**Atelier 1 : Comprendre le fonctionnement des tables Hive**

Données utilisées :

* Hive/names.txt
* Hive/visits.txt
* Hive/wh\_visits.hive
  1. Créer le répertoire wh\_visits s’il n’existe pas dans :
* Hortonworks :/apps/hive/warehouse
* Cloudera :/user/hive/warehouse/

hadoop fs –mkdir /app/hive/warehouse/wh\_visits

hadoop fs –mkdir /user/hive/warehouse/wh\_visits

* 1. Copier le fichier visits du local dans hdfs dans le dossier wh\_visits

hadoop fs -put visits.txt /apps/hive/warehouse/wh\_visits

hadoop fs -put visits.txt /user/hive/warehouse/wh\_visits

1. Définition des tables HIVE
   1. Créer le script wh\_visits.hive contenant le script de :

Créer la table wh\_visits ayant les informations suivantes :

lname string

fname string

time\_of\_arrival string

appt\_scheduled\_time string

meeting\_location string

info\_comment string

Le séparateur est '\t' ;

**REMARQUE**:

Vous ne pouvez pas utiliser des colonnes **comment** ou **location** Ceux sont des mots clés réservés à Hive, donc si c’est le cas il faut les modifier.

2.2 lancer le script wh\_visits.hive

Remarque :

Pour lancer un script hive, vous n’avez pas besoin d’être dans le mode commande hive

Il faut lancer le script dans le dossier qui contient le script en local ou spécifier son chemin

Si le script s’est exécuté correctement, vous devriez voir affiché un OK sur le terminal d’exécution

hive -f wh\_visits.hive

beeline -u jdbc:hive2://localhost:10000/default -f wh\_visits.hive

3- Vérifier que la table a été bien créée

3.1 Accèder au mode commande hive ou beeline:

hive

beeline -u jdbc:hive2://localhost:10000/default

3.2 taper la commande vous permettant d’afficher toutes tables hive :

Vous devez voir wh\_visits dans la liste des tables :

hive> show tables;

3.3 Utilisez la commande describe pour afficher les détails de wh\_visits:

hive> describe wh\_visits;

OK

lname string

fname string

time\_of\_arrival string

appt\_scheduled\_time string

meeting\_location string

info\_comment string

Time taken: 0.785 seconds, Fetched: 6 row(s)

3.4 Lancer une requête pour vérifier l’état de la table en limitant à 20 lignes ( si vide ou non ) ?

Que constatez-vous ? expliquer pourquoi la table est renseignée ?

hive> select \* from wh\_visits limit 20;

3.5 Afficher les lignes ayant la colonne lname qui commence par OME et limiter 20 lignes.

select lname from wh\_visits where lname like 'OME%' limit 20;

4 compter le nombre de ligne de la table wh\_visits et comparer ce nombre au nombre de ligne du fichier visits.txt

hive> select count(\*) from wh\_visits;

MapReduce Total cumulative CPU time: 4 seconds 950 msec

Ended Job = job\_1481982442814\_0001

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.95 sec HDFS Read: 1142933 HDFS Write: 6 SUCCESS

Total MapReduce CPU Time Spent: 4 seconds 950 msec

**OK**

**21819**

Time taken: 62.878 seconds, Fetched: 1 row(s)

hive> [cloudera@quickstart ~]$

5- Sélectionner Input File Name

Hive possède deux colonnes virtuelles créées automatiquement pour chaque table:

INPUTNOM DE FICHIER et BLOC OFFSET INSIDE FILE. Vous pouvez utiliser ces noms de colonne dans  
Vos requêtes comme n'importe quelle autre colonne de la table.

