

Cas Offline

Contexte

Ton entreprise a besoin de créer un pipeline de données pour extraire, transformer et charger des données provenant d'une API externe dans BigQuery, afin de permettre des analyses futures. Ici, l'objectif est de mesurer tes compétences en SQL, en Python et en extraction de données via une API.

Partie 1: Extraction des données via une API externe

Tâche:

- Utilise l'API publique "[OpenWeatherMap](#)" pour récupérer les données météo d'une ville (par exemple, Paris) sur les 7 derniers jours.
- Documente et explique le processus de récupération des données. Décrit également comment tu gères les erreurs ou les limitations d'appel de l'API.

Données attendues:

- Les informations météo pour chaque jour doivent inclure : la date, la température moyenne, la température minimale, la température maximale, l'humidité, et une description générale du temps.

Livrable attendu:

- Un script Python qui récupère les données et les structure dans un format facile à exploiter (par exemple, un DataFrame Pandas ou un fichier CSV).

Partie 2: Transformation et chargement dans BigQuery

Tâche:

- À partir des données extraites via l'API, prépare un script SQL (BigQuery) pour créer une table qui stocke ces données.
- Le script doit inclure la création de la table (si elle n'existe pas déjà), et l'insertion des nouvelles données de manière optimale (en évitant les duplications).

Livrable attendu:

- Un script SQL qui crée la table et y insère les données.

Partie 3: Analyse des données dans BigQuery

Tâche:

- Rédige une requête SQL pour répondre à la question suivante : "Quel a été le jour le plus chaud et le jour le plus froid au cours des 7 derniers jours à Paris?"
- La requête doit retourner la date, la température maximale et la température minimale pour ces deux jours.

Livrable attendu:

- Une requête SQL commentée qui produit les résultats attendus.

Instructions supplémentaires

1. **Documentation** : Tu dois fournir une brève documentation qui:
 - a. explique l'approche utilisée pour chaque partie du cas
 - b. explique comment n'importe qui peut exécuter le code fourni depuis son environnement
 - c. Nice to have: Une image Docker basée sur Python pour exécuter le script Python
2. **Critères d'évaluation** :
 - **SQL** : Clarté et efficacité des requêtes SQL & gestion des duplications
 - **Python** : Qualité du code, gestion des erreurs, structure du code
 - **API** : Compréhension de l'intégration API, robustesse du code face aux limitations d'API (par exemple, limites de taux d'appel, erreurs de connexion).