Collège de Bois-de-Boulogne | FORMATION CONTINUE

Projet de session

420-BD4-BB
Acquisition, Ingestion et gestion des données

Samir LAKEHAL
(1895291)

TP1: Capture et nettoyage de données

Juin 2018

Introduction

L'objectif de ce travail est d'apprendre, en utilisant le langage Python, comment extraire des données de sources différentes en termes de structure et de type de base de données, et de charger ces données dans une table destination en prenant soin de les transformer avant (nettoyer). Il s'agit donc de créer et exécuter un processus ETL avec Python, et ainsi confirmer la puissance de ce langage.

Caractéristiques du processus

Le processus consiste à lire successivement trois fichiers de données clients, fichiers de types différents (Mysql, CSV et Json), et pour chaque fichier, effectuer éventuellement des transformations sur les données et ensuite les charger dans une table destination (j'ai choisi Mysql) comme le montre le schéma ci-dessous :

Le contenu des fichiers sources de données est préalablement analysé pour décider quelles transformations pourraient être appliquées aux données d'une part, et de la structure du fichier destination d'autre part.

Après analyse des trois tables, j'ai décidé :

- d'uniformiser le genre avec les seules valeurs possibles M ou F (transformer Male et Female)
- si le genre est manquant, mettre 'unknown' et garder l'enregistrement, le client décidera ultérieurement s'il veut corriger ou supprimer ces enregistrements.
- si la ville est manquante, mettre 'unknown' et garder l'enregistrement, le client décidera ultérieurement s'il veut corriger ou supprimer ces enregistrements.
- d'ajouter une clé primaire que je génère séquentiellement lors du chargement, en gardant le ID original, le client décidera ultérieurement s'il veut corriger ou supprimer la colonne (cl_Orig_Id).

Étapes principales de mise en place du processus

- Analyser les structures et données des trois tables sources pour déterminer quelle sera la structure de la table destination ainsi que les transformations à faire aux données sources avant de les charger
- Exécuter le script SQL de création de la table client_data (sous mySQL) en prenant soin d'ajouter la commande de création de la base de données source.
- Création du projet python

- Attacher les autres fichiers sources de données (csv et json files) au projet python

- Une fois la structure de table destination définie, écrire et exécuter le script SQL de création de cette base/table (sous mySQL).

- Écrire le code ETL qui devra pour chacune des trois tables en input extraire les données, effectuer les transformations nécessaires et insérer ces données dans la table destination.

- Effectuer les tests nécessaires pour valider le traitement

Comment tester l'application

Au moyen de quelques requêtes SQL pour vérifier :

- si le nombre d'enregistrements du fichier en sortie correspond au total des enregistrements en input.

  ```
  select count(*) as "Nombre de clients" from clients
  ```

- si le total des enregistrements suite aux transformations du 'genre' correspond au nombre total d'enregistrements du fichier en sortie :

  ```
  select count(*) as Count, "M" as Gender from clients where cl_gender = 'M' union
  select count(*), "F" from clients where cl_gender = 'F' union
  select count(*), "unknown" from clients where cl_gender = 'unknown' union
  select count(*), "Total" from clients
  ```

- si le total des enregistrements suite aux transformations du 'state' correspond au nombre total d'enregistrements du fichier en sortie :

  ```
  select count(*) as Count, "valid" as State from clients where cl_state <> 'unknown' union
  select count(*), "unknown" from clients where cl_state = 'unknown' union
  select count(*), "Total" from clients
  ```

- l'unicité des clés en tentant un Insert d'une clé existante : (on peut utiliser également l'onglet 'Insérer' de phpMyAdmin)

  ```
  insert into clients (cl_key, cl_original_id, cl_first_name, cl_last_name, cl_email, cl_gender, cl_state) values
  (1,1, 'Al', 'Pacino', 'M', 'alpacino@gmail.com', 'New York');
  ```

Conclusion

L'objectif de ce travail a été atteint puisqu'on a pu, au moyen du langage Python, exécuter un processus ETL, qui a consisté à extraire des données de tables de différents types, effectuer des transformations de ces données et finalement les charger dans une table destination, et ensuite valider les résultats de requêtes SQL.

Pour tout ce processus ETL, on a eu besoin de tout au plus une soixantaine de lignes de code Python, confirmant ainsi toute la puissance de ce langage.