**A2.mapreduce  
  
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));

}

}

**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();

}

}

}

**code**1) ls –al  
2) sudo chmod +r \*.\*  
  
  
4) javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java -cp $(hadoop classpath)  
//\*\*\*\* create a new file Where you have created map reduce folder (Manifest.txt)  
5) sudo gedit Manifest.txt  
**ADD THIS LINE IN THAT TEXT FILE** Main-Class: SalesCountry.SalesCountryDriver

7) jar cfm ProductSalePerCountry.jar Manifest.txt SalesCountry/\*.class  
8) $HADOOP\_HOME/sbin/start-dfs.sh

9) $HADOOP\_HOME/sbin/start-yarn.sh

10) $HADOOP\_HOME/bin/hdfs dfs -copyFromLocal ~/inputMapReduce /

11) $HADOOP\_HOME/bin/hdfs dfs -ls /inputMapReduce

12) $HADOOP\_HOME/bin/hadoop jar ProductSalePerCountry.jar /inputMapReduce /mapreduce\_output\_sales

13) $HADOOP\_HOME/bin/hdfs dfs -cat /mapreduce\_output\_sales/part-00000

Localhost:54310  
  
  
in worst case: https://www.guru99.com/create-your-first-hadoop-program.html