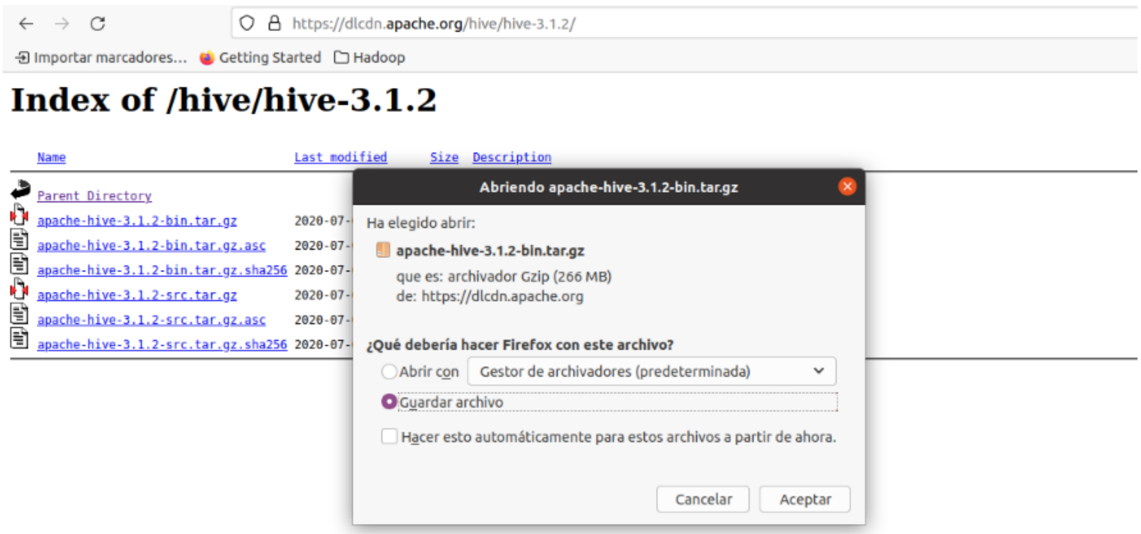
6. HIVE

6.1. Instalación

Descargar a partir del sitio https://hive.apache.org/downloads.html. Verificar la versión de HIVE compatible con HADOOP, en este ejemplo, a versión de Hive 3.1.2



Index of /hive/hive-3.1.2

Name	Last modified	Size	Description
Parent Directory			
apache-hive-3.1.2-bin.tar.gz	2020-07-		
apache-hive-3.1.2-bin.tar.gz.asc	2020-07-		
apache-hive-3.1.2-bin.tar.gz.sha256	2020-07-		
apache-hive-3.1.2-src.tar.gz	2020-07-		
apache-hive-3.1.2-src.tar.gz.asc	2020-07-		
apache-hive-3.1.2-src.tar.gz.sha256	2020-07-		

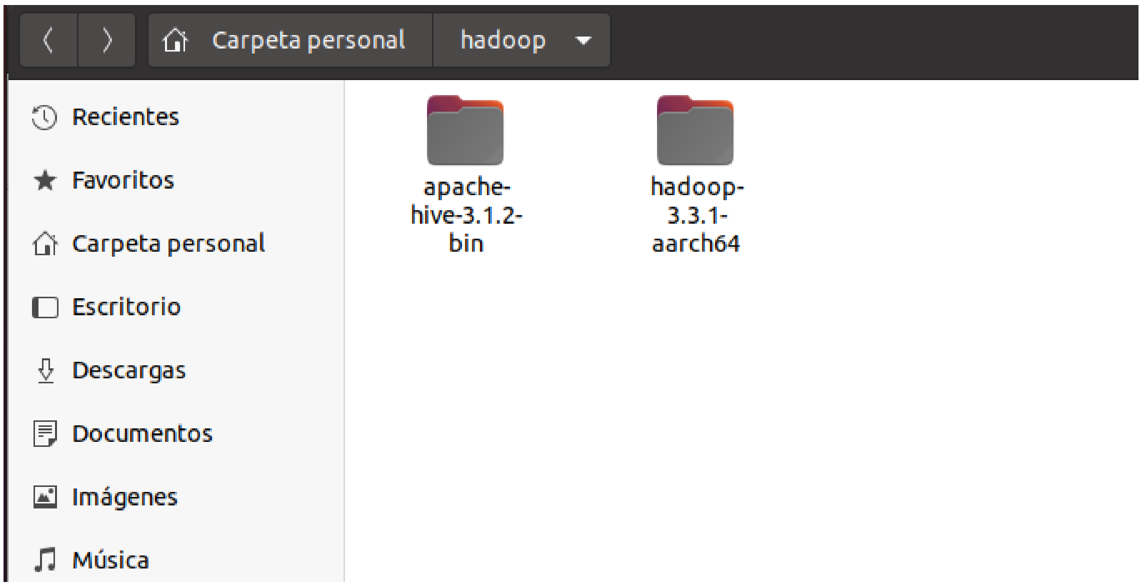
Abriendo apache-hive-3.1.2-bin.tar.gz

Ha elegido abrir:
apache-hive-3.1.2-bin.tar.gz
que es: archivador Gzip (266 MB)
de: https://dlcdn.apache.org

¿Qué debería hacer Firefox con este archivo?
Abrir con: Gestor de archivadores (predeterminada)
[✓ Guardar archivo]
Hacer esto automáticamente para estos archivos a partir de ahora.

Cancelar Aceptar

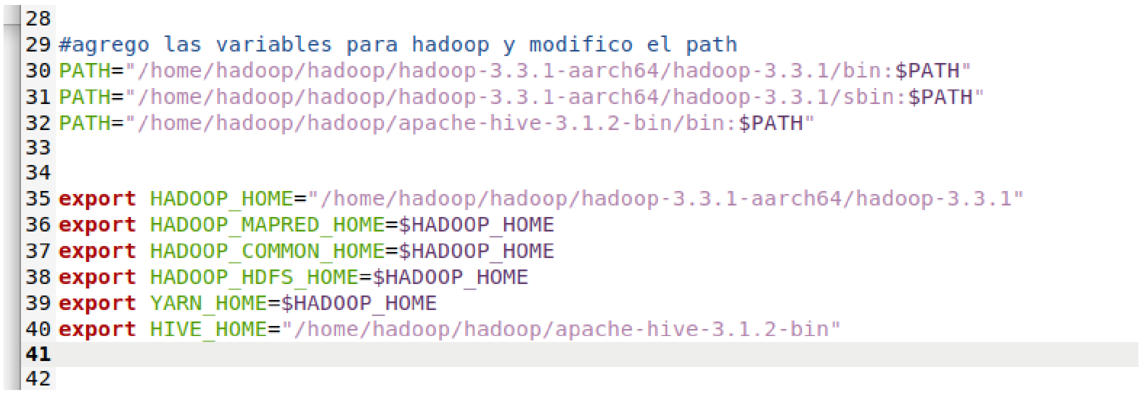
Descomprimir el archivo (en el ejemplo utilizo la carpeta hdoop)



Crear la variable de entorno HIVE_HOME y agregarla al PATH. Abrir el archivo de configuración "~/.profile" y agregar:

```
PATH="/home/hadoop/hadoop/apache-hive-3.1.2-bin/bin:$PATH"

export HIVE_HOME="/home/hadoop/hadoop/apache-hive-3.1.2-bin"
```

Quedando el fichero ~/.profile de la siguiente manera.

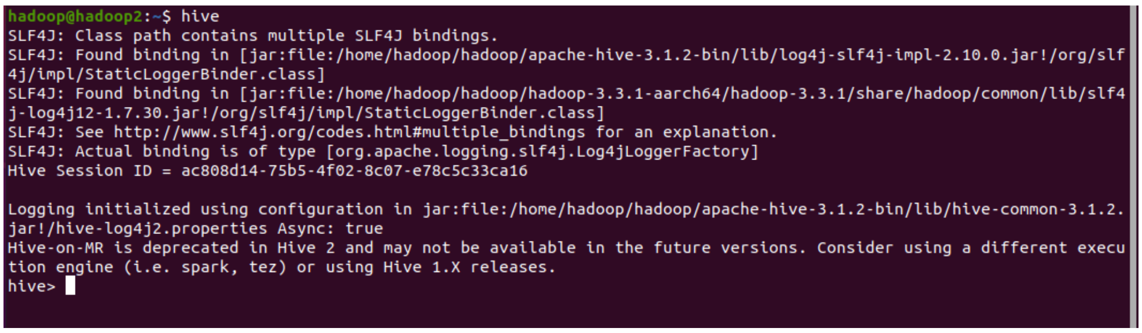


```

#agrego las variables para hadoop y modifico el path
PATH="/home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/bin:$PATH"
PATH="/home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/sbin:$PATH"
PATH="/home/hadoop/hadoop/apache-hive-3.1.2-bin/bin:$PATH"


export HADOOP_HOME="/home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1"
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HIVE_HOME="/home/hadoop/hadoop/apache-hive-3.1.2-bin"


```

Reiniciar session para que se carguen las nuevas variables de entorno y se actualice el path

Iniciar el shell de Hive con

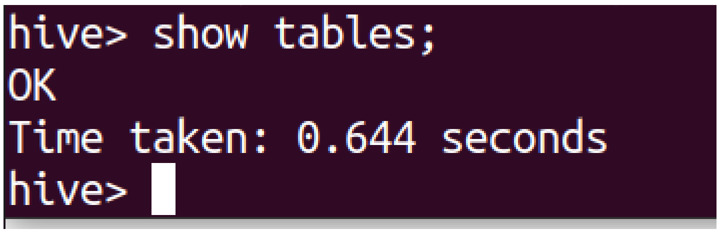
```
$ hive
```



```
hadoop@hadoop2:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/share/hadoop/common/lib/slf4
j-log4j12-1.7.30.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]
Hive Session ID = ac808d14-75b5-4f02-8c07-e78c5c33ca16

Logging initialized using configuration in jar:file:/home/hadoop/hadoop/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.
jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
hive>
```

Verificar funcionamiento

```
hive> show tables hiv;
```



```
hive> show tables;
OK
Time taken: 0.644 seconds
hive>
```

En el caso de error, reiniciar la base de datos desde el fuera del Shell de hive y volver a ejecutar

```
$ rm -rf metastore_db/

$ $HIVE_HOME/bin/schematool -initSchema -dbType derby
```

