-- Add the time\_of\_day column

SELECT

time,

(CASE

WHEN time BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"

WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"

ELSE "Evening"

END) AS time\_of\_day

FROM sales;

ALTER TABLE sales ADD COLUMN time\_of\_day VARCHAR(20);

-- For this to work turn off safe mode for update

-- Edit > Preferences > SQL Edito > scroll down and toggle safe mode

-- Reconnect to MySQL: Query > Reconnect to server

UPDATE sales

SET time\_of\_day = (

CASE

WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"

WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"

ELSE "Evening"

END

);

-- Add day\_name column

SELECT

date,

DAYNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN day\_name VARCHAR(10);

UPDATE sales

SET day\_name = DAYNAME(date);

-- Add month\_name column

SELECT

date,

MONTHNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN month\_name VARCHAR(10);

UPDATE sales

SET month\_name = MONTHNAME(date);

-- --------------------------------------------------------------------

-- ---------------------------- Generic ------------------------------

-- --------------------------------------------------------------------

-- How many unique cities does the data have?

SELECT

DISTINCT city

FROM sales;

-- In which city is each branch?

SELECT

DISTINCT city,

branch

FROM sales;

## Product Lines

1. How many unique product lines does the data have?

SELECT Count(distinct Product\_line)

FROM sales ;

1. What is the most common payment method?

SELECT Payment,Count(\*)

FROM sales

Group by Payment

Order By count(\*) DESC

LIMIT 1 ;

1. What is the most selling product line?

SELECT Product\_line,Count(\*)

FROM sales

Group by Product\_line

Order By count(\*) DESC

LIMIT 1

1. What is the total revenue by month?

SELECT Month(Date) as Month, sum(Cogs) as Total\_sales

FROM sales

GROUP BY Month

ORDER BY Month;

1. What month had the largest COGS?

SELECT monthname(Date) as Months,MAX(Cogs) as Cogss

FROM sales

Group by Months;

1. What product line had the largest revenue?

SELECT Product\_line, sum(Total) as Total

FROM sales

GROUP BY Product\_line

ORDER BY Total DESC

LIMIT 1 ;

1. What is the city with the largest revenue?

Method 1 :

SELECT City , sum(Total) as Total

FROM sales

GROUP BY City

ORDER BY Total DESC

LIMIT 1

Method 2 :

SELECT City , MAX(Total) as Total

FROM sales

GROUP BY City

Order by City DESC

LIMIT 1;

1. What product line had the largest VAT?

SELECT Product\_line , MAX(Tax\_5\_percent) as VAT

FROM sales

GROUP BY Product\_line

Order by Product\_line DESC

LIMIT 1;

1. Fetch each product line and add a column to those product line showing "Good",

"Bad". Good if its greater than average sales

WITH ProductLineSales AS (

SELECT

Product\_line,

SUM(Total) AS total\_sales

FROM

sales

GROUP BY

Product\_line

),

AverageSales AS (

SELECT

AVG(total\_sales) AS avg\_sales

FROM

ProductLineSales

)

SELECT

pls.Product\_line,

pls.total\_sales,

CASE

WHEN pls.total\_sales > avg\_sales THEN 'Good'

ELSE 'Bad'

END AS performance

FROM

ProductLineSales pls,

AverageSales

ORDER BY

pls.Product\_line;

1. Which branch sold more products than average product sold?

SET @Total\_avg = (SELECT AVG(Quantity) from sales);

With Branch\_avg AS (SELECT Branch, avg(Quantity) as AVG\_branch\_qty

FROM sales

GROUP BY Branch)

Select Branch

From Branch\_avg

where AVG\_branch\_qty> @Total\_avg;

1. What is the most common product line by gender?

WITH Pro\_line as (SELECT Gender,Product\_line,count(Product\_line) as total\_product\_line,

ROW\_NUMBER() OVER (PARTITION BY GENDER ORDER BY Product\_line) as Rnk

FROM sales

Group by Gender,Product\_line)

SELECT Gender,Product\_line

FROM Pro\_line

where Rnk = 1;

-- --------------------------------------------------------------------

-- --------------------------------------------------------------------

-- --------------------------------------------------------------------

-- ---------------------------- Sales ---------------------------------

-- --------------------------------------------------------------------

-- Number of sales made in each time of the day per weekday

SELECT

time\_of\_day,

COUNT(\*) AS total\_sales

FROM sales

WHERE day\_name = "Sunday"

GROUP BY time\_of\_day

ORDER BY total\_sales DESC;

-- Evenings experience most sales, the stores are

-- filled during the evening hours

-- Which of the customer types brings the most revenue?

