**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

package cdac2;

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 AllTime {

public static class Map extends Mapper<LongWritable, Text, Text, LongWritable> {

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

try {

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

Long high = Long.parseLong(str[4]);

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

} catch (Exception e) {

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

}

}

}

public static class Reduce extends Reducer<Text LongWritable,Text,LongWritable> {

private Longwritable result = new LongWritable();

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

double high =0;

double temp =0;

for(Longwritable val : values) {

temp = val.get();

if (temp > high) {

high = temp;

}

}

result.set(high);

context.write(key,result)

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "High Price For Each stock");

job.setJarByClass(AllTime);

job.setMapperClass(Map);

job.setReducerClass(Reduce);

job.setNumReduceTasks(1);

job.setMapOutputKeyClass(Text);

job.setMapOutputValueClass(Longwritable);

job.setOutputKeyClass(Text);

job.setOutputValueClass(LongWritable);

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

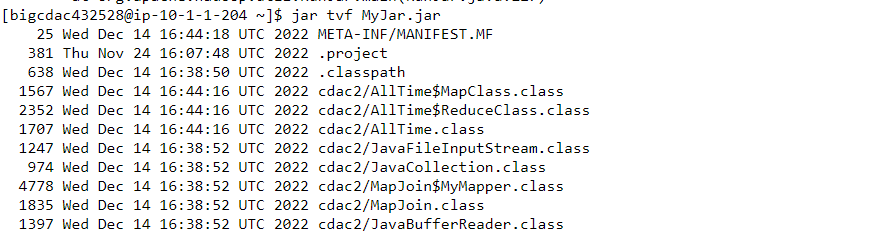
FileOutputFormat.setOutputPath(job, new Path(args[1]));

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

}

}

}



**Question 2 : Find all time High price for each stock [15 marks]**

**Hive**

Please find the customer data set.

