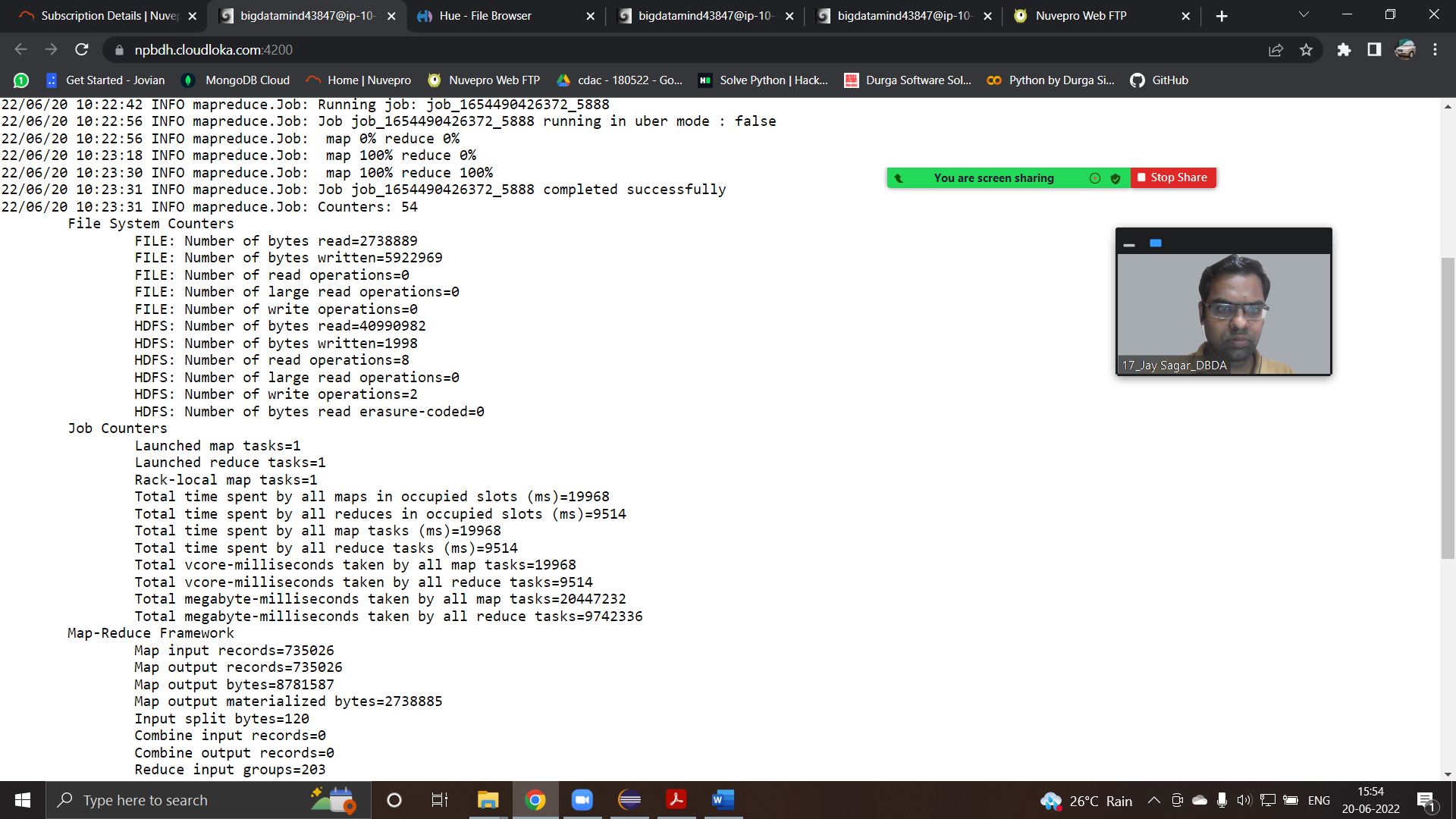
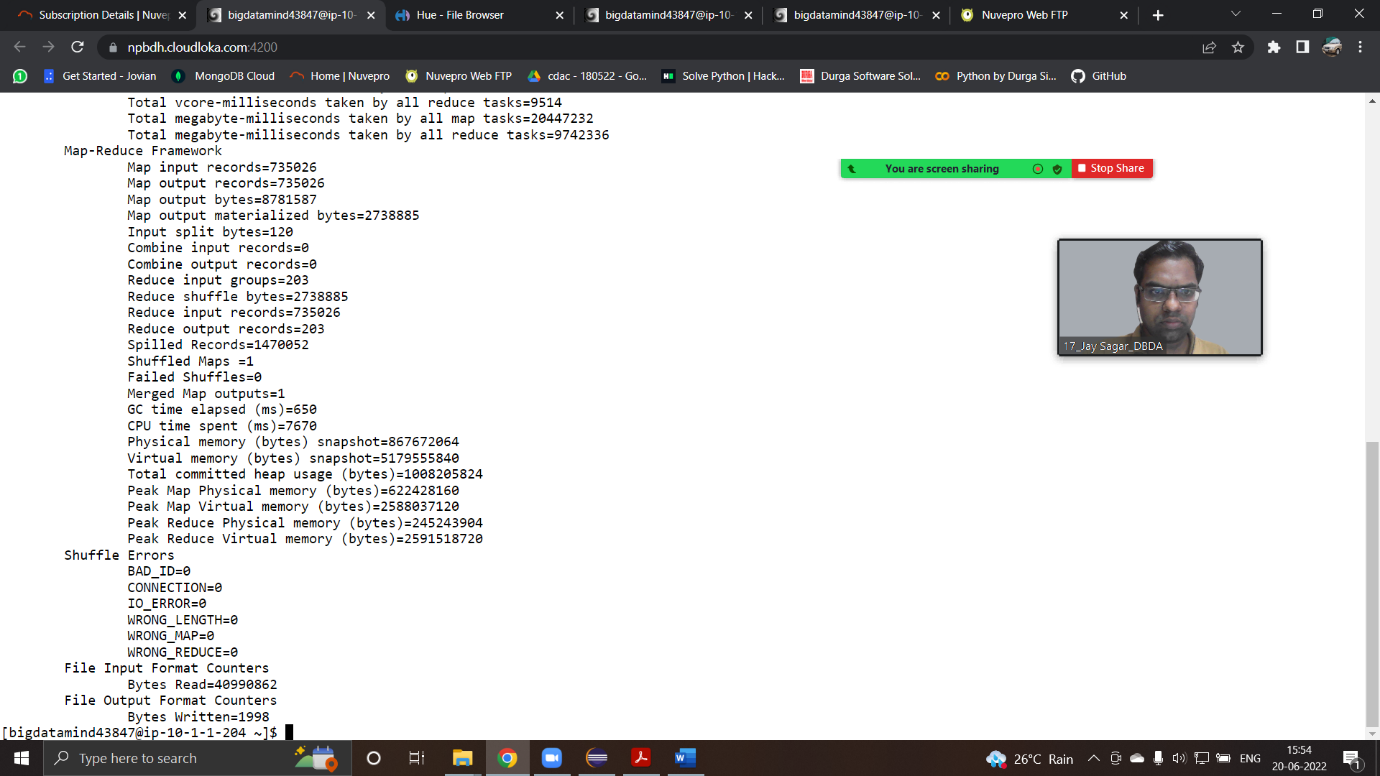
