PROJECT OVERVIEW: SALES DATA ANALYSIS

Project Introduction

This project is designed to refresh your memory on the concepts you've learned and to challenge you to explore new ideas beyond the course content. While collaboration with your peers is encouraged, I strongly recommend that you put in the effort to solve the tasks independently before seeking help. The process of figuring things out on your own will deepen your understanding and enhance your problem-solving skills.

Remember, the purpose of this project is not just to assess your current knowledge, but also to cultivate a growth mindset—the ability to seek out and learn new concepts on your own. Don’t hesitate to use online resources, documentation, and forums as you work through the tasks. These are valuable skills in the real world, where knowing how to find the answer is just as important as knowing the answer itself.

Take this as an opportunity to challenge yourself, build confidence, and showcase your skills! You’ll be sharing your work on GitHub and LinkedIn, so make sure to put your best effort into the project. Let it be a testament to your dedication and readiness to advance in the field of data analysis.

BUSINESS QUESTIONS

1. Database Creation & Schema Design

- Create a database called `sales_data`.
- Create the following tables:
- products: `product_id` (INT, PRIMARY KEY), `product_name`, `category`, `price`.
- customers: `customer_id` (INT, PRIMARY KEY), `first_name`, `last_name`, `email`, `phone_number`.
- sales: `sale_id` (INT, PRIMARY KEY), `product_id` (INT), `customer_id` (INT), `sale_date`, `quantity`, `total_amount`.
- payments: `payment_id` (INT, PRIMARY KEY), `sale_id` (INT), `payment_method`, `payment_date`, `amount_paid`.

2. Data Insertion

- Insert at least 10 products into the `products` table.
- Insert at least 40 customers into the `customers` table.
- Insert at least 20 records into the `sales` table, making sure some customers buy multiple products.
- Insert payment records corresponding to the sales, with varying amounts and dates.

3. Analysis Using SQL Concepts

- GROUP BY:

➢ Write a query to calculate the total quantity sold for each product category.
➢ Write a query to count the number of sales for each customer.
➢ Write a query to find the total revenue generated by each payment method.

- ORDER BY:

➢ Write a query to retrieve a list of products, sorted by price in descending order.
➢ Write a query to get the sales records ordered by `sale_date` in ascending order.
➢ Write a query to display customers sorted by their last name alphabetically.

- HAVING Clause:

➢ Write a query to find products that have been sold more than 5 times, using `HAVING` with the `GROUP BY` statement.
➢ Write a query to list customers who have made total purchases (sum of `total_amount`) exceeding $100.
➢ Write a query to find categories with a total sales revenue greater than $500.

- LIMIT Clause:

➢ Write a query to display the top 5 most expensive products.
➢ Write a query to display the 3 most recent sales.
➢ Write a query to list the first 5 customers who made purchases.

- ALIASING:

➢ Write a query to calculate the total revenue (sum of `total_amount`) for each product and use an alias `total_revenue` for the calculated field.
➢ Write a query to display each customer's full name as `customer_name` by concatenating `first_name` and `last_name`.
➢ Use aliasing to create a column called `sales_value` representing the product of `price` and `quantity` for each sale.

- String Functions:

➢ Write a query to display the product names in uppercase.

- Write a query to extract the domain from the `email` field of each customer.
- Write a query to display the first three characters of each customer's `last_name`.
- Write a query that trims extra spaces around product names and displays them.

- Aggregate Functions:

- Write a query to calculate the average price of all products.
- Write a query to find the maximum `total_amount` from the `sales` table.
- Write a query to calculate the total amount paid for all sales.
- Write a query to calculate the minimum and maximum sale quantities.
- Write a query to determine the number of distinct product categories available.

4. Additional Analysis Tasks

- Write a query to identify the customer with the highest total purchase amount.
- Calculate the total number of products sold for each product category and sort the result in descending order.
- Write a query that retrieves the details of sales made in the last 30 days.
- Create a query to display the top 3 customers with the most sales in terms of quantity.
- Write a query to list all customers whose names start with the letter "J".
- Write a query to calculate the total unpaid amount for each sale by subtracting `amount_paid` from `total_amount`.
- Write a query to find sales where the payment method was "Credit Card" and the `total_amount` is greater than $50.
- Write a query that lists all sales made by customers whose last names contain the letter "e".

Submission Instructions

1. Upload the Project to GitHub:

- Create a repository named `Sales-Data-Analysis`(or any name of your choice) on GitHub.

- Upload all the SQL scripts.

2. Share on LinkedIn:

- Write a post describing your experience with this project, what you learned, and include a link to your GitHub repository.

- Use relevant hashtags like `MySQL`, `DataAnalytics`, `SQLProject`, and `SalesDataAnalysis`.

3. Share in the WhatsApp Group:

- Send the linkedin post link to the whatsapp group.