SELECT

customer\_type,

SUM(total) AS total\_revenue

FROM sales

GROUP BY customer\_type

ORDER BY total\_revenue;

-- Which city has the largest tax/VAT percent?

SELECT

city,

ROUND(AVG(tax\_pct), 2) AS avg\_tax\_pct

FROM sales

GROUP BY city

ORDER BY avg\_tax\_pct DESC;

-- Which customer type pays the most in VAT?

SELECT

customer\_type,

AVG(tax\_pct) AS total\_tax

FROM sales

GROUP BY customer\_type

ORDER BY total\_tax;

-- --------------------------------------------------------------------

-- --------------------------------------------------------------------

- --------------------------------------------------------------------

-- --------------------------------------------------------------------

-- --------------------------------------------------------------------

-- -------------------------- Customers -------------------------------

-- --------------------------------------------------------------------

-- How many unique customer types does the data have?

SELECT

DISTINCT customer\_type

FROM sales;

-- How many unique payment methods does the data have?

SELECT

DISTINCT payment

FROM sales;

-- What is the most common customer type?

SELECT

customer\_type,

count(\*) as count

FROM sales

GROUP BY customer\_type

ORDER BY count DESC;

-- Which customer type buys the most?

SELECT

customer\_type,

COUNT(\*)

FROM sales

GROUP BY customer\_type;

-- What is the gender of most of the customers?

SELECT

gender,

COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;

-- What is the gender distribution per branch?

SELECT

gender,

COUNT(\*) as gender\_cnt

FROM sales

WHERE branch = "C"

GROUP BY gender

ORDER BY gender\_cnt DESC;

-- Gender per branch is more or less the same hence, I don't think has

-- an effect of the sales per branch and other factors.

-- Which time of the day do customers give most ratings?

SELECT

time\_of\_day,

AVG(rating) AS avg\_rating

FROM sales

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;

-- Looks like time of the day does not really affect the rating, its

-- more or less the same rating each time of the day.alter

-- Which time of the day do customers give most ratings per branch?

SELECT

time\_of\_day,

AVG(rating) AS avg\_rating

FROM sales

WHERE branch = "A"

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;

-- Branch A and C are doing well in ratings, branch B needs to do a

-- little more to get better ratings.

-- Which day fo the week has the best avg ratings?

SELECT

day\_name,

AVG(rating) AS avg\_rating

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;

-- Mon, Tue and Friday are the top best days for good ratings

-- why is that the case, how many sales are made on these days?

-- Which day of the week has the best average ratings per branch?

SELECT

day\_name,

COUNT(day\_name) total\_sales

FROM sales

WHERE branch = "C"

GROUP BY day\_name

ORDER BY total\_sales DESC;

**## SOLUTION of 50 questions**

### Easy

1. Retrieve all sales transactions.

SELECT \* FROM walmart\_sales.sales;

2. List all unique branches.

SELECT Distinct Branch

FROM sales;

3. Count the number of sales transactions.

SELECT Count(\*) as No\_of\_sales\_transactions

FROM sales;

4. Find the total sales amount for each branch.

SELECT Branch, sum(Total) as Total\_sales

FROM sales

Group by Branch

Order by Branch

5. Get the average unit price of products sold.

SELECT avg(Unit\_price) as Avg\_Unit\_Price

FROM sales;

6. List all transactions made by female customers.

SELECT \*

FROM sales

WHERE Gender = 'Female';

7. Find the total number of products sold.

SELECT Sum(Quantity) as Total\_Quantity

FROM sales;

8. Retrieve the sales transactions where the quantity is greater than 6.

SELECT \*

FROM sales

Where Quantity > 6;

