H1-B CASE STUDY

1 a) Is the number of petitions with Data Engineer job title increasing over time?
b) Find top 5 job titles who are having highest growth in applications.

2 a) Which part of the US has the most Data Engineer jobs for each year?
b) find top 5 locations in the US who have got certified visa for each year.

3)Which industry has the most number of Data Scientist positions?

4)Which top 5 employers file the most petitions each year?

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

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

7) Create a bar graph to depict the number of applications for each year

8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate)

9) Which are top ten employers who have the highest success rate in petitions?

10) Which are the top 10 job positions which have the highest success rate in petitions?

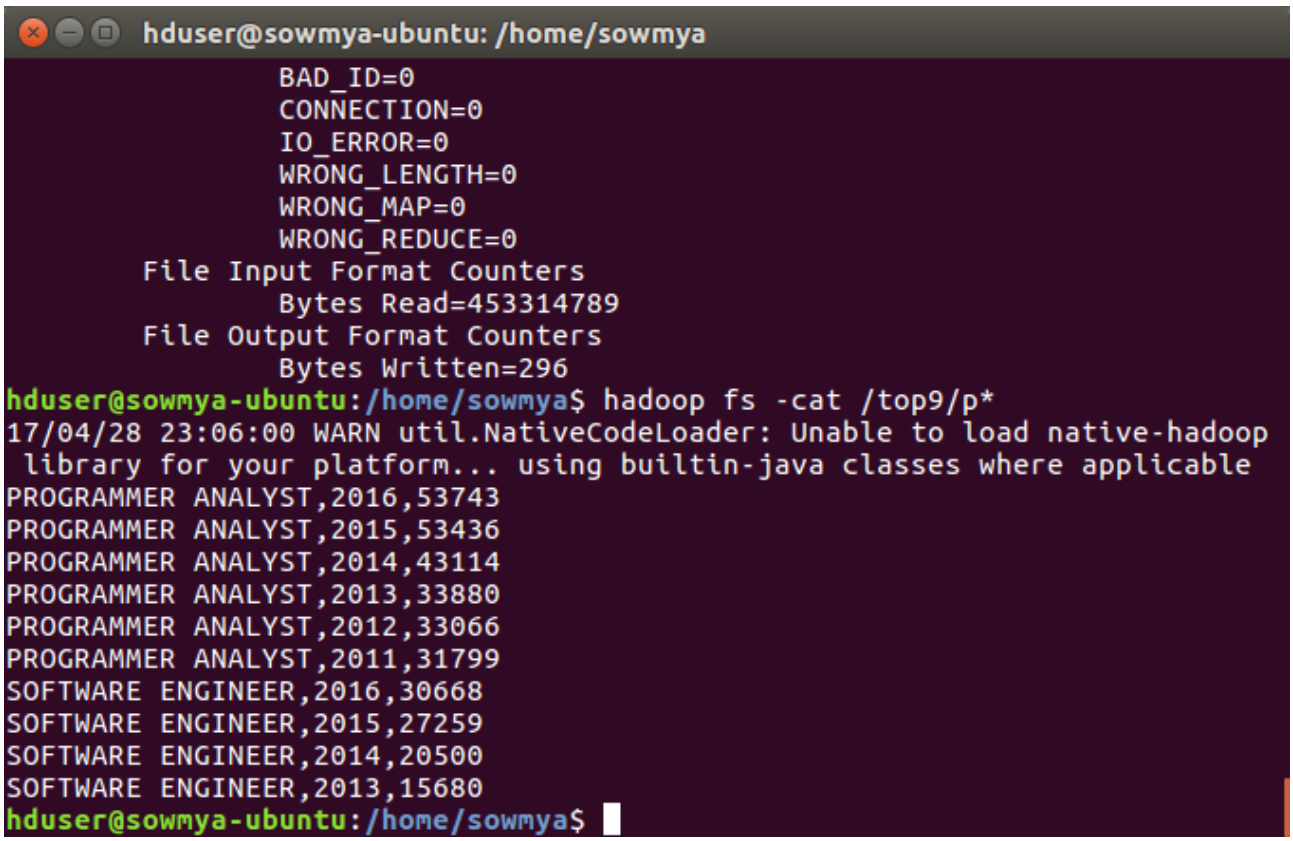
11) Export result for question no 10 to MySql database.

Mapreduce

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

```
import java.io.IOException;
import java.util.TreeMap;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
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 top10positions
public static class Top10Mapper extends Mapper<LongWritable, Text, Text, Text> {
      public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {

                        try {
```

