ASSIGNMENT 18.1

**Input Datasets:**

We have an airline data with us:

**user details:**

user\_id, name, age

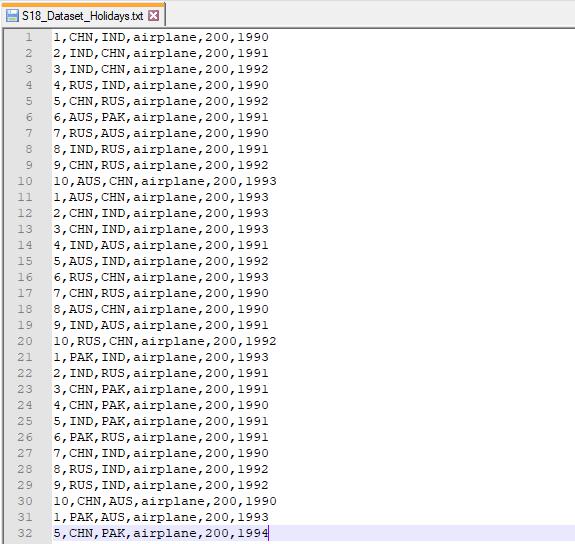
**holidays:**

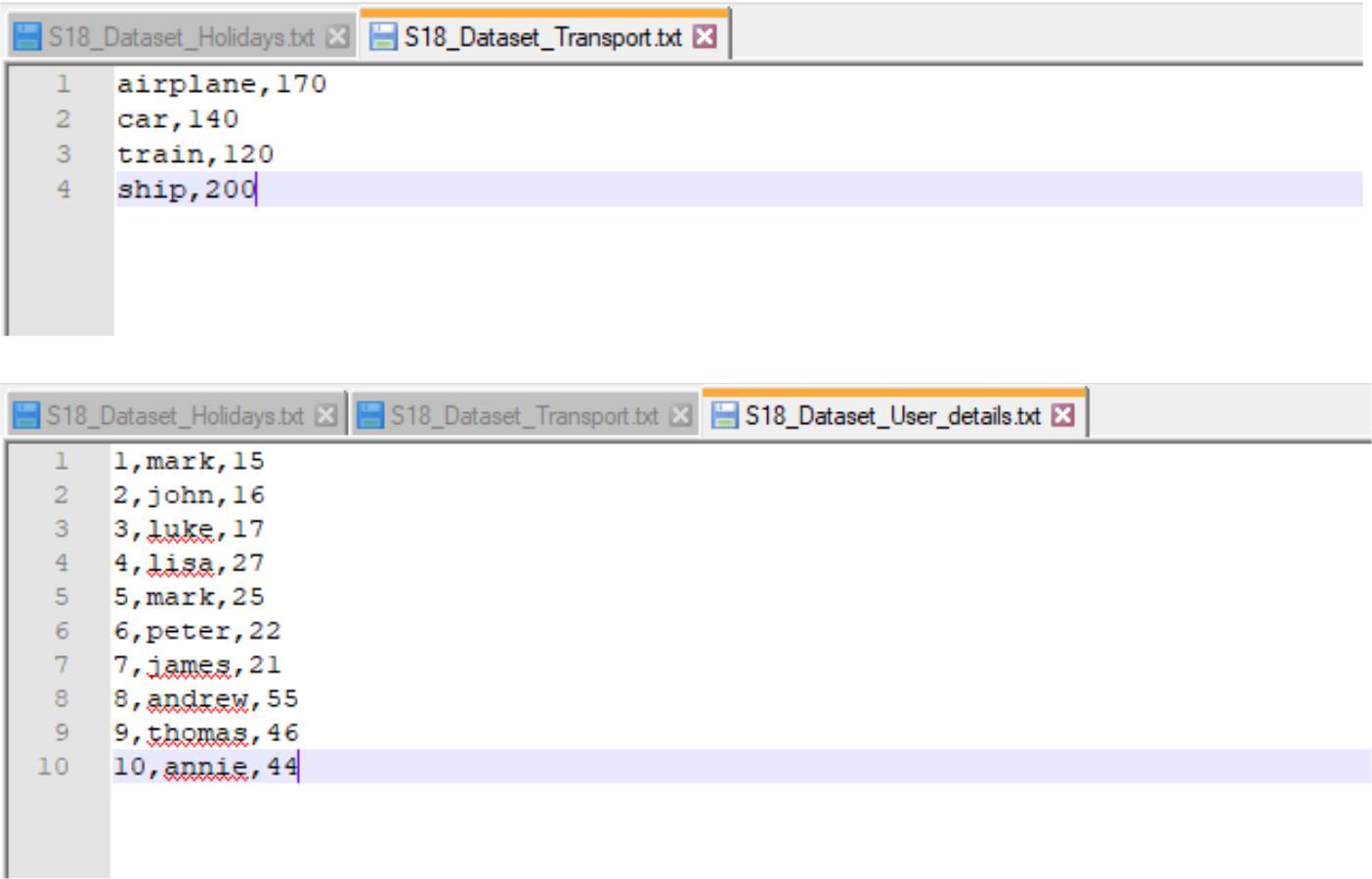
user\_id, src, dest, travel\_mode, distance, year\_of\_travel

