BDA - Lab 2 : Spark SQL

Student1: Akshay Gurudath(Aksgu350)

Student2: Keshav Padiyar Manuru(Kespa139)

In [ ]:

```
from pyspark import SparkContext
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from operator import add
import sys

# Set up Spark Context
sc = SparkContext(appName = "BDA Lab2")
spark = SparkSession.builder.getOrCreate()

# Reading Data
df_tempReadings = spark.read.csv("file:///home/x_kesma/Lab1/input_data/temperature-read
ings.csv", header = False, sep = ';' )
df_tempReadings = df_tempReadings.withColumnRenamed("_c0", "stationNumber")\
                                 .withColumnRenamed("_c1", "date")\
                                 .withColumnRenamed("_c2", "time")\
                                 .withColumnRenamed("_c3", "airTemperature")\
                                 .withColumnRenamed("_c4", "quality")

df_precipitation = spark.read.csv("file:///home/x_kesma/Lab1/input_data/precipitation-r
eadings.csv", header = False, sep = ';' )
df_precipitation = df_precipitation.withColumnRenamed("_c0", "stationNumber")\
                                 .withColumnRenamed("_c1", "date")\
                                 .withColumnRenamed("_c2", "time")\
                                 .withColumnRenamed("_c3", "precipitation")\
                                 .withColumnRenamed("_c4", "quality")

rdd_OstStations = sc.textFile("file:///home/x_kesma/Lab1/input_data/stations-Ostergotla
nd.csv")\
                            .map(lambda line: line.split(";"))\
                            .map(lambda line:line[0])
```

In [ ]:

```
# Assignment 1: What are the lowest and highest temperatures measured each year for the
period 1950-2014.
# Using Dataframes


df_filtered_1 = df_tempReadings.select("stationNumber", F.year(F.col('date')).alias("Ye
ar"),\
                                       F.col("airTemperature").cast("float"))\
                                .filter((F.col("Year")>=1950) & ((F.col("Year")<=2014)))

out = df_filtered_1.groupBy("Year")\
            .agg(F.min('airTemperature').alias('MinTemp'),F.max('airTemperature').alias
('MaxTemp'))\
            .orderBy("Year")

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q1"
,sep=",", header=True)
```

Result:

	Year	MinTemp	MaxTemp
2	2014	-42.5	34.4
3	2013	-40.7	31.6
4	2012	-42.7	31.3
5	2011	-42	32.5
6	2010	-41.7	34.4
7	2009	-38.5	31.5
8	2008	-39.3	32.2
9	2007	-40.7	32.2
10	2006	-40.6	32.7
11	2005	-39.4	32.1
12	2004	-39.7	30.2
13	2003	-41.5	32.2
14	2002	-42.2	33.3
15	2001	-44	31.9
16	2000	-37.6	33
17	1999	-49	32.4
18	1998	-42.7	29.2
19	1997	-40.2	31.8
20	1996	-41.7	30.8
21	1995	-37.6	30.8

In [ ]:

```
# 2_1 Count the number of readings for each month in the period of 1950-2014 which are
 higher than 10 degrees

df_filtered_2 = df_tempReadings.select("stationNumber", F.year(F.col('date')).alias("Ye
ar"),\
                                       F.month(F.col("date")).alias("Month"),\
                                       F.col("airTemperature").cast("float"))\
                               .filter(((F.col("Year")>=1950) & ((F.col("Year")<=2014)))
&(F.col("airTemperature")>10))

out = df_filtered_2.groupBy("Year", "Month")\
             .agg(F.count("stationNumber").alias("Value"))\
             .orderBy("value",ascending=False)

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q2_
1",sep=",", header=True)
```

Result:

	Year	Month	Value
2	2014	7	147681
3	2011	7	146656
4	2010	7	143419
5	2012	7	137477
6	2013	7	133657
7	2009	7	133008
8	2011	8	132734
9	2009	8	128349
10	2013	8	128235
11	2003	7	128133
12	2002	7	127956
13	2006	8	127622
14	2008	7	126973
15	2002	8	126073
16	2005	7	125294
17	2011	6	125193
18	2012	8	125037
19	2006	7	124794
20	2010	8	124417
21	2014	8	124045

In [ ]:

```
# 2_2 Repeat the exercise,this time taking only distinct readings from each station.
# That is, if a station reported a reading above 10 degrees in some month, then itappea
rs only
# once in the count for that month

out = df_filtered_2.groupBy("Year", "Month")\
             .agg(F.countDistinct("stationNumber").alias("Value"))\
             .orderBy("value",ascending=False)

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q2_
2",sep=",", header=True)
```

Result:

	Year	Month	Value
2	1972	10	378
3	1973	5	377
4	1973	6	377
5	1973	9	376
6	1972	8	376
7	1972	6	375
8	1972	5	375
9	1971	8	375
10	1972	9	375
11	1971	6	374
12	1971	9	374
13	1972	7	374
14	1971	5	373
15	1973	8	373
16	1974	8	372
17	1974	6	372
18	1974	9	370
19	1970	8	370
20	1973	7	370
21	1974	5	370

