

UNIVERSIDAD EAFIT
ALMACENAMIENTO Y RECUPERACION DE LA INFORMACIÓN

PROYECTO 1
INGENIERIA DE DATOS

ESTUDIANTES:
MATEO HOLGUIN CARVALHO
DIEGO F. GUASCO LOAIZA
FABIÁN D. SÁNCHEZ MARTINEZ

ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA
MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

EDWIN N. MONTOYA MUNERA

MARZO 15, 2024

PROYECTO 1 – Ingeniería de Datos

1. Selección de los diferentes datasets.

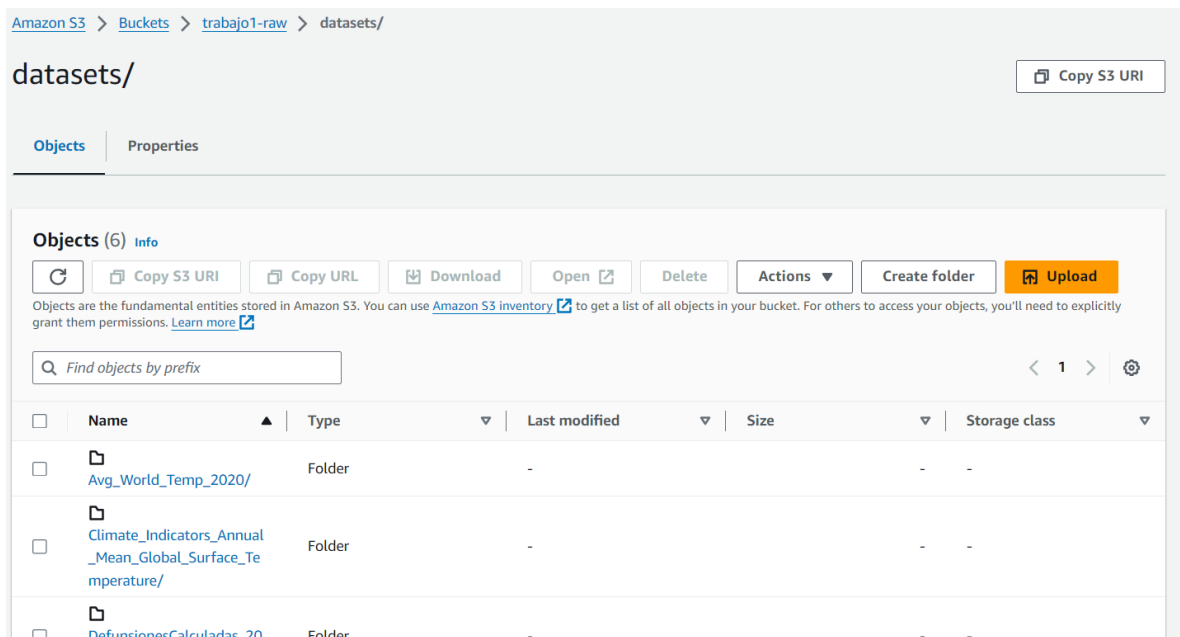
Para la realización del taller nos basamos en extracciones de datasets principalmente de Kaggle, del Portal de Datos Abiertos del Área Metropolitana del Valle de Aburrá y Datos del Cambio Climático del Fondo Monetario Internacional (Climate Change Dashboard – IMF).

Datasets	URL
Avg_World_Temp_2020	https://www.kaggle.com/datasets/efradgamer/world-average-temperature?resource=download
DefuncionesCalculadas_2008-2017	https://datosabiertos.metropol.gov.co/dataset/7281fa8e-c666-4fed-a8bd-a4af7d0ea749
Climate_Indicators_Annual_Mean_Global_Surface_Temperature	https://climatedata.imf.org/datasets/4063314923d74187be9596f10d034914_0/explore
global air pollution dataset	https://www.kaggle.com/datasets/hasibalmuzdadid/global-air-pollution-dataset
Water Quality	https://www.kaggle.com/datasets/patricklford/water-and-air-quality?select=Cities1.csv
Temperatura	https://datosabiertos.metropol.gov.co/sites/default/files/uploaded_resources/Datos_SIATA_Vaisala_temperatura.json

2. Ingesta de datos como archivos / datasets desde Almacenamiento de datos en AWS S3

En este paso, se toman los archivos csv descargados previamente y se cargan dentro del bucket creado en S3, en este caso de la zona RAW.

URI S3 Raw: s3://trabajo1-raw/datasets/



3. Diseño de un datalake como almacenamiento de los datos diferentes orígenes, tipos y estructuras.

En este paso se crean las 3 diferentes zonas: Raw, Trusted y Refined como se muestra a continuación y se establece la estructura de directorios:

- URI Raw: s3://trabajo1-raw/datasets/
- URI Trusted: s3://trabajo1-trusted/datatrustrusted/
- URI Refined: s3://trabajo1-refined/datarefine/

General purpose buckets

Directory buckets

General purpose buckets (8) Info

Refresh

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

< 1 > ⚙

	Name	AWS Region	Access	Creation date
<input type="radio"/>	trabajo1-refined	US East (N. Virginia) us-east-1	Objects can be public	March 9, 2024, 13:33:39 (UTC-05:00)
<input type="radio"/>	trabajo1-trusted	US East (N. Virginia) us-east-1	Objects can be public	March 9, 2024, 13:32:58 (UTC-05:00)
<input type="radio"/>	trabajo1-raw	US East (N. Virginia) us-east-1	Objects can be public	March 9, 2024, 13:31:20 (UTC-05:00)

4. Catalogación:

En este paso se transforman los datos utilizando AWS Glue, inicialmente los datos se veían así:

AWS Glue > Tables > global_air_pollution_dataset > Edit schema

Edit schema: global_air_pollution_dataset

Schema (12)

Delete

Edit

Add

View and manage the table schema.

Filter schemas

< 1 > ⚙

<input type="checkbox"/>	#	Column name	Data type	Partition key	Comment
<input type="checkbox"/>	1	country	string	-	-
<input type="checkbox"/>	2	city	string	-	-
<input type="checkbox"/>	3	aqi value	bigint	-	-
<input type="checkbox"/>	4	aqi category	string	-	-
<input type="checkbox"/>	5	co aqi value	bigint	-	-
<input type="checkbox"/>	6	co aqi category	string	-	-
<input type="checkbox"/>	7	ozone aqi value	bigint	-	-
<input type="checkbox"/>	8	ozone aqi categ...	string	-	-
<input type="checkbox"/>	9	no2 aqi value	bigint	-	-
<input type="checkbox"/>	10	no2 aqi category	string	-	-
<input type="checkbox"/>	11	pm2.5 aqi value	bigint	-	-
<input type="checkbox"/>	12	pm2.5 aqi categ...	string	-	-

Posterior a la transformación, los datos quedan así, de 12 columnas quedaron 10 con un tipo de data modificado también:

Schema

Partitions

Indexes

Column statistics - new

Schema (10)

Edit schema as JSON

Edit schema

View and manage the table schema.

Q Filter schemas

< 1 > ⚙

#	Column name	Data type	Partition key	Comment
1	country	string	-	-
2	city	string	-	-
3	aqi value	int	-	-
4	aqi category	string	-	-
5	co aqi value	int	-	-
6	co aqi category	string	-	-
7	ozone aqi value	bigint	-	-
8	ozone aqi category	string	-	-
9	pm2.5 aqi value	int	-	-
10	pm2.5 aqi category	string	-	-

Esta es una segunda transformación con AWS Glue, los datos inicialmente se cargaron así:

#	Column name	Data type	Partition key	Comment						
1	id	string	-	-						
2	country	string	-	-						
3	city	string	-	-						
4	jan	double	-	-						
5	feb	double	-	-						
6	mar	double	-	-						
7	apr	double	-	-						
8	may	double	-	-						
9	jun	double	-	-						
10	jul	double	-	-						
11	aug	double	-	-						
12	sep	double	-	-						
13	oct	double	-	-						
14	nov	double	-	-						
15	dec	double	-	-						
16	avg_year	double	-	-						
17	continent	string	-	-						

Y así quedaron posterior a la transformación, se redujeron a 16 columnas:

Schema (16)							Edit schema as JSON		Edit schema	
View and manage the table schema.							Filter schemas			< 1 > ⚙
#	Column name	Data type	Partition key	Comment						
1	country	string	-	-						
2	city	string	-	-						
3	jan	double	-	-						
4	feb	double	-	-						
5	mar	double	-	-						
6	apr	double	-	-						
7	may	double	-	-						
8	jun	double	-	-						
9	jul	double	-	-						
10	aug	double	-	-						
11	sep	double	-	-						
12	oct	double	-	-						
13	nov	double	-	-						
14	dec	double	-	-						

Views (0)

< 1 >

Run again

Explain

Cancel

Clear

Create

Reuse query results

up to 60 minutes ago

Query results

Query stats

Completed

Time in queue: 101 ms

Run time: 566 ms

Data scanned: 1.56 MB

Results (4)