**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. 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?

**Solution:**

1. Here is the Spark code snippet to find distribution of total number of air-travelers per year: // import required Spark packages

import org.apache.spark.sql.SparkSession

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

object Assignment18\_1 {

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\_df = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18\_Datasets\\S18\_ Dataset\_

Holidays.txt") // load input data file – holidays.txt

val holidaysDF = input\_df.select ( // define schema for input data loaded

input\_df("\_c0").cast(IntegerType).as("user\_id"), //assign column names to the data frame

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

input\_df("\_c2").cast(StringType).as("dest"),

input\_df("\_c3").cast(StringType).as("travel\_mode"),

input\_df("\_c4").cast(IntegerType).as("distance"),

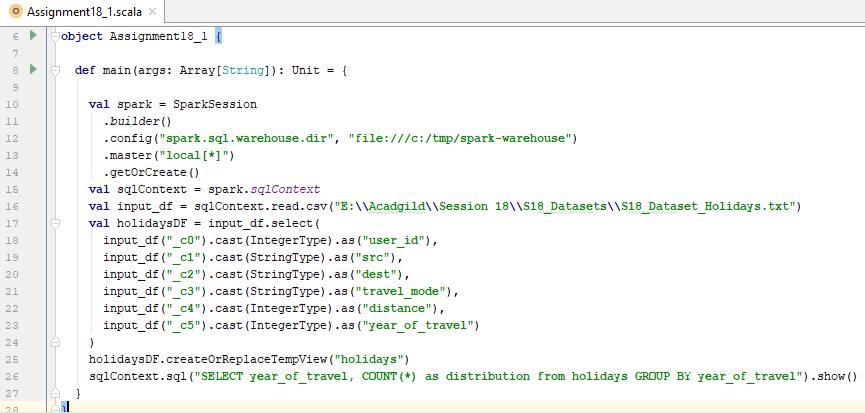
input\_df("\_c5").cast(IntegerType).as("year\_of\_travel"))

holidaysDF.createOrReplaceTempView("holidays") // create a temporary view - holidays

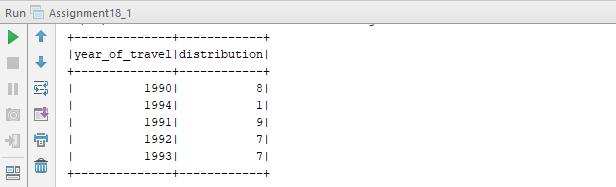
sqlContext.sql("SELECT year\_of\_travel, COUNT (\*) as distribution from holidays GROUP

BY year\_of\_travel").show() // SQL query to show distribution per year }

}



**Output:**



1. Here is the Spark code snippet to find total air distance covered by each user per year: // import required Spark packages

import org.apache.spark.sql.SparkSession

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

object Assignment18\_1 {

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\_df = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18\_Datasets\\S18\_ Dataset\_

Holidays.txt") // load input data file – holidays.txt

val holidaysDF = input\_df.select ( // define schema for input data loaded

input\_df("\_c0").cast(IntegerType).as("user\_id"), //assign column names to the data frame

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

input\_df("\_c2").cast(StringType).as("dest"),

input\_df("\_c3").cast(StringType).as("travel\_mode"),

input\_df("\_c4").cast(IntegerType).as("distance"),

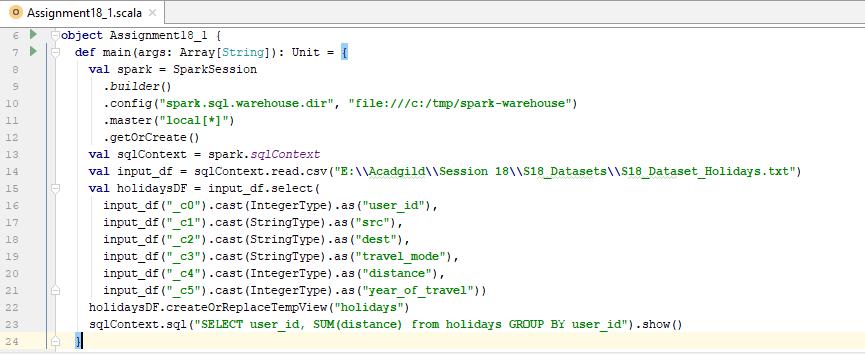
input\_df("\_c5").cast(IntegerType).as("year\_of\_travel"))

holidaysDF.createOrReplaceTempView("holidays") // create a temporary view - holidays

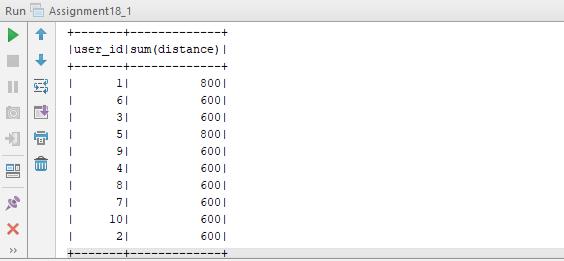
sqlContext.sql("SELECT user\_id, SUM(distance) from holidays GROUP BY user\_id").show() // SQL query to show distance covered by each user

