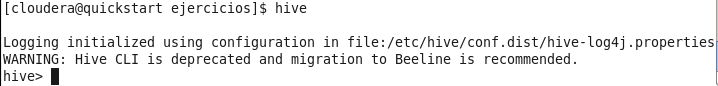
**Ejercicios Hive**

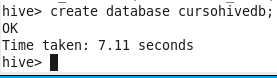
1. Entrar en Hive
   1. Hive



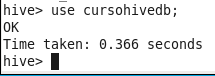
1. Modificar la propiedad correspondiente para mostrar por pantalla las cabeceras de las tablas
   1. “set hive.cli.print.header=true;”



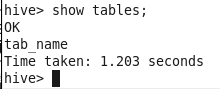
1. Crear una base de datos llamada “cursohivedb”
   1. Créate database cursohivedb



1. Situarnos en la base de datos recién creada para trabajar con ella
   1. Use cursohivedb



1. Comprobar que la base de datos está vacía
   1. Show tables;



1. Crear una tabla llamada “iris” en nuestra base de datos que contenga 5 columnas (s\_length float,s\_width float,p\_length float,p\_width float,clase string) cuyos campos estén separados por comas (ROW FORMAT DELIMITED FIELDS TERMINATED BY ',')

CREATE TABLE iris (

S\_length float,

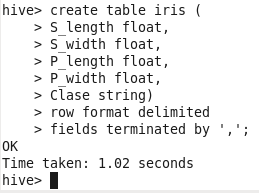
S\_width float,

P\_length float,

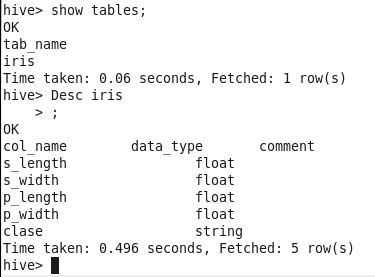
P\_width float,

Clase string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘id’



1. Comprobar que la tabla se ha creado y el tipado de sus columnas
   1. Show tables
   2. Desc iris



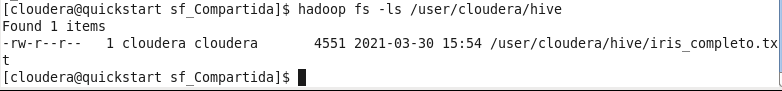
1. Importar el fichero “iris\_completo.txt” al local file system del cluster en la carpeta /home/cloudera/ejercicios/ejercicios\_HIVE



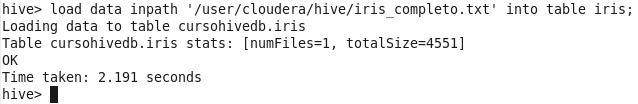
1. Copiar el fichero a HDFS en la ruta /user/cloudera/hive. Realizar las acciones necesarias
   1. Hadoop fs -mkdir /user/Cloudera/hive
   2. Hadoop fs -put /home/Cloudera/ejercicios/ejercicios\_hive/iris\_completo.txt /user/Cloudera/hive



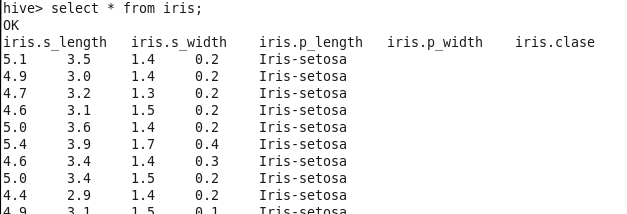
1. Comprueba que el fichero está en la ruta en HDFS indicada



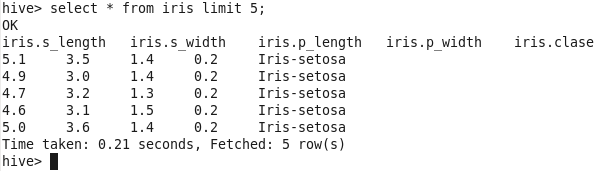
1. Importa el fichero en la tabla iris que acabamos de crear desde HDFS
   1. Load data inpath ‘/user/Cloudera/hive/iris\_completo.txt’ into table iris;



