Vijay's Assignment – Spark SQL 1

Task 1

Common areas of the program to declare the SparkConf and SparkSession

```
package vksp1

import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.types._
object vksprksql1 {

  case class User(id: Int, name: String, age: Long)
  case class Transport(transport_mode: String, cost_per_unit: Long)
  case class Holidays(id: Int, source: String, destination: String,
transport_mode: String, distance: Long,year: Int)

  def main(args: Array[String]) {

    val conf = new SparkConf()
      .setAppName("vjsprksql1")
      .setMaster("local")

    val sc = new SparkContext(conf)

    val spark = SparkSession
      .builder()
      .appName("Spark SQL basic example")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    import spark.implicits._
```

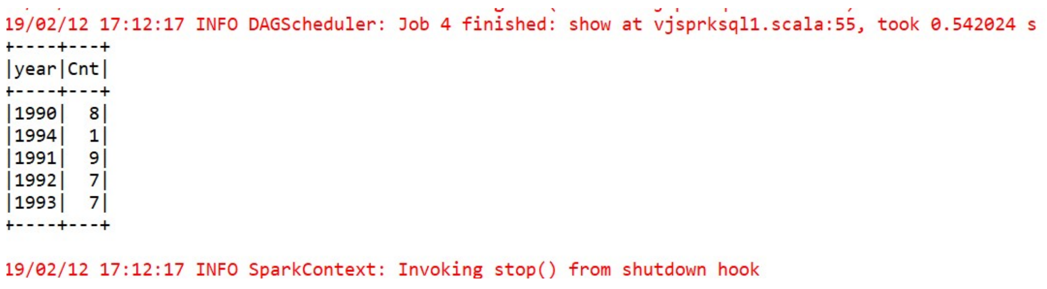
Create three dataframes based on the input file

```
val userDF = spark.sparkContext
      .textFile("C:\\Users\\VIJAYLAKSHMANAN\\spark\\users.txt")
      .map(_.split(","))
      .map(attributes => User(attributes(0).trim.toInt, attributes(1),
attributes(2).trim.toInt))
      .toDF()
    userDF.createOrReplaceTempView("user")

    val transportDF = spark.sparkContext
      .textFile("C:\\Users\\VIJAYLAKSHMANAN\\spark\\transport.txt")
      .map(_.split(","))
      .map(attributes => Transport(attributes(0), attributes(1).trim.toInt))
      .toDF()
    transportDF.createOrReplaceTempView("transport")


    val holidayDF = spark.sparkContext
      .textFile("C:\\Users\\VIJAYLAKSHMANAN\\spark\\holidays.txt")
      .map(_.split(","))
      .map(attributes => Holidays(attributes(0).trim.toInt, attributes(1),
attributes(2), attributes(3), attributes(4).trim.toInt, attributes(5).trim.toInt))
      .toDF()
    holidayDF.createOrReplaceTempView("holiday")
```

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

```
val query1DF = spark.sql("SELECT year, count(*) as Cnt FROM holiday group by
year")
query1DF.show()
```

Output:

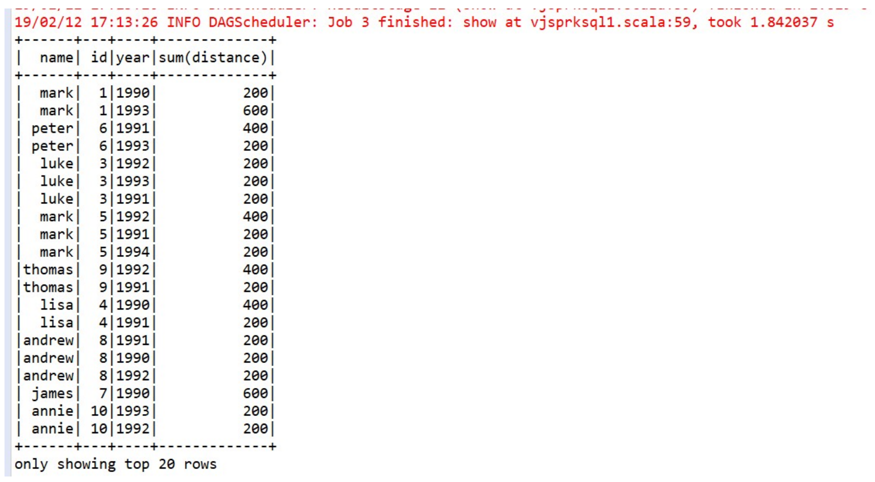


```
19/02/12 17:12:17 INFO DAGScheduler: Job 4 finished: show at vjsprksql1.scala:55, took 0.542024 s
+----+---+
|year|Cnt|
+----+---+
|1990|  8|
|1994|  1|
|1991|  9|
|1992|  7|
|1993|  7|
+----+---+

19/02/12 17:12:17 INFO SparkContext: Invoking stop() from shutdown hook
```

