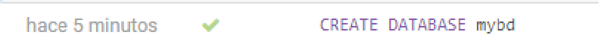
3. Hive (Parte 1)

miércoles, 29 de abril de 2020 15:57

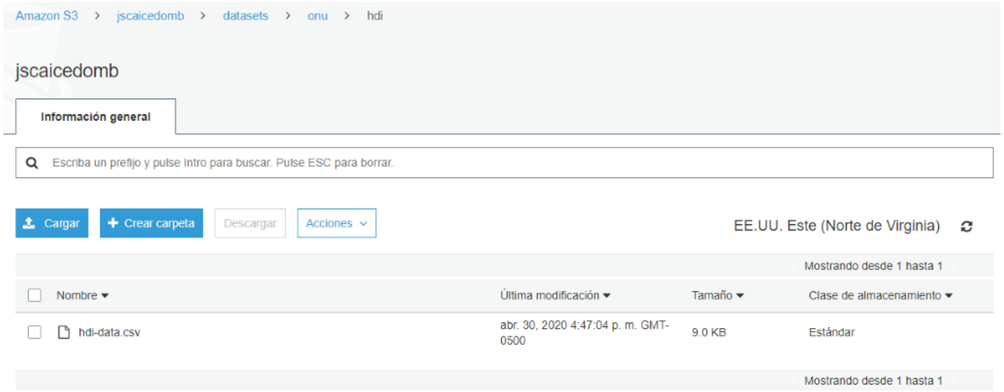
- **Crear base de datos, crear tablas 'hdi' y 'wordcount':**

Creamos la base de datos



```
hace 5 minutos ✓ CREATE DATABASE mybd
```

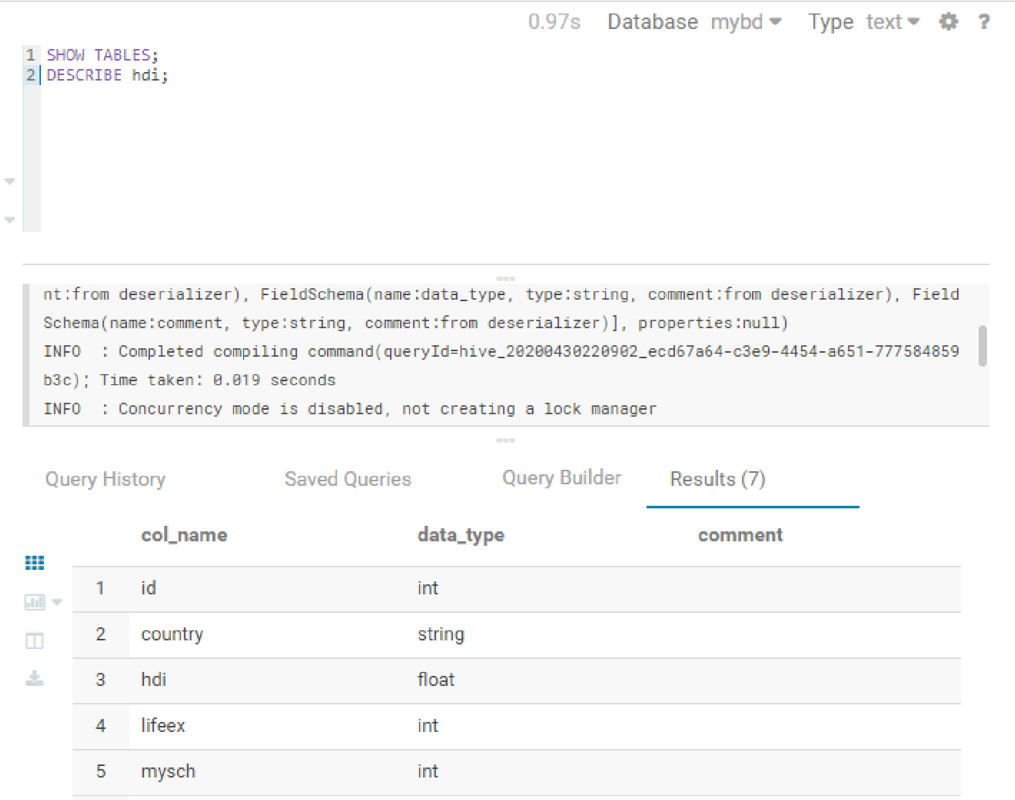
Actualizamos los datasets de s3



Amazon S3 › jscaicedomb › datasets › onu › hdi
jscaicedomb
Información general
Escriba un prefijo y pulse Intro para buscar. Pulse ESC para borrar.
Cargar
Crear carpeta
Descargar
Acciones
EE.UU. Este (Norte de Virginia)
Mostrando desde 1 hasta 1

☐	Nombre	Última modificación	Tamaño	Clase de almacenamiento
☐	hdi-data.csv	abr. 30, 2020 4:47:04 p. m. GMT-0500	9.0 KB	Estándar

Creamos la tabla con datos de s3



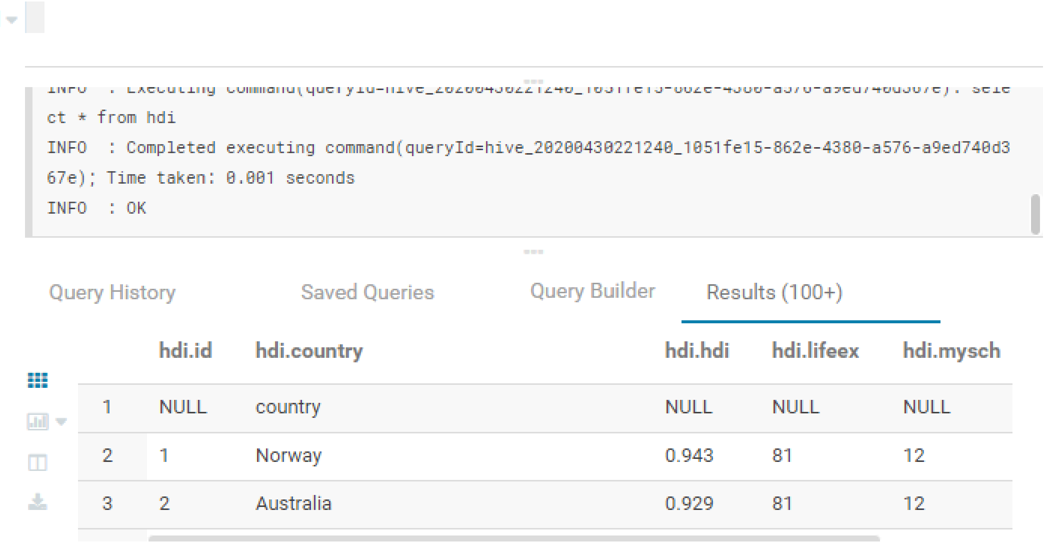
```
SHOW TABLES;
DESCRIBE hdi;
```

```
nt:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), Field
Schema(name:comment, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20200430220902_ecd67a64-c3e9-4454-a651-777584859
b3c); Time taken: 0.019 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
```

	col_name	data_type	comment
1	id	int	
2	country	string	
3	hdi	float	
4	lifeex	int	
5	mysch	int	

Verificamos que tenga datos con un select

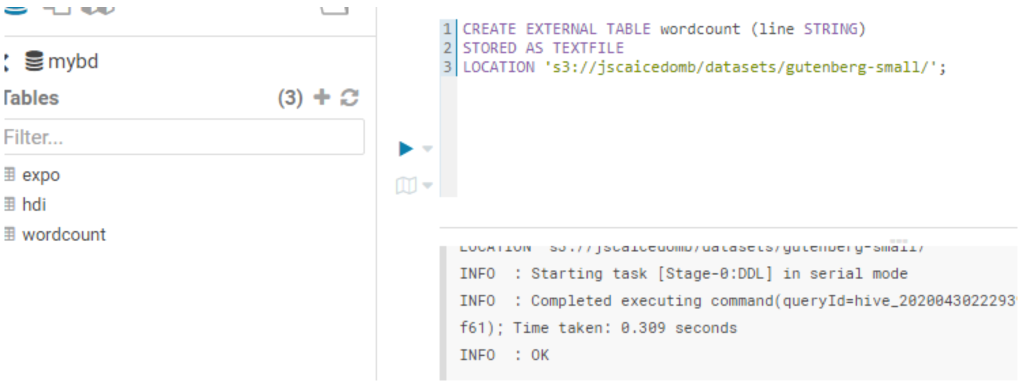
```
select * from hdi;
```



```
INFO  : Executing command(queryId=hive_20200430221240_1051fe15-862e-4380-a576-a9ed740d367e): sele
ct * from hdi
INFO  : Completed executing command(queryId=hive_20200430221240_1051fe15-862e-4380-a576-a9ed740d3
67e); Time taken: 0.001 seconds
INFO  : OK
```

	hdi.id	hdi.country	hdi.hdi	hdi.lifeex	hdi.mysch
1	NULL	country	NULL	NULL	NULL
2	1	Norway	0.943	81	12
3	2	Australia	0.929	81	12

Creamos la tabla wordcount que tendrá todos los datos de gutenbergsmall

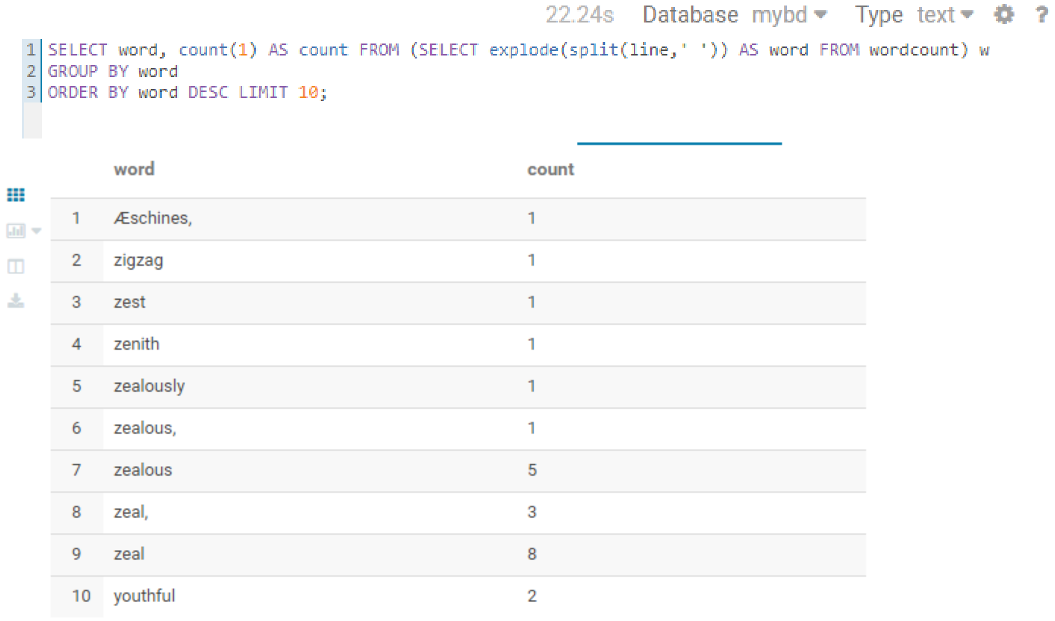


```
CREATE EXTERNAL TABLE wordcount (line STRING)
STORED AS TEXTFILE
LOCATION 's3://jscaicedomb/datasets/gutenberg-small/';
```

```
LOCATION 's3://jscaicedomb/datasets/gutenberg-small/
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_2020043022293
f61); Time taken: 0.309 seconds
INFO  : OK
```

- Wordcount

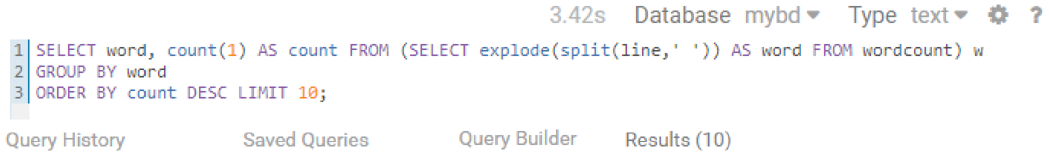
Ordenado por palabra



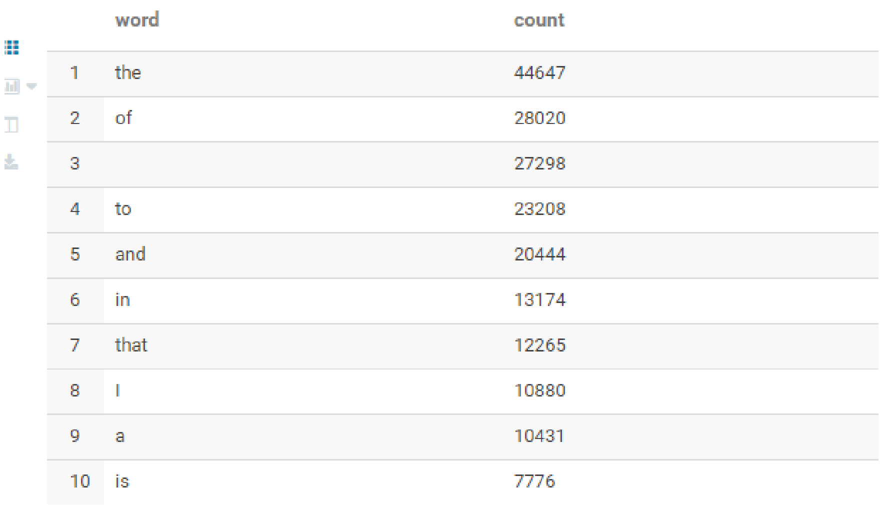
```
SELECT word, count(1) AS count FROM (SELECT explode(split(line,' ')) AS word FROM wordcount) w
GROUP BY word
ORDER BY word DESC LIMIT 10;
```

	word	count
1	Æschines,	1
2	zigzag	1
3	zest	1
4	zenith	1
5	zealously	1
6	zealous,	1
7	zealous	5
8	zeal,	3
9	zeal	8
10	youthful	2

Ordenado de mayor a menor por frecuencia



```
SELECT word, count(1) AS count FROM (SELECT explode(split(line,' ')) AS word FROM wordcount) w
GROUP BY word
ORDER BY count DESC LIMIT 10;
```

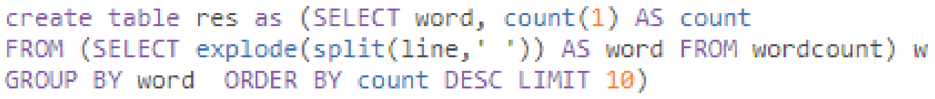


	word	count
1	the	44647
2	of	28020
3		27298
4	to	23208
5	and	20444
6	in	13174
7	that	12265
8	I	10880
9	a	10431
10	is	7776

RETO:

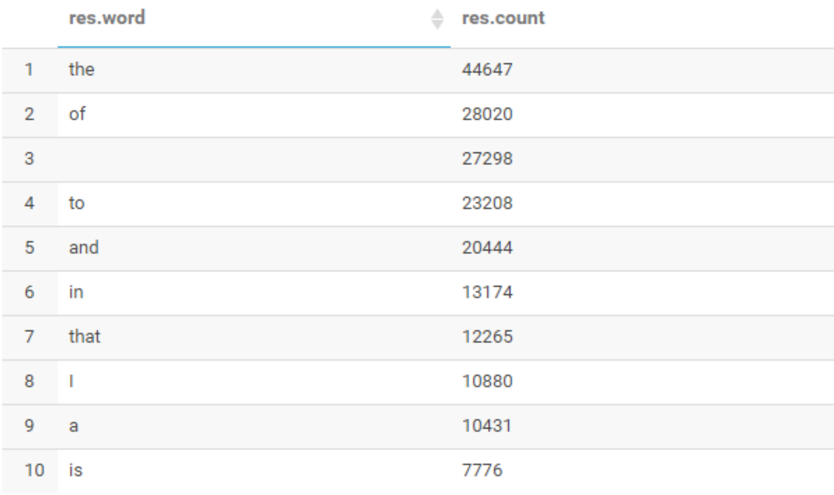
¿cómo llenar una tabla con los resultados de un Query? por ejemplo, como almacenar en una tabla el diccionario de frecuencia de palabras en el wordcount?

Creamos una tabla y la guardamos como los resultados del query de frecuencia de mayor a menor de los 10 primeros datos.



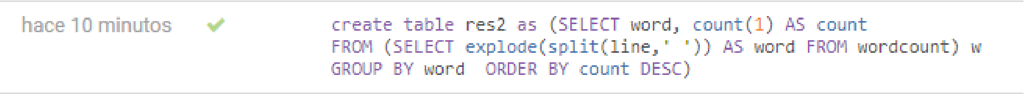
```
create table res as (SELECT word, count(1) AS count
FROM (SELECT explode(split(line,' ')) AS word FROM wordcount) w
GROUP BY word  ORDER BY count DESC LIMIT 10)
```

Mostramos con Select * from res



	res.word	res.count
1	the	44647
2	of	28020
3		27298
4	to	23208
5	and	20444
6	in	13174
7	that	12265
8	I	10880
9	a	10431
10	is	7776

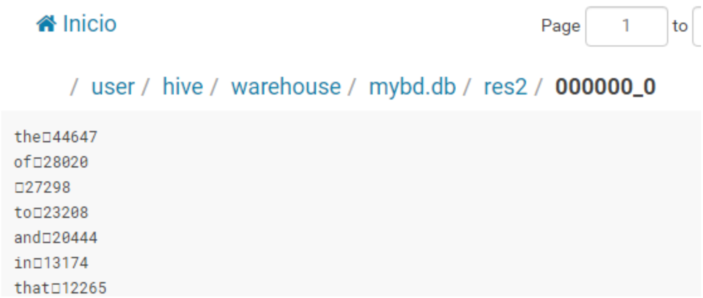
Ahora creamos una tabla res2 con todos los registros



hace 10 minutos ✓

```
create table res2 as (SELECT word, count(1) AS count
FROM (SELECT explode(split(line,' ')) AS word FROM wordcount) w
GROUP BY word  ORDER BY count DESC)
```

Verificamos donde se guarda por defecto los resultados



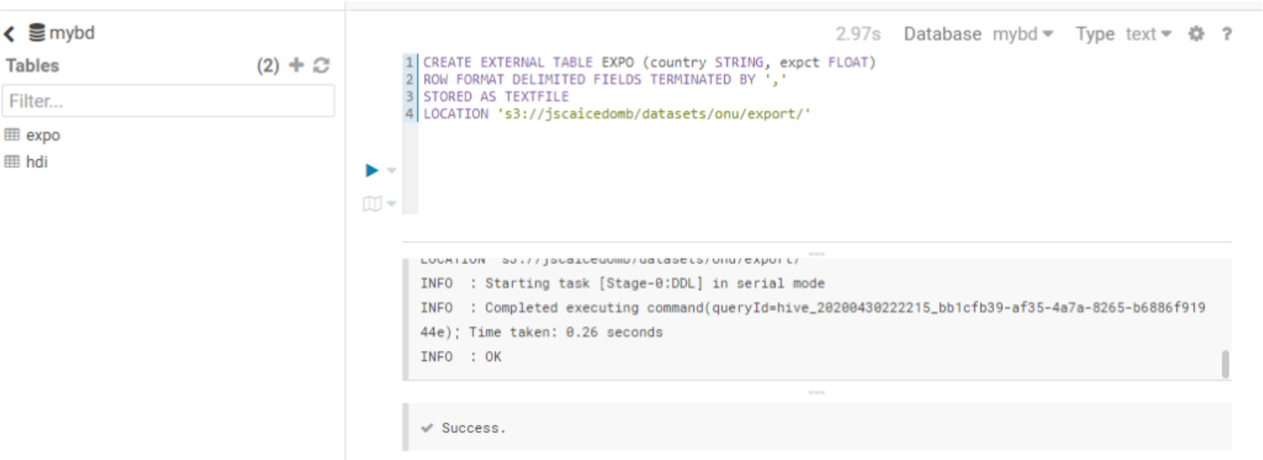
/ user / hive / warehouse / mybd.db / res2 / 000000_0

```
the□44647
of□28020
□27298
to□23208
and□20444
in□13174
that□12265
```

```
I□10880
a□10431
is□7776
be□7148
it□6899
as□6473
not□5920
for□5658
have□5060
by□4571
you□4328
he□4111
```

- ## Realizar consultas SQL

Se crea la tabla expo con los datos en s3

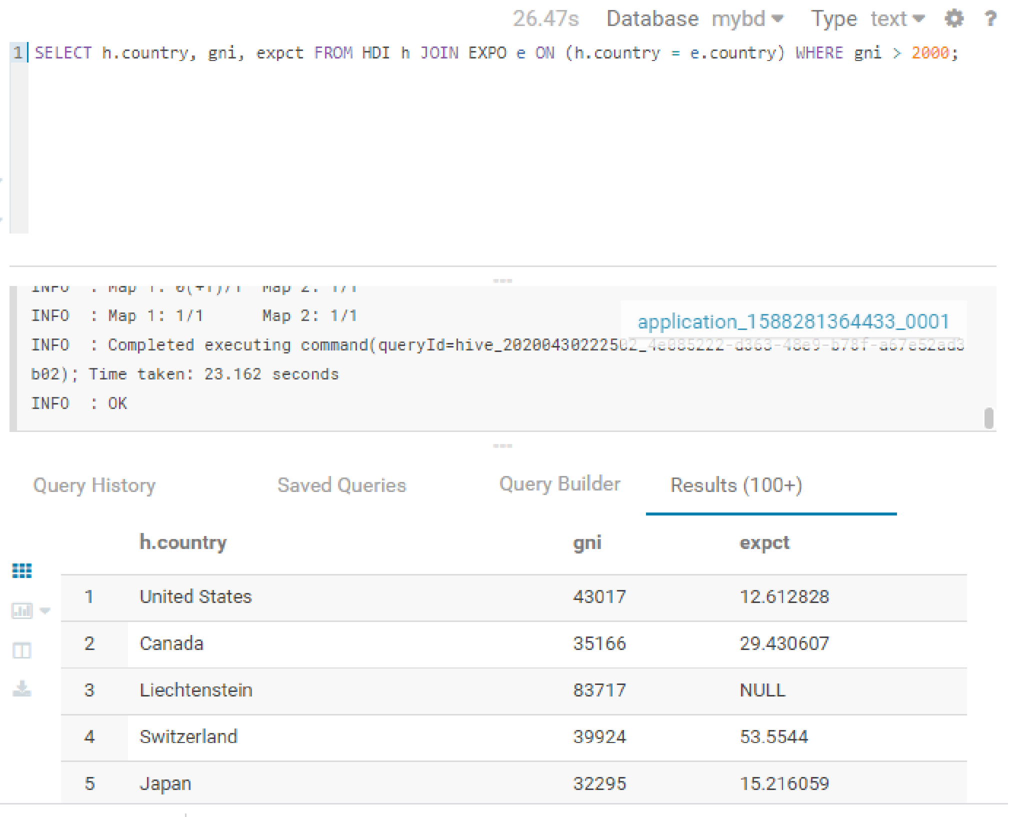


```
CREATE EXTERNAL TABLE EXPO (country STRING, expct FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 's3://jscaicedomb/datasets/onu/export/'
```

```
LOCATION 's3://jscaicedomb/datasets/onu/export/'
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200430222215_bb1cfb39-af35-4a7a-8265-b6886f919
44e); Time taken: 0.26 seconds
INFO  : OK
```

✓ Success.

Hacemos un join


26.47s Database mybd Type text
application_1588281364433_0001
Query History
Saved Queries
Query Builder
Results (100+)

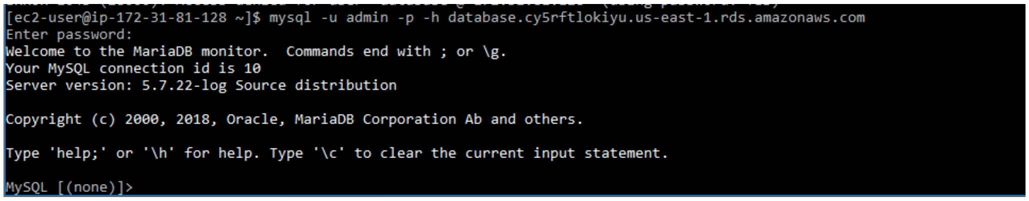
```
SELECT h.country, gni, expct FROM HDI h JOIN EXPO e ON (h.country = e.country) WHERE gni > 2000;
```

```
INFO  : Map 1: 0(+1)/1    Map 2: 1/1
INFO  : Map 1: 1/1      Map 2: 1/1
INFO  : Completed executing command(queryId=hive_20200430222502_4e085222-d363-48e9-b78f-a67e52ad3
b02); Time taken: 23.162 seconds
INFO  : OK
```

	h.country	gni	expct
1	United States	43017	12.612828
2	Canada	35166	29.430607
3	Liechtenstein	83717	NULL
4	Switzerland	39924	53.5544
5	Japan	32295	15.216059

- ## Transferir datos vía Sqoop de la base de datos:'cursodb' y tabla: 'employee'

Creo una instancia ec2 para el cliente mysql y una base de datos rds mysql y me conecto a la base de datos por medio de mi cliente sql:

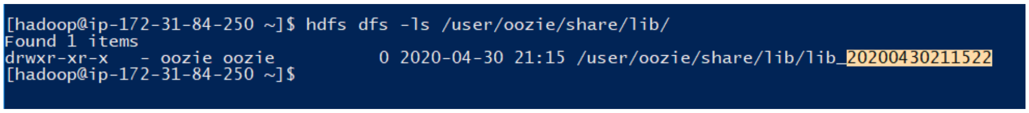


```
[ec2-user@ip-172-31-81-128 ~]$ mysql -u admin -p -h database.cy5rftlokiyu.us-east-1.rds.amazonaws.com
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.22-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

PARA SQOOP DESDE HUE EN EMR-AMAZON

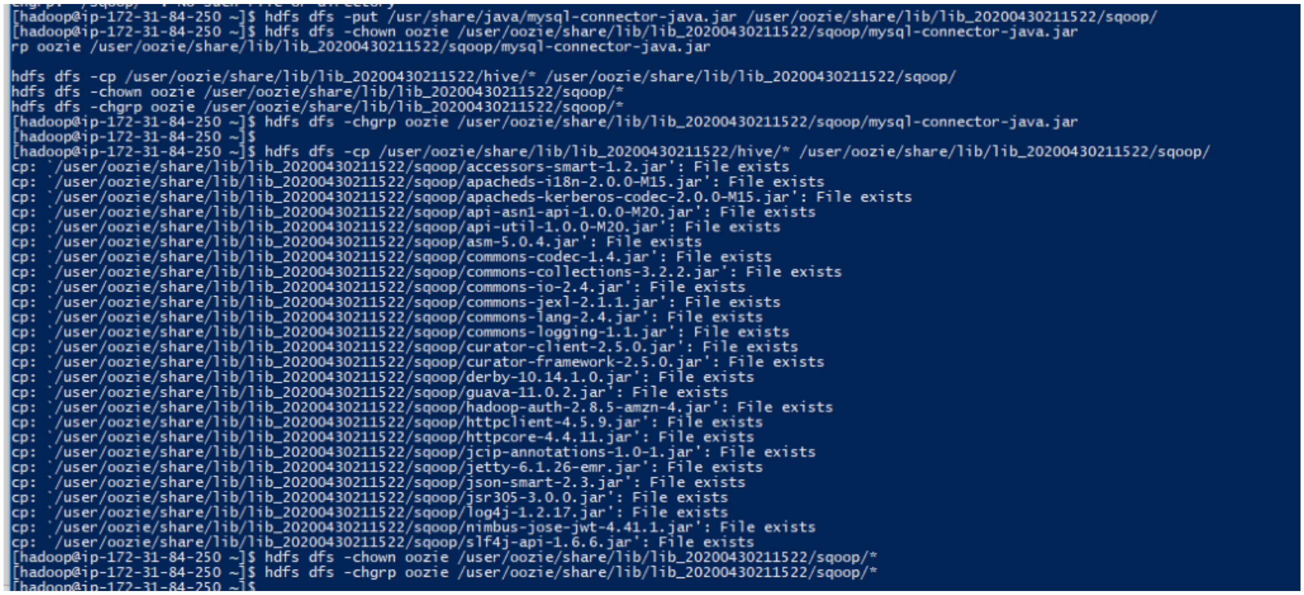
Para que funcione SQOOP desde la interfaz web HUE se hace la siguiente configuración:

Se busca la lib correspondiente al clúster, conectándonos primero vía ssh y luego listando lib



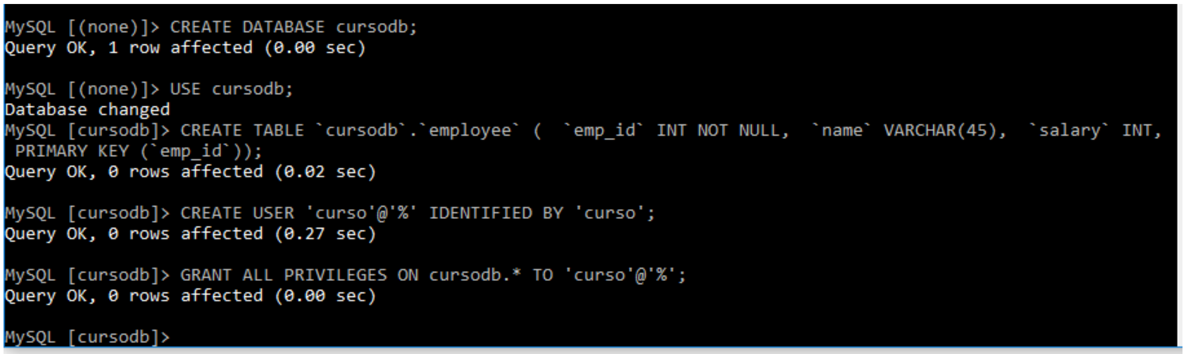
```
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -ls /user/oozie/share/lib/
Found 1 items
drwxr-xr-x   - oozie oozie          0 2020-04-30 21:15 /user/oozie/share/lib/lib_20200430211522
[hadoop@ip-172-31-84-250 ~]$
```

Copiamos el número y lo reemplazamos en la configuración en settings-emr.txt



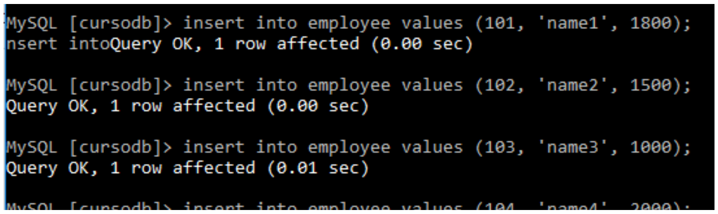
```
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -put /usr/share/java/mysql-connector-java.jar /user/oozie/share/lib/lib_20200430211522/sqoop/
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20200430211522/sqoop/mysql-connector-java.jar
rp oozie /user/oozie/share/lib/lib_20200430211522/sqoop/mysql-connector-java.jar

hdfs dfs -cp /user/oozie/share/lib/lib_20200430211522/hive/* /user/oozie/share/lib/lib_20200430211522/sqoop/
hdfs dfs -chown oozie /user/oozie/share/lib/lib_20200430211522/sqoop/*
hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20200430211522/sqoop/*
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20200430211522/sqoop/mysql-connector-java.jar
[hadoop@ip-172-31-84-250 ~]$
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -cp /user/oozie/share/lib/lib_20200430211522/hive/* /user/oozie/share/lib/lib_20200430211522/sqoop/
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/accessors-smart-1.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/apacheds-i18n-2.0.0-M15.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/apacheds-kerberos-codec-2.0.0-M15.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/api-asn1-api-1.0.0-M20.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/api-util-1.0.0-M20.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/asm-5.0.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-codec-1.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-collections-3.2.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-io-2.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-jexl-2.1.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-lang-2.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/commons-logging-1.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/curator-client-2.5.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/curator-framework-2.5.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/derby-10.14.1.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/guava-11.0.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/hadoop-auth-2.8.5-amzn-4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/httpclient-4.5.9.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/httpcore-4.4.11.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/jcip-annotations-1.0-1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/jetty-6.1.26-emr.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/json-smart-2.3.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/jsr305-3.0.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/log4j-1.2.17.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/nimbus-jose-jwt-4.41.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200430211522/sqoop/slf4j-api-1.6.6.jar': File exists
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20200430211522/sqoop/*
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20200430211522/sqoop/*
[hadoop@ip-172-31-84-250 ~]$
```

Creamos la base de datos cursodb y la tabla employee

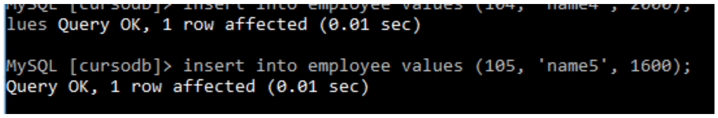


```
MySQL [(none)]> CREATE DATABASE cursodb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> USE cursodb;
Database changed
MySQL [cursodb]> CREATE TABLE `cursodb`.`employee` (  `emp_id` INT NOT NULL,  `name` VARCHAR(45),  `salary` INT,
 PRIMARY KEY (`emp_id`));
Query OK, 0 rows affected (0.02 sec)

MySQL [cursodb]> CREATE USER 'curso'@'%' IDENTIFIED BY 'curso';
Query OK, 0 rows affected (0.27 sec)

MySQL [cursodb]> GRANT ALL PRIVILEGES ON cursodb.* TO 'curso'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [cursodb]>
```

Llenamos la tabla según la configuración dada en scripts-rdbms

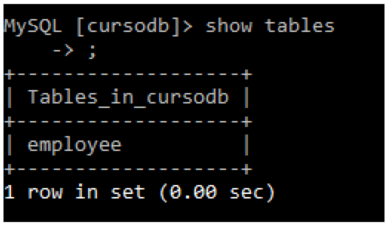


```
MySQL [cursodb]> insert into employee values (101, 'name1', 1800);
nsert intoQuery OK, 1 row affected (0.00 sec)

MySQL [cursodb]> insert into employee values (102, 'name2', 1500);
Query OK, 1 row affected (0.00 sec)

MySQL [cursodb]> insert into employee values (103, 'name3', 1000);
Query OK, 1 row affected (0.01 sec)

MySQL [cursodb]> insert into employee values (104, 'name4', 2000);
```



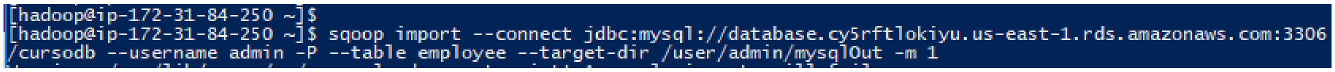
```
MySQL [cursodb]> insert into employee values (104, 'name4', 2000);
lues Query OK, 1 row affected (0.01 sec)

MySQL [cursodb]> insert into employee values (105, 'name5', 1600);
Query OK, 1 row affected (0.01 sec)
```

Vemos la tabla creada



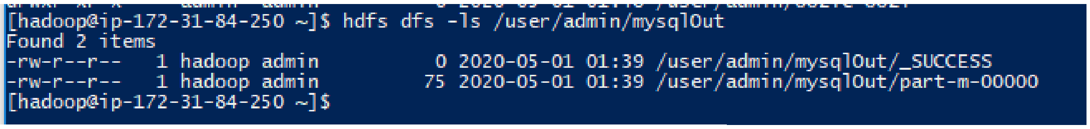
```
MySQL [cursodb]> show tables
    -> ;
+-------------------+
| Tables_in_cursodb |
+-------------------+
| employee          |
+-------------------+
1 row in set (0.00 sec)
```

//Transferir datos de una base de datos (tipo mysql) hacia HDFS:

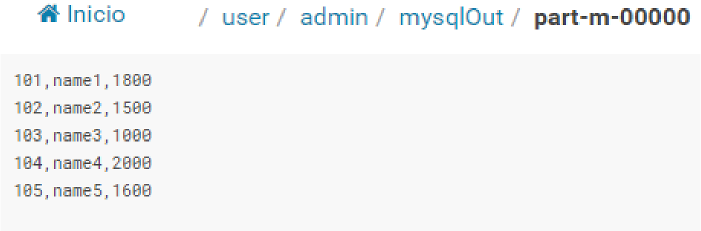


```
[hadoop@ip-172-31-84-250 ~]$
[hadoop@ip-172-31-84-250 ~]$ sqoop import --connect jdbc:mysql://database.cy5rftlokiyu.us-east-1.rds.amazonaws.com:3306
/cursodb --username admin -P --table employee --target-dir /user/admin/mysqlOut -m 1
```

Vemos que ya están en hdfs

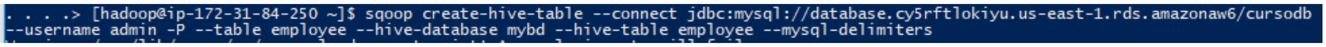


```
[hadoop@ip-172-31-84-250 ~]$ hdfs dfs -ls /user/admin/mysqlOut
Found 2 items
-rw-r--r--   1 hadoop admin          0 2020-05-01 01:39 /user/admin/mysqlOut/_SUCCESS
-rw-r--r--   1 hadoop admin         75 2020-05-01 01:39 /user/admin/mysqlOut/part-m-00000
[hadoop@ip-172-31-84-250 ~]$
```



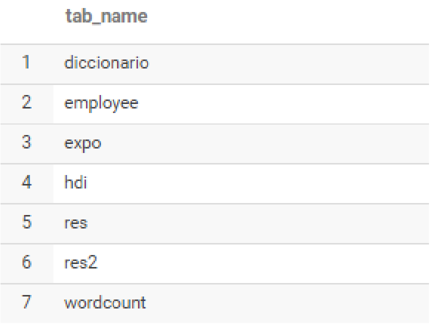
```
101,name1,1800
102,name2,1500
103,name3,1000
104,name4,2000
105,name5,1600
```

// Crear tabla HIVE a partir de definición tabla Mysql:



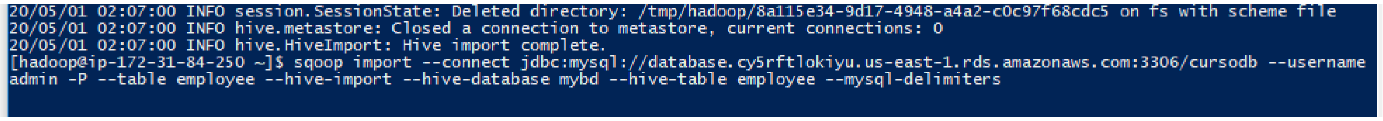
```
. . . .> [hadoop@ip-172-31-84-250 ~]$ sqoop create-hive-table --connect jdbc:mysql://database.cy5rftlokiyu.us-east-1.rds.amazonaw6/cursodb
--username admin -P --table employee --hive-database mybd --hive-table employee --mysql-delimiters
```

Vemos que employee esté en hue

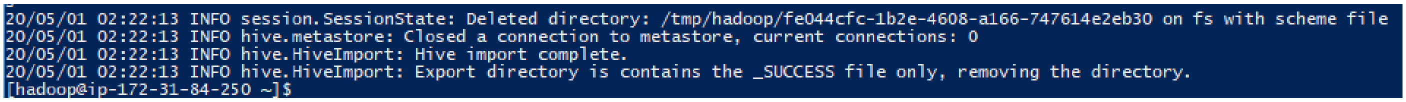


	tab_name
1	diccionario
2	employee
3	expo
4	hdi
5	res
6	res2
7	wordcount

// Transferir datos de una base de datos (tipo mysql) hacia HIVE vía HDFS:

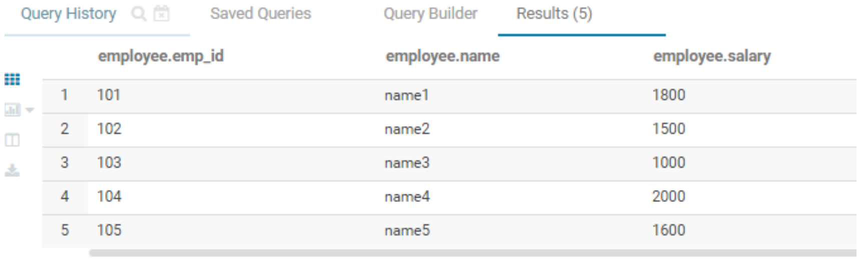


```
20/05/01 02:07:00 INFO session.SessionState: Deleted directory: /tmp/hadoop/8a115e34-9d17-4948-a4a2-c0c97f68cdc5 on fs with scheme file
20/05/01 02:07:00 INFO hive.metastore: Closed a connection to metastore, current connections: 0
20/05/01 02:07:00 INFO hive.HiveImport: Hive import complete.
[hadoop@ip-172-31-84-250 ~]$ sqoop import --connect jdbc:mysql://database.cy5rftlokiyu.us-east-1.rds.amazonaws.com:3306/cursodb --username
admin -P --table employee --hive-import --hive-database mybd --hive-table employee --mysql-delimiters
```



```
20/05/01 02:22:13 INFO session.SessionState: Deleted directory: /tmp/hadoop/fe044cfc-1b2e-4608-a166-747614e2eb30 on fs with scheme file
20/05/01 02:22:13 INFO hive.metastore: Closed a connection to metastore, current connections: 0
20/05/01 02:22:13 INFO hive.HiveImport: Hive import complete.
20/05/01 02:22:13 INFO hive.HiveImport: Export directory is contains the _SUCCESS file only, removing the directory.
[hadoop@ip-172-31-84-250 ~]$
```

Vemos que se copiaron bien con select * from employee



	employee.emp_id	employee.name	employee.salary
1	101	name1	1800
2	102	name2	1500
3	103	name3	1000
4	104	name4	2000
5	105	name5	1600