

# **Ateliers pratiques Big Data sous HDP**

**Prof. Karim Baïna**

karim.baina@gmail.com

Expert Big Data Engineering & Analytics

## Atelier 6 : Utiliser Apache Hive pour accéder aux données Hadoop

**But : Cet exercice vous présente Apache Hive. Vous allez acquérir de l'expérience avec le shell Hive CLI pour stocker, traiter et accéder aux données Hadoop. Vous apprendrez également à obtenir des informations sur la configuration Hive à l'aide de l'interface utilisateur Web Ambari.**

L'entrepôt de données Apache Hive facilite l'interrogation et la gestion de grands ensembles de données qui résident dans le stockage distribué. Construit sur Apache Hadoop, il fournit :

- Outils permettant d'extraire/transformer/charger facilement les données (ETL).
- Un mécanisme pour imposer une structure sur divers formats de données.
- Accès aux fichiers stockés directement dans Apache HDFS ou dans d'autres systèmes de stockage de données tels que Apache HBase.
- Exécution de requêtes via MapReduce.

Hive définit un langage de requête simple de type SQL, appelé HiveQL, qui permet aux utilisateurs familiarisés avec SQL d'interroger les données. Dans le même temps, ce langage permet également aux programmeurs familiarisés avec le framework MapReduce de brancher leurs mappeurs et réducteurs personnalisés pour effectuer des analyses plus sophistiquées qui pourraient ne pas être prises en charge par les capacités intégrées du langage. QL peut également être étendu avec des fonctions scalaires personnalisées définies par l'utilisateur (UDF), des agrégations (UDAF) et des fonctions de table (UDTF).

Hive n'exige pas que les données lues ou écrites soient au "format Hive", il n'y a rien de tel. Hive fonctionne aussi bien sur Thrift, control délimité ou vos formats de données spécialisés.

Hive n'est pas conçu pour les charges de travail OLTP et n'offre pas de requêtes en temps réel ni de mises à jour au niveau des lignes. Il est préférable de l'utiliser pour les tâches par lots sur de grands ensembles de données d'ajout uniquement (comme les journaux Web). Les valeurs Hive les plus importantes sont l'évolutivité (évolutivité avec plus de machines ajoutées dynamiquement au cluster Hadoop), l'extensibilité (avec le framework MapReduce et UDF/UDAF/UDTF), la tolérance aux pannes et le couplage lâche avec ses formats d'entrée. En bref, considérez Hive comme une base de données SQL à ajout uniquement. Hive est utile pour l'intégration SQL si vous souhaitez stocker des données à long terme à traiter, à résumer et à recharger. La principale limitation de Hive est la vitesse des requêtes. Lorsqu'il s'agit de milliards de lignes, il n'y a pas d'interrogation en direct des données qui serait suffisamment rapide pour une interface interactive avec les données. Par exemple, avec l'enregistrement des données, les quantités de données peuvent être énormes, mais ce dont vous avez souvent besoin, c'est d'une requête rapide et flexible sur des données résumées ou extrêmes, c'est-à-dire des défauts et des échecs.

Dans cet exercice, vous allez apprendre à obtenir des informations sur les services Hive à l'aide de l'interface utilisateur Web Ambari. Vous utiliserez également la CLI Hive pour créer des tables Hive, importer des données dans Hive et interroger des données Hive. Enfin, vous utilisez la CLI Beeline pour interroger des données dans Hive.

### Tâche 1. Apprendre à gérer des données Hadoop à travers des tables Hive managées relationnelles

Vous êtes déjà connecté en tant que student sur votre VM HDP.

**NB. Si le fichier CDR.csv est déjà mis sur HDFS /user/student/data sauter-les étapes 1-4 !!**

#### 1. Uploader un fichier de données CDR depuis votre machine vers le compte student de la machine virtuelle

(Sous Windows) Clicker sur le menu **Démarrer**, puis rechercher **cmd**.

Mettez-vous sur le répertoire contenant le fichier de données CDR.csv (par ex. C:)

Taper par exemple :

```
scp -r -p -P 2222 CDR.csv student@192.168.50.129:CDR.csv
```

**NB.** Dans la figure suivante, Remplacez 192.168.50.129 par l'URL (ou l'adresse IP) de votre VM HDP !!

```
I:\DATA\LAKE\20210105\async-ensias\async-Teaching\Big_Data_AMOA\Demos\R&D\CDRGenerator>scp -r -p -P 2222 CDR.csv student@192.168.50.129:CDR.csv
student@192.168.50.129's password:
CDR.csv 100% 128MB 94.8MB/s 00:01
```

## 2. S'assurer du bon transfert du fichier de données depuis votre machine au compte de l'utilisateur (student) dans la VM

Retourner à votre console de la VM HDP et Taper :

ls

```
[student@sandbox-hdp ~]$ ls
CDR.csv
[student@sandbox-hdp ~]$
```

## 3. Uploader ce fichier de données depuis Linux de la VM vers l'espace HDFS associé à l'utilisateur (student)

Sur votre console de la VM HDP, Taper :

### Besoin de création du répertoire data/cdr

```
hadoop fs -mkdir /user/student/data/cdr
```

```
hadoop fs -put /home/student/CDR.csv /user/student/data/cdr/
```

## 4. Afficher les informations à propos du fichier de données HDFS

```
hadoop fs -ls data/cdr/CDR.csv
```

(oubien `hadoop fs -ls /user/student/data/cdr/CDR.csv` vu que `/user/student/` est par défaut)

```
[student@sandbox-hdp ~]$ hadoop fs -ls data/cdr/CDR.csv
-rw-r--r-- 1 student student 132428102 2021-09-28 22:05 data/cdr/CDR.csv
```

## 5. Exploration des données du fichier HDFS CDR.csv

```
hadoop fs -cat /user/student/data/cdr/CDR.csv | head -n 10
```

```
[student@sandbox-hdp ~]$ hadoop fs -cat /user/student/data/cdr/CDR.csv | head -n 10
c1070550-08d7-42bf-84b2-9f67ce9a1dfa|0699176182|0684508434|2021-09-13 15:14:57.041|2021-09-13 15:14:57.041|SMS|0.8072003|ANSWERED
5def1fde-ae78-4902-a5ac-54cd790d5bcb|0629728131|0616679734|2021-09-19 14:23:32.791|2021-09-19 14:26:06.988|VOICE|0.8080974|ANSWERED
e7bbe98b-90f7-43f3-a878-efe4d6fbfcb5|0670914600|0691495655|2021-10-14 18:02:38.434|2021-10-14 18:04:49.222|VOICE|0.013672888|ANSWERED
b741c7d6-9d23-47f1-bea1-f2788d65faf4|0612353921|0625883917|2021-09-18 16:17:40.110|2021-09-18 16:17:40.110|SMS|0.30302763|ANSWERED
6038f390-90df-418c-995a-e95ebf42df54|0640784239|0676557336|2021-09-22 00:44:44.629|2021-09-22 00:47:10.159|VOICE|0.07670128|ANSWERED
dcd1751c-4358-4354-aaa0-c630438da009|0611901116|0696479453|2021-10-01 10:17:47.399|2021-10-01 10:19:56.727|VOICE|0.07953727|ANSWERED
cd83425d-9e9b-41e2-8436-dfdbaf33907a|0631374947|0600587955|2021-09-30 16:54:46.061|2021-09-30 16:57:19.763|VOICE|0.73849326|ANSWERED
5985d6ca-969f-4ce2-9e77-a530d1a1b418|0696524704|0623395634|2021-09-07 20:12:53.066|2021-09-07 20:15:01.913|VOICE|0.6276111|ANSWERED
095ebfb5-f6fb-4992-99f1-0714664641d4|0618422948|0664041874|2021-09-04 01:42:31.907|2021-09-04 01:45:27.026|VOICE|0.9577195|ANSWERED
1793a21b-127a-4950-9fe1-c77f014d3726|0657102291|0663944393|2021-10-04 00:09:00.737|2021-10-04 00:09:00.737|SMS|0.756527|ANSWERED
```

1. id : STRING
2. calling\_num : STRING (numéro appelant)
3. called\_num : STRING (numéro appelé)
4. start\_time : TIMESTAMP (début appel)
5. end\_time : TIMESTAMP (fin appel)
6. call\_type : STRING (Voice, SMS)
7. charge : FLOAT (coût de l'appel chargé à l'appelant)
8. call\_result : STRING (ANSWERED, BUSY) (statut de l'appel)

Source du dataset : génération aléatoire automatique.

## 6. Assurez-vous que tous les services Hadoop sont démarrés à travers Ambari Web UI



## 7. Lancer le shell Hive CLI de datawarehouse Hive

/usr/bin/hive shell

```
[student@sandbox-hdp ~]$ /usr/bin/hive shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zooKeeperNamespace=hiveserver2
21/09/29 09:44:17 [main]: INFO jdbc.HiveConnection: Connected to sandbox-hdp.hortonworks.com:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

## 8. Créer une base de données pour le compte courant (student)

CREATE DATABASE db\_student;

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> CREATE DATABASE db_student;
INFO : Compiling command(queryId=hive_20210929094908_3bf20a38-5dfe-47fb-8b1d-d9db3e1c1012): CREATE DATABASE db_student
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20210929094908_3bf20a38-5dfe-47fb-8b1d-d9db3e1c1012); Time taken: 0.085 seconds
INFO : Executing command(queryId=hive_20210929094908_3bf20a38-5dfe-47fb-8b1d-d9db3e1c1012): CREATE DATABASE db_student
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929094908_3bf20a38-5dfe-47fb-8b1d-d9db3e1c1012); Time taken: 0.58 seconds
INFO : OK
No rows affected (0.851 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

## 9. Vérifier la création de la base de données du compte courant (student)

SHOW DATABASES;

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> SHOW DATABASES;
INFO : Compiling command(queryId=hive_20210929095059_465efcb0-cc0d-4bf7-a754-7e08d6be9ca4): SHOW DATABASES
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20210929095059_465efcb0-cc0d-4bf7-a754-7e08d6be9ca4); Time taken: 0.017 seconds
INFO : Executing command(queryId=hive_20210929095059_465efcb0-cc0d-4bf7-a754-7e08d6be9ca4): SHOW DATABASES
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929095059_465efcb0-cc0d-4bf7-a754-7e08d6be9ca4); Time taken: 0.005 seconds
INFO : OK
+-----+
| database_name |
+-----+
| db_student    |
| default       |
| foodmart      |
| information_schema |
| sys           |
+-----+
5 rows selected (0.222 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

## 10. Vérifier la création de la base de données du compte courant (student)

use db\_student;

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> use db_student ;
INFO : Compiling command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607): use db_student
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607); Time taken: 0.026 seconds
INFO : Executing command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607): use db_student
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607); Time taken: 0.015 seconds
INFO : OK
No rows affected (0.053 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

## 11. Créer une table Hive CDR

```
CREATE TABLE CDR (
id STRING,
calling_num STRING,
called_num STRING,
start_time TIMESTAMP,
end_time TIMESTAMP,
call_type STRING,
charge FLOAT,
call_result STRING) row format delimited
fields terminated by '|'
lines terminated by '\n'
stored as textfile;
```

## 12. Vérifier la création de la table Hive CDR dans la base de données Hive associée à l'utilisateur courant (student)

```
use db_student;
show tables;
```

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> use db_student;
INFO : Compiling command(queryId=hive_20210929100126_69958a1c-233f-489a-bb6f-d2f1858cdd34): use db_student
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20210929100126_69958a1c-233f-489a-bb6f-d2f1858cdd34); Time taken: 0.021 seconds
INFO : Executing command(queryId=hive_20210929100126_69958a1c-233f-489a-bb6f-d2f1858cdd34): use db_student
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929100126_69958a1c-233f-489a-bb6f-d2f1858cdd34); Time taken: 0.01 seconds
INFO : OK
No rows affected (0.043 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> show tables ;
INFO : Compiling command(queryId=hive_20210929100129_c59dc6ff-2e3d-459b-b8ce-c23a197bdfa2): show tables
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20210929100129_c59dc6ff-2e3d-459b-b8ce-c23a197bdfa2); Time taken: 0.028 seconds
INFO : Executing command(queryId=hive_20210929100129_c59dc6ff-2e3d-459b-b8ce-c23a197bdfa2): show tables
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929100129_c59dc6ff-2e3d-459b-b8ce-c23a197bdfa2); Time taken: 0.015 seconds
INFO : OK
+-----+
| tab_name |
+-----+
| cdr      |
+-----+
1 row selected (0.064 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

## 13. Vérifier le schéma de la table CDR dans Hive

```
describe cdr;
```

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> describe cdr ;
INFO : Compiling command(queryId=hive_20210926135822_8893d7f7-e044-4cd1-b8bb-3cc621dc6be0): describe cdr
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comment, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20210926135822_8893d7f7-e044-4cd1-b8bb-3cc621dc6be0); Time taken: 0.045 seconds
INFO : Executing command(queryId=hive_20210926135822_8893d7f7-e044-4cd1-b8bb-3cc621dc6be0): describe cdr
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210926135822_8893d7f7-e044-4cd1-b8bb-3cc621dc6be0); Time taken: 0.025 seconds
INFO : OK
+-----+-----+-----+
| col_name | data_type | comment |
+-----+-----+-----+
| id        | string    |         |
| calling_num | string    |         |
| called_num | string    |         |
| start_time | timestamp |         |
| end_time   | timestamp |         |
| call_type  | string    |         |
| charge     | float     |         |
| call_result | string    |         |
+-----+-----+-----+
8 rows selected (0.115 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

## 14. Charger le fichier de données CDR.csv depuis HDFS vers la table Hive CDR

```
LOAD DATA INPATH '/user/student/data/cdr/CDR.csv' OVERWRITE INTO TABLE cdr;
```

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> LOAD DATA LOCAL INPATH '/home/student/cdr/CDR.csv' OVERWRITE INTO TABLE CDR ;
INFO : Compiling command(queryId=hive_20210926134033_a78dada8-759f-41d3-9b58-f4e6a1cc4597): LOAD DATA LOCAL INPATH '/home/student/CDR.csv' OVERWRITE INTO TABLE CDR
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20210926134033_a78dada8-759f-41d3-9b58-f4e6a1cc4597); Time taken: 0.256 seconds
INFO : Executing command(queryId=hive_20210926134033_a78dada8-759f-41d3-9b58-f4e6a1cc4597): LOAD DATA LOCAL INPATH '/home/student/CDR.csv' OVERWRITE INTO TABLE CDR
INFO : Starting task [Stage-0:MOVE] in serial mode
INFO : Loading data to table default.cdr from file:/home/student/CDR.csv
INFO : Starting task [Stage-1:STATS] in serial mode
INFO : Completed executing command(queryId=hive_20210926134033_a78dada8-759f-41d3-9b58-f4e6a1cc4597); Time taken: 2.895 seconds
INFO : OK
No rows affected (5.071 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

## 15. Vérifier le chargement du fichier de données CDR.csv depuis HDFS vers la table Hive CDR

select \* from cdr LIMIT 10;

```
INFO : Compiling command(queryId=hive_20210926141730_5c5c544c-7490-412e-84c3-73960049a8f4): select * from CDR LIMIT 10
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:cdr.id, type:string, comment:null), FieldSchema(name:cdr.calling_num, type:string, comment:null), FieldSchema(name:cdr.called_num, type:string, comment:null), FieldSchema(name:cdr.start_time, type:timestamp, comment:null), FieldSchema(name:cdr.end_time, type:timestamp, comment:null), FieldSchema(name:cdr.call_type, type:string, comment:null), FieldSchema(name:cdr.charge, type:float, comment:null), FieldSchema(name:cdr.call_result, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=hive_20210926141730_5c5c544c-7490-412e-84c3-73960049a8f4); Time taken: 0.321 seconds
INFO : Executing command(queryId=hive_20210926141730_5c5c544c-7490-412e-84c3-73960049a8f4): select * from CDR LIMIT 10
INFO : Completed executing command(queryId=hive_20210926141730_5c5c544c-7490-412e-84c3-73960049a8f4); Time taken: 0.007 seconds
INFO : OK
```

cdr.id	cdr.calling_num	cdr.called_num	cdr.start_time	cdr.end_time	cdr.call_type	cdr.charge	cdr.call_result
c1070550-08d7-42bf-84b2-9f67ce9a1dfa	0699176182	0684508434	2021-09-13 15:14:57.041	2021-09-13 15:14:57.041	SMS	0.8072003	ANSWERED
5deff1de-ae78-4902-a5ac-54cd790d5bcb	0629728131	0616679734	2021-09-19 14:23:32.791	2021-09-19 14:26:06.988	VOICE	0.8080974	ANSWERED
e7bbe90b-90f7-43f3-a878-efed66fbcb5	0670914600	0691495655	2021-10-14 18:02:38.434	2021-10-14 18:04:49.222	VOICE	0.013672888	ANSWERED
b741c7d6-9d23-47f1-bea1-f2788d65faf4	0612353921	0625883917	2021-09-18 16:17:40.11	2021-09-18 16:17:40.11	SMS	0.30302763	ANSWERED
6038f390-90df-418c-995a-e95abf42df54	0640784239	0676557336	2021-09-22 00:44:44.629	2021-09-22 00:47:10.159	VOICE	0.87676128	ANSWERED
dc01751c-4358-4354-aaa0-c630438da009	0611901116	0696479453	2021-10-01 10:17:47.399	2021-10-01 10:19:56.727	VOICE	0.87953727	ANSWERED
cd83425d-9e9b-41e2-8436-dfdba733907a	0631374947	0600587955	2021-09-30 16:54:46.061	2021-09-30 16:57:19.763	VOICE	0.73849326	ANSWERED
5985d6ca-969f-4ce2-9e77-a530d1a1b418	0696524704	0623395634	2021-09-07 20:12:53.066	2021-09-07 20:15:01.913	VOICE	0.6276111	ANSWERED
095ebfb5-f6fb-4992-99f1-0714664614d4	0618422948	0664041874	2021-09-04 01:42:31.907	2021-09-04 01:45:27.026	VOICE	0.9577195	ANSWERED
1793a21b-127a-4950-9fe1-c77f014d3726	0657102291	0663944393	2021-10-04 00:09:00.737	2021-10-04 00:09:00.737	SMS	0.756527	ANSWERED

```
10 rows selected (0.453 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

NB. Sous les versions (2.6/2.7 sous Azure), la requête affichera les données sans les noms des colonnes.

## 16. Exploration de la force de SQL sur les données réparties sous HDFS de la table Hive CDR

```
select CALLING_NUM, (END_TIME-START_TIME) AS DURATION
from CDR
ORDER BY DURATION DESC
LIMIT 10;
```

calling_num	duration
0664625205	0 00:03:00.000000000
0643567331	0 00:03:00.000000000
0695884845	0 00:03:00.000000000
0620971959	0 00:03:00.000000000
0668758824	0 00:03:00.000000000
0656468109	0 00:03:00.000000000
0613677842	0 00:03:00.000000000
0634486253	0 00:03:00.000000000
0647313423	0 00:03:00.000000000
0677003366	0 00:03:00.000000000

```
10 rows selected (10.437 seconds)
```

```
select Date(START_TIME) AS CALL_DATE, Count(*) AS CALLS
from CDR
where CALL_TYPE = "VOICE"
GROUP BY Date(START_TIME)
ORDER BY CALLS DESC
LIMIT 10;
```

call_date	calls
2021-10-18	10250
2021-09-21	10247
2021-09-11	10237
2021-09-09	10226
2021-10-15	10223
2021-10-07	10206
2021-09-18	10195
2021-10-11	10193
2021-09-15	10192
2021-09-05	10177

```
10 rows selected (13.246 seconds)
```

```
select MONTH(START_TIME) AS CALL_MONTH, YEAR(START_TIME) AS CALL_YEAR, Count(*) AS
CALLS
from CDR
where CALL_TYPE = "VOICE"
GROUP BY MONTH(START_TIME), YEAR(START_TIME)
ORDER BY CALLS DESC
LIMIT 1;
```

call_month	call_year	calls
9	2021	294148

1 row selected (10.576 seconds)

```
select YEAR(START_TIME), AVG(DATEDIFF(END_TIME,START_TIME)) AS CALL_DURATION_AVG
from CDR
where CALL_TYPE = "VOICE"
GROUP BY YEAR(START_TIME);
```

```
select Date(START_TIME) AS SMS_DATE, Count(*) AS SMS
from CDR
where CALL_TYPE = "SMS"
GROUP BY Date(START_TIME)
ORDER BY SMS DESC
LIMIT 10;
```

sms_date	sms
2021-09-20	10274
2021-09-30	10204
2021-09-17	10195
2021-09-29	10185
2021-10-02	10170
2021-10-01	10160
2021-09-03	10141
2021-10-05	10140
2021-10-09	10134
2021-10-08	10132

10 rows selected (11.484 seconds)

```
SELECT NUMBER, SUM(DEGREE_PLUS) DP, SUM(DEGREE_MOINS) DM, SUM(DEGREE_PLUS) +
SUM(DEGREE_MOINS) DEGREE
FROM
(
(select CALLING_NUM NUMBER, count(CALLED_NUM) DEGREE_PLUS, 0 DEGREE_MOINS
from CDR
GROUP BY CALLING_NUM
HAVING count(CALLED_NUM) != 0)

UNION

(select CALLED_NUM NUMBER, 0 DEGREE_PLUS, count(CALLING_NUM) DEGREE_MOINS
from CDR
GROUP BY CALLED_NUM
HAVING count(CALLING_NUM) !=0)
) T
GROUP BY NUMBER
ORDER BY DEGREE DESC, NUMBER ASC
LIMIT 10;
```

number	dp	dm	degree
0649349936	1	3	4
0649402765	1	3	4
0600021946	2	1	3
0600606028	3	0	3
0601243493	1	2	3
0601871839	1	2	3
0601945203	1	2	3
0602189738	2	1	3
0606468972	3	0	3
0607708592	1	2	3

10 rows selected (21.996 seconds)

0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █

## 17. Quitter le shell Hive CLI



!q

## Tâche 2. Apprendre à gérer des données Hadoop à travers des tables Hive non managées (externes)

**Pré-requis : Atelier 5 :** Développer et Exécuter une Application Spark en Python – **Tâche 3.** Apprendre à créer et exécuter une application Spark en Python.

En Atelier 5 (Tâche 3), la programmation Spark vous a permis de faire le pré-traitement des données SMS brutes, et en Atelier 6 (Tâche 2), le requêtage Hive vous permettra de faire de l'analyse sur les données pré-traitées.

### 1. Se logger en tant qu'utilisateur hdfs

```
su -
su - hdfs
```

### 2. Donner les privilèges sur le répertoire de données à l'utilisateur hive du groupe hdfs

```
hadoop fs -chown -R hive:hdfs /user/student/output-spamsms/
```

### 3. quitter la session hdfs

```
exit
```

### 4. quitter la session root

```
exit
```

### 5. Lancer le shell Hive CLI de datawarehouse Hive

```
/usr/bin/hive shell
```

### 6. Créer une table externe Hive pointant sur les données HDFS (de comptage de mot des sms) non managées par Hive

```
CREATE EXTERNAL TABLE IF NOT EXISTS
db_student.SMSWordCount (word STRING, countinspam INT, countinham INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/student/output-spamsms';
```

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> CREATE EXTERNAL TABLE IF NOT EXISTS
...>
...> db_student.SMSWordCount(word STRING, countinspam INT, countinham INT)
...>
...> ROW FORMAT DELIMITED
...>
...> FIELDS TERMINATED BY ','
...>
...> LINES TERMINATED BY '\n'
...>
...> STORED AS TEXTFILE
...>
...> LOCATION '/user/student/output-spamsms/';
INFO : Compiling command(queryId=hive_20211004094501_7bd79d13-f6a5-416d-b10b-d49acc089dc1): CREATE EXTERNAL TABLE IF NOT EXISTS
db_student.SMSWordCount(word STRING, countinspam INT, countinham INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/student/output-spamsms/'
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20211004094501_7bd79d13-f6a5-416d-b10b-d49acc089dc1); Time taken: 0.076 seconds
INFO : Executing command(queryId=hive_20211004094501_7bd79d13-f6a5-416d-b10b-d49acc089dc1): CREATE EXTERNAL TABLE IF NOT EXISTS
db_student.SMSWordCount(word STRING, countinspam INT, countinham INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/student/output-spamsms/'
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20211004094501_7bd79d13-f6a5-416d-b10b-d49acc089dc1); Time taken: 0.225 seconds
INFO : OK
No rows affected (0.586 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> □
```



```
FROM db_student.SMSWordCount
SELECT word, countinspam, countinham
WHERE word ='money' ;
```

word	countinspam	countinham
money	3	33

```
FROM db_student.SMSWordCount
SELECT word, countinspam, countinham
WHERE word ='work' ;
```

word	countinspam	countinham
work	2	56

```
FROM db_student.SMSWordCount
SELECT word, countinspam, countinham
WHERE word ='dating' ;
FROM db_student.SMSWordCount
SELECT word, countinspam, countinham
WHERE word ='chat' ;
```

word	countinspam	countinham
dating	15	NULL

word	countinspam	countinham
chat	37	11

### Tâche 3. Apprendre à gérer des données Hadoop à travers des tables Hive managées multidimensionnelles

**NB.** Si le fichier *city.csv* est déjà mis sur HDFS /user/student/data/cities sauter-les étapes 1-5 !!

**1. Uploader un fichier de données city.csv depuis votre machine vers le compte student de la machine virtuelle**

(Sous Windows) Cliquer sur le menu **Démarrer**, puis rechercher **cmd**.

Mettez-vous sur le répertoire contenant le fichier de données city.csv (par ex. I:)

Taper par exemple :

```
scp -P 2222 city.csv student@kbaina.westeurope.cloudapp.azure.com:city.csv
```

**NB.** Dans la figure suivante, Remplacez [kbaina.westeurope.cloudapp.azure.com](https://kbaina.westeurope.cloudapp.azure.com) par l'URL (ou l'adresse IP) de votre VM HDP !!

```

Microsoft Windows [version 10.0.19042.1348]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\workstation>i:

I:\DATA\LAKE\20210105\async-ensias\async-Teaching\Big_Data_Introduction_Platform_HDP\Lecture\V7\0.Pre-requisites\Datasets
>scp -P 2222 city.csv student@192.168.50.128:city.csv
student@192.168.50.128's password:
city.csv                                     100% 144MB 36.9MB/s 00:03

I:\DATA\LAKE\20210105\async-ensias\async-Teaching\Big_Data_Introduction_Platform_HDP\Lecture\V7\0.Pre-requisites\Datasets
>

```

## 2. S'assurer du bon transfert du fichier de données depuis votre machine au compte de l'utilisateur (student) dans la VM

Retourner à votre console de la VM HDP

exemple : `ssh student@192.168.50.128 -p 2222`

et Taper :

ls

**NB.** Dans la figure suivante, Remplacez `192.168.50.128` par l'URL (ou l'adresse IP) de votre VM HDP !!

```

student@sandbox-hdp:~
Microsoft Windows [version 10.0.19042.1348]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\workstation>ssh student@192.168.50.128 -p 2222
student@192.168.50.128's password:
Last login: Mon Nov 29 00:06:36 2021
[student@sandbox-hdp ~]$ ls
CDR.csv                               ChurnerCountByGenderReducer.py      CustomerCountByGenderReducer.py      spam-sms.csv
CDR.java                             city.csv                             Customer.csv                           spam-sms.py
ChurnerCountByGenderMapper.py         CustomerCountByGenderMapper.py      DE0121.signe.pdf                     tweets.json
[student@sandbox-hdp ~]$

```

## 3. Uploader ce fichier de données depuis Linux de la VM vers l'espace HDFS associé à l'utilisateur (student)

Sur votre console de la VM HDP, Taper :

### Besoin de création du répertoire data/cities

`hadoop fs -mkdir /user/student/data/cities/`

`hadoop fs -put '/home/student/city.csv' /user/student/data/cities/`

## 4. Afficher les informations à propos du fichier de données HDFS

`hadoop fs -ls data/cities/city.csv`

(oubien `hadoop fs -ls /user/student/data/cities/city.csv` vu que `/user/student/` est par défaut)

```

student@sandbox-hdp:~
student@192.168.50.128's password:
Last login: Mon Nov 29 00:06:36 2021
[student@sandbox-hdp ~]$ ls
CDR.csv                               ChurnerCountByGenderReducer.py      CustomerCountByGenderReducer.py      spam-sms.csv
CDR.java                             city.csv                             Customer.csv                           spam-sms.py
ChurnerCountByGenderMapper.py         CustomerCountByGenderMapper.py      DE0121.signe.pdf                     tweets.json
[student@sandbox-hdp ~]$ hadoop fs -mkdir /user/student/data/cities/
[student@sandbox-hdp ~]$ hadoop fs -put '/home/student/city.csv' /user/student/data/cities/
[student@sandbox-hdp ~]$ hadoop fs -ls data/cities/city.csv
-rw-r--r--  1 student student  151145261 2021-12-01 00:31 data/cities/city.csv
[student@sandbox-hdp ~]$

```

## 5. Exploration des données du fichier HDFS city.csv

`hadoop fs -cat /user/student/data/cities/city.csv | head -n 10`

```

student@sandbox-hdp:~$ hadoop fs -cat /user/student/data/cities/city.csv | head -n 10
ad,aixas,Aixs,06,,42.4833333,1.4666667
ad,aixirivali,Aixirivali,06,,42.4666667,1.5
ad,aixirivall,Aixirivall,06,,42.4666667,1.5
ad,aixirvall,Aixirvall,06,,42.4666667,1.5
ad,aixovall,Aixovall,06,,42.4666667,1.4833333
ad,andorra,Andorra,07,,42.5,1.5166667
ad,andorra la vella,Andorra la Vella,07,20430,42.5,1.5166667
ad,andorra-vieille,Andorra-Vieille,07,,42.5,1.5166667
ad,andorre,Andorre,07,,42.5,1.5166667
ad,andorre-la-vieille,Andorre-la-Vieille,07,,42.5,1.5166667
cat: Unable to write to output stream.
[student@sandbox-hdp ~]$

```

- Country\_Code : STRING (Code du pays)
- Name : STRING (Nom de la ville)
- AccentCity : STRING (Nom de la ville avec accents éventuels)
- Region : STRIN (région à laquelle appartient la ville)
- Population : INT (nombre d'habitants)
- Latitude : FLOAT (latitude)
- Longitude : FLOAT (longitude)

Source du dataset : Open data de : <https://www.maxmind.com>

## 6. Se logger en tant qu'utilisateur hdfs

su -

su - hdfs

## 7. Donner les privilèges sur le répertoire de données à l'utilisateur hive du groupe hdfs

hadoop fs -chown -R hive:hdfs /user/student/cities/

## 8. quitter la session hdfs

exit

## 9. quitter la session root

exit

```

C:\Users\workstation>ssh student@192.168.50.128 -p 2222
student@192.168.50.128's password:
Last login: Wed Dec  1 01:21:42 2021 from 172.18.0.3
[student@sandbox-hdp ~]$ su -
Password:
Last login: Wed Dec  1 01:21:51 UTC 2021 on pts/0
[root@sandbox-hdp ~]# su - hdfs
Last login: Wed Dec  1 01:21:56 UTC 2021 on pts/0
[hdfs@sandbox-hdp ~]$ hadoop fs -chown -R hive:hdfs /user/student/data/cities/
[hdfs@sandbox-hdp ~]$ exit
logout
[root@sandbox-hdp ~]# exit

```

**NB. Dans la figure précédente, Remplacez 192.168.50.128 par l'URL (ou l'adresse IP) de votre VM HDP !!**

## 6. Lancer le shell Hive CLI de datawarehouse Hive

/usr/bin/hive shell

```
[student@sandbox-hdp ~]$ /usr/bin/hive shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zooKeeperNamespace=hiveserver2
21/09/29 09:44:17 [main]: INFO jdbc.HiveConnection: Connected to sandbox-hdp.hortonworks.com:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

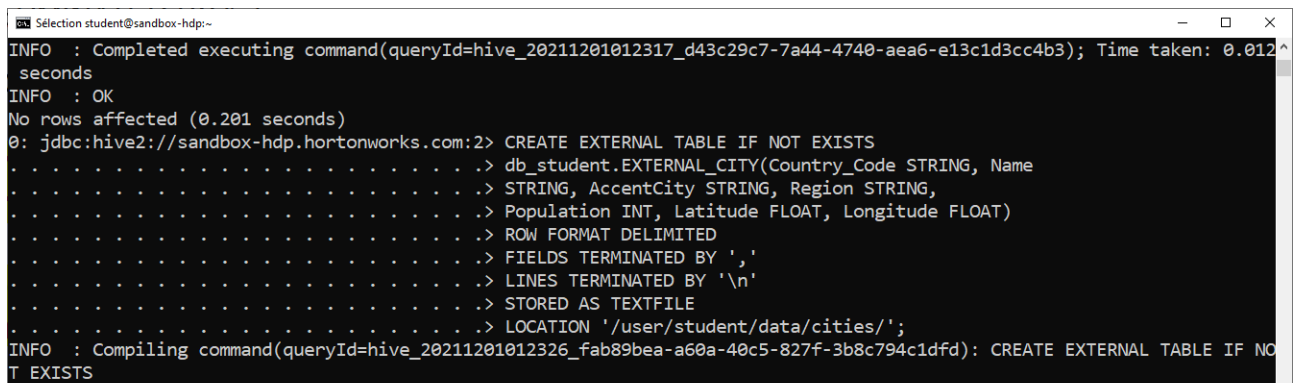
## 7. Se Positionner sur la base de données du compte courant (student)

use db\_student;

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> use db_student ;
INFO : Compiling command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607): use db_student
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607); Time taken: 0.026 seconds
INFO : Executing command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607): use db_student
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210929095231_9119d209-1801-4c9d-a192-08f086f8b607); Time taken: 0.015 seconds
INFO : OK
No rows affected (0.053 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

## 7. Créer une table Externe (non managée) Hive EXTERNAL\_CITY

```
CREATE EXTERNAL TABLE IF NOT EXISTS
db_student.EXTERNAL_CITY(Country_Code STRING, Name
STRING, AccentCity STRING, Region STRING,
Population INT, Latitude FLOAT, Longitude FLOAT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/student/data/cities/';
```



```
Sélection student@sandbox-hdp:~
INFO : Completed executing command(queryId=hive_20211201012317_d43c29c7-7a44-4740-aea6-e13c1d3cc4b3); Time taken: 0.012^
seconds
INFO : OK
No rows affected (0.201 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> CREATE EXTERNAL TABLE IF NOT EXISTS
. . . . .> db_student.EXTERNAL_CITY(Country_Code STRING, Name
. . . . .> STRING, AccentCity STRING, Region STRING,
. . . . .> Population INT, Latitude FLOAT, Longitude FLOAT)
. . . . .> ROW FORMAT DELIMITED
. . . . .> FIELDS TERMINATED BY ','
. . . . .> LINES TERMINATED BY '\n'
. . . . .> STORED AS TEXTFILE
. . . . .> LOCATION '/user/student/data/cities/';
INFO : Compiling command(queryId=hive_20211201012326_fab89bea-a60a-40c5-827f-3b8c794c1dfd): CREATE EXTERNAL TABLE IF NO
T EXISTS
```

## 8. Vérifier la création de la table Hive CDR dans la base de données Hive associée à l'utilisateur courant (student)

show tables;

```

student@sandbox-hdp:~
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> show tables;
INFO : Compiling command(queryId=hive_20211201012823_62bafb7c-fff6-44af-8fee-e9a8160835b0): show tables
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)],
properties:null)
INFO : Completed compiling command(queryId=hive_20211201012823_62bafb7c-fff6-44af-8fee-e9a8160835b0); Time taken: 0.028
seconds
INFO : Executing command(queryId=hive_20211201012823_62bafb7c-fff6-44af-8fee-e9a8160835b0): show tables
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20211201012823_62bafb7c-fff6-44af-8fee-e9a8160835b0); Time taken: 0.012
seconds
INFO : OK
+-----+
| tab_name |
+-----+
| cdr      |
| customer |
| employee |
| external_city |
| smswordcount |
+-----+
5 rows selected (0.238 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>

```

## 9. Vérifier le schéma de la table external\_city dans Hive

describe db\_student.external\_city;

```

Selection student@sandbox-hdp:~
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> describe db_student .external_city;
INFO : Compiling command(queryId=hive_20211201012944_949ad9bd-45ef-459e-a082-0321d4e5c4e1): describe db_student .extern
al_city
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer),
FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comment, type:string, comment:from
deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20211201012944_949ad9bd-45ef-459e-a082-0321d4e5c4e1); Time taken: 0.073
seconds
INFO : Executing command(queryId=hive_20211201012944_949ad9bd-45ef-459e-a082-0321d4e5c4e1): describe db_student .extern
al_city
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20211201012944_949ad9bd-45ef-459e-a082-0321d4e5c4e1); Time taken: 0.046
seconds
INFO : OK
+-----+-----+-----+
| col_name | data_type | comment |
+-----+-----+-----+
| country_code | string   |         |
| name        | string   |         |
| accentcity   | string   |         |
| region       | string   |         |
| population   | int      |         |
| latitude     | float    |         |
| longitude    | float    |         |
+-----+-----+-----+
7 rows selected (0.159 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>

```

## 10. Explorer les données de la table external\_city à travers quelques requêtes Hive

```

SELECT Country_Code, sum(Population) FROM db_student.external_city
WHERE Country_Code='br'
GROUP BY Country_Code ;

```

```
student@sandbox-hdp:~
INFO : TaskCounter_Reducer_2_OUTPUT_out_Reducer_2:
INFO : OUTPUT_RECORDS: 0
INFO : org.apache.hadoop.hive.q1.exec.tez.HiveInputCounters:
INFO : GROUPED_INPUT_SPLITS_Map_1: 3
INFO : INPUT_DIRECTORIES_Map_1: 1
INFO : INPUT_FILES_Map_1: 1
INFO : RAW_INPUT_SPLITS_Map_1: 3
INFO : Completed executing command(queryId=hive_20211201013416_5a43f79d-ef8b-441e-9838-336b768f3fc4); Time taken: 27.07
7 seconds
INFO : OK
+-----+-----+
| country_code | _c1 |
+-----+-----+
| br           | 133449921 |
+-----+-----+
1 row selected (31.749 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

SELECT Country\_Code, SUM(Population) AS somme  
FROM db\_student.external\_city GROUP BY Country\_Code HAVING  
somme >= 100000000 ORDER BY somme DESC;

