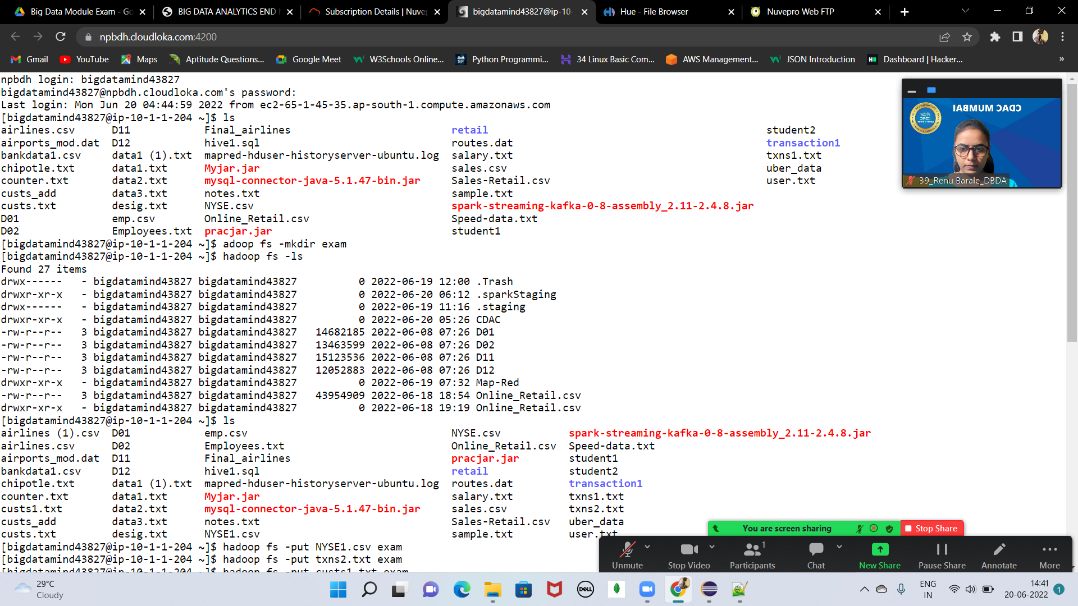
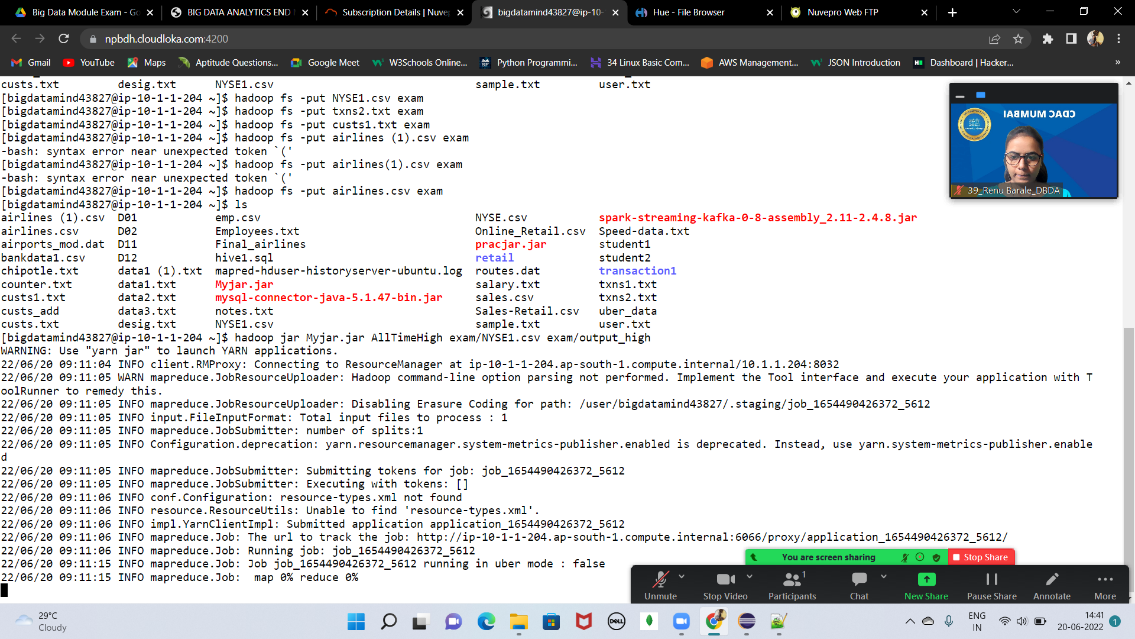
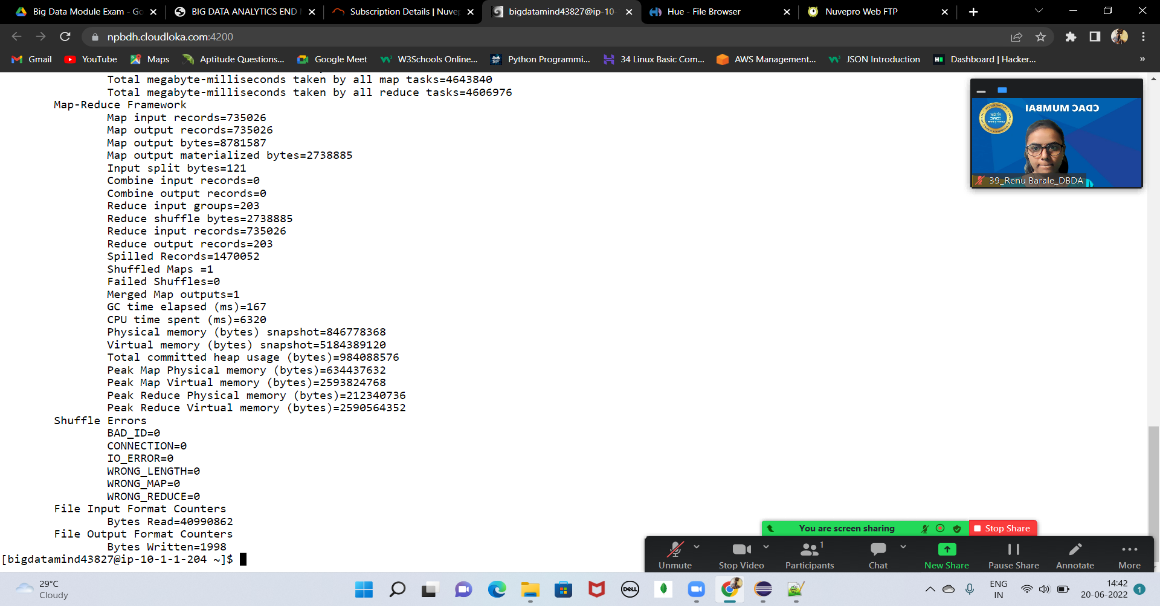