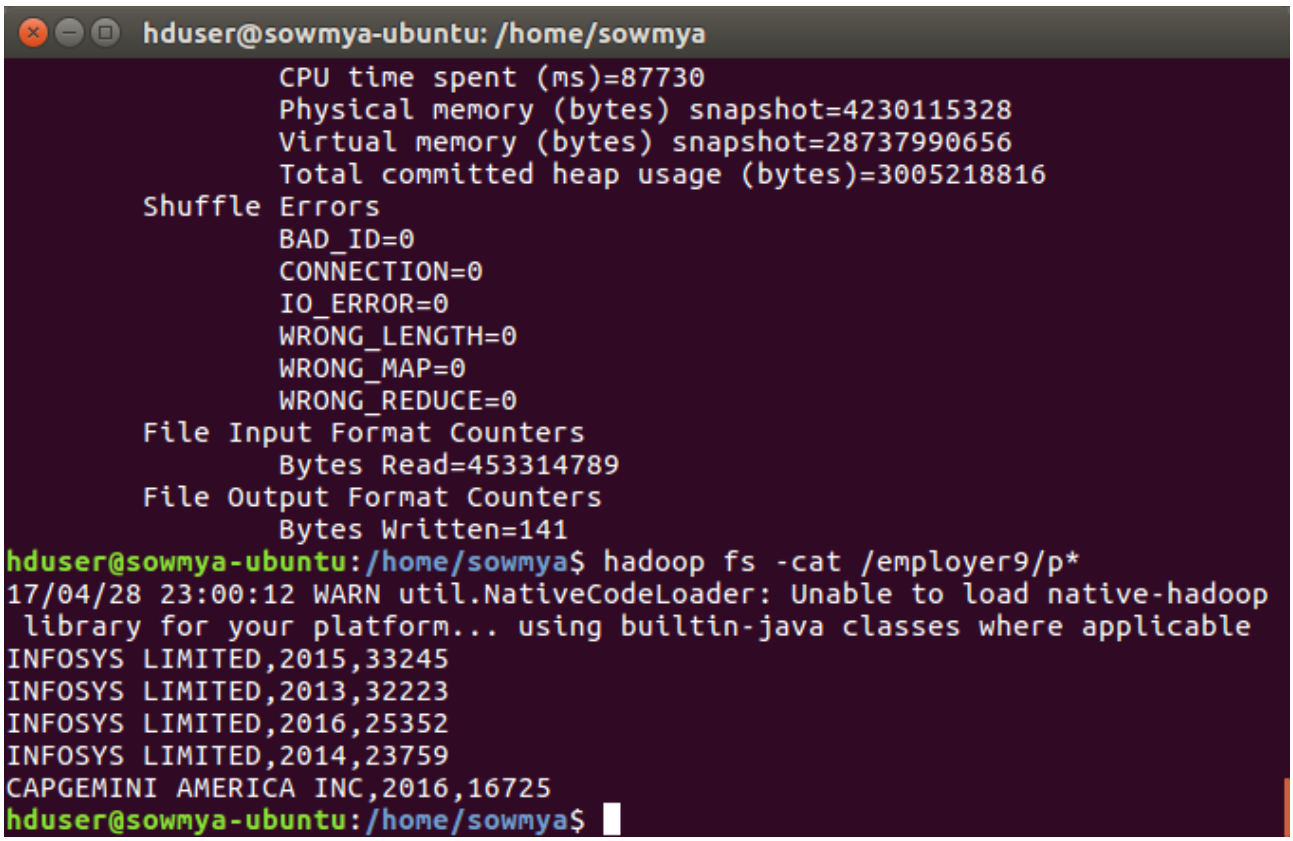
```
                                String[] str = value.toString().split("\t");
                                String position= str[4];
                                int year= Integer.parseInt(str[7]);
                                String status= str[1];
                                String myyear= String.format("%d", year);
                                String myvalue= position + ',' +  myyear;
                                context.write(new Text(myvalue),new Text(status));

                        }
                        catch(Exception e)
                        {
                           System.out.println(e.getMessage());
                        }
                     }
                  }

            public static class Top10Reducer extends
Reducer<Text, Text, NullWritable, Text> {
private TreeMap<Long, Text> repToRecordMap = new TreeMap<Long, Text>();
public void reduce(Text key, Iterable<Text> values,
                                Context context) throws IOException,
InterruptedException {
                           long count=0;
                           String myvalue= "";
                           String mycount= "";
                                 for (Text val : values) {
                                 //String[] token= key.toString().split(",");
                                  count++;
                                 }
                                 myvalue= key.toString();
                                 mycount= String.format("%d", count);
                           myvalue= myvalue + ',' + mycount;
                     repToRecordMap.put(new Long(mycount), new Text(myvalue));
                                 if (repToRecordMap.size() > 10) {
                                 repToRecordMap.remove(repToRecordMap.firstKey());
                           }
                     }

            protected void cleanup(Context context) throws IOException,
                                 InterruptedException {
                     for (Text t : repToRecordMap.descendingMap().values()) {
                     // Output our five records to the file system with a null key
                                       context.write(NullWritable.get(), t);
                                       }
                                 }
            }
public static void main(String[] args) throws Exception {

                     Configuration conf = new Configuration();
                     Job job = Job.getInstance(conf, "Top 10 positions");
                  job.setJarByClass(top10positions.class);
                  job.setMapperClass(Top10Mapper.class);
                  job.setReducerClass(Top10Reducer.class);
                  job.setMapOutputKeyClass(Text.class);
                  job.setMapOutputValueClass(Text.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);
               }
      }
```



2) Which top 5 employers file the most petitions each year?

```
import java.io.IOException;
import java.util.TreeMap;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
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 top5employer {
public static class Top5Mapper extends Mapper<LongWritable, Text, Text, Text> {
      public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {

                              try {
                        String[] str = value.toString().split("\t");
                        String employer_name= str[2];
                        int year= Integer.parseInt(str[7]);
                        String status= str[1];
                        String myyear= String.format("%d", year);
                        String myvalue= employer_name + ',' + myyear;
                        context.write(new Text(myvalue),new Text (status));
```

```
                        }
                        catch(Exception e)
                        {
                           System.out.println(e.getMessage());
                        }
                     }
                  }
               public static class Top5Reducer extends
               Reducer<Text, Text, NullWritable, Text> {
      private TreeMap<Long, Text> repToRecordMap = new TreeMap<Long, Text>();

public void reduce(Text key, Iterable<Text> values,Context context) throws
IOException, InterruptedException {
                        long count=0;
                        String myvalue= "";
                        String mycount= "";
                              for (Text val : values) {
                              //String[] token= key.toString().split(",");
                               count++;
                              }
                              myvalue= key.toString();
                              mycount= String.format("%d", count);
                              myvalue= myvalue + ',' + mycount;
            repToRecordMap.put(new Long(mycount), new Text(myvalue));
                              if (repToRecordMap.size() > 5) {

      repToRecordMap.remove(repToRecordMap.firstKey());

                              }
                     }

protected void cleanup(Context context) throws IOException,InterruptedException
{
            for (Text t : repToRecordMap.descendingMap().values()) {
            // Output our five records to the file system with a null key
                                    context.write(NullWritable.get(), t);
                                    }
                              }
               }
      public static void main(String[] args) throws Exception {

                        Configuration conf = new Configuration();
                        Job job = Job.getInstance(conf, "Top 5 employer");
                     job.setJarByClass(top5employer.class);
                     job.setMapperClass(Top5Mapper.class);
                     job.setReducerClass(Top5Reducer.class);
                     job.setMapOutputKeyClass(Text.class);
                     job.setMapOutputValueClass(Text.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);
                  }
                  }
```



3) Which part of the US has the most Data Engineer jobs for each year?

```
import java.io.IOException;
import java.util.TreeMap;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
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 countengineer {

        public static class datamap extends Mapper<LongWritable,Text,Text,Text> {
                public void map(LongWritable key,Text value,Context context)throws IOException,InterruptedException {
                        String[] word = value.toString().split("\t");
                        String word1 = word[8] +','+ word[7];
                        if(word[4].contains("DATA ENGINEER"))
                        {
```

```
        context.write(new Text(word1), new Text (word[4]));
                        }
                }
        }
        public static class datared extends Reducer<Text,Text,NullWritable,Text>
        {
private TreeMap<Long,Text> repToRecordMap= new TreeMap<Long,Text>();
        public void reduce(Text key, Iterable<Text> values, Context context)throws IOException, InterruptedException
                {
                        long count=0;
                        String myvalue= "";
                        for(Text val:values)
                        {
                                count++;
                        }
                        myvalue= key.toString();
                        String mycount= String.format("%d", count);
                        myvalue= myvalue + ',' + mycount;
                        repToRecordMap.put(new Long(mycount), new Text(myvalue));
                        if(repToRecordMap.size() > 10)
                        {
                                repToRecordMap.remove(repToRecordMap.firstKey());
                        }

                }
protected void cleanup(Context context) throws IOException, InterruptedException
                {
                        for(Text t: repToRecordMap.descendingMap().values())
                        {
                                context.write(NullWritable.get(), t);
                        }
                }
        }

                public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException{
            Configuration conf = new Configuration();
            Job job = Job.getInstance(conf, "data count");
                job.setJarByClass(countengineer.class);
                job.setMapperClass(datamap.class);
                job.setReducerClass(datared.class);
                job.setNumReduceTasks(1);
                job.setMapOutputKeyClass(Text.class);
                job.setMapOutputValueClass(Text.class);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
```

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

        }
    }
```

```
                Map output records=1721
                Map output bytes=79257
                Map output materialized bytes=82783
                Input split bytes=1400
                Combine input records=0
                Combine output records=0
                Reduce input groups=414
                Reduce shuffle bytes=82783
                Reduce input records=1721
                Reduce output records=10
                Spilled Records=3442
                Shuffled Maps =14
                Failed Shuffles=0
                Merged Map outputs=14
                GC time elapsed (ms)=3989
                CPU time spent (ms)=26600
                Physical memory (bytes) snapshot=3948535808
                Virtual memory (bytes) snapshot=28711673856
                Total committed heap usage (bytes)=2899312640
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=453314789
        File Output Format Counters
                Bytes Written=294
hduser@sowmya-ubuntu:~$ hadoop fs -cat /data2/p*
17/04/29 21:09:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
SEATTLE, WASHINGTON,2016,128
SAN FRANCISCO, CALIFORNIA,2016,90
NEW YORK, NEW YORK,2016,70
SEATTLE, WASHINGTON,2015,61
SEATTLE, WASHINGTON,2013,46
SEATTLE, WASHINGTON,2014,45
NEW YORK, NEW YORK,2015,41
MENLO PARK, CALIFORNIA,2016,39
SAN FRANCISCO, CALIFORNIA,2014,34
SEATTLE, WASHINGTON,2012,30
```

4) find top 5 locations in the US who have got certified visa for each year.

