**ADBMS Final Project**

**Problem Statement:**

The travel industry is a multi-billion-dollar industry, with quite rapidly fluctuating prices and uncertainty. This can be quite frustrating from a consumer point of view, especially during peak seasons. Presently, the airline is consolidated with just a few major key players that allow them to fluctuate the prices at will.

I plan to analyze the past trends in the fickle industry and try to provide future predictions of how the prices will vary over days and weeks. I also plan to analyze the travelling patterns of people over a quarter and check how the airlines perform in that period..

**Dataset:**

BTS Airlines Dataset from 2016 which contains Itinerary of passengers.

Size: 25 GB (Approx.)

Link:<https://www.transtats.bts.gov/Tables.asp?DB_ID=125&DB_Name=Airline%20Origin%20and%20Destination%20Survey%20%28DB1B%29&DB_Short_Name=Origin%20and%20Destination%20Survey>

**Analysis:**

1. **Cost aggregation for quarters between locations.**

Calculated minimum fare, maximum fare and avg fare for all the locations using custom writable and stored it in Hbase.

@Override

protected void reduce(LocationCustomWritable key, Iterable<AggCustomWritable> values,

Reducer<LocationCustomWritable, AggCustomWritable, LocationCustomWritable, AggCustomWritable>.Context context)

throws IOException, InterruptedException {

float sum = 0;

float count = 0;

float min = 100000f;

float max = 0f;

for(AggCustomWritable ac : values) {

sum += ac.getAvg() \* ac.getCount();

count += ac.getCount();

if(ac.getMin() < min) {

min = ac.getMin();

}

if(ac.getMax() > max ) {

max = ac.getMax();

}

}

acw.setCount(count);

acw.setAvg(sum/count);

acw.setMax(max);

acw.setMin(min);

acw.setSum(Math.round(sum));

LocationsHbase.addLocationsToHbase(table, "2016Q4", key.toString(), acw.getMin(), acw.getMax(), acw.getAvg());

context.write(key, acw);

}

public class LocationsHbase {

public static void addLocationsToHbase (Table table, String cf, String srcDes, Float minFare, Float maxFare, Float avgFare) {

try {

Put put = new Put (Bytes.toBytes(srcDes));

put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("minFare"), Bytes.toBytes(minFare.toString()));

put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("maxFare"), Bytes.toBytes(maxFare.toString()));

put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("avgFare"), Bytes.toBytes(avgFare.toString()));

table.put(put);

}

catch (Exception e) {

e.printStackTrace();

}

}

public static void addTopCarrierToHbase (Table table, String cf, String srcDes, String carrierCode) {

try {

Put put = new Put (Bytes.toBytes(srcDes));

put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("carrierCode"), Bytes.toBytes(carrierCode));

table.put(put);

}

catch (Exception e) {

e.printStackTrace();

}

}

}

1. **Cost aggregation for quarters for carriers.**

Calculated minimum fare, maximum fare and avg fare for all the carriers using custom writable.

@Override

protected void reduce(Text key, Iterable<AggCustomWritable> values,

Reducer<Text, AggCustomWritable, Text, AggCustomWritable>.Context context)

throws IOException, InterruptedException {

long sum = 0;

float count = 0;

float min = 100000f;

float max = 0f;

for(AggCustomWritable ac : values) {

sum += ac.getAvg() \* ac.getCount();

count += ac.getCount();

if(ac.getMin() < min) {

min = ac.getMin();

}

if(ac.getMax() > max ) {

max = ac.getMax();

}

}

acw.setCount(count);

acw.setAvg(sum/count);

acw.setMax(max);

acw.setMin(min);

acw.setSum(sum);

CarrierHbase.addCarriersToHbase(table, "2016Q4", key.toString(), acw.getMin(), acw.getMax(), acw.getAvg(), acw.getSum());

context.write(key, acw);

}

package com.adbms.project.airlines.tktcarrier.topk;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class TktCarrierTopCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> {

private Text outKey = new Text();

private LongWritable one = new LongWritable(1);

@Override

protected void map(LongWritable key, Text value,

Mapper<LongWritable, Text, Text, LongWritable>.Context context)

throws IOException, InterruptedException {

String fields[] = value.toString().split(",");

outKey.set(fields[25]);

context.write(outKey, one);

}

}

1. **Most frequent origin**

Used the TopK pattern to calculate the top origins in a quarter and converted it to json

public void reduce(Text key, Iterable<NullWritable> values,Context context)

throws IOException, InterruptedException {

if (count < 25) {

JsonObject origin = nestElements(key.toString());

topOrigins.add(origin);

//context.write(key, NullWritable.get());

count ++;

}

}

private JsonObject nestElements(String originTopK) {

try{

JsonObject origin = new JsonObject();

String fields[] = originTopK.toString().split(" ");

String org = fields[1].substring(1, fields[1].length()-1);

origin.addProperty("origin", org);

Long count = Long.parseLong(fields[0]);

origin.addProperty("count", count);

return origin;

}

catch(Exception e) {

e.printStackTrace();

}

return null;

}

@Override

protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context)

throws IOException, InterruptedException {

Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create();

context.write(new Text(gson.toJson(topOrigins)), NullWritable.get());

}

1. **Most frequent destination**

Used the TopK pattern to calculate the top origins in a quarter and wrote the output in json format

public void reduce(Text key, Iterable<NullWritable> values,Context context)

throws IOException, InterruptedException {

if (count < 25) {

JsonObject destination = nestElements(key.toString());

topDestinations.add(destination);

//context.write(key, NullWritable.get());

count ++;

}

}

private JsonObject nestElements(String originTopK) {

try{

JsonObject destination = new JsonObject();

String fields[] = originTopK.toString().split(" ");

String des = fields[1].substring(1, fields[1].length()-1);

destination.addProperty("destination", des);

Long count = Long.parseLong(fields[0]);

destination.addProperty("count", count);

return destination;

}

catch(Exception e) {

e.printStackTrace();

}

return null;

}

@Override

protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context)

throws IOException, InterruptedException {

Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create();

context.write(new Text(gson.toJson(topDestinations)), NullWritable.get());

}

1. Famous Airline carriers

Aggregated the count and total revenue of the carriers and wrote it in json

1. Quarter wise revenue of carriers

private LongWritable total = new LongWritable();

@Override

protected void reduce(Text key, Iterable<LongWritable> values,

Reducer<Text, LongWritable, Text, LongWritable>.Context context)

throws IOException, InterruptedException {

Long sum = 0l;

for (LongWritable l : values) {

sum += l.get();

}

total.set(sum);

context.write(key, total);

}

public void reduce(Text key, Iterable<NullWritable> values,Context context)

throws IOException, InterruptedException {

if (count < 25) {

tktCarriers.add(nestElements(key.toString()));

//context.write(key, NullWritable.get());

count ++;

}

}

private JsonObject nestElements(String tktTopK) {

try{

JsonObject tktCar = new JsonObject();

String fields[] = tktTopK.toString().split(" ");

String tkt = fields[1];

tktCar.addProperty("tktCarrier", tkt);

Long count = Long.parseLong(fields[0]);

tktCar.addProperty("sum", count);

return tktCar;

}

catch(Exception e) {

e.printStackTrace();

}

return null;

}

@Override

protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context)

throws IOException, InterruptedException {

Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create();

context.write(new Text(gson.toJson(tktCarriers)), NullWritable.get());

}

1. Top carrier per source and destination

Found the top carrier for each locations using a sorted map writable and stored them in hbase

@Override

protected void reduce(LocationCustomWritable key, Iterable<SortedMapWritable> values,

Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, SortedMapWritable>.Context context)

throws IOException, InterruptedException {

SortedMapWritable sortedMap = new SortedMapWritable();

for (SortedMapWritable val : values) {

for (Entry<WritableComparable, Writable> entry : val.entrySet()) {

LongWritable count = (LongWritable) sortedMap.get(entry.getKey());

if (count != null ) {

count.set(count.get() + ((LongWritable) entry.getValue()).get());

} else {

sortedMap.put(entry.getKey(), new LongWritable(((LongWritable) entry.getValue()).get()));

}

}

val.clear();

}

context.write(key, sortedMap);

}

protected void setup(

Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context)

throws IOException, InterruptedException {

conn = HBaseConf.getConnection();

table = HBaseConf.getLocationsTable(conn);

}

@Override

protected void reduce(LocationCustomWritable key, Iterable<SortedMapWritable> values,

Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context)

throws IOException, InterruptedException {

Text airlines = new Text();

Long max = 0L;

for (SortedMapWritable val : values) {

for (Entry<WritableComparable, Writable> entry : val.entrySet()) {

if (((LongWritable) entry.getValue()).get() > max) {

max = ((LongWritable) entry.getValue()).get();

airlines.set(entry.getKey().toString() + " : " + entry.getValue().toString());

}

}

}

LocationsHbase.addTopCarrierToHbase(table, "2016Q4", key.toString(), airlines.toString());

context.write(key, airlines);

}

@Override

protected void cleanup(

Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context)

throws IOException, InterruptedException {

HBaseConf.closeConnection(conn, table);

}

1. Reactive Visualization charts and APIs using spring boot

@RequestMapping(value = "/api/v1/getLocationAgg", method = RequestMethod.GET, produces = "application/json")

public ArrayList<Location>  getLocationAgg (HttpServletRequest request) {

System.out.println("inside the rest controller");

String src = request.getParameter("src");

String des = request.getParameter("des");

ArrayList<Location> locationsArray = new ArrayList<>();

ArrayList<String> array = new ArrayList<>();

array.add("2017Q2");

array.add("2017Q1");

array.add("2016Q4");

array.add("2016Q3");

array.add("2016Q2");

array.add("2016Q1");

for (String a: array) {

Location location = LocationsHbase.getLocationsFromHbase(src, des, a);

locationsArray.add(location);

}

return locationsArray;

}

public static Location getLocationsFromHbase (String src, String des, String cf) {

try{

Connection conn = HBaseConf.getConnection();

Table table = conn.getTable(TableName.valueOf( Bytes.toBytes("locationsAdbms")));

Get get = new Get(Bytes.toBytes("\"" + src + "\"" + "-" + "\"" + des + "\""));

Result result = table.get(get);

HBaseConf.closeConnection(conn, table);

NavigableMap<byte[], byte[]> map = result.getFamilyMap(Bytes.toBytes(cf));

Location location = new Location();

location.setSrc(src);

location.setDes(des);

location.setTime(cf);

for(byte[] key : map.keySet()){

if(Bytes.toString(key).equals("avgFare")) {

location.setAvgFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("minFare")) {

location.setMinFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("maxFare")) {

location.setMaxFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("carrierCode")) {

location.setCc(Bytes.toString(map.get(key)));

}

}

return location;

}

catch (Exception e) {

}

return null;

}

public static Carrier getCarriersFromHbase (String cc, String cf) {

try{

Connection conn = HBaseConf.getConnection();

Table table = conn.getTable(TableName.valueOf( Bytes.toBytes("carriersAdbms")));

Get get = new Get(Bytes.toBytes("\"" + cc + "\""));

Result result = table.get(get);

HBaseConf.closeConnection(conn, table);

NavigableMap<byte[], byte[]> map = result.getFamilyMap(Bytes.toBytes(cf));

Carrier carrier = new Carrier();

carrier.setCc(cc);

carrier.setTime(cf);

for(byte[] key : map.keySet()){

if(Bytes.toString(key).equals("avgFare")) {

carrier.setAvgFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("minFare")) {

carrier.setMinFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("maxFare")) {

carrier.setMaxFare(Float.parseFloat(Bytes.toString(map.get(key))));

}

if(Bytes.toString(key).equals("sumFare")) {

carrier.setSumFare(Long.parseLong(Bytes.toString(map.get(key))));

}

}

return carrier;

}

catch (Exception e) {

}

return null;

}

**Map Reduce Patterns used:**

* Join Pattern

Joined 3 dataset to get a complete dataset. The joined output was 5gb each for each quarter. Used multiple inputs and chaining of jobs

public static void main(String[] args) throws Exception {

if (args.length != 4) {

System.err.println("Usage: MaxSubmittedCharge <input path> <output path>");

System.exit(-1);

}

BasicConfigurator.configure();

Path file1Path = new Path(args[0]);

Path file2Path = new Path(args[1]);

Path file3Path = new Path(args[2]);

Path interDir = new Path(INTER\_PATH);

Path outputDir = new Path(args[3]);

// Create configuration

Configuration conf = new Configuration(true);

conf.set(TextOutputFormat.SEPERATOR, ",");

// Create job

Job job = Job.getInstance(conf, "Job1");

job.setJarByClass(JoinMapper1.class);

job.setNumReduceTasks(1);

// Setup MapReduce

MultipleInputs.addInputPath(job, file1Path, TextInputFormat.class, JoinMapper1.class);

MultipleInputs.addInputPath(job, file2Path, TextInputFormat.class, JoinMapper2.class);

job.setReducerClass(JoinReducer1.class);

// Specify key / value

job.setMapOutputKeyClass(LongWritable.class);

job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

// Output

FileOutputFormat.setOutputPath(job, interDir);

FileSystem hdfs = FileSystem.get(conf);

if (hdfs.exists(interDir))

hdfs.delete(interDir, true);

job.waitForCompletion(true);

//     System.exit(code);

// Execute job

Configuration conf2 = new Configuration(true);

Job job2 = Job.getInstance(conf2, "Job 2");

    job2.setJarByClass(JoinMapper2.class);

    job2.setNumReduceTasks(1);

    MultipleInputs.addInputPath(job2, interDir, TextInputFormat.class, JoinMapper3.class);

MultipleInputs.addInputPath(job2, file3Path, TextInputFormat.class, JoinMapper4.class);

    job2.setReducerClass(JoinReducer2.class);

    job2.setMapOutputKeyClass(LongWritable.class);

job2.setMapOutputValueClass(Text.class);

    job2.setOutputKeyClass(Text.class);

    job2.setOutputValueClass(Text.class);

    job2.setOutputFormatClass(TextOutputFormat.class);

    FileOutputFormat.setOutputPath(job2, outputDir);

if (hdfs.exists(outputDir))

hdfs.delete(outputDir, true);

    int code1 =  job2.waitForCompletion(true) ? 0 : 1;

    System.exit(code1);

}

* Filter pattern

Filtered the data to remove the outliers and the unused colums.

@Override

protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, NullWritable>.Context context)

throws IOException, InterruptedException {

String fields[] = value.toString().split(",");

String filteredString = fields[0] + "," + fields[1] + "," + fields[4] + "," + fields[8] + "," + fields[9] + ","

+ fields[10] + "," + fields[12] + "," + fields[13] + "," + fields[18] + "," + fields[19] + "," + fields[21] + ","

+ fields[22] + "," + fields[26] + "," + fields[27] + "," + fields[28] + "," + fields[29] + "," + fields[45] + ","

+ fields[46] + "," + fields[48] + "," + fields[49] + "," + fields[54] + "," + fields[55] + "," + fields[57] + ","

+ fields[58] + "," + fields[66] + "," + fields[67] + "," + fields[68] + ","  + fields[86] + "," + fields[71]

+ "," + fields[92] + "," + fields[94];

outkey.set(filteredString);

try {

if (Float.parseFloat(fields[71]) > 30 && fields[9] != fields[18] && Float.parseFloat(fields[71]) < 3500) {

context.write(outkey, NullWritable.get() );

}

}

catch (Exception e) {

context.write(outkey, NullWritable.get());

}

}

* Top K pattern

Performed TopK to find the top locations, top origins, top destinations and top carriers

* Data organization pattern

Used to convert the data to a structural form like json.

* Data summarization pattern

Summarize the data to fin average, minimum, maximum and sum

* Counting with counters

Used the counting with counters pattern to calculate the number of roundtrips and single trips.

@Override

protected void map(Object key, Text value, Mapper<Object, Text, Text, NullWritable>.Context context)

throws IOException, InterruptedException {

try {

String[] fields = value.toString().split(",");

        //context.write(t1, NullWritable.get());

        if (Float.parseFloat(fields[27]) == 1.0) {

        context.getCounter(ROUNDTRIP, "roundtrip").increment(1);

        }

        context.getCounter(ROUNDTRIP, "sum").increment(1);

}

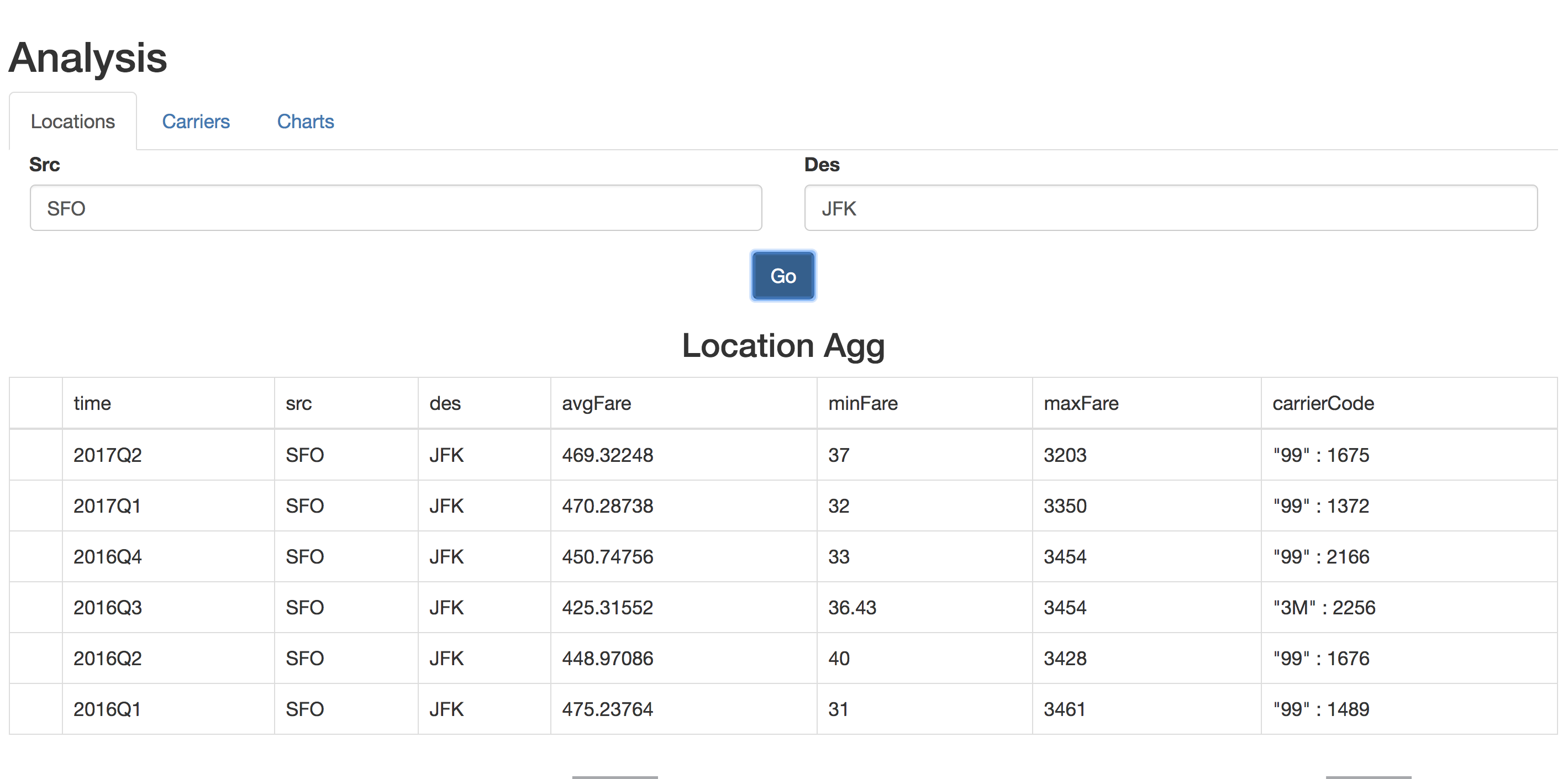
catch (Exception e) {

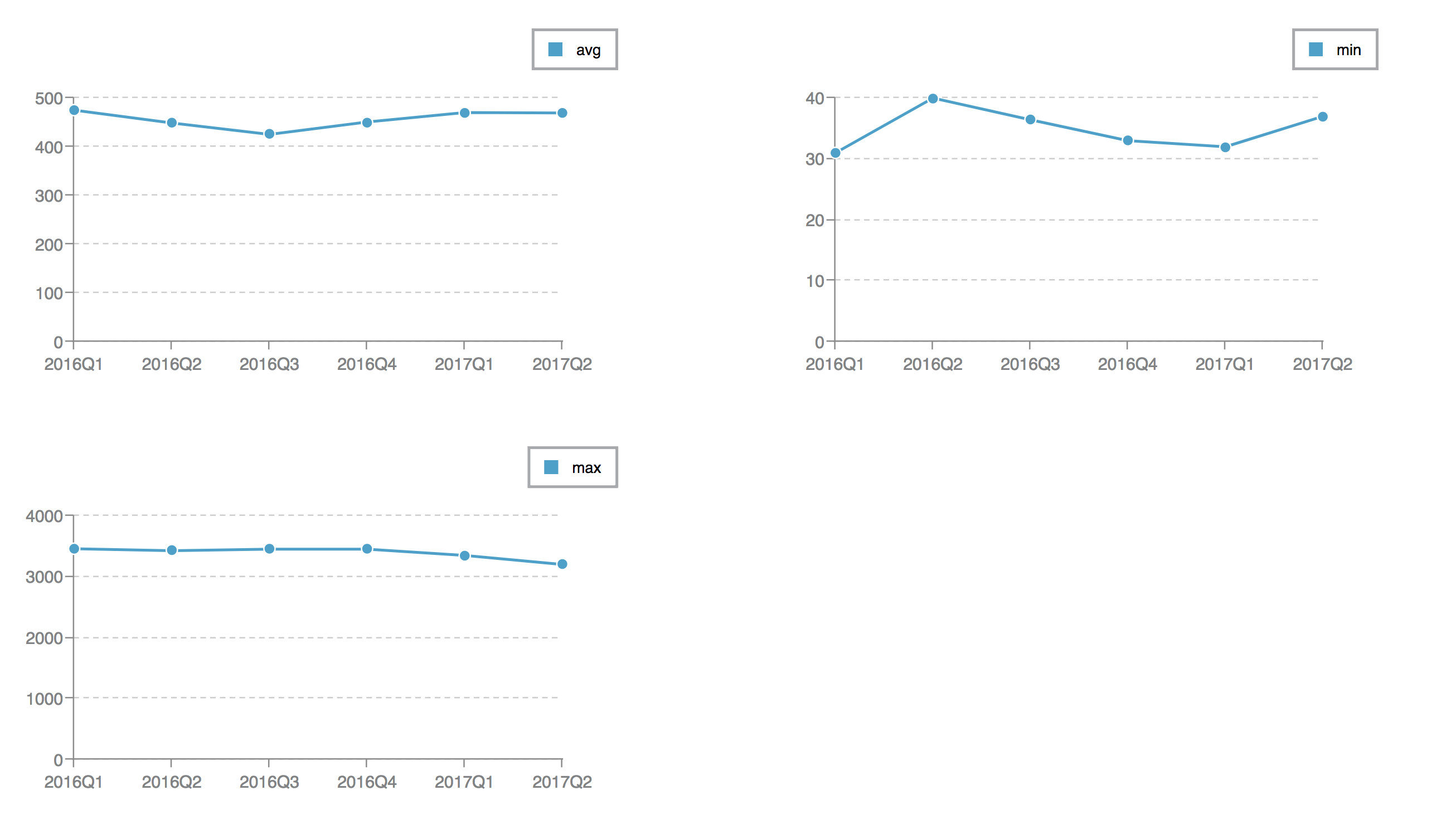
e.printStackTrace();

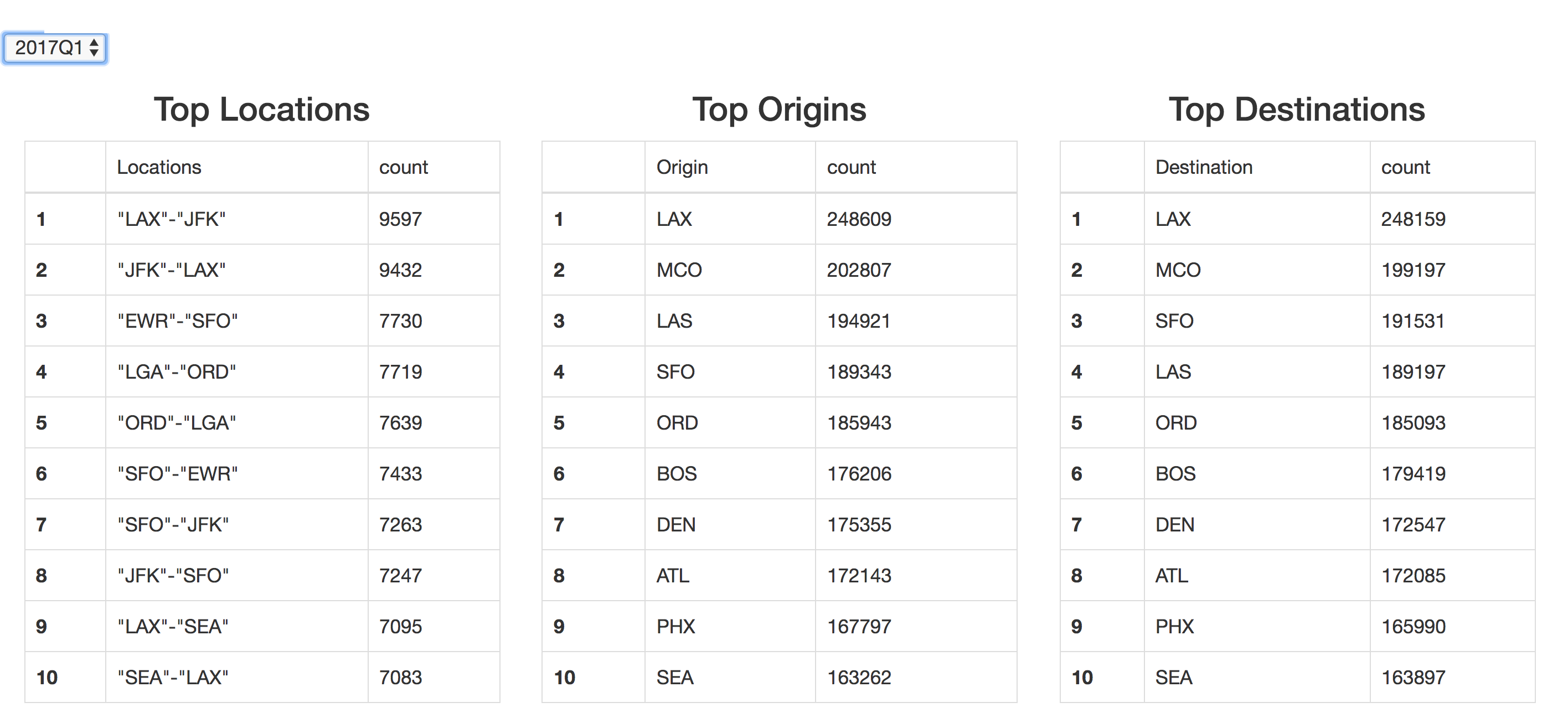
}

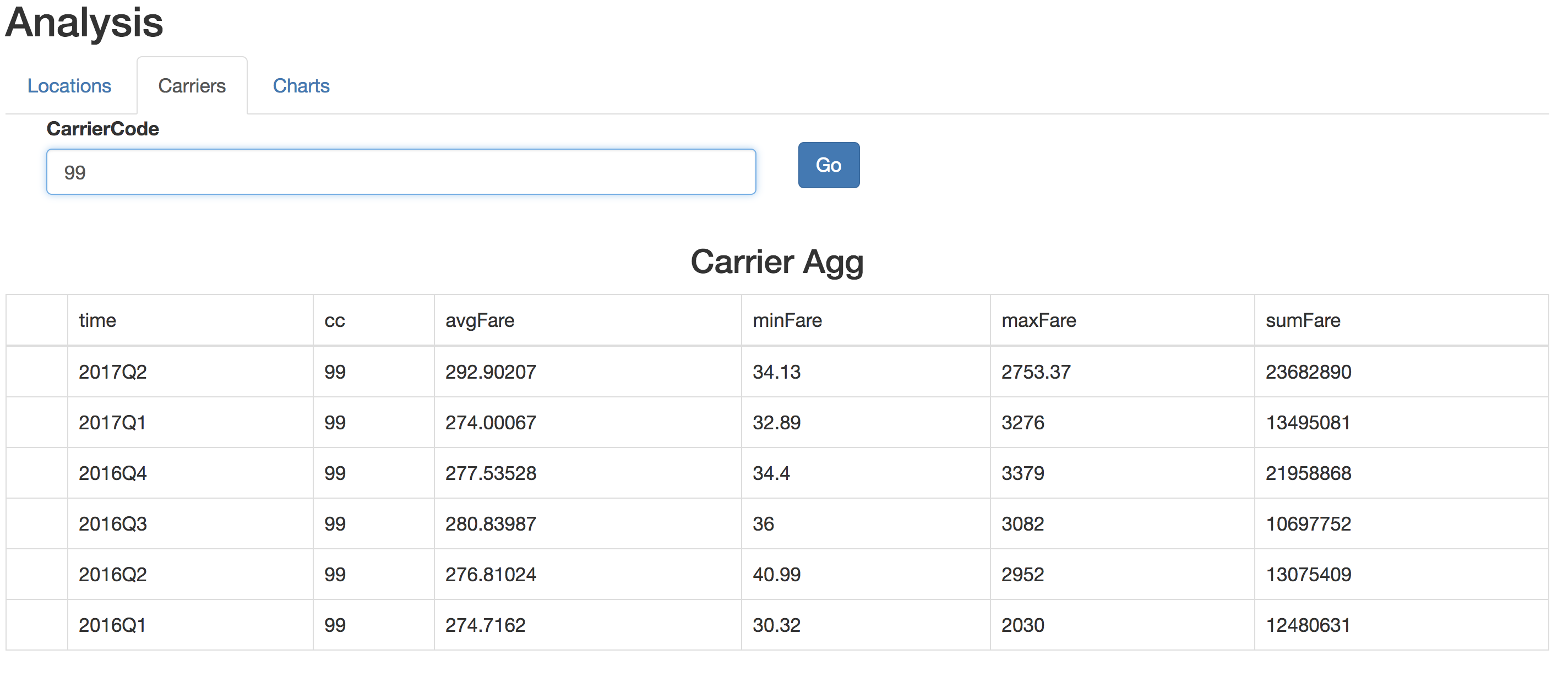
}

**Charts and UI:**

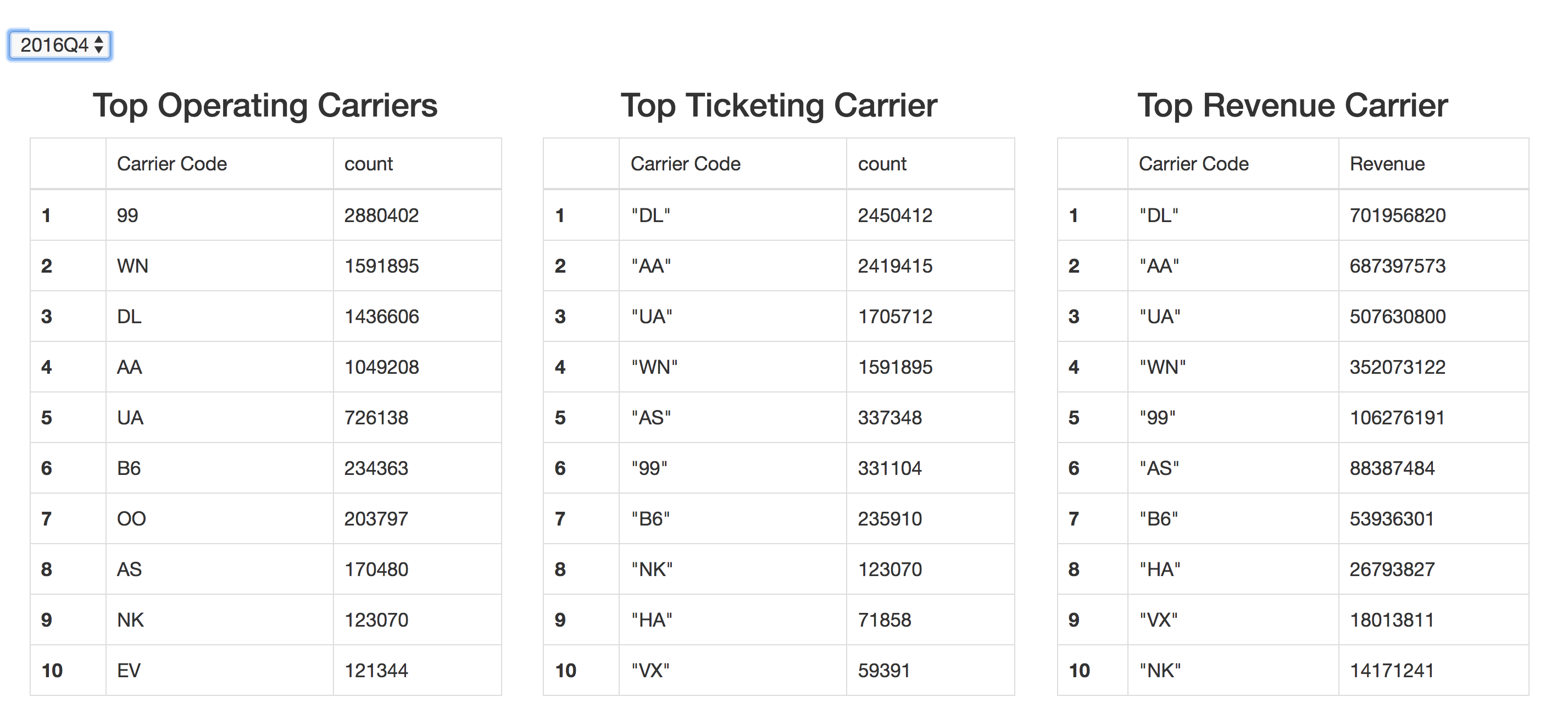
****

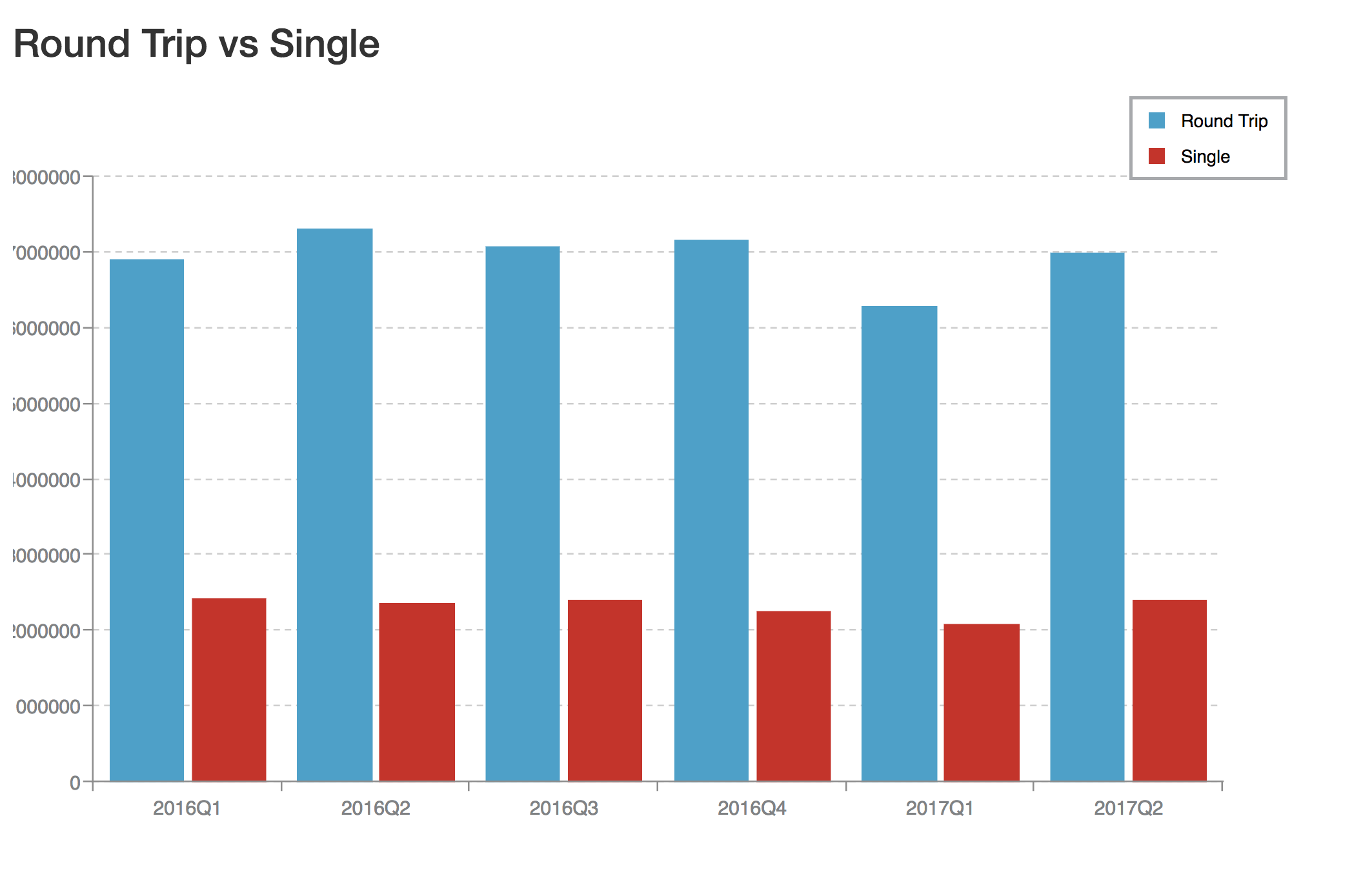
****

****

****

****

****

****

**47** [FinalProject/src/main/java/com/adbms/project/airlines/BloomFilterDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-485662f0004a9a5a314a9c676e7dac39)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,47 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import java.io.BufferedReader; |
|  |  | +import java.io.File; |
|  |  | +import java.io.FileReader; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FSDataOutputStream; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.util.bloom.BloomFilter; |
|  |  | +import org.apache.hadoop.util.bloom.Key; |
|  |  | +import org.apache.hadoop.util.hash.Hash; |
|  |  | +import org.apache.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +public class BloomFilterDriver { |
|  |  | + public static void main (String[] args) throws Exception { |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + Path outputDir = new Path("bloom/hotvalues"); |
|  |  | + BasicConfigurator.configure(); |
|  |  | + BloomFilter filter = new org.apache.hadoop.util.bloom.BloomFilter(2876, 20, Hash.MURMUR\_HASH); |
|  |  | + File file1 = new File("topkLocations/part-r-00000"); |
|  |  | + try(BufferedReader br = new BufferedReader(new FileReader(file1))) { |
|  |  | + for(String line; (line = br.readLine()) != null; ) { |
|  |  | + System.out.println("lines" + line); |
|  |  | + String fields[] = line.split(" "); |
|  |  | + String orgDes = fields[1]; |
|  |  | + filter.add(new Key(orgDes.getBytes())); |
|  |  | + } |
|  |  | + // line is not visible here. |
|  |  | + } |
|  |  | + |
|  |  | + System.setProperty("hadoop.home.dir", "/"); |
|  |  | + FileSystem fs = FileSystem.getLocal(conf); |
|  |  | + |
|  |  | + |
|  |  | + if (fs.exists(outputDir)) |
|  |  | + fs.delete(outputDir, true); |
|  |  | + |
|  |  | + FSDataOutputStream strm = fs.create(outputDir); |
|  |  | + filter.write(strm); |
|  |  | + strm.flush(); |
|  |  | + strm.close(); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**62** [FinalProject/src/main/java/com/adbms/project/airlines/CarrierAggDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-838d259f44ede8082d7a27892a6d9bb0)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,62 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.carriers.aggregation.CarrierAggMapper; |
|  |  | +import com.adbms.project.airlines.carriers.aggregation.CarrierAggReducer; |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | + |
|  |  | +public class CarrierAggDriver { |
|  |  | + |
|  |  | +public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path outputDir = new Path("2016Q4/carrierAggOutput"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Carrier Aggregates"); |
|  |  | + job.setJarByClass(CarrierAggMapper.class); |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(CarrierAggMapper.class); |
|  |  | + job.setCombinerClass(CarrierAggReducer.class); |
|  |  | + job.setReducerClass(CarrierAggReducer.class); |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(AggCustomWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(AggCustomWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | +} |

