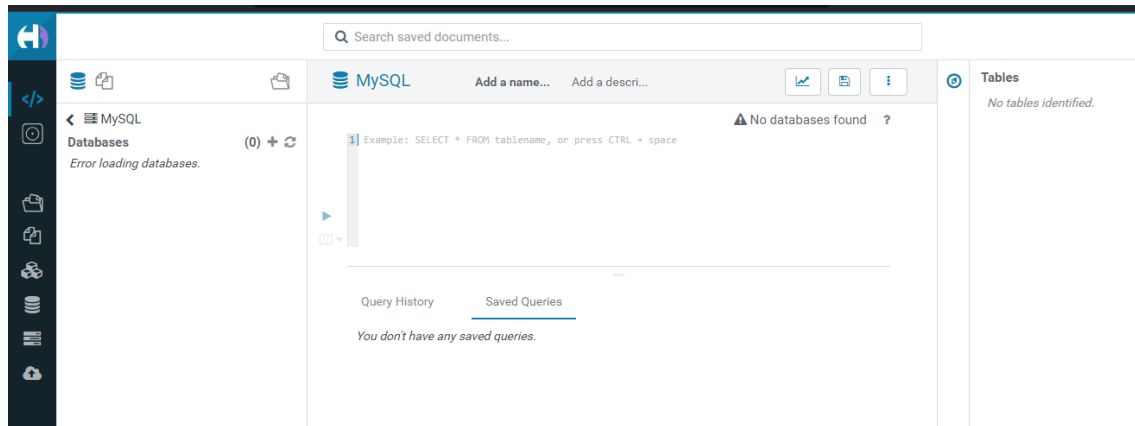
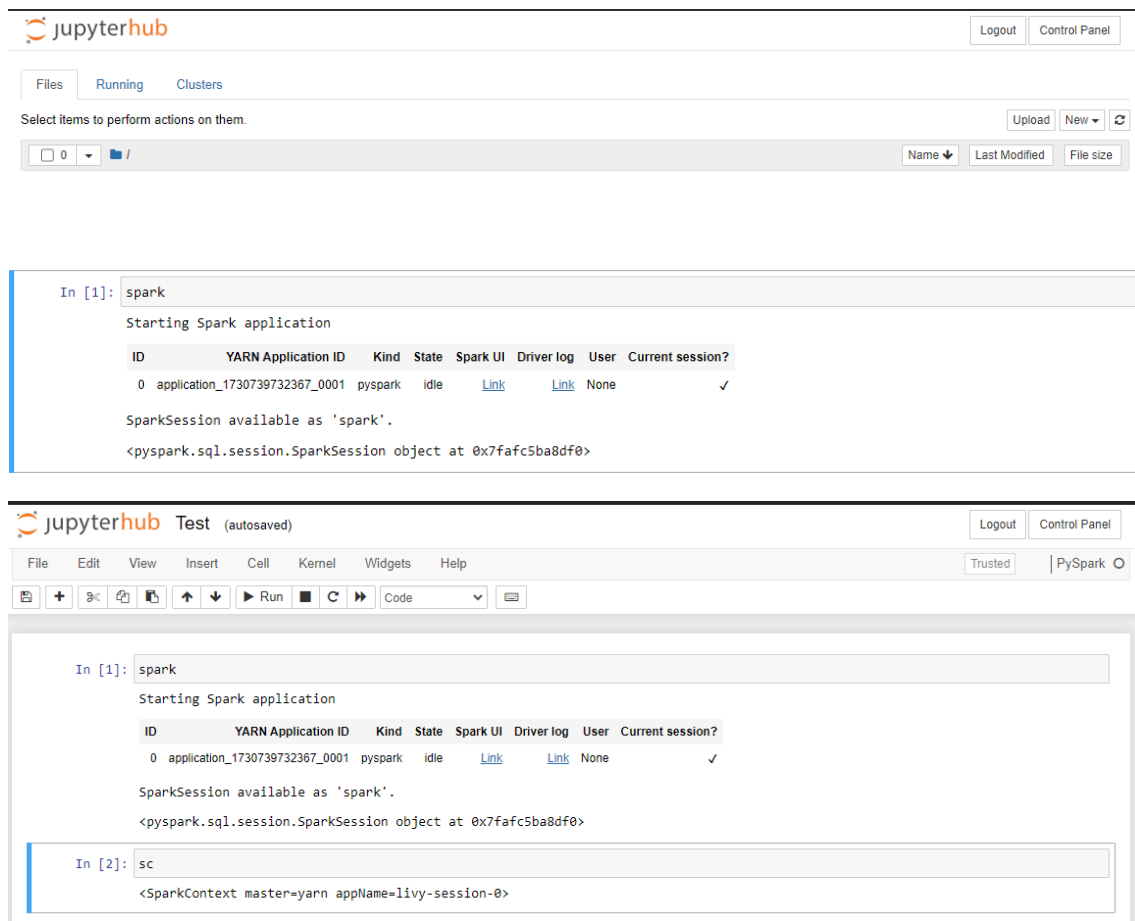


David Grisales Posada

Creación Hadoop:



Creación JupyterHub



Usando Zeppelin

%spark.pyspark

spark

FINISHED

<spark.sql.session.SparkSession object at 0x7f4d2b076fa0>

Took 43 sec. Last updated by anonymous at November 04 2024, 12:50:24 PM.

%spark.pyspark

sc

FINISHED

<SparkContext master=yarn appName=Zeppelin>

Took 0 sec. Last updated by anonymous at November 04 2024, 12:51:00 PM.

%sql

Show databases

FINISHED

namespace

default

Archivos S3 desde Hue.

File Browser

Search for file name

Actions

Copy Path

Open in Importer

Upload

New

us-east-1

s3a://davidnotebook/jupyter/jovyan

	Name	Size	User	Group	Permissions	Date
					drwxrwxrwx	
	.				drwxrwxrwx	
	.s3keep	0 bytes			-rw-rw-rw-	November 04, 2024 09:35 AM
	Test.ipynb	2.5 KB			-rw-rw-rw-	November 04, 2024 09:48 AM

Show 45 of 2 items

Page 1 of 1

Archivos montados en datasets de hadoop.

Home /user/hadoop/datasets







	Name	Size	User	Group	Permissions	Date
			hadoop	hdfsadmin	drwxrwxrwx	November 04, 2024 11:29 AM
	.		hadoop	hdfsadmin	drwxr-xr-x	November 04, 2024 11:30 AM
	gutenberg-small		hadoop	hdfsadmin	drwxr-xr-x	November 04, 2024 11:31 AM

Home /user/hadoop/datasets

	Name	Size	User	Group	Permissions	Date
			hadoop	hdfsadmin	drwxrwxrwx	November 04, 2024 11:29 AM
	.		hadoop	hdfsadmin	drwxr-xr-x	November 04, 2024 11:37 AM
	gutenberg-small		hadoop	hdfsadmin	drwxr-xr-x	November 04, 2024 11:31 AM
	onu		hadoop	hdfsadmin	drwxr-xr-x	November 04, 2024 11:38 AM

Show 45 of 2 items

Page 1 of 1

<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	 ↑		hadoop	hdfsadmingroup	drwxr-xr-x	November 04, 2024 11:51 AM
<input type="checkbox"/>	 .		hadoop	hdfsadmingroup	drwxr-xr-x	November 04, 2024 11:51 AM
<input type="checkbox"/>	 mrjob.zip	420.3 KB	hadoop	hdfsadmingroup	-rw-r--	November 04, 2024 11:51 AM
<input type="checkbox"/>	 setup-wrapper.sh	389 bytes	hadoop	hdfsadmingroup	-rw-r--	November 04, 2024 11:51 AM
<input type="checkbox"/>	 wordcount-mr.py	333 bytes	hadoop	hdfsadmingroup	-rw-r--	November 04, 2024 11:51 AM
Show <input type="text" value="45"/> of 3 items						
			Page	<input type="text" value="1"/>	of 1	

Hive SparkSQL

```
1 show databases;
```

```
INFO : Starting task [stage-0.000] in serial mode
INFO : Completed executing command(queryId=hive_20241104204518_46b23b54-8c7d-4a40-9cca-5bcf0ac67082); Time taken: 0.104 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History

Saved Queries

Results (2)

database_name



1 default

2 warehouse

0.219 GB

1 use warehouse;
2 show tables;

INFO : Starting task [stage-0.000] in serial mode

INFO : Completed executing command(queryId=hive_20241104204549_804a5a0e-8190-4447-857c-70af60aec3e0); Time taken: 0.187 seconds

INFO : OK

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

Query History

Saved Queries

Results (1)

tab_name

1 hdi

```
[hadoop@ip-172-31-88-238 ~]$ hdfs dfs -put ~/st0263-242/bigdata/datasets/* /user/hadoop/datasets/
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
```

```
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R:::::::::R
EE::::::::EEEEEEEE::::E M::::::::M M::::::::M R::::RRRRRR::::R
E:::E EEEEE M::::::::M M::::::::M RR:::R R:::R
E:::E M::::::::M M:::M M:::M R:::R R:::R
E:::EEEEEEEEEE M:::M M:::M M:::M R::RRRRRR::::R
E::::::::::::E M:::M M:::M M:::M R:::::::::RR
E:::EEEEEEEEEE M:::M M:::M M:::M R::RRRRRR::::R
E:::E M:::M M:::M M:::M R:::R R:::R
E:::E EEEEE M:::M MMM M:::M R:::R R:::R
EE::::::::EEEEEEEE::::E M:::M M:::M R:::R R:::R
E::::::::::::E M:::M M:::M RR:::R R:::R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$
[hadoop@ip-172-31-88-238 ~]$ sudo yum install git
```

```
[hadoop@ip-172-31-88-238 ~]$ hdfs dfs -ls /user/hadoop/datasets
Found 11 items
-rw-r--r-- 1 hadoop hdfsadmingroup 780058 2024-11-06 18:24 /user/hadoop/datasets/airlines.csv
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/all-news
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/covid19
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/flights
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/gutenberg
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/gutenberg-small
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/onu
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/otros
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/retail_logs
-rw-r--r-- 1 hadoop hdfsadmingroup 534 2024-11-06 18:24 /user/hadoop/datasets/sample_data.csv
drwxr-xr-x - hadoop hdfsadmingroup 0 2024-11-06 18:24 /user/hadoop/datasets/spark
[hadoop@ip-172-31-88-238 ~]$
```

Lab 3-2:

```
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
```

```
INFO : Starting task [Stage-0:DOL] in serial mode
INFO : Completed executing command(queryId=hive_20241106180313_2013480e-e938-4ce9-ac67-
aaaaeb7d5e4f8); Time taken: 0.563 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

0.32s default

```
1 CREATE EXTERNAL 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 LOCATION '/user/hadoop/datasets/onu/hdi/'
```

```
INFO : Starting task [Stage-0:DOL] in serial mode
INFO : Completed executing command(queryId=hive_20241106184045_31800d5f-6ff8-4c00-9d1d-1680670007bc); Time taken: 0.349 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

✓ Success.

```
1 show tables;
2
```

```
INFO : Starting task [Stage-0:DOL] in serial mode
INFO : Completed executing command(queryId=hive_20241106180735_bb0ff4c0-1dc1-45d2-a197-
0dc3cc57997b); Time taken: 0.213 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History

Saved Queries

Results (1)

tab_name

1	hdi
---	-----

```
1 desc hdi;
2
```

```
INFO : Executing command(queryId=hive_20241106180751_e0490543-e4cc-4311-a208-fa57ecef6c0e): desc hdi
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20241106180751_e0490543-e4cc-4311-a208-fa57ecef6c0e); Time taken: 0.117 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History Saved Queries Results (7)

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

```
1 select * from hdi;
2
```

```
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20241106180910_53ea1b75-4d77-4864-adea-ee7e714a466): select * from hdi
INFO : Completed executing command(queryId=hive_20241106180910_53ea1b75-4d77-4864-adea-ee7e714a466); Time taken: 0.001 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History Saved Queries Results (100+)

	hdi.id	hdi.country	hdi.hdi	hdi.lifeex	hdi.mysch	hdi.eysoh	hdi.gni
1	NULL	country	NULL	NULL	NULL	NULL	NULL
2	1	Norway	0.943	81	12	17	47557
3	2	Australia	0.929	81	12	18	34431
4	3	Netherlands	0.91	80	11	16	36402
5	4	United States	0.91	78	12	16	43017
6	5	New Zealand	0.908	80	12	18	23737

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

```
INFO : Concurrently mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20241106181003_ed8c4d38-e2b6-4946-bb7d-d1819fc3a4b4): select country, gni from hdi where gni>2000
INFO : Completed executing command(queryId=hive_20241106181003_ed8c4d38-e2b6-4946-bb7d-d1819fc3a4b4); Time taken: 0.0 seconds
INFO : OK
INFO : Concurrently 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

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

```
LOCATION 's3://davidnotebook/datasets/onu/export/'
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20241106181605_2c992803-fadc-473f-bc32-a6182)
INFO : OK
INFO : Concurrently mode is disabled, not creating a lock manager
```

✓ Success.

Tables (2) + ↺

Filter...

expo

hdi

1

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

▶

📄

INFO : Map 1: 1/1 Map 2: 1/1

INFO : Completed executing command(queryId=hive_20241106181706_acf5e93e-84cf-4169-ac43-5f194df982dd); Time taken: 7.723 seconds

INFO : OK

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

Query History

Saved Queries

Results (100+)

📄

📄

📄

	h.country	gni	expct
1	Albania	7803	29.77231
2	Algeria	7658	30.830406
3	Andorra	36095	NULL
4	Angola	4874	56.835884
5	Antigua and Barbuda	15521	44.08267
6	Argentina	14527	21.706469

Wordcount

1

CREATE EXTERNAL TABLE docs (line STRING)

2

STORED AS TEXTFILE

3

LOCATION 's3://davidnotebook/datasets/gutenberg-small/';

▶

📄

LOCATION 's3://davidnotebook/datasets/gutenberg-small/'

INFO : Starting task [Stage-0:DDL] in serial mode

INFO : Completed executing command(queryId=hive_20241106182602_513ebb89-14e6-47c4-88aa-3789fa3f3a02); Time taken: 0.402 seconds

INFO : OK

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

✔ Success.