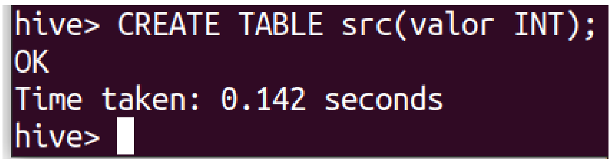
6.2. Funcionamiento básico

Además del Shell de Hive CLI tenemos otro Shell BeeLine, que se basa en el anterior pero basado en conexión JDBCexit

6.2.1. Creación de una tabla de ejemplo

Crear una tabla basada en el fichero /examples/test.dat

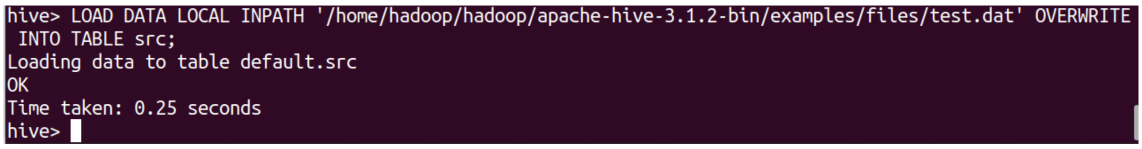
```
CREATE TABLE src(valor INT);
```



```
hive> CREATE TABLE src(valor INT);
OK
Time taken: 0.142 seconds
hive>
```

Cargar los datos desde el fichero test.dats

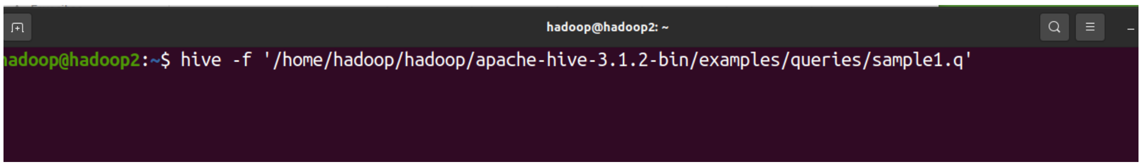
```
LOAD DATA LOCAL INPATH '/home/hadoop/hadoop/apache-hive-3.1.2-
bin/examples/files/test.dat' OVERWRITE INTO TABLE src;
```



```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/hadoop/apache-hive-3.1.2-bin/examples/files/test.dat' OVERWRITE
 INTO TABLE src;
Loading data to table default.src
OK
Time taken: 0.25 seconds
hive>
```

6.2.2. Ejecutar instrucciones

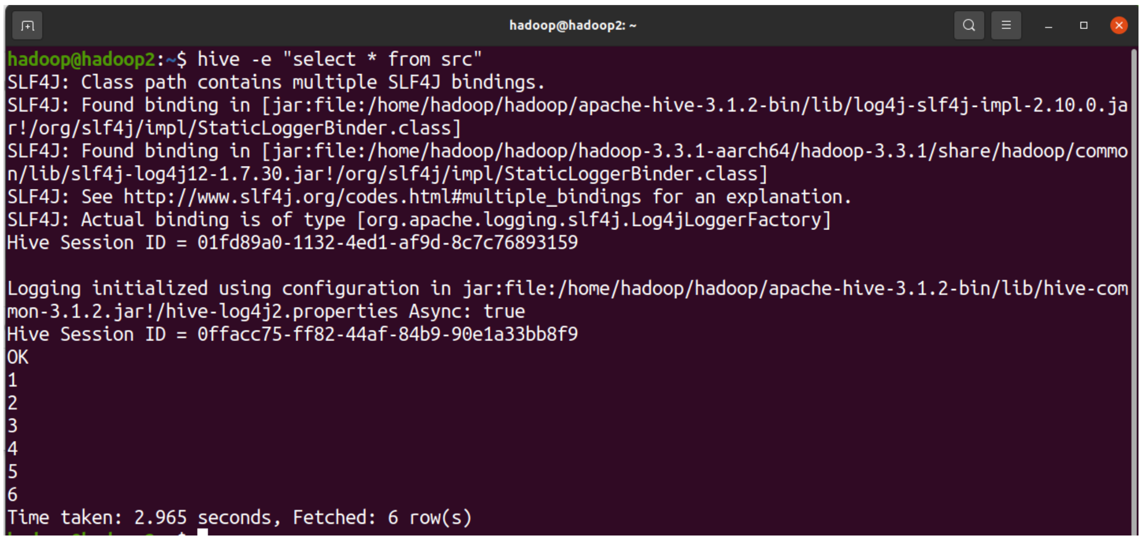
- -f permite ejecutar los comandos desde un archivo



```
adoop@hadoop2:~$ hive -f '/home/hadoop/hadoop/apache-hive-3.1.2-bin/examples/queries/sample1.q'
```

- -e , permite ejecutar el comando sin iniciar el Shell

```
hive -e "select * from src"
```



```
hadoop@hadoop2:~$ hive -e "select * from src"
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.ja
r!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/share/hadoop/commo
n/lib/slf4j-log4j12-1.7.30.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]
Hive Session ID = 01fd89a0-1132-4ed1-af9d-8c7c76893159

Logging initialized using configuration in jar:file:/home/hadoop/hadoop/apache-hive-3.1.2-bin/lib/hive-com
mon-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 0ffacc75-ff82-44af-84b9-90e1a33bb8f9
OK
1
2
3
4
5
6
Time taken: 2.965 seconds, Fetched: 6 row(s)
```

Notas sintaxis HiveQL :

- Las instrucciones no son casesensitive
- Las sentencias tienen que finalizar con “;”
- La tecla TAB completa la sintaxis

