BDA2 Spark Sql

Martynas Lukosevicius, Alejo Perez Gomez

26/04/2021

1)

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

sc = SparkContext(appName = "exercise 1")

# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))

# dataframe
data_temp = lines.map(lambda x: Row(year = int(x[1][0:4]), station = x[0], value = float(x[3])))

sqlContext = SQLContext(sc)

data = sqlContext.createDataFrame(data_temp)
data.registerTempTable("data_temp")

#filter
data_selected = data.filter((data["year"]>=1950) & (data["year"]<=2014)).groupBy('year').agg(F.first("s

#print(max_temperatures.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

Row(year=1975, station=u'133470', maxvalue=36.1)

Row(year=1992, station=u'102210', maxvalue=35.4)

Row(year=1994, station=u'123250', maxvalue=34.7)

Row(year=2014, station=u'106160', maxvalue=34.4)

Row(year=2010, station=u'108320', maxvalue=34.4)

Row(year=1989, station=u'112080', maxvalue=33.9)

Row(year=1982, station=u'123250', maxvalue=33.8)

Row(year=1968, station=u'133470', maxvalue=33.7)

Row(year=1966, station=u'108640', maxvalue=33.5)

Row(year=1983, station=u'133260', maxvalue=33.3)

Row(year=2002, station=u'140360', maxvalue=33.3)

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

sc = SparkContext(appName = "exercise 1")

# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))

# dataframe
data_temp = lines.map(lambda x: Row(year = int(x[1][0:4]), station = x[0], value = float(x[3])))

sqlContext = SQLContext(sc)

data = sqlContext.createDataFrame(data_temp)
data.registerTempTable("data_temp")

#filter
data_selected = data.filter((data["year"]>=1950) & (data["year"]<=2014)).groupBy('year').agg(F.first("s

#print(max_temperatures.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

Row(year=1990, station=u'133260', minvalue=-35.0)

Row(year=1952, station=u'124020', minvalue=-35.5)

Row(year=1974, station=u'112080', minvalue=-35.6)

Row(year=1954, station=u'108640', minvalue=-36.0)

Row(year=1992, station=u'123250', minvalue=-36.1)

Row(year=1975, station=u'106270', minvalue=-37.0)

Row(year=1972, station=u'140480', minvalue=-37.5)

2)

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

sc = SparkContext(appName = "exercise 1")

# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
```

```
lines = temperature_file.map(lambda line: line.split(";"))

# dataframe
data_temp = lines.map(lambda x: Row(year = int(x[1][0:4]), month = int(x[1][5:7]), temp = float(x[3])))
sqlContext = SQLContext(sc)
data = sqlContext.createDataFrame(data_temp)
data.registerTempTable("data")
#filter
data_selected = data.filter((data["year"]>=1950) & (data["year"]<=2014) & (data["temp"]> 10)).groupBy('
.agg( F.count("temp").alias("count"))\
.orderBy(['count'], ascending = False)
#print(max_temperatures.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

Row(year=2014, month=7, count=147681)

Row(year=2011, month=7, count=146656)

Row(year=2010, month=7, count=143419)

Row(year=2012, month=7, count=137477)

Row(year=2013, month=7, count=133657)

Row(year=2009, month=7, count=133008)

Row(year=2011, month=8, count=132734)

Row(year=2009, month=8, count=128349)

Row(year=2013, month=8, count=128235)

Row(year=2003, month=7, count=128133)

Distinct:

Row(year=1972, month=10, count=378)

Row(year=1973, month=6, count=377)

Row(year=1973, month=5, count=377)

Row(year=1972, month=8, count=376)

Row(year=1973, month=9, count=376)

Row(year=1972, month=5, count=375)

Row(year=1972, month=9, count=375)

Row(year=1972, month=6, count=375)

Row(year=1971, month=8, count=375)

Row(year=1971, month=6, count=374)

Row(year=1972, month=7, count=374)

Row(year=1971, month=9, count=374)

3)

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

sc = SparkContext(appName = "exercise 1")

# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))

# dataframe
data_temp = lines.map(lambda x: Row(year = int(x[1][0:4]), month = int(x[1][5:7]), day = int(x[1][8:10]
sqlContext = SQLContext(sc)
data = sqlContext.createDataFrame(data_temp)
data.registerTempTable("data")
#filter
data_selected = data.filter((data["year"]>=1950) & (data["year"]<=2014)).groupBy('year', "month", "day"
.agg( F.min("temp").alias("min"), F.max("temp").alias("max"))\
.groupBy('year', "month") \
.agg( F.sum("min").alias("min"), F.sum("max").alias("max"), F.count("min").alias("count"))\
.withColumn("average", ((F.col("min")+F.col("max")) / (2*F.col("count")))).drop("min", "max", "count")\
.sort("average", ascending = False)
#print(max_temperatures.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

```
Row(year=2014, month=7, station=u’96000’, average=26.3)
Row(year=1994, month=7, station=u’96550’, average=23.071052631578947)
Row(year=1983, month=8, station=u’54550’, average=23.0)
Row(year=1994, month=7, station=u’78140’, average=22.970967741935482)
Row(year=1994, month=7, station=u’85280’, average=22.87258064516129)
Row(year=1994, month=7, station=u’75120’, average=22.858064516129033)
Row(year=1994, month=7, station=u’65450’, average=22.85645161290323)
Row(year=1994, month=7, station=u’96000’, average=22.80806451612903)
Row(year=1994, month=7, station=u’95160’, average=22.76451612903226)
Row(year=1994, month=7, station=u’86200’, average=22.711290322580645)
Row(year=2002, month=8, station=u’78140’, average=22.7)
Row(year=1994, month=7, station=u’76000’, average=22.698387096774198)
Row(year=1997, month=8, station=u’78140’, average=22.666129032258063)
```

4)

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

sc = SparkContext(appName = "exercise 1")

# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
temp = temperature_file.map(lambda line: line.split(";")).map(lambda x: Row(station = x[0], temp = floa
prec = precipitation_file.map(lambda line: line.split(";")).map(lambda x: Row(station = x[0], prec = fl

print("test1")
# dataframe
sqlContext = SQLContext(sc)
df_temp = sqlContext.createDataFrame(temp)
df_temp.registerTempTable("temp")

df_prec = sqlContext.createDataFrame(prec)
df_prec.registerTempTable("prec")
#filter

df_prec = df_prec.groupBy("station").agg(F.max("prec").alias("prec"))
df_temp = df_temp.groupBy("station").agg(F.max("temp").alias("temp"))

data_selected = df_temp.join(df_prec, ["station"])

data_selected = data_selected.filter((data_selected["temp"] >= 25) & (data_selected["temp"] <= 30) & (da
    .orderBy('station', ascending=False)

print("test4")

#print(max_temperatures.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

empty

5)

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


sc = SparkContext(appName = "exercise 1")
sqlContext = SQLContext(sc)
stations = sc.textFile("BDA/input/stations-Ostergotland.csv").map(lambda x: x.split(";")[0]).collect()
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")

prec = precipitation_file.map(lambda line: line.split(";"))
prec = prec.map(lambda x: Row(year = int(x[1][0:4]), month = int(x[1][5:7]), station = x[0], day = int(

df_prec = sqlContext.createDataFrame(prec)
df_prec.registerTempTable("prec")



#filter
data_selected = df_prec.where((F.col("year")>=1993) & (F.col("year")<=2016) & (F.col("station").isin(sta
    .agg( F.sum("prec").alias("prec"))\
    .groupBy("year", "month").agg( F.avg("prec").alias("average"))\
    .sort("average", ascending = False)

data_selected.rdd.saveAsTextFile("BDA/output")
```

Results:

```
+----+-----+------------------+
|year|month|           average|
+----+-----+------------------+
|2006|    8| 148.0833333333333|
|2008|    8|138.51666666666657|
|2000|    7|135.86666666666662|
|1995|    9|            134.55|
|2012|    6|132.19999999999996|
|2015|    7|119.09999999999997|
|2006|   10|118.16666666666664|
|2003|    7|113.46666666666665|
|2009|    7|113.16666666666664|
|2001|    9|110.63333333333327|
|2000|   10|110.29999999999997|
|2007|    6|            108.95|
|2000|   11| 108.1166666666666|
|2010|    8|            108.05|
|2005|    7|104.34999999999997|
|2015|    9|             101.3|
|2002|    6|  98.7833333333333|
|1995|    6| 97.19999999999987|
|2004|    7| 95.99999999999996|
|2007|    7| 95.96666666666664|
+----+-----+------------------+
```