

## **Assessment test: Data Engineering**

This test is intended to verify basic skills in Data Engineering (data extraction, modeling, critical reasoning) for candidates to the role of Data Engineer in ShopFully.

The deadline for returning the output is specified to the candidate when sharing the test. The results will then be used as a topic for discussion in the subsequent interview.

The candidate is left free to decide the preferred tools with which to carry out the test, it is sufficient that all the results and the coding are then shared through a text document for further discussion.

### **Project description**

For the ShopFully team it is very important to understand the weather conditions for certain locations.

Use the [OpenWeatherMap API](#) (use a free tier subscription) to get the current weather conditions for 3 cities where ShopFully has offices: Milano, Bologna, Cagliari.

The task has two main parts, one focuses on the modeling while the other one is centered on script writing.

### **Part 1 - data modeling**

- Look at the data structure provided by the API documentation.
- Decide which data could be considered important and bring value and discard the data which looks less relevant.
- The data granularity should be 1-hour (we want to have hourly temperature to be able to analyze historical data in the future).
- Create a logical and physical model for this data having the following questions in mind:
  - How many distinct weather conditions were observed (rain/snow/clear/...) in a certain period?
  - Rank the most common weather conditions in a certain period of time per city?
  - What are the temperature averages observed in a certain period per city?
  - What city had the highest absolute temperature in a certain period

of time?



- Which city had the highest daily temperature variation in a certain period of time?
- What city had the strongest wind in a certain period of time?

Deliverable:

- visualized logical schema,
- complete DDL for physical database implementation,
- SQL queries.

## **Part 2 - script writing**

- Automate the data download process.
- Store the raw (response) data in the format you find the most suitable. •
- Identify the information you find useful and create a dataframe with it. •
- Write the data into the table(s) you identified in the modeling process. •
- Write the data to a relational database. You have the freedom to decide how to organize your data/relationships, data types, primary, foreign keys, indices, etc.
  - Could you answer the questions from the previous section using aggregations in python applied on the denormalized dataframe?

## **Note**

- If you are not able to provide a script, you can concentrate on the modeling part of the data by looking at the expected API output/response in the documentation.
- Organize the project taking into consideration that data needs could grow and involve:
  - All Italian municipalities and beyond.
  - Different ways to get the information and consequently different API calls (always the same endpoint).