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

```
val query2DF = spark.sql("SELECT a.name, b.id, b.year,  sum(b.distance) FROM user
a,holiday b where a.id=b.id group by a.name, b.id, b.year")
query2DF.show()
```

Output:

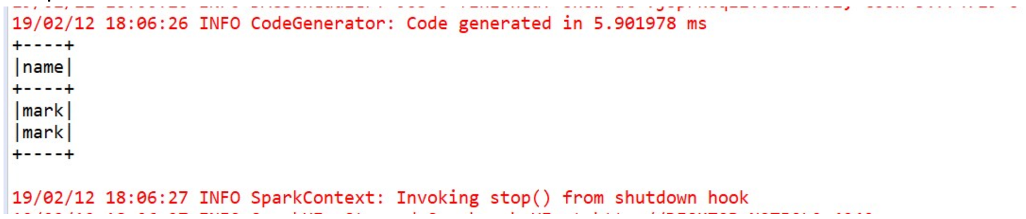


```
19/02/12 17:13:26 INFO DAGScheduler: Job 3 finished: show at vjsprksql1.scala:59, took 1.842037 s
+------+---+----+-------------+
|  name| id|year|sum(distance)|
+------+---+----+-------------+
|  mark|  1|1990|          200|
|  mark|  1|1993|          600|
| peter|  6|1991|          400|
| peter|  6|1993|          200|
|  luke|  3|1992|          200|
|  luke|  3|1993|          200|
|  luke|  3|1991|          200|
|  mark|  5|1992|          400|
|  mark|  5|1991|          200|
|  mark|  5|1994|          200|
|thomas|  9|1992|          400|
|thomas|  9|1991|          200|
|  lisa|  4|1990|          400|
|  lisa|  4|1991|          200|
|andrew|  8|1991|          200|
|andrew|  8|1990|          200|
|andrew|  8|1992|          200|
| james|  7|1990|          600|
| annie| 10|1993|          200|
| annie| 10|1992|          200|
+------+---+----+-------------+
only showing top 20 rows
```

3) Which user has travelled the largest distance till date

```
val query3DF = spark.sql("select a.name,  b.id, sum(b.distance) as s from user a,
holiday b where a.id = b.id group by a.name, b.id order by s desc limit 2")
    query3DF.select("name").show()
```

Output:

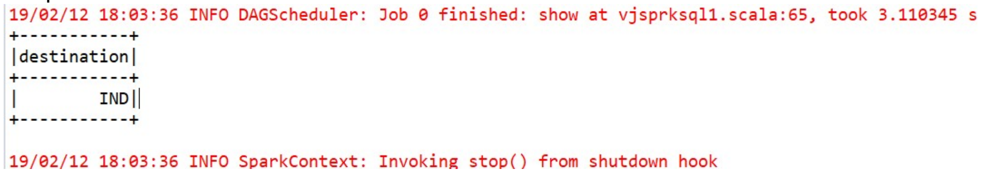


```
19/02/12 18:06:26 INFO CodeGenerator: Code generated in 5.901978 ms
+-----+
|name|
+-----+
|mark|
|mark|
+-----+

19/02/12 18:06:27 INFO SparkContext: Invoking stop() from shutdown hook
```

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

```
val query4DF = spark.sql("select destination, count(*) as c from holiday group by
destination order by c desc limit 1")
    query4DF.select("destination").show()
```

Output:

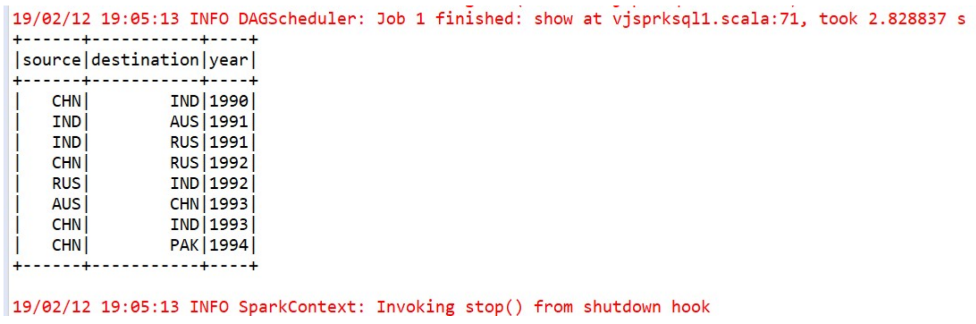


```
19/02/12 18:03:36 INFO DAGScheduler: Job 0 finished: show at vjsprksql1.scala:65, took 3.110345 s
+-----------+
|destination|
+-----------+
|        IND||
+-----------+

19/02/12 18:03:36 INFO SparkContext: Invoking stop() from shutdown hook
```

5)Which route is generating the most revenue per year

```
val query5DF = spark.sql("select source, destination, year, sum(a.distance *
b.cost_per_unit) as x from holiday a, transport b where a.transport_mode =
b.transport_mode group by source, destination, year order by year asc,x desc limit
50")
    //query5DF.select("source", "destination", "year", "x").show()
    query5DF.createOrReplaceTempView("query5f")
    val query6DF = spark.sql("select source, destination, year from query5f a
where a.x = (select max(b.x) from query5f b where a.year = b.year)")
    query6DF.show()
```

Output:

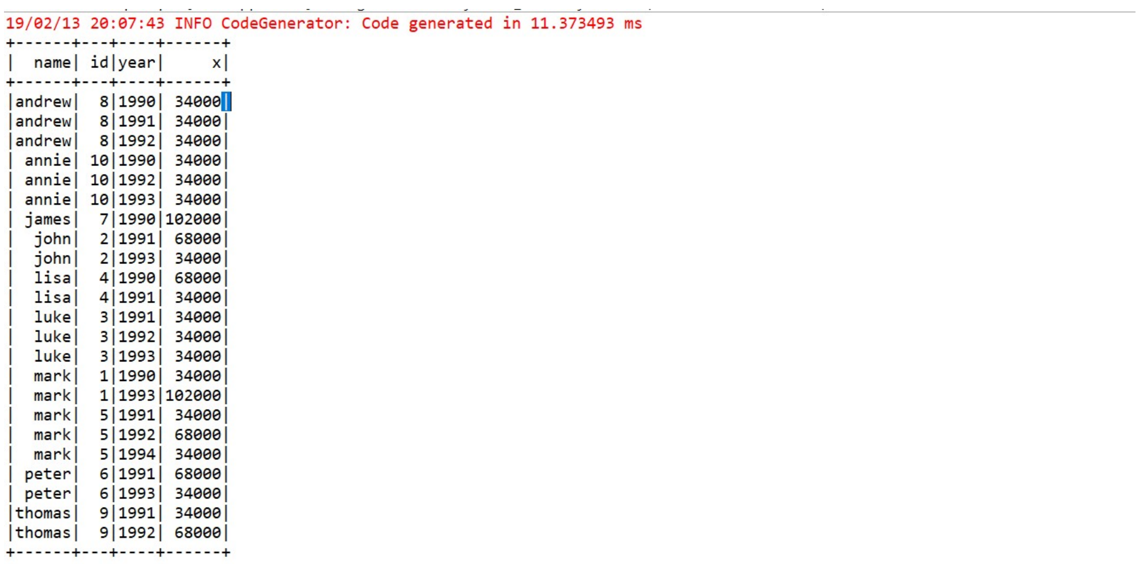


```
19/02/12 19:05:13 INFO DAGScheduler: Job 1 finished: show at vjsprksql1.scala:71, took 2.828837 s
+------+-----------+----+
|source|destination|year|
+------+-----------+----+
|   CHN|        IND|1990|
|   IND|        AUS|1991|
|   IND|        RUS|1991|
|   CHN|        RUS|1992|
|   RUS|        IND|1992|
|   AUS|        CHN|1993|
|   CHN|        IND|1993|
|   CHN|        PAK|1994|
+------+-----------+----+

19/02/12 19:05:13 INFO SparkContext: Invoking stop() from shutdown hook
```

