

Project name: "COVID-19 Data Integration, Analysis, and Visualization Platform"

To be able to work on project you must create free Snowflake trial account. It must use AWS cloud (Ohio region)

Leverage the COVID-19 dataset available on Snowflake to create an integrated data analytics and visualization platform that offers insights into the spread, effects, and patterns of the virus across different regions and populations. The platform should combine structured and semi-structured data sources, present an API for querying, and visualize results dynamically.

This project not only allows you to dive deep into Snowflake's capabilities but also pushes to integrate multiple technologies to produce a holistic, user-friendly analytical tool. It effectively assesses your skills gained during this Bootcamp in data engineering, data analysis and backend development.

For assignment completion please provide 1-2 page report on project implementation. Report must contain implementation for every task and insights on COVID-19 topic you were able to get based on project. Report format: PDF or MS Word. Please include student name.

Technologies Involved:

Snowflake (for structured data storage and SQL operations)

Python (for backend development, data manipulation, and API development)

Any NoSQL DB (like MongoDB for storing user preferences, comments, or supplementary semi-structured data)

Any visualization library in Python (Plotly, Matplotlib, or Dash for the frontend).

You are free to make any necessary choices as a developer.

Tasks:

Task 1. Use Snowflake Marketplace and get COVID-19 free dataset.

Setup Snowflake Resource monitors.

Task 2. Data Exploration and Enhancement.

Begin with analyzing the Snowflake COVID-19 dataset using SQL to understand its structure, patterns, and any gaps.

Use Python & Snowflake to augment this dataset. For instance, combine it with any other datasets (e.g., demographic or economic data from Kaggle) to provide richer insights.

Task 3. Data Modeling in NoSQL.

Design a schema in a NoSQL database (e.g., MongoDB) to store supplementary data, like user comments on data points, annotations, or additional sources not found in Snowflake. Include schema in the final report.

Task 4. API Development with Python:

Develop an API using a framework like Flask or FastAPI.

This API should be able to:

Query Snowflake for data based on user inputs.

Interact with the NoSQL database for relevant additional data or metadata.

Perform any on-the-fly data processing required.

Task 5. Interactive Visualization with Python:

Use libraries such as Dash or Plotly to create interactive web dashboards. Alternatively you can use Snowflake visualization techniques.

Visualize various metrics like infection rates, mortality rates, demographic breakdown, etc.

You are free to use any naming conventions and any dashboard and technology combinations.

Bonus: Allow users to add annotations or comments, which are then stored in the NoSQL DB.

Task 6. Analytical Features:

Time Series Forecasting: Use Python to predict future infection rates or other metrics.

Bonus: Clustering. Segment regions based on similarities in COVID-19 spread patterns or outcomes.

Task 7. Performance Optimization:

Use appropriate Snowflake features to ensure that the SQL queries on the COVID-19 dataset are optimized for performance.

Task 8. Implement caching in your API for frequently requested data.

Task 9. Pattern Recognition with MATCH_RECOGNIZE.

Identify patterns within a COVID dataset.

Task 10. Share all project structures with account Snowflake account RM76933 <https://iodzogj-rm76933.snowflakecomputing.com>. Account locator: HR82191

Evaluation Criteria:

Data Integration: Efficacy in merging Snowflake's dataset with other sources and the NoSQL database.

Analysis Depth: How profound and varied are the insights drawn from the data?

Visualization Quality: Intuitiveness, aesthetics, and interactivity of the visualization tool.

API Robustness: The API's efficiency, flexibility, and error-handling capabilities.

Code Quality: Code organization, clarity, and documentation.