_superstor
GROUP BY Category
HAVING Total_Profit > 5000;
```

□ 3. Date Functions

```
-- 11. Orders placed in 2023
SELECT * FROM Simple_superstor
WHERE YEAR(`Order Date`) = 2023;

-- 12. Orders shipped within 3 days of order date
SELECT *, DATEDIFF(`Ship Date`, `Order Date`) AS Ship_Delay
FROM Simple_superstor
WHERE DATEDIFF(`Ship Date`, `Order Date`) <= 3;
```

```
-- 13. Monthly sales
SELECT MONTH(`Order Date`) AS Month, SUM(Sales) AS Monthly_Sales
FROM Simple_superstor
GROUP BY Month;

-- 14. Yearly profit
SELECT YEAR(`Order Date`) AS Year, SUM(Profit) AS Yearly_Profit
FROM Simple_superstor
GROUP BY Year;

-- 15. Average delivery days
SELECT AVG(DATEDIFF(`Ship Date`, `Order Date`)) AS Avg_Delivery_Days
FROM Simple_superstor;
```

□ 4. Location Analysis

```
-- 16. Top 5 states by sales
SELECT State, SUM(Sales) AS Total_Sales
FROM Simple_superstor
GROUP BY State
ORDER BY Total_Sales DESC
LIMIT 5;

-- 17. Cities with negative profit
SELECT DISTINCT City
FROM Simple_superstor
WHERE Profit < 0;

-- 18. Total orders by region
SELECT Region, COUNT(*) AS Total_Orders
FROM Simple_superstor
GROUP BY Region;

-- 19. Average sales by city
SELECT City, AVG(Sales) AS Avg_Sales
FROM Simple_superstor
GROUP BY City;

-- 20. Top country in terms of profit
SELECT Country, SUM(Profit) AS Total_Profit
FROM Simple_superstor
GROUP BY Country
ORDER BY Total_Profit DESC
LIMIT 1;
```

□□ 5. Customer Analysis

```
-- 21. Total customers
SELECT COUNT(DISTINCT `Customer ID`) AS Total_Customers FROM
Simple_superstor;

-- 22. Top 5 customers by sales
SELECT `Customer Name`, SUM(Sales) AS Total_Sales
FROM Simple_superstor
GROUP BY `Customer Name`
ORDER BY Total_Sales DESC
```

```
LIMIT 5;

-- 23. Customer with most orders
SELECT `Customer Name`, COUNT(*) AS Order_Count
FROM Simple_superstor
GROUP BY `Customer Name`
ORDER BY Order_Count DESC
LIMIT 1;

-- 24. Customers with negative profit orders
SELECT DISTINCT `Customer Name`
FROM Simple_superstor
WHERE Profit < 0;

-- 25. Segment-wise customer count
SELECT Segment, COUNT(DISTINCT `Customer ID`) AS Customers
FROM Simple_superstor
GROUP BY Segment;
```

□ 6. Product Analysis

```
-- 26. Most sold product
SELECT `Product Name`, SUM(Quantity) AS Total_Sold
FROM Simple_superstor
GROUP BY `Product Name`
ORDER BY Total_Sold DESC
LIMIT 1;

-- 27. Top category by profit
SELECT Category, SUM(Profit) AS Total_Profit
FROM Simple_superstor
GROUP BY Category
ORDER BY Total_Profit DESC
LIMIT 1;

-- 28. Products with average discount > 0.3
SELECT `Product Name`, AVG(Discount) AS Avg_Discount
FROM Simple_superstor
GROUP BY `Product Name`
HAVING Avg_Discount > 0.3;

-- 29. Category-wise total sales and quantity
SELECT Category, SUM(Sales) AS Sales, SUM(Quantity) AS Quantity
FROM Simple_superstor
GROUP BY Category;

-- 30. Products with negative total profit
SELECT `Product Name`, SUM(Profit) AS Total_Profit
FROM Simple_superstor
GROUP BY `Product Name`
HAVING Total_Profit < 0;
```

□ 7. Window Functions (MySQL 8.0+)

```
-- 31. Running total of sales
SELECT `Order Date`, Sales,
       SUM(Sales) OVER (ORDER BY `Order Date`) AS Running_Total
```

```
FROM Simple_superstor;

-- 32. Rank products by sales within each category
SELECT Category, `Product Name`, SUM(Sales) AS Product_Sales,
       RANK() OVER (PARTITION BY Category ORDER BY SUM(Sales) DESC) AS
Sales_Rank
FROM Simple_superstor
GROUP BY Category, `Product Name`;

-- 33. Profit percentage of each row in its category
SELECT Category, `Product Name`, Profit,
       Profit / SUM(Profit) OVER (PARTITION BY Category) * 100 AS
Profit_Percent
FROM Simple_superstor;
```

□ 8. Subqueries and Joins

```
-- 34. Orders with profit greater than average
SELECT * FROM Simple_superstor
WHERE Profit > (SELECT AVG(Profit) FROM Simple_superstor);

-- 35. Products that have been sold in more than 10 different cities
SELECT `Product Name`
FROM Simple_superstor
GROUP BY `Product Name`
HAVING COUNT(DISTINCT City) > 10;
```