import java.io.IOException;

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.Mapper;

import org.apache.hadoop.mapreduce.Partitioner;

import org.apache.hadoop.mapreduce.Reducer;

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

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

public class Que {

public static class Mymapper 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("\t");

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

{

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

context.write(ans,one);

}

}

}

}

public static class Mypart extends Partitioner<Text,LongWritable>

{

public int getPartition(Text key,LongWritable values,int numReduceTasks)

{

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

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

return 0;

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

return 1;

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

return 2;

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

return 3;

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

return 4;

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

return 5;

else

return 6;

}

}

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

{

public void reduce(Text tt,Iterable<LongWritable> value,Context cc)throws IOException,InterruptedException

{

long total=0;

for(LongWritable dd:value)

{

total+=dd.get();

}

cc.write(tt,new LongWritable(total));

}

}

public static void main(String[] args)

throws IOException, ClassNotFoundException, InterruptedException

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf);

job.setJarByClass(Que.class);

job.setMapperClass(Mymapper.class);

job.setPartitionerClass(Mypart.class);

job.setReducerClass(Myred.class);

job.setNumReduceTasks(6);

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);

}

}