**Problem:**

10. Enhance the task 8 (refer session 6, assignment 1) to calculate the top 3 state-wise sales for each company.

You may use multiple reducers for this activity.

**Task1.java**

job.setJarByClass(Task1.**class**);

job.setMapOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(IntWritable.**class**);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(IntWritable.**class**);

job.setMapperClass(Task1Mapper.**class**);

job.setReducerClass(Task1Reducer.**class**);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

**Task1Mapper.java**

IntWritable one=**new** IntWritable();

Text companyState = **new** Text();

**public** **void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

String[] lineArray = value.toString().split("\\|");

**if**(!("NA".equalsIgnoreCase(lineArray[0])) &&!("NA".equalsIgnoreCase(lineArray[1])))

{

Text companyState = **new** Text(lineArray[0]+"\t"+lineArray[3]);

one.set(1);

context.write(companyState, one);

}

}

**Task1Reducer.java**

**public** **class** Task1Reducer **extends** Reducer<Text, IntWritable, Text, IntWritable>

{

**public** **void** reduce(Text key, Iterable<IntWritable> values,Context context) **throws** IOException, InterruptedException

{

**int** unitSold=0;

**for** (IntWritable value : values) {

unitSold=unitSold+value.get();

}

context.write(key, **new** IntWritable(unitSold));

}

}

**Output:**

hadoop jar S6\_A3.jar mapreduce.demo.task1.T2.Task2 /niki\_output/A6\_3\_1 niki\_output/A6\_3\_2

The output key will be company and state separated by tab and value would be the count for company state pair.

