SPARK SQL1 ASSIGNMENT1

TASK 1: What is the distribution of the total number of air-travelers per year

```
Applications  Places  System                                   Fri Jul  6, 12:14 AM   Acadgild
Browse and run installed applications        acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
18/07/05 23:57:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/07/05 23:57:04 WARN util.Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 10.0
.3.15 instead (on interface eth16)
18/07/05 23:57:04 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
18/07/05 23:57:11 WARN util.Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://10.0.3.15:4041
Spark context available as 'sc' (master = local[*], app id = local-1530815233447).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.2.1
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val lines = sc.textFile("/user/spark/Holidays.txt").map(_.split(","))
lines: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[2] at map at <console>:24

scala> case class Holidays(id:Int,source:String,destination:String,transportmode:String,distance:Float,year:Int)
defined class Holidays

scala> val holidayDF = lines.map(att => Holidays(att(0).toInt,att(1),att(2),att(3),att(4).toFloat,att(5).toInt)).toDF;
```

```
18/07/06 00:14:13 WARN metastore.ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
holidayDF: org.apache.spark.sql.DataFrame = [id: int, source: string ... 4 more fields]

scala> holidayDF.registerTempTable("TASK1")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val sql =spark.sql("select * from TASK1")
sql: org.apache.spark.sql.DataFrame = [id: int, source: string ... 4 more fields]

scala> sql.show()
+---+------+-----------+-------------+--------+----+
| id|source|destination|transportmode|distance|year|
+---+------+-----------+-------------+--------+----+
|  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|
| 10|   RUS|        CHN|     airplane|   200.0|1992|
+---+------+-----------+-------------+--------+----+
only showing top 20 rows
```

SPARK SQL1 ASSIGNMENT1

RESULT OF TASK 1:

```
scala> val task1 = spark.sql("select count(*),year from TASK1 group by year order by year desc")
task1: org.apache.spark.sql.DataFrame = [count(1): bigint, year: int]

scala> task1.show()
+--------+----+
|count(1)|year|
+--------+----+
|       1|1994|
|       7|1993|
|       7|1992|
|       9|1991|
|       8|1990|
+--------+----+


scala>
[acadgild@localhost:~]  acadgild@localhost:~
```

TASK2: What is the total air distance covered by each user per year

```
scala> val userDetail = sc.textFile("/user/spark/User_Details.txt").map(_.split(","))
userDetail: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[28] at map at <console>:24

scala> case class UserDetail(id:Int,name:String,age:Int)
defined class UserDetail

scala> val userDetailDF = userDetail.map(attr => UserDetail(attr(0).toInt,attr(1),attr(2).toInt)).toDF;
userDetailDF: org.apache.spark.sql.DataFrame = [id: int, name: string ... 1 more field]

scala> userDetailDF.show()
+---+------+---+
| 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> val user = holidayDF.join(userDetailDF, "id")
user: org.apache.spark.sql.DataFrame = [id: int, source: string ... 6 more fields]

scala>
[acadgild@localhost:~]  acadgild@localhost:~
```

SPARK SQL1 ASSIGNMENT1

```
scala> user.show()
+---+------+-----------+-------------+--------+----+------+---+
| id|source|destination|transportmode|distance|year|  name|age|
+---+------+-----------+-------------+--------+----+------+---+
|  1|   CHN|        IND|     airplane|   200.0|1990|  mark| 15|
|  1|   AUS|        CHN|     airplane|   200.0|1993|  mark| 15|
|  1|   PAK|        IND|     airplane|   200.0|1993|  mark| 15|
|  1|   PAK|        AUS|     airplane|   200.0|1993|  mark| 15|
|  6|   AUS|        PAK|     airplane|   200.0|1991| peter| 22|
|  6|   RUS|        CHN|     airplane|   200.0|1993| peter| 22|
|  6|   PAK|        RUS|     airplane|   200.0|1991| peter| 22|
|  3|   IND|        CHN|     airplane|   200.0|1992|  luke| 17|
|  3|   CHN|        IND|     airplane|   200.0|1993|  luke| 17|
|  3|   CHN|        PAK|     airplane|   200.0|1991|  luke| 17|
|  5|   CHN|        RUS|     airplane|   200.0|1992|  mark| 25|
|  5|   AUS|        IND|     airplane|   200.0|1992|  mark| 25|
|  5|   IND|        PAK|     airplane|   200.0|1991|  mark| 25|
|  5|   CHN|        PAK|     airplane|   200.0|1994|  mark| 25|
|  9|   CHN|        RUS|     airplane|   200.0|1992|thomas| 46|
|  9|   IND|        AUS|     airplane|   200.0|1991|thomas| 46|
|  9|   RUS|        IND|     airplane|   200.0|1992|thomas| 46|
|  4|   RUS|        IND|     airplane|   200.0|1990|  lisa| 27|
|  4|   IND|        AUS|     airplane|   200.0|1991|  lisa| 27|
|  4|   CHN|        PAK|     airplane|   200.0|1990|  lisa| 27|
+---+------+-----------+-------------+--------+----+------+---+
only showing top 20 rows

scala>
```

