Atelier SQOOP :

1- Importer les données d'une Base de données vers HDFS

L'objectif :

- Importer les données d'une base de données vers HDFS

Emplacement du fichier : /formation/ateliers/sqoop/

Réalisation :

Chapitre correspondant : SQOOP

1- Installer mysql :

http://www.cloudera.com/documentation/archive/manager/4-x/4-5-1/Cloudera-Manager-Enterprise-Edition-Installation-Guide/cmeeig_topic_5_5.html

suivre les consignes Redhat

```
2- $ sudo yum install mysql-server
3- $ sudo service mysqld start
4- $ sudo /sbin/chkconfig mysqld on
5- $ sudo /sbin/chkconfig --list mysqld
6- mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

7- Création d'une table MYSQL :

a- Ouvrir le terminal de la ligne de commande de votre VM

b- Aller dans le dossier formation/sqoop

```
cd formation/ateliers/sqoop/
```

c- Visualiser le fichier formation/ateliers/sqoop/salaries.txt

```
[root@sandbox sqoop]# tail salaries.txt
```

Le fichier représente le sexe, age, le salaire et le code postal.

d- Afin de créer une table MYSQL, exécuter le script formation/ateliers/sqoop/salaries.sql qui définit une nouvelle table dans MySQL nommée.
Pour que ce script fonctionne, vous devez copier salaries.txt dans le répertoire / tmp

e- Créer la database test dans mysql
Cloudera :

```
mysql -u root -p
Enter password:cloudera
```

f- Copier le fichier salaries dans /tmp ou modifier le chemin dans le script salaries.sql

```
[root@sandbox sqoop]# cp salaries.txt /tmp
```

g- Donnez tous les droits pour exécuter le fichier Salaries.txt

```
chmod 777 salaries.txt
```

h- Vérifier si la base de données test existe :

```
mysql> show databases;
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
12 rows in set (0.01 sec)
```

i- Si elle n'existe pas, il faut la créer :

```
mysql> create database test;
Query OK, 1 row affected (0.00 sec)
```

j- Vérifie que la base de données a été bien créée.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
```

```
| rman          |
| sentry        |
| test          |
+---------------+
13 rows in set (0.00 sec)
```

k- Exécuter le script salaries.sql en utilisant la commande suivante :
Attention : lancer en mode command shell et non pas dans mysql

Cloudera :

```
[root@sandbox sqoop]# mysql -u root -p test < salaries.sql
```

8- Visualiser la table MYSQL :

a- Pour vérifier que la table a été bien créée, ouvrir la commande MYSQL comme suit :
Cloudera :

```
mysql – u root -p
```

Taper le mot de passe cloudera

b-Connectez-vous à la base de données où la table salaries a été créée avec la commande suivante :

```
mysql> use test
```

c- Exécuter la commande show tables ; et vérifier que la table salaries a été bien créée

```
 mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| salaries       |
+----------------+
1 row in set (0.00 sec)

mysql>
```

d-Lancer la requête ci-dessous pour vérifier que la table est bien alimentée.

```
mysql> select * from salaries limit 10;
+--------+------+--------+---------+----+
| gender | age  | salary | zipcode | id |
+--------+------+--------+---------+----+
| F      |   66 |  41000 |   95103 |  1 |
| M      |   40 |  76000 |   95102 |  2 |
| F      |   58 |  95000 |   95103 |  3 |
| F      |   68 |  60000 |   95105 |  4 |
| M      |   85 |  14000 |   95102 |  5 |
| M      |   14 |      0 |   95105 |  6 |
| M      |   52 |   2000 |   94040 |  7 |
| M      |   67 |  99000 |   94040 |  8 |
| F      |   43 |  11000 |   94041 |  9 |
| F      |   37 |  65000 |   94040 | 10 |
+--------+------+--------+---------+----+
10 rows in set (0.00 sec)
```

e- Sortez de MYSQL PROMPT

```
mysql> exit
```

9- Importer la table dans HDFS :

Attention :

Lancer en mode command shell et non pas dans mysql

Lancer la commande en une seule ligne.

a- Entrez la ligne de commande ci-dessous pour importer la table salaries dans HDFS

Cloudera :

```
sqoop import --connect jdbc:mysql://localhost/test --table salaries    --username root -P
```

b- Le MapReduce doit s'exécuter pour lancer l'importation de la table, attendre quelques minutes pour que ça se finisse.

10- Vérifier l'import de la table :

a- Vérifier le contenu de HDFS (répertoire courant)

```
[root@sandbox ~]# hadoop fs -ls
```

b- Vous devriez voir un nouveau répertoire salaries, vérifier son contenu :

```
[root@sandbox ~]# hadoop fs -ls salaries
Found 5 items
-rw-r--r--   3 root hdfs          0 2016-08-24 10:49
salaries/_SUCCESS
-rw-r--r--   3 root hdfs 272 2016-08-24 10:49 salaries/part-m-00000
-rw-r--r--   3 root hdfs 241 2016-08-24 10:49 salaries/part-m-00001
-rw-r--r--   3 root hdfs 238 2016-08-24 10:49 salaries/part-m-00002
-rw-r--r--   3 root hdfs 272 2016-08-24 10:49 salaries/part-m-00003
```

c- Vous constaté la présence de 4 fichiers part-m-000X (1 à 4), pourquoi 4 fichiers ?

