Assignment 21.2

Step1: Load datasets as given in assignment

* Load datasets flights, airports, airlines as given in assignment

Code is as below:

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

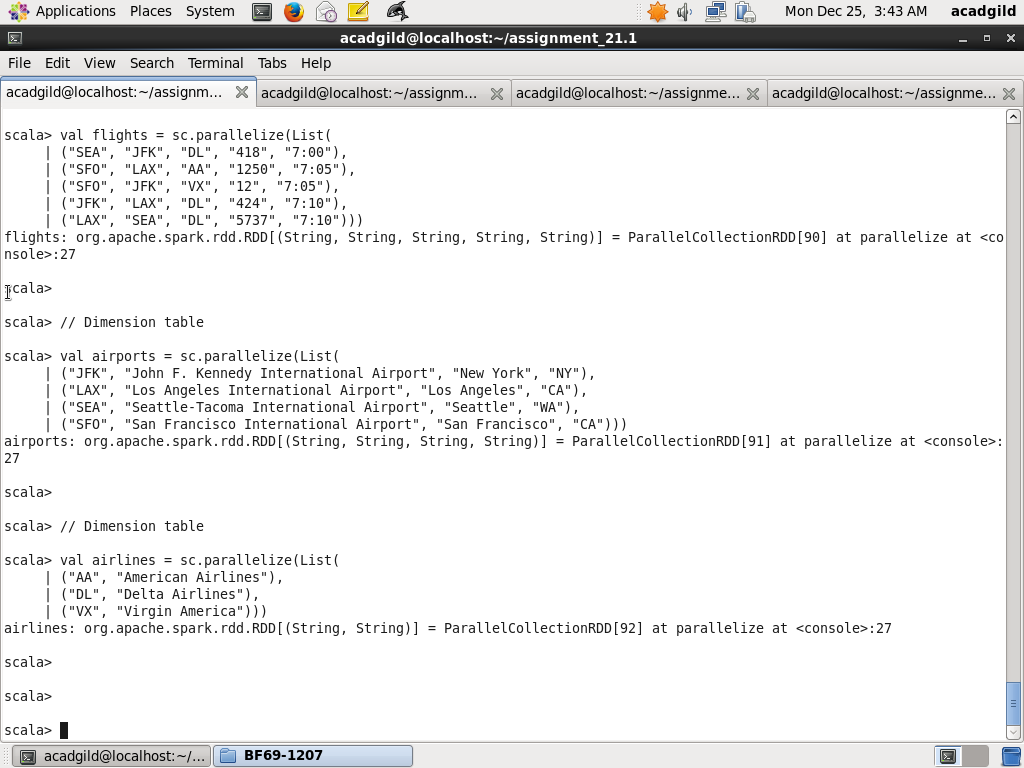
val airlines = sc.parallelize(List(

("AA", "American Airlines"),

("DL", "Delta Airlines"),

("VX", "Virgin America")))

Screenshot is as below:



Step2: Create map airportsMap from airports with key being airport code and value being airport city. Next create broadcast variable broadcastAirportsMap.

Create map linesMap from airlibnes with key being airlines code and value being airline name. Next create broadcast variable broadcastAirlinesMap.

// Create broadcast variable broadcastAirportsMap

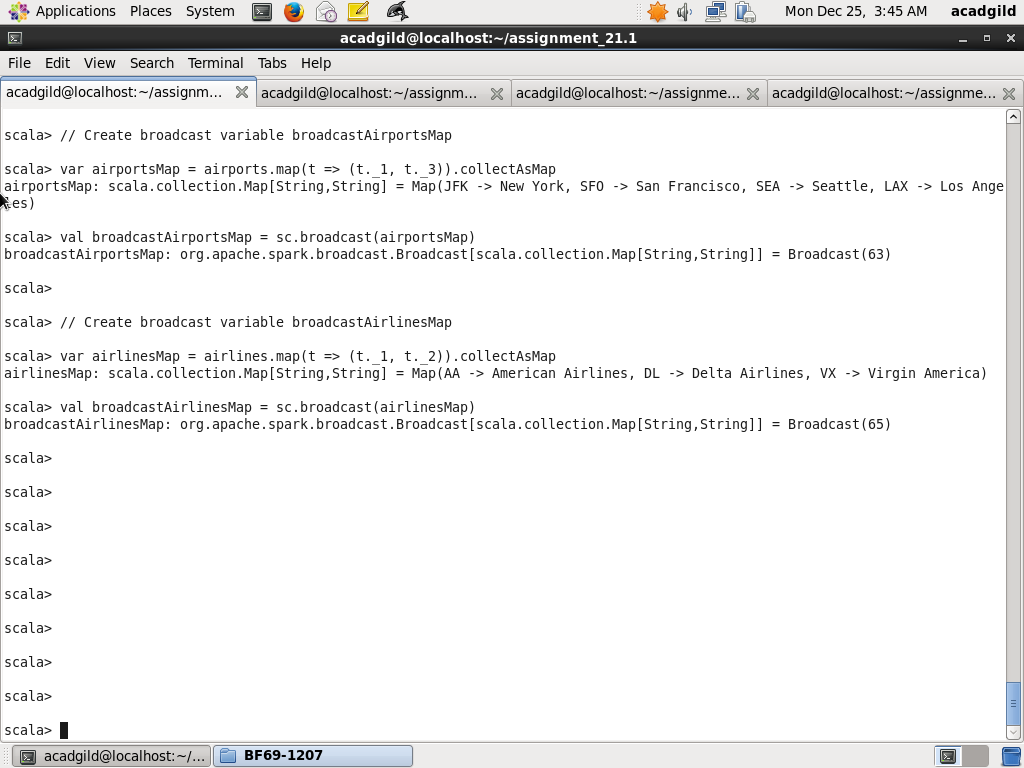
var airportsMap = airports.map(t => (t.\_1, t.\_3)).collectAsMap

val broadcastAirportsMap = sc.broadcast(airportsMap)

// Create broadcast variable broadcastAirlinesMap

var airlinesMap = airlines.map(t => (t.\_1, t.\_2)).collectAsMap

val broadcastAirlinesMap = sc.broadcast(airlinesMap)



Step3: Create dataframe

* Map filghts by populating airtport city from broadcastAirportsMap based on airport code and airlines name from broadcastAirlinesMap based on airlines code and create new RDD flightsDetailed
* Create dataframe flightsDetailedDF from flightsDetailed with fields source\_airport, dest\_airport, "airlines", price, departure\_time
* Show the records from dataframe flightsDetailedDF

// Create Dataframe flightsDetailedDF and display records

val flightsDetailed = flights.map(t=> ( broadcastAirportsMap.value.get(t.\_1), broadcastAirportsMap.value.get(t.\_2), broadcastAirlinesMap.value.get(t.\_3), t.\_4, t.\_5))

val flightsDetailedDF = flightsDetailed.toDF("source\_airport", "dest\_airport", "airlines", "price", "departure\_time")

flightsDetailedDF.show

Screenshot is as below:

