ASSIGNMENT 18.2

**Input Datasets:**

We have an airline data with us:

**user details:**

user\_id, name, age

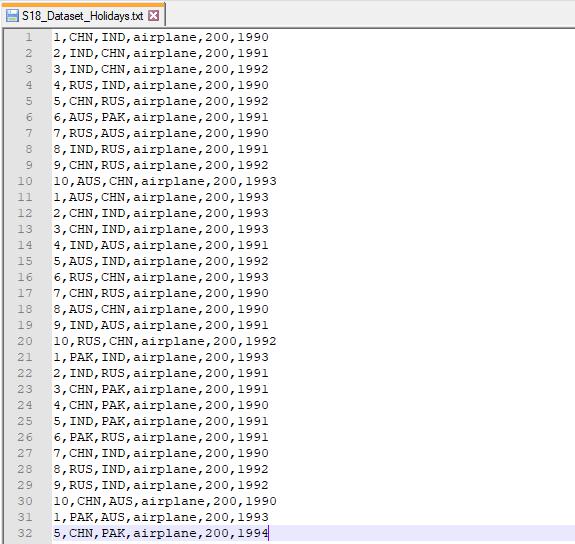
**holidays:**

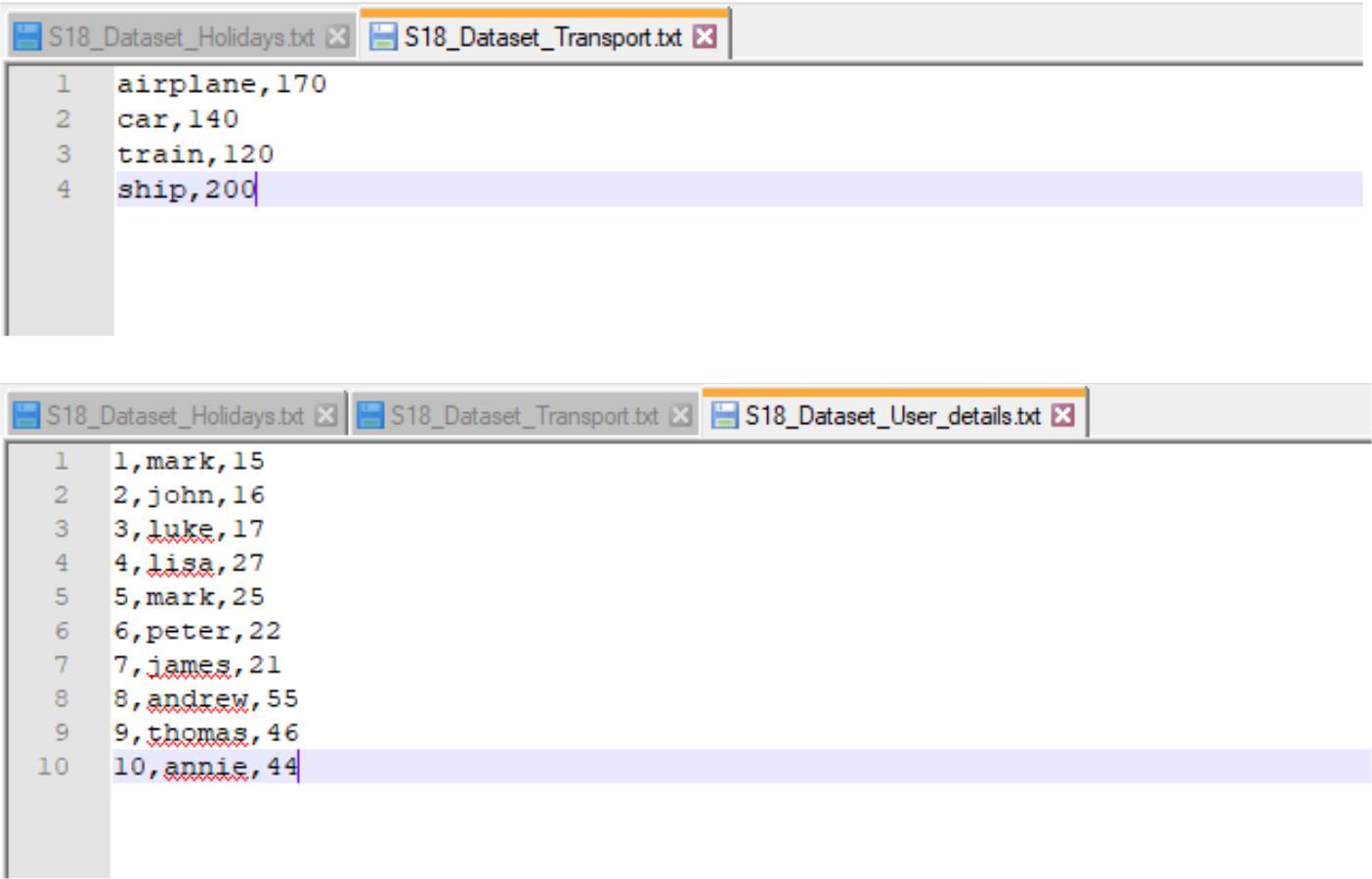
user\_id, src, dest, travel\_mode, distance, year\_of\_travel

**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:





**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: // 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

.builder()

