Projet Hadoop Spark

Etape 01 : Git

Créer un nouveau projet sur github :

name : HadoopSparkProjetPMN

description : yourName

allez dans le workspace depuis le terminal :

```
git init
echo "# HadoopSparkProjetPMN" >> README.md
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/[yourId]/HadoopSparkProjetPMN.git
git push -u origin master
```

Etape 02 : Maven & architecture

- créer le projet maven HadoopSparkProjetPMN et ajouter les principales dépendances (scala & spark)

- Ajouter dans src/scala

package utils

Utils.scala

UserDefinedFunction.scala

package common

ApplicationProperties.scala

Constants.scala

package main

DataframeResult.scala

Main.scala

Etape 03 : Bash

créer les scripts bash suivants **POUR LA MANIPULATION HDFS** :

- *verify_input.sh* : un script pour vérifier si le dépot input est vide ou pas
- *copie_data_input.sh* :un script pour copier les fichiers data dans le dépot input
- *copie_data_tmp.sh* : un script pour créer un dossier data_tmp sur lequel on va copier les fichiers qui sont présents dans l'input.
- *delete_tmp.sh :* un script pour supprimer le dossier data_tmp et son contenu
- *delete_content_input.sh* : un script pour supprimer le contenu du dépôt input
- *run_spark.sh* : un script pour lancer l’appication spark

Etape 04 : Développement

N:B :

- *bien inspecter les données avant de commencer les requêtes*
- *afin de tester les requêtes : mettez les fichiers csv dans le dossier ressources*
- *ne pas oublier de créer le dossier input dans hdfs*
- *ne pas oublier le fichier application.properties*
- *pour chaque requête ⇒ 1 fonction ⇒ une sauvegarde (csv et/ou parquet)*
- *BONUS : tester les fonctions ;)*
- *pour les ajouts des columns,on peut avoir juste un dataframe final comme sauvegarde*

Bonne pratique :

- utiliser les persists tant que c'est nécessaire
- ne pas écrire les String en dure (les récupérer depuis constants)
- bien découper les fonctions

dans l'object **Utils.scala** :

- fonction pour créer un spark session
- fonction pour lire un fichier csv
- fonction pour écrire un fichier csv
- fonction pour écrire un fichier parquet

dans l'object **UserDefinedFunction.scala** :

- les fonctions pour les udfs que vous allez (éventuellement) crées

dans l'object **ApplicationProperties.scala** :

- la déclaration des valeurs path de l'input et des output

dans l'objet **Constants.scala** :

- les chaînes de caractère que vous allez utiliser dans votre projet

dans l’objet **Main.scala**

- l’appel à la spak session
- la lecture des fichiers csv
- l’appel aux fonctions de création de dataframe (résultat)
- l'écriture des fichiers csv + parquet (des résultats)

dans l’objet **DataframeResult.scala** :

création de fonctions pour les requêtes suivantes :

- ajouter un column date qui prend la column time_ref : de 202206 en 01/06/2022
- Ajouter une column year qui prend comme valeur l’année depuis la date
- Ajouter la column nom_Pays qui associe le code pays a son nom
- Ajouter une column détails service (filtrage sur service)
- Ajouter une column détails Good (filtrage sur goods)
- classement des pays exportateurs (par goods et par services)
- classement des pays importateurs (par goods et par services)
- regroupement par good
- regroupement par service
- la liste des services exporté de la france
- la listes des goods importés de la france
- classement des services les moins demandés
- classement des goods les plus demandé
- Ajouter la column status_import_export : si import > export : négative, Sinon positive (par pays)

- Ajouter la column difference_import_export : qui va calculer les exports - imports (par pays)
- Ajouter la column Somme_good qui va calculer la somme des goods par pays
- Ajouter la column somme_service qui va Calculer la somme des services par pays
- Ajouter la column pourcentages_good qui va Calculer le pourcentage de la column good par rapport à tous les good d'un seul pays (regroupement par import et export)
- Même choses mais Pour les services
- regrouper les goods selon leur type (Code HS2)
- classement des pays exportateur de pétrole
- classement des pays importateur de viandes
- classement des pays qui ont le plus de demandes sur les services informatique (computer services, computer software et other computer services)
- (bonus) ajouter une column description : qui prend comme valeur : “le pays XXXX fait un [IMPORT ou EXPORT] sur [goods ou Services]”

Etape 05 : script final

dans un script bash ajoutez les actions suivantes dans cette ordre :

- verification_input
- copie dans input
- copie dans tmp
- application spark
- suppression dossier temporaire
- suppression contenu input

Etape 06 : Push git

Dans le dossier du projet :

- Ajouter un .gitignore pour ne pas envoyer les fichiers/dossier non nécessaire
- Ajouter un dossier bash ou vous allez mettre les scripts (étape 03)
- Ajouter le fichier application.properties
- Bonus : ajouter dans le readMe.md les étapes pour le lancement du projet