**Assignment 18.1 Spark**

2. Problem Statement

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

2) What is the total air distance covered by each user per year

3) Which user has travelled the largest distance till date

4) What is the most preferred destination for all users.

Use below link to download the dataset:

<https://drive.google.com/drive/folders/0B_P3pWagdIrrVThBaUdVSUtzbms>

import sqlContext.implicits.\_

import org.apache.spark.sql.functions.\_

val sqlcontext = new org.apache.spark.sql.SQLContext(sc)

------\ data frame 1

case class TRANSPORT (tname: String , ref: Int)

val tr = sc.textFile("Transport.txt").map(\_.split(",")).map(p => TRANSPORT(p(0),p(1).trim.toInt)).toDF()

tr.registerTempTable("Transport");

val trp = sqlContext.sql("SELECT tname, ref FROM Transport")

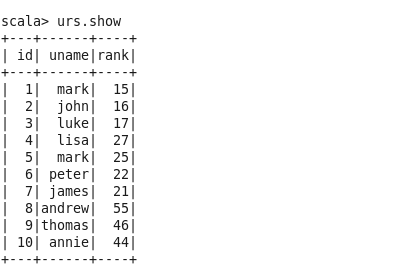
-----\ data frame 2

case class Users (id: Int, uname: String , rank: Int)

val ur = sc.textFile("Users.txt").map(\_.split(",")).map(p => Users(p(0).toInt,p(1),p(2).trim.toInt)).toDF()

ur.registerTempTable("Users");

val urs = sqlContext.sql("SELECT id, uname, rank FROM Users")



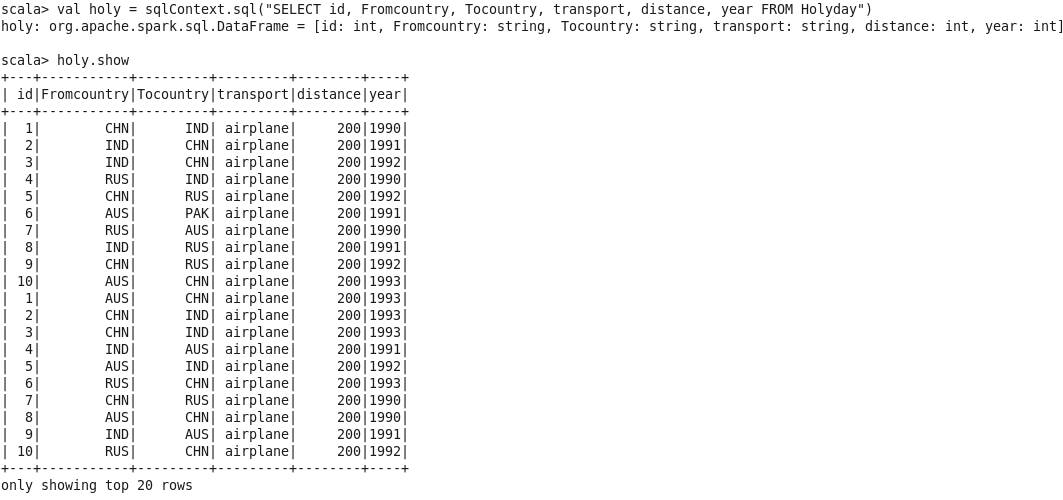
------\ data frame 3

case class Holyday (id: Int, Fromcountry: String ,Tocountry: String, transport: String, distance: Int, year: Int)

val hday = sc.textFile("Holidays.txt").map(\_.split(",")).map(p => Holyday(p(0).toInt, p(1), p(2), p(3),p(4).toInt ,p(5).trim.toInt)).toDF()

hday.registerTempTable("Holyday");

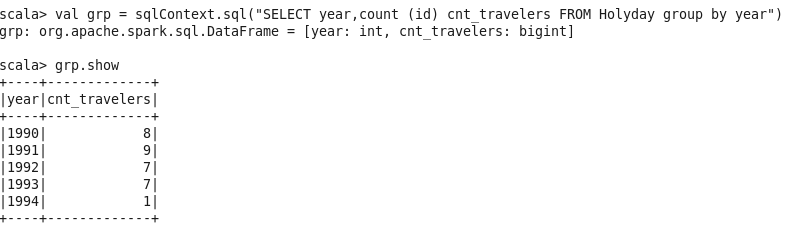
val holy = sqlContext.sql("SELECT id, Fromcountry, Tocountry, transport, distance, year FROM Holyday")



PS : --1

**1) What is the distribution of the total number of air-travellers per year**

val grp = sqlContext.sql("SELECT year,count (id) cnt\_travelers FROM Holyday group by year")



val joindf = df\_3.join(itemdf, df\_3("\_1") === itemdf("\_1") , "inner")

.select("name","description")

.show

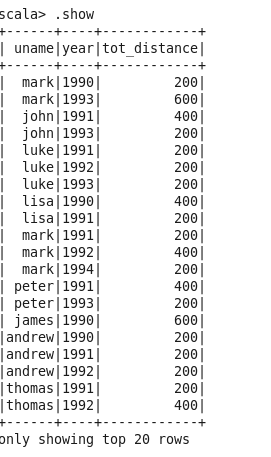
**2.What is the total air distance covered by each user per year**

val grp = sqlContext.sql("SELECT id,year,sum(distance) tot\_distance FROM Holyday group by id,year")

val joindf = grp.join(urs, grp("id") === urs("id") , "inner")

.select("uname","year","tot\_distance")

.show



**3. Which user has travelled the largest distance till date?**

val grp = sqlContext.sql("Select distinct id, max(sum\_distance) topdis from(SELECT distinct id,sum(distance) sum\_distance FROM Holyday group by id) A group by id")

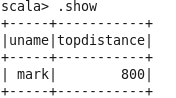
val grp1 = sqlContext.sql("Select max(sum\_distance) topdistance from(SELECT id,sum(distance) sum\_distance FROM Holyday group by id) A")

val joindf = grp.join(grp1, grp("topdis") === grp1("topdistance") , "inner")

.join(urs, grp("id") === urs("id") , "inner")

.select("uname","topdistance").distinct

.show



4 What is the most preferred destination for all users.

val grp = sqlContext.sql("SELECT Tocountry, count (Tocountry) cnt FROM Holyday group by Tocountry order by count (Tocountry) desc").limit(1)

**most preferred destination:**