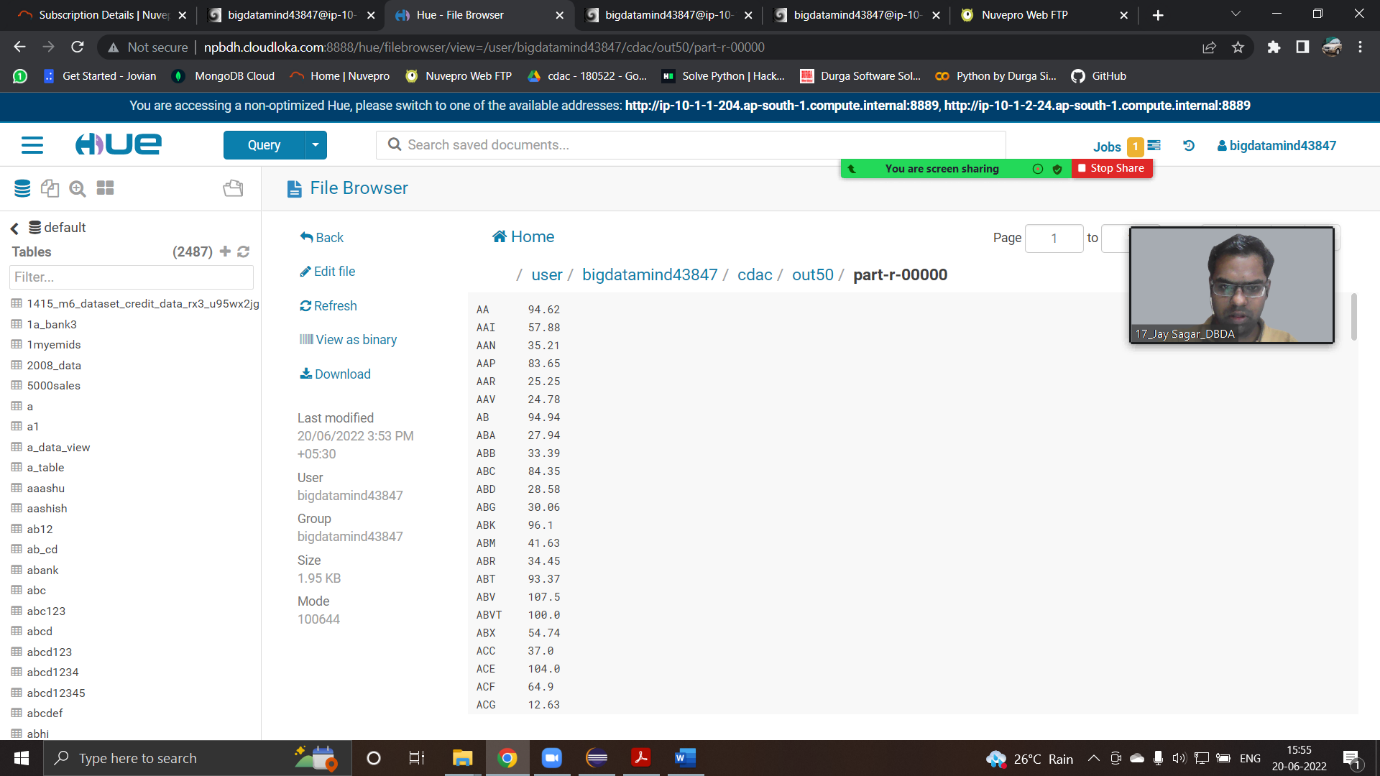
Q 1 Map Reduce

hadoop jar myjar.jar AllTimeHigh cdac/NYSE.csv cdac/out50







Q2 ) create external table customer(custno string, firstname

String, lastname string, age int,profession string)

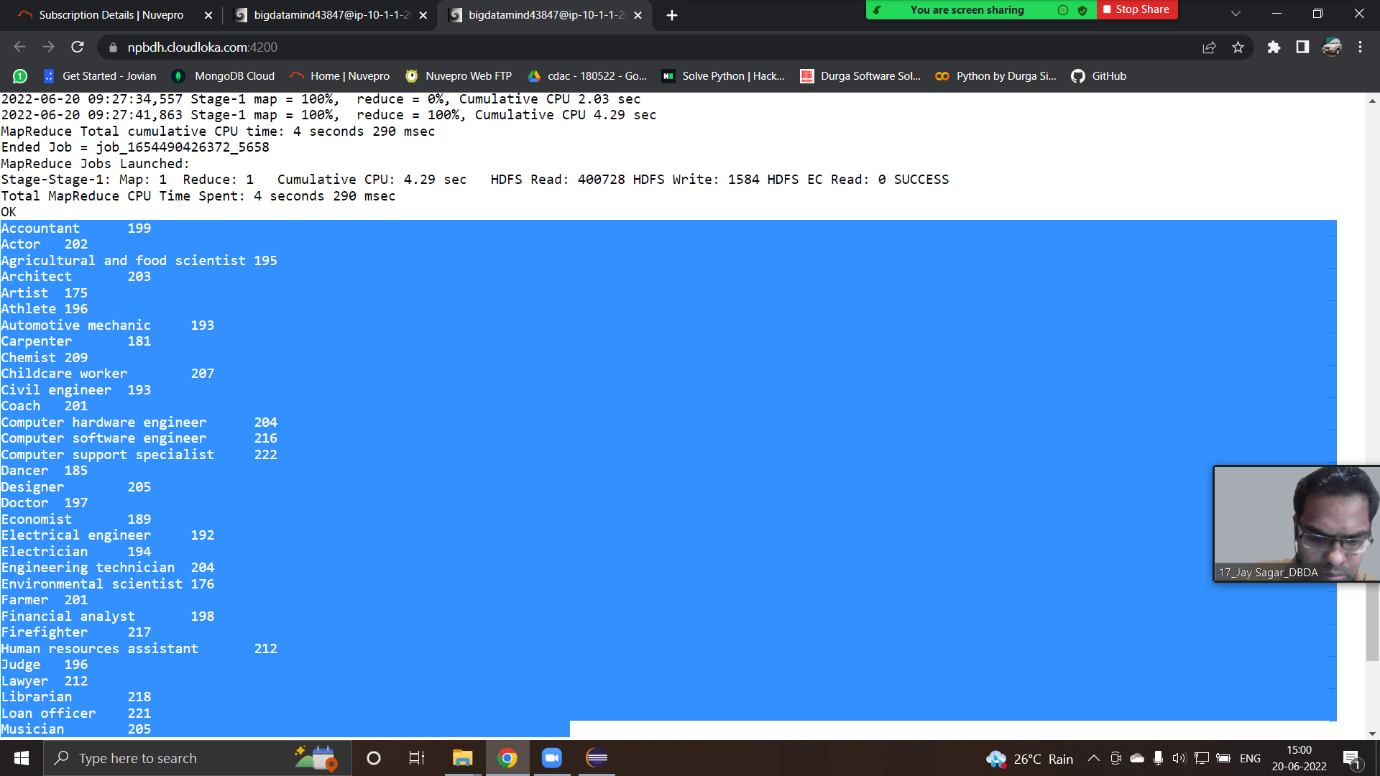
row format delimited

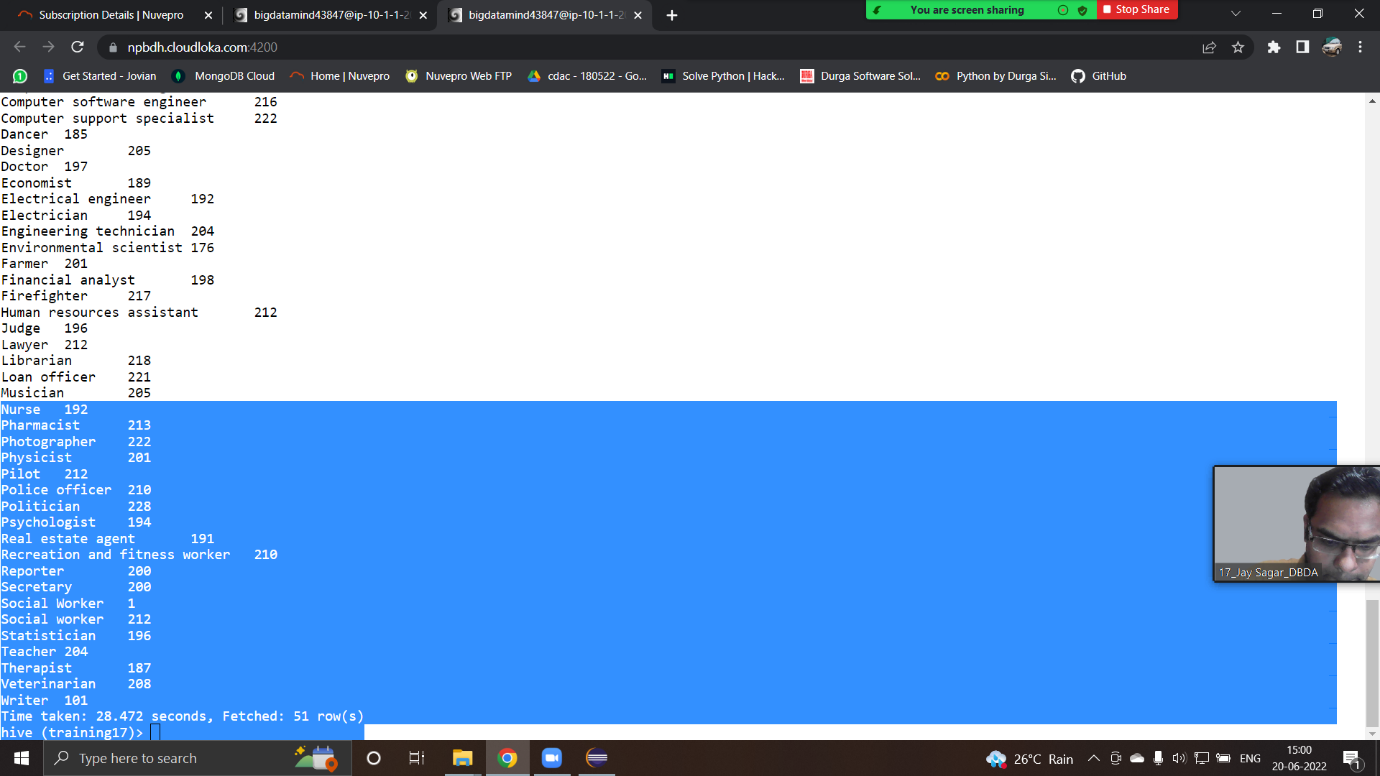
fields terminated by ','

stored as textfile

location '/user/bigdatamind43847/custs.txt';

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



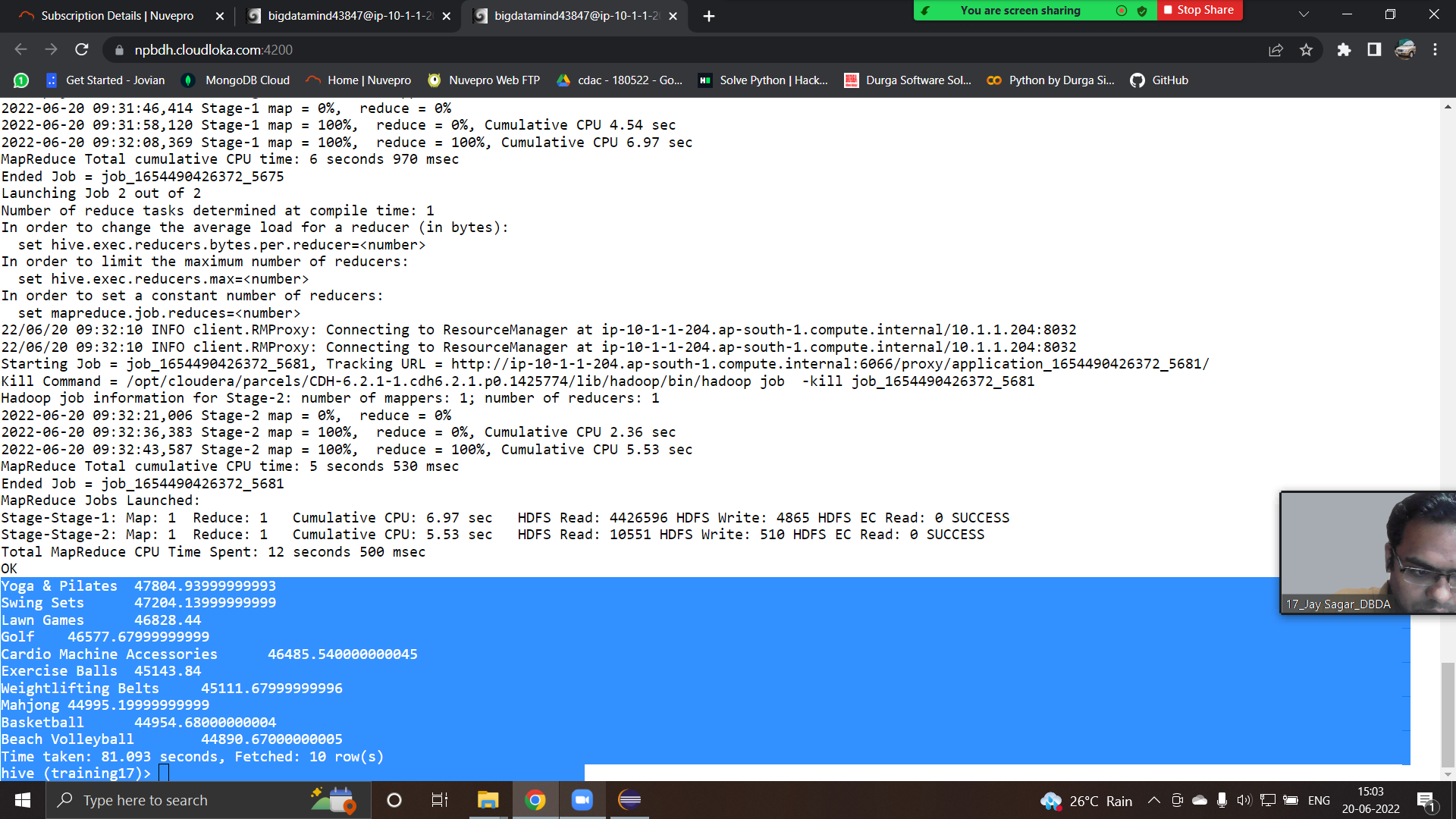


Q2 create table txnrecsByCat(txnno INT, txndate STRING, custno INT, amount DOUBLE, product STRING, city STRING, state STRING, spendby STRING) row format

delimited fields terminated by ',' stored as textfile;

LOAD DATA LOCAL INPATH 'txns1.txt' OVERWRITE INTO TABLE txnrecords;

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

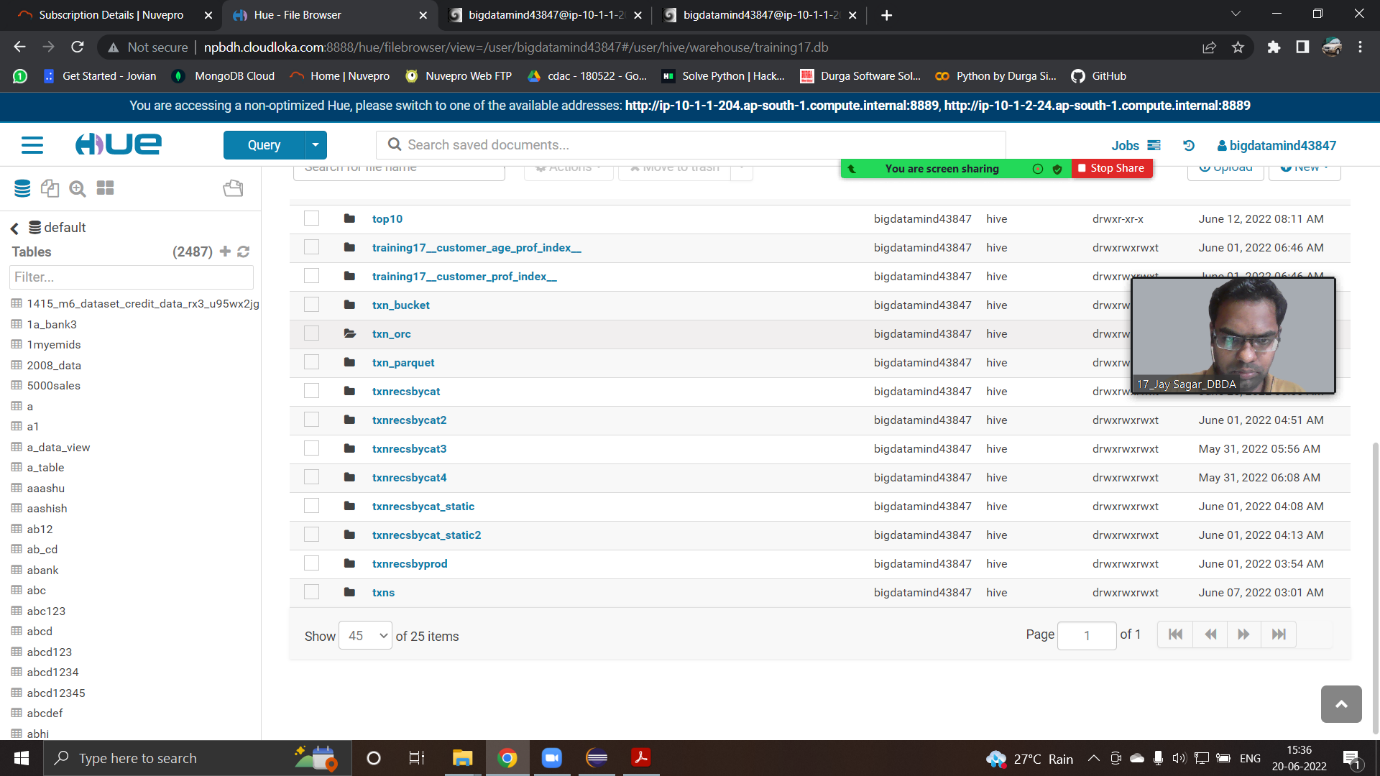