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

```
val query7DF = spark.sql("select c.name, c.id, a.year, sum(a.distance *
b.cost_per_unit) as x from holiday a, transport b, user c where a.transport_mode =
b.transport_mode and a.id = c.id and a.transport_mode = 'airplane' group by
c.name, c.id, year order by c.name, c.id, a.year")
    query7DF.show(100)
```

Output:



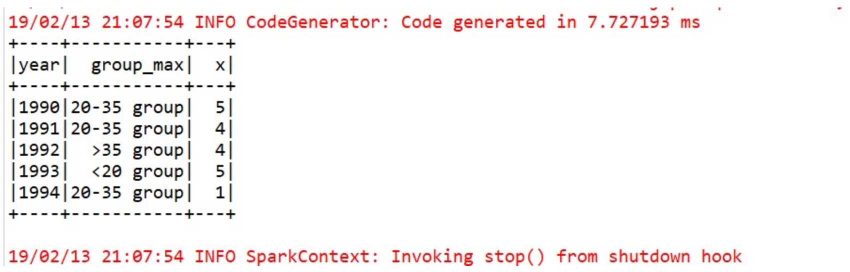
```
19/02/13 20:07:43 INFO CodeGenerator: Code generated in 11.373493 ms
+------+---+----+------+
|  name| id|year|     x|
+------+---+----+------+
|andrew|  8|1990| 34000|
|andrew|  8|1991| 34000|
|andrew|  8|1992| 34000|
| annie| 10|1990| 34000|
| annie| 10|1992| 34000|
| annie| 10|1993| 34000|
| james|  7|1990|102000|
|  john|  2|1991| 68000|
|  john|  2|1993| 34000|
|  lisa|  4|1990| 68000|
|  lisa|  4|1991| 34000|
|  luke|  3|1991| 34000|
|  luke|  3|1992| 34000|
|  luke|  3|1993| 34000|
|  mark|  1|1990| 34000|
|  mark|  1|1993|102000|
|  mark|  5|1991| 34000|
|  mark|  5|1992| 68000|
|  mark|  5|1994| 34000|
| peter|  6|1991| 68000|
| peter|  6|1993| 34000|
|thomas|  9|1991| 34000|
|thomas|  9|1992| 68000|
+------+---+----+------+
```

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

every year.

```
val query8DF = spark.sql("select b.year, CASE WHEN (a.age < 20) THEN '<20 group'
WHEN a.age between 20 and 35 THEN '20-35 group' ELSE '>35 group' end as group_max,
count(b.*) as x from user a, holiday b where a.id = b.id group by b.year,
group_max order by b.year, group_max")
    query8DF.createOrReplaceTempView("query8f")
    val query9DF = spark.sql("select a.year, a.group_max, a.x from query8f a where
a.x = (select max(x) from query8f b where a.year = b.year)")
    query9DF.show()
```

Output:



```
19/02/13 21:07:54 INFO CodeGenerator: Code generated in 7.727193 ms
+----+-----------+---+
|year|  group_max|  x|
+----+-----------+---+
|1990|20-35 group|  5|
|1991|20-35 group|  4|
|1992|  >35 group|  4|
|1993|  <20 group|  5|
|1994|20-35 group|  1|
+----+-----------+---+

19/02/13 21:07:54 INFO SparkContext: Invoking stop() from shutdown hook
```