Clono el repositorio

git clone https://github.com/lopezdar222/herramientas_big_data

1. HDFS

1.1. Ingreso a la carpeta "herramientas_big_data"

cd herramientas_big_data

```
ubuntu@servidor_ubuntu:~$ git clone https://github.com/lopezdar222/herramientas_
big_data
Cloning into 'herramientas_big_data'...
remote: Enumerating objects: 206, done.
remote: Counting objects: 100% (206/206), done.
remote: Compressing objects: 100% (142/142), done.
remote: Total 206 (delta 78), reused 162 (delta 39), pack-reused 0
Receiving objects: 100% (206/206), 18.97 MiB | 1.68 MiB/s, done.
Resolving deltas: 100% (78/78), done.
ubuntu@servidor_ubuntu:~$ cd herramientas_big_data
ubuntu@servidor_ubuntu:~/herramientas_big_data$ ls
Datasets                                docker-compose-kafka.yml
Generacion_Ventas.ipynb                 docker-compose-v1.yml
Mongo                                   docker-compose-v2.yml
Parquet                                 docker-compose-v3.yml
Paso00.sh                               docker-compose-v4.yml
Paso01.sh                               docker-compose.yml
Paso02.hql                              ejemploNeo4J.txt
Paso02_ConConsultas.hql                 hadoop-hive.env
Paso03.hql                              hadoop.env
Paso04.hql                              hbase-distributed-local.env
Paso04_ConConsulta.hql                  iris.hql
Paso05.py                               pruebaPySpark.py
Paso06_GeneracionVentasNuevasPorDia.py  pruebaScala.scala
Paso06_IncrementalVentas.py             pyspark-ETL.ipynb
README.md
```

1.2. Levanto el contenedor "docker-compose-v1.yml"

sudo docker-compose -f docker-compose-v1.yml up -d

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker-compose -f docker-compose-v1.yml up
-d
Creating network "herramientasbigdata_default" with the default driver
Creating volume "herramientasbigdata_hadoop_historyserver" with default driver
Creating volume "herramientasbigdata_hadoop_namenode" with default driver
Creating volume "herramientasbigdata_hadoop_datanode" with default driver
Creating volume "herramientasbigdata_hadoop_datasets" with default driver
Pulling namenode (bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8)...
2.0.0-hadoop3.2.1-java8: Pulling from bde2020/hadoop-namenode
3192219afd04: Pull complete
7127a1d8cced: Pull complete
883a89599900: Pull complete
77920a3e82af: Pull complete
92329e81aec4: Pull complete
f373218fec59: Pull complete
aa53513fe997: Pull complete
8b1800105b98: Pull complete
c3a84a3e49c8: Pull complete
a65640a64a76: Pull complete
facffb3a6de3: Pull complete
c71a6df73788: Pull complete
73b8c0ccb707: Pull complete
Digest: sha256:51ad9293ec52083c5003ef0aaab00c3dd7d6335ddf495cc1257f97a272cab4c0
Status: Downloaded newer image for bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
Pulling historyserver (bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8)...
2.0.0-hadoop3.2.1-java8: Pulling from bde2020/hadoop-historyserver
3192219afd04: Already exists
```

1.3. Veo los contenedores activos

sudo docker ps

1.4. Entro al contenedor “namenode” que está corriendo.

sudo docker exec -it namenode bash

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED         STATUS
  PORTS                                                                                                  NAMES
c2cebddf5ad9   bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   5 minutes ago   Up 5 minutes (healthy)
  8088/tcp                                                                                               resourcemanager
5a7b07310012   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   5 minutes ago   Up 5 minutes (healthy)
  0.0.0.0:9864->9864/tcp, :::9864->9864/tcp                                                              datanode
792b26d02ae0   bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8     "/entrypoint.sh /run…"   5 minutes ago   Up 5 minutes (healthy)
  8188/tcp                                                                                               historyserver
c1a034b10553   bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   5 minutes ago   Up 5 minutes (healthy)
  8042/tcp                                                                                               nodemanager
8b012c00b188   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   5 minutes ago   Up 5 minutes (healthy)
  0.0.0.0:9870->9870/tcp, :::9870->9870/tcp, 0.0.0.0:9010->9000/tcp, :::9010->9000/tcp   namenode
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker exec -it namenode bash
root@8b012c00b188:/#
```