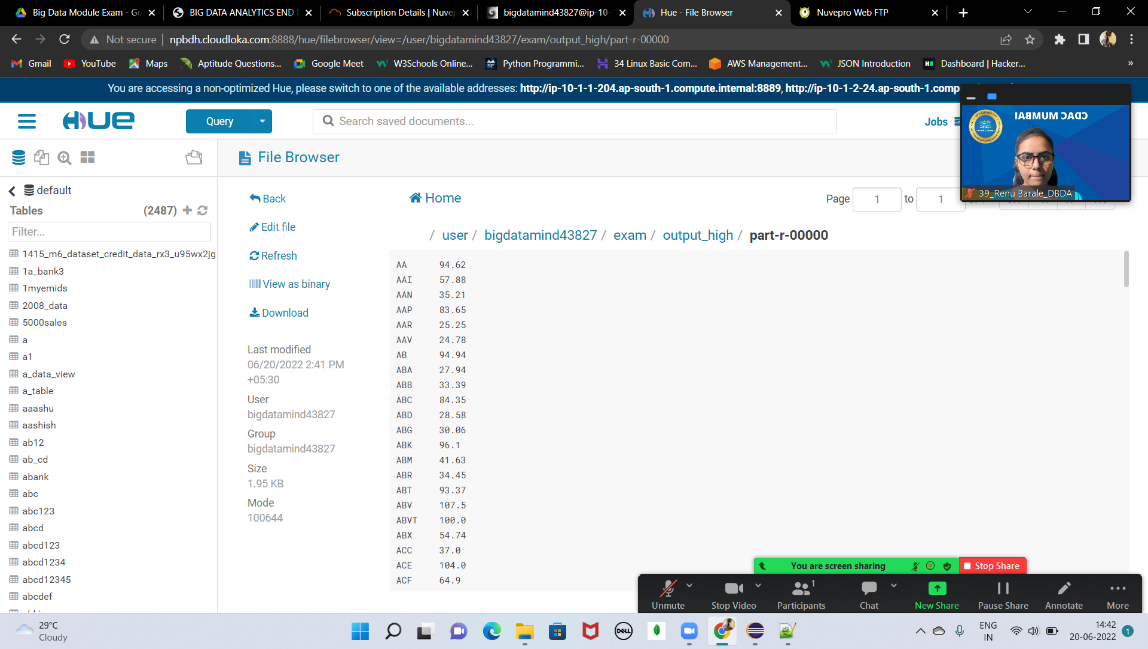
Question 2 : Find all time High price for each stock









**Hive**

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

create table customer\_exam(

cust\_id bigint, firstname string,

lastname string, age int,

profession string

)

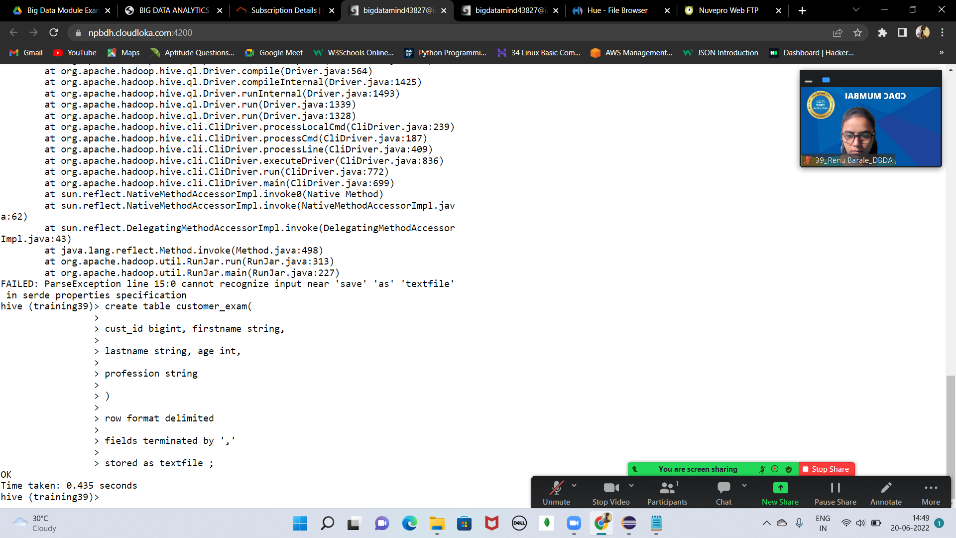
row format delimited

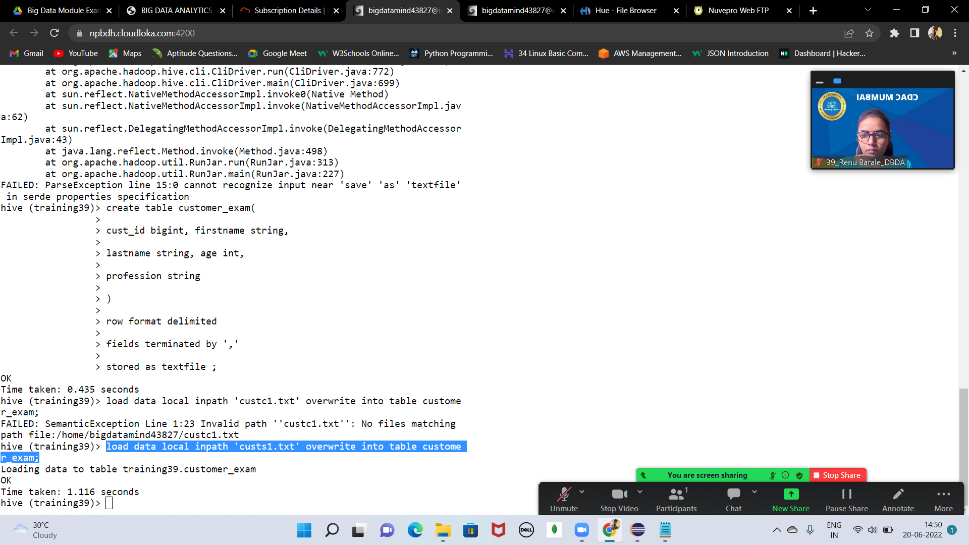
fields terminated by ','

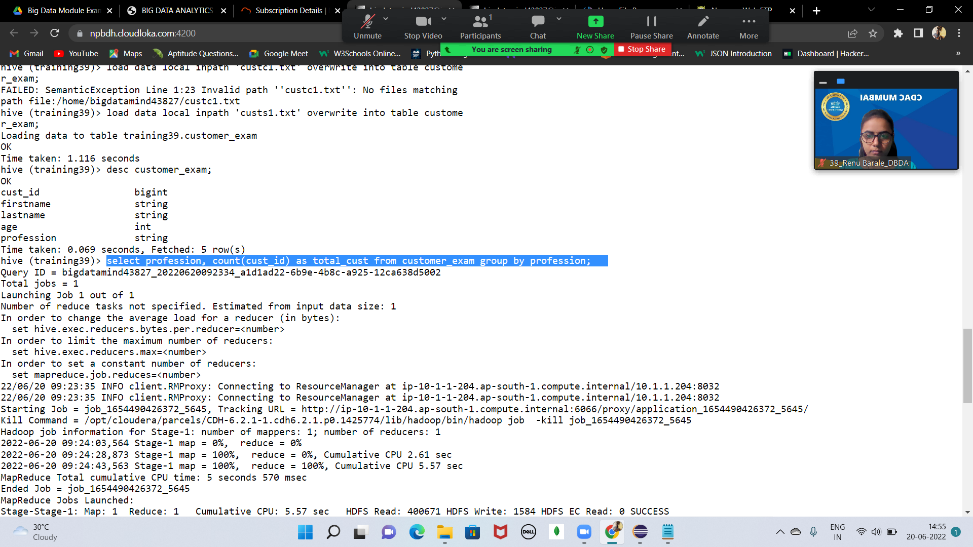
stored as textfile ;

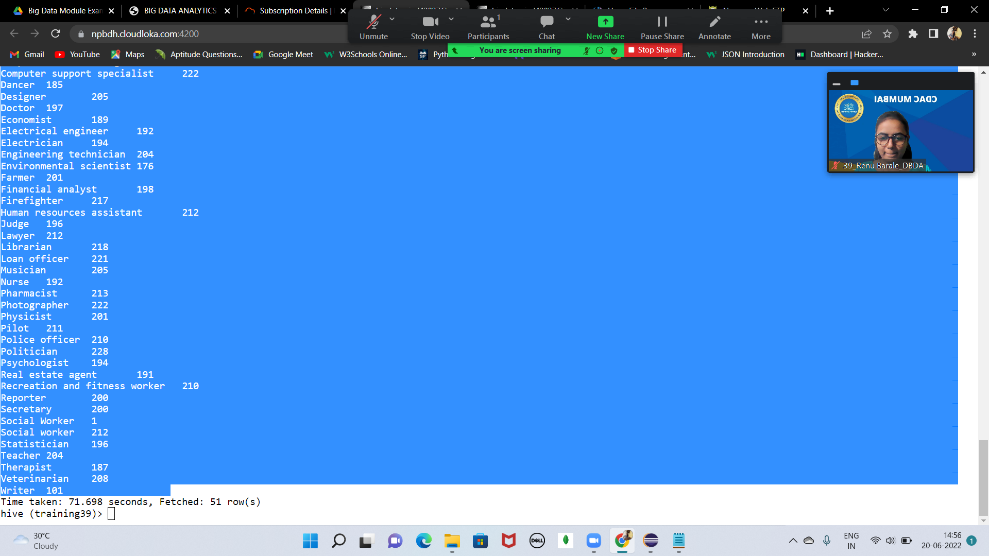
load data local inpath 'custs1.txt' overwrite into table customer\_exam;

select profession, count(cust\_id) as total\_cust from customer\_exam group by profession;









1. Write a program to find the top 10 products sales wise 3) Write a program to create partiioned table on category

create table txn\_exam(

txn\_id bigint,

txn\_date string,

cust\_id bigint,

amount bigint,

category string,

product string,

city string,

state string,

spendby string

)

row format delimited

fields terminated by ','

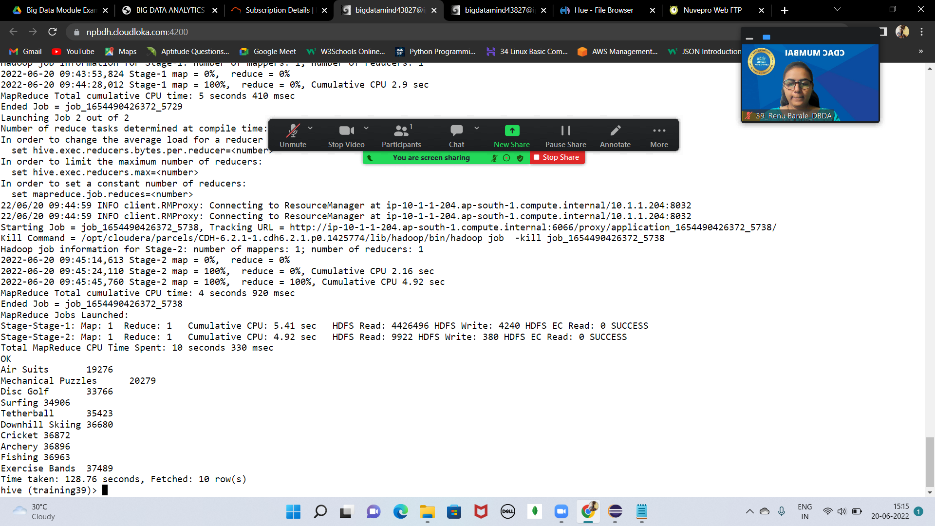
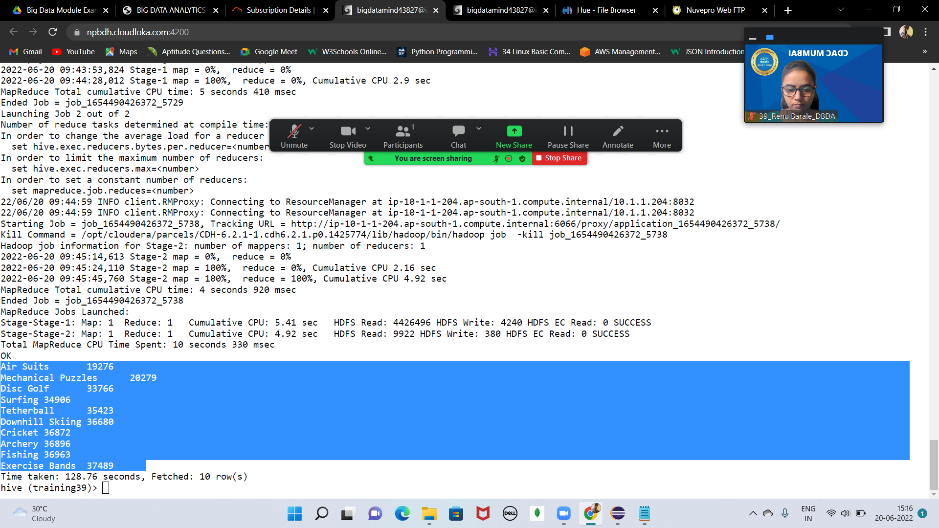
stored as textfile;

