**Crime Data Analysis**

Dataset: crime\_incidents\_2013.csv

hdfs dfs -put crime\_incidents\_2013.csv

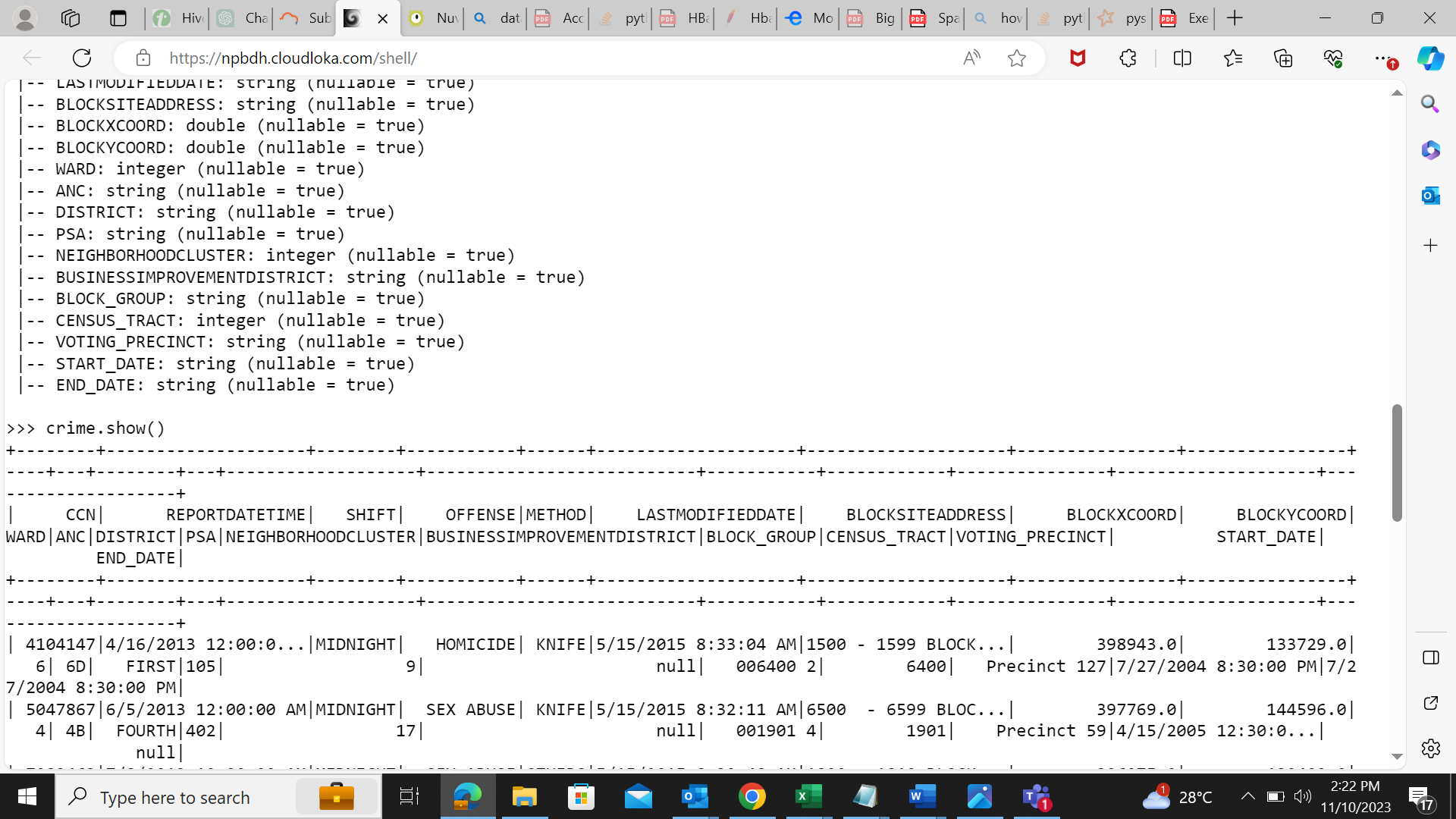
hdfs dfs -ls

crime=spark.read.option("header",True).option("inferschema",True).csv("crime\_incidents\_2013.csv")

crime.createOrReplaceTempView("crime1")

crime.printSchema()

crime.show()



1. How many offenses were there in 2013?

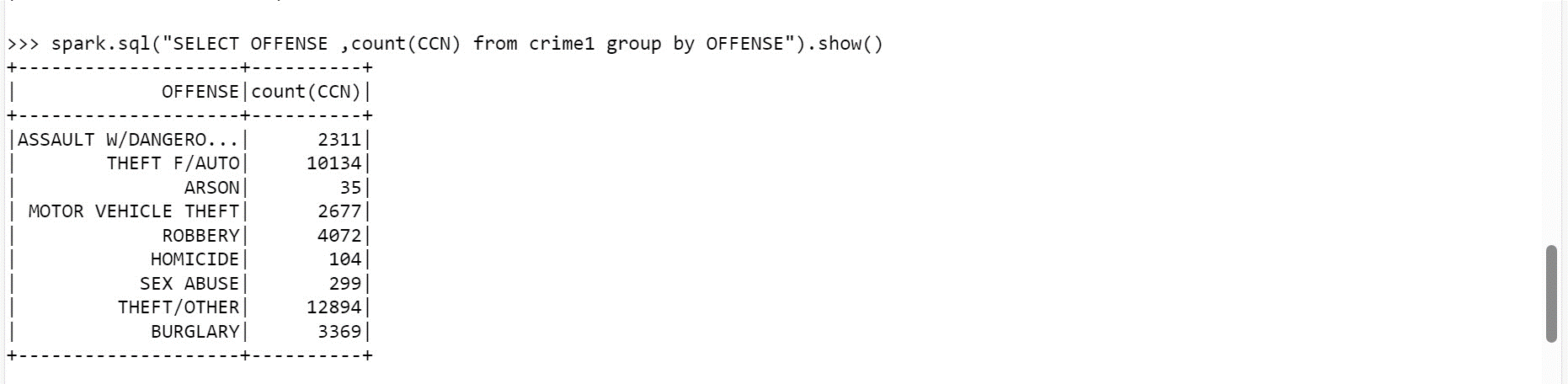
spark.sql("SELECT count(distinct OFFENSE) from crime1").show()

A close-up of a white background

Description automatically generated

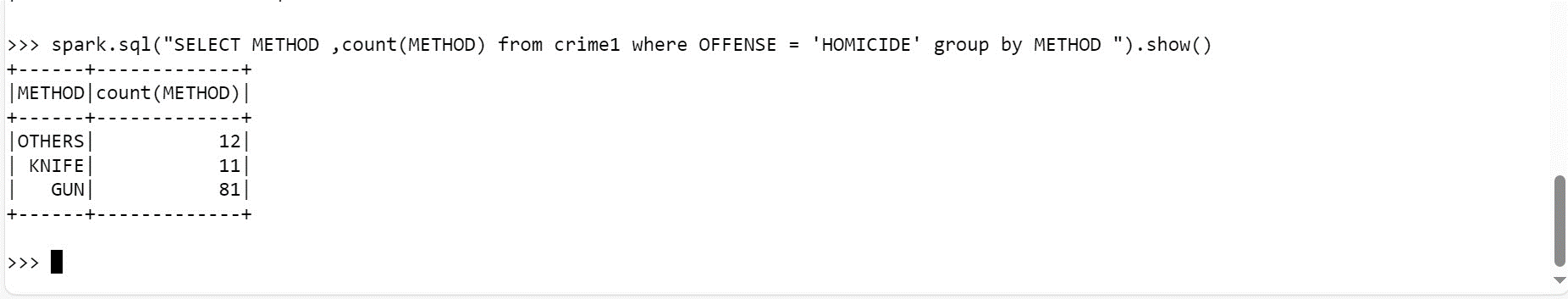
1. . How many crimes were committed in each offense?

spark.sql("SELECT OFFENSE ,count(CCN) from crime1 group by OFFENSE").show()



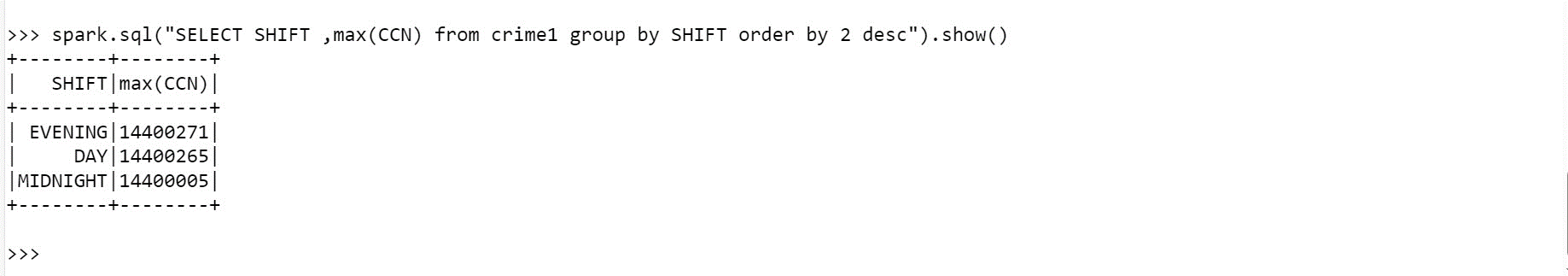
1. How many different methods were used (And their count) in offense “Homicide”

spark.sql("SELECT METHOD ,count(METHOD) from crime1 where OFFENSE = 'HOMICIDE' group by METHOD ").show()



1. Which shift had the maximum crimes?

spark.sql("SELECT SHIFT ,max(CCN) from crime1 group by SHIFT order by 2 desc").show()



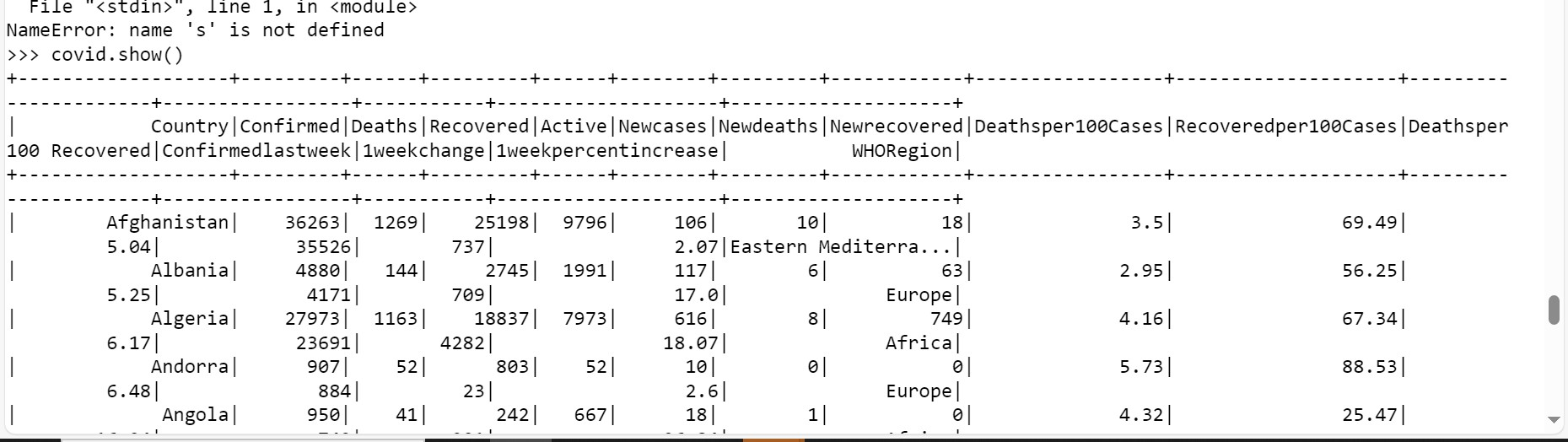
**Covid 19 Data analysis**

1. 19 Data analysis 1. Load the covid 19 data in to a spark dataframe (country\_wise\_latest.csv) with the correct schema definition

covid=spark.read.option("header",True).option("inferschema",True).csv("country\_wise\_latest.csv")

covid.printSchema()

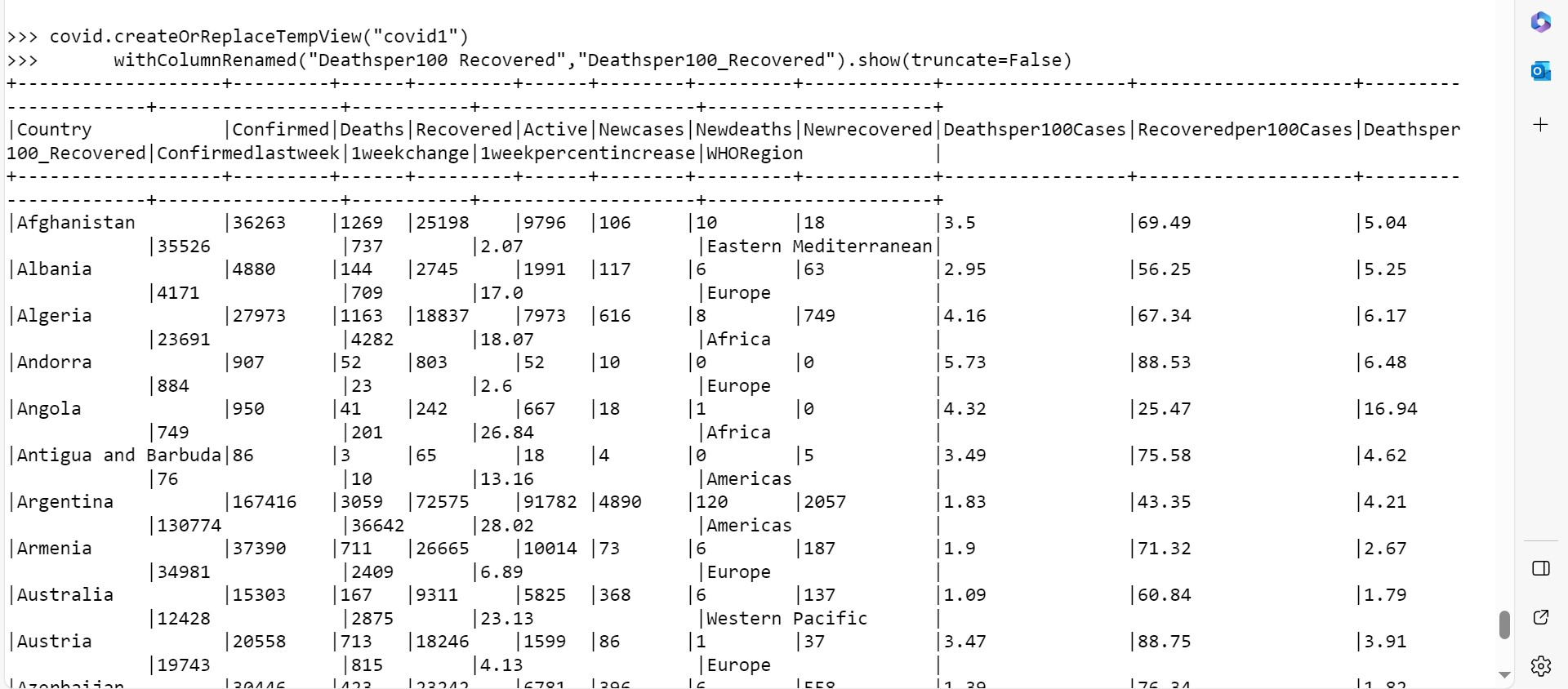
covid.show()



1. **The are some column names which are long, contains special characters, spaces etc. Rename all such column names accordingly. Example Country/Region → country New cases → New\_cases etc**

covid.createOrReplaceTempView("covid1")

covi.withColumnRenamed("Deathsper100 Recovered","Deathsper100\_Recovered").show(truncate=False)



1. **Count and check if there any null values in any of the columns**

import pyspark.sql.functions as F

covid\_agg = covid.agg(\*[F.count(F.when(F.isnull(c), c)).alias(c) for c in covid.columns])

covid\_agg.show()