Copy

Download results

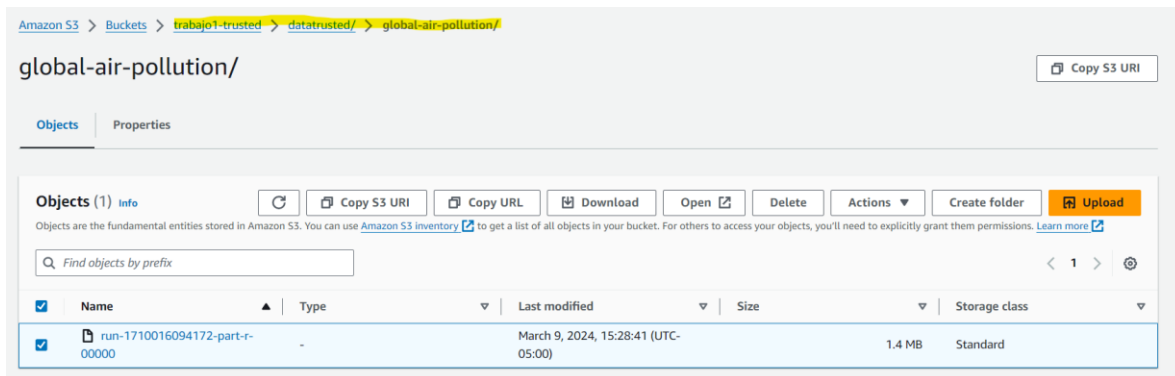
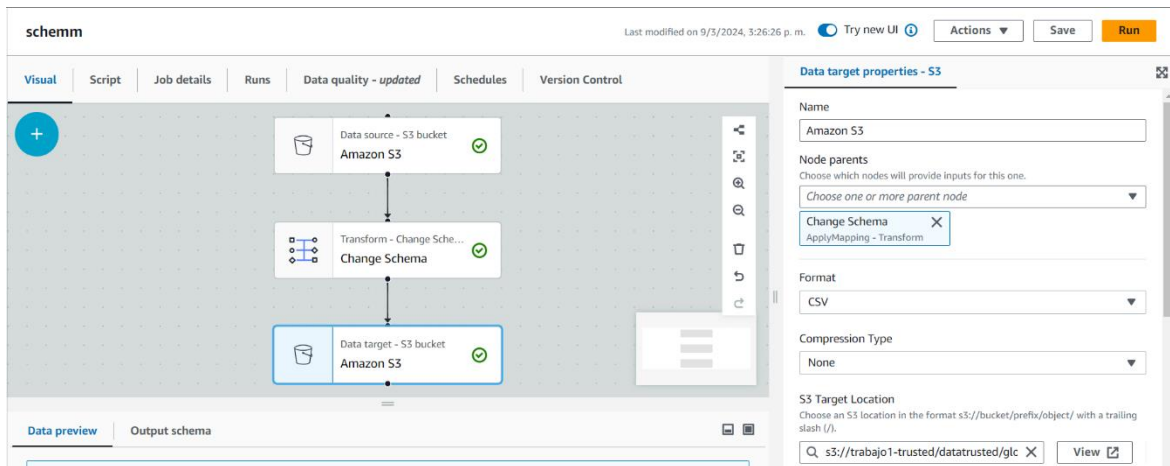
Search rows

< 1 >

#	country	city	aqi value	aqi category	co aqi value	co aqi category	ozone aqi value
1	Republic of Korea	Seoul	421	Hazardous	27	Good	0
2	United States of America	Durango	500	Hazardous	133	Unhealthy for Sensitive Groups	0
3	China	Chengdu	386	Hazardous	28	Good	0
4	China	Xian	307	Hazardous	19	Good	0

5. Proceso transformación con ETL job con AWS Glue

En este paso se usa Glue para la catalogación, indicando como zona de salida el bucket trusted



URI S3: s3://aws-glue-assets-891377086651-us-east-1/

Script (catalogación Glue – ETL Jobs)

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
```

```
job = Job(glueContext)
job.init(args["JOB_NAME"], args)
```

```
# Script generated for node Amazon S3
```

```
AmazonS3_node1710014273779 = glueContext.create_dynamic_frame.from_options(
    format_options={
        "quoteChar": "'",
        "withHeader": True,
        "separator": ",",
        "optimizePerformance": False,
    },
    connection_type="s3",
    format="csv",
    connection_options={
        "paths": [
            "s3://trabajo1-raw/datasets/global air pollution dataset/global air pollution dataset.csv"
        ],
        "recurse": True,
    },
    transformation_ctx="AmazonS3_node1710014273779",
)
```

```
# Script generated for node Change Schema
```

```
ChangeSchema_node1710014282069 = ApplyMapping.apply(
    frame=AmazonS3_node1710014273779,
    mappings=[
        ("Country", "string", "Country", "string"),
        ("City", "string", "City", "string"),
        ("AQI Value", "string", "AQI Value", "int"),
        ("AQI Category", "string", "AQI Category", "string"),
        ("CO AQI Value", "string", "CO AQI Value", "int"),
        ("CO AQI Category", "string", "CO AQI Category", "string"),
        ("NO2 AQI Value", "string", "NO2 AQI Value", "int"),
        ("NO2 AQI Category", "string", "NO2 AQI Category", "string"),
        ("`PM2.5 AQI Value`", "string", "`PM2.5 AQI Value`", "int"),
        ("`PM2.5 AQI Category`", "string", "`PM2.5 AQI Category`", "string"),
    ],
    transformation_ctx="ChangeSchema_node1710014282069",
)
```