```
import java.io.IOException;
import java.util.TreeMap;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
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 certifiedvisa {
public static class Top5Mapper extends Mapper<LongWritable, Text, Text, Text> {
      public void map(LongWritable key, Text value, Context context
                    ) throws IOException, InterruptedException {

                        try {
                              String[] str = value.toString().split("\t");
                              String worksite= str[8];
```

```
                        int year= Integer.parseInt(str[7]);
                        String status= str[1];
                        String myyear= String.format("%d", year);
                        if(str[1].contains("CERTIFIED"))
                        {
                         String myvalue= worksite + ',' +  myyear;
                        context.write(new Text(myvalue),new Text(status));

                    }
                       }
                    catch(Exception e)
                    {
                       System.out.println(e.getMessage());
                    }
                }
              }

            public static class Top5Reducer extends
            Reducer<Text, Text, NullWritable, Text> {
private TreeMap<Long, Text> repToRecordMap = new TreeMap<Long, Text>();

public void reduce(Text key, Iterable<Text> values,Context context) throws
IOException, InterruptedException {
                        long count=0;
                        String myvalue= "";
                        String mycount= "";
                              for (Text val : values) {
                              //String[] token= key.toString().split(",");
                               count++;
                              }
                  myvalue= key.toString();
                  mycount= String.format("%d", count);
                  myvalue= myvalue + ',' + mycount;
                  repToRecordMap.put(new Long(mycount), new Text(myvalue));
                              if (repToRecordMap.size() > 5) {
                              repToRecordMap.remove(repToRecordMap.firstKey());
                                          }
                  }

                  protected void cleanup(Context context) throws IOException,
                              InterruptedException {
            for (Text t : repToRecordMap.descendingMap().values()) {
                  // Output our five records to the file system with a null key
                        context.write(NullWritable.get(), t);
                                    }
                              }
            }
                public static void main(String[] args) throws Exception {

                Configuration conf = new Configuration();
                Job job = Job.getInstance(conf, "Top 10 positions");
                job.setJarByClass(certifiedvisa.class);
                job.setMapperClass(Top5Mapper.class);
                job.setReducerClass(Top5Reducer.class);
                job.setMapOutputKeyClass(Text.class);
                job.setMapOutputValueClass(Text.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);
              }
```

}

```
                Total megabyte-seconds taken by all map tasks=177128448
                Total megabyte-seconds taken by all reduce tasks=27943936
        Map-Reduce Framework
                Map input records=3002458
                Map output records=2818274
                Map output bytes=102443731
                Map output materialized bytes=108080363
                Input split bytes=1400
                Combine input records=0
                Combine output records=0
                Reduce input groups=52284
                Reduce shuffle bytes=108080363
                Reduce input records=2818274
                Reduce output records=5
                Spilled Records=5636548
                Shuffled Maps =14
                Failed Shuffles=0
                Merged Map outputs=14
                GC time elapsed (ms)=4785
                CPU time spent (ms)=83910
                Physical memory (bytes) snapshot=4217241600
                Virtual memory (bytes) snapshot=28742262784
                Total committed heap usage (bytes)=2999451648
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=453314789
        File Output Format Counters
                Bytes Written=150
hduser@sowmya-ubuntu:/home/sowmya$ hadoop fs -cat /cert/p*
17/04/29 22:26:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
NEW YORK, NEW YORK,2016,37746
NEW YORK, NEW YORK,2015,34216
NEW YORK, NEW YORK,2014,30132
NEW YORK, NEW YORK,2012,26160
NEW YORK, NEW YORK,2013,25888
hduser@sowmya-ubuntu:/home/sowmya$
```

HIVE

1) Which are the top 10 job positions which have the highest success rate in petitions?

Select ROUND(((count(s_no)/t.tot)*100),2) as ctn,job_title from h1b_pro,(select count(s_no) as tot from h1b_pro)t where case_status ='CERTIFIED' or case_status= 'CERTIFIED-WITHDRAWN' group by job_title,t.tot order by ctn desc limit 10;

```
hduser@sowmya-ubuntu: /home/sowmya
Ended Job = job_1493542042611_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 31.18 sec   HDFS Read: 492282
873 HDFS Write: 117 SUCCESS
Stage-Stage-6: Map: 2   Cumulative CPU: 42.86 sec   HDFS Read: 492286296 HDFS Wr
ite: 18387619 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 16.54 sec   HDFS Read: 183935
26 HDFS Write: 16521718 SUCCESS
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 6.52 sec   HDFS Read: 1652623
3 HDFS Write: 254 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 37 seconds 100 msec
OK
7.97    PROGRAMMER ANALYST
3.84    SOFTWARE ENGINEER
2.25    COMPUTER PROGRAMMER
1.98    SYSTEMS ANALYST
1.35    SOFTWARE DEVELOPER
1.26    BUSINESS ANALYST
1.11    COMPUTER SYSTEMS ANALYST
0.94    TECHNOLOGY LEAD - US
0.87    TECHNOLOGY ANALYST - US
0.85    SENIOR SOFTWARE ENGINEER
Time taken: 118.878 seconds, Fetched: 10 row(s)
hive>
```

2) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate)

select ROUND((AVG(prevailing_wage)),2), job_title,year,full_time_position from h1b_pro where full_time_position!='NULL' group by full_time_position,job_title,year order by full_time_position desc;

