**BIG DATA MODULE**

**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. Kindly Find 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

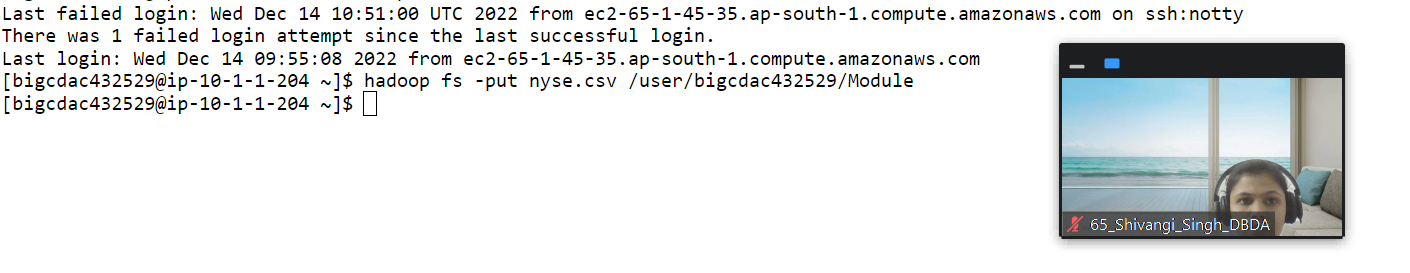
6. Intra day low price of the stock

7. Closing price of the stock

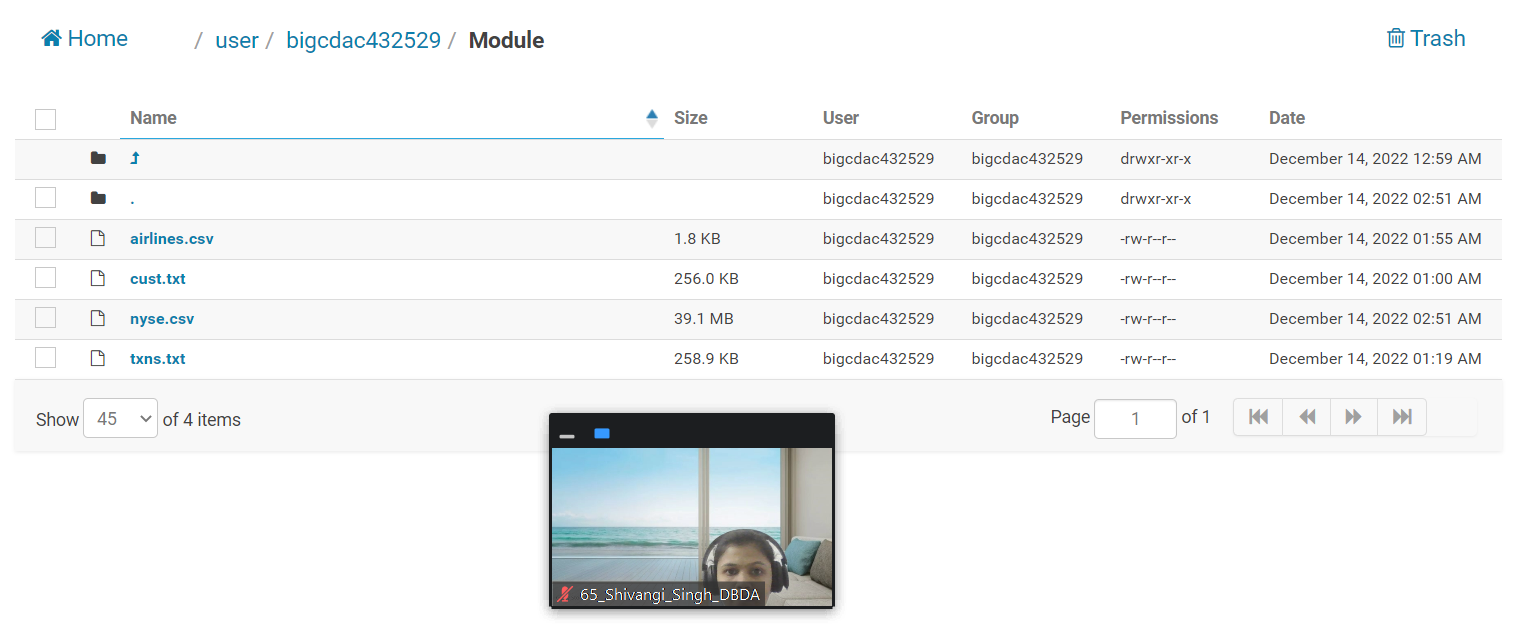
8. Total Volume of the stock on the particular day

