

HDFS con hadoop

Se puede poner `hdfs` en vez de `hadoop fs`, es equivalente.

Si ejecutas “`hadoop fs -ls`” puedes ver los archivos y directorios en formato HDFS.

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 5 items
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:49 count2
drwxr-xr-x - cloudera cloudera      0 2021-05-14 05:01 hive
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:31 pwords
drwxr-xr-x - cloudera cloudera      0 2021-05-13 05:16 shakespeare
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:12 wordcounts
[cloudera@quickstart ~]$
```

Si queremos poner un archivo en HDFS tendremos que ejecutar “`hadoop fs -put (nombre del archivo)`”

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 6 items
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:49 count2
drwxr-xr-x - cloudera cloudera      0 2021-05-14 05:01 hive
drwxr-xr-x - cloudera cloudera      0 2021-05-20 02:56 prueba
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:31 pwords
drwxr-xr-x - cloudera cloudera      0 2021-05-13 05:16 shakespeare
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:12 wordcounts
[cloudera@quickstart ~]$
```

Si queremos borrar una carpeta tenemos que ejecutar un “`hadoop fs -rm -R (nombre del directorio)`”

Tras la ejecución podremos ver que se ha borrado en HDFS pero si hacemos un “`ls`” normal y corriente sigue estando ahí.

```
[cloudera@quickstart ~]$ hadoop fs -rm -R prueba/
Deleted prueba
[cloudera@quickstart ~]$ ls
cloudera-manager enterprise-deployment.json Music tabla_prueba.java
cm_api.py         example2.conf   parcels         Templates
Desktop          example3.conf  Pictures       Videos
Documents        express-deployment.json pig_1621245510622.log wordcount
Downloads        iplistw       prueba         workspace
eclipse          kerberos
ejercicios       lib
[cloudera@quickstart ~]$ hadoop fs -ls
Found 5 items
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:49 count2
drwxr-xr-x - cloudera cloudera      0 2021-05-14 05:01 hive
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:31 pwords
drwxr-xr-x - cloudera cloudera      0 2021-05-13 05:16 shakespeare
drwxr-xr-x - cloudera cloudera      0 2021-05-14 03:12 wordcounts
[cloudera@quickstart ~]$
```

Si queremos copiar el contenido de un archivo a otro utilizaremos un “`hadoop fs -get (ruta archivo origen) (ruta archivo destino)`”

```
[cloudera@quickstart ~]$ hadoop fs -get /user/cloudera/shakespeare/poems /home/cloudera/ejercicios/shakespeare/shakepoems.txt
get: `/home/cloudera/ejercicios/shakespeare/shakepoems.txt': File exists
[cloudera@quickstart ~]$
```

Hive

Para crear y usar una base de datos:

```
hive> create database test;
OK
Time taken: 0.353 seconds
hive> use test;
OK
Time taken: 0.038 seconds
```

Para mostrar las tablas de las que se compone la base de datos

```
hive> show tables;
OK
tab_name
Time taken: 0.23 seconds
```

Para crear una tabla:

```
hive> drop table prueba;
OK
Time taken: 0.105 seconds
hive> create table prueba(
  > s_length float,
  > s_width float,
  > p_length float,
  > p_width float,
  > clase string
  > )
  > row format delimited
  > fields terminated by ',';
OK
Time taken: 0.241 seconds
hive> show tables;
OK
tab_name
prueba
Time taken: 0.044 seconds, Fetched: 1 row(s)
```

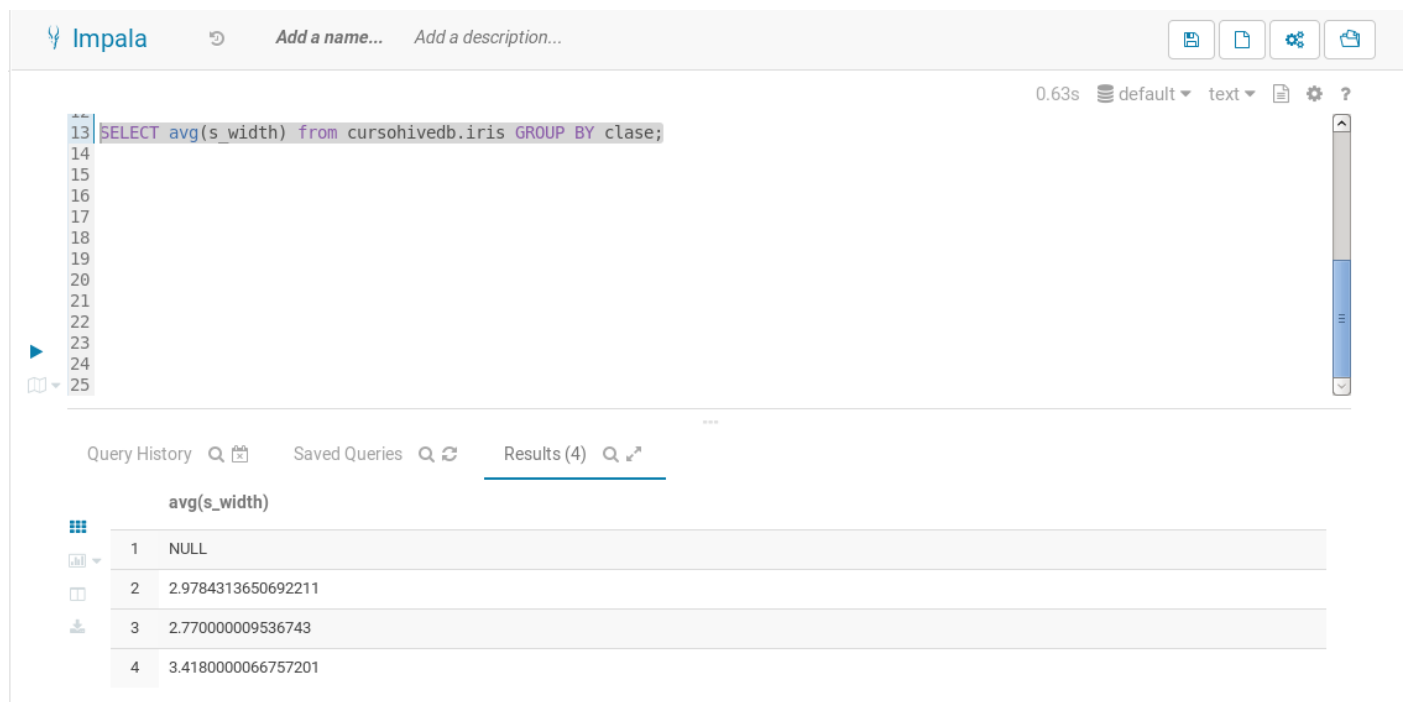
Para mostrar los campos de una tabla:

```
hive> desc prueba
  > ;
OK
col_name      data_type      comment
s_length      float
s_width        float
p_length       float
p_width        float
clase          string
Time taken: 0.128 seconds, Fetched: 5 row(s)
hive> █
```

Realizar consultas:

```
hive> select avg(s_width) from iris group by clase;
Query ID = cloudera_20210520090808_8746820d-8b6a-4b51-a4e7-76b2c6f9706e
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_1621520656632_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1621520656632_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1621520656632_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-20 09:09:13,581 Stage-1 map = 0%, reduce = 0%
2021-05-20 09:09:23,384 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.01 sec
2021-05-20 09:09:30,782 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 1.98 sec
MapReduce Total cumulative CPU time: 1 seconds 980 msec
Ended Job = job_1621520656632_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 1.98 sec HDFS Read: 13220 HDFS Write: 57 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 980 msec
OK
NULL
3.41800000667572
2.770000009536743
2.9739999914169313
Time taken: 33.947 seconds, Fetched: 4 row(s)
```

La misma consulta en impala:



The screenshot shows the Impala web interface. At the top, there's a header with the Impala logo and some navigation links. Below the header, there's a text area where a SQL query is entered: `SELECT avg(s_width) from cursohivedb.iris GROUP BY clase;`. To the right of the query, there's a status bar showing "0.63s" and some icons. Below the query, there's a section titled "Results (4)" which displays a table with 4 rows. The table has a single column labeled "avg(s_width)". The rows contain the values: NULL, 2.9784313650692211, 2.770000009536743, and 3.4180000066757201.

avg(s_width)
1 NULL
2 2.9784313650692211
3 2.770000009536743
4 3.4180000066757201

PIG

Pig es un lenguaje de transformación de datos.

Se abre con `pig -x local`

Luego estableces una variable en la que almacenas los campos y su tipo:

Para filtrar los datos creamos otra variable en la que se le indica los parámetros para ello:

```
grunt> result = filter data by pais=='USA'  
>> ;
```

Ahora si queremos ordenar los campos almacenaremos los datos de nuestra variable en otra en la que establezcamos nosotros el orden de los campos y guardamos el resultado en una ruta:

```
grunt> ordenado = foreach result generate campana, fecha, tiempo, key, display, lugar, accion, cpc;  
grunt> store ordenado into '/home/cloudera/ejercicios/pig/resultado';
```

Si vamos a la ruta:

```
[cloudera@quickstart ~]$ ls /home/cloudera/ejercicios/pig/resultado/  
part-m-000000 SUCCESS  
[cloudera@quickstart ~]$
```

Sqoop

Para crear una base de datos y poder verla en sqoop se hace de la siguiente manera:

Se crea mediante "create database (nombre de la base)"

Para mostrar las bases de datos creadas se introduce "show databases"

```
mysql> create database testdb
-> ;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> show databases;
```

Database
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
oozie
pruebadb
retail_db
rman
sentry
testdb

```
14 rows in set (0.00 sec)
```

```
mysql> █
```

Funciona de una manera muy similar a cualquier sintaxis de sql, de forma que se utiliza un “use” para seleccionar la tabla, un “create table” para crear una tabla y un “show tables” para ver las tablas:

```
mysql> use testdb
```

```
Database changed
```

```
mysql> create table tabla_test (nombre varchar(30), edad int);
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> show tables;
```

Tables_in_testdb
tabla_test

```
1 row in set (0.00 sec)
```

Se pueden realizar tambien inserts:

```
mysql> INSERT INTO tabla_test VALUES ("Alberto",22);INSERT INTO tabla_test VALUES ("Luis", 23);INSERT INTO tabla_test VALUES ("Pablo", 24);
Query OK, 1 row affected (0.00 sec)
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Query OK, 1 row affected (0.00 sec)
```

Así como consultas select:

```
mysql> select * from tabla_test;
```

nombre	edad
Alberto	22
Luis	23
Pablo	24

```
3 rows in set (0.00 sec)
```

Si escribes “describe (nombre de la tabla)” puedes ver los detalles de la tabla:

```
mysql> describe tabla_test;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(30)	YES		NULL	
edad	int(11)	YES		NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> █
```