6.3. Utilización de tablas externas

Para este ejemplo tenemos varios archivos con los campos separados por # y con la siguiente estructura

movies:

Estructura: id#name#genre

Ejemplo :
1#Toy Story (1995)#Animation|Children's|Comedy
2#Jumanji (1995)#Adventure|Children's|Fantasy
3#Grumpier Old Men (1995)#Comedy|Romance

users:

Estructura:id#gender#age#occupationid#zipcode

Ejemplo:
1#F#1#10#48067
2#M#56#16#70072
3#M#25#15#55117

ratings:

estructura:userid#movieid#rating#tmstmp

Ejemplo:
1#1193#5#978300760
1#661#3#978302109
1#914#3#978301968

occupations:

Estructura:id#category#place

Ejemplo:
0#other/not specified
1#academic/educator
2#artist

6.3.1. Creo la estructura y copio los ficheros al sistema HDFS

Creo la estructura en HDFS

```
$ hdfs dfs -mkdir /hive

$ hdfs dfs -mkdir /hive/data

$ hdfs dfs -mkdir /hive/data/occupation
$ hdfs dfs -mkdir /hive/data/movie

$ hdfs dfs -mkdir /hive/data/rating

$ hdfs dfs -mkdir /hive/data/user
```

Copio los archivos (ejemplo suponiendo que estoy en el mismo directorio)

```
$ hdfs dfs -put occupations.dat /hive/data/occupation

$ hdfs dfs -put movies.dat /hive/data/movie

$ hdfs dfs -put ratings.dat /hive/data/rating

$ hdfs dfs -put users.dat /hive/data/user
```

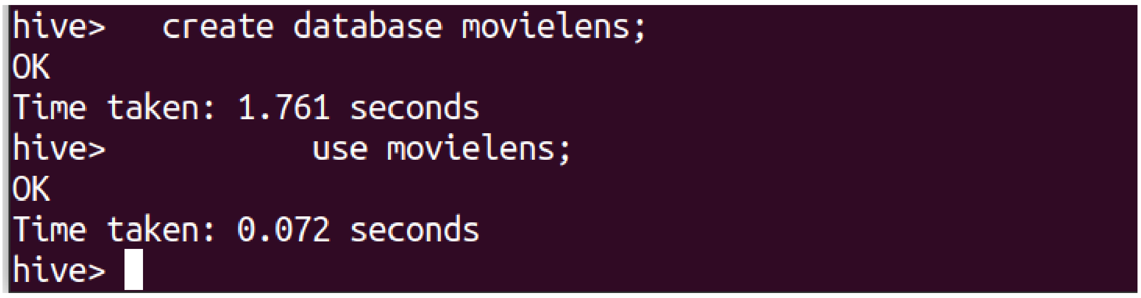
Teniendo un resultado similar a:

6.3.2. Creación de la base de datos y tablas

A partir de este conjunto de datos, estructuramos la base de datos y tablas en Hive para poder hacer las consultas.

Creamos la base de datos

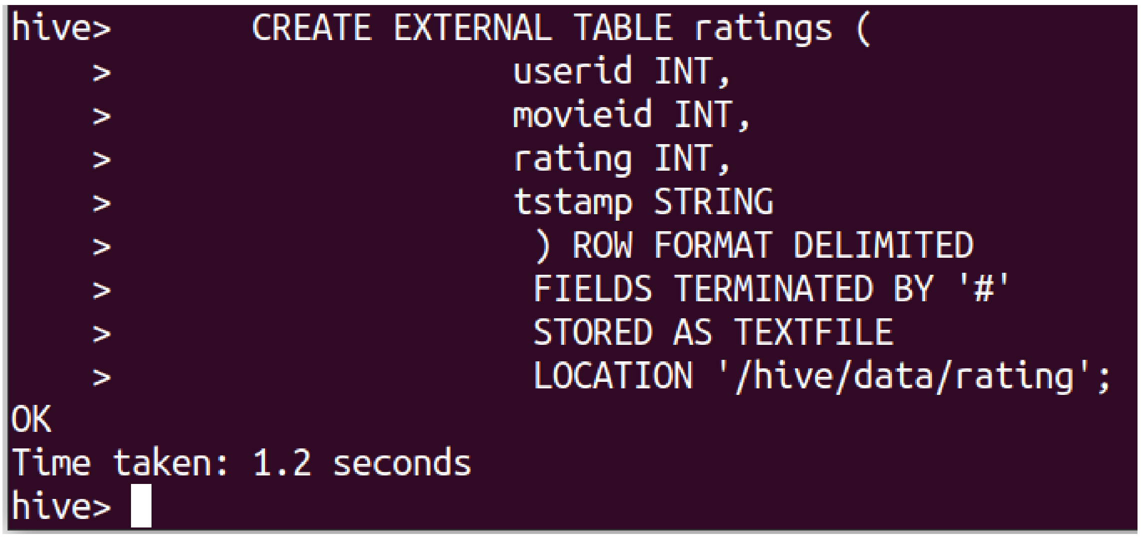
```
hive>   create database movielens;

hive>           use movielens;
```



```
hive>   create database movielens;
OK
Time taken: 1.761 seconds
hive>           use movielens;
OK
Time taken: 0.072 seconds
hive>
```