// create a SparkSession object that can be used to

// 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")

val holidaysDF = input\_df2.select(

input\_df2("\_c0").cast(IntegerType).as("user\_id"),

input\_df2("\_c1").cast(StringType).as("src"),

// load input data file – holidays.txt

// define schema for input data loaded

//assign column names to the data frame

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

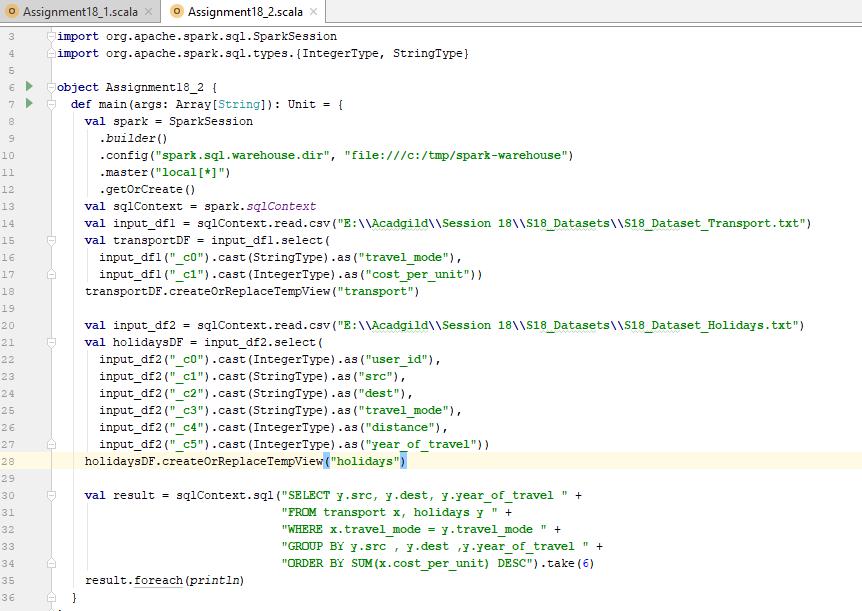
val result = sqlContext.sql("SELECT y.src, y.dest, y.year\_of\_travel " +

"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)

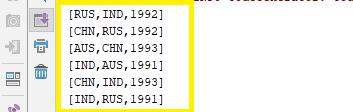
result.foreach(println) // print the result

}

}



**Output:**



1. 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")

val usersDF = input\_df3.select(

input\_df3("\_c0").cast(IntegerType).as("user\_id"),

input\_df3("\_c1").cast(StringType).as("name"),

input\_df3("\_c2").cast(IntegerType).as("age"))

// load input data file – user\_details.txt // define schema for input data loaded //assign column names to the data frame

usersDF.createOrReplaceTempView("users") // create a temporary view - users

* SQL query to find the total amount spent by every user on air-travel per year 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 " +

"GROUP BY z.user\_id, z.name, y.year\_of\_travel " +

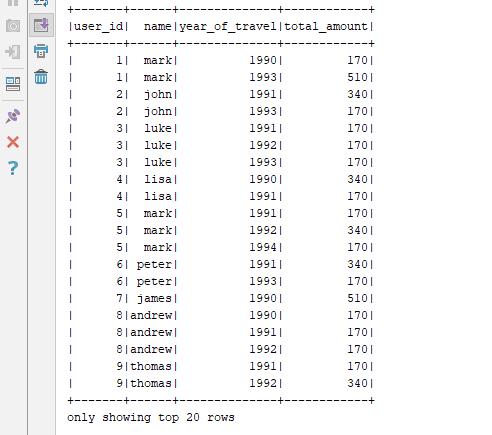
"ORDER by z.user\_id, z.name, y.year\_of\_travel").show() // print the result

}

}



**Output:**



1. Here is the Spark code to find the age group that is travelling the most every 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\_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18\_Datasets\\S18\_Dataset\_

Holidays.txt")

val holidaysDF = input\_df2.select(

input\_df2("\_c0").cast(IntegerType).as("user\_id"),

input\_df2("\_c1").cast(StringType).as("src"),

// load input data file – holidays.txt

// define schema for input data loaded

//assign column names to the data frame

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")

val usersDF = input\_df3.select(

input\_df3("\_c0").cast(IntegerType).as("user\_id"),

// load input data file – user\_details.txt // define schema for input data loaded //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 " +

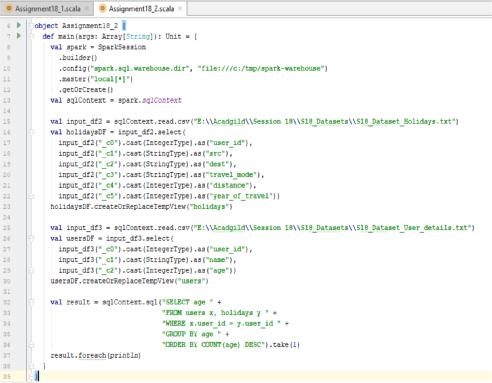
"FROM users x, holidays y " + "WHERE x.user\_id = y.user\_id " + "GROUP BY age " +

"ORDER BY COUNT(age) DESC").take(1)

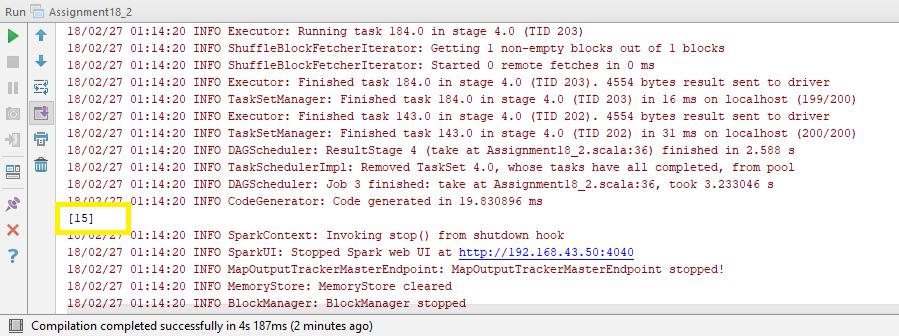
result.foreach(println) // print the result

}

}



**Output:**



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