|  |  |
| --- | --- |
|  | |
| import java.io.IOException; | |
|  | import java.text.ParseException; |
|  | import java.text.SimpleDateFormat; |
|  | import java.util.Date; |
|  | import java.util.TreeMap; |
|  |  |
|  | import org.apache.hadoop.conf.Configuration; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.io.LongWritable; |
|  | import org.apache.hadoop.io.Text; |
|  | 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.FileInputFormat; |
|  | import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; |
|  |  |
|  |  |
|  | //use retail data D11,D12,D01 and D02 |
|  |  |
|  | public class DayWiseSales { |
|  |  |
|  | // Mapper Class |
|  |  |
|  | public static class DayWiseSalesMapperClass extends Mapper<LongWritable,Text,Text,LongWritable> |
|  | { |
|  | public void map(LongWritable key, Text value, Context context) |
|  | { |
|  | try{ |
|  | String[] str = value.toString().split(";"); |
|  | String trndate = str[0]; |
|  | long sales = Long.parseLong(str[8]); |
|  | String trnDay = toDay(trndate); |
|  | context.write(new Text(trnDay), new LongWritable(sales)); |
|  | } |
|  | catch(Exception e) |
|  | { |
|  | System.out.println(e.getMessage()); |
|  | } |
|  | } |
|  |  |
|  | private String toDay(String date) { |
|  |  |
|  | SimpleDateFormat format = new SimpleDateFormat( |
|  | "yyyy-MM-dd HH:mm:ss"); |
|  | SimpleDateFormat newDateFormat = new SimpleDateFormat("EEEE"); |
|  |  |
|  | Date dateFrm = null; |
|  | try { |
|  | dateFrm = format.parse(date); |
|  | } |
|  | catch (ParseException e) |
|  | { |
|  |  |
|  | e.printStackTrace(); |
|  | } |
|  | return newDateFormat.format(dateFrm); |
|  | } |
|  | } |
|  |  |
|  | //Reducer class |
|  |  |
|  | public static class DayWiseSalesReducerClass extends Reducer<Text,LongWritable,Text,Text> |
|  | { |
|  | private TreeMap<Long, Text> repToRecordMap = new TreeMap<Long, Text>(); |
|  | long grand\_total = 0; |
|  |  |
|  |  |
|  | public void reduce(Text key, Iterable <LongWritable> values, Context context) throws IOException, InterruptedException |
|  | { |
|  | long totalsales = 0; |
|  |  |
|  | for (LongWritable val : values) |
|  | { |
|  | totalsales = totalsales + val.get(); |
|  | grand\_total = grand\_total + val.get(); |
|  | } |
|  |  |
|  | String myValue = key.toString(); |
|  | String mytotal = String.format("%d", totalsales); |
|  |  |
|  | myValue = myValue + ',' + mytotal ; |
|  |  |
|  | repToRecordMap.put(new Long(totalsales), new Text(myValue)); |
|  | } |
|  |  |
|  | protected void cleanup(Context context) throws IOException, |
|  | InterruptedException |
|  | { |
|  | String myKey = ""; |
|  | String myText = ""; |
|  | double myPercent = 0.00; |
|  | long totalsales = 0; |
|  |  |
|  | for (Text t : repToRecordMap.values()) |
|  | { |
|  | String[] token = t.toString().split(","); |
|  | myKey = token[0]; |
|  | totalsales = Long.parseLong(token[1]); |
|  | myPercent = ((double)totalsales\*100)/(double)grand\_total; |
|  | String SalesPercent = String.format("%f", myPercent); |
|  | myText = token[1] + ',' + SalesPercent; |
|  | context.write(new Text(myKey), new Text(myText)); |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  | //Main class |
|  |  |
|  | public static void main(String[] args) throws Exception { |
|  |  |
|  | Configuration conf = new Configuration(); |
|  | Job job = Job.getInstance(conf, "Day wise sales report"); |
|  | job.setJarByClass(DayWiseSales.class); |
|  | job.setMapperClass(DayWiseSalesMapperClass.class); |
|  | job.setReducerClass(DayWiseSalesReducerClass.class); |
|  | job.setMapOutputKeyClass(Text.class); |
|  | job.setMapOutputValueClass(LongWritable.class); |
|  | job.setOutputKeyClass(Text.class); |
|  | job.setOutputValueClass(Text.class); |
|  | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  | } |
|  | } |