Assignment 18.2

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")

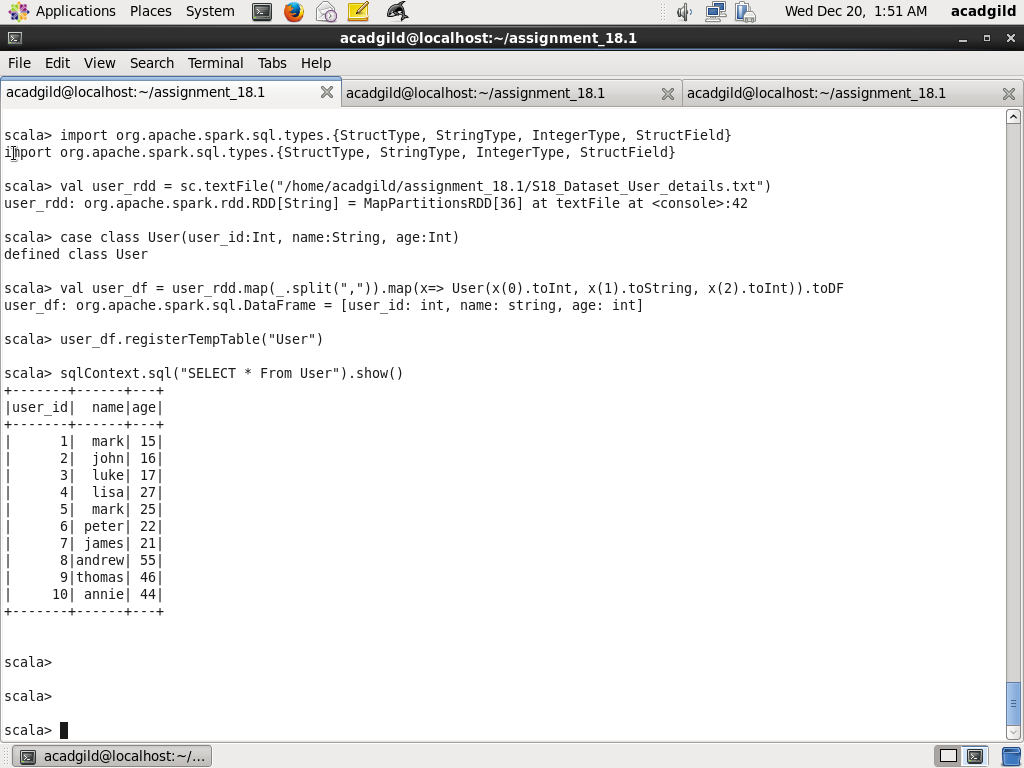
case class User(user\_id:Int, name:String, age:Int)

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")

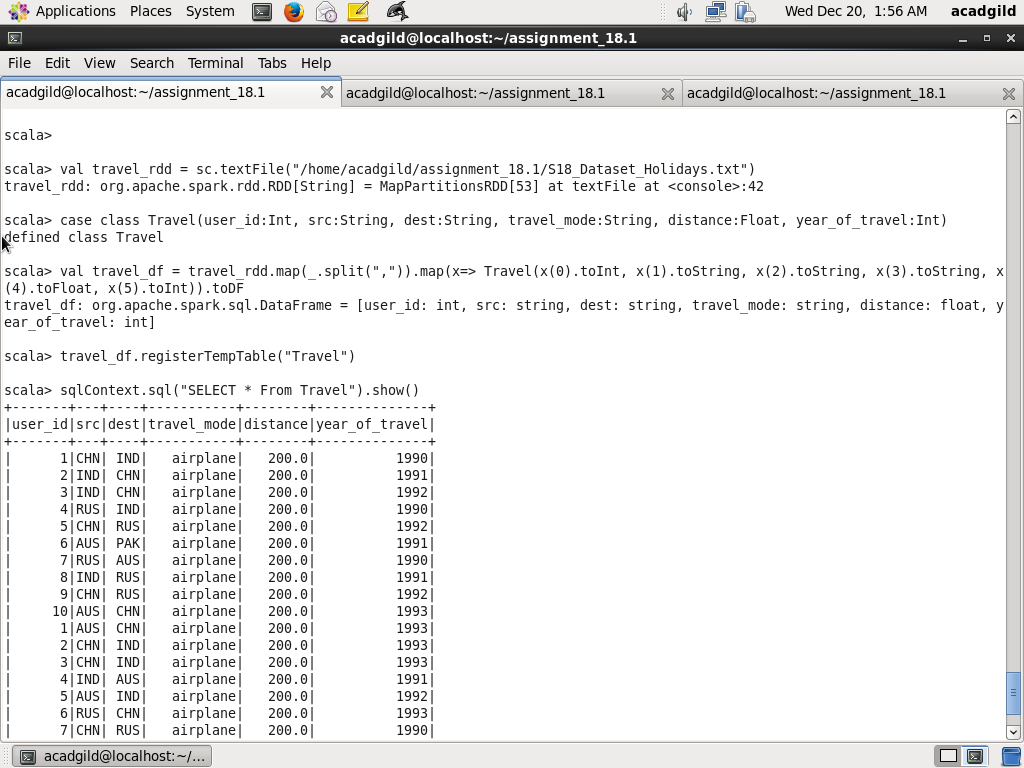
case class Travel(user\_id:Int, src:String, dest:String, travel\_mode:String, distance:Float, year\_of\_travel:Int)

