POM.xml

<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>com.training.section3</groupId>

<artifactId>section3</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-sdks-java-core</artifactId>

<version>2.19.0</version>

</dependency>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-runners-direct-java</artifactId>

<version>2.19.0</version>

</dependency>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-sdks-java-io-parquet</artifactId>

<version>2.19.0</version>

</dependency>

</dependencies>

</project>

FilterExample.java

package section3;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.DoFn;

import org.apache.beam.sdk.transforms.Filter;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.transforms.ParDo;

import org.apache.beam.sdk.transforms.SerializableFunction;

import org.apache.beam.sdk.transforms.SerializableFunctions;

import org.apache.beam.sdk.values.PCollection;

import org.apache.beam.sdk.values.TypeDescriptors;

class MyFilter implements SerializableFunction<String, Boolean>{

@Override

public Boolean apply(String input) {

// TODO Auto-generated method stub

return input.contains("Los Angeles");

}

}

public class FilterExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pCustList = p.apply(TextIO.read().from("C:\\Beam\\customer\_pardo.csv"));

// Using Filter

PCollection<String> pOutput = pCustList.apply(Filter.by(new MyFilter()));

pOutput.apply(TextIO.write().to("C:\\Beam\\customer\_Filter\_output.csv").withHeader("Id,Name,Last Name,City").withNumShards(1).withSuffix(".csv"));

p.run();

}

}

MapElementsExample.java

package section3;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.values.PCollection;

import org.apache.beam.sdk.values.TypeDescriptors;

public class MapElementsExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pCustList= p.apply(TextIO.read().from("C:\\Beam\\customer.csv"));

//Using TypeDescriptors

PCollection<String> pOutput=pCustList.apply(MapElements.into(TypeDescriptors.strings()).via((String obj) -> obj.toUpperCase()));

pOutput.apply(TextIO.write().to("C:\\Beam\\cust\_output.csv").withNumShards(1).withSuffix(".csv"));

p.run();

}

}

MapElementsSimpleFunction.java

package section3;

import java.util.List;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.transforms.SimpleFunction;

import org.apache.beam.sdk.values.PCollection;

class User extends SimpleFunction<String, String>{

@Override

public String apply(String input) {

// TODO Auto-generated method stub

String arr[] = input.split(",");

String SId= arr[0];

String UId= arr[1];

String Uname= arr[2];

String VId= arr[3];

String duration= arr[4];

String startTime= arr[5];

String sex= arr[6];

String ouput="";

if(sex.equals("1")) {

ouput=SId+","+UId+","+","+Uname+","+VId+","+duration+","+startTime+","+"M";

}else if(sex.equals("2")) {

ouput=SId+","+UId+","+","+Uname+","+VId+","+duration+","+startTime+","+"F";

}

else {

ouput=input;

}

return ouput;

}

}

public class MapElementsSimpleFunction {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pUserList= p.apply(TextIO.read().from("C:\\Beam\\user.csv"));

//Using Simple Function

PCollection<String> pOutput= pUserList.apply(MapElements.via(new User()));

pOutput.apply(TextIO.write().to("C:\\Beam\\user\_output.csv").withNumShards(1).withSuffix(".csv"));

p.run();

}

}

ParDoExample.java

package section3;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.DoFn;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.transforms.ParDo;

import org.apache.beam.sdk.values.PCollection;

import org.apache.beam.sdk.values.TypeDescriptors;

class CustFilter extends DoFn<String, String> {

@ProcessElement

public void processElement(ProcessContext c) {

String line = c.element();

String arr[] = line.split(",");

if(arr[3].equals("Los Angeles")) {

c.output(line);

}

}

}

public class ParDoExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pCustList = p.apply(TextIO.read().from("C:\\Beam\\customer\_pardo.csv"));

// Using ParDO

PCollection<String> pOutput = pCustList.apply(ParDo.of(new CustFilter()));

pOutput.apply(TextIO.write().to("C:\\Beam\\customer\_pardo\_output.csv").withHeader("Id,Name,Last Name,City").withNumShards(1).withSuffix(".csv"));

p.run();

}

}

PartitionExample.java

package section3;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.DoFn;

import org.apache.beam.sdk.transforms.Filter;

import org.apache.beam.sdk.transforms.Flatten;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.transforms.ParDo;

import org.apache.beam.sdk.transforms.Partition;

import org.apache.beam.sdk.transforms.Partition.PartitionFn;

import org.apache.beam.sdk.transforms.SerializableFunction;

import org.apache.beam.sdk.transforms.SerializableFunctions;

import org.apache.beam.sdk.transforms.windowing.PartitioningWindowFn;

import org.apache.beam.sdk.values.PCollection;

import org.apache.beam.sdk.values.PCollectionList;

import org.apache.beam.sdk.values.TypeDescriptors;

class MyCityPartition implements PartitionFn<String>{

@Override

public int partitionFor(String elem, int numPartitions) {

// TODO Auto-generated method stub

String arr[] = elem.split(",");

if(arr[3].equals("Los Angeles")) {

return 0;

}

else if(arr[3].equals("Phoenix")) {

return 1;

}

else {

return 2;

}

}

}

public class PartitionExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pCustList1 = p.apply(TextIO.read().from("C:\\Beam\\Partition.csv"));

PCollectionList<String> partition = pCustList1.apply(Partition.of(3, new MyCityPartition()));

PCollection<String> p0 = partition.get(0);

PCollection<String> p1 = partition.get(1);

PCollection<String> p2 = partition.get(2);

p0.apply(TextIO.write().to("C:\\Beam\\p0.csv").withNumShards(1).withSuffix(".csv"));

p1.apply(TextIO.write().to("C:\\Beam\\p1.csv").withNumShards(1).withSuffix(".csv"));

p2.apply(TextIO.write().to("C:\\Beam\\p2.csv").withNumShards(1).withSuffix(".csv"));

p.run();

}

}

FlattenExample.java

package section3;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.transforms.DoFn;

import org.apache.beam.sdk.transforms.Filter;

import org.apache.beam.sdk.transforms.Flatten;

import org.apache.beam.sdk.transforms.MapElements;

import org.apache.beam.sdk.transforms.ParDo;

import org.apache.beam.sdk.transforms.SerializableFunction;

import org.apache.beam.sdk.transforms.SerializableFunctions;

import org.apache.beam.sdk.values.PCollection;

import org.apache.beam.sdk.values.PCollectionList;

import org.apache.beam.sdk.values.TypeDescriptors;

public class FlatternExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

PCollection<String> pCustList1 = p.apply(TextIO.read().from("C:\\Beam\\customer\_1.csv"));

PCollection<String> pCustList2 = p.apply(TextIO.read().from("C:\\Beam\\customer\_2.csv"));

PCollection<String> pCustList3 = p.apply(TextIO.read().from("C:\\Beam\\customer\_3.csv"));

PCollectionList<String> list=PCollectionList.of(pCustList1).and(pCustList2).and(pCustList3);

PCollection<String> merged=list.apply(Flatten.pCollections());

merged.apply(TextIO.write().to("C:\\Beam\\customer\_flatteren\_output.csv").withHeader("Id,Name,Last Name,City").withNumShards(1).withSuffix(".csv"));

p.run();

}

}