

Consultas en Hive - Laboratorio 3-2

Objetivo

El objetivo de este laboratorio fue realizar consultas SQL en un dataset almacenado en HDFS utilizando Hive a través de la interfaz de Hue. Adicionalmente, se utilizó SparkSQL en JupyterHub para realizar consultas sobre el mismo dataset.

Instrucciones Paso a Paso

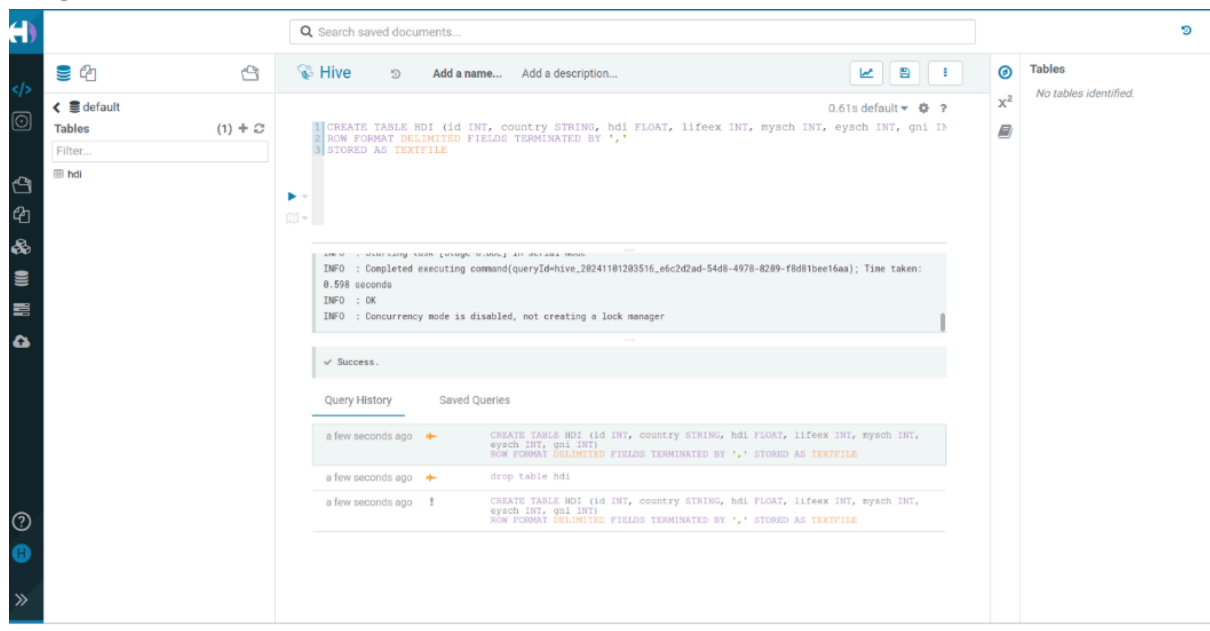
1. Acceso a Hue y Selección de Hive

Primero, accedimos a la interfaz de Hue en el cluster EMR de AWS. Una vez en Hue, seleccionamos la opción de Hive para ejecutar consultas SQL sobre el dataset previamente cargado en HDFS.

2. Consulta SQL Básica en Hive

En Hue, ejecutamos una serie de consultas SQL para explorar el contenido del dataset. Estas consultas incluyeron operaciones básicas como selección y filtrado de datos para obtener una visión general del contenido.

Imagen:



Add a name...
Add a description...

0.15s default
?

```

1 CREATE TABLE HDI (id INT, country STRING, hdi FLOAT, lifeex INT, mysch INT, eysch INT, gni INT)
2 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
3 STORED AS TEXTFILE
4
5 select * from hdi;
6
7 select country, gni from hdi where gni > 2000;

```

INFO : Completed executing command(queryId=hive_20241101203852_6bef9aaa-83a0-47cb-a0a1-7dbd3819617f); Time taken: 0.0 seconds

INFO : OK

INFO : Concurrency mode is disabled, not creating a lock manager

Query History
Saved Queries
Results (100+)

	country	gni
1	Norway	47557
2	Australia	34431
3	Netherlands	36402
4	United States	43017
5	New Zealand	23737
6	Canada	35166
7	Ireland	29322
8	Liechtenstein	83717

Add a name...
Add a description...

1.75s default
?

```

1 CREATE TABLE HDI (id INT, country STRING, hdi FLOAT, lifeex INT, mysch INT, eysch INT, gni INT)
2 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
3 STORED AS TEXTFILE
4
5 select * from hdi;
6
7 select country, gni from hdi where gni > 2000;
8
9 CREATE EXTERNAL TABLE EXPO (country STRING, expct FLOAT)
10 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
11 STORED AS TEXTFILE
12 LOCATION 's3://big-data-topicos/onu/'
13
14 SELECT h.country, gni, expct FROM HDI h JOIN EXPO e ON (h.country = e.country) WHERE gni > 2000;

```

INFO : Completed executing command(queryId=hive_20241101204042_73832b2e-7921-44f1-b587-6c1ccabbe73e); Time taken: 1.271 seconds

INFO : OK

INFO : Concurrency mode is disabled, not creating a lock manager

✓ Success.

Query History
Saved Queries

a few seconds ago	<pre>CREATE EXTERNAL TABLE EXPO (country STRING, expct FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 's3://big-data-topicos/onu/'</pre>
2 minutes ago	<pre>select country, gni from hdi where gni > 2000</pre>
2 minutes ago	<pre>select * from hdi</pre>
6 minutes ago	<pre>CREATE TABLE HDI (id INT, country STRING, hdi FLOAT, lifeex INT, mysch INT, eysch INT, gni INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE</pre>

0.59s default

```
1 CREATE EXTERNAL TABLE docs (line STRING)
2 STORED AS TEXTFILE
3 LOCATION 's3://big-data-topicos/bigdata/datasets/gutenberg-small/';
```

```
INFO : Completed executing command(queryId=hive_20241101204721_5e69ba51-f678-4657-9792-1674405d01ff); Time taken:
0.538 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Success.

Query History

Saved Queries

a few seconds ago	✓	CREATE EXTERNAL TABLE docs (line STRING) STORED AS TEXTFILE LOCATION 's3://big-data-topicos/bigdata/datasets/gutenberg-small/'
4 minutes ago	✓	select * from expo
7 minutes ago	⚙	SELECT h.country, gni, expct FROM HDI h JOIN EXPO e ON (h.country = e.country) WHERE gni > 2000
7 minutes ago	⚙	CREATE EXTERNAL TABLE EXPO (country STRING, expct FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 's3://big-data-topicos/onu/'
9 minutes ago	⚙	select country, gni from hdi where gni > 2000
9 minutes ago	⚙	select * from hdi
12 minutes ago	⚙	CREATE TABLE HDI (id INT, country STRING, hdi FLOAT, lifeex INT, mysch INT, ...)

18.68s default

```
1 CREATE EXTERNAL TABLE docs (line STRING)
2 STORED AS TEXTFILE
3 LOCATION 's3://big-data-topicos/bigdata/datasets/gutenberg-small/';
4
5 SELECT word, count(1) AS count FROM (SELECT explode(split(line, ' ')) AS word FROM docs) w
6 GROUP BY word
7 ORDER BY word DESC LIMIT 10;
```

```
INFO : Completed executing command(queryId=hive_20241101204802_38a82189-851d-4a10-b3a5-000111111111); Time taken:
18.306 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History