Le MapReduce qui a exécuté ce job a utilisé 4 mapper, chaque mapper a généré un fichier.

d- Utiliser cat pour visualiser le contenu du fichier part-m-00000

```
[root@sandbox ~]# hadoop fs -tail salaries/part-m-00000
```

Vous constatez que les 4 fichiers correspondent au contenu de la table salaries MYSQL, dans cette atelier nous avons importé toutes colonnes de la table. Dans les exercices suivants, nous allons voir comment extraire qu'une partie des données de la table

11- Spécifier les colonnes à importer :

a- Utiliser l'argument –columns argument pour spécifier les colonnes à importer, dans notre cas, ça sera les colonnes salary et age (dans l'ordre) de la table salaries MYSQL vers un fichier salaries2 dans HDFS. Enfin, utiliser l'argument –m à 1 pour générer le résultant en un seul fichier dans HDFS.

Solution : Entrez en une seule ligne la commande suivante :

Cloudera :

```
sqoop import --connect jdbc:mysql://localhost/test --table salaries     --username root -P
--columns salary,age -m 1 --target-dir salaries2
```

L'argument --target-dir indique le chemin et le répertoire dans lequel la table MYSQL doit être importée.

Important :

Les commandes doivent être exécutées sur en seule ligne, dans le cas contraire, il faut rajouter \ à la fin de chaque ligne. Dans notre cas :

```
# sqoop import --connect jdbc:mysql://sandbox/test?user=root \
--table salaries \
--columns salary,age \
-m 1 \
--target-dir salaries2
```

b- Après l'import, vérifier que vous avez bien un seul fichier dans HDFS

```
[root@sandbox ~]# hadoop fs -ls salaries2
Found 2 items
-rw-r--r--   3 root hdfs  0 2016-08-24 11:34 salaries2/_SUCCESS
-rw-r--r--   3 root hdfs  482 2016-08-24 11:34 salaries2/part-m-00000
```

c- Vérifier le contenu du fichier part-m-00000

```
[root@sandbox ~]# hadoop fs -cat salaries2/part-m-00000
41000.0,66
76000.0,40
95000.0,58
60000.0,68
14000.0,85
0.0,14
2000.0,52
```

12- Importer les données d'une requête SQL :

a- Ecrivez une commande Sqoop qui permet de charger dans HDFS les données de la table salaries ayant un salaire supérieur à 90,000.00

Informations : Uitliser :

- ✓ --query à la place de –table car notre objectif est de faire une requête.
- ✓ --split by « column » en ajoutant le champ souhaité pour séparer les résultats selon votre besoin.
- ✓ Deux Mapper pour ce job
- ✓ La cible HDFS est le répertoire salaries3
- ✓ A la fin de la condition de notre requête, ajouter toujours **and \$CONDITIONS**

Cloudera :

```
sqoop import --connect jdbc:mysql://localhost/test --username root -P --driver com.mysql.jdbc.Driver --query "select * from salaries s where s.salary > 90000.00 and \$CONDITIONS" --split-by gender -m 2 --target-dir salaries3
```

b- Vérifier que le répertoire salaries a été bien créé et contient les deux fichiers

```
hadoop  fs -ls salaries3
```

c- Visualiser le contenu du fichier part-m-00000 et le fichier part-m-00001

```
[root@sandbox ~]# hadoop fs -cat salaries3/part-m-00000
F,58,95000.0,95103,3
```

```
[root@sandbox ~]# hadoop fs -cat salaries3/part-m-00001
M,67,99000.0,94040,8
M,44,96000.0,94040,33
M,31,95000.0,94041,39
M,48,91000.0,95102,46
```

d- Vous remarquez qu'un fichier contient les femmes, l'autres contient les hommes, pourquoi ?

Nous avons utilisé le commande –split by gender, donc, deux mapper, un pour les femmes l'autres pour les hommes.

e- Vérifier que les deux fichiers contiennent bien uniquement les personnes ayant un salaire supérieur à 90,000.00

Résultat :

Vous avez importé les données les données de toute la table dans HDFS, mais aussi via des requêtes pour extraire qu'une partie des données.

2- Importer les données de HDFS vers une Base de données.

Export HDFS to BDR :

Créer un répertoire dans HDFS nommé `salarydata`.

```
# hdfs dfs -mkdir salarydata
```

d. Mettre salarydata.txt dans le repertoire HDFS `salarydata`.

```
# hdfs dfs -put salarydata.txt salarydata
```

Créer une table salaries2 sous mysql gender en varchar2

```
use test;
drop table if exists salaries2;
create table salaries2 (
gender varchar(1),
age int,
salary double,
zipcode int);
```

Vérifier que la table a été créée avec succès:

```
# mysql
mysql> use test;
mysql> describe salaries2;
```

Exporter les données avec Sqoop dans la table salaries2

```
sqoop export \
--connect jdbc:mysql://localhost/test \
--username root --password cloudera \
--table salaries2 \
--export-dir salarydata \
--input-fields-terminated-by ","
```