1. question

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

from pyspark.sql.window import Window

spark=SparkSession.builder.appName("Testing").getOrCreate()

df=spark.read.format("CSV").option("header","true").option("delimiter",",").option("inferschema","true").load("E:\Bigdata\JK.csv")

df.show()

result = df.withColumn("lead", lead("director\_id").over(Window.orderBy("TIMESTAMP"))).withColumn("lag", lag("director\_id").over(Window.orderBy("TIMESTAMP"))).filter((col("DIRECTOR\_ID") == col("lead")) & (col("DIRECTOR\_ID") == col("lag")) & (col("lead") == col("lag")))

# Show the result

result.show()

-------------------------------------------------------------

3)

from pyspark.sql import SparkSession

spark=SparkSession.builder.appName("Testing").getOrCreate()

df\_person=spark.read.format("CSV").option("header","true").option("delimiter","|").option("inferschema","true").load("D:/Persons.csv")

#df\_person.show()

df\_Address=spark.read.format("CSV").option("header","true").option("delimiter","|").option("inferschema","true").load("D:/Addresses.csv")#.show()

df\_person.createOrReplaceTempView("persons")

df\_Address.createOrReplaceTempView("addresses")

dfjoin = spark.sql("""

SELECT p.PERSONID as person\_id, p.LASTNAME, p.FIRSTNAME, a.ADDRESSID, a.CITY, a.STATE

FROM persons p

LEFT JOIN addresses a

ON p.PERSONID = a.PERSONID

ORDER BY ADDRESSID DESC

""")

dfjoin.show()

+---------+--------+---------+---------+-------------+-------+

|person\_id|LASTNAME|FIRSTNAME|ADDRESSID| CITY| STATE|

+---------+--------+---------+---------+-------------+-------+

| 2| Alice| Bob| 1|New York City|Newyork|

| 1| Wang| Allen| null| null| null|

+---------+--------+---------+---------+-------------+-------+

--------------------------------------------------------------------------

4)Write a PySpark program to find employees

earning more than their managers.

from pyspark.sql import SparkSession

spark=SparkSession.builder.appName("Testing").getOrCreate()

df\_empmgr=spark.read.format("csv").option("header","true").option("delimiter","|").option("inferschema","true").load("D:/EMPMGR.txt").show()

df\_empmgr.show()

+---+-----+------+---------+

| ID| NAME|SALARY|MANAGERID|

+---+-----+------+---------+

| 1| JOE| 70000| 3|

| 2|HENRY| 80000| 4|

| 3| SAM| 60000| null|

| 4| MAX| 90000| null|

+---+-----+------+---------+

from pyspark.sql import SparkSession

# Read the CSV file

df\_empmgr = spark.read.format("csv") \

.option("header", "true") \

.option("delimiter", "|") \

.option("inferschema", "true") \

.load("D:/EMPMGR.txt")

# Show the DataFrame

df\_empmgr.show()

# Register the DataFrame as a temporary view

df\_empmgr.createOrReplaceTempView("EMPMGRR")

# Perform SQL join

df\_em = spark.sql("""

SELECT e.name

FROM EMPMGRR e

JOIN EMPMGRR m

ON e.MANAGERID = m.ID

WHERE e.SALARY>=m.SALARY

""")

# Show the joined DataFrame

df\_em.show()

by DF API

from pyspark.sql import SparkSession

from pyspark.sql.functions import col

# Create a Spark session

spark = SparkSession.builder.appName("EmpMgr").getOrCreate()

df=spark.read.format("csv").option("header","true").option("delimiter","|").option("inferschema","true").load("D:/EMPMGR.txt")

# Load your EMPMGR DataFrame (assuming it's already created as 'df')

# Self-join to find employees earning more than their managers

result\_df = df.alias('emp')\

.join(df.alias('mgr'), col('emp.MANAGERID') == col('mgr.ID'))\

.where(col('emp.SALARY') > col('mgr.SALARY'))\

.select(col('emp.NAME').alias('EMPLOYEE'), col('emp.SALARY').alias('EMPLOYEE\_SALARY'),

col('mgr.NAME').alias('MANAGER'), col('mgr.SALARY').alias('MANAGER\_SALARY'))

# Show the result

result\_df.show()

+--------+---------------+-------+--------------+

|EMPLOYEE|EMPLOYEE\_SALARY|MANAGER|MANAGER\_SALARY|

+--------+---------------+-------+--------------+

| JOE| 70000| SAM| 60000|

+--------+---------------+-------+--------------+

--------------------------------------------------------------