**Covid 19 Data analysis**

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

**Solution:**

# Create SparkSession

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("DemoApp").master("local").getOrCreate()

df = spark.read.csv("dbfs:/FileStore/tables/country\_wise\_latest.csv", header=True, inferSchema=True)

df.display()

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.**

**Solution:**

from pyspark.sql.functions import col

new\_column\_names = [col(c).alias(c.replace("/", "\_").replace(" ", "\_").replace(".", "").replace("-", "\_")) for c in df.columns]

df = df.select(\*new\_column\_names)

df.display()

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

**Solution:**

from pyspark.sql.functions import col

null\_counts = []

for col\_name in df.columns:

null\_count = df.filter(col(col\_name).isNull()).count()

null\_counts.append((col\_name, null\_count))

# Display the counts

for col\_name, count in null\_counts:

print(f"Column '{col\_name}' has {count} null values")

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

**Solution:**

from pyspark.sql.functions import desc

top\_10\_countries = df.orderBy(desc("Confirmed")).limit(10)

top\_10\_countries.show()

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

**Solution:**

from pyspark.sql import functions as F

from pyspark.sql.window import Window

region\_filter = (F.col('WHO\_Region') == 'YourRegion')

filtered\_data = df.filter(region\_filter)

windowSpec = Window.orderBy(F.col('Confirmed').asc())

ranked\_data = filtered\_data.withColumn('rank', F.rank().over(windowSpec))

bottom\_10\_countries = ranked\_data.filter(F.col('rank') <= 10)

bottom\_10\_countries.show()

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

**Solution:**

total\_countries = df.select("Country\_Region").distinct().count()

total\_who\_regions = df.select("WHO\_Region").distinct().count()

who\_regions\_list = df.select("WHO\_Region").distinct().rdd.map(lambda x: x[0]).collect()

print(f"Total number of countries: {total\_countries}")

print(f"Total number of WHO regions: {total\_who\_regions}")

print(f"List of WHO regions: {who\_regions\_list}")