9. Adjustment Closing price of the stock

hadoop fs -put nyse.csv /user/bigcdac432529/Module







Field Separator – comma

Question 2 : Find all time High price for each stock

**package** cdac18;

**import** java.io.IOException;

**import** java.io.\*;

**import** org.apache.hadoop.io.Text;

**import** java.util.StringTokenizer;

**import** org.apache.hadoop.conf.Configuration;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.LongWritable;

**import** org.apache.hadoop.io.DoubleWritable;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapreduce.Job;

**import** org.apache.hadoop.mapreduce.Mapper;

**import** org.apache.hadoop.mapreduce.Reducer;

**import** org.apache.hadoop.fs.\*;

**import** org.apache.hadoop.conf.\*;

**import** org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**public** **class** AllTime {

**public** **static** **class** Mapclass **extends** Mapper<LongWritable, Text, Text, DoubleWritable>{

**public** **void** map(LongWritable key, Text value, Context context)

{

**try**

{

String [] str= value.toString().split(",");

**double** high= Double.*parseDouble*(str[4]);

context.write(**new** Text(str[1]), **new** DoubleWritable(high));

}

**catch**(Exception e)

{

System.***out***.println(e.getMessage());

}

}

}

**public** **static** **class** Reduceclass **extends** Reducer<Text,IntWritable,Text,DoubleWritable> {

**private** DoubleWritable result= **new** DoubleWritable();

**public** **void** reduce(Text key, Iterable<DoubleWritable>value, Context context) **throws** IOException, InterruptedException

{

**double** max =0.0;

**for** (DoubleWritable val : value)

{

**if**(val.get()>max)

{

max=val.get();

}

}

result.set(max);

context.write(key,result);

}

}

**public** **static** **void** main(String[] args) **throws** Exception {

Configuration conf = **new** Configuration();

Job job = Job.*getInstance*(conf, "Al time high ");

job.setJarByClass(AllTime.**class**);

job.setMapperClass(Mapclass.**class**);

job.setReducerClass(Reduceclass.**class**);

job.setNumReduceTasks(1);

//job.setMapOutputKeyClass(Text.class);

//job.setMapOutputValueClass(DoubleWritable.class);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(DoubleWritable.**class**);

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

System.*exit*(job.waitForCompletion(**true**) ? 0 : 1);

}

}

hadoop jar mynew.jar cdac18/AllTime /user/bigcdac432529/Module/nyse.csv /user/bigcdac432529/Module/Output1

[15 marks]

Hive

Please find the customer data set.

cust id

firstname

lastname

age

profession

OUTPUT-

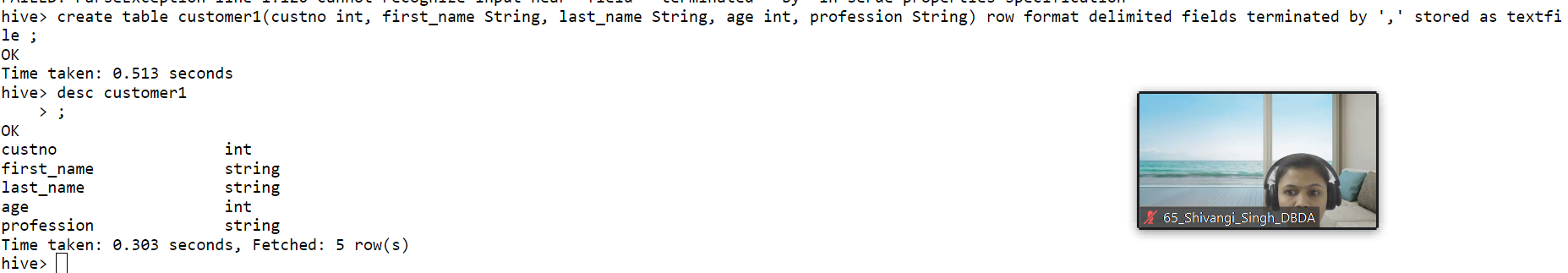
create table customer1(custno int, first\_name String, last\_name String, age int, profession String) row format delimited fields terminated by ',' stored as textfi

le ;

hive> load data local inpath 'cust.txt' overwrite into table customer1;

