**H1B VISA**

**DATA ANALYSIS**

**USING**

**BIG DATA - HADOOP**

Work Done By:

**JAYANT KORE,**

**S180010500059, NIIT-PDTBD**

**ACKNOWLEDGEMENT**

I find immense pleasure to convey my sincere and grateful thanks to **NIIT** and the management for providing necessary facilities in carrying out this project.

I greatly indebted to my Tech Mentor **Ms. Nandita,** the batch instructor **Mr. Annu** and the SLT faculty **Mr. Sandeep** for constant support throughout the course and also for useful suggestions, constant encouragement and kind advice in bringing out this project as a success.

* **The importance of the employment details**

The h1b visa data set, wide-range activity, which takes place once a decade in the entire country. Its purpose is to gather information about the general population, in order to present a full and reliable picture of the population in the country - its housing conditions and demographic, social and economic characteristics.

The information collected includes data on,

**S\_no**

**Case\_status**

**Employer\_name**

**Job\_title**

**Soc\_name**

**Soc\_code**

**Job\_title**

**Full\_time\_position**

**Previling\_wages**

**Year**

**Work\_site**

**Lon**

**Lat**

* **Total Data and Statistics**

Data is a total for 2011, 2012, 2013 , 2014 , 2015 , 2016. Which consist of serial no, case status, job title, employer name, latitude , longitude, worksite ,prevailing wages, years.

* **Term in this Project**

The columns in the dataset include:

* CASE\_STATUS: Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.

Certified: Employer filed the LCA, which was approved by DOL

Certified Withdrawn: LCA was approved but later withdrawn by employer

Withdrawn: LCA was withdrawn by employer before approval

Denied: LCA was denied by DOL

* EMPLOYER\_NAME: Name of employer submitting labour condition application.
* SOC\_NAME: the Occupational name associated with the SOC\_CODE. SOC\_CODE is the occupational code associated with the job being requested for temporary labour condition, as classified by the Standard Occupational Classification (SOC) System.
* JOB\_TITLE: Title of the job
* FULL\_TIME\_POSITION: Y = Full Time Position; N = Part Time Position
* PREVAILING\_WAGE: Prevailing Wage for the job being requested for temporary labour condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer’s minimum requirements for the position.
* YEAR: Year in which the H1B visa petition was filed
* WORKSITE: City and State information of the foreign worker’s intended area of employment
* lon: longitude of the Worksite
* lat: latitude of the Worksite

SUCCESS RATE % = (Certified + Certified Withdrawn)/Total x 100

Q. 1 a) is the number of petitions with Data Engineer job title increasing over time?

I/P:

-------------------------- main class ------------------

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

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

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

public class Question1a {

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

{

Configuration conf= new Configuration();

Job job= Job.getInstance(conf,"Question 1a");

job.setJarByClass(Question1a.class);

job.setMapperClass(Question1aMapper.class);

job.setReducerClass(Question1aReducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(LongWritable.class);

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)?1:0);

}

}

-------------------------- Mapper class ------------------

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class Question1aMapper extends Mapper<LongWritable,Text,Text,LongWritable>{

LongWritable one =new LongWritable(1);

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

{

if(key.get()> 0)

{

String [] token=values.toString().split(",(?=([^\"]\*\"[^\"]\*\")\*[^\"]\*$)");

if(token[4]!=null && token[4].contains("DATA ENGINEER") && token[7]!=null && !token[7].equals("NA"))

{ Text answer= new Text("DATA ENGINEER"+","+token[7]);

context.write(answer,one);

}

}

}

}

-------------------------- Reducer class ------------------

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Question1aReducer extends Reducer <Text,LongWritable,Text,LongWritable>

{ LongWritable SUM=new LongWritable(0); int i=0;

String[] years={"2011","2012","2013","2014","2015","2016"};

long [] arr=new long[6];

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

{

long sum=0;

for(LongWritable val:values)

sum+=val.get();

arr[i++]=sum;

}

public void cleanup(Context context) throws IOException, InterruptedException

{

for (int i=0;i<6;i++)

if (i==0)

context.write(new Text(years[i]), new LongWritable(0));

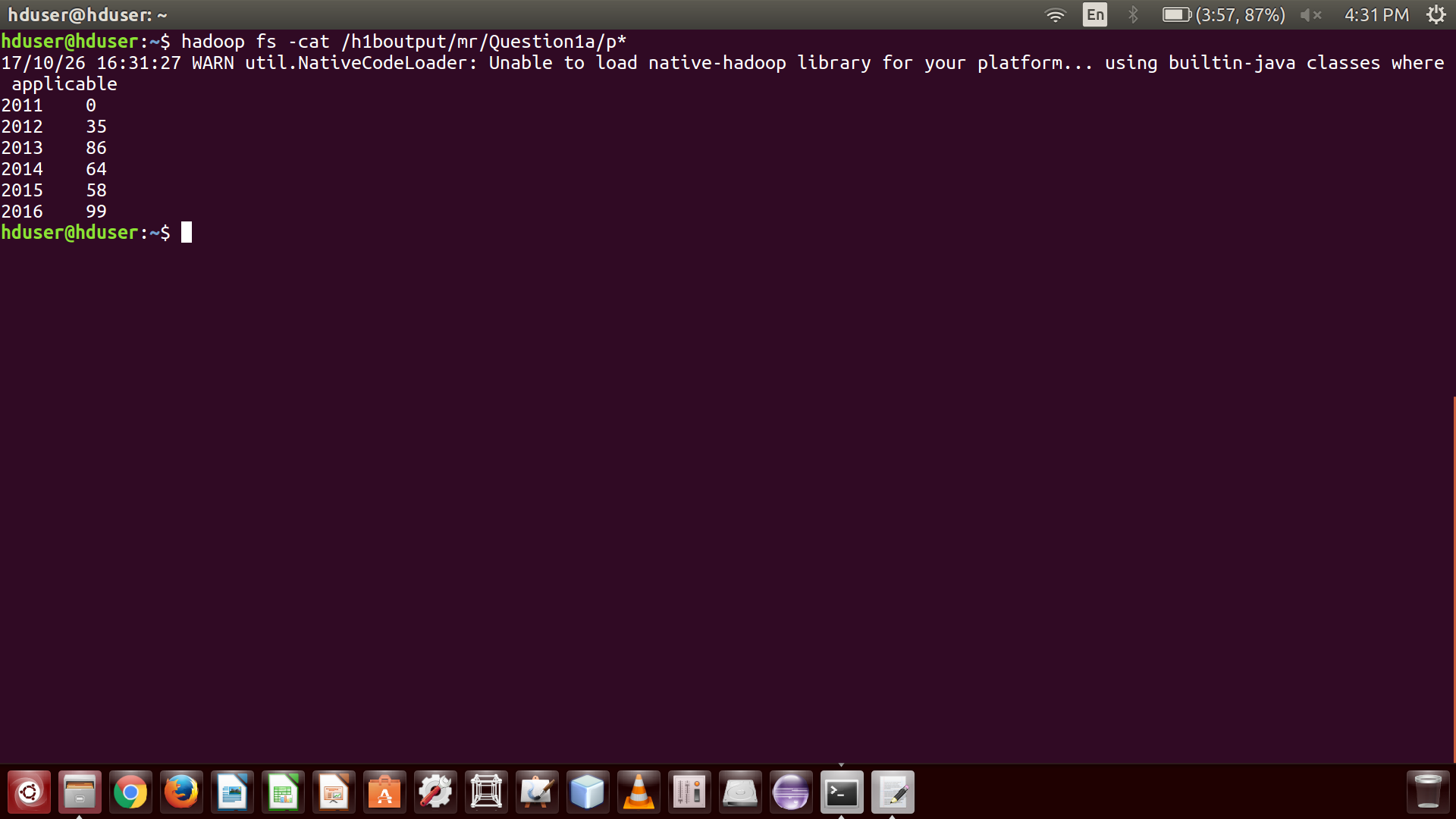
else

context.write(new Text(years[i]),new LongWritable((arr[i]-arr[i-1])\*100/arr[i-1]));

}

}

O/P:

****

Q. 1. b) Find top 5 job titles who are having highest avg growth in applications.[ALL]

I/P:

register /usr/local/hive/lib/hive-exec-1.2.1.jar

register /usr/local/hive/lib/hive-common-1.2.1.jar

data1 = LOAD 'hdfs://localhost:54310/user/hive/warehouse/h1b\_final' USING PigStorage('\t') as (s\_no:double,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:chararray,worksite:chararray,longitude,latitude);

cleansed= filter data1 by $7=='2011';

a= group cleansed by $4;

step\_a= foreach a generate group,COUNT($1);

describe step\_a;

cleansed1= filter data1 by $7=='2012';

b= group cleansed1 by $4;

step\_b= foreach b generate group,COUNT($1);

describe step\_b;

cleansed2= filter data1 by $7=='2013';

c= group cleansed2 by $4;

step\_c= foreach c generate group,COUNT($1);

describe step\_c;

cleansed3= filter data1 by $7=='2014';

d= group cleansed3 by $4;

step\_d= foreach d generate group,COUNT($1);

describe step\_d;

cleansed4= filter data1 by $7=='2015';

e= group cleansed4 by $4;

step\_e= foreach e generate group,COUNT($1);

describe step\_e;

cleansed5= filter data1 by $7=='2016';

f= group cleansed5 by $4;

step\_f= foreach f generate group,COUNT($1);

describe step\_f;

joined= join step\_a by $0,step\_b by $0,step\_c by $0,step\_d by $0,step\_e by $0,step\_f by $0;

describe joined;

yearwiseapplications= foreach joined generate $0,$1,$3,$5,$7,$9,$11;

progressivegrowth= foreach yearwiseapplications generate $0,ROUND\_TO((long)($6-$5)\*100/$5,2),ROUND\_TO((long)($5-$4)\*100/$4,2),ROUND\_TO((long)($4-$3)\*100/$3,2),ROUND\_TO((long)($3-$2)\*100/$2,2),ROUND\_TO((long)($2-$1)\*100/$1,2);

