TP Revision Hive

- Ouvrir la machine virtuelle et télécharger le fichier **user-posts.csv** à partir du Classroom et le mettre dans le **dossier data,** que vous créez dans le dossier personnel
- Taper : Start-all.sh
- Puis démarrer le shell hive avec : hive

1) Afficher le contenu du fichier **user-posts**

```
! cat /home/u1/data/user-posts.csv;
```

2) Créer une table « posts » ayant les champs suivants :
Users, post, time (de type BIGINT)

```
hive> CREATE TABLE posts (users STRING, post STRING, time BIGINT)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',’
   > STORED AS TEXTFILE;
```

3) Afficher la liste de tables `hive> show tables;`

4) Afficher le **schéma** de la table posts `hive> describe posts;`

5) Charger les données du fichier users-post.csv, dans la table posts.

```
hive> LOAD DATA LOCAL INPATH ‘/home/u1/data/user-posts.csv’
    > OVERWRITE INTO TABLE posts;
```

6) Ouvrir un autre terminal. Visualiser le contenu du dossier hive

```
Hdfs dfs –ls /user/hive
```

7) Visualiser le contenu du fichier users-post.csv

```
Hdfs dfs –cat  /user/hive/warehouse/posts/user-posts.csv
```

8) Basculer vers la fenêtre de hive interactif. Afficher le nombre d'enregistrements de la table posts.

```
hive> select count (1) from posts;
```

9) Afficher l'enregistrement user2

```
hive> select * from posts where users="user2";
```

10) Afficher deux enregistrements de la table avec le champs time <=1343182133839

```
hive> select * from posts where time<=1343182133839 limit 2;
```

11) Créer la table likes contenant le nombre « j'aime » pour chaque utilisateur:

users, likes_count, time(BIGINT)

```
hive> CREATE TABLE likes (users STRING, likes_count INT, time BIGINT)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',’
   > STORED AS TEXTFILE;
```

12) Créer une table qui permet d'avoir les champs: user, post, likes_count

```
hive> CREATE TABLE posts_likes (users STRING, post STRING, likes_count INT);
```

13) Donner la requête permettant d'avoir les valeurs des champs de la table post_likes

```
hive> INSERT OVERWRITE TABLE posts_likes
> SELECT p.users, p.post, l.likes_count
> FROM posts p JOIN likes l ON (p.users = l.users);
```

14) Effacer la table posts.

```
hive> DROP TABLE posts;
```

15) Vérifier l'existence de la table posts dans warehouse (dans le deuxième terminal)

```
Hdfs dfs -ls  /user/hive/warehouse/
```

16) Créer une table posts en partitionnant suivant « country »

```
hive> CREATE TABLE posts (user STRING, post STRING, time BIGINT)
> PARTITIONED BY(country STRING)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> STORED AS TEXTFILE
```

17) Afficher le schéma de la table posts

```
hive> describe posts;
```

18) Afficher les partitions de posts

```
hive>show partitions posts;
```

```
hive> CREATE TABLE posts (user STRING, post STRING, time BIGINT)
> PARTITIONED BY(country STRING)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> STORED AS TEXTFILE
```

19) Afficher le schéma de la table posts

```
hive> describe posts;
```

20) Afficher les partitions de posts

```
hive>show partitions posts;
```

```
Aucune partition n'est affichée
```

21) Charger les données du fichier users-post-US.csv, dans la table posts.

```
hive> LOAD DATA LOCAL INPATH 'data/user-posts-US.txt'
> OVERWRITE INTO TABLE posts;
```

```
FAILED: Error in semantic analysis: Need to specify partition
columns because the destination table is partitioned
```

22) Charger les données du fichier users-post-US.csv, dans la table posts, la partition country=« us ».

```
hive>LOAD DATA LOCAL INPATH 'data/user-posts-US.csv'
> OVERWRITE INTO TABLE posts PARTITION(country='US');
```

23) Charger les données du fichier users-post- Australia.csv, dans la table posts, la partition country=«Australia».

```
hive>LOAD DATA LOCAL INPATH 'data/user-posts- Australia.csv'
> OVERWRITE INTO TABLE posts PARTITION(country=' Australia ');
```

24) Afficher les partitions de posts

```
hive>show partitions posts;
```

```
OK
country=AUSTRALIA
country=US
Time taken: 0.095 seconds
```

25) Afficher les 5 enregistrements de la table posts avec la condition country=US

```
hive>select * from posts where country='US' limit 5;
```

Uniquement la partition: « country=US » sera considérée

```
user1 Funny Story 1343182026191 US
user2 Cool Deal 1343182133839 US
user2 Great Interesting Note 13431821339485 US
user4 Interesting Post 1343182154633 US
user1 Humor is good 1343182039586 US
```

26) Basculer vers le deuxième terminal (hadoop), afficher la table posts avec ses partitions

```
$ hdfs dfs -ls -R /user/hive/warehouse/posts
```

/user/hive/warehouse/posts/**country=AUSTRALIA**
/user/hive/warehouse/posts/country=AUSTRALIA/user-posts-AUSTRALIA.txt
/user/hive/warehouse/posts/**country=US**
/user/hive/warehouse/posts/country=US/user-posts-US.txt