**UNIVERSIDAD EAFIT**

**ST0263 TOPICOS ESPECIALES EN TELEMÁTICA**

**2025-2**

**Nombre: \_Sara Zuluaga Trujillo\_\_ - email:\_szuluagat@eafit.edu.co\_**

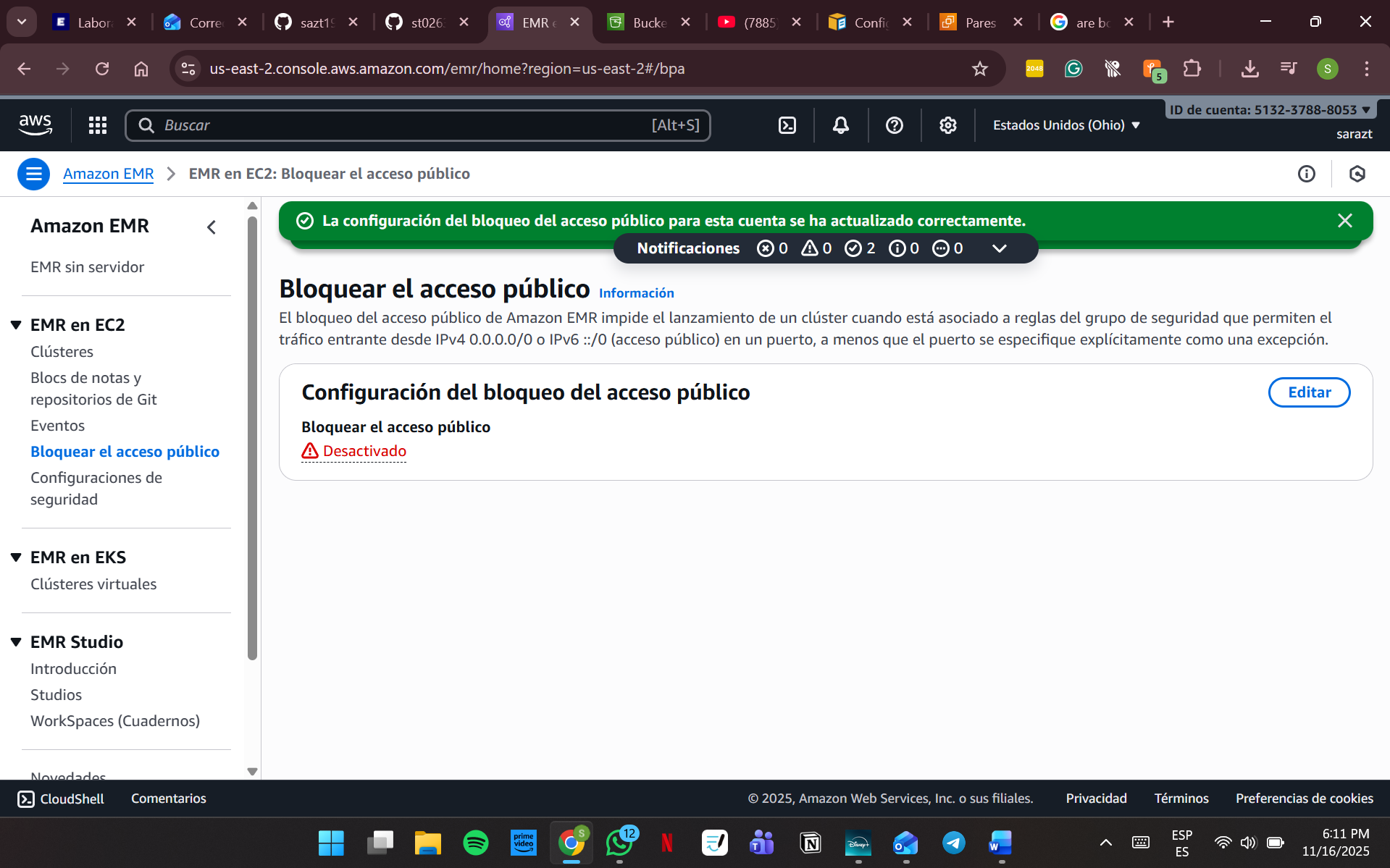
Laboratorios de BIG DATA:

Curso en AWS ACADEMY:

LABORATORIO 3-0: CREACIÓN DE CLUSTER EMR en AWS ACADEMY

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.