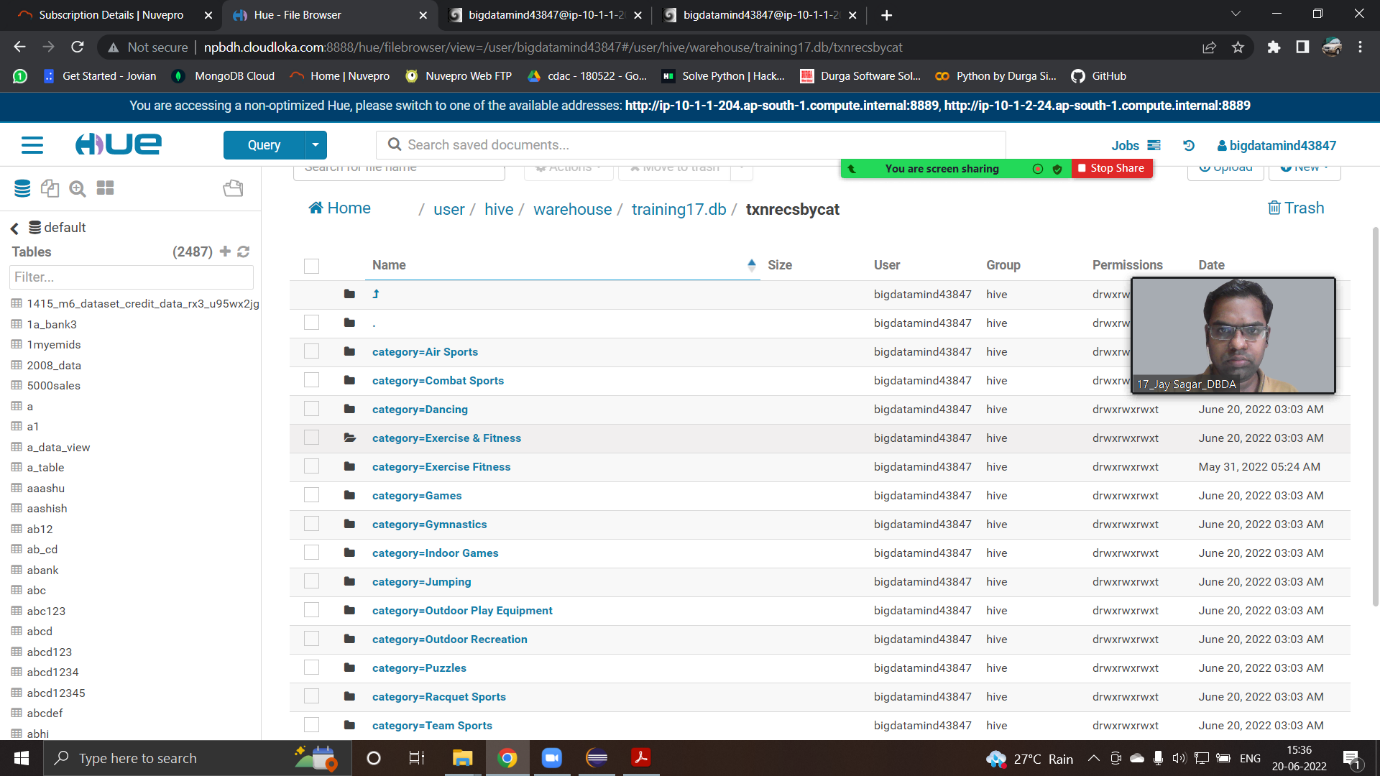
set hive.exec.dynamic.partition.mode = nonstrict;

set hive.exec.dynamic.partition = true;

