TP2 utilisation de hive

Préambule:

Je vous conseille d'utiliser Vim pour modifier les fichiers sur la VM.
Pour ceux n'ayant pas l'habitude d'un éditeur de texte sans interface graphiques, vous pouvez installer gnu nano.

Après vous être connectés en ssh:

sudo yum update
sudo yum install nano

Vous pourrez ouvrir et modifier des fichiers avec nano <nom_fichier>. ctrl o pour enregistrer, ctrl x pour quitter.

Vous pouvez toujours utilisez un éditeur graphique type Kate et uploader vos fichiers sur la VM avec FileZila, mais ce sera bien plus pratique de faire les modifications sur la VM directement:

Pour un éditeur graphique, travaillez en local avec Kate (applications → accessoires → Kate) et uploadez le code sur la Sandbox avec FileZilla (protocole sftp , host localhost, port 2222, user/mdp raj_ops)

Nano:
https://www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/

Vim:
https://devhints.io/vim

Fichiers nécessaires au TP:

https://s3.eu-west-3.amazonaws.com/hadoop-telecom/TP2/create_table_iris.hql
https://s3.eu-west-3.amazonaws.com/hadoop-telecom/TP2/create_table_population.hql
https://s3.eu-west-3.amazonaws.com/hadoop-telecom/TP2/iris.csv
https://s3.eu-west-3.amazonaws.com/hadoop-telecom/TP2/population_us.json

Exercice 1 :

Objectif de l'exercice :
Utiliser Hive et s'approprier le cli pour lancer des requêtes.
Vous pouvez également utiliser Beeline si vous le souhaitez (beeline -u jdbc:hive2://localhost:10000/ -n raj_ops -p raj_ops)
Tâches à réaliser :

·
· Lancer hive depuis votre session ssh avec la commande hive.
· Créer votre database pour le tp2.
  - *create database tp2;*
·
· Quitter votre session hive exit ;

Visualisez le contenu du fichier *create_table_iris.hql*. (cat *create_table_iris.hql*)

· Creez la table à partir du fichier avec *hive –f create_table_iris.hql*

Uploadez iris.csv dans un dossier /user/raj_ops/TP2/iris sur HDFS en utilisant ambari files view ou la CLI (en fonction duquel fonctionne)

localhost:8080

· Relancez hive

Executez *use tp2;* Pour vous placer dans la bonne database.

· Visualiser les résultats avec SELECT * FROM IRIS limit 10

Essayez d'utiliser maintenant l'interface Hive View 2.0 d'ambari si elle fonctionne.

· Afficher le top dix des fleurs avec les petales les plus long avec ORDER BY et LIMIT 10
· Idem mais parmi les fleurs 'virginica' et 'versicolor'

Pour vous aider :
https://cwiki.apache.org/confluence/display/Hive/LanguageManual

Exercice 2 :

Objectif de l'exercice :
Utiliser unix et Hive pour faire quelques transformations de données.

· Chargez le fichier population_us.json dans votre répertoire TP2 sur la VM (pas HDFS).
· Le fichier est un tableau d'enregistrements json. Pour manipuler plus facilement les données sur Hive, on va le transformer en un enregistrement par ligne
  - cat population_us.json | tr -d "\n" | tr -d "]" | tr -d "[" | sed 's/},{/}\n{/g'> population_us_flat.json
  - tr supprime des caractères, sed les remplace. Faites *tr --help* ou *sed --help* pour en savoir plus.

- Créez une table hive externe dans la database tp2 ayant pour source de données sur hdfs /user/raj_ops/TP2/population

solution en fin de TP

  - Cette table ne doit avoir qu'une seule colonne json_data, de type string. uploadez votre fichier population_us_flat.json sur HDFS dans /user/raj_ops/TP2/population
- Visualisez le contenu de la table.
- Utilisez la fonction get_json_object de hive pour extraire des données du json
  - https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-get_json_object
  - get_json_object(population.json_data, '$') pour le document json entier
  - get_json_object(population.json_data, '$.females') pour selectionner le champs females.
- Grace à cette fonction, affichez le nombre d'hommes et de femmes par age.

Solution en fin de TP

- Créez une table externe sur le même format qu'iris (une colonne par champ json) afin de créer un fichier au format csv, avec homme, femme, age, année, pays et total (avec les bons types) dans un dossier TP2/population_tab sur hdfs.
  La commande de création de cette table Hive doit générer la création d'un fichier HDFS de type csv (Utilisez FIELDS TERMINATED BY ...).
- Utilisez INSERT OVERWRITE TABLE pour inserer les données de *population* vers *population_tab*.
  - https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML#LanguageManualDML-InsertingdataintoHiveTablesfromqueries
  - Attention, l'ordre des colonne dans votre select doit correspondre a celui defini lors de la création de la table cible.
- Observez sur HDFS le fichier créé dans TP2/population_tab.

Solution en fin de TP

Vérifiez le fichier créé:
hadoop fs -cat TP2/population_tab

Aller plus loin:
Afin de créer une table Hive avec une colonne par champ json, Hive propose d'utiliser un SERDE json déjà développé:
https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL#LanguageManualDDL-RowFormats&SerDe

Solution en fin de TP

Exercice 3 :

Récupérez cette les des élus municipaux en opendata sur la VM dans TP2:

wget https://s3.eu-west-3.amazonaws.com/hadoop-telecom/TP2/elus_mun2014.zip

Dézippez le avec unzip
Utilisez head pour voir le header (premiere ligne contenant les noms de colonne)
Creez une table externe dans hive vers /user/raj_ops/TP2/elus_mun avec toute les colonnes.

Afficher les 10 premières lignes
Compter le nombre d'ingénieur
Afficher le top 5 des prénoms les plus utilisés parmis les femmes et le nombre d'occurrence associé

Bonus :

- Récupérer le fsimage et déserialisez-le en format csv
  https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsImageViewer.html#Delimited_Processor
  Faire un sudo hdfs pour récupérer le fsimage et utiliser les commandes dfsadmin !
- Créez une table Hive externe sur ce fichier.
- Créez une vue avec un nouveau champs (file_type) qui comporte 2 valeurs possibles: 'file' ou 'dir' (utilisez le champs permission avec la fonction SUBSTR() et l'opérateur CASE)
- Affichez les 20 plus petits fichiers de taille supérieure à 0

Si le fsimage est de petite taille, une solution beaucoup plus rapide est d'utiliser AWK et hdfs dfs -ls -R, vous pouvez afficher le même résultat en une ligne de commande.
Par contre s'il de grande taille (un fsimage atteint rapidement les millions de lignes), autant utiliser les outils à disposition sur le cluster Hadoop.

Solutions:

Create avec Location:

```
create external table population (
json_data string
  )
location '/user/raj_ops/TP2/population';
```

JSON:

```
select get_json_object(population.json_data, '$.females') as femmes ,
    get_json_object(population.json_data, '$.males') as hommes ,
    get_json_object(population.json_data, '$.age') as age
    from json_table;
```

Créer le fichier au format CSV à partir de la table json_table:

```
create external table population_tab (
    femmes int
    ,hommes int
    ,age int
    ,annee int
    ,pays string
    ,total int
     )
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ","
    location '/user/raj_ops/TP2/population_tab';


INSERT OVERWRITE TABLE population_tab SELECT
    get_json_object(population.json_data, '$.females'),get_json_object(population.json_data,
    '$.males')
,get_json_object(population.json_data, '$.age'),get_json_object(population.json_data,
    '$.year'),get_json_object(population.json_data, '$.country')
,get_json_object(population.json_data, '$.total') from population;
```

Aller plus loin:

```
CREATE EXTERNAL TABLE population_serde (
 females int,
 country string,
 age int,
 males int,
 year int,
 total int)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
```

LOCATION '/user/raj_ops/TP2/population';

SELECT * FROM population_serde LIMIT 10;

Dans ce cas, il n’y a pas de fichier créé (utilisatton de la table externe à partir du fichier HDFS existant TP2/population/population_us_flat.json)