BDA2 - Spark SQL

Roshni Sundaramurthy (rossu809) and Yusur Al-Mter (yusal621)

May 12, 2019

Contents

Exercises BDA 2 using Spark SQL

Imports

```
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
```

```
######################## Temperature file ###########################################

# Load a text file and convert each line to a tuple.
sc = SparkContext()
rdd = sc.textFile("/user/x_rossu/bdlab1/temperature-readings.csv")

sqlContext = SQLContext(sc)
parts = rdd.map(lambda l: l.split(";"))
tempReadingsRow = parts.map(lambda p: (p[0], p[1], int(p[1].split("-")[0]),
int(p[1].split("-")[1]), p[2], float(p[3]), p[4] ))

# Specifying the schema programatically and registering the DataFrame as a table
tempReadingsString = ["station", "date", "year", "month", "time", "value",
"quality"]
```

```
# Apply the schema to the RDD.
schemaTempReadings = sqlContext.createDataFrame(tempReadingsRow,
tempReadingsString)

# Register the DataFrame as a table.
schemaTempReadings.registerTempTable("tempReadingsTable")

######################## Precipitation file ###########################################

# Data importing
rdd = sc.textFile("/user/x_rossu/bdlab1/precipitation-readings.csv")
parts = rdd.map(lambda l: l.split(";"))
precipReadingsRow = parts.map(lambda p: (int(p[0]), p[1],  int(p[1].split("-")[0]),
                               int(p[1].split("-")[1]), p[2], float(p[3]), p[4]))

precipReadingsString = ["station", "date", "year", "month", "time", "precipitation", "quality"]

# Apply the schema to the RDD
schemaPrecipReadings = sqlContext.createDataFrame(precipReadingsRow, precipReadingsString)

# Register the DataFrame as a table
schemaPrecipReadings.registerTempTable("precipReadingsTable")


######################## Station Ostergotland file ###########################################

rdd = sc.textFile("C:/Users/roshn/Desktop/Bigdata/stations-Ostergotland.csv")
parts = rdd.map(lambda l: l.split(";"))
StationReadingsRow = parts.map(lambda p: (int(p[0]), p[1] ))

StationReadingsString = ["station", "stn_name"]

# Apply the schema to the RDD
schemaStationReadings = sqlContext.createDataFrame(StationReadingsRow, StationReadingsString)

# Register the DataFrame as a table
schemaStationReadings.registerTempTable("StationReadingsTable")

# Can run queries now
```

1 Question 1.

year, station with the max, maxValue ORDER BY maxValue DESC
year, station with the min, minValue ORDER BY minValue DESC

1.1 Query

```
# Running SQL queries - API methods
```

```
#max temp

schemaTempReadingsMax = schemaTempReadings.filter(schemaTempReadings['year'] >= 1950)
                        .filter(schemaTempReadings.year <= 2014)
schemaTempReadingsMax = schemaTempReadingsMax.groupBy('year').agg(F.max('value').alias('value'))
                        .join(schemaTempReadings, ['year', 'value']).drop_duplicates(['year'])
                        .select(['year', 'station', 'value']).orderBy(['value'],ascending=[0])
schemaTempReadingsMax.show()
```

```
# min temp

schemaTempReadingsMin = schemaTempReadings.filter(schemaTempReadings['year'] >= 1950)
                        .filter(schemaTempReadings.year <= 2014)
schemaTempReadingsMin = schemaTempReadingsMin.groupBy('year').agg(F.min('value').alias('value'))
                        .join(schemaTempReadings, ['year', 'value']).dropDuplicates(['year'])
                        .select(['year', 'station', 'value']).orderBy(['value'],ascending=[0])
schemaTempReadingsMin.show()
```

