**HIVE =>**

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

CREATE table custo (

cust\_id int,

firstname string,

lastname string,

age int,

profession string

)

row format delimited

fields terminated by ','

stored as textfile;

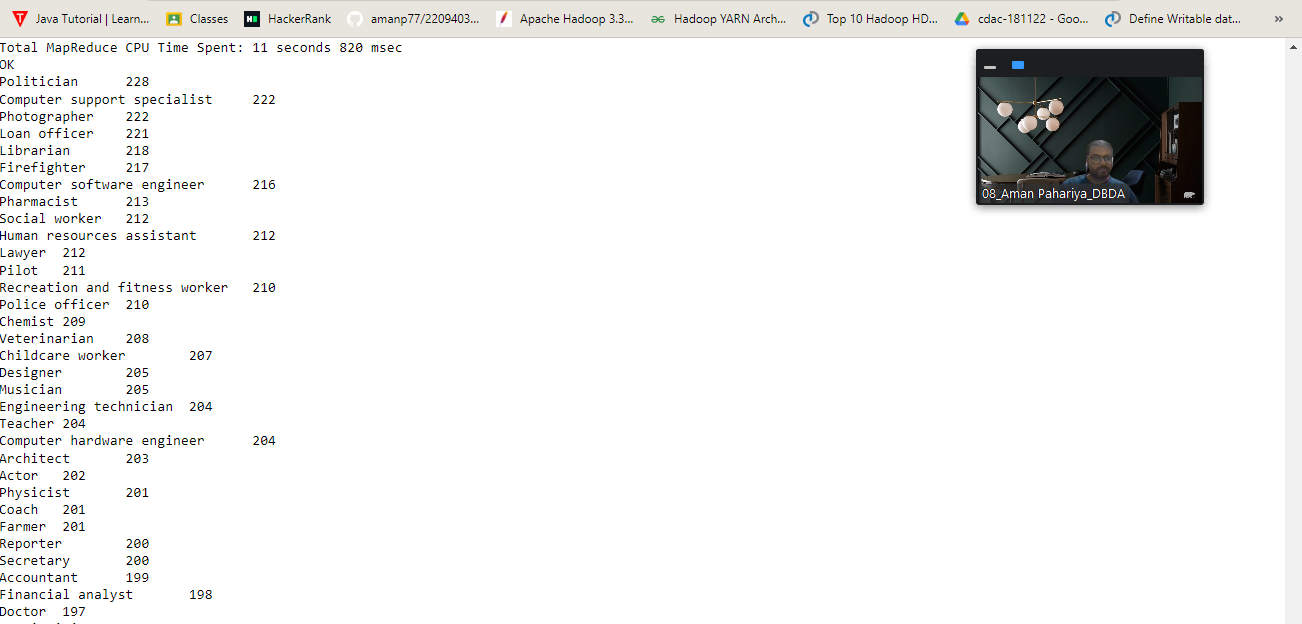
Load data local inpath "custs1.txt" overwrite into table custo;

**select profession, count(profession) as total**

**from custo**

**group by profession**

**order by total desc;**



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

create table txn(

txn\_id int,

txn\_date string,

cust\_id int,

amount double,

category string,

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 txn;

**select product , sum(amount) as total\_sales**

**from txn**

**group by product**

**order by total\_sales desc**

**limit 10;**

****

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

create table txn\_part(

txn\_id int,

txn\_date string,

cust\_id int,

amount double,

product string,

city string,

state string,

spendby string

)

partition by (category String)

row format delimited

fields terminated by ','

stored as textfile;

insert overwrite table txn\_part partition(category)

select t.txn\_id, t.txn\_date, t.cust\_id, t.amount, t.product, t.city, t.state,

t.spendby, t.category

from txn t distribute by category;

**PySpark =>**

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

from pyspark.sql.types import StructType, IntegerType, StringType, DoubleType, LongType

schema1 = StructType().add("year", IntegerType(), True).add("Quarter",IntegerType(), True).add("avg\_revenue", IntegerType(), True).add("total\_book\_tic", IntegerType(), True)

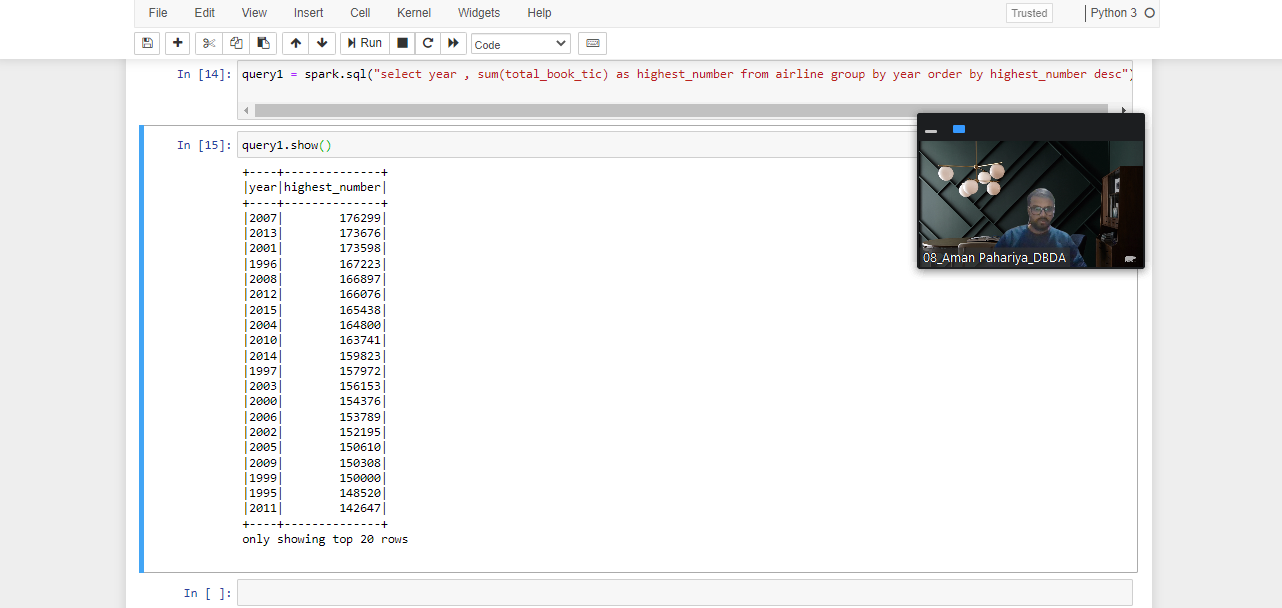
df\_to\_schma = spark.read.format("csv").option("header", "True").schema(schema2).load("/user/bigcdac432585/exam/airlines1.csv")

df\_to\_schma.printSchema()

df\_to\_schema.registerTempTable("airline")

**query1 = spark.sql("select year , sum(total\_book\_tic) as highest\_number from airline group by year order by highest\_number desc")**

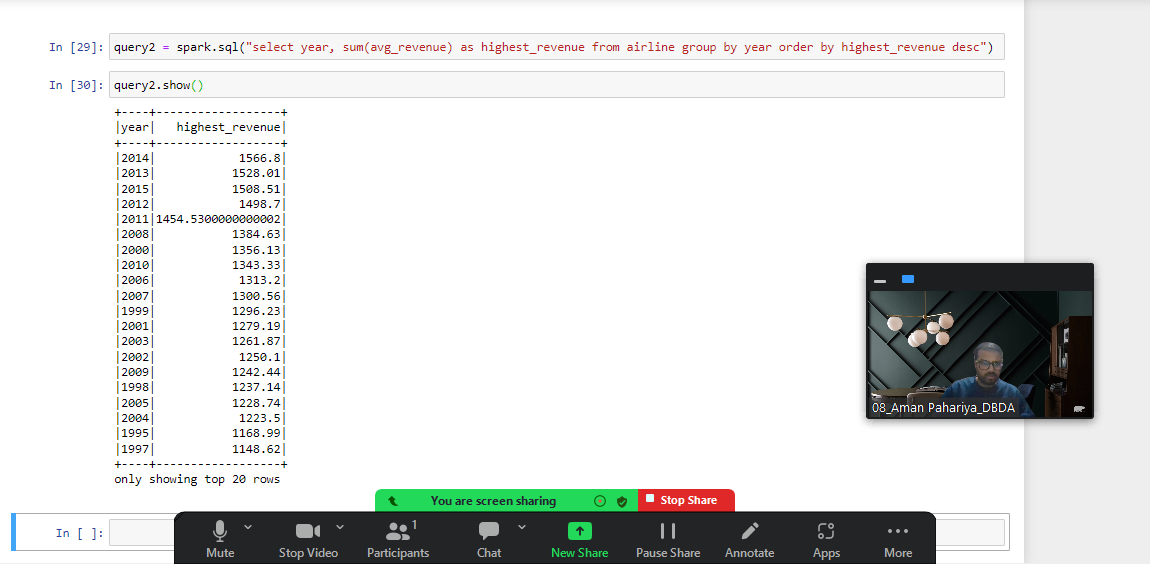
**query1.show()**



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

query2 = spark.sql("select year, sum(avg\_revenue) as highest\_revenue from airline group by year order by highest\_revenue desc")

query2.show()



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

group)

query3 = spark.sql("select year, Quarter sum(avg\_revenue) as highest\_revenue from airline group by year, Quarter order by highest\_revenue desc ")

query3.show()

**MAPREDUCE =>**

**Q1. MapReduce**

**and**

**Question 2 : Find all time High price for each stock**

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.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 EXAM {

public static class MapClass extends Mapper<LongWritable,Text,Text,LongWritable>

{

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

{

try{

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

long vol = Long.parseLong(str[7]);

context.write(new Text(str[1]),new LongWritable(vol));

}

catch(Exception e)

{

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

}

}

}

public static class ReduceClass extends Reducer<Text,LongWritable,Text,LongWritable>

{

private LongWritable result = new LongWritable();

public void reduce(Text key, Iterable<LongWritable> values,Context context) throws IOException, InterruptedException {

long sum = 0;

for (LongWritable val : values)

{

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Volume Count");

job.setJarByClass(EXAM.class);

job.setMapperClass(MapClass.class);

job.setCombinerClass(ReduceClass.class);

job.setReducerClass(ReduceClass.class);

job.setNumReduceTasks(1);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(LongWritable.class);

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

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

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

}

}

**Command to run on hadoop Shell =>**

Hadoop fs myjar.jar EXAM.java /user/bigcdac432585/training/NYSE.csv /user/bigcdac432585/training/result