Hive & Sqoop

miércoles, 29 de abril de 2020 15:52

- Conexión con la DB

```
[ec2-user@ip-172-31-93-194 ~]$ mysql -u admin -p -h database-1.czsfk9ugozci.us-east-1.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2152
Server version: 8.0.17 Source distribution

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

- Crear base de datos

hace 2 horas ✔

```
CREATE DATABASE mybd
```

- Crear tabla "hdi" con datos de s3

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

```
INFO  : Compiling command(queryId=hive_20200501031410_4ba81de5-8712-42f4-b15e-0516e72c7be0):
DESCRIBE hdi
```

Query History | Saved Queries | Query Builder | Results (7)

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

- Consultas y cálculos sobre la tabla HDI:

```
select * from hdi;
```

```
INFO  : Completed executing command(queryId=hive_20200501031601_b3b8c146-75cd-4914-abe2-88e81a6d88d7); Time taken: 0.001 seconds
INFO  : OK
```

Query History | Saved Queries | Query Builder | Results (100+)

	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
4	3	Netherlands	0.91	80	11
5	4	United States	0.91	78	12
6	5	New Zealand	0.908	80	12
7	6	Canada	0.908	81	12

```
select country, gni from hdi where gni > 2000;
```

```
INFO  : Compiling command(queryId=hive_20200501031816_132d0c2f-daa7-455e-baa4-d3c997964553): select country, gni from hdi where gni > 2000
INFO  : Semantic Analysis Completed
```

Query History | Saved Queries | Query Builder | Results (100+)

country	gni

1	Norway	47557
2	Australia	34431

- Ejecutar join con hive

  Se crea la tabla expo

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

  Join

WORDCOUNT EN HIVE:

- Creamos la tabla que tendrá todos los datos de gutenbergsmall

```
hace unos segundos ✔
CREATE EXTERNAL TABLE docs (line STRING)
STORED AS TEXTFILE
LOCATION
's3://amartinezvdatasets/datasets/onu/datasets/gutenberg-
small/'
```

- ordenado por palabra

```
27.53s  Database mybd  Type text
1 SELECT word, count(1) AS count FROM (SELECT explode(split(line,' ')) AS word FROM docs) w
2 GROUP BY word
3 ORDER BY word DESC LIMIT 10;
```

```
INFO  : Compiling command(queryId=hive_20200501032752_986
2): SELECT word, count(1) AS count FROM (SELECT explode(split(line,
w
```

application_1588297838585_0001

Query History | Saved Queries | Query Builder | Results (10)

	word	count
1	Æschines,	1
2	zigzag	1
3	zest	1
4	zenith	1
5	zealously	1
6	zealous,	1

- ordenado por frecuencia de menor a mayor

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

```
INFO  : Compiling command(queryId=hive_20200501033003_c3b
f): SELECT word, count(1) AS count FROM (SELECT explode(split(line,
w
```

application_1588297838585_0001

Query History | Saved Queries | Query Builder | Results (10)

	word	count
1	the	44647
2	of	28020
3		27298
4	to	23208
5	and	20444
6	in	13174

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 guardamos los resultados ahí:

```
hace 4 minutos ✔
CREATE TABLE result as (SELECT word, count(1) AS count
FROM (SELECT explode(split(line, ' ')) AS word FROM docs) w
GROUP BY word ORDER BY count DESC LIMIT 10)
```

Consultamos la tabla result

result.word	result.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

Inicio Page 1

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

```
the□44647
of□28020
□27298
to□23208
and□20444
in□13174
that□12265
I□10880
a□10431
is□7776
```

Apache Sqoop

- Nos conectamos al clúster vía ssh y buscamos la lib correspondiente

```
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -ls /user/oozie/share/lib/
Found 1 items
drwxr-xr-x   - oozie oozie          0 2020-05-01 20:01 /user/oozie/share/lib/lib_20200501200113
[hadoop@ip-172-31-86-245 ~]$
```

- Configuraciones

