

Retail Sales Data Analysis Project (SQL)

1. Project Objective

The objective of this project is to **analyze retail sales data** using SQL. The dataset contains sales transactions with details such as date, time, customer info, category, price, and quantity. The project includes:

- Database and table setup
 - Data validation and cleaning
 - Basic exploration
 - Business-focused analytics through SQL queries
-

Database and Table Setup

2.1 Creating the Database and Schema

Began by ensuring the database and schema existed. If not, they were created:

```
CREATE DATABASE IF NOT EXISTS Analysis;
```

```
CREATE SCHEMA IF NOT EXISTS Analysis;
```

This step sets up the environment where the table and data will reside.

2.2 Creating the Retail Sales Table

Defined the structure of the retail_sales table using appropriate data types for each column:

```
CREATE TABLE IF NOT EXISTS 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  
);
```

This schema design enables comprehensive sales tracking with detailed attributes.

Data Validation and Cleaning

3.1 Viewing Initial Records

You used a simple query to preview the first 10 records:

```
SELECT * FROM retail_sales LIMIT 10;
```

3.2 Checking Total Records

You checked the total number of records in the dataset:

```
SELECT COUNT(*) FROM retail_sales;
```

3.3 Identifying NULL Values

To ensure data quality, you searched for rows with missing values across all fields:

```
SELECT * FROM retail_sales  
WHERE transactions_id IS NULL  
OR sale_date IS NULL  
...  
OR total_sale IS NULL;
```

3.4 Cleaning the Data

You deleted all rows containing NULL values to maintain data integrity:

```
DELETE FROM retail_sales
```

WHERE transactions_id IS NULL

OR sale_date IS NULL

...

OR total_sale IS NULL;

Basic Data Exploration

4.1 Total Valid Sales Records

You verified the number of clean records:

SELECT COUNT() AS total_sale FROM retail_sales;*

4.2 Unique Customers

To understand your customer base:

SELECT COUNT(DISTINCT customer_id) AS total_sales FROM retail_sales;

4.3 Product Categories

You checked the diversity of product offerings:

SELECT DISTINCT category AS total_categories FROM retail_sales;

Business Questions & Data Analysis

Q1: Sales on a Specific Date

Retrieve all transactions from **November 5, 2022**:

*SELECT * FROM retail_sales WHERE sale_date = '2022-11-05';*

Q2: High Quantity Clothing Sales in Nov-2022

Find transactions where category is **clothing**, quantity ≥ 4 , in **November 2022**:

```
SELECT * FROM retail_sales  
WHERE category = 'clothing' AND quantity >= 4  
AND sale_date BETWEEN '2022-11-01' AND '2022-11-30';
```

Q3: Total Sales by Category

Calculate **net sales per category**:

```
SELECT category, SUM(total_sale) AS net_sale FROM retail_sales  
GROUP BY category;
```

Q4: Average Age for Beauty Category

Determine the **average age** of customers buying **Beauty** products:

```
SELECT ROUND(AVG(age) AS Average_age FROM retail_sales  
WHERE category = 'Beauty';
```

Q5: High Value Transactions

Find all transactions with **total sales over 1000**:

```
SELECT * FROM retail_sales WHERE total_sale > 1000;
```

Q6: Gender-wise Sales per Category

Count number of transactions grouped by **category and gender**:

```
SELECT category, gender, COUNT(*) AS total_transactions FROM retail_sales  
GROUP BY category, gender;
```

Q7: Monthly Average Sales

Evaluate **average sales per month per year** and rank by best months:

```
SELECT YEAR(sale_date) AS year, MONTH(sale_date) AS month, AVG(total_sale) AS Average_Sales
FROM retail_sales
GROUP BY year, month
ORDER BY year, Average_Sales DESC;
```

Q8: Top 5 Customers by Total Sales

Find top 5 customers with the **highest total sales**:

```
SELECT customer_id, SUM(total_sale) AS total_sales
FROM retail_sales
GROUP BY customer_id
ORDER BY total_sales DESC
LIMIT 5;
```

Q9: Customer Count per Category

Get **unique customer count per product category**:

```
SELECT category, COUNT(DISTINCT customer_id) AS Number_of_Customers
FROM retail_sales
GROUP BY category;
```

Q10: Sales Shift Analysis

Categorize sales by time of day:

```
SELECT shift, COUNT(*) AS total_orders
FROM (
    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
) AS hourly_sale
GROUP BY shift;
```

This helps in identifying peak transaction hours by defining **Morning**, **Afternoon**, and **Evening** shifts.

6. Summary

Through this project, you:

- Set up a structured SQL database for retail transactions
- Performed data validation and cleansing
- Carried out basic and advanced exploratory analysis
- Answered key business questions such as top customers, peak sales periods, and category performance

This kind of structured SQL project is an excellent example of **data-driven decision making**, using SQL as the core tool for business intelligence.