cust id

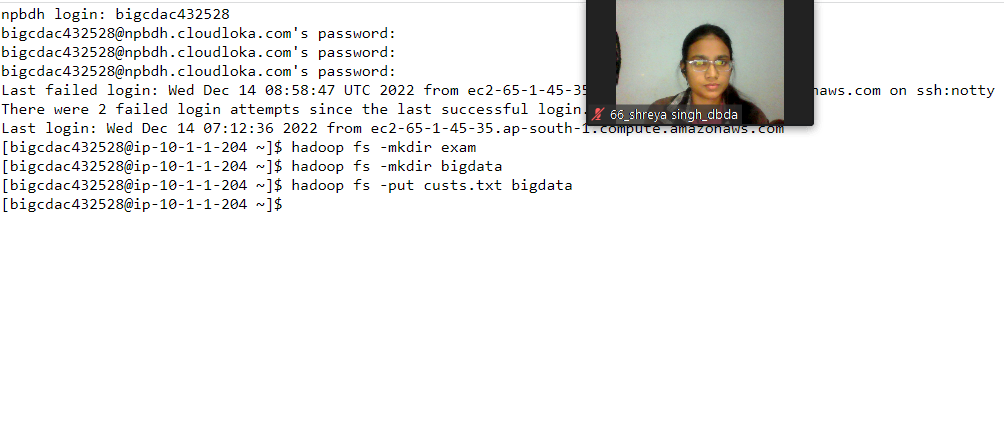
firstname

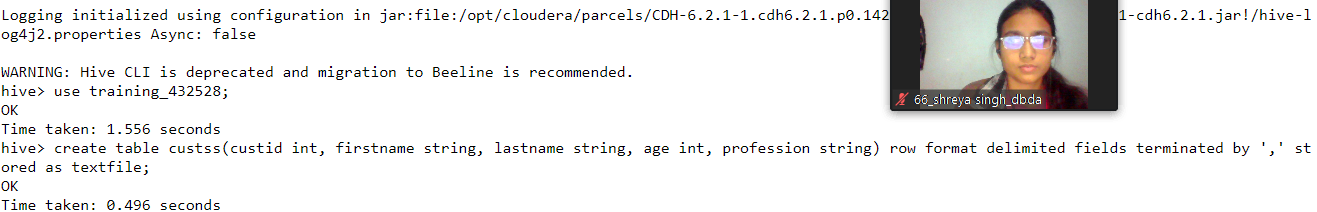
lastname

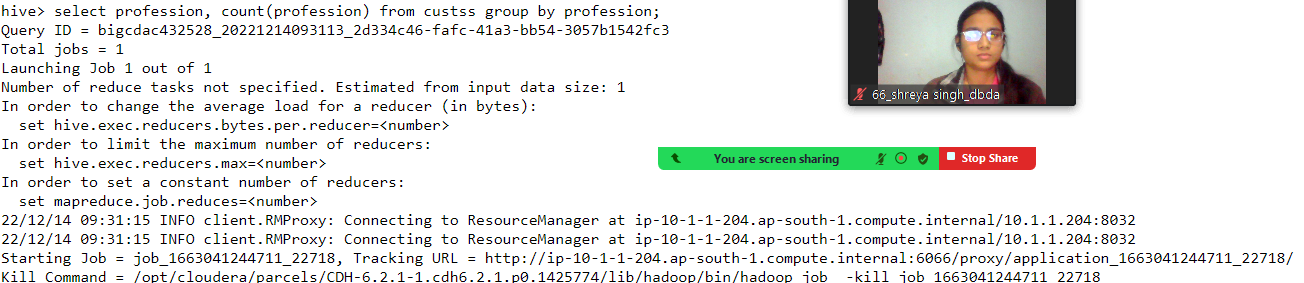
age

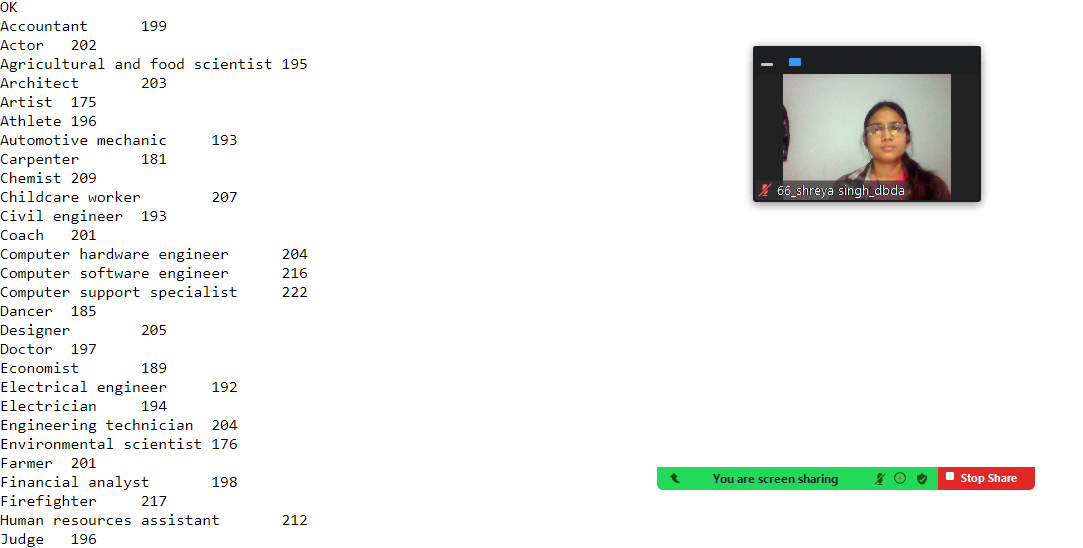
profession

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

**Ans- **

****

****

****

Please find the sales data set.

txn id

txn date

cust id

amount

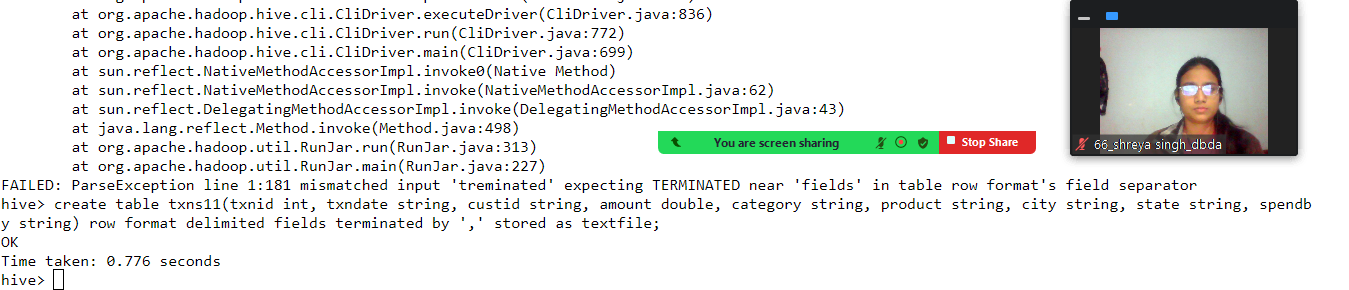
category

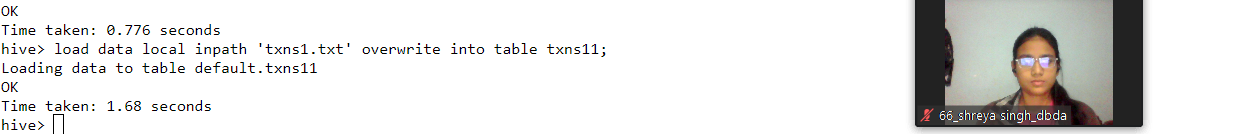
product

city

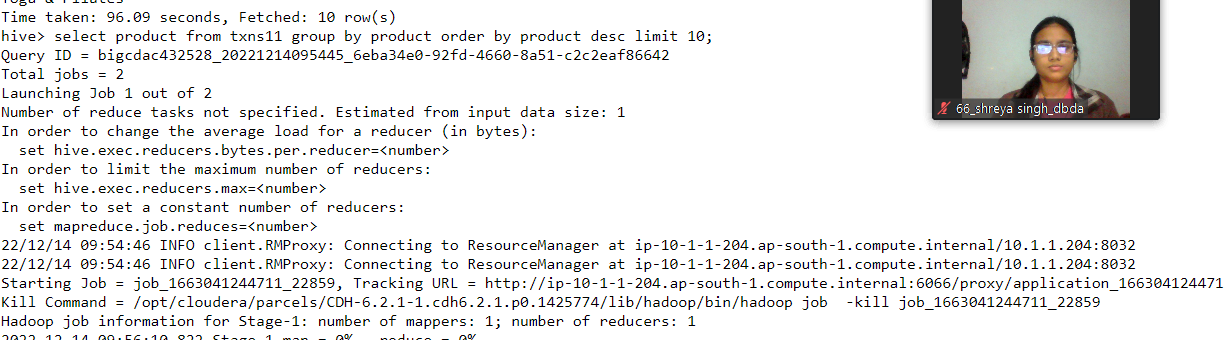
state

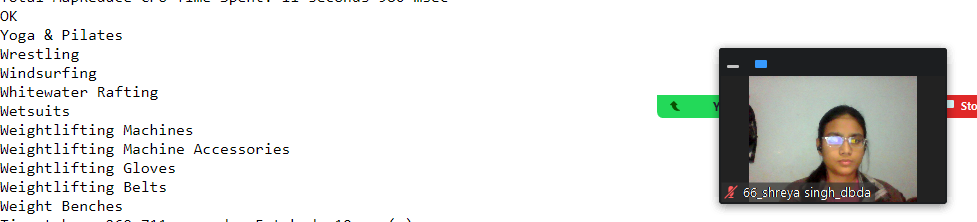
spendby



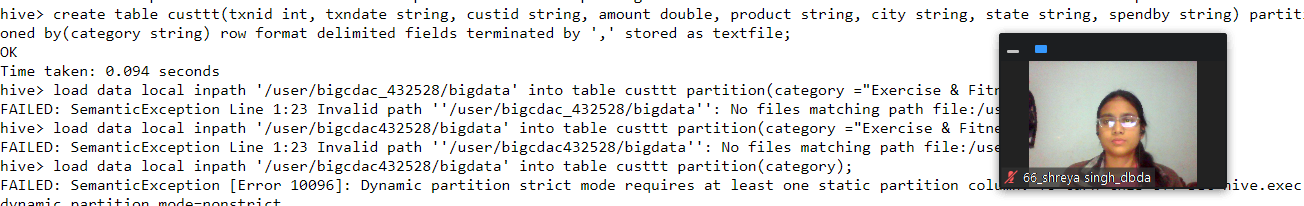


**2) Write a program to find the top 10 products sales wise**

****

****

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

****

**QUESTION 3 [15 marks]**

**PySpark**

Please find the AIRLINES data set

Year

Quarter

Average revenue per seat

Total number of booked seats

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

**Year?**

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

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