**We have a dataset of sales of different TV sets across different locations.**

**Records look like:**

**Samsung|Optima|14|Madhya Pradesh|132401|14200**

**The fields are arranged like:**

**Company Name|Product Name|Size in inches|State|Pin Code|Price**

**There are some invalid records which contain 'NA' in either Company Name or Product Name.**

**1. Write a Map Reduce program to filter out the invalid records.**

**The output of this program will act as input for subsequent tasks.**

**Map only job will fit for this context.**

MapperProgram.java  
  
package assignment;  
  
  
  
import org.apache.hadoop.fs.Path;   
import org.apache.hadoop.conf.\*;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;   
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;   
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;   
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
  
public class MapperProgram{  
    @SuppressWarnings("deprecation")  
    public static void main(String[] args) throws Exception {  
        Configuration conf = new Configuration();  
        Job job = new Job(conf, "DemoTask4");  
        job.setJarByClass(MapperProgram.class);  
  
        job.setMapOutputKeyClass(NullWritable.class);  
        job.setMapOutputValueClass(Text.class);  
  
        job.setOutputKeyClass(NullWritable.class);  
        job.setOutputValueClass(Text.class);  
        job.setMapperClass(Task1Mapper.class);  
          
        job.setNumReduceTasks(0);  
          
        job.setInputFormatClass(TextInputFormat.class);  
        job.setOutputFormatClass(TextOutputFormat.class);  
  
        FileInputFormat.addInputPath(job, new Path(args[0]));   
        FileOutputFormat.setOutputPath(job,new Path(args[1]);  
          
        job.waitForCompletion(true);  
    }  
}  
  
**=============================================================**

**task1mapper.java**

package assignment;  
  
import java.io.IOException;  
  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
  
public class Task1Mapper extends Mapper<LongWritable, Text, NullWritable, Text> {  
     public void map(LongWritable key, Text value, Context context)   
       throws IOException, InterruptedException {  
      String[] lineArray = value.toString().split("\\|");  
      if((lineArray.length > 0) &&(lineArray[0] != null) & (lineArray[1] != null) &&  
        (! lineArray[0].equalsIgnoreCase("NA")) &&(! lineArray[1].equalsIgnoreCase("NA"))  
        ) {  
        context.write(NullWritable.get(), value);  
       }  
     }  
    }

**EXECUTION:**

**[acadgild@localhost training]$ hadoop jar a1.jar /user/acadgild/television.txt /tele**

**[acadgild@localhost training]$ hadoop fs -cat /tele**  
17/04/26 15:50:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
  
**[acadgild@localhost training]$ hadoop fs -cat /tele/part-m-00000**  
17/04/26 15:51:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Samsung|Optima|14|Madhya Pradesh|132401|14200  
Onida|Lucid|18|Uttar Pradesh|232401|16200  
Akai|Decent|16|Kerala|922401|12200  
Lava|Attention|20|Assam|454601|24200  
Zen|Super|14|Maharashtra|619082|9200  
Samsung|Optima|14|Madhya Pradesh|132401|14200  
Onida|Lucid|18|Uttar Pradesh|232401|16200  
Onida|Decent|14|Uttar Pradesh|232401|16200  
Lava|Attention|20|Assam|454601|24200  
Zen|Super|14|Maharashtra|619082|9200  
Samsung|Optima|14|Madhya Pradesh|132401|14200  
Samsung|Decent|16|Kerala|922401|12200  
Lava|Attention|20|Assam|454601|24200  
Samsung|Super|14|Maharashtra|619082|9200  
Samsung|Super|14|Maharashtra|619082|9200  
Samsung|Super|14|Maharashtra|619082|9200