ASSIGNMENT 20.1

Based on the data set given find the following

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.
5)Which route is generating the most revenue per year
6) What is the total amount spent by every user on air-travel per year
7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year.

Solution#
Answers to the above questions are given below as the program output. The complete source code is given below

Ans.1 Distribution of the total number of air-travelers per year

```
+----+-----+
|year|count|
+----+-----+
|1992|    7|
|1994|    1|
|1993|    7|
|1990|    8|
|1991|    9|
+----+-----+
```

Ans.2 Total air distance covered by each user per year

```
+------+----+-------------+
|  name|year|sum(distance)|
+------+----+-------------+
|  mark|1990|          200|
|  mark|1993|          600|
| peter|1993|          200|
| peter|1991|          400|
|  luke|1992|          200|
|  luke|1991|          200|
|  luke|1993|          200|
|  mark|1991|          200|
|  mark|1994|          200|
|  mark|1992|          400|
|thomas|1992|          400|
|thomas|1991|          200|
|  lisa|1990|          400|
```

```
|  lisa|1991|         200|
|andrew|1990|         200|
|andrew|1992|         200|
|andrew|1991|         200|
| james|1990|         600|
| annie|1992|         200|
| annie|1993|         200|
+------+----+-------------+
```

Ans.3 The user who has travelled the largest distance till date
[mark,1600]

Ans.4 The most preferred destination for all users
[IND,9]

Ans.5 Which route is generating the most revenue per year
[CHN,IND,136000]

Ans.6 total amount spent by every user on air-travel per year

```
+------+----+-------+
|  name|year|expense|
+------+----+-------+
|  mark|1991|  34000|
| annie|1992|  34000|
| annie|1993|  34000|
|  mark|1994|  34000|
|  luke|1992|  34000|
|  lisa|1991|  34000|
|thomas|1991|  34000|
|  john|1993|  34000|
| peter|1991|  68000|
|  mark|1993| 102000|
|thomas|1992|  68000|
|  john|1991|  68000|
| james|1990| 102000|
|  mark|1990|  34000|
|andrew|1990|  34000|
|  luke|1991|  34000|
|andrew|1992|  34000|
| peter|1993|  34000|
|  luke|1993|  34000|
|andrew|1991|  34000|
+------+----+-------+
only showing top 20 rows
```

Ans.7 age group < 20 =7 age group > 35 =8 age group between 30 and 35 =0

Process finished with exit code 0

```
package demo

import org.apache.spark.sql.SparkSession

object SparkSql {


  case class user(uid:Int,name:String,age:Int)
  case class transport(modes:String,cost_per_unit:Int)
  case class
holidays(id:Int,src:String,dst:String,mode:String,distance:Int,year:String)


  def main(args: Array[String]): Unit = {

    // create spark session
    val spark = SparkSession.builder().master(master="local")
      .appName(name="spark sql example")
      .config("spark.some.config.option","some-value").getOrCreate()
    spark.sparkContext.setLogLevel("WARN")
    // use this to create dataframes
    import spark.implicits._
    // create dataframes by reading these files
    val userDF = spark.sparkContext

.textFile("/Users/mmisra/Desktop/acad/assignments/assignment_20.1/S20_Dataset_User_det
ails.txt")
      .map(_.split(","))
      .map(attributes => user(attributes(0).toInt,attributes(1),attributes(2).toInt))
      .toDF()
    //userDF.show()

    val transportDF = spark.sparkContext

.textFile("/Users/mmisra/Desktop/acad/assignments/assignment_20.1/S20_Dataset_Transpor
t.txt")
      .map(_.split(","))
      .map(attributes => transport(attributes(0),attributes(1).toInt))
      .toDF()
    //transportDF.show()


    val holidayDF = spark.sparkContext

.textFile("/Users/mmisra/Desktop/acad/assignments/assignment_20.1/S20_Dataset_Holidays
.txt")
      .map(_.split(","))
      .map(attributes =>
holidays(attributes(0).toInt,attributes(1),attributes(2),attributes(3),attributes(4).t
oInt,attributes(5)))
      .toDF()
    //1) What is the distribution of the total number of air-travelers per year

    // filter based on mode=airplane and then group by year and then count
    val r1 = holidayDF.filter($"mode"==="airplane").groupBy($"year").count()

    println("Ans.1 Distribution of the total number of air-travelers per year")
    r1.show()

    //2) What is the total air distance covered by each user per year
    val r2 =
holidayDF.filter($"mode"==="airplane").groupBy($"id",$"year").sum("distance")
    // join with user table to get the names and select appropriate columns for
```

```
printing
    val r21 = r2.join(userDF,$"id"===$"uid").select($"name",$"year",$"sum(distance)")

    println("Ans.2 Total air distance covered by each user per year")
    r21.show()

    //3) Which user has travelled the largest distance till date
    // we first join the user table with the holiday table
    // we group by the user name and sum the distance for each name
    // then we sort the sum column in descending order and get the first/top row
    val r3 = holidayDF.filter($"mode"==="airplane")
    val r31 = r3.join(userDF,$"id"===$"uid").select($"name",$"distance")
              .groupBy($"name").sum("distance").sort($"sum(distance)".desc).first()

    println("Ans.3 The user who has travelled the largest distance till date")
    println(r31)

    //4)What is the most preferred destination for all users
    // we select the destination and count
    val r4 = holidayDF.groupBy($"dst").count().sort($"count".desc).first()
    println("Ans.4 The most preferred destination for all users")
    println(r4)

    //5)Which route is generating the most revenue per year
    //group by src+dst and find the sum of distance
    val r5 = holidayDF.groupBy($"src", $"dst", $"mode").sum("distance")
    // join with the table where the cost of each mode of transport is mentioned
    val r51= r5.join(transportDF,$"mode"===$"modes")select($"src",
                                    $"dst",$"sum(distance)",$"cost_per_unit")
    // calculate the revewnue by multilying total distance with cost per unit , sort
in descending order and take the op one
    println("Ans.5 Which route is generating the most revenue per year")
    val r52 = r51.select($"src",$"dst",($"sum(distance)"*$"cost_per_unit")
                              .as("revenue")).sort($"revenue".desc).first()
    println(r52)

    //6) What is the total amount spent by every user on air-travel per year
    val r6 = holidayDF.filter($"mode"==="airplane")
    val r61 = r6.join(userDF,$"id"===$"uid").select($"name",
                                                    $"distance",$"year",$"mode")
      .groupBy($"name",$"year",$"mode").sum("distance")
    // join r61 with the transport DF to get the cost/per unit. The spend is total
distance x cost per unit for travel

     val r62= r61.join(transportDF,$"mode"===$"modes")select($"name",$"year",
            ($"sum(distance)"* $"cost_per_unit").as("expense"))
    println("Ans.6 total amount spent by every user on air-travel per year")
     r62.show()

    //7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling
the most
    //every year.
    // total distance traveled by each person each year
    val r7 = holidayDF.groupBy($"id",$"year").sum("distance")
    // join with user table to get the age for each travel
    val r71 = r7.join(userDF,$"id"===$"uid").select($"age")
    // count number of rows of travel based on the age criteria
    val x1 = r71.filter($"age" < 20).count()
    val x2=r71.filter($"age" > 35).count()
    val x3= r71.filter($"age" > 30 && $"age" <=35).count()
    println("Ans.7 age group < 20 =" + x1 + " age group > 35 =" + x2 +" age group
between 30 and 35 =" + x3)
```

```
    }


}
```