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")

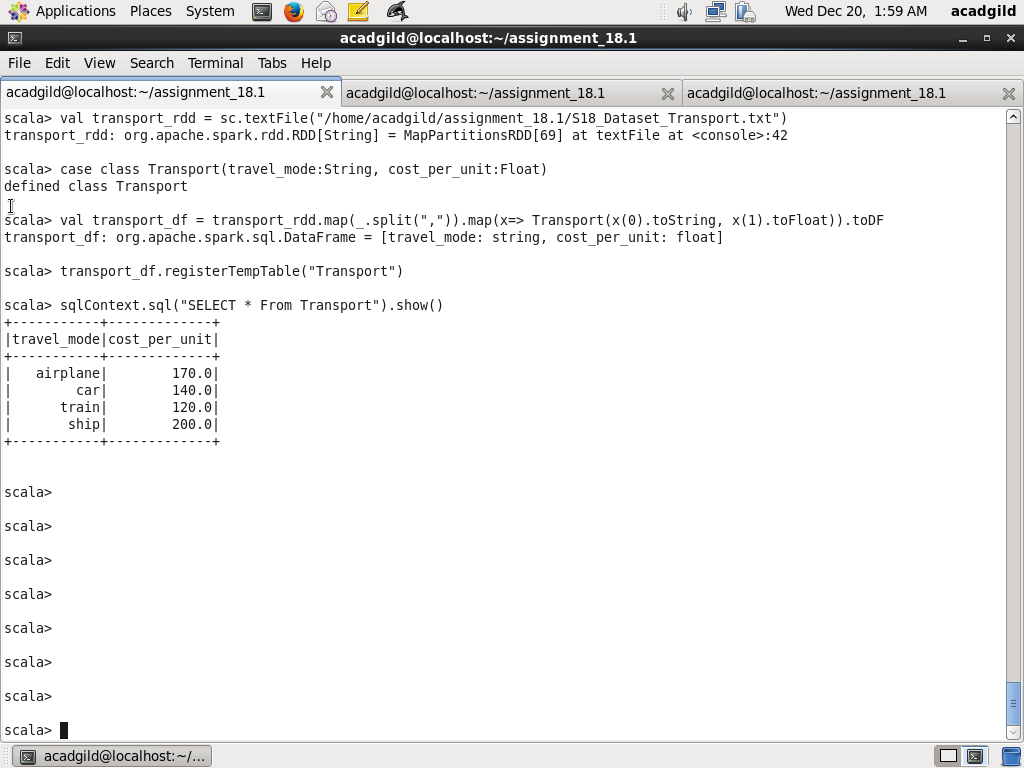
case class Transport(travel\_mode:String, cost\_per\_unit:Float)

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:



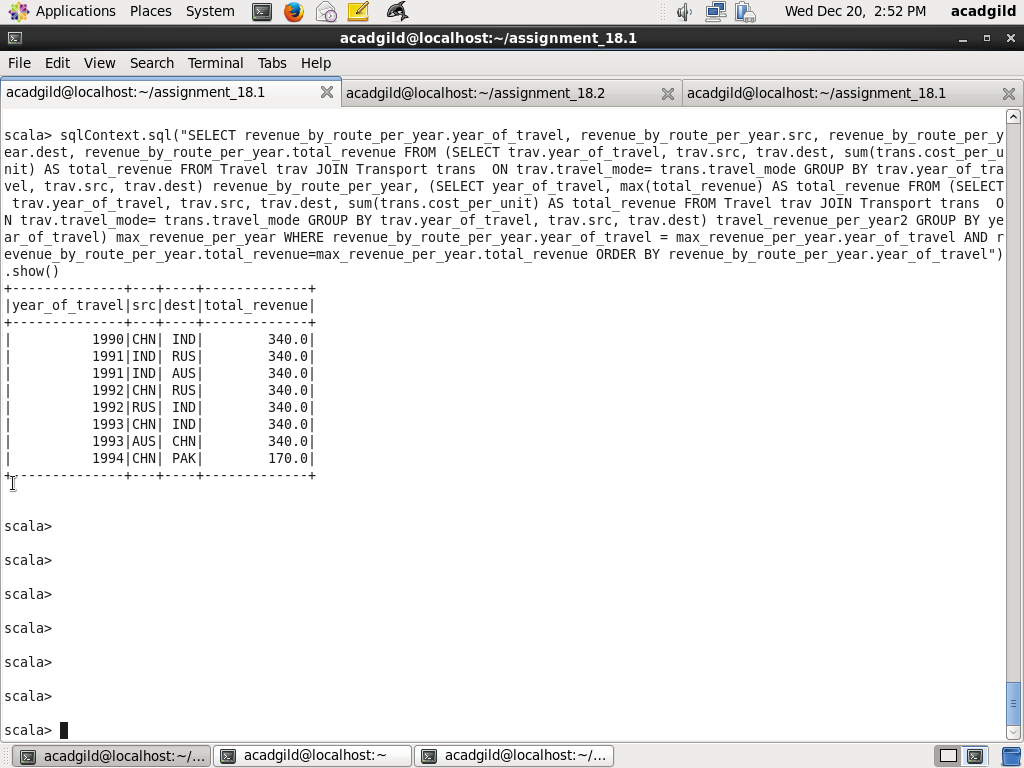
Task1: Which route is generating the most revenue per year

Note: This is a complex query as view was not working in this installation of acadgild VM. I first find the revenue by route per year and create a alias revenue\_by\_route\_per\_year. Next I find the maximum revenue per year and create alias max\_revenue\_per\_year. Next I join both the table aliases revenue\_by\_route\_per\_year and max\_revenue\_per\_year on year\_of\_travel and total\_revenue and order them by year

Code is as below:

sqlContext.sql("SELECT revenue\_by\_route\_per\_year.year\_of\_travel, revenue\_by\_route\_per\_year.src, revenue\_by\_route\_per\_year.dest, revenue\_by\_route\_per\_year.total\_revenue FROM (SELECT trav.year\_of\_travel, trav.src, trav.dest, sum(trans.cost\_per\_unit) AS total\_revenue FROM Travel trav JOIN Transport trans ON trav.travel\_mode= trans.travel\_mode GROUP BY trav.year\_of\_travel, trav.src, trav.dest) revenue\_by\_route\_per\_year, (SELECT year\_of\_travel, max(total\_revenue) AS total\_revenue FROM (SELECT trav.year\_of\_travel, trav.src, trav.dest, sum(trans.cost\_per\_unit) AS total\_revenue FROM Travel trav JOIN Transport trans ON trav.travel\_mode= trans.travel\_mode GROUP BY trav.year\_of\_travel, trav.src, trav.dest) travel\_revenue\_per\_year2 GROUP BY year\_of\_travel) max\_revenue\_per\_year WHERE revenue\_by\_route\_per\_year.year\_of\_travel = max\_revenue\_per\_year.year\_of\_travel AND revenue\_by\_route\_per\_year.total\_revenue=max\_revenue\_per\_year.total\_revenue ORDER BY revenue\_by\_route\_per\_year.year\_of\_travel").show()Screenshot is as below:

Screenshot is as below:



Task2: What is the total amount spent by each user on air travel per year

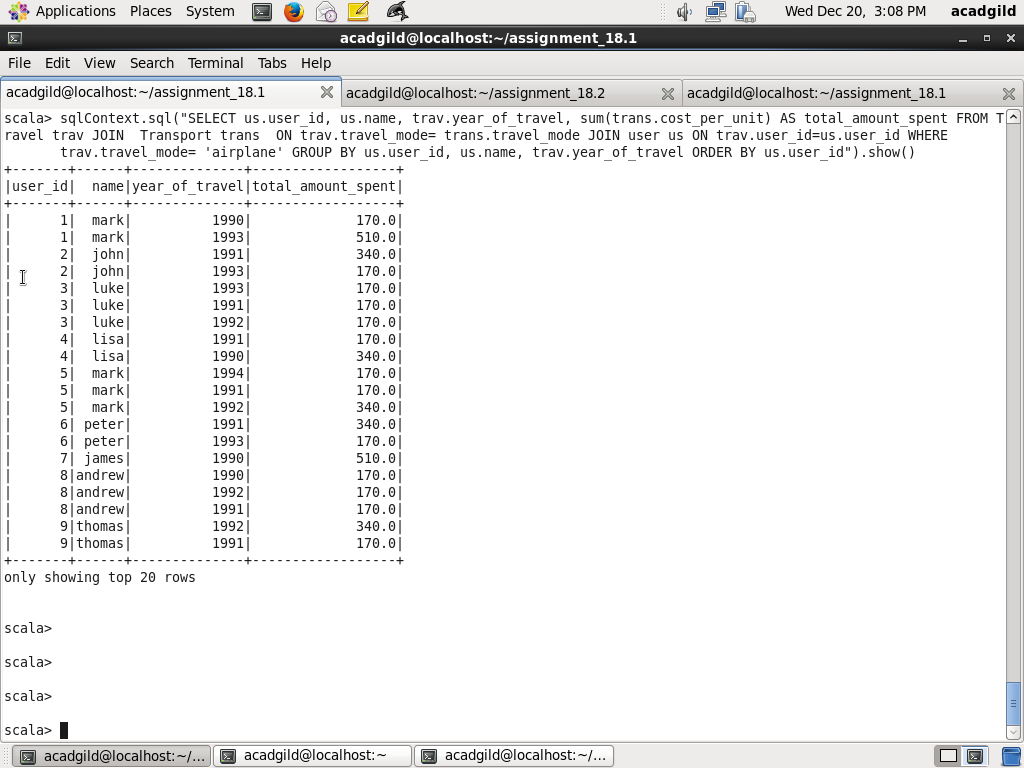
Here I have joined temporary tables Travel, User, Transport using the common fields (travel\_mode between Travel and Transport, user\_id between Travel and User) and using sum function on cost\_per\_unit group by user\_id, name, with travel\_mode as airplane

Code is as below:

sqlContext.sql("SELECT us.user\_id, us.name, trav.year\_of\_travel, sum(trans.cost\_per\_unit) AS total\_amount\_spent FROM Travel trav JOIN Transport trans ON trav.travel\_mode= trans.travel\_mode JOIN user us ON trav.user\_id=us.user\_id WHERE trav.travel\_mode= 'airplane' GROUP BY us.user\_id, us.name, trav.year\_of\_travel ORDER BY us.user\_id").show()

(NOTE: I have used both user\_id and user\_name as there are two different users who have same name but different id. For example, user\_id 1 and 5 both have name as Mark)

Screenshot is as below:



Task3: Considering age group < 20, 20-35 , > 35. Which age group has travelled most every yeat

Using CASE WHEN first filter the age groups (<20, 20-35, > 35) joining tables User and Travel on common field user\_id. Next find count number of travels per year per age group and create a table alias just\_travel\_count. Also found the maximum count of number of travels per year and create table alias max\_travel\_count. Next join these two aliases on field on travel\_count and year\_of\_travel

(Note: As view is not working in this setup, I have to use complex query to solve this problem)

Code is as below:

sqlContext.sql("SELECT DISTINCT just\_travel\_count.year\_of\_travel, just\_travel\_count.age\_group FROM (SELECT age\_group\_count.year\_of\_travel,age\_group\_count.age\_group, COUNT(\*) AS travel\_count 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 FROM Travel trav JOIN User us ON trav.user\_id=us.user\_id) age\_group\_count GROUP BY age\_group\_count.year\_of\_travel, age\_group\_count.age\_group) just\_travel\_count, (SELECT year\_agegroup\_travel\_count.year\_of\_travel, max(year\_agegroup\_travel\_count.travel\_count) AS travel\_count FROM (SELECT age\_group\_count.year\_of\_travel AS year\_of\_travel,age\_group\_count.age\_group, COUNT(\*) travel\_count 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 FROM Travel trav JOIN User us ON trav.user\_id=us.user\_id) age\_group\_count GROUP BY age\_group\_count.year\_of\_travel, age\_group\_count.age\_group) year\_agegroup\_travel\_count GROUP BY year\_agegroup\_travel\_count.year\_of\_travel) max\_travel\_count WHERE just\_travel\_count.year\_of\_travel = max\_travel\_count.year\_of\_travel AND just\_travel\_count.travel\_count = max\_travel\_count.travel\_count ORDER BY just\_travel\_count.year\_of\_travel").show()

Screenshot is as below:

