PROJECT FINAL REPORT

**Name : **Hitesh Rajendra Deore

Organization : Elevate Labs

1. Introduction

This report summarizes the final project phase of the SQL Developer Internship at Elevate Labs. The objective of this phase was to apply the cumulative skills learned—from database design (DDL) and data manipulation (DML) to advanced querying (DQL) and analysis—into practical, portfolio-worthy projects.

This report details the development of two distinct projects:

1. **Online Retail Sales Database:** A database engineering project focused on designing, building, and deploying a normalized, relational database from scratch.
2. **COVID-19 Data Analytics:** A data analysis project focused on ingesting, cleaning, and analyzing a large, real-world dataset to derive meaningful insights.

Together, these projects demonstrate a comprehensive ability to manage the full data lifecycle, from architectural design to advanced analytics.

2. Tools and Technologies Used

- **Database:** MySQL Server
- **IDE/Client:** MySQL Workbench
- **Database Design:** dbdiagram.io
- **Data Source:** Kaggle (for Project 2)
- **Version Control:** Git & GitHub

3. Project 1: Online Retail Sales Database Design

3.1. Project Abstract

The goal of this project was to design and implement a scalable, normalized (3NF) relational database for a modern e-commerce platform. The system successfully models complex relationships between customers, products, and orders, and includes professional-grade features like a self-referencing categories table, a separate payments table, and historical price tracking. The final deliverable is a robust, secure, and efficient database backend ready to support a real-world application.

3.2. Steps Involved

1. **Database Design (ERD):** I began by identifying all core business entities (`Customers`, `Addresses`, `Products`, `Categories`, `Orders`, `Order_Items`, `Payments`). I used `dbdiagram.io` to create a detailed Entity-Relationship Diagram, mapping the one-to-many and many-to-many relationships. Key design features include:
   - A separate `Addresses` table to allow one customer to have multiple shipping and billing addresses.
   - A self-referencing `ParentCategoryID` in the `Categories` table to allow for infinite subcategories.
   - A `PriceAtPurchase` column in the `Order_Items` table to ensure accurate historical sales reporting, even if product prices change.
2. **Schema Implementation (DDL):** Using the ERD as a blueprint, I wrote the `schema.sql` script. This DDL script created the `retail_sales_db` and all tables, implementing all `PRIMARY KEY`, `FOREIGN KEY`, `UNIQUE`, and `CHECK` constraints to ensure data integrity and referential integrity.
3. **Data Population (DML):** I wrote a `data.sql` script to populate the database with realistic sample data. The `INSERT` statements were carefully ordered to respect foreign key constraints, populating parent tables (like `Customers`) before child tables (like `Orders`).
4. **Reporting & Analysis (DQL):** Finally, I wrote `reports.sql` to demonstrate the database's value. This included:
   - Creating a `v_SalesDetails` view to simplify complex 5-table joins.
   - Creating a `v_CustomerSpending` view to aggregate sales data per customer.
   - Writing queries against these views to answer key business questions like "Who is our top customer?" and "What is our best-selling product category?"

---

4. Project 2: COVID-19 Data Analytics

4.1. Project Abstract

The objective of this project was to leverage SQL for data analysis by ingesting a large, real-world dataset. I sourced a COVID-19 dataset from Kaggle, imported it into a MySQL database, and wrote a series of advanced analytical queries to identify trends, rankings, and daily figures. This project demonstrates the power of SQL as an analytical tool, including the use of window functions to derive insights from over 300,000 rows of data.

4.2. Steps Involved

1. **Data Ingestion:** I sourced the "Novel Corona Virus 2019 Dataset" from Kaggle, which contained over 300,000 individual case records. I used the MySQL Workbench Table Data Import Wizard to import the `covid_19_data.csv` file into a new `covid_analytics_db` database.
2. **Data Exploration:** After a failed import attempt due to data type mismatches, I re-imported the data, identifying the correct table name (`covid_19_data`) and schema structure for use in my queries.
3. **Data Analysis (DQL):** I wrote the `covid_analytics.sql` script to analyze the data. This script answered several key questions:

  - **Aggregation:** Used `GROUP BY` and `SUM()` to calculate the total confirmed cases, deaths, and recoveries for every country.
  - **Ranking:** Used `ORDER BY` and `LIMIT 10` to find the top 10 most-affected countries.
  - **Window Functions:**
    - Used `RANK()` to generate a global ranking of countries by their total confirmed cases.
    - Used `LAG()` to look at the previous day's "Confirmed" count, allowing for the calculation of *new daily cases* for a specific country—a critical metric that is not present in the raw data.

---

5. Overall Conclusion

These two projects solidified my practical SQL skills across the entire data lifecycle. Project 1 proved my ability in **database engineering**, covering architecture, normalization, and DDL implementation. Project 2 demonstrated my strengths in **data analytics**, including data ingestion and the use of advanced window functions for trend analysis. I am confident in my ability to use SQL to build robust database solutions and extract clear, actionable insights from complex datasets.