Assignment 18.2:

Problem Statement:

Initial Steps:

Step1: Create a temporary table User

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

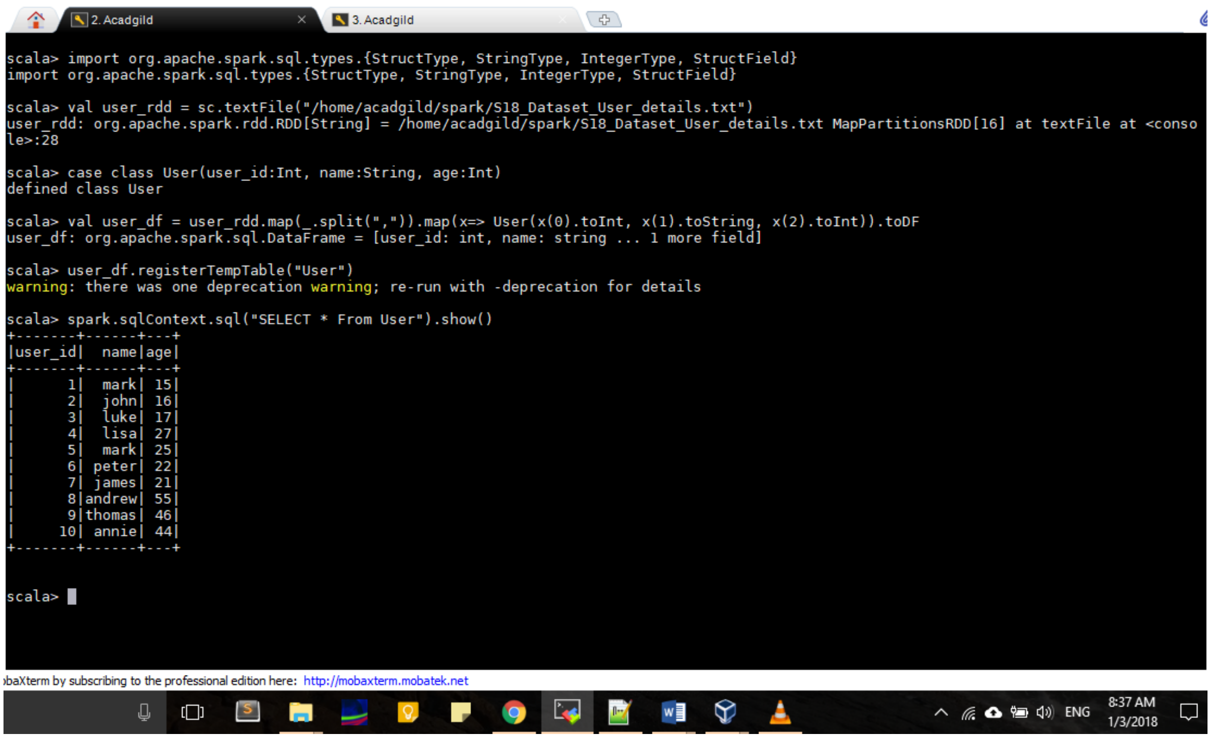
val user_rdd = sc.textFile("/home/acadgild/assignment_18.1/S18_Dataset_User_details.txt")

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")

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


2. Acadgild
3. Acadgild
baXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
ENG
8:37 AM
1/3/2018

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

scala> val user_rdd = sc.textFile("/home/acadgild/spark/S18_Dataset_User_details.txt")
user_rdd: org.apache.spark.rdd.RDD[String] = /home/acadgild/spark/S18_Dataset_User_details.txt MapPartitionsRDD[16] at textFile at <conso
le>:28

scala> case class User(user_id:Int, name:String, age:Int)
defined class User

scala> val user_df = user_rdd.map(_.split(",")).map(x=> User(x(0).toInt, x(1).toString, x(2).toInt)).toDF
user_df: org.apache.spark.sql.DataFrame = [user_id: int, name: string ... 1 more field]

scala> user_df.registerTempTable("User")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> spark.sqlContext.sql("SELECT * From User").show()
+-------+------+---+
|user_id|  name|age|
+-------+------+---+
|      1|  mark| 15|
|      2|  john| 16|
|      3|  luke| 17|
|      4|  lisa| 27|
|      5|  mark| 25|
|      6| peter| 22|
|      7| james| 21|
|      8|andrew| 55|
|      9|thomas| 46|
|     10| annie| 44|
+-------+------+---+


scala> 
```

Step2: Create a temporary table Travel

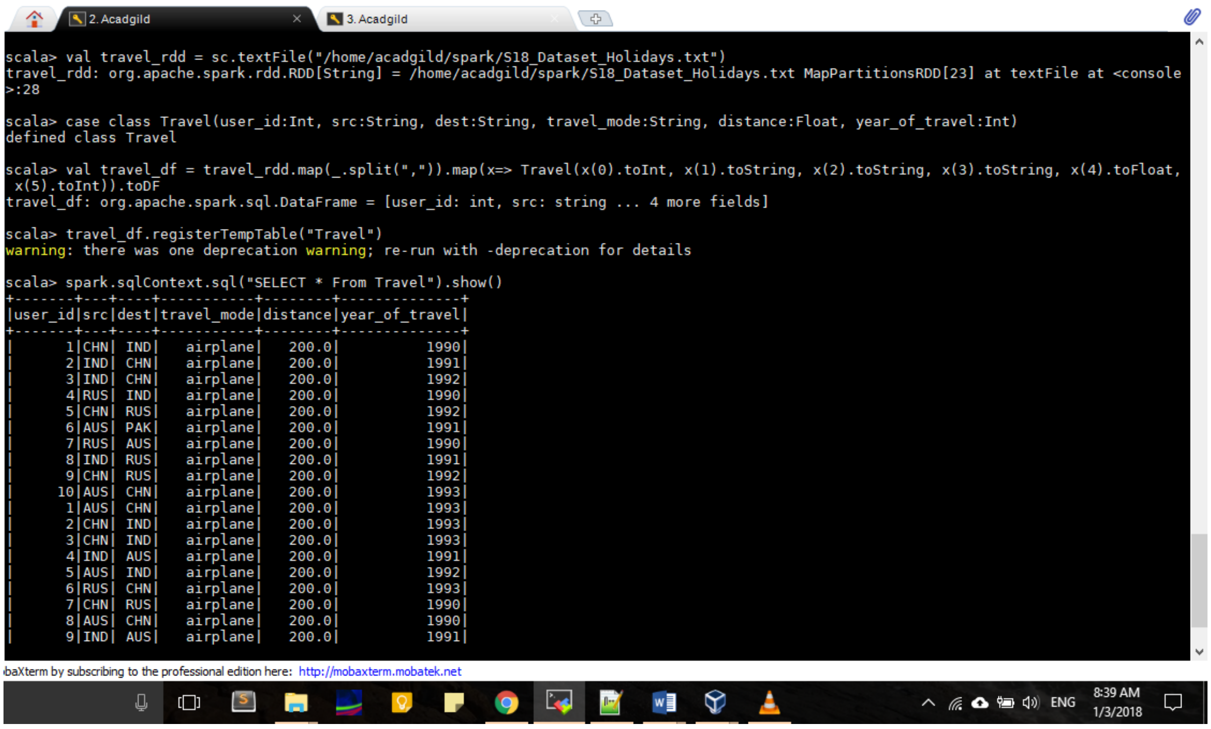
val travel_rdd = sc.textFile("/home/acadgild/spark/S18_Dataset_Holidays.txt")

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")

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


2. Acadgild
3. Acadgild
baXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
ENG
8:39 AM
1/3/2018

