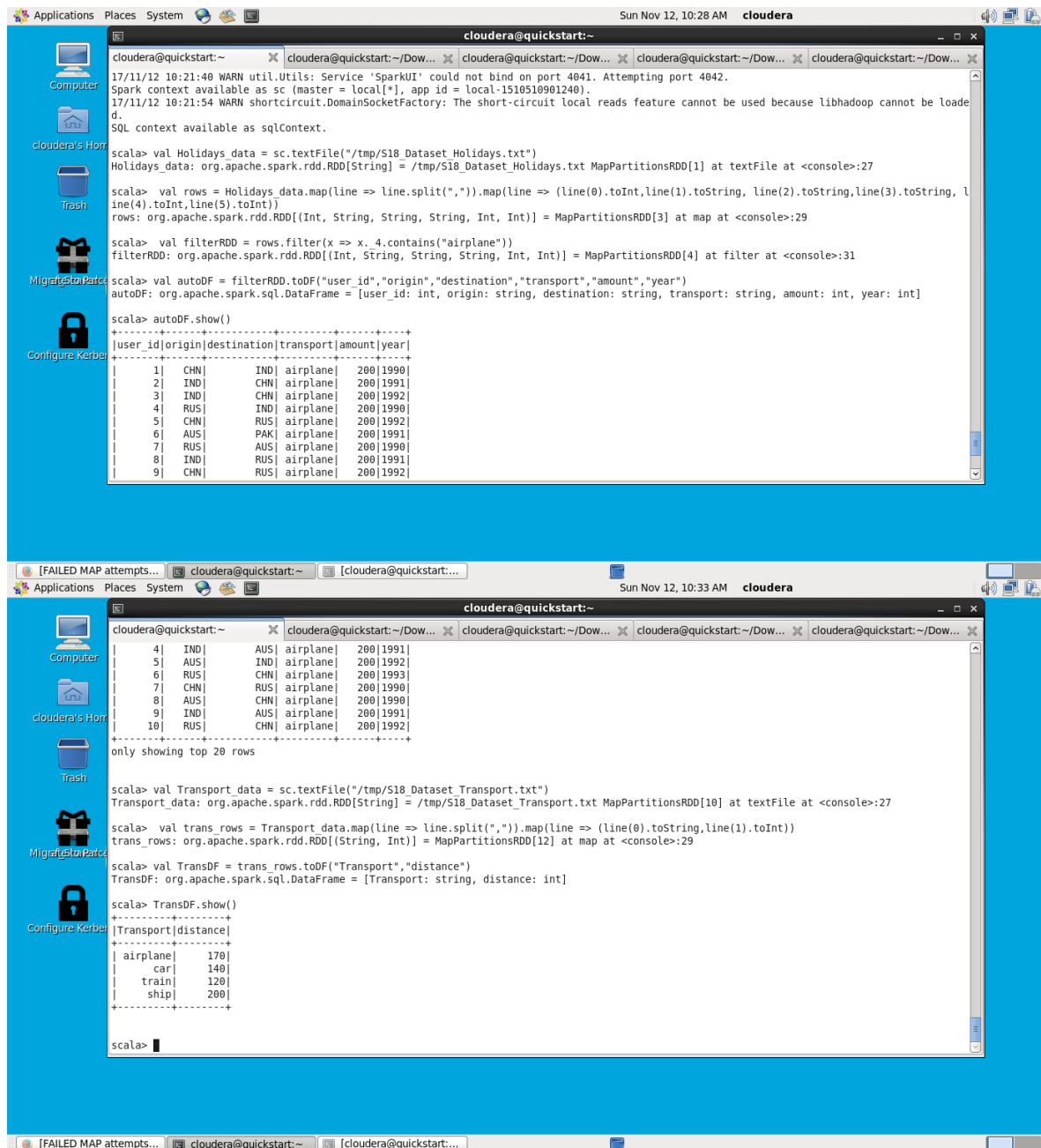


Convert text file to data frame to run queries:



The screenshot shows a Cloudera Quickstart terminal window with the following Scala code and output:

```
scala> val Holidays_data = sc.textFile("/tmp/S18_Dataset_Holidays.txt")
Holidays_data: org.apache.spark.rdd.RDD[String] = /tmp/S18_Dataset_Holidays.txt MapPartitionsRDD[1] at textFile at <console>:27

scala> val rows = Holidays_data.map(line => line.split(",")).map(line => (line(0).toInt, line(1).toString, line(2).toString, line(3).toString, line(4).toInt, line(5).toInt))
rows: org.apache.spark.rdd.RDD[(Int, String, String, String, Int, Int)] = MapPartitionsRDD[3] at map at <console>:29

scala> val filterRDD = rows.filter(x => x._4.contains("airplane"))
filterRDD: org.apache.spark.rdd.RDD[(Int, String, String, String, Int, Int)] = MapPartitionsRDD[4] at filter at <console>:31

scala> val autoDF = filterRDD.toDF("user_id", "origin", "destination", "transport", "amount", "year")
autoDF: org.apache.spark.sql.DataFrame = [user_id: int, origin: string, destination: string, transport: string, amount: int, year: int]

scala> autoDF.show()
+-----+-----+-----+-----+-----+
|user_id|origin|destination|transport|amount|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|
+-----+-----+-----+-----+
```

The second screenshot shows the continuation of the terminal session:

```
scala> val Transport_data = sc.textFile("/tmp/S18_Dataset_Transport.txt")
Transport_data: org.apache.spark.rdd.RDD[String] = /tmp/S18_Dataset_Transport.txt MapPartitionsRDD[10] at textFile at <console>:27

scala> val trans_rows = Transport_data.map(line => line.split(",")).map(line => (line(0).toString, line(1).toInt))
trans_rows: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[12] at map at <console>:29

scala> val TransDF = trans_rows.toDF("Transport", "distance")
TransDF: org.apache.spark.sql.DataFrame = [Transport: string, distance: int]

scala> TransDF.show()
+-----+-----+
|Transport|distance|
+-----+-----+
|airplane|170|
|car|140|
|train|120|
|ship|200|
+-----+-----+
```

The terminal window also shows a message: "only showing top 20 rows".

```
Applications Places System cloudera Sun Nov 12, 10:35 AM cloudera
Terminal
Use the command line
cloudera@quickstart:~/Downloads
train| 120|
ship| 200|
-----+-----
scala> val User_data = sc.textFile("/tmp/S18_Dataset_User_Details.txt")
User_data: org.apache.spark.rdd.RDD[String] = /tmp/S18_Dataset_User_Details.txt MapPartitionsRDD[18] at textFile at <console>:27
scala> val user_rows = User_data.map(line => line.split(",")).map(line => (line(0).toInt, line(1).toString, line(2).toInt))
user_rows: org.apache.spark.rdd.RDD[(Int, String, Int)] = MapPartitionsRDD[20] at map at <console>:29
scala> val UserDF = user_rows.toDF("user_id", "name", "age")
UserDF: org.apache.spark.sql.DataFrame = [user_id: int, name: string, age: int]
scala> UserDF.show()
+-----+-----+
|user_id| 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|
+-----+-----+
scala>
```

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

```
Applications Places System cloudera Sun Nov 12, 9:59 AM cloudera
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
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
scala> autoDF.groupBy("year").count().show()
+-----+-----+
|year| count|
+-----+-----+
|1990| 8|
|1991| 9|
|1992| 7|
|1993| 7|
|1994| 1|
+-----+-----+
scala>
```

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

```
cloudera@quickstart:~  
only showing top 20 rows  
  
scala> autoDF.join(TransDF, autoDF("transport") === TransDF("Transport")).groupBy("year","user_id").sum("distance").show()  
+-----+  
|year|user_id|sum(distance)|  
+-----+  
|1993|6|170|  
|1993|10|170|  
|1994|5|170|  
|1990|1|170|  
|1990|4|340|  
|1990|7|510|  
|1990|8|170|  
|1990|10|170|  
|1991|2|340|  
|1991|3|170|  
|1991|4|170|  
|1991|5|170|  
|1991|6|340|  
|1991|8|170|  
|1991|9|170|  
|1992|3|170|  
|1992|5|340|  
|1992|8|170|  
|1992|9|340|  
|1992|10|170|  
+-----+  
only showing top 20 rows  
  
scala>
```

3) Which user has travelled the largest distance till date

```
cloudera@quickstart:~  
only showing top 20 rows  
  
scala> autoDF.join(TransDF, autoDF("transport") === TransDF("Transport")).groupBy("user_id").sum("distance").show()  
+-----+  
|user_id|sum(distance)|  
+-----+  
|1|680|  
|2|510|  
|3|510|  
|4|510|  
|5|680|  
|6|510|  
|7|510|  
|8|510|  
|9|510|  
|10|510|  
+-----+  
  
scala>
```

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

