**Assignment 21.2**

**Problem Statement**

**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.**

**The fact table can be very large, while dimension tables are often quite small.**

**Let’s use the following sample data (one fact and two dimension tables):**

// 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")))

We need to join the fact and dimension tables to get the following result:

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

**Code**



