

**Almacenamiento y Recuperación de Información**  
**Maestría de Ciencia de Datos y Analítica**  
**2024 -01**  
**Trabajo 1**

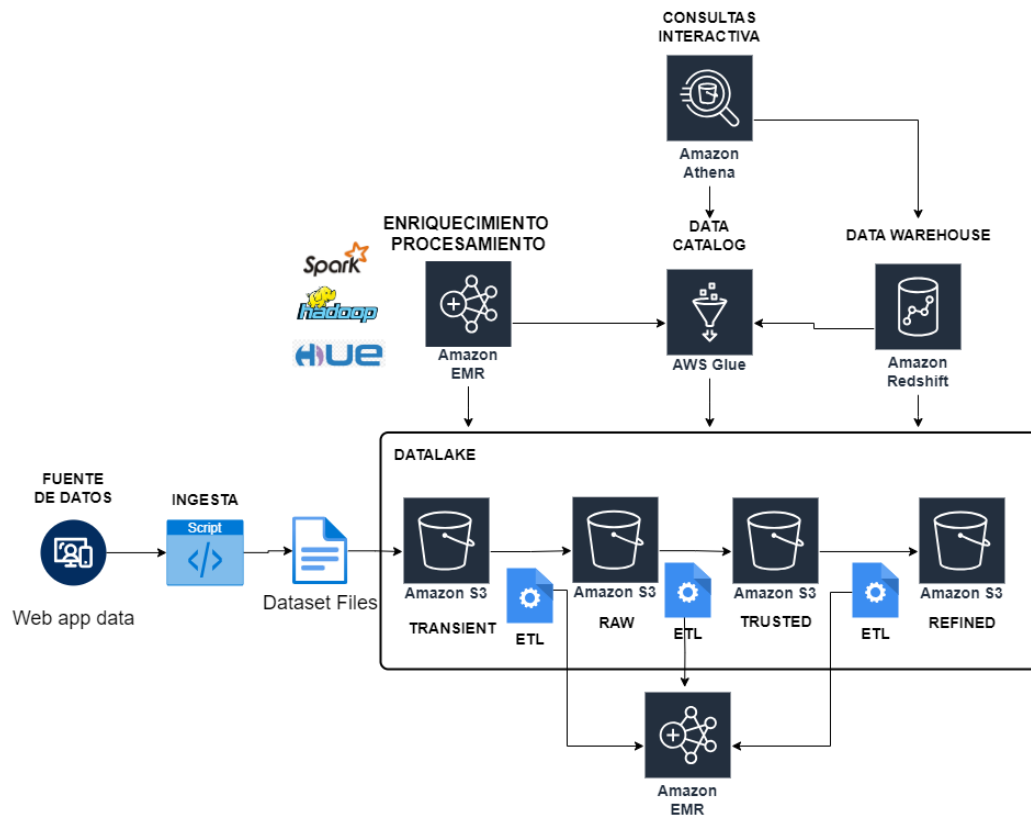
**Integrantes:**

**Daniel Molina Leon**  
**Andres Felipe Restrepo Acevedo**  
**John Jairo Gonzalez Ruiz**

**1. Proceso de Implementación**

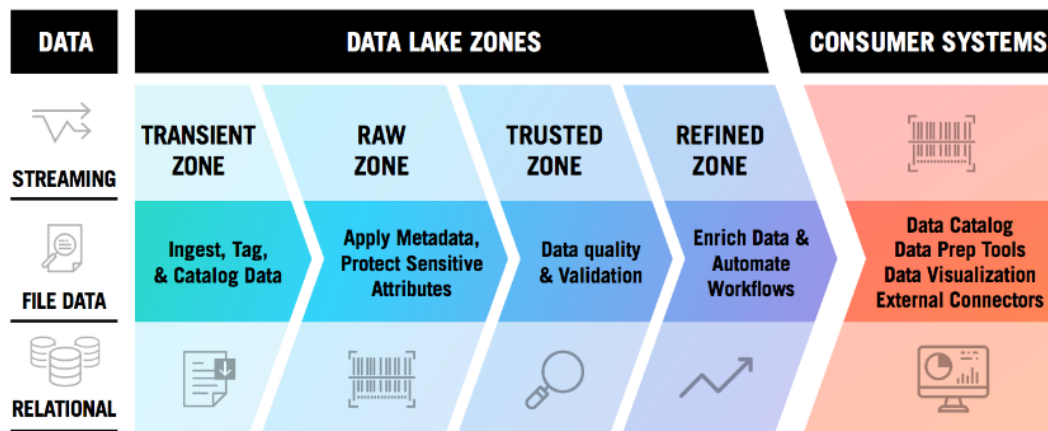
Datalake

Para desarrollar un Datalake puede tenerse como guía alguna arquitectura de referencia, se usará la siguiente:



El datalake cuenta con cuatro zonas, 'Transient' donde se descargan los archivos temporales manualmente o por medio de una automatización (script). Zona 'Raw' donde se acumula la información de los objetos (archivos) tomados de la zona Transient y se guarda la información en formato parquet sin ninguna transformación. Zona 'Trusted' donde se almacena la información almacenada en Raw pero con el

preprocesamiento de limpieza de los datos y finalmente la zona ‘*Refined*’ donde se guardan las tablas y archivos con información enriquecida.



## 2. Datalake AWS

### a. Fuentes

Para el desarrollo de este trabajo se identificaron 2 fuentes de datos sobre cambio climático y calentamiento global. Para la selección de estas fuentes de datos, se realizó una búsqueda basada en las variables relacionadas con las causas y efectos del cambio climático referenciado en:

<https://www.kaggle.com/code/andradaolteanu/plotly-advanced-global-warming-analysis/notebook>

Donde se hace un análisis avanzado del calentamiento global y cómo este ha evolucionado a lo largo de los años, abordando preguntas relacionadas con el cambio climático, sus patrones y los efectos que este cambio ha tenido en el incremento de las temperaturas en las diferentes regiones del planeta, el cambio en el nivel de los océanos y eventos climáticos extremos que se han vivido en los últimos años.

