BD-4 Extraction, transformation et gestion de données

Liudmila Aleksandrova (1895677)

TP1 : Mise en place d'un processus ETL avec python

1. Introduction

L'objectif du TP est la mise en place le processus de capture et de transformation de données de source différents.

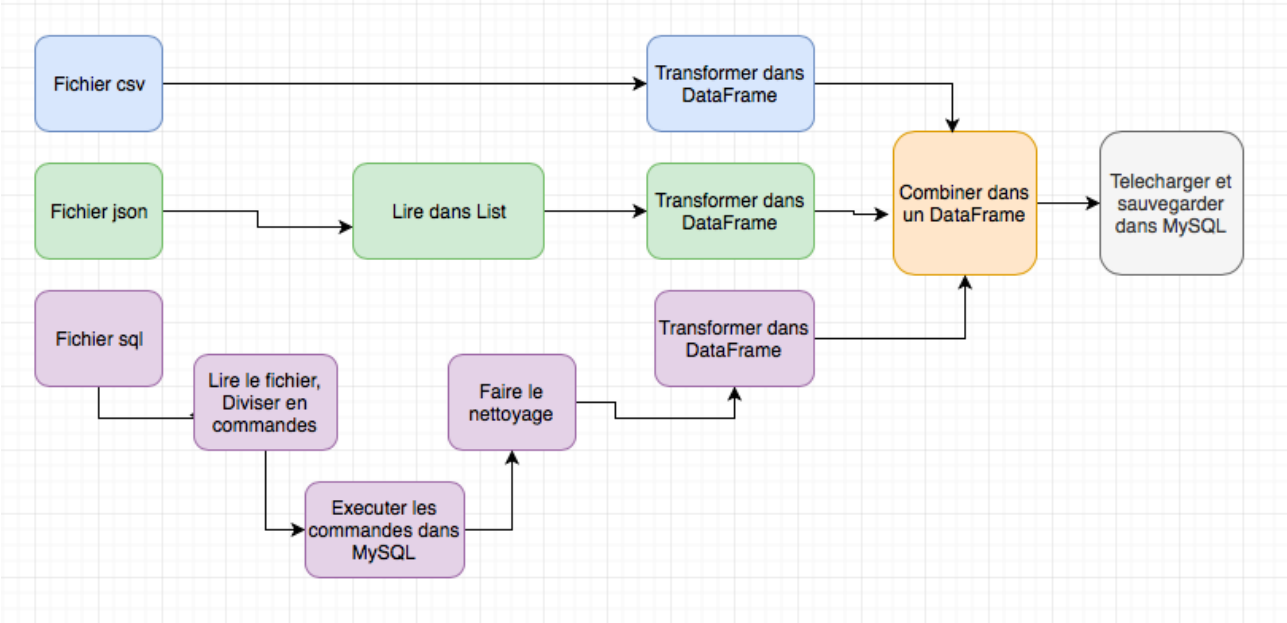
Pour ce TP on a les fichiers de source csv, json et sql. La destination est une table MySQL.

2. Caractéristiques de processus

- Le langage de programmation – Python 3
- Le module pour travailler avec les données de source json – json
- Les modules pour travailler avec les données de source sql – MySQLdb et sqlalchemy
- Le module pour travailler avec les Data Frames – pandas

3. Étapes principales de mise en place du processus

1. Déterminer un workflow du processus



2. Importer les modules nécessaires

```
import json
import pandas as pd
import MySQLdb
from sqlalchemy import create_engine
```

3. Créer la connexion avec base de données MySQL

```
engine = create_engine("mysql://root:qazwsxedc@localhost/tp1")
connection = engine.connect()
```

4. Charger les données de sources (json, sql, csv) dans Data Frames et faire les transformations

```
def load_json_data(file_name):
    with open(file_name, 'r') as file_json:
        data_json = json.load(file_json)
    d_json = pd.DataFrame.from_dict(data_json)
    d_json = d_json[['id', 'first_name', 'last_name', 'email', 'gender', 'ville']]
    file_json.close()
    return d_json

def load_sql_data(file_name):
    file_sql = open(file_name)
    cmds = file_sql.read().split(';')
    i = 0
   while i < len(cmds) - 1:
        connection.execute(cmds[i])
        i += 1
    formattedGenderQuery = '''SELECT id, first_name, last_name, email,
                                     CASE WHEN gender = 'F' THEN 'Female'
                                          WHEN gender = 'M' THEN 'Male'
                                     END AS gender, ville
                              FROM client_DATA'''
    data_sql = pd.read_sql(formattedGenderQuery, connection)
    file_sql.close()
    return data_sql

df_csv = pd.read_csv('week_cust.csv')
df_json = load_json_data('cust_data.json')
df_sql = load_sql_data('client_DATA.sql')
```

5. Faire concaténation des 3 Data Frames

```
df_all = pd.concat([df_json, df_csv, df_sql], ignore_index = True)
```

6. Charger le Data Frame final dans une table MySQL

```
df_all.to_sql(name = 'all_DATA', con = connection, if_exists = 'fail', index = False)
```

4. Test l'application

- Le code de programmation : script.py
- Les fichiers de données de sources différents : 'client_DATA.sql ', 'cust_data.json', 'week_cust.csv'
- La destination – une table MySQL
- Voire les détails sur vidéo – Test_processusETL_python_LiudmilaAleksandrova.

5. Conclusion

Mise en place le processus de capture et de transformation de données est facile avec le langage Python et ses modules.