```

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 : Map 1: 1/1      Reducer 2: 2/2 Reducer 3: 0/1/1
INFO : Completed executing command(queryId=hive_20241106182719_2d513ef1-f6ab-4a00-b3eb-c9dd3e
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

```

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 : Map 1: 1/1      Reducer 2: 2/2 Reducer 3: 0/1/1
INFO : Completed executing command(queryId=hive_20241106182906_a3aff338-7963-4945-812b-e90619a59bd7); Time taken: 12.100 sec
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager

```

Query History Saved Queries Results (10)

	word	count
1	the	44647
2	of	28020
3		27298
4	to	23208
5	and	20444
6	in	13174
7	at	10000

```

1 insert into storing
2 SELECT word, count(1) AS count FROM (SELECT explode(split(line, ' ')) AS word FROM docs) w
3 GROUP BY word
4 ORDER BY count;
5

```

INFO : Executing stats task
 INFO : Table default.storing stats: [numFiles=1, numRows=38683, totalSize=438215, rawDataSize=399532]
 INFO : Completed executing command(queryId=hive_20241106183342_dc3399b1-c222-44d7-8376-59b2af5c2a72); Time taken: 13.001 seconds
 INFO : OK
 INFO : Concurrency mode is disabled, not creating a lock manager

✓ Success.

```

6 select * from storing;

```

INFO : Concurrency mode is disabled, not creating a lock manager
 INFO : Executing command(queryId=hive_20241106183417_f0ac9619-5e0e-4fb2-87c6-86c142deaaab): select * from storing
 INFO : Completed executing command(queryId=hive_20241106183417_f0ac9619-5e0e-4fb2-87c6-86c142deaaab); Time taken: 0.001 seconds
 INFO : OK
 INFO : Concurrency mode is disabled, not creating a lock manager

Query History Saved Queries Results (100+)

	storing.word	storing.count
1	Æschines,	1
2	"But	1
3	"KANSAS	1
4	"Nothing	1
5	"One	1
6	"PETERSBURG,	1
7	"--	1

Lab 3-3:

2.1 Columnas:

Google Colab	<pre> 1 df.columns 2 ['fecha reporte web', 3 'ID de caso', 4 'Fecha de notificación', 5 'Código DIVIPOLA departamento', 6 'Nombre departamento', 7 'Código DIVIPOLA municipio', 8 'Nombre municipio', 9 'Edad', 10 'Unidad de medida de edad', 11 'Sexo', 12 'Tipo de contagio', 13 'Ubicación del caso', 14 'Estado', 15 'Código ISO del país', 16 'Nombre del país', 17 'Recuperado', 18 'Fecha de inicio de síntomas', 19 'Fecha de muerte', 20 'Fecha de diagnóstico', 21 'Fecha de recuperación', 22 'Tipo de recuperación', 23 'Pertenencia étnica', 24 'Nombre del grupo étnico'] </pre>
JupyterHub	<pre> In [5]: df.columns ['fecha reporte web', 'ID de caso', 'Fecha de notificación', 'Código DIVIPOLA departamento', 'Nombre departamento', 'Código DIVIPOLA municipio', 'Nombre municipio', 'Edad', 'Unidad de medida de edad', 'Sexo', 'Tipo de contagio', 'Ubicación del caso', 'Estado', 'Código ISO del país', 'Nombre del país', 'Recuperado', 'Fecha de inicio de síntomas', 'Fecha de muerte', 'Fecha de diagnóstico', 'Fecha de recuperación', 'Tipo de recuperación', 'Pertenencia étnica', 'Nombre del grupo étnico'] </pre>