```
# Script generated for node Amazon S3
```

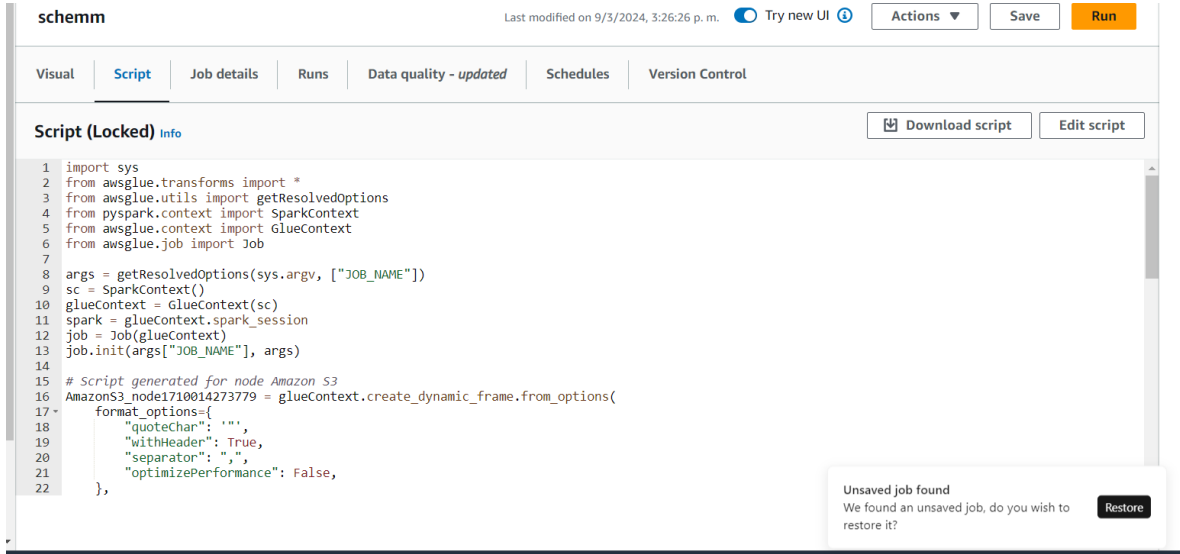
```
AmazonS3_node1710014287417 = glueContext.write_dynamic_frame.from_options(
    frame=ChangeSchema_node1710014282069,
    connection_type="s3",
    format="csv",
    connection_options={
        "path": "s3://trabajo1-trusted/datatrustrusted/global-air-pollution/",
        "partitionKeys": [],
    },
)
```

```

    },
    transformation_ctx="AmazonS3_node1710014287417",
)

job.commit()

```



6. Consultas

A continuación, se muestran unas consultas realizadas posterior a la catalogación y transformación de los datos en Athena (más adelante se muestran las consultas realizadas en Hive):

Para el dataset “global_air_pollution_dataset”:

```

SELECT * FROM "tpollution"."global_air_pollution_dataset"
WHERE "aqi category" = 'Hazardous'
AND "pm2.5 aqi value" >40

```



Segunda consulta

Athena now supports typeahead code suggestions to speed up SQL query development

Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.

Edit preferences

Data

- Data source
AwsDataCatalog
- Database
t-avg-wt2
- Tables and views
 Filter tables and views
- Tables (1)
avg_world_temp_2020
- Views (0)

Query 7 ✕ | Query 8 ✕ | **Query 9** ✕ | Query 10 ✕

```

1 SELECT * FROM "t-avg-wt2"."avg_world_temp_2020"
2

```

SQL Ln 1, Col 1

Run again
 Explain
 Cancel
 Clear
 Create

☒ Reuse query results
up to 60 minutes ago

Query results | Query stats

Completed
 Time in queue: 106 ms Run time: 545 ms Data scanned: 40.96 KB

Completed

Time in queue: 106 ms Run time: 545 ms Data scanned: 40.96 KB

Results (425)

Copy

Download results

< 1 ... > ⌕

# ▾	id ▾	country ▾	city ▾	jan ▾	feb ▾	mar ▾	apr ▾	may ▾	jun ▾
1	0	Algeria	Algiers	11.2	11.9	12.8	14.7	17.7	21.3
2	1	Algeria	Tamanrasset	12.8	15.0	18.1	22.2	26.1	28.9
3	2	Algeria	Reggane	16.0	18.2	23.1	27.9	32.2	36.4
4	3	Angola	Luanda	26.7	28.5	28.6	28.2	27.0	23.9
5	4	Benin	Cotonou	27.3	28.5	28.9	28.6	27.8	26.5
6	5	Benin	Parakou	26.5	28.7	29.6	29.0	27.5	26.1
7	6	Benin	Kandi		24.9	27.8	30.9	32.1	30.4
8	7	Botswana	Maun		25.4	25.1	24.2	22.6	18.7
9	8	Botswana	Gaborone	26.0	25.0	24.0	21.0	17.0	13.0
10	9	Botswana	Ghanzi	25.9	25.3	24.1	21.2	17.6	14.3

Tercera consulta

Query 11

```
1 SELECT * FROM "t-avg-wt2"."avg_world_temp_2020"
2 where country = 'Colombia' and avg_year > 20.0
```

Results (2)

#	id	country	city	jan	feb	mar	apr	may	jun	jul	aug
1	410	Colombia	Barranquilla	26.5	26.7	27.0	27.5	28.1	28.1	27.9	27.9
2	411	Colombia	Medellin	22.4	22.7	22.7	22.4	22.6	22.9	23.1	23.1

7. Tablas almacenadas en Redshift desde S3 y su respectiva consulta:

Archivo cargado en Redshift

Redshift query editor v2

```
1 load-data-avgairpollu-81d7
1 1 air pollution dataset.csv' IAM_ROLE 'arn:aws:iam::851725270465:role/LabRole' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1'
```

Summary

Info:

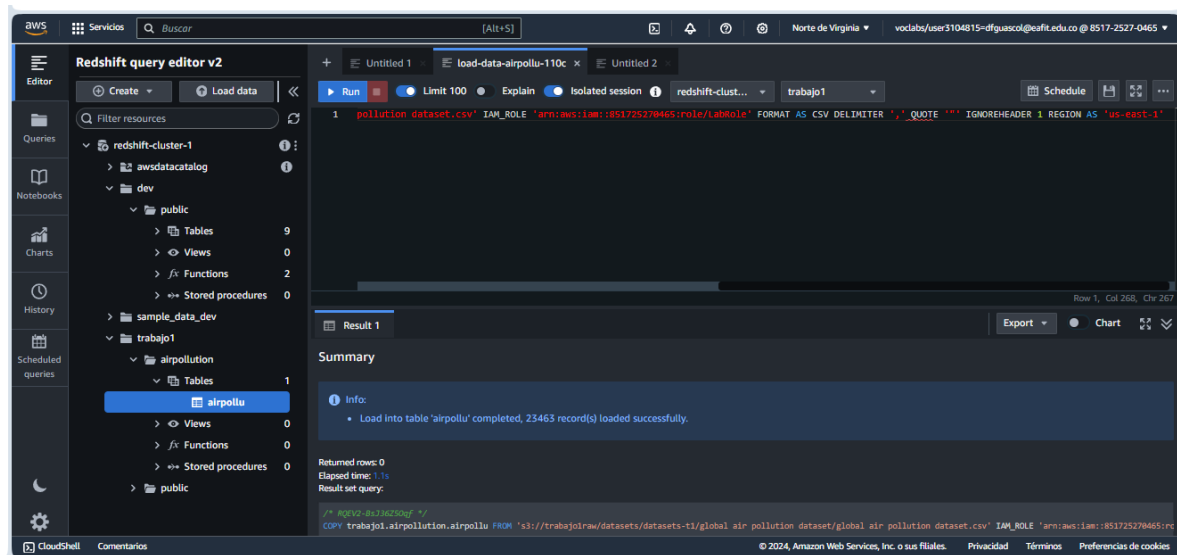
- Load into table 'avgairpollu' completed, 23463 record(s) loaded successfully.

Returned rows: 0
Elapsed time: 578ms
Result set query:

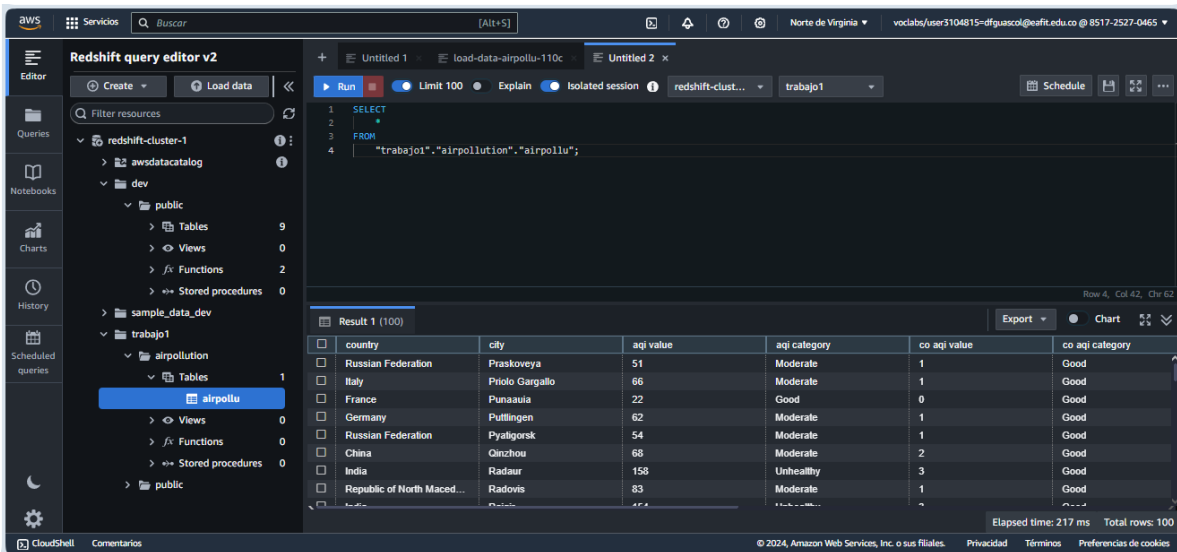
```
/* Redshift console */
COPY trabajo1.airpollution.avgairpollu FROM 's3://trabajo1raw/datasets/datasets-t1/global air pollution dataset/global air pollution dataset.csv' IAM_ROLE 'arn:aws:iam::851725270465:role/LabRole'
```

Accediendo a los datos en S3

```
COPY trabajo1.airpollution.airpollu FROM 's3://trabajo1raw/datasets/datasets-t1/global air
pollution dataset/global air pollution dataset.csv' IAM_ROLE
'arn:aws:iam::851725270465:role/LabRole' FORMAT AS CSV DELIMITER ',' QUOTE ''
IGNOREHEADER 1 REGION AS 'us-east-1'
```



Consultas



```
SELECT
    country, count(city) as city_cnt
FROM
    "trabajo1"."airpollution"."airpollu"
WHERE
    "aqi category" = 'Hazardous'
GROUP BY
```

country
ORDER BY
city_cnt DESC

Redshift query editor v2

Filter resources

redshift-cluster-1

awsdatacatalog

dev

public

Tables 9

Views 0

Functions 2

Stored procedures 0

sample_data_dev

trabajo1

airpollution

airpollu 1

Views 0

Functions 0

Stored procedures 0

public

