BDA Spark SQL