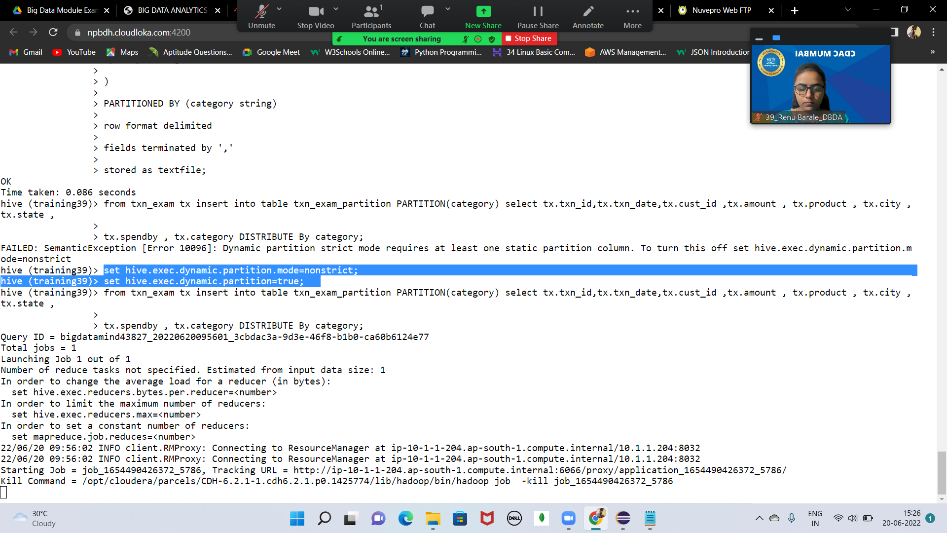
load data local inpath 'txns2.txt' overwrite into table txn\_exam;

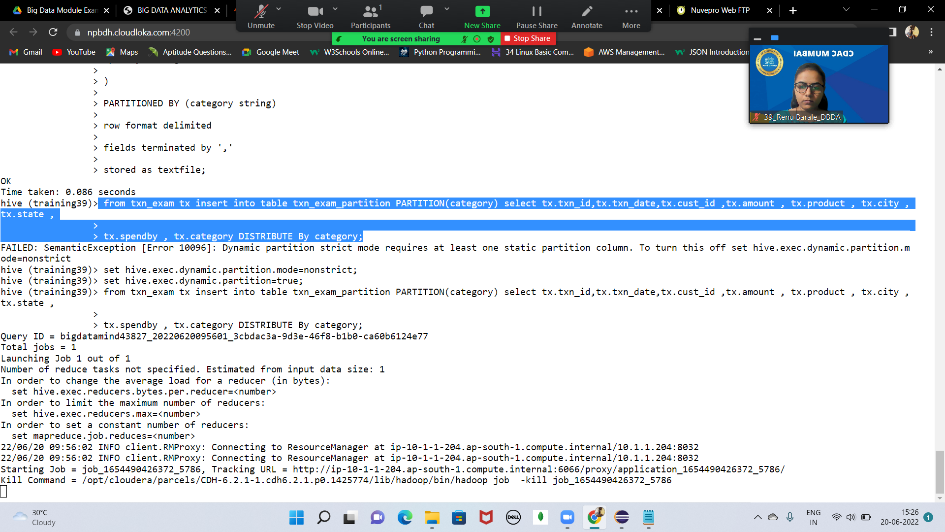
Q2) Write a program to find the top 10 products sales wise

==>

select product , sum(amount) as total from txn\_exam group by product order by total limit 10;







create table txn\_exam\_partition(

txn\_id bigint,

txn\_date string,

cust\_id bigint,

amount bigint,

product string,

city string,

state string,

spendby string

)

PARTITIONED BY (category string)

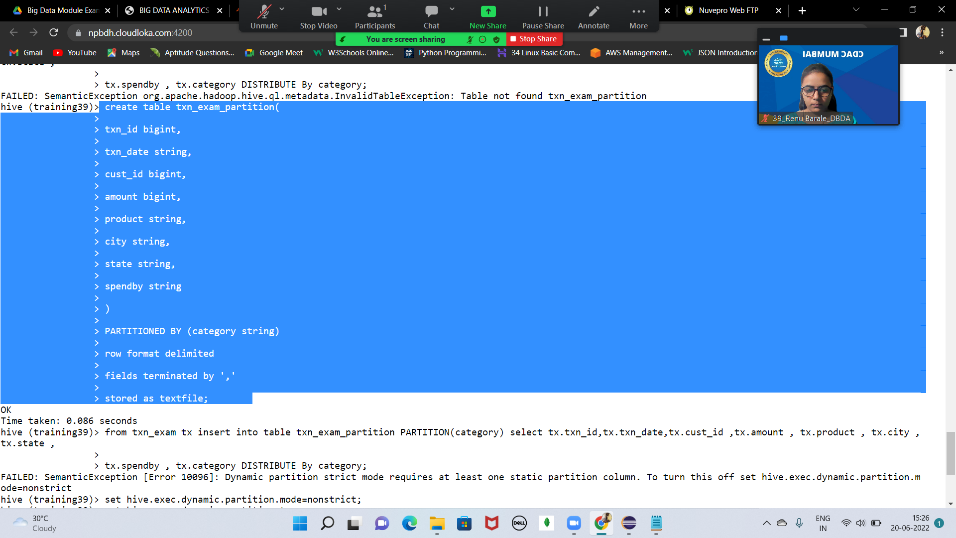
row format delimited

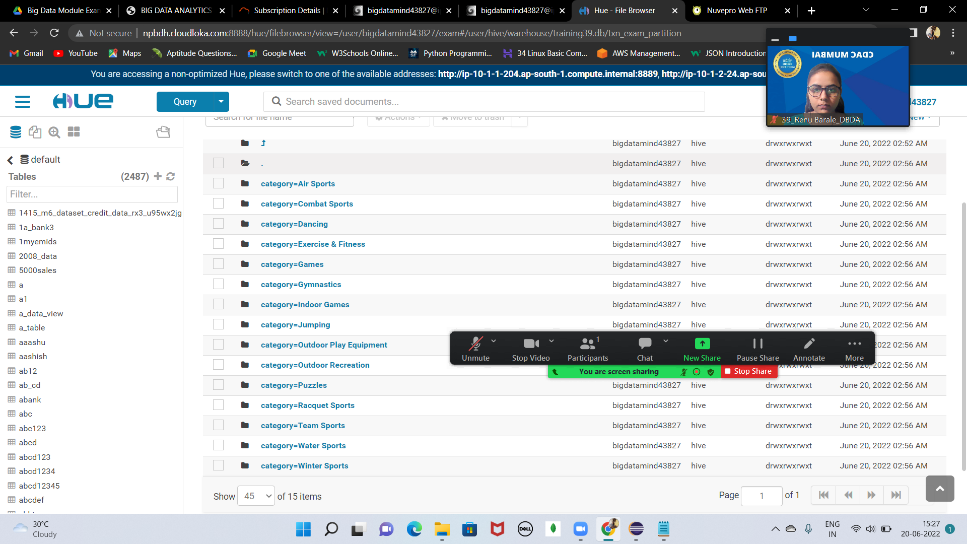
fields terminated by ','

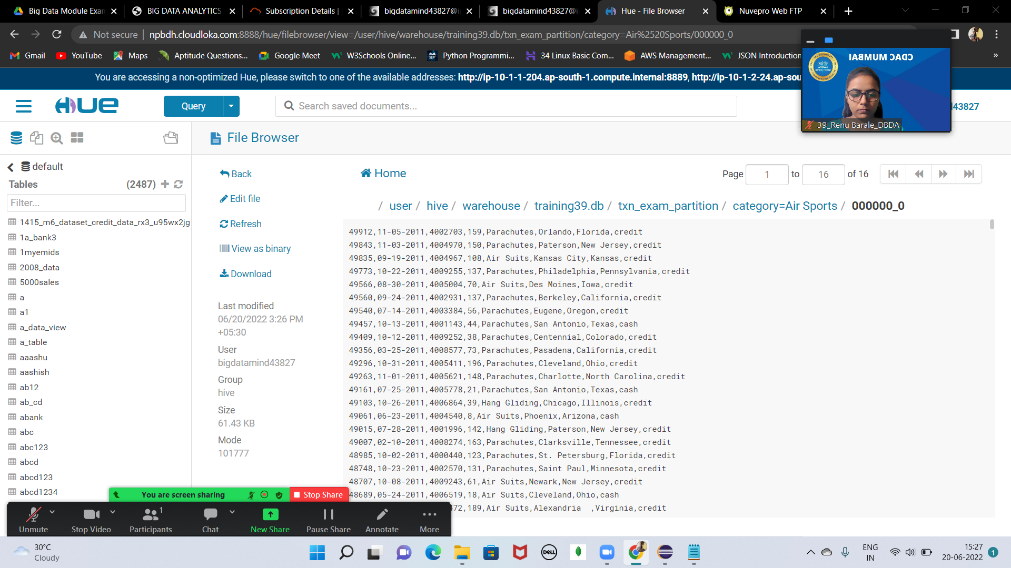
stored as textfile;

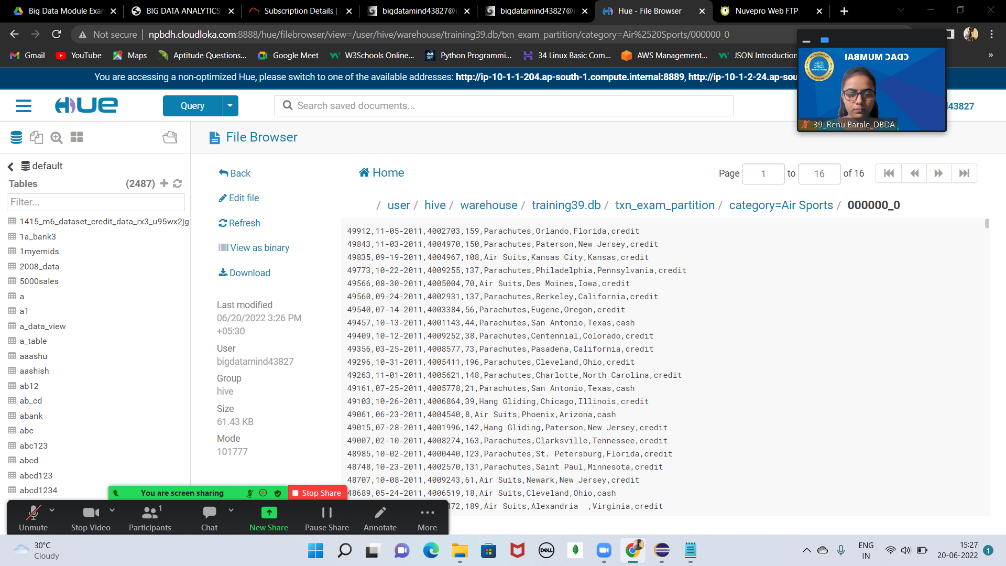
from txn\_exam tx insert into table txn\_exam\_partition PARTITION(category) select tx.txn\_id,tx.txn\_date,tx.cust\_id ,tx.amount , tx.product , tx.city ,tx.state ,

tx.spendby , tx.category DISTRIBUTE By category;









**Spark**

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()

