CDAC MUMBAI

PG-DBDA SEP 2022 BATCH KHARGHAR

MODULE: BIG DATA ANALYTICS

DATE : 14TH DEC, 2022

MARKS : 40 MARKS

Please create a doc/txt/pdf file with 12 digits student id, which will contain the code along with the screenshots of the output or result. While taking the screenshot make sure that you are visible in all the images.

Q1.

MapReduce

Problem Statement **[10 marks]**

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

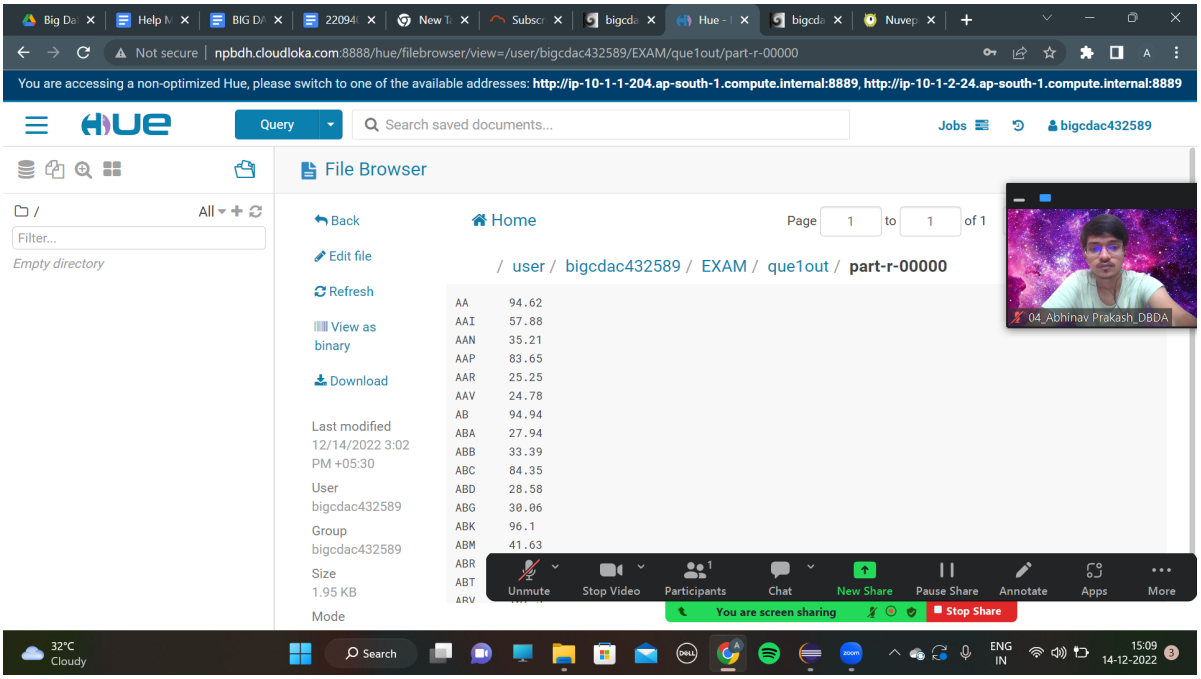
9. Adjustment Closing price of the stock

Field Separator – comma

<u>Question 2 : F</u>ind all time High price for each stock [15 marks]

```
import java.io.IOException;
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.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class allTimeHigh {
        public static class MapClass extends Mapper<LongWritable, Text, Text, DoubleWritable>{
                public void map(LongWritable key, Text value,Context context)throws
IOException,InterruptedException {
                        String[] str = value.toString().split(",");
                        String stk_id = str[1];
                        double high = Double.parseDouble(str[4]);
                        context.write(new Text(stk_id), new DoubleWritable(high));
                }

 }

  public static class ReduceClass  extends Reducer<Text,DoubleWritable,Text,DoubleWritable> {

          public void reduce(Text key, Iterable<DoubleWritable> values,Context context)throws
IOException,InterruptedException  {
                double max = 0;
                for (DoubleWritable val:values) {
                        if (val.get() > max) {
                                max = val.get();
                        }
                }
                context.write(key, new DoubleWritable(max));
          }
  }

public static void main(String[] args) throws Exception {
   Configuration conf = new Configuration();
   Job job = Job.getInstance(conf, "All Time High");
   job.setJarByClass(allTimeHigh.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);
  }
}
```