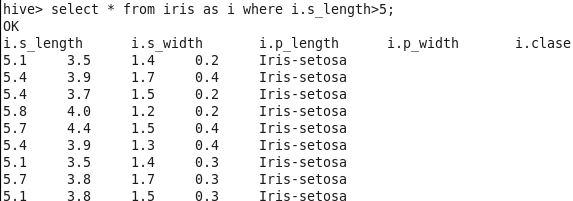
1. Comprobar que la table tiene datos



1. Mostrar las 5 primeras filas de la tabla iris
   1. Select \* from iris limit 5;



1. 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
   1. Select \* from iris as i where i.s\_length>5

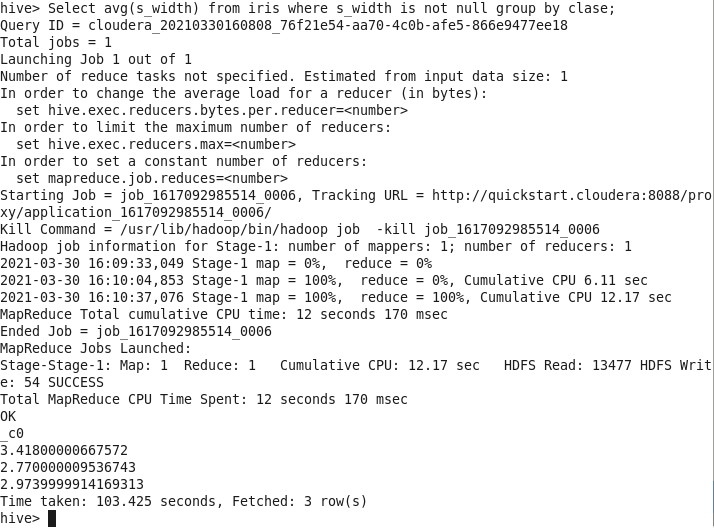


1. Seleccionar la media de s\_width agrupados por clase. Observad que ahora el tiempo de ejecución aumenta considerablemente.
   1. Select avg(s\_width) from iris GROUP BY clase;

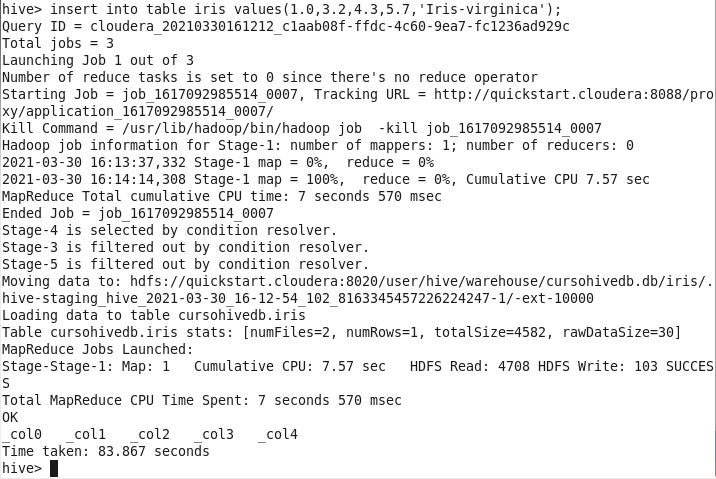


1. Pregunta: vemos que aparece un valor NULL como resultado en la query anterior. ¿Por qué? ¿cómo los eliminarías?

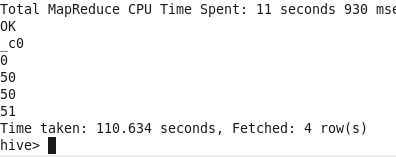
Porque había algún dato erróneo, no numérico o nulo en el campo de alguna clase. Para eliminarlos podríamos añadir la condición where para que fuera distinto de null.