Texto

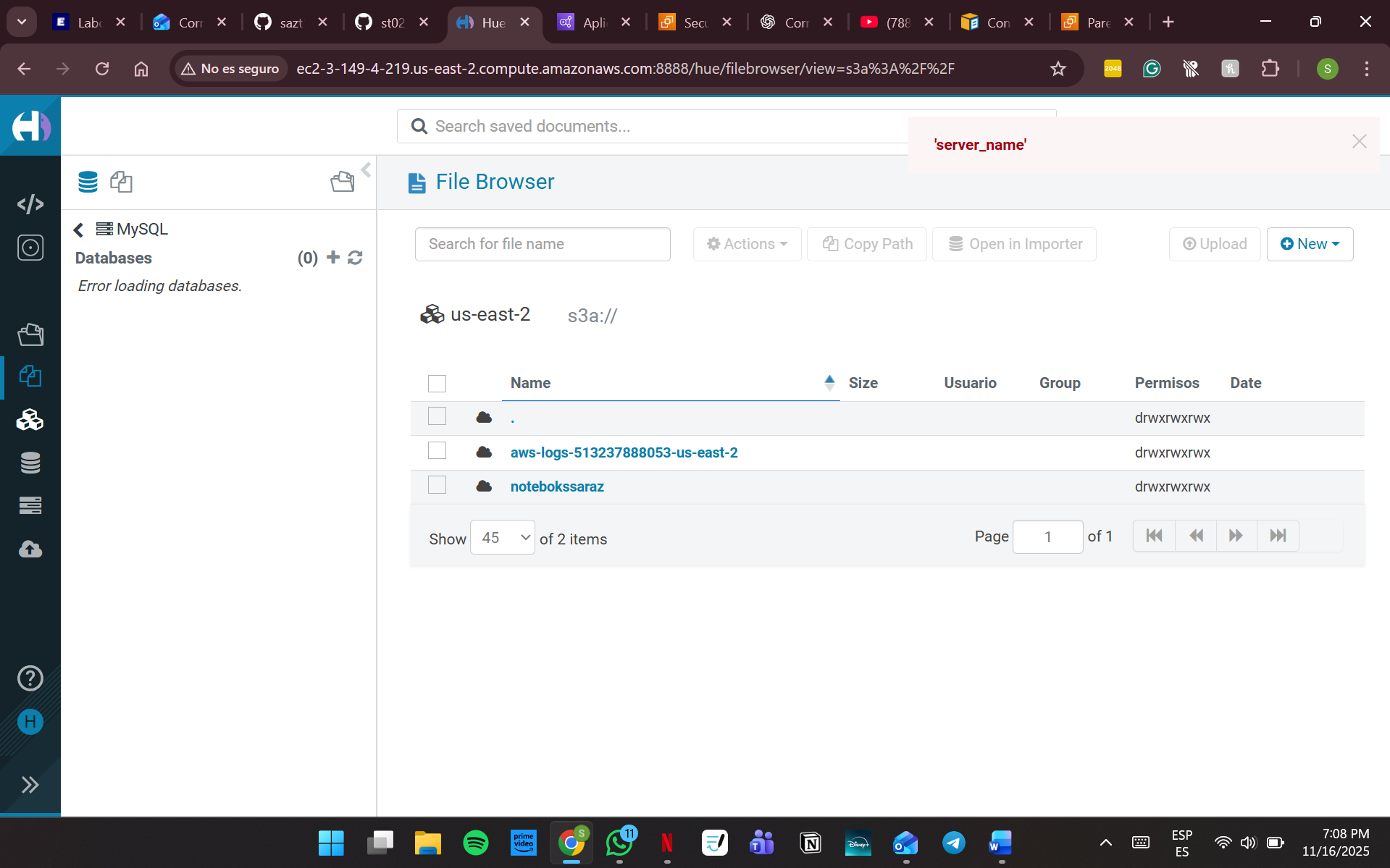
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

