Assignment 18.3

Initial Steps:

Step1: Create a temporary table User

Read dataset from /home/acadgild/assignment\_18.1/S18\_Dataset\_User\_details.txt and create RDD user\_rdd. Create case class User with field user\_id, name. Create dataframe user\_df by mapping records splitting fields by and populating the User class object. Next create temporary table User

Code is as below:

import org.apache.spark.sql.types.{StructType, StringType, IntegerType, StructField}

val user\_rdd = sc.textFile("/home/acadgild/assignment\_18.1/S18\_Dataset\_User\_details.txt")

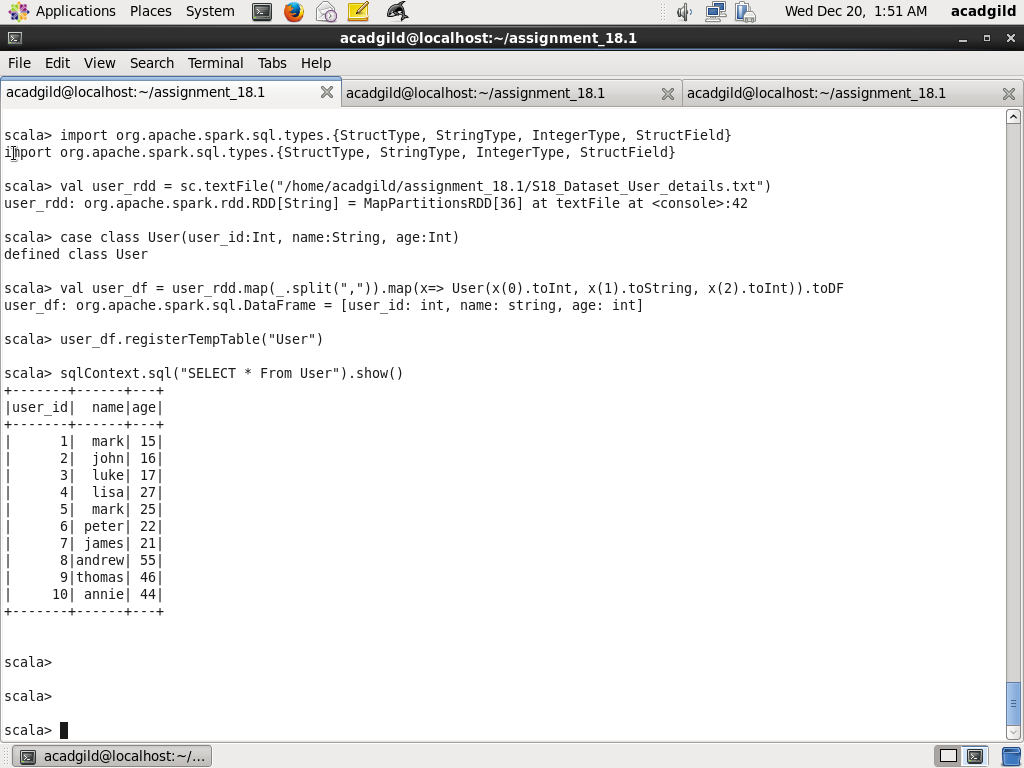
case class User(user\_id:Int, name:String, age:Int)

val user\_df = user\_rdd.map(\_.split(",")).map(x=> User(x(0).toInt, x(1).toString, x(2).toInt)).toDF

user\_df.registerTempTable("User")

sqlContext.sql("SELECT \* From User").show()

Screenshot is as below:



Step2: Create a temporary table Travel

Read dataset from /home/acadgild/assignment\_18.1/S18\_Dataset\_ Holidays.txt and create RDD user\_rdd. Create case class Travel with field user\_id, src, dest, travel\_mode distance, year\_of\_travel Create dataframe travel\_df by mapping records splitting fields by , and populating the Travel class object. Next create temporary table Travel

Code is as below:

val travel\_rdd = sc.textFile("/home/acadgild/assignment\_18.1/S18\_Dataset\_Holidays.txt")

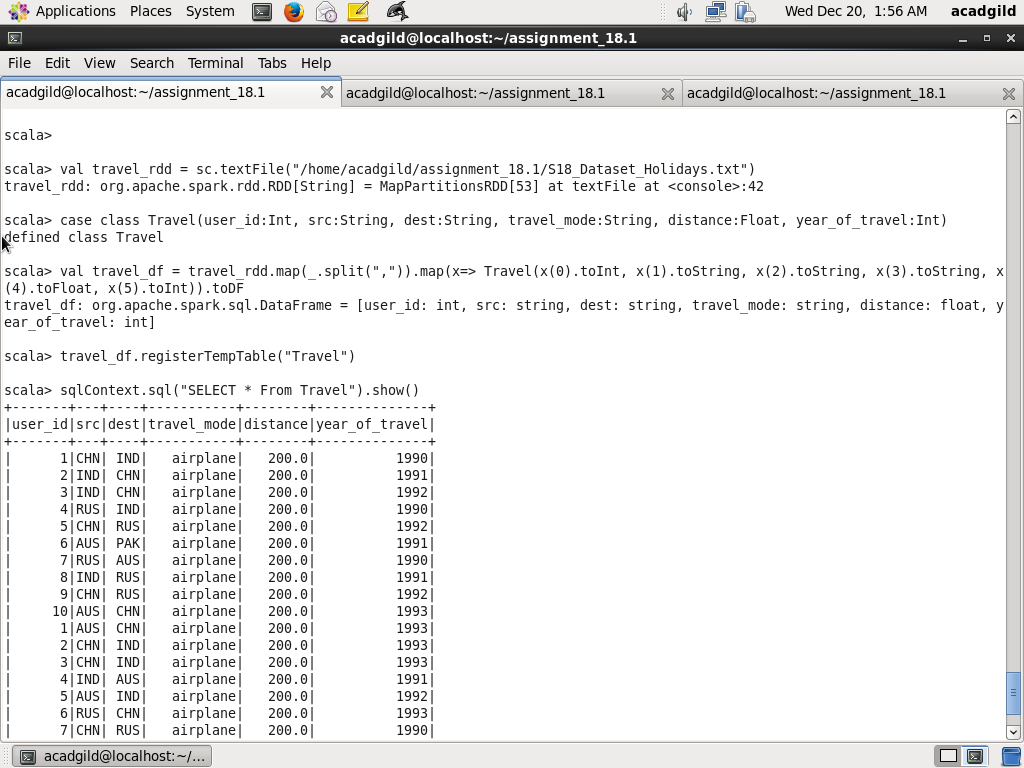
case class Travel(user\_id:Int, src:String, dest:String, travel\_mode:String, distance:Float, year\_of\_travel:Int)

val travel\_df = travel\_rdd.map(\_.split(",")).map(x=> Travel(x(0).toInt, x(1).toString, x(2).toString, x(3).toString, x(4).toFloat, x(5).toInt)).toDF

travel\_df.registerTempTable("Travel")

sqlContext.sql("SELECT \* From Travel").show()

Screenshot is as below:



Step3:

Read dataset from /home/acadgild/assignment\_18.1/S18\_Dataset\_ Transport.txt and create RDD transport\_rdd. Create case class Transport with fields travel\_mode,cost\_per\_unit Create dataframe transport\_df by mapping records splitting fields by , and populating the Transport class object. Next create temporary table Transport

val transport\_rdd = sc.textFile("/home/acadgild/assignment\_18.1/S18\_Dataset\_Transport.txt")

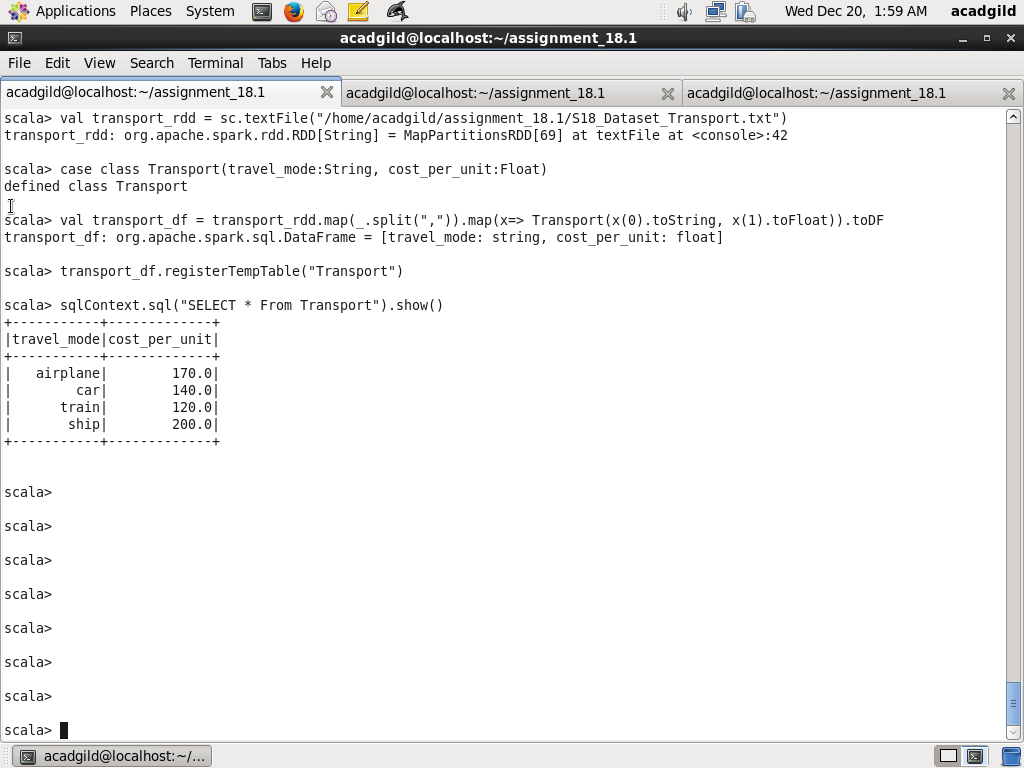
case class Transport(travel\_mode:String, cost\_per\_unit:Float)

val transport\_df = transport\_rdd.map(\_.split(",")).map(x=> Transport(x(0).toString, x(1).toFloat)).toDF

transport\_df.registerTempTable("Transport")

sqlContext.sql("SELECT \* From Transport").show()

Screenshot is as below:



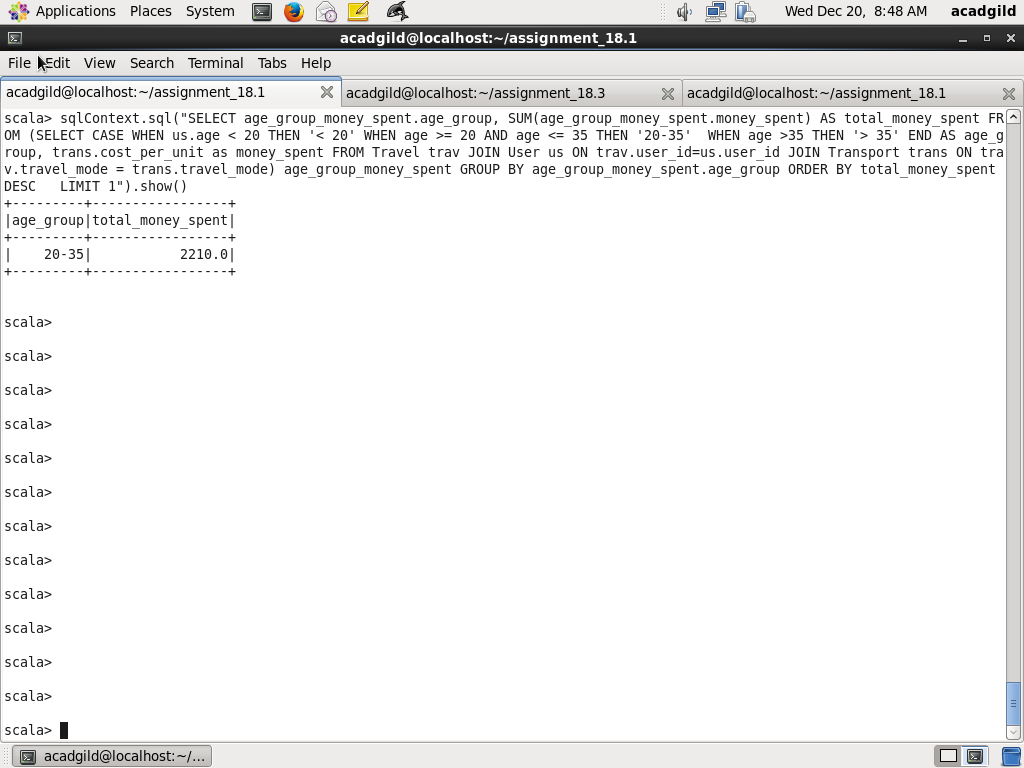
Task1: Which age group spent most amount in travelling

Here I joined temporary tables Travel and User on user\_id field and filter the age groups (<20, 20-35, >35) and using the sum function on cost\_per\_unit, calculate total\_money\_spent, group by age group and order by age\_group and take the first record using LIMIT 1

Code is as below:

sqlContext.sql("SELECT age\_group\_money\_spent.age\_group, SUM(age\_group\_money\_spent.money\_spent) AS total\_money\_spent FROM (SELECT CASE WHEN us.age < 20 THEN '< 20' WHEN age >= 20 AND age <= 35 THEN '20-35' WHEN age >35 THEN '> 35' END AS age\_group, trans.cost\_per\_unit as money\_spent FROM Travel trav JOIN User us ON trav.user\_id=us.user\_id JOIN Transport trans ON trav.travel\_mode = trans.travel\_mode) age\_group\_money\_spent GROUP BY age\_group\_money\_spent.age\_group ORDER BY total\_money\_spent DESC LIMIT 1").show()

Screenshot is as below:



Task2: What is the amount spent by each age group every year in travelling

Here I joined temporary tables Travel and User on user\_id field and filter the age groups (<20, 20-35, >35) and using the sum function on cost\_per\_unit, calculate total\_money\_spent, group by year and age group

Code is as below:

sqlContext.sql("SELECT age\_group\_money\_spent.year\_of\_travel, age\_group\_money\_spent.age\_group, SUM(age\_group\_money\_spent.money\_spent) AS total\_money\_spent FROM (SELECT trav.year\_of\_travel, CASE WHEN us.age < 20 THEN '< 20' WHEN age >= 20 AND age <= 35 THEN '20-35' WHEN age >35 THEN '> 35' END AS age\_group, trans.cost\_per\_unit as money\_spent FROM Travel trav JOIN User us ON trav.user\_id=us.user\_id JOIN Transport trans ON trav.travel\_mode = trans.travel\_mode) age\_group\_money\_spent GROUP BY age\_group\_money\_spent.year\_of\_travel, age\_group\_money\_spent.age\_group ORDER BY age\_group\_money\_spent.year\_of\_travel, age\_group\_money\_spent.age\_group ").show()

Screenshot is as below:

