**Retail Sales Analysis SQL Project**

**Project Overview**

**Project Title**: Retail Sales Analysis  
**Level**: Beginner  
**Database**: potfolio\_project

This project is designed to demonstrate SQL skills and techniques typically used by data analysts to explore, clean, and analyze sales data. The project involves setting up a sales database, performing exploratory data analysis (EDA), and answering specific business questions via SQL queries.

**Objectives**

1. **Set up a sales database**: Create and populate a sales database with the provided sales data.
2. **Data Cleaning**: (a). Identify and remove any records with missing or null values. (b). Check for outliers using interquartile range method. And (c). Check for duplicate values and remove them if there is any duplicate values.
3. **Exploratory Data Analysis (EDA)**: Perform basic exploratory data analysis to understand the dataset.
4. **Business Analysis**: Use POASTGRESQL to answer specific business questions and derive insights from the sales data.

**Project Structure**

**1. Database Setup**

* **Database Creation**: The project starts by creating a database named potfolio\_project.
* **Table Creation**: A table named retail\_sale is created to store the sales data. The table structure includes columns for transaction\_id, sale\_date, sale\_time, customer\_id, gender, age, product category, quantity sold, price per unit, cost of goods sold (COGS), and total\_sale amount.

CREATE DATABASE potfolio\_project;

CREATE TABLE retail\_sale

(

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

);

After the creation of Table, the dataset was imported into postgreSQL.

**2. Data Cleaning**

* **Check for Null Value**: Check for any null values in the dataset and delete records with missing data.
* **Check for Duplicate**: Check for any duplicate records and remove the duplicate record in the dataset
* **Check for Outliers**: Check for outliers in the dataset using Interquartile range (IGR) method.
* Check the total number of records remain in the dataset.

--(1). Check for NULL value

SELECT \*

FROM retail\_sale

WHERE transaction\_id IS NULL;

SELECT \*

FROM retail\_sale

WHERE sale\_date IS NULL;

SELECT \*

FROM retail\_sale

WHERE sale\_time IS NULL;

SELECT \*

FROM retail\_sale

WHERE gender IS NULL;

SELECT \*

FROM retail\_sale

WHERE age IS NULL;

--Since age has null

(2). View the number of null in age column

SELECT COUNT (\*)

FROM retail\_sale

WHERE age IS NULL;

--Delete all the null value

DELETE FROM retail\_sale

WHERE

transaction\_id IS NULL OR 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 quantity IS NULL OR price\_per\_unit IS NULL OR cogs IS NULL OR total\_sale IS NULL;

--(3). Check for DUPLICATE value.

SELECT transaction\_id, COUNT(\*)

FROM retail\_sale

GROUP BY transaction\_id

HAVING COUNT(\*) > 1;

--(4). Check for OUTLIERS using Interuartlies range (IGR)

WITH quartiles AS (SELECT PERCENTILE\_CONT(0.25)

WITHIN GROUP (ORDER BY age) AS Q1,

PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY age) AS Q3

FROM retail\_sale),

iqr\_calc AS (SELECT Q1,Q3, (Q3 -Q1) AS IQR

FROM quartiles)

select \*

FROM retail\_sale

CROSS JOIN iqr\_calc

WHERE age < Q1 -1.5 \* IQR

OR age > Q3 + 1.5 \*IQR;

--(5). Check the total number of dataset

SELECT COUNT (\*) AS total\_number\_of\_dataset

FROM retail\_sale;

**2. Data Exploration**

* **View dataset**: View the first 10 rows of the entire dataset.
* **Record Count**: Determine the total number of records in the dataset.
* **Customer Count**: Find out how many unique customers are in the dataset.
* **Category Count**: Identify all unique product categories in the dataset.

--(a). What is the total sales?

SELECT COUNT(\*) AS total\_sale

FROM retail\_sale;

--(b). How many unique customers did we have?

SELECT COUNT(DISTINCT customer\_id) AS total\_number\_of\_customers

FROM retail\_sale;

--(c). How many male customers did we have in the dataset?

SELECT COUNT (\*) AS male\_customers

FROM retail\_sale

WHERE gender = 'Male';

--(d). How many female customers we have in the dataset?

SELECT COUNT (\*) AS female\_customers

FROM retail\_sale

WHERE gender = 'Female';

--(e). What is the percentage of male customers?