9. List all unique customer types.

SELECT DISTINCT Customer\_type

FROM sales;

10. Get the total VAT collected.

SELECT SUM(Tax\_5\_percent) as Total\_VAT

FROM sales;

### Intermediate

11. Find the average unit price of products sold in Mandalay.

SELECT city, round(AVG(Unit\_price),2) FROM sales

group by city

having city = 'Mandalay';

12. List the total quantity of products sold for each branch.

SELECT Branch, sum(Quantity) as Total\_Quantity FROM sales

group by branch

order by Total\_Quantity DESC;

13. Retrieve transactions where the gross margin percentage is above 30%.

NA

14 a. Get the highest rating received and the corresponding invoice\_id.

SELECT

Invoice\_ID,

Rating

FROM sales

order by Rating DESC

LIMIT 1;

14 b. Get the 2nd highest rating received and the corresponding invoice\_id.

**Method 1** : Generic query :

SELECT \*

FROM employee

WHERE salary= ( SELECT DISTINCT(salary)

FROM employee ORDER BY salary DESC LIMIT n-1,1) ;

**Actual query :**

SELECT \*

FROM sales

order by Rating DESC

LIMIT 1,1 ;

**Method 2 :**

SELECT \*

FROM

( SELECT Invoice\_ID, Rating,

DENSE\_RANK() OVER (ORDER BY Rating Desc) AS Dns\_Rnk FROM sales) as B

where Dns\_Rnk = 2 ;

15. Find the most common payment method used.

SELECT Payment, count(payment)

FROM sales

GROUP BY payment

order by count(payment) DESC ;

16. Calculate the average total cost per city.

SELECT city, avg(Total) as Avg\_total\_cost

FROM sales

GROUP BY city

order by Avg\_total\_cost DESC ;

17. Find the total gross income for each product line.

SELECT Product\_line,sum(Gross\_income) as Total\_Gross\_Income

FROM sales

GROUP BY Product\_line

order by Total\_Gross\_Income DESC ;

18. List all transactions made on weekends (Saturday and Sunday).

SELECT Invoice\_ID,dayname(Date) AS DA

FROM sales

**HAVING** DA="Saturday" or DA="Sunday"

19. Get the product line with the highest total sales.

SELECT Product\_line,sum(Total) as Total\_Sales

FROM sales

GROUP BY Product\_line

order by Total\_Sales DESC

LIMIT 1 ;

**Without using LIMIT ( Using subquery) :**

select Product\_line, Max(Total\_Sales)

From

(SELECT sum(Total) as Total\_Sales,Product\_line

FROM sales

GROUP BY Product\_line

order by Total\_Sales DESC) as t

group by Product\_line;

20. Find the top 5 highest grossing transactions.

Select

Invoice\_ID,Gross\_income

From sales

Order By Gross\_income DESC

LIMIT 5 ;

21. Determine the average rating for each branch.

Select

Branch, Avg(Rating) as Avg\_rating

From sales

GROUP BY Branch

Order By Avg\_rating DESC ;

22. Calculate the total sales and the number of transactions for each customer type.

SELECT Customer\_type,count(\*) as No\_of\_transc, sum(Total) as Total

FROM sales

Group by Customer\_type;

23. Find the product line with the lowest average unit price.

SELECT DISTINCT Product\_line, avg(Unit\_price)as Avg\_unit\_price FROM walmart\_sales.sales

group by Product\_line

order by Avg\_unit\_price ASC

LIMIT 1 ;

24. Get the percentage contribution of each branch to the total sales.

SET @Total = (SELECT sum(Total) as Total\_Sum

FROM walmart\_sales.sales);

SELECT Branch, (sum(Total)/@Total)\*100 as Total\_Sum

FROM walmart\_sales.sales

group by Branch

order by Total\_Sum ASC;

25. Retrieve the average gross margin percentage for each city.

NA

26. List the total VAT collected per month.

Use walmart\_sales;

SELECT monthname(Date) as Months, sum(Tax\_5\_percent) as VAT

FROM sales

Group by Months;

27. Calculate the total quantity of products sold for each product line per branch.

Use walmart\_sales;

