

Rapports individuels évolutifs sur le travail réalisé.

Période de travail : 07/11/2022 => 05/12/2022

Manager de la période : Claude

Collaborateurs : Lucas, Quentin et Samuel

Claude :

Les tâches :

- Modification des scripts SQL et ajout de script SQL
- Python :
 - Améliorer le code du script Python qui rajoute les accolades
 - Connexion à la Base de données
 - Exécution de script SQL en Python
 - Indicateurs de performances

Ressenti :

Ayant réussi à améliorer le script Python rajoutant les accolades à certaines colonnes précises en ajoutant également les indicateurs de performances par la même occasion, je me suis vite aperçu qu'un problème survenait. L'ancien script SQL que nous avions n'était pas bon du tout par conséquent, j'ai dû modifier les contraintes ainsi que les types des tables. À la suite de cela, j'ai pu charger les données dans la base de données à travers la commande Copy from de PostgreSQL. Évidemment, si tout marchait du premier coup, cela ne serait pas drôle par conséquent, certaines tables ne parvenaient pas à y être insérées et cela est dû à la contrainte de la clé étrangère. En effet certaines données sont manquantes dans la quasi-totalité des tables par conséquent, il est quasiment impossible de pouvoir faire référence d'une table à une autre puisque si un identifiant n'est pas présent dans la table servant de référence cela viole la contrainte de clé étrangère.

Voici une illustration du problème l'identifiant sera ici Tconst :

Table test

Tconst	Ordering
Tt0001	1
Tt0002	2
Tt0004	3

Table test1

Tconst	Ordering
Tt001	1
Tt002	2
Tt003	3
Tt004	4

La table test fait référence à la table test1 à travers la colonne Tconst cependant comme vous pouvez le voir Tt0003 n'est pas présente dans la table test1 cela viole la contrainte de clé étrangère nous empêchant ainsi de « relier » les deux tables.

2 Choix s'offre à nous pour palier à ce problème, ajouter Tt003 dans la table test ou bien supprimer Tt003 dans la table test1. Comme je ne vais pas inventer des valeurs pour les différentes colonnes de la table, j'ai décidé d'avoir un Script SQL qui enlèvera les « mauvaises valeurs » de la table correspondante. Cependant, pour faire cela, je dois faire en sorte de ne pas prendre en compte la contrainte de clé étrangère. Pour cela, j'utilise une commande assez particulière de postgresSQL NOT VALID cette commande nous permet de ne pas valider la contrainte pour le moment cela règle le problème de la contrainte nous permettant ainsi de copier les données du tsv à la table correspondante. Nous validerons la contrainte lorsque les mauvaises données seront traitées à travers le script DeleteFrom. Pour le moment, il est impossible aux autres personnes du groupe d'avoir accès à la base de données, car pour cela, je dois ouvrir les ports de ma box ainsi que les pare-feu et cela reste plus ou moins compliqué d'autant plus que des modifications sont à faire sur les fichiers de configuration de PostgreSQL afin que toutes les adresses IP (0.0.0.0/0) ou bien ceux que je définisse puisse se connecter à ma base de données, par conséquent, nous allons sûrement utiliser AWS (Amazon) avec RDS (merci à Fatih pour l'information). Cependant, le temps de traitement prendrait beaucoup plus de temps que prévu (environ une dizaine d'heures pour une table).

Les différents scripts fonctionneront ainsi : un Script Bash ou Shell exécutant un fichier Python à des horaires périodiques (ce n'est pas demandé, mais nous essayons de voir pour cela.). Un script qui récupère les données tsv d'IMDB à travers l'URL, un script qui modifie les données pour rajouter les accolades et enfin un script qui exécutera les différents scripts SQL en se connectant au préalable à la base de données.

Lucas :

Les tâches :

Ressenti :

Quentin :

Les tâches :

Ressenti :

Samuel :

Les tâches :

Ressenti :