SELECT COUNT(CASE WHEN gender = 'Male' THEN 1 END) AS male\_count,

COUNT (\*) AS total\_count,

(COUNT(CASE WHEN gender = 'Male'THEN 1 END)\*100.0 / COUNT (\*))

AS Male\_percentage

FROM retail\_sale;

--(f). What is the percentage of female customers in the dataset?

SELECT COUNT(CASE WHEN gender = 'Female'THEN 1 END) AS female\_count,

COUNT (\*) AS total\_count,

(COUNT(CASE WHEN gender = 'Female'THEN 1 END)\*100.0 / COUNT (\*)) AS Female\_percentage

FROM retail\_sale;

--(g). What is the Minimum age of the customers in the dataset?

SELECT MIN(age) AS minimum\_age

FROM retail\_sale;

--(h). What is the maximum age of the customers in the dataset?

SELECT MAX(age) AS maximum\_age

FROM retail\_sale;

--(i). What is the average age of the customers in whole number?

SELECT ROUND(AVG(age), 0) AS Average\_age

FROM retail\_sale;

--(j). How many category of the product did we have?

SELECT DISTINCT category AS product\_cateory

FROM retail\_sale;

**3. Data Analysis & Findings**

The following SQL queries were developed to answer specific business questions:

**Q.1 Write SQL query to retrieve total sales, total transaction and all sales made on both 2022-02-03 and 2022-02-05?**

SELECT SUM(total\_sale)

FROM retail\_sale;

SELECT SUM(transaction\_id)

FROM retail\_sale;

SELECT \*

FROM retail\_sale

WHERE sale\_date = '2022-02-03' OR sale\_date = '2022-02-05'

ORDER BY sale\_date, age ASC;

**Q.2 Write a query to retrieve all transactions where the category is Beauty and the Quantity sold is more than 2 in the month of July. 2022**

SELECT \*

FROM retail\_sale

WHERE category = 'Beauty'

AND TO\_CHAR(sale\_date, 'YYYY-MM') = '2022-07'

AND quantity >= 2;

**Q.3 Writ a SQL query to calculate the total sales (total\_sales) for each category**

SELECT category, COUNT(\*) AS Total\_Orders,

SUM(total\_sale) AS Total\_sales\_for\_each\_category

FROM retail\_sale

GROUP BY category;

**Q.4 Writ the query to find the average age of customers who purchased items from the 'Beauty' category**

SELECT category, ROUND(AVG(age),4) AS Beauty\_average\_age

FROM retail\_sale

GROUP BY category

HAVING category = 'Beauty';

**Q.5 Write the query to find the average sales per each age group and obtain the number of quantity sold per age group.**

SELECT

CASE

WHEN age < 18 THEN 'Under 18'

WHEN age BETWEEN 18 AND 25 THEN '18-25'

WHEN age BETWEEN 26 AND 35 THEN '26-35'

WHEN age BETWEEN 36 AND 50 THEN '36-50'

ELSE 'Above 50'

END AS age\_group,

SUM(total\_sale) AS total\_sale\_per\_age\_group, COUNT(quantity) AS quantity\_sold

FROM retail\_sale

GROUP BY CASE

WHEN age < 18 THEN 'Under 18'

WHEN age BETWEEN 18 AND 25 THEN '18-25'

WHEN age BETWEEN 26 AND 35 THEN '26-35'

WHEN age BETWEEN 36 AND 50 THEN '36-50'

ELSE 'Above 50'

END

ORDER BY total\_sale\_per\_age\_group DESC;

**Q.6 Write a SQL query to find all transactions where the total\_sale is greater than 1000.**

SELECT COUNT(total\_sale) AS Transaction\_greater\_1000

FROM retail\_sale

WHERE total\_sale > 1000;

**Q.7 Write a SQL query to find the total number of transactions (transaction\_id) made by each gender in each category.**

SELECT category, gender, COUNT(transaction\_id) AS Total\_sale\_by\_gender\_for\_each\_category

FROM retail\_sale

GROUP BY category, gender

ORDER BY 1;

**Q.8 Write a SQL query to calculate the average sale for each month. Find out best selling month in each year:**

SELECT

EXTRACT(YEAR from sale\_date) AS Year,

EXTRACT(MONTH FROM sale\_date) AS Month,

AVG(total\_sale) AS Average\_sales

FROM retail\_sale

GROUP BY 1,2

ORDER BY 3 DESC;

**--Q.9 Determine the top 5 customers who contributed to total sales and calculate their contribution percentage**

-- Calculate total sales by all customers

WITH total\_sale AS (

SELECT SUM(total\_sale) AS total\_sale\_amount

FROM retail\_sale

),

-- Calculate sales by each customer

customer\_sale AS (

SELECT customer\_id,

SUM(total\_sale) AS customer\_total\_sale

FROM retail\_sale

GROUP BY customer\_id

),

-- Rank customers by sales and select the top 5

top\_5\_customer AS (

SELECT customer\_id,

customer\_total\_sale

FROM customer\_sale

ORDER BY customer\_total\_sale DESC

LIMIT 5

)

-- Calculate the percentage contribution of the top 5 customers

SELECT customer\_id,

customer\_total\_sale,

(customer\_total\_sale / total\_sale.total\_sale\_amount) \* 100 AS contribution\_percentage

FROM top\_5\_customer, total\_sale

ORDER BY contribution\_percentage DESC;

**Q.10 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\_unique\_customer

FROM retail\_sale

GROUP BY 1

ORDER BY 2 DESC;

**Q.11 Write an SQL query to calculate the average age for each category product**

SELECT

category,

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

FROM retail\_sale

GROUP BY category

ORDER BY average\_age DESC;

**Q.12 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 retail\_sale

)

SELECT shift,

COUNT(\*) AS total\_orders

FROM hourly\_sale

GROUP BY shift

ORDER BY 2 DESC;

### ****Exploratory Data Analysis (EDA)****

Key observations from the cleaned dataset:

1. **Customer Distribution**:
   * Total unique customers: **[1987]**.
   * Male customers: **[There are 975 Male customers taking over 49.0689% of the entire population of the customers]**.
   * Female customers: **[There are 975 Female customers taking over 50.9311% of the entire population of the customers]**.
2. **Age Insights**:
   * Minimum age: **[The minimum age of the customer is 18 years]**.
   * Maximum age: **[While the maximum age of the customer is 64]**.
   * Average age: **[The average age of the customer is 41]**.
3. **Product Categories**:
   * Unique product categories: **[There are 3 categories of products in the dataset]**.

### ****Business Analysis****

Using SQL, several business questions were answered to derive actionable insights:

#### **1. Total Sales**

* Total transactions: **[ is 1988020]**.
* Total sales amount: **[is 908230]**.

#### **2. Customer Insights**

* Average age of customers purchasing from the "Beauty" category: **[average age for beauty]**.
* Percentage of male and female customers:
  + Male: **[**Percentage of male **is 49.0689]%**
  + Female: **[**Percentage of female **is 50.9311]%**

#### **3. Product and Sales Trends**

* Top-selling categories based on total sales:



#### **4. Age Group Analysis**

* Highest spending age group: **[The** highest spending age group **is age\_group 36-50 with 279245 total sale]**
* Age group contributing most to quantity sold: **[**Age group contributing most to quantity sold **is age\_group 36-50 with 625 quantity sold]**



#### **5. High-Value Transactions**

* Total transactions with sales > 1000: **[is 306]**.

#### **6. Time-Based Analysis**

* Best-selling month: **[The overall best-selling month is July in year 2022 with average sales of 541.3415 followed by February in year 2023 with the average sales of 535.5319]**.

****

* Most orders were placed during the **evening shift**, with **1062** orders.



#### **7. Customer Performance**

* Top 5 customers contributed to total sales with their percentage. These customers were identified for loyalty programs.



### ****Key Insights****

1. **Category Insights**:
   * The Clothing category has a higher average age of customers compared to other categories, indicating that marketing strategies can target this demographic more effectively.
2. **High-Spending Age Group**:
   * The **36-50** age group spends the most, making it a primary target for premium product marketing.
3. **Time and Sales Patterns**:
   * Sales peak during the **Evening shift**, suggesting potential for targeted promotions or staffing optimizations during this period.
4. **High-Value Transactions**:
   * **306** of transactions account for significant sales, emphasizing the need to prioritize high-value customer segments.
5. **Seasonality**:
   * **July in year 2022** shows seasonal trends, which can guide inventory and marketing campaigns.

### ****Conclusion****

The analysis highlights patterns in customer demographics, product preferences, and sales trends. These findings can inform strategic decisions in areas such as inventory management, targeted marketing, and customer segmentation. Future work could incorporate predictive analytics and deeper time-series analysis to anticipate sales trends.

**THANKS. ADEKUNLE TIMOTHY**