**SQL RETAIL SALES PROJECT**

**PROJECT TITLE: RETAIL SALES ANALYSIS**

**PROJECT OVERVIEW**

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

OBJECTIVES

1. **Set up a retail sales database**: Create and populate a retail sales database with the provided sales data.
2. **Data Cleaning**: Identify and remove any records with missing or null values.
3. **Exploratory Data Analysis (EDA)**: Perform basic exploratory data analysis to understand the dataset.
4. **Business Analysis**: Use SQL 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 ‘RetailDB’

**Table Creation**: A table named retail\_sales is created to store the sales data.

CREATE DATABASE RetailDB;

USE RetailDB;

--Create TABLE

CREATE TABLE retail\_sales

(

transaction\_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

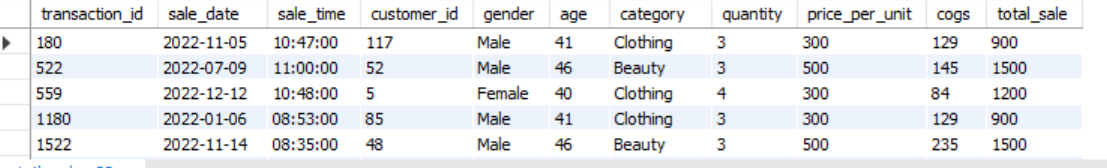
);

**2. Data Exploration & Cleaning**

* **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.
* **Null Value Check**: Check for any null values in the dataset and delete records with missing data.

SELECT \* FROM retail\_sales

LIMIT 10;



-- How many sales we have?

SELECT COUNT(\*) AS total\_sale

FROM retail\_sales;

A close-up of a text

AI-generated content may be incorrect.

-- How many unique customers we have ?

SELECT COUNT(DISTINCT customer\_id) AS Distinct\_total\_sale

FROM retail\_sales;

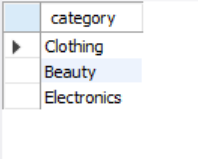
A screenshot of a computer

AI-generated content may be incorrect.

-- How many uniuque categories we have ?

SELECT DISTINCT category

FROM retail\_sales;



SELECT \* FROM retail\_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

quantity IS NULL OR price\_per\_unit IS NULL OR cogs IS NULL;

Alter Table retail\_sales

Rename column ï»¿transactions\_id to transaction\_id;

Alter Table retail\_sales

Rename column quantiy to quantity;

**Data Analysis & Findings**

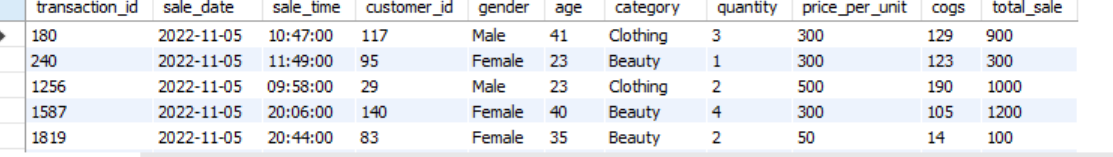
The following SQL queries were developed to answer specific business questions:

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

SELECT \*

FROM retail\_sales

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



-- Q.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 \*

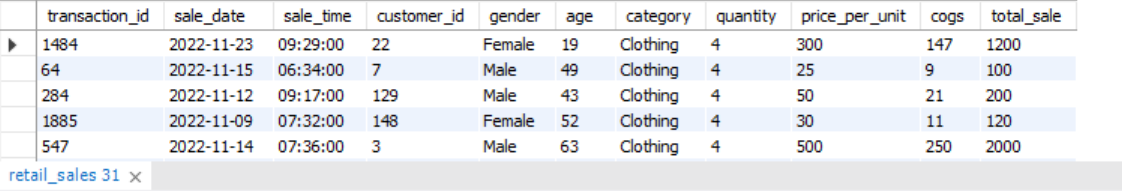
FROM retail\_sales

WHERE

category = 'Clothing'

AND DATE\_FORMAT(sale\_date, '%Y-%m') = '2022-11'

AND quantity >= 4;



-- Q.3 Write a SQL query to calculate the total sales (total\_sale) for each category.