Hive

Please find the customer data set.

cust id

firstname

lastname

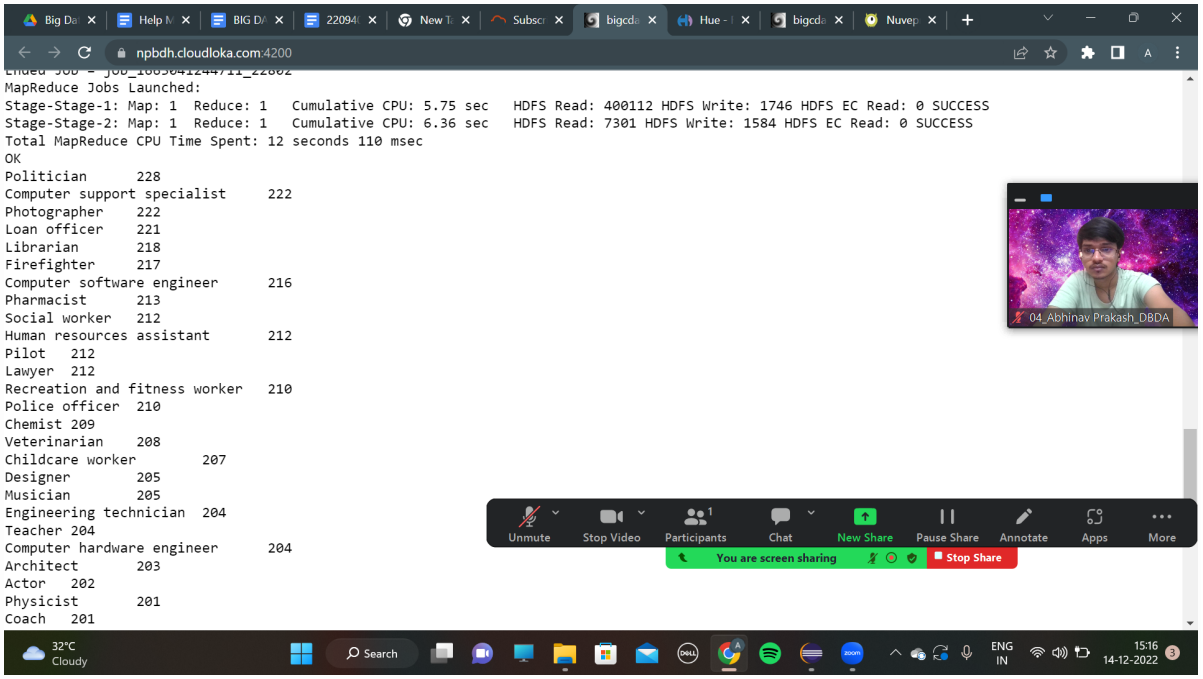
age

profession

```
CREATE TABLE customer (
custno int,
firstname String,
lastname String,
age  int,
profession String
)
Row format delimited
Fields terminated by ‘,’
Stored as textFile;

Load data local inpath ‘custs.txt’ overwrite into table customer;
```

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

```
SELECT profession , count(DISTINCT custno) AS num_of_cust FROM
customer GROUP BY profession ORDER BY num_of_cust DESC;
```



Please find the sales data set.

txn id

txn date

cust id

amount

category

product

city

state

spendby

Creating Table:---

CREATE TABLE sales (

txn_id int,

txn_date String,

Custno int,

amount float,

category String,

product String,

city String,

state String,

spendby String

)

Row format delimited

Fields terminated by ‘,’

Stored as textFile;

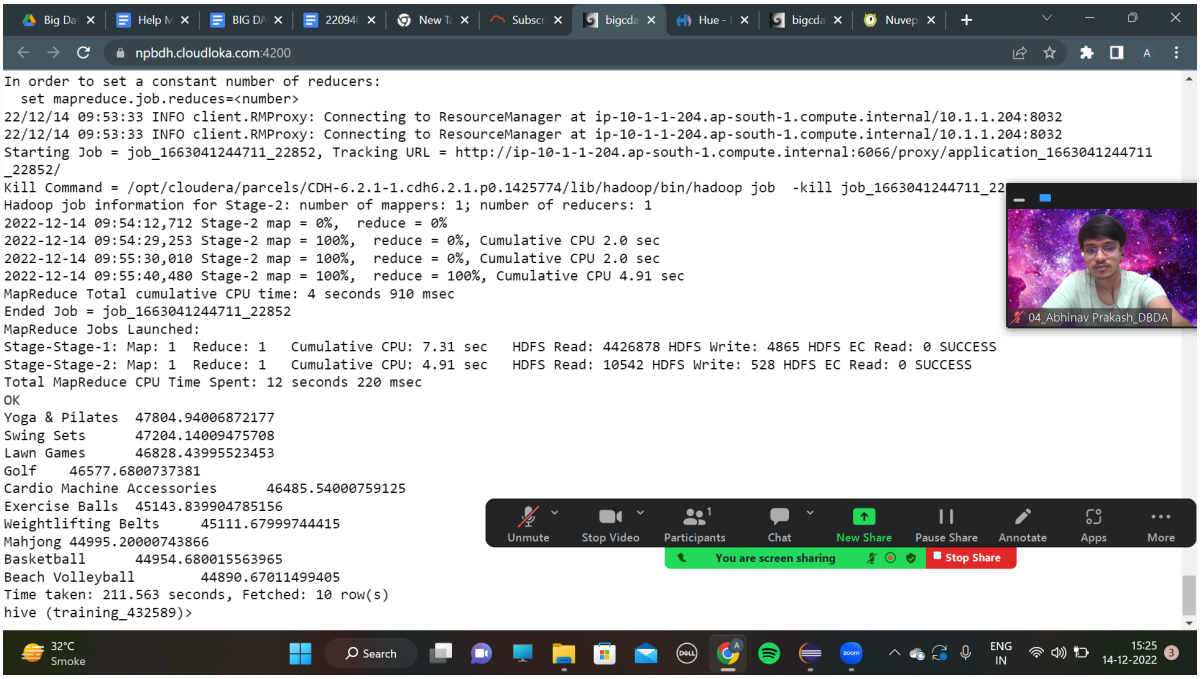
Loading Data into table:----------

Load data local inpath ‘txns1.txt’ overwrite into table sales;

---

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

SELECT product, sum(amount) as total_sales from sales GROUP BY product ORDER BY total_sales DESC LIMIT 10;



---

3) Write a program to create partitioned table on category

First set dynamic partitioning true:---
Set hive.exec.dynamic.partition = true;
Set hive.exec.dynamic.partition.mode = nonstrict;

Create table partitioned_txns (

txn_id int,

txn_date String,

Custno int,

amount float,

product String,

city String,

state String,

spendby String

)

PARTITIONED BY (category String)

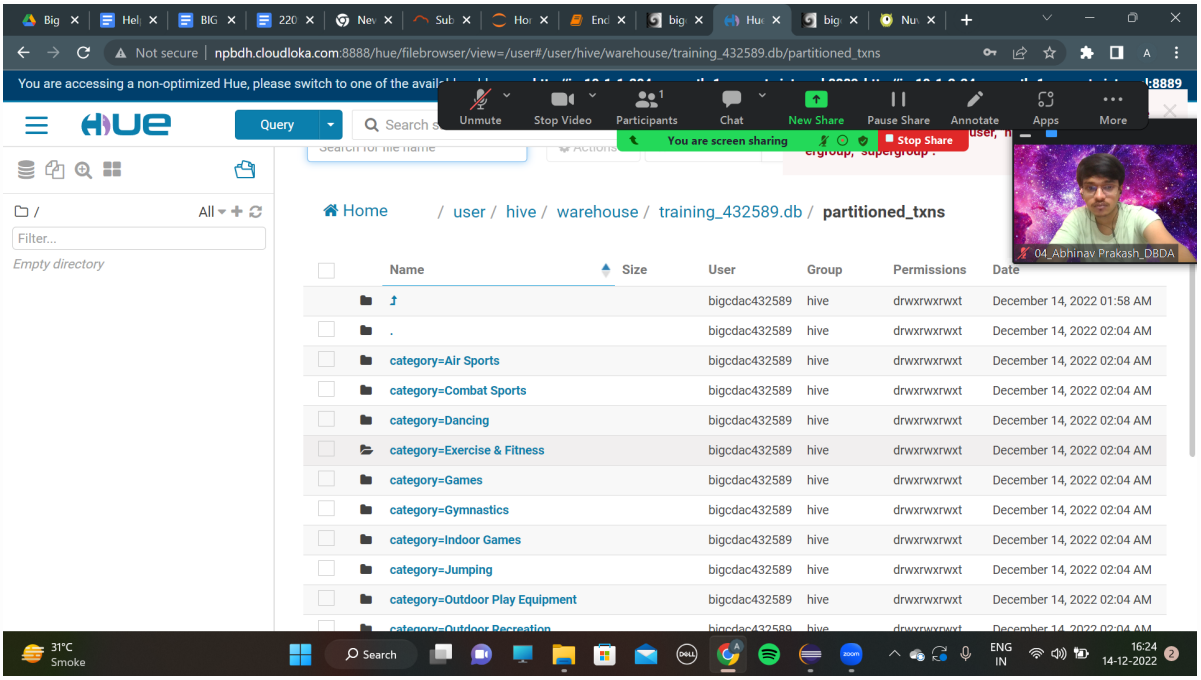
Row format delimited

Fields terminated by ‘,’

stored as textFile;

<u>Inserting Data:</u>--------

INSERT OVERWRITE TABLE partitioned_txns PARTITION(category)
SELECT txn_id , txn_date, custno,amount,product,city,state,spendby,category FROM sales
DISTRIBUTE BY category;



QUESTION 3 [15 marks]
PySpark

Please find the AIRLINES data set

Year

Quarter

Average revenue per seat

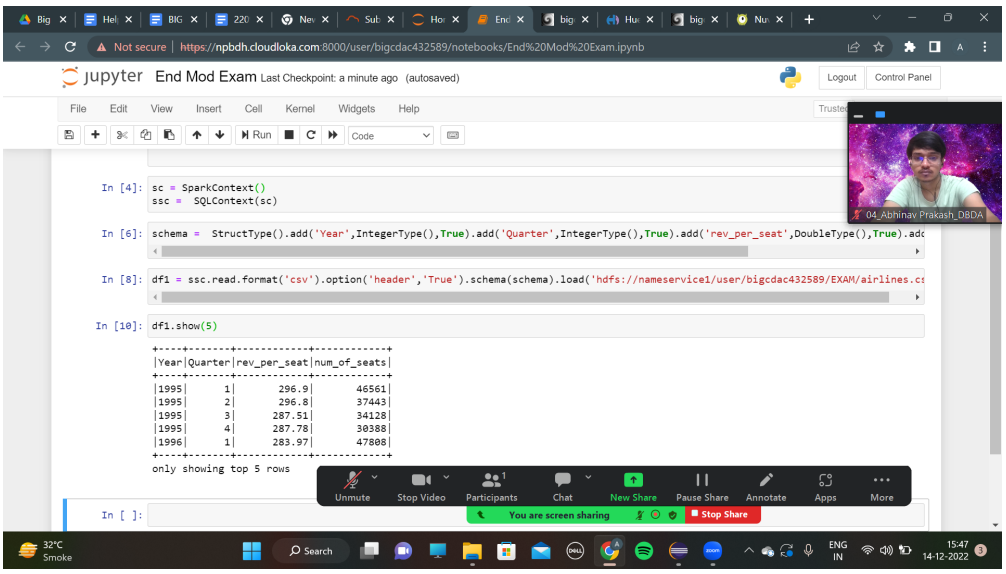
Total number of booked seats

SOLUTION USING DATAFRAMES.