1.5. Veo los archivos en la carpeta actual

ls

1.6. Me ubico en la carpeta “home”

cd home

```
root@8b012c00b188:/# ls
KEYS  boot  entrypoint.sh  hadoop       home  lib64  mnt  proc  run     sbin  sys  usr
bin   dev   etc            hadoop-data  lib   media  opt  root  run.sh  srv   tmp  var
root@8b012c00b188:/# cd home
root@8b012c00b188:/home#
```

1.7. Creo un directorio o carpeta llamada “Datasets”

mkdir Datasets

```
root@8b012c00b188:/home# mkdir Datasets
root@8b012c00b188:/home# exit
exit
ubuntu@servidor_ubuntu:~/herramientas_big_data$ pwd
/home/ubuntu/herramientas_big_data
```

1.8. Copio el archivo al contenedor de Docker "Datasets"

OPC 1: sudo docker cp **/home/ubuntu/herramientas_big_data/Datasets** namenode:**/home/Datasets**

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ pwd
/home/ubuntu/herramientas_big_data
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets namenode:/home/Datasets/
```

OPC 2:

sudo docker cp **/home/ubuntu/herramientas_big_data/Datasets/calendario** namenode:**/home/Datasets**

sudo docker cp **/home/ubuntu/herramientas_big_data/Datasets/canaldeventa** namenode:**/home/Datasets**

.

.

sudo docker cp **/home/ubuntu/herramientas_big_data/Datasets/raw-flight-data.csv** namenode:**/home/Datasets**

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/canaldeventa namenode:/home/D
atasets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/cliente namenode:/home/Datase
ts/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/compra namenode:/home/Dataset
s/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/data_nvo namenode:/home/Datas
ets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/empleado namenode:/home/Datas
ets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/gasto namenode:/home/Datasets
/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/producto namenode:/home/Datas
ets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/proveedor namenode:/home/Data
sets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/sucursal namenode:/home/Datas
ets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/tipodegasto namenode:/home/Da
tasets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/venta namenode:/home/Datasets
/
```

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/airports.csv namenode:/home/D
atasets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/iris.csv namenode:/home/Datas
ets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/iris.json namenode:/home/Data
sets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/personal.csv namenode:/home/D
atasets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker cp /home/ubuntu/herramientas_big_data/Datasets/raw-flight-data.csv namenode:
/home/Datasets/
ubuntu@servidor_ubuntu:~/herramientas_big_data$
```

(*) Copié todo el contenido de la carpeta Datasets porque no me especificaba cuales.

1.9. Compruebo que los archivos se hayan copiado correctamente

1.10. Entro al contenedor “namenode”

sudo docker exec -it namenode bash

1.11. Entro al archivo “home”

cd home

1.12. Entro al archivo “Datasets”

cd Datasets

ls

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker exec -it namenode bash
root@8b012c00b188:/# ls
KEYS  boot  entrypoint.sh  hadoop       home  lib64  mnt  proc  run     sbin  sys  usr
bin   dev   etc            hadoop-data  lib   media  opt  root  run.sh  srv   tmp  var
root@8b012c00b188:/# cd home
root@8b012c00b188:/home# ls
Datasets
root@8b012c00b188:/home# cd Datasets
root@8b012c00b188:/home/Datasets# ls
airports.csv  canaldeventa  compra    empleado  iris.csv   personal.csv  proveedor           sucursal     venta
calendario    cliente       data_nvo  gasto     iris.json  producto      raw-flight-data.csv  tipodegasto
root@8b012c00b188:/home/Datasets#
```