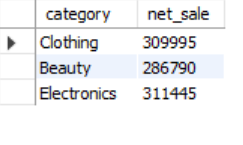
SELECT

category,

SUM(total\_sale) as net\_sale

FROM retail\_sales

GROUP BY category;



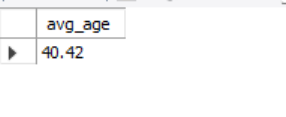
-- Q.4 Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

SELECT

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

FROM retail\_sales

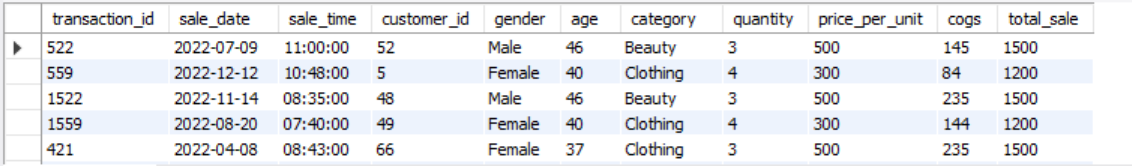
WHERE category = 'Beauty';



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

SELECT \* FROM retail\_sales

WHERE total\_sale > 1000;



-- Q.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 retail\_sales

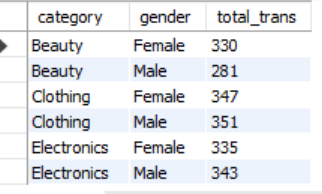
GROUP

BY

category,

gender

ORDER BY 1;



-- Q.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,

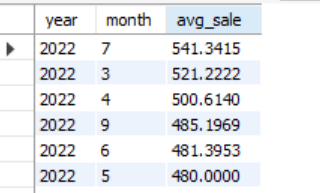
AVG(total\_sale) as avg\_sale,

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

FROM retail\_sales

GROUP BY 1, 2

) as t1;



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 ranks

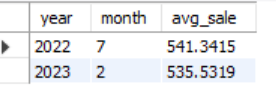
FROM retail\_sales

GROUP BY 1, 2

) as t1

WHERE ranks = 1

ORDER BY 1, 3 DESC;



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

SELECT

customer\_id,

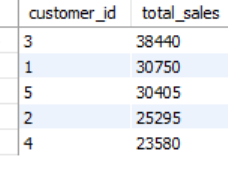
SUM(total\_sale) as total\_sales

FROM retail\_sales

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5;



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

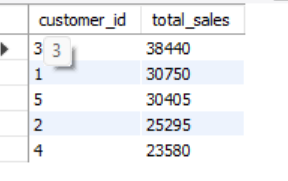
SELECT

category,

COUNT(DISTINCT customer\_id) as cnt\_unique\_cs

FROM retail\_sales

GROUP BY category;



-- Q.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 retail\_sales

)

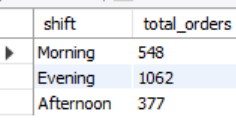
SELECT

shift,

COUNT(\*) as total\_orders

FROM hourly\_sale

GROUP BY shift;



**Findings**

* **Customer Demographics**: The dataset includes customers from various age groups, with sales distributed across different categories such as Clothing and Beauty.
* **High-Value Transactions**: Several transactions had a total sale amount greater than 1000, indicating premium purchases.
* **Sales Trends**: Monthly analysis shows variations in sales, helping identify peak seasons.
* **Customer Insights**: The analysis identifies the top-spending customers and the most popular product categories.

**Reports**

* **Sales Summary**: A detailed report summarizing total sales, customer demographics, and category performance.
* **Trend Analysis**: Insights into sales trends across different months and shifts.
* **Customer Insights**: Reports on top customers and unique customer counts per category.