In [ ]:

```
# 3 Find the average monthly temperature for each available station in Sweden. Your res
ult
#should include average temperature for each station for each month in the period of 19
60-
#2014. Bear in mind that not every station has the readings for each month in this time
frame.

df_filtered_3 = df_tempReadings.select("stationNumber", F.year(F.col('date')).alias("Ye
ar"),\
                                       F.month(F.col("date")).alias("Month"),\
                                       F.col("airTemperature").cast("float"))\
                               .filter((F.col("Year")>=1960) & ((F.col("Year")<=2014)))

out = df_filtered_3.groupBy("stationNumber","Year", "Month")\
             .agg(F.avg("airTemperature").alias("avgMonthlyTemperature"))\
             .orderBy("stationNumber","Year", "Month",ascending=False)

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q3"
,sep=",", header=True)
```

Result:

	stationNu	Year	Month	avgMonthlyTemperature
2	99450	2014	12	1.989784944
3	99450	2014	11	5.973888883
4	99450	2014	10	9.300811914
5	99450	2014	9	13.71222223
6	99450	2014	8	16.91505378
7	99450	2014	7	18.45551076
8	99450	2014	6	11.00694446
9	99450	2014	5	7.565456982
10	99450	2014	4	4.473472222
11	99450	2014	3	2.797446236
12	99450	2014	2	1.833333333
13	99450	2014	1	-0.976478491
14	99450	2013	12	3.663907734
15	99450	2013	11	5.528194455
16	99450	2013	10	9.186290327
17	99450	2013	9	13.62097221
18	99450	2013	8	17.18333336
19	99450	2013	7	15.39784947
20	99450	2013	6	13.85180556
21	99450	2013	5	8.732795694

In [ ]:

```
### Fixed Code: Added additonal group by to get the daily max precipitation.

# 4 Provide a list of stations with their associated maximum measured temperatures and
# maximum measured daily precipitation. Show only those stations where the maximum
# temperature is between 25 and 30 degrees and maximum daily precipitation is between 1
00mm and 200mm


df_filtered_temp = df_tempReadings.select("stationNumber",\
                                    F.col("airTemperature").cast("float"))\
                                  .groupBy("stationNumber")\
                                  .agg(F.max("airTemperature").alias("maxTemp"))\
                                  .filter((F.col("maxTemp")>=25) & ((F.col("maxTemp")<=
30)))

df_filtered_preci = df_precipitation.select("stationNumber","date",\
                                    F.col("precipitation").cast("float"))\
                                .groupBy("stationNumber")\
                                .agg(F.sum("precipitation").alias("precipitation"))\
                                .select("stationNumber","precipitation")\
                                .groupBy("stationNumber")\
                                .agg(F.max("precipitation").alias("maxDailyPrecipitation"
))\
                                .filter((F.col("maxDailyPrecipitation")>=100) & ((F.col(
"maxDailyPrecipitation")<=200)))

out = df_filtered_temp.alias("a").join(df_filtered_preci.alias("b"),
                                       F.col("a.stationNumber")==F.col("b.stationNu
mber"),"inner")\
                                  .select("a.stationNumber", "a.maxTemp", "b.maxDail
yPrecipitation")

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q4"
,sep=",", header=True)
```

Result:

No Resultset Obtained

In [ ]:

```
# 5 Calculate the average monthly precipitation for the Ostergotland region (list of st
ations is provided in the separate file)
# for the period 1993-2016. In order to do this, you will first need to calculate the t
otalmonthly precipitation for each
# station before calculating the monthly average (by averaging over stations).

list_OstStations = rdd_OstStations.collect()

broadcastVar = sc.broadcast(list_OstStations)

df_filtered_preci_5 = df_precipitation.select("stationNumber",\
                                      F.year(F.col('date')).alias("Year"),\
                                      F.month(F.col("date")).alias("Month"),\
                                      F.col("precipitation").cast("float"))\
                                .filter(((F.col("Year")>=1993) & ((F.col("Year")<=2016)))
& (F.col("stationNumber").isin(broadcastVar.value)))

out = df_filtered_preci_5.groupBy("Year", "Month","stationNumber")\
             .agg(F.sum("precipitation").alias("Sum"))\
             .groupBy("Year", "Month")\
             .agg(F.avg("Sum").alias("avgMonthlyPrecipitation"))\
             .orderBy("year","Month",ascending=False)

out.repartition(1).write.csv("file:///home/x_kesma/Lab1/input_data/results/BDA_LAB2/Q5"
,sep=",", header=True)

sys.exit(0)
```

Result:

	Year	Month	avgMonthlyPrecipitation
2	2016	7	0
3	2016	6	47.66250009
4	2016	5	29.2500002
5	2016	4	26.90000024
6	2016	3	19.96250029
7	2016	2	21.56250028
8	2016	1	22.32500034
9	2015	12	28.92500019
10	2015	11	63.88750029
11	2015	10	2.262500035
12	2015	9	101.3000003
13	2015	8	26.98750011
14	2015	7	119.0999999
15	2015	6	78.66250023
16	2015	5	93.2250002
17	2015	4	15.33750008
18	2015	3	42.61250029
19	2015	2	24.82500036
20	2015	1	59.11250053
21	2014	12	35.46250028