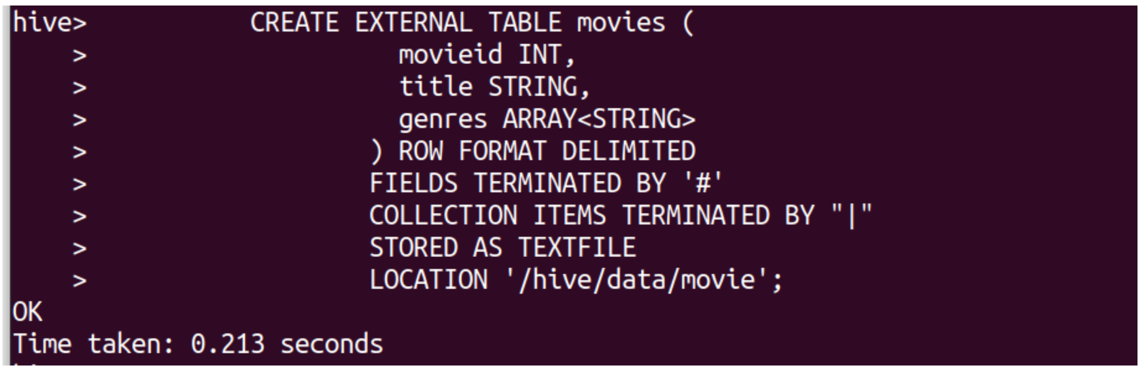
Creamos las tablas enlazadas con los datos externos al metastore de hive

```
        CREATE EXTERNAL TABLE ratings (
                     userid INT,
                     movieid INT,
                     rating INT,
                     tstamp STRING
                      ) ROW FORMAT DELIMITED
                      FIELDS TERMINATED BY '#'
                      STORED AS TEXTFILE
                      LOCATION '/hive/data/rating';
```



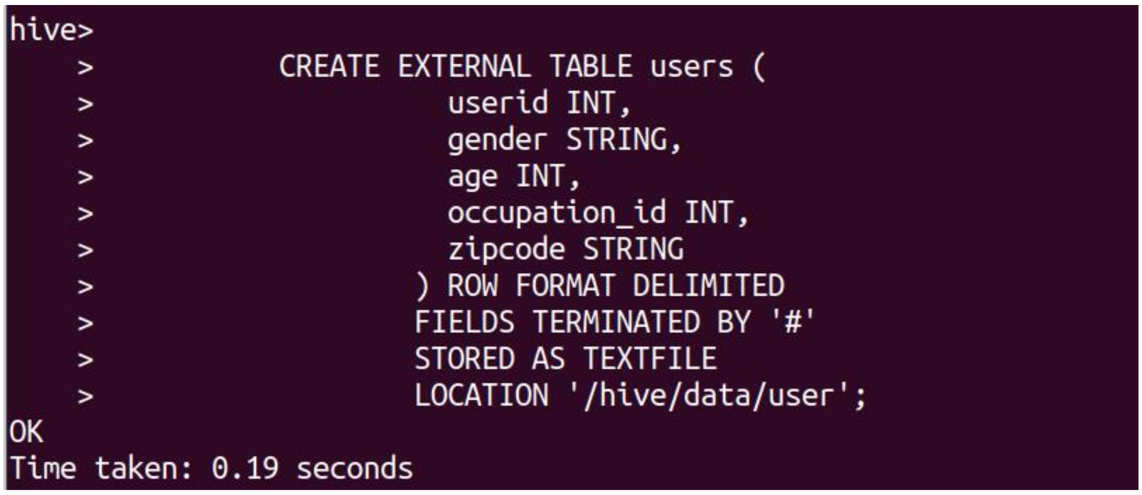
```
hive>         CREATE EXTERNAL TABLE ratings (
    >                     userid INT,
    >                     movieid INT,
    >                     rating INT,
    >                     tstamp STRING
    >                      ) ROW FORMAT DELIMITED
    >                      FIELDS TERMINATED BY '#'
    >                      STORED AS TEXTFILE
    >                      LOCATION '/hive/data/rating';
OK
Time taken: 1.2 seconds
hive>
```

```
        CREATE EXTERNAL TABLE movies (
                  movieid INT,
                  title STRING,
                  genres ARRAY<STRING>
                ) ROW FORMAT DELIMITED
                FIELDS TERMINATED BY '#'
                COLLECTION ITEMS TERMINATED BY "|"
                STORED AS TEXTFILE
                LOCATION '/hive/data/movie';
```



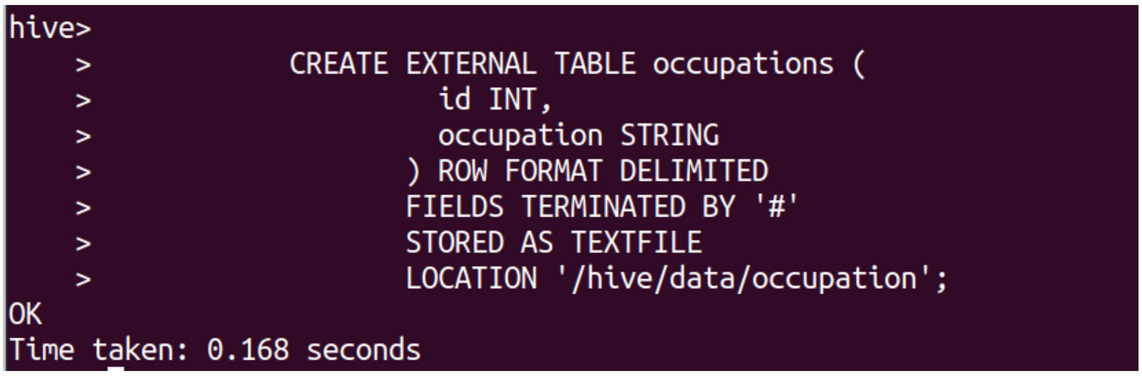
```
hive>         CREATE EXTERNAL TABLE movies (
    >                   movieid INT,
    >                   title STRING,
    >                   genres ARRAY<STRING>
    >                 ) ROW FORMAT DELIMITED
    >                 FIELDS TERMINATED BY '#'
    >                 COLLECTION ITEMS TERMINATED BY "|"
    >                 STORED AS TEXTFILE
    >                 LOCATION '/hive/data/movie';
OK
Time taken: 0.213 seconds
```

```
        CREATE EXTERNAL TABLE users (
                  userid INT,
                  gender STRING,
                  age INT,
                  occupation_id INT,
                  zipcode STRING
                ) ROW FORMAT DELIMITED
                FIELDS TERMINATED BY '#'
                STORED AS TEXTFILE
                LOCATION '/hive/data/user';
```



```
hive>
    >           CREATE EXTERNAL TABLE users (
    >                   userid INT,
    >                   gender STRING,
    >                   age INT,
    >                   occupation_id INT,
    >                   zipcode STRING
    >                 ) ROW FORMAT DELIMITED
    >                 FIELDS TERMINATED BY '#'
    >                 STORED AS TEXTFILE
    >                 LOCATION '/hive/data/user';
OK
Time taken: 0.19 seconds
```

```
            CREATE EXTERNAL TABLE occupations (
                    id INT,
                    occupation STRING
                  ) ROW FORMAT DELIMITED
                  FIELDS TERMINATED BY '#'
                  STORED AS TEXTFILE
                  LOCATION '/hive/data/occupation';
```



```
hive>
    >           CREATE EXTERNAL TABLE occupations (
    >                   id INT,
    >                   occupation STRING
    >                 ) ROW FORMAT DELIMITED
    >                 FIELDS TERMINATED BY '#'
    >                 STORED AS TEXTFILE
    >                 LOCATION '/hive/data/occupation';
OK
Time taken: 0.168 seconds
```

6.3.3. Verificar datos de las tablas

Mediante Hive, realizar una consulta simple obteniendo los tres primeros registros de cada tabla

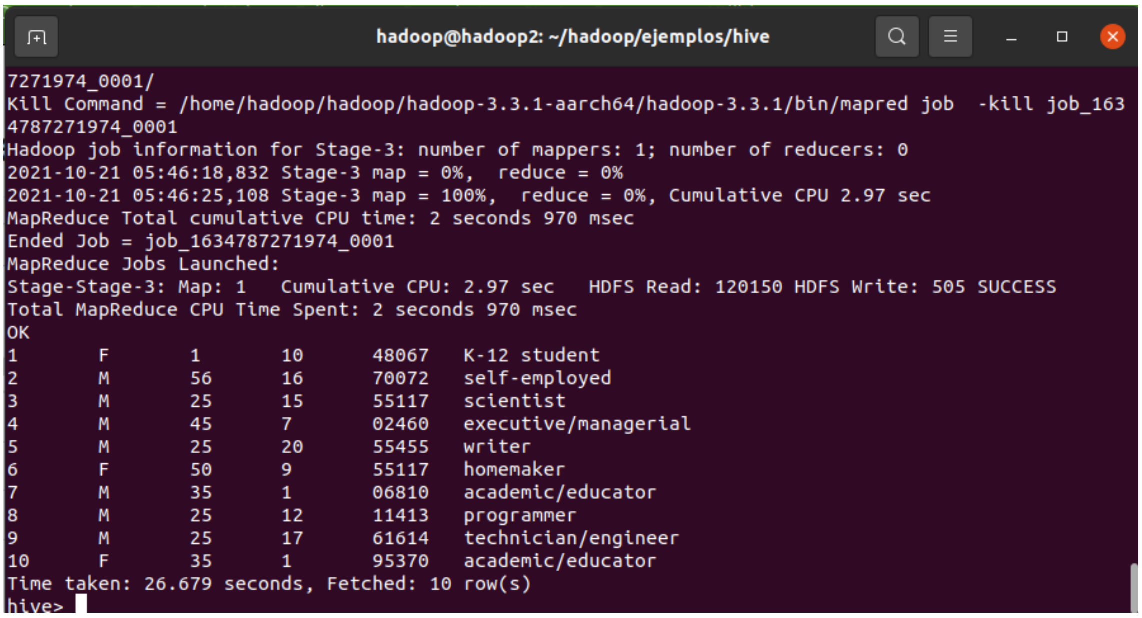
```
hive> select * from <tabla> limit 3;
```

