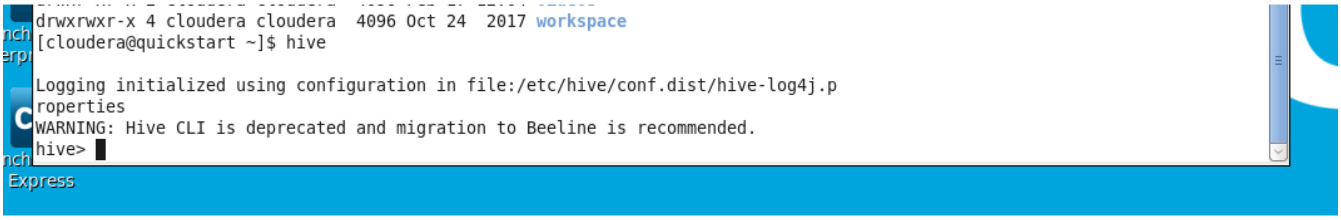
HIVE

Es una infraestructura para el almacenaje y consultade datos basada en Hadoop

Hadoop proporciona escalabilidad masiva y con capacidades de tolerancia a fallos para el procesamiento y almacenamiento de datos

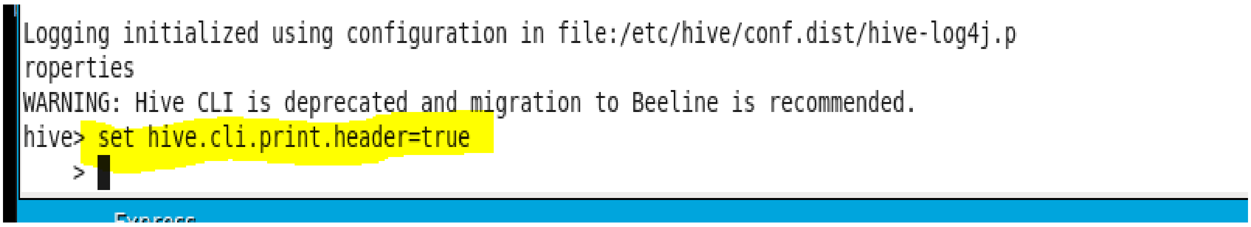
Ejercicios Hive

1. Entrar en Hive



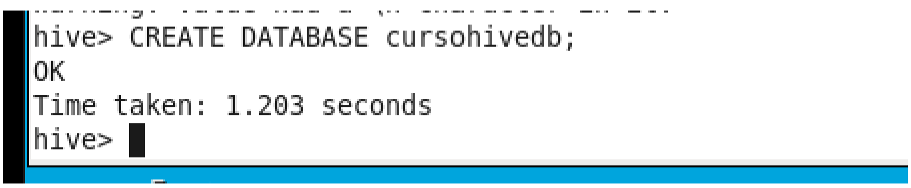
```
drwxrwxr-x 4 cloudera cloudera  4096 Oct 24  2017 workspace
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> 
```

2. Modificar la propiedad correspondiente para mostrar por pantalla las cabeceras de las tablas



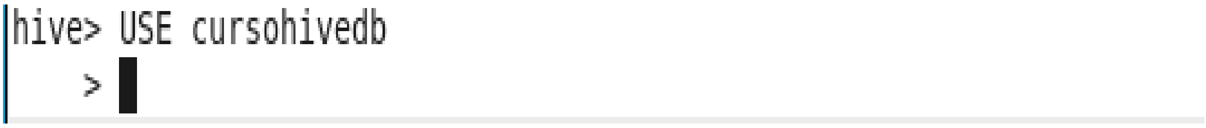
```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> set hive.cli.print.header=true
    > 
```

3. Crear una base de datos llamada “cursohivedb”



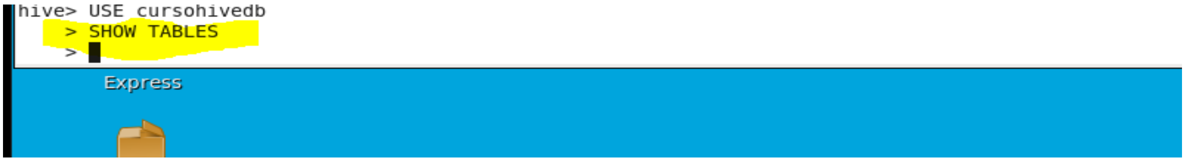
```
hive> CREATE DATABASE cursohivedb;
OK
Time taken: 1.203 seconds
hive> 
```

