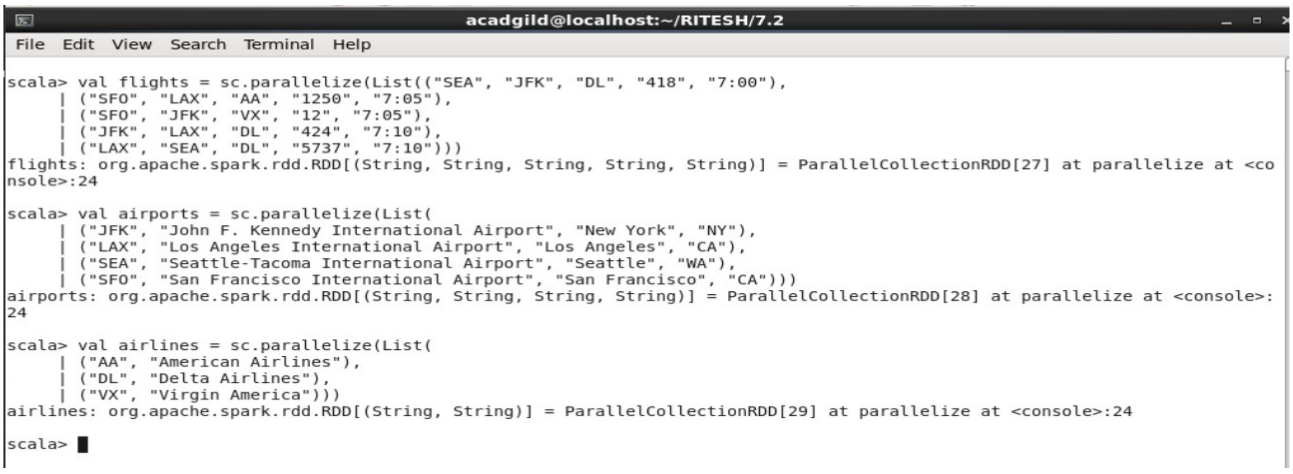
Bigdata Assignment 7.4

1) Join of two or more data sets is one of the most widely used operations you do with your data, but in distributed systems it can be a huge headache. In general, since your data are distributed among many nodes, they have to be shuffled before a join that causes significant network I/O and slow performance.
2) Fortunately, if you need to join a large table with relatively small tables you can avoid sending all data of the large table over the network. This type of join is called map-side join in Hadoop community. In other distributed systems, it is often called replicated or broadcast join.

```
// Fact table

val flights = sc.parallelize(List(
("SEA", "JFK", "DL", "418", "7:00"),
("SFO", "LAX", "AA", "1250", "7:05"),
("SFO", "JFK", "VX", "12", "7:05"),
("JFK", "LAX", "DL", "424", "7:10"),
("LAX", "SEA", "DL", "5737", "7:10")))


// Dimension table

val airports = sc.parallelize(List(
("JFK", "John F. Kennedy International Airport", "New York", "NY"),
("LAX", "Los Angeles International Airport", "Los Angeles", "CA"),
("SEA", "Seattle-Tacoma International Airport", "Seattle", "WA"),
("SFO", "San Francisco International Airport", "San Francisco", "CA")))


// Dimension table

val airlines = sc.parallelize(List(
("AA", "American Airlines"),
("DL", "Delta Airlines"),
("VX", "Virgin America")))
```



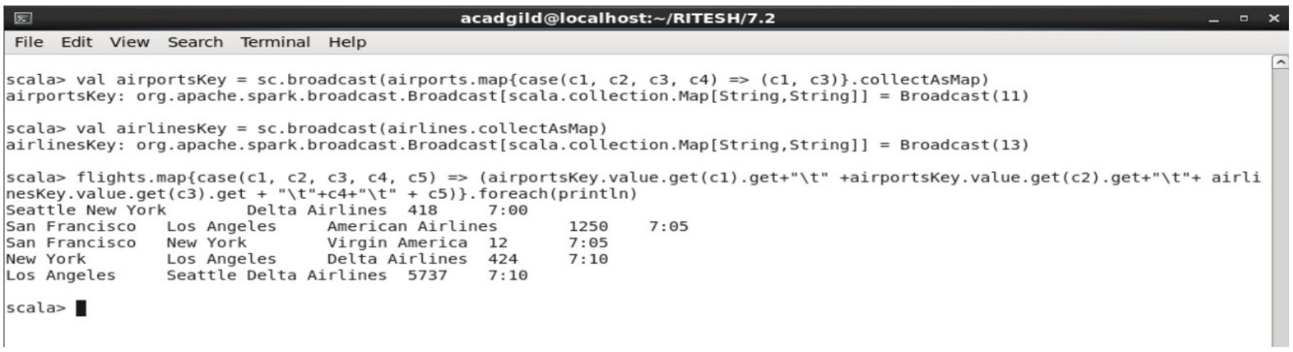
```
acadgild@localhost:~/RITESH/7.2
File  Edit  View  Search  Terminal  Help

scala> val flights = sc.parallelize(List(("SEA", "JFK", "DL", "418", "7:00"),
     | ("SFO", "LAX", "AA", "1250", "7:05"),
     | ("SFO", "JFK", "VX", "12", "7:05"),
     | ("JFK", "LAX", "DL", "424", "7:10"),
     | ("LAX", "SEA", "DL", "5737", "7:10")))
flights: org.apache.spark.rdd.RDD[(String, String, String, String, String)] = ParallelCollectionRDD[27] at parallelize at <co
nsole>:24

scala> val airports = sc.parallelize(List(
     | ("JFK", "John F. Kennedy International Airport", "New York", "NY"),
     | ("LAX", "Los Angeles International Airport", "Los Angeles", "CA"),
     | ("SEA", "Seattle-Tacoma International Airport", "Seattle", "WA"),
     | ("SFO", "San Francisco International Airport", "San Francisco", "CA")))
airports: org.apache.spark.rdd.RDD[(String, String, String, String)] = ParallelCollectionRDD[28] at parallelize at <console>:
24

scala> val airlines = sc.parallelize(List(
     | ("AA", "American Airlines"),
     | ("DL", "Delta Airlines"),
     | ("VX", "Virgin America")))
airlines: org.apache.spark.rdd.RDD[(String, String)] = ParallelCollectionRDD[29] at parallelize at <console>:24

scala> █
```

Solution -

//Selecting city and its short form with city as key using broadcast variable.
val airportsKey = sc.broadcast(airports.map{case(c1, c2, c3, c4) => (c1, c3)}.collectAsMap)

//Selecting airlines as key and its value is its full form using broadcast variable.
val airlinesKey = sc.broadcast(airlines.collectAsMap)

//Selecting airport ' shortform as key and its value and its required columns as given in the question.
flights.map{case(c1, c2, c3, c4, c5) => (airportsKey.value.get(c1).get+"\t" +airportsKey.value.get(c2).get+"\t"+ airlinesKey.value.get(c3).get + "\t"+c4+"\t" + c5)}.foreach(println)



```
acadgild@localhost:~/RITESH/7.2
File  Edit  View  Search  Terminal  Help

scala> val airportsKey = sc.broadcast(airports.map{case(c1, c2, c3, c4) => (c1, c3)}.collectAsMap)
airportsKey: org.apache.spark.broadcast.Broadcast[scala.collection.Map[String,String]] = Broadcast(11)

scala> val airlinesKey = sc.broadcast(airlines.collectAsMap)
airlinesKey: org.apache.spark.broadcast.Broadcast[scala.collection.Map[String,String]] = Broadcast(13)

scala> flights.map{case(c1, c2, c3, c4, c5) => (airportsKey.value.get(c1).get+"\t" +airportsKey.value.get(c2).get+"\t"+ airli
nesKey.value.get(c3).get + "\t"+c4+"\t" + c5)}.foreach(println)
Seattle New York        Delta Airlines  418     7:00
San Francisco   Los Angeles     American Airlines       1250    7:05
San Francisco   New York        Virgin America  12      7:05
New York        Los Angeles     Delta Airlines  424     7:10
Los Angeles     Seattle Delta Airlines  5737    7:10

scala> █
```