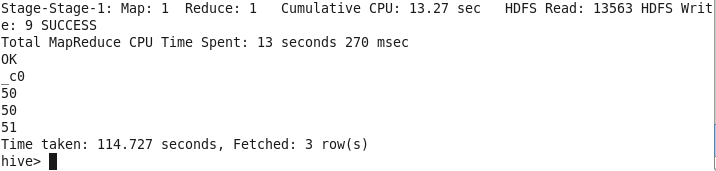
1. Insertar en la tabla la siguiente fila (1.0,3.2,4.3,5.7,"Iris-virginica")
   1. Insert into table iris values (1.0,3.2,4.3,5.7,"Iris-virginica")



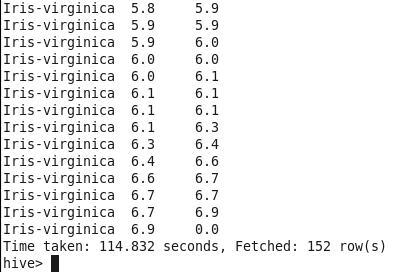
1. Contar el número de ocurrencias de cada clase
   1. Select count(clase) from iris group by clase;



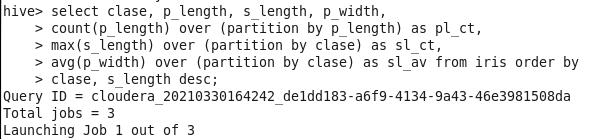
1. Seleccionar las clases que tengan más de 45 ocurrencias
   1. Select clase from iris group by clase having count(\*)>45;

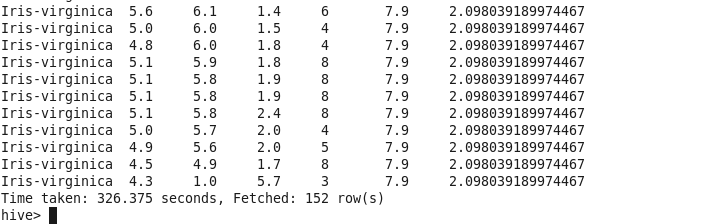


1. Utilizando la función LEAD, ejecutar una query que devuelva la clase, p\_length y el LEAD de p\_length con Offset=1 y Default\_Value =0, particionado por clase y ordenado por p\_length.
   1. select clase, p\_length, LEAD(p\_length,1,0) OVER (PARTITION BY clase ORDER BY p\_length) as Lead from iris;



1. 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.
   1. 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;





Preguntas:

1. Cita algunas razones por las que no reemplazarías una RDBM por Hive

Si la base de datos es relativamente pequeña y no se van a trabajar con grandes cantidades de datos, no tendría sentido hacer el cambio a Hive.

Tampoco si se quiere poder trabajar con la base de datos en tiempo real.

2. Cuáles son los beneficios de Hive y Hadoop sobre DWH tradicionales

DWH solo procesa datos estructurados, mientras que Hadoop puede procesar todo tipo de datos(Estructurados, semi-estructurados, no-estructurados)

Hive y Hadoop funcionan mejor con grandes cantidades de datos.

3. Qué datos almacena el metastore de hive

Se almacena toda la información referente a las tablas y sus relaciones

4.Cuando hacemos una consulta en hive sobre una tabla, dónde reside físicamente esa tabla

Reside en el sistema de datos de hadoop, HDFS. Por defecto en la ruta /user/hive/warehouse.

5.Qué comando se usa para cambiar el foco a otra tabla en hive

Mediante el comando SELECT … FROM se puede especificar la tabla de la que quieres ver los datos.

6.Cuál es el comando usado para combinar el resultado de varias queries en un solo resultado

Concat()

7.Cuál es el directorio por defecto del warehouse de hive

Hive almacena la información por defecto en /user/hive/warehouse del sistema HDFS

8.Dónde se almacenan las tablas particionadas en Hive

Si la tabla particionada está almacenada en diferentes localizaciones, estas se pueden ver con el siguiente comando:

DESCRIBE FORMATTED db\_name.table\_name PARTITION (name=value)

9.Cual es la diferencia entre el tipo de datos SequenceFile y Parquet

10.Cuál es la diferencia entre Arrays y Maps

Los Maps son conjuntos de elementos clave-valor, mientras que los Arrays solo están formados por valores sin ninguna clave que haga referencia a ellos.

11.Cuál es la query más rápida en Hive

La query más rápida sería aquella que consulta el metastore para obtener los datos, ya que así no tiene que hacer tareas map o reduce, como por ejemplo “show tables;” De las queries que consultan las filas de la tabla sería “select \* from table”, la cual no tiene que hacer operaciones reduce, ya que mostramos todos los datos.