```
scala> val travel_rdd = sc.textFile("/home/acadgild/spark/S18_Dataset_Holidays.txt")
travel_rdd: org.apache.spark.rdd.RDD[String] = /home/acadgild/spark/S18_Dataset_Holidays.txt MapPartitionsRDD[23] at textFile at <console
>:28

scala> case class Travel(user_id:Int, src:String, dest:String, travel_mode:String, distance:Float, year_of_travel:Int)
defined class Travel

scala> 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: org.apache.spark.sql.DataFrame = [user_id: int, src: string ... 4 more fields]

scala> travel_df.registerTempTable("Travel")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> spark.sqlContext.sql("SELECT * From Travel").show()
+-------+---+----+-----------+--------+--------------+
|user_id|src|dest|travel_mode|distance|year_of_travel|
+-------+---+----+-----------+--------+--------------+
|      1|CHN| IND|   airplane|   200.0|          1990|
|      2|IND| CHN|   airplane|   200.0|          1991|
|      3|IND| CHN|   airplane|   200.0|          1992|
|      4|RUS| IND|   airplane|   200.0|          1990|
|      5|CHN| RUS|   airplane|   200.0|          1992|
|      6|AUS| PAK|   airplane|   200.0|          1991|
|      7|RUS| AUS|   airplane|   200.0|          1990|
|      8|IND| RUS|   airplane|   200.0|          1991|
|      9|CHN| RUS|   airplane|   200.0|          1992|
|     10|AUS| CHN|   airplane|   200.0|          1993|
|      1|AUS| CHN|   airplane|   200.0|          1993|
|      2|CHN| IND|   airplane|   200.0|          1993|
|      3|CHN| IND|   airplane|   200.0|          1993|
|      4|IND| AUS|   airplane|   200.0|          1991|
|      5|AUS| IND|   airplane|   200.0|          1992|
|      6|RUS| CHN|   airplane|   200.0|          1993|
|      7|CHN| RUS|   airplane|   200.0|          1990|
|      8|AUS| CHN|   airplane|   200.0|          1990|
|      9|IND| AUS|   airplane|   200.0|          1991|
```

Step3: Create temporary table Transport

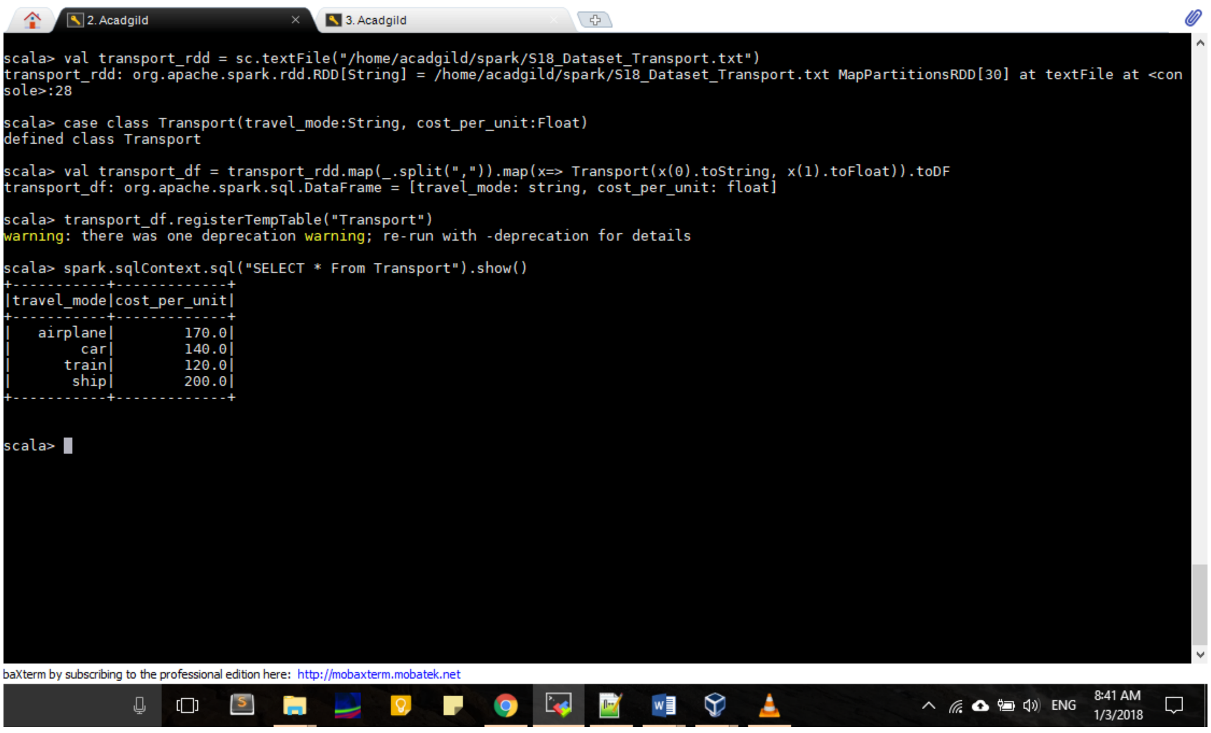
val transport_rdd = sc.textFile("/home/acadgild/spark/S18_Dataset_Transport.txt")

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")

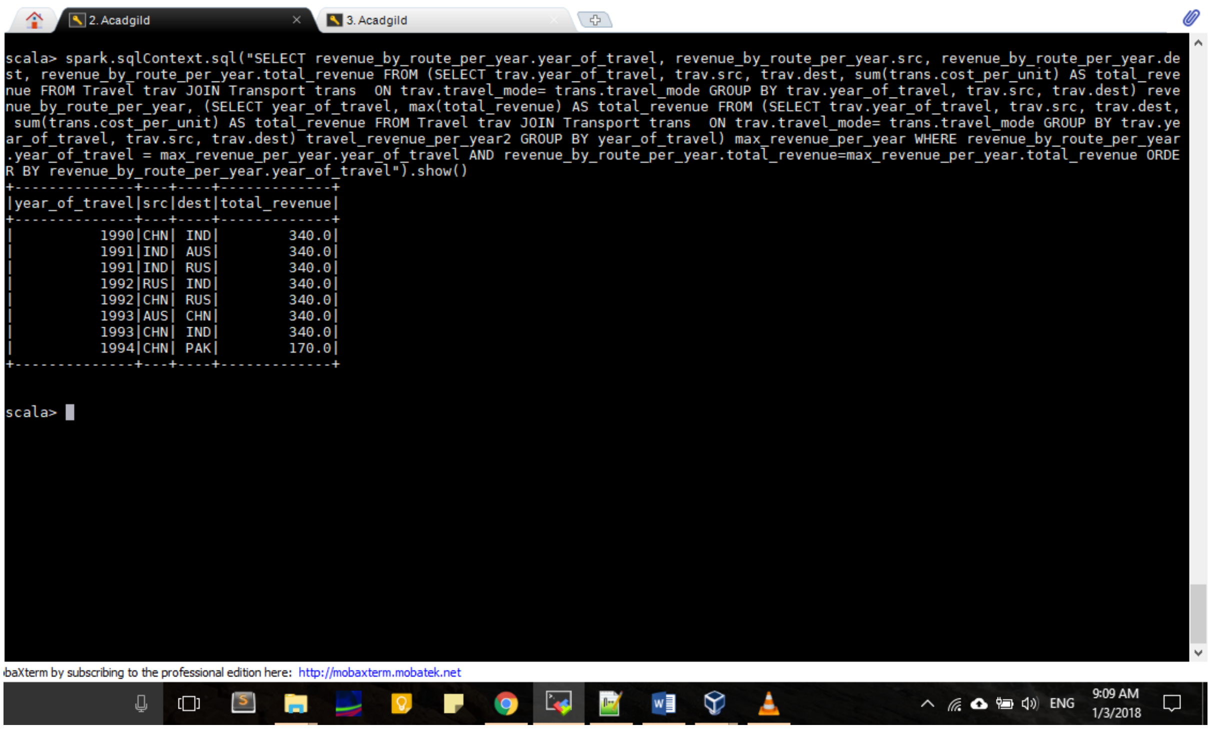
spark.sqlContext.sql("SELECT * From Transport").show()



