**Retail Sales Analysis - Case Study**

Problem Statements

1. Write a SQL query to retrieve all columns for sales made on '2022-11-05:
2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 4 in the month of Nov-2022:
3. Write a SQL query to calculate the total sale (net\_sale) and total orders for each category.:
4. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.:
5. Write a SQL query to find all transactions where the total\_sale is greater than 1000.:
6. Write a SQL query to find the total number of transactions (transaction\_id) made by each gender in each category.:
7. Write a SQL query to calculate the average sale for each month. Find out bestselling month in each year:
8. Write a SQL query to find the top 5 customers based on the highest total sales:
9. Write a SQL query to find the number of unique customers who purchased items from each category:
10. Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17):

--create a database

CREATE DATABASE retail\_sales;

--create a table

CREATE TABLE sales

(

transactions\_id INT PRIMARY KEY,

sale\_date DATE,

sale\_time TIME,

customer\_id INT,

gender VARCHAR(10),

age INT,

category VARCHAR(35),

quantity INT,

price\_per\_unit FLOAT,

cogs FLOAT,

total\_sale FLOAT

);

-- count rows

SELECT COUNT(\*) FROM sales;

-- count customer without duplicate

SELECT COUNT(DISTINCT customer\_id) FROM sales;

-- show category

SELECT DISTINCT category FROM sales;

-- check null

SELECT \* FROM sales

WHERE

sale\_date IS NULL

OR

sale\_time IS NULL

OR

customer\_id IS NULL

OR

gender IS NULL

OR

age IS NULL

OR

category IS NULL

OR

quantiy IS NULL

OR

price\_per\_unit IS NULL

OR

cogs IS NULL;

-- delete the row with a null value

DELETE FROM sales

WHERE

sale\_date IS NULL

OR

sale\_time IS NULL

OR

customer\_id IS NULL

OR

gender IS NULL

OR

age IS NULL

OR

category IS NULL

OR

quantiy IS NULL

OR

price\_per\_unit IS NULL

OR

cogs IS NULL;

-- View table data

SELECT \* FROM sales

-- 1. Write a SQL query to retrieve all columns for sales made on '2022-11-05:

SELECT

\*

FROM

sales

WHERE

sale\_date = '2022-11-05'

-- 2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 4 in the month of Nov-2022:

SELECT

\*

FROM

sales

WHERE

category = 'Clothing'

AND CAST ( sales.sale\_date AS DATE ) >= DATE'2022-11-01'

AND CAST ( sales.sale\_date AS DATE ) < DATE'2022-12-01'

AND sales.quantiy >= 4

ORDER BY

sale\_date

-- 3. Write a SQL query to calculate the total sale (net\_sale) and total orders for each category.:

SELECT

category,

SUM ( total\_sale ) AS net\_sale,

COUNT ( \* ) AS total\_orders

FROM

sales

GROUP BY

1

-- 4. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.:

SELECT

ROUND( AVG ( age ), 2 ) AS avg\_age

FROM

sales

WHERE

category = 'Beauty'

-- 5. Write a SQL query to find all transactions where the total\_sale is greater than 1000.:

SELECT

\*

FROM

sales

WHERE

total\_sale > 1000

-- 6. Write a SQL query to find the total number of transactions (transaction\_id) made by each gender in each category.:

SELECT

category,

gender,

COUNT ( \* ) AS total\_trans

FROM

sales

GROUP BY

1,

2

ORDER BY

1

--7. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year:

SELECT YEAR

,

MONTH,

avg\_sale

FROM

(

SELECT EXTRACT

( YEAR FROM sale\_date ) AS YEAR,

EXTRACT ( MONTH FROM sale\_date ) AS MONTH,

AVG ( total\_sale ) AS avg\_sale,

RANK ( ) OVER ( PARTITION BY EXTRACT ( YEAR FROM sale\_date ) ORDER BY AVG ( total\_sale ) DESC ) AS RANK

FROM

sales

GROUP BY

1,

2

) AS t1

WHERE

RANK = 1

--8. Write a SQL query to find the top 5 customers based on the highest total sales

SELECT

customer\_id,

SUM ( total\_sale ) AS total\_sale

FROM

sales

GROUP BY

customer\_id

ORDER BY

2 DESC

LIMIT 5

--9. Write a SQL query to find the number of unique customers who purchased items from each category.

SELECT

category,

COUNT ( DISTINCT customer\_id ) AS count\_of\_unique\_cust

FROM

sales

GROUP BY

category

--10. Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17)

WITH hourly\_sale AS (

SELECT

CASE

WHEN EXTRACT

( HOUR FROM sale\_time ) < 12 THEN

'Morning'

WHEN EXTRACT ( HOUR FROM sale\_time ) BETWEEN 12

AND 17 THEN

'Afternoon' ELSE'Evening'

END AS shift

FROM

sales

) SELECT

shift,

COUNT ( \* ) AS total\_orders

FROM

hourly\_sale

GROUP BY

shift