```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.types._

val schmhdays = StructType(Seq(
StructField("usrid", IntegerType, true),
StructField("fromLoc", StringType, true),
StructField("toLoc", StringType, true),
StructField("transport", StringType, true),
StructField("distance", IntegerType, true),
StructField("year", IntegerType, true)))
val dfhdays = spark.read.format("csv").option("header",
"false").schema(schmhdays).load("resources/Holidays.csv")

val schmusers = StructType(Seq(
StructField("usrid", IntegerType, true),
StructField("username", StringType, true),
StructField("age", IntegerType, true)))
val dfusers = spark.read.format("csv").option("header",
"false").schema(schmusers).load("resources/Users.csv")

val schmtransp = StructType(Seq(
StructField("transport", StringType, true),
StructField("fare", IntegerType, true)))
val dftransp = spark.read.format("csv").option("header",
"false").schema(schmtransp).load("resources/Transport.csv")


dfusers.createOrReplaceTempView("Vdfusers ")
val dfVusers = sql("select * from Vdfusers ")
dfVusers.show

dftransp.createOrReplaceTempView("Vdftransp")
val dfVtransp = sql("select * from Vdftransp")
dfVtransp.show

dfhdays.createOrReplaceTempView("Vdfhdays ")
val dfVhdays = sql("select * from Vdfhdays")
dfVhdays.show
```

Task 1

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

```
scala> dfhdays.show()
+-----+-------+-----+---------+--------+----+
|usrid|fromLoc|toLoc|transport|distance|year|
+-----+-------+-----+---------+--------+----+
|    1|    CHN|  IND| airplane|     200|1990|
|    2|    IND|  CHN| airplane|     200|1991|
|    3|    IND|  CHN| airplane|     200|1992|
|    4|    RUS|  IND| airplane|     200|1990|
|    5|    CHN|  RUS| airplane|     200|1992|
|    6|    AUS|  PAK| airplane|     200|1991|
|    7|    RUS|  AUS| airplane|     200|1990|
|    8|    IND|  RUS| airplane|     200|1991|
|    9|    CHN|  RUS| airplane|     200|1992|
|   10|    AUS|  CHN| airplane|     200|1993|
|    1|    AUS|  CHN| airplane|     200|1993|
|    2|    CHN|  IND| airplane|     200|1993|
|    3|    CHN|  IND| airplane|     200|1993|
|    4|    IND|  AUS| airplane|     200|1991|
|    5|    AUS|  IND| airplane|     200|1992|
|    6|    RUS|  CHN| airplane|     200|1993|
|    7|    CHN|  RUS| airplane|     200|1990|
|    8|    AUS|  CHN| airplane|     200|1990|
|    9|    IND|  AUS| airplane|     200|1991|
|   10|    RUS|  CHN| airplane|     200|1992|
+-----+-------+-----+---------+--------+----+
only showing top 20 rows

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

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

var dfjoinhdayspax = dfhdays.join(dfusers,"usrid")

```
scala> var dftotDistByYear = dfjoinhdayspax.groupBy("usrid", "username", "year").sum("distance").orderBy("usrid","year")
dftotDistByYear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [usrid: int, username: string ... 2 more fields]

scala> dftotDistByYear.show
+-----+--------+----+-------------+
|usrid|username|year|sum(distance)|
+-----+--------+----+-------------+
|    1|    mark|1990|          200|
|    1|    mark|1993|          600|
|    2|    john|1991|          400|
|    2|    john|1993|          200|
|    3|    luke|1991|          200|
|    3|    luke|1992|          200|
|    3|    luke|1993|          200|
|    4|    lisa|1990|          400|
|    4|    lisa|1991|          200|
|    5|    mark|1991|          200|
|    5|    mark|1992|          400|
|    5|    mark|1994|          200|
|    6|   peter|1991|          400|
|    6|   peter|1993|          200|
|    7|   james|1990|          600|
|    8|  andrew|1990|          200|
|    8|  andrew|1991|          200|
|    8|  andrew|1992|          200|
|    9|  thomas|1991|          200|
|    9|  thomas|1992|          400|
+-----+--------+----+-------------+
only showing top 20 rows
```

3) Which user has travelled the largest distance till date

```
scala> var dflargestDistusr = dfjoinhdayspax.groupBy("usrid", "username").sum("distance").orderBy("usrid")
dflargestDistusr: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [usrid: int, username: string ... 1 more field]

scala> dflargestDistusr.show
+-----+--------+-------------+
|usrid|username|sum(distance)|
+-----+--------+-------------+
|    1|    mark|          800|
|    2|    john|          600|
|    3|    luke|          600|
|    4|    lisa|          600|
|    5|    mark|          800|
|    6|   peter|          600|
|    7|   james|          600|
|    8|  andrew|          600|
|    9|  thomas|          600|
|   10|   annie|          600|
+-----+--------+-------------+
```

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