```
1 SELECT
2   country, count(city) as city_cnt
3 FROM
4   "trabajo1"."airpollution"."airpollu"
5 WHERE
6   "aqi category" = 'Hazardous'
7 GROUP BY
8   country
9 ORDER BY
10  city_cnt DESC
```

Result 1 (11)

country	city_cnt
India	158
Pakistan	13
Mexico	6
China	3
Democratic Republic of th...	3
South Africa	3
Chile	1
Uzbekistan	1
United States of America	1

Elapsed time: 221 ms Total rows: 11

8. Ecosistema Hadoop/Spark basado en AWS EMR

Desplegar clústeres EMR

Amazon EMR > EMR on EC2: Clusters

Clusters (11) Info

Filter clusters by status

Find clusters

Filter clusters by creation date-time

Cluster ID	Cluster name	Status	Creation time (UTC-05:00)	Elapsed time
j-3GV9Y8Y244D8T	Cluster MHOLGUINC2	Terminated User request	March 14, 2024, 19:52	1 hour, 51 mi
j-J7ZEBCE4IMJ6	Cluster MHOLGUINC2	Terminated with errors Instance failure	March 13, 2024, 20:15	3 hours, 55 m
j-287SFJFCQLG3T	Cluster MHOLGUINC2	Terminated User request	March 12, 2024, 22:20	44 minutes, 1
j-5J4AMRD3HH18	Cluster MHOLGUINC2	Terminated Auto-terminate	March 12, 2024, 21:00	1 hour, 17 mi

Almacenamiento de datos temporales en HDFS

File Browser

Search for file name

Actions

Copy Path

Open in Importer

Upload

New

Home /user/hadoop/datasets/avg_wt

Name	Size	Usuario	Group	Permisos	Date
.		hadoop	hdfsadmin	drwxr-xr-x	March 14, 2024 08:31 PM
Avg_World_Temp_2020.csv	41.0 KB	hadoop	hdfsadmin	-rw-r--r--	March 14, 2024 08:31 PM

Show 45 of 1 items

Page 1 of 1

Creación de tabla externa en Hive desde S3 (catalogación)

USE trabajo1_fdm

```
CREATE EXTERNAL TABLE HDI (country STRING, city STRING, aqi_value INT, aqi_category STRING,  
co_aqi_value INT, co_aqi_category STRING, oz_aqi_value INT, oz_aqi_category STRING,  
no2_aqi_value INT, no2_aqi_category STRING, pm25_aqi_value INT, pm25_aqi_category STRING)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
STORED AS TEXTFILE  
LOCATION 's3://trabajo1-raw/datasets/global air pollution dataset/'
```

The screenshot displays the Hive web interface. On the left, a sidebar shows the database 'trabajo1_fdm' and a table 'hpolution'. The main area contains the SQL query for creating the 'hpolution' table, which is stored as a text file in an S3 location. Below the query editor, the execution status is shown as 'Success'. The 'Query History' tab is active, showing the query was executed 'hace unos segundos' (a few seconds ago). The 'Saved Queries' tab is also visible.

```
1 CREATE EXTERNAL TABLE hpolution (country STRING, city STRING, aqi_value INT, aqi_category STRING, co_aqi_value INT, co_aqi_category STRING, oz_aqi_value INT, oz_aqi_category STRING, no2_aqi_value INT, no2_aqi_category STRING, pm25_aqi_value INT, pm25_aqi_category STRING)  
2 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
3 STORED AS TEXTFILE  
4 LOCATION 's3://trabajo1-raw/datasets/global air pollution dataset/'
```

INFO : Compiling command(queryId=hive_20240314020843_11dc3d76-813c-4955-8168-83d14a627b19): CREATE EXTERNAL TABLE hpolution (country STRING, city STRING, aqi_value INT, aqi_category STRING, co_aqi_value INT, co_aqi_category STRING, oz_aqi_value INT, oz_aqi_category STRING, no2_aqi_value INT, no2_aqi_category STRING, pm25_aqi_value INT, pm25_aqi_category STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 's3://trabajo1-raw/datasets/global air pollution dataset/'

✓ Success.

Query History	Saved Queries
hace unos segundos ✓	CREATE EXTERNAL TABLE hpolution (country STRING, city STRING, aqi_value INT, aqi_category STRING, co_aqi_value INT, co_aqi_category STRING, oz_aqi_value INT, oz_aqi_category STRING, no2_aqi_value INT, no2_aqi_category STRING, pm25_aqi_value INT, pm25_aqi_category STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 's3://trabajo1-raw/datasets/global air pollution dataset/'
hace un minuto ✓	USE trabajo1_fdm

Consultas:

Hive

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

< trabajo1_fdm

Tables (1) + ↺

Filter...

hpolution

10.60s trabajo1_fdm ▾ ⚙ ?

```
1 SELECT country, count(city) as count_city
2 from hpolution
3 where aqi_value > 60
4 group by country
5 order by count_city DESC
```

▶ ▾

📖 ▾

INFO : Completed executing command(queryId=hive_20240314024309_045c5b047aa040e00013-application_1710379701539_000137); Time taken: 8.124 seconds

INFO : OK

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

Query History

Saved Queries

Results (155)

country

count_city

1	India	2206
2	United States of America	1053
3	China	701
4	Italy	474
5	Mexico	347
6	Germany	326
7	Pakistan	306
8	Brazil	265

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

7.54s trabajo1_fdm ▾ ⚙ ?

```

1 SELECT country, count(city) as count_city
2 from hpolution
3 where aqi_value > 60
4 group by country
5 order by count_city DESC;
6

```

INFO : map 1: 1/1 READER 2: 2/2 REDUCER 3: 1/1

INFO : Completed executing command(queryId=hive_20240314025124_application_1710379701539_0002)

e); Time taken: 7.539 seconds

INFO : OK

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

Query History Saved Queries Results (100+)

TYPE

Bars

X-AXIS

country

Y-AXIS

☒ count_city

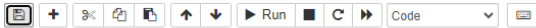
GRUPO

Choose a colu...

LIMIT

-

Country	count_city
India	~2.2M
United States of America	~1.1M
China	~700k
Italy	~500k
Mexico	~400k



In [23]: df.show(5)

```
+-----+-----+-----+-----+-----+
|      City      |      Region      |      Country      |      AirQuality      |      WaterPollution      |
+-----+-----+-----+-----+-----+
| New York City  | New York         | United States of ... | 46.81603774         | 49.5049505             |
| Washington, D.C. | District of Columbia | United States of ... | 66.12903226         | 49.10714286            |
| San Francisco  | California        | United States of ... | 60.51401869         | 43.0                   |
| Berlin         | null             | Germany             | 62.36413043         | 28.61271676            |
| Los Angeles    | California        | United States of ... | 36.62162162         | 61.29943503            |
+-----+-----+-----+-----+-----+
only showing top 5 rows
```

In [24]: df.describe().show()

```
+-----+-----+-----+-----+-----+
|summary| City      |      Region      |      Country      |      AirQuality      |      WaterPollution      |
+-----+-----+-----+-----+-----+
| count | 3963      | 3538             | 3963              | 3963                 | 3963                     |
| mean  | NaN       | NaN             | null             | 62.2534517395642    | 44.63537214186169       |
| stddev| null      | null            | null            | 30.94475340271576   | 25.663910130285892      |
| min   | Aachen    | ???:il Province | Afghanistan       | 0.0                  | 0.0                      |
| max   | Zyrwiec   | canton of Fribourg | Zimbabwe          | 100.0                | 100.0                    |
+-----+-----+-----+-----+-----+
```



```
+-----+-----+-----+-----+-----+
| mean  | NaN       | NaN             | null             | 62.2534517395642    | 44.63537214186169       |
| stddev| null      | null            | null            | 30.94475340271576   | 25.663910130285892      |
| min   | Aachen    | ???:il Province | Afghanistan       | 0.0                  | 0.0                      |
| max   | Zyrwiec   | canton of Fribourg | Zimbabwe          | 100.0                | 100.0                    |
+-----+-----+-----+-----+-----+
```

In [40]: from pyspark.sql.functions import udf

```
In [42]: ## Creando una columna con datos 1 = mayor que la media y 0 = menor que la media

AQMean = udf(lambda AirQuality: "0" if AirQuality <= 62.25 else "1")

df.withColumn("AQ>Mean", AQMean(df.AirQuality)).show(10,False)
```

```
+-----+-----+-----+-----+-----+-----+
|City      |Region      |Country      |AirQuality|WaterPollution|AQ>Mean|
+-----+-----+-----+-----+-----+-----+
|New York City|New York    |United States of America|46.81603774|49.5049505     |0       |
|Washington, D.C.|District of Columbia|United States of America|66.12903226|49.10714286    |1       |
|San Francisco |California  |United States of America|60.51401869|43.0           |0       |
|Berlin        |null       |Germany       |62.36413043|28.61271676    |1       |
|Los Angeles   |California  |United States of America|36.62162162|61.29943503    |0       |
|Bern          |Canton of Bern|Switzerland   |94.31818182|12.5           |1       |
|Geneva        |Canton of Geneva|Switzerland   |71.53846154|17.37288136    |1       |
|Zurich        |Canton of Zurich|Switzerland   |83.80952381|10.71428571    |1       |
|Basel         |null       |Switzerland   |81.66666667|26.92307692    |1       |
|London        |England     |United Kingdom|37.04225352|40.71637427    |0       |
+-----+-----+-----+-----+-----+-----+
only showing top 10 rows
```


jupyterhub

Trabajo1_FDM

Last Checkpoint: hace un minuto (autosaved)

Logout

Control Panel

FileEditViewInsertCellKernelWidgetsHelp

TrustedPySpark

+

↶

↷

↕

↕

▶ Run

■

↺

↻

▶▶

Code

⌵

⌵

In [72]:

```
## No se guarda La columna AQMean
df.show(5)
```

City	Region	Country	AirQuality	WaterPollution
New York City	New York	United States of ...	46.81603774	49.5849505
Washington, D.C.	District of Columbia	United States of ...	66.12903226	49.10714286
San Francisco	California	United States of ...	60.51401869	43.0
Berlin	null	Germany	62.36413043	28.61271676
Los Angeles	California	United States of ...	36.62162162	61.29943503

only showing top 5 rows

In [12]:

```
## Ciudades de Colombia que tienen contaminación del agua superior a La media mundial
df.filter(df['Country']=='Colombia').filter(df['WaterPollution'] >44.63).show()
```

City	Region	Country	AirQuality	WaterPollution
Bogota	null	Colombia	23.35025381	48.15789474
Pereira	Risaralda Department	Colombia	25.0	75.0
Barranquilla	Atlantico Department	Colombia	52.77777778	50.0
Santa Marta	Magdalena Department	Colombia	50.0	71.42857143
Ibague	Tolima Department	Colombia	56.61764706	79.04411765
Tunja	Boyaca Department	Colombia	100.0	75.0
San Juan de Pasto	Narino Department	Colombia	70.0	75.0
Valledupar	Cesar Department	Colombia	50.0	50.0
Neiva	Huila Department	Colombia	75.0	75.0
Cartagena	Bolivar Department	Colombia	60.0	65.0

jupyterhub

Trabajo1_FDM

Last Checkpoint: hace un minuto (autosaved)

Logout

Control Panel

FileEditViewInsertCellKernelWidgetsHelp

TrustedPySpark

+

↶

↷

↕

↕

▶ Run

■

↺

↻

▶▶

Code

⌵

⌵

In [13]:

```
## Regiones con más ciudades en dataframe
df.groupBy('Country','Region').count().orderBy('count',ascending=False).show(20,False)
```

Country	Region	count
United Kingdom	England	142
United States of America	California	122
Czech Republic	null	57
United States of America	Texas	51
United States of America	Florida	48
Canada	Ontario	47
France	null	42
Canada	British Columbia	38
Philippines	null	36
Brazil	Sao Paulo	32
United States of America	Georgia	31
United States of America	New York	31
Germany	North Rhine-Westphalia	30
United States of America	Washington	29
United States of America	North Carolina	27
United States of America	Indiana	26
United States of America	Illinois	26
United States of America	Michigan	24
Finland	null	23
Germany	Bavaria	23

only showing top 20 rows

Trabajo1_FDM
Last Checkpoint: hace 2 minutos (autosaved)
Logout
Control Panel

File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
PySpark

In [14]:
from pyspark.sql.types import StringType, DoubleType, IntegerType

In [15]:
df.filter(df['Country']=='Colombia').filter(df['AirQuality'] > 30).show()

City	Region	Country	AirQuality	WaterPollution
Cali	Valle del Cauca	Colombia	67.04545455	42.1875
Barranquilla	Atlantico	Colombia	52.77777778	50.0
Santa Marta	Magdalena	Colombia	50.0	71.42857143
Ibague	Tolima	Colombia	56.61764706	79.04411765
Tunja	Boyaca	Colombia	100.0	75.0
Bucaramanga	Santander	Colombia	40.0	25.0
San Juan de Pasto	Narino	Colombia	70.0	75.0
Armenia	Quindio	Colombia	62.5	37.5
Popayan	Cauca	Colombia	100.0	25.0
Valledupar	Cesar	Colombia	50.0	50.0
Neiva	Huila	Colombia	75.0	75.0
Cartagena	Bolivar	Colombia	60.0	65.0
Villavicencio	Meta	Colombia	100.0	25.0

In [17]:
from pyspark.sql.functions import pandas_udf, PandasUDFType

In [18]:
def critical_airq(AirQuality):
 quality_left= 2 * df.mean()-AirQuality
 return quality_left

Trabajo1_FDM
Last Checkpoint: hace 2 minutos (autosaved)
Logout
Control Panel

File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
PySpark

Neiva
Huila
Department
Colombia
75.0
75.0

Cartagena
Bolivar
Department
Colombia
60.0
65.0

Villavicencio
Meta
Colombia
100.0
25.0

click to scroll output; double click to hide

In [17]:
from pyspark.sql.functions import pandas_udf, PandasUDFType

In [18]:
def critical_airq(AirQuality):
 quality_left= 2 * df.mean()-AirQuality
 return quality_left

In [73]:
Intentamos guardar el DataFrame con la nueva columna en una nueva variable, pero al correr el código nos mostró el siguiente error

Rdf.write.format('csv').option('header','true').save('s3://trabajo1-refined/datarefine/water_quality/Rdf_csv',mode='overwrite')

An error was encountered:
'NoneType' object has no attribute 'write'
Traceback (most recent call last):
AttributeError: 'NoneType' object has no attribute 'write'

In [74]:
df.write.format('csv').option('header','true').save('s3://trabajo1-refined/datarefine/water_quality/df_csv',mode='overwrite')

URI S3 archivo exportado: s3://trabajo1-refined/datarefine/water_quality/

Amazon S3 > Buckets > trabajo1-refined > datarefine/ > water_quality/

water_quality/

S3 URI copied

Copy S3 URI

Objects | Properties

Objects (1) Info

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	df_csv/	Folder	-	-	-

Notebook creado en S3 desde jupyterhub

URI S3: s3://mholguinc2notebooks/jupyter/

Amazon S3 > Buckets > mholguinc2notebooks > jupyter/ > jovyan/

jovyan/

Copy S3 URI

Objects | Properties

Objects (6) Info

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	.s3keep	s3keep	March 14, 2024, 20:11:35 (UTC-05:00)	0 B	Standard
<input type="checkbox"/>	demo1.ipynb	ipynb	March 3, 2024, 15:55:36 (UTC-05:00)	2.5 KB	Standard
<input type="checkbox"/>	Trabajo1-FDM.ipynb	ipynb	March 14, 2024, 21:12:22 (UTC-05:00)	17.8 KB	Standard
<input type="checkbox"/>	Untitled.ipynb	ipynb	March 3, 2024, 15:51:15 (UTC-05:00)	648.0 B	Standard