INSERT OVERWRITE TABLE txnrecsByCat PARTITION(category) select txn.txnno, txn.txndate,txn.custno, txn.amount, txn.product, txn.city,txn.state,txn.spe

ndby, txn.category from txnrecords txn DISTRIBUTE By category;

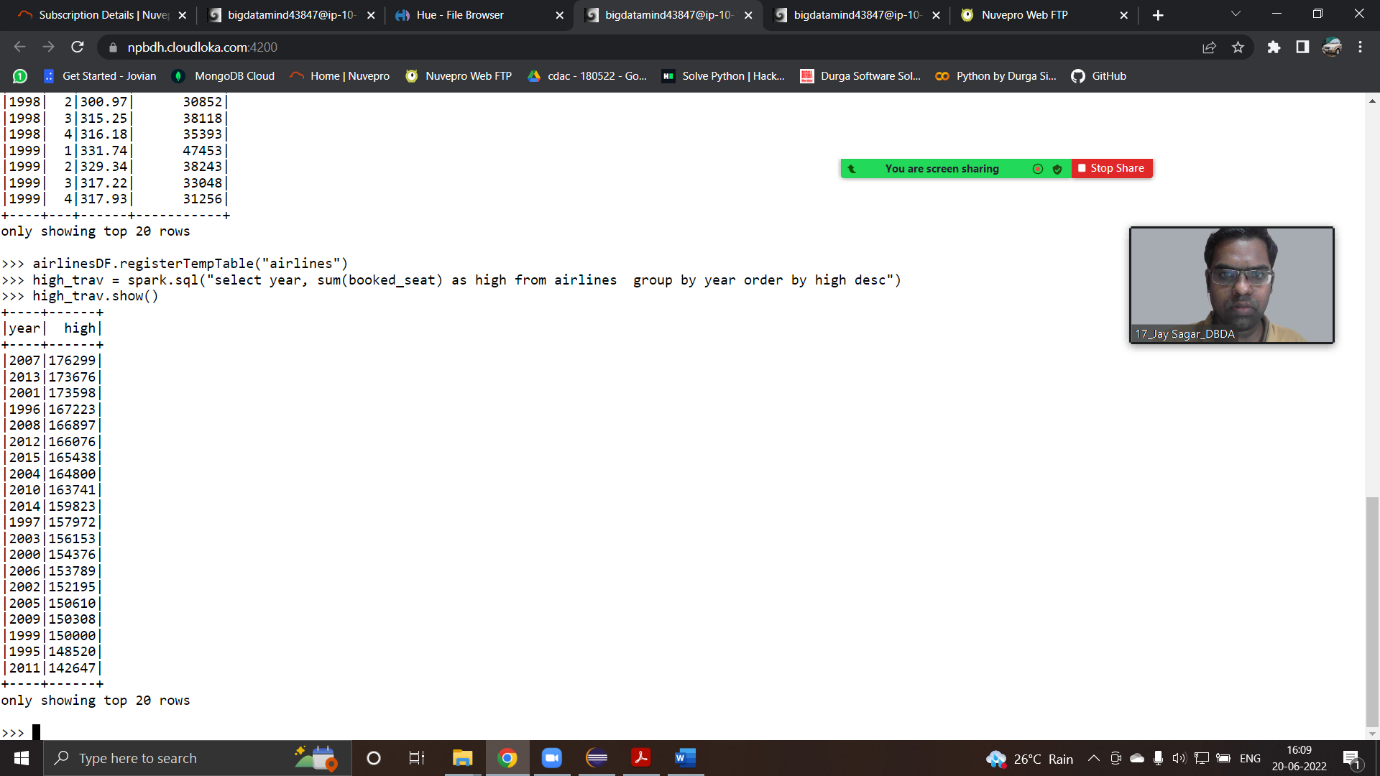




Q3

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

>>> high\_trav.show()



high\_rev = spark.sql("select year, round(sum(arps\*booked\_seat)/1000000) as in\_millions from airlines group by year order by in\_millions desc")

>>> high\_rev.show()