1.13. Crear un directorio en HDFS llamado "/data".

hdfs dfs -mkdir -p /data

```
root@8b012c00b188:/# hdfs dfs -mkdir -p /data
```

1.14. Copiar los archivos csv provistos a HDFS

hdfs dfs -put **/home/Datasets/*** /data

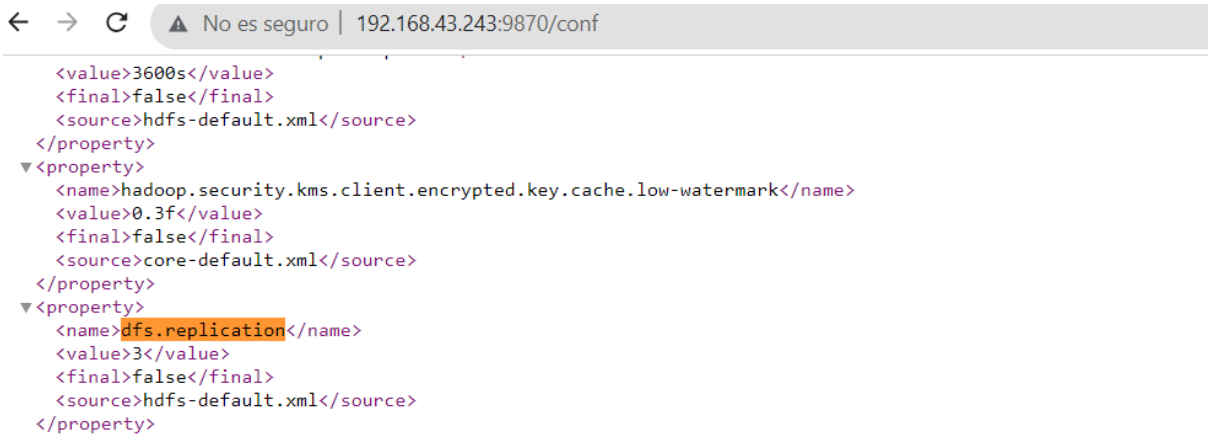
```
root@8b012c00b188:/# hdfs dfs -put /home/Datasets/* /data
2023-10-16 22:10:51,193 INFO sasl.SaslDataTransferClient: SASL encryption t
se
2023-10-16 22:10:51,609 INFO sasl.SaslDataTransferClient: SASL encryption t
se
2023-10-16 22:10:51,642 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1252)
        at java.lang.Thread.join(Thread.java:1326)
        at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.
        at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:6
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
2023-10-16 22:10:51,744 INFO sasl.SaslDataTransferClient: SASL encryption t
se
2023-10-16 22:10:51,782 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1252)
        at java.lang.Thread.join(Thread.java:1326)
        at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.
        at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:6
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
2023-10-16 22:10:51,872 INFO sasl.SaslDataTransferClient: SASL encryption t
se
2023-10-16 22:10:51,983 INFO sasl.SaslDataTransferClient: SASL encryption t
```

1.15. Compruebo que los archivos esten en /data

hdfs dfs -ls /data

```
root@8b012c00b188:/# hdfs dfs -ls /data
Found 17 items
-rw-r--r--   3 root supergroup      16308 2023-10-16 22:10 /data/airports.csv
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/calendario
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/canaldeventa
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/cliente
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/compra
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/data_nvo
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/empleado
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/gasto
-rw-r--r--   3 root supergroup       4813 2023-10-16 22:10 /data/iris.csv
-rw-r--r--   3 root supergroup      15802 2023-10-16 22:10 /data/iris.json
-rw-r--r--   3 root supergroup         94 2023-10-16 22:10 /data/personal.csv
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/producto
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/proveedor
-rw-r--r--   3 root supergroup   69772435 2023-10-16 22:10 /data/raw-flight-data.csv
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/sucursal
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/tipodegasto
drwxr-xr-x   - root supergroup          0 2023-10-16 22:10 /data/venta
root@8b012c00b188:/#
```

Nota: Busque dfs.blocksize y dfs.replication en http://<IP_Anfitrion>:9870/conf para encontrar los valores de tamaño de bloque y factor de réplica respectivamente entre otras configuraciones del sistema Hadoop.



Resumen:

Creé una carpeta Datasets y copié los archivos de local a la carpeta que creé. Luego, creé una carpeta dentro de HDFS llamada data y copié ahí los archivos de la carpeta Datasets.

2. HIVE

```
exit
```

Crear tablas en Hive, a partir de los csv ingestados en HDFS.

