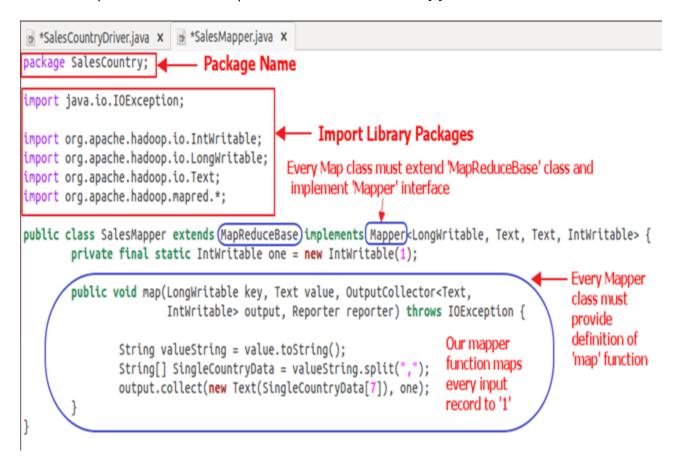
## **Explanation of SalesMapper Class**

In this section, we will understand the implementation of **SalesMapper** class.

1. We begin by specifying a name of package for our class. **SalesCountry** is a name of our package. Please note that output of compilation, **SalesMapper.class** will go into a directory named by this package name: **SalesCountry**.

Followed by this, we import library packages.

Below snapshot shows an implementation of SalesMapper class-



Sample Code Explanation:

## 1. SalesMapper Class Definition-

public class SalesMapper extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {

Every mapper class must be extended from **MapReduceBase** class and it must implement **Mapper** interface.

## 2. Defining 'map' function-

The main part of Mapper class is a 'map()' method which accepts four arguments.

At every call to 'map()' method, a key-value pair ('key' and 'value' in this code) is passed.

'map()' method begins by splitting input text which is received as an argument. It uses the tokenizer to split these lines into words.

```
String valueString = value.toString();
String[] SingleCountryData = valueString.split(",");
```

Here, ',' is used as a delimiter.

After this, a pair is formed using a record at 7th index of array 'SingleCountryData' and a value '1'.

output.collect(new Text(SingleCountryData[7]), one);

We are choosing record at 7th index because we need **Country** data and it is located at 7th index in array **'SingleCountryData'**.

Please note that our input data is in the below format (where **Country** is at 7<sup>th</sup> index, with 0 as a starting index)-

Transaction\_date,Product,Price,Payment\_Type,Name,City,State,**Country**,Account\_Created,Last\_Login,Latitude,Longitude

An output of mapper is again a **key-value** pair which is outputted using **'collect()'** method of **'OutputCollector'**.

## **Explanation of SalesCountryReducer Class**

In this section, we will understand the implementation of **SalesCountryReducer** class.

1. We begin by specifying a name of the package for our class. **SalesCountry** is a name of out package. Please note that output of compilation, **SalesCountryReducer.class** will go into a directory named by this package name: **SalesCountry**.

Followed by this, we import library packages.

Below snapshot shows an implementation of **SalesCountryReducer** class-

```
*SalesCountryDriver.java × *SalesMapper.java × *SalesCountryReducer.java ×
package SalesCountry;
                                  Package Name
import java.io.IOException:
import java.util.*;
                                                      Import Library Packages
                                              Every 'Reducer' class must extend 'MapReduceBase' class
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
                                              and implement 'Reducer' interface
import org.apache.hadoop.mapred.*;
public class SalesCountryReducer extends (MapReduceBase) implements (Reducer Text, IntWritable, Text, In
        public void reduce(Text t key, Iterator<IntWritable> values,
                           OutputCollector<Text,IntWritable> output,
                           Reporter reporter) throws IOException {
                Text key = t_key;
                                                                                           Every
                int frequencyForCountry = 0;
                                                                                           'Reducer'
                                                                                           class must
                while (values.hasNext()) {
                                                                                           provide
                         // replace type of value with the actual type of our value
                                                                                           definition of
                         IntWritable value = (IntWritable) values.next();
                                                                                           'reduce'
                         frequencyForCountry += value.get();
                                                                                           function
                }
                output.collect(key, new IntWritable(frequencyForCountry));
```