}

}



**Output:**



1. Here is the Spark code snippet to find which user has travelled the largest distance till date: // import required Spark packages

import org.apache.spark.sql.SparkSession

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

object Assignment18\_1 {

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\_df = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18\_Datasets\\S18\_ Dataset\_

Holidays.txt") // load input data file – holidays.txt val holidaysDF = input\_df.select ( // define schema for input data loaded input\_df("\_c0").cast(IntegerType).as("user\_id"), //assign column names to the data frame input\_df("\_c1").cast(StringType).as("src"),

input\_df("\_c2").cast(StringType).as("dest"),

input\_df("\_c3").cast(StringType).as("travel\_mode"),

input\_df("\_c4").cast(IntegerType).as("distance"),

input\_df("\_c5").cast(IntegerType).as("year\_of\_travel"))

holidaysDF.createOrReplaceTempView("holidays") // create a temporary view - holidays

val result = sqlContext.sql("SELECT user\_id, SUM(distance) AS total\_distance " +

"FROM holidays " +

"GROUP BY user\_id " +

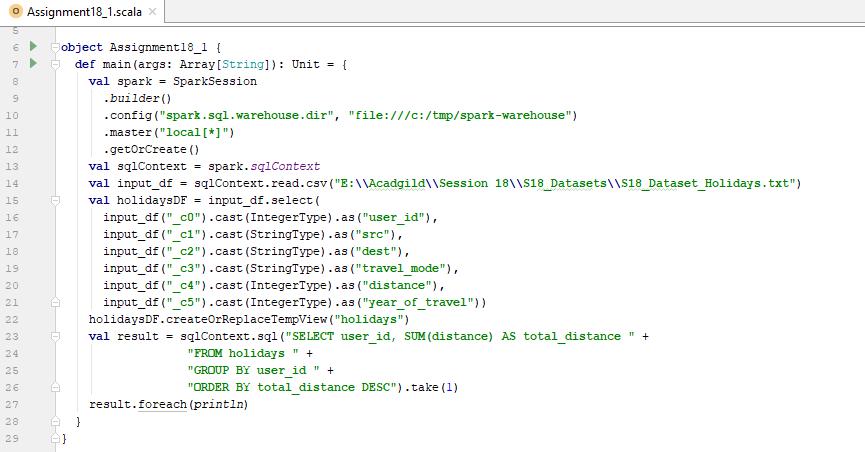
"ORDER BY total\_distance DESC").take(1)

// SQL query to find the user who covered largest distance

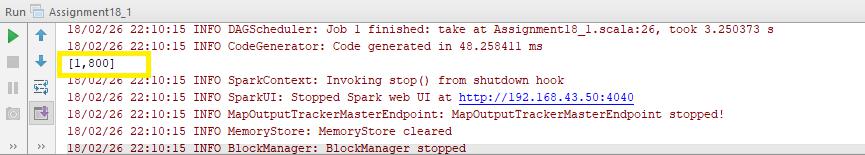
result.foreach(println) // print the result

}

}



**Output:**



We can conclude from the above output that the user with id **1** has covered the largest distance of **800 kms** till date.

1. Here is the Spark code snippet to find the most preferred destination for all users: // import required Spark packages

import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.types.{IntegerType, StringType} object Assignment18\_1 {

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\_df = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18\_Datasets\\S18\_ Dataset\_

Holidays.txt") // load input data file – holidays.txt

val holidaysDF = input\_df.select ( // define schema for input data loaded

input\_df("\_c0").cast(IntegerType).as("user\_id"), //assign column names to the data frame

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

input\_df("\_c2").cast(StringType).as("dest"),

input\_df("\_c3").cast(StringType).as("travel\_mode"),

input\_df("\_c4").cast(IntegerType).as("distance"),

input\_df("\_c5").cast(IntegerType).as("year\_of\_travel"))

holidaysDF.createOrReplaceTempView("holidays") // create a temporary view - holidays

val result = sqlContext.sql("SELECT dest, COUNT(\*) AS distribution " +

"FROM holidays " +

"GROUP BY dest " + // SQL query to find the most

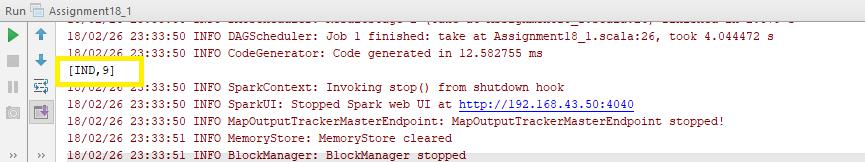
"ORDER BY distribution DESC").take(1) // preferred destination for all users

result.foreach(println) // print the result

}

}

**Output:**



We can conclude from the above output that **India (IND)** is the most preferred destination among all **9** users.

**Spark code screenshot:**