2.1. Levanto el contenedor “docker-compose-v1.yml”

```
sudo docker-compose -f docker-compose-v2.yml up -d
```

2.2. Veo los contenedores activos

```
sudo docker ps
```

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker-compose -f docker-compose-v2.yml up -d
Starting resourcemanager ...
Starting hive-metastore-postgresql ...
Starting resourcemanager
Starting datanode ...
Starting hive-metastore-postgresql
Starting datanode
Starting historyserver ...
Starting hive-metastore ...
Starting historyserver
Starting nodemanager ...
Starting namenode ...
Starting hive-metastore
Starting nodemanager
Starting namenode ... done
Starting hive-server ...
Starting hive-server ... done
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED         STATUS
          PORTS                                                                                  NAMES
41263b038d53   bde2020/hive:2.3.2-postgresql-metastore              "entrypoint.sh /opt/…"   5 minutes ago   Up 23 seconds
          0.0.0.0:10000->10000/tcp, :::10000->10000/tcp, 10002/tcp                               hive-server
67bd55036dcc   bde2020/hive:2.3.2-postgresql-metastore              "entrypoint.sh /opt/…"   5 minutes ago   Up 25 seconds
          10000/tcp, 0.0.0.0:9083->9083/tcp, :::9083->9083/tcp, 10002/tcp                        hive-metastore
eebfdfa128c8   bde2020/hive-metastore-postgresql:2.3.0              "/docker-entrypoint.…"   5 minutes ago   Up 25 seconds
          0.0.0.0:5432->5432/tcp, :::5432->5432/tcp                                              hive-metastore-postgresql
c2cebddf5ad9   bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   2 hours ago     Up 28 seconds (health:
starting)   8088/tcp                                                                               resourcemanager
5a7b07310012   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   2 hours ago     Up 26 seconds (health:
starting)   0.0.0.0:9864->9864/tcp, :::9864->9864/tcp                                              datanode
792b26d02ae0   bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8     "/entrypoint.sh /run…"   2 hours ago     Up 25 seconds (health:
starting)   8188/tcp                                                                               historyserver
c1a034b10553   bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   2 hours ago     Up 26 seconds (health:
starting)   8042/tcp                                                                               nodemanager
8b012c00b188   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   2 hours ago     Up 24 seconds (health:
starting)   0.0.0.0:9870->9870/tcp, :::9870->9870/tcp, 0.0.0.0:9010->9000/tcp, :::9010->9000/tcp   namenode
ubuntu@servidor_ubuntu:~/herramientas_big_data$
```

2.3. Me ubico dentro del contenedor correspondiente al servidor de Hive, es decir entro al contenedor “hive-server”, que está activo.

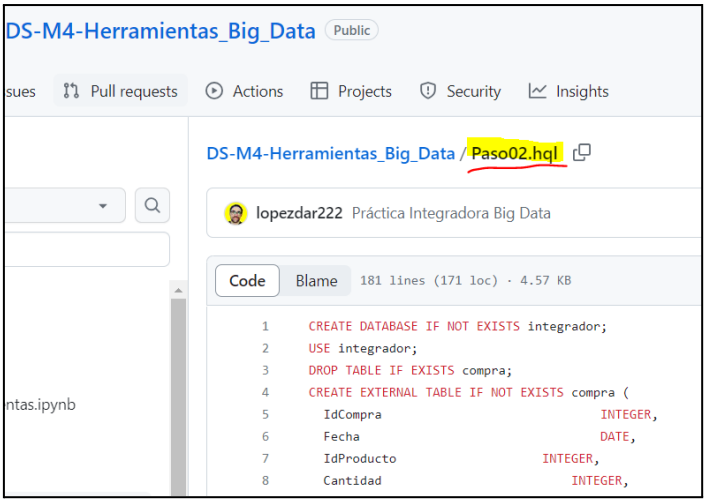
sudo docker exec -it hive-server bash

hive

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker exec -it hive-server bash
root@41263b038d53:/opt# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/imp
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Asy
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
r using Hive 1.X releases.
hive>
```