4. Situarnos en la base de datos recién creada para trabajar con ella



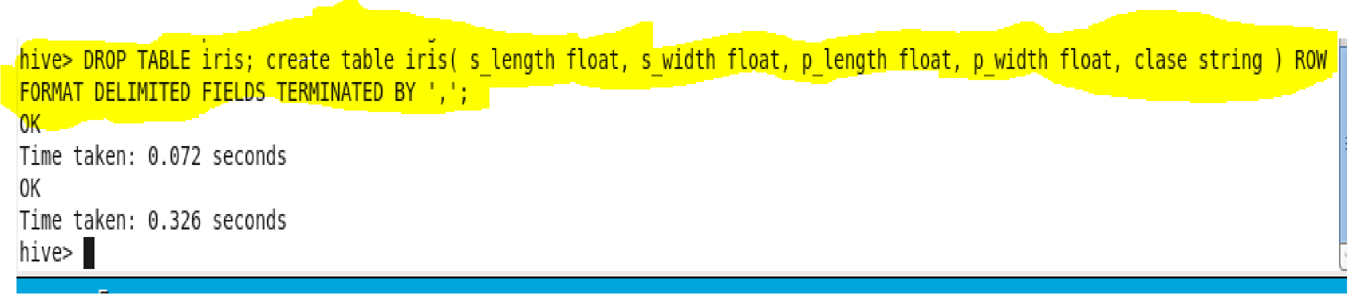
```
hive> USE cursohivedb
    > 
```

5. Comprobar que la base de datos está vacía



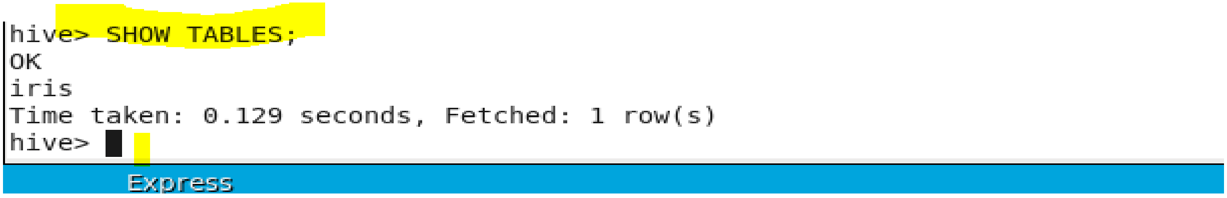
```
hive> USE cursohivedb
    > SHOW TABLES
    > 
```

6. Crear una tabla llamada “iris” en nuestra base de datos que contenga 5 columnas



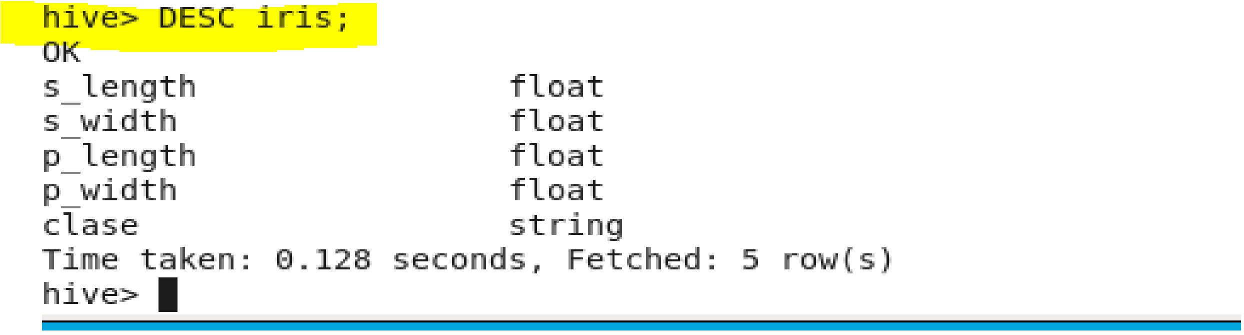
```
hive> DROP TABLE iris; create table iris( s_length float, s_width float, p_length float, p_width float, clase string ) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.072 seconds
OK
Time taken: 0.326 seconds
hive> 
```

7. Comprobar que la tabla se ha creado y el tipado de sus columnas



```
hive> SHOW TABLES;
OK
iris
Time taken: 0.129 seconds, Fetched: 1 row(s)
hive> 
```