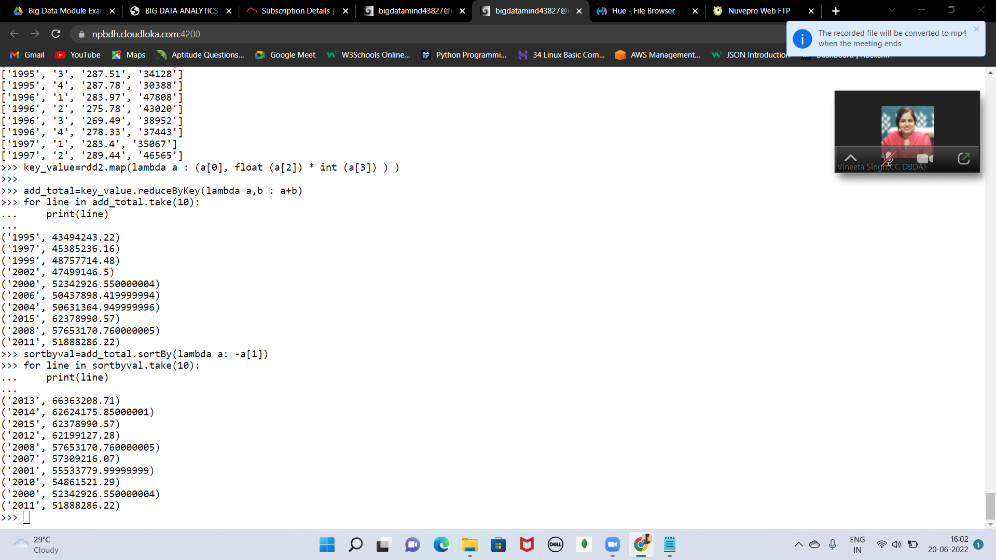
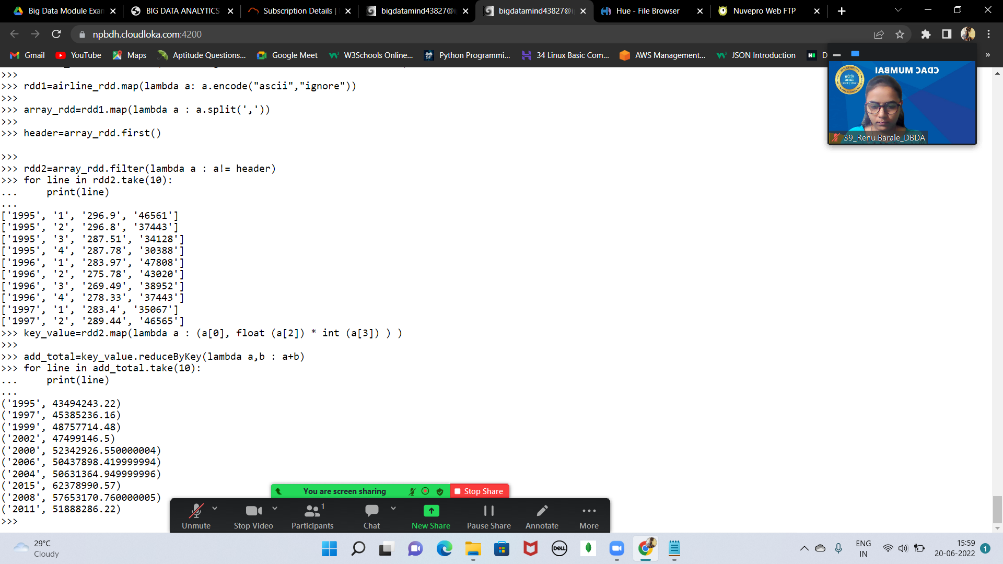
rdd2=array\_rdd.filter(lambda a : a!= header)

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)



Q1)What was the highest number of people travelled in which year?

==>

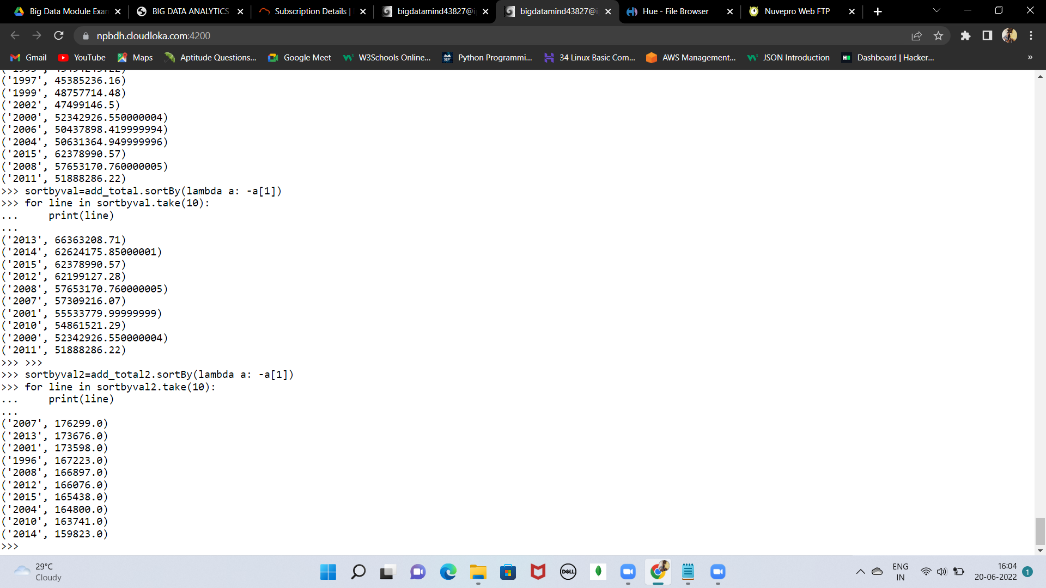
key\_value2=rdd2.map(lambda a : (a[0], float(a[3]) ))

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

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

('2007', 176299.0)





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

group)

==>

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])