```
scala> var mostprefdest = dfmostprefDest.groupBy("toLoc").agg(max("count"))
mostprefdest: org.apache.spark.sql.DataFrame = [toLoc: string, max(count): bigint]

scala> mostprefdest.show
+-----+----------+
|toLoc|max(count)|
+-----+----------+
|  AUS|         1|
|  PAK|         2|
|  RUS|         1|
|  IND|         2|
|  CHN|         2|
+-----+----------+
```

5)Which route is generating the most revenue per year

```
scala> var dfjointransp = dfhdays.join(dftransp,"transport")
dfjointransp: org.apache.spark.sql.DataFrame = [transport: string, usrid: int ... 5 more fields]

scala> var dfmostrevenue = dfjointransp.groupBy("transport", "year").sum("fare")
dfmostrevenue: org.apache.spark.sql.DataFrame = [transport: string, year: int ... 1 more field]

scala> dfmost
dfmostprefDest   dfmostrevenue

scala> dfmostrevenue.show
+---------+----+---------+
|transport|year|sum(fare)|
+---------+----+---------+
| airplane|1993|     1190|
| airplane|1991|     1530|
| airplane|1990|     1360|
| airplane|1994|      170|
| airplane|1992|     1190|
+---------+----+---------+


scala> var dfmostrevenue = dfjointransp.groupBy("transport", "year").sum("fare") as "mostrevenue_year"
dfmostrevenue: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [transport: string, year: int ... 1 more field]

scala> dfmostrevenue.show
+---------+----+---------+
|transport|year|sum(fare)|
+---------+----+---------+
| airplane|1993|     1190|
| airplane|1991|     1530|
| airplane|1990|     1360|
| airplane|1994|      170|
| airplane|1992|     1190|
+---------+----+---------+
```

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

var dfjointranspuser = dfhdays.join(dftransp,"transport").join(dfusers,"usrid")

```
scala> var dfmospentuser = dfjointranspuser.groupBy("transport", "year","usrid","username").sum("fare") as "SpentByUser_year"
dfmospentuser: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [transport: string, year: int ... 3 more fields]

scala> dfmospentuser.show
+---------+----+-----+--------+---------+
|transport|year|usrid|username|sum(fare)|
+---------+----+-----+--------+---------+
| airplane|1993|    1|    mark|      510|
| airplane|1991|    2|    john|      340|
| airplane|1992|   10|   annie|      170|
| airplane|1992|    5|    mark|      340|
| airplane|1991|    3|    luke|      170|
| airplane|1990|    8|  andrew|      170|
| airplane|1991|    5|    mark|      170|
| airplane|1992|    9|  thomas|      340|
| airplane|1991|    4|    lisa|      170|
| airplane|1993|    2|    john|      170|
| airplane|1993|   10|   annie|      170|
| airplane|1991|    8|  andrew|      170|
| airplane|1991|    9|  thomas|      170|
| airplane|1990|    1|    mark|      170|
| airplane|1993|    6|   peter|      170|
| airplane|1991|    6|   peter|      340|
| airplane|1990|   10|   annie|      170|
| airplane|1990|    7|   james|      510|
| airplane|1994|    5|    mark|      170|
| airplane|1990|    4|    lisa|      340|
+---------+----+-----+--------+---------+
only showing top 20 rows
```

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

```
scala> dfagegroup.show
+-----+---------+-------+-----+--------+----+----+--------+---+
|usrid|transport|fromLoc|toLoc|distance|year|fare|username|age|
+-----+---------+-------+-----+--------+----+----+--------+---+
|    4| airplane|    RUS|  IND|     200|1990| 170|    lisa| 27|
|    5| airplane|    CHN|  RUS|     200|1992| 170|    mark| 25|
|    6| airplane|    AUS|  PAK|     200|1991| 170|   peter| 22|
|    7| airplane|    RUS|  AUS|     200|1990| 170|   james| 21|
|    4| airplane|    IND|  AUS|     200|1991| 170|    lisa| 27|
|    5| airplane|    AUS|  IND|     200|1992| 170|    mark| 25|
|    6| airplane|    RUS|  CHN|     200|1993| 170|   peter| 22|
|    7| airplane|    CHN|  RUS|     200|1990| 170|   james| 21|
|    4| airplane|    CHN|  PAK|     200|1990| 170|    lisa| 27|
|    5| airplane|    IND|  PAK|     200|1991| 170|    mark| 25|
|    6| airplane|    PAK|  RUS|     200|1991| 170|   peter| 22|
|    7| airplane|    CHN|  IND|     200|1990| 170|   james| 21|
|    5| airplane|    CHN|  PAK|     200|1994| 170|    mark| 25|
+-----+---------+-------+-----+--------+----+----+--------+---+


scala> dfagegroup.count()
res38: Long = 13

scala> dfjointransp
dfjointransp   dfjointranspuser

scala> dfjointranspuser.filter($"age" === 20).count()
res39: Long = 0

scala> dfjointranspuser.filter($"age" === 35).count()
res40: Long = 0

scala> dfjointranspuser.filter($"age" >= 35).count()
res41: Long = 9

scala> dfjointranspuser.filter($"age" <= 20).count()
res42: Long = 10
```