Saved Queries

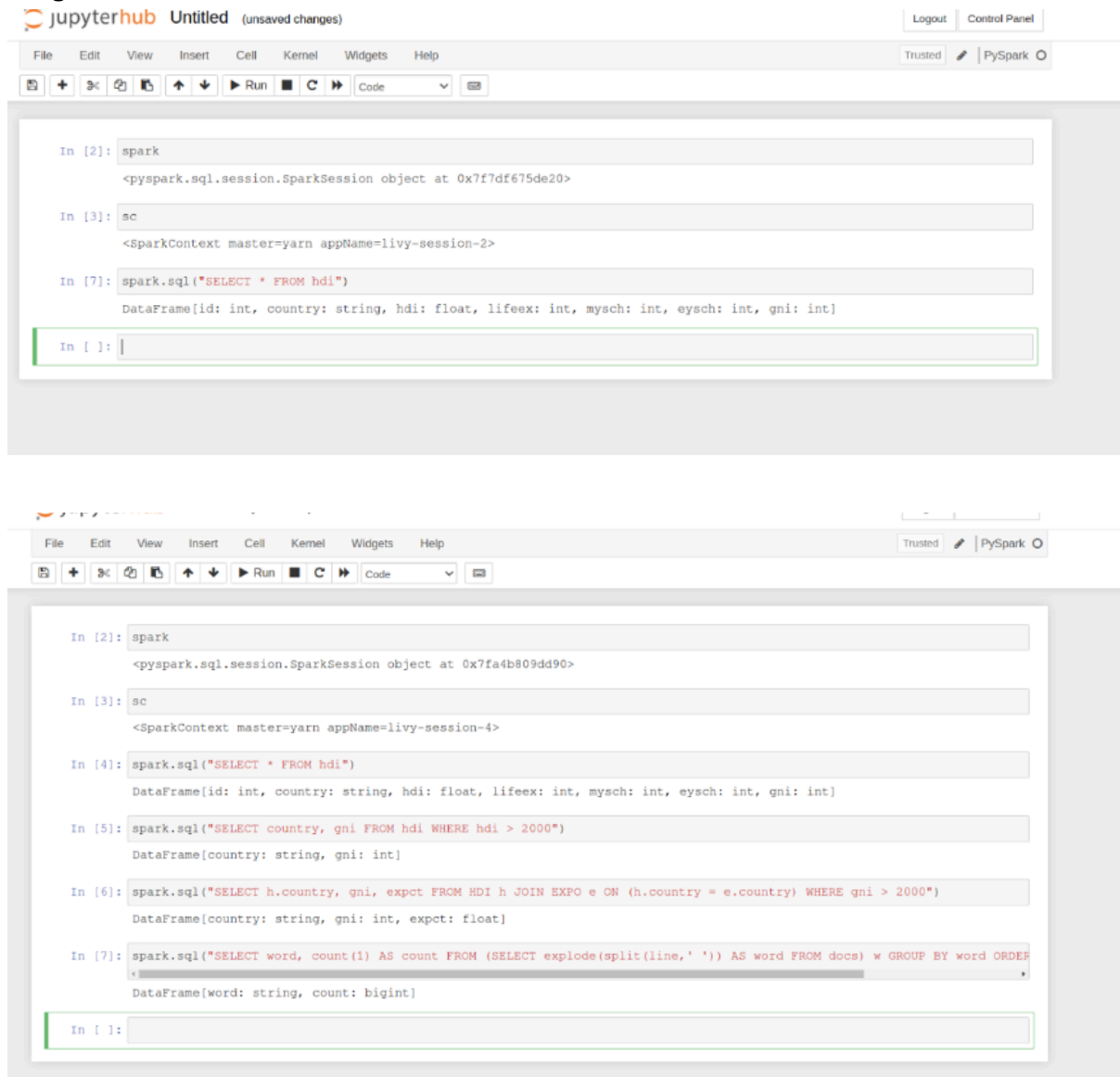
Results (10)

	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

4. Consultas en JupyterHub usando SparkSQL

Además de usar Hive en Hue, se realizaron consultas sobre el mismo dataset en JupyterHub usando SparkSQL. Esto permitió comparar el rendimiento y facilidad de uso de SparkSQL frente a Hive para la misma tarea.

Imagen:



The image displays two screenshots of a JupyterHub interface. The top screenshot shows the initial setup of a Spark session and a DataFrame. The bottom screenshot shows a series of SQL queries being executed on the DataFrame.

Top Screenshot:

```
In [2]: spark
<pyspark.sql.session.Session object at 0x7f7df675de20>

In [3]: sc
<SparkContext master=yarn appName=livy-session-2>

In [7]: spark.sql("SELECT * FROM hdi")
DataFrame[id: int, country: string, hdi: float, lifeex: int, mysch: int, eysch: int, gni: int]

In [ ]:
```

Bottom Screenshot:

```
In [2]: spark
<pyspark.sql.session.Session object at 0x7fa4b809dd90>

In [3]: sc
<SparkContext master=yarn appName=livy-session-4>

In [4]: spark.sql("SELECT * FROM hdi")
DataFrame[id: int, country: string, hdi: float, lifeex: int, mysch: int, eysch: int, gni: int]

In [5]: spark.sql("SELECT country, gni FROM hdi WHERE hdi > 2000")
DataFrame[country: string, gni: int]

In [6]: spark.sql("SELECT h.country, gni, expct FROM HDI h JOIN EXPO e ON (h.country = e.country) WHERE gni > 2000")
DataFrame[country: string, gni: int, expct: float]

In [7]: spark.sql("SELECT word, count(1) AS count FROM (SELECT explode(split(line, ' ')) AS word FROM docs) w GROUP BY word ORDER BY count DESC")
DataFrame[word: string, count: bigint]

In [ ]:
```

Conclusión

Este laboratorio permitió explorar el uso de Hive en la interfaz de Hue para consultas SQL sobre datos en HDFS, así como el uso de SparkSQL en JupyterHub para realizar las mismas consultas. La práctica brindó experiencia en la manipulación de grandes volúmenes de datos y en la comparación de herramientas de consulta en el entorno Hadoop.