1.2 Extract

Lowest and highest temperatures measured each year for the period 1950-2014:

```
# Max-temperatures

+----+-------+-----+
|year|station|value|
+----+-------+-----+
|1975|  86200| 36.1|
|1992|  63600| 35.4|
|1994| 117160| 34.7|
|2010|  75250| 34.4|
|2014|  96560| 34.4|
|1989|  63050| 33.9|
|1982|  94050| 33.8|
|1968| 137100| 33.7|
|1966| 151640| 33.5|
|1983|  98210| 33.3|
|2002|  78290| 33.3|
|1970| 103080| 33.2|
|1986|  76470| 33.2|
|2000|  62400| 33.0|
|1956| 145340| 33.0|
|1959|  65160| 32.8|
|2006|  75240| 32.7|
|1991| 137040| 32.7|
|1988| 102540| 32.6|
|2011| 172770| 32.5|
+----+-------+-----+
only showing top 20 rows


# Min-temperaures
```

```
+----+-------+-----+
|year|station|value|
+----+-------+-----+
|1990| 166870|-35.0|
|1952| 192830|-35.5|
|1974| 179950|-35.6|
|1954| 113410|-36.0|
|1992| 179960|-36.1|
|1975| 157860|-37.0|
|1972| 167860|-37.5|
|1995| 182910|-37.6|
|2000| 169860|-37.6|
|1957| 159970|-37.8|
|1989| 166870|-38.2|
|1983| 191900|-38.2|
|1953| 183760|-38.4|
|2009| 179960|-38.5|
|1993| 191900|-39.0|
|1984| 191900|-39.2|
|1991| 179960|-39.3|
|1973| 166870|-39.3|
|2008| 179960|-39.3|
|2005| 155790|-39.4|
+----+-------+-----+
only showing top 20 rows
```

2 Question 2.

year, month, value ORDER BY value DESC

year, month, value ORDER BY value DESC

2.1 Query using APIs

```
schemaTempReadingscount = schemaTempReadings.filter(schemaTempReadings['value'] > 10)
                          .filter(schemaTempReadings['year'] >= 1950)
                          .filter(schemaTempReadings.year <= 2014)
schemaTempReadingscount = schemaTempReadingscount.groupBy('year','month').agg(F.count('value')
                          .alias('count')).orderBy(['count'],ascending=[0])
schemaTempReadingscount.show()
```

```
schemaTempReadingscount = schemaTempReadings.filter(schemaTempReadings['value'] > 10)
                          .filter(schemaTempReadings['year'] >= 1950)
                          .filter(schemaTempReadings.year <= 2014)
schemaTempReadingscount = schemaTempReadingscount.groupBy('year','month')
                          .agg(F.countDistinct('station')
                          .alias('count')).orderBy(['count'],ascending=[0])
schemaTempReadingscount.show()
```

2.2 Regular query

```
count = sqlContext.sql("SELECT year,month, count(value) as value FROM tempReadingsTable
                       WHERE year>=1950 and year<=2014 and value>=10.0
                       GROUP BY year, month ORDER BY value DESC")
count.show()
```

```
count = sqlContext.sql("SELECT year,month, count(tmp) as value
                       FROM (SELECT year, month, station, count(value) as tmp
                       FROM tempReadingsTable
                       WHERE year>=1950 and year<=2014 and value>=10.0
                       GROUP BY year, month, station)
                       GROUP BY year, month ORDER BY value DESC")
count.show()
```

2.3 Extract

Number of readings above 10 degrees for each month in the period of 1950-2014:

```
|year|month| value|
+----+-----+------+
|2014|    7|147910|
|2011|    7|147060|
|2010|    7|143860|
|2012|    7|138166|
|2013|    7|134297|
|2009|    7|133570|
|2011|    8|133483|
|2009|    8|129007|
|2013|    8|128920|
|2003|    7|128360|
|2002|    7|128354|
|2006|    8|128039|
|2008|    7|127627|
|2002|    8|126495|
|2011|    6|126084|
|2012|    8|125947|
|2005|    7|125651|
|2006|    7|125192|
|2010|    8|125135|
|2014|    8|125006|
+----+-----+------+
```

Distinct readings above 10 degrees for each month in the period of 1950-2014:

```
|year|month|count|
+----+-----+-----+
|1972|   10|  378|
|1973|    5|  377|
|1973|    6|  377|
|1973|    9|  376|
|1972|    8|  376|
|1972|    6|  375|
|1972|    9|  375|
|1972|    5|  375|
|1971|    8|  375|
|1972|    7|  374|
|1971|    9|  374|
|1971|    6|  374|
|1971|    5|  373|
|1973|    8|  373|
|1974|    8|  372|
|1974|    6|  372|
|1973|    7|  370|
|1970|    8|  370|
|1974|    9|  370|
|1971|    7|  370|
+----+-----+-----+
```

3 Question 3.

year, month, station, avgMonthlyTemperature ORDER BY avgMonthlyTemperature DESC

3.1 Query

```
schemayearReadings = schemaTempReadings.filter(schemaTempReadings.year >= 1960)
                                       .filter(schemaTempReadings.year <= 2014)
avg_temp = schemayearReadings.groupBy(['station','date', 'year', 'month'])
                            .agg(F.max('value').alias('max_temp')
                                 ,F.min('value').alias('min_temp'))
avg_temp = avg_temp.select(['station', 'year', 'month', 'min_temp', 'max_temp'])
avg_temp = avg_temp.withColumn('sum', avg_temp['min_temp'] + avg_temp['max_temp'])
                    .groupBy(['station', 'year', 'month']).agg(F.avg('sum').alias('value'))
avg_temp = avg_temp.withColumn('avg_temp', avg_temp['value']/2)
                  .select(['year', 'month', 'station', 'avg_temp'])
                  .orderBy('avg_temp', ascending = [0])

avg_temp.show()
```

3.2 Extract

Average monthly temperature for each available station in Sweden in the period of 1960-2014:

```
+----+-----+-------+------------------+
|year|month|station|          avg_temp|
+----+-----+-------+------------------+
|2014|    7|  96000|              26.3|
|1994|    7|  96550|23.071052631578947|
|1983|    8|  54550|              23.0|
|1994|    7|  78140|22.970967741935485|
|1994|    7|  85280| 22.87258064516129|
|1994|    7|  75120|22.858064516129037|
|1994|    7|  65450| 22.85645161290323|
|1994|    7|  96000|22.808064516129033|
|1994|    7|  95160|22.764516129032256|
|1994|    7|  86200| 22.71129032258065|
|2002|    8|  78140|              22.7|
|1994|    7|  76000|22.698387096774194|
|1997|    8|  78140|22.666129032258066|
|1994|    7| 105260| 22.65967741935484|
|1975|    8|  54550|22.642857142857142|
|2006|    7|  76530|22.598387096774193|
|1994|    7|  86330| 22.54838709677419|
|2006|    7|  75120| 22.52741935483871|
|1994|    7|  54300|22.469354838709677|
|2006|    7|  78140|22.458064516129028|
+----+-----+-------+------------------+
```

4 Question 4.

station, maxTemp, maxDailyPrecipitation ORDER BY station DESC

4.1 Query

```
# Temp
max_temp = schemaTempReadings.groupBy('station').agg(F.max('value').alias('maxtempvalue'))
max_temp = max_temp.filter(max_temp['maxtempvalue'] >= 25)
                   .filter(max_temp['maxtempvalue'] <=30)
                   .orderBy(['station'],ascending=[0])
max_temp.show()

# Precipitation
max_precip = schemaPrecipReadings.groupBy('station', 'date')
                                 .agg(F.sum('precipitation').alias('total_precip'))
max_precip = max_precip.select('station', 'total_precip').groupBy('station')
                       .agg(F.max('total_precip').alias('maxprecipvalue'))
max_precip = max_precip.filter(max_precip['maxprecipvalue'] >= 100)
                       .filter(max_precip['maxprecipvalue'] <= 200)
                       .orderBy(['station'],ascending=[0])
max_precip.show()
```

4.2 Extract

Stations with their associated maximum measured temperatures and maximum measured daily precipitation:

```
# Max_Temperatures
|station|maxtempvalue|
+-------+------------+
|  99450|        26.0|
|  99280|        25.5|
|  99270|        27.8|
|  99090|        27.6|
|  98610|        29.2|
|  98170|        27.0|
|  98140|        26.4|
|  96600|        26.2|
|  96370|        30.0|
|  96220|        30.0|
|  95640|        29.5|
|  95620|        29.8|
|  95380|        26.1|
|  95230|        29.9|
|  94660|        29.3|
|  94450|        29.5|
|  94190|        30.0|
|  93640|        30.0|
|  93250|        28.6|
|  91620|        30.0|
+-------+------------+


# Max_precipitation

+-------+------------------+
|station|    maxprecipvalue|
+-------+------------------+
|  97510|103.99999999999999|
|  75250|             101.8|
|  71420|             106.3|
|  52350|             101.6|
+-------+------------------+
```

5 Question 5.

year, month, avgMonthlyPrecipitation ORDER BY year DESC, month DESC

5.1 Query

```
precip = schemaStationReadings.join(schemaPrecipReadings, ['station'])
precip = precip.select('station', 'year', 'month', 'precipitation')
              .filter(precip['year'] >= 1993).filter(precip.year <= 2016)
precip = precip.groupBy('station', 'year', 'month')
               .agg(F.sum('precipitation').alias('precip'))
precip = precip.select('year', 'month', 'precip')
precip = precip.groupBy('year', 'month').agg(F.avg('precip').alias('avg_precip'))
              .orderBy(['year', 'month'], ascending = False)

precip.show()
```

5.2 Extract

Average monthly precipitation for the Östergotland region for the period 1993-2016:

```
+----+-----+------------------+
|year|month|        avg_precip|
+----+-----+------------------+
|2016|    7|               0.0|
|2016|    6|47.662499999999994|
|2016|    5|29.250000000000004|
|2016|    4| 26.90000000000001|
|2016|    3|19.962500000000002|
|2016|    2|           21.5625|
|2016|    1|            22.325|
|2015|   12|28.925000000000004|
|2015|   11| 63.88750000000001|
|2015|   10|            2.2625|
|2015|    9|101.29999999999997|
|2015|    8|           26.9875|
|2015|    7|119.09999999999997|
|2015|    6| 78.66250000000001|
|2015|    5| 93.22499999999998|
|2015|    4|15.337499999999999|
|2015|    3| 42.61250000000001|
|2015|    2|24.824999999999996|
|2015|    1|59.112500000000026|
|2014|   12| 35.46250000000001|
+----+-----+------------------+
```

6 Question 6.

year, month, difference ORDER BY year DESC, month DESC

6.1 Query

```
temp = schemaStationReadings.join(schemaTempReadings, ['station'])
temp = temp.select('station', 'date', 'year', 'month', 'value')
            .filter(temp['year'] >= 1950).filter(temp['year'] <= 2014)
temp = temp.groupBy(['station', 'date', 'year', 'month'])
            .agg(F.min('value').alias('min'), F.max('value').alias('max'))
temp = temp.select(['station', 'year', 'month', 'min', 'max'])
            .withColumn('sum_temp', temp['min'] + temp['max'])
temp = temp.groupBy(['station', 'year', 'month']).agg(F.avg('sum_temp').alias('value'))
temp = temp.withColumn('avg_temp', temp.value/2)
avg_temp = temp.select(['year', 'month', 'station', 'avg_temp'])
              .groupBy(['year', 'month']).agg(F.avg('avg_temp').alias('avg_temp'))
long_temp = avg_temp.filter(avg_temp['year'] >= 1950).filter(avg_temp['year'] <= 1980)
long_temp = long_temp.groupBy('month').agg(F.avg('avg_temp')
                      .alias('long_temp')).join(avg_temp, 'month')
long_temp = long_temp.withColumn('Compared_Temperature'
                                 , long_temp['avg_temp'] - long_temp['long_temp'])
long_temp = long_temp.select(['year', 'month', 'Compared_Temperature'])
long_temp = long_temp.orderBy(['year', 'month'], ascending = False)
long_temp.show()
```

```
# if plot
#long_temp.rdd.coalesce(1).saveAsTextFile("comp_temp")
```

6.2 Extract

Compared average monthly temperature in the period 1950-2014 with long-term monthly averages in the period of 1950-1980:

```
+----+-----+--------------------+
|year|month Compared_temperature|
+----+-----+--------------------+
|2014|   12|  0.8238893537957012|
|2014|   11|  2.0635396726928987|
|2014|   10|  1.5225549840378134|
|2014|    9| 0.06105818643721861|
|2014|    8| -0.6426470719706909|
|2014|    7|  2.0939107758930824|
|2014|    6| -1.8073686197315197|
|2014|    5| 0.26719065014069976|
|2014|    4|  2.0661931589915454|
|2014|    3|   4.486748343574566|
|2014|    2|   5.420311314566043|
|2014|    1|  0.9325880207201753|
|2013|   12|  3.8796729966761174|
|2013|   11|  0.9342517939050206|
|2013|   10|  0.7529068901961731|
|2013|    9|   -1.00505751604212|
|2013|    8|-0.31464120686804975|
|2013|    7|0.008280277359357768|
|2013|    6| -0.5441868015497029|
|2013|    5|   1.573920855419292|
+----+-----+--------------------+
```