Create database amazon\_database;

Use amazon\_database;

CREATE TABLE amazon (

`Invoice\_id` VARCHAR(30) NOT NULL,

`branch` VARCHAR(5) NOT NULL,

`city` VARCHAR(30) NOT NULL,

`customer\_type` VARCHAR(30) NOT NULL,

`gender` VARCHAR(10) NOT NULL,

`product\_line` VARCHAR(100) NOT NULL,

`unit\_price` DECIMAL(10,2) NOT NULL,

`quantity` INT NOT NULL,

`VAT` FLOAT NOT NULL,

`total` DECIMAL(10,2) NOT NULL,

`date` DATE NOT NULL,

`time` TIMESTAMP NOT NULL,

`payment\_method` DECIMAL(10,2) NOT NULL,

`cogs` DECIMAL(10,2) NOT NULL,

`gross\_margin\_percentage` FLOAT NOT NULL,

`gross\_income` DECIMAL(10,2) NOT NULL,

`rating` FLOAT NOT NULL

);

select \* from amazon;

-- Feature Engineering

-- 1 Add a new column named timeofday to give insight of sales in the Morning, Afternoon and Evening.

Alter Table amazon

ADD COLUMN time\_of\_day varchar(50);

UPDATE amazon

SET time\_of\_day = CASE

WHEN HOUR(amazon.time) >= 0 AND HOUR(amazon.time) < 12 THEN 'Morning'

WHEN HOUR(amazon.time) >= 12 AND HOUR(amazon.time) < 18 THEN 'Afternoon'

ELSE 'Evening'

END;

--2 Add a new column named dayname that contains the extracted days of the week on which the given transaction took place.

Alter Table amazon

ADD COLUMN day\_name varchar(50);

Update amazon

SET day\_name =DATE\_FORMAT(amazon.Date,'%a');

-- 3 Add a new column named monthname that contains the extracted months of the year on which the given transaction took place.

Alter Table amazon

ADD COLUMN month\_name varchar(50);

Update amazon

SET month\_name =DATE\_FORMAT(amazon.Date,'ba');

-- 1 What is the count of distinct cities in the dataset?

select count(distinct city) as cities from amazon;

--2 For each branch, what is the corresponding city?

select distinct branch,city from amazon;

--3 What is the count of distinct product lines in the dataset?

SELECT COUNT(DISTINCT 'product line') AS distinct\_Productline

FROM amazon;

--4 Which payment method occurs most frequently?

SELECT payment, COUNT(\*) AS payment\_count

FROM amazon

GROUP BY payment

ORDER BY payment\_count DESC

LIMIT 1;

--5 Which product line has the highest sales?

SELECT 'product\_line', Count('Invoice ID') as sales\_count

FROM amazon

GROUP BY 'product\_line'

ORDER BY sales\_count DESC

LIMIT 1;

-- 6 How much revenue is generated each month?

SELECT

YEAR(date) AS year,

MONTH(date) AS month,

SUM(total) AS monthly\_revenue

FROM

amazon

GROUP BY YEAR(date) , MONTH(date)

ORDER BY year , month;

-- 7 In which month did the cost of goods sold reach its peak?

SELECT

YEAR(date) AS year,

MONTH(date) AS month,

SUM(cogs) AS total\_cogs

FROM

amazon

GROUP BY

YEAR(date), MONTH(date)

ORDER BY

total\_cogs DESC

LIMIT 1;

--8 Which product line generated the highest revenue?

SELECT

'product\_line',

SUM(total) AS total\_revenue

FROM

amazon

GROUP BY

'product\_line'

ORDER BY

total\_revenue DESC

LIMIT 1;

-- 9 In which city was the highest revenue recorded?

SELECT

city,

SUM(total) AS total\_revenue

FROM

amazon

GROUP BY

city

ORDER BY

total\_revenue DESC

LIMIT 1;

-- 10 Which product line incurred the highest Value Added Tax?

SELECT

'product\_line',

SUM('VAT') AS total\_VAT

FROM

amazon

GROUP BY

'product\_line'

ORDER BY

total\_VAT DESC

LIMIT 1;

-- 11 For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."

SELECT

'product\_line',

total\_sales,

CASE

WHEN total\_sales > AVG(total\_sales) OVER () THEN 'Good'

ELSE 'Bad'

END AS sales\_status

FROM (

SELECT

'product\_line',

SUM(total) AS total\_sales

FROM

amazon

GROUP BY

'product\_line'

) AS sales\_summary;

-- 12 Identify the branch that exceeded the average number of products sold.