2.2 Tipos de datos:

Google Colab	JupyterHub
<pre>1 df.printSchema() root -- fecha reporte web: string (nullable = true) -- ID de caso: integer (nullable = true) -- Fecha de notificación: string (nullable = true) -- Código DIVIPOLA departamento: integer (nullable = true) -- Nombre departamento: string (nullable = true) -- Código DIVIPOLA municipio: integer (nullable = true) -- Nombre municipio: string (nullable = true) -- Edad: integer (nullable = true) -- Unidad de medida de edad: integer (nullable = true) -- Sexo: string (nullable = true) -- Tipo de contagio: string (nullable = true) -- Ubicación del caso: string (nullable = true) -- Estado: string (nullable = true) -- Código ISO del país: integer (nullable = true) -- Nombre del país: string (nullable = true) -- Recuperado: string (nullable = true) -- Fecha de inicio de síntomas: string (nullable = true) -- Fecha de muerte: string (nullable = true) -- Fecha de diagnóstico: string (nullable = true) -- Fecha de recuperación: string (nullable = true) -- Tipo de recuperación: string (nullable = true) -- Pertenencia étnica: integer (nullable = true) -- Nombre del grupo étnico: string (nullable = true)</pre>	<pre>In [6]: df.printSchema() root -- fecha reporte web: string (nullable = true) -- ID de caso: integer (nullable = true) -- Fecha de notificación: string (nullable = true) -- Código DIVIPOLA departamento: integer (nullable = true) -- Nombre departamento: string (nullable = true) -- Código DIVIPOLA municipio: integer (nullable = true) -- Nombre municipio: string (nullable = true) -- Edad: integer (nullable = true) -- Unidad de medida de edad: integer (nullable = true) -- Sexo: string (nullable = true) -- Tipo de contagio: string (nullable = true) -- Ubicación del caso: string (nullable = true) -- Estado: string (nullable = true) -- Código ISO del país: integer (nullable = true) -- Nombre del país: string (nullable = true) -- Recuperado: string (nullable = true) -- Fecha de inicio de síntomas: string (nullable = true) -- Fecha de muerte: string (nullable = true) -- Fecha de diagnóstico: string (nullable = true) -- Fecha de recuperación: string (nullable = true) -- Tipo de recuperación: string (nullable = true) -- Pertenencia étnica: integer (nullable = true) -- Nombre del grupo étnico: string (nullable = true)</pre>

2.3 Seleccionando columnas

GoogleColab	JupyterHub
<pre>1 df.select('fecha reporte web', 'ID de caso', 'Sexo', 'Tipo de contagio').show(10, False) +-----+-----+-----+ fecha reporte web ID de caso Sexo Tipo de contagio +-----+-----+-----+ 6/3/2020 0:00:00 1 F Importado 9/3/2020 0:00:00 2 M Importado 9/3/2020 0:00:00 3 F Importado 11/3/2020 0:00:00 4 M Relacionado 11/3/2020 0:00:00 5 M Relacionado 11/3/2020 0:00:00 6 F Relacionado 11/3/2020 0:00:00 7 F Importado 11/3/2020 0:00:00 8 F Importado 11/3/2020 0:00:00 9 F Importado 12/3/2020 0:00:00 10 F Importado +-----+-----+-----+ only showing top 10 rows</pre>	<pre>In [7]: df.select('fecha reporte web', 'ID de caso', 'Sexo', 'Tipo de contagio').show(10, False) +-----+-----+-----+ fecha reporte web ID de caso Sexo Tipo de contagio +-----+-----+-----+ 6/3/2020 0:00:00 1 F Importado 9/3/2020 0:00:00 2 M Importado 9/3/2020 0:00:00 3 F Importado 11/3/2020 0:00:00 4 M Relacionado 11/3/2020 0:00:00 5 M Relacionado 11/3/2020 0:00:00 6 F Relacionado 11/3/2020 0:00:00 7 F Importado 11/3/2020 0:00:00 8 F Importado 11/3/2020 0:00:00 9 F Importado 12/3/2020 0:00:00 10 F Importado +-----+-----+-----+ only showing top 10 rows</pre>

2.4 Renombrando Columnas

GoogleColab	<pre>1 df.withColumnRenamed('fecha reporte web', 'fecha_reporte') 2 df.withColumnRenamed('ID de caso', 'ID', 'Nombre del país', 'pais', 'Nombre departamento', 'departamento', 'Nombre municipio', 'municipio', 'Ubicación del caso', 'ubicación')</pre>
-------------	--

JupyterHub

```
In [8]: df=df.withColumnRenamed('fecha reporte web', 'fecha_reporte')

In [9]: df=df.withColumnRenamed({'ID de caso': 'ID', 'Nombre del país': 'pais', 'Nombre departamento': 'departamento',
<
<

In [10]: df.printSchema()

root
|-- fecha_reporte: string (nullable = true)
|-- ID: integer (nullable = true)
|-- Fecha de notificación: string (nullable = true)
|-- Código DIVIPOLA departamento: integer (nullable = true)
|-- departamento: string (nullable = true)
|-- Código DIVIPOLA municipio: integer (nullable = true)
|-- municipio: string (nullable = true)
|-- Edad: integer (nullable = true)
|-- Unidad de medida de edad: integer (nullable = true)
|-- Sexo: string (nullable = true)
|-- Tipo de contagio: string (nullable = true)
|-- Ubicación del caso: string (nullable = true)
|-- Estado: string (nullable = true)
|-- Código ISO del país: integer (nullable = true)
|-- Nombre del país: string (nullable = true)
|-- Recuperado: string (nullable = true)
|-- Fecha de inicio de síntomas: string (nullable = true)
|-- Fecha de muerte: string (nullable = true)
|-- Fecha de diagnóstico: string (nullable = true)
|-- Fecha de recuperación: string (nullable = true)
|-- Tipo de recuperación: string (nullable = true)
|-- Pertenencia étnica: integer (nullable = true)
|-- Nombre del grupo étnico: string (nullable = true)
```

2.5 Añadiendo columnas

Google
Colab

```
1 df=df.withColumn('Edad_mas_10', df['Edad'] + 10)
2 df.show(5, False)
```

ubicación	estado	código ISO del país	recuperado	fecha de inicio de síntomas	fecha de muerte	fecha de diagnóstico	fecha de recuperación	tipo de recuperación	portencia étnica	nombre del grupo étnico	edad_max
Casa	leve	ITA	ITALIA	Recuperado [27/2/2020 0:00:00]	NMLL	16/3/2020 0:00:00	13/3/2020 0:00:00	PCR	16	NMLL	729
Casa	leve	ESP	ESPAÑA	Recuperado [9/3/2020 0:00:00]	NMLL	9/3/2020 0:00:00	23/3/2020 0:00:00	PCR	15	NMLL	144
Casa	leve	ESP	ESPAÑA	Recuperado [9/3/2020 0:00:00]	NMLL	9/3/2020 0:00:00	15/3/2020 0:00:00	PCR	16	NMLL	140
Casa	leve	NMLL	NMLL	Recuperado [9/3/2020 0:00:00]	NMLL	13/3/2020 0:00:00	26/3/2020 0:00:00	PCR	16	NMLL	165
Casa	leve	NMLL	NMLL	Recuperado [11/3/2020 0:00:00]	NMLL	11/3/2020 0:00:00	22/3/2020 0:00:00	PCR	16	NMLL	135

JupyterHub

```
In [11]: df=df.withColumn('Edad_mas_10', df['Edad'] + 10)
df.show(5, False)
```

[fecha_reporte]	[ID]	[Fecha de notificación]	[código DIVPOLA departamento]	[departamento]	[código DIVPOLA municipio]	[municipio]	[Edad]	[unidad de medida de edad]	[sexo]	[Tipo de contagio]	[ubicación del caso]	[estado]	[código ISO del país]	[Nombre del país]	[Recuperado]	[Fecha inicio de síntomas]	[Fecha de muerte]	[Fecha de diagnóstico]	[Fecha de recuperación]	[Tipo de recuperación]	[Pertenencia étnica]	[Nombre del grupo étnico]	[Edad_mas_18]
[6/3/2020 0:00:00]	[1]	[2/3/2020 0:00:00]	[11]	[casa]	[BOGOTA]	[11001]			[F]	[Importado]	[6/3/2020 0:00:00]	[13/3/2020 0:00:00]	[PCR]	[ITALIA]	[6]	[Recuperado]	[27/2/2020 0:00:00]	[NULL]					[NULL]

2.6 Eliminando columnas

GoogleColab

```
1 columns_to_remove=['Nombre del grupo étnico', 'Pertenencia étnica']
2 df=df.drop(*columns_to_remove)
```

JupyterHub

```
columns_to_remove=['Tipo de contagio', 'Unidad de medida de edad']
df=df.drop(*columns_to_remove)
df.columns
```

```
['fecha reporte web', 'ID', 'Fecha de notificación', 'Código DIVIPOLA departamento', 'departamento', 'Código DIVIPOLA municipio', 'municipio', 'Edad', 'Sexo', 'Ubicación del caso', 'Estado', 'Código ISO del país', 'Nombre del país', 'Recuperado', 'Fecha de inicio de síntomas', 'Fecha de muerte', 'Fecha de diagnóstico', 'Fecha de recuperación', 'Tipo de recuperación', 'Pertenencia étnica', 'Nombre del grupo étnico']
```

2.7 Filtrando datos

Google
Colab

```
1 filtered_df=df.filter(df.Sexo == 'F')
2 filtered_df.show(5, False)
```

fecha_reporte	id	fecha de notificación	código DIVIPOLA	departamento	departamento	código DIVIPOLA	municipio	municipio	edad	unidad de medida de edad	sexo
6/3/2020 0:00:00	1	12/3/2020 0:00:00	11	BOGOTÁ	11001	BOGOTÁ	119	1		F	M
9/3/2020 0:00:00	1	7/3/2020 0:00:00	5	ANTIOQUIA	5001	MEDELLÍN	508	1		F	M
11/3/2020 0:00:00	1	8/3/2020 0:00:00	5	ANTIOQUIA	5009	ITAGUÁ	512	1		F	M
11/3/2020 0:00:00	17	8/3/2020 0:00:00	13001	CARTAGENA	13001	CARTAGENA	85	1		F	M
11/3/2020 0:00:00	18	9/3/2020 0:00:00	11	BOGOTÁ	11001	BOGOTÁ	22	1		F	M

only showing top 5 rows

```
+-----+
only showing top 5 rows
```


Spark SQL	<pre>1 spark.sql("select municipio, count(*) as count from covid19 group by municipio order by count desc").show(10)</pre> <pre>+-----+-----+ municipio count +-----+-----+ BOGOTA 30016 BARRAMQUILLA 13065 CARTAGENA 8333 CALI 7747 SOLEDAD 6233 LETICIA 2194 MEDELLIN 2137 TUMACO 1501 BUENAVENTURA 1453 QUITBO 1367 +-----+-----+ only showing top 10 rows</pre>
-----------	--

