

SalesJan2009.CSV

Transaction_date	Product	Price	Payment_Type	Name	City	State	Country	Account_Created	Last_Login	Latitude	Longitude
#####	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdom	#####	#####	51.5	1.11667
#####	Product1	1200	Visa	Betina	Parkville	MO	United States	#####	#####	39.195	94.6819
#####	Product1	1200	Mastercard	Federica Andrea	Astoria	OR	United States	#####	#####	46.18806	123.83
#####	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	9/25/05 21:13	#####	36.1333	144.75
#####	Product2	3600	Visa	Gerd W	Cahaba Heights	AL	United States	11/15/08 15:47	#####	33.52056	86.8025
#####	Product1	1200	Visa	LAUREN CE	Mickleton	NJ	United States	9/24/08 15:19	#####	39.79	75.2381
#####	Product1	1200	Mastercard	Fleur	Peoria	IL	United States	#####	#####	40.69361	89.5889
#####	Product1	1200	Mastercard	adam	Martin	TN	United States	#####	#####	36.34333	88.8503
#####	Product1	1200	Mastercard	Renee Elisabeth	Tel Aviv	Tel Aviv	Israel	#####	#####	32.06667	34.76667
#####	Product1	1200	Visa	Aidan	Chatou	Ile-de-France	France	#####	#####	48.88333	2.15
#####	Product1	1200	Diners	Stacy	New York	NY	United States	#####	#####	40.71417	74.0064
#####	Product1	1200	Amex	Heidi	Eindhoven	Noord-Brabant	Netherlands	#####	#####	51.45	5.46667
#####	Product1	1200	Mastercard	Sean	Shavano Park	TX	United States	#####	#####	29.42389	98.4933
#####	Product1	1200	Visa	Georgia	Eagle	ID	United States	#####	#####	43.69556	116.353
#####	Product1	1200	Visa	Richard	Riverside	NJ	United States	#####	#####	40.03222	74.9578
#####	Product1	1200	Diners	Leanne	Julianstown	Meath	Ireland	#####	#####	53.67722	6.31917
#####	Product1	1200	Visa	Janet	Ottawa	Ontario	Canada	#####	#####	45.41667	-75.7
#####	Product1	1200	Diners	barbara	Hyderabad	Andhra Pradesh	India	#####	#####	17.38333	78.46667
#####	Product2	3600	Visa	Sabine	London	England	United Kingdom	#####	#####	51.52721	0.14559
#####	Product1	1200	Diners	Hani	Salt Lake City	UT	United States	12/30/08 5:44	#####	40.76083	111.89
#####	Product1	1200	Visa	Jeremy	Manchester	England	United Kingdom	#####	#####	53.5	2.21667
#####	Product1	1200	Diners	Janis	Ballynora	Cork	Ireland	#####	#####	51.86306	-8.58
#####	Product1	1200	Mastercard	Nicola	Roodepoort	Gauteng	South Africa	#####	#####	26.1667	27.86667

```
//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
```

```
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.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
        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();
        }
    }
}
```