

Objectif

Mettre en valeur les aptitudes en programmation, la compréhension des bases de données et du traitement des données.

S'assurer que le code est propre et exécutable. Qu'il y a une gestion d'erreurs.

S'assurer que le livrable contient tous les éléments : SQL, code, commandes exécutées...

Prérequis

Connaissance de MySQL, Python, Pour la partie 1.

Connaissance de Docker et Docker compose pour la partie 2.

Besoin de Docker et de Docker Compose installés pour la partie 2.

Le test inclus des données d'exemple et un code de programme. Le schéma d'exemple crée une table, avec un code d'exemple en Python, pour charger des données à partir d'un fichier CSV et les exporter dans un fichier JSON.

Il y a des instructions dans ce document expliquant comment utiliser les conteneurs Docker, démarrer la base de données et utiliser le script python.

Contenu de l'archive

- **Le présent README.**
- **Un fichier docker-compose.yml** qui configure un conteneur pour la base de données MySQL et un conteneur d'environnement python.
- **Un dossier image-python** contenant un exemple de script pour accéder à la base de données.
- **Un fichier example_schema.sql** contenant un schéma de table utilisé par le script d'exemple.
- **Un dossier data** contenant quatre fichiers :
 - example.csv : Un petit ensemble de données utilisé par le script d'exemple.
 - places.csv : Chaque ligne contient le nom d'une ville, d'un comté et d'un pays.
 - people.csv : Chaque ligne contient un prénom / nom / date_naissance / ville.
 - sample_output.json : Fichier de sortie d'exemple.

Problème

PARTIE 1 OBLIGATOIRE :

1. Concevoir le schéma de base de données pour contenir les données des fichiers CSV people et places.
2. Adapter le fichier example.py pour lire les fichiers CSV people.csv et places.csv, et permettant d'écrire un fichier JSON contenant la liste des pays, avec le nombre de naissances pour chacun. Le fichier data/sample_output.json donne un aperçu de contenu attendu.

PARTIE 2 FACULTATIVE : Stocker l'ordre des commandes dans un fichier

1. Exécuter le fichier SQL pour construire la base de données MySQL, via un script, via le client de ligne de commande MySQL ou via un client GUI.
2. Créer une image Docker pour charger les fichiers CSV places.csv et people.csv dans les tables de base de données créées. La configuration doit apparaître dans le fichier docker-compose.
3. Créer une image Docker pour produire un résumé en langage python du contenu de la base de données. La sortie doit être au format JSON et être écrite dans un fichier du dossier data appelé data/summary_output.json. Il doit contenir la liste des pays, avec le nombre de naissances pour chacun. Le fichier data/sample_output.json donne un aperçu de contenu attendu.

Aide DOCKER

Construction des images

Cela construira toutes les images référencées dans le fichier Docker Compose. Vous devrez le relancer après avoir apporté des modifications au code. (Vous pouvez également spécifier des services individuels à construire si cela est plus pratique.)

```
docker compose build
```

Démarrage de MySQL

Cela prendra un court instant pour exécuter les scripts de démarrage de la base de données.

```
docker compose up database
```

Facultatif : si vous souhaitez vous connecter à la base de données MySQL via le client en ligne de commande. Cela peut être utile pour consulter le schéma ou les données de la base de données.

```
docker compose run database mysql --host=database --user=user_dteng --  
password=cirilgroupt db_dteng
```

Scripts d'exemple

Nous avons fourni du code d'exemple écrit en Python et en R, qui montre comment utiliser un programme dans un conteneur Docker séparé pour se connecter à la base de données, en utilisant une bibliothèque ORM si nécessaire, pour charger des données à partir d'un fichier CSV et interroger les données pour les exporter sous forme de fichier JSON. Celles-ci doivent être considérées comme illustratives; il est possible d'utiliser l'un de ces exemples comme base pour votre propre solution, mais nous préférons que vous utilisiez les technologies avec lesquelles vous vous sentez à l'aise.

Assurez-vous que la base de données MySQL fonctionne, puis chargez le schéma d'exemple avec :

```
docker compose run --no-TTY database mysql --host=database --user=user_dteng --  
password=cirilgroupt db_dteng <example_schema.sql
```

Assurez-vous ensuite que les conteneurs ont été construits avec docker compose build et exécutez un ou plusieurs des programmes d'exemple avec :

```
docker compose run example-python
```

```
docker compose run example-r
```

Dans chaque cas, le programme charge les données du fichier data/example.csv dans cette table et exporte les données de la table de la base de données sous forme de fichier JSON dans le dossier data. Notez que les scripts ne tronquent pas la table, donc chaque fois que vous en exécutez un, du contenu supplémentaire sera ajouté.

Nettoyage

Pour nettoyer en arrêtant tous les conteneurs et en les supprimant.

```
docker compose down
```