Assignment Day 20

Task 1:

1) What is the distribution of the total number of air-travellers 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?

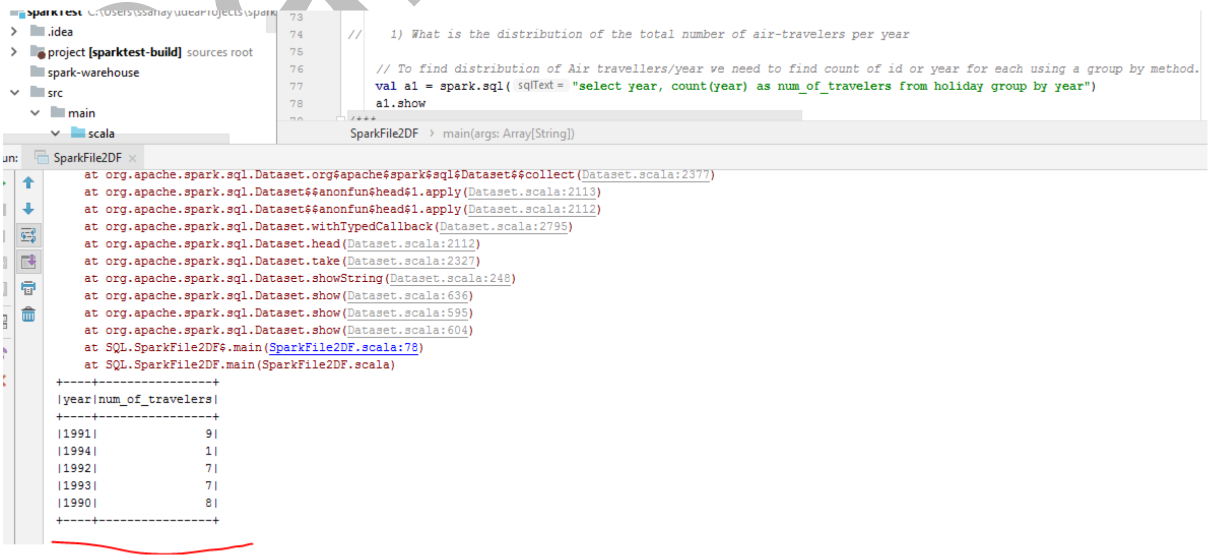
7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year?

Ans:

Note: Program files are properly documented for a detailed description of each instruction used within the program.

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

ScreenShot:



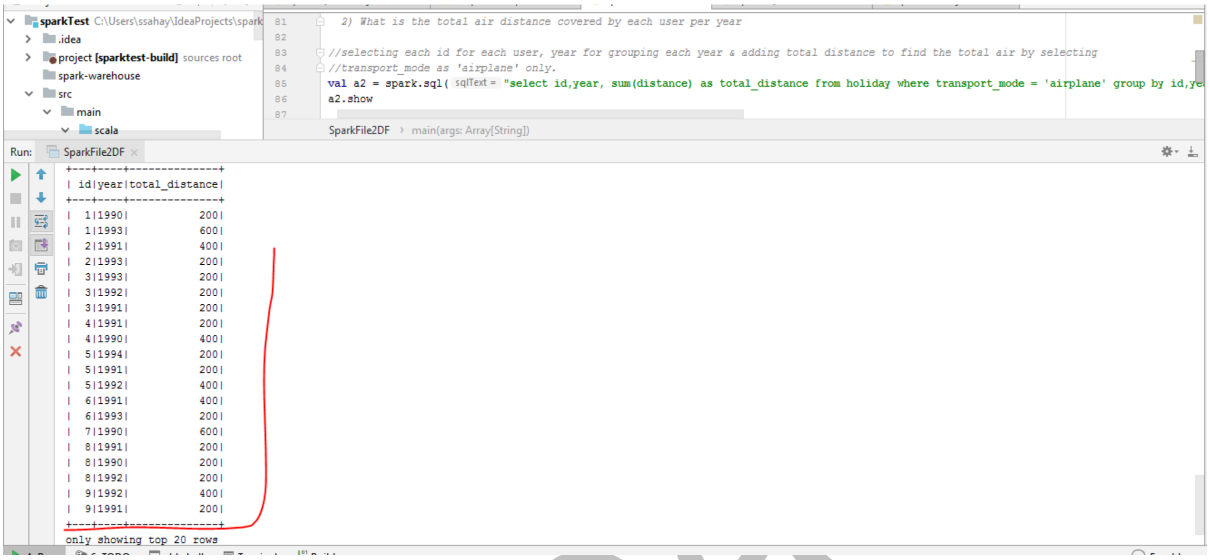
sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
SparkFile2DF
main(args: Array[String])
un:
SparkFile2DF

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

    // To find distribution of Air travellers/year we need to find count of id or year for each using a group by method.
    val a1 = spark.sql( sqlText = "select year, count(year) as num_of_travelers from holiday group by year")
    a1.show
```

```
	at org.apache.spark.sql.Dataset.org$apache$spark$sql$Dataset$$collect(Dataset.scala:2377)
	at org.apache.spark.sql.Dataset$$anonfun$head$1.apply(Dataset.scala:2113)
	at org.apache.spark.sql.Dataset$$anonfun$head$1.apply(Dataset.scala:2112)
	at org.apache.spark.sql.Dataset.withTypedCallback(Dataset.scala:2795)
	at org.apache.spark.sql.Dataset.head(Dataset.scala:2112)
	at org.apache.spark.sql.Dataset.take(Dataset.scala:2327)
	at org.apache.spark.sql.Dataset.showString(Dataset.scala:248)
	at org.apache.spark.sql.Dataset.show(Dataset.scala:636)
	at org.apache.spark.sql.Dataset.show(Dataset.scala:595)
	at org.apache.spark.sql.Dataset.show(Dataset.scala:604)
	at SQL.SparkFile2DF$.main(SparkFile2DF.scala:78)
	at SQL.SparkFile2DF.main(SparkFile2DF.scala)
+----+----------------+
|year|num_of_travelers|
+----+----------------+
|1991|               9|
|1994|               1|
|1992|               7|
|1993|               7|
|1990|               8|
+----+----------------+
```

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

ScreenShot:



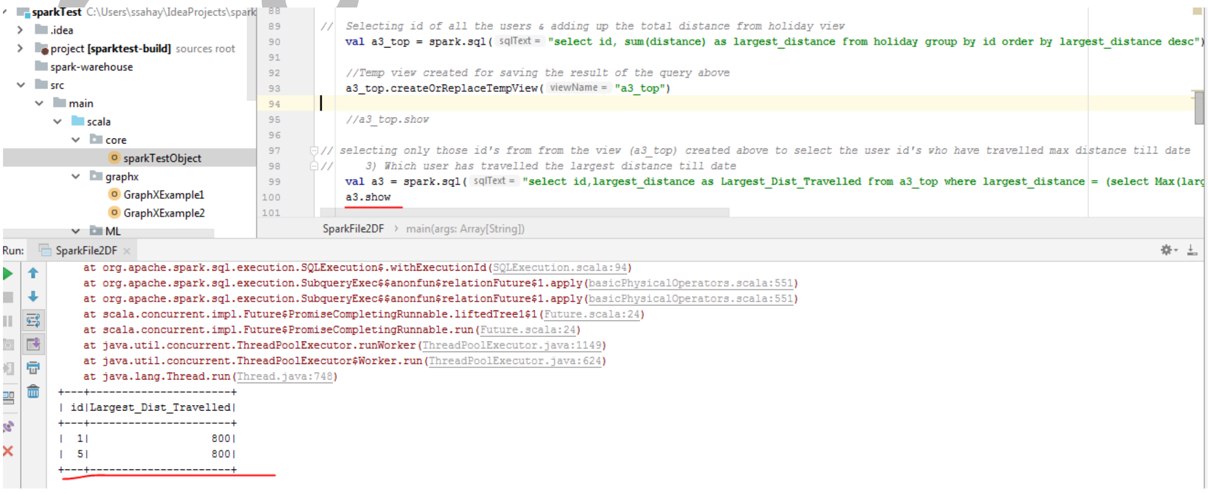
sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
SparkFile2DF › main(args: Array[String])
Run: SparkFile2DF

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

//selecting each id for each user, year for grouping each year & adding total distance to find the total air by selecting
//transport_mode as 'airplane' only.
val a2 = spark.sql( sqlText = "select id,year, sum(distance) as total_distance from holiday where transport_mode = 'airplane' group by id,ye
a2.show
```

```
+---+----+--------------+
| id|year|total_distance|
+---+----+--------------+
|  1|1990|           200|
|  1|1993|           600|
|  2|1991|           400|
|  2|1993|           200|
|  3|1993|           200|
|  3|1992|           200|
|  3|1991|           200|
|  4|1991|           200|
|  4|1990|           400|
|  5|1994|           200|
|  5|1991|           200|
|  5|1992|           400|
|  6|1991|           400|
|  6|1993|           200|
|  7|1990|           600|
|  8|1991|           200|
|  8|1990|           200|
|  8|1992|           200|
|  9|1992|           400|
|  9|1991|           200|
+---+----+--------------+
only showing top 20 rows
```

