Part 1: Address Enrichment for Orders

We are working with a third-party service provider for order analysis. While most orders information include an address, some do not. In such cases, the service provider provides an IP address instead. Our goal is to extract the city, state, and zip code using these IP addresses and update our orders database accordingly.

Tasks:

1. Load Orders Data:

Develop a function to load order data from a CSV file (orders\_file.csv).

Insert data into orders table, ensuring duplicate uploads are ignored.

2. Load and Process IP Addresses:

Create a function to load IP addresses from a CSV file (ip\_address.csv).

Query multiple APIs to retrieve the corresponding city (optional), state, and zip code (mandatory).

Optimize API calls by reducing redundant requests.

This information must be saved in a database for re-using.

3. Update Orders and Export Data:

Update the orders table with retrieved location details.

Generate an export file containing order\_number, city, state, and zip.

4. Prevent Reprocessing:

Most of the customers place multiple orders. So, the IP address remains the same. Implement a mechanism to track and skip previously processed IP addresses, and use the information saved in the database instead.

5. Performance Optimization:

Ensure efficient data lookup and processing for large datasets (100,000+ records).

Optimize API usage to fetch location data for all IPs in under 5 minutes.

Part 2: Sales Report Generation

We need to generate a quarterly sales report by state and city for internal auditing.

Tasks:

1. Custom Report Generation:

Develop a function that accepts state and year as inputs.

Aggregate sales by state, city, and quarter.

Generate an Excel file containing the relevant sales data. (Use IL\_state\_sales\_report\_2021.xlsx as a reference. Output should be in the same format)

Orders without a city value can be ignored.

Deliverables

1. Python Script:

Well-documented code implementing all required functionalities (using pandas).

Clear function definitions with comments explaining the logic.

2. SQL Statements:

Scripts for creating and updating necessary database tables.

Queries for inserting, updating, and retrieving data.

3. Final Code Submission:

The complete project must be submitted via a Git repository.

The repository should include a README file explaining the setup, execution steps, and any dependencies.

Note:

While AI tools can be used for assistance, clear understanding of logic in the implementation must be demonstrated.

The approach, optimizations, and decisions should be well explained in the documentation or comments.