**PROJECT TITLE: RETAIL SALES ANALYSIS**

This project involves the design and execution of an advanced SQL-based system to analyze retail sales data. The goal is to gain valuable insights into sales performance, customer behavior, product trends, and operational efficiency, which can help drive strategic business decisions.

**OBJECTIVES OF THE RETAIL SALES ANALYSIS PROJECT**

1. **Sales Trend Analysis**: Analyze the total sales over time (daily, monthly, yearly) to identify patterns, seasonal trends, and any fluctuations in sales volume.
2. **Customer Segmentation**: Use SQL to segment customers based on their purchasing behavior, such as total spending, frequency of purchases, and preferred product categories.
3. **Category and Product Performance Evaluation**: Evaluate the sales performance of different product categories and identify top-performing products based on total revenue and number of transaction.
4. **High-Value Transaction Identification**: Identify transactions where the total sale exceeds a certain threshold and analyze these high-value transactions to understand their contribution to overall sales.
5. **Time-Based Sales Analysis**: Classify sales transactions into different shifts (morning, afternoon, evening) and analyze how sales vary across different times of the day.
6. **Business Analysis**: Use SQL queries to answer specific business questions, such as identifying top customers, most popular categories, and factors influencing sales performance.

**PROJECT STRUCTURE**

1. **DATABASE SETUP**

* **Database Creation**: The project begins by setting up a database named retail\_db to store all sales-related data.

CREATE DATABASE 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 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

);

1. **DATA RETRIEVAL QUERIES**

* **Counting the total number of transactions in retail\_sales:**

SELECT COUNT(\*) FROM retail\_sales;

* **Counting the number of distinct customers in retail\_sales:**

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

1. **FILTERED DATA QUERIES**

* **Total Sales Trend Over Time for a Specific Date Range**

*Query : Retrieve the total sales trend for a specific date range (the year 2022), broken down by month.*

SELECT DATE\_FORMAT(sale\_date, '%Y-%m') AS month,

SUM(total\_sale) AS total\_sales

FROM retail\_sales

WHERE sale\_date BETWEEN '2022-01-01' AND '2022-12-31' -- filter for the year 2022

GROUP BY month

ORDER BY month;

* **Calculate total sales and orders by category:**

*Query: Retrieves the sum of sales and the total number of orders for each category*

SELECT category,

SUM(total\_sale) as net\_sale,

COUNT(\*) as total\_orders

FROM retail\_sales

GROUP BY category;

* **Find the average age of customers who purchased items from the 'Beauty' category:**

*Query: Retrieves the sum of sales and the total number of orders for the 'Beauty' category*

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

FROM retail\_sales

WHERE category = 'Beauty';

* **Find all transactions where the total sale is greater than 1000:**

*Query: Retrieves all the transactions where the total sale amount exceeds 1000.*

SELECT \*

FROM retail\_sales

WHERE total\_sale > 1000;

1. **AGGREGATED DATA QUERIES**

* Count transactions by gender and category:

*Query: Retrieves a count of the total transactions for each gender within each product category.*

SELECT category,

gender,

COUNT(\*) as total\_trans

FROM retail\_sales

GROUP BY category, gender

ORDER BY category;

* **Top 5 customers based on highest total sales:**

*Query: Retrieve the five customers who have made the highest total purchases.*

SELECT

customer\_id,

SUM(total\_sale) as total\_sales

FROM retail\_sales

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5;

* **Determine the number of unique customers who made purchases in each category:**

*Query: Retrieves the number of unique customers who have made purchases in each product category.*

SELECT

category,

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

FROM retail\_sales

GROUP BY category;

1. **TIME-BASED DATA QUERIES**

* **Classifying sales transactions into shifts and calculating total orders for each shift:**

*Query: Classify each transaction into a specific shift based on the sale time and return the total number of orders for each shift.*

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;

* **Identify the category with the highest number of total purchases:**

*Query: Retrieves the category with the highest number of total purchases.*

SELECT

category,

COUNT(\*) AS total\_purchases

FROM retail\_sales

GROUP BY category

ORDER BY total\_purchases DESC

LIMIT 1;

* **Calculate the total revenue generated by each category:**

*Query: Retrieves the total revenue generated by each product category.*

SELECT

category,

SUM(total\_sale) AS total\_revenue

FROM retail\_sales

GROUP BY category;

1. **DATA SEGREGATION**

* **Retrieving all data again to segregate it for documentation:**

SELECT \* FROM retail\_sales;

**CONCLUSION**

The **Retail Sales Analysis Project** uses SQL to analyze sales trends, customer behavior, and product performance. It provides insights into top-selling products, high-value transactions, and sales patterns over time. This analysis helps businesses make informed decisions to optimize marketing, inventory, and customer engagement.