**Retail Sales Analysis SQL Project**

**Project Overview**

**Project Title**: Retail Sales Analysis  
**Level**: Beginner  
**Database**: project\_retail\_db

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. This project is ideal for those who are starting their journey in data analysis and want to build a solid foundation in SQL.

**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 p1\_retail\_db.
* **Table Creation**: A table named retail\_sales 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 p1\_retail\_db;

CREATE TABLE retail\_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

);

**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 COUNT(\*) FROM retail\_sales;

SELECT COUNT(DISTINCT customer\_id) FROM retail\_sales;

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;

DELETE 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;

**3. Data Analysis & Findings**

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

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';

1. **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

TO\_CHAR(sale\_date, 'YYYY-MM') = '2022-11'

AND

quantity >= 4

1. **Write a SQL query to calculate the total sales (total\_sale) for each category.**:

SELECT

category,

SUM(total\_sale) as net\_sale,

COUNT(\*) as total\_orders

FROM retail\_sales

GROUP BY 1

1. **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'

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

SELECT \* FROM retail\_sales

WHERE total\_sale > 1000

1. **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

1. **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 retail\_sales

GROUP BY 1, 2

) as t1

WHERE rank = 1

1. \*\*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

1. **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

1. **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.

**Conclusion**

This project serves as a comprehensive introduction to SQL for data analysts, covering database setup, data cleaning, exploratory data analysis, and business-driven SQL queries. The findings from this project can help drive business decisions by understanding sales patterns, customer behavior, and product performance.

**How to Use**

1. **Clone the Repository**: Clone this project repository from GitHub.
2. **Set Up the Database**: Run the SQL scripts provided in the database\_setup.sql file to create and populate the database.
3. **Run the Queries**: Use the SQL queries provided in the analysis\_queries.sql file to perform your analysis.
4. **Explore and Modify**: Feel free to modify the queries to explore different aspects of the dataset or answer additional business questions.