schema = StructType().add('Year',IntegerType(),True).add('Quarter',IntegerType(),True).add('rev_per_seat',DoubleType(),True).add('num_of_seats',IntegerType(),True)

df1 = ssc.read.format('csv').option('header','True').schema(schema).load('hdfs://nameservice1/user/bigcdac432589/EXAM/airlines.csv')

df1.registerTempTable('airlines')

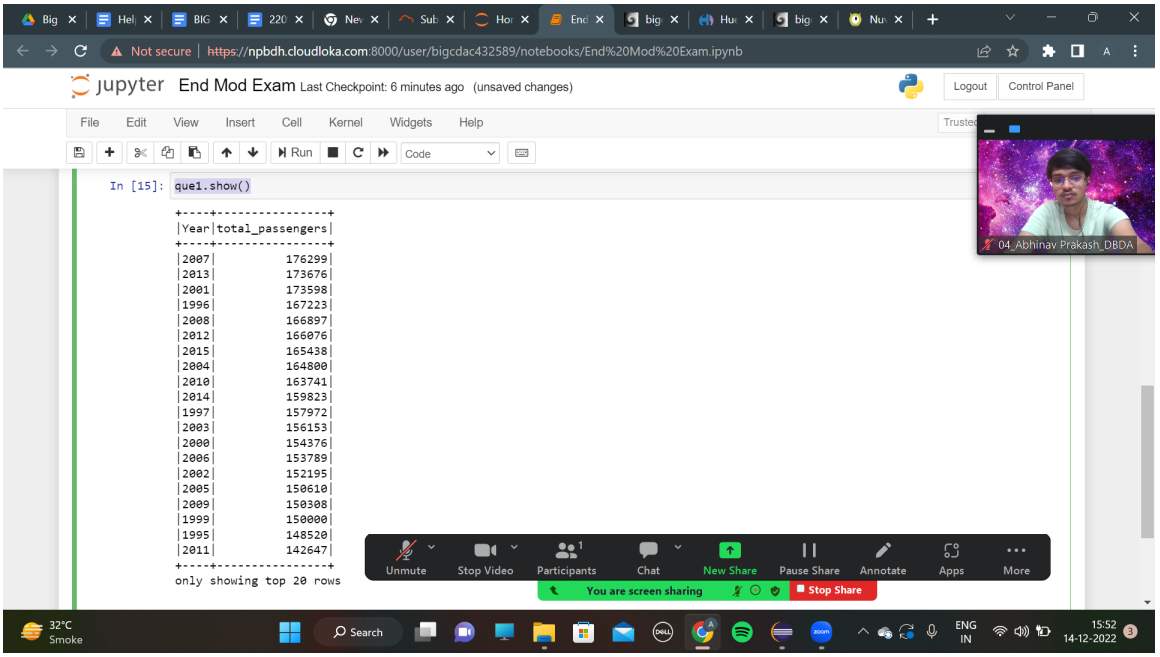


1) What was the highest number of people travelled in which

year?

que1 = ssc.sql('SELECT Year, sum(num_of_seats) as total_passengers FROM airlines GROUP BY Year ORDER BY total_passengers DESC')

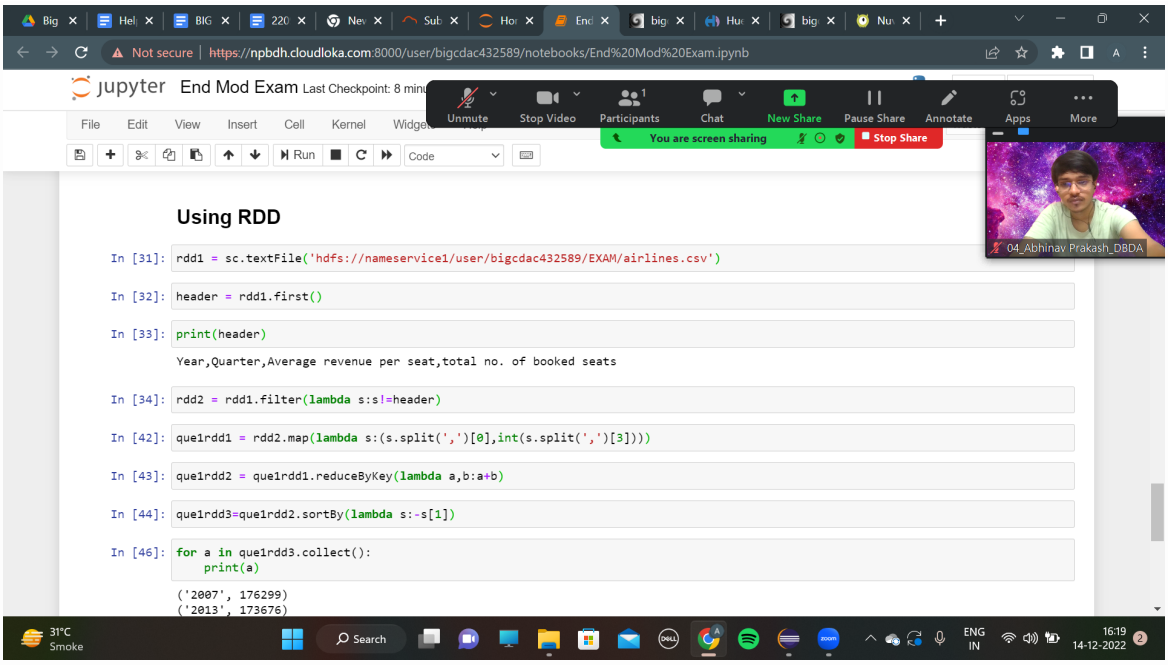
que1.show()




```
In [16]: que1.first()

Out[16]: Row(Year=2007, total_passengers=176299)
```

Highest Number of people traveled in 2007 with a total of 1,76,299 passengers.

Solution Using RDD

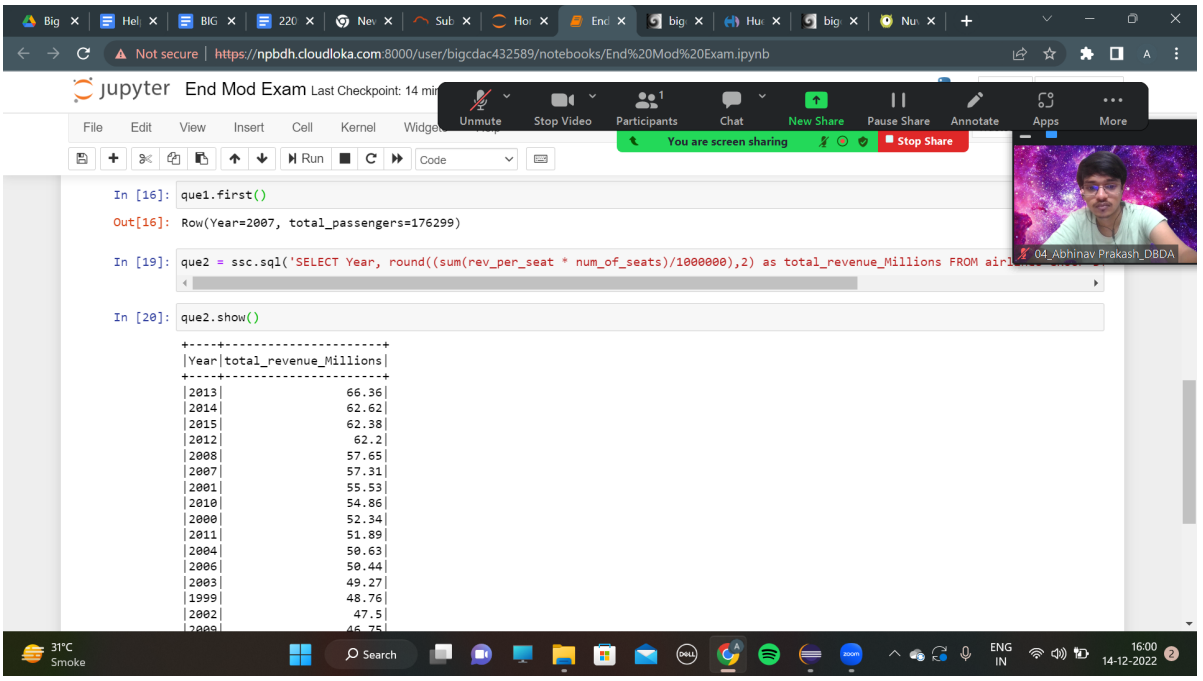



```
In [47]: que1rdd3.first()

Out[47]: ('2007', 176299)
```

2) Identifying the highest revenue generation for which year

que2 = ssc.sql('SELECT Year, round((sum(rev_per_seat * num_of_seats)/1000000),2) as total_revenue_Millions FROM airlines GROUP BY Year ORDER BY total_revenue_Millions DESC')

que2.show()

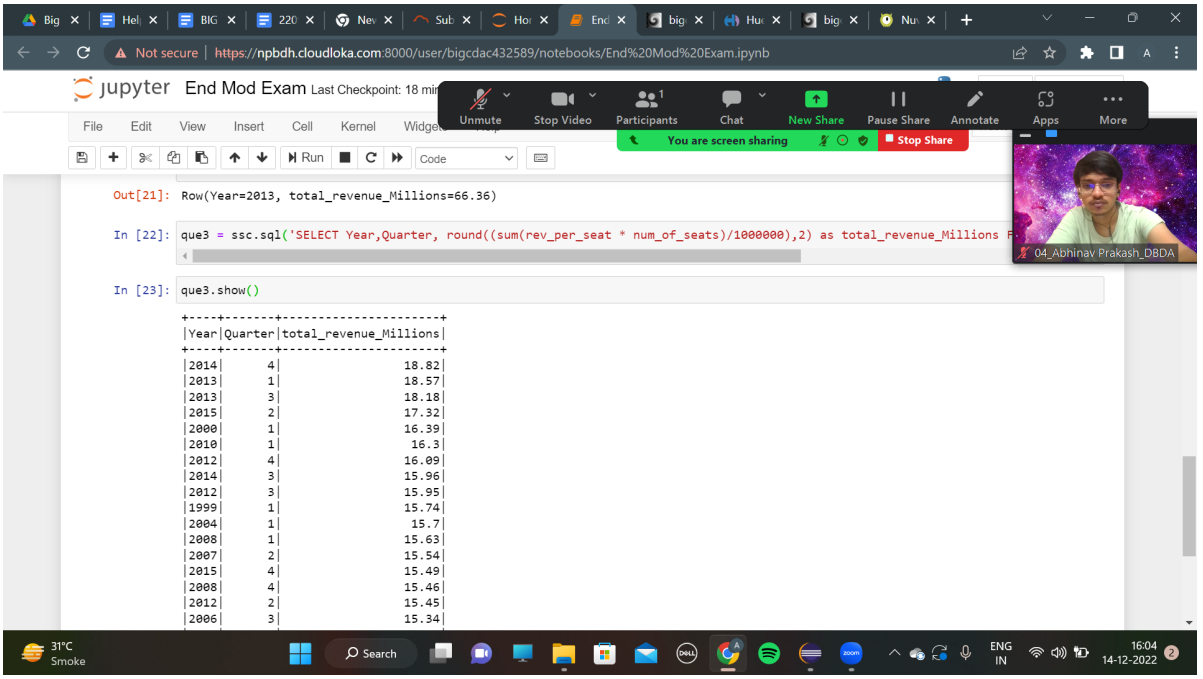



```
In [21]: que2.first()

Out[21]: Row(Year=2013, total_revenue_Millions=66.36)
```

Highest Revenue was generated in 2013 with a total of 66.36 Million.

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

que3 = ssc.sql('SELECT Year,Quarter, round((sum(rev_per_seat * num_of_seats)/1000000),2) as total_revenue_Millions FROM airlines GROUP BY Year,Quarter ORDER BY total_revenue_Millions DESC')




```
In [24]: que3.first()
Out[24]: Row(Year=2014, Quarter=4, total_revenue_Millions=18.82)
```

Highest Revenue was generated in **2014's fourth** Quarter with a total sum of 18.82 Millions.