Pour démontrer :

5.1 Exécutez les requêtes suivantes :

select INPUT\_\_FILE\_\_NAME, lname, fname FROM wh\_visits WHERE lname LIKE 'Y%';

5.2 A quoi correspond l’information issue du résultat de la requête ci-dessus ?

hdfs://quickstart.cloudera:8020/user/hive/warehouse/wh\_visits/visits.txt

6- Suppression de table

6.1 Pour commencer créer la table names utilisant les informations ci-dessous :

id int

name string

Le séparateur est '\t';

create table names (id int, name string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

6.2 Copier le fichier names dans le repertoire /user/hive/warehouse/names

hadoop fs -put names.txt /apps/hive/warehouse/names

hadoop fs -put names.txt /user/hive/warehouse/names

6.3 Vérifier le contenu du repertoire /user/hive/warehouse/names :

hadoop fs -ls /apps/hive/warehouse/names

hadoop fs -ls /user/hive/warehouse/names

6.4 Avec une requête HQL, afichier le contenu de la table names :

hive> select \* from names;

OK

0 Rich

1 Barry

2 George

3 Ulf

4 Danielle

5 Tom

6 manish

7 Brian

8 Mark

Time taken: 0.288 seconds, Fetched: 9 row(s)

hive>

6.5 Supprimer la table names :

hive> drop table names;

OK

Time taken: 2.519 seconds

hive>

6.6 Consulter à nouveau le dossier /user/hive/warehouse/names, que constatez-vous ?

hadoop fs -ls /user/hive/warehouse/names

ls: `/user/hive/warehouse/names': No such file or directory

[cloudera@quickstart ~]$

Important : Soyez prudent en supprimant les tables Hive, assurer vous que les données sont stockées ailleurs si vous souhaitez les réutiliser.

7- Définir une table externe

7.1 copier le fichier names.txt dans le répertoire /user/root dans HDFS

hadoop fs -put names.txt /user/root/

7.2 Créer le folder hivedemo dans le /user/ pour l’utiliser dans la table externe

hadoop fs -mkdir /user/hivedemo;

7.3 Créer la table externe names ayant les mêmes colonnes que la question 6.1

Spécifier la location :/user/hivedemo

create external table names (id int, name string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

LOCATION '/user/hivedemo';

;

7.4 Alimenter la table names avec les données du fichier names.txt en utilisant la function load

hive> load data inpath '/user/root/names.txt' into table names;

Loading data to table default.names

Table default.names stats: [numFiles=1, numRows=0, totalSize=78, rawDataSize=0]

OK

Time taken: 1.418 seconds

hive>

7.5 Vérifier que la table a été bien chargée :

hive> select \* from names;

OK

0 Rich

1 Barry

2 George

3 Ulf

4 Danielle

5 Tom

6 manish

7 Brian

8 Mark

Time taken: 0.207 seconds, Fetched: 9 row(s)

hive>

7.6

Vérifier si le fichier names.txt que vous avez copié dans le répertoire /user/root/ existe toujours ?

Afficher le contenu du dossier /user/hivedemo

Que constatez-vous ?

hadoop fs –ls /user/root

* Le fichier names.txt n’existe plus dans le répertoire courant

hadoop fs -ls /user/hivedemo/

Found 1 items

-rwxr-xr-x 1 cloudera cloudera 78 2016-12-17 11:29 /user/hivedemo/names.txt

* Le fichier names.txt a été déplacé vers le dossier /user/hivedemo/

7.7 vérifier que dans le répertoire /user/hive/wharehouse/ ne contient pas un répertoire names

hadoop fs -ls /user/hive/warehouse/

7.8 supprimer la table names

hive> drop table names;

OK

Time taken: 0.434 seconds

hive>

7.8 Vérifier le contenu du répertoire /user/hivedemo

hadoop fs -ls /user/hivedemo/

Found 1 items

-rwxr-xr-x 1 cloudera cloudera 78 2016-12-17 11:29 /user/hivedemo/names.txt

* Le fichier existe toujours

**Conclusion**:

Table interne :

* A la creation de la table interne le fichier est stocké dans le repertoire /user/hive/warehouse/nom\_table
* Lors de la suppression de la table, le fichier et le répertoire créés dans warehouse seront aussi supprimés.

Table externe :

* Au moment du chargement de la table externe le fichier sera déplacé à l’endroit (location) spécifié lors de la création de la table.
* Lors de la suppression de la table, le fichier restera toujours dans location