Tipado de sus columnas

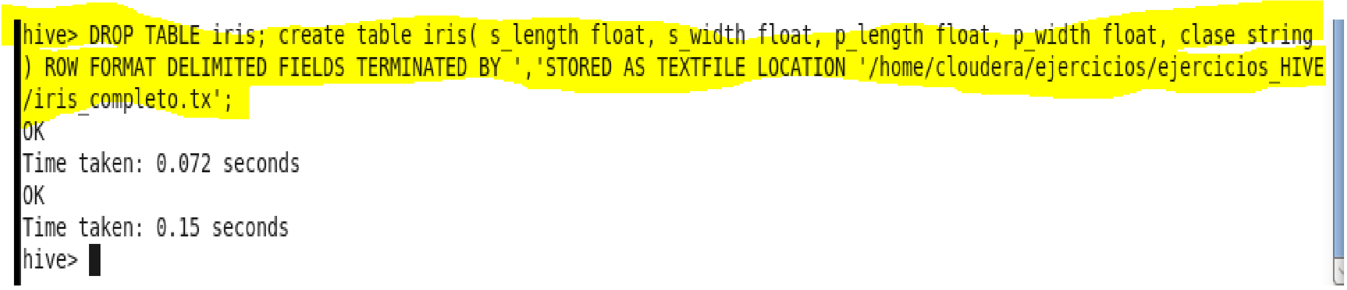


```
hive> DESC iris;
OK
s_length                float
s_width                 float
p_length                float
p_width                 float
clase                   string
Time taken: 0.128 seconds, Fetched: 5 row(s)
hive> 
```

8. Importar el fichero "iris_completo.txt" al local file system del cluster en la carpeta /home/cloudera/ejercicios/ejercicios_HIVE

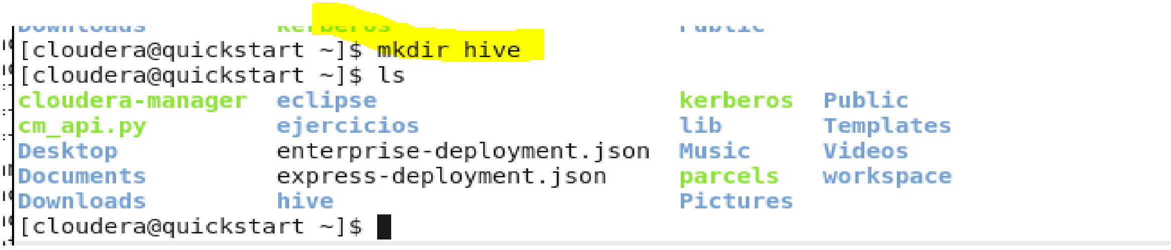
a. Copiar el fichero a HDFS en la ruta /user/cloudera/hive. Realizar las acciones

Necesarias

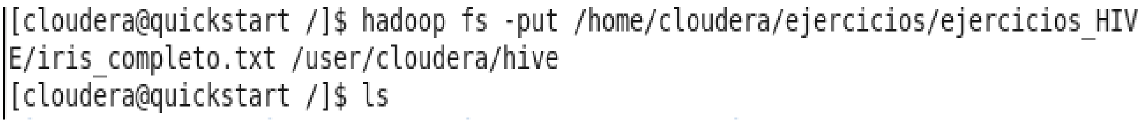


```
hive> DROP TABLE iris; create table iris( s_length float, s_width float, p_length float, p_width float, clase string
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','STORED AS TEXTFILE LOCATION '/home/cloudera/ejercicios/ejercicios_HIVE
/iris_completo.tx';
OK
Time taken: 0.072 seconds
OK
Time taken: 0.15 seconds
hive> 
```

9. Comprueba que el fichero está en la ruta en HDFS indicada

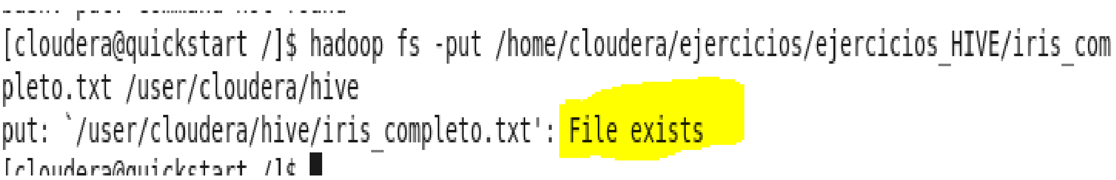


```
[cloudera@quickstart ~]$ mkdir hive
[cloudera@quickstart ~]$ ls
cloudera-manager  eclipse                     kerberos  Public
cm_api.py         ejercicios                  lib       Templates
Desktop           enterprise-deployment.json  Music     Videos
Documents         express-deployment.json     parcels   workspace
Downloads         hive                        Pictures
[cloudera@quickstart ~]$ 
```



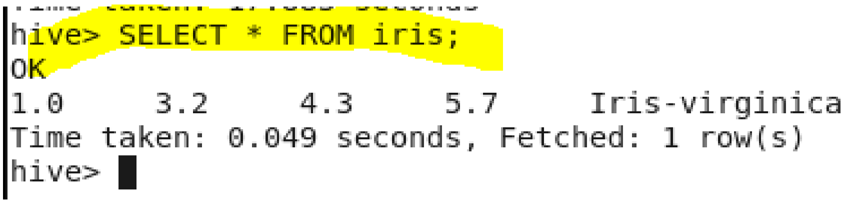
```
[cloudera@quickstart /]$ hadoop fs -put /home/cloudera/ejercicios/ejercicios_HIV
E/iris_completo.txt /user/cloudera/hive
[cloudera@quickstart /]$ ls
```

10. Importa el fichero en la tabla iris que acabamos de crear desde HDFS



```
[cloudera@quickstart /]$ hadoop fs -put /home/cloudera/ejercicios/ejercicios_HIVE/iris_com
pleto.txt /user/cloudera/hive
put: `/user/cloudera/hive/iris_completo.txt': File exists
[cloudera@quickstart /]$ 
```

11. Comprobar que la table tiene datos



```
hive> SELECT * FROM iris;
OK
1.0     3.2     4.3     5.7     Iris-virginica
Time taken: 0.049 seconds, Fetched: 1 row(s)
hive> 
```

12. Mostrar las 5 primeras filas de la tabla iris

```
Hive> SELECT * FROM iris Limit 5;
```

13. Mostrar solo aquellas filas cuyo s_length sea mayor que 5. Observad que se ejecuta un MapReduce y que el tiempo de ejecución es un poco mayor

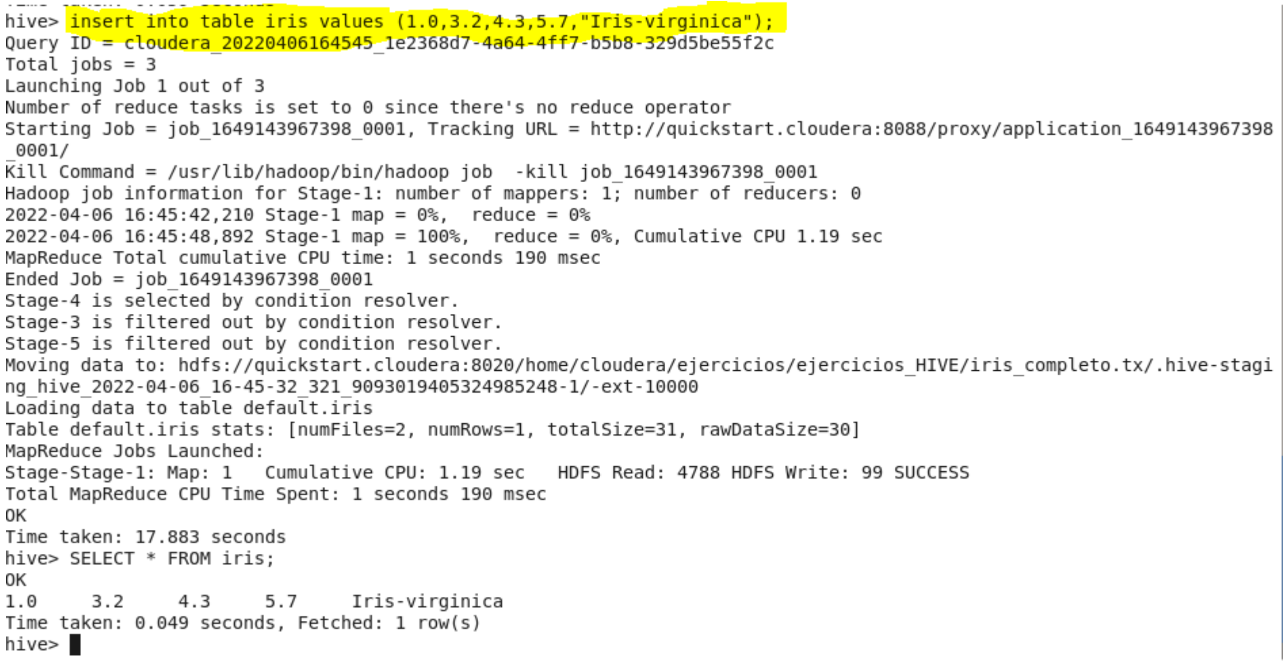
```
Hive> Select * from iris as i where i.s_length>5
```

14. Seleccionar la media de s_width agrupados por clase. Observad que ahora el tiempo de ejecución aumenta considerablemente.

15. Pregunta: vemos que aparece un valor NULL como resultado en la query anterior. ¿Por qué? ¿cómo los eliminarías? Porque había algún dato erróneo, no numérico o nulo en el campo de

alguna clase. Para eliminarlos podríamos añadir la condición where para que fuera distinto de null.

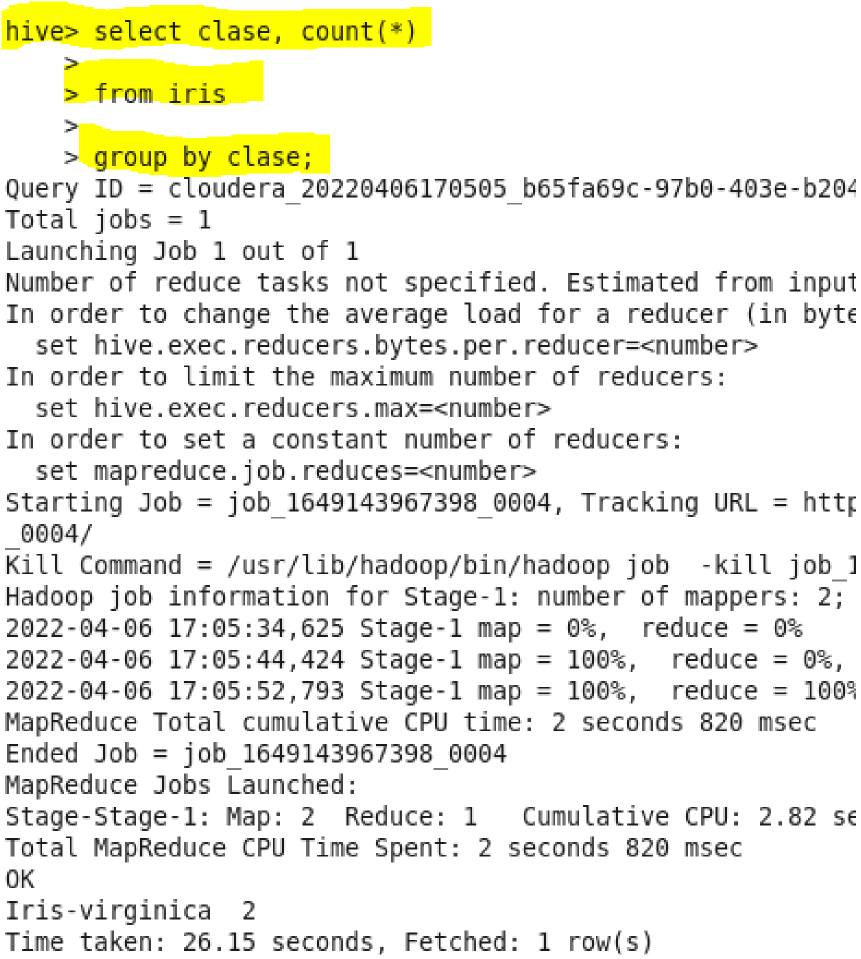
16. Insertar en la tabla la siguiente fila (1.0,3.2,4.3,5.7,"Iris-virginica")



```
hive> insert into table iris values (1.0,3.2,4.3,5.7,"Iris-virginica");
Query ID = cloudera_20220406164545_1e2368d7-4a64-4ff7-b5b8-329d5be55f2c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1649143967398_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1649143967398
_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1649143967398_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-04-06 16:45:42,210 Stage-1 map = 0%,  reduce = 0%
2022-04-06 16:45:48,892 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.19 sec
MapReduce Total cumulative CPU time: 1 seconds 190 msec
Ended Job = job_1649143967398_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.cloudera:8020/home/cloudera/ejercicios/ejercicios_HIVE/iris_completo.tx/.hive-stagi
ng_hive_2022-04-06_16-45-32_321_9093019405324985248-1/-ext-10000
Loading data to table default.iris
Table default.iris stats: [numFiles=2, numRows=1, totalSize=31, rawDataSize=30]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.19 sec   HDFS Read: 4788 HDFS Write: 99 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 190 msec
OK
Time taken: 17.883 seconds
hive> SELECT * FROM iris;
OK
1.0     3.2     4.3     5.7     Iris-virginica
Time taken: 0.049 seconds, Fetched: 1 row(s)
hive> █
```