```
scala> val transport_rdd = sc.textFile("/home/acadgild/spark/S18_Dataset_Transport.txt")
transport_rdd: org.apache.spark.rdd.RDD[String] = /home/acadgild/spark/S18_Dataset_Transport.txt MapPartitionsRDD[30] at textFile at <con
sole>:28

scala> case class Transport(travel_mode:String, cost_per_unit:Float)
defined class Transport

scala> val transport_df = transport_rdd.map(_.split(",")).map(x=> Transport(x(0).toString, x(1).toFloat)).toDF
transport_df: org.apache.spark.sql.DataFrame = [travel_mode: string, cost_per_unit: float]

scala> transport_df.registerTempTable("Transport")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> spark.sqlContext.sql("SELECT * From Transport").show()
+-----------+-------------+
|travel_mode|cost_per_unit|
+-----------+-------------+
|   airplane|        170.0|
|        car|        140.0|
|      train|        120.0|
|       ship|        200.0|
+-----------+-------------+


scala> 
```

1) Which route is generating the most revenue per year?

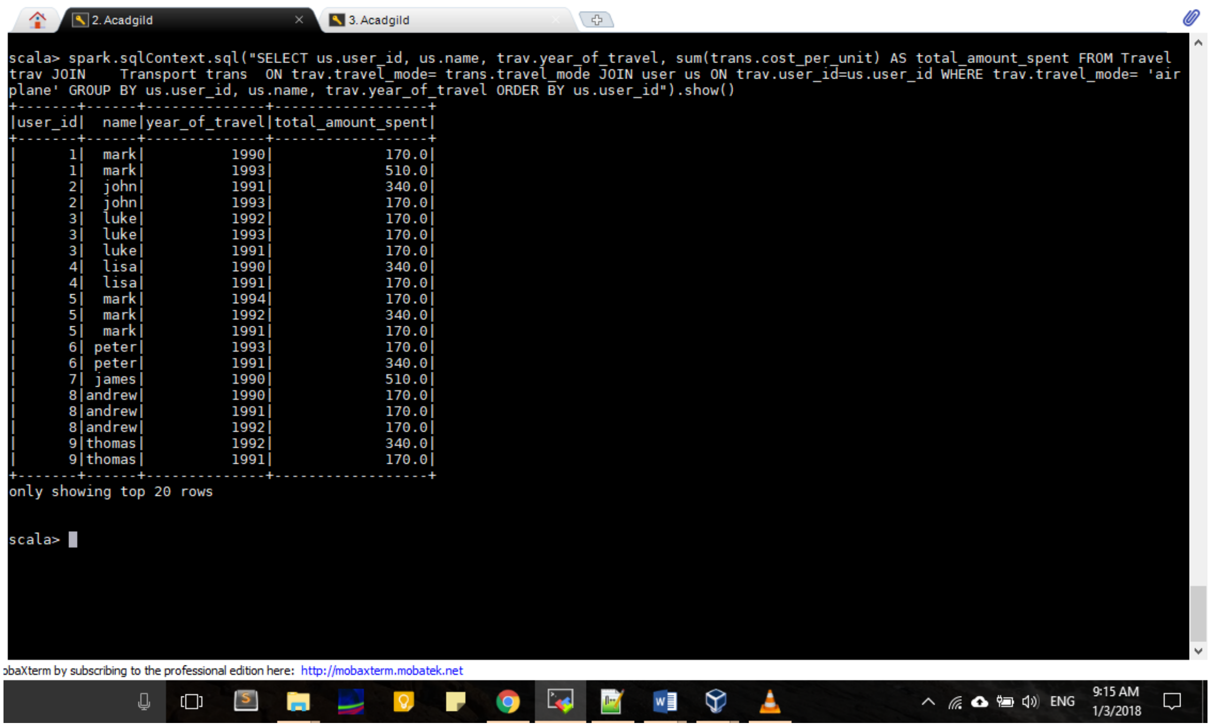
spark.sqlContext.sql("SELECT revenue_by_route_per_year.year_of_travel, revenue_by_route_per_year.src, revenue_by_route_per_year.dest, revenue_by_route_per_year.total_revenue FROM (SELECT trav.year_of_travel, trav.src, trav.dest, sum(trans.cost_per_unit) AS total_revenue FROM Travel trav JOIN Transport 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()



```
scala> spark.sqlContext.sql("SELECT revenue_by_route_per_year.year_of_travel, revenue_by_route_per_year.src, revenue_by_route_per_year.de
st, revenue_by_route_per_year.total_revenue FROM (SELECT trav.year_of_travel, trav.src, trav.dest, sum(trans.cost_per_unit) AS total_reve
nue FROM Travel trav JOIN Transport trans  ON trav.travel_mode= trans.travel_mode GROUP BY trav.year_of_travel, trav.src, trav.dest) reve
nue_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.ye
ar_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 ORDE
R BY revenue_by_route_per_year.year_of_travel").show()
+--------------+---+----+-------------+
|year_of_travel|src|dest|total_revenue|
+--------------+---+----+-------------+
|          1990|CHN| IND|        340.0|
|          1991|IND| AUS|        340.0|
|          1991|IND| RUS|        340.0|
|          1992|RUS| IND|        340.0|
|          1992|CHN| RUS|        340.0|
|          1993|AUS| CHN|        340.0|
|          1993|CHN| IND|        340.0|
|          1994|CHN| PAK|        170.0|
+--------------+---+----+-------------+


scala>
```

2) What is the total amount spent by every user on air-travel per year

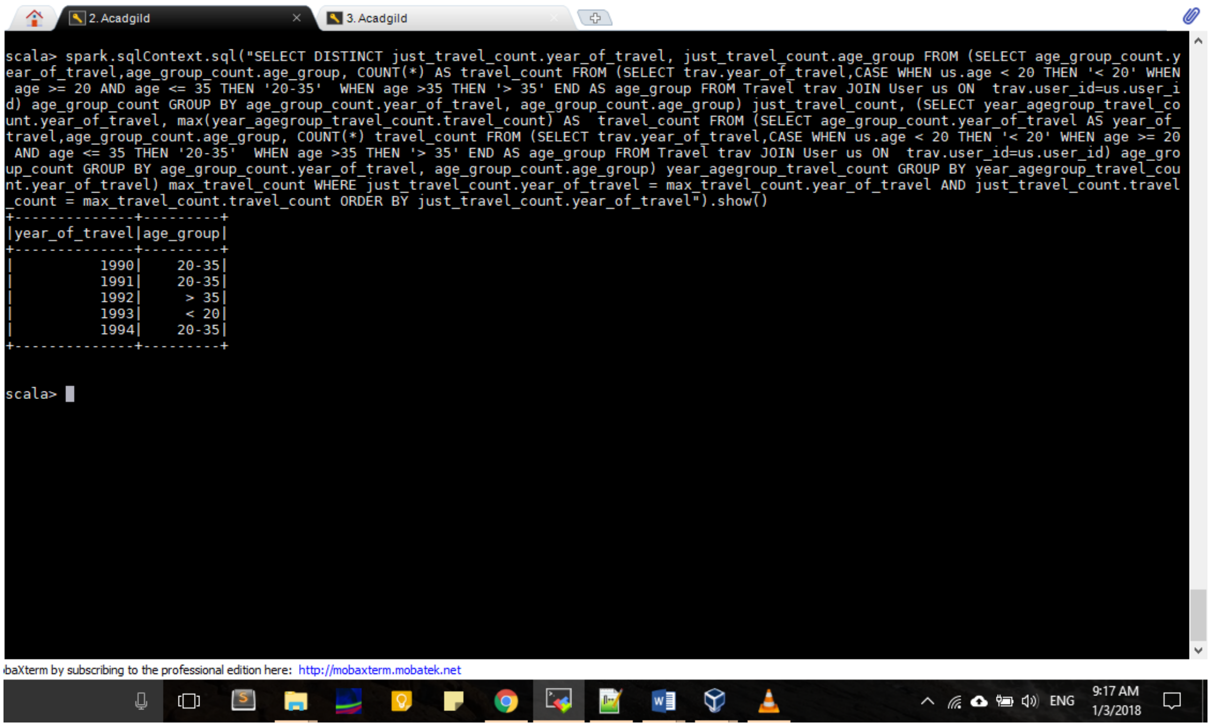
spark.sqlContext.sql("SELECT us.user_id, us.name, trav.year_of_travel, sum(trans.cost_per_unit) AS total_amount_spent FROM Travel trav JOIN   Transport trans ON trav.travel_mode= trans.travel_mode JOIN user us ON trav.user_id=us.user_id WHERE trav.travel_mode= 'airplane' GROUP BY us.user_id, us.name, trav.year_of_travel ORDER BY us.user_id").show()



```
scala> spark.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= 'air
plane' GROUP BY us.user_id, us.name, trav.year_of_travel ORDER BY us.user_id").show()
+-------+------+--------------+------------------+
|user_id|  name|year_of_travel|total_amount_spent|
+-------+------+--------------+------------------+
|      1|  mark|          1990|             170.0|
|      1|  mark|          1993|             510.0|
|      2|  john|          1991|             340.0|
|      2|  john|          1993|             170.0|
|      3|  luke|          1992|             170.0|
|      3|  luke|          1993|             170.0|
|      3|  luke|          1991|             170.0|
|      4|  lisa|          1990|             340.0|
|      4|  lisa|          1991|             170.0|
|      5|  mark|          1994|             170.0|
|      5|  mark|          1992|             340.0|
|      5|  mark|          1991|             170.0|
|      6| peter|          1993|             170.0|
|      6| peter|          1991|             340.0|
|      7| james|          1990|             510.0|
|      8|andrew|          1990|             170.0|
|      8|andrew|          1991|             170.0|
|      8|andrew|          1992|             170.0|
|      9|thomas|          1992|             340.0|
|      9|thomas|          1991|             170.0|
+-------+------+--------------+------------------+
only showing top 20 rows


scala>
```

3) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year.

spark.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()



```
scala> spark.sqlContext.sql("SELECT DISTINCT just_travel_count.year_of_travel, just_travel_count.age_group FROM (SELECT age_group_count.y
ear_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_i
d) age_group_count GROUP BY age_group_count.year_of_travel, age_group_count.age_group) just_travel_count, (SELECT year_agegroup_travel_co
unt.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_gro
up_count GROUP BY age_group_count.year_of_travel, age_group_count.age_group) year_agegroup_travel_count GROUP BY year_agegroup_travel_cou
nt.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()
+--------------+---------+
|year_of_travel|age_group|
+--------------+---------+
|          1990|    20-35|
|          1991|    20-35|
|          1992|     > 35|
|          1993|     < 20|
|          1994|    20-35|
+--------------+---------+


scala> 
```

2. Acadgild
3. Acadgild
baXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
ENG
9:17 AM
1/3/2018