A continuación, se comparte información general y los enlaces de orígenes de datos:

**Fuentes de datos #1** – Aquí se encuentran set de datos con información relacionada con emisiones de dióxido de carbono, gases de efecto invernadero, nivel de los océanos y temperatura de la superficie:

<https://climatedata.imf.org/pages/access-data>

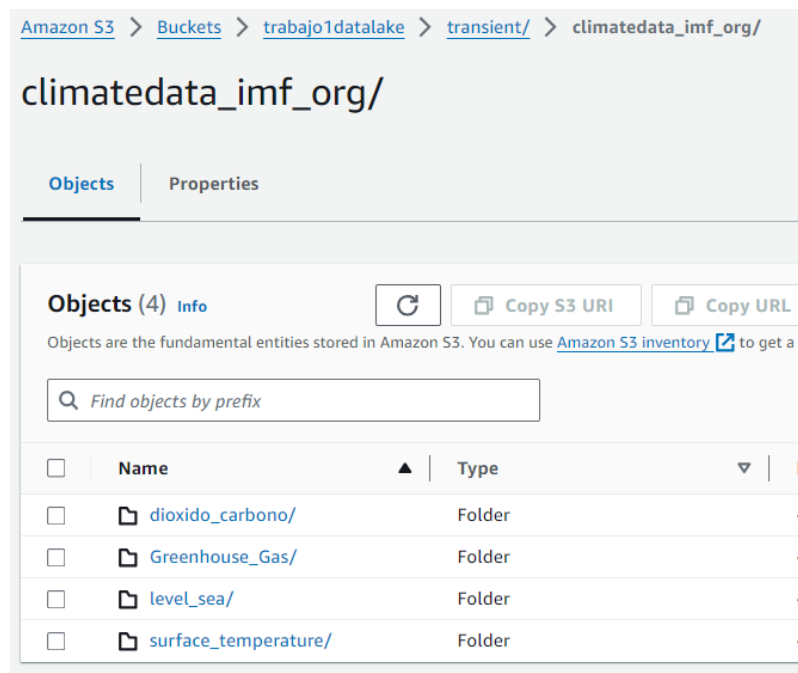
**Fuentes de datos #2** - Acá se encuentran set de datos con información relacionada con emisiones de CO2 anuales, per cápita, acumuladas y basadas en el consumo:

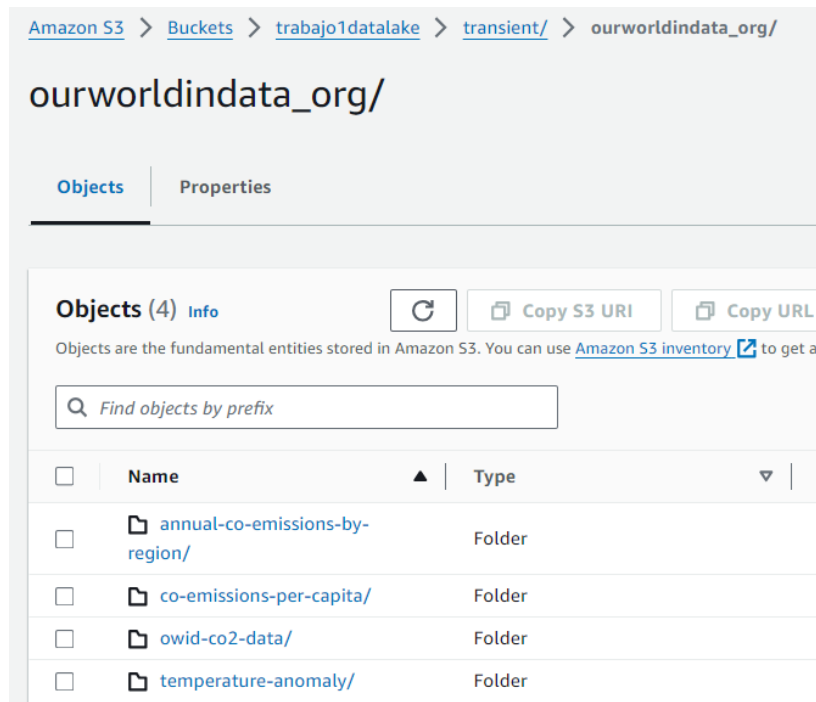
<https://ourworldindata.org/co2-and-greenhouse-gas-emissions#explore-data-on-co2-and-greenhouse-gas-emissions>

<https://github.com/owid/co2-data>

b. Buckets s3

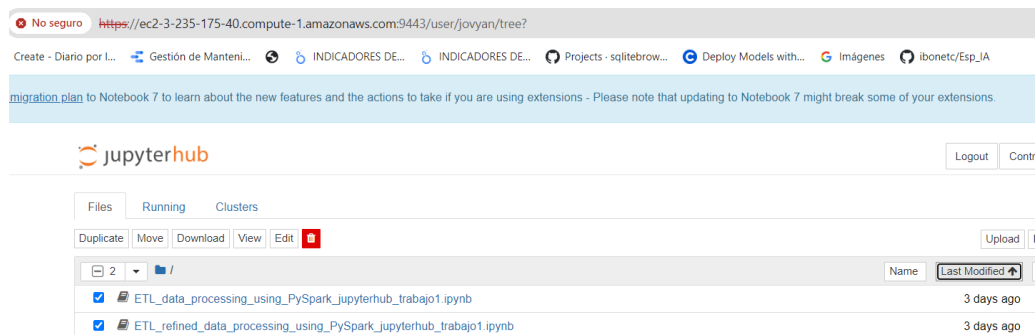
Los sets de datos se descargaron manualmente y fueron llevados a la zona 'transient' del datalake desplegado en el bucket 'trabajo1datalake' de S3. Esto se podría automatizar con un script o un RPA.





### c. ETL

Se desarrollaron los ETL desde los notebooks de Jupyter con Spark para crear y poblar las diferentes zonas del Datalake.



**Notebook:** [ETL\\_data\\_processing\\_using\\_PySpark\\_jupyterhub\\_trabajo1.ipynb](#)

Los **fuentes de entrada** son los data set de los folder [climatedata\\_imf\\_org](#) y [ourworldindata\\_org](#) en la zona transient del datalake, los cuales son archivos temporales en formato 'csv' y son ingestados manualmente o mediante una automatización cada mes que se actualicen en las páginas web origen.

Amazon S3 > Buckets > trabajo1datalake > transient/ > climatedata\_imf\_org/ > level\_sea/

## level\_sea/

Objects | Properties

**Objects (1)** Info Refresh Copy S3 URI Copy URL Download

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects.

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	<a href="#">Change_in_Mean_Sea_Levels.csv</a>	CSV	March 7, 2024 05:00

Los **data sets de salida** son procesados y pasados por las zonas del Datalake donde son almacenados en formato parquet (notebook de ETL en Spark). En 'raw' se consolida la información en formato delta, en 'trusted' se limpian los datos y se hace preprocesamiento de limpieza y en 'refine' se crean datasets con columnas seleccionadas, agrupadas por promedios y acumulados, y la unión de tablas que generan valor a los usuarios finales.

jupyterhub ETL\_data\_processing\_using\_PySpark\_jupyterhub\_trabajo1 (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Not Trusted PySpark

FloatProgress(value=0.0, bar\_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),...

### Raw to trusted

Limpiar los datos:

- Identificar, eliminar o completar nulos
- eliminar duplicados
- cambiar formatos de datos
- corregir valores y/o strings

### Clean data source 1

```
In [11]: from pyspark.sql.functions import regexp_replace, col
from pyspark.sql.types import StringType, DoubleType, IntegerType
from pyspark.sql.functions import isnan, when, count, col

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),...

In [13]: path_raw_source_1='s3://trabajo1datalake/raw/climatedata_imf_org/level_sea'
# Load raw data
df_raw_source_1=spark.read.parquet(path_raw_source_1,
                                   inferSchema=True,
                                   header=True)
```

[Amazon S3](#) > [Buckets](#) > [trabajo1datalake](#) > [raw/](#) > [climatedata\\_imf\\_org/](#) > [level\\_sea/](#)

level\_sea/

Objects

Properties

Objects (3) Info

Copy S3 URI

Copy URL

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	_SUCCESS	-	March 7, 2024, 05:00
<input type="checkbox"/>	part-00000-4c3dab97-ae5b-41b3-978b-e415376ff701-c000.snappy.parquet	parquet	March 7, 2024, 05:00
<input type="checkbox"/>	part-00001-4c3dab97-ae5b-41b3-978b-e415376ff701-c000.snappy.parquet	parquet	March 7, 2024, 05:00

[Amazon S3](#) > [Buckets](#) > [trabajo1datalake](#) > [trusted/](#) > [climatedata\\_imf\\_org/](#) > [level\\_sea/](#)

level\_sea/

Objects

Properties

Objects (20) 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](#)

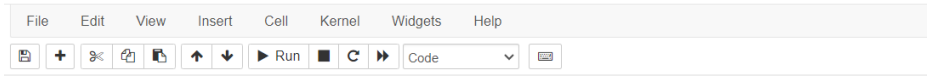
Find objects by prefix

< 1 > ⌕

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	_SUCCESS	-	March 7, 2024, 15:55:32 (UTC-05:00)	0 B	Standard
<input type="checkbox"/>	part-00000-5c05c037-2a47-4724-b7fc-614063813e6e-c000.snappy.parquet	parquet	March 7, 2024, 15:55:28 (UTC-05:00)	32.0 KB	Standard
<input type="checkbox"/>	part-00001-5c05c037-2a47-4724-b7fc-614063813e6e-c000.snappy.parquet	parquet	March 7, 2024, 15:55:28 (UTC-05:00)	32.3 KB	Standard
<input type="checkbox"/>	part-00002-5c05c037-2a47-4724-b7fc-614063813e6e-c000.snappy.parquet	parquet	March 7, 2024, 15:55:28 (UTC-05:00)	32.2 KB	Standard

## Zona Refined

jupyterhub ETL\_refined\_data\_processing\_using\_PySpark\_jupyterhub\_trabajo1 (unsaved changes)



### Data Processing and EDA using Pyspark

#### Data from trusted to Refine

- Cargar dataset zona trusted
- Seleccionar columnas de interes
- Agrupar tablas por mes año (valor promedio de nivel del mar y dióxido de carbono)
- Unir tablas level\_sea y dióxido de carbono por año y mes
- Explorar datos

#### load data from trusted

```
In [2]: path_trusted_source_1='s3://trabajo1datalake/trusted/climatedata_imf_org/level_sea'
# Load raw data
df_trusted_source_1=spark.read.parquet(path_trusted_source_1,
                                       inferSchema=True,
                                       header=True)
```

```
In [3]: path_trusted_source_1='s3://trabajo1datalake/trusted/climatedata_imf_org/dioxido_carbono'
# Load raw data
df_trusted_source_1=spark.read.parquet(path_trusted_source_1,
                                       inferSchema=True,
                                       header=True)
```

```
In [4]: path_trusted_source_2='s3://trabajo1datalake/trusted/ourworldindata_org/owid-co2-data'
# Load raw data
df_trusted_source_2=spark.read.parquet(path_trusted_source_2,
```

Notebook:

[ETL\\_refined\\_data\\_processing\\_using\\_PySpark\\_jupyterhub\\_trabajo1.ipynb](#)

[Amazon S3](#) > [Buckets](#) > [trabajo1datalake](#) > [refined/](#)

## refined/

[Objects](#) | [Properties](#)

**Objects (2)** [Info](#) [Refresh](#) [Copy S3 URI](#) [Copy URI](#)

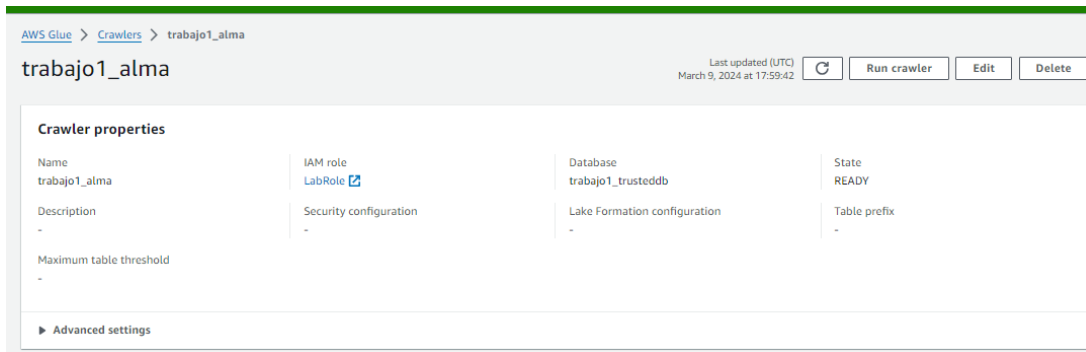
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<a href="#">co2_and_level_sea/</a>	Folder
<input type="checkbox"/>	<a href="#">df_co2_level_sea/</a>	Folder

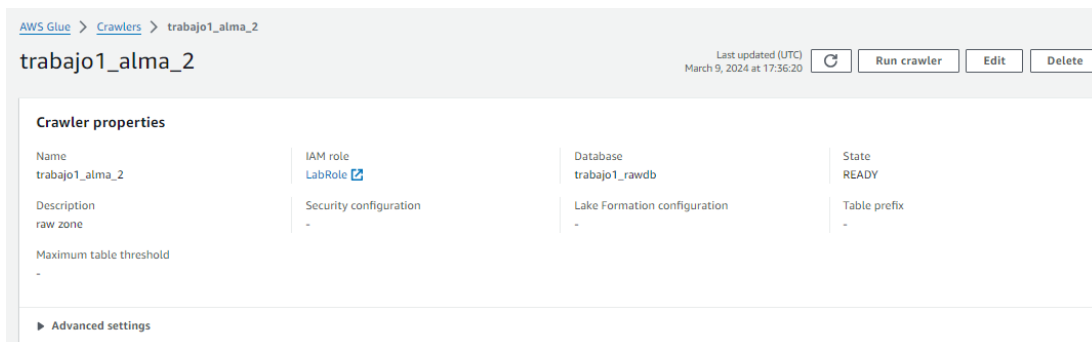
#### d. Glue Catalogo

Para lograr realizar consultas al datalake en los buckets de s3 por Athena es necesario crear unos catálogos en Glue, se crearon dos crawlers (uno para cada zona interesada en consultar) que traducen la zona Raw y la zona Trusted. Estos crawlers son trabajo1\_alma y trabajo1\_alma\_2.

