|  |  |
| --- | --- |
|  | |
| import java.io.IOException; | |
|  | import org.apache.hadoop.conf.Configuration; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.io.Text; |
|  | import org.apache.hadoop.io.LongWritable; |
|  | import org.apache.hadoop.mapreduce.Job; |
|  | import org.apache.hadoop.mapreduce.Mapper; |
|  | import org.apache.hadoop.mapreduce.Reducer; |
|  | import org.apache.hadoop.mapreduce.lib.input.MultipleInputs; |
|  | import org.apache.hadoop.mapreduce.lib.input.TextInputFormat; |
|  | import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; |
|  |  |
|  | public class SalesJoin { |
|  |  |
|  | public static class NovMapper extends |
|  | Mapper<LongWritable, Text, Text, Text> { |
|  | public void map(LongWritable key, Text value, Context context) |
|  | throws IOException, InterruptedException { |
|  | String record1 = value.toString(); |
|  | String[] parts1 = record1.split(";"); |
|  | context.write(new Text(parts1[5]), new Text("nov\t" + parts1[8])); |
|  | } |
|  | } |
|  |  |
|  | public static class DecMapper extends |
|  | Mapper<LongWritable, Text, Text, Text> { |
|  | public void map(LongWritable key, Text value, Context context) |
|  | throws IOException, InterruptedException { |
|  | String record2 = value.toString(); |
|  | String[] parts2 = record2.split(";"); |
|  | context.write(new Text(parts2[5]), new Text("dec\t" + parts2[8])); |
|  | } |
|  | } |
|  |  |
|  | public static class JanMapper extends |
|  | Mapper<LongWritable, Text, Text, Text> { |
|  | public void map(LongWritable key, Text value, Context context) |
|  | throws IOException, InterruptedException { |
|  | String record3 = value.toString(); |
|  | String[] parts3 = record3.split(";"); |
|  | context.write(new Text(parts3[5]), new Text("jan\t" + parts3[8])); |
|  | } |
|  | } |
|  |  |
|  | public static class FebMapper extends |
|  | Mapper<LongWritable, Text, Text, Text> { |
|  | public void map(LongWritable key, Text value, Context context) |
|  | throws IOException, InterruptedException { |
|  | String record4 = value.toString(); |
|  | String[] parts4 = record4.split(";"); |
|  | context.write(new Text(parts4[5]), new Text("feb\t" + parts4[8])); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | public static class SalesJoinReducer extends |
|  | Reducer<Text, Text, Text, Text> { |
|  | public void reduce(Text key, Iterable<Text> values, Context context) |
|  | throws IOException, InterruptedException { |
|  | long nov\_total = 0; |
|  | long dec\_total = 0; |
|  | long jan\_total = 0; |
|  | long feb\_total = 0; |
|  | double growth1 = 0.00; |
|  | double growth2 = 0.00; |
|  | double growth3 = 0.00; |
|  | double avg\_growth = 0.00; |
|  |  |
|  | for (Text t : values) { |
|  | String parts[] = t.toString().split("\t"); |
|  | if (parts[0].equals("nov")) { |
|  | nov\_total += Long.parseLong(parts[1]); |
|  | } |
|  | if (parts[0].equals("dec")) { |
|  | dec\_total += Long.parseLong(parts[1]); |
|  | } |
|  | if (parts[0].equals("jan")) { |
|  | jan\_total += Long.parseLong(parts[1]); |
|  | } |
|  | if (parts[0].equals("feb")) { |
|  | feb\_total += Long.parseLong(parts[1]); |
|  | } |
|  | } |
|  | if (nov\_total!=0) |
|  | { |
|  | growth1 = (((double)dec\_total) - ((double)nov\_total))/((double)nov\_total)\*100; |
|  | } |
|  | else |
|  | { |
|  | growth1 = ((double)dec\_total) \*100; |
|  | } |
|  |  |
|  | if (dec\_total!=0) |
|  | { |
|  | growth2 = (((double)jan\_total) - ((double)dec\_total))/((double)dec\_total)\*100; |
|  | } |
|  | else |
|  | { |
|  | growth2 = ((double)jan\_total) \*100; |
|  | } |
|  |  |
|  | if (jan\_total!=0) |
|  | { |
|  | growth3 = (((double)feb\_total) - ((double)jan\_total))/((double)jan\_total)\*100; |
|  | } |
|  | else |
|  | { |
|  | growth3 = ((double)feb\_total) \*100; |
|  | } |
|  |  |
|  |  |
|  | avg\_growth = (growth1+growth2+growth3)/3; |
|  |  |
|  | String str1 = String.format("%f", growth1); |
|  | String str2 = String.format("%f", growth2); |
|  | String str3 = String.format("%f", growth3); |
|  | String str4 = String.format("%f", avg\_growth); |
|  |  |
|  | String str = str1 + ',' + str2 + ',' + str3 + ',' + str4; |
|  |  |
|  | context.write(key, new Text(str)); |
|  | } |
|  | } |
|  |  |
|  | public static void main(String[] args) throws Exception { |
|  |  |
|  | Configuration conf = new Configuration(); |
|  | conf.set("mapreduce.output.textoutputformat.separator", ","); |
|  | Job job = Job.getInstance(conf); |
|  | job.setJarByClass(SalesJoin.class); |
|  | job.setJobName("Sales Join"); |
|  | job.setReducerClass(SalesJoinReducer.class); |
|  | job.setOutputKeyClass(Text.class); |
|  | job.setOutputValueClass(Text.class); |
|  |  |
|  | MultipleInputs.addInputPath(job, new Path(args[0]),TextInputFormat.class, NovMapper.class); |
|  | MultipleInputs.addInputPath(job, new Path(args[1]),TextInputFormat.class, DecMapper.class); |
|  | MultipleInputs.addInputPath(job, new Path(args[2]),TextInputFormat.class, JanMapper.class); |
|  | MultipleInputs.addInputPath(job, new Path(args[3]),TextInputFormat.class, FebMapper.class); |
|  |  |
|  | Path outputPath = new Path(args[4]); |
|  | FileOutputFormat.setOutputPath(job, outputPath); |
|  | //outputPath.getFileSystem(conf).delete(outputPath); |
|  |  |
|  | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  | } |
|  | } |