

## **Big Data Assignment 2**

**Code:**

**Sales Mapper Code:**

```
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);

    }

}
```

## **Sales Country Driver Code:**

```
package SalesCountry;

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

public class SalesCountryDriver {
    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();
}
}
```

### **Sales Country Reducer Code:**

```
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:

