Code:

```
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.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));
                 }
       }
SalesCountryDriver.java
 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
```

import org.apache.hadoop.io.IntWritable;

```
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 <math>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();
              }
       }
}
```

SalesJan2009.csv

	Transaction_date	Product	Price	Payment_Type	Name	City	State	Country	Account_Created	Last_Login	Latitude
)	1/2/09 6:17	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdom	1/2/09 6:00	1/2/09 6:08	51.5
	1/2/09 4:53	Product1	1200	Visa	Betina	Parkville	MO	United States	1/2/09 4:42	1/2/09 7:49	39.195
	1/2/09 13:08	Product1	1200	Mastercard	Federica e Andrea	Astoria	OR	United States	1/1/09 16:21	1/3/09 12:32	46.18806
	1/3/09 14:44	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	9/25/05 21:13	1/3/09 14:22	-36.133333
	1/4/09 12:56	Product2	3600	Visa	Gerd W	Cahaba Heights	AL	United States	11/15/08 15:47	1/4/09 12:45	33.52056
	1/4/09 13:19	Product1	1200	Visa	LAURENCE	Mickleton	NJ	United States	9/24/08 15:19	1/4/09 13:04	39.79
3	1/4/09 20:11	Product1	1200	Mastercard	Fleur	Peoria	L	United States	1/3/09 9:38	1/4/09 19:45	40.69361

Output :

part-00000

Argentina

1

38 Australia Austria7 Bahrain1 Belgium 8 Bermuda 1 Brazil 5 Bulgaria 1 CO 1 Canada 76 Cayman Isls China 1 Costa Rica Country 1 Czech Republic 3 Denmark 15 Dominican Republic Finland 2 France 27 Germany 25 Greece 1 Guatemala 1 Hong Kong Hungary 3 Iceland1 India 2

Ireland 49

```
Israel 1
```

Italy 15

Japan 2

Jersey 1

Kuwait 1

Latvia 1

Luxembourg 1

Malaysia 1

Malta 2

Mauritius 1

Moldova1

Monaco 2

Netherlands 22

New Zealand 6

Norway 16

Philippines 2

Poland 2

Romania1

Russia 1

South Africa 5

South Korea 1

Spain 12

Sweden 13

Switzerland 36

Thailand 2

The Bahamas 2

Turkey 6

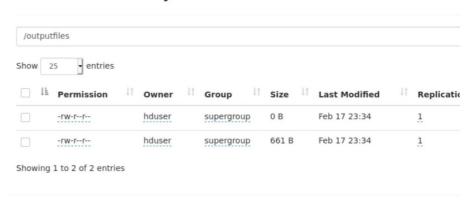
Ukraine1

United Arab Emirates 6

United Kingdom 100

United States 462

Browse Directory



Hadoop, 2017.