LABORATORIO 3-1: GESTIÓN DE ARCHIVOS EN HDFS EN EMR

Texto

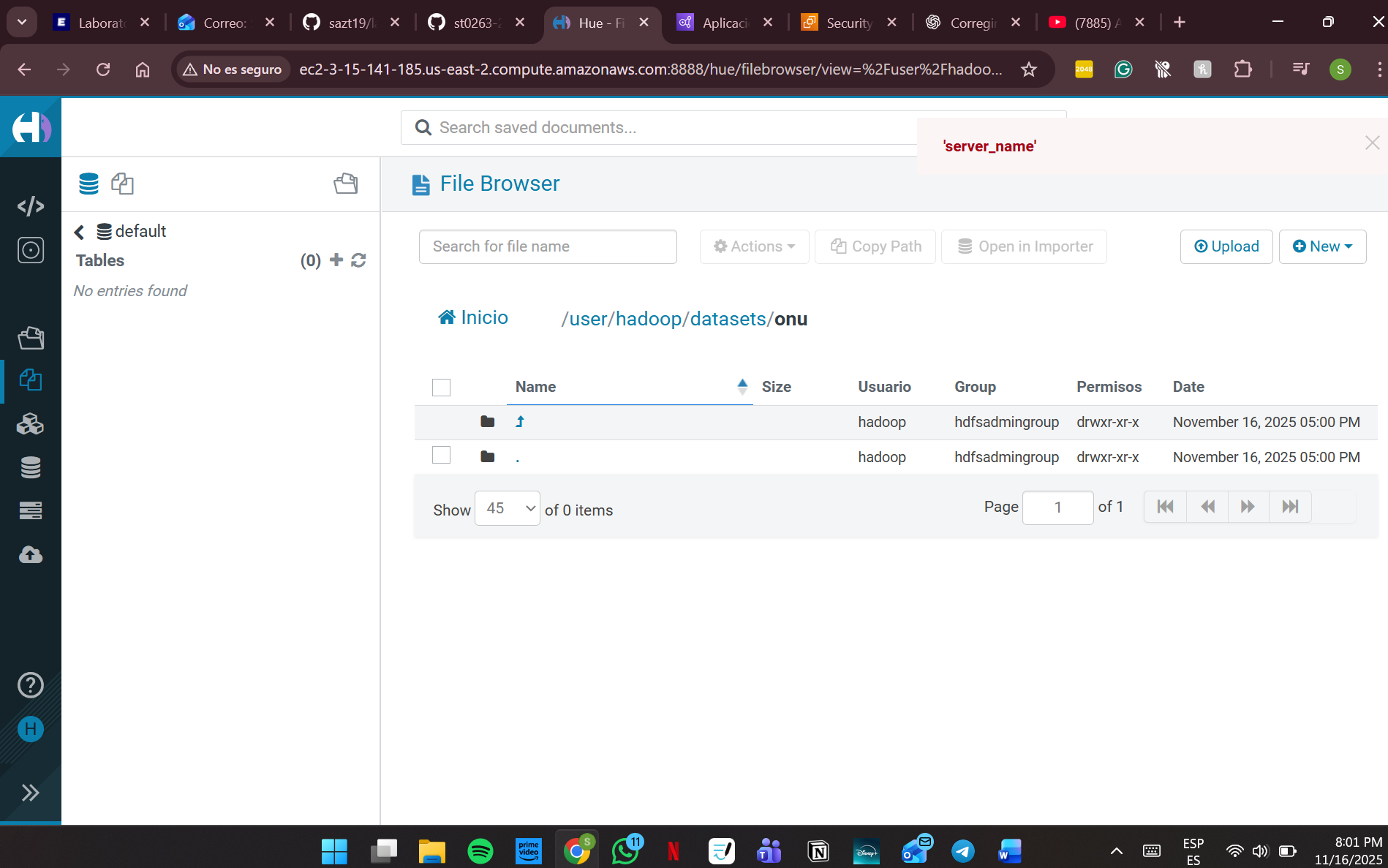
El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.



