```
/* In Query_1 is a Map-only job */
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class Query_1 {

    public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, IntWritable, Text> {
      private IntWritable ID = new IntWritable();
      private Text record = new Text();

      public void map(LongWritable key, Text value, OutputCollector<IntWritable, Text> output, Reporter reporter) throws IOException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line, ",");//the space is the default delimiter
        //assign the value of attribute ID to ID
        ID.set(Integer.parseInt(tokenizer.nextToken()));
        //assign the value of attribute CountryCode to cc
        tokenizer.nextToken();
        tokenizer.nextToken();
        int cc=Integer.parseInt(tokenizer.nextToken());
        if(cc<=6 && cc>=2)
        {
                record.set(line);
            output.collect(ID, record);  //adds the key-value pair to the output
        }


      }
    }

    public static void main(String[] args) throws Exception {
     JobConf conf = new JobConf(Query_1.class);
              conf.setJobName("Query_1");



      conf.setOutputKeyClass(IntWritable.class);
      conf.setOutputValueClass(Text.class);

      conf.setMapperClass(Map.class);

      conf.setInputFormat(TextInputFormat.class);   // An InputFormat for plain text files. Files are broken into lines. Either linefeed
      conf.setOutputFormat(TextOutputFormat.class); // or carriage-return are used to signal end of line. !!!Keys are the position in the
                                                    // file, and values are the line of text..
      FileInputFormat.setInputPaths(conf, new Path(args[0]));  //args[0] => /usr/joe/wordcount/input
      FileOutputFormat.setOutputPath(conf, new Path(args[1])); //args[1] => /usr/joe/wordcount/

      JobClient.runJob(conf); //Normally the user creates the application, describes various facets of the job via JobConf and
                                                  //then uses the JobClient to submit the job and monitor its progress.
    }
}
```