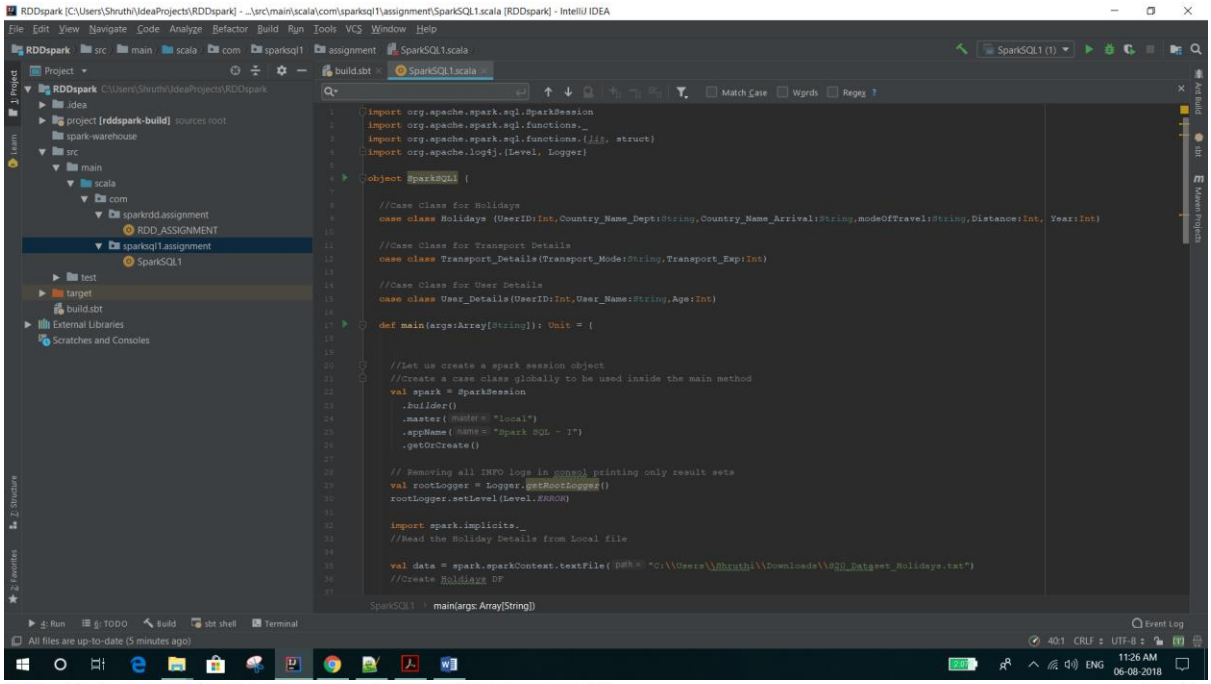
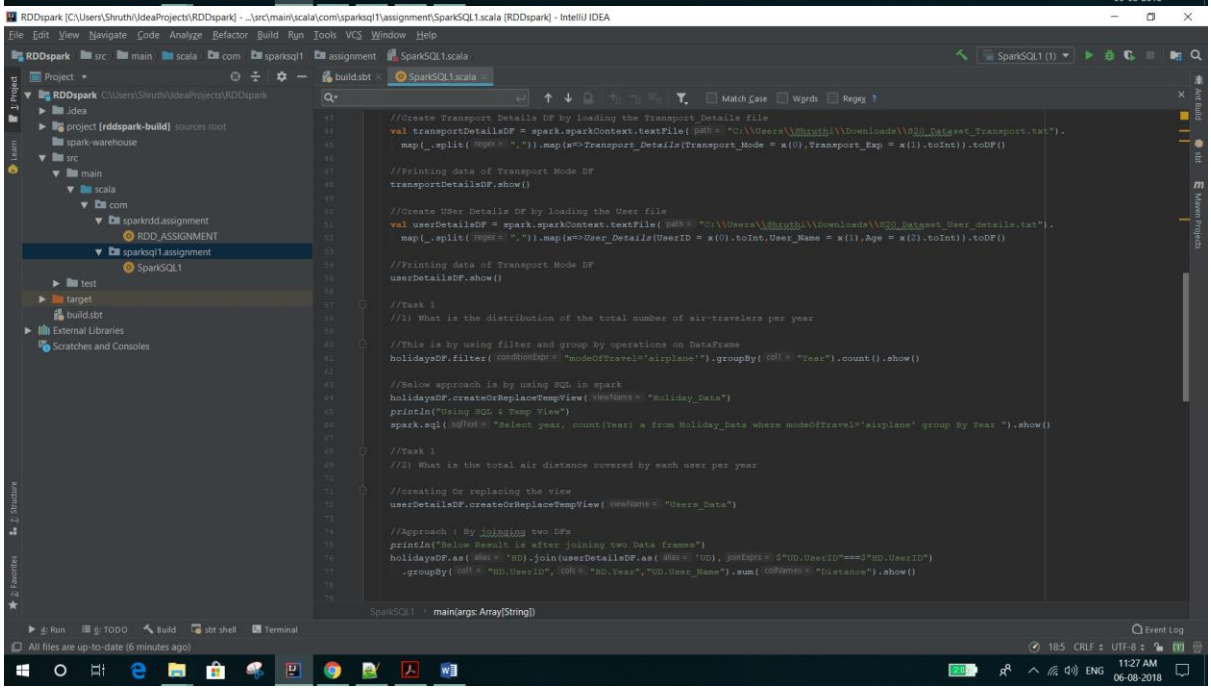
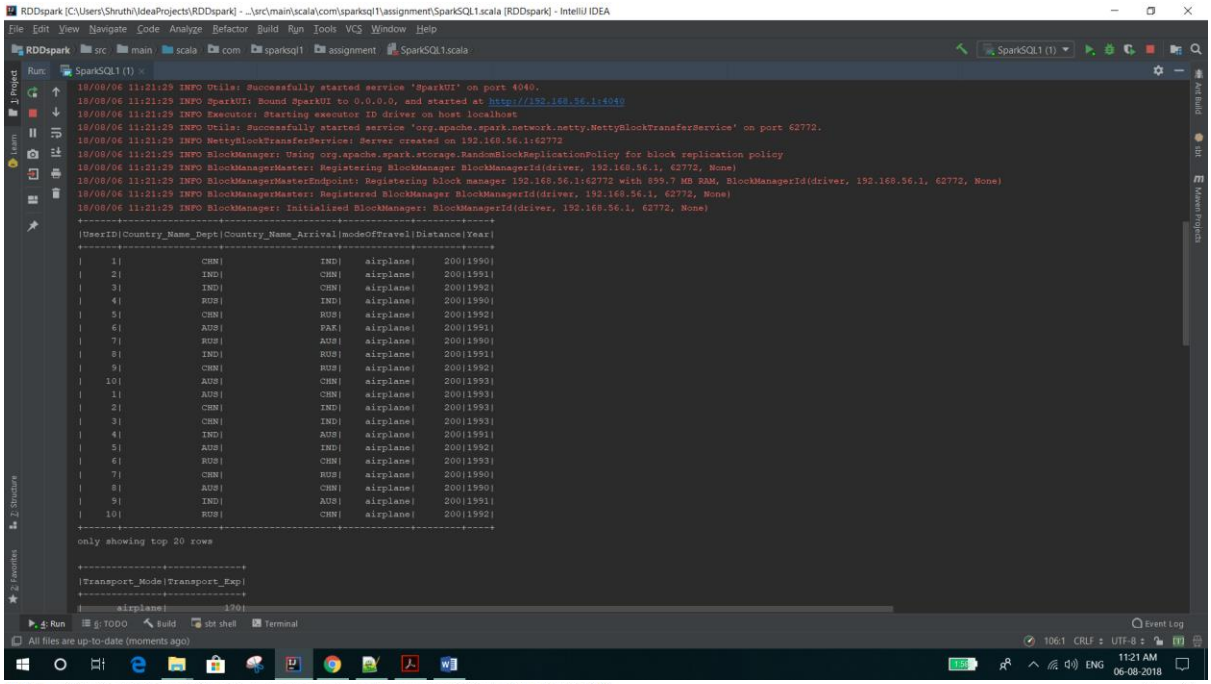
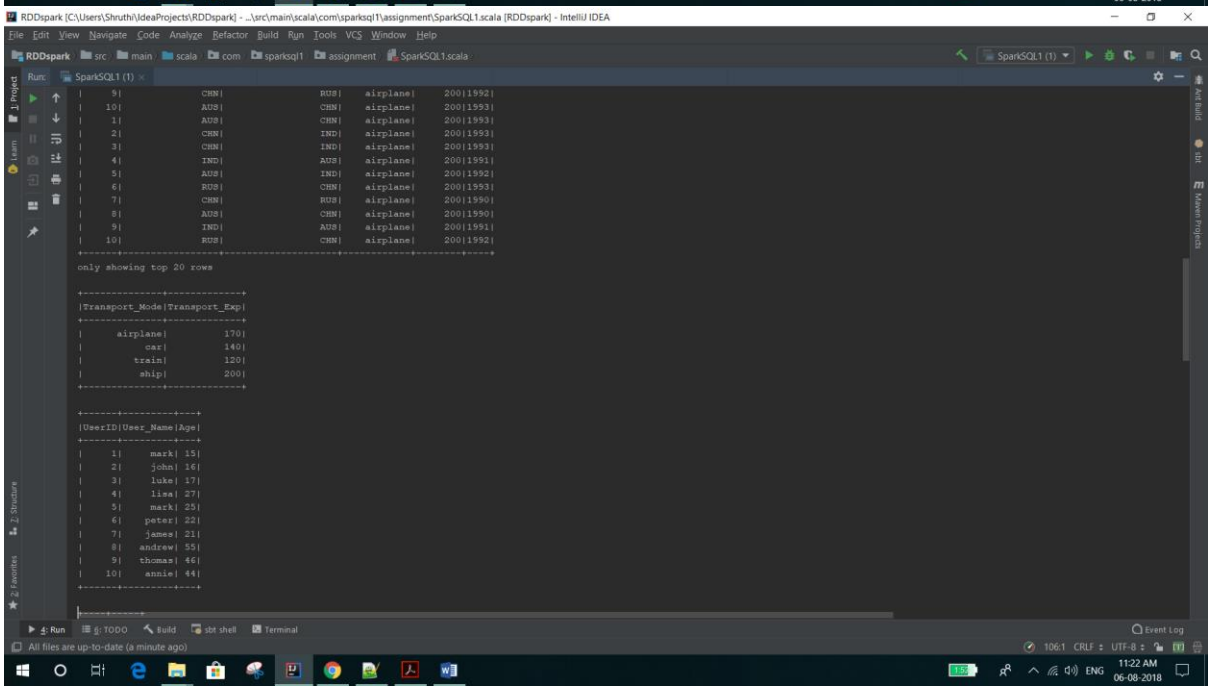
SPARK SQL 1


RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions.{lit, struct}
import org.apache.log4j.{Level, Logger}
object SparkSQL1 {
//Case Class for Holidays
case class Holidays (UserID:Int,Country_Name_Dept:String,Country_Name_Arrival:String,modeOfTravel:String,Distance:Int, Year:Int)
//Case Class for Transport Details
case class Transport_Details(Transport_Mode:String,Transport_Exp:Int)
//Case Class for User Details
case class User_Details(UserID:Int,User_Name:String,Age:Int)
def main(args:Array[String]): Unit = {
//Let us create a spark session object
//Create a case class globally to be used inside the main method
val spark = SparkSession
.builder()
.master("local")
.appName("Spark SQL - 1")
.getOrCreate()
// Removing all INFO logs in console printing only result sets
val rootLogger = Logger.getRootLogger()
rootLogger.setLevel(Level.ERROR)
import spark.implicits._
//Read the Holiday Details from Local file
val data = spark.sparkContext.textFile("C:\\Users\\Shruthi\\Downloads\\S20_Dataset_Holidays.txt")
//Create Holidays DF

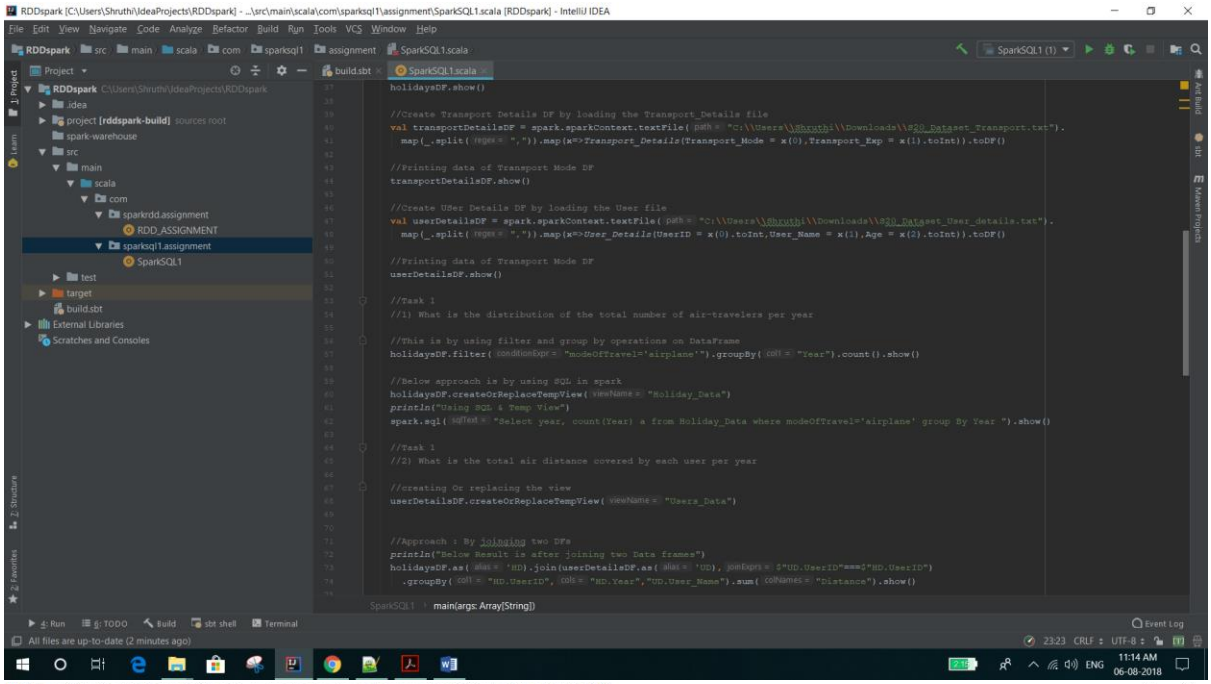

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
//Create Transport Details DF by loading the Transport_Details file
val transportDetailsDF = spark.sparkContext.textFile("C:\\Users\\Shruthi\\Downloads\\S20_Dataset_Transport.txt").
map(_.split(",")).map(x=>Transport_Details(Transport_Mode = x(0),Transport_Exp = x(1).toInt)).toDF()
//Printing data of Transport Mode DF
transportDetailsDF.show()
//Create USer Details DF by loading the User file
val userDetailsDF = spark.sparkContext.textFile("C:\\Users\\Shruthi\\Downloads\\S20_Dataset_User_details.txt").
map(_.split(",")).map(x=>User_Details(UserID = x(0).toInt,User_Name = x(1),Age = x(2).toInt)).toDF()
//Printing data of Transport Mode DF
userDetailsDF.show()
//Task 1
//1) What is the distribution of the total number of air-travelers per year
//This is by using filter and group by operations on DataFrame
holidaysDF.filter("modeOfTravel='airplane'").groupBy("Year").count().show()
//Below approach is by using SQL in spark
holidaysDF.createOrReplaceTempView("Holiday_Data")
println("Using SQL & Temp View")
spark.sql("Select year, count(Year) a from Holiday_Data where modeOfTravel='airplane' group By Year ").show()
//Task 1
//2) What is the total air distance covered by each user per year
//creating Or replacing the view
userDetailsDF.createOrReplaceTempView("Users_Data")
//Approach : By joining two DFs
println("Below Result is after joining two Data frames")
holidaysDF.as("HD").join(userDetailsDF.as("UD"),$"UD.UserID"===$"HD.UserID")
.groupBy("HD.UserID","HD.Year","UD.User_Name").sum("Distance").show()

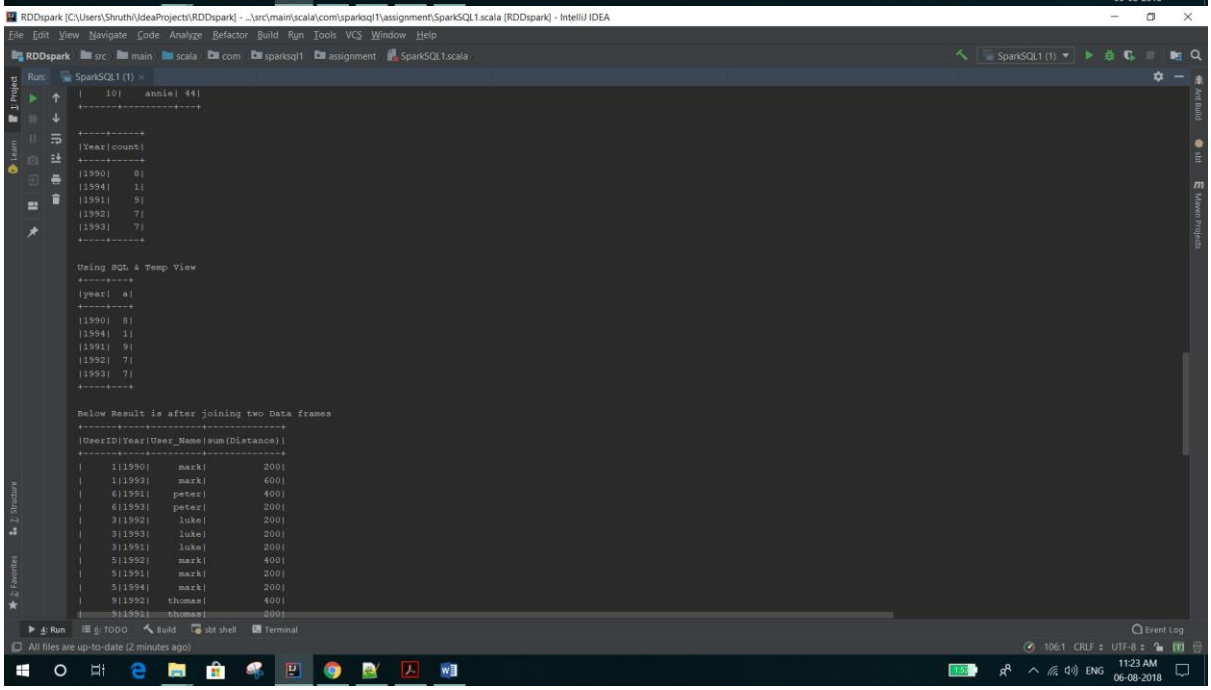
SPARK SQL 1


RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
18/08/06 11:21:29 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/08/06 11:21:29 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://192.168.56.1:4040
18/08/06 11:21:29 INFO Executor: Starting executor ID driver on host localhost
18/08/06 11:21:29 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 62772.
18/08/06 11:21:29 INFO NettyBlockTransferService: Server created on 192.168.56.1:62772
18/08/06 11:21:29 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
18/08/06 11:21:29 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, 192.168.56.1, 62772, None)
18/08/06 11:21:29 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.56.1:62772 with 899.7 MB RAM, BlockManagerId(driver, 192.168.56.1, 62772, None)
18/08/06 11:21:29 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 192.168.56.1, 62772, None)
18/08/06 11:21:29 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 192.168.56.1, 62772, None)
UserID	Country_Name_Dept	Country_Name_Arrival	modeOfTravel	Distance	Year
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
1	AUS	CHN	airplane	200	1993
2	CHN	IND	airplane	200	1993
3	CHN	IND	airplane	200	1993
4	IND	AUS	airplane	200	1991
5	AUS	IND	airplane	200	1992
6	RUS	CHN	airplane	200	1993
7	CHN	RUS	airplane	200	1990
8	AUS	CHN	airplane	200	1990
9	IND	AUS	airplane	200	1991
10	RUS	CHN	airplane	200	1992
only showing top 20 rows
Transport_Mode	Transport_Exp
airplane	170
4: Run
6: TODO
Build
sbt shell
Terminal
Event Log
All files are up-to-date (moments ago)
11:21 AM
06-08-2018

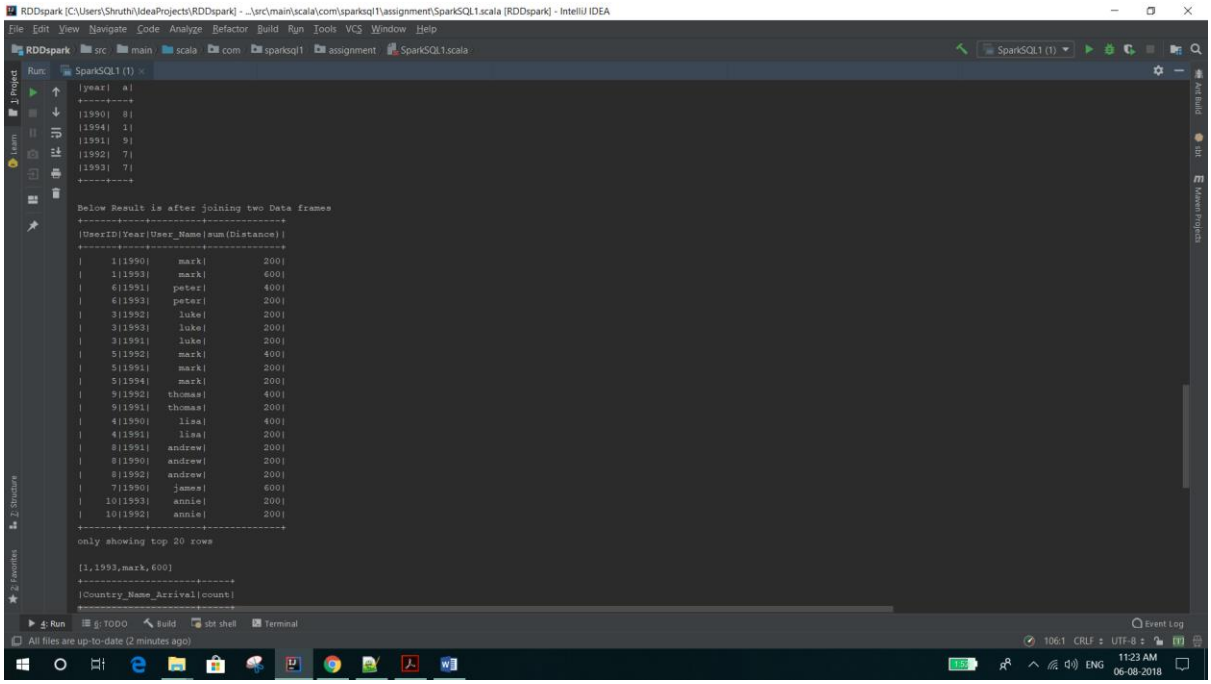

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
9	CHN	RUS	airplane	200	1992
10	AUS	CHN	airplane	200	1993
1	AUS	CHN	airplane	200	1993
2	CHN	IND	airplane	200	1993
3	CHN	IND	airplane	200	1993
4	IND	AUS	airplane	200	1991
5	AUS	IND	airplane	200	1992
6	RUS	CHN	airplane	200	1993
7	CHN	RUS	airplane	200	1990
8	AUS	CHN	airplane	200	1990
9	IND	AUS	airplane	200	1991
10	RUS	CHN	airplane	200	1992
only showing top 20 rows
Transport_Mode	Transport_Exp
airplane	170
car	140
train	120
ship	200
UserID	User_Name	Age
1	mark	15
2	john	16
3	luke	17
4	lisa	27
5	mark	25
6	peter	22
7	james	21
8	andrew	55
9	thomas	46
10	annie	44
4: Run
6: TODO
Build
sbt shell
Terminal
Event Log
All files are up-to-date (a minute ago)
11:22 AM
06-08-2018

SPARK SQL 1

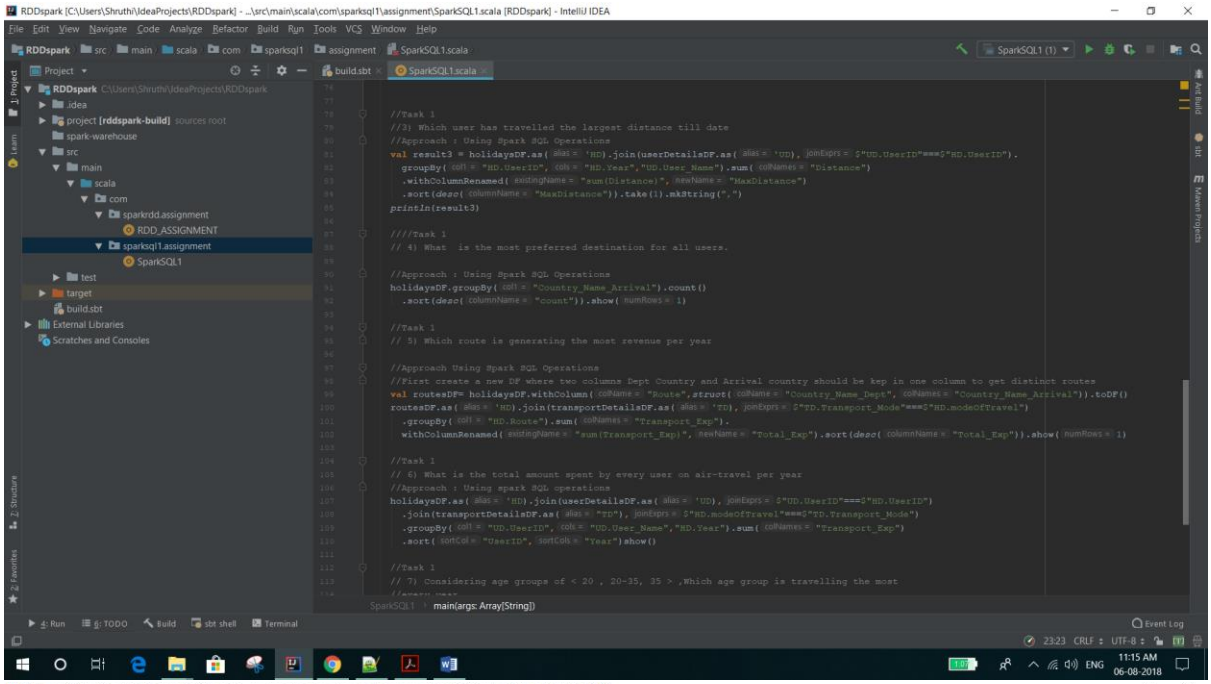

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RDDspark src main scala com sparksql1 assignment SparkSQL1.scala
SparkSQL1 (1)
Project
RDDspark C:\Users\Shruthi\IdeaProjects\RDDspark
.idea
project [rddspark-build] sources root
spark-warehouse
src
main
scala
com
sparkrdd.assignment
RDD_ASSIGNMENT
sparksql1.assignment
SparkSQL1
test
target
build.sbt
External Libraries
Scratches and Consoles
build.sbt
SparkSQL1.scala
holidaysDF.show()
//Create Transport Details DF by loading the Transport_Details file
val transportDetailsDF = spark.sparkContext.textFile(path = "C:\\Users\\Shruthi\\Downloads\\$20_Dataset_Transport.txt").
map(_.split(regex = ",")).map(x=>Transport_Details(Transport_Mode = x(0),Transport_Exp = x(1).toInt)).toDF()
//Printing data of Transport Mode DF
transportDetailsDF.show()
//Create USer Details DF by loading the User file
val userDetailsDF = spark.sparkContext.textFile(path = "C:\\Users\\Shruthi\\Downloads\\$20_Dataset_User_details.txt").
map(_.split(regex = ",")).map(x=>User_Details(UserID = x(0).toInt,User_Name = x(1),Age = x(2).toInt)).toDF()
//Printing data of Transport Mode DF
userDetailsDF.show()
//Task 1
//1) What is the distribution of the total number of air-travelers per year
//This is by using filter and group by operations on DataFrame
holidaysDF.filter(conditionExpr = "modeOfTravel='airplane'").groupBy(col1 = "Year").count().show()
//Below approach is by using SQL in spark
holidaysDF.createOrReplaceTempView(viewName = "Holiday_Data")
println("Using SQL & Temp View")
spark.sql(sqlText = "Select year, count(Year) a from Holiday_Data where modeOfTravel='airplane' group By Year ").show()
//Task 1
//2) What is the total air distance covered by each user per year
//creating Or replacing the view
userDetailsDF.createOrReplaceTempView(viewName = "Users_Data")
//Approach : By joinging two DFs
println("Below Result is after joining two Data frames")
holidaysDF.as(alias = "HD").join(userDetailsDF.as(alias = "UD"), joinExprs = $"UD.UserID"==="HD.UserID")
.groupBy(col1 = "HD.UserID", cols = "HD.Year","UD.User_Name").sum(colNames = "Distance").show()
SparkSQL1 main(args: Array[String])
4: Run 6: TODO Build sbt shell Terminal
Event Log
All files are up-to-date (2 minutes ago)
23:23 CRLF UTF-8
ENG 11:14 AM 06-08-2018

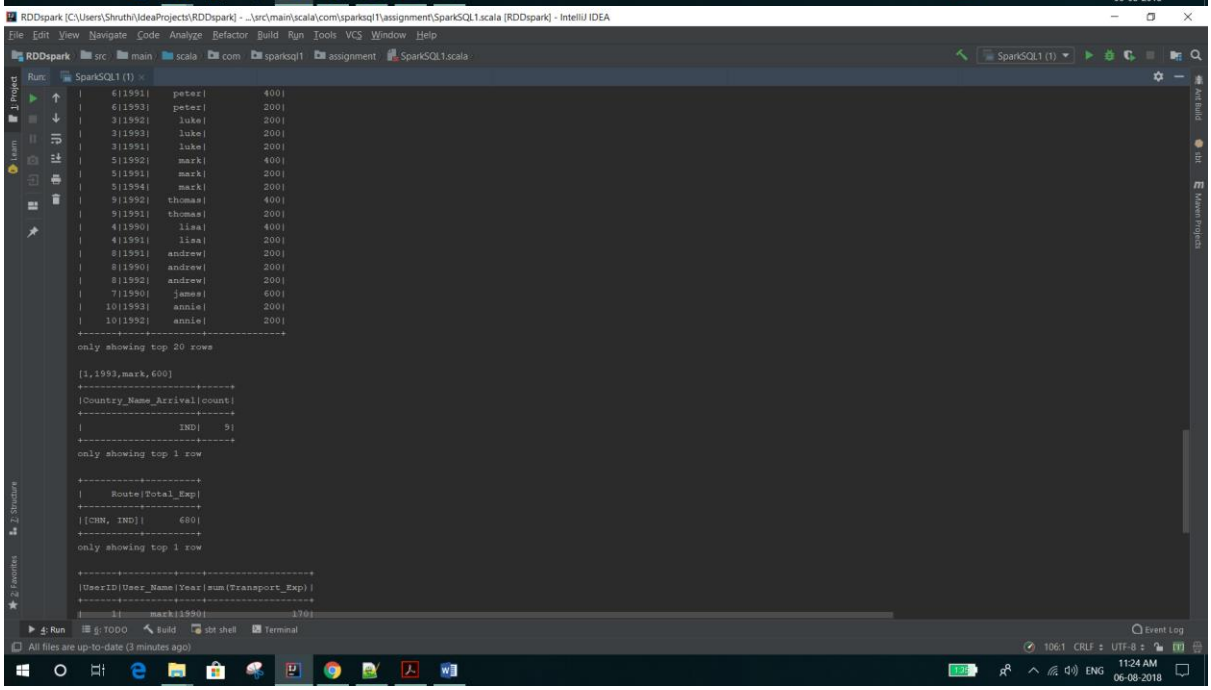

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Run: SparkSQL1 (1)
| 10| annie| 44|
+------+---------+---+
+----+-----+
|Year|count|
+----+-----+
|1990| 8|
|1994| 1|
|1991| 9|
|1992| 7|
|1993| 7|
+----+-----+
Using SQL & Temp View
+----+---+
|year| a|
+----+---+
|1990| 8|
|1994| 1|
|1991| 9|
|1992| 7|
|1993| 7|
+----+---+
Below Result is after joining two Data frames
+------+----+---------+-------------+
|UserID|Year|User_Name|sum(Distance)|
+------+----+---------+-------------+
1	1990	mark	200
1	1993	mark	600
6	1991	peter	400
6	1993	peter	200
3	1992	luke	200
3	1993	luke	200
3	1991	luke	200
5	1992	mark	400
5	1991	mark	200
5	1994	mark	200
9	1992	thomas	400
9	1991	thomas	200
4: Run 6: TODO Build sbt shell Terminal
Event Log
All files are up-to-date (2 minutes ago)
106:1 CRLF UTF-8
ENG 11:23 AM 06-08-2018

SPARK SQL 1

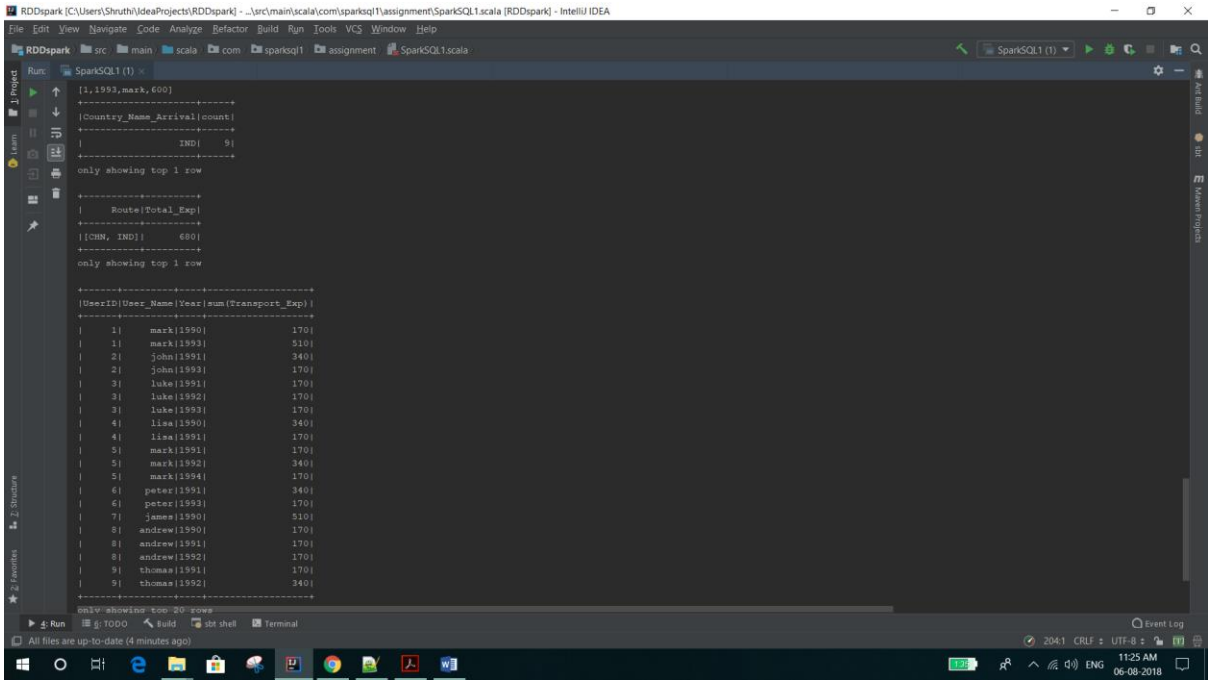

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkSQL1 (1)
|year| a|
|1990| 8|
|1994| 1|
|1991| 9|
|1992| 7|
|1993| 7|
Below Result is after joining two Data frames
UserID	Year	User_Name	sum(Distance)
1	1990	mark	200
1	1993	mark	600
6	1991	peter	400
6	1993	peter	200
3	1992	luke	200
3	1993	luke	200
3	1991	luke	200
5	1992	mark	400
5	1991	mark	200
5	1994	mark	200
9	1992	thomas	400
9	1991	thomas	200
4	1990	lisa	400
4	1991	lisa	200
8	1991	andrew	200
8	1990	andrew	200
8	1992	andrew	200
7	1990	james	600
10	1993	annie	200
10	1992	annie	200
only showing top 20 rows
[1,1993,mark,600]
|Country_Name_Arrival|count|
4: Run
6: TODO
Build
sbt shell
Terminal
Event Log
All files are up-to-date (2 minutes ago)
106:1 CRLF UTF-8
ENG
11:23 AM
06-08-2018

SPARK SQL 1

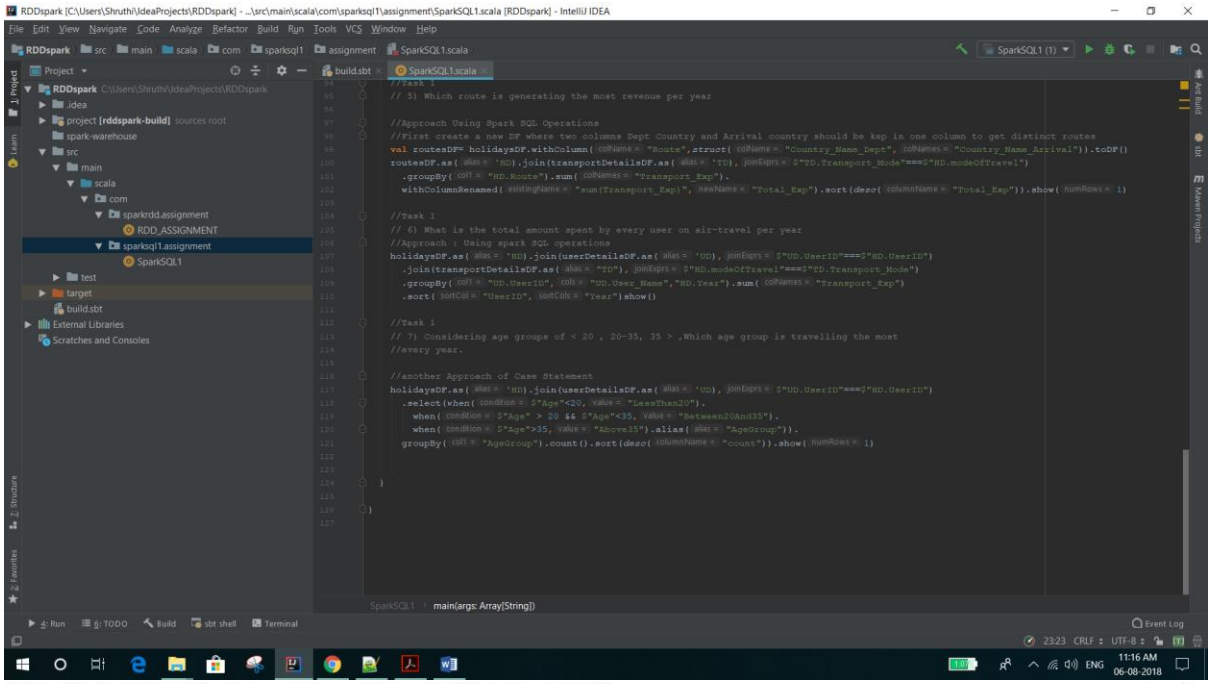

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RDDspark src main scala com sparksql1 assignment SparkSQL1.scala
SparkSQL1 (1)
Project
RDDspark C:\Users\Shruthi\IdeaProjects\RDDspark
.idea
project [rddspark-build] sources root
spark-warehouse
src
main
scala
com
sparkrdd.assignment
RDD_ASSIGNMENT
sparksql1.assignment
SparkSQL1
test
target
build.sbt
External Libraries
Scratches and Consoles
build.sbt
SparkSQL1.scala
//Task 1
//3) Which user has travelled the largest distance till date
//Approach : Using Spark SQL Operations
val result3 = holidaysDF.as(alias = 'HD').join(userDetailsDF.as(alias = 'UD'), joinExprs = $"UD.UserID"===$"HD.UserID").
groupBy(col1 = "HD.UserID", cols = "HD.Year","UD.User_Name").sum(colNames = "Distance")
.withColumnRenamed(existingName = "sum(Distance)", newName = "MaxDistance")
.sort(desc(columnName = "MaxDistance")).take(1).mkString(",")
println(result3)
////Task 1
// 4) What is the most preferred destination for all users.
//Approach : Using Spark SQL Operations
holidaysDF.groupBy(col1 = "Country_Name_Arrival").count()
.sort(desc(columnName = "count")).show(numRows = 1)
//Task 1
// 5) Which route is generating the most revenue per year
//Approach Using Spark SQL Operations
//First create a new DF where two columns Dept Country and Arrival country should be kep in one column to get distinct routes
val routesDF= holidaysDF.withColumn(colName = "Route",struct(colName = "Country_Name_Dept", colNames = "Country_Name_Arrival")).toDF()
routesDF.as(alias = 'HD').join(transportDetailsDF.as(alias = 'TD'), joinExprs = $"TD.Transport_Mode"===$"HD.modeOfTravel")
.groupBy(col1 = "HD.Route").sum(colNames = "Transport_Exp").
withColumnRenamed(existingName = "sum(Transport_Exp)", newName = "Total_Exp").sort(desc(columnName = "Total_Exp")).show(numRows = 1)
//Task 1
// 6) What is the total amount spent by every user on air-travel per year
//Approach : Using spark SQL operations
holidaysDF.as(alias = 'HD').join(userDetailsDF.as(alias = 'UD'), joinExprs = $"UD.UserID"===$"HD.UserID")
.join(transportDetailsDF.as(alias = "TD"), joinExprs = $"HD.modeOfTravel"===$"TD.Transport_Mode")
.groupBy(col1 = "UD.UserID", col = "UD.User_Name","HD.Year").sum(colNames = "Transport_Exp")
.sort(sortCol = "UserID", sortCols = "Year").show()
//Task 1
// 7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most
SparkSQL1 main(args: Array[String])
4: Run 6: TODO Build sbt shell Terminal
Event Log
23:23 CRLF UTF-8
ENG
11:15 AM
06-08-2018

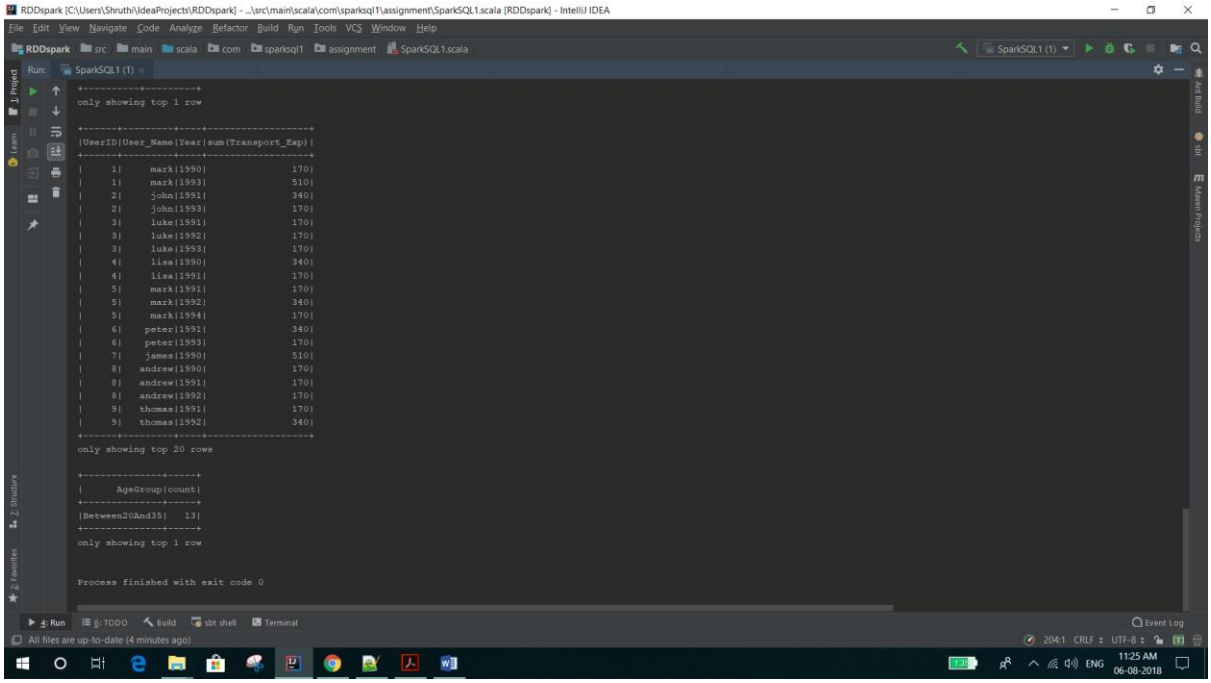

RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Run: SparkSQL1 (1)
6	1991	peter	400
6	1993	peter	200
3	1992	luke	200
3	1993	luke	200
3	1991	luke	200
5	1992	mark	400
5	1991	mark	200
5	1994	mark	200
9	1992	thomas	400
9	1991	thomas	200
4	1990	lisa	400
4	1991	lisa	200
8	1991	andrew	200
8	1990	andrew	200
8	1992	andrew	200
7	1990	james	600
10	1993	annie	200
10	1992	annie	200
only showing top 20 rows
[1,1993,mark,600]
|Country_Name_Arrival|count|
| IND| 9|
only showing top 1 row
| Route|Total_Exp|
|[CHN, IND]| 680|
only showing top 1 row
|UserID|User_Name|Year|sum(Transport_Exp)|
| 1| mark|1990| 170|
4: Run 6: TODO Build sbt shell Terminal
Event Log
All files are up-to-date (3 minutes ago)
106:1 CRLF UTF-8
ENG
11:24 AM
06-08-2018

SPARK SQL 1


RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RDDspark src main scala com sparksql1 assignment SparkSQL1.scala
SparkSQL1 (1)
Run: SparkSQL1 (1)
[1,1993,mark,600]
+--------------------+-----+
|Country_Name_Arrival|count|
+--------------------+-----+
| IND| 9|
+--------------------+-----+
only showing top 1 row
+----------+---------+
| Route|Total_Exp|
+----------+---------+
|[CHN, IND]| 680|
+----------+---------+
only showing top 1 row
+------+---------+----+--------------------+
|UserID|User_Name|Year|sum(Transport_Exp)|
+------+---------+----+--------------------+
1	mark	1990	170
1	mark	1993	510
2	john	1991	340
2	john	1993	170
3	luke	1991	170
3	luke	1992	170
3	luke	1993	170
4	lisa	1990	340
4	lisa	1991	170
5	mark	1991	170
5	mark	1992	340
5	mark	1994	170
6	peter	1991	340
6	peter	1993	170
7	james	1990	510
8	andrew	1990	170
8	andrew	1991	170
8	andrew	1992	170
9	thomas	1991	170
9	thomas	1992	340
+------+---------+----+--------------------+
only showing top 20 rows
4: Run 6: TODO Build sbt shell Terminal
Event Log
All files are up-to-date (4 minutes ago)
204:1 CRLF UTF-8
ENG
11:25 AM
06-08-2018

SPARK SQL 1


RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RDDspark src main scala com sparksql1 assignment SparkSQL1.scala
SparkSQL1 (1)
Project
RDDspark C:\Users\Shruthi\IdeaProjects\RDDspark
.idea
project [rddspark-build] sources root
spark-warehouse
src
main
scala
com
sparkrdd.assignment
RDD_ASSIGNMENT
sparksql1.assignment
SparkSQL1
test
target
build.sbt
External Libraries
Scratches and Consoles
build.sbt
SparkSQL1.scala
//Task 1
// 5) Which route is generating the most revenue per year
//Approach Using Spark SQL Operations
//First create a new DF where two columns Dept Country and Arrival country should be kep in one column to get distinct routes
val routesDF= holidaysDF.withColumn(colName = "Route",struct(colName = "Country_Name_Dept", colNames = "Country_Name_Arrival")).toDF()
routesDF.as(alias = 'RD').join(transportDetailsDF.as(alias = 'TD'), joinExprs = $"TD.Transport_Mode"==="RD.modeOfTravel")
.groupBy(col = "RD.Route").sum(colNames = "Transport_Exp").
withColumnRenamed(existingName = "sum(Transport_Exp)", newName = "Total_Exp").sort(desc(columnName = "Total_Exp")).show(numRows = 1)
//Task 1
// 6) What is the total amount spent by every user on air-travel per year
//Approach : Using spark SQL operations
holidaysDF.as(alias = 'HD').join(userDetailsDF.as(alias = 'UD'), joinExprs = $"UD.UserID"==="HD.UserID")
.join(transportDetailsDF.as(alias = "TD"), joinExprs = $"HD.modeOfTravel"==="TD.Transport_Mode")
.groupBy(col1 = "UD.UserID", cols = "UD.User_Name","HD.Year").sum(colNames = "Transport_Exp")
.sort(sortCol = "UserID", sortCols = "Year").show()
//Task 1
// 7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most
//every year.
//another Approach of Case Statement
holidaysDF.as(alias = 'HD').join(userDetailsDF.as(alias = 'UD'), joinExprs = $"UD.UserID"==="HD.UserID")
.select(when(condition = $"Age"<20, value = "LessThan20").
when(condition = $"Age" > 20 && $"Age"<35, value = "Between20And35").
when(condition = $"Age">35, value = "Above35").alias(alias = "AgeGroup")).
groupBy(col1 = "AgeGroup").count().sort(desc(columnName = "count")).show(numRows = 1)
SparkSQL1 main(args: Array[String])
4: Run 6: TODO Build sbt shell Terminal
Event Log
23:23 CRLF UTF-8
ENG
11:16 AM
06-08-2018


RDDspark [C:\Users\Shruthi\IdeaProjects\RDDspark] - ...\src\main\scala\com\sparksql1\assignment\SparkSQL1.scala [RDDspark] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Run: SparkSQL1 (1)
+----------+----------+
only showing top 1 row
+------+---------+----+------------------+
|UserID|User_Name|Year|sum(Transport_Exp)|
+------+---------+----+------------------+
1	mark	1990	170
1	mark	1993	510
2	john	1991	340
2	john	1993	170
3	luke	1991	170
3	luke	1992	170
3	luke	1993	170
4	lisa	1990	340
4	lisa	1991	170
5	mark	1991	170
5	mark	1992	340
5	mark	1994	170
6	peter	1991	340
6	peter	1993	170
7	james	1990	510
8	andrew	1990	170
8	andrew	1991	170
8	andrew	1992	170
9	thomas	1991	170
9	thomas	1992	340
+------+---------+----+------------------+
only showing top 20 rows
+--------------+-----+
| AgeGroup|count|
+--------------+-----+
|Between20And35| 13|
+--------------+-----+
only showing top 1 row
Process finished with exit code 0
4: Run 6: TODO Build sbt shell Terminal
Event Log
All files are up-to-date (4 minutes ago)
204:1 CRLF UTF-8
ENG
11:25 AM
06-08-2018