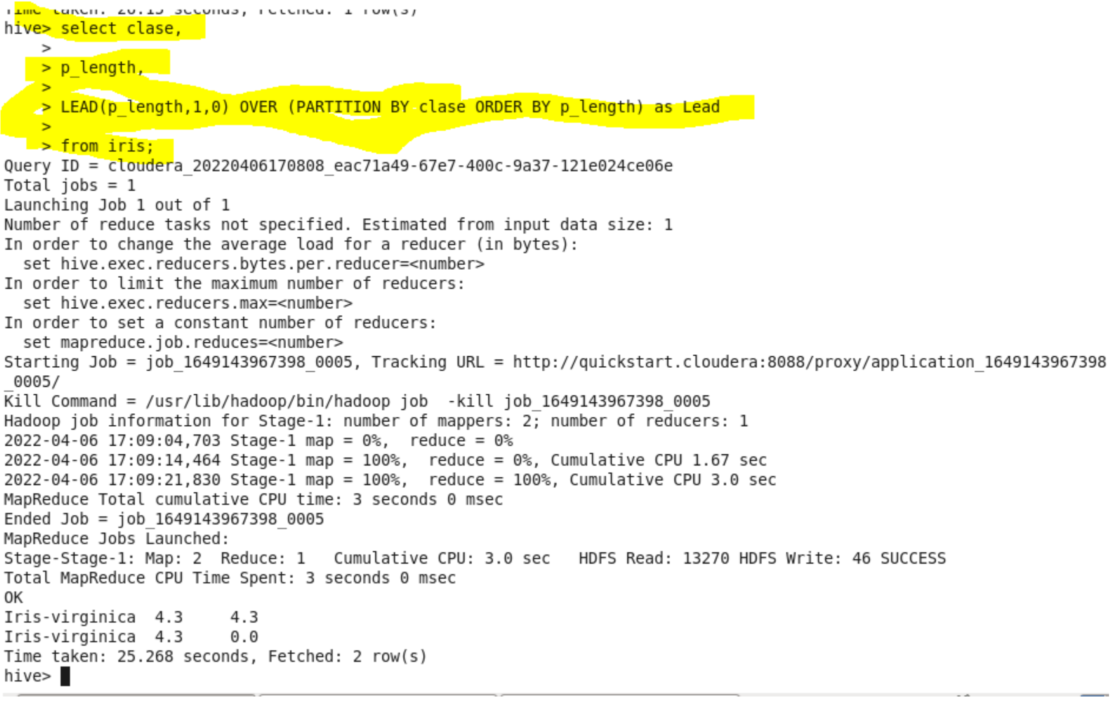
17. Contar el número de ocurrencias de cada clase

18. Seleccionar las clases que tengan más de 45 ocurrencias a. Select clase from iris group by clase having count(*)>45;



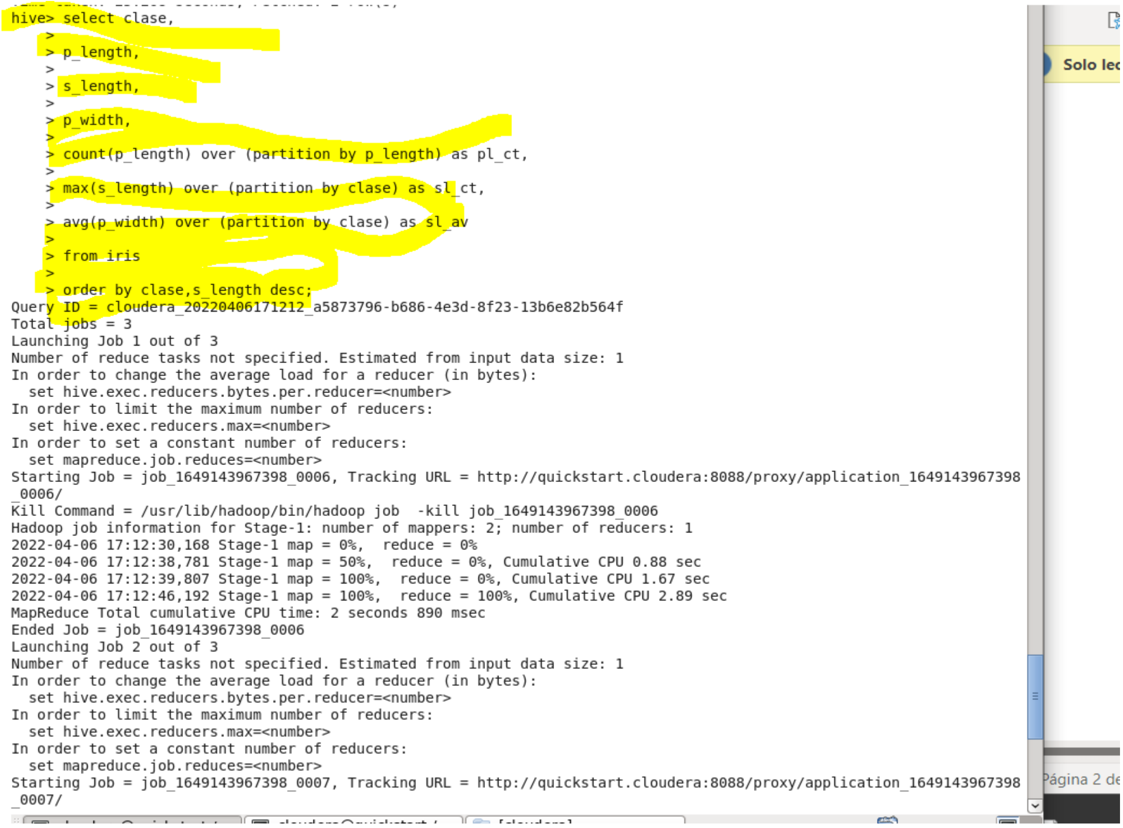
```
hive> select clase, count(*)
    >
    > from iris
    >
    > group by clase;
Query ID = cloudera_20220406170505_b65fa69c-97b0-403e-b204
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input
In order to change the average load for a reducer (in byte
  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_1649143967398_0004, Tracking URL = http
_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1
Hadoop job information for Stage-1: number of mappers: 2;
2022-04-06 17:05:34,625 Stage-1 map = 0%,  reduce = 0%
2022-04-06 17:05:44,424 Stage-1 map = 100%,  reduce = 0%,
2022-04-06 17:05:52,793 Stage-1 map = 100%,  reduce = 100%
MapReduce Total cumulative CPU time: 2 seconds 820 msec
Ended Job = job_1649143967398_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 2.82 se
Total MapReduce CPU Time Spent: 2 seconds 820 msec
OK
Iris-virginica  2
Time taken: 26.15 seconds, Fetched: 1 row(s)
```

19. Utilizando la función LEAD, ejecutar una query que devuelva la clase, p_length y el LEAD de p_length con Offset=1 y Default_Value =0, particionado por clase y ordenado por p_length.

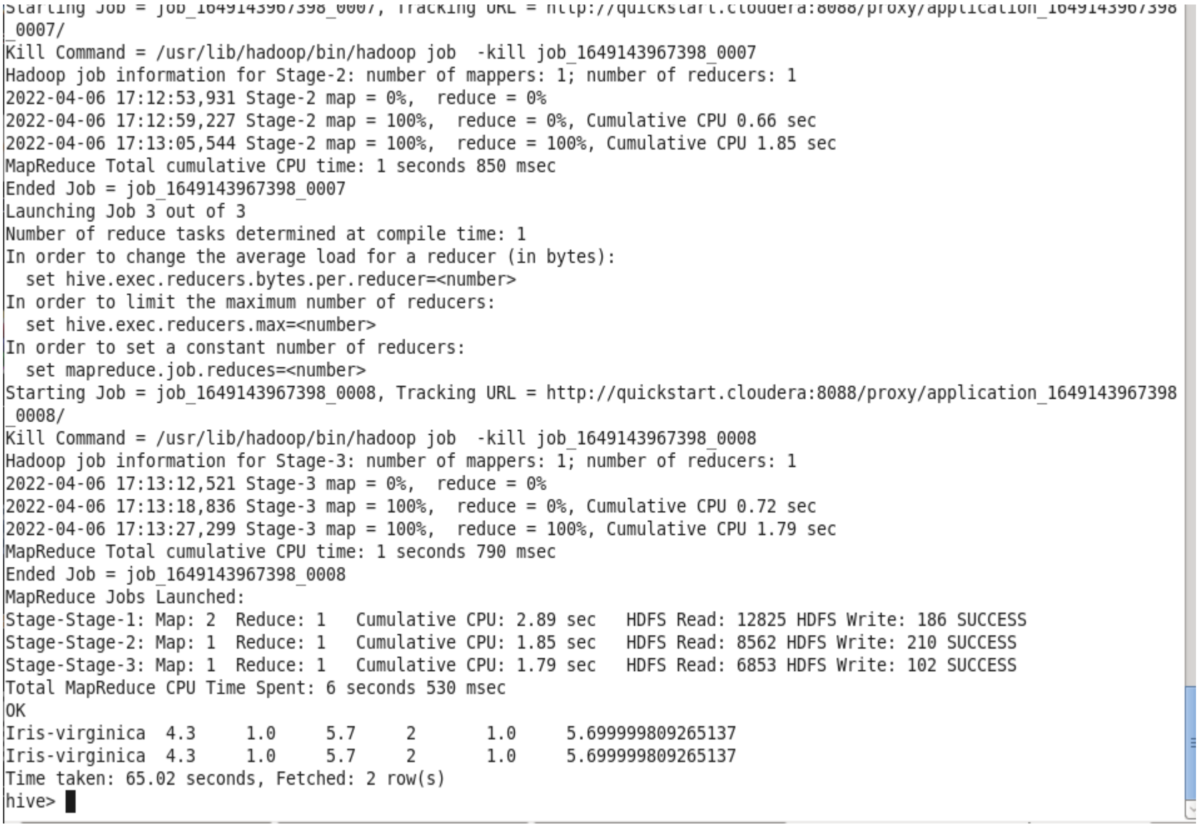


```
Time taken: 20.15 seconds, Fetched: 1 row(s)
hive> select clase,
    >
    > p_length,
    >
    > LEAD(p_length,1,0) OVER (PARTITION BY clase ORDER BY p_length) as Lead
    >
    > from iris;
Query ID = cloudera_20220406170808_eac71a49-67e7-400c-9a37-121e024ce06e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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_1649143967398_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1649143967398
_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1649143967398_0005
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-04-06 17:09:04,703 Stage-1 map = 0%,  reduce = 0%
2022-04-06 17:09:14,464 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.67 sec
2022-04-06 17:09:21,830 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.0 sec
MapReduce Total cumulative CPU time: 3 seconds 0 msec
Ended Job = job_1649143967398_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 3.0 sec   HDFS Read: 13270 HDFS Write: 46 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 0 msec
OK
Iris-virginica  4.3     4.3
Iris-virginica  4.3     0.0
Time taken: 25.268 seconds, Fetched: 2 row(s)
hive> █
```

20. Utilizando funciones de ventanas, seleccionar la clase, p_length, s_length, p_width, el número de valores distintos de p_length en todo el dataset, el valor máximo de s_length por clase y la media de p_width por clase, ordenado por clase y s_length de manera descendente.



```
hive> select clase,
    >
    > p_length,
    >
    > s_length,
    >
    > p_width,
    >
    > count(p_length) over (partition by p_length) as pl_ct,
    >
    > max(s_length) over (partition by clase) as sl_ct,
    >
    > avg(p_width) over (partition by clase) as sl_av
    >
    > from iris
    >
    > order by clase,s_length desc;
Query ID = cloudera_20220406171212_a5873796-b686-4e3d-8f23-13b6e82b564f
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
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_1649143967398_0006, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1649143967398
_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1649143967398_0006
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-04-06 17:12:30,168 Stage-1 map = 0%,  reduce = 0%
2022-04-06 17:12:38,781 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 0.88 sec
2022-04-06 17:12:39,807 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.67 sec
2022-04-06 17:12:46,192 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.89 sec
MapReduce Total cumulative CPU time: 2 seconds 890 msec
Ended Job = job_1649143967398_0006
Launching Job 2 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
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_1649143967398_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1649143967398
_0007/
```



```
_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1649143967398_0007
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-04-06 17:12:53,931 Stage-2 map = 0%,  reduce = 0%
2022-04-06 17:12:59,227 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 0.66 sec
2022-04-06 17:13:05,544 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 1.85 sec
MapReduce Total cumulative CPU time: 1 seconds 850 msec
Ended Job = job_1649143967398_0007
Launching Job 3 out of 3
Number of reduce tasks determined at compile time: 1
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_1649143967398_0008, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1649143967398
_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1649143967398_0008
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2022-04-06 17:13:12,521 Stage-3 map = 0%,  reduce = 0%
2022-04-06 17:13:18,836 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 0.72 sec
2022-04-06 17:13:27,299 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 1.79 sec
MapReduce Total cumulative CPU time: 1 seconds 790 msec
Ended Job = job_1649143967398_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 2.89 sec   HDFS Read: 12825 HDFS Write: 186 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 1.85 sec   HDFS Read: 8562 HDFS Write: 210 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 1.79 sec   HDFS Read: 6853 HDFS Write: 102 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 530 msec
OK
Iris-virginica  4.3     1.0     5.7     2       1.0     5.699999809265137
Iris-virginica  4.3     1.0     5.7     2       1.0     5.699999809265137
Time taken: 65.02 seconds, Fetched: 2 row(s)
hive> 
```