#### Glue – Zona Trusted



#### Glue – Zona Raw



Luego de implementar los crawlers es posible realizar las consultas por Athena a cada una de las zonas.

#### e. Athena

Luego de implementar los crawlers es posible realizar las consultas por Athena a cada una de las zonas.



## Athena – Query Zona Trusted

The screenshot shows the AWS Athena console interface. On the left, the 'Data' sidebar shows the 'trabajo1\_trusteddb' database. The main panel displays a query: `SELECT * FROM "trabajo1_trusteddb"."climatedata_inf_org" limit 10;`. The query is completed, with a run time of 1.264 seconds and 222.04 KB of data scanned. The results table shows 10 rows, with the first three rows displayed:

#	objectid	country	iso2	iso3	Indicator	unit	source
1	1093	World	NA	WLD	Change_in_mean_sea_level_Sea_level_TOPEXPoseidon	Millimeters	National_Oceanic_and_Atmosph
2	4171	World	NA	WLD	Change_in_mean_sea_level_Sea_level_TOPEXPoseidon	Millimeters	National_Oceanic_and_Atmosph
3	4503	World	NA	WLD	Change_in_mean_sea_level_Sea_level_TOPEXPoseidon	Millimeters	National_Oceanic_and_Atmosph

## Athena – Query zona Raw

The screenshot shows the AWS Athena console interface for a query in the raw zone. The query is: `SELECT * FROM "trabajo1_rawdb"."climatedata_inf_org" limit 10;`. The query is completed, with a run time of 1.245 seconds and 17.70 KB of data scanned. The results table shows 10 rows, with the first three rows displayed:

#	objectid	country	iso2	iso3	Indicator	unit	source
1	1	World		WLD	Monthly Atmospheric Carbon Dioxide Concentrations	Parts Per Million	Dr. Pieter Tans, National Oceanic and Atmospheric Administration
2	4	World		WLD	Monthly Atmospheric Carbon Dioxide Concentrations	Parts Per Million	Dr. Pieter Tans, National Oceanic and Atmospheric Administration
3	5	World		WLD	Monthly Atmospheric Carbon Dioxide Concentrations	Parts Per Million	Dr. Pieter Tans, National Oceanic and Atmospheric Administration

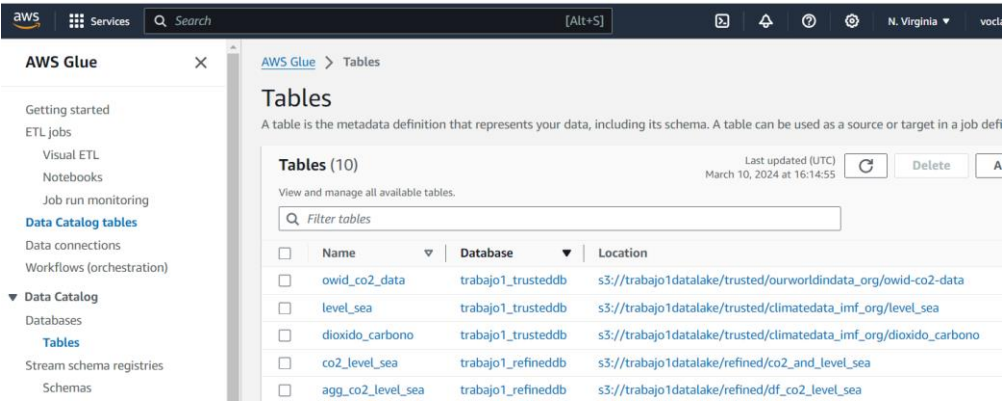
## f. EMR/Hive

## Despliegue de Cluster EMR

The screenshot shows the Amazon EMR console for a cluster named 'cluster\_trabajo1'. The cluster is in the 'Waiting' status. The summary section provides the following details:

- Cluster info:** Cluster ID: j-3671OTL841HR1, Instance groups: 1 Primary, 1 Core, 1 Task.
- Applications:** Amazon EMR version: emr-6.14.0, Installed applications: Flink 1.17.1, Hadoop 3.3.3, HCatalog 3.1.3, Hive 3.1.3, Hue 4.11.0, JupyterEnterpriseGateway 2.6.0, JupyterHub 1.5.0, Livy 0.7.1, Spark 3.4.1, Sqoop 1.4.7, Tez 0.10.2, Zeppelin 0.10.1, ZooKeeper 3.5.10.
- Cluster management:** Log destination in Amazon S3: [aws-logs-637423206752-us-east-1/elasticmapreduce](#), Persistent application UIs: [Spark History Server](#), [YARN timeline server](#), [Tez UI](#), Primary node public DNS: [ec2-3-236-66-75.compute-1.amazonaws.com](#), Connect to the Primary node using SSH, Connect to the Primary node using SSM.
- Status and time:** Status: [Waiting](#), Creation time: March 07, 2024, 12:46, Elapsed time: 1 hour, 6 minutes.

Bases de datos y tablas en aws Glue, catalogadas en Hive desde EMR.



Para el desarrollo de este trabajo, usaremos los servicios:

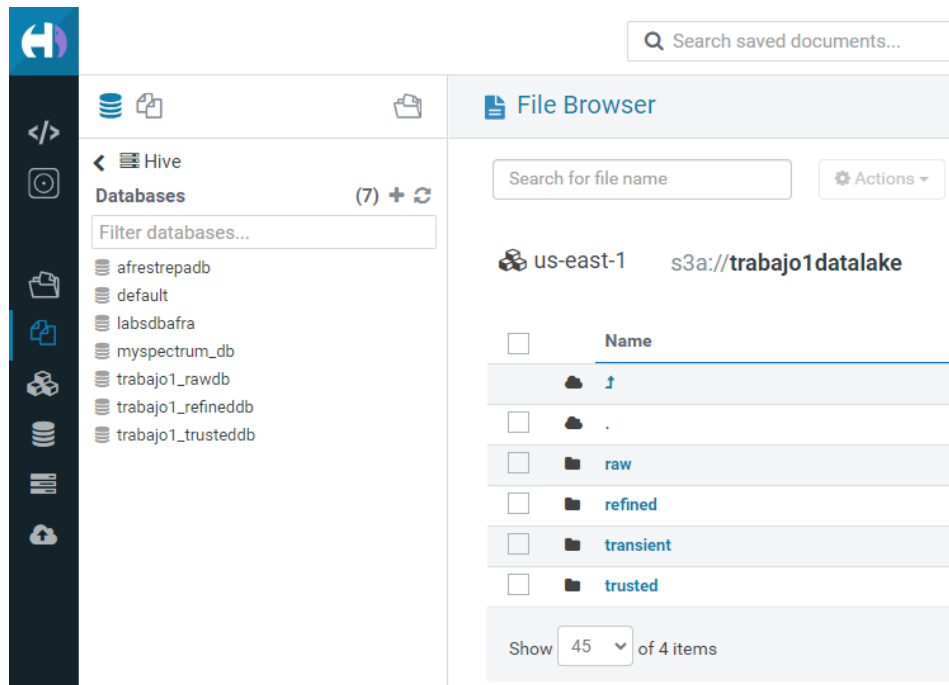
**HUE:** Generar los scripts para catalogar y hacer un EDA para explorar algunas tablas.

**JupyterHub:** Desarrollar 2 notebooks con las ETLs entre zonas del datalake y análisis exploratorio de algunas zonas usando Spark.

Application UIs on the primary node	
These require SSH tunneling to be enabled.	
Application	UI URL
HDFS Name Node	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:9870/">http://ec2-3-235-175-40.compute-1.amazonaws.com:9870/</a>
Hue	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:8888/">http://ec2-3-235-175-40.compute-1.amazonaws.com:8888/</a>
JupyterHub	<a href="https://ec2-3-235-175-40.compute-1.amazonaws.com:9443/">https://ec2-3-235-175-40.compute-1.amazonaws.com:9443/</a>
Livy	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:8998/">http://ec2-3-235-175-40.compute-1.amazonaws.com:8998/</a>
Resource Manager	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:8088/">http://ec2-3-235-175-40.compute-1.amazonaws.com:8088/</a>
Spark History Server	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:18080/">http://ec2-3-235-175-40.compute-1.amazonaws.com:18080/</a>
Tez UI	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:8080/tez-ui">http://ec2-3-235-175-40.compute-1.amazonaws.com:8080/tez-ui</a>
Zeppelin	<a href="http://ec2-3-235-175-40.compute-1.amazonaws.com:8890/">http://ec2-3-235-175-40.compute-1.amazonaws.com:8890/</a>

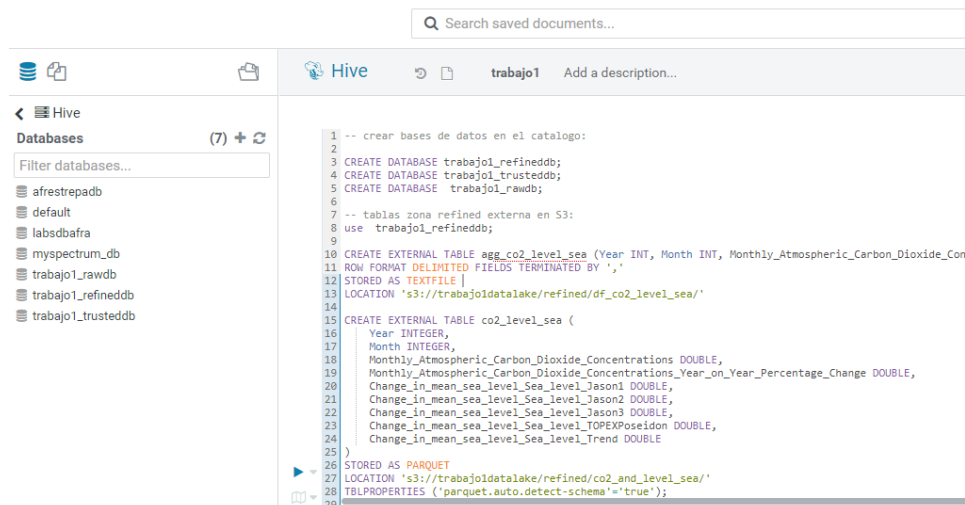
Hue

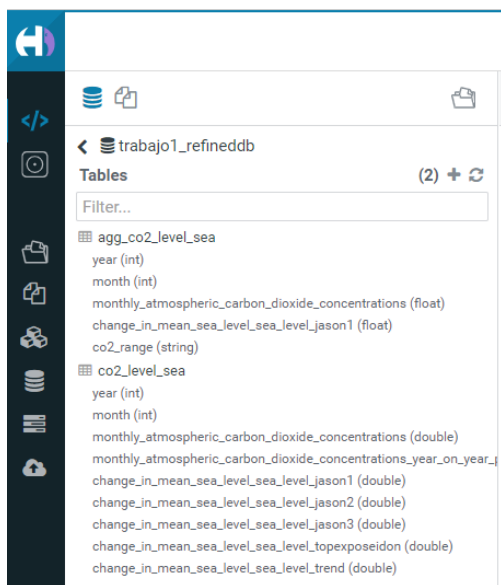
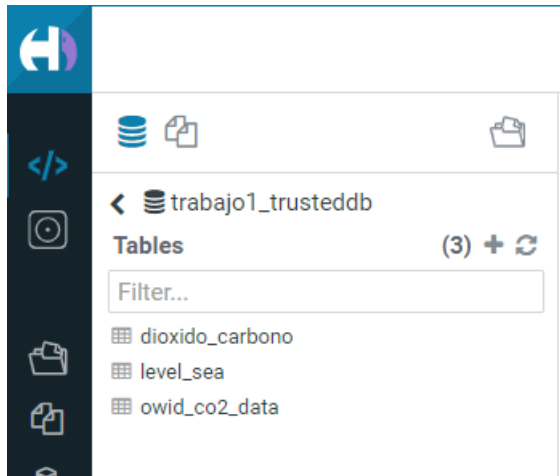
Creación catálogo de bases de datos y tablas por zona del datalake.



Scripts en SQL de creación de bases de datos y tablas zonas datalake:

[https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa\\_eafit\\_edu\\_co/EVsWwBGbWGFEqlu6Nk8o6EsB7j-unabbM-OA3uNx5q5uCtg?e=WLna2d](https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa_eafit_edu_co/EVsWwBGbWGFEqlu6Nk8o6EsB7j-unabbM-OA3uNx5q5uCtg?e=WLna2d)






## Exploración de tablas desde HIVE

Scripts SQL de exploración de tablas usando Hive:

[https://eafit-my.sharepoint.com/:t/g/personal/afrestrepa\\_eafit\\_edu\\_co/EVPxTOP-15RKnq4c2Yw-XdgBLN4C0v6EMvZHZ0lcFAPGRA?e=H48ZIB](https://eafit-my.sharepoint.com/:t/g/personal/afrestrepa_eafit_edu_co/EVPxTOP-15RKnq4c2Yw-XdgBLN4C0v6EMvZHZ0lcFAPGRA?e=H48ZIB)

 Hive

consultas BD catalogo

Add a description...

```
1 use trabajo1_refineddb;
2 show tables;
3 describe agg_co2_level_sea;
```

INFO : Calculating combiner(queryId=hive\_20240308040718\_a5f693a4-7352-4ae8-a0b9-a43c6b052009); User: hive\_agg\_co2\_level\_sea

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

INFO : Completed executing command(queryId=hive\_20240308040718\_a5f693a4-7352-4ae8-a0b9-a43c6b052009); Time taken: 0.117 sec

INFO : OK


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

Query History

Saved Queries

Results (5)

	col_name	data_type
1	year	int
2	month	int
3	monthly_atmospheric_carbon_dioxide_concentrations	float
4	change_in_mean_sea_level_sea_level_json1	float
5	co2_range	string

 Hive

consultas BD catalogo

Add a description...

```
1 use trabajo1_refineddb;
2 show tables;
3 describe agg_co2_level_sea;
4 select * from agg_co2_level_sea;
5
6 select distinct(co2_range) from agg_co2_level_sea where co2_range;
7
8 select * from agg_co2_level_sea where co2_range = 'High';
9
10 SELECT COUNT(*) AS total_meses_con_alta_concentracion_co2
11 FROM agg_co2_level_sea
12 WHERE co2_range = 'High';
13
14 SELECT Year, COUNT(*) AS total_meses_con_alta_concentracion_co2
15 FROM agg_co2_level_sea
16 WHERE co2_range = 'High'
17 GROUP BY Year;
```

INFO : Map 1: 1/1 Reducer 2: 1/1/1/4

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

INFO : Completed executing command(queryId=hive\_20240308042018\_909338dc-4e0f

INFO : OK

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

Query History

Saved Queries

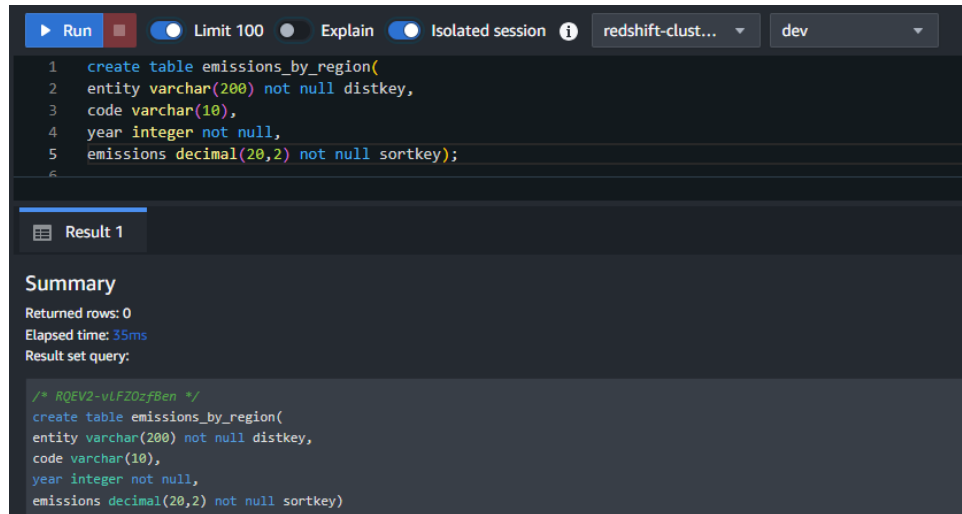
Results (7)

	year	total_meses_con_alta_concentracion_co2
1	2018	12
2	2019	12
3	2020	12
4	2021	12
5	2016	4
6	2017	10
7	2022	11

g. RedShift - Spectrum

Para el manejo de tablas nativas se crean en RedShift las tablas de los data sets para las emisiones por región y per cápita.

Tabla para el data set de emisiones por región:



The screenshot shows the AWS RedShift console interface. At the top, there are buttons for 'Run', 'Limit 100', 'Explain', and 'Isolated session'. Below these, the SQL query to create the 'emissions\_by\_region' table is displayed. The query is as follows:

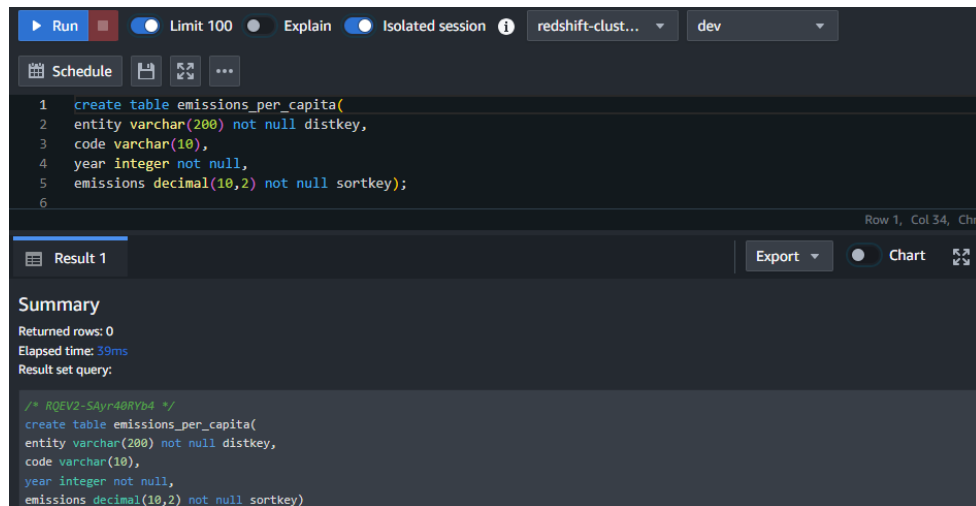
```
1 create table emissions_by_region(  
2 entity varchar(200) not null distkey,  
3 code varchar(10),  
4 year integer not null,  
5 emissions decimal(20,2) not null sortkey);
```

Below the query, the 'Result 1' section shows a 'Summary' with the following details:

- Returned rows: 0
- Elapsed time: 35ms
- Result set query: 

```
/* RQEV2-vLFZ0zfBen */  
create table emissions_by_region(  
entity varchar(200) not null distkey,  
code varchar(10),  
year integer not null,  
emissions decimal(20,2) not null sortkey)
```

Tabla para el data set de emisiones per cápita:



The screenshot shows the AWS RedShift console interface. At the top, there are buttons for 'Run', 'Limit 100', 'Explain', and 'Isolated session'. Below these, the SQL query to create the 'emissions\_per\_capita' table is displayed. The query is as follows:

```
1 create table emissions_per_capita(  
2 entity varchar(200) not null distkey,  
3 code varchar(10),  
4 year integer not null,  
5 emissions decimal(10,2) not null sortkey);
```

Below the query, the 'Result 1' section shows a 'Summary' with the following details:

- Returned rows: 0
- Elapsed time: 39ms
- Result set query: 

```
/* RQEV2-S4yr40RYb4 */  
create table emissions_per_capita(  
entity varchar(200) not null distkey,  
code varchar(10),  
year integer not null,  
emissions decimal(10,2) not null sortkey)
```

Se cargan los datos desde los archivos almacenados en S3 en la zona 'Transient'

Carga de los datos para el data set de emisiones por región:

```
Run Limit 100 Explain Isolated session redshift-clust... dev Schedule
8
9 COPY emissions_by_region FROM 's3://trabajoidatalake/transient/ourworldindata_org/annual-co-emissions-by-region/annual-co-emissions-by-region.csv'
10 iam_role 'arn:aws:iam::851725398557:role/LabRole'
11 FORMAT CSV
12 IGNOREHEADER 1;
```

Result 1

Summary

Info:

- Load into table 'emissions\_by\_region' completed, 30308 record(s) loaded successfully.

Returned rows: 0  
Elapsed time: 139ms  
Result set query:

Carga de los datos para el data set de emisiones per cápita:

```
Run Limit 100 Explain Isolated session redshift-clust... dev Schedule
9 COPY emissions_per_capita FROM 's3://trabajoidatalake/transient/ourworldindata_org/co-emissions-per-capita/co-emissions-per-capita.csv'
10 iam_role 'arn:aws:iam::851725398557:role/LabRole'
11 FORMAT CSV
12 IGNOREHEADER 1;
```

Result 1

Summary

Info:

- Load into table 'emissions\_per\_capita' completed, 26600 record(s) loaded successfully.

Returned rows: 0  
Elapsed time: 483ms  
Result set query:

Consultas de las tablas almacenadas directamente en RedShift.

Teniendo en cuenta las emisiones por región y per cápita, se realiza un análisis por cada una de las regiones y países agrupándolos con el promedio de cada una de estas emisiones.

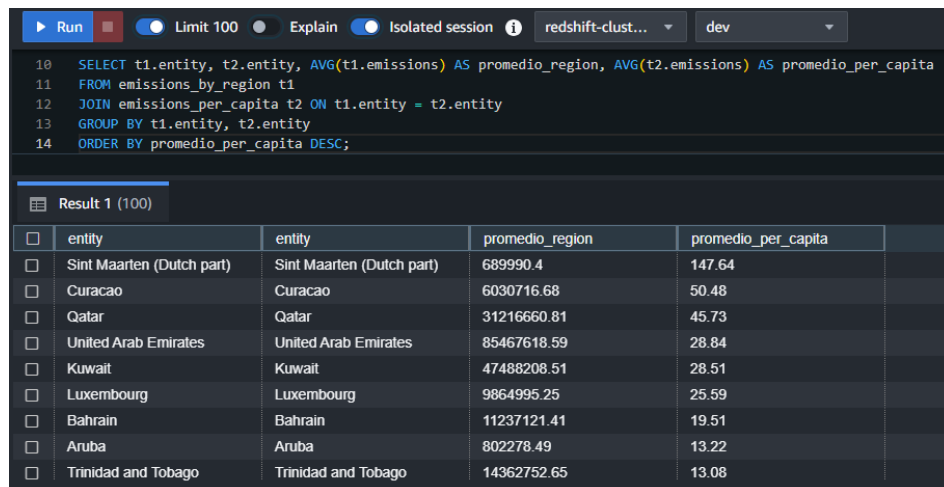
Regiones y países ordenados de mayor a menor por el promedio de cada una de sus emisiones en su historial:

```
Run Limit 100 Explain Isolated session redshift-clust... dev
10 SELECT t1.entity, t2.entity, AVG(t1.emissions) AS promedio_region, AVG(t2.emissions) AS promedio_per_capita
11 FROM emissions_by_region t1
12 JOIN emissions_per_capita t2 ON t1.entity = t2.entity
13 GROUP BY t1.entity, t2.entity
14 ORDER BY promedio_region DESC;
```

Result 1 (100)

entity	entity	promedio_region	promedio_per_capita
<input type="checkbox"/> World	World	6494020342.16	1.8
<input type="checkbox"/> High-income countries	High-income countries	3668989580.59	5.24
<input type="checkbox"/> China	China	2246717607.44	1.8
<input type="checkbox"/> Asia	Asia	2118771524.02	0.72
<input type="checkbox"/> Europe	Europe	1987031371.47	3.86
<input type="checkbox"/> Upper-middle-income cou...	Upper-middle-income cou...	1971178063.46	1.21
<input type="checkbox"/> United States	United States	1914415061.21	10.26
<input type="checkbox"/> North America	North America	1795756149.45	7.34
<input type="checkbox"/> European Union (28)	European Union (28)	1372910925.52	4.02

Regiones y países ordenados de mayor a menor por el promedio per cápita de cada una de sus emisiones en su historial:



```
10 SELECT t1.entity, t2.entity, AVG(t1.emissions) AS promedio_region, AVG(t2.emissions) AS promedio_per_capita
11 FROM emissions_by_region t1
12 JOIN emissions_per_capita t2 ON t1.entity = t2.entity
13 GROUP BY t1.entity, t2.entity
14 ORDER BY promedio_per_capita DESC;
```