3.3

Dataframes	<pre>1 df.dropna(subset=['Fecha de inicio de síntomas']).groupBy('Fecha de inicio de síntomas').count().orderBy('count', ascending=False).show(10, False)</pre> <pre>+-----+-----+ Fecha de inicio de síntomas count +-----+-----+ 10/6/2020 0:00:00 2731 16/6/2020 0:00:00 2558 18/6/2020 0:00:00 2479 12/6/2020 0:00:00 2452 1/6/2020 0:00:00 2429 8/6/2020 0:00:00 2398 17/6/2020 0:00:00 2344 5/6/2020 0:00:00 2266 9/6/2020 0:00:00 2224 19/6/2020 0:00:00 2162 +-----+-----+ only showing top 10 rows</pre>
SparkSQL	<pre>1 spark.sql("""select `Fecha de inicio de síntomas` as fecha, count(*) as count from covid19 where 2 `Fecha de inicio de síntomas` is not NULL group by fecha order by count desc limit 10""").show()</pre> <pre>+-----+-----+ fecha count +-----+-----+ 10/6/2020 0:00:00 2731 16/6/2020 0:00:00 2558 18/6/2020 0:00:00 2479 12/6/2020 0:00:00 2452 1/6/2020 0:00:00 2429 8/6/2020 0:00:00 2398 17/6/2020 0:00:00 2344 5/6/2020 0:00:00 2266 9/6/2020 0:00:00 2224 19/6/2020 0:00:00 2162 +-----+-----+</pre>

3.4

Dataframes	<pre>1 df.select('edad', 'age_group', 'ID').groupBy('edad', 'age_group').count().orderBy('count', ascending=False).show()</pre> <pre>+-----+-----+-----+ edad age_group count +-----+-----+-----+ 30 young adult 2735 29 young adult 2611 31 senior 2569 28 young adult 2540 27 young adult 2494 26 young adult 2436 33 senior 2371 32 senior 2362 25 young adult 2335 34 senior 2318 35 senior 2292 24 young adult 2214 36 senior 2175 37 senior 2132 38 senior 2098 40 senior 2058 23 young adult 2041 39 senior 1985 22 young adult 1879 41 senior 1783 +-----+-----+-----+ only showing top 20 rows</pre>
SparkSQL	<pre>1 spark.sql('select edad, age_group,count(*) as count from covid19 group by edad, age_group order by count desc').show()</pre> <pre>+-----+-----+-----+ edad age_group count +-----+-----+-----+ 30 young adult 2735 29 young adult 2611 31 senior 2569 28 young adult 2540 27 young adult 2494 26 young adult 2436 33 senior 2371 32 senior 2362 25 young adult 2335 34 senior 2318 35 senior 2292 24 young adult 2214 36 senior 2175 37 senior 2132 38 senior 2098 40 senior 2058 23 young adult 2041 39 senior 1985 22 young adult 1879 41 senior 1783 +-----+-----+-----+ only showing top 20 rows</pre>

3.5

Muertes por grupo de edad

Dataframes	<pre>[3]: 1 from pyspark.sql.functions import col 2 df.filter(col('Fecha de muerte').isNotNull()).groupBy('age_group').count().orderBy('count', ascending=False).show()</pre> <pre>+-----+-----+ age_group count +-----+-----+ senior 5435 young adult 151 kid 47 +-----+-----+</pre>
SparkSQL	<pre>1 spark.sql("""select age_group, count(*) as count from covid19 where 2 Fecha de muerte' is not null group by age_group order by count desc""").show()</pre> <pre>+-----+-----+ age_group count +-----+-----+ senior 5435 young adult 147 kid 51 +-----+-----+</pre>

Punto 3 JupyterHub

3.1

Spark Dataframes	<pre>In [22]: df.groupBy('departamento').count().orderBy('count', ascending=False).show(10, False)</pre> <pre>+-----+-----+ departamento count +-----+-----+ BOGOTA 30016 BARRANQUILLA 13065 ATLANTICO 10994 VALLE 10404 CARTAGENA 8333 ANTIOQUIA 4554 NARIÑO 3520 CUNDINAMARCA 2827 AMAZONAS 2317 CHOCO 1636 +-----+-----+ only showing top 10 rows</pre>
SparkSQL	<pre>In [79]: spark.sql("select departamento, count(*) as count from covid19 group by departamento order by count desc").show(10)</pre> <pre>+-----+-----+ departamento count +-----+-----+ BOGOTA 30016 BARRANQUILLA 13065 ATLANTICO 10994 VALLE 10404 CARTAGENA 8333 ANTIOQUIA 4554 NARIÑO 3520 CUNDINAMARCA 2827 AMAZONAS 2317 CHOCO 1636 +-----+-----+ only showing top 10 rows</pre>

