Session 21 Assignment 2

Import the flight details list

```
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")))
```

Convert the flight details for DataFrame

```
val flightsDF=flights.toDF("src","dest","airline","sno","duration")
flightsDF.show
```

```
scala> val flightsDF=flights.toDF("src","dest","airline","sno","deptime")
flightsDF: org.apache.spark.sql.DataFrame = [src: string, dest: string ... 3 more fields]

scala> flightsDF.show
+---+----+-------+----+-------+
|src|dest|airline| sno|deptime|
+---+----+-------+----+-------+
|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|
+---+----+-------+----+-------+
```

Convert the airport details list to DataFrame

```
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")))
```

val airportsDF=airports.toDF("code","descr","city","state")

```
scala> val airportsDF=airports.toDF("code","descr","city","state")
airportsDF: org.apache.spark.sql.DataFrame = [code: string, descr: string ... 2 more fields]

scala> airportsDF.show
+----+--------------------+-------------+-----+
|code|               descr|         city|state|
+----+--------------------+-------------+-----+
| JFK|John F. Kennedy I...|     New York|   NY|
| LAX|Los Angeles Inter...|  Los Angeles|   CA|
| SEA|Seattle-Tacoma In...|      Seattle|   WA|
| SFO|San Francisco Int...|San Francisco|   CA|
+----+--------------------+-------------+-----+
```

Convert the airlines details to DataFrame

val airlines = sc.parallelize(List(

("AA", "American Airlines"),

("DL", "Delta Airlines"),

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

val airlineDF=airlines.toDF("airline","airlinename")

```
scala> val airlineDF=airlines.toDF("airline","airlinename")
airlineDF: org.apache.spark.sql.DataFrame = [airline: string, airlinename: string]

scala> airlineDF.show
+-------+-----------------+
|airline|      airlinename|
+-------+-----------------+
|     AA|American Airlines|
|     DL|   Delta Airlines|
|     VX|   Virgin America|
+-------+-----------------+


scala>
```

Step 1: Join flights dataframe and airport dataframe to get the source city.

val flightsAirportsSRCDF=flightsDF.as("flights").join(airportsDF.as("airports"),$"flights.src"===$"airports.code").select($"flights.src",$"flights.dest",$"airports.city",$"flights.airline",$"flights.sno",$"flights.duration")

```
scala> val flightsAirportsSRCDF=flightsDF.as("flights").join(airportsDF.as("airports"),$"flights.src"===$"airports.code").select($"flights.src",$"flights.dest"
,$"airports.city",$"flights.airline",$"flights.sno",$"flights.duration")
flightsAirportsSRCDF: org.apache.spark.sql.DataFrame = [src: string, dest: string ... 4 more fields]

scala> flightsAirportsSRCDF.show
+---+----+-------------+-------+----+--------+
|src|dest|         city|airline| sno|duration|
+---+----+-------------+-------+----+--------+
|SEA| JFK|      Seattle|     DL| 418|    7:00|
|SFO| LAX|San Francisco|     AA|1250|    7:05|
|SFO| JFK|San Francisco|     VX|  12|    7:05|
|LAX| SEA|  Los Angeles|     DL|5737|    7:10|
|JFK| LAX|     New York|     DL| 424|    7:10|
+---+----+-------------+-------+----+--------+
```

Step 2: Again, perform the same join as above to get the destination city.

```
val flightsAirportsDESTDF=flightsDF.as("flights").join(airportsDF.as("airports"),$"flights.dest"===$"airports.code").select($"flights.src",$"flights.dest",$"airports.city")
```

```
scala> val flightsAirportsDESTDF=flightsDF.as("flights").join(airportsDF.as("airports"),$"flights.dest"===$"airports.code").select($"flights.src",$"flights.des
t",$"airports.city")
flightsAirportsDESTDF: org.apache.spark.sql.DataFrame = [src: string, dest: string ... 1 more field]

scala> flightsAirportsDESTDF.show
+---+----+-----------+
|src|dest|       city|
+---+----+-----------+
|LAX| SEA|    Seattle|
|SFO| LAX|Los Angeles|
|JFK| LAX|Los Angeles|
|SEA| JFK|   New York|
|SFO| JFK|   New York|
+---+----+-----------+


scala>
```

Step 3: Again, perform join on dataframes in step 1 and 2 to get the source city and destination city.the same join as above to get the destination city.

```
val flightsAirportsDESTDF=flightsDF.as("flights").join(airportsDF.as("airports"),$"flights.dest"===$"airports.code").select($"flights.src",$"flights.dest",$"airports.city")
```

```
scala> val flightsAirportsDF=flightsAirportsSRCDF.as("source").join(flightsAirportsDESTDF.as("destination"),$"source.src"===$"destination.src" && $"source.dest
"===$"destination.dest").select($"source.city",$"destination.city",$"source.airline",$"source.sno",$"source.duration").toDF("sourcecity","destinationcity","air
line","sno","duration")
flightsAirportsDF: org.apache.spark.sql.DataFrame = [sourcecity: string, destinationcity: string ... 3 more fields]

scala> flightsAirportsDF.show
+-------------+---------------+-------+----+--------+
|   sourcecity|destinationcity|airline| sno|duration|
+-------------+---------------+-------+----+--------+
|     New York|    Los Angeles|     DL| 424|    7:10|
|San Francisco|       New York|     VX|  12|    7:05|
|San Francisco|    Los Angeles|     AA|1250|    7:05|
|      Seattle|       New York|     DL| 418|    7:00|
|  Los Angeles|        Seattle|     DL|5737|    7:10|
+-------------+---------------+-------+----+--------+
```

Step 4: Now, to get the final result make a join of step 3 with airline dataframe

```
val result=flightsAirportsDF.alias("f").join(airlineDF.alias("a"),$"a.airline"===$"f.airline").select($"f.sourcecity",$"f.destinationcity",$"a.airlinename",$"f.sno",$"f.duration")
```

```
scala> val result=flightsAirportsDF.alias("f").join(airlineDF.alias("a"),$"a.airline"===$"f.airline").select($"f.sourcecity",$"f.destinationcity",$"a.airlinena
me",$"f.sno",$"f.duration")
result: org.apache.spark.sql.DataFrame = [sourcecity: string, destinationcity: string ... 3 more fields]

scala> result.show
+-------------+---------------+-----------------+----+--------+
|   sourcecity|destinationcity|      airlinename| sno|duration|
+-------------+---------------+-----------------+----+--------+
|San Francisco|    Los Angeles|American Airlines|1250|    7:05|
|     New York|    Los Angeles|   Delta Airlines| 424|    7:10|
|      Seattle|       New York|   Delta Airlines| 418|    7:00|
|  Los Angeles|        Seattle|   Delta Airlines|5737|    7:10|
|San Francisco|       New York|   Virgin America|  12|    7:05|
+-------------+---------------+-----------------+----+--------+
```