**63** [FinalProject/src/main/java/com/adbms/project/airlines/CarrierAggToJsonDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-5ef873796d601e1e67c2c7b6ee27a7de)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,63 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.carriers.tojson.CarrierAggToJson; |
|  |  | +import com.adbms.project.airlines.locations.tojson.LocationsToJsonMapper; |
|  |  | + |
|  |  | +public class CarrierAggToJsonDriver { |
|  |  | +public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + |
|  |  | + Path inputPath = new Path("carrierAggOutput/part-r-00000"); |
|  |  | + Path outputDir = new Path("CarrierAggJson"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "CarrierAggToJson"); |
|  |  | + job.setJarByClass(CarrierAggToJson.class); |
|  |  | + |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(CarrierAggToJson.class); |
|  |  | + job.setNumReduceTasks(0); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(NullWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**90** [FinalProject/src/main/java/com/adbms/project/airlines/DestinationTopKDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-4f67ba1c832551c1cc9a53b90c796086)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,90 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.destination.topk.DestinationTopCountMapper; |
|  |  | +import com.adbms.project.airlines.destination.topk.DestinationTopCountReducer; |
|  |  | +import com.adbms.project.airlines.destination.topk.DestinationTopKMapper; |
|  |  | +import com.adbms.project.airlines.destination.topk.DestinationTopKReducer; |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | + |
|  |  | +public class DestinationTopKDriver { |
|  |  | + |
|  |  | + private static final String INTER\_PATH = "2016Q4/topkDestination\_intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path(INTER\_PATH); |
|  |  | + Path outputDir = new Path("2016Q4/topkDestinations"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(DestinationTopCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(DestinationTopCountMapper.class); |
|  |  | + job.setReducerClass(DestinationTopCountReducer.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(DestinationTopKMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(DestinationTopKMapper.class); |
|  |  | + job2.setReducerClass(DestinationTopKReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**101** [FinalProject/src/main/java/com/adbms/project/airlines/Driver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ab024ff3a5960ca82c24356aada8942c)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,101 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.IntWritable; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.MultipleInputs; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.filter.FilterMapper; |
|  |  | +import com.adbms.project.airlines.join.JoinMapper1; |
|  |  | +import com.adbms.project.airlines.join.JoinMapper2; |
|  |  | +import com.adbms.project.airlines.join.JoinMapper3; |
|  |  | +import com.adbms.project.airlines.join.JoinMapper4; |
|  |  | +import com.adbms.project.airlines.join.JoinReducer1; |
|  |  | +import com.adbms.project.airlines.join.JoinReducer2; |
|  |  | + |
|  |  | +public class Driver { |
|  |  | +private static final String INTER\_PATH = "intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + if (args.length != 4) { |
|  |  | + System.err.println("Usage: MaxSubmittedCharge <input path> <output path>"); |
|  |  | + System.exit(-1); |
|  |  | + } |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path file1Path = new Path(args[0]); |
|  |  | + Path file2Path = new Path(args[1]); |
|  |  | + Path file3Path = new Path(args[2]); |
|  |  | + Path interDir = new Path(INTER\_PATH); |
|  |  | + Path outputDir = new Path(args[3]); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + conf.set(TextOutputFormat.SEPERATOR, ","); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(JoinMapper1.class); |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + MultipleInputs.addInputPath(job, file1Path, TextInputFormat.class, JoinMapper1.class); |
|  |  | + MultipleInputs.addInputPath(job, file2Path, TextInputFormat.class, JoinMapper2.class); |
|  |  | + job.setReducerClass(JoinReducer1.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(LongWritable.class); |
|  |  | + job.setMapOutputValueClass(Text.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(Text.class); |
|  |  | + |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + job.waitForCompletion(true); |
|  |  | +// System.exit(code); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + |
|  |  | + Configuration conf2 = new Configuration(true); |
|  |  | + Job job2 = Job.getInstance(conf2, "Job 2"); |
|  |  | + job2.setJarByClass(JoinMapper2.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + MultipleInputs.addInputPath(job2, interDir, TextInputFormat.class, JoinMapper3.class); |
|  |  | + MultipleInputs.addInputPath(job2, file3Path, TextInputFormat.class, JoinMapper4.class); |
|  |  | + job2.setReducerClass(JoinReducer2.class); |
|  |  | + |
|  |  | + job2.setMapOutputKeyClass(LongWritable.class); |
|  |  | + job2.setMapOutputValueClass(Text.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(Text.class); |
|  |  | + |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code1 = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code1); |
|  |  | + |
|  |  | + } |
|  |  | +} |

**91** [FinalProject/src/main/java/com/adbms/project/airlines/FilterDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ecddc9cc25d95eb14aae0823cbbe9a0c)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,91 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FSDataInputStream; |
|  |  | +import org.apache.hadoop.fs.FSDataOutputStream; |
|  |  | +import org.apache.hadoop.fs.FileStatus; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.filter.FilterMapper; |
|  |  | + |
|  |  | +public class FilterDriver { |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + if (args.length != 2) { |
|  |  | + System.err.println("Usage: MaxSubmittedCharge <input path> <output path>"); |
|  |  | + System.exit(-1); |
|  |  | + } |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path(args[0]); |
|  |  | + Path outputDir = new Path(args[1]); |
|  |  | + Path mergeDir = new Path("filtermerge"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Filter"); |
|  |  | + job.setJarByClass(FilterMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(FilterMapper.class); |
|  |  | + job.setNumReduceTasks(0); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(NullWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + FileSystem local = FileSystem.getLocal(conf); |
|  |  | + |
|  |  | + //merge all files |
|  |  | + try { |
|  |  | + FileStatus[] inputFiles = hdfs.listStatus(outputDir); |
|  |  | + FSDataOutputStream out = hdfs.create(mergeDir); |
|  |  | + |
|  |  | + for(int i = 0; i<inputFiles.length; i++) { |
|  |  | + System.out.println(inputFiles[i].getPath().getName()); |
|  |  | + FSDataInputStream in = local.open(inputFiles[i].getPath()); |
|  |  | + byte buffer[] = new byte[256]; |
|  |  | + int bytesRead = 0; |
|  |  | + while ((bytesRead= in.read(buffer))>0) { |
|  |  | + out.write(buffer, 0, bytesRead); |
|  |  | + } |
|  |  | + in.close(); |
|  |  | + } |
|  |  | + out.close(); |
|  |  | + } |
|  |  | + catch (IOException e){ |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | +} |

**63** [FinalProject/src/main/java/com/adbms/project/airlines/LocationAggDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-1dd6b5ae6a4ae94356a9a54e185816f2)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,63 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.aggregation.LocationAggMapper; |
|  |  | +import com.adbms.project.airlines.locations.aggregation.LocationAggReducer; |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationAggDriver { |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path outputDir = new Path("2016Q4/locations/aggOutput"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "LocationAgg"); |
|  |  | + job.setJarByClass(LocationAggMapper.class); |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(LocationAggMapper.class); |
|  |  | + job.setCombinerClass(LocationAggReducer.class); |
|  |  | + job.setReducerClass(LocationAggReducer.class); |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setMapOutputValueClass(AggCustomWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setOutputValueClass(AggCustomWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**66** [FinalProject/src/main/java/com/adbms/project/airlines/LocationTopCarDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-a5725530bff6e322fd76916068617708)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,66 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.SortedMapWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.topcarrier.LocationsTopCarCombiner; |
|  |  | +import com.adbms.project.airlines.locations.topcarrier.LocationsTopCarMapper; |
|  |  | +import com.adbms.project.airlines.locations.topcarrier.LocationsTopCarReducer; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationTopCarDriver { |
|  |  | + |
|  |  | +public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path outputDir = new Path("2016Q4/locationTopCarrierOutput"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Carrier Aggregates"); |
|  |  | + job.setJarByClass(LocationsTopCarMapper.class); |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(LocationsTopCarMapper.class); |
|  |  | + job.setCombinerClass(LocationsTopCarCombiner.class); |
|  |  | + job.setReducerClass(LocationsTopCarReducer.class); |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setMapOutputValueClass(SortedMapWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setOutputValueClass(Text.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**67** [FinalProject/src/main/java/com/adbms/project/airlines/LocationsToJsonDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-75c767d122fd53ac176368757f175a7a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,67 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FSDataInputStream; |
|  |  | +import org.apache.hadoop.fs.FSDataOutputStream; |
|  |  | +import org.apache.hadoop.fs.FileStatus; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.tojson.LocationsToJsonMapper; |
|  |  | + |
|  |  | +public class LocationsToJsonDriver { |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + |
|  |  | + Path inputPath = new Path("aggOutput/part-r-00000"); |
|  |  | + Path outputDir = new Path("LocationsAggJson"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "LocationsToJson"); |
|  |  | + job.setJarByClass(LocationsToJsonMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(LocationsToJsonMapper.class); |
|  |  | + job.setNumReduceTasks(0); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(NullWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + |
|  |  | + // Delete output if exists |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**86** [FinalProject/src/main/java/com/adbms/project/airlines/OpCarrierTopKDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-93427ca1761bbca6748fd49edc04b755)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,86 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.opcarrier.topk.OpCarrierTopCountMapper; |
|  |  | +import com.adbms.project.airlines.opcarrier.topk.OpCarrierTopCountReducer; |
|  |  | +import com.adbms.project.airlines.opcarrier.topk.OpCarrierTopKMapper; |
|  |  | +import com.adbms.project.airlines.opcarrier.topk.OpCarrierTopKReducer; |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | + |
|  |  | +public class OpCarrierTopKDriver { |
|  |  | +public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path("2016Q4/topKOpCarrier\_intermiadte\_output"); |
|  |  | + Path outputDir = new Path("2016Q4/topkOpCarrier"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(OpCarrierTopCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(OpCarrierTopCountMapper.class); |
|  |  | + job.setReducerClass(OpCarrierTopCountReducer.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(OpCarrierTopKMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(OpCarrierTopKMapper.class); |
|  |  | + job2.setReducerClass(OpCarrierTopKReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | +} |

**90** [FinalProject/src/main/java/com/adbms/project/airlines/OriginTopKDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ea361015d69eede205ad75aa59f1bcd6)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,90 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.origin.topk.OriginTopCountMapper; |
|  |  | +import com.adbms.project.airlines.origin.topk.OriginTopCountReducer; |
|  |  | +import com.adbms.project.airlines.origin.topk.OriginTopKMapper; |
|  |  | +import com.adbms.project.airlines.origin.topk.OriginTopKReducer; |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | + |
|  |  | +public class OriginTopKDriver { |
|  |  | + |
|  |  | + private static final String INTER\_PATH = "2016Q4/topkOrigin\_intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path(INTER\_PATH); |
|  |  | + Path outputDir = new Path("2016Q4/topkOrigins"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(OriginTopCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(OriginTopCountMapper.class); |
|  |  | + job.setReducerClass(OriginTopCountReducer.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(OriginTopKMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(OriginTopKMapper.class); |
|  |  | + job2.setReducerClass(OriginTopKReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**58** [FinalProject/src/main/java/com/adbms/project/airlines/RoundTripDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-6433b5f1e77bfdea5e35357fa298e9bb)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,58 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Counter; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.MRJobConfig; |
|  |  | +import org.apache.hadoop.mapreduce.counters.Limits; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.tojson.LocationsToJsonMapper; |
|  |  | +import com.adbms.project.airlines.roundtrip.RoundTripMapper; |
|  |  | + |
|  |  | +public class RoundTripDriver { |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + Configuration conf = new Configuration(); |
|  |  | + conf.setInt(MRJobConfig.COUNTERS\_MAX\_KEY, 6000); |
|  |  | + BasicConfigurator.configure(); |
|  |  | + Limits.init(conf); |
|  |  | + |
|  |  | + |
|  |  | + Path input = new Path("2016Q2/filtermerge"); |
|  |  | + Path outputDir = new Path("2016Q2/RoundtripCounter"); |
|  |  | + |
|  |  | + Job job = Job.getInstance(conf, "Count IP Address"); |
|  |  | + job.setJarByClass(RoundTripMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setMapperClass(RoundTripMapper.class); |
|  |  | + job.setNumReduceTasks(0); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job, input); |
|  |  | + FileOutputFormat.setOutputPath(job, outputDir); |
|  |  | + int code = job.waitForCompletion(true) ? 0 : 1; |
|  |  | + |
|  |  | + if (code == 0) { |
|  |  | + for (Counter counter : job.getCounters().getGroup( |
|  |  | + RoundTripMapper.ROUNDTRIP)) { |
|  |  | + System.out.println(counter.getDisplayName() + "\t" |
|  |  | + + counter.getValue()); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + FileSystem.get(conf).delete(outputDir, true); |
|  |  | + |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | +} |

**89** [FinalProject/src/main/java/com/adbms/project/airlines/TktCarrierTopKDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-34ad828732b41f68c77fb7d3241ddf8e)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,89 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | +import com.adbms.project.airlines.tktcarrier.topk.TktCarrierTopCountMapper; |
|  |  | +import com.adbms.project.airlines.tktcarrier.topk.TktCarrierTopCountReducer; |
|  |  | +import com.adbms.project.airlines.tktcarrier.topk.TktCarrierTopKMapper; |
|  |  | +import com.adbms.project.airlines.tktcarrier.topk.TktCarrierTopKReducer; |
|  |  | + |
|  |  | +public class TktCarrierTopKDriver { |
|  |  | +private static final String INTER\_PATH = "2016Q4/topktktcarrier\_intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path(INTER\_PATH); |
|  |  | + Path outputDir = new Path("2016Q4/topkTktCarrier"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(TktCarrierTopCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(TktCarrierTopCountMapper.class); |
|  |  | + job.setReducerClass(TktCarrierTopCountReducer.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(TktCarrierTopKMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(TktCarrierTopKMapper.class); |
|  |  | + job2.setReducerClass(TktCarrierTopKReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**91** [FinalProject/src/main/java/com/adbms/project/airlines/TktCarrierTopRevDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-acd0bf907e588d830f15553176f34460)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,91 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | +import org.apache.log4j.Level; |
|  |  | +import org.apache.log4j.LogManager; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | +import com.adbms.project.airlines.tktcarrier.toprev.TktCarrierTopRevCountMapper; |
|  |  | +import com.adbms.project.airlines.tktcarrier.toprev.TktCarrierTopRevCountReducer; |
|  |  | +import com.adbms.project.airlines.tktcarrier.toprev.TktCarrierTopRevMapper; |
|  |  | +import com.adbms.project.airlines.tktcarrier.toprev.TktCarrierTopRevReducer; |
|  |  | + |
|  |  | +public class TktCarrierTopRevDriver { |
|  |  | + private static final String INTER\_PATH = "2016Q4/toprevtktcarrier\_intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path(INTER\_PATH); |
|  |  | + Path outputDir = new Path("2016Q4/TopRevTktCarrier"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(TktCarrierTopRevCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(TktCarrierTopRevCountMapper.class); |
|  |  | + job.setReducerClass(TktCarrierTopRevCountReducer.class); |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + job.setMapOutputKeyClass(Text.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(Text.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(TktCarrierTopRevMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(TktCarrierTopRevMapper.class); |
|  |  | + job2.setReducerClass(TktCarrierTopRevReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**93** [FinalProject/src/main/java/com/adbms/project/airlines/TopKDriver.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-c0f6b1800f836971c3de420e1cb3285d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,93 @@ |
|  |  | +package com.adbms.project.airlines; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.fs.FileSystem; |
|  |  | +import org.apache.hadoop.fs.Path; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Job; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  |  | +import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat; |
|  |  | +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.log4j.BasicConfigurator; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.topk.LocationTopCountMapper; |
|  |  | +import com.adbms.project.airlines.locations.topk.LocationTopCountReducer; |
|  |  | +import com.adbms.project.airlines.locations.topk.LocationTopKMapper; |
|  |  | +import com.adbms.project.airlines.locations.topk.LocationTopKReducer; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | +import com.adbms.project.airlines.sort.DescendingSortComparator; |
|  |  | + |
|  |  | +public class TopKDriver { |
|  |  | + |
|  |  | +private static final String OUTPUT\_PATH = "2016Q4/topkLocations\_intermediate\_output"; |
|  |  | + |
|  |  | + public static void main(String[] args) throws Exception { |
|  |  | + |
|  |  | + BasicConfigurator.configure(); |
|  |  | + |
|  |  | + Path inputPath = new Path("2016Q4/filtermerge"); |
|  |  | + Path interDir = new Path(OUTPUT\_PATH); |
|  |  | + Path outputDir = new Path("2016Q4/topkLocations"); |
|  |  | + |
|  |  | + // Create configuration |
|  |  | + Configuration conf = new Configuration(true); |
|  |  | + |
|  |  | + // Create job |
|  |  | + Job job = Job.getInstance(conf, "Job1"); |
|  |  | + job.setJarByClass(LocationTopCountMapper.class); |
|  |  | + |
|  |  | + |
|  |  | + job.setNumReduceTasks(1); |
|  |  | + // Setup MapReduce |
|  |  | + job.setMapperClass(LocationTopCountMapper.class); |
|  |  | + //job.setCombinerClass(MovieCombiner.class); |
|  |  | + job.setReducerClass(LocationTopCountReducer.class); |
|  |  | + |
|  |  | + |
|  |  | + // Specify key / value |
|  |  | + |
|  |  | + job.setMapOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setMapOutputValueClass(LongWritable.class); |
|  |  | + |
|  |  | + job.setOutputKeyClass(LocationCustomWritable.class); |
|  |  | + job.setOutputValueClass(LongWritable.class); |
|  |  | + // Input |
|  |  | + FileInputFormat.addInputPath(job, inputPath); |
|  |  | + job.setInputFormatClass(TextInputFormat.class); |
|  |  | + |
|  |  | + // Output |
|  |  | + FileOutputFormat.setOutputPath(job, interDir); |
|  |  | + FileSystem hdfs = FileSystem.get(conf); |
|  |  | + if (hdfs.exists(interDir)) |
|  |  | + hdfs.delete(interDir, true); |
|  |  | + |
|  |  | + // Execute job |
|  |  | + job.waitForCompletion(true); |
|  |  | + |
|  |  | + Job job2 = Job.getInstance(conf, "Job 2"); |
|  |  | + job2.setJarByClass(LocationTopKMapper.class); |
|  |  | + |
|  |  | + job2.setNumReduceTasks(1); |
|  |  | + job2.setSortComparatorClass(DescendingSortComparator.class); |
|  |  | + job2.setMapperClass(LocationTopKMapper.class); |
|  |  | + job2.setReducerClass(LocationTopKReducer.class); |
|  |  | + |
|  |  | + job2.setOutputKeyClass(Text.class); |
|  |  | + job2.setOutputValueClass(NullWritable.class); |
|  |  | + |
|  |  | + job2.setInputFormatClass(KeyValueTextInputFormat.class); |
|  |  | + job2.setOutputFormatClass(TextOutputFormat.class); |
|  |  | + |
|  |  | + FileInputFormat.addInputPath(job2, interDir); |
|  |  | + FileOutputFormat.setOutputPath(job2, outputDir); |
|  |  | + if (hdfs.exists(outputDir)) |
|  |  | + hdfs.delete(outputDir, true); |
|  |  | + |
|  |  | + int code = job2.waitForCompletion(true) ? 0 : 1; |
|  |  | + System.exit(code); |
|  |  | + } |
|  |  | +} |

**42** [...oject/src/main/java/com/adbms/project/airlines/carriers/aggregation/CarrierAggMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f9662720afb95ac0139c250c5c7ff333)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,42 @@ |
|  |  | +package com.adbms.project.airlines.carriers.aggregation; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | + |
|  |  | + |
|  |  | +public class CarrierAggMapper extends Mapper<LongWritable, Text, Text, AggCustomWritable> { |
|  |  | + |
|  |  | + private AggCustomWritable acw = new AggCustomWritable(); |
|  |  | + private Text outKey = new Text(); |
|  |  | + |
|  |  | + static int count = 0; |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + Long sum = (long) Math.round(Float.parseFloat(fields[28])); |
|  |  | + try { |
|  |  | + |
|  |  | + outKey.set(fields[26]); |
|  |  | + acw.setCount(1); |
|  |  | + acw.setAvg(Float.parseFloat(fields[28])); |
|  |  | + acw.setMax(Float.parseFloat(fields[28])); |
|  |  | + acw.setMin(Float.parseFloat(fields[28])); |
|  |  | + acw.setSum(sum); |
|  |  | + context.write(outKey, acw); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**61** [...ject/src/main/java/com/adbms/project/airlines/carriers/aggregation/CarrierAggReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-794ab6852e0301c91a6f7e0af1aae805)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,61 @@ |
|  |  | +package com.adbms.project.airlines.carriers.aggregation; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | +import com.adbms.project.hbase.CarrierHbase; |
|  |  | +import com.adbms.project.hbase.HBaseConf; |
|  |  | + |
|  |  | +public class CarrierAggReducer extends Reducer<Text, AggCustomWritable, Text, AggCustomWritable>{ |
|  |  | + |
|  |  | + private AggCustomWritable acw = new AggCustomWritable(); |
|  |  | + private Table table = null; |
|  |  | + private Connection conn = null; |
|  |  | + @Override |
|  |  | + protected void setup( |
|  |  | + Reducer<Text, AggCustomWritable, Text, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + conn = HBaseConf.getConnection(); |
|  |  | + table = CarrierHbase.getCarriersTable(conn); |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<AggCustomWritable> values, |
|  |  | + Reducer<Text, AggCustomWritable, Text, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + long sum = 0; |
|  |  | + float count = 0; |
|  |  | + float min = 100000f; |
|  |  | + float max = 0f; |
|  |  | + |
|  |  | + for(AggCustomWritable ac : values) { |
|  |  | + sum += ac.getAvg() \* ac.getCount(); |
|  |  | + count += ac.getCount(); |
|  |  | + if(ac.getMin() < min) { |
|  |  | + min = ac.getMin(); |
|  |  | + } |
|  |  | + |
|  |  | + if(ac.getMax() > max ) { |
|  |  | + max = ac.getMax(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + acw.setCount(count); |
|  |  | + acw.setAvg(sum/count); |
|  |  | + acw.setMax(max); |
|  |  | + acw.setMin(min); |
|  |  | + acw.setSum(sum); |
|  |  | + CarrierHbase.addCarriersToHbase(table, "2016Q4", key.toString(), acw.getMin(), acw.getMax(), acw.getAvg(), acw.getSum()); |
|  |  | + context.write(key, acw); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**64** [FinalProject/src/main/java/com/adbms/project/airlines/carriers/tojson/CarrierAggToJson.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-247565b45c6ffe7eba88553826e77a79)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,64 @@ |
|  |  | +package com.adbms.project.airlines.carriers.tojson; |
|  |  | + |
|  |  | +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.Mapper; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonElement; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class CarrierAggToJson extends Mapper<LongWritable, Text, Text, NullWritable>{ |
|  |  | + static int count = 0; |
|  |  | + JsonArray carriers = new JsonArray(); |
|  |  | + JsonObject jsonOut; |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + JsonElement carrier = nestElements(value.toString()); |
|  |  | + carriers.add(carrier); |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String locationsAgg) { |
|  |  | + try{ |
|  |  | + JsonObject carrier = new JsonObject(); |
|  |  | + |
|  |  | + String fields[] = locationsAgg.toString().split("\t"); |
|  |  | + String carrierCode = fields[0]; |
|  |  | + String cc = carrierCode.substring(1, carrierCode.length()-1);; |
|  |  | + |
|  |  | + carrier.addProperty("carrierCode", cc); |
|  |  | + String agg[] = fields[1].split("\\|"); |
|  |  | + String minkv[] = agg[0].split(":"); |
|  |  | + String maxkv[] = agg[1].split(":"); |
|  |  | + String avgkv[] = agg[2].split(":"); |
|  |  | + String sumkv[] = agg[3].split(":"); |
|  |  | + carrier.addProperty("minFare", Float.parseFloat(minkv[1])); |
|  |  | + carrier.addProperty("maxFare", Float.parseFloat(maxkv[1])); |
|  |  | + carrier.addProperty("avgFare", Float.parseFloat(avgkv[1])); |
|  |  | + carrier.addProperty("sumFare", Float.parseFloat(sumkv[1])); |
|  |  | + |
|  |  | + return carrier; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(carriers)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**24** [.../src/main/java/com/adbms/project/airlines/destination/topk/DestinationTopCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-fc016f95f3422b996f636864bb347cdb)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,24 @@ |
|  |  | +package com.adbms.project.airlines.destination.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class DestinationTopCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> { |
|  |  | + |
|  |  | + private Text outKey = new Text(); |
|  |  | + private LongWritable one = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + outKey.set(fields[20]); |
|  |  | + context.write(outKey, one); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**25** [...src/main/java/com/adbms/project/airlines/destination/topk/DestinationTopCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-c5e7b82c3c7e7e27ad50dfcea76f736a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.destination.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class DestinationTopCountReducer extends Reducer<Text, LongWritable, Text, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<LongWritable> values, |
|  |  | + Reducer<Text, LongWritable, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**20** [...ject/src/main/java/com/adbms/project/airlines/destination/topk/DestinationTopKMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2f3fc58f68c8f0520f7ef0c20ab79de0)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,20 @@ |
|  |  | +package com.adbms.project.airlines.destination.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class DestinationTopKMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | +} |

**62** [...ect/src/main/java/com/adbms/project/airlines/destination/topk/DestinationTopKReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-bae7f44ab93bccb1e6a178e424af626b)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,62 @@ |
|  |  | +package com.adbms.project.airlines.destination.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class DestinationTopKReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray topDestinations = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 25) { |
|  |  | + JsonObject destination = nestElements(key.toString()); |
|  |  | + topDestinations.add(destination); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String originTopK) { |
|  |  | + try{ |
|  |  | + |
|  |  | + JsonObject destination = new JsonObject(); |
|  |  | + String fields[] = originTopK.toString().split(" "); |
|  |  | + String des = fields[1].substring(1, fields[1].length()-1); |
|  |  | + destination.addProperty("destination", des); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + destination.addProperty("count", count); |
|  |  | + return destination; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(topDestinations)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**42** [FinalProject/src/main/java/com/adbms/project/airlines/filter/FilterMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f7ef87be930888daed2565894e45187a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,42 @@ |
|  |  | +package com.adbms.project.airlines.filter; |
|  |  | + |
|  |  | +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.Mapper; |
|  |  | + |
|  |  | +public class FilterMapper extends Mapper<LongWritable, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + private Text outkey = new Text(); |
|  |  | + |
|  |  | + static int count = 0; |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + String filteredString = fields[0] + "," + fields[1] + "," + fields[4] + "," + fields[8] + "," + fields[9] + "," |
|  |  | + + fields[10] + "," + fields[12] + "," + fields[13] + "," + fields[18] + "," + fields[19] + "," + fields[21] + "," |
|  |  | + + fields[22] + "," + fields[26] + "," + fields[27] + "," + fields[28] + "," + fields[29] + "," + fields[45] + "," |
|  |  | + + fields[46] + "," + fields[48] + "," + fields[49] + "," + fields[54] + "," + fields[55] + "," + fields[57] + "," |
|  |  | + + fields[58] + "," + fields[66] + "," + fields[67] + "," + fields[68] + "," + fields[86] + "," + fields[71] |
|  |  | + + "," + fields[92] + "," + fields[94]; |
|  |  | + outkey.set(filteredString); |
|  |  | + |
|  |  | + try { |
|  |  | + if (Float.parseFloat(fields[71]) > 30 && fields[9] != fields[18] && Float.parseFloat(fields[71]) < 3500) { |
|  |  | + context.write(outkey, NullWritable.get() ); |
|  |  | + } |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + context.write(outkey, NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**36** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinMapper1.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0bd57fa594b821d694b4653947d3f7c5)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,36 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class JoinMapper1 extends Mapper<LongWritable, Text, LongWritable, Text> { |
|  |  | + |
|  |  | + private LongWritable outKey = new LongWritable(); |
|  |  | + private Text outValue = new Text(); |
|  |  | + private LongWritable l = new LongWritable(100000); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, LongWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + try { |
|  |  | + |
|  |  | + Long mktId = Long.parseLong(fields[1]); |
|  |  | + outKey.set(mktId); |
|  |  | + outValue.set("A" + value.toString()); |
|  |  | + context.write(outKey, outValue); |
|  |  | + } |
|  |  | + catch (Exception e){ |
|  |  | + outValue.set("A" + value.toString()); |
|  |  | + context.write(l, outValue); |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**32** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinMapper2.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-138120ea03d1a3c2c6836acea66f35d8)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,32 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class JoinMapper2 extends Mapper<LongWritable, Text, LongWritable, Text> { |
|  |  | + |
|  |  | + private LongWritable outKey = new LongWritable(); |
|  |  | + private Text outValue = new Text(); |
|  |  | + private LongWritable l = new LongWritable(100000); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, LongWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + try { |
|  |  | + |
|  |  | + Long mktId = Long.parseLong(fields[1]); |
|  |  | + outKey.set(mktId); |
|  |  | + outValue.set("B" + value.toString()); |
|  |  | + context.write(outKey, outValue); |
|  |  | + } |
|  |  | + catch (Exception e){ |
|  |  | + outValue.set("B" + value.toString()); |
|  |  | + context.write(l, outValue); |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | +} |

**34** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinMapper3.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-c8786bc851aa5d691aa96159b41159f7)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,34 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | + |
|  |  | +public class JoinMapper3 extends Mapper<LongWritable, Text, LongWritable, Text> { |
|  |  | + |
|  |  | + private LongWritable outKey = new LongWritable(); |
|  |  | + private Text outValue = new Text(); |
|  |  | + private LongWritable l = new LongWritable(100000); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, LongWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + try { |
|  |  | + Long itinId = Long.parseLong(fields[0]); |
|  |  | + outKey.set(itinId); |
|  |  | + outValue.set("A" + value.toString()); |
|  |  | + context.write(outKey, outValue); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + outValue.set("A" + value.toString()); |
|  |  | + context.write(l, outValue); |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | +} |

**33** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinMapper4.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-947b09c7d7473a6a39b483257429010f)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,33 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class JoinMapper4 extends Mapper<LongWritable, Text, LongWritable, Text> { |
|  |  | + |
|  |  | + private LongWritable outKey = new LongWritable(); |
|  |  | + private Text outValue = new Text(); |
|  |  | + private LongWritable l = new LongWritable(100000); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, LongWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + try { |
|  |  | + Long itinId = Long.parseLong(fields[0]); |
|  |  | + outKey.set(itinId); |
|  |  | + outValue.set("B" + value.toString()); |
|  |  | + context.write(outKey, outValue); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + outValue.set("B" + value.toString()); |
|  |  | + context.write(l, outValue); |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | +} |

**41** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinReducer1.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-b64c2d7232bbce9fe8515c2531a24bb2)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,41 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | +import java.util.ArrayList; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class JoinReducer1 extends Reducer<LongWritable, Text, Text, Text> { |
|  |  | + |
|  |  | + private ArrayList<Text> listA = new ArrayList<Text>(); |
|  |  | + private ArrayList<Text> listB = new ArrayList<Text>(); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(LongWritable key, Iterable<Text> values, Reducer<LongWritable, Text, Text, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + listA.clear(); |
|  |  | + listB.clear(); |
|  |  | + for (Text value: values) { |
|  |  | + if(value.charAt(0)== 'A'){ |
|  |  | + listA.add(new Text(value.toString().substring(1))); |
|  |  | + } else if(value.charAt(0)== 'B'){ |
|  |  | + listB.add(new Text(value.toString().substring(1))); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + if(!listA.isEmpty() && !listB.isEmpty()) { |
|  |  | + for (Text a: listA){ |
|  |  | + for (Text b: listB){ |
|  |  | + context.write(a, b); |
|  |  | + } |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**39** [FinalProject/src/main/java/com/adbms/project/airlines/join/JoinReducer2.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d6dba4f58886333d1a35b3c21acc1e22)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,39 @@ |
|  |  | +package com.adbms.project.airlines.join; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | +import java.util.ArrayList; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class JoinReducer2 extends Reducer<LongWritable, Text, Text, Text> { |
|  |  | + |
|  |  | + private ArrayList<Text> listA = new ArrayList<Text>(); |
|  |  | + private ArrayList<Text> listB = new ArrayList<Text>(); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(LongWritable key, Iterable<Text> values, Reducer<LongWritable, Text, Text, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + listA.clear(); |
|  |  | + listB.clear(); |
|  |  | + for (Text value: values) { |
|  |  | + if(value.charAt(0)== 'A'){ |
|  |  | + listA.add(new Text(value.toString().substring(1))); |
|  |  | + } else if(value.charAt(0)== 'B'){ |
|  |  | + listB.add(new Text(value.toString().substring(1))); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + if(!listA.isEmpty() && !listB.isEmpty()) { |
|  |  | + for (Text a: listA){ |
|  |  | + for (Text b: listB){ |
|  |  | + context.write(a, b); |
|  |  | + } |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | +} |

**44** [...ect/src/main/java/com/adbms/project/airlines/locations/aggregation/LocationAggMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ded6b583183900f918bb676d7918a002)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,44 @@ |
|  |  | +package com.adbms.project.airlines.locations.aggregation; |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationAggMapper extends Mapper<LongWritable, Text, LocationCustomWritable, AggCustomWritable> { |
|  |  | + |
|  |  | + private LocationCustomWritable lcw = new LocationCustomWritable(); |
|  |  | + private AggCustomWritable acw = new AggCustomWritable(); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, LocationCustomWritable, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + try { |
|  |  | + lcw.setOrigin(fields[16]); |
|  |  | + lcw.setDestination(fields[20]); |
|  |  | + acw.setCount(1); |
|  |  | + acw.setAvg(Float.parseFloat(fields[28])); |
|  |  | + acw.setMax(Float.parseFloat(fields[28])); |
|  |  | + acw.setMin(Float.parseFloat(fields[28])); |
|  |  | + acw.setSum(0l); |
|  |  | + context.write(lcw, acw); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**72** [...ct/src/main/java/com/adbms/project/airlines/locations/aggregation/LocationAggReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-e009cbb4ee49f452028786e48c13f79c)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,72 @@ |
|  |  | +package com.adbms.project.airlines.locations.aggregation; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | +import com.adbms.project.hbase.HBaseConf; |
|  |  | +import com.adbms.project.hbase.LocationsHbase; |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +public class LocationAggReducer extends Reducer<LocationCustomWritable, AggCustomWritable, LocationCustomWritable, AggCustomWritable>{ |
|  |  | + |
|  |  | + private AggCustomWritable acw = new AggCustomWritable(); |
|  |  | + private Table table = null; |
|  |  | + private Connection conn = null; |
|  |  | + @Override |
|  |  | + protected void setup( |
|  |  | + Reducer<LocationCustomWritable, AggCustomWritable, LocationCustomWritable, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + conn = HBaseConf.getConnection(); |
|  |  | + table = HBaseConf.getLocationsTable(conn); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(LocationCustomWritable key, Iterable<AggCustomWritable> values, |
|  |  | + Reducer<LocationCustomWritable, AggCustomWritable, LocationCustomWritable, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + float sum = 0; |
|  |  | + float count = 0; |
|  |  | + float min = 100000f; |
|  |  | + float max = 0f; |
|  |  | + |
|  |  | + for(AggCustomWritable ac : values) { |
|  |  | + sum += ac.getAvg() \* ac.getCount(); |
|  |  | + count += ac.getCount(); |
|  |  | + if(ac.getMin() < min) { |
|  |  | + min = ac.getMin(); |
|  |  | + } |
|  |  | + |
|  |  | + if(ac.getMax() > max ) { |
|  |  | + max = ac.getMax(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + acw.setCount(count); |
|  |  | + acw.setAvg(sum/count); |
|  |  | + acw.setMax(max); |
|  |  | + acw.setMin(min); |
|  |  | + acw.setSum(Math.round(sum)); |
|  |  | + LocationsHbase.addLocationsToHbase(table, "2016Q4", key.toString(), acw.getMin(), acw.getMax(), acw.getAvg()); |
|  |  | + context.write(key, acw); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup( |
|  |  | + Reducer<LocationCustomWritable, AggCustomWritable, LocationCustomWritable, AggCustomWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + HBaseConf.closeConnection(conn, table); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**79** [...ject/src/main/java/com/adbms/project/airlines/locations/tojson/LocationsToJsonMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-998a71ce7f0698365a8b585abdced5a4)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,79 @@ |
|  |  | +package com.adbms.project.airlines.locations.tojson; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | +import java.io.StringWriter; |
|  |  | +import java.util.HashSet; |
|  |  | + |
|  |  | +import javax.xml.parsers.DocumentBuilder; |
|  |  | +import javax.xml.parsers.DocumentBuilderFactory; |
|  |  | +import javax.xml.transform.OutputKeys; |
|  |  | +import javax.xml.transform.Transformer; |
|  |  | +import javax.xml.transform.TransformerFactory; |
|  |  | +import javax.xml.transform.dom.DOMSource; |
|  |  | +import javax.xml.transform.stream.StreamResult; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonElement; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class LocationsToJsonMapper extends Mapper<LongWritable, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + static int count = 0; |
|  |  | + JsonArray locations = new JsonArray(); |
|  |  | + JsonObject jsonOut; |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + JsonElement location = nestElements(value.toString()); |
|  |  | + locations.add(location); |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String locationsAgg) { |
|  |  | + try{ |
|  |  | + JsonObject location = new JsonObject(); |
|  |  | + String fields[] = locationsAgg.toString().split("\t"); |
|  |  | + String orgDes[] = fields[0].split("-") ; |
|  |  | + String org = orgDes[0].substring(1, orgDes[0].length()-1);; |
|  |  | + location.addProperty("origin", org); |
|  |  | + String des = orgDes[1].substring(1, orgDes[1].length()-1); |
|  |  | + location.addProperty("destination", des); |
|  |  | + String agg[] = fields[1].split("\\|"); |
|  |  | + |
|  |  | + String minkv[] = agg[0].split(":"); |
|  |  | + String maxkv[] = agg[1].split(":"); |
|  |  | + String avgkv[] = agg[2].split(":"); |
|  |  | + location.addProperty("minFare", Float.parseFloat(minkv[1])); |
|  |  | + location.addProperty("maxFare", Float.parseFloat(maxkv[1])); |
|  |  | + location.addProperty("avgFare", Float.parseFloat(avgkv[1])); |
|  |  | + |
|  |  | + return location; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(locations)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**38** [...rc/main/java/com/adbms/project/airlines/locations/topcarrier/LocationsTopCarCombiner.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-6d1567c61da7d96d04740c0f86320e47)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,38 @@ |
|  |  | +package com.adbms.project.airlines.locations.topcarrier; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | +import java.util.Map.Entry; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.IntWritable; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.SortedMapWritable; |
|  |  | +import org.apache.hadoop.io.Writable; |
|  |  | +import org.apache.hadoop.io.WritableComparable; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationsTopCarCombiner extends Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, SortedMapWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(LocationCustomWritable key, Iterable<SortedMapWritable> values, |
|  |  | + Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, SortedMapWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + SortedMapWritable sortedMap = new SortedMapWritable(); |
|  |  | + for (SortedMapWritable val : values) { |
|  |  | + for (Entry<WritableComparable, Writable> entry : val.entrySet()) { |
|  |  | + LongWritable count = (LongWritable) sortedMap.get(entry.getKey()); |
|  |  | + |
|  |  | + if (count != null ) { |
|  |  | + count.set(count.get() + ((LongWritable) entry.getValue()).get()); |
|  |  | + } else { |
|  |  | + sortedMap.put(entry.getKey(), new LongWritable(((LongWritable) entry.getValue()).get())); |
|  |  | + } |
|  |  | + } |
|  |  | + val.clear(); |
|  |  | + } |
|  |  | + |
|  |  | + context.write(key, sortedMap); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**39** [.../src/main/java/com/adbms/project/airlines/locations/topcarrier/LocationsTopCarMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-91a1c95bff270f8a959c3f4462c6ffd6)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,39 @@ |
|  |  | +package com.adbms.project.airlines.locations.topcarrier; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.SortedMapWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationsTopCarMapper extends Mapper<LongWritable, Text, LocationCustomWritable, SortedMapWritable> { |
|  |  | + |
|  |  | + private LocationCustomWritable lcw = new LocationCustomWritable(); |
|  |  | + private SortedMapWritable sm = new SortedMapWritable(); |
|  |  | + private static LongWritable one = new LongWritable(1); |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, LocationCustomWritable, SortedMapWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + try { |
|  |  | + lcw.setOrigin(fields[16]); |
|  |  | + lcw.setDestination(fields[20]); |
|  |  | + sm.put(new Text(fields[26]), one); |
|  |  | + |
|  |  | + context.write(lcw, sm); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**59** [...src/main/java/com/adbms/project/airlines/locations/topcarrier/LocationsTopCarReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-615002966559636c5ab95d65ef69830e)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,59 @@ |
|  |  | +package com.adbms.project.airlines.locations.topcarrier; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | +import java.util.Map.Entry; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.SortedMapWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.io.Writable; |
|  |  | +import org.apache.hadoop.io.WritableComparable; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.AggCustomWritable; |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | +import com.adbms.project.hbase.HBaseConf; |
|  |  | +import com.adbms.project.hbase.LocationsHbase; |
|  |  | + |
|  |  | +public class LocationsTopCarReducer extends Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text> { |
|  |  | + |
|  |  | + private Table table = null; |
|  |  | + private Connection conn = null; |
|  |  | + @Override |
|  |  | + protected void setup( |
|  |  | + Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + conn = HBaseConf.getConnection(); |
|  |  | + table = HBaseConf.getLocationsTable(conn); |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void reduce(LocationCustomWritable key, Iterable<SortedMapWritable> values, |
|  |  | + Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Text airlines = new Text(); |
|  |  | + Long max = 0L; |
|  |  | + for (SortedMapWritable val : values) { |
|  |  | + for (Entry<WritableComparable, Writable> entry : val.entrySet()) { |
|  |  | + if (((LongWritable) entry.getValue()).get() > max) { |
|  |  | + max = ((LongWritable) entry.getValue()).get(); |
|  |  | + airlines.set(entry.getKey().toString() + " : " + entry.getValue().toString()); |
|  |  | + } |
|  |  | + } |
|  |  | + } |
|  |  | + LocationsHbase.addTopCarrierToHbase(table, "2016Q4", key.toString(), airlines.toString()); |
|  |  | + context.write(key, airlines); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup( |
|  |  | + Reducer<LocationCustomWritable, SortedMapWritable, LocationCustomWritable, Text>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + HBaseConf.closeConnection(conn, table); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**66** [...a/com/adbms/project/airlines/locations/topcarrier/tojson/LocationsTopCarToJsonMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d69c7fbc57680649d58a24f436da6ee9)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,66 @@ |
|  |  | +package com.adbms.project.airlines.locations.topcarrier.tojson; |
|  |  | + |
|  |  | +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.Mapper; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonElement; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class LocationsTopCarToJsonMapper extends Mapper<LongWritable, Text, Text, NullWritable>{ |
|  |  | + static int count = 0; |
|  |  | + JsonArray carriers = new JsonArray(); |
|  |  | + JsonObject jsonOut; |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + JsonElement carrier = nestElements(value.toString()); |
|  |  | + carriers.add(carrier); |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String locationsAgg) { |
|  |  | + try{ |
|  |  | + JsonObject carrier = new JsonObject(); |
|  |  | + |
|  |  | + String fields[] = locationsAgg.toString().split("\t"); |
|  |  | + String carrierCode = fields[0]; |
|  |  | + String cc = carrierCode.substring(1, carrierCode.length()-1);; |
|  |  | + |
|  |  | + carrier.addProperty("carrierCode", cc); |
|  |  | + String agg[] = fields[1].split("\\|"); |
|  |  | + String minkv[] = agg[0].split(":"); |
|  |  | + String maxkv[] = agg[1].split(":"); |
|  |  | + String avgkv[] = agg[2].split(":"); |
|  |  | + String sumkv[] = agg[3].split(":"); |
|  |  | + carrier.addProperty("minFare", Float.parseFloat(minkv[1])); |
|  |  | + carrier.addProperty("maxFare", Float.parseFloat(maxkv[1])); |
|  |  | + carrier.addProperty("avgFare", Float.parseFloat(avgkv[1])); |
|  |  | + carrier.addProperty("sumFare", Float.parseFloat(sumkv[1])); |
|  |  | + |
|  |  | + return carrier; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(carriers)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**28** [...oject/src/main/java/com/adbms/project/airlines/locations/topk/LocationTopCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f7041a9ac6e0f7c84bf3e9b3eabd57e5)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,28 @@ |
|  |  | +package com.adbms.project.airlines.locations.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationTopCountMapper extends Mapper<LongWritable, Text, LocationCustomWritable, LongWritable> { |
|  |  | + |
|  |  | + private LocationCustomWritable lcw = new LocationCustomWritable(); |
|  |  | + private LongWritable one = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, LocationCustomWritable, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + lcw.setOrigin(fields[16]); |
|  |  | + lcw.setDestination(fields[20]); |
|  |  | + |
|  |  | + context.write(lcw, one); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**26** [...ject/src/main/java/com/adbms/project/airlines/locations/topk/LocationTopCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f1393f8331d71bce04fd4687f17bd253)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,26 @@ |
|  |  | +package com.adbms.project.airlines.locations.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.adbms.project.airlines.locations.writables.LocationCustomWritable; |
|  |  | + |
|  |  | +public class LocationTopCountReducer extends Reducer<LocationCustomWritable, LongWritable, LocationCustomWritable, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(LocationCustomWritable key, Iterable<LongWritable> values, |
|  |  | + Reducer<LocationCustomWritable, LongWritable, LocationCustomWritable, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**19** [FinalProject/src/main/java/com/adbms/project/airlines/locations/topk/LocationTopKMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-a22244c90e631147225583898601b43d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,19 @@ |
|  |  | +package com.adbms.project.airlines.locations.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class LocationTopKMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | +} |

**58** [...lProject/src/main/java/com/adbms/project/airlines/locations/topk/LocationTopKReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-49f4e86825041952d340984aa1fed4f5)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,58 @@ |
|  |  | +package com.adbms.project.airlines.locations.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | + |
|  |  | +public class LocationTopKReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray topLocations = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 100) { |
|  |  | + topLocations.add(nestElements(key.toString())); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String srcDesTopK) { |
|  |  | + try{ |
|  |  | + JsonObject srcDes = new JsonObject(); |
|  |  | + String fields[] = srcDesTopK.toString().split(" "); |
|  |  | + String sd = fields[1]; |
|  |  | + srcDes.addProperty("srcDes", sd); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + srcDes.addProperty("count", count); |
|  |  | + return srcDes; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(topLocations)), NullWritable.get()); |
|  |  | + } |
|  |  | +} |

**20** [...ain/java/com/adbms/project/airlines/locations/topk/bloom/LocationsAggTopKBloomMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-3c80b489d83199f7f0091e3341157e7a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,20 @@ |
|  |  | +package com.adbms.project.airlines.locations.topk.bloom; |
|  |  | + |
|  |  | +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.Mapper; |
|  |  | + |
|  |  | +public class LocationsAggTopKBloomMapper extends Mapper<LongWritable, Text, Text, NullWritable>{ |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void setup(Mapper<LongWritable, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + // TODO Auto-generated method stub |
|  |  | + super.setup(context); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**91** [...oject/src/main/java/com/adbms/project/airlines/locations/writables/AggCustomWritable.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-cfca988122caee797112459cc5334ce2)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,91 @@ |
|  |  | +package com.adbms.project.airlines.locations.writables; |
|  |  | + |
|  |  | +import java.io.DataInput; |
|  |  | +import java.io.DataOutput; |
|  |  | +import java.io.IOException; |
|  |  | +import java.util.TreeMap; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.Writable; |
|  |  | +import org.apache.hadoop.io.WritableUtils; |
|  |  | + |
|  |  | +public class AggCustomWritable implements Writable { |
|  |  | + |
|  |  | + private float avg; |
|  |  | + private float min; |
|  |  | + private float max; |
|  |  | + private float count; |
|  |  | + private Long sum; |
|  |  | + private float stdDev; |
|  |  | + |
|  |  | + public void readFields(DataInput dI) throws IOException { |
|  |  | + avg = Float.parseFloat(WritableUtils.readString(dI)); |
|  |  | + min = Float.parseFloat(WritableUtils.readString(dI)); |
|  |  | + max = Float.parseFloat(WritableUtils.readString(dI)); |
|  |  | + count = Float.parseFloat(WritableUtils.readString(dI)); |
|  |  | + sum = (long) Math.round(Float.parseFloat(WritableUtils.readString(dI))); |
|  |  | + stdDev = Float.parseFloat(WritableUtils.readString(dI)); |
|  |  | + } |
|  |  | + |
|  |  | + public void write(DataOutput dO) throws IOException { |
|  |  | + WritableUtils.writeString(dO, Float.toString(avg)); |
|  |  | + WritableUtils.writeString(dO, Float.toString(min)); |
|  |  | + WritableUtils.writeString(dO, Float.toString(max)); |
|  |  | + WritableUtils.writeString(dO, Float.toString(count)); |
|  |  | + WritableUtils.writeString(dO, Float.toString(sum)); |
|  |  | + WritableUtils.writeString(dO, Float.toString(stdDev)); |
|  |  | + } |
|  |  | + |
|  |  | + public float getAvg() { |
|  |  | + return avg; |
|  |  | + } |
|  |  | + |
|  |  | + public void setAvg(float avg) { |
|  |  | + this.avg = avg; |
|  |  | + } |
|  |  | + |
|  |  | + public float getMin() { |
|  |  | + return min; |
|  |  | + } |
|  |  | + |
|  |  | + public void setMin(float min) { |
|  |  | + this.min = min; |
|  |  | + } |
|  |  | + |
|  |  | + public float getMax() { |
|  |  | + return max; |
|  |  | + } |
|  |  | + |
|  |  | + public void setMax(float max) { |
|  |  | + this.max = max; |
|  |  | + } |
|  |  | + |
|  |  | + public float getCount() { |
|  |  | + return count; |
|  |  | + } |
|  |  | + |
|  |  | + public void setCount(float count) { |
|  |  | + this.count = count; |
|  |  | + } |
|  |  | + |
|  |  | + public long getSum() { |
|  |  | + return sum; |
|  |  | + } |
|  |  | + |
|  |  | + public void setSum(long sum) { |
|  |  | + this.sum = sum; |
|  |  | + } |
|  |  | + |
|  |  | + public float getStdDev() { |
|  |  | + return stdDev; |
|  |  | + } |
|  |  | + |
|  |  | + public void setStdDev(float stdDev) { |
|  |  | + this.stdDev = stdDev; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + public String toString() { |
|  |  | + return "min:" + this.min + "|max:" + this.max + "|avg:" + this.avg ; |
|  |  | + } |
|  |  | + |
|  |  | +} |

**58** [.../src/main/java/com/adbms/project/airlines/locations/writables/LocationCustomWritable.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0f60c907f64e72d6d6f745a880c71683)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,58 @@ |
|  |  | +package com.adbms.project.airlines.locations.writables; |
|  |  | + |
|  |  | +import java.io.DataInput; |
|  |  | +import java.io.DataOutput; |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.WritableComparable; |
|  |  | +import org.apache.hadoop.io.WritableUtils; |
|  |  | + |
|  |  | +public class LocationCustomWritable implements WritableComparable<LocationCustomWritable> { |
|  |  | + |
|  |  | + private String origin; |
|  |  | + private String destination; |
|  |  | + |
|  |  | + public void readFields(DataInput dI) throws IOException { |
|  |  | + origin = WritableUtils.readString(dI); |
|  |  | + destination = WritableUtils.readString(dI); |
|  |  | + } |
|  |  | + |
|  |  | + public void write(DataOutput dO) throws IOException { |
|  |  | + WritableUtils.writeString(dO, origin); |
|  |  | + WritableUtils.writeString(dO, destination); |
|  |  | + } |
|  |  | + |
|  |  | + public int compareTo(LocationCustomWritable lc) { |
|  |  | + String cur = this.origin + this.destination; |
|  |  | + String prev = lc.origin + lc.destination; |
|  |  | + return cur.compareTo(prev); |
|  |  | + } |
|  |  | + |
|  |  | + public String getOrigin() { |
|  |  | + return origin; |
|  |  | + } |
|  |  | + |
|  |  | + public void setOrigin(String origin) { |
|  |  | + this.origin = origin; |
|  |  | + } |
|  |  | + |
|  |  | + public String getDestination() { |
|  |  | + return destination; |
|  |  | + } |
|  |  | + |
|  |  | + public void setDestination(String destination) { |
|  |  | + this.destination = destination; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + public String toString() { |
|  |  | + return this.origin + "-" + this.destination; |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**25** [...ject/src/main/java/com/adbms/project/airlines/opcarrier/topk/OpCarrierTopCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2a3fd0b0f5556415dc222f0ee2c3eabc)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.opcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class OpCarrierTopCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> { |
|  |  | + |
|  |  | + private Text outKey = new Text(); |
|  |  | + private LongWritable one = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + outKey.set(fields[26]); |
|  |  | + context.write(outKey, one); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**25** [...ect/src/main/java/com/adbms/project/airlines/opcarrier/topk/OpCarrierTopCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-9ffc76fabb50fdb6e16625e483c4910e)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.opcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class OpCarrierTopCountReducer extends Reducer<Text, LongWritable, Text, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<LongWritable> values, |
|  |  | + Reducer<Text, LongWritable, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**21** [...lProject/src/main/java/com/adbms/project/airlines/opcarrier/topk/OpCarrierTopKMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f5609e5a81abee7c465c0a945ed647b8)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,21 @@ |
|  |  | +package com.adbms.project.airlines.opcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class OpCarrierTopKMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**61** [...Project/src/main/java/com/adbms/project/airlines/opcarrier/topk/OpCarrierTopKReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f2039930aa01965dc30cadc54575c56f)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,61 @@ |
|  |  | +package com.adbms.project.airlines.opcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | + |
|  |  | +public class OpCarrierTopKReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray opCarriers = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 25) { |
|  |  | + opCarriers.add(nestElements(key.toString())); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String opCarrierTopK) { |
|  |  | + try{ |
|  |  | + |
|  |  | + JsonObject opCar = new JsonObject(); |
|  |  | + String fields[] = opCarrierTopK.toString().split(" "); |
|  |  | + String op = fields[1].substring(1, fields[1].length()-1); |
|  |  | + opCar.addProperty("opCarrier", op); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + opCar.addProperty("count", count); |
|  |  | + return opCar; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(opCarriers)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**25** [FinalProject/src/main/java/com/adbms/project/airlines/origin/topk/OriginTopCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-bf55d218db75943bdcebf147fd910d12)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.origin.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class OriginTopCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> { |
|  |  | + |
|  |  | + private Text outKey = new Text(); |
|  |  | + private LongWritable one = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + outKey.set(fields[16]); |
|  |  | + context.write(outKey, one); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**26** [FinalProject/src/main/java/com/adbms/project/airlines/origin/topk/OriginTopCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-7ff39031191dc6a7200a156bbffb8e89)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,26 @@ |
|  |  | +package com.adbms.project.airlines.origin.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class OriginTopCountReducer extends Reducer<Text, LongWritable, Text, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<LongWritable> values, |
|  |  | + Reducer<Text, LongWritable, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**20** [FinalProject/src/main/java/com/adbms/project/airlines/origin/topk/OriginTopKMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-6446259586c6bcd722fe8e9ac59463bf)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,20 @@ |
|  |  | +package com.adbms.project.airlines.origin.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class OriginTopKMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | +} |

**68** [FinalProject/src/main/java/com/adbms/project/airlines/origin/topk/OriginTopKReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-366bd965eca6ca3121a2824a9630cd04)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,68 @@ |
|  |  | +package com.adbms.project.airlines.origin.topk; |
|  |  | + |
|  |  | +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.Mapper; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | + |
|  |  | +public class OriginTopKReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray topOrigins = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 25) { |
|  |  | + JsonObject origin = nestElements(key.toString()); |
|  |  | + topOrigins.add(origin); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String originTopK) { |
|  |  | + try{ |
|  |  | + |
|  |  | + JsonObject origin = new JsonObject(); |
|  |  | + String fields[] = originTopK.toString().split(" "); |
|  |  | + String org = fields[1].substring(1, fields[1].length()-1); |
|  |  | + origin.addProperty("origin", org); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + origin.addProperty("count", count); |
|  |  | + return origin; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(topOrigins)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**30** [FinalProject/src/main/java/com/adbms/project/airlines/roundtrip/RoundTripMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-a27a7244f656cc5396ef4a089180c823)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,30 @@ |
|  |  | +package com.adbms.project.airlines.roundtrip; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class RoundTripMapper extends Mapper<Object, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + public static final String ROUNDTRIP = "Round\_Trip"; |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Object key, Text value, Mapper<Object, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + try { |
|  |  | + String[] fields = value.toString().split(","); |
|  |  | + //context.write(t1, NullWritable.get()); |
|  |  | + if (Float.parseFloat(fields[27]) == 1.0) { |
|  |  | + context.getCounter(ROUNDTRIP, "roundtrip").increment(1); |
|  |  | + } |
|  |  | + context.getCounter(ROUNDTRIP, "sum").increment(1); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | +} |

**28** [FinalProject/src/main/java/com/adbms/project/airlines/sort/DescendingSortComparator.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2a8d5f46dd15340d2a7835f3ac0cd619)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,28 @@ |
|  |  | +package com.adbms.project.airlines.sort; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.IntWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.io.WritableComparable; |
|  |  | +import org.apache.hadoop.io.WritableComparator; |
|  |  | + |
|  |  | +public class DescendingSortComparator extends WritableComparator { |
|  |  | + |
|  |  | + protected DescendingSortComparator() { |
|  |  | + super(Text.class, true); |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + public int compare(WritableComparable w1, WritableComparable w2) { |
|  |  | + Text key1 = (Text) w1; |
|  |  | + Text key2 = (Text) w2; |
|  |  | + String[] fields1 = key1.toString().split(" "); |
|  |  | + int total1 = Integer.parseInt(fields1[0]); |
|  |  | + IntWritable t1 = new IntWritable(total1); |
|  |  | + |
|  |  | + String[] fields2 = key2.toString().split(" "); |
|  |  | + int total2 = Integer.parseInt(fields2[0]); |
|  |  | + IntWritable t2 = new IntWritable(total2); |
|  |  | + |
|  |  | + return -1 \* t1.compareTo(t2); |
|  |  | + } |
|  |  | +} |

**25** [...ct/src/main/java/com/adbms/project/airlines/tktcarrier/topk/TktCarrierTopCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-8fd38c6135bad2dbf9c45eb97776932a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class TktCarrierTopCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> { |
|  |  | + |
|  |  | + private Text outKey = new Text(); |
|  |  | + private LongWritable one = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + outKey.set(fields[25]); |
|  |  | + context.write(outKey, one); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**26** [...t/src/main/java/com/adbms/project/airlines/tktcarrier/topk/TktCarrierTopCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-23270a76f16c35740157d5c221e5f5f1)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,26 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class TktCarrierTopCountReducer extends Reducer<Text, LongWritable, Text, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<LongWritable> values, |
|  |  | + Reducer<Text, LongWritable, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**21** [...roject/src/main/java/com/adbms/project/airlines/tktcarrier/topk/TktCarrierTopKMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d33ff2b8bbeb0554f3d34f16ef57cf15)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,21 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class TktCarrierTopKMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**61** [...oject/src/main/java/com/adbms/project/airlines/tktcarrier/topk/TktCarrierTopKReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-e1f316db76c02b6e48b71fe5e7be509d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,61 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.topk; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class TktCarrierTopKReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray tktCarriers = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 25) { |
|  |  | + System.out.println(key.toString()); |
|  |  | + tktCarriers.add(nestElements(key.toString())); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String tktTopK) { |
|  |  | + try{ |
|  |  | + |
|  |  | + JsonObject tktCar = new JsonObject(); |
|  |  | + String fields[] = tktTopK.toString().split(" "); |
|  |  | + String tkt = fields[1]; |
|  |  | + tktCar.addProperty("tktCarrier", tkt); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + tktCar.addProperty("count", count); |
|  |  | + return tktCar; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(tktCarriers)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**32** [...c/main/java/com/adbms/project/airlines/tktcarrier/toprev/TktCarrierTopRevCountMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-4ba2ec6d4441feb965b81a67d6ff14d3)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,32 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.toprev; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class TktCarrierTopRevCountMapper extends Mapper<LongWritable, Text, Text, LongWritable> { |
|  |  | + |
|  |  | + private Text outKey = new Text(); |
|  |  | + private LongWritable outValue = new LongWritable(1); |
|  |  | + @Override |
|  |  | + protected void map(LongWritable key, Text value, |
|  |  | + Mapper<LongWritable, Text, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + try{ |
|  |  | + String fields[] = value.toString().split(","); |
|  |  | + |
|  |  | + outKey.set(fields[25]); |
|  |  | + outValue.set((long) Math.round(Float.parseFloat(fields[28]))); |
|  |  | + context.write(outKey, outValue); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |
|  |  | + |
|  |  | + |

**25** [.../main/java/com/adbms/project/airlines/tktcarrier/toprev/TktCarrierTopRevCountReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d5da39962ebb982377ab3f2e641d1365)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,25 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.toprev; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.LongWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +public class TktCarrierTopRevCountReducer extends Reducer<Text, LongWritable, Text, LongWritable> { |
|  |  | + |
|  |  | + private LongWritable total = new LongWritable(); |
|  |  | + @Override |
|  |  | + protected void reduce(Text key, Iterable<LongWritable> values, |
|  |  | + Reducer<Text, LongWritable, Text, LongWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Long sum = 0l; |
|  |  | + for (LongWritable l : values) { |
|  |  | + sum += l.get(); |
|  |  | + } |
|  |  | + total.set(sum); |
|  |  | + context.write(key, total); |
|  |  | + } |
|  |  | + |
|  |  | +} |

**20** [...ct/src/main/java/com/adbms/project/airlines/tktcarrier/toprev/TktCarrierTopRevMapper.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0bb196331ab83058ca023388aa165068)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,20 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.toprev; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Mapper; |
|  |  | + |
|  |  | +public class TktCarrierTopRevMapper extends Mapper<Text, Text, Text, NullWritable> { |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void map(Text key, Text value, Mapper<Text, Text, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + Text outKey = new Text(value.toString() + " " + key.toString()); |
|  |  | + context.write(outKey, NullWritable.get()); |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | +} |

**61** [...t/src/main/java/com/adbms/project/airlines/tktcarrier/toprev/TktCarrierTopRevReducer.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2ba034c421d00f3e6d9d42fad528ad2d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,61 @@ |
|  |  | +package com.adbms.project.airlines.tktcarrier.toprev; |
|  |  | + |
|  |  | +import java.io.IOException; |
|  |  | + |
|  |  | +import org.apache.hadoop.io.NullWritable; |
|  |  | +import org.apache.hadoop.io.Text; |
|  |  | +import org.apache.hadoop.mapreduce.Reducer; |
|  |  | + |
|  |  | +import com.google.gson.FieldNamingPolicy; |
|  |  | +import com.google.gson.Gson; |
|  |  | +import com.google.gson.GsonBuilder; |
|  |  | +import com.google.gson.JsonArray; |
|  |  | +import com.google.gson.JsonObject; |
|  |  | + |
|  |  | +public class TktCarrierTopRevReducer extends Reducer<Text, NullWritable, Text, NullWritable> { |
|  |  | + |
|  |  | + private int count = 0; |
|  |  | + JsonArray tktCarriers = new JsonArray(); |
|  |  | + @Override |
|  |  | + protected void setup(Context context) throws IOException, InterruptedException { |
|  |  | + count = 0; |
|  |  | + } |
|  |  | + |
|  |  | + public void reduce(Text key, Iterable<NullWritable> values,Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + |
|  |  | + if (count < 25) { |
|  |  | + tktCarriers.add(nestElements(key.toString())); |
|  |  | + //context.write(key, NullWritable.get()); |
|  |  | + count ++; |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + private JsonObject nestElements(String tktTopK) { |
|  |  | + try{ |
|  |  | + |
|  |  | + JsonObject tktCar = new JsonObject(); |
|  |  | + String fields[] = tktTopK.toString().split(" "); |
|  |  | + String tkt = fields[1]; |
|  |  | + tktCar.addProperty("tktCarrier", tkt); |
|  |  | + Long count = Long.parseLong(fields[0]); |
|  |  | + tktCar.addProperty("sum", count); |
|  |  | + return tktCar; |
|  |  | + } |
|  |  | + catch(Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + @Override |
|  |  | + protected void cleanup(Reducer<Text, NullWritable, Text, NullWritable>.Context context) |
|  |  | + throws IOException, InterruptedException { |
|  |  | + Gson gson = new GsonBuilder().setPrettyPrinting().serializeNulls().setFieldNamingPolicy(FieldNamingPolicy.UPPER\_CAMEL\_CASE).create(); |
|  |  | + context.write(new Text(gson.toJson(tktCarriers)), NullWritable.get()); |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**37** [FinalProject/src/main/java/com/adbms/project/hbase/CarrierHbase.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-515dbba17e181640e51c655fc0b75544)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,37 @@ |
|  |  | +package com.adbms.project.hbase; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.TableName; |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Put; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | +public class CarrierHbase { |
|  |  | + |
|  |  | + public static Table getCarriersTable(Connection conn) { |
|  |  | + try { |
|  |  | + Table table = conn.getTable(TableName.valueOf(Bytes.toBytes("carriersAdbms"))); |
|  |  | + return table; |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | + public static void addCarriersToHbase (Table table, String cf, String carrierCode, Float minFare, Float maxFare, Float avgFare, Long sum) { |
|  |  | + try { |
|  |  | + Put put = new Put (Bytes.toBytes(carrierCode)); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("minFare"), Bytes.toBytes(minFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("maxFare"), Bytes.toBytes(maxFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("avgFare"), Bytes.toBytes(avgFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("sumFare"), Bytes.toBytes(sum.toString())); |
|  |  | + table.put(put); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | +} |

**53** [FinalProject/src/main/java/com/adbms/project/hbase/HBaseConf.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-19537a708c239b3421737a269230329a)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,53 @@ |
|  |  | +package com.adbms.project.hbase; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.hbase.HBaseConfiguration; |
|  |  | +import org.apache.hadoop.hbase.TableName; |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.ConnectionFactory; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | +public class HBaseConf { |
|  |  | + |
|  |  | + public static Connection getConnection() { |
|  |  | + Connection conn = null; |
|  |  | + try { |
|  |  | + Configuration conf = HBaseConfiguration.create(); |
|  |  | + conn = ConnectionFactory.createConnection(conf); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + if (conn != null) { |
|  |  | + System.out.println("Successfull connection"); |
|  |  | + } else { |
|  |  | + System.out.println("Connection failed"); |
|  |  | + } |
|  |  | + return conn; |
|  |  | + } |
|  |  | + |
|  |  | + public static void closeConnection (Connection conn, Table table) { |
|  |  | + try { |
|  |  | + conn.close(); |
|  |  | + table.close(); |
|  |  | + System.out.println("Connection closed"); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + public static Table getLocationsTable(Connection conn) { |
|  |  | + try { |
|  |  | + Table table = conn.getTable(TableName.valueOf(Bytes.toBytes("locationsAdbms"))); |
|  |  | + return table; |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | +} |

**32** [FinalProject/src/main/java/com/adbms/project/hbase/LocationsHbase.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-99088aba23c0acbdbe323475c9c51e87)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,32 @@ |
|  |  | +package com.adbms.project.hbase; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.client.Put; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | + |
|  |  | +public class LocationsHbase { |
|  |  | + public static void addLocationsToHbase (Table table, String cf, String srcDes, Float minFare, Float maxFare, Float avgFare) { |
|  |  | + try { |
|  |  | + Put put = new Put (Bytes.toBytes(srcDes)); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("minFare"), Bytes.toBytes(minFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("maxFare"), Bytes.toBytes(maxFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("avgFare"), Bytes.toBytes(avgFare.toString())); |
|  |  | + table.put(put); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + public static void addTopCarrierToHbase (Table table, String cf, String srcDes, String carrierCode) { |
|  |  | + try { |
|  |  | + Put put = new Put (Bytes.toBytes(srcDes)); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("carrierCode"), Bytes.toBytes(carrierCode)); |
|  |  | + table.put(put); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | +} |

**5** [FinalProject/target/classes/META-INF/MANIFEST.MF](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-9476456f7fe5f46458869fbb13205e69)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,5 @@ |
|  |  | +Manifest-Version: 1.0 |
|  |  | +Built-By: saravandeepak |
|  |  | +Build-Jdk: 1.8.0\_111 |
|  |  | +Created-By: Maven Integration for Eclipse |
|  |  | + |

**7** [FinalProject/target/classes/META-INF/maven/FinalProject/FinalProject/pom.properties](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-72d113384df52d3bf85e264e6a12fe2b)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,7 @@ |
|  |  | +#Generated by Maven Integration for Eclipse |
|  |  | +#Mon Dec 11 10:41:31 EST 2017 |
|  |  | +version=0.0.1-SNAPSHOT |
|  |  | +groupId=FinalProject |
|  |  | +m2e.projectName=FinalProject |
|  |  | +m2e.projectLocation=/Users/saravandeepak/ADBMS/FinalProject |
|  |  | +artifactId=FinalProject |

**63** [FinalProject/target/classes/META-INF/maven/FinalProject/FinalProject/pom.xml](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-564893b9964067beb0509690553a1b51)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,63 @@ |
|  |  | +<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"> |
|  |  | + <modelVersion>4.0.0</modelVersion> |
|  |  | + <groupId>FinalProject</groupId> |
|  |  | + <artifactId>FinalProject</artifactId> |
|  |  | + <version>0.0.1-SNAPSHOT</version> |
|  |  | + <name>FinalProject</name> |
|  |  | + <description>ADBMS Final Project</description> |
|  |  | + <dependencies> |
|  |  | + <dependency> |
|  |  | + <groupId>junit</groupId> |
|  |  | + <artifactId>junit</artifactId> |
|  |  | + <version>3.8.1</version> |
|  |  | + <scope>test</scope> |
|  |  | + </dependency> |
|  |  | + <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-client-core --> |
|  |  | + |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hadoop</groupId> |
|  |  | + <artifactId>hadoop-mapreduce-client-core</artifactId> |
|  |  | + <version>2.8.0</version> |
|  |  | + </dependency> |
|  |  | + <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-common --> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hadoop</groupId> |
|  |  | + <artifactId>hadoop-common</artifactId> |
|  |  | + <version>2.8.0</version> |
|  |  | + </dependency> |
|  |  | + <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-client-common --> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hadoop</groupId> |
|  |  | + <artifactId>hadoop-mapreduce-client-common</artifactId> |
|  |  | + <version>2.8.0</version> |
|  |  | + </dependency> |
|  |  | + <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-client-jobclient --> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hadoop</groupId> |
|  |  | + <artifactId>hadoop-mapreduce-client-jobclient</artifactId> |
|  |  | + <version>2.8.0</version> |
|  |  | + <scope>provided</scope> |
|  |  | + </dependency> |
|  |  | + <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core --> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.logging.log4j</groupId> |
|  |  | + <artifactId>log4j-core</artifactId> |
|  |  | + <version>2.8</version> |
|  |  | + </dependency> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hbase</groupId> |
|  |  | + <artifactId>hbase-common</artifactId> |
|  |  | + <version>1.0.0</version> |
|  |  | + </dependency> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hbase</groupId> |
|  |  | + <artifactId>hbase-client</artifactId> |
|  |  | + <version>1.0.0</version> |
|  |  | + </dependency> |
|  |  | + <dependency> |
|  |  | + <groupId>org.apache.hbase</groupId> |
|  |  | + <artifactId>hbase-server</artifactId> |
|  |  | + <version>1.0.0</version> |
|  |  | + </dependency> |
|  |  | + </dependencies> |
|  |  | +</project> |

**16** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/SpringmvcApplication.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-3bdf83435370620c7110d4e3eb5566eb)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,16 @@ |
|  |  | +package com.adbms.springmvc; |
|  |  | + |
|  |  | +import org.springframework.boot.SpringApplication; |
|  |  | +import org.springframework.boot.autoconfigure.SpringBootApplication; |
|  |  | +import org.springframework.boot.builder.SpringApplicationBuilder; |
|  |  | + |
|  |  | +@SpringBootApplication |
|  |  | +public class SpringmvcApplication { |
|  |  | + |
|  |  | + protected SpringApplicationBuilder configure (SpringApplicationBuilder application) { |
|  |  | + return application.sources(SpringmvcApplication.class); |
|  |  | + } |
|  |  | + public static void main(String[] args) { |
|  |  | + SpringApplication.run(SpringmvcApplication.class, args); |
|  |  | + } |
|  |  | +} |

**38** [...ngMVC/springmvc/src/main/java/com/adbms/springmvc/controllers/CarriersRestController.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-1d8977150d1ccdf3bef265593fc9cf9d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,38 @@ |
|  |  | +package com.adbms.springmvc.controllers; |
|  |  | + |
|  |  | +import java.util.ArrayList; |
|  |  | + |
|  |  | +import javax.servlet.http.HttpServletRequest; |
|  |  | + |
|  |  | +import org.springframework.web.bind.annotation.RequestMapping; |
|  |  | +import org.springframework.web.bind.annotation.RequestMethod; |
|  |  | +import org.springframework.web.bind.annotation.RestController; |
|  |  | + |
|  |  | +import com.adbms.springmvc.hbase.CarriersHbase; |
|  |  | +import com.adbms.springmvc.pojo.Carrier; |
|  |  | + |
|  |  | +@RestController |
|  |  | +public class CarriersRestController { |
|  |  | + |
|  |  | + @RequestMapping(value = "/api/v1/getCarrierAgg", method = RequestMethod.GET, produces = "application/json") |
|  |  | + public ArrayList<Carrier> getLocationAgg (HttpServletRequest request) { |
|  |  | + System.out.println("inside the rest controller"); |
|  |  | + String cc = request.getParameter("cc"); |
|  |  | + ArrayList<Carrier> carriersArray = new ArrayList<>(); |
|  |  | + ArrayList<String> array = new ArrayList<>(); |
|  |  | + array.add("2017Q2"); |
|  |  | + array.add("2017Q1"); |
|  |  | + array.add("2016Q4"); |
|  |  | + array.add("2016Q3"); |
|  |  | + array.add("2016Q2"); |
|  |  | + array.add("2016Q1"); |
|  |  | + |
|  |  | + |
|  |  | + for (String a: array) { |
|  |  | + Carrier carrier = CarriersHbase.getCarriersFromHbase(cc, a); |
|  |  | + carriersArray.add(carrier); |
|  |  | + } |
|  |  | + |
|  |  | + return carriersArray; |
|  |  | + } |
|  |  | +} |

**14** [...jectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/controllers/HomeController.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ba2ea8d750a061e8d2f5181e261d6c16)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,14 @@ |
|  |  | +package com.adbms.springmvc.controllers; |
|  |  | + |
|  |  | +import org.springframework.stereotype.Controller; |
|  |  | +import org.springframework.web.bind.annotation.GetMapping; |
|  |  | + |
|  |  | +@Controller |
|  |  | +public class HomeController { |
|  |  | + |
|  |  | + @GetMapping(value= {"/", "home.htm"}) |
|  |  | + public String index() { |
|  |  | + return "index"; |
|  |  | + } |
|  |  | + |
|  |  | +} |

**38** [...gMVC/springmvc/src/main/java/com/adbms/springmvc/controllers/LocationsRestController.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-083c764ab17e0b8c08dd5c3f3a9268ef)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,38 @@ |
|  |  | +package com.adbms.springmvc.controllers; |
|  |  | + |
|  |  | +import java.util.ArrayList; |
|  |  | + |
|  |  | +import javax.servlet.http.HttpServletRequest; |
|  |  | + |
|  |  | +import org.springframework.web.bind.annotation.RequestMapping; |
|  |  | +import org.springframework.web.bind.annotation.RequestMethod; |
|  |  | +import org.springframework.web.bind.annotation.RestController; |
|  |  | + |
|  |  | +import com.adbms.springmvc.hbase.LocationsHbase; |
|  |  | +import com.adbms.springmvc.pojo.Location; |
|  |  | + |
|  |  | +@RestController |
|  |  | +public class LocationsRestController { |
|  |  | + |
|  |  | + @RequestMapping(value = "/api/v1/getLocationAgg", method = RequestMethod.GET, produces = "application/json") |
|  |  | + public ArrayList<Location> getLocationAgg (HttpServletRequest request) { |
|  |  | + System.out.println("inside the rest controller"); |
|  |  | + String src = request.getParameter("src"); |
|  |  | + String des = request.getParameter("des"); |
|  |  | + ArrayList<Location> locationsArray = new ArrayList<>(); |
|  |  | + ArrayList<String> array = new ArrayList<>(); |
|  |  | + array.add("2017Q2"); |
|  |  | + array.add("2017Q1"); |
|  |  | + array.add("2016Q4"); |
|  |  | + array.add("2016Q3"); |
|  |  | + array.add("2016Q2"); |
|  |  | + array.add("2016Q1"); |
|  |  | + |
|  |  | + for (String a: array) { |
|  |  | + Location location = LocationsHbase.getLocationsFromHbase(src, des, a); |
|  |  | + locationsArray.add(location); |
|  |  | + } |
|  |  | + |
|  |  | + return locationsArray; |
|  |  | + } |
|  |  | +} |

**51** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/hbase/CarriersHbase.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-7e2ed36312067af4a30af0f81a9661d8)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,51 @@ |
|  |  | +package com.adbms.springmvc.hbase; |
|  |  | + |
|  |  | +import java.util.NavigableMap; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.TableName; |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Get; |
|  |  | +import org.apache.hadoop.hbase.client.Result; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | +import com.adbms.springmvc.pojo.Carrier; |
|  |  | + |
|  |  | +public class CarriersHbase { |
|  |  | + public static Carrier getCarriersFromHbase (String cc, String cf) { |
|  |  | + try{ |
|  |  | + Connection conn = HBaseConf.getConnection(); |
|  |  | + Table table = conn.getTable(TableName.valueOf( Bytes.toBytes("carriersAdbms"))); |
|  |  | + Get get = new Get(Bytes.toBytes("\"" + cc + "\"")); |
|  |  | + Result result = table.get(get); |
|  |  | + HBaseConf.closeConnection(conn, table); |
|  |  | + NavigableMap<byte[], byte[]> map = result.getFamilyMap(Bytes.toBytes(cf)); |
|  |  | + Carrier carrier = new Carrier(); |
|  |  | + carrier.setCc(cc); |
|  |  | + carrier.setTime(cf); |
|  |  | + |
|  |  | + for(byte[] key : map.keySet()){ |
|  |  | + if(Bytes.toString(key).equals("avgFare")) { |
|  |  | + carrier.setAvgFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("minFare")) { |
|  |  | + carrier.setMinFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("maxFare")) { |
|  |  | + carrier.setMaxFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("sumFare")) { |
|  |  | + carrier.setSumFare(Long.parseLong(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + return carrier; |
|  |  | + |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | +} |

**53** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/hbase/HBaseConf.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-b6d939b9d059cc6b45dce7bfb72e1ddb)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,53 @@ |
|  |  | +package com.adbms.springmvc.hbase; |
|  |  | + |
|  |  | +import org.apache.hadoop.conf.Configuration; |
|  |  | +import org.apache.hadoop.hbase.HBaseConfiguration; |
|  |  | +import org.apache.hadoop.hbase.TableName; |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.ConnectionFactory; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | +public class HBaseConf { |
|  |  | + |
|  |  | + public static Connection getConnection() { |
|  |  | + Connection conn = null; |
|  |  | + try { |
|  |  | + Configuration conf = HBaseConfiguration.create(); |
|  |  | + conn = ConnectionFactory.createConnection(conf); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + |
|  |  | + if (conn != null) { |
|  |  | + System.out.println("Successfull connection"); |
|  |  | + } else { |
|  |  | + System.out.println("Connection failed"); |
|  |  | + } |
|  |  | + return conn; |
|  |  | + } |
|  |  | + |
|  |  | + public static void closeConnection (Connection conn, Table table) { |
|  |  | + try { |
|  |  | + conn.close(); |
|  |  | + table.close(); |
|  |  | + System.out.println("Connection closed"); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + public static Table getLocationsTable(Connection conn) { |
|  |  | + try { |
|  |  | + Table table = conn.getTable(TableName.valueOf(Bytes.toBytes("locationsAdbms"))); |
|  |  | + return table; |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | + |
|  |  | +} |

**69** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/hbase/LocationsHbase.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-60c3bbb6873b2305d19a6d1ceec841d9)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,69 @@ |
|  |  | +package com.adbms.springmvc.hbase; |
|  |  | + |
|  |  | +import java.util.ArrayList; |
|  |  | +import java.util.Map; |
|  |  | +import java.util.NavigableMap; |
|  |  | + |
|  |  | +import org.apache.hadoop.hbase.TableName; |
|  |  | +import org.apache.hadoop.hbase.client.Connection; |
|  |  | +import org.apache.hadoop.hbase.client.Get; |
|  |  | +import org.apache.hadoop.hbase.client.Put; |
|  |  | +import org.apache.hadoop.hbase.client.Table; |
|  |  | +import org.apache.hadoop.hbase.util.Bytes; |
|  |  | + |
|  |  | +import com.adbms.springmvc.pojo.Location; |
|  |  | +import org.apache.hadoop.hbase.client.Result; |
|  |  | + |
|  |  | + |
|  |  | +public class LocationsHbase { |
|  |  | + public static void addLocationsToHbase (Table table, String cf, String srcDes, Float minFare, Float maxFare, Float avgFare) { |
|  |  | + try { |
|  |  | + Put put = new Put (Bytes.toBytes(srcDes)); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("minFare"), Bytes.toBytes(minFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("maxFare"), Bytes.toBytes(maxFare.toString())); |
|  |  | + put.addColumn(Bytes.toBytes(cf), Bytes.toBytes("avgFare"), Bytes.toBytes(avgFare.toString())); |
|  |  | + table.put(put); |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + e.printStackTrace(); |
|  |  | + } |
|  |  | + } |
|  |  | + |
|  |  | + public static Location getLocationsFromHbase (String src, String des, String cf) { |
|  |  | + try{ |
|  |  | + Connection conn = HBaseConf.getConnection(); |
|  |  | + Table table = conn.getTable(TableName.valueOf( Bytes.toBytes("locationsAdbms"))); |
|  |  | + Get get = new Get(Bytes.toBytes("\"" + src + "\"" + "-" + "\"" + des + "\"")); |
|  |  | + Result result = table.get(get); |
|  |  | + HBaseConf.closeConnection(conn, table); |
|  |  | + NavigableMap<byte[], byte[]> map = result.getFamilyMap(Bytes.toBytes(cf)); |
|  |  | + Location location = new Location(); |
|  |  | + location.setSrc(src); |
|  |  | + location.setDes(des); |
|  |  | + location.setTime(cf); |
|  |  | + |
|  |  | + for(byte[] key : map.keySet()){ |
|  |  | + if(Bytes.toString(key).equals("avgFare")) { |
|  |  | + location.setAvgFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("minFare")) { |
|  |  | + location.setMinFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("maxFare")) { |
|  |  | + location.setMaxFare(Float.parseFloat(Bytes.toString(map.get(key)))); |
|  |  | + } |
|  |  | + if(Bytes.toString(key).equals("carrierCode")) { |
|  |  | + location.setCc(Bytes.toString(map.get(key))); |
|  |  | + } |
|  |  | + |
|  |  | + } |
|  |  | + |
|  |  | + return location; |
|  |  | + |
|  |  | + } |
|  |  | + catch (Exception e) { |
|  |  | + |
|  |  | + } |
|  |  | + return null; |
|  |  | + } |
|  |  | +} |

**51** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/pojo/Carrier.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d16a628045c0caed08e2ec5eaaa17d92)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,51 @@ |
|  |  | +package com.adbms.springmvc.pojo; |
|  |  | + |
|  |  | +public class Carrier { |
|  |  | + private String cc; |
|  |  | + private Float minFare; |
|  |  | + private Float maxFare; |
|  |  | + private Float avgFare; |
|  |  | + private String time; |
|  |  | + private Long sumFare; |
|  |  | + |
|  |  | + public String getCc() { |
|  |  | + return cc; |
|  |  | + } |
|  |  | + public void setCc(String cc) { |
|  |  | + this.cc = cc; |
|  |  | + } |
|  |  | + public Float getMinFare() { |
|  |  | + return minFare; |
|  |  | + } |
|  |  | + public void setMinFare(Float minFare) { |
|  |  | + this.minFare = minFare; |
|  |  | + } |
|  |  | + public Float getMaxFare() { |
|  |  | + return maxFare; |
|  |  | + } |
|  |  | + public void setMaxFare(Float maxFare) { |
|  |  | + this.maxFare = maxFare; |
|  |  | + } |
|  |  | + public Float getAvgFare() { |
|  |  | + return avgFare; |
|  |  | + } |
|  |  | + public void setAvgFare(Float avgFare) { |
|  |  | + this.avgFare = avgFare; |
|  |  | + } |
|  |  | + public String getTime() { |
|  |  | + return time; |
|  |  | + } |
|  |  | + public void setTime(String time) { |
|  |  | + this.time = time; |
|  |  | + } |
|  |  | + public Long getSumFare() { |
|  |  | + return sumFare; |
|  |  | + } |
|  |  | + public void setSumFare(Long sumFare) { |
|  |  | + this.sumFare = sumFare; |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**58** [FinalProjectSpringMVC/springmvc/src/main/java/com/adbms/springmvc/pojo/Location.java](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-16375eaf51d5fb0989f0b66757054add)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,58 @@ |
|  |  | +package com.adbms.springmvc.pojo; |
|  |  | + |
|  |  | +public class Location { |
|  |  | + |
|  |  | + private String src; |
|  |  | + private String des; |
|  |  | + private Float minFare; |
|  |  | + private Float maxFare; |
|  |  | + private Float avgFare; |
|  |  | + private String time; |
|  |  | + private String cc; |
|  |  | + |
|  |  | + public String getSrc() { |
|  |  | + return src; |
|  |  | + } |
|  |  | + public void setSrc(String src) { |
|  |  | + this.src = src; |
|  |  | + } |
|  |  | + public String getDes() { |
|  |  | + return des; |
|  |  | + } |
|  |  | + public void setDes(String des) { |
|  |  | + this.des = des; |
|  |  | + } |
|  |  | + public Float getMinFare() { |
|  |  | + return minFare; |
|  |  | + } |
|  |  | + public void setMinFare(Float minFare) { |
|  |  | + this.minFare = minFare; |
|  |  | + } |
|  |  | + public Float getMaxFare() { |
|  |  | + return maxFare; |
|  |  | + } |
|  |  | + public void setMaxFare(Float maxFare) { |
|  |  | + this.maxFare = maxFare; |
|  |  | + } |
|  |  | + public Float getAvgFare() { |
|  |  | + return avgFare; |
|  |  | + } |
|  |  | + public void setAvgFare(Float avgFare) { |
|  |  | + this.avgFare = avgFare; |
|  |  | + } |
|  |  | + public String getTime() { |
|  |  | + return time; |
|  |  | + } |
|  |  | + public void setTime(String time) { |
|  |  | + this.time = time; |
|  |  | + } |
|  |  | + public String getCc() { |
|  |  | + return cc; |
|  |  | + } |
|  |  | + public void setCc(String cc) { |
|  |  | + this.cc = cc; |
|  |  | + } |
|  |  | + |
|  |  | + |
|  |  | + |
|  |  | +} |

**3** [FinalProjectSpringMVC/springmvc/src/main/resources/application.properties](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d1c5bd3c1fc8e370d7a17aaddb062ca4)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,3 @@ |
|  |  | +server.context-path = /springmvc |
|  |  | + |
|  |  | +server.port: 4444 |

**BIN**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/.filtermerge.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-e7d17091f741ecbfc0100bf2c9032f6c)

Binary file not shown.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q1/TopRevTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f49f711d5800c2abdf971fba67917f5f)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/TopRevTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-48dc501e673109d6c4b4a6a338820e49)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/TopRevTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-271b8b0699ab983a3d03e8031922f6d8)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "sum": 678058105 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "sum": 629214278 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "sum": 455175009 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "sum": 313256695 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "sum": 107478346 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "sum": 78009197 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "sum": 51345090 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "sum": 20602335 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "sum": 15755989 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "sum": 10641444 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "sum": 7749156 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "sum": 2338642 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "sum": 2327974 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "sum": 10932 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "sum": 4635 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "sum": 3599 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VA\"", |
|  |  | + "sum": 3336 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NZ\"", |
|  |  | + "sum": 3056 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "sum": 2926 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "sum": 2675 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "sum": 2396 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EI\"", |
|  |  | + "sum": 1281 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "sum": 1218 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EK\"", |
|  |  | + "sum": 1214 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SK\"", |
|  |  | + "sum": 470 |
|  |  | + } |
|  |  | +] |

**BIN**[...MVC/springmvc/src/main/resources/static/2016Q1/locationTopCarrierOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-80e6e44af7dda7a0910995aa6fa57a88)

Binary file not shown.

**0**[...ectSpringMVC/springmvc/src/main/resources/static/2016Q1/locationTopCarrierOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-50284c33aefbe67df8cd1aac9da75cf6)

No changes.

**62,262** [...pringMVC/springmvc/src/main/resources/static/2016Q1/locationTopCarrierOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-27af356d8da9ec7abef0cb17be1414ab)

Load diff

Large diffs are not rendered by default.

**BIN**[...pringMVC/springmvc/src/main/resources/static/2016Q1/locations/aggOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-78e34b0a8960b988dfefb2d7f97c7e9f)

Binary file not shown.

**0**[...lProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/locations/aggOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ec264c7b8457d3260010114e150be54b)

No changes.

**62,261** [...jectSpringMVC/springmvc/src/main/resources/static/2016Q1/locations/aggOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-a1f06b5691fc74cfd82a9624c2a8d562)

Load diff

Large diffs are not rendered by default.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q1/topkDestinations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-f693577938c87675fb08ad5a71f76206)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkDestinations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-8796495568139e8f63bd4631cd0f02d6)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkDestinations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-06426d3b564ac6d537d31f353b6f1ba7)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "destination": "LAX", |
|  |  | + "count": 261861 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SFO", |
|  |  | + "count": 212068 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MCO", |
|  |  | + "count": 201333 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ORD", |
|  |  | + "count": 195120 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LAS", |
|  |  | + "count": 186471 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "BOS", |
|  |  | + "count": 186149 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ATL", |
|  |  | + "count": 180927 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DEN", |
|  |  | + "count": 179892 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHX", |
|  |  | + "count": 178196 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SEA", |
|  |  | + "count": 164481 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LGA", |
|  |  | + "count": 161293 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DFW", |
|  |  | + "count": 156397 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "EWR", |
|  |  | + "count": 149984 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DCA", |
|  |  | + "count": 149832 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MSP", |
|  |  | + "count": 149352 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "FLL", |
|  |  | + "count": 143521 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SAN", |
|  |  | + "count": 143289 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "TPA", |
|  |  | + "count": 133433 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHL", |
|  |  | + "count": 130999 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DTW", |
|  |  | + "count": 123228 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SLC", |
|  |  | + "count": 117771 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "IAH", |
|  |  | + "count": 116424 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PDX", |
|  |  | + "count": 112634 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MIA", |
|  |  | + "count": 111464 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "AUS", |
|  |  | + "count": 108731 |
|  |  | + } |
|  |  | +] |

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkLocations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-5aaaff1dee15e59aee95b0ec9695b04d)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkLocations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-7c387535ec3549612995638a2ee036db)

No changes.

**402** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkLocations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-8ea81de4e1f53e66e7944586f755e073)

Load diff

Large diffs are not rendered by default.

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOpCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-465aa24d1a43c5894519ac52e559fc14)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOpCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-aa02a0d27fe64e4d3f67f3406c0d6f04)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOpCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ff3d304294990c1698f3a62fdc8d6411)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "opCarrier": "99", |
|  |  | + "count": 2715794 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "WN", |
|  |  | + "count": 1384712 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "DL", |
|  |  | + "count": 1209236 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AA", |
|  |  | + "count": 1092597 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "UA", |
|  |  | + "count": 600848 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "B6", |
|  |  | + "count": 218327 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OO", |
|  |  | + "count": 185437 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AS", |
|  |  | + "count": 155911 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "EV", |
|  |  | + "count": 118079 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "NK", |
|  |  | + "count": 88860 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YX", |
|  |  | + "count": 81440 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "9E", |
|  |  | + "count": 72300 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OH", |
|  |  | + "count": 71136 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YV", |
|  |  | + "count": 67782 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "MQ", |
|  |  | + "count": 63480 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "HA", |
|  |  | + "count": 60707 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "F9", |
|  |  | + "count": 56789 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "VX", |
|  |  | + "count": 53034 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "S5", |
|  |  | + "count": 42723 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "CP", |
|  |  | + "count": 40269 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "ZW", |
|  |  | + "count": 39498 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "QX", |
|  |  | + "count": 37395 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G7", |
|  |  | + "count": 33396 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G4", |
|  |  | + "count": 24601 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "SY", |
|  |  | + "count": 10622 |
|  |  | + } |
|  |  | +] |

**BIN**[...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOrigins/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-320eb653960ba36e1bd1194e11004a7a)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOrigins/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-1aeb36bdf192643940c8df33b5a2b655)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkOrigins/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-79a9e9d59d7b146a0005639f99996052)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "origin": "LAX", |
|  |  | + "count": 263788 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SFO", |
|  |  | + "count": 211352 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MCO", |
|  |  | + "count": 206093 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ORD", |
|  |  | + "count": 196324 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LAS", |
|  |  | + "count": 192102 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "BOS", |
|  |  | + "count": 182634 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ATL", |
|  |  | + "count": 181415 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DEN", |
|  |  | + "count": 181073 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHX", |
|  |  | + "count": 180305 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SEA", |
|  |  | + "count": 162379 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LGA", |
|  |  | + "count": 160418 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DFW", |
|  |  | + "count": 157556 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MSP", |
|  |  | + "count": 148764 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DCA", |
|  |  | + "count": 148516 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "FLL", |
|  |  | + "count": 147917 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "EWR", |
|  |  | + "count": 147908 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SAN", |
|  |  | + "count": 143515 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "TPA", |
|  |  | + "count": 134869 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHL", |
|  |  | + "count": 129886 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DTW", |
|  |  | + "count": 122949 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SLC", |
|  |  | + "count": 118984 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "IAH", |
|  |  | + "count": 117083 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MIA", |
|  |  | + "count": 115731 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PDX", |
|  |  | + "count": 109538 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "AUS", |
|  |  | + "count": 108547 |
|  |  | + } |
|  |  | +] |

**BIN**[...jectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-3c75d175110d3d2aa049059338c33b42)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-26f3feb96a14ebd96a2db0f06aea15fa)

No changes.

**90** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q1/topkTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-c57934ba1346b58b5ac895f04a461646)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,90 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "count": 2365654 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "count": 2135185 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "count": 1494763 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "count": 1384712 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "count": 336056 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "count": 305888 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "count": 220025 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "count": 88860 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "count": 64195 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "count": 56789 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "count": 53024 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "count": 24601 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "count": 10622 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "count": 112 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "count": 29 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "count": 27 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "count": 26 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "count": 10 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "count": 4 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"MW\"", |
|  |  | + "count": 3 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NZ\"", |
|  |  | + "count": 2 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"LX\"", |
|  |  | + "count": 1 |
|  |  | + } |
|  |  | +] |

**BIN**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/.filtermerge.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-c29935d51a10b246301ba9f3c2776843)

Binary file not shown.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q2/TopRevTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-efc5f16632cfef05406bf67cfbc8910e)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/TopRevTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-939329f050cdd8792b60bed498931c45)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/TopRevTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-62d67e331623b7228bc6ce55b140b985)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "sum": 741452432 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "sum": 727977766 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "sum": 507756653 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "sum": 372704067 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "sum": 117393217 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "sum": 92101357 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "sum": 54918107 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "sum": 24156934 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "sum": 17135495 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "sum": 12163671 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "sum": 6603966 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "sum": 2683680 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "sum": 2154510 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "sum": 12940 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "sum": 6025 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VA\"", |
|  |  | + "sum": 4473 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "sum": 3876 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "sum": 3635 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "sum": 3520 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EI\"", |
|  |  | + "sum": 2018 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "sum": 1379 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"LH\"", |
|  |  | + "sum": 1342 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"LY\"", |
|  |  | + "sum": 949 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "sum": 914 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"CA\"", |
|  |  | + "sum": 712 |
|  |  | + } |
|  |  | +] |

**BIN**[...MVC/springmvc/src/main/resources/static/2016Q2/locationTopCarrierOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-4d5921877503ce936d79f5bd8b5f7a47)

Binary file not shown.

**0**[...ectSpringMVC/springmvc/src/main/resources/static/2016Q2/locationTopCarrierOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2e4e77a10a0afb8b4d4f37eee1f26e36)

No changes.

**64,906** [...pringMVC/springmvc/src/main/resources/static/2016Q2/locationTopCarrierOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-494967b931c6eded4315086014aae5fc)

Load diff

Large diffs are not rendered by default.

**BIN**[...pringMVC/springmvc/src/main/resources/static/2016Q2/locations/aggOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0b62233923569f111de42f3b35b649de)

Binary file not shown.

**0**[...lProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/locations/aggOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-20c89bcf6b79cb2e4472deb8f2ca6df2)

No changes.

**64,905** [...jectSpringMVC/springmvc/src/main/resources/static/2016Q2/locations/aggOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-93704e172e4d81db69e4c7454e8eaf1e)

Load diff

Large diffs are not rendered by default.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q2/topkDestinations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-410818d51f91ebe7278dd31bae6e57ee)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkDestinations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-007676b02ceaae0a93cbeadc6a99490d)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkDestinations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-00c8058ccba527f72d95f80d21900773)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "destination": "LAX", |
|  |  | + "count": 293253 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SFO", |
|  |  | + "count": 246171 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "BOS", |
|  |  | + "count": 227214 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ORD", |
|  |  | + "count": 225458 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ATL", |
|  |  | + "count": 210613 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SEA", |
|  |  | + "count": 208894 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DEN", |
|  |  | + "count": 201912 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MCO", |
|  |  | + "count": 197300 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LGA", |
|  |  | + "count": 191339 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LAS", |
|  |  | + "count": 188621 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "EWR", |
|  |  | + "count": 173385 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DCA", |
|  |  | + "count": 170624 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHX", |
|  |  | + "count": 170601 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DFW", |
|  |  | + "count": 167719 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MSP", |
|  |  | + "count": 165391 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SAN", |
|  |  | + "count": 158908 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHL", |
|  |  | + "count": 149628 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DTW", |
|  |  | + "count": 141963 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PDX", |
|  |  | + "count": 139156 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "TPA", |
|  |  | + "count": 131372 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "IAH", |
|  |  | + "count": 130371 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "FLL", |
|  |  | + "count": 127119 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "AUS", |
|  |  | + "count": 124851 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "JFK", |
|  |  | + "count": 120901 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "BWI", |
|  |  | + "count": 119754 |
|  |  | + } |
|  |  | +] |

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkLocations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-5b02b0f7f22d95b5e6c1a061f7636570)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkLocations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d14c5bbfa427b8d567f16e640d6ac6a3)

No changes.

**402** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkLocations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-e4b837591e1c7c7f2f0d9de2f43c1030)

Load diff

Large diffs are not rendered by default.

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOpCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-506a5bb2f31d4591878dc1c1be517803)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOpCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-086b5afa0f22d4d61a31445b5d399d9c)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOpCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-01b12f08bab4f722c1724f1cfc3348a2)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "opCarrier": "99", |
|  |  | + "count": 3063474 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "WN", |
|  |  | + "count": 1622763 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "DL", |
|  |  | + "count": 1487258 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AA", |
|  |  | + "count": 1098729 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "UA", |
|  |  | + "count": 671740 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "B6", |
|  |  | + "count": 237193 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OO", |
|  |  | + "count": 205546 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AS", |
|  |  | + "count": 175349 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "EV", |
|  |  | + "count": 129027 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "NK", |
|  |  | + "count": 111657 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YX", |
|  |  | + "count": 102860 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "9E", |
|  |  | + "count": 87124 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OH", |
|  |  | + "count": 84117 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "MQ", |
|  |  | + "count": 80688 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YV", |
|  |  | + "count": 73271 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "HA", |
|  |  | + "count": 66777 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "VX", |
|  |  | + "count": 57154 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "F9", |
|  |  | + "count": 55568 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "ZW", |
|  |  | + "count": 44730 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "S5", |
|  |  | + "count": 43782 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "CP", |
|  |  | + "count": 42984 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "QX", |
|  |  | + "count": 41896 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G7", |
|  |  | + "count": 40057 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G4", |
|  |  | + "count": 25039 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "SY", |
|  |  | + "count": 13839 |
|  |  | + } |
|  |  | +] |

**BIN**[...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOrigins/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-dd23009bfa58621a2ba8fdbec96bf5d2)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOrigins/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0f8401ba53e06d5c93123f9e1909adf0)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkOrigins/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-b17d71d490265b1c4690f7ddaa500363)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "origin": "LAX", |
|  |  | + "count": 294943 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SFO", |
|  |  | + "count": 243867 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ORD", |
|  |  | + "count": 226094 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "BOS", |
|  |  | + "count": 224130 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ATL", |
|  |  | + "count": 212234 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SEA", |
|  |  | + "count": 206283 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DEN", |
|  |  | + "count": 202081 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MCO", |
|  |  | + "count": 200594 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LAS", |
|  |  | + "count": 194900 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LGA", |
|  |  | + "count": 190734 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHX", |
|  |  | + "count": 175446 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DFW", |
|  |  | + "count": 173275 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DCA", |
|  |  | + "count": 172148 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "EWR", |
|  |  | + "count": 170769 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MSP", |
|  |  | + "count": 162059 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SAN", |
|  |  | + "count": 159006 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHL", |
|  |  | + "count": 149971 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DTW", |
|  |  | + "count": 139808 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PDX", |
|  |  | + "count": 137339 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "TPA", |
|  |  | + "count": 134506 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "IAH", |
|  |  | + "count": 132733 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "FLL", |
|  |  | + "count": 130682 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "AUS", |
|  |  | + "count": 126293 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "JFK", |
|  |  | + "count": 120119 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "BWI", |
|  |  | + "count": 118836 |
|  |  | + } |
|  |  | +] |

**BIN**[...jectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-faab87c5b1ff1b5e598816edd62ddf53)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2139065a0b5c3cd31e557557d8a6d12a)

No changes.

**90** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q2/topkTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-24ee642006fb1aab87ca14529a250194)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,90 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "count": 2552481 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "count": 2543452 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "count": 1676340 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "count": 1622763 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "count": 365337 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "count": 348334 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "count": 238964 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "count": 111657 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "count": 71024 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "count": 57145 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "count": 55568 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "count": 25039 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "count": 13839 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "count": 132 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "count": 49 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "count": 37 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "count": 28 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "count": 23 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "count": 10 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EI\"", |
|  |  | + "count": 4 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "count": 2 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BA\"", |
|  |  | + "count": 1 |
|  |  | + } |
|  |  | +] |

**BIN**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/.filtermerge.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-817bccdb86a844b0dc3dccf60c4612d2)

Binary file not shown.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q3/TopRevTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-dfb69f970de139e5b5629731e2a5de93)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/TopRevTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-649c9f52e2a6e007bb1592c27301f80a)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/TopRevTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-d94502e8367c5f1ad89e71ce80adc8a0)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "sum": 705523188 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "sum": 700726615 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "sum": 504517199 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "sum": 355436578 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "sum": 125094362 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "sum": 93141669 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "sum": 49118781 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "sum": 24474257 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "sum": 17372769 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "sum": 12747237 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "sum": 12485539 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "sum": 2903374 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "sum": 1991642 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "sum": 14391 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "sum": 9426 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "sum": 4639 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "sum": 4505 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "sum": 4279 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "sum": 4126 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VA\"", |
|  |  | + "sum": 3193 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "sum": 2626 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"KS\"", |
|  |  | + "sum": 2373 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EI\"", |
|  |  | + "sum": 2199 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EK\"", |
|  |  | + "sum": 1686 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"OS\"", |
|  |  | + "sum": 1200 |
|  |  | + } |
|  |  | +] |

**BIN**[...MVC/springmvc/src/main/resources/static/2016Q3/locationTopCarrierOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-be406c395271b2318ece16519f894079)

Binary file not shown.

**0**[...ectSpringMVC/springmvc/src/main/resources/static/2016Q3/locationTopCarrierOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-076759af1cb62171bee75d64a0a2b59c)

No changes.

**67,196** [...pringMVC/springmvc/src/main/resources/static/2016Q3/locationTopCarrierOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-3766f9d523955c4ed205f86766b0e55e)

Load diff

Large diffs are not rendered by default.

**BIN**[...pringMVC/springmvc/src/main/resources/static/2016Q3/locations/aggOutput/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-bde9c6e5ea8c8c30a76cf1a533aeb27f)

Binary file not shown.

**0**[...lProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/locations/aggOutput/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-36aae1ad61baaa534b2219782cb54e92)

No changes.

**67,195** [...jectSpringMVC/springmvc/src/main/resources/static/2016Q3/locations/aggOutput/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-6f59ed1ceab04b131eff80e2204b4edb)

Load diff

Large diffs are not rendered by default.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q3/topkDestinations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-950edaa01f8761245a577f19647216a2)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkDestinations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-0a7ecd33ec4c5d91efc4ff76043c73cc)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkDestinations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-56c30cbf1f5dbd236448362e4fbc2bb9)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "destination": "LAX", |
|  |  | + "count": 298060 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SFO", |
|  |  | + "count": 245491 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SEA", |
|  |  | + "count": 221644 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "BOS", |
|  |  | + "count": 220197 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ORD", |
|  |  | + "count": 219247 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DEN", |
|  |  | + "count": 210062 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "ATL", |
|  |  | + "count": 201834 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LAS", |
|  |  | + "count": 183453 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "LGA", |
|  |  | + "count": 182576 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MCO", |
|  |  | + "count": 176494 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DFW", |
|  |  | + "count": 166289 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "MSP", |
|  |  | + "count": 166182 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "EWR", |
|  |  | + "count": 165815 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SAN", |
|  |  | + "count": 164930 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DCA", |
|  |  | + "count": 158209 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHX", |
|  |  | + "count": 155430 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PDX", |
|  |  | + "count": 147969 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "PHL", |
|  |  | + "count": 147103 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "DTW", |
|  |  | + "count": 137251 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "IAH", |
|  |  | + "count": 127090 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "SLC", |
|  |  | + "count": 121543 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "JFK", |
|  |  | + "count": 121268 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "AUS", |
|  |  | + "count": 120628 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "TPA", |
|  |  | + "count": 119266 |
|  |  | + }, |
|  |  | + { |
|  |  | + "destination": "BWI", |
|  |  | + "count": 115959 |
|  |  | + } |
|  |  | +] |

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkLocations/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-32bdfd74b214914cd0ff895b6d832551)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkLocations/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-3d9ce124609742e1e1f766d0b6d04685)

No changes.

**402** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkLocations/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-8fede5f0870b12bb8cd9ee7be61fc1ab)

Load diff

Large diffs are not rendered by default.

**BIN**[...ojectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOpCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-cefecbfd1cf0e19315b4d7dd223e769c)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOpCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-9c3bf4a93e42103ca736e09e2ad931d2)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOpCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-24232582f77f94b54cb595dcfd50026e)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "opCarrier": "99", |
|  |  | + "count": 3046125 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "WN", |
|  |  | + "count": 1572408 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "DL", |
|  |  | + "count": 1421225 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AA", |
|  |  | + "count": 1032786 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "UA", |
|  |  | + "count": 695271 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "B6", |
|  |  | + "count": 224650 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OO", |
|  |  | + "count": 204349 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "AS", |
|  |  | + "count": 172913 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "EV", |
|  |  | + "count": 127276 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "NK", |
|  |  | + "count": 114853 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YX", |
|  |  | + "count": 96381 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "MQ", |
|  |  | + "count": 89811 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "F9", |
|  |  | + "count": 88241 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "9E", |
|  |  | + "count": 86255 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "OH", |
|  |  | + "count": 85868 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "HA", |
|  |  | + "count": 66760 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "YV", |
|  |  | + "count": 66146 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "VX", |
|  |  | + "count": 58244 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "CP", |
|  |  | + "count": 42275 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "QX", |
|  |  | + "count": 42009 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "ZW", |
|  |  | + "count": 39822 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G7", |
|  |  | + "count": 35634 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "S5", |
|  |  | + "count": 34871 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "G4", |
|  |  | + "count": 23970 |
|  |  | + }, |
|  |  | + { |
|  |  | + "opCarrier": "SY", |
|  |  | + "count": 14521 |
|  |  | + } |
|  |  | +] |

**BIN**[...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOrigins/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-e0d63b94cac1847a7bb68f9bafa5fbe9)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOrigins/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-5372d85cacd53ab76ec3304823023142)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkOrigins/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-29a2e2270cf94bbcd7c745b5ec5b9f1d)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "origin": "LAX", |
|  |  | + "count": 298073 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SFO", |
|  |  | + "count": 243144 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ORD", |
|  |  | + "count": 224816 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SEA", |
|  |  | + "count": 222706 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "BOS", |
|  |  | + "count": 222448 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DEN", |
|  |  | + "count": 211348 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "ATL", |
|  |  | + "count": 201309 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LAS", |
|  |  | + "count": 187605 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "LGA", |
|  |  | + "count": 185921 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MCO", |
|  |  | + "count": 177352 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DFW", |
|  |  | + "count": 169121 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "MSP", |
|  |  | + "count": 166447 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "EWR", |
|  |  | + "count": 165845 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SAN", |
|  |  | + "count": 163230 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DCA", |
|  |  | + "count": 160564 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHX", |
|  |  | + "count": 154396 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PHL", |
|  |  | + "count": 147757 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "PDX", |
|  |  | + "count": 147333 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "DTW", |
|  |  | + "count": 137693 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "IAH", |
|  |  | + "count": 126957 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "SLC", |
|  |  | + "count": 122785 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "JFK", |
|  |  | + "count": 120819 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "TPA", |
|  |  | + "count": 119023 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "AUS", |
|  |  | + "count": 118229 |
|  |  | + }, |
|  |  | + { |
|  |  | + "origin": "BWI", |
|  |  | + "count": 116867 |
|  |  | + } |
|  |  | +] |

**BIN**[...jectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-685f881ce45ba2fa7ac4a83a3452b960)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-352ae0b18eb1e23ca866db13c42ee2a3)

No changes.

**102** [FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q3/topkTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-a867d148cac208991f3ec9365ad692fa)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "count": 2462043 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "count": 2452651 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "count": 1676601 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "count": 1572408 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "count": 393255 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "count": 347808 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "count": 226057 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "count": 114853 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "count": 88241 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "count": 71100 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "count": 58231 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "count": 23970 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "count": 14521 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "count": 93 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "count": 88 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "count": 43 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "count": 28 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "count": 25 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "count": 19 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"KS\"", |
|  |  | + "count": 13 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"MW\"", |
|  |  | + "count": 5 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EY\"", |
|  |  | + "count": 4 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VA\"", |
|  |  | + "count": 3 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"OO\"", |
|  |  | + "count": 2 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"KE\"", |
|  |  | + "count": 1 |
|  |  | + } |
|  |  | +] |

**BIN**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q4/.filtermerge.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-2a71a22e7bf4ab0a689930770e1c2383)

Binary file not shown.

**BIN**[...ctSpringMVC/springmvc/src/main/resources/static/2016Q4/TopRevTktCarrier/.part-r-00000.crc](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-630ddd85dfc4d3f1923ed91e5becf292)

Binary file not shown.

**0**[FinalProjectSpringMVC/springmvc/src/main/resources/static/2016Q4/TopRevTktCarrier/\_SUCCESS](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-ded4b1c6abb22854cbab9ccb526997d2)

No changes.

**102** [...ProjectSpringMVC/springmvc/src/main/resources/static/2016Q4/TopRevTktCarrier/part-r-00000](https://github.com/saravanvadivel/ADBMSFinalProject/commit/25bf3eb0ade3182a9cdaa0096a81db15c4b2c436#diff-318b237dbaa61632ee59382d9150fcff)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -0,0 +1,102 @@ |
|  |  | +[ |
|  |  | + { |
|  |  | + "tktCarrier": "\"DL\"", |
|  |  | + "sum": 701956820 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AA\"", |
|  |  | + "sum": 687397573 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"UA\"", |
|  |  | + "sum": 507630800 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WN\"", |
|  |  | + "sum": 352073122 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"99\"", |
|  |  | + "sum": 106276191 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"AS\"", |
|  |  | + "sum": 88387484 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"B6\"", |
|  |  | + "sum": 53936301 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"HA\"", |
|  |  | + "sum": 26793827 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VX\"", |
|  |  | + "sum": 18013811 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NK\"", |
|  |  | + "sum": 14171241 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"F9\"", |
|  |  | + "sum": 6823172 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SY\"", |
|  |  | + "sum": 3627323 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"G4\"", |
|  |  | + "sum": 2524090 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"WP\"", |
|  |  | + "sum": 8285 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"7H\"", |
|  |  | + "sum": 6212 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"BB\"", |
|  |  | + "sum": 5637 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"SN\"", |
|  |  | + "sum": 5437 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"9K\"", |
|  |  | + "sum": 3498 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"ZK\"", |
|  |  | + "sum": 2300 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"3M\"", |
|  |  | + "sum": 1919 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"LH\"", |
|  |  | + "sum": 1815 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"EI\"", |
|  |  | + "sum": 1349 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"LY\"", |
|  |  | + "sum": 1342 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"NZ\"", |
|  |  | + "sum": 1323 |
|  |  | + }, |
|  |  | + { |
|  |  | + "tktCarrier": "\"VA\"", |
|  |  | + "sum": 1263 |
|  |  | + } |
|  |  | +] |