

**Background:**

You are tasked with creating an ETL process in Python that extracts data from a common free API, transforms the data, and loads it into a local database. The API you will be working with is the Open Meteo API, which provides weather data for various locations around the world.

**Requirements:**

- **Extract (E):** Write Python code to fetch weather data from the Open Meteo API for three different cities of your choice. The forecast returned should contain hourly predictions for the following 10 days.

Example of API usage:

curl

["https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current\\_weather=true&hourly=temperature\\_2m,relativehumidity\\_2m,windspeed\\_10m"](https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current_weather=true&hourly=temperature_2m,relativehumidity_2m,windspeed_10m)

- **Transform (T):** Perform the following transformations on the data obtained from the API. Store only the following information for each city:
  - City Name
  - Temperature (in Celsius)
  - Temperature (in Fahrenheit)
  - Temperature (in Kelvin)
  - Humidity
  - Wind Speed (m/s)
  - Measure datetime
  - Round the temperature, humidity, and wind speed values to two decimal places.
  - Add a timestamp indicating when the data was collected.
- **Load (L):** Create a local SQLite database and store the transformed data in a table named **weather\_data**. Use Python and the SQLite library for this task. Each row in the table should represent the weather data for one city.
- **BigQuery Test:** Export this table and load it in BigQuery with the best practices (using any free GCP account). This process can be a python code (preferably and it will add points to the final score of the test) or a manually documented process. Perform a query that retrieves the mean, minimum and maximum of temperature per day per city for the following 10 days. Store the query as a SQL file in the repository.
- **Data Engineering Concepts (DE):** In the comments of your code, explain the following data engineering concepts that you applied in your solution:
  - Data extraction method used.
  - Data transformation techniques applied.
  - Data persistence strategy employed.
  - Any error handling or data validation you implemented.

**Submission:**

- Provide a Python script that demonstrates your ETL process. You can use comments to explain your code and data engineering concepts.

- Include a README file with instructions on how to run your script and any necessary dependencies.
- Submit your code and README in a compressed (zip) file.

**Evaluation Criteria:**

Your solution will be evaluated based on the following criteria:

- Correctness of data extraction, transformation, and loading.
- Proper use of Python and relevant libraries (e.g., requests, SQLite).
- Clarity and correctness of comments explaining data engineering concepts.
- Error handling and data validation.
- Organized and well-documented code.
- Adherence to best practices in data engineering.

**Test duration: 3 days**