ASSIGNMENT 18.2

Input Datasets:

We have an airline data with us:

user details:

user_id, name, age

holidays:

user_id, src, dest, travel_mode, distance, year_of_travel

transport:

travel_mode, cost_per_unit

Here are the datasets which we will be using for this assignment in all problems. It has been kept in local file system:

```
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
 10
      1, AUS, CHN, airplane, 200, 1993
 11
 12
      2, CHN, IND, airplane, 200, 1993
 13
      3,CHN,IND,airplane,200,1993
 14
      4, IND, AUS, airplane, 200, 1991
 15
      5, AUS, IND, airplane, 200, 1992
 16
      6, RUS, CHN, airplane, 200, 1993
 17
      7, CHN, RUS, airplane, 200, 1990
 18
      8, AUS, CHN, airplane, 200, 1990
 19
      9, IND, AUS, airplane, 200, 1991
 20
      10, RUS, CHN, airplane, 200, 1992
 21
      1, PAK, IND, airplane, 200, 1993
 22
      2, IND, RUS, airplane, 200, 1991
 23
      3, CHN, PAK, airplane, 200, 1991
      4, CHN, PAK, airplane, 200, 1990
      5, IND, PAK, airplane, 200, 1991
 26
      6, PAK, RUS, airplane, 200, 1991
       7, CHN, IND, airplane, 200, 1990
 28
      8, RUS, IND, airplane, 200, 1992
      9, RUS, IND, airplane, 200, 1992
      10, CHN, AUS, airplane, 200, 1990
 31
      1, PAK, AUS, airplane, 200, 1993
      5, CHN, PAK, airplane, 200, 1994
```

```
S18 Dataset Holidays.txt 🗵 🔚 S18 Dataset Transport.txt 🔀
     airplane,170
     car, 140
     train, 120
     ship,200
S18_Dataset_Holidays.txt 🗵 📙 S18_Dataset_Transport.txt 🗵 블 S18_Dataset_User_details.txt 🗵
     1, mark, 15
     2, john, 16
     3, luke, 17
     4, lisa, 27
    5, mark, 25
     6, peter, 22
 7
     7, james, 21
     8, andrew, 55
     9,thomas,46
10 10, annie, 44
```

Problem Statement:

- 1) Which route is generating the most revenue per year?
- 2) What is the total amount spent by every user on air-travel per year?
- 3) Considering age groups of <20, 20-35, 35>, which age group is travelling the most every year?

Solution:

1. Here is the Spark code snippet to find the route generating the most revenue per year:

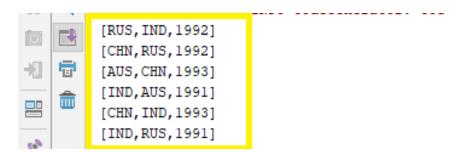
```
.config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
   .master("local[*]")
   .getOrCreate()
  val sqlContext = spark.sqlContext
                                                          // initialize sqlContext
  val input_df1 = sqlContext.read.csv ("E:\\Acadgild\\Session 18\\S18_Datasets \\\S18_Dataset
_Transport.txt")
                                                           // load input data file – transport.txt
  val transportDF = input_df1.select(
                                                           // define schema for input data loaded
   input_df1("_c0").cast(StringType).as("travel_mode"), //assign column names to the data frame
   input_df1("_c1").cast(IntegerType).as("cost_per_unit"))
  transportDF.createOrReplaceTempView("transport")
                                                          // create a temporary view - transport
  val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
Holidays.txt")
                                                           // load input data file – holidays.txt
  val holidaysDF = input_df2.select(
                                                           // define schema for input data loaded
   input df2(" c0").cast(IntegerType).as("user id"),
                                                          //assign column names to the data frame
   input_df2("_c1").cast(StringType).as("src"),
   input_df2("_c2").cast(StringType).as("dest"),
   input_df2("_c3").cast(StringType).as("travel_mode"),
   input_df2("_c4").cast(IntegerType).as("distance"),
   input_df2("_c5").cast(IntegerType).as("year_of_travel"))
  holidaysDF.createOrReplaceTempView("holidays")
                                                          // create a temporary view - holidays
// SQL query to find the route generating the most revenue per year
  val result = sqlContext.sql("SELECT y.src, y.dest, y.year_of_travel " +
                   "FROM transport x, holidays y " +
                   "WHERE x.travel_mode = y.travel_mode " +
                   "GROUP BY y.src , y.dest ,y.year_of_travel " +
                   "ORDER BY SUM(x.cost_per_unit) DESC").take(6)
  result.foreach(println)
                                                          // print the result
 }
}
```

```
    Assignment18_1.scala ×

    Assignment18_2.scala ×

        import org.apache.spark.sql.SparkSession
       import org.apache.spark.sql.types.{IntegerType, StringType}
5
   •
      object Assignment18 2 {
7
      def main(args: Array[String]): Unit = {
           val spark = SparkSession
8
             .builder()
9
             .config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
             .master("local[*]")
             .getOrCreate()
13
           val sqlContext = spark.sqlContext
           val input_dfl = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset Transport.txt")
14
           val transportDF = input_dfl.select(
15
16
            input_dfl(" c0").cast(StringType).as("travel mode"),
17
            input_dfl(" c1").cast(IntegerType).as("cost per unit"))
18
           transportDF.createOrReplaceTempView("transport")
19
           val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset Holidays.txt")
20
           val holidaysDF = input_df2.select(
21
             input_df2("_c0").cast(IntegerType).as("user_id"),
             input_df2(" c1").cast(StringType).as("src"),
24
             input_df2("_c2").cast(StringType).as("dest"),
             input_df2(" c3").cast(StringType).as("travel mode"),
             input_df2("_c4").cast(IntegerType).as("distance"),
26
27
             input_df2(" c5").cast(IntegerType).as("year of travel"))
           holidaysDF.createOrReplaceTempView("holidays")
29
30
            val result = sqlContext.sql("SELECT y.src, y.dest, y.year_of_travel " +
                                        "FROM transport x, holidays y " +
31
                                        "WHERE x.travel mode = y.travel mode " +
32
33
                                        "GROUP BY y.src , y.dest ,y.year_of_travel " +
                                        "ORDER BY SUM(x.cost per unit) DESC").take(6)
34
35
            result.foreach(println)
```