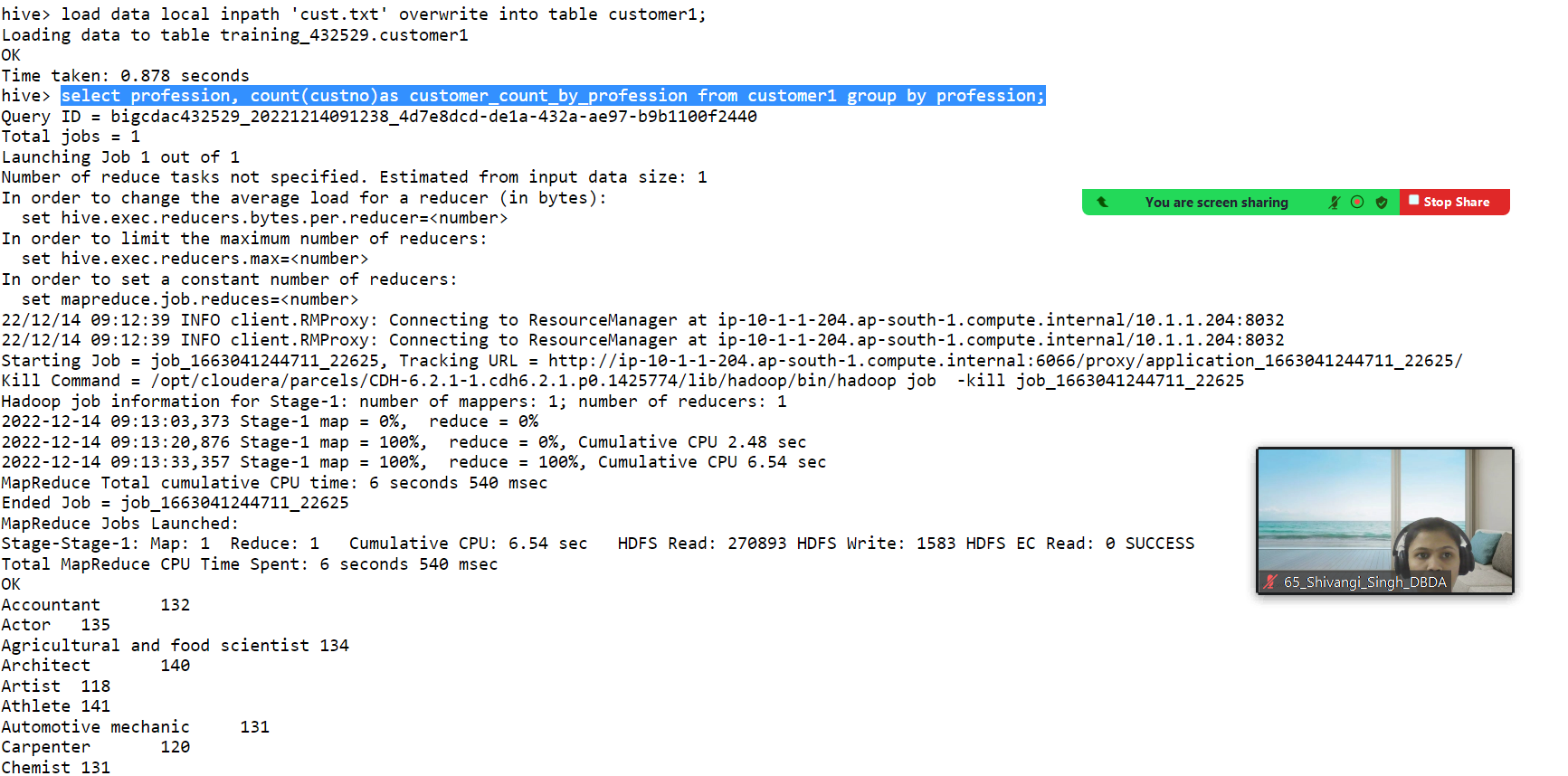
Loading data to table training\_432529.customer1