```
student@sandbox-hdp:~
INFO : INPUT_FILES_Map_1: 1
INFO : RAW_INPUT_SPLITS_Map_1: 3
INFO : Completed executing command(queryId=hive_20211201013525_d3c7a71c-f398-4c9a-81e7-55ec00ab114a); Time taken: 14.07
2 seconds
INFO : OK
+-----+-----+
| country_code | somme |
+-----+-----+
| in           | 259227307 |
| cn           | 218884084 |
| us           | 179123400 |
| br           | 133449921 |
| ru           | 109505345 |
| jp           | 101577008 |
+-----+-----+
6 rows selected (14.484 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

## 11. Créer une table partitionnée partitioned\_city dans Hive

```
CREATE TABLE IF NOT EXISTS
db_student.PARTITIONED_CITY(Name STRING, AccentCity
STRING, Region STRING, Population INT,
Latitude FLOAT, Longitude FLOAT)
PARTITIONED BY (Country_Code STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE ;
```

```
student@sandbox-hdp:~
6 rows selected (14.484 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> CREATE TABLE IF NOT EXISTS
. . . . .> db_student.PARTITIONED_CITY(Name STRING, AccentCity
. . . . .> STRING, Region STRING, Population INT,
. . . . .> Latitude FLOAT, Longitude FLOAT)
. . . . .> PARTITIONED BY (Country_Code STRING)
. . . . .> ROW FORMAT DELIMITED
. . . . .> FIELDS TERMINATED BY ','
. . . . .> LINES TERMINATED BY '\n'
. . . . .> STORED AS TEXTFILE ;
INFO : Compiling command(queryId=hive_20211201013825_7c5a132d-1953-4189-acba-1bf2f1efb1f3): CREATE TABLE IF NOT EXISTS
```

**11. Créer la partition statique (i.e. relative à une valeur donnée de la clé de partitionnement (Country\_Code STRING) – un pays donné) dans la table Hive PARTITIONED\_CITY et l'alimenter depuis la table externe Hive EXTERNAL\_CITY**

```
INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code = 'ma')
SELECT Name, AccentCity, Region, Population, Latitude,
Longitude
FROM db_student.EXTERNAL_CITY
WHERE Country_Code = 'ma';
```

```
student@sandbox-hdp:~
INFO : OK
No rows affected (0.513 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> INSERT INTO db_student.PARTITIONED_CITY
. . . . .> PARTITION (Country_Code = 'ma')
. . . . .> SELECT Name, AccentCity, Region, Population, Latitude,
. . . . .> Longitude
. . . . .> FROM db_student.EXTERNAL_CITY
. . . . .> WHERE Country_Code = 'ma';
INFO : Compiling command(queryId=hive_20211201014319_fc27dcb0-2aa2-4a73-ba12-39d9ed12c6f0): INSERT INTO db_student.PARTITIONED_CITY
```

```
select * from db_student.partitioned_city where Name ='casablanca' ;
```

```
student@sandbox-hdp:~
+-----+-----+-----+-----+-----+-----+
| partitioned_city.name | partitioned_city.accentcity | partitioned_city.region | partitioned_city.population | partitioned_city.latitude | partitioned_city.longitude | partitioned_city.country_code |
+-----+-----+-----+-----+-----+-----+
| casablanca            | Casablanca                  | 45                      | 3609698                    | 33.592777                | -7.619157                  | ma                             |
+-----+-----+-----+-----+-----+-----+
1 row selected (0.347 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

**12. Quitter Hive CLI momentanément**

```
!q
```

**13. Explorer la structure de stockage HDFS réalisé par Hive pour la table partitionnée partitioned\_city**

```
hadoop fs -ls /warehouse/tablespace/managed/hive/db_student.db/partitioned_city
```

NB. Sous les versions (2.6/2.7 sous Azure), tapez :

```
hadoop fs -ls /apps/hive/warehouse/db_student.db/partitioned_city
```

```
student@sandbox-hdp:~
[student@sandbox-hdp ~]$ hadoop fs -ls /warehouse/tablespace/managed/hive/db_student.db/partitioned_city
Found 1 items
drwxr-xr-x+ - hive hadoop      0 2021-12-01 01:43 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ma
[student@sandbox-hdp ~]$
```

Remarque : Particularité des tables partitionnées. Différemment des tables normales stockées sous HDFS comme des buckets plats dans une sous répertoire portant le même nom de la table sous /warehouse/tablespace/managed/hive/, pour la table managée partitionnée **db\_student.db/partitioned\_city**, Hive crée une arborescence des niveaux multi-dimensionnel imbriqués de répertoire HDFS (ici une seule dimension/niveau le Country\_Code) sur son /warehouse/tablespace/managed/hive/ pour les clefs de partitions (entant q'un pur Big data datawarehouse).

NB. Sous les versions (2.6/2.7 sous Azure), les données partitionnées sont sur /apps/hive/warehouse/db\_student.db/partitioned\_city.

**14. Revenir sur le shell Hive CLI de datawarehouse Hive**

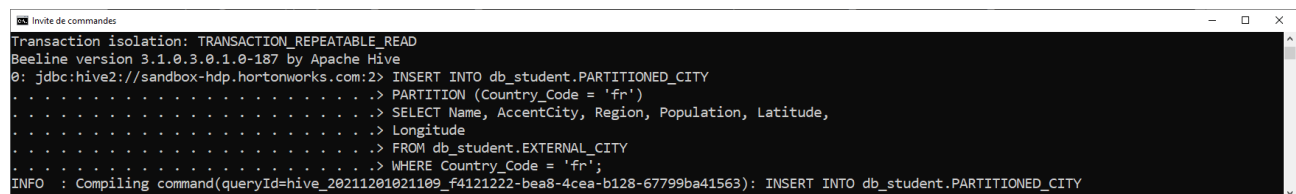


```
/usr/bin/hive shell
```

```
[student@sandbox-hdp ~]$ /usr/bin/hive shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zooKeeperNamespace=hiveserver2
21/09/29 09:44:17 [main]: INFO jdbc.HiveConnection: Connected to sandbox-hdp.hortonworks.com:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> █
```

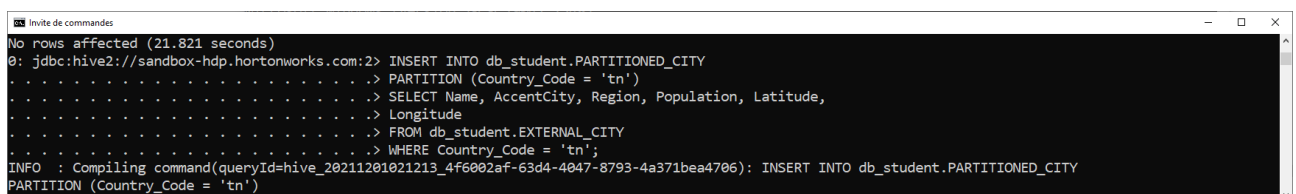
**15. Créer deux nouvelles partitions statiques (i.e. relative à des valeurs données de la clé de partitionnement (Country\_Code STRING) – des pays donné) dans la table Hive PARTITIONED\_CITY et les alimenter depuis la table externe Hive EXTERNAL\_CITY**

```
INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code = 'fr')
SELECT Name, AccentCity, Region, Population, Latitude,
Longitude
FROM db_student.EXTERNAL_CITY
WHERE Country_Code = 'fr';
```



```
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> INSERT INTO db_student.PARTITIONED_CITY
. . . . .-> PARTITION (Country_Code = 'fr')
. . . . .-> SELECT Name, AccentCity, Region, Population, Latitude,
. . . . .-> Longitude
. . . . .-> FROM db_student.EXTERNAL_CITY
. . . . .-> WHERE Country_Code = 'fr';
INFO : Compiling command(queryId=hive_20211201021109_f4121222-bea8-4cea-b128-67799ba41563): INSERT INTO db_student.PARTITIONED_CITY
```

```
INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code = 'tn')
SELECT Name, AccentCity, Region, Population, Latitude,
Longitude
FROM db_student.EXTERNAL_CITY
WHERE Country_Code = 'tn';
```



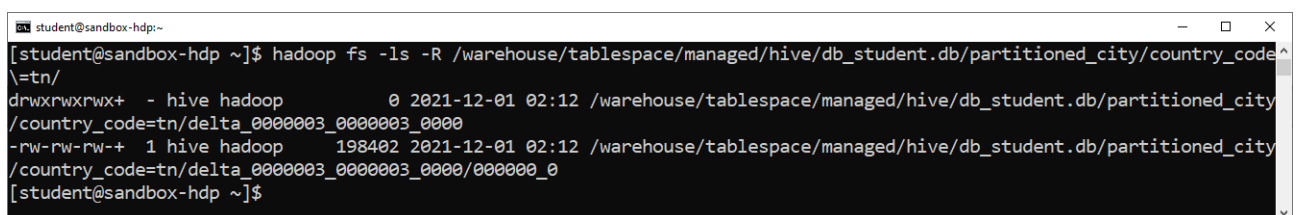
```
No rows affected (21.821 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> INSERT INTO db_student.PARTITIONED_CITY
. . . . .-> PARTITION (Country_Code = 'tn')
. . . . .-> SELECT Name, AccentCity, Region, Population, Latitude,
. . . . .-> Longitude
. . . . .-> FROM db_student.EXTERNAL_CITY
. . . . .-> WHERE Country_Code = 'tn';
INFO : Compiling command(queryId=hive_20211201021213_4f6002af-63d4-4047-8793-4a371bea4706): INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code = 'tn')
```

**16. Explorer plus en profondeur la structure et le contenu du stockage réalisé par Hive pour la table partitionnée partitioned\_city sur HDFS**

```
hadoop fs -ls -R /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code\=tn/
```

NB. Sous les versions (2.6/2.7 sous Azure), tapez :

```
hadoop fs -ls /apps/hive/warehouse/db_student.db/partitioned_city/country_code\=tn/
```



```
student@sandbox-hdp:~$ hadoop fs -ls -R /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code\=tn/
drwxrwxrwx+ - hive hadoop 0 2021-12-01 02:12 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=tn/delta_0000003_0000003_0000
-rw-rw-rw-+ 1 hive hadoop 198402 2021-12-01 02:12 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=tn/delta_0000003_0000003_0000/000000_0
[student@sandbox-hdp ~]$
```

```
hadoop fs -cat /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code\=tn/delta_0000003_0000003_0000/000000_0 | head -n 10
```

NB. Sous les versions (2.6/2.7 sous Azure), tapez :

`hadoop fs -cat /app/shive/warehouse/db_student.db/partitioned_city/country_code\=tn/..... | head -n 10`  
 Le nom du bucket dépendra ainsi de votre exécution du partitionnement statique.

**NB.** Dans la commande précédente Mettre les bons sous répertoires de la partition choisie !

```
student@sandbox-hdp:~$ hadoop fs -cat /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code\=tn/delta_0000003_0000003_0000/000000_0 | head -n 10
abarda,Abarda,28,\N,33.779213,10.768174
abbah qusur,Abbah Qusur,14,\N,35.94388,8.827776
`abbas,`Abbas,35,\N,33.896503,8.135551
abbes,Abbes,35,\N,33.896503,8.135551
abd er rahmane,Abd er Rahmane,23,\N,36.15,10.25
abd er rahmane el garsi zaouia,Abd er Rahmane el Garsi Zaoua,23,\N,36.15,10.25
abes,Abes,29,\N,33.88146,10.098196
abzeg,Abzeg,32,\N,35.028053,10.960506
ad dabadab,Ad Dabadab,29,\N,33.872475,9.797324
ad dakhilah,Ad Dakhilah,18,\N,36.86667,9.7
cat: Unable to write to output stream.
[student@sandbox-hdp ~]$
```

**17. Créer des partitions dynamiques (i.e. relatives à toutes les valeurs de la clé de partitionnement (Country\_Code STRING) – des pays restant non encore créé) dans la table Hive PARTITIONED\_CITY et les alimenter depuis la table externe Hive EXTERNAL\_CITY**

```
INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code)
SELECT Name, AccentCity, Region, Population, Latitude,
Longitude, Country_Code
FROM db_student.EXTERNAL_CITY
WHERE Country_Code not in('fr', 'ma', 'tn');
```

Créer des partitions dynamiques ressemble à invoquer une sorte d'itérateur !

NB. Sous les versions (2.6/2.7 sous Azure), il faudra forcer des variables de partitionnement dynamique :

```
SET hive.exec.dynamic.partition = true;
SET hive.exec.dynamic.partition.mode = nonstrict;
SET hive.exec.max.dynamic.partitions = 1000;
SET hive.exec.max.dynamic.partitions.pernode = 1000;
```

#les 4 propriétés peuvent être configurées également dans hive-site.xml

Puis refaire

```
INSERT INTO db_student.PARTITIONED_CITY
PARTITION (Country_Code)
SELECT Name, AccentCity, Region, Population, Latitude,
Longitude, Country_Code
FROM db_student.EXTERNAL_CITY
WHERE Country_Code not in('fr', 'ma', 'tn');
```

```
student@sandbox-hdp:~$ jdb:hive2://sandbox-hdp.hortonworks.com:2> INSERT INTO db_student.PARTITIONED_CITY
...> PARTITION (Country_Code)
...> SELECT Name, AccentCity, Region, Population, Latitude,
...> Longitude, Country_Code
...> FROM db_student.EXTERNAL_CITY
...> WHERE Country_Code not in('fr', 'ma', 'tn');
```

Normalement, la commande devra réussir. Cependant, sous les versions (2.6/2.7 sous Azure), d'autres problèmes de répliquions surgissent. Caused by: org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /apps/hive/warehouse/db\_student.db/partitioned\_city/.hive-staging\_hive\_2021-12-02\_13-33-27\_937\_8827931192441516354-1/\_task\_tmp.-ex t-10000/country\_code=ca/\_tmp.000001\_3 could only be replicated to 0 nodes instead of minReplication (=1). There are 1 datanode(s) running and no node(s) are excluded in this operation.

## 18. Explorer les données de la table external\_city à travers quelques requêtes Hive

show partitions db\_student.partitioned\_city LIMIT 15;

...

OK

- country\_code=ad
- country\_code=ae
- country\_code=af
- country\_code=ag
- country\_code=ai
- country\_code=al
- ...