Output:



2. Here is the Spark code to find the total amount spent by every user on air-travel per year:

// import required Spark packages

import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.types.{IntegerType, StringType}

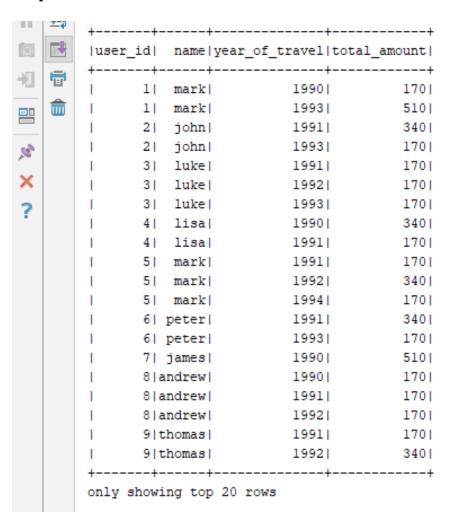
```
object Assignment18_2 {
 def main(args: Array[String]): Unit = {
  val spark = SparkSession
                                            // create a SparkSession object that can be used to
   .builder()
                                            // create various contexts of Spark such as sqlContext
   .config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
   .master("local[*]")
   .getOrCreate()
  val sqlContext = spark.sqlContext
                                                           // initialize sqlContext
  val input_df1 = sqlContext.read.csv ("E:\\Acadgild\\Session 18\\S18_Datasets \\S18_Dataset
Transport.txt")
                                                           // load input data file – transport.txt
  val transportDF = input_df1.select(
                                                           // define schema for input data loaded
   input_df1("_c0").cast(StringType).as("travel_mode"), //assign column names to the data frame
   input_df1("_c1").cast(IntegerType).as("cost_per_unit"))
  transportDF.createOrReplaceTempView("transport")
                                                           // create a temporary view - transport
  val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
Holidays.txt")
                                                           // load input data file – holidays.txt
  val holidaysDF = input_df2.select(
                                                           // define schema for input data loaded
   input_df2("_c0").cast(IntegerType).as("user_id"),
                                                           //assign column names to the data frame
   input_df2("_c1").cast(StringType).as("src"),
   input_df2("_c2").cast(StringType).as("dest"),
   input_df2("_c3").cast(StringType).as("travel_mode"),
   input_df2("_c4").cast(IntegerType).as("distance"),
   input_df2("_c5").cast(IntegerType).as("year_of_travel"))
  holidaysDF.createOrReplaceTempView("holidays")
                                                           // create a temporary view - holidays
val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\\S18_Datasets\\\S18_Dataset_
User_details.txt")
                                                           // load input data file – user_details.txt
  val usersDF = input_df3.select(
                                                           // define schema for input data loaded
   input_df3("_c0").cast(IntegerType).as("user_id"),
                                                           //assign column names to the data frame
   input_df3("_c1").cast(StringType).as("name"),
   input_df3("_c2").cast(IntegerType).as("age"))
```

```
    Assignment18_1.scala ×

    Assignment18_2.scala ×

13
            val sqlContext = spark.sqlContext
            val input dfl = sqlContext.read.csv("E:\\Acadqild\\Session 18\\S18 Datasets\\S18 Dataset Transport.txt")
14
15
            val transportDF = input dfl.select(
            input_dfl("_c0").cast(StringType).as("travel_mode"),
16
17
            input_dfl(" c1").cast(IntegerType).as("cost per unit"))
            transportDF.createOrReplaceTempView("transport")
18
19
            val input df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset Holidays.txt")
20
            val holidaysDF = input_df2.select(
            input_df2(" c0").cast(IntegerType).as("user id"),
             input_df2("_c1").cast(StringType).as("src"),
             input_df2(" c2").cast(StringType).as("dest"),
24
             input_df2("_c3").cast(StringType).as("travel_mode"),
26
              input_df2(" c4").cast(IntegerType).as("distance"),
              input_df2(" c5").cast(IntegerType).as("year of travel"))
27
            holidaysDF.createOrReplaceTempView("holidays")
28
29
            val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset User details.txt")
30
            val usersDF = input_df3.select(
31
            input_df3("_c0").cast(IntegerType).as("user_id"),
32
33
             input_df3("_c1").cast(StringType).as("name"),
         input_df3(" c2").cast(IntegerType).as("age"))
34
35
            usersDF.createOrReplaceTempView("users")
36
37
            sqlContext.sql("SELECT z.user_id, z.name, y.year_of_travel, SUM(x.cost_per_unit) as total_amount " +
                           "FROM transport x, holidays y ,users z " +
                           "WHERE x.travel mode = y.travel mode AND x.travel mode = 'airplane' and y.user id = z.user id " +
39
40
                           "GROUP BY z.user_id, z.name, y.year_of_travel " +
                           "ORDER by z.user id, z.name, y.year of travel").show()
41
42
43
```

