// ASSIGNMENT B2 | PARTH SUPEKAR 20CO130

// SalesCountryRunner.java

package SalesCountry;

import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.\*; import org.apache.hadoop.mapred.\*;

public class SalesCountryRunner { public static void main(String[] args) {

JobClient my\_client = new JobClient();

// Create a configuration object for the job

JobConf job\_conf = new JobConf(SalesCountryDriver.class);

// Set a name of the Job job\_conf.setJobName("SalePerCountry");

// Specify data type of output key and value job\_conf.setOutputKeyClass(Text.class); job\_conf.setOutputValueClass(IntWritable.class);

// Specify names of Mapper and Reducer Class job\_conf.setMapperClass(SalesCountry.SalesMapper.class); job\_conf.setReducerClass(SalesCountry.SalesCountryReducer.class);

// Specify formats of the data type of Input and output job\_conf.setInputFormat(TextInputFormat.class); job\_conf.setOutputFormat(TextOutputFormat.class);

// Set input and output directories using command line arguments,

//arg[0] = name of input directory on HDFS, and arg[1] = name of output

directory to be created to store the output file.

FileInputFormat.setInputPaths(job\_conf, new Path(args[0])); FileOutputFormat.setOutputPath(job\_conf, new Path(args[1]));

my\_client.setConf(job\_conf); try {

// Run the job

JobClient.runJob(job\_conf);

} catch (Exception e) {

e.printStackTrace();

}

}

}

// SalesMapper.java

package SalesCountry;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.LongWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.\*;

public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> { private final static IntWritable one = new IntWritable(1);

public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable>

output, Reporter reporter) throws IOException {

String valueString = value.toString(); String[] SingleCountryData = valueString.split(","); output.collect(new Text(SingleCountryData[7]), one);

}

}

// SalesCountryReducer.java

package SalesCountry;

import java.io.IOException; import java.util.\*;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.\*;

public class SalesCountryReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

public void reduce(Text t\_key, Iterator<IntWritable> values,

OutputCollector<Text,IntWritable> output, Reporter reporter) throws IOException { Text key = t\_key; int frequencyForCountry = 0; while (values.hasNext()) {

// replace type of value with the actual type of our value IntWritable value = (IntWritable) values.next(); frequencyForCountry += value.get();

} output.collect(key, new IntWritable(frequencyForCountry));

}

}

Output:

|  |  |  |  |
| --- | --- | --- | --- |
| Argentina 1  Australia 38  Austria 7  Bahrain 1  Belgium 8  Bermuda 1  Brazil5  Bulgaria 1  CO 1  Canada76  Cayman Isls 1  China 1  Costa Rica 1  Country 1 | | |  |
| Czech Republic  Denmark 15 | | | 3 |
| Dominican Republic | | | 1 |
| Finland  France27 | 2 | |
| Germany  Greece1 | 25 | |  |
| Guatemala | 1 | |  |
| Hong Kong | 1 | |  |
| Hungary | 3 | |  |
| Iceland  India 2 | 1 | |  |
| Ireland  Israel1  Italy 15  Japan 2  Jersey1  Kuwait1  Latvia1 | 49 | |  |
| Luxembourg | 1 | |  |
| Malaysia  Malta 2 | 1 | |  |
| Mauritius | 1 | |  |
| Moldova  Monaco2 | 1 | |  |
| Netherlands | 22 | |  |
| New Zealand  Norway16 | 6 | |  |
| Philippines  Poland2 | 2 | |  |
| Romania  Russia1 | 1 | |  |
| South Africa 5 | | |
| South Korea  Spain 12  Sweden13 | 1 |
| Switzerland | 36 |
| Thailand | 2 |
| The Bahamas  Turkey6 | 2 |
| Ukraine | 1 |

United Arab Emirates 6

United Kingdom 100

United States462