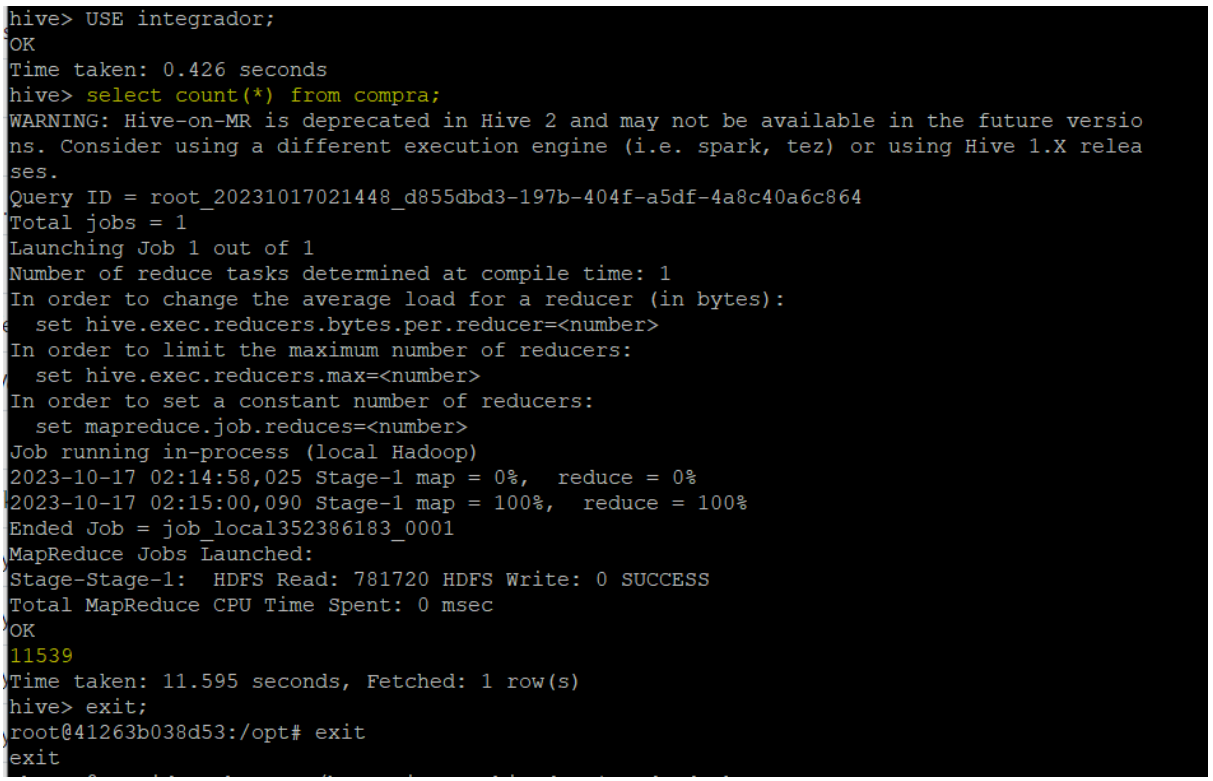
2.4. Copio lo que está en este archivo .sql “Paso02.hql”




```
hive> USE integrador;
OK
Time taken: 3.203 seconds
hive>
    > DROP TABLE IF EXISTS compra;
OK
Time taken: 0.3 seconds
hive>
    > CREATE EXTERNAL TABLE IF NOT EXISTS compra (IdCompra INTEGER, Fecha DATE, IdProducto INTEGE
veedor INTEGER) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIE
.count'='1') LOCATION '/data/compra';
OK
Time taken: 0.636 seconds
hive>
    > DROP TABLE IF EXISTS gasto;
OK
Time taken: 0.034 seconds
hive>
    > CREATE EXTERNAL TABLE IF NOT EXISTS gasto (IdGasto INTEGER, IdSucursal INTEGER, IdTipoGasto
ORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ('separatorChar'=',
N '/data/gasto';
OK
Time taken: 0.295 seconds
hive>
    > DROP TABLE IF EXISTS tipo_gasto;
OK
Time taken: 0.054 seconds
```

LISTO!

Compruebo la creación en Hive



Lo que hacen los comandos de hive es crear tablas externas que apuntan al archivo CSV en HDFS directamente.