Uploading and creating dataframe

scala> val df =sqlContext.read.format("com.databricks.spark.csv").option("header", "true").option("delimiter", ",").load("/user/datafiles/casestudy.csv")

scala> case class buses(year:String,run\_type:String,bus\_no:String,route\_no:String,reason:String,occured\_on:String,no\_of\_student:Integer)

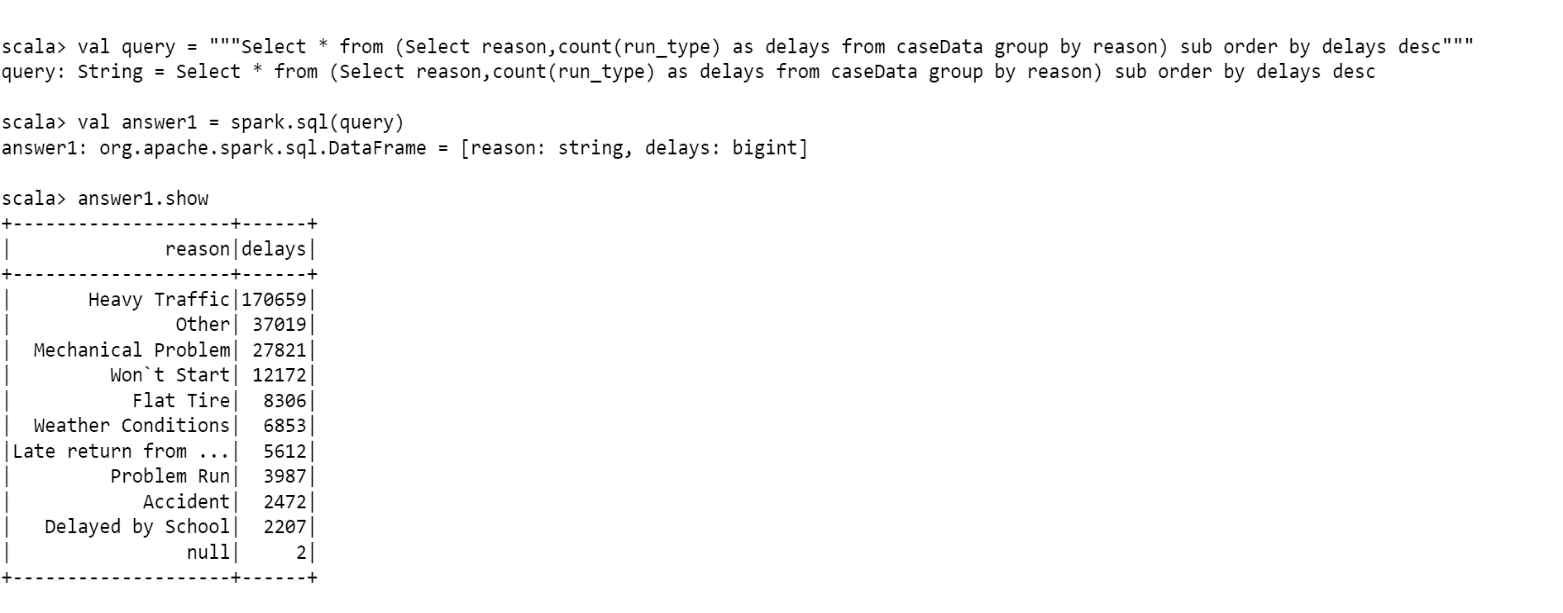
scala> val df1 = sc.textFile("/user/datafiles/casestudy.csv").map(\_.split(",")).map(p=>casestudy(p(0),p(1),p(2),p(3),p(4),p(5),p(6).trim.toInt)).toDF()

**Use Case 1 - Most common reasons for either a delay or breaking down of the bus**

scala> val query = """Select \* from (Select reason,count(Run\_Type) as delays from caseData group by Reason) sub order by delays desc"""

scala> val answer1 = spark.sql(query)

scala> answer1.show

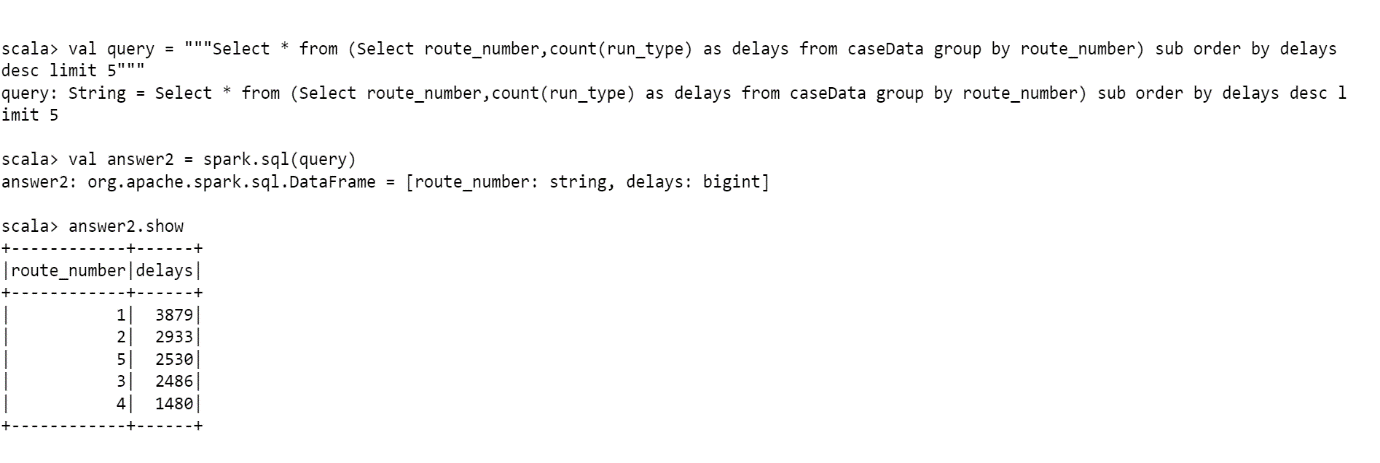


**Use Case 2 - Top five route numbers where the bus was either delayed or broke down**

scala> val query = """Select \* from (Select Route\_Number,count(Run\_Type) as delays from caseData group by Route\_No) sub order by delays desc limit 5"""

scala> val answer2 = spark.sql(query)

scala> answer2.show



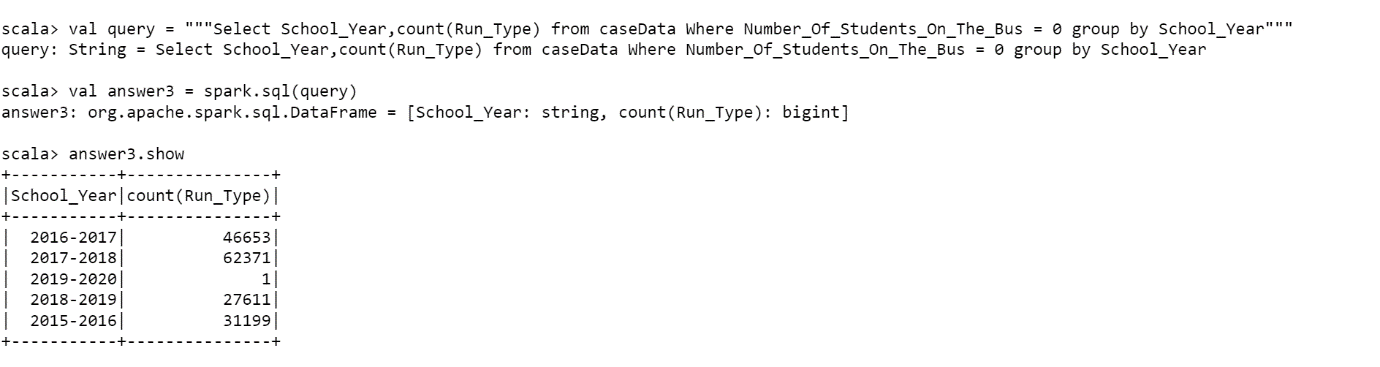
**Use Case 3 - The total number of incidents, year-wise, when the students were**

**3.1** **when students were not in the bus**

scala> val query = """Select School\_Year,count(Run\_Type) from caseData where Number\_Of\_Students\_In\_The\_Bus=0 group by School\_year"""

scala> val answer3 = spark.sql(query)

scala> answer3.show

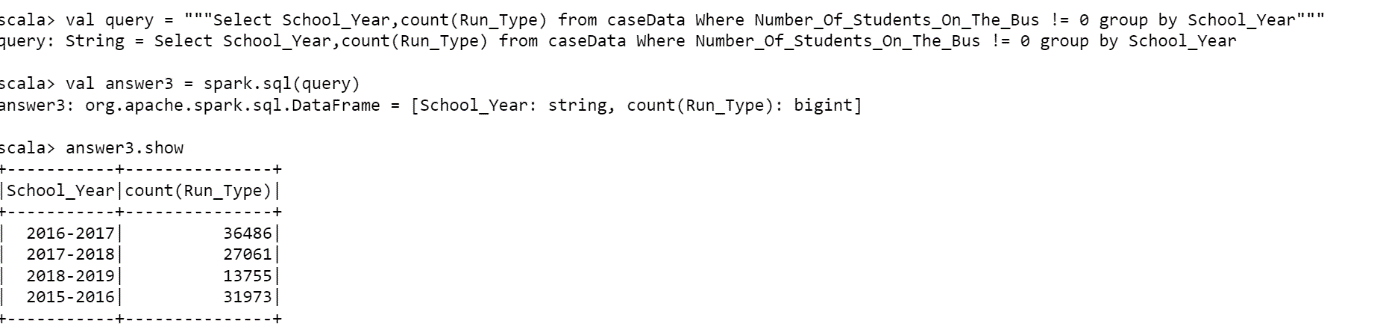


**3.2** **when students were in the bus**

scala> val query = """Select School\_Year,count(Run\_Type) from caseData where Number\_Of\_Students\_In\_The\_Bus!=0 group by School\_year"""

scala> val answer3 = spark.sql(query)

scala> answer3.show

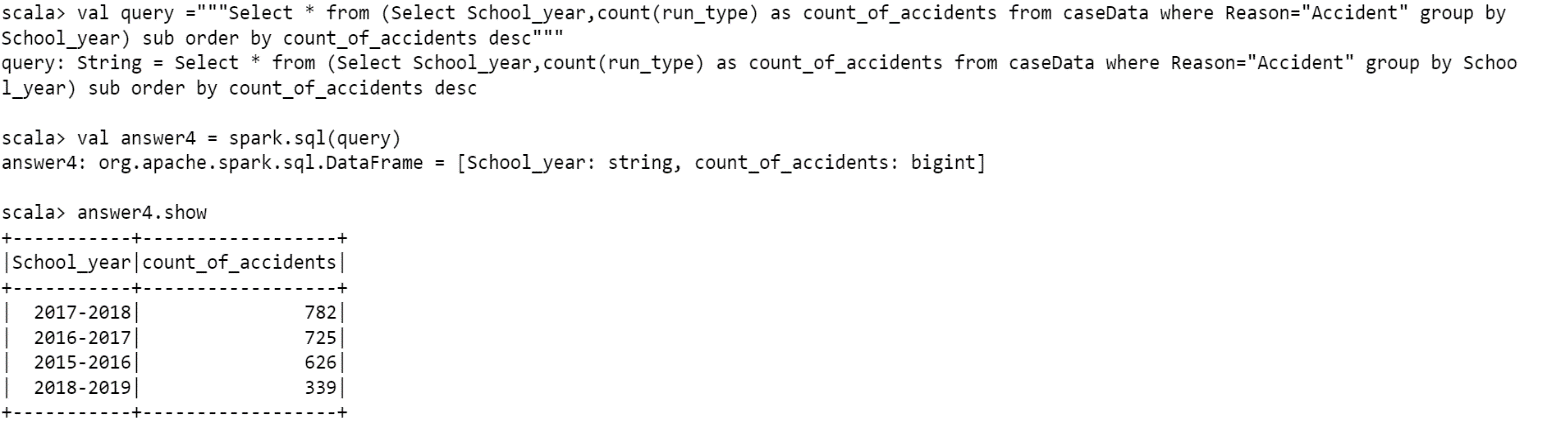


**Use Case 4 -** **The year in which accidents were less**

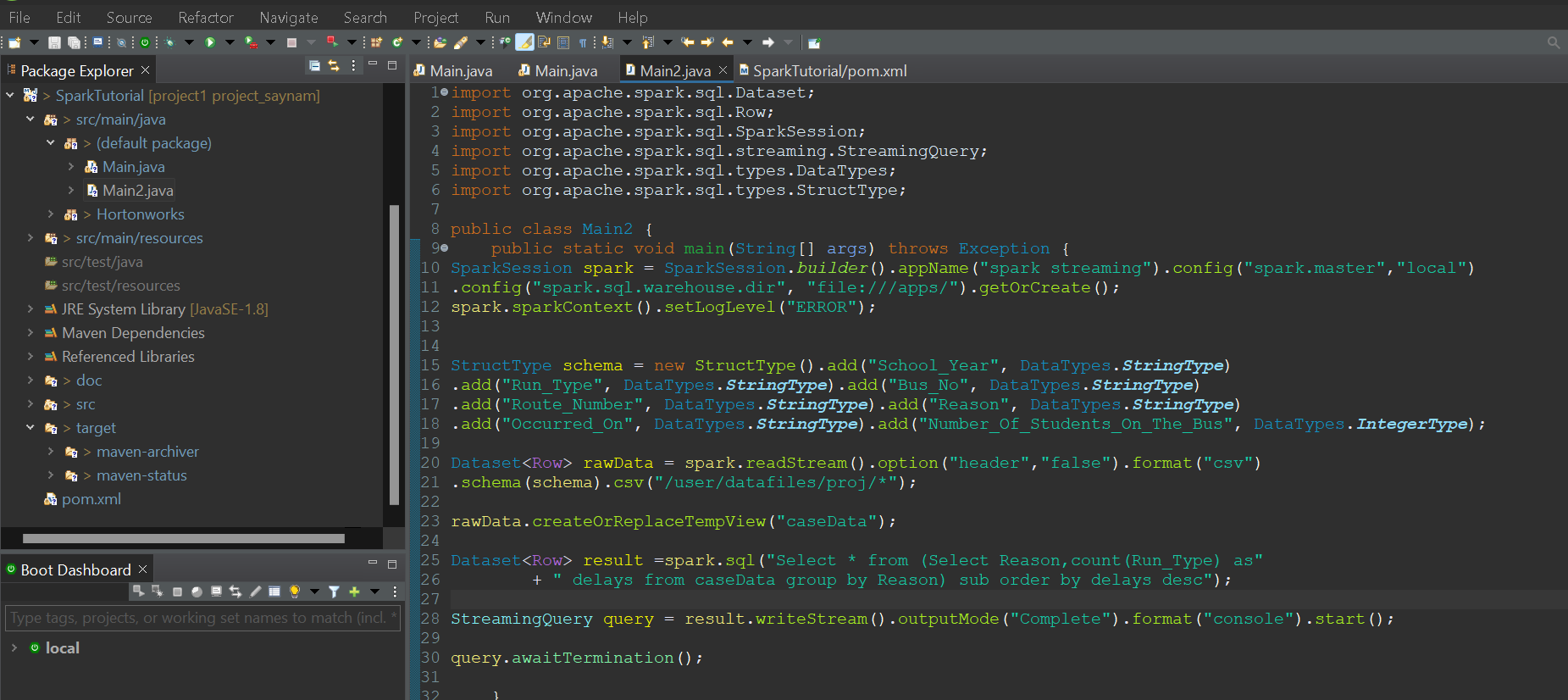
scala> val query = """ Select \* from (Select School\_year,count(run\_type) as count\_of\_accidents from caseData where Reason="Accident" group by School\_year) sub order by count\_of\_accidents """

scala> val answer4 = spark.sql(query)

scala> answer4.show



**JAVA Program**



**Execution**