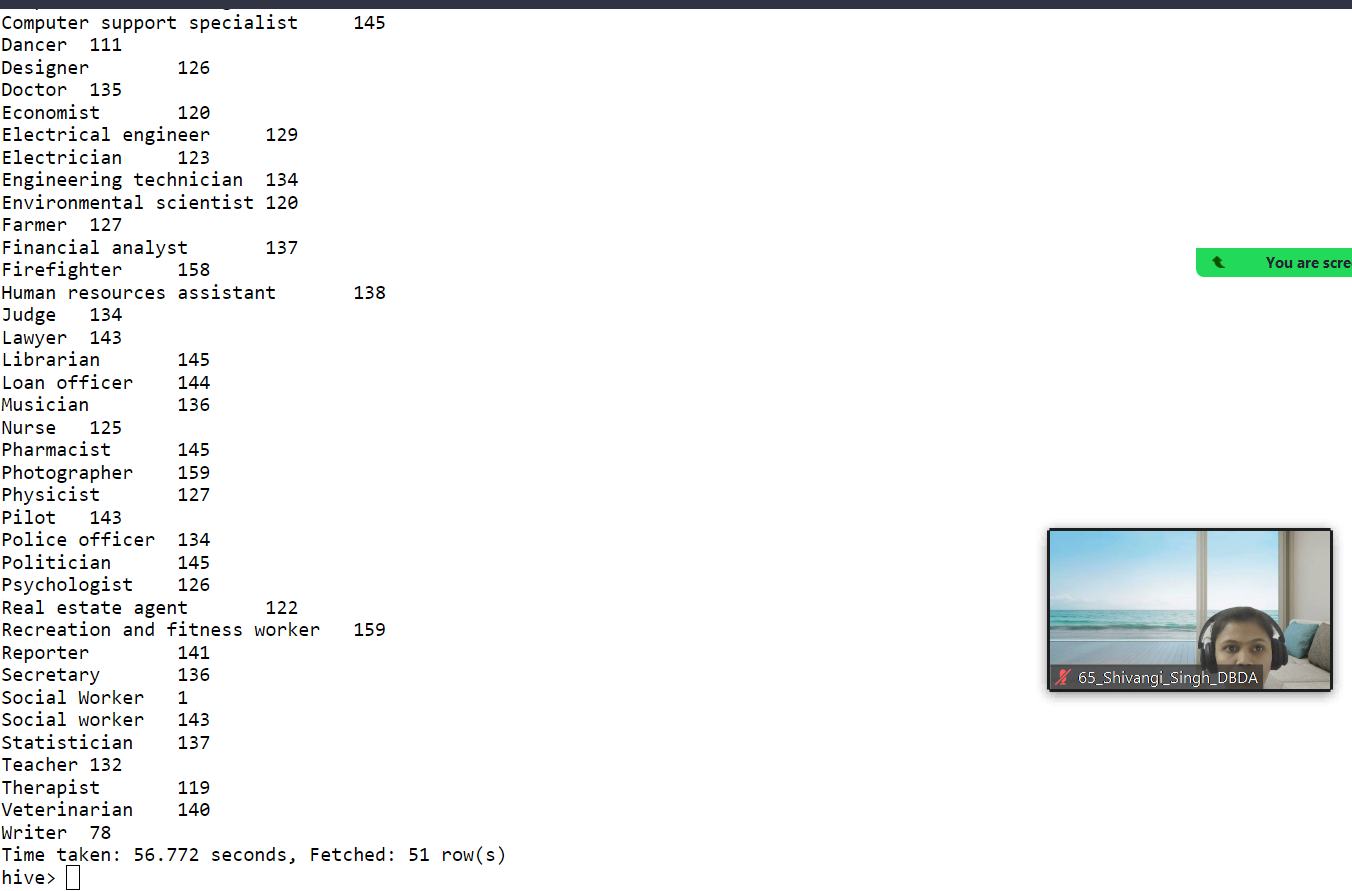
OK



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

select profession, count(custno)as customer\_count\_by\_profession from customer1 group by profession;





Please find the sales data set.

txn id

txn date

cust id

amount

category

product

city

state

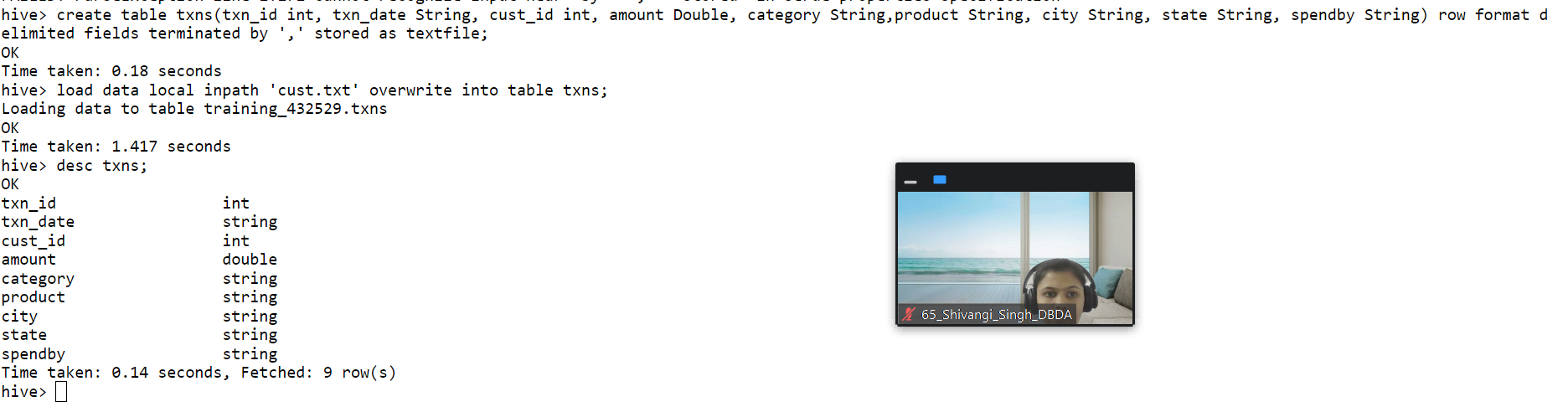
spendby

create table txns(txn\_id int, txn\_date String, cust\_id int, amount Double, category String,product String, city String, state String, spendby String) row format d

elimited fields terminated by ',' stored as textfile;

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

Loading data to table training\_432529.txns

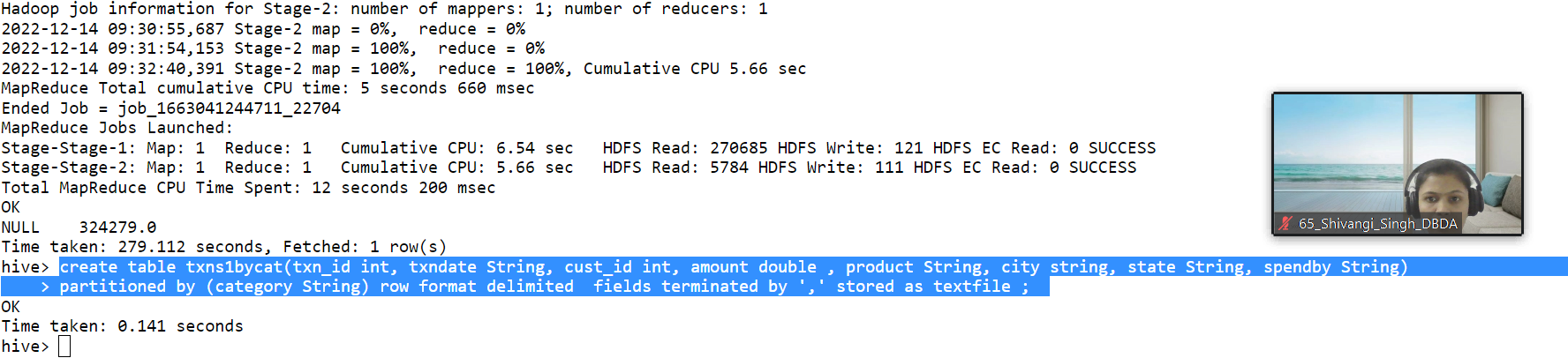


1. Write a program to find the top 10 products sales wise

1. Write a program to create partiioned table on category

create table txns1bycat(txn\_id int, txndate String, cust\_id int, amount double , product String, city string, state String, spendby String)

> partitioned by (category String) row format delimited fields terminated by ',' stored as textfile ;



QUESTION 3 [15 marks]

PySpark

Please find the AIRLINES data set

Year

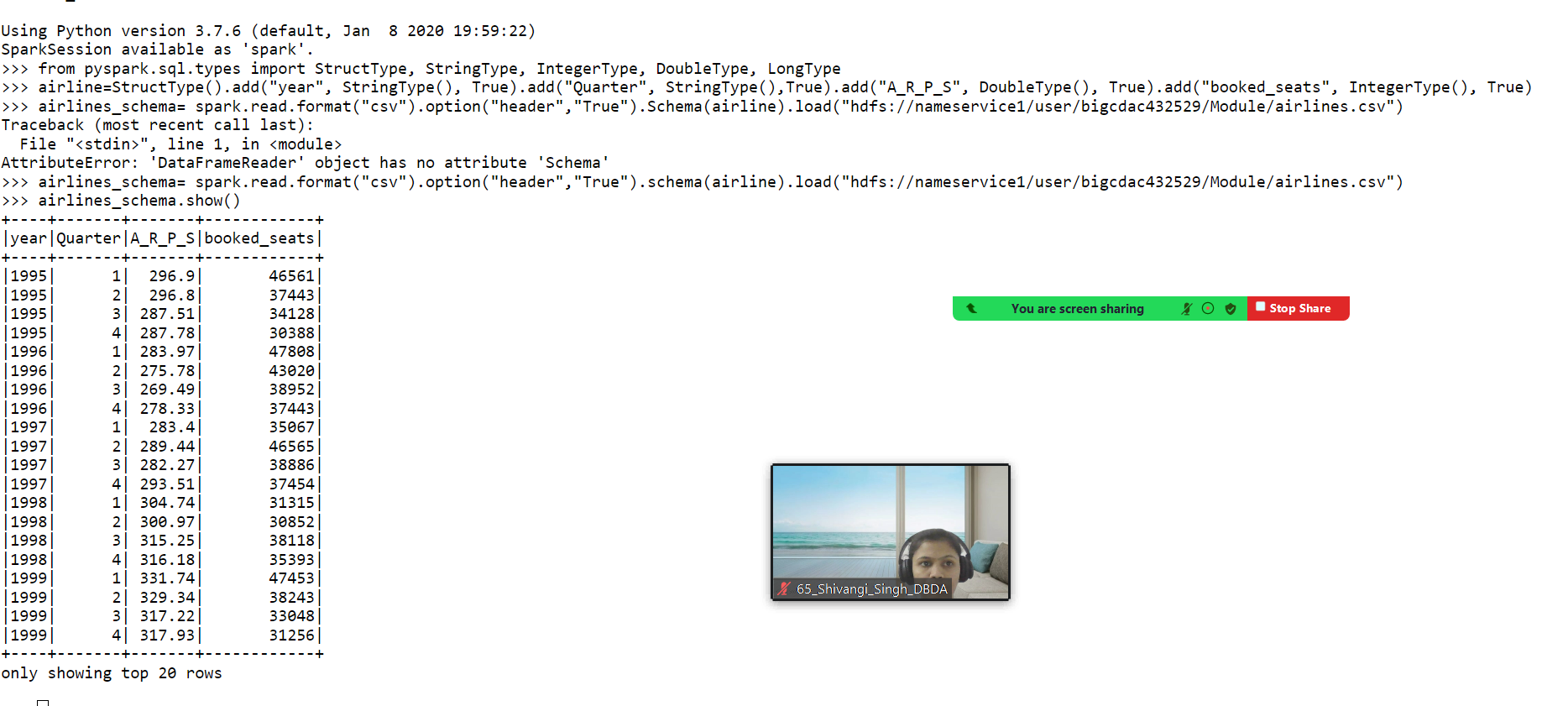
Quarter

Average revenue per seat

Total number of booked seats

airline=StructType().add("year", StringType(), True).add("Quarter", StringType(),True).add("A\_R\_P\_S", DoubleType(), True).add("booked\_seats", IntegerType(), True)