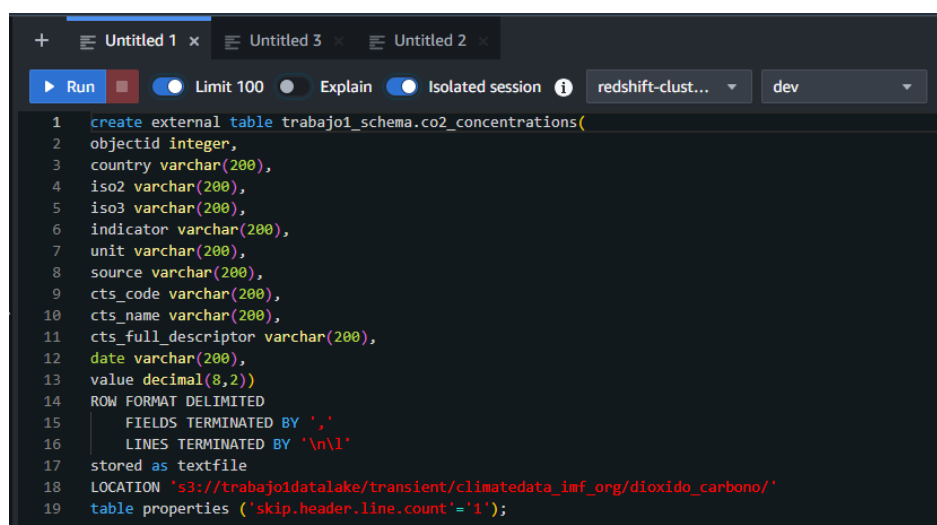
Result 1 (100)

	entity	entity	promedio_region	promedio_per_capita
<input type="checkbox"/>	Sint Maarten (Dutch part)	Sint Maarten (Dutch part)	689990.4	147.64
<input type="checkbox"/>	Curacao	Curacao	6030716.68	50.48
<input type="checkbox"/>	Qatar	Qatar	31216660.81	45.73
<input type="checkbox"/>	United Arab Emirates	United Arab Emirates	85467618.59	28.84
<input type="checkbox"/>	Kuwait	Kuwait	47488208.51	28.51
<input type="checkbox"/>	Luxembourg	Luxembourg	9864995.25	25.59
<input type="checkbox"/>	Bahrain	Bahrain	11237121.41	19.51
<input type="checkbox"/>	Aruba	Aruba	802278.49	13.22
<input type="checkbox"/>	Trinidad and Tobago	Trinidad and Tobago	14362752.65	13.08

Consultas de tablas con datos en S3.

Haciendo uso de **Reshift Spectrum** ejecutamos las consultas SQL en datos almacenados en los archivos de Amazon S3 sin la necesidad de cargar los datos directamente en Redshift.

Se define inicialmente el esquema y la tabla externa en Redshift haciendo referencia a la carpeta en la zona 'Transient' donde se encuentra el archivo de emisiones de CO2 atmosféricas en formato CSV en S3:

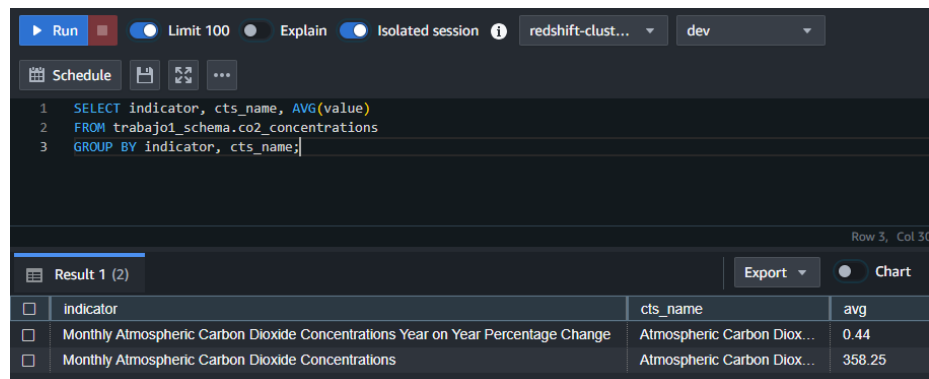


```
1 create external table trabajo1_schema.co2_concentrations(
2   objectid integer,
3   country varchar(200),
4   iso2 varchar(200),
5   iso3 varchar(200),
6   indicator varchar(200),
7   unit varchar(200),
8   source varchar(200),
9   cts_code varchar(200),
10  cts_name varchar(200),
11  cts_full_descriptor varchar(200),
12  date varchar(200),
13  value decimal(8,2))
14 ROW FORMAT DELIMITED
15   FIELDS TERMINATED BY ','
16   LINES TERMINATED BY '\n\\'
17 stored as textfile
18 LOCATION 's3://trabajo1datalake/transient/climatedata_imf_org/dioxido_carbono/'
19 table properties ('skip.header.line.count'='1');
```



Se realiza la consulta a la tabla externa “co2\_concentrations” que hace referencia al archivo almacenado en S3.

En este caso se obtiene el promedio de emisiones de dióxido de carbono en el historial por cada indicador:



The screenshot shows a Redshift SQL query interface. At the top, there are buttons for 'Run', 'Limit 100', 'Explain', and 'Isolated session'. Below these are icons for 'Schedule', 'Save', and 'More'. The SQL query is as follows:

```
1 SELECT indicator, cts_name, AVG(value)
2 FROM trabajo1_schema.co2_concentrations
3 GROUP BY indicator, cts_name;
```

Below the query, the results are displayed in a table. The table has three columns: 'indicator', 'cts\_name', and 'avg'. There are two rows of data.

indicator	cts_name	avg
Monthly Atmospheric Carbon Dioxide Concentrations Year on Year Percentage Change	Atmospheric Carbon Diox...	0.44
Monthly Atmospheric Carbon Dioxide Concentrations	Atmospheric Carbon Diox...	358.25

### 3. Notebooks

#### ETL Zonas Datalake

[https://eafit-](https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa_eafit_edu_co/EadoQY1FC2tHv836X7U80QIBzXuo5SRq6PXz-3DwVe5V4Q?e=il49LL)

[my.sharepoint.com/:u:/g/personal/afrestrepa\\_eafit\\_edu\\_co/EadoQY1FC2tHv836X7U80QIBzXuo5SRq6PXz-3DwVe5V4Q?e=il49LL](https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa_eafit_edu_co/EadoQY1FC2tHv836X7U80QIBzXuo5SRq6PXz-3DwVe5V4Q?e=il49LL)

#### Zona Refined

[https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa\\_eafit\\_edu\\_co/EaylIJOs9rROvhh-9S91RhQBCxA8\\_xtJekyXVxRqxa2syA?e=hwX7j6](https://eafit-my.sharepoint.com/:u:/g/personal/afrestrepa_eafit_edu_co/EaylIJOs9rROvhh-9S91RhQBCxA8_xtJekyXVxRqxa2syA?e=hwX7j6)