package F33;

import java.io.IOException;

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.input.MultipleInputs;

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

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

import org.apache.hadoop.util.GenericOptionsParser;

public class F333

{

public static class Map1 extends Mapper<LongWritable,Text,Text,IntWritable>

{

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

{

String line = value.toString();

String[] line1=line.split(",");

String gender=line1[3];

Text outputKey = new Text(gender);

int salary=Integer.parseInt(line1[2]);

IntWritable outputValue = new IntWritable(salary);

con.write(outputKey, outputValue);

}

}

public static class Map2 extends Mapper<LongWritable,Text,Text,IntWritable>

{

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

{

String line = value.toString();

String[] line1=line.split(",");

String gender=line1[2];

Text outputKey = new Text(gender);

int salary=Integer.parseInt(line1[3]);

IntWritable outputValue = new IntWritable(salary);

con.write(outputKey, outputValue);

}

}

public static class Red extends Reducer<Text,IntWritable,Text,IntWritable>

{

public void reduce(Text prod\_id, Iterable<IntWritable> total\_sales, Context con)

throws IOException , InterruptedException

{

int sum = 0;

for(IntWritable value : total\_sales)

{

sum += value.get();

}

con.write(prod\_id, new IntWritable(sum));

}

}

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

{

Configuration c=new Configuration();

String[] files=new GenericOptionsParser(c,args).getRemainingArgs();

Path p1=new Path(files[0]);

Path p2=new Path(files[1]);

Path p3=new Path(files[2]);

Job j = new Job(c,"multiple");

j.setJarByClass(F333.class);

j.setMapperClass(Map1.class);

j.setMapperClass(Map2.class);

j.setReducerClass(Red.class);

j.setOutputKeyClass(Text.class);

j.setOutputValueClass(IntWritable.class);

MultipleInputs.addInputPath(j, p1, TextInputFormat.class, Map1.class);

MultipleInputs.addInputPath(j,p2, TextInputFormat.class, Map2.class);

FileOutputFormat.setOutputPath(j, p3);

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

}

}

Inputf31

1,A,1000,M

2,B,2000,F

3,C,1500,M

4,D,2500,F

Inputf32

5,L,M,5000

6,G,F,4500

7,H,F,7000

8,I,M,1000

Outputf3

F 16000

M 8500