A black and white text

Description automatically generated

1. **What are the top 10 countries under the WHO region with covid 19 Confirmed cases**

spark.sql( "SELECT Country,WHORegion,sum(Confirmed) FROM covid1 GROUP BY Country, WHORegion order by 3 desc").show(10)

A close-up of a computer screen

Description automatically generated

1. **What are the bottom 10 countries under the WHO region with covid 19 Confirmed cases**

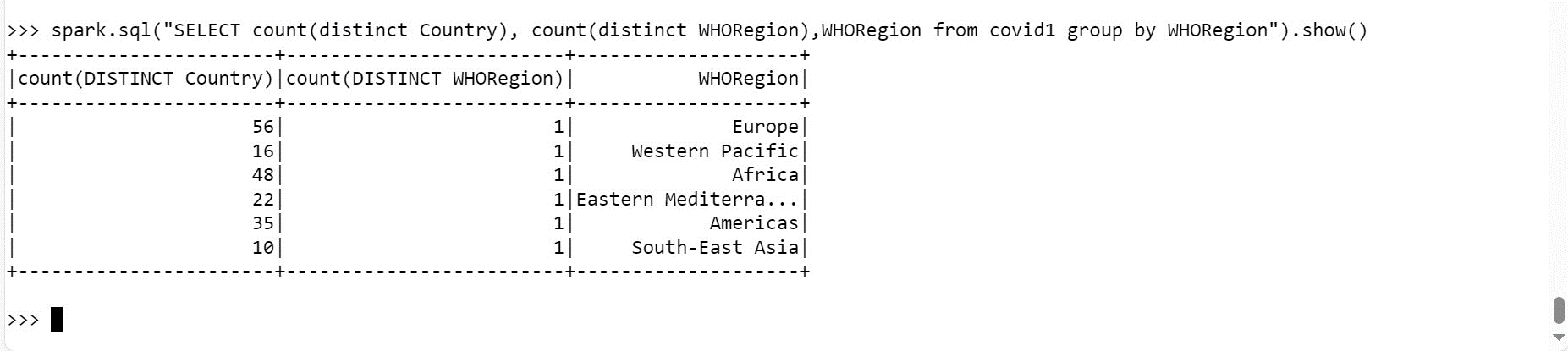
spark.sql( "SELECT Country,WHORegion,sum(Confirmed) FROM covid1 GROUP BY Country, WHORegion order by 3 asc").show(10)

A screenshot of a computer

Description automatically generated

1. What are the total number of countries/ total no. of WHO regions and also list the various WHO regions

spark.sql("SELECT count(distinct Country), count(distinct WHORegion),WHORegion from covid1 group by WHORegion").show()



**Order Data analysis (Joins)**

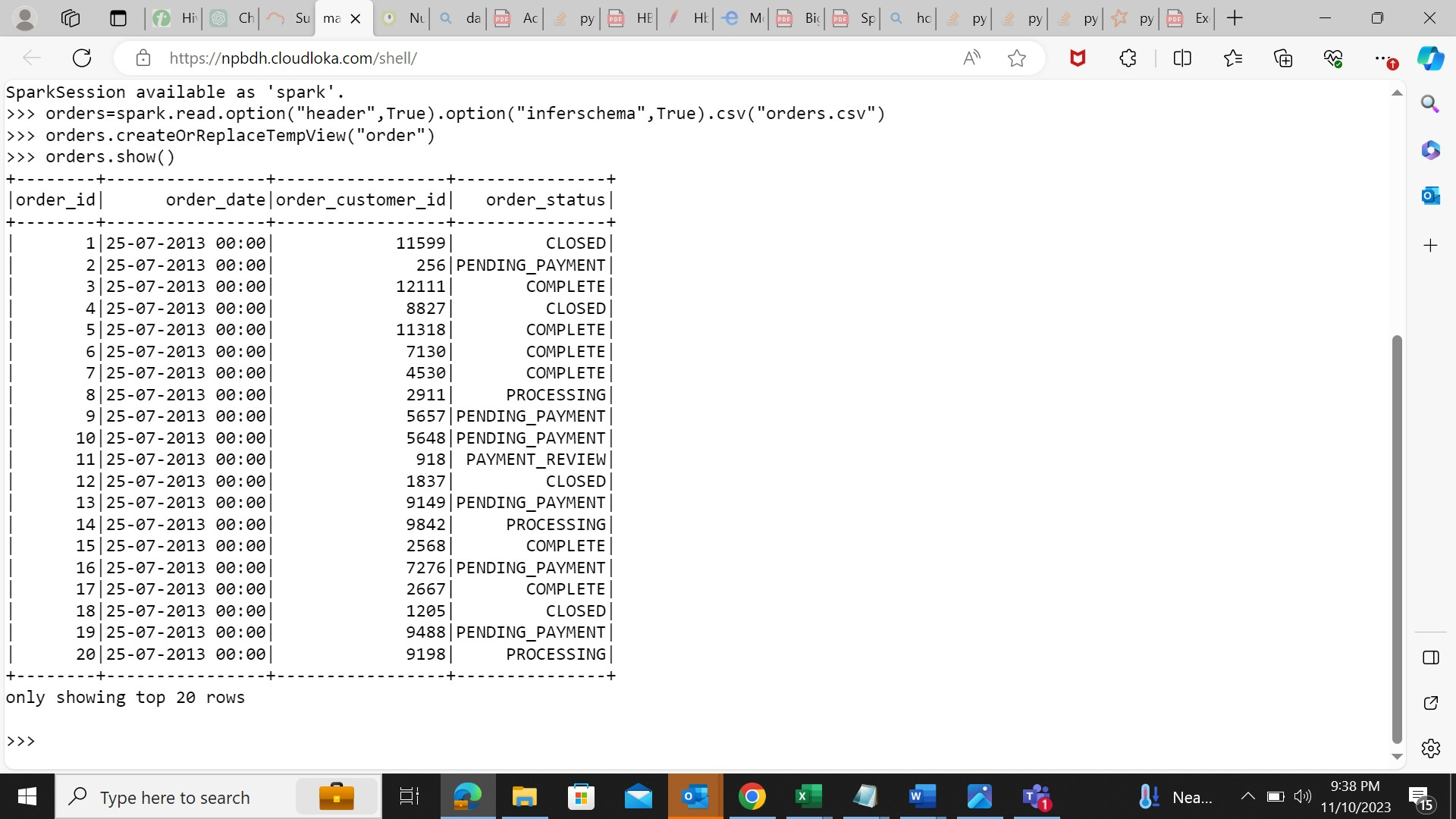
1. . **Load the required data in to DF like categories, customer,departments,order\_items,orders and products**

**orders**

orders=spark.read.option("header",True).option("inferschema",True).csv("orders.csv")

orders.createOrReplaceTempView("order")

orders.show()



**order\_items**

order\_items=spark.read.option("header",True).option("inferschema",True).csv("order\_items.csv")

order\_items.createOrReplaceTempView("order1")

order\_items.show()

A screenshot of a computer

Description automatically generated

**products**

products=spark.read.option("header",True).option("inferschema",True).csv("products.csv")

products.createOrReplaceTempView("product1")

products.show()

A screenshot of a computer

Description automatically generated

1. **Get the count for each order status**

spark.sql("SELECT order\_status, count(order\_id) from order group by order\_status").show()

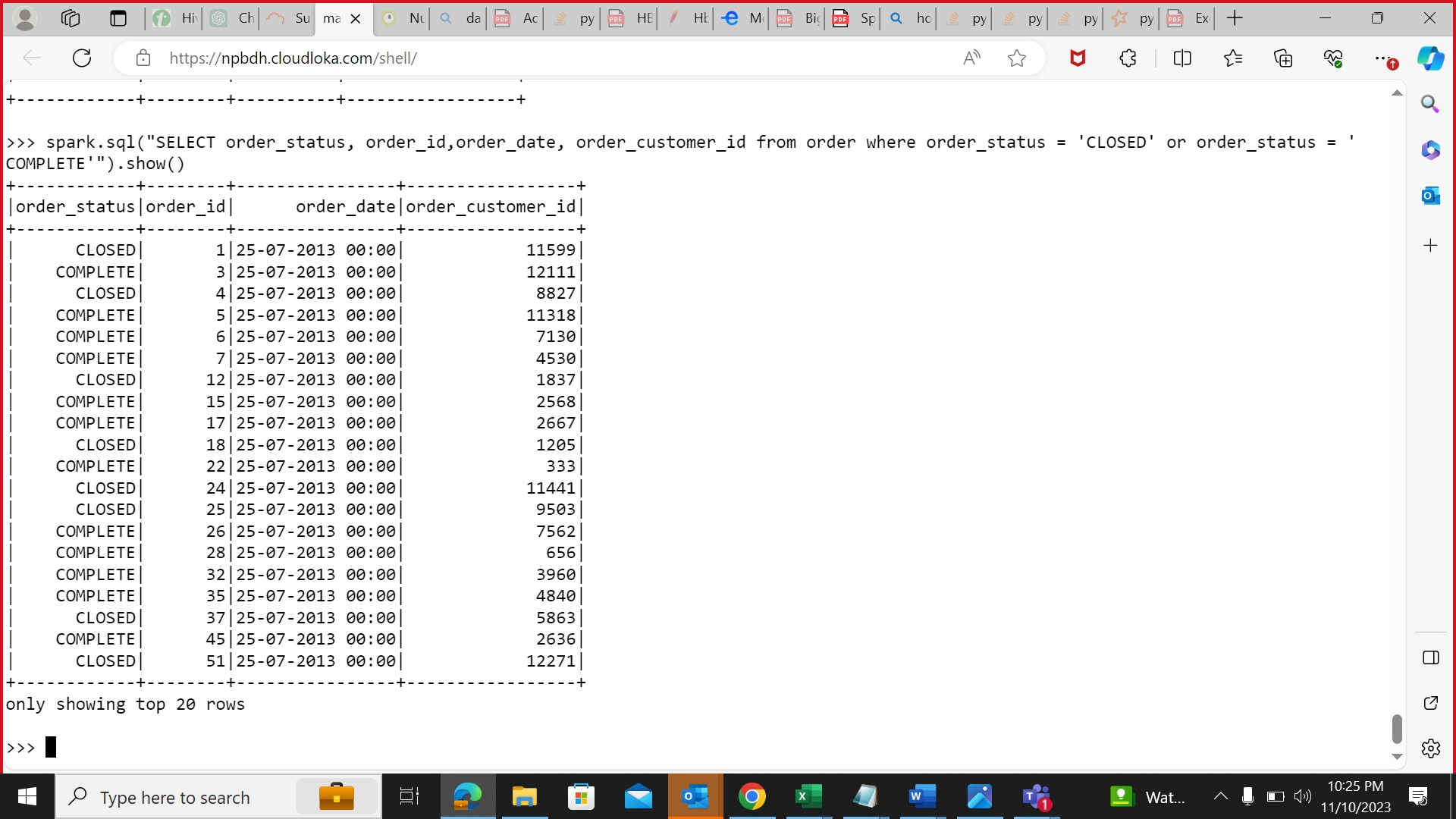
A screenshot of a computer

Description automatically generated

1. **Filter only COMPLETE or CLOSED orders**

spark.sql("SELECT order\_status, order\_id,order\_date, order\_customer\_id from order where order\_status = 'CLOSED' or order\_status = '

COMPLETE'").show()



1. **Problem Statement: What is the daily product revenue for CLOSED or COMPLETE orders?**

spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_id =

ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group b

y o.order\_date, p.product\_name").show()

A screenshot of a computer

Description automatically generated

1. **Join the products , order\_items and orders tables and calculate daily product revenue**

spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_id =

ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id group by o.order\_date, p.product\_name").show()

A screenshot of a computer

Description automatically generated

1. **Write the data in to the table Daily product revenue in Hive**

from pyspark.sql import SparkSession

spark=SparkSession.builder.appName("Spark Hive").enableHiveSupport().config("spark.sql.warehouse.dir","/user/hive/warehouse").getOr

Create()

spark.sparkContext.setLogLevel("ERROR")

df = spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_

id = ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id group by o.order\_date, p.product\_name")

df.show()

df.write.partitionBy("product\_name").mode("overwrite").saveAsTable("airline\_db.daily\_revenue")

A screenshot of a computer

Description automatically generated