SELECT

branch,

AVG(quantity) AS avg\_quantity,

CASE

WHEN AVG(quantity) > (SELECT AVG(quantity) FROM amazon) THEN 'Exceeded'

ELSE 'Below or Equal'

END AS sales\_status

FROM

amazon

GROUP BY

branch;

-- 13 Which product line is most frequently associated with each gender?

SELECT

gender,

'product\_line',

COUNT(\*) AS frequency

FROM

amazon

GROUP BY

gender, 'product\_line'

ORDER BY

gender, frequency DESC;

-- 14 Calculate the average rating for each product line.

SELECT

'product\_line',

AVG(rating) AS average\_rating

FROM

amazon

GROUP BY

'product\_line';

-- 15 Count the sales occurrences for each time of day on every weekday.

SELECT

DAYNAME(date) AS weekday,

HOUR(time) AS hour\_of\_day,

COUNT(\*) AS sales\_occurrences

FROM

amazon

GROUP BY

weekday, hour\_of\_day

ORDER BY

FIELD(weekday, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'), hour\_of\_day;

-- 16 Identify the customer type contributing the highest revenue.

SELECT

'customer\_type',

SUM(total) AS total\_revenue

FROM

amazon

GROUP BY

'customer\_type'

ORDER BY

total\_revenue DESC

LIMIT 1;

-- 17 Determine the city with the highest VAT percentage.

SELECT

city,

SUM('VAT') AS total\_VAT,

SUM(total) AS total\_sales,

(SUM('VAT') / SUM(total)) \* 100 AS VAT\_percentage

FROM

amazon

GROUP BY

city

ORDER BY

VAT\_percentage DESC

LIMIT 1;

-- 18 Identify the customer type with the highest VAT payments.

SELECT

'customer\_type',

SUM('VAT') AS total\_VAT\_payments

FROM

amazon

GROUP BY

'customer\_type'

ORDER BY

total\_VAT\_payments DESC

LIMIT 1;

-- 19 What is the count of distinct customer types in the dataset?

SELECT COUNT(DISTINCT 'customer\_type') AS distinct\_customer\_types

FROM amazon;

-- 20 What is the count of distinct payment methods in the dataset?

SELECT COUNT(DISTINCT 'payment\_method') AS distinct\_payment\_methods\_count

FROM amazon;

-- 21 Which customer type occurs most frequently?

SELECT

'customer\_type',

COUNT(\*) AS frequency

FROM

amazon

GROUP BY

'customer\_type'

ORDER BY

frequency DESC

LIMIT 1;

-- 22 Identify the customer type with the highest purchase frequency.

SELECT

'customer\_type',

COUNT(\*) AS purchase\_frequency

FROM

amazon

GROUP BY

'customer\_type'

ORDER BY

purchase\_frequency DESC

LIMIT 1;

-- 23 Determine the predominant gender among customers.

SELECT gender, COUNT(\*) AS gender\_count

FROM amazon

GROUP BY gender

ORDER BY gender\_count DESC

LIMIT 1;

--24 Examine the distribution of genders within each branch.

SELECT branch, gender, COUNT(\*) AS gender\_count

FROM amazon

GROUP BY branch, gender

ORDER BY branch, gender\_count DESC;

--25 Identify the time of day when customers provide the most ratings.

SELECT HOUR(time) AS hour\_of\_day, COUNT(\*) AS rating\_count

FROM amazon

GROUP BY hour\_of\_day

ORDER BY rating\_count DESC

LIMIT 1;

--26 Determine the time of day with the highest customer ratings for each branch.

SELECT branch, HOUR(time) AS hour\_of\_day, COUNT(\*) AS rating\_count

FROM amazon

GROUP BY branch, hour\_of\_day

ORDER BY branch, rating\_count DESC;

--27 Identify the day of the week with the highest average ratings.

SELECT DAYNAME(date) AS day\_of\_week, AVG(rating) AS average\_rating

FROM amazon

GROUP BY day\_of\_week

ORDER BY average\_rating DESC

LIMIT 1;

--28 Determine the day of the week with the highest average ratings for each branch.

SELECT branch, DAYNAME(date) AS day\_of\_week, AVG(rating) AS average\_rating

FROM amazon

GROUP BY branch, day\_of\_week

ORDER BY branch, average\_rating DESC;