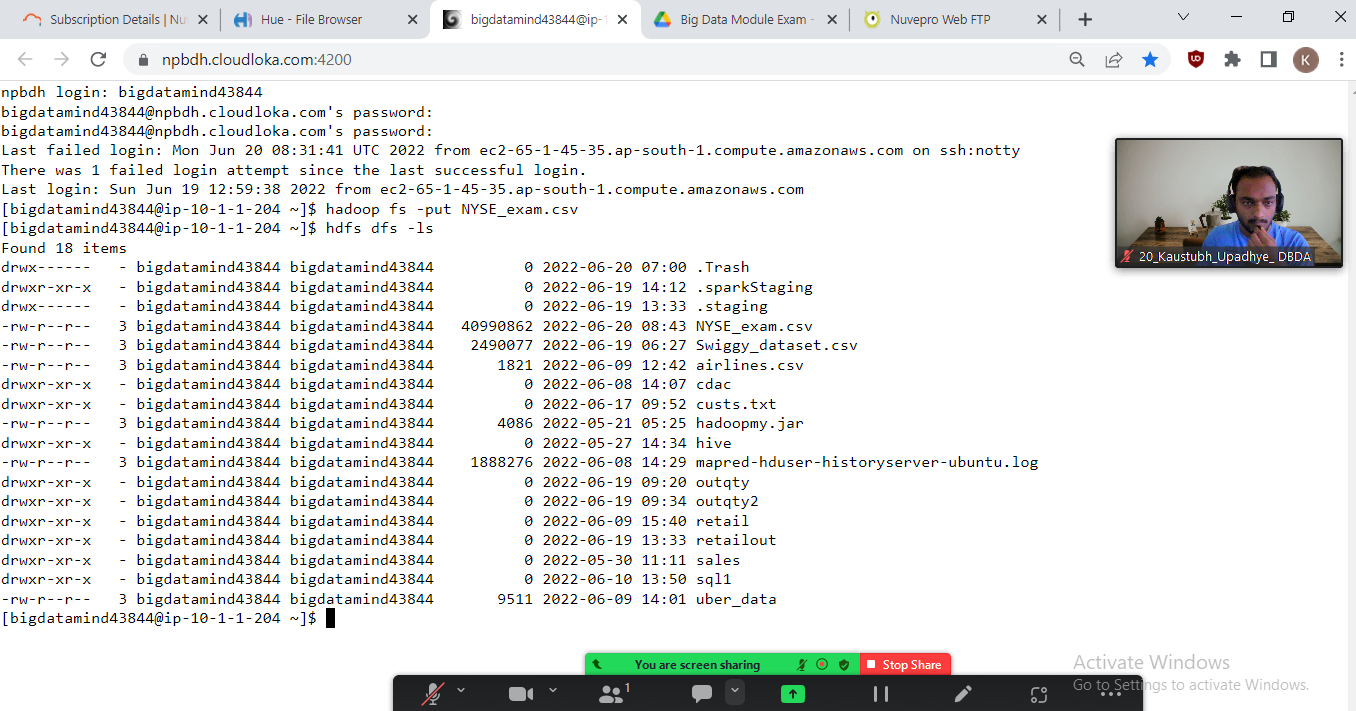
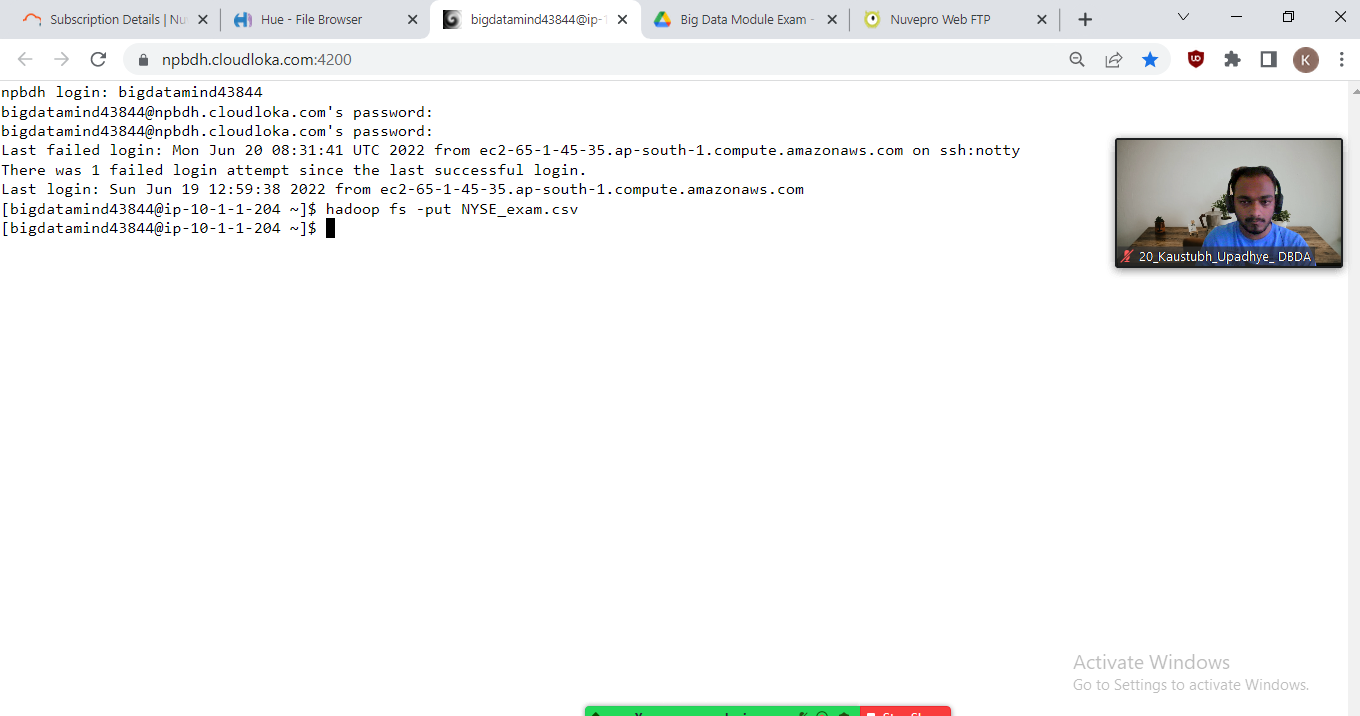
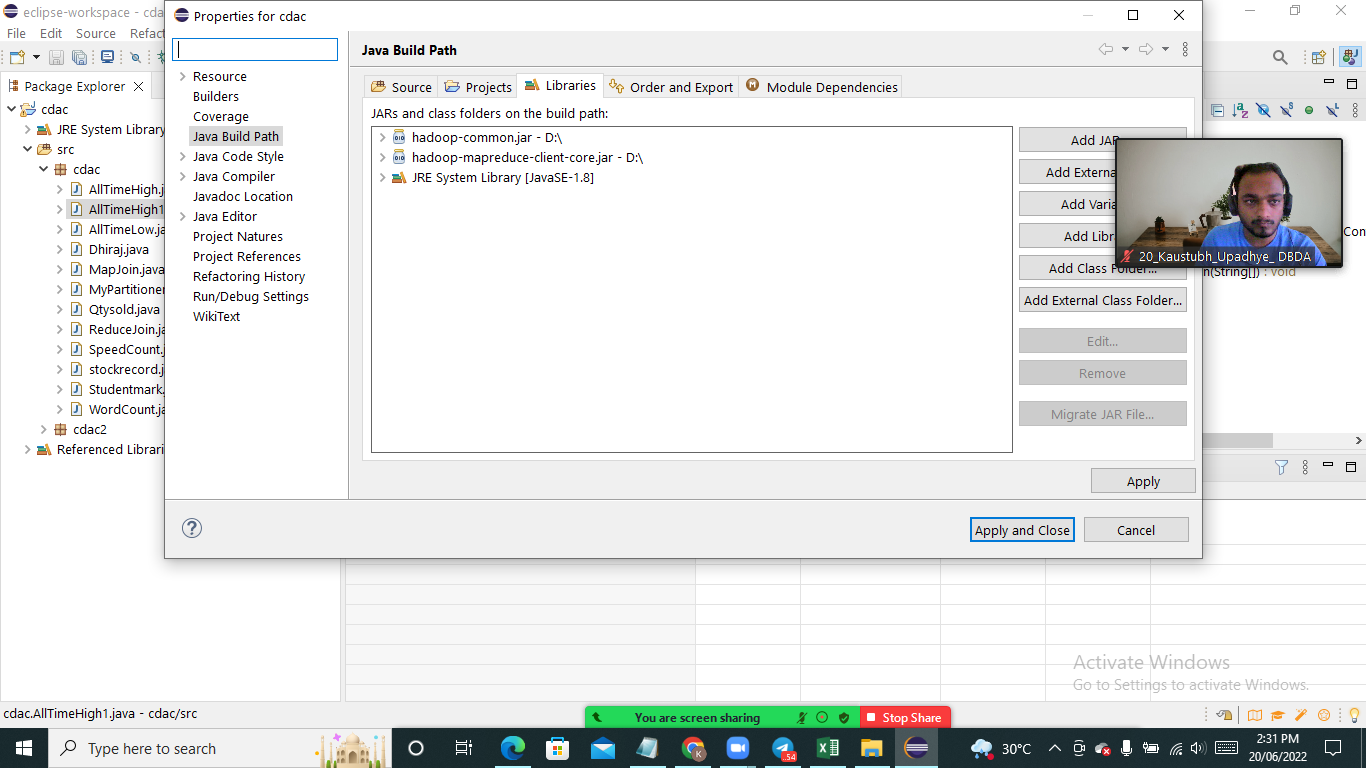
**Q1. MapReduce Problem Statement Here, we have chosen the stock market dataset on which we have performed map-reduce operations. Following is the structure of the data. Kindlyfind the solutions to the questions below**

****

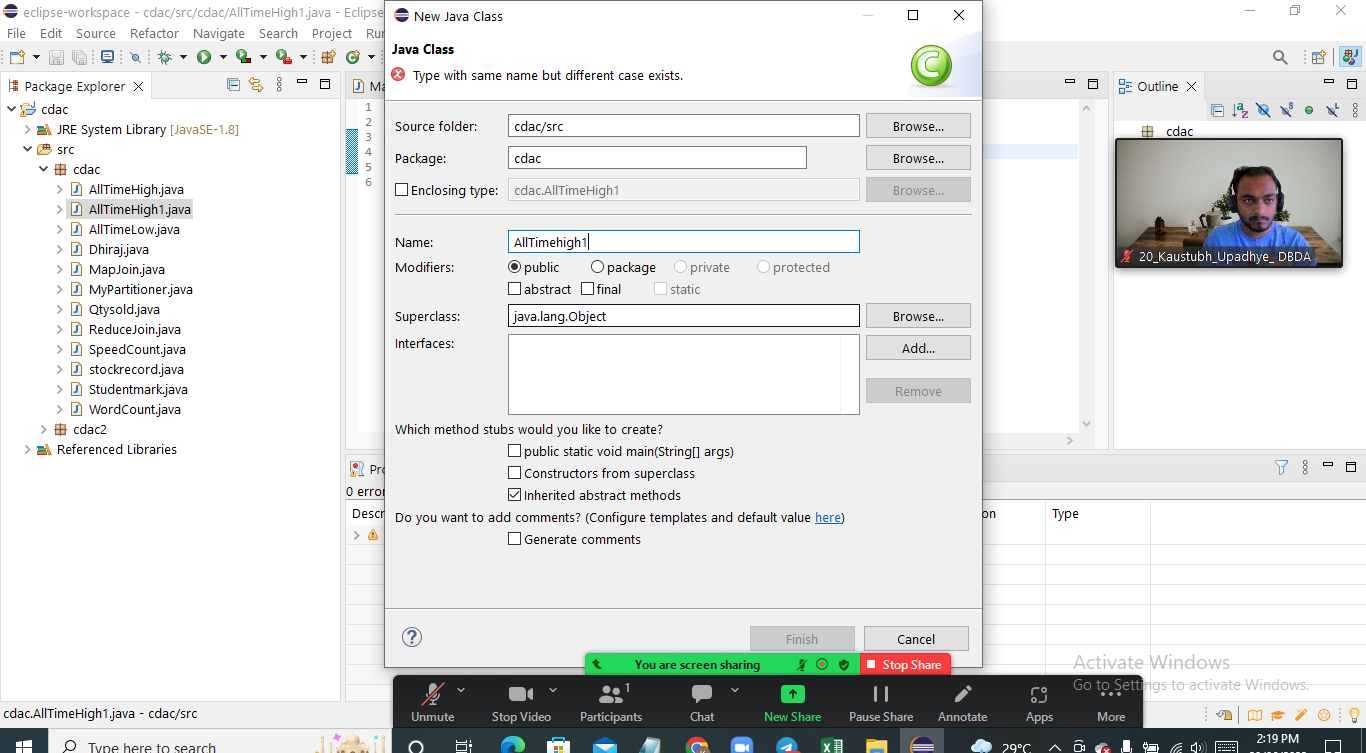
**#hadoop fs-ls**

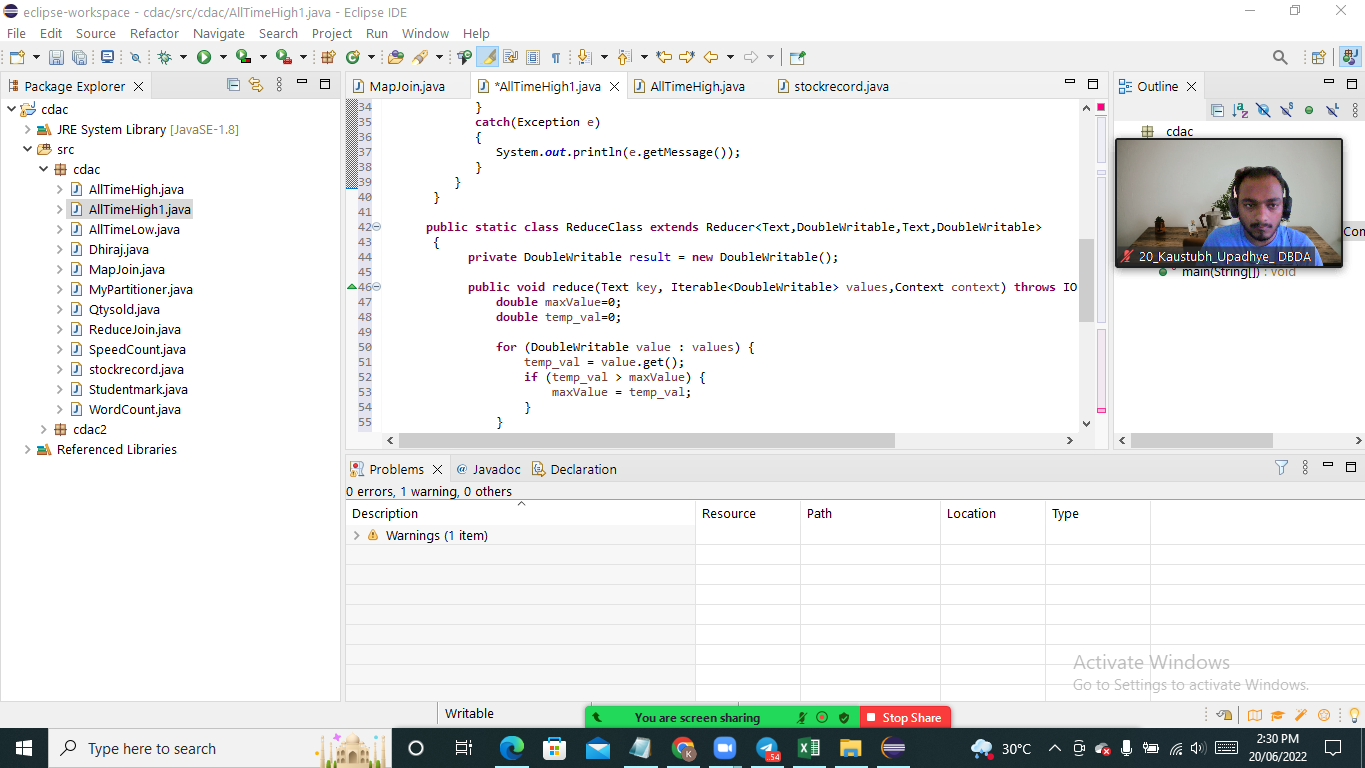
****

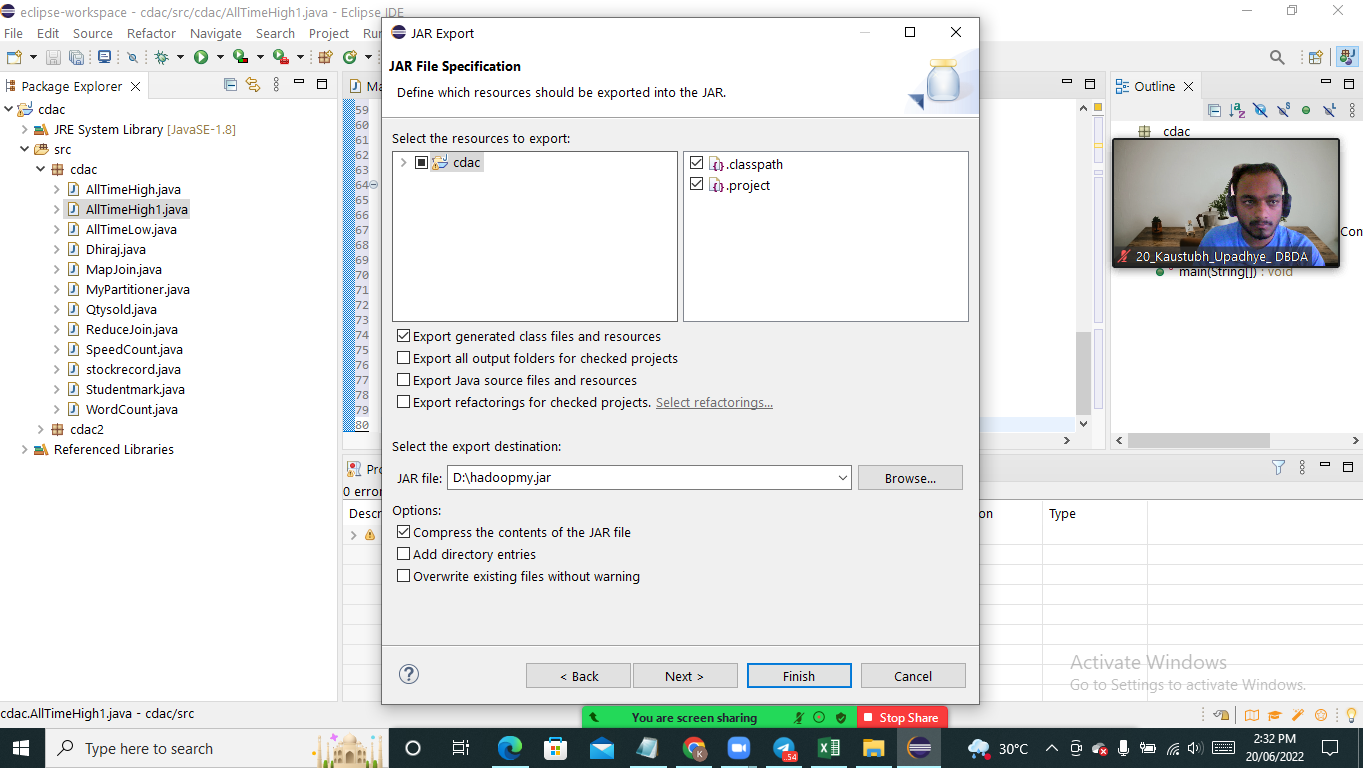
**#adding jars**

hadoop fs -put NYSE\_exam.csv

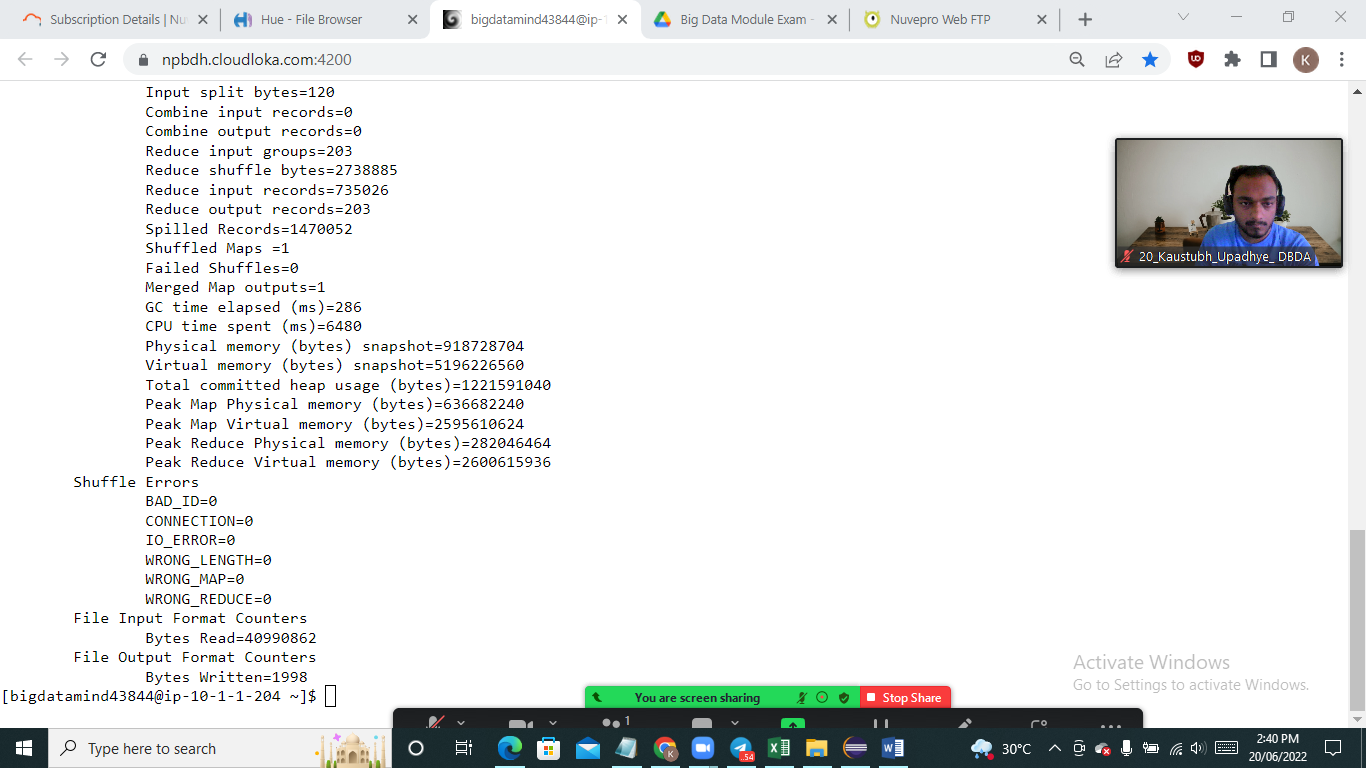
Question 2 : Find all time High price for each stock

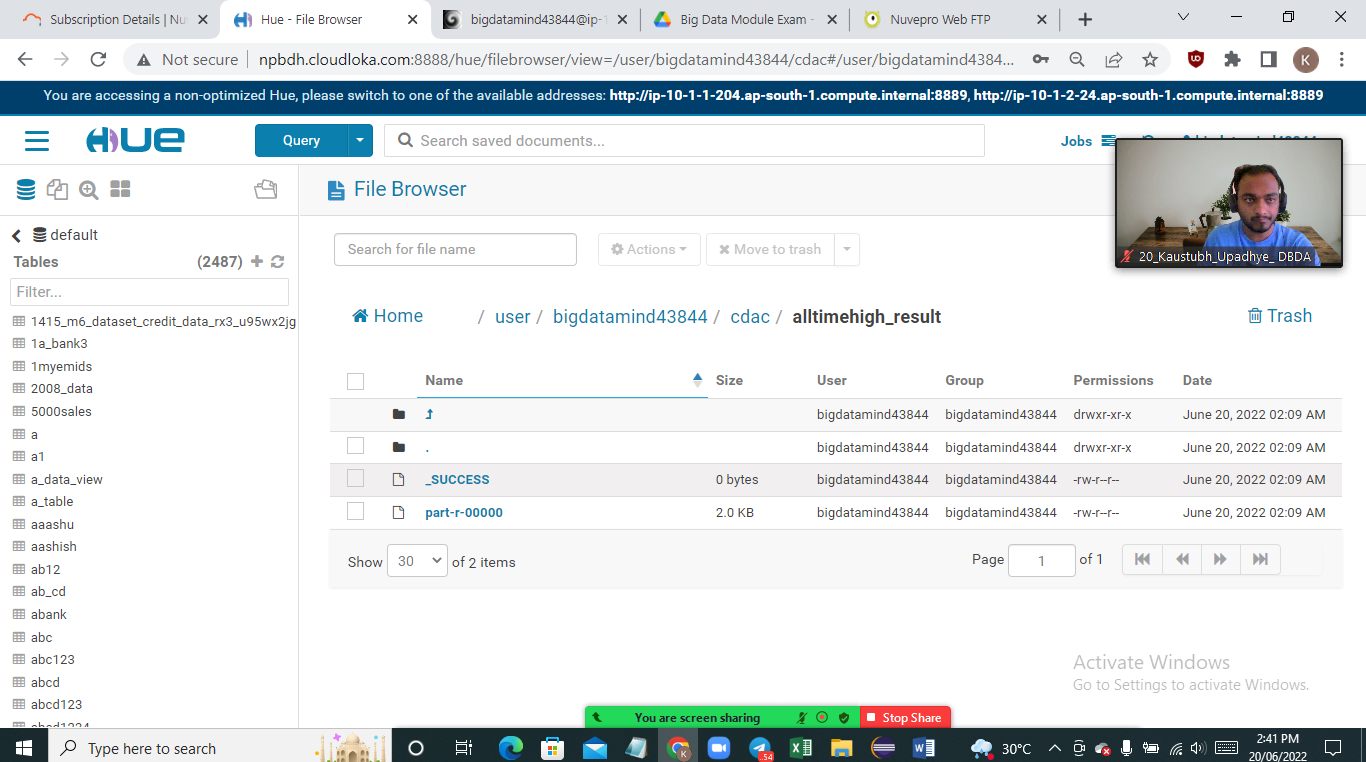


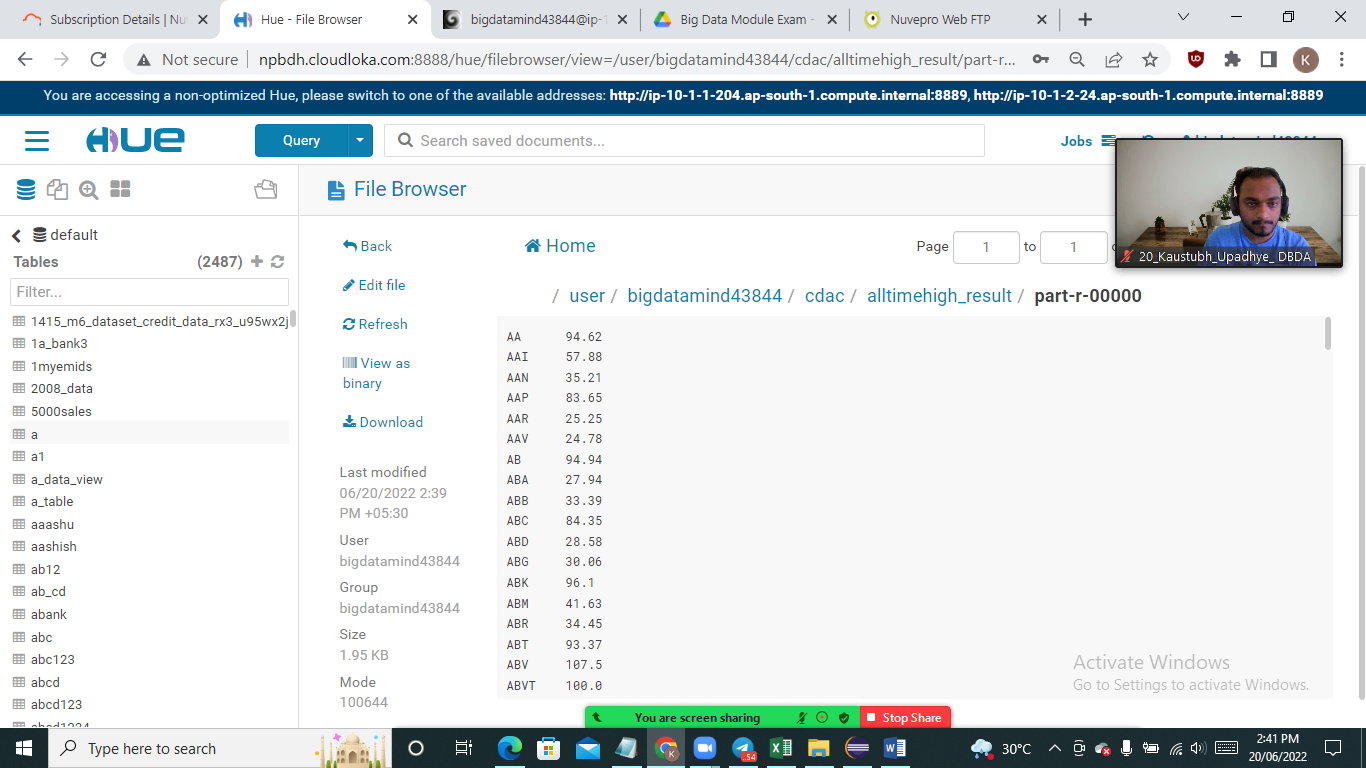




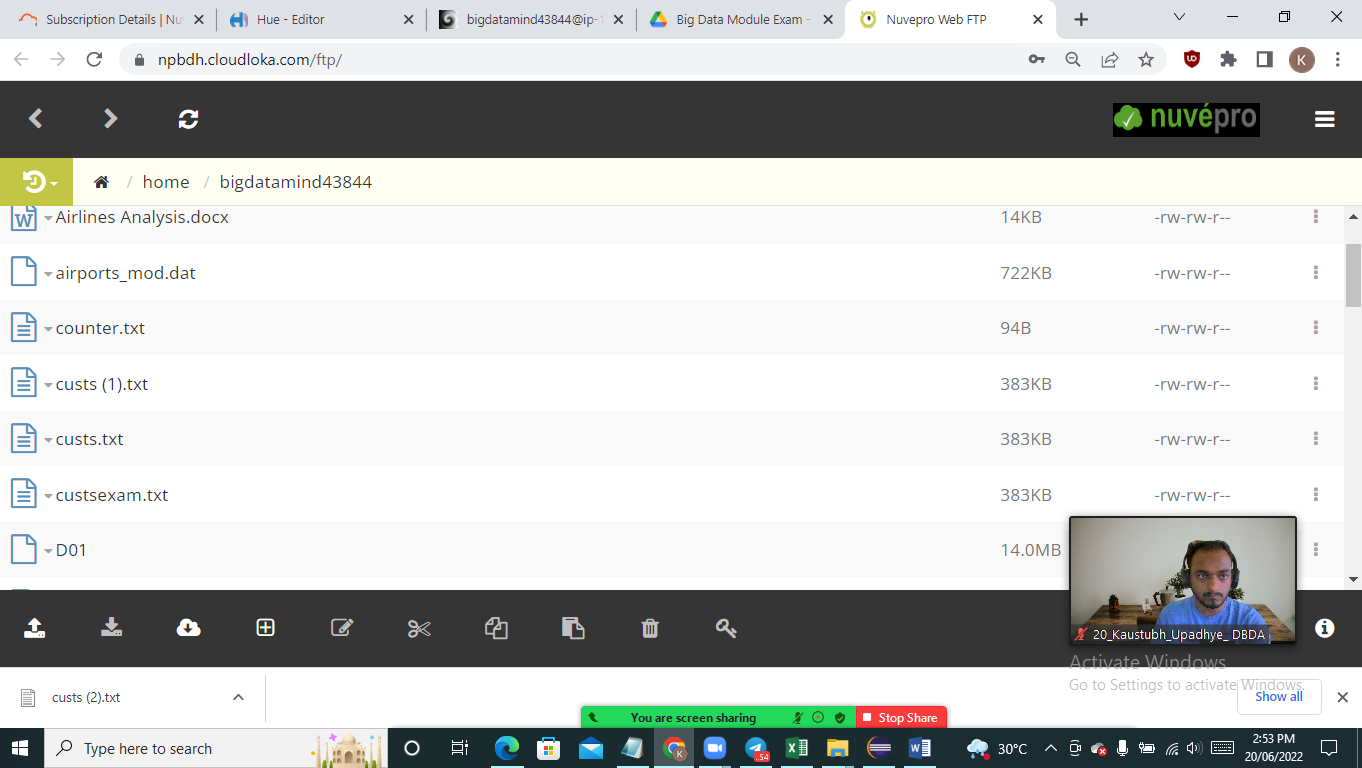
[bigdatamind43844@ip-10-1-1-204 ~]$ hadoop jar hadoopmy.jar cdac/AllTimeHigh1 NYSE\_exam.csv cdac/alltimehigh\_result

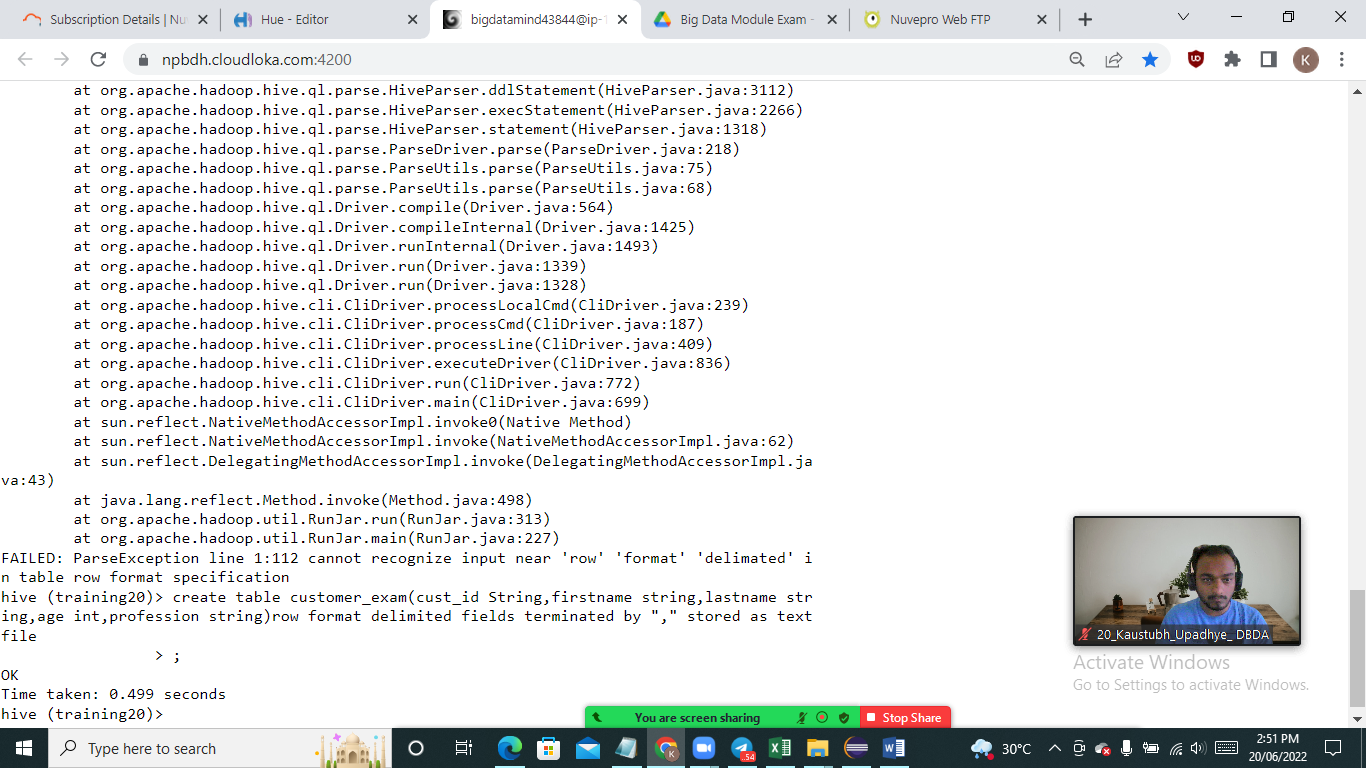


****

**#Final output**

**Hive Please find the customer data set. cust id firstname lastname age profession**

****

****

hive (training20)> create table customer\_exam(cust\_id String,firstname string,lastname str

ing,age int,profession string)row format delimited fields terminated by "," stored as text

file

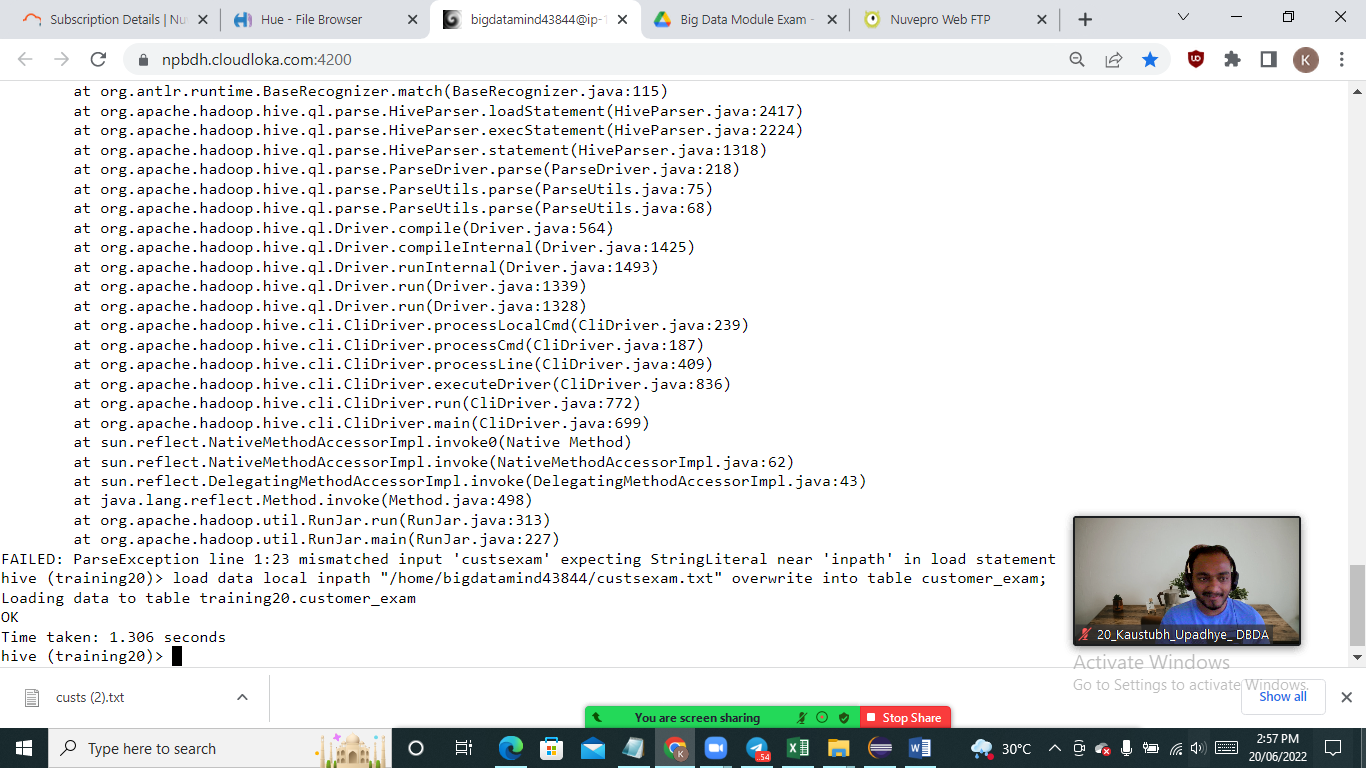
> ;

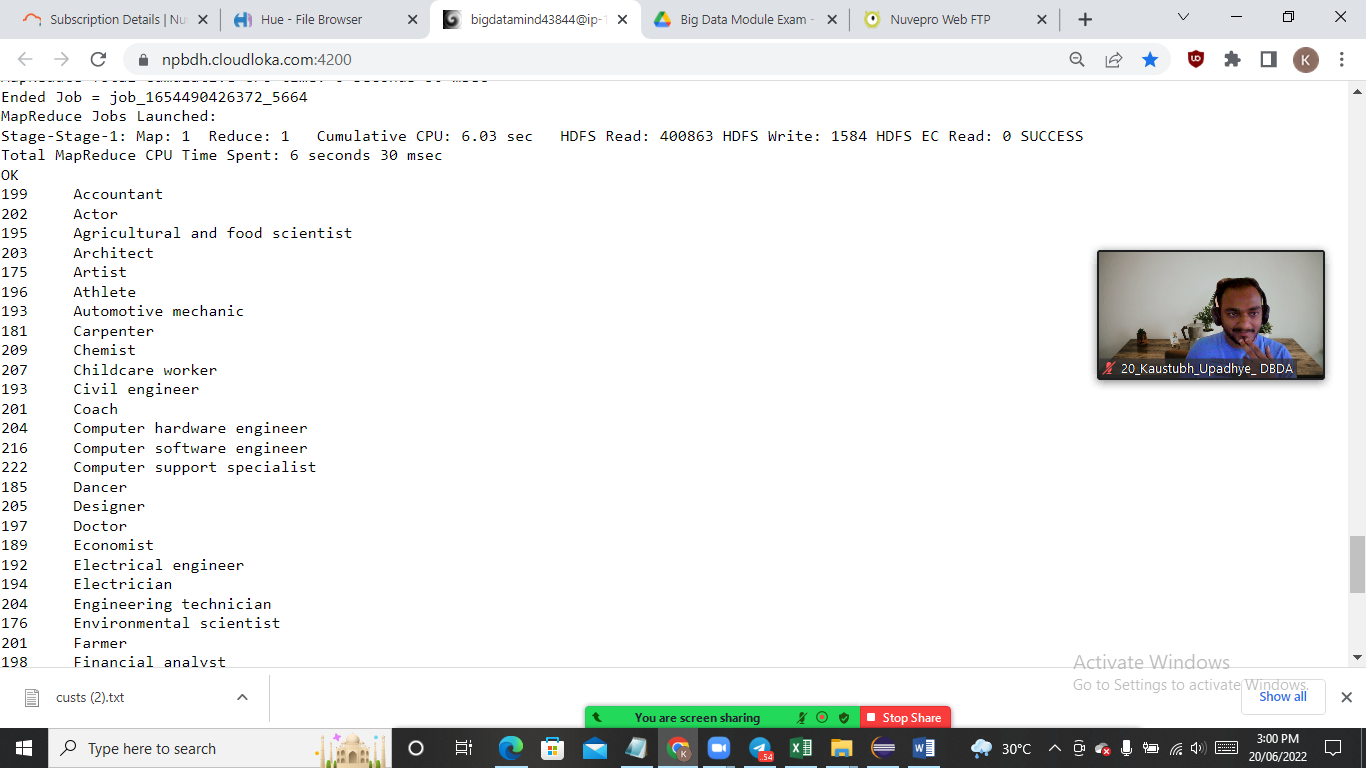
OK

Time taken: 0.499 seconds

hive (training20)>

load data local inpath “/home/bigdatamind43844/custs\_exam.txt” overwrite into table customer\_exam

****

****

**Que1)Output Cutomer –**

**select count(cust\_id)as Countof\_customers,profession from customer\_exam group by profession;**

**--------------------------------------------------------------------------------**

**2)Write a program to find the top 10 products sales wise**

Loading data to table training20.txnrecords\_exam

OK

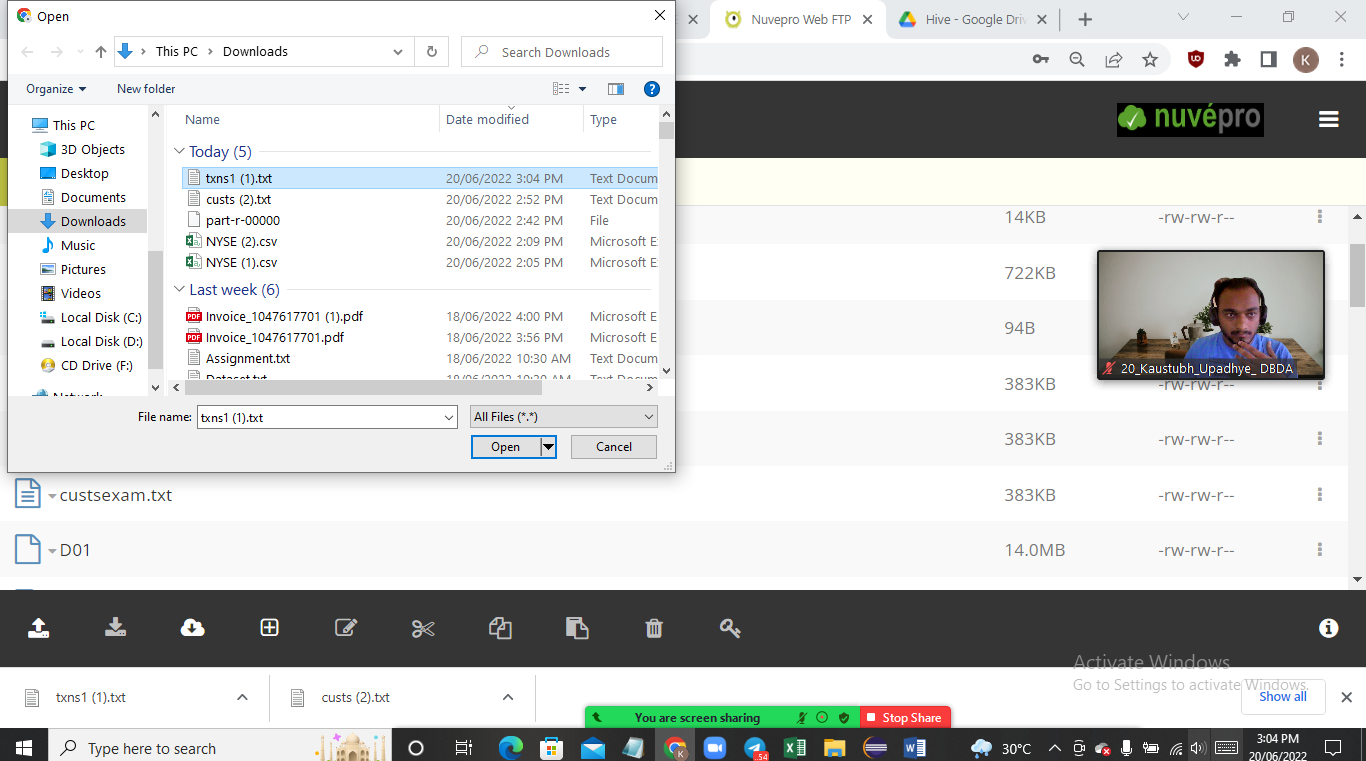
Time taken: 0.742 seconds

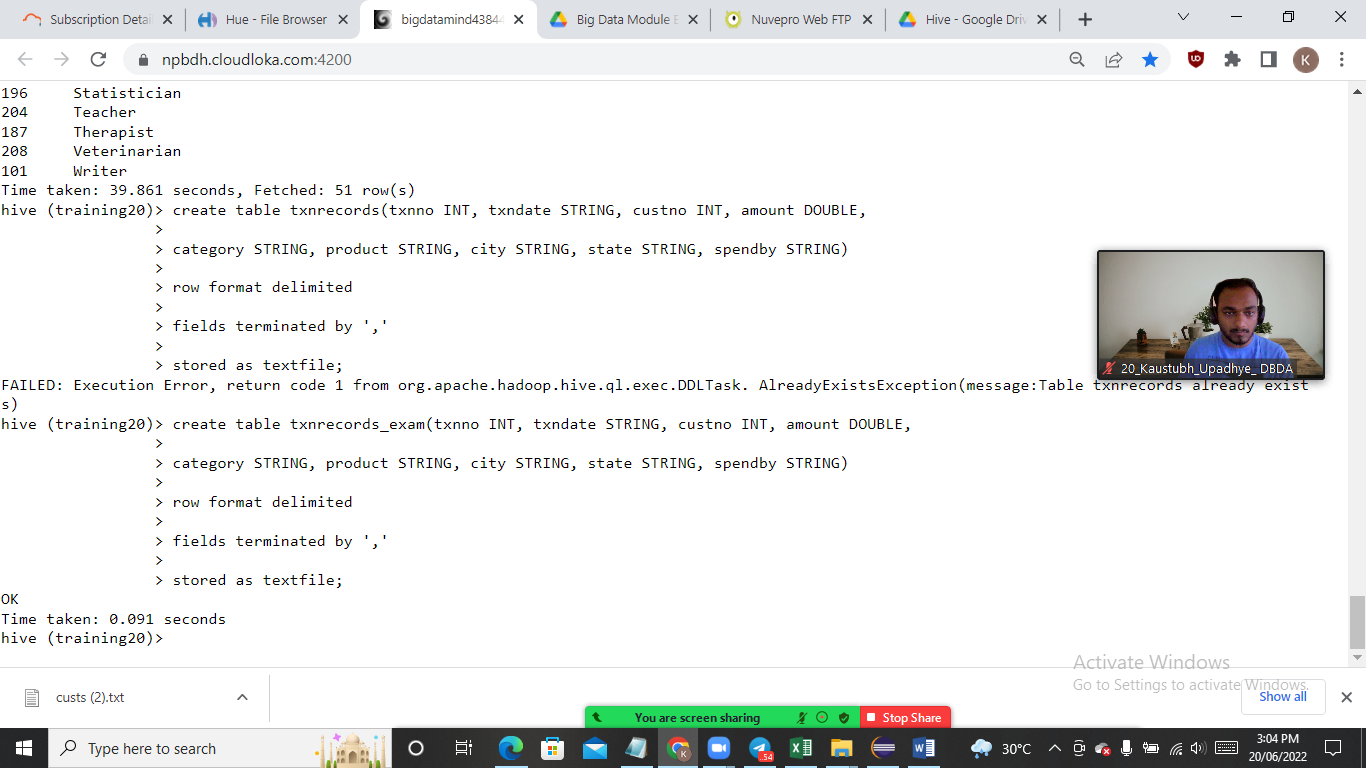
hive (training20)> **select product,sum(amount) as total from txnrecords\_exam group by product order by total limit 10;**

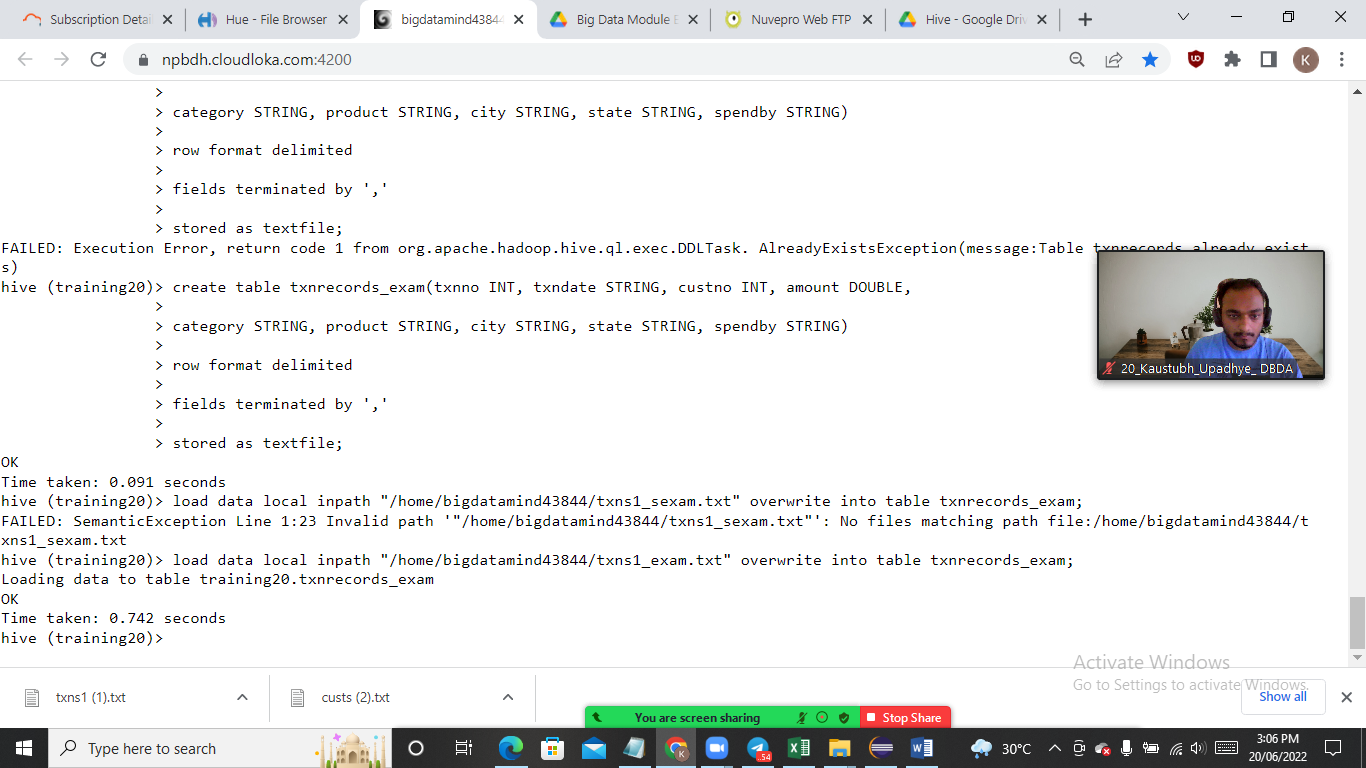
Query ID = bigdatamind43844\_20220620094318\_3b84add4-739d-4304-8de9-b6cfca81885c

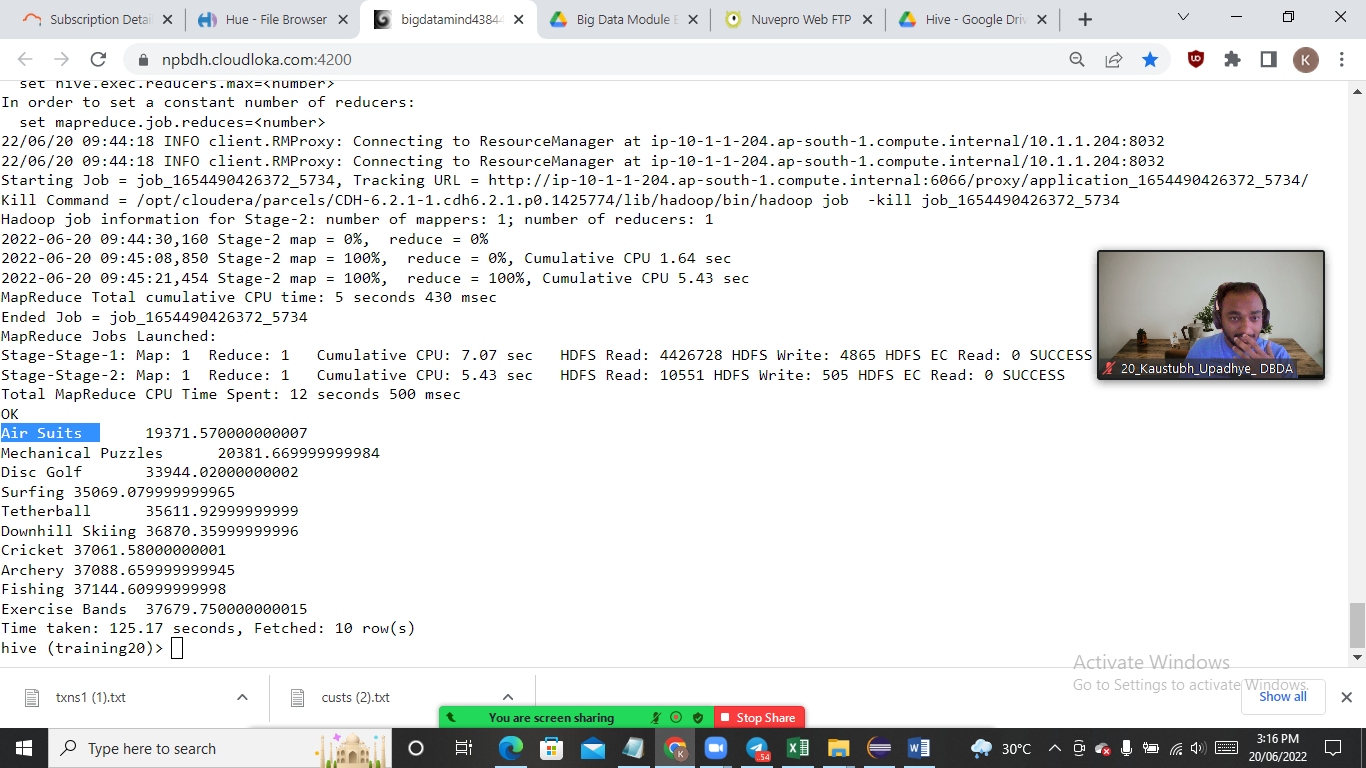
Total jobs = 2

Launching Job 1 out of 2

****

****

****

****

**Output –Que 2)**

**3) Write a program to create partiioned table on category**

hive> set hive.cli.print.current.db=true;

set hive.exec.dynamic.partition=true;

hive (default)> use training20;

create table txnrecs\_part(txnno INT, txndate STRING, custno INT, amount DOUBLE,

product STRING, city STRING, state STRING, spendby STRING)

partitioned by (category STRING)

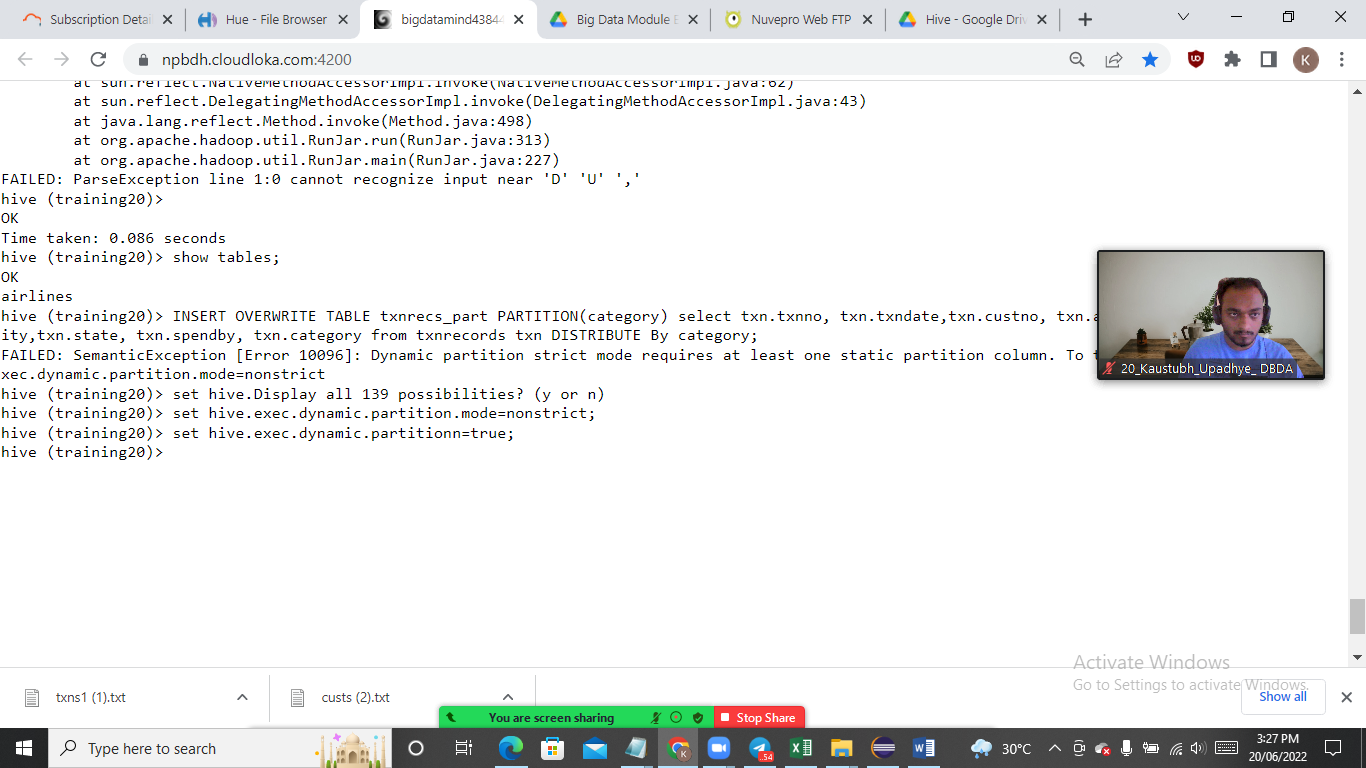
row format delimited

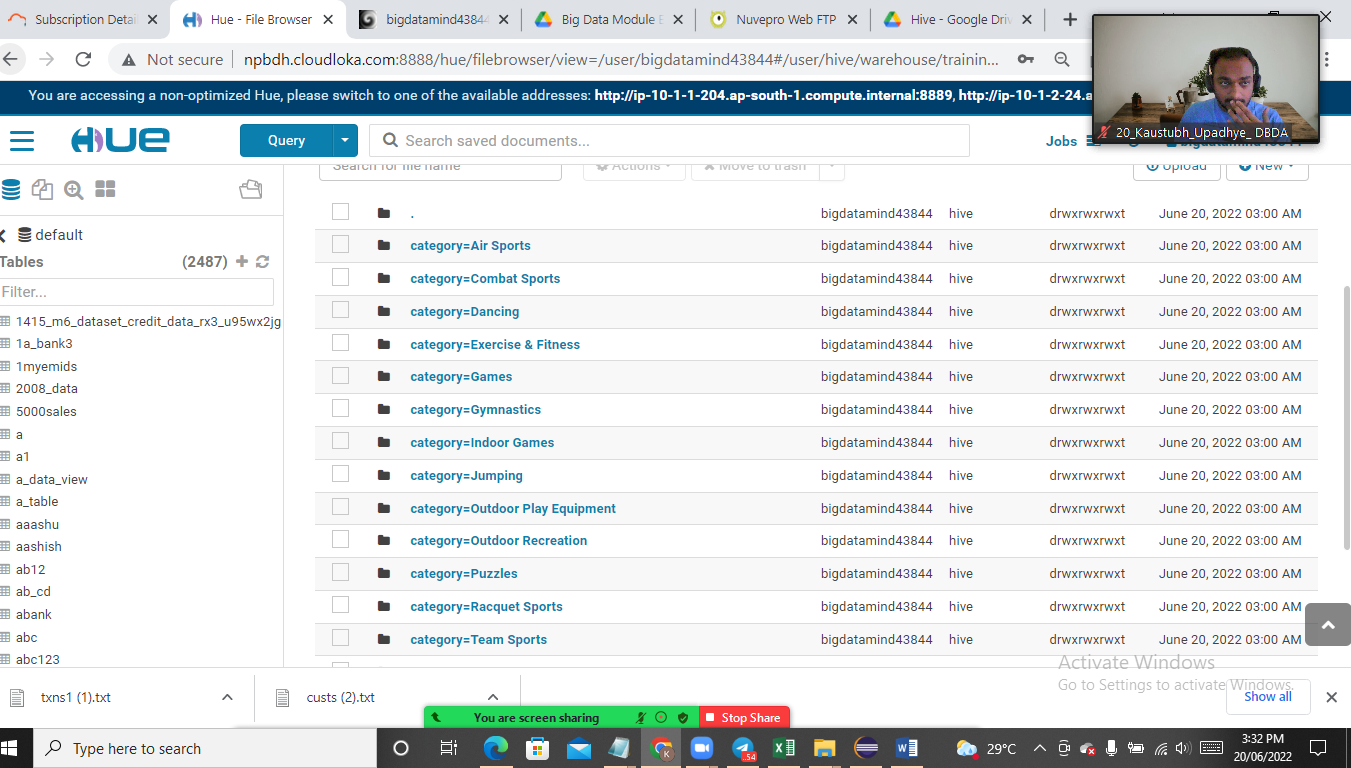
fields terminated by ','

stored as textfile;

INSERT OVERWRITE TABLE txnrecs\_part PARTITION(category) select txn.txnno, txn.txndate,txn.custono, txn.amount,txn.product,

> txn.city,txn.state, txn.spendby, txn.category from txnrecords txn DISTRIBUTE By category;

****

****

**QUESTION 3 [15 marks] PySpark Please find the AIRLINES data set Year Quarter Average revenue per seat Total number of booked seats**

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

**2) Identifying the highest revenue generation for which year**

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

**Que1)**

airline\_rdd=sc.textFile('/user/bigdatamind43827/exam/airlines.csv')

rdd1=airline\_rdd.map(lambda a: a.encode("ascii","ignore"))

array\_rdd=rdd1.map(lambda a : a.split(','))

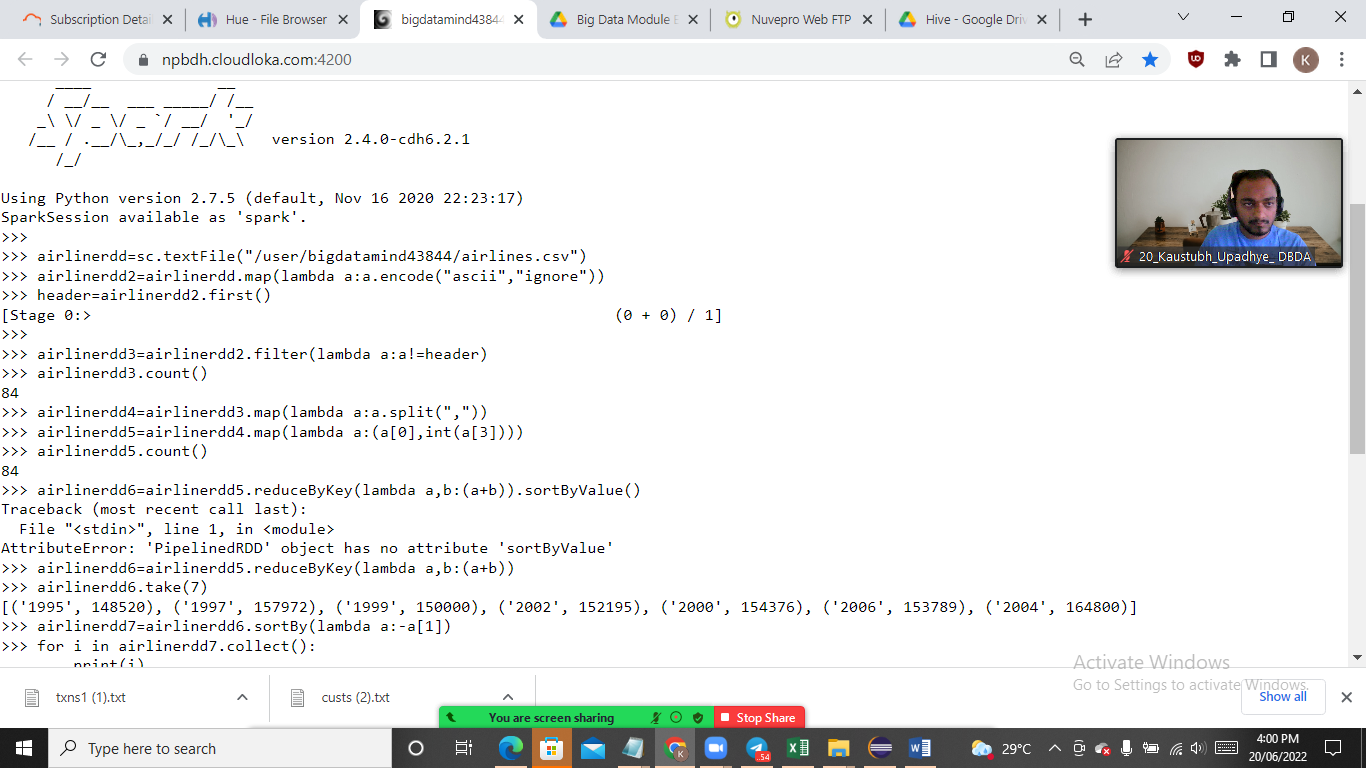
header=array\_rdd.first()

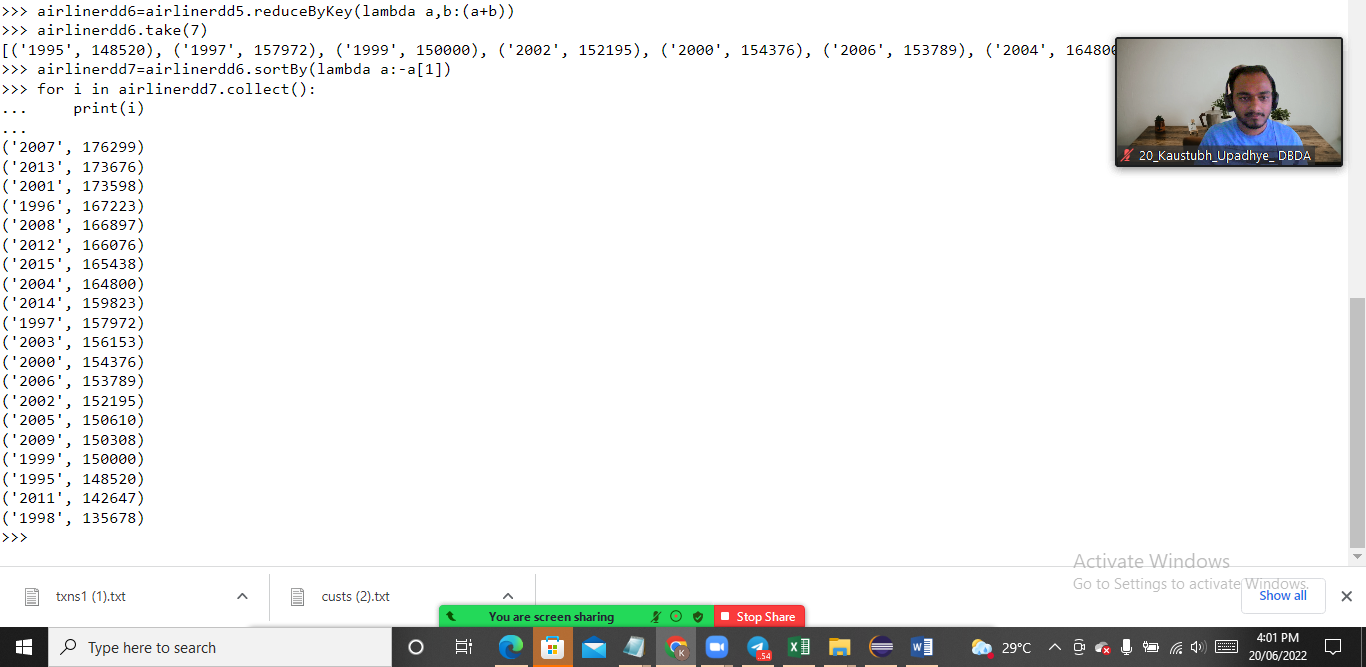
rdd2=array\_rdd.filter(lambda a : a!= header)

key\_value2=rdd2.map(lambda a : (a[0], float(a[3]) ))

add\_total2=key\_value2.reduceByKey(lambda a,b : a+b)

sortbyval2=add\_total2.sortBy(lambda a: -a[1])

****

****

Q2) Identifying the highest revenue generation for which year

==>

airline\_rdd=sc.textFile('/user/bigdatamind43827/exam/airlines.csv')

rdd1=airline\_rdd.map(lambda a: a.encode("ascii","ignore"))

array\_rdd=rdd1.map(lambda a : a.split(','))

header=array\_rdd.first()

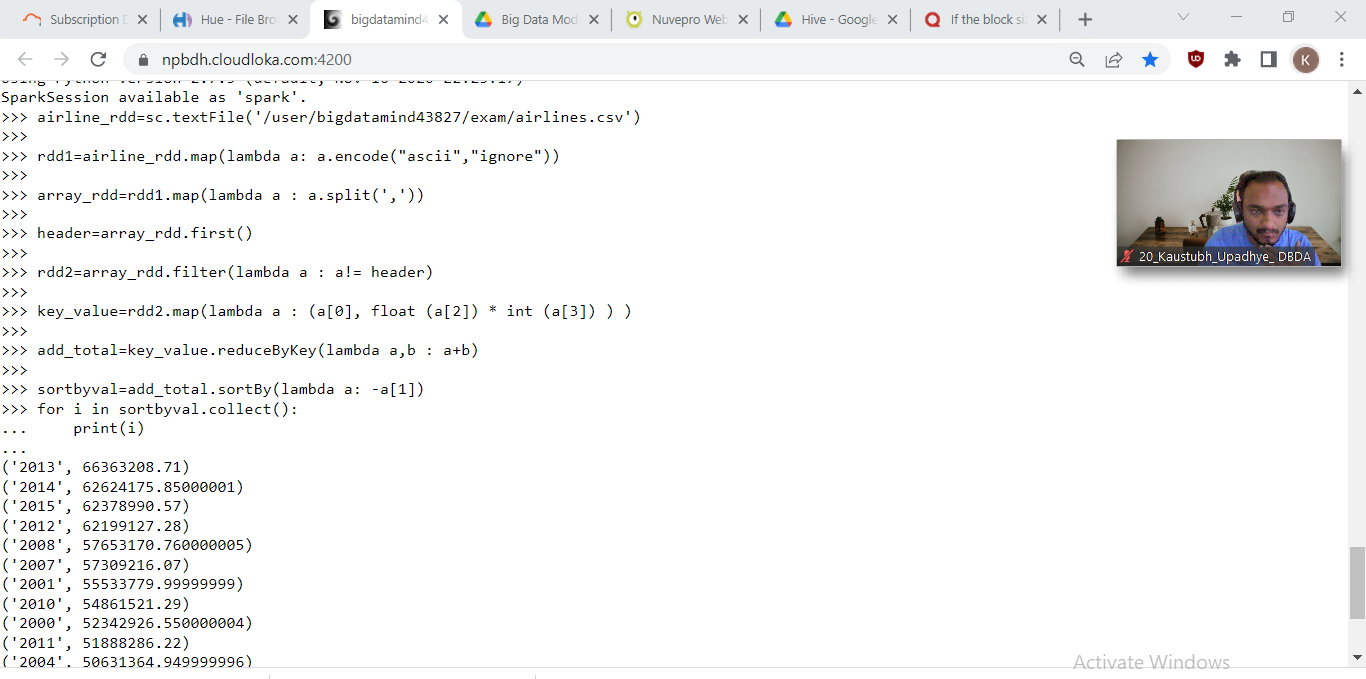
rdd2=array\_rdd.filter(lambda a : a!= header)

key\_value=rdd2.map(lambda a : (a[0], float (a[2]) \* int (a[3]) ) )

add\_total=key\_value.reduceByKey(lambda a,b : a+b)

sortbyval=add\_total.sortBy(lambda a: -a[1])

('2013', 66363208.71)



Q3)) Identifying the highest revenue generation for which year and quarter (Common

group)

==>

airline\_rdd=sc.textFile('/user/bigdatamind43827/exam/airlines.csv')

rdd1=airline\_rdd.map(lambda a: a.encode("ascii","ignore"))

array\_rdd=rdd1.map(lambda a : a.split(','))

header=array\_rdd.first()

rdd2=array\_rdd.filter(lambda a : a!= header)

key\_value2=rdd2.map(lambda a : ((a[0]+" "+a[1]), float(a[2])\*int(a[3]) ))

add\_total2=key\_value2.reduceByKey(lambda a,b : a+b)

sortbyval2=add\_total2.sortBy(lambda a: -a[1])