SELECT Branch, Product\_line,Sum(Quantity) as Total\_Quantity

FROM sales

Group by Branch,Product\_line

order by Branch,Product\_line;

28. Find the average rating of products sold for each customer type.

Use walmart\_sales;

SELECT Customer\_type, Round(Avg(Rating),2) as Avg\_Rating

FROM sales

Group By Customer\_type

29. Get the total sales and total quantity of products sold for each gender.

Select Gender, Sum(Quantity) As Total\_Quantity, Sum(Total) as Total\_sales

From sales

Group BY Gender;

30. Determine the number of transactions and total gross income for each payment method.

Select

Payment,

count(\*) as No\_of\_Transactions,

Sum(Gross\_income) As Total\_Gross\_income

From sales

Group BY Payment

Order By Payment

**### Advanced**

, suma

31. Find the total sales made on the 15th of each month.

Select

date(Date) as Monthn, SUM(total) as Total\_Sales

From sales

GROUP BY Monthn

having Monthn like '%15' ;

32. Get the number of transactions made each day of the week.

Select

weekday(Date) as Weekday, count(\*) as No\_of\_transactions

From sales

GROUP BY Weekday

Order by Weekday;

33. Calculate the average total cost of transactions made during weekends.

Select

dayname(Date) as day, AVG(Total) as Total\_Cost\_of\_transactions

From sales

GROUP BY day

HAVING day in ('Saturday', 'Sunday')

Order by day

34. List the total sales made in each quarter of the year.

SELECT

quarter(Date) as Quarter, AVG(Total) as Total\_Cost\_of\_transactions

From sales

GROUP BY Quarter

Order by Quarter;

35. Find the average rating of transactions made in the evening (after 6 PM).

SELECT

quarter(Date) as Quarter, AVG(Total) as Total\_Cost\_of\_transactions

From sales

GROUP BY Quarter

Order by Quarter;

36. Find the invoice with the highest total sales in each branch.

SELECT

Branch, Max(Total) as Total\_sales

From sales

GROUP BY Branch

Order by Branch;

37. Get the branches where the average unit price is above the overall average unit price.

SELECT

DISTINCT Branch,City, Unit\_price

From sales

where Unit\_price > ( SELECT avg(Unit\_price) FROM sales)

ORDER BY Branch;

38. Find the total sales for each city that exceed the overall average total sales.

SELECT City,Sum(Total\_sales) as Total

FROM

(SELECT City,Total as Total\_sales

From sales

having Total\_sales > ( SELECT avg(Total) FROM sales)

order by City) as T

Group by City;

39. Retrieve the product lines that have a higher total gross income than the average gross income of all product lines.

SELECT

Product\_line,

Avg(Gross\_income)

FROM sales

Group by Product\_line

Having Avg(Gross\_income) > ( SELECT Avg(Gross\_income) FROM sales);

40. List all transactions where the unit price is above the average unit price of products sold in the same branch.

WITH BranchAvgPrice AS (

SELECT

Branch,

AVG(Unit\_price) AS avg\_unit\_price

FROM

sales

GROUP BY

Branch

)

SELECT

s.Invoice\_ID,

s.Branch,

s.City,

s.Product\_line,

s.Total,

s.Unit\_price,

b.avg\_unit\_price

FROM

sales s

JOIN

BranchAvgPrice b

ON

s.Branch = b.Branch

WHERE

s.Unit\_price > b.avg\_unit\_price;

41. Find the top 3 product lines by total sales for each city.

WITH CityProductSales AS (

SELECT

City,

Product\_line,

SUM(Total) AS total\_sales

FROM

sales

GROUP BY

City, Product\_line

),

RankedProductSales AS (

SELECT

City,

Product\_line,

total\_sales,

ROW\_NUMBER() OVER (PARTITION BY City ORDER BY total\_sales DESC) AS ranks

FROM

CityProductSales

)

SELECT

City,

Product\_line,

total\_sales

FROM

RankedProductSales

WHERE

ranks <= 3

ORDER BY

City,

ranks;