3.2

Spark Dataframes	<pre>In [24]: df.groupBy('municipio').count().orderBy('count', ascending=False).show(10, False)</pre> <pre>+-----+-----+ municipio count +-----+-----+ BOGOTA 30016 BARRANQUILLA 13065 CARTAGENA 8333 CALI 7747 SOLEDAD 6233 LETICIA 2194 MEDELLIN 2137 TUMACO 1501 BUENAVENTURA 1453 QUIBDO 1367 +-----+-----+ only showing top 10 rows</pre>
Spark SQL	<pre>In [80]: spark.sql("select municipio, count(*) as count from covid19 group by municipio order by count desc").show(10)</pre> <pre>+-----+-----+ municipio count +-----+-----+ BOGOTA 30016 BARRANQUILLA 13065 CARTAGENA 8333 CALI 7747 SOLEDAD 6233 LETICIA 2194 MEDELLIN 2137 TUMACO 1501 BUENAVENTURA 1453 QUIBDO 1367 +-----+-----+ only showing top 10 rows</pre>

3.3

Dataframes	<pre>In [71]: df.dropna(subset=['fecha_reporte']).groupBy('fecha_reporte').count().orderBy('count', ascending=False).show(10, false)</pre> <pre>-----+-----+ fecha_reporte count -----+-----+ 27/6/2020 0:00:00 4149 26/6/2020 0:00:00 3843 24/6/2020 0:00:00 3541 25/6/2020 0:00:00 3486 29/6/2020 0:00:00 3274 28/6/2020 0:00:00 3178 18/6/2020 0:00:00 3171 19/6/2020 0:00:00 3059 21/6/2020 0:00:00 3019 30/6/2020 0:00:00 2803 -----+-----+ only showing top 10 rows</pre>
SparkSQL	<pre>In [53]: spark.sql("""select fecha_reporte as fecha, count(*) as count from covid19 where fecha_reporte is not NULL group by fecha order by count desc limit 10""").show()</pre> <pre>-----+-----+ fecha count -----+-----+ 27/6/2020 0:00:00 4149 26/6/2020 0:00:00 3843 24/6/2020 0:00:00 3541 25/6/2020 0:00:00 3486 29/6/2020 0:00:00 3274 28/6/2020 0:00:00 3178 18/6/2020 0:00:00 3171 19/6/2020 0:00:00 3059 21/6/2020 0:00:00 3019 30/6/2020 0:00:00 2803 -----+-----+</pre>

3.4

Dataframes	<pre>In [26]: df.select('edad', 'age_group', 'ID').groupBy('edad', 'age_group').count().orderBy('count', ascending=False).show()</pre> <pre>-----+-----+-----+ edad age_group count -----+-----+-----+ 30 young adult 2735 29 young adult 2611 31 senior 2569 28 young adult 2540 27 young adult 2494 26 young adult 2436 33 senior 2371 32 senior 2362 25 young adult 2335 34 senior 2310 35 senior 2292 24 young adult 2214 36 senior 2175 37 senior 2132 38 senior 2098 40 senior 2050 23 young adult 2041 39 senior 1985 22 young adult 1879 41 senior 1783 -----+-----+-----+ only showing top 20 rows</pre>
SparkSQL	<pre>In [36]: spark.sql('select edad, age_group, count(*) as count from covid19 group by edad, age_group order by count desc').show()</pre> <pre>-----+-----+-----+ edad age_group count -----+-----+-----+ 30 young adult 2735 29 young adult 2611 31 senior 2569 28 young adult 2540 27 young adult 2494 26 young adult 2436 33 senior 2371 32 senior 2362 25 young adult 2335 34 senior 2310 35 senior 2292 24 young adult 2214 36 senior 2175 37 senior 2132 38 senior 2098 40 senior 2050 23 young adult 2041 39 senior 1985 22 young adult 1879 41 senior 1783 -----+-----+-----+ only showing top 20 rows</pre>

3.5

Muertes por grupo de edad

Dataframes	<pre>In [77]: from pyspark.sql.functions import col df.filter(col('fecha de muerte').isNotNull()).groupBy('age_group').count().orderBy('count', ascending=False).show()</pre> <pre>-----+-----+ age_group count -----+-----+ senior 5435 young adult 151 kid 47 -----+-----+</pre>
SparkSQL	<pre>In [36]: spark.sql("""select age_group, count(*) as count from covid19 where 'fecha de muerte' is not null group by age_group order by count desc""").show()</pre> <pre>-----+-----+ age_group count -----+-----+ senior 5435 young adult 151 kid 47 -----+-----+</pre>

URI S3: s3://davidnotebook/jupyter/