**CDAC MUMBAI**

**PG-DBDA MARCH 2022 BATCH KHARGHAR**

**MODULE: BIG DATA ANALYTICS**

**DATE : 20TH JUNE 2022**

**MARKS : 40 MARKS**

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

[10 marks]

**Q1.**

**MapReduce**

**Problem Statement**

Here, we have chosen the stock market dataset on which we have performed map-reduce

operations. Following is the structure of the data. Kindlyfind the solutions to the questions

below.

Data Structure

1. Exchange Name

2 Stock symbol

3. Transaction date

4. Opening price of the stock

5. Intra day high price of the stock

6. Intra day low price of the stock

7. Closing price of the stock

8. Total Volume of the stock on the particular day

9. Adjustment Closing price of the stock

Field Separator – comma

**Question 2 : Find all time High price for each stock**

[15 marks]

Answer:-

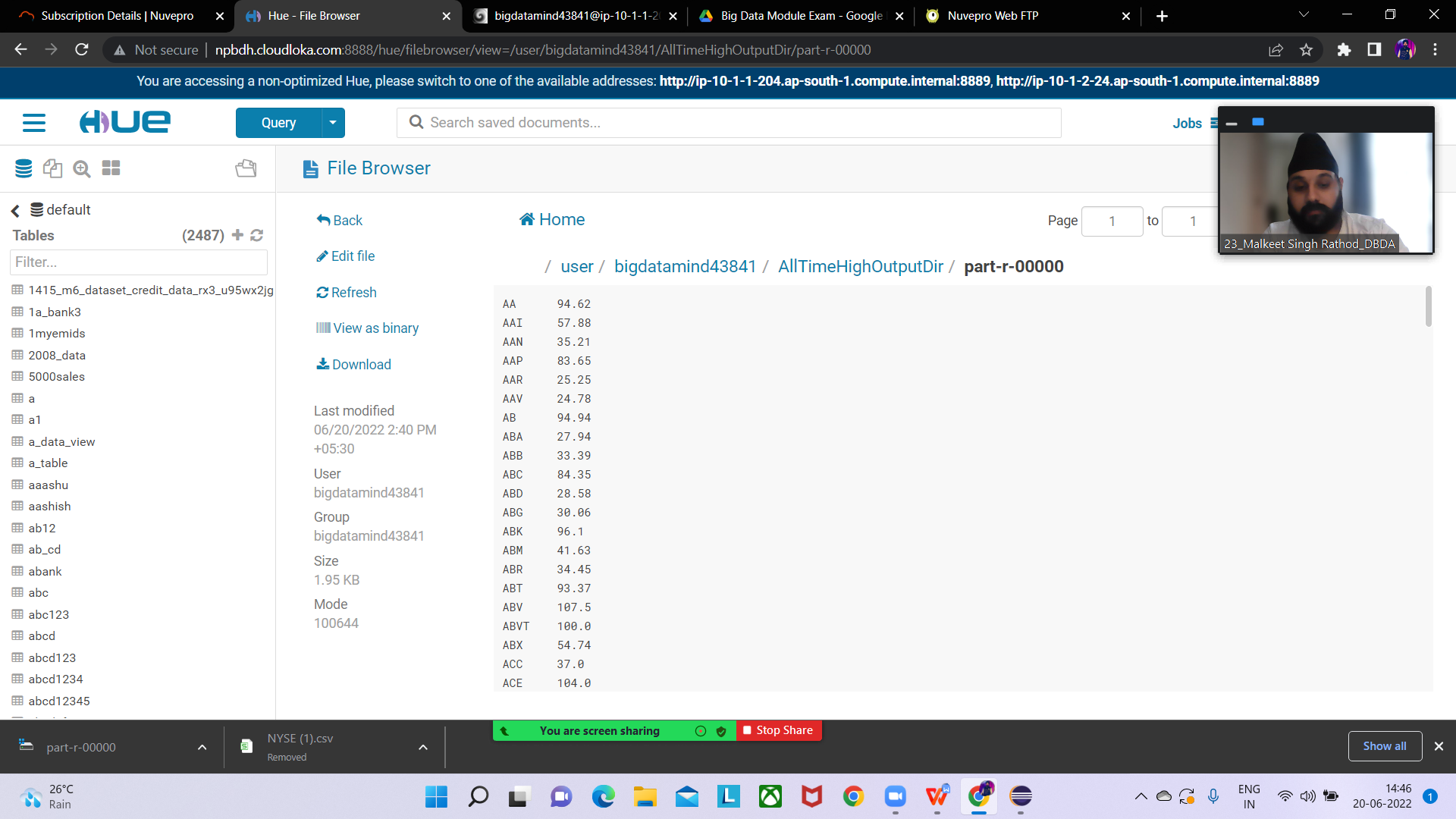
Query:-

hadoop fs -put NYSEdata.csv //uploading NYSE file

jar tvf myjar.jar //checking class in jar

hadoop jar myjar.jar AllTimehigh1 NYSEdata.csv AllTimeHighOutputDir

//using class present in jar to process NYSEdata.csv and storing in AllTimeHighOutputDir



**Hive**

Please find the customer data set.

cust id

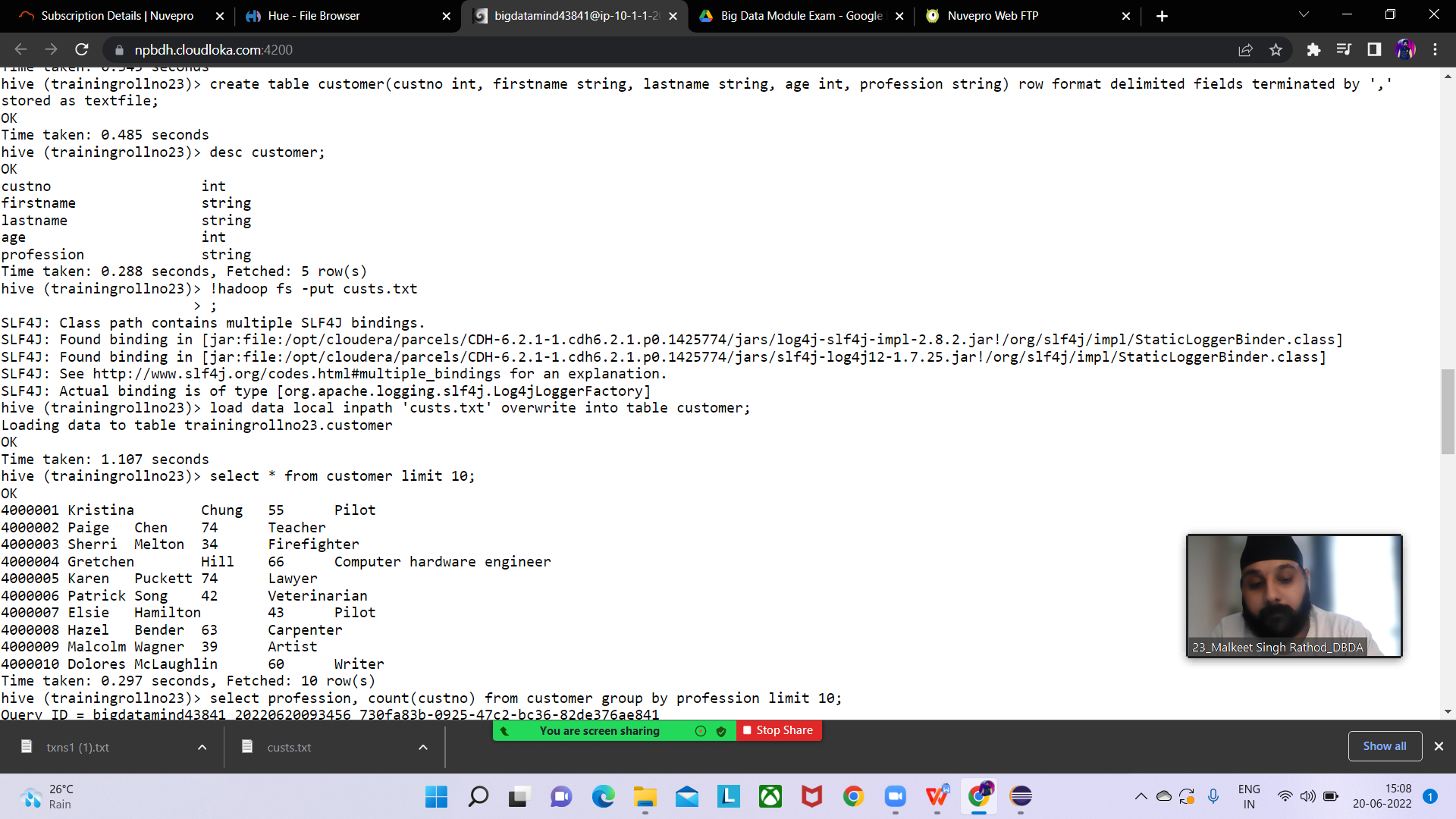
firstname

lastname

age

profession

Answer:-

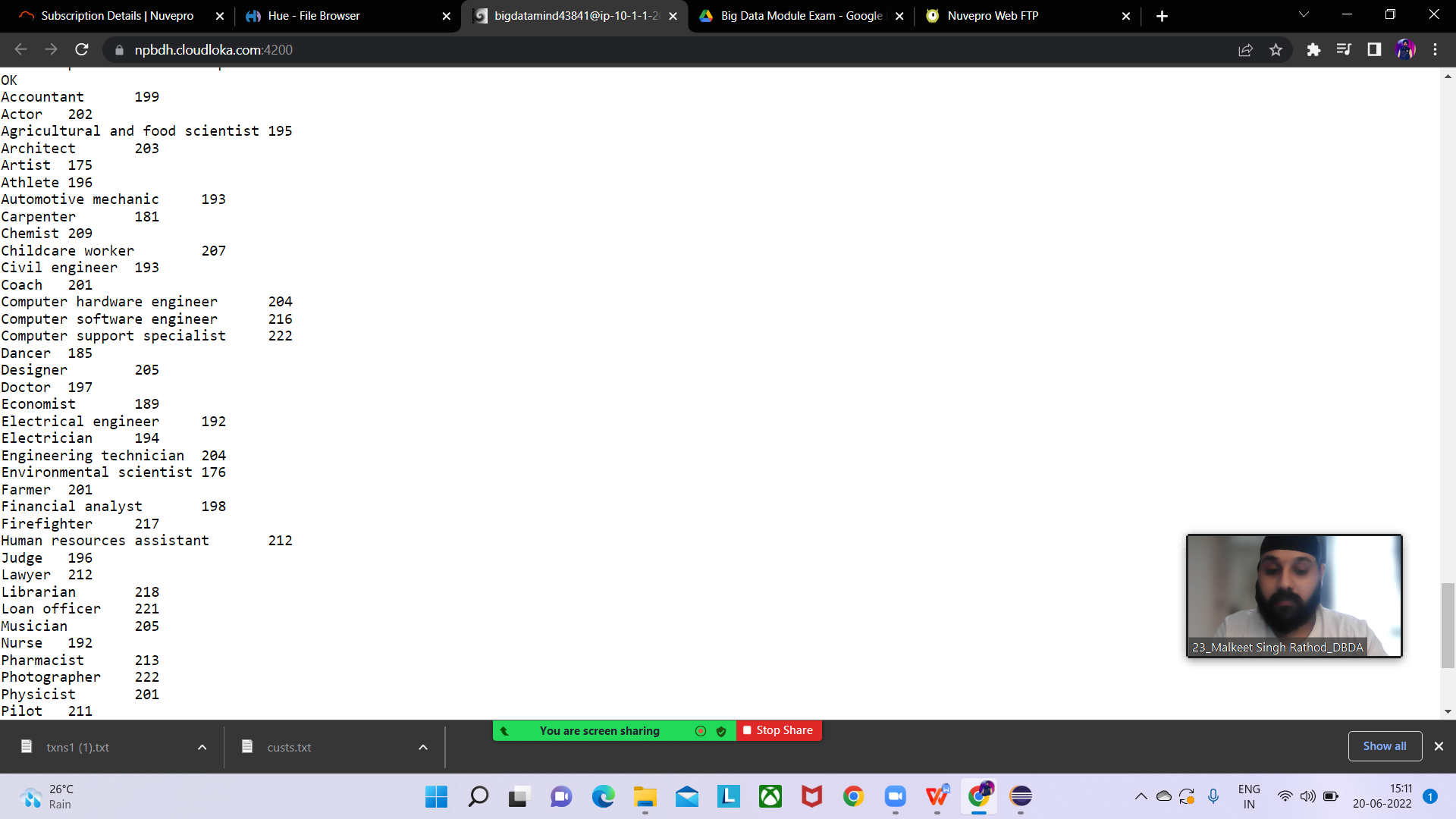


1. **Write a program to find the count of customers for each profession.**