42. Get the total quantity of products sold per product line for each gender.

SELECT Gender, Product\_line,sum(QUANTITY) as Total\_quantity

From sales

Group by Gender, Product\_line

ORDER BY Gender, Product\_line;

43. Calculate the total VAT collected for each customer type per branch.

SELECT

Branch,

Customer\_type,

SUM(Tax\_5\_percent) AS total\_vat\_collected

FROM

sales

GROUP BY

Branch,

Customer\_type

ORDER BY

Branch,

Customer\_type;

44. List the highest gross income for each product line and the corresponding branch.

Select

Branch,

Product\_line,

max(Gross\_income) as Gross

From sales

Group by Product\_line, Branch

Order BY Branch,Product\_line ;

45. Retrieve the total sales and gross income for each product line in each quarter of the year.

Select quarter(date) as Quarter,

Product\_line, Sum(Total) as Total\_sales,

Sum(Gross\_income) as Total\_Gross\_income

From sales

Group by Quarter,Product\_line

Order BY Quarter,Product\_line ;

46. Calculate the **cumulative** total sales for each branch over time.

SELECT

Branch,

Date,

SUM(Total) OVER (PARTITION BY Branch ORDER BY Date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative\_total\_sales

FROM

sales

ORDER BY

Branch,

Date;

New concept :

#### **Components:**

1. **SUM(Total)**:
   * This is the aggregate function that calculates the sum of the Total column, which represents the sales amount.
2. **OVER Clause**:
   * The OVER clause is what makes this a window function, allowing the SUM to be calculated over a specific range of rows defined by the window specification.
3. **PARTITION BY Branch**:
   * This divides the result set into partitions based on the Branch column. The cumulative sum will be calculated separately for each branch.
4. **ORDER BY Date**:
   * This orders the rows within each partition by the Date column. The cumulative sum will be calculated in the order of dates within each branch.
5. **ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW**:
   * This specifies the window frame.
   * UNBOUNDED PRECEDING: The window starts at the first row of the partition.
   * CURRENT ROW: The window ends at the current row.
   * Together, this means the sum is calculated from the first row of the partition up to the current row, resulting in a cumulative sum.

47. Find the average rating of the last 5 transactions for each product line.

WITH LastFiveTransactions AS (

SELECT

Product\_line,

Rating,

ROW\_NUMBER() OVER (PARTITION BY Product\_line ORDER BY Date DESC, Time DESC) AS row\_num

FROM

sales

)

SELECT

Product\_line,

AVG(Rating) AS average\_rating

FROM

LastFiveTransactions

WHERE

row\_num <= 5

GROUP BY

Product\_line;

48. Determine the average gross margin percentage for each product line.

SELECT

Product\_line,

AVG(Gross\_margin\_percentage) AS avg\_gross\_margin\_percentage

FROM

sales

GROUP BY

Product\_line

ORDER BY

Product\_line;

49. Get the total sales for each product line where the average rating is above 4.

Method 1 : With Tab AS (SELECT Product\_line, avg(rating) as Avg\_Rating

From sales

Group by Product\_line

Having Avg\_Rating>4)

SELECT sales.Product\_line, SUM(total) as Total\_sales

From Sales

JOIN Tab

ON sales.Product\_line = Tab.Product\_line

GROUP BY Product\_line;

Method 2 :

WITH AvgRating AS (

SELECT

Product\_line,

AVG(Rating) AS avg\_rating

FROM

sales

GROUP BY

Product\_line

)

SELECT

s.Product\_line,

SUM(s.Total) AS total\_sales

FROM

sales s

JOIN

AvgRating a

ON

s.Product\_line = a.Product\_line

WHERE

a.avg\_rating > 4

GROUP BY

s.Product\_line;

50. List the product lines that contribute to more than 16% of the total sales.

WITH TotalSales AS (

SELECT

Product\_line,

SUM(Total) AS total\_sales

FROM

sales

GROUP BY

Product\_line

),

OverallSales AS (

SELECT

SUM(total\_sales) AS overall\_sales

FROM

TotalSales

)

SELECT

ts.Product\_line,

ts.total\_sales

FROM

TotalSales ts,

OverallSales os

WHERE

ts.total\_sales > 0.16 \* os.overall\_sales;