LABORATORIO 3-2: DATALAKE EN S3, CATALOGACIÓN EN GLUE Y MOTOR DE CONSULTA ATHENA

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

s3://szuluagatlab1/datasets/

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

LABORATORIO 3-3: MOTOR DE CONSULTA SQL EN HADOOP HIVE, INTEGRADO CON S3, GLUE Y ATHENA

* TABLAS HIVE CON ALMACENAMIENTO EN HDFS Y S3
* CONSULTAS DESDE HIVE DE TABLAS GLUE
* CONSULTAS DESDE ATHENA DE TABLAS HIVE

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Reto: se puede hacer de dos maneras:

1. Crear la tabla a partir del query

CREATE TABLE wordcount AS

SELECT word, COUNT(1) AS count

FROM (

SELECT explode(split(line, ' ')) AS word

FROM docs

) w

GROUP BY word;

Hive crea una tabla y la llena automáticamente con el resultado del query.

1. Llenar una table que ya existe:

INSERT OVERWRITE TABLE wordcount

SELECT word, COUNT(1) AS count

FROM (

SELECT explode(split(line, ' ')) AS word

FROM docs

) w

GROUP BY word;

Cuando ya se tiene la tabla y solo se requiere llenarla o actualizarla..

LABORATORIO 3-4: APACHE SPARK DESDE EMR/JUPYTERHUB Y GOOGLE COLAB

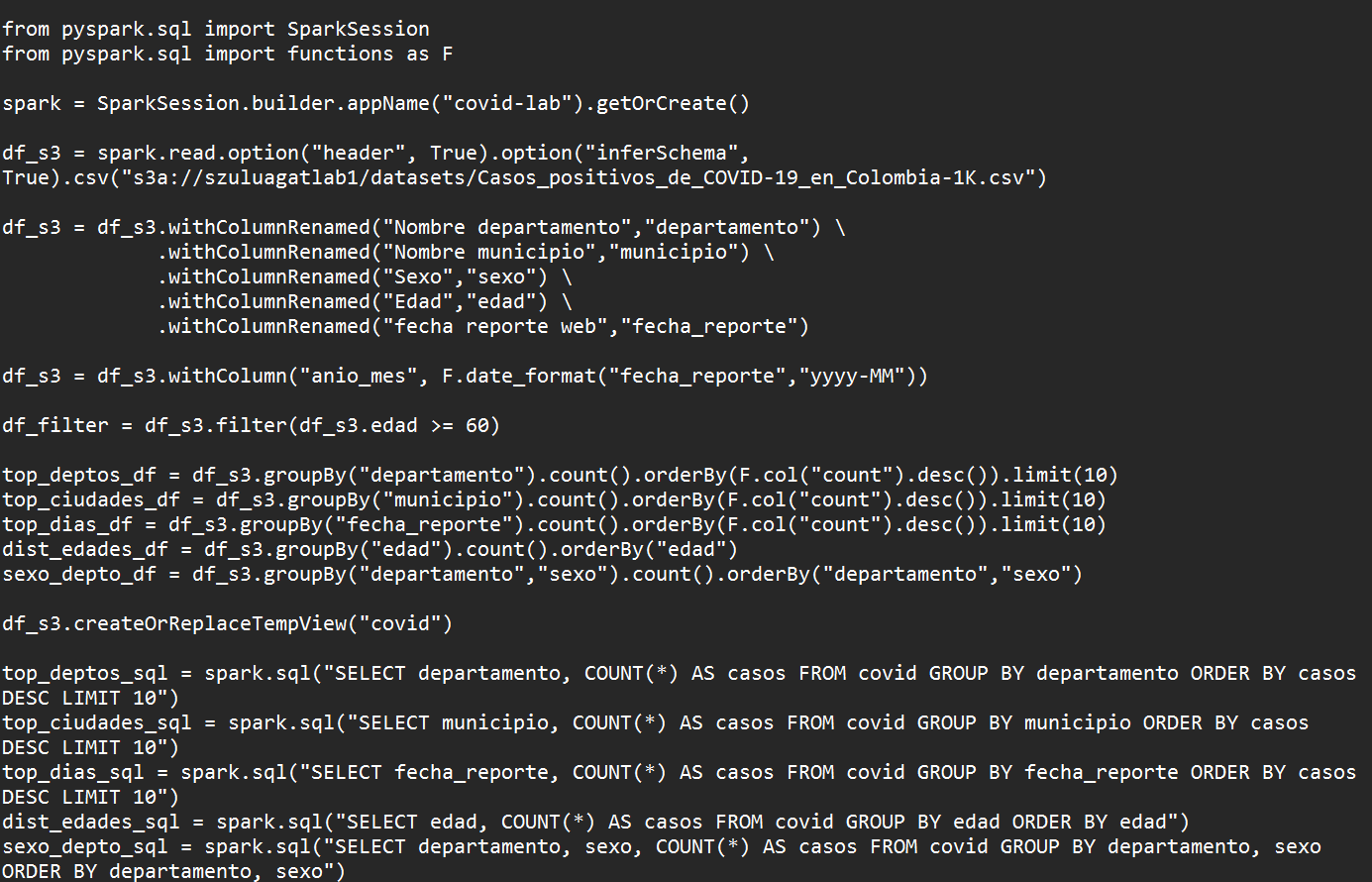


Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

5 preguntas:

from pyspark.sql import SparkSession  
from pyspark.sql import functions as F

spark = SparkSession.builder.appName("covid-lab").getOrCreate()

df = (  
spark.read  
.option("header", True)  
.option("inferSchema", True)  
.csv("s3a://szuluagatlab1/datasets/Casos\_positivos\_de\_COVID-19\_en\_Colombia-1K.csv")  
)

df = df.withColumnRenamed("Nombre departamento","departamento")   
.withColumnRenamed("Nombre municipio","municipio")   
.withColumnRenamed("Sexo","sexo")   
.withColumnRenamed("Edad","edad")   
.withColumnRenamed("fecha reporte web","fecha\_reporte")

df = df.withColumn("fecha\_reporte", F.to\_date("fecha\_reporte"))

**Con dataframes:**

1. 10 departamentos con más casos

top\_deptos\_df = (  
df.groupBy("departamento")  
.count()  
.orderBy(F.col("count").desc())  
.limit(10)  
)

top\_deptos\_df.show()

1. 10 ciudades (municipios) con más casos

top\_ciudades\_df = (  
df.groupBy("municipio")  
.count()  
.orderBy(F.col("count").desc())  
.limit(10)  
)

top\_ciudades\_df.show()

1. 10 días con más casos

top\_dias\_df = (  
df.groupBy("fecha\_reporte")  
.count()  
.orderBy(F.col("count").desc())  
.limit(10)  
)

top\_dias\_df.show()

1. Distribución de casos por edades

dist\_edades\_df = (  
df.groupBy("edad")  
.count()  
.orderBy("edad")  
)

dist\_edades\_df.show()

1. Pregunta de negocio que tú eliges  
   (ejemplo: casos por sexo y departamento)

sexo\_depto\_df = (  
df.groupBy("departamento", "sexo")  
.count()  
.orderBy("departamento", "sexo")  
)

sexo\_depto\_df.show()

**Con spark sql:**

1. 10 departamentos con más casos

top\_deptos\_sql = spark.sql("""  
SELECT departamento, COUNT(\*) AS casos  
FROM covid  
GROUP BY departamento  
ORDER BY casos DESC  
LIMIT 10  
""")

top\_deptos\_sql.show()

1. 10 ciudades con más casos

top\_ciudades\_sql = spark.sql("""  
SELECT municipio, COUNT(\*) AS casos  
FROM covid  
GROUP BY municipio  
ORDER BY casos DESC  
LIMIT 10  
""")

top\_ciudades\_sql.show()

1. 10 días con más casos

top\_dias\_sql = spark.sql("""  
SELECT fecha\_reporte, COUNT(\*) AS casos  
FROM covid  
GROUP BY fecha\_reporte  
ORDER BY casos DESC  
LIMIT 10  
""")

top\_dias\_sql.show()

1. Distribución de casos por edades

dist\_edades\_sql = spark.sql("""  
SELECT edad, COUNT(\*) AS casos  
FROM covid  
GROUP BY edad  
ORDER BY edad  
""")

dist\_edades\_sql.show()

1. Misma pregunta de negocio (sexo x departamento)

sexo\_depto\_sql = spark.sql("""  
SELECT departamento, sexo, COUNT(\*) AS casos  
FROM covid  
GROUP BY departamento, sexo  
ORDER BY departamento, sexo  
""")

sexo\_depto\_sql.show()