airlines\_schema= spark.read.format("csv").option("header","True").schema(airline).load("hdfs://nameservice1/user/bigcdac432529/Module/airlines.csv")



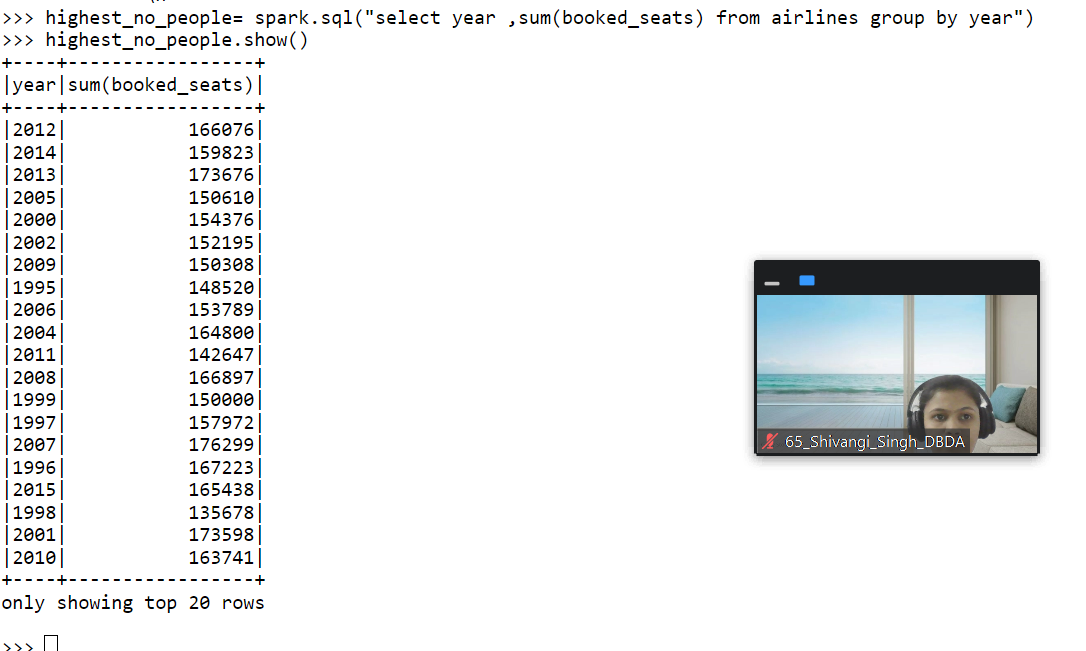
1) What was the highest number of people travelled in which

year?

airlines\_schema.registerTempTable("airlines")

highest\_no\_people= spark.sql("select year ,sum(booked\_seats) from airlines group by year")

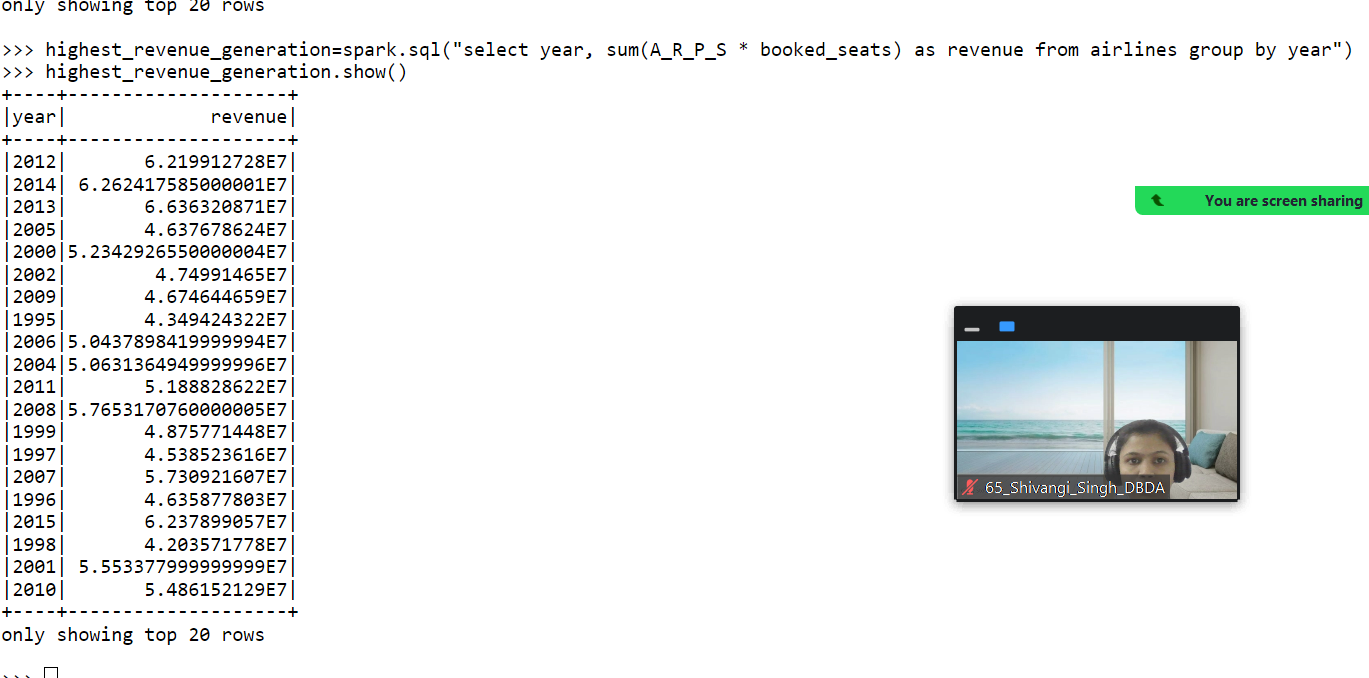
highest\_no\_people.show()



2) Identifying the highest revenue generation for which year

highest\_revenue\_generation=spark.sql("select year, sum(A\_R\_P\_S \* booked\_seats) as revenue from airlines group by year")

highest\_revenue\_generation.show()



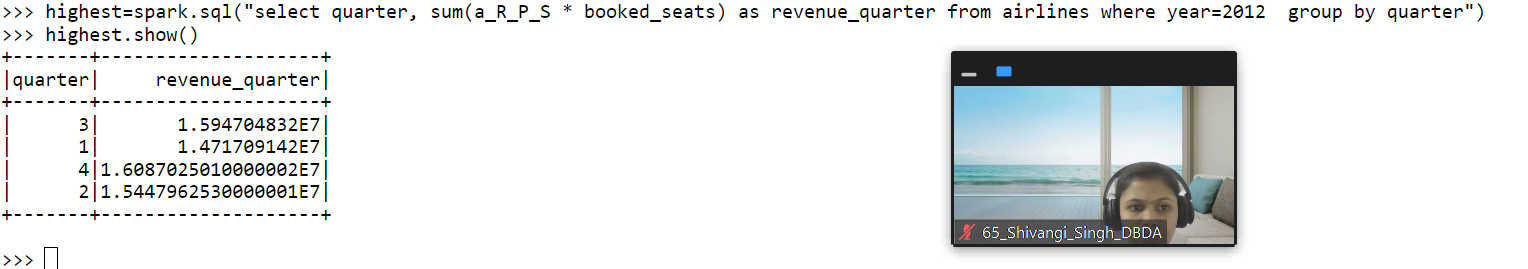
**o./p – 2012**

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

group)

highest=spark.sql("select quarter, sum(a\_R\_P\_S \* booked\_seats) as revenue\_quarter from airlines where year=2012 group by quarter")

highest.show()



**o/p – year 2012 highest revenue generated for the quarter 3**