**Query:-**

select profession, count(custno) from customer group by profession ;

Output:-



Please find the sales data set.

txn id

txn date

cust id

amount

category

product

city

state

spendby

Answer:-

**create table txnrecords(txnno int,txndate string,custno int,amount double, category string, product string,city string,state string,spendby str**

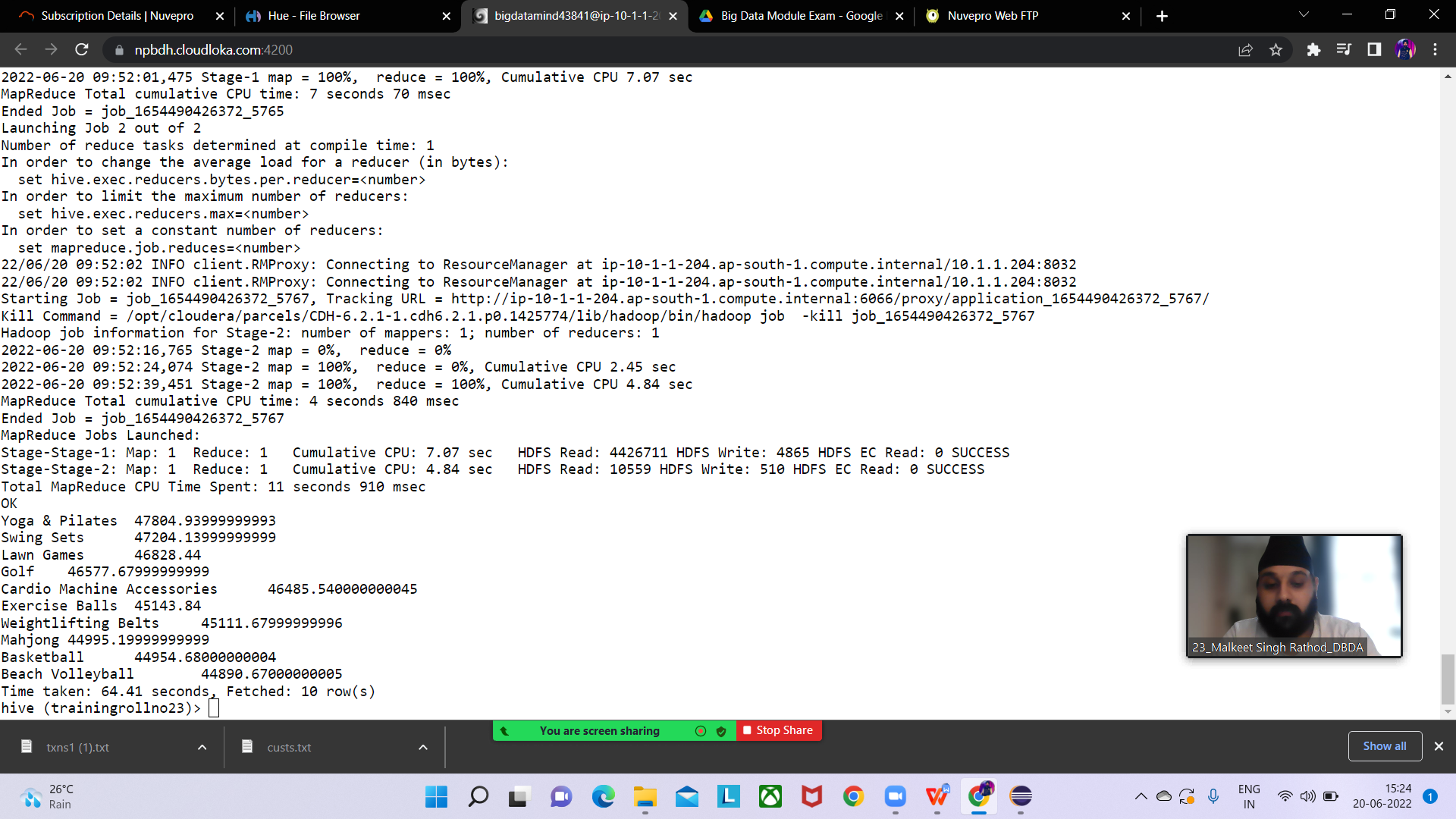
**ing) row format delimited fields terminated by ',' stored as textfile;**

**load data local inpath 'txns.txt' overwrite into table txnrecords;**

1. **Write a program to find the top 10 products sales wise**

Query-

select product,sum(amount) as max\_sales from txnrecords group by product order by max\_sales desc limit 10;



1. **Write a program to create partiioned table on category**

Answer:-

set hive.exec.dynamic.partition.mode=nonstrict;

set hive.exec.dynamic.partition=true;

//Enabling partitioning

create table txnrecordcatpartition(txnno int,txndate string,custno int,amount double,product string,city string,state string,spendby string) pa

rtitioned by (category string) row format delimited fields terminated by ',' stored as textfile;

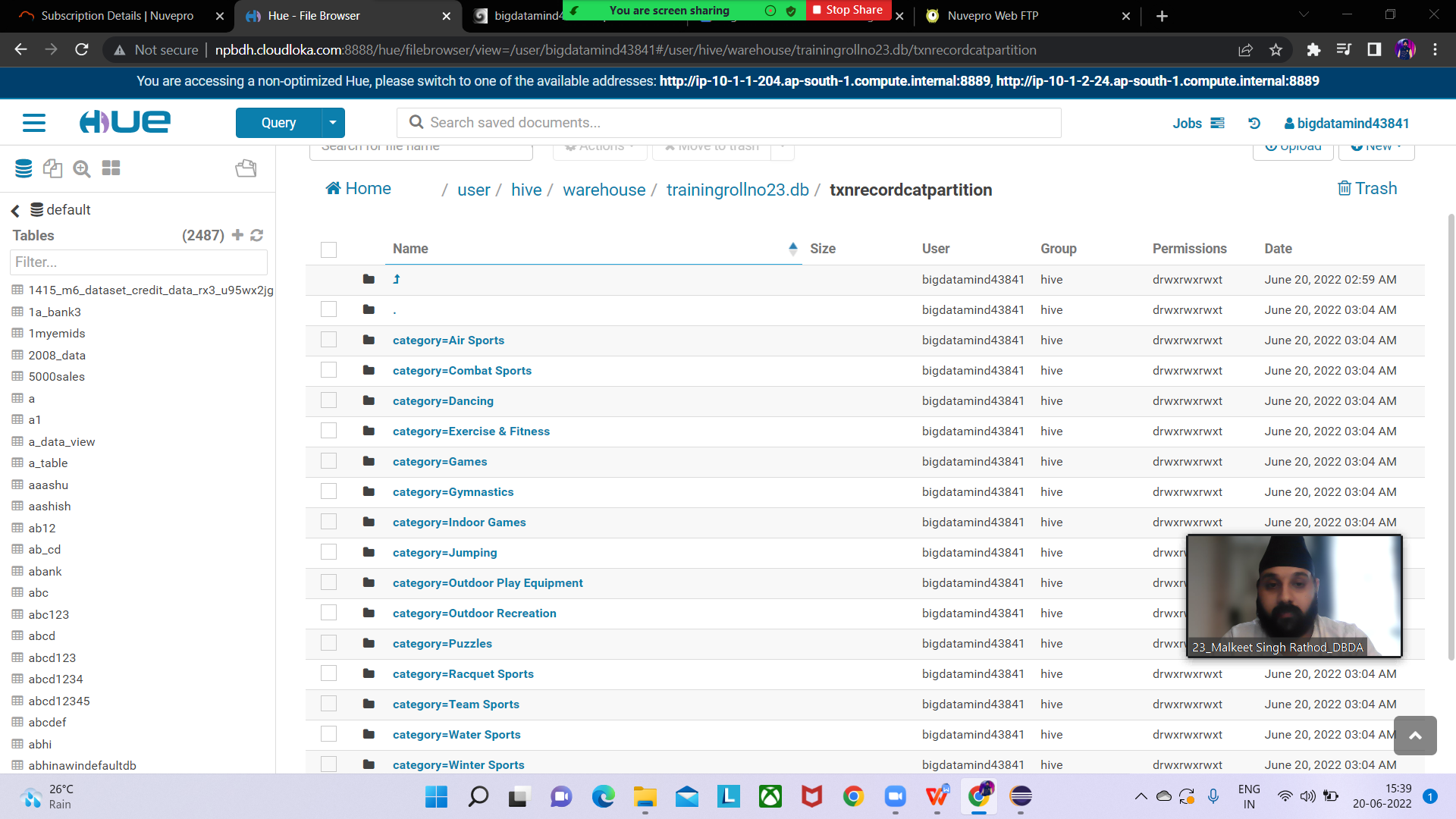
// creating partition table

insert overwrite table txnrecordcatpartition partition(category) select txn.txnno,txn.txndate,txn.custno,txn.amount,txn.product,txn.city,txn.st

ate,txn.spendby,txn.category from txnrecords txn Distribute by category;

//inserting data from txnrecords table in partitioned table

Partition created acc to category viewed in hue



**QUESTION 3**

[15 marks]

**PySpark**

Please find the AIRLINES data set

Year

Quarter

Average revenue per seat

Total number of booked seats

Answer:-

from pyspark.sql.types import StructType,StringType,IntegerType,DoubleType,LongType

>>> schema1=StructType().add("year",StringType(),True).add("qtr",StringType(),True).add("arps",DoubleType(),True).add("booked\_seats",LongType(),True)

>>> airlinesDF=spark.read.format("csv").option("header","TRUE").schema(schema1).load("/user/bigdatamind43841/airlines.csv")

>>> airlines.registerTempTable("airlines")

**What was the highest number of people travelled in which year?**

Query:-

query1DF=spark.sql("select year,sum(booked\_seats) as high from airlines group by year order by high desc")

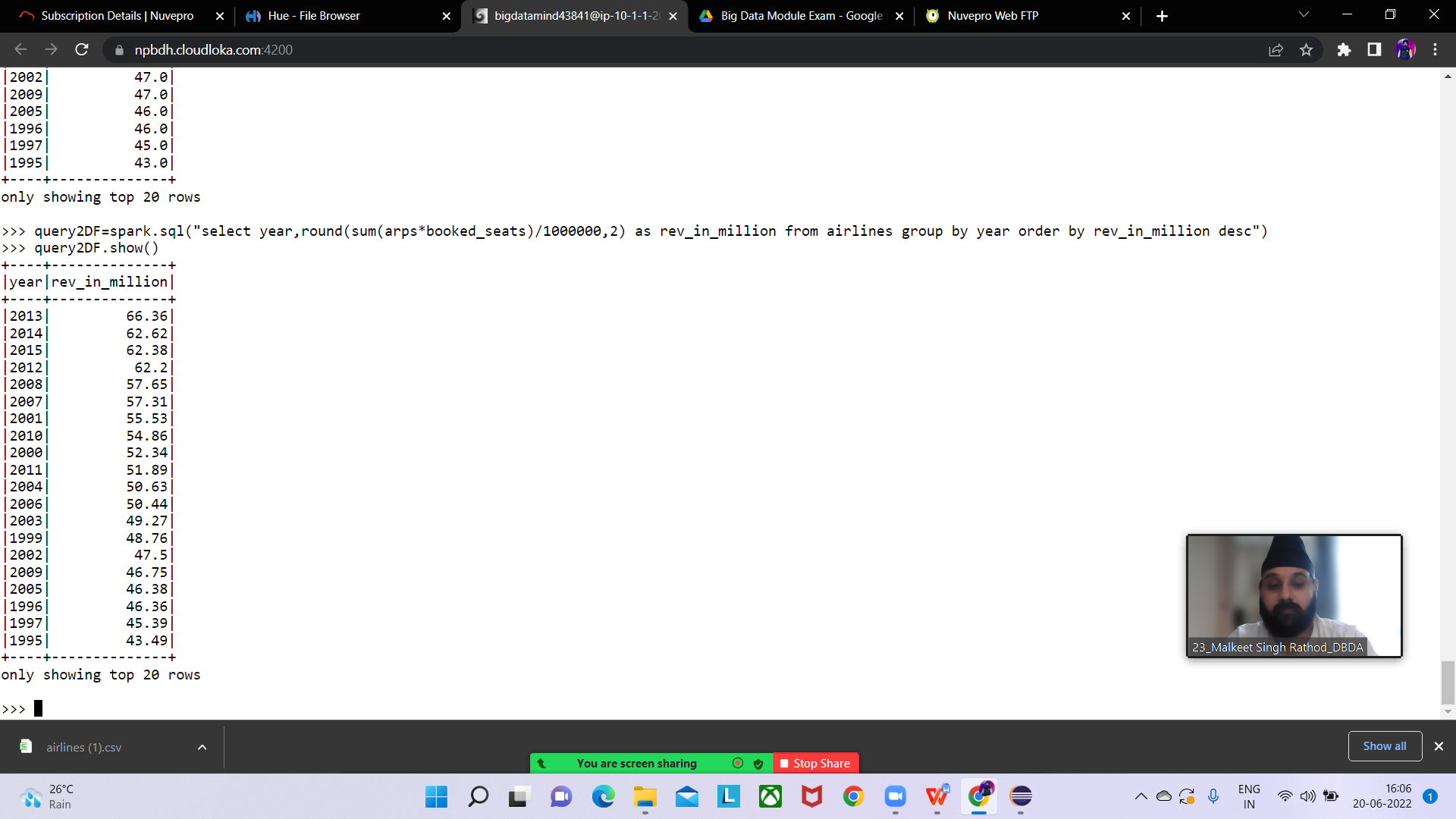
query1DF.show()



1. **Identifying the highest revenue generation for which year**

>>> query2DF=spark.sql("select year,round(sum(arps\*booked\_seats)/1000000,2) as rev\_in\_million from airlines group by year order by rev\_in\_million desc")

>>> query2DF.show()



**3) Identifying the highest revenue generation for which year and quarter (Common**

**group)**

>>> query3DF=spark.sql("select year,qtr,round(sum(arps\*booked\_seats)/1000000,2) as rev\_in\_million from airlines group by year,qtr order by rev\_in\_million desc limit 1")

>>> query3DF.show()

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

|year|qtr|rev\_in\_million|

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

|2014| 4| 18.82|

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