**Problem statement**

Write an application using Python, Scala or Java that will use Spark to do the following:

• Read the data file ‘data.csv’.

• Create an optimised parquet file with the same data

• Load the parquet file into Spark

• Aggregate the values by country

• Write the results to a parquet file

**Solution:**

**Code in Java**

package ProjectFriday;  
  
  
import org.apache.spark.api.java.JavaRDD;  
import org.apache.spark.api.java.function.Function;  
import org.apache.spark.sql.\*;  
import org.apache.spark.sql.types.DataTypes;  
import org.apache.spark.sql.types.StructField;  
import org.apache.spark.sql.types.StructType;  
import java.util.Properties;  
import java.util.Objects;  
  
import static org.apache.spark.sql.functions.*collect\_set*;  
import static org.apache.spark.sql.functions.*concat\_ws*;  
import static org.apache.spark.sql.types.DataTypes.\*;  
  
public class Friday\_Black {  
  
 public static void main(String[] args) {  
  
 SparkSession session = SparkSession.*builder*().master(**"local"**).appName(**"jsonreader"**).getOrCreate();  
 SQLContext sqlContext = new SQLContext(session);  
  
  
 Dataset<Row> list1 = session.read().option(**"header"**, **"true"**).csv(**"E:**\\**data.csv"**);  
 list1.show();  
 list1.write().parquet(**"E:**\\**data.parquet"**);  
 list1.createOrReplaceTempView(**"list"**);  
 JavaRDD<Row> data = list1.toJavaRDD().filter(Objects::*nonNull*);  
 ;  
 JavaRDD<Row> data1 = data.map(new Function<Row, Row>() {  
 **@Override** public Row call(Row v1) throws Exception {  
 String[] words = v1.toString().split(**","**);  
 String[] value = words[1].split(**";"**);  
 int l1, l2;  
 String s, e;  
 l1 = value[4].length();  
 l2 = words[0].length();  
 s = value[4].substring(0, l1 - 1);  
 e = words[0].substring(1, l2 - 0);  
 return RowFactory.*create*(  
 e,  
 Integer.*parseInt*(value[0]),  
 Integer.*parseInt*(value[1]),  
 Integer.*parseInt*(value[2]),  
 Integer.*parseInt*(value[3]),  
 Integer.*parseInt*(s)  
  
  
 );  
 }  
 });  
  
 StructType schema = DataTypes.*createStructType*(new StructField[]{  
  
  
 *createStructField*(**"Country"**, *StringType*, true),  
 *createStructField*(**"v1"**, *IntegerType*, true),  
 *createStructField*(**"v2"**, *IntegerType*, true),  
 *createStructField*(**"v3"**, *IntegerType*, true),  
 *createStructField*(**"v4"**, *IntegerType*, true),  
 *createStructField*(**"v5"**, *IntegerType*, true),  
  
  
 });  
  
 Dataset<Row> df = sqlContext.createDataFrame(data1, schema);  
 df.show();  
 df.createOrReplaceTempView(**"DataView"**);  
  
 Dataset<Row> result = sqlContext.sql(**"select Country, CONCAT( SUM(v1),';',SUM(v2),';',SUM(v3),';',SUM(v4) ,';', SUM(v5)) AS Values from DataView group by Country order by Country"**);  
 result.show();  
 result.coalesce(1).write().parquet(**"E:**\\**Result"**);  
 }  
}

**output screen shots**