6.4. Consultas con HiveSQ

A partir de las tablas anteriores, algunos ejemplos de consulta

6.4.1. Encontrar ocupación de todos los usuarios

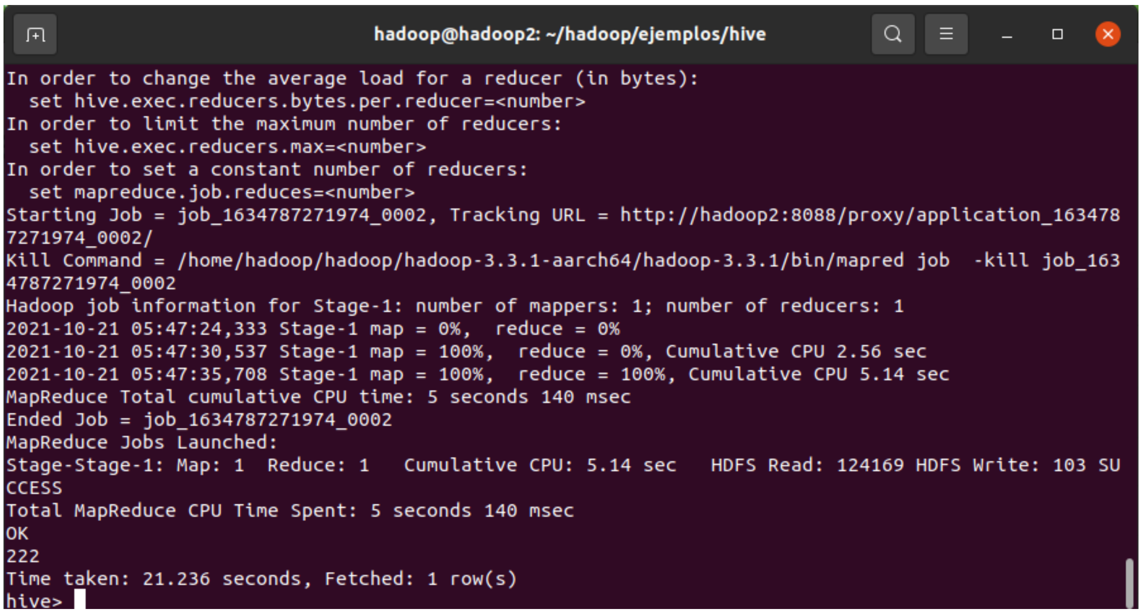
```
hive>select u.*, o.occupation from users u, occupations o where
u.occupation_id= o.id limit 10;
```



```
7271974_0001/
Kill Command = /home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/bin/mapred job  -kill job_163
4787271974_0001
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2021-10-21 05:46:18,832 Stage-3 map = 0%,  reduce = 0%
2021-10-21 05:46:25,108 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.97 sec
MapReduce Total cumulative CPU time: 2 seconds 970 msec
Ended Job = job_1634787271974_0001
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 2.97 sec   HDFS Read: 120150 HDFS Write: 505 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 970 msec
OK
1       F       1       10      48067   K-12 student
2       M       56      16      70072   self-employed
3       M       25      15      55117   scientist
4       M       45      7       02460   executive/managerial
5       M       25      20      55455   writer
6       F       50      9       55117   homemaker
7       M       35      1       06810   academic/educator
8       M       25      12      11413   programmer
9       M       25      17      61614   technician/engineer
10      F       35      1       95370   academic/educator
Time taken: 26.679 seconds, Fetched: 10 row(s)
hive>
```

6.4.2. Buscar el número de menores que han valorado películas

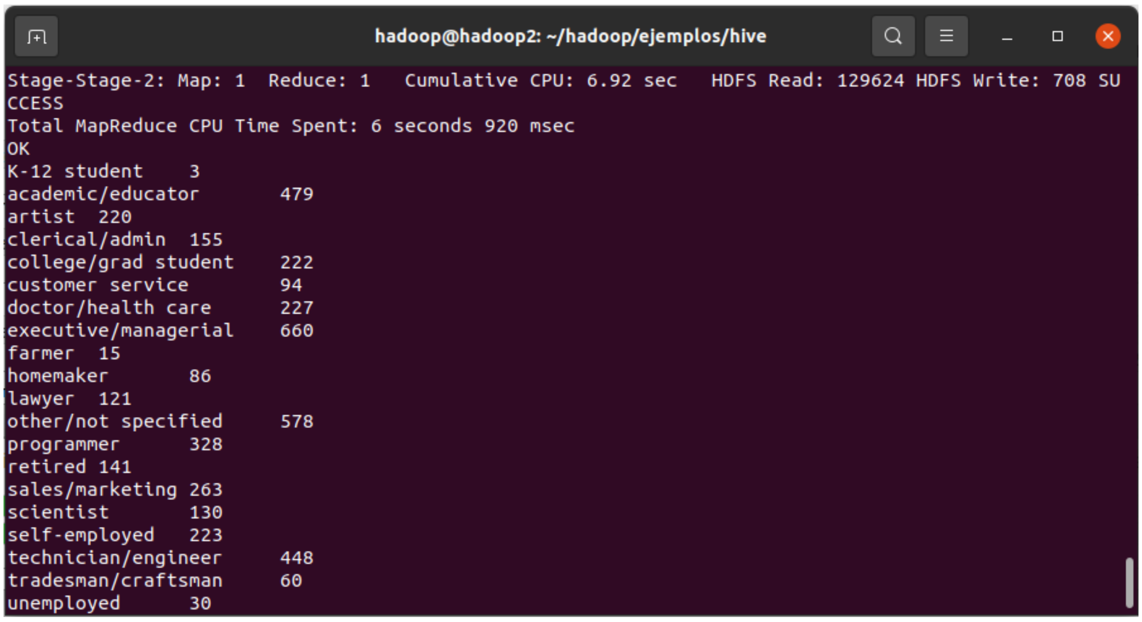
```
hive> select count(*) from users where age < 18;
```



```
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1634787271974_0002, Tracking URL = http://hadoop2:8088/proxy/application_163478
7271974_0002/
Kill Command = /home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/bin/mapred job  -kill job_163
4787271974_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-10-21 05:47:24,333 Stage-1 map = 0%,  reduce = 0%
2021-10-21 05:47:30,537 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.56 sec
2021-10-21 05:47:35,708 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.14 sec
MapReduce Total cumulative CPU time: 5 seconds 140 msec
Ended Job = job_1634787271974_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.14 sec   HDFS Read: 124169 HDFS Write: 103 SU
CCESS
Total MapReduce CPU Time Spent: 5 seconds 140 msec
OK
222
Time taken: 21.236 seconds, Fetched: 1 row(s)
hive>
```

6.4.3. Buscar el número de usuarios con la misma ocupación y con más de 25 años. Mostrar los detalles de la ocupación

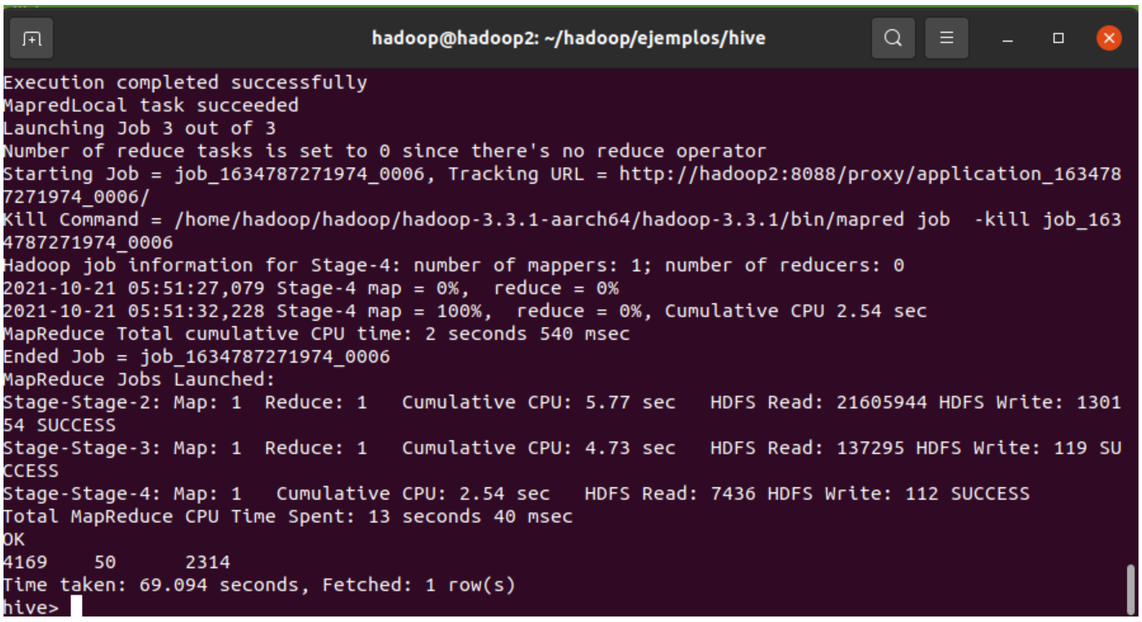
```
hive> select o.occupation, count(1)  from users u join occupations o where
u.occupation_id= o.id AND u.age > 24 group by o.occupation;
```



```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.92 sec   HDFS Read: 129624 HDFS Write: 708 SU
CCESS
Total MapReduce CPU Time Spent: 6 seconds 920 msec
OK
K-12 student    3
academic/educator       479
artist  220
clerical/admin  155
college/grad student    222
customer service        94
doctor/health care      227
executive/managerial    660
farmer  15
homemaker       86
lawyer  121
other/not specified     578
programmer      328
retired 141
sales/marketing 263
scientist       130
self-employed   223
technician/engineer     448
tradesman/craftsman     60
unemployed      30
```

6.4.4. Encontrar la edad el usuario con más valoraciones

```
hive> select u.userid, u.age, x.count from users u join ( select r.userid,
count(rating) count from ratings r group by (r.userid) order by count DESC
limit 1) x where u.userid = x.userid;
```



```
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1634787271974_0006, Tracking URL = http://hadoop2:8088/proxy/application_163478
7271974_0006/
Kill Command = /home/hadoop/hadoop/hadoop-3.3.1-aarch64/hadoop-3.3.1/bin/mapred job  -kill job_163
4787271974_0006
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2021-10-21 05:51:27,079 Stage-4 map = 0%,  reduce = 0%
2021-10-21 05:51:32,228 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 2.54 sec
MapReduce Total cumulative CPU time: 2 seconds 540 msec
Ended Job = job_1634787271974_0006
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.77 sec   HDFS Read: 21605944 HDFS Write: 1301
54 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 4.73 sec   HDFS Read: 137295 HDFS Write: 119 SU
CCESS
Stage-Stage-4: Map: 1   Cumulative CPU: 2.54 sec   HDFS Read: 7436 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 40 msec
OK
4169    50      2314
Time taken: 69.094 seconds, Fetched: 1 row(s)
hive>
```