Output:



3. Here is the Spark code to find the age group that is travelling the most every year:

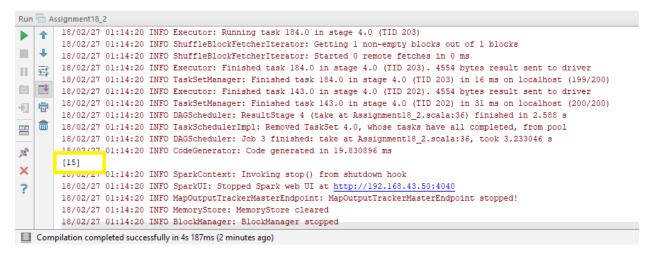
```
.master("local[*]")
   .getOrCreate()
  val sqlContext = spark.sqlContext
                                                          // initialize sqlContext
  val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
Holidays.txt")
                                                          // load input data file – holidays.txt
  val holidaysDF = input_df2.select(
                                                          // define schema for input data loaded
   input_df2("_c0").cast(IntegerType).as("user_id"),
                                                          //assign column names to the data frame
   input_df2("_c1").cast(StringType).as("src"),
   input_df2("_c2").cast(StringType).as("dest"),
   input_df2("_c3").cast(StringType).as("travel_mode"),
   input_df2("_c4").cast(IntegerType).as("distance"),
   input_df2("_c5").cast(IntegerType).as("year_of_travel"))
  holidaysDF.createOrReplaceTempView("holidays")
                                                          // create a temporary view - holidays
val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
User details.txt")
                                                          // load input data file – user details.txt
  val usersDF = input_df3.select(
                                                          // define schema for input data loaded
   input_df3("_c0").cast(IntegerType).as("user_id"),
                                                          //assign column names to the data frame
   input_df3("_c1").cast(StringType).as("name"),
   input_df3("_c2").cast(IntegerType).as("age"))
  usersDF.createOrReplaceTempView("users")
                                                          // create a temporary view – users
// SQL query to find the age group that is travelling the most every year
   val result = sqlContext.sql("SELECT age " +
                   "FROM users x, holidays y " +
                   "WHERE x.user_id = y.user_id " +
                   "GROUP BY age " +
                   "ORDER BY COUNT(age) DESC").take(1)
                                                          // print the result
  result.foreach(println)
 }
```

```
Assignment18_1.scala ×

    Assignment18_2.scala

6
       object Assignment18 2 {
7
          def main(args: Array[String]): Unit = {
8
            val spark = SparkSession
9
              .builder()
              .config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
             .master("local[*]")
11
12
             .getOrCreate()
13
           val sqlContext = spark.sqlContext
14
            val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset Holidays.txt")
            val holidaysDF = input_df2.select(
16
17
             input_df2("_c0").cast(IntegerType).as("user_id"),
             input_df2("_c1").cast(StringType).as("src"),
19
             input_df2("_c2").cast(StringType).as("dest"),
             input_df2("_c3").cast(StringType).as("travel_mode"),
             input_df2("_c4").cast(IntegerType).as("distance"),
             input_df2("_c5").cast(IntegerType).as("year_of_travel"))
           holidaysDF.createOrReplaceTempView("holidays")
23
24
            val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18 Datasets\\S18 Dataset User details.txt")
            val usersDF = input_df3.select(
             input df3(" c0").cast(IntegerType).as("user id"),
             input df3(" c1").cast(StringType).as("name"),
28
             input_df3(" c2").cast(IntegerType).as("age"))
29
30
            usersDF.createOrReplaceTempView("users")
31
32
           val result = sqlContext.sql("SELECT age " +
33
                                         "FROM users x, holidays y " +
34
                                         "WHERE x.user id = y.user id " +
35
                                        "GROUP BY age " +
                                        "ORDER BY COUNT(age) DESC").take(1)
            result.foreach(println)
```

Output:



Conclusion: People who are under the age of 20 (to be precise, people with age 15) travels the most every year.