3) Which user has travelled the largest distance till date?

ScreenShot:



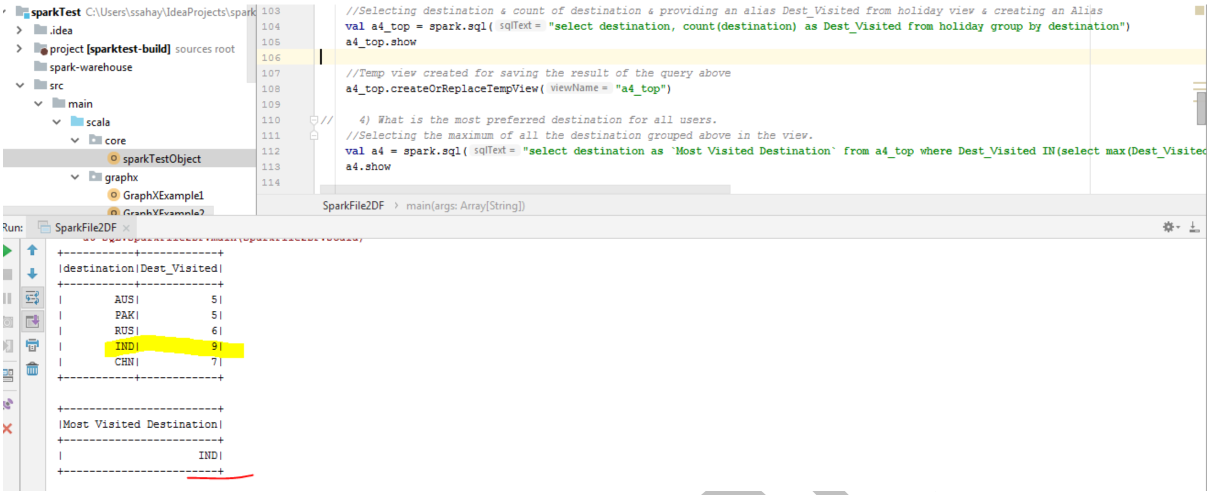
sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
core
sparkTestObject
graphx
GraphXExample1
GraphXExample2
ML
SparkFile2DF › main(args: Array[String])
Run: SparkFile2DF

```
//  Selecting id of all the users & adding up the total distance from holiday view
    val a3_top = spark.sql( sqlText = "select id, sum(distance) as largest_distance from holiday group by id order by largest_distance desc")

    //Temp view created for saving the result of the query above
    a3_top.createOrReplaceTempView( viewName = "a3_top")

    //a3_top.show

// selecting only those id's from from the view (a3_top) created above to select the user id's who have travelled max distance till date
//     3) Which user has travelled the largest distance till date
    val a3 = spark.sql( sqlText = "select id,largest_distance as Largest_Dist_Travelled from a3_top where largest_distance = (select Max(larg
    a3.show
```

```
	at org.apache.spark.sql.execution.SQLExecution$.withExecutionId(SQLExecution.scala:94)
	at org.apache.spark.sql.execution.SubqueryExec$$anonfun$relationFuture$1.apply(basicPhysicalOperators.scala:551)
	at org.apache.spark.sql.execution.SubqueryExec$$anonfun$relationFuture$1.apply(basicPhysicalOperators.scala:551)
	at scala.concurrent.impl.Future$PromiseCompletingRunnable.liftedTree1$1(Future.scala:24)
	at scala.concurrent.impl.Future$PromiseCompletingRunnable.run(Future.scala:24)
	at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
	at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
	at java.lang.Thread.run(Thread.java:748)
+---+--------------------+
| id|Largest_Dist_Travelled|
+---+--------------------+
|  1|                 800|
|  5|                 800|
+---+--------------------+
```

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

ScreenShot:

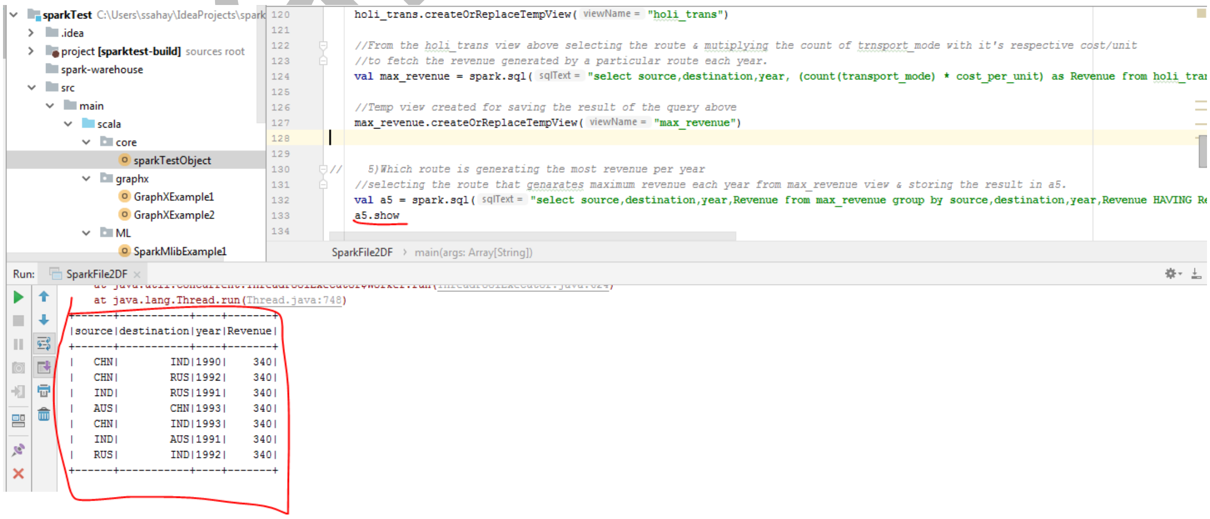

sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
core
sparkTestObject
graphx
GraphXExample1
GraphXExample2
SparkFile2DF › main(args: Array[String])
Run: SparkFile2DF

```
//Selecting destination & count of destination & providing an alias Dest_Visited from holiday view & creating an Alias
val a4_top = spark.sql( sqlText = "select destination, count(destination) as Dest_Visited from holiday group by destination")
a4_top.show

//Temp view created for saving the result of the query above
a4_top.createOrReplaceTempView( viewName = "a4_top")

//    4) What is the most preferred destination for all users.
//Selecting the maximum of all the destination grouped above in the view.
val a4 = spark.sql( sqlText = "select destination as `Most Visited Destination` from a4_top where Dest_Visited IN(select max(Dest_Visited
a4.show
```

```
+-----------+------------+
|destination|Dest_Visited|
+-----------+------------+
|        AUS|           5|
|        PAK|           5|
|        RUS|           6|
|        IND|           9|
|        CHN|           7|
+-----------+------------+

+------------------------+
|Most Visited Destination|
+------------------------+
|                     IND|
+------------------------+
```

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

ScreenShot:

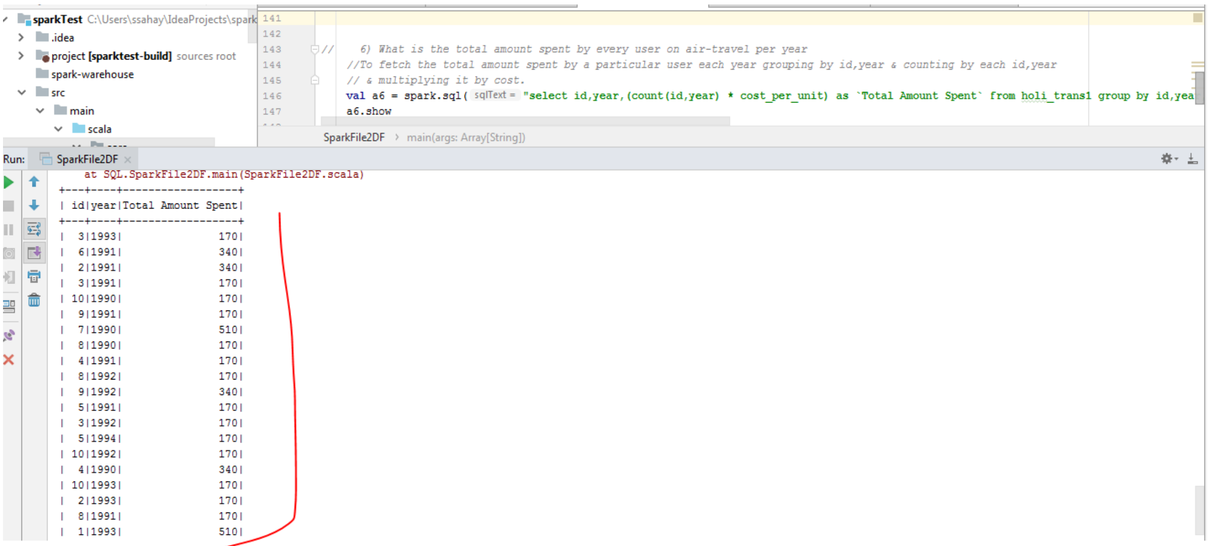

sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
core
sparkTestObject
graphx
GraphXExample1
GraphXExample2
ML
SparkMlibExample1
SparkFile2DF › main(args: Array[String])
Run: SparkFile2DF

```
holi_trans.createOrReplaceTempView( viewName = "holi_trans")

//From the holi_trans view above selecting the route & mutiplying the count of trnsport_mode with it's respective cost/unit
//to fetch the revenue generated by a particular route each year.
val max_revenue = spark.sql( sqlText = "select source,destination,year, (count(transport_mode) * cost_per_unit) as Revenue from holi_tra

//Temp view created for saving the result of the query above
max_revenue.createOrReplaceTempView( viewName = "max_revenue")


//    5)Which route is generating the most revenue per year
//selecting the route that genarates maximum revenue each year from max_revenue view & storing the result in a5.
val a5 = spark.sql( sqlText = "select source,destination,year,Revenue from max_revenue group by source,destination,year,Revenue HAVING Re
a5.show
```

```
    at java.lang.Thread.run(Thread.java:748)
+------+-----------+----+-------+
|source|destination|year|Revenue|
+------+-----------+----+-------+
|   CHN|        IND|1990|    340|
|   CHN|        RUS|1992|    340|
|   IND|        RUS|1991|    340|
|   AUS|        CHN|1993|    340|
|   CHN|        IND|1993|    340|
|   IND|        AUS|1991|    340|
|   RUS|        IND|1992|    340|
+------+-----------+----+-------+
```

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

ScreenShot:

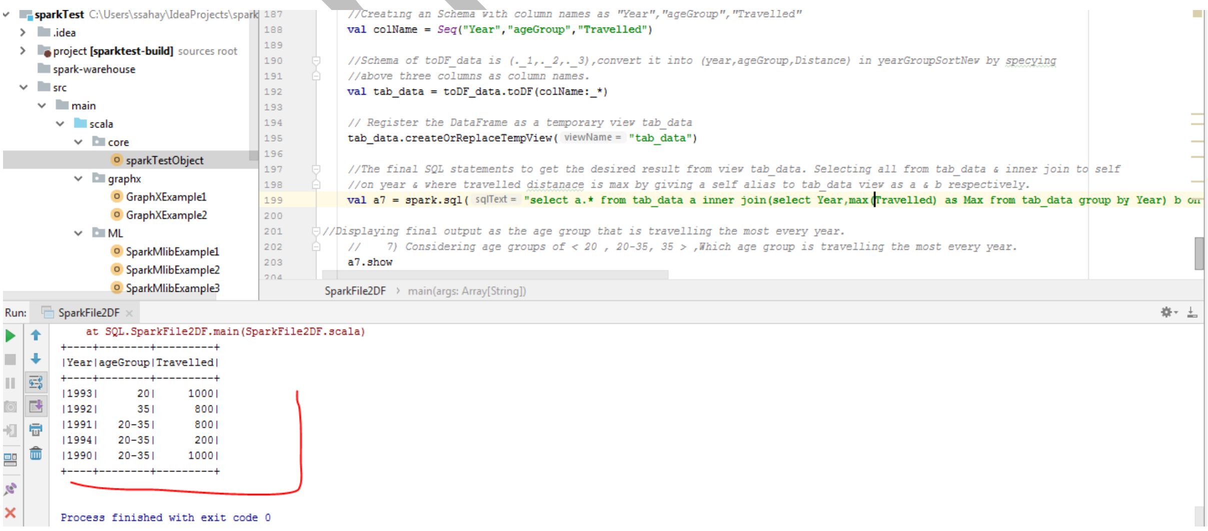

sparkTest C:\Users\ssahay\IdeaProjects\spark
.idea
project [sparktest-build] sources root
spark-warehouse
src
main
scala
SparkFile2DF › main(args: Array[String])
Run: SparkFile2DF

```
//    6) What is the total amount spent by every user on air-travel per year
//To fetch the total amount spent by a particular user each year grouping by id,year & counting by each id,year
// & multiplying it by cost.
val a6 = spark.sql( sqlText = "select id,year,(count(id,year) * cost_per_unit) as `Total Amount Spent` from holi_trans1 group by id,yea
a6.show
```

```
    at SQL.SparkFile2DF.main(SparkFile2DF.scala)
+---+----+------------------+
| id|year|Total Amount Spent|
+---+----+------------------+
|  3|1993|               170|
|  6|1991|               340|
|  2|1991|               340|
|  3|1991|               170|
| 10|1990|               170|
|  9|1991|               170|
|  7|1990|               510|
|  8|1990|               170|
|  4|1991|               170|
|  8|1992|               170|
|  9|1992|               340|
|  5|1991|               170|
|  3|1992|               170|
|  5|1994|               170|
| 10|1992|               170|
|  4|1990|               340|
| 10|1993|               170|
|  2|1993|               170|
|  8|1991|               170|
|  1|1993|               510|
```

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

ScreenShot:



```
//Creating an Schema with column names as "Year","ageGroup","Travelled"
val colName = Seq("Year","ageGroup","Travelled")

//Schema of toDF_data is (._1,._2,._3),convert it into (year,ageGroup,Distance) in yearGroupSortNew by specying
//above three columns as column names.
val tab_data = toDF_data.toDF(colName:_*)

// Register the DataFrame as a temporary view tab_data
tab_data.createOrReplaceTempView( viewName = "tab_data")

//The final SQL statements to get the desired result from view tab_data. Selecting all from tab_data & inner join to self
//on year & where travelled distanace is max by giving a self alias to tab_data view as a & b respectively.
val a7 = spark.sql( sqlText = "select a.* from tab_data a inner join(select Year,max(Travelled) as Max from tab_data group by Year) b on

//Displaying final output as the age group that is travelling the most every year.
//     7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year.
a7.show
```

```
    at SQL.SparkFile2DF.main(SparkFile2DF.scala)
+----+--------+---------+
|Year|ageGroup|Travelled|
+----+--------+---------+
|1993|      20|     1000|
|1992|      35|      800|
|1991|   20-35|      800|
|1994|   20-35|      200|
|1990|   20-35|     1000|
+----+--------+---------+


Process finished with exit code 0
```

******************************** **End** ********************************