```
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -put /usr/share/java/mysql-connector-java.jar /user/oozie/share/lib/lib_20200501200113/sqoop/
put: `/user/oozie/share/lib/lib_20200501200113/sqoop/mysql-connector-java.jar': File exists
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20200501200113/sqoop/mysql-connector-java.jar
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20200501200113/sqoop/mysql-connector-java.jar
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -cp /user/oozie/share/lib/lib_20200501200113/hive/* /user/oozie/share/lib/lib_20200501200113/sqoop/
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/accessors-smart-1.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/apacheds-i18n-2.0.0-M15.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/apacheds-kerberos-codec-2.0.0-M15.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/api-asn1-api-1.0.0-M20.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/api-util-1.0.0-M20.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/asm-5.0.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-codec-1.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-collections-3.2.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-io-2.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-jexl-2.1.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-lang-2.4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/commons-logging-1.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/curator-client-2.5.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/curator-framework-2.5.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/derby-10.14.1.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/guava-11.0.2.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/hadoop-auth-2.8.5-amzn-4.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/httpclient-4.5.9.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/httpcore-4.4.11.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/jcip-annotations-1.0-1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/jetty-6.1.26-emr.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/json-smart-2.3.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/jsr305-3.0.0.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/log4j-1.2.17.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/nimbus-jose-jwt-4.41.1.jar': File exists
cp: `/user/oozie/share/lib/lib_20200501200113/sqoop/slf4j-api-1.6.6.jar': File exists
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20200501200113/sqoop/*
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20200501200113/sqoop/*
[hadoop@ip-172-31-86-245 ~]$
```

- Creamos la base de datos cursodb y la tabla employee

```
MySQL [(none)]> use cursodb;
Database changed
MySQL [cursodb]> CREATE TABLE `cursodb`.`employee` (  `emp_id` INT NOT NULL,  `n
ame` VARCHAR(45),  `salary` INT,  PRIMARY KEY (`emp_id`));
Query OK, 0 rows affected (0.04 sec)

MySQL [cursodb]> CREATE USER 'curso'@'%' IDENTIFIED BY 'curso';
Query OK, 0 rows affected (0.01 sec)

MySQL [cursodb]> GRANT ALL PRIVILEGES ON cursodb.* TO 'curso'@'%';
Query OK, 0 rows affected (0.01 sec)
```

- Llenamos la tabla

```
MySQL [cursodb]> insert into employee values (101, 'name1', 1800);
```

```
Query OK, 1 row affected (0.01 sec)

MySQL [cursodb]> insert into employee values (102, 'name2', 1500);
Query OK, 1 row affected (0.01 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);
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)
```

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

```
[hadoop@ip-172-31-86-245 ~]$ sqoop import --connect jdbc:mysql://database-2.cyuc3m1ondou.us-east-1.rds.amazonaws.com:3306/cursodb
 --username admin -P --table employee --target-dir /user/admin/mysqlOut -m 1
```

- Listamos los archivos:

```
[hadoop@ip-172-31-86-245 ~]$ hdfs dfs -ls /user/admin/mysqlOut
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2020-05-01 21:48 /user/admin/mysqlOut/_SUCCESS
-rw-r--r--   1 hadoop hadoop         75 2020-05-01 21:48 /user/admin/mysqlOut/part-m-00000
[hadoop@ip-172-31-86-245 ~]$
```

/ user / admin / mysqlOut / **part-m-00000**

```
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-172-31-86-245 ~]$ sqoop create-hive-table --connect jdbc:mysql://d
atabase-2.cyuc3m1ondou.us-east-1.rds.amazonaws.com:3306/cursodb --username ad
min -P --table employee --hive-database mydb --hive-table employee --mysql-de
limiters
```

mydb

Tables

Filter...

employee
- emp_id (int)
- name (string)
- salary (int)

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

```
[hadoop@ip-172-31-86-245 ~]$ sqoop import --connect jdbc:mysql://database-2.cyuc3m1ondou.us-east-1.rds.amazo
naws.com:3306/cursodb --username admin -P --table employee --hive-import --hive-database mydb --hive-table e
mployee --mysql-delimiters
```

Query History | Saved Queries | Query Builder | Results (5)

	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