**Introduction**

The following is an Exploratory Data Analysis (EDA) of a sales data using MySQL. The aim of the analysis is to retrieve some insights about the data set, so that to advise business owners on the best strategies to put in place to improve sales and customers service in their enterprises.

**Creation Of Database and Sales Table**

First, we create a database and a table where the data is to be stored. After the table is creating the dataset downloaded from Kaggle is then import into MySQL table for exploration.

CREATE database MySQL\_ProjectP1;

CREATE TABLE retail sales

(

transactions\_id INT PRIMARY KEY,

sale\_date DATE,

sale\_time TIME,

customer\_id INT,

gender VARCHAR (15),

age INT,

category VARCHAR (15),

quantity INT,

price\_per\_unit FLOAT,

cogs FLOAT,

total sale FLOAT

)

**Checking That the Data Is Correctly Import**

To ensure the all the records are correctly imported we check the number of records

SELECT COUNT(\*) from retail\_sales;

In this case we have a total of 1987 records

**Check For the Null in The Dataset**

Null values interfere with proper analysis of the data and so they have to be checked and deleted from the data set if they do not have significant effect on the analysis.

SELECT \* FROM retail\_sales

WHERE

transactions\_id IS NULL

or

sale\_date IS NULL

or

sale\_time IS NULL

or

gender IS NULL

or

category IS NULL

or

quantity IS NULL

or

cogs IS NULL

or

total\_sale is NULL;

There are no null values in the dataset set and therefore we can proceed to answer some business-related problems

**BUSINESS RELATED PROBLEMS**

Q1. What are the sales made on ‘2022-11-06’

SELECT \* FROM retail\_sales

WHERE sale\_date = '2022-11-05';

This gives all the transactions the happened on that specific date.

Q2. Write an SQL query where the category is Beauty and the quantity is more 4 in the month of November 2022.

SELECT \* FROM retail\_sales

WHERE category = 'Beauty'

AND YEAR(sale\_date) = '2022'

AND MONTH(sale\_date) = '11'

AND quantity >=4 ;

In in the dataset only six transactions meet all the criteria listed

Q3. Write a SQL query to calculate the total sale for each category and total number of orders

ELECT category,sum(total\_sale) as category\_sale, COUNT(\*) as total\_orders

FROM retail\_sales

group by 1;

Q4. Write an SQl query to get the average age of customers who bought from the beauty category

SELECT ROUND(AVG(age), 2) as avg\_age

FROM retail\_sales

WHERE category = 'Beauty';

And this case the average age for a person buying from beauty category is 40 years

Q5. Write an SQL query find all the transactions where the total\_sale is greater than 1000

SELECT

\* FROM retail\_sales

WHERE total\_sale > 1000;

Q6. Write an SQL query to find the total number of transactions made by each gender in each category

SELECT category, gender, COUNT(\*) as total\_transactions

FROM retail\_sales

GROUP BY gender, category

ORDER BY gender;

Q7. Write a SQL query to calculate the average sale for each month. Find out the best selling month in each year

In this case we have to use the window function Rank() to get the best selling month in each year

SELECT year,

month,

avg\_sale

FROM

(

SELECT YEAR(sale\_date) as year , MONTH(sale\_date) as month,

AVG(total\_sale) as avg\_sale,

RANK() over (PARTITION BY YEAR(sale\_date) ORDER BY AVG(total\_sale) DESC) as rnk

FROM retail\_sales

GROUP BY 1,2

) as t1 WHERE rnk =1;

Q8. Write a query to find the top five customers based on total sales

SELECT DISTINCT customer\_id , SUM(total\_sale ) as total\_customer\_sale

FROM retail\_sales

GROUP BY customer\_id

ORDER BY SUM(total\_sale) DESC

LIMIT 5;

Q9. Write a SQL query to find the number of unique customers who purchased items from each category

SELECT category, COUNT( DISTINCT customer\_id) AS total\_customers

FROM retail\_sales

GROUP BY category;

Q10. Write a SQL query to create each shift and the number of sales in each shift (Morning, Afternoon and Evening)

WITH sales\_per\_hour

AS

(

SELECT

CASE

WHEN HOUR(sale\_time) <= 12 THEN 'Morning'

WHEN HOUR(sale\_time) BETWEEN 12 AND 17 THEN 'Afternoon'

ELSE 'Evening' END as Shift

FROM retail\_sales

)

SELECT Shift,

COUNT(\*) AS total\_sales

FROM sales\_per\_hour

GROUP BY Shift;

We determine that the evening shift has the highest number of orders