```

student@sandbox-hdp-~
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> show partitions db_student.partitioned_city;
INFO : Compiling command(queryId=hive_20211201023919_993a3911-3780-4273-8bc8-cc5f6a4f173e): show partitions db_student.partitioned_city
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:partition, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20211201023919_993a3911-3780-4273-8bc8-cc5f6a4f173e); Time taken: 0.197 seconds
INFO : Executing command(queryId=hive_20211201023919_993a3911-3780-4273-8bc8-cc5f6a4f173e): show partitions db_student.partitioned_city
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20211201023919_993a3911-3780-4273-8bc8-cc5f6a4f173e); Time taken: 0.054 seconds
INFO : OK

+-----+
| partition |
+-----+
| country_code=ad |
| country_code=ae |
| country_code=af |
| country_code=ag |
| country_code=ai |
| country_code=al |
| country_code=am |
| country_code=an |
| country_code=ao |
| country_code=ar |
| country_code=at |
| country_code=au |
| country_code=aw |
| country_code=az |
| country_code=ba |

```

DESCRIBE FORMATTED db\_student.partitioned\_city  
PARTITION (country\_code='us');

```

student@sandbox-hdp-~
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> DESCRIBE FORMATTED db_student.partitioned_city
PARTITION (country_code='us');
INFO : Compiling command(queryId=hive_20211201024133_a53db0d6-aa79-48cf-8c0f-014c3f9f98c2): DESCRIBE FORMATTED db_student.partitioned_city
PARTITION (country_code='us')
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20211201024133_a53db0d6-aa79-48cf-8c0f-014c3f9f98c2); Time taken: 0.166 seconds
INFO : Executing command(queryId=hive_20211201024133_a53db0d6-aa79-48cf-8c0f-014c3f9f98c2): DESCRIBE FORMATTED db_student.partitioned_city
PARTITION (country_code='us')
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20211201024133_a53db0d6-aa79-48cf-8c0f-014c3f9f98c2); Time taken: 0.093 seconds
INFO : OK

+-----+-----+-----+
| col_name | data_type | comment |
+-----+-----+-----+
| # col_name | data_type | comment |
| name | string | |
| accentcity | string | |
| region | string | |
| population | int | |
| latitude | float | |
| longitude | float | |
| # Partition Information | NULL | NULL |
| # col_name | data_type | comment |
| country_code | string | |
| | NULL | NULL |
| # Detailed Partition Information | NULL | NULL |

```

select \* from db\_student.partitioned\_city  
where Country\_Code = 'fr' LIMIT 5;

```

student@sandbox-hdp:~$
39 rows selected (0.307 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> select * from db_student.partitioned_city
where Country_Code = 'fr' LIMIT 5;
INFO : Compiling command(queryId=hive_20211201024304_3cc8c2dd-b1ec-40af-89ef-92735584522b): select * from db_student.partitioned_city
where Country_Code = 'fr' LIMIT 5
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:partitioned_city.name, type:string, comment:null), FieldSchema(name:partitioned_city.accentcity, type:string, comment:null), FieldSchema(name:partitioned_city.region, type:string, comment:null), FieldSchema(name:partitioned_city.population, type:int, comment:null), FieldSchema(name:partitioned_city.latitude, type:float, comment:null), FieldSchema(name:partitioned_city.longitude, type:float, comment:null), FieldSchema(name:partitioned_city.country_code, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=hive_20211201024304_3cc8c2dd-b1ec-40af-89ef-92735584522b); Time taken: 0.628 seconds
INFO : Executing command(queryId=hive_20211201024304_3cc8c2dd-b1ec-40af-89ef-92735584522b): select * from db_student.partitioned_city
where Country_Code = 'fr' LIMIT 5
INFO : Completed executing command(queryId=hive_20211201024304_3cc8c2dd-b1ec-40af-89ef-92735584522b); Time taken: 0.009 seconds
INFO : OK

+-----+-----+-----+-----+-----+-----+
| partitioned_city.name | partitioned_city.accentcity | partitioned_city.region | partitioned_city.population | partitioned_city.latitude | partitioned_city.longitude | partitioned_city.country_code |
+-----+-----+-----+-----+-----+-----+
| aas | fr | Aas | 97 | NULL | 42.983334 | -0.4 |
| aast | fr | Aast | 97 | NULL | 43.283333 | -0.083333 |
| abacourt | fr | Abacourt | 84 | NULL | 50.234695 | 3.212651 |
| abainville | fr | Abainville | 82 | NULL | 48.530514 | 5.494462 |
| abancourt | fr | Abancourt | A7 | NULL | 49.591 | 1.623268 |
+-----+-----+-----+-----+-----+-----+

```

```

select * from db_student.partitioned_city
where Country_Code = 'it' LIMIT 5;

```

```

student@sandbox-hdp:~$
INFO : Compiling command(queryId=hive_20211201024351_51e199b7-0a59-4189-965a-523b9980fce6): select * from db_student.partitioned_city
where Country_Code = 'it' LIMIT 5
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:partitioned_city.name, type:string, comment:null), FieldSchema(name:partitioned_city.accentcity, type:string, comment:null), FieldSchema(name:partitioned_city.region, type:string, comment:null), FieldSchema(name:partitioned_city.population, type:int, comment:null), FieldSchema(name:partitioned_city.latitude, type:float, comment:null), FieldSchema(name:partitioned_city.longitude, type:float, comment:null), FieldSchema(name:partitioned_city.country_code, type:string, comment:null)], properties:null)
INFO : Completed compiling command(queryId=hive_20211201024351_51e199b7-0a59-4189-965a-523b9980fce6); Time taken: 0.243 seconds
INFO : Executing command(queryId=hive_20211201024351_51e199b7-0a59-4189-965a-523b9980fce6): select * from db_student.partitioned_city
where Country_Code = 'it' LIMIT 5
INFO : Completed executing command(queryId=hive_20211201024351_51e199b7-0a59-4189-965a-523b9980fce6); Time taken: 0.007 seconds
INFO : OK

+-----+-----+-----+-----+-----+-----+
| partitioned_city.name | partitioned_city.accentcity | partitioned_city.region | partitioned_city.population | partitioned_city.latitude | partitioned_city.longitude | partitioned_city.country_code |
+-----+-----+-----+-----+-----+-----+
| abadia a isola | it | Abadia a Isola | 16 | NULL | 43.38333 | 11.2 |
| abano | it | Abano | 20 | NULL | 45.35 | 11.783333 |
| abano terme | it | Abano Terme | 20 | 19112 | 45.35 | 11.783333 |
| abatemarco | it | Abatemarco | 04 | NULL | 40.13333 | 15.35 |
| abbadia | it | Abbadia | 10 | NULL | 43.216667 | 13.4 |
+-----+-----+-----+-----+-----+-----+
5 rows selected (0.293 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>

```

## 19. Quitter Hive CLI momentanément

```
!q
```

## 20. Explorer la structure de stockage HDFS réalisé par Hive pour la table partitionnée partitioned\_city

```
hadoop fs -ls /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/ | head -n 15
```

```

student@sandbox-hdp:~$
5 rows selected (0.293 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> !q
Closing: 0: jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zooKeeperNamespace=hiveserver2
[student@sandbox-hdp ~]$ hadoop fs -ls /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/ | head -n 15
Found 234 items
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ad
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ae
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=af
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ag
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ai
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=al
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=am
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=an
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ao
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=ar
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=at
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=au
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=aw
drwxrwxrwx+ - hive hadoop      0 2021-12-01 02:29 /warehouse/tablespace/managed/hive/db_student.db/partitioned_city/country_code=az
[student@sandbox-hdp ~]$

```

**Question.** Que pensez-vous des trois styles de développement, test et déploiement de flux de données (dataflow) d'analyse de données Hadoop :

- *Jobs Map Reduce (Programmation impérative du pattern MR sous Python et HadoopStreaming) en Atelier 4 ?*
- *Applications Spark (Programmation fonctionnelle du pattern MR, Spark SQL sous PySpark et spark-submit) en Atelier 5 ?*
- *Requêtage Hive QL (sous Hive datawarehouse store) en Atelier 6 ? Bien qu'il n'y ait pas vraiment de comparaison objective entre un dataflow language (Python) sous deux moteurs d'exécution différentes (mapred avec une communication M&R on disk et spark avec une communication M&R in memory) et un data store non comparables du fait que chacun à son propre rôle et qu'ils se complètent vraiment ?*

**Résultats :** Cet atelier vous a permis d'Obtenir des informations sur les services Hive à l'aide de l'interface utilisateur Web Ambari, d'Utiliser la CLI Hive pour créer des tables Hive, importer des données dans Hive et interroger des données sur Hive.