

## BRIEF PROJET

### *Analyser les ventes d'une PME*

Projet pour la préparation à la journée de sélection pour le parcours de formation Data Engineer.

## Objectifs visés

- Créer et mettre en œuvre un environnement à deux services : un pour l'exécution des scripts (import de données) et un autre pour la base de données;
- Analyser un jeu de données et en expliquer ses caractéristiques;
- Créer une base de données adaptée pour le stockage du jeu de données;
- Importer les données;
- Réaliser un premier niveau d'analyses de données avec SQL;
- Stocker les résultats des analyses.

## Durée estimée

1 à 2 jours

**rappel** : Si vous êtes bloqué/bloquée à un moment donné dans votre travail, cela n'est pas excluant et n'est pas un motif de rejet de votre candidature.

Vous pouvez aussi demander de l'aide autour de vous (sans faire faire à votre place bien entendu).

## Énoncé du travail à réaliser

### Scénario de départ pour le travail à réaliser

Imaginons que vous êtes administrateur/administratrice système et que pour un nouveau projet vous assistez un·e data engineer, au sein d'une PME de services numériques.

Ce projet concerne un client qui souhaite pouvoir analyser la dynamique de ses ventes dans le temps et dans l'espace afin d'améliorer leur prise de décision stratégique. Pour cela, le client vous a fourni [un extrait de leur jeu de données](#) des ventes sur 30 jours et des données relatives à ses [produits](#) et à ses [magasins](#) distribués dans plusieurs villes de France.

Dans un premier temps, votre responsable data engineer vous demande de :

1. concevoir une architecture en deux services :
  - un premier service pour l'exécution des scripts dans le langage de programmation de votre choix.
  - un deuxième service pour le stockage des données en base de données avec SQLite.
2. implémenter l'architecture conçue :
  - créez les fichiers Dockerfile et Docker Compose
  - assurez la bonne exécution de l'architecture

Puis, vous devrez prendre connaissance des trois fichiers de données partagés par le client et en

comprendre les principales caractéristiques pour modéliser les données en tables et relations. Une fois le schéma de la base de données créé vous devrez :

1. créer la base de données SQLite;
2. effectuer une première analyse des ventes avec SQL;
3. stocker les résultats des analyses.

Comme le fichier de données sur les ventes s'actualise en temps réel, vous devrez vous assurer que les lignes de données à importer ne sont pas déjà stockées en base de données.

## Proposition d'organisation de votre travail

### Etape 1 : Concevoir l'architecture

Créer sous forme de schéma une architecture qui identifie :

1. Le service d'exécution des scripts et le service de stockage de données avec :
  - a. le nom du service
  - b. l'objectif du service
2. les entrées et sorties de communication entre les services indiquant
  - a. les ports exposés
  - b. le sens de la communication

### Etape 2 : Réaliser l'architecture

1. Créez le Dockerfile pour créer l'image du service qui exécutera les scripts. Prenez en compte ce qui suit :
  - a. trouvez une image de base avec l'environnement d'exécution que vous avez choisi. Par exemple, si vous décidez d'écrire les scripts en Python, votre image de base pourrait être celle-ci : [python - Official Image | Docker Hub](#)
  - b. créez un script "hello-world" dans le langage de programmation de votre choix et tester sa bonne exécution
2. Créer le fichier Docker compose qui orchestre le lancement des deux services. Prenez en compte ce qui suit :
  - a. Pour lancer le service qui stocke les données, utilisez une image Docker avec SQLite3 (vous pouvez en trouver une sur Docker Hub) et assurez-vous qu'elle fonctionne lorsque vous la lancez localement (par exemple, vous pouvez accéder à la ligne de commande SQLite dans le conteneur).
  - b. Assurez-vous que le service qui exécute les scripts s'exécute après le démarrage du service qui stocke les données.

## Etape 3 : Réaliser une analyse préalable des données

1. Créez le schéma de la base de données :
  - a. les tables définies par un nom de table, les attributs de la table et la clé primaire sous forme de diagramme
  - b. les relations entre les tables inclus sur le diagramme
  - c. n'oubliez pas de créer les tables nécessaires pour pouvoir ajouter les résultats des analyses à la base de données.
2. Intégrez dans le script "hello-world" (vous pouvez le renommer si vous voulez) la logique pour créer les éléments suivantes :
  - a. La base de données
  - b. Les tables
3. Intégrez dans le script (ou en créer un autre) la logique pour importer les données partagées par le client dans les tables de la base de données :
  - a. Collecter les fichiers de données directement à partir des URLs partagées par le client. Vous devrez effectuer les requêtes HTTP.
  - b. Pour les données de vente, créez une condition dans le script pour n'importer que les nouvelles données qui ne sont pas déjà dans la base de données.
4. Créez un script pour l'exécution de requêtes SQL, pour répondre aux questions clés sur les ventes de l'entreprise :
  - a. une requête pour obtenir le chiffre d'affaires total,
  - b. une requête pour obtenir les ventes par produit,
  - c. une requête pour obtenir les ventes par région.

## Ressources associées

- Conteneurisation
  - [Docker : comprendre l'essentiel en 7 minutes](#)
  - [La conteneurisation, qu'est-ce que c'est ?](#)
  - [Qu'est-ce que la conteneurisation ? | IBM](#)
  - [Overview | Docker Documentation](#)
  - [Docker Hub](#)
  - [Dockerfile: the building block of the Docker ecosystem - IONOS](#)
- Web / http
  - [Les requêtes HTTP expliquées simplement](#)
  - [Exemple de requêtes HTTP avec Python](#)
- SQL

- <https://sql.sh/>
- Les données client
  - [CSV produits](#)
  - [CSV magasins](#)
  - [CSV ventes](#)

## Livrables

- Le schéma de l'architecture conçu
- Le schéma des données (sous une forme standard, MCD par exemple)
- Le Dockerfile
- Le fichier yaml du Docker Compose
- Le(s) script(s) d'exécution pour la collecte, transformation, et import des données
- Le fichier sql
- Une note rappelant les résultats d'analyse obtenus (point 4.a, 4.b, 4.c).

**NB** : dans le cas où votre candidature serait retenue, vous devrez apporter vos livrables le jour des entretiens de sélection, à l'aide du support de votre choix : usb / url en ligne (git, cloud...) / pc portable / ...;