```
68661.0 EPC PLANNER     2016    N
52228.8 ENVIRONMENTALLY PREFERRED PURCHASING COORDINATOR         2016    N
69389.0 ENVIRONMENTAL/PERMITS ENGINEER II       2016    N
50315.0 ENVIRONMENTAL, HEALTH & SAFETY ADVISOR  2016    N
55286.4 ENVIRONMENTAL TEST ENGINEER     2015    N
54433.6 ENVIRONMENTAL TEST ENGINEER     2013    N
50294.4 ENVIRONMENTAL SYSTEMS ANALYST   2012    N
49951.2 ENVIRONMENTAL SPECIALISTS       2016    N
43784.0 ENVIRONMENTAL SPECIALIST, AIR AND WATER (ASSOCIATE 6)   2016    N
35297.6 ENVIRONMENTAL SPECIALIST / RESEARCH ASSOCIATE   2015    N
53071.2 ENVIRONMENTAL SPECIALIST        2015    N
56049.07        ENVIRONMENTAL SPECIALIST        2013    N
42697.2 ENVIRONMENTAL SPECIALIST        2011    N
38709.0 ENVIRONMENTAL SERVICE ENGINEER TECHNICIAN       2016    N
53435.0 ENVIRONMENTAL SCIENTIST/GEOLOGIST       2016    N
60715.0 ENVIRONMENTAL SCIENTIST (ENVIRONMENTAL ENGINEER)        2016    N
50112.59        ENVIRONMENTAL SCIENTIST 2016    N
51168.0 ENVIRONMENTAL SCIENTIST 2014    N
52012.13        ENVIRONMENTAL SCIENTIST 2012    N
48333.0 ENVIRONMENTAL SCIENCE TEACHER   2016    N
61193.6 ENVIRONMENTAL SCIENCE INSTRUCTOR        2013    N
33000.0 ENVIRONMENTAL SCIENCE AND PROTECTION TECHNICIAN 2016    N
39353.6 ENVIRONMENTAL PROJECT MANAGER   2016    N
44325.0 ENVIRONMENTAL PROJECT ENGINEER  2016    N
35464.0 ENVIRONMENTAL PROJECT DESIGNER  2016    N
29494.4 ENVIRONMENTAL PROGRAM ASSISTANT 2016    N
37294.0 ENVIRONMENTAL POLICY ANALYST    2016    N
55827.0 ENVIRONMENTAL PLANNER   2016    N
38438.4 ENVIRONMENTAL MICROBIOLOGIST    2014    N
58177.6 ENVIRONMENTAL MANAGER   2013    N
58011.0 ENVIRONMENTAL INFORMATICS SOFTWARE DEVELOPER    2016    N
52124.8 ENVIRONMENTAL HYGIENIST 2016    N
57907.0 ENVIRONMENTAL HEALTH AND SAFETY SPECIALIST      2016    N
69805.0 ENVIRONMENTAL HEALTH & SAFETY SPECIALIST III    2016    N
99860.8 ENVIRONMENTAL HEALTH & SAFETY SPECIALIST        2011    N
43118.0 ENVIRONMENTAL HEALTH & SAFETY PRACTITIONER      2016    N
40580.8 ENVIRONMENTAL FIELD SCIENTIST   2016    N
65811.0 ENVIRONMENTAL FATE SCIENTIST    2016    N
38563.2 ENVIRONMENTAL EVENT SPECILIST   2014    N
53269.0 ENVIRONMENTAL ENGINEERS 2016    N
34444.8 ENVIRONMENTAL ENGINEERING TECHINCIANS   2016    N
54870.0 ENVIRONMENTAL ENGINEERING SPECIALIST 2  2016    N
44637.0 ENVIRONMENTAL ENGINEER IV       2016    N
```

3) Which are top ten employers who have the highest success rate in petitions?

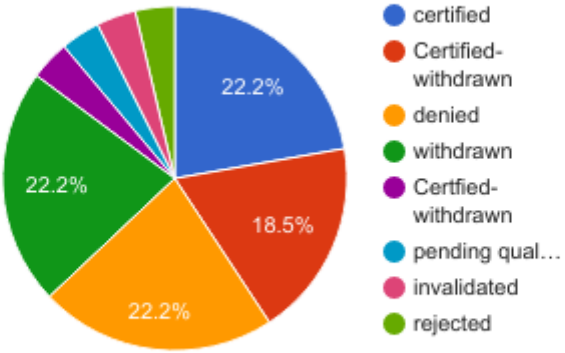
select ROUND(((count(s_no)/t.tot)*100),2) as ctn,employer_name from h1b_pro,(select count(s_no) as tot from h1b_pro)t where case_status ='CERTIFIED' or case_status= 'CERTIFIED-WITHDRAWN' group by employer_name,t.tot order by ctn desc limit 10;

```
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1493474794542_0009
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 2
2017-04-29 20:00:31,413 Stage-1 map = 0%,  reduce = 0%
2017-04-29 20:00:49,152 Stage-1 map = 37%,  reduce = 0%, Cumulative CPU 36.87 sec
2017-04-29 20:00:52,268 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 44.84 sec
2017-04-29 20:00:53,320 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 46.62 sec
2017-04-29 20:01:01,693 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 54.79 sec
2017-04-29 20:01:03,762 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 64.13 sec
MapReduce Total cumulative CPU time: 1 minutes 4 seconds 130 msec
Ended Job = job_1493474794542_0009
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1493474794542_0010, Tracking URL = http://sowmya-ubuntu:8088/proxy/application_1493474794542_0010/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1493474794542_0010
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2017-04-29 20:01:14,566 Stage-2 map = 0%,  reduce = 0%
2017-04-29 20:01:19,926 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.41 sec
2017-04-29 20:01:26,154 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.31 sec
MapReduce Total cumulative CPU time: 7 seconds 310 msec
Ended Job = job_1493474794542_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 2   Cumulative CPU: 64.13 sec   HDFS Read: 492290928 HDFS Write: 11502572 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.31 sec   HDFS Read: 11507350 HDFS Write: 271 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 11 seconds 440 msec
OK
4.33    INFOSYS LIMITED
2.14    TATA CONSULTANCY SERVICES LIMITED
1.58    WIPRO LIMITED
1.2     DELOITTE CONSULTING LLP
1.11    ACCENTURE LLP
1.0     IBM INDIA PRIVATE LIMITED
0.84    MICROSOFT CORPORATION
0.75    HCL AMERICA, INC.
0.6     ERNST & YOUNG U.S. LLP
0.56    LARSEN & TOUBRO INFOTECH LIMITED
Time taken: 61.774 seconds, Fetched: 10 row(s)
hive>
```

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

select count(s_no) as count,round(((count(s_no)/t.tot)*100),2) as ctn,year,case_status from h1b_pro,(select count(s_no) as tot from h1b_pro)t where year!='NULL' or year!='NA' group by case_status,year,t.tot order by ctn desc;

```
2017-04-29 20:21:02,342 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.13 sec
2017-04-29 20:21:07,547 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.1 sec
MapReduce Total cumulative CPU time: 3 seconds 100 msec
Ended Job = job_1493477287371_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 2   Cumulative CPU: 38.83 sec   HDFS Read: 492289698 HDFS Write: 1630 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.1 sec   HDFS Read: 6798 HDFS Write: 883 SUCCESS
Total MapReduce CPU Time Spent: 41 seconds 930 msec
OK
569646  18.97   2016    CERTIFIED
547278  18.23   2015    CERTIFIED
455144  15.16   2014    CERTIFIED
382948  12.75   2013    CERTIFIED
352667  11.75   2012    CERTIFIED
307936  10.26   2011    CERTIFIED
47092   1.57    2016    CERTIFIED-WITHDRAWN
41071   1.37    2015    CERTIFIED-WITHDRAWN
36349   1.21    2014    CERTIFIED-WITHDRAWN
35432   1.18    2013    CERTIFIED-WITHDRAWN
31117   1.04    2012    CERTIFIED-WITHDRAWN
29130   0.97    2011    DENIED
21890   0.73    2016    WITHDRAWN
21096   0.7     2012    DENIED
19455   0.65    2015    WITHDRAWN
16034   0.53    2014    WITHDRAWN
12126   0.4     2013    DENIED
11896   0.4     2014    DENIED
11589   0.39    2013    WITHDRAWN
11596   0.39    2011    CERTIFIED-WITHDRAWN
10923   0.36    2015    DENIED
10725   0.36    2012    WITHDRAWN
10105   0.34    2011    WITHDRAWN
9175    0.31    2016    DENIED
1       0.0     2014    INVALIDATED
13      0.0     NA      NA
15      0.0     2013    PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED
2       0.0     NULL    CERTIFIED-WITHDRAWN
1       0.0     NULL    WITHDRAWN
4       0.0     NULL    CERTIFIED
2       0.0     2014    REJECTED
1       0.0     YEAR    CASE_STATUS
Time taken: 51.931 seconds, Fetched: 32 row(s)
hive>
```



PIG

1)Is the number of petitions with Data Engineer job title increasing over time?

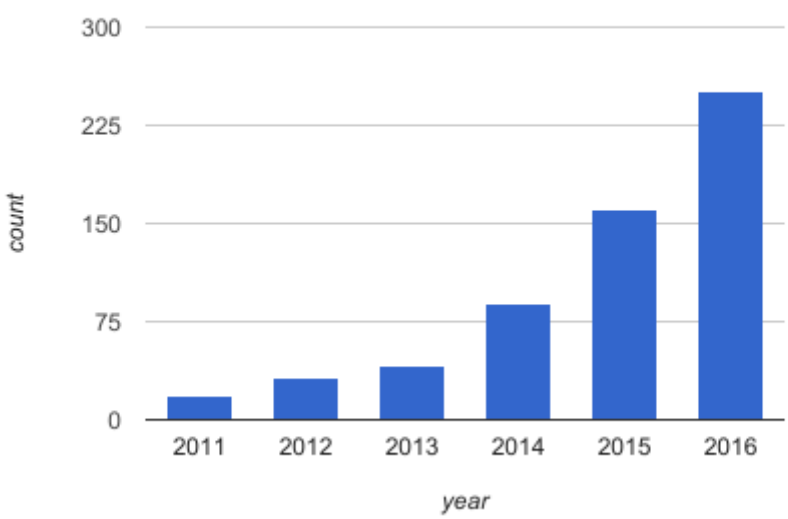
h1b= LOAD '/home/sowmya/Desktop/mapreduce' using PigStorage('\t') AS (s_no:int,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevailing_wage,year, worksite1:chararray,longitute, latitute);

filter_bag= filter h1b by job_title== 'DATA ENGINEER';

group_all= group filter_bag by year;

count_all= foreach group_all generate $0, COUNT(filter_bag) as headcount;

```
hduser@sowmya-ubuntu: ~
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-04-27 14:15:07,417 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-04-27 14:15:07,417 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.addr
ess
2017-04-27 14:15:07,417 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2017-04-27 14:15:07,422 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2017-04-27 14:15:07,423 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(2011,18)
(2012,32)
(2013,41)
(2014,89)
(2015,160)
(2016,251)
grunt> describe filter_bag;
filter_bag: {s_no: int,case_status: chararray,employer_name: chararray,soc_name:
 chararray,job_title: chararray,full_time_position: chararray,prevailing_wage: b
ytearray,year: bytearray,worksite1: chararray,longitute: bytearray,latitute: byt
earray}
grunt>
```



2) Find top 5 job titles who are having highest growth in applications.

h1b= LOAD '/home/sowmya/Desktop/mapreduce' using PigStorage('\t') AS (s_no:int,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevailing_wage,year,worksite1:chararray,longitute, latitute);

groupbyyear= GROUP h1b by (year, job_title);

countcust= foreach groupbyyear generate group as year, COUNT(h1b) as headcount;

orderbycount = order countcust by $1 desc;

limit2= limit orderbycount 5;

dump limit2;

```
ER
job_local1397506383_0003        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     orderbycount    SAMPLER
job_local1823924649_0005        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     orderbycount            file:/t
mp/temp-116192712/tmp1580809765,
job_local877122179_0002 14      1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     countcust,groupbyyear,h1b       GROUP_B
Y,COMBINER

Input(s):
Successfully read 3002458 records from: "/home/sowmya/Desktop/mapreduce"

Output(s):
Successfully stored 5 records in: "file:/tmp/temp-116192712/tmp1580809765"

Counters:
Total records written : 5
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local877122179_0002 ->      job_local1397506383_0003,
job_local1397506383_0003        ->      job_local1365126249_0004,
job_local1365126249_0004        ->      job_local1823924649_0005,
job_local1823924649_0005


2017-04-30 20:22:22,307 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-04-30 20:22:22,308 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.b
ytes-per-checksum
2017-04-30 20:22:22,309 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultF
S
2017-04-30 20:22:22,309 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduc
e.jobtracker.address
2017-04-30 20:22:22,309 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-04-30 20:22:22,317 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-04-30 20:22:22,317 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
((2016,PROGRAMMER ANALYST),53743)
((2015,PROGRAMMER ANALYST),53436)
((2014,PROGRAMMER ANALYST),43114)
((2013,PROGRAMMER ANALYST),33880)
((2012,PROGRAMMER ANALYST),33066)
grunt>
```

3)Which industry has the most number of Data Scientist positions?

h1b= LOAD '/home/sowmya/Desktop/mapreduce' using PigStorage('\t') AS (s_no:int,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevailing_wage,year, worksite1:chararray,longitute, latitute);

filteration= filter h1b by job_title == 'DATA SCIENTIST';

groupbyemployer= group filteration by employer_name;

countbypos= foreach groupbyemployer generate $0, COUNT(filteration) as headcount;

dump countbypos;

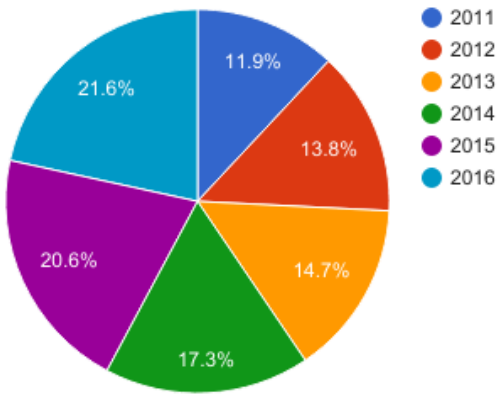
```
(UNIVISION INTERACTIVE MEDIA, INC.,1)
(VOICEBOX TECHNOLOGIES CORPORATION,1)
(AMERICAN EXPRESS TRS, COMPANY INC.,1)
(BLUESTAR TECHNOLOGY SOLUTIONS, LLC,1)
(MOTION PICTURES LABORATORIES, INC.,1)
(OMNICOM MEDIA GROUP HOLDINGS, INC.,1)
(TRENDALYTICS INNOVATION LABS, INC.,2)
(CHAR SOFTWARE INC. (DBA LOCALYTICS),1)
(ENDURANCE INTERNATIONAL GROUP, INC.,1)
(PIONEER HI-BRED INTERNATIONAL, INC.,3)
(RUTHS ANALYTICS AND INNOVATION, LLC,1)
(SCHLUMBERGER TECHNOLOGY CORPORATION,11)
(CHILDREN'S HOSPITAL OF ORANGE COUNTY,1)
(E. I. DU PONT DE NEMOURS AND COMPANY,1)
(EXPERIAN INFORMATION SOLUTIONS, INC.,2)
(INSTITUTE FOR MEDICAL RESEARCH, INC.,1)
(T3C, INC. DBA RETAIL SOLUTIONS, INC.,1)
(COMPREHENSIVE HEALTH MANAGEMENT, INC.,1)
(MASTERCARD INTERNATIONAL INCORPORATED,2)
(SEARS HOLDINGS MANAGEMENT CORPORATION,1)
(DAVIDSON KEMPNER CAPITAL MANAGEMENT LP,1)
(GEMOLOGICAL INSTITUTE OF AMERICA (GIA),1)
(MCKINSEY & COMPANY, INC. UNITED STATES,2)
(DAVIDSON KEMPNER CAPITAL MANAGEMENT, LP,1)
(FANATICS RETAIL GROUP FULFILLMENT, INC.,5)
(GLOBAL TOUCHPOINTS, INC. DUNS# 13-8058305,1)
(ACCIDENT FUND INSURANCE COMPANY OF AMERICA,2)
(HEALTHCARE ANALYTICS SERVICES HOLDING INC.,1)
(UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL,1)
(AXIAL NETWORKS, INC. (FORMERLY AXIAL MARKET),1)
(HOUGHTON MIFFLIN HARCOURT PUBLISHING COMPANY,1)
(SONY NETWORK ENTERTAINMENT INTERNATIONAL LLC,2)
(CAESARS ENTERTAINMENT OPERATING COMPANY, INC.,1)
(PHILIPS ELECTRONICS NORTH AMERICA CORPORATION,1)
(PLAYDOM, INC., PART OF THE WALT DISNEY COMPANY,1)
(HEALTHCARE BUSINESS INTELLIGENCE SOLUTIONS INC.,1)
(NOVARTIS INSTITUTE FOR FUNCTIONAL GENOMICS, INC.,1)
(AMERICAN EXPRESS TRAVEL RELATED SERVICES COMPANY,,2)
(ADVANCED INFORMATION MANAGEMENT TECHNOLOGY PARTNER,2)
(AMERICAN EXPRESS TRAVEL RELATED SERVICES CO., INC.,1)
(CAPITAL IQ, INC. (SUBSIDIARY OF THE MCGRAW-HILL CO,1)
(MORNINGSTAR, INC. (HELLOWALLET, A MORNINGSTAR COMPANY),1)
grunt>
```

4) Create a bar graph to depict the number of applications for each year

h1b= LOAD '/home/sowmya/Desktop/mapreduce' using PigStorage('\t') AS (s_no:int,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevailing_wage,year, worksite1:chararray,longitute, latitute);

groupbyyear= GROUP h1b by year;

countapp= foreach groupbyyear generate group as year, COUNT(h1b) as headcount;



```
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianR
educetime       Alias   Feature Outputs
job_local1271765203_0001        14      1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     countapp,groupbyyear,h1b      G
ROUP_BY,COMBINER        file:/tmp/temp-1048292517/tmp422138752,

Input(s):
Successfully read 3002458 records from: "/home/sowmya/Desktop/mapreduce"

Output(s):
Successfully stored 15 records in: "file:/tmp/temp-1048292517/tmp422138752"

Counters:
Total records written : 15
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1271765203_0001


2017-04-29 20:37:19,431 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-04-29 20:37:19,434 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.b
ytes-per-checksum
2017-04-29 20:37:19,435 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultF
S
2017-04-29 20:37:19,435 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduc
e.jobtracker.address
2017-04-29 20:37:19,435 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-04-29 20:37:19,449 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-04-29 20:37:19,449 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(NA,13)
(\N,7)
(2011,358766)
(2012,415602)
(2013,442110)
(2014,519426)
(2015,618725)
(2016,647802)
```

SQOOP

10) Which are the top 10 job positions which have the highest success rate in petitions?

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

Make an input file for the top 10 job positions who have highest success rate in petitions and put it in HDFS. The command for it:

hduser@sowmya-ubuntu:~$ hadoop fs -put /home/sowmya/visa.txt /niit

Enter the mysql server: mysql -u root -p

Create a table in Mysql to export data from HDFS:

```
create table h1b_app(
    success_rate FLOAT NOT NULL,
     position VARCHAR(40) NOT NULL,
     status VARCHAR(40) NOT NULL,
PRIMARY KEY(success_rate));
```

SQOOP QUERY:

```
sqoop export --connect jdbc:mysql://localhost/app --username root -P --table h1b_app --update-mode allowinsert --update-key success_rate --export-dir /niit/visa.txt --input-fields-terminated-by ',';
```

```
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
17/04/29 23:20:28 INFO mapreduce.ExportJobBase: Transferred 1.1367 KB in 38.2204 seconds (30.4549 bytes/sec)
17/04/29 23:20:28 INFO mapreduce.ExportJobBase: Exported 8 records.
17/04/29 23:20:28 ERROR tool.ExportTool: Error during export: Export job failed!
hduser@sowmya-ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.18-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use app;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from h1b_app;
+--------------+--------------------------+------------+
| success_rate | position                 | status     |
+--------------+--------------------------+------------+
|         0.78 | SENIOR SOFTWARE ENGINEER | CERTIFIED  |
|         0.87 | TECNOLOGY ANALYST        | CERTIFIED  |
|         0.94 | TECHNICAL LEAD-US        | CERTIFIED  |
|         1.05 | COMPUTER SYSTEMS ANALYST | CERTIFIED  |
|         1.17 | BUSINESS ANALYST         | CERTIFIED  |
|         1.26 | SOFTWARE DEVELOPER       | CERTIFIED  |
|         1.86 |  SOFTWARE ANALYST        | CERTIFIED  |
|         2.13 |  COMPUTER PROGRAMMER     |  CERTIFIED |
|         3.43 |  SOFTWARE ENGINEER       |   CERTIFED |
|         7.42 |  PROGRAMMER ANALYST      |  CERTIFIED |
+--------------+--------------------------+------------+
10 rows in set (0.00 sec)

mysql>
```