RESULT TASK2:

```
Applications  Places  System                                  Fri Jul  6, 12:30 AM   Acadgild
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help

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

scala> user.registerTempTable("TASK2")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val task2 = spark.sql("SELECT name,year,sum(distance) as Total_Distance from USER group by name,year order by name")
task2: org.apache.spark.sql.DataFrame = [name: string, year: int ... 1 more field]

scala> task2.show()
+------+----+--------------+
|  name|year|Total_Distance|
+------+----+--------------+
|andrew|1991|         200.0|
|andrew|1992|         200.0|
|andrew|1990|         200.0|
| annie|1993|         200.0|
| annie|1990|         200.0|
| annie|1992|         200.0|
| james|1990|         600.0|
|  john|1993|         200.0|
|  john|1991|         400.0|
|  lisa|1991|         200.0|
|  lisa|1990|         400.0|
|  luke|1991|         200.0|
|  luke|1992|         200.0|
|  luke|1993|         200.0|
|  mark|1994|         200.0|
|  mark|1991|         200.0|
|  mark|1993|         600.0|
|  mark|1990|         200.0|
|  mark|1992|         400.0|
| peter|1993|         200.0|
+------+----+--------------+
only showing top 20 rows

scala>
[acadgild@localhost:~]   acadgild@localhost:~
```

SPARK SQL1 ASSIGNMENT1

TASK 3: Which user has travelled the largest distance till date

```
scala> val task3 = spark.sql("SELECT name,sum(distance) as Total_Distance from USER group by name order by sum(distance) desc
")
task3: org.apache.spark.sql.DataFrame = [name: string, Total_Distance: double]

scala> task3.show(1)
+----+--------------+
|name|Total_Distance|
+----+--------------+
|mark|        1600.0|
+----+--------------+
only showing top 1 row


scala> task3.show()
+------+--------------+
|  name|Total_Distance|
+------+--------------+
|  mark|        1600.0|
| peter|         600.0|
| annie|         600.0|
|  lisa|         600.0|
|andrew|         600.0|
|  john|         600.0|
|  luke|         600.0|
|thomas|         600.0|
| james|         600.0|
+------+--------------+


scala>
[acadgild@localhost:~]  acadgild@localhost:~
```

TASK 4: What is the most preferred destination for all users.

```
scala> val task4 = spark.sql("SELECT name,destination,count(destination) as Total  from USER group by name,destination order
by count(destination) desc")
task4: org.apache.spark.sql.DataFrame = [name: string, destination: string ... 1 more field]

scala> task4.registerTempTable("MAINTASK4")
warning: there was one deprecation warning; re-run with -deprecation for details
```

SPARK SQL1 ASSIGNMENT1

```
... 48 elided

scala> val task4 = spark.sql("select * from MAINTASK4")
task4: org.apache.spark.sql.DataFrame = [name: string, destination: string ... 1 more field]

scala> task4.show()
+------+-----------+-----+
|  name|destination|Total|
+------+-----------+-----+
|  mark|        IND|    3|
| annie|        CHN|    2|
|  mark|        PAK|    2|
| annie|        AUS|    1|
|  john|        RUS|    1|
|andrew|        RUS|    1|
|  mark|        AUS|    1|
|thomas|        IND|    1|
|  john|        CHN|    1|
|  lisa|        IND|    1|
| james|        IND|    1|
|  luke|        IND|    1|
|andrew|        CHN|    1|
|  luke|        PAK|    1|
| peter|        PAK|    1|
|thomas|        RUS|    1|
|  mark|        RUS|    1|
|thomas|        AUS|    1|
|  john|        IND|    1|
| peter|        CHN|    1|
+------+-----------+-----+
only showing top 20 rows
```

```
scala> val task4 = spark.sql("select name,destination,max(Total) as MostPrefDest from MAINTASK4 group by name,destination ord
er by name desc")
task4: org.apache.spark.sql.DataFrame = [name: string, destination: string ... 1 more field]

scala> task4.show()
+------+-----------+------------+
|  name|destination|MostPrefDest|
+------+-----------+------------+
|thomas|        RUS|           1|
|thomas|        AUS|           1|
|thomas|        IND|           1|
| peter|        PAK|           1|
| peter|        RUS|           1|
| peter|        CHN|           1|
|  mark|        IND|           3|
|  mark|        CHN|           1|
|  mark|        RUS|           1|
|  mark|        AUS|           1|
|  mark|        PAK|           2|
|  luke|        IND|           1|
|  luke|        PAK|           1|
|  luke|        CHN|           1|
|  lisa|        PAK|           1|
|  lisa|        AUS|           1|
|  lisa|        IND|           1|
|  john|        IND|           1|
|  john|        RUS|           1|
|  john|        CHN|           1|
+------+-----------+------------+
only showing top 20 rows
```

SPARK SQL1 ASSIGNMENT1

TASK 5: Which route is generating the most revenue per year

```
Applications  Places  System                                   Fri Jul  6, 12:55 AM   Acadgild
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
scala> val transport = sc.textFile("/user/spark/Transport.txt").map(_.split(","))
transport: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[142] at map at <console>:24

scala> case class TransportDetail(transportmode:String,cost:Int)
defined class TransportDetail

scala> val transportDF = transport.map(attr => TransportDetail(attr(0),attr(1).toInt)).toDF
transportDF: org.apache.spark.sql.DataFrame = [transportmode: string, cost: int]

scala> val userInfo = user.join(transportDF, "transportmode").toDF
userInfo: org.apache.spark.sql.DataFrame = [transportmode: string, id: int ... 7 more fields]

scala> userInfo.registerTempTable("TASK5")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val task5 = spark.sql("select *from TASK5")
task5: org.apache.spark.sql.DataFrame = [transportmode: string, id: int ... 7 more fields]

scala> val task5 = spark.sql("select * from TASK5")
task5: org.apache.spark.sql.DataFrame = [transportmode: string, id: int ... 7 more fields]

scala> task5.show()
+-------------+---+------+-----------+--------+----+------+---+----+
|transportmode| id|source|destination|distance|year|  name|age|cost|
+-------------+---+------+-----------+--------+----+------+---+----+
|     airplane|  1|   CHN|        IND|   200.0|1990|  mark| 15| 170|
|     airplane|  1|   AUS|        CHN|   200.0|1993|  mark| 15| 170|
|     airplane|  1|   PAK|        IND|   200.0|1993|  mark| 15| 170|
|     airplane|  1|   PAK|        AUS|   200.0|1993|  mark| 15| 170|
|     airplane|  6|   AUS|        PAK|   200.0|1991| peter| 22| 170|
|     airplane|  6|   RUS|        CHN|   200.0|1993| peter| 22| 170|
|     airplane|  6|   PAK|        RUS|   200.0|1991| peter| 22| 170|
|     airplane|  3|   IND|        CHN|   200.0|1992|  luke| 17| 170|
|     airplane|  3|   CHN|        IND|   200.0|1993|  luke| 17| 170|
|     airplane|  3|   CHN|        PAK|   200.0|1991|  luke| 17| 170|
|     airplane|  5|   CHN|        RUS|   200.0|1992|  mark| 25| 170|
|     airplane|  5|   AUS|        IND|   200.0|1992|  mark| 25| 170|
|     airplane|  5|   IND|        PAK|   200.0|1991|  mark| 25| 170|
|     airplane|  5|   CHN|        PAK|   200.0|1994|  mark| 25| 170|
[acadgild@localhost:~]  acadgild@localhost:~  acadgild  Downloads
```

```
scala> val resulttask5 = spark.sql("select year,source,destination,sum(cost) from TASK5 group by year,source,destination")
resulttask5: org.apache.spark.sql.DataFrame = [year: int, source: string ... 2 more fields]

scala> resulttask5.show()
+----+------+-----------+---------+
|year|source|destination|sum(cost)|
+----+------+-----------+---------+
|1993|   RUS|        CHN|      170|
|1991|   IND|        PAK|      170|
|1993|   PAK|        IND|      170|
|1990|   CHN|        IND|      340|
|1990|   CHN|        RUS|      170|
|1991|   IND|        CHN|      170|
|1992|   RUS|        CHN|      170|
|1994|   CHN|        PAK|      170|
|1991|   AUS|        PAK|      170|
|1992|   RUS|        IND|      340|
|1992|   AUS|        IND|      170|
|1991|   CHN|        PAK|      170|
|1990|   CHN|        AUS|      170|
|1993|   AUS|        CHN|      340|
|1990|   RUS|        AUS|      170|
|1993|   PAK|        AUS|      170|
|1990|   CHN|        PAK|      170|
|1991|   IND|        AUS|      340|
|1990|   AUS|        CHN|      170|
|1993|   CHN|        IND|      340|
+----+------+-----------+---------+
only showing top 20 rows


scala> val resulttask5 = spark.sql("select year,source,destination,sum(cost) as TotalCost from TASK5 group by year,source,des
tination")
resulttask5: org.apache.spark.sql.DataFrame = [year: int, source: string ... 2 more fields]
```

SPARK SQL1 ASSIGNMENT1

```
scala> val output = spark.sql("select year,max(TotalCost) from MAINTASK5 group by year order by year desc")
output: org.apache.spark.sql.DataFrame = [year: int, max(TotalCost): bigint]

scala> output.show()
+----+--------------+
|year|max(TotalCost)|
+----+--------------+
|1994|           170|
|1993|           340|
|1992|           340|
|1991|           340|
|1990|           340|
+----+--------------+
```

```
scala> val result = output.join(resulttask5, "year", "max(TotalCost")).toDF
```

```
scala> val result =spark.sql("select a.year,a.source,a.destination,b.TotalCost from MAINTASK5 a inner join RESULT b on (a.yea
r=b.year) and (a.TotalCost=b.TotalCost) order by year desc")
result: org.apache.spark.sql.DataFrame = [year: int, source: string ... 2 more fields]

scala> result.show()
+----+------+-----------+---------+
|year|source|destination|TotalCost|
+----+------+-----------+---------+
|1994|   CHN|        PAK|      170|
|1993|   CHN|        IND|      340|
|1993|   AUS|        CHN|      340|
|1992|   RUS|        IND|      340|
|1992|   CHN|        RUS|      340|
|1991|   IND|        AUS|      340|
|1991|   IND|        RUS|      340|
|1990|   CHN|        IND|      340|
+----+------+-----------+---------+

scala>
```

TASK 6: What is the total amount spent by every user on air-travel per year

```
scala> val task6 = spark.sql("select id,year,sum(cost) from TASK5 group by id,year order by id")
task6: org.apache.spark.sql.DataFrame = [id: int, year: int ... 1 more field]

scala> task6.show()
+---+----+---------+
| id|year|sum(cost)|
+---+----+---------+
|  1|1990|      170|
|  1|1993|      510|
|  2|1991|      340|
|  2|1993|      170|
|  3|1991|      170|
|  3|1992|      170|
|  3|1993|      170|
|  4|1990|      340|
|  4|1991|      170|
|  5|1992|      340|
|  5|1991|      170|
|  5|1994|      170|
|  6|1991|      340|
|  6|1993|      170|
|  7|1990|      510|
|  8|1991|      170|
|  8|1990|      170|
|  8|1992|      170|
|  9|1991|      170|
|  9|1992|      340|
+---+----+---------+
only showing top 20 rows
```

SPARK SQL1 ASSIGNMENT1

TASK 7: Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year.

```
scala> val task7 =spark.sql("select id,age,year,count(id) as TotalCount from TASK5 where age >35 or age <20 or age between 20
 and 35  group by id,age,year order by age,year desc")
task7: org.apache.spark.sql.DataFrame = [id: int, age: int ... 2 more fields]

scala> task7.show()
+---+---+----+----------+
| id|age|year|TotalCount|
+---+---+----+----------+
|  1| 15|1993|         3|
|  1| 15|1990|         1|
|  2| 16|1993|         1|
|  2| 16|1991|         2|
|  3| 17|1993|         1|
|  3| 17|1992|         1|
|  3| 17|1991|         1|
|  7| 21|1990|         3|
|  6| 22|1993|         1|
|  6| 22|1991|         2|
|  5| 25|1994|         1|
|  5| 25|1992|         2|
|  5| 25|1991|         1|
|  4| 27|1991|         1|
|  4| 27|1990|         2|
| 10| 44|1993|         1|
| 10| 44|1992|         1|
| 10| 44|1990|         1|
|  9| 46|1992|         2|
|  9| 46|1991|         1|
+---+---+----+----------+
only showing top 20 rows
```