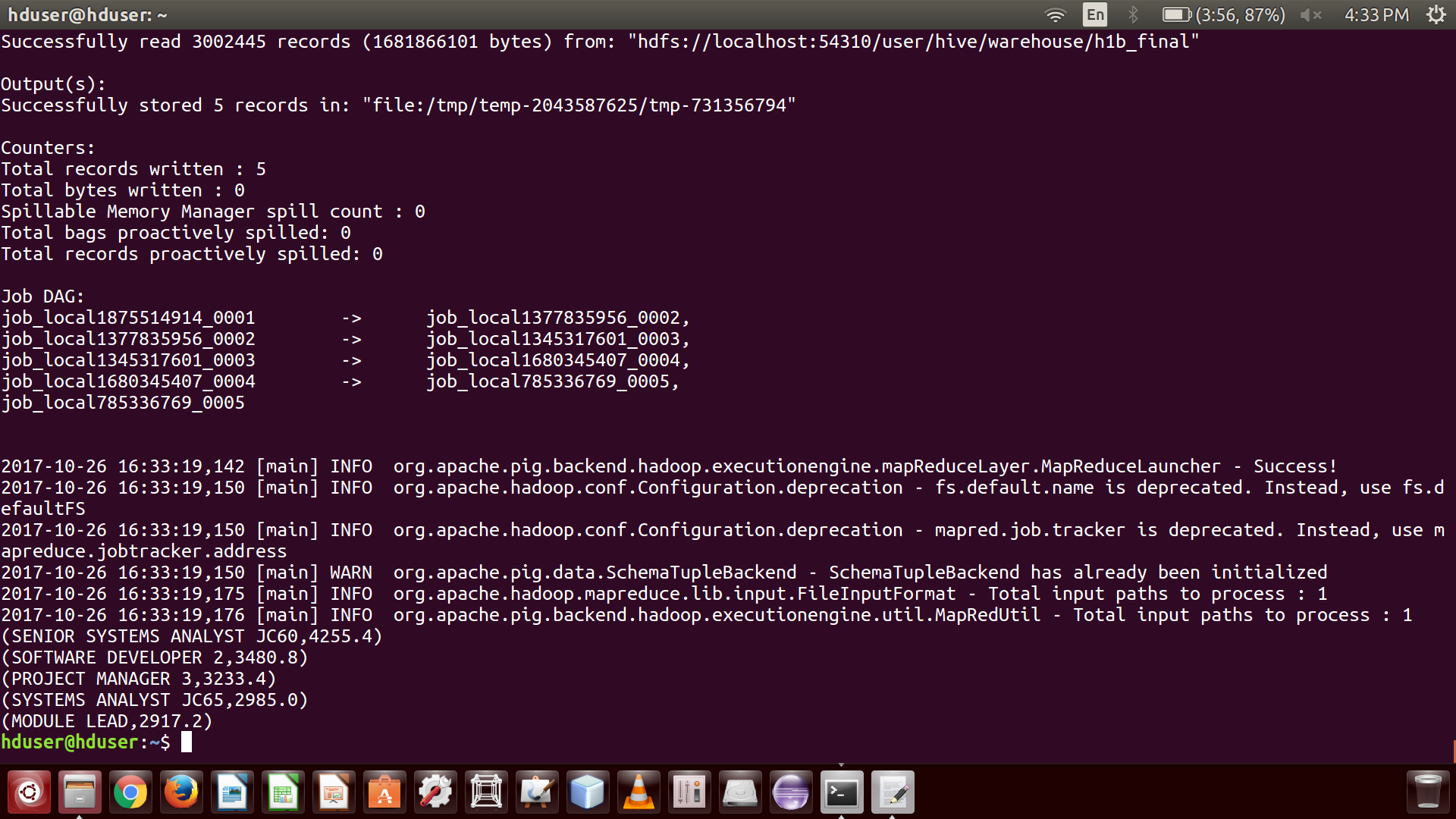
avgprogressivegrowth= foreach progressivegrowth generate $0,($1+$2+$3+$4+$5)/5;

orderedavggrowth= order avgprogressivegrowth by $1 desc;

answer = limit orderedavggrowth 5;

dump answer;

O/P:

****

Q. 2. a) Which part of the US has the most Data Engineer jobs for each year?

I/P:

-------------------------- main class ------------------

import java.io.IOException;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

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

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

public class Question2a

{

public static void main(String args[]) throws IOException, InterruptedException, ClassNotFoundException

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Top 5 Data Engineer in a worksite");

job.setJarByClass(Question2a.class);

job.setMapperClass(Question2aMapper.class);

job.setPartitionerClass(Question2aPartitioner.class);

job.setReducerClass(Question2aReducer.class);

job.setNumReduceTasks(7);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(LongWritable.class);

job.setOutputKeyClass(NullWritable.class);

job.setOutputValueClass(Text.class);

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

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

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

}

}

-------------------------- Mapper class ------------------

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class Question2aMapper extends Mapper < LongWritable, Text, Text, LongWritable > {

LongWritable one = new LongWritable(1);

public void map(LongWritable key, Text values, Context context) throws IOException,

InterruptedException {

if (key.get() > 0)

{

String[] token = values.toString().split(",(?=([^\"]\*\"[^\"]\*\")\*[^\"]\*$)");

if (token[4] != null && token[4].contains("DATA ENGINEER") && token[8] != null && !token[8].equals("NA")) {

Text answer = new Text(token[8].replaceAll("\"", "") + "\t" + token[7]);

context.write(answer, one);

}

}

}

}

-------------------------- Reducer class ------------------

import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Question2aReducer extends Reducer<Text,LongWritable,NullWritable,Text>

{

private TreeMap<LongWritable, Text> Top5DataEngineer = new TreeMap<LongWritable, Text>();

long sum=0;

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

{

sum=0;

for(LongWritable val:values)

{

sum+=val.get();

}

Top5DataEngineer.put(new LongWritable(sum),new Text(key+","+sum));

if (Top5DataEngineer.size()>5)

Top5DataEngineer.remove(Top5DataEngineer.firstKey());

}

protected void cleanup(Context context)throws IOException, InterruptedException

{

for (Text t : Top5DataEngineer.descendingMap().values())

context.write(NullWritable.get(), t);

}

}

-------------------------- Partitioner class ------------------

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Partitioner;

public class Question2aPartitioner extends

Partitioner < Text, LongWritable > {

@Override

public int getPartition(Text key, LongWritable value, int numReduceTasks) {

String[] str = key.toString().split("\t");

if (str[1].equals("2011"))

return 0;

if (str[1].equals("2012"))

return 1;

if (str[1].equals("2013"))

return 2;

if (str[1].equals("2014"))

return 3;

if (str[1].equals("2015"))

return 4;

if (str[1].equals("2016"))

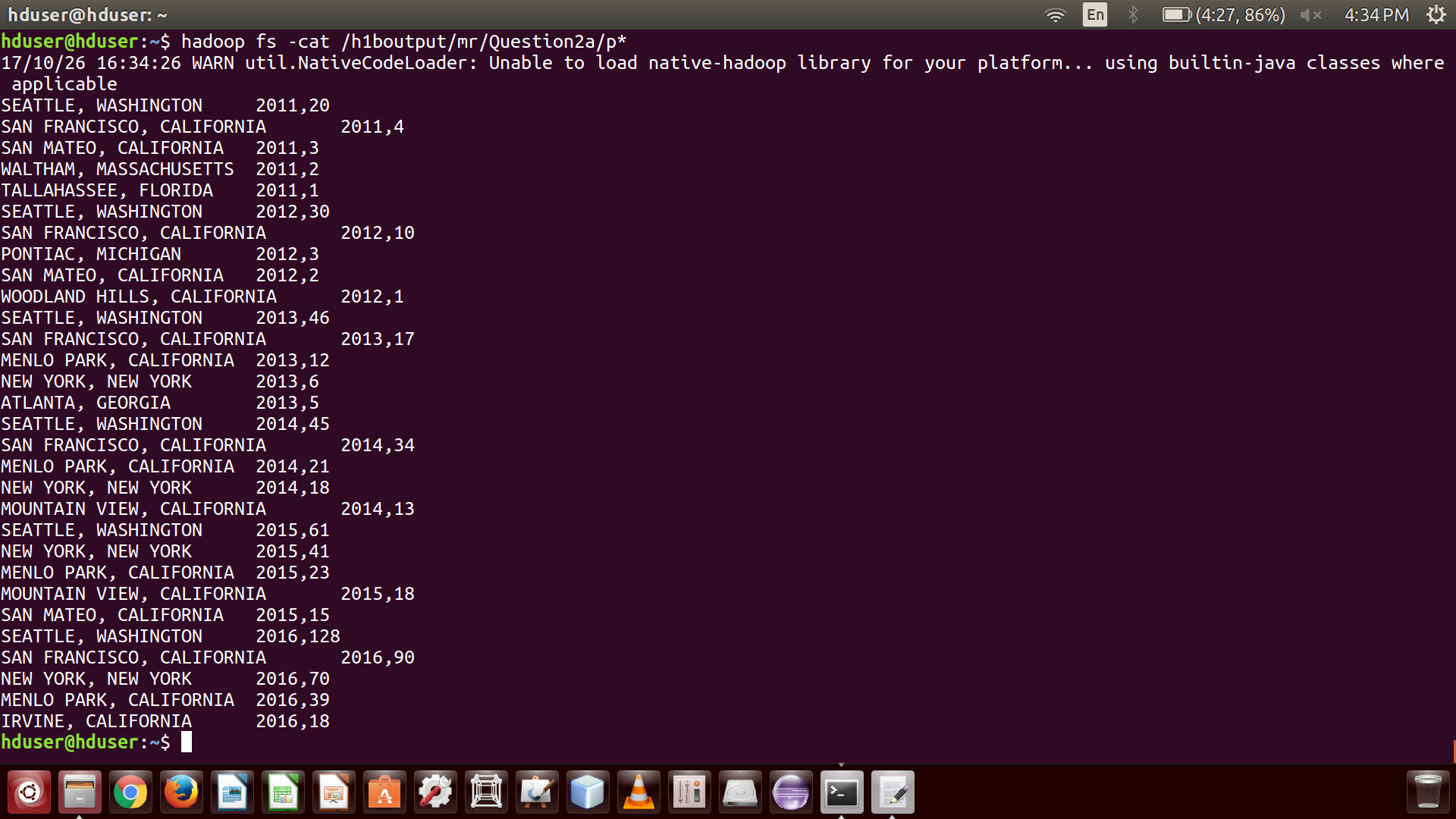
return 5;

else

return 6;

}

}

O/P:

Q. 2. b) find top 5 locations in the US who have got certified visa for each year.[certified]

I/P:

rm /home/hduser/project/2b.txt

hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2011' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" > /home/hduser/project/2b.txt

hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2012' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" >> /home/hduser/project/2b.txt

hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2013' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" >> /home/hduser/project/2b.txt

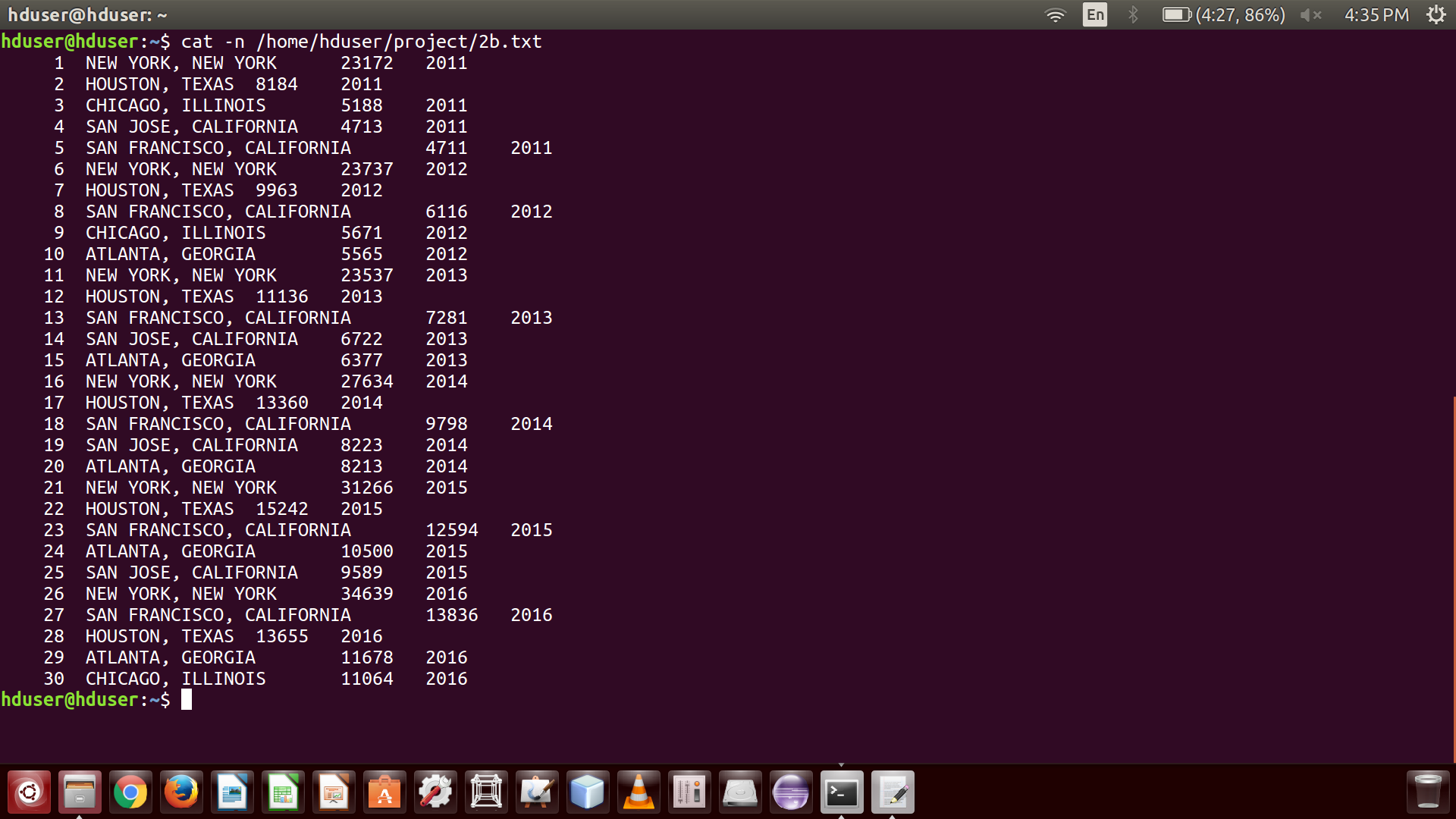
hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2014' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" >> /home/hduser/project/2b.txt

hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2015' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" >> /home/hduser/project/2b.txt

hive -e "select worksite,count(case\_status) as t,year from h1b\_final where year ='2016' and case\_status='CERTIFIED' group by worksite,year order by t desc limit 5;" >> /home/hduser/project/2b.txt

cat -n /home/hduser/project/2b.txt

O/P:



Q.3) Which industry(SOC\_NAME) has the most number of Data Scientist positions?

[certified]

I/P:

-------------------------- Main class ------------------

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

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

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

public class Question3 {

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Data Scientist jobs");

job.setJarByClass(Question3.class);

job.setMapperClass(Question3Mapper.class);

job.setReducerClass(Question3Reducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(LongWritable.class);

job.setOutputKeyClass(NullWritable.class);

job.setOutputValueClass(Text.class);

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

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

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

}

}

-------------------------- Mapper class ------------------

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class Question3Mapper extends Mapper < LongWritable, Text, Text, LongWritable > {

LongWritable one = new LongWritable(1);

public void map(LongWritable key, Text values, Context context) throws IOException,

InterruptedException {

if (key.get() > 0) {

String[] token = values.toString().split(",(?=([^\"]\*\"[^\"]\*\")\*[^\"]\*$)");

if (token[4].contains("DATA SCIENTIST")) {

Text answer = new Text(token[3].replaceAll("\"", ""));

context.write(answer, one);

}

}

}

}

-------------------------- Reducer class ------------------

import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Question3Reducer extends Reducer < Text, LongWritable, NullWritable, Text > {

private TreeMap < LongWritable,

Text > DataScientistJobs = new TreeMap < LongWritable,

Text > ();

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

InterruptedException {

long sum = 0;

for (LongWritable val: values)

sum += val.get();

DataScientistJobs.put(new LongWritable(sum), new Text(key.toString().replaceAll("\"", "") + "," + sum));

if (DataScientistJobs.size() > 5)

DataScientistJobs.remove(DataScientistJobs.firstKey());

}

protected void cleanup(Context context) throws IOException,

InterruptedException {

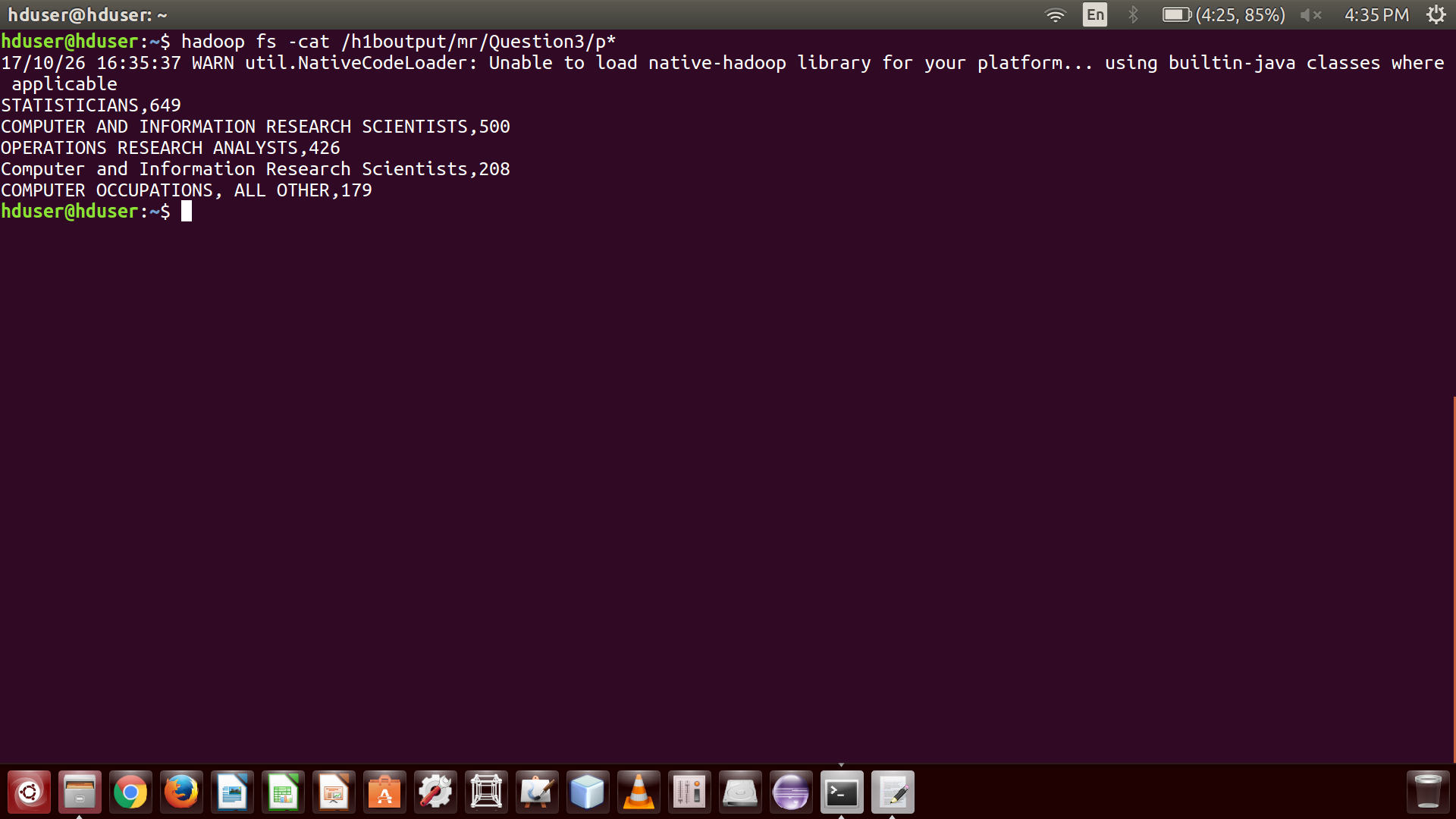
for (Text t: DataScientistJobs.descendingMap().values())

context.write(NullWritable.get(), t);

}

}

O/P:

****

Q.4) Which top 5 employers file the most petitions each year? - Case Status - ALL

I/P:

-------------------------- Main class ------------------

import java.io.IOException;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

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

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

public class Question4 {

public static void main(String args[]) throws IOException, InterruptedException, ClassNotFoundException

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Top 5 Employers");

job.setJarByClass(Question4.class);

job.setMapperClass(Question4Mapper.class);

job.setPartitionerClass(Question4Partitioner.class);

job.setReducerClass(Question4Reducer.class);

job.setNumReduceTasks(7);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(LongWritable.class);

job.setOutputKeyClass(NullWritable.class);

job.setOutputValueClass(Text.class);

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

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

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

}

}

-------------------------- Mapper class ------------------

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class Question4Mapper extends Mapper < LongWritable, Text, Text, LongWritable > {

LongWritable one = new LongWritable(1);

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

InterruptedException {

if (key.get() > 0)

{

String[] token = value.toString().split(",(?=([^\"]\*\"[^\"]\*\")\*[^\"]\*$)");

if (!token[1].equals("NA") && token[1] != null && !token[2].equals("NA") && token[2] != null && !token[7].equals("NA") && token[7] != null) {

Text answer = new Text(token[2].replaceAll("\"", "") + "\t" + token[7]);

context.write(answer, one);

}

}

}

}

-------------------------- Reducer class ------------------

import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Question4Reducer extends Reducer < Text, LongWritable, NullWritable, Text > {

private TreeMap < LongWritable,

Text > Top5Employers = new TreeMap < LongWritable,

Text > ();

long sum = 0;

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

InterruptedException {

sum = 0;

for (LongWritable val: values) {

sum += val.get();

}

Top5Employers.put(new LongWritable(sum), new Text(key + "," + sum));

if (Top5Employers.size() > 5)

Top5Employers.remove(Top5Employers.firstKey());

}

protected void cleanup(Context context) throws IOException,

InterruptedException {

for (Text t: Top5Employers.descendingMap().values())

context.write(NullWritable.get(), t);

}

}

-------------------------- Partitioner class ------------------

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Partitioner;

public class Question4Partitioner extends

Partitioner < Text, LongWritable > {

@Override

public int getPartition(Text key, LongWritable value, int numReduceTasks) {

String[] str = key.toString().split("\t");

if (str[1].equals("2011"))

return 0;

if (str[1].equals("2012"))

return 1;

if (str[1].equals("2013"))

return 2;

if (str[1].equals("2014"))

return 3;

if (str[1].equals("2015"))

return 4;

if (str[1].equals("2016"))

return 5;

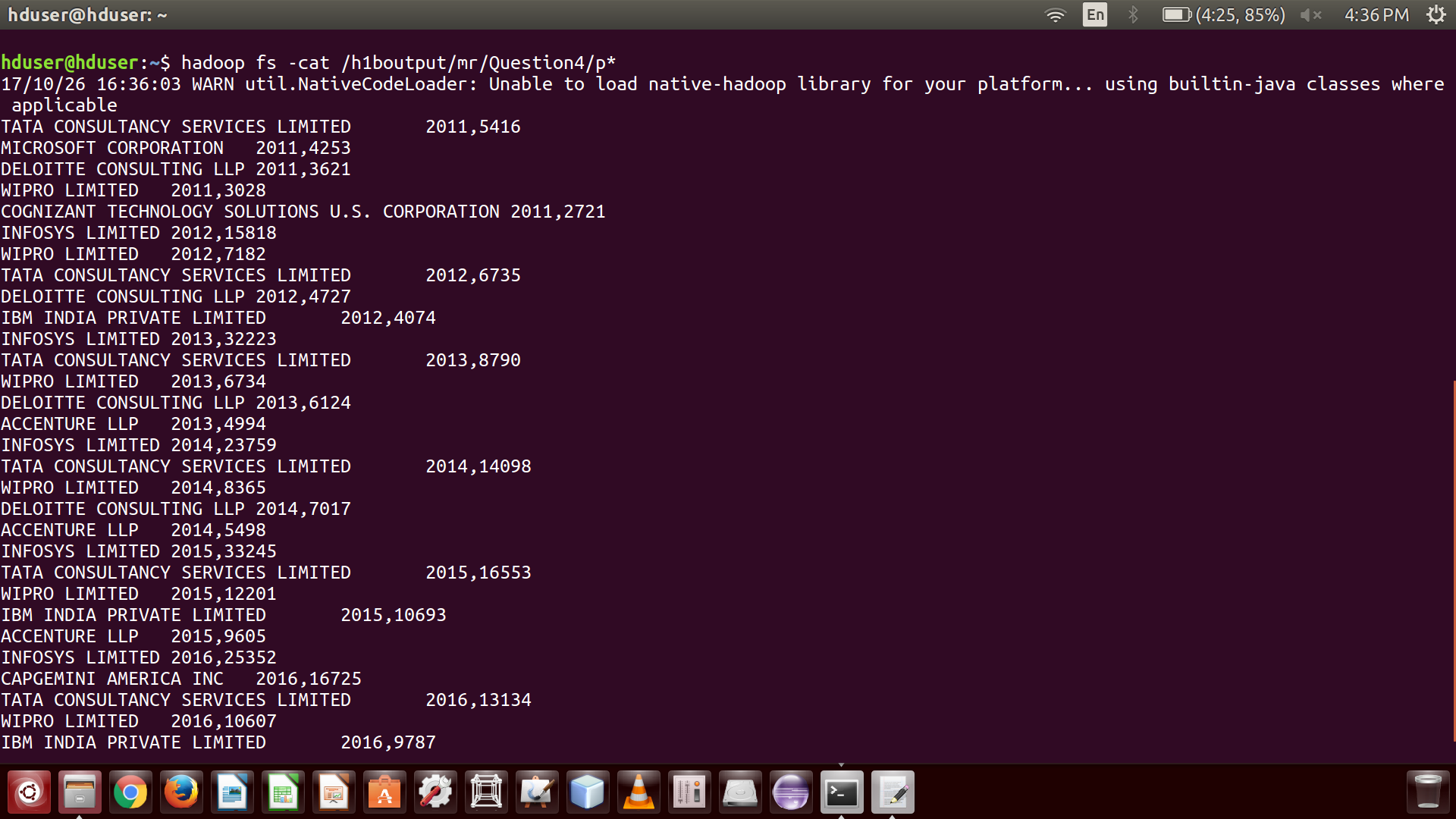
else

return 6;

}

}

O/P:

****

Q.5) Find the most popular top 10 job positions for H1B visa applications for each year?

a) for all the applications

b) for only certified applications.

I/P:

rm /home/hduser/project/5.txt

rm /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2011 group by job\_title,year order by temp desc limit 10;" > /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2012 group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2013 group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2014 group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2015 group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2016 group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2011 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" > /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2012 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2013 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2014 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2015 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5b.txt

hive -e "select job\_title,year,count(case\_status ) as temp from h1b\_final where year = 2016 and case\_status= 'CERTIFIED' group by job\_title,year order by temp desc limit 10;" >> /home/hduser/project/5b.txt

echo -e "Top 10 Job Titles for all the applications"

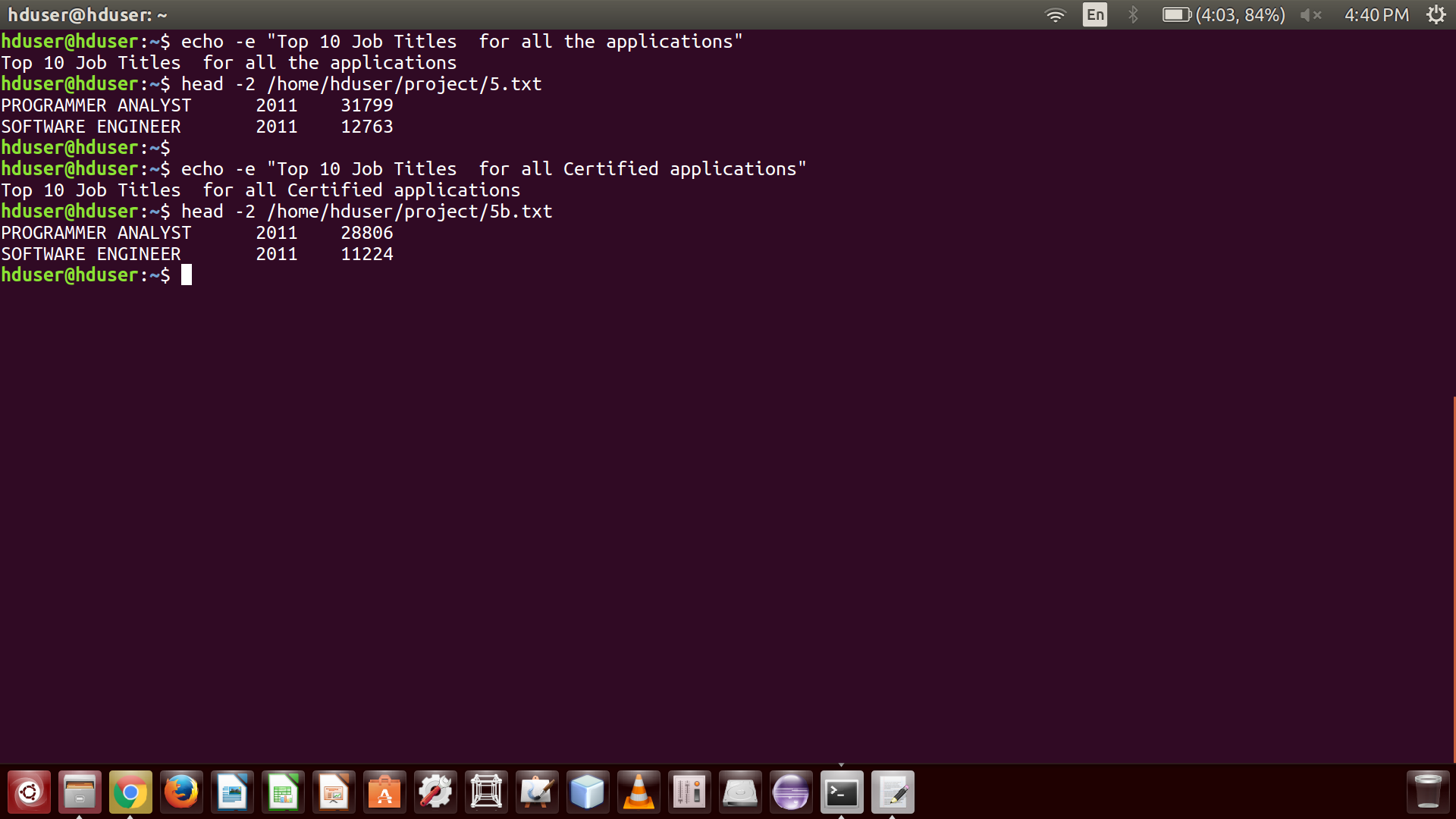
cat -n /home/hduser/project/5.txt

echo -e "\n"

echo -e "Top 10 Job Titles for all Certified applications"

cat -n /home/hduser/project/5b.txt

O/P:

****

Q.6) Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time.

I/P:

register /usr/local/hive/lib/hive-exec-1.2.1.jar

register /usr/local/hive/lib/hive-common-1.2.1.jar

data1 = LOAD 'hdfs://localhost:54310/user/hive/warehouse/h1b\_final' USING PigStorage('\t') as (s\_no:double,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:chararray,worksite:chararray,longitude,latitude);

cleansed= filter data1 by $1 is not null and $1!='NA';

temp= group cleansed by $7;

total= foreach temp generate group,COUNT(cleansed.$0);

cleansed1= filter data1 by $7 is not null and $7!='NA';

temp1= group cleansed1 by ($7,$1);

yearsoccount= foreach temp1 generate group,group.$0,COUNT($1);

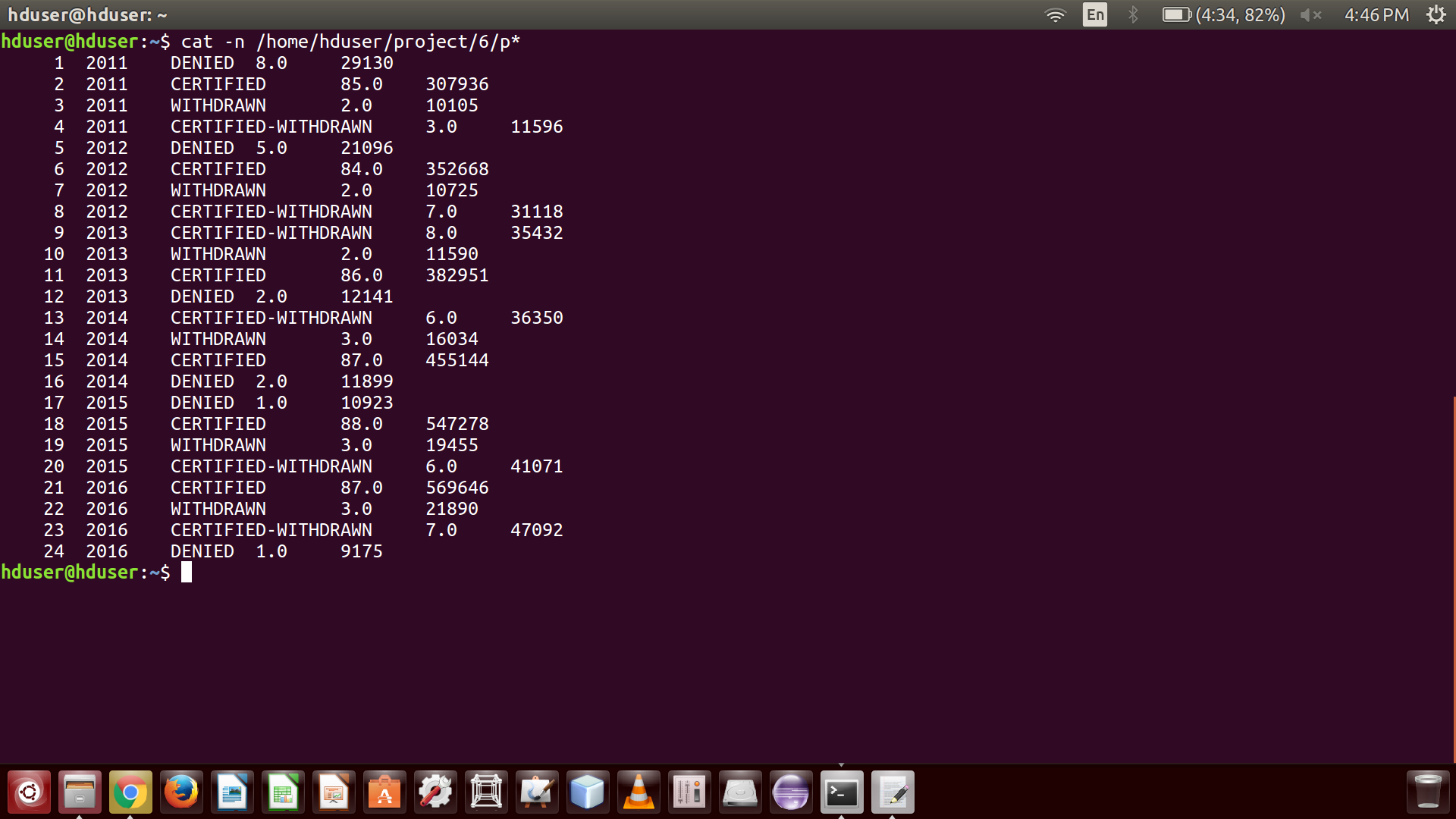
joined= join yearsoccount by $1,total by $0;

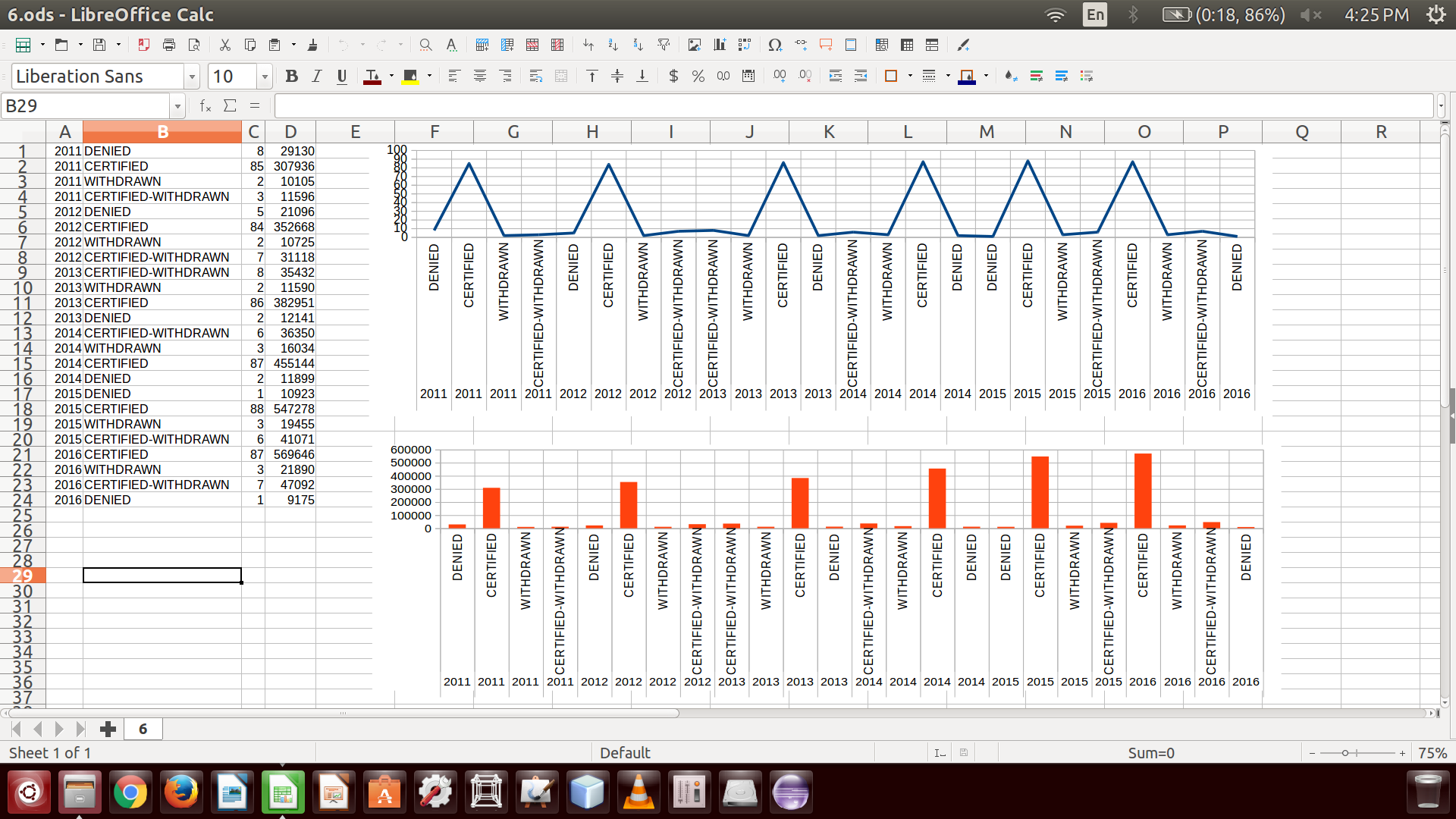
ans= foreach joined generate FLATTEN($0),ROUND\_TO(((long)$2\*100)/(long)$4,2),$2;

store ans into '/home/hduser/project/6' using PigStorage('\t');

dump ans;

O/P:

****

****

Q.7) Create a bar graph to depict the number of applications for each year [All]

I/P:

rm /home/hduser/project/1.dat

hive -e "select year,count(\*) as applications from h1b\_final where year like '201%' group by year;" > /home/hduser/project/1.dat

gnuplot -persist <<-EOFMarker

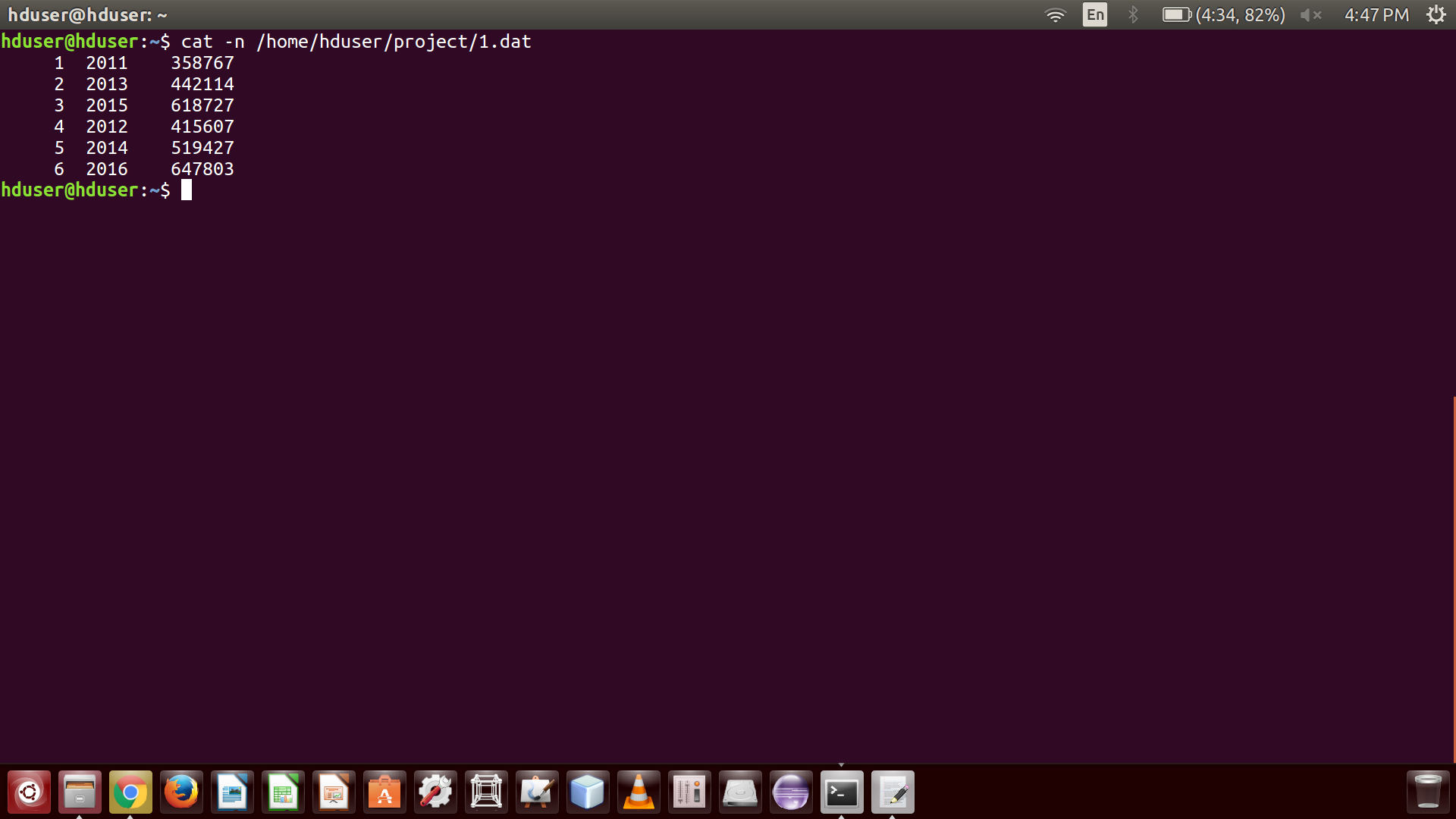
set boxwidth 0.5

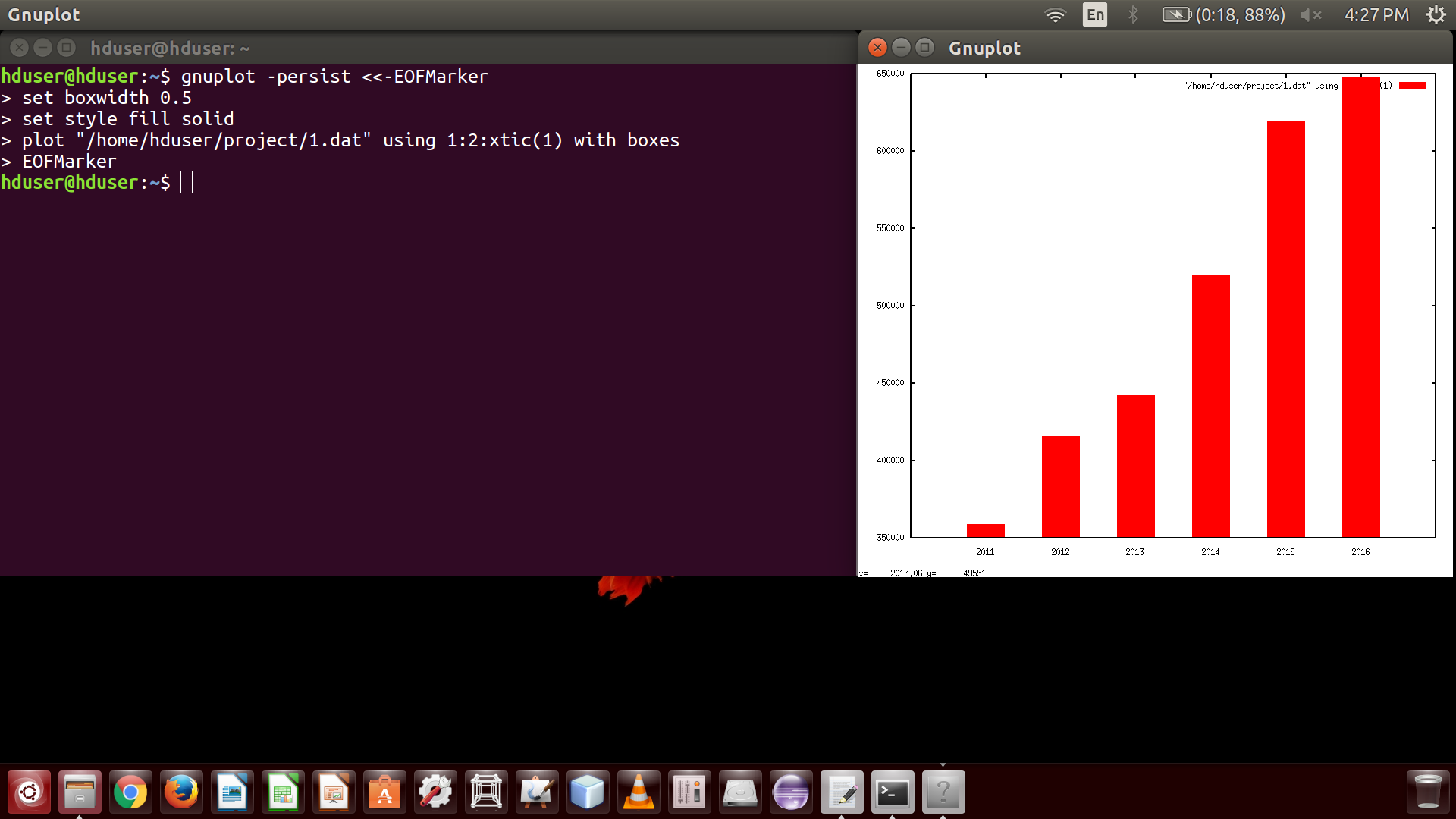
set style fill solid

plot "/home/hduser/project/1.dat" using 1:2:xtic(1) with boxes

EOFMarker

O/P:

****

****

Q.8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order - [Certified and Certified Withdrawn.]

I/P:

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2011' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" > /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2011' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" > /home/hduser/project/8n.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2012' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2012' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8n.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2013' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2013' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8n.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2014' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2014' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8n.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2015' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2015' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8n.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2016' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8y.txt;

hive -e "select job\_title,full\_time\_position,year,avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2016' and case\_status in('CERTIFIED','CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,year order by average desc" >> /home/hduser/project/8n.txt;

echo -e "Average with Full-Time Job"

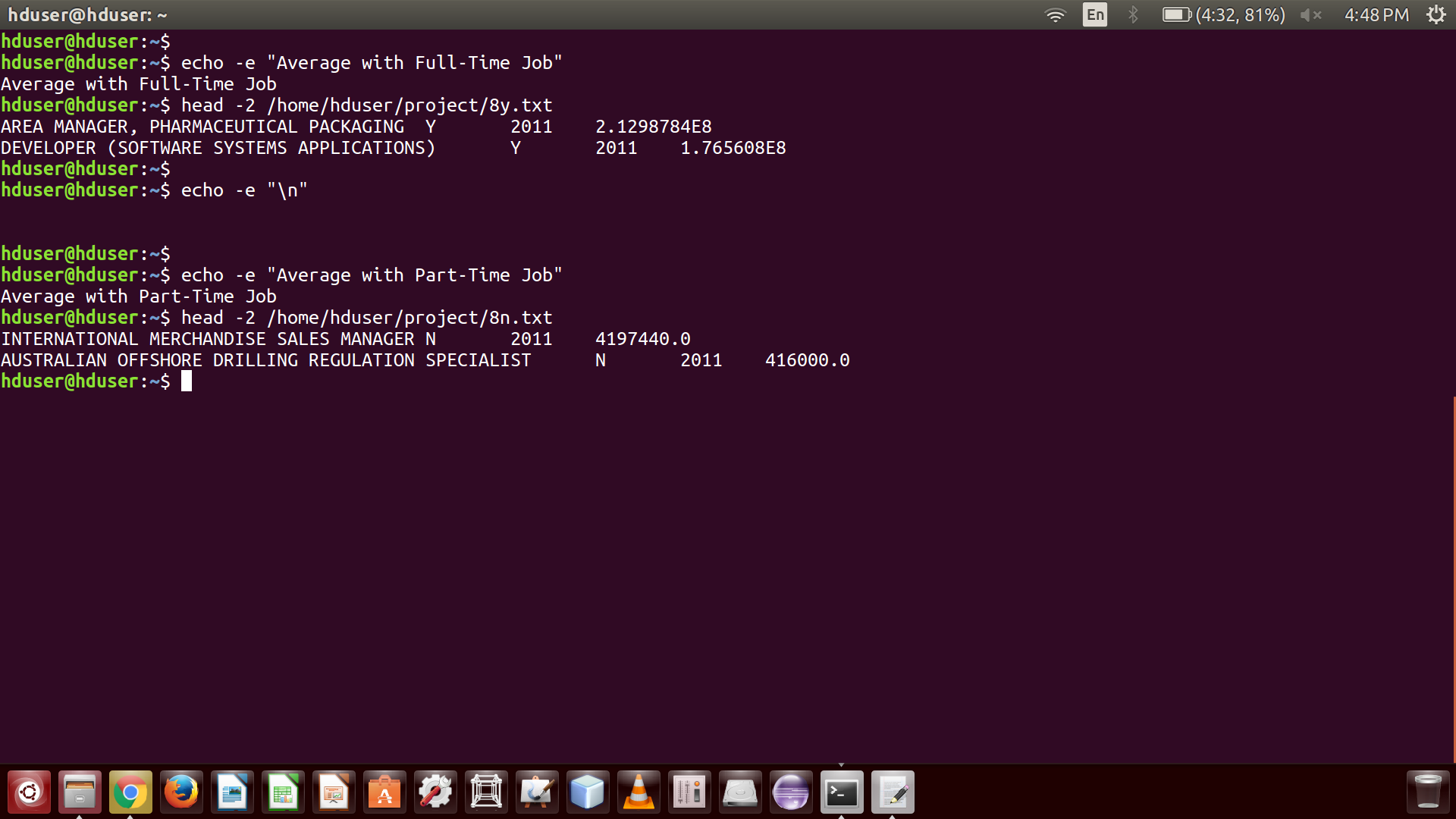
cat /home/hduser/project/8y.txt

echo -e "\n"

echo -e "Average with Part-Time Job"

cat /home/hduser/project/8n.txt

O/P:

****

Q.9) Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000) ?

I/P:

hadoop fs -rmr /h1boutput/pig/question9

register /usr/local/hive/lib/hive-exec-1.2.1.jar

register /usr/local/hive/lib/hive-common-1.2.1.jar

data1 = LOAD 'hdfs://localhost:54310/user/hive/warehouse/h1b\_final' USING PigStorage('\t') as (s\_no:double,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:chararray,worksite:chararray,longitude,latitude);

cleansed= filter data1 by $1 is not null and $1!='NA';

temp= group cleansed by $2;

total= foreach temp generate group,COUNT(cleansed.$0);

certified= filter data1 by $1 == 'CERTIFIED';

temp1= group certified by $2;

totalcertified= foreach temp1 generate group,COUNT(certified.$0);

certified\_with= filter data1 by $1 == 'CERTIFIED-WITHDRAWN';

temp2= group certified\_with by $2;

totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified\_with.$0);

joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;

joined= foreach joined generate $0,$1,$3,$5;

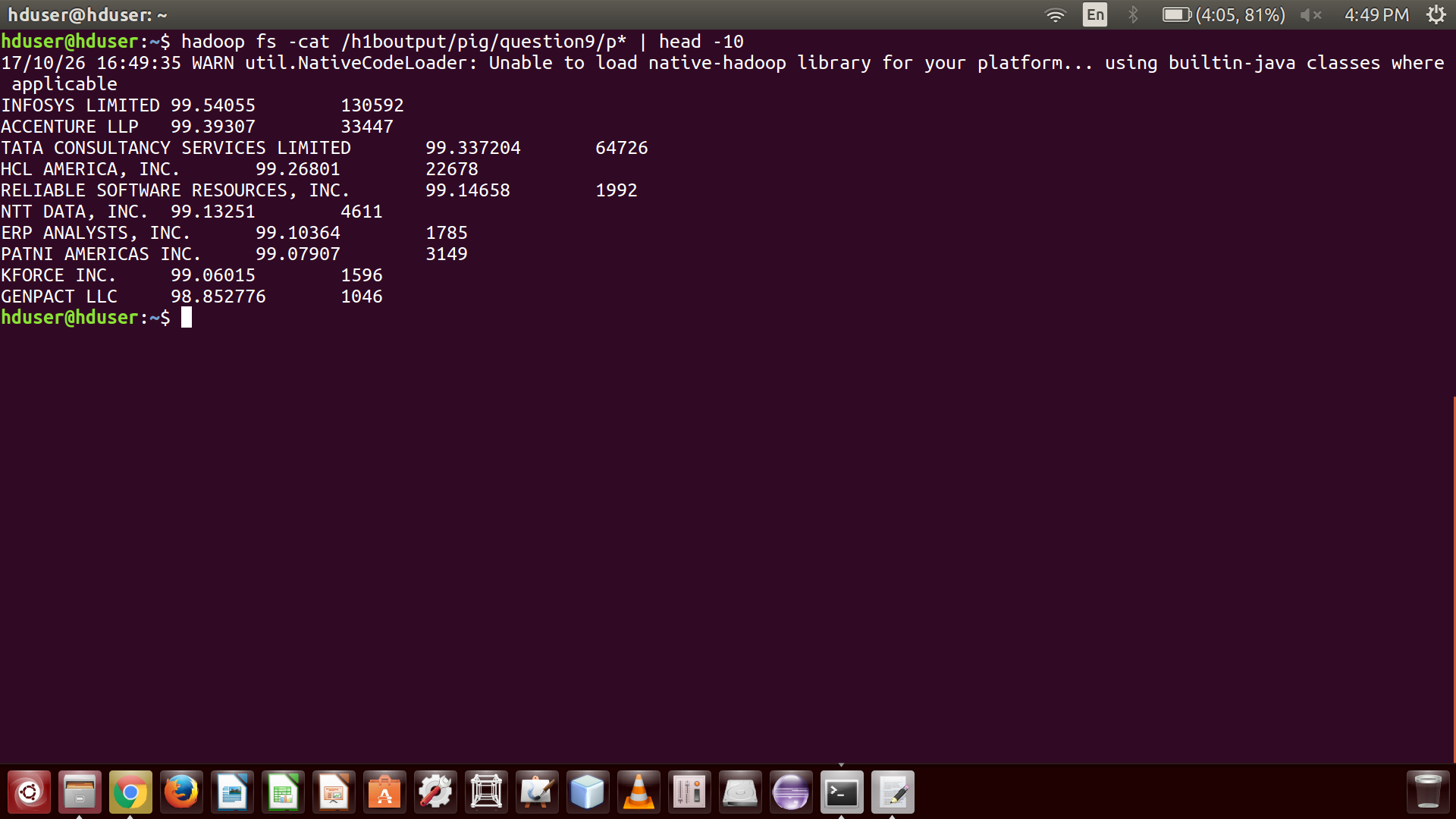
intermediateoutput= foreach joined generate $0,(float)($1+$2)\*100/($3),$3;

intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000;

finaloutput= order intermediateoutput2 by $1 DESC;

store finaloutput into 'hdfs://localhost:54310/h1boutput/pig/question9' using PigStorage('\t');

hadoop fs -cat /h1boutput/pig/question9/p\*

O/P:

Q.10) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than 1000)?

I/P:

hadoop fs -rmr /h1boutput/pig/question10

register /usr/local/hive/lib/hive-exec-1.2.1.jar

register /usr/local/hive/lib/hive-common-1.2.1.jar

data1 = LOAD 'hdfs://localhost:54310/user/hive/warehouse/h1b\_final' USING PigStorage('\t') as (s\_no:double,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:chararray,worksite:chararray,longitude,latitude);

cleansed= filter data1 by $1 is not null and $1!='NA';

temp= group cleansed by $4;

total= foreach temp generate group,COUNT(cleansed.$0);

certified= filter data1 by $1 == 'CERTIFIED';

temp1= group certified by $4;

totalcertified= foreach temp1 generate group,COUNT(certified.$0);

certified\_with= filter data1 by $1 == 'CERTIFIED-WITHDRAWN';

temp2= group certified\_with by $4;

totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified\_with.$0);

joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;

joined= foreach joined generate $0,$1,$3,$5;

intermediateoutput= foreach joined generate $0,(float)($1+$2)\*100/($3),$3;

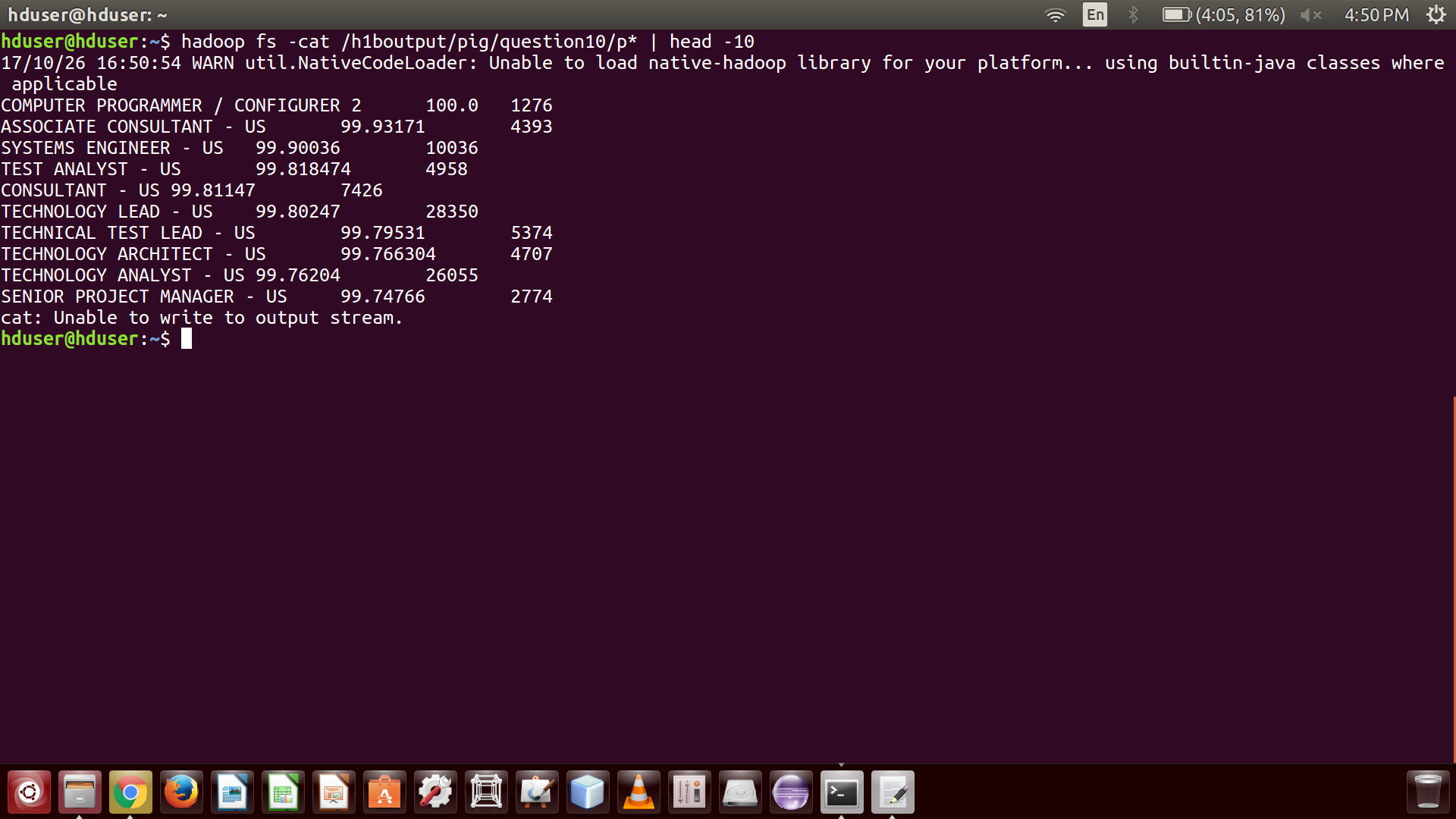
intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000;

finaloutput= order intermediateoutput2 by $1 DESC;

store finaloutput into 'hdfs://localhost:54310/h1boutput/pig/question10' using PigStorage('\t');

hadoop fs -cat /h1boutput/pig/question10/p\*

O/P:

****

Q.11) Export result for question no 10 to MySql database.

I/P:

hadoop fs -rmr /h1boutput/Question11

hadoop fs -mkdir -p /h1boutput/Question11

hadoop fs -cp /h1boutput/pig/question10/p\* /h1boutput/Question11/

mysql -u root -p'1234' -e "drop database question11;create database if not exists question11;use question11;create table question11(job\_title varchar(100),success\_rate float,petitions int);"

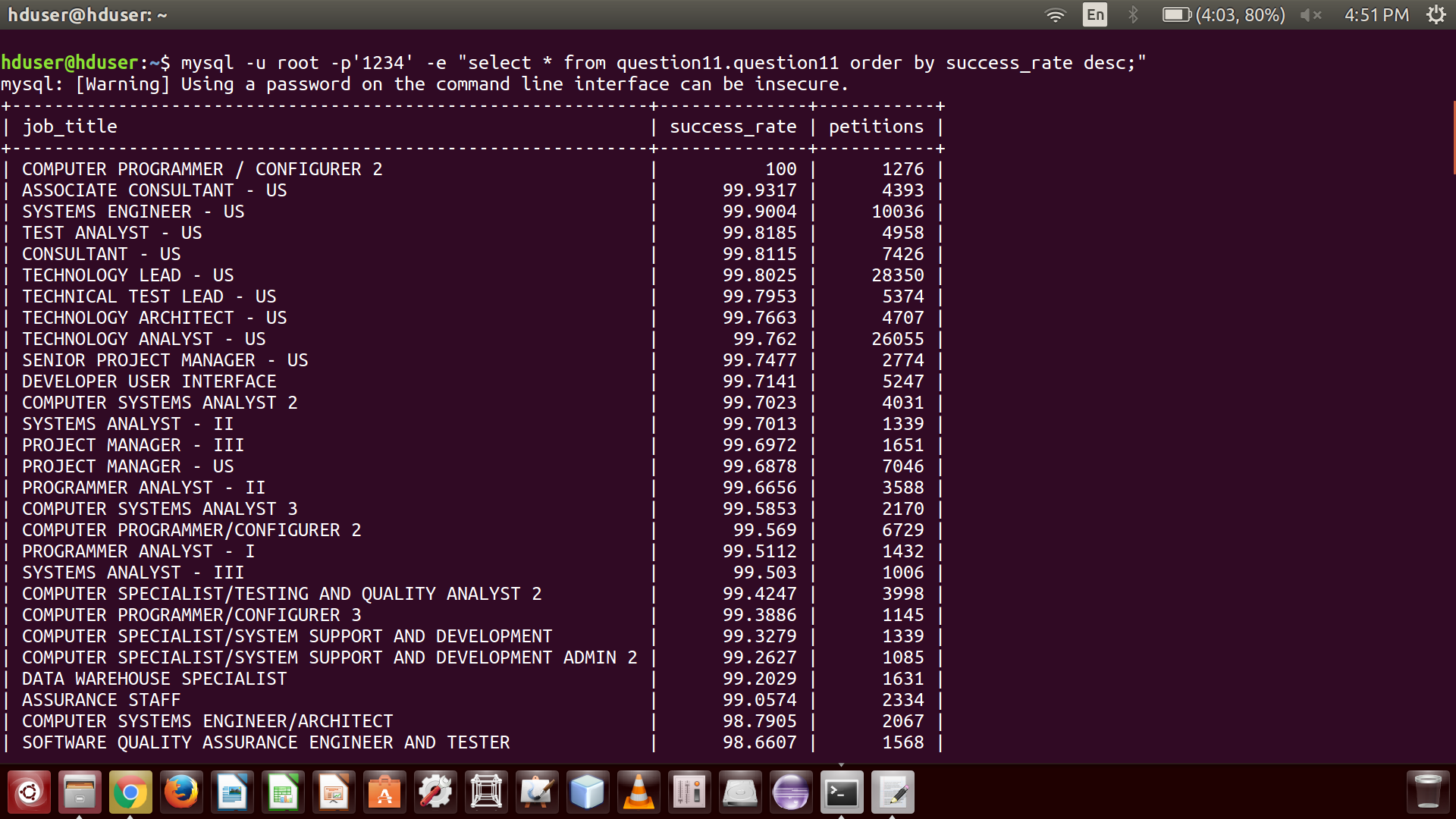
sqoop export --connect jdbc:mysql://localhost/question11 --username root --password '1234' --table question11 --update-mode allowinsert --export-dir /h1boutput/Question11/p\* --input-fields-terminated-by '\t' ;

echo -e '\n\nDisplay contents from MySQL Database.\n\n'

echo -e '\n10) Which are the top 10 job positions that have success rate more than 70% in petitions and total petitions filed more than 1000?\n\n'

mysql -u root -p'1234' -e "select \* from question11.question11 order by success\_rate desc;"

O/P:

****

* **PROGRAM – Shell Script:**

start-all.sh

show\_menu()

{

NORMAL=`echo "\033[m"`

MENU=`echo "\033[36m"` #blue

NUMBER=`echo "\033[33m"` #yellow

FGRED=`echo "\033[41m"`

RED\_TEXT=`echo "\033[31m"`

ENTER\_LINE=`echo "\033[33m"`

echo -e "${MENU}\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*H1B APPLICATIONS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 1) ${RED\_TEXT} [1(a)] ${MENU} Is the number of petitions with Data Engineer job title increasing over time?${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 2) ${RED\_TEXT} [1(b)] ${MENU} Find top 5 job titles who are having highest avg growth in applications.[ALL]${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 3) ${RED\_TEXT} [2(a)] ${MENU} Which part of the US has the most Data Engineer jobs for each year?${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 4) ${RED\_TEXT} [2(b)] ${MENU} find top 5 locations in the US who have got certified visa for each year.[certified]${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 5) ${RED\_TEXT} [3] ${MENU} Which industry(SOC\_NAME) has the most number of Data Scientist positions?

[certified]${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 6) ${RED\_TEXT} [4] ${MENU} Which top 5 employers file the most petitions each year? - Case Status - ALL ${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 7) ${RED\_TEXT} [5] ${MENU} Find the most popular top 10 job positions for H1B visa applications for each year?

a) for all the applications

b) for only certified applications.${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 8) ${RED\_TEXT} [6] ${MENU} Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time.${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 9) ${RED\_TEXT} [7] ${MENU} Create a bar graph to depict the number of applications for each year [All]${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 10) ${RED\_TEXT} [8] ${MENU}Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order ${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 11) ${RED\_TEXT} [9] ${MENU} Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000) ?${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 12) ${RED\_TEXT} [10] ${MENU} Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than 1000)? ${NORMAL}"

echo -e "${MENU}\*\*${NUMBER} 13) ${RED\_TEXT} [11] ${MENU}Export result for question no 10 to MySql database.${NORMAL}"

echo -e "${MENU}\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*${NORMAL}"

echo -e "${ENTER\_LINE}Please enter a menu option and enter or ${RED\_TEXT}press enter to exit. ${NORMAL}"

read opt

}

function option\_picked()

{

COLOR='\033[01;31m' # bold red

RESET='\033[00;00m' # normal white

MESSAGE="$1" #modified to post the correct option selected

echo -e "${COLOR}${MESSAGE}${RESET}"

}

clear

show\_menu

while [ opt != '' ]

do

if [[ $opt = "" ]]; then

exit;

else

case $opt in

1) clear;

option\_picked "1 a) Is the number of petitions with Data Engineer job title increasing over time?";

hadoop fs -rmr /h1boutput/mr/Question1a

hadoop jar h1b.jar Question1a /h1b /h1boutput/mr/Question1a

echo -e "1 Is the number of petitions with Data Engineer job title increasing over time?\n\n"

hadoop fs -cat /h1boutput/mr/Question1a/p\*

show\_menu;

;;

2) clear;

option\_picked "1 b) Find top 5 job titles who are having highest avg growth in applications.[ALL]";

pig -x local /home/hduser/project/Codes/Pig/question1b.pig

show\_menu;

;;

3) clear;

option\_picked "2 a) Which part of the US has the most Data Engineer jobs for each year?";

hadoop fs -rmr /h1boutput/mr/Question2a

hadoop jar h1b.jar Question2a /h1b /h1boutput/mr/Question2a

hadoop fs -cat /h1boutput/mr/Question2a/p\*

show\_menu;

;;

4) clear;

option\_picked "2 b) find top 5 locations in the US who have got certified visa for each year.[certified]";

bash /home/hduser/project/Codes/Hive/2b.sh

show\_menu;

;;

5) clear;

option\_picked "3) Which industry(SOC\_NAME) has the most number of Data Scientist positions? [certified]";

hadoop fs -rmr /h1boutput/mr/Question3

hadoop jar h1b.jar Question3 /h1b /h1boutput/mr/Question3

hadoop fs -cat /h1boutput/mr/Question3/p\*

show\_menu;

;;

6) clear;

option\_picked "4)Which top 5 employers file the most petitions each year? - Case Status - ALL";

hadoop fs -rmr /h1boutput/mr/Question4

hadoop jar h1b.jar Question4 /h1b /h1boutput/mr/Question4

hadoop fs -cat /h1boutput/mr/Question4/p\*

show\_menu;

;;

7) clear;

option\_picked "5)Find the most popular top 10 job positions for H1B visa applications for each year?

a) for all the applications

b) for only certified applications.";

bash /home/hduser/project/Codes/Hive/5.sh

show\_menu;

;;

8) clear;

option\_picked "6) Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time.";

rm -r /home/hduser/project/6

pig -x local /home/hduser/project/Codes/Pig/question6.pig

libreoffice /home/hduser/project/6.ods

show\_menu;

;;

9) clear;

option\_picked "7) Create a bar graph to depict the number of applications for each year";

bash /home/hduser/project/Codes/Hive/7.sh

show\_menu;

;;

10) clear;

option\_picked "8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order ";

bash /home/hduser/project/Codes/Hive/8.sh

show\_menu;

;;

11) clear;

option\_picked "9) Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000) ?";

pig -x local /home/hduser/project/Codes/Pig/question9.pig

show\_menu;

;;

12) clear;

option\_picked "10) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than 1000)?";

rm -r /home/mohith/Pig/question10

pig -x local /home/hduser/project/Codes/Pig/question10.pig

cat /home/mohith/Pig/question10/p\*

show\_menu;

;;

13) clear;

option\_picked "11) Export result for question no 10 to MySql database.";

bash /home/hduser/project/Codes/Sqoop-MySql/11.sh

show\_menu;

;;

\n) exit;

;;

\*) clear;

option\_picked "Pick an option from the menu";

show\_menu;

;;

esac

fi

done

* **CONCLUSION:**

1) H1b data set project shows that the amount of case status that are with-drawn and denied in year wise.

2) From the H1b dataset the data engineer job title increases over time.

3) And shows that in year 2011 new jersey

Is the state that has most number of data scientists each year

4) Top job position which the H1b visa is applied and in year wise.

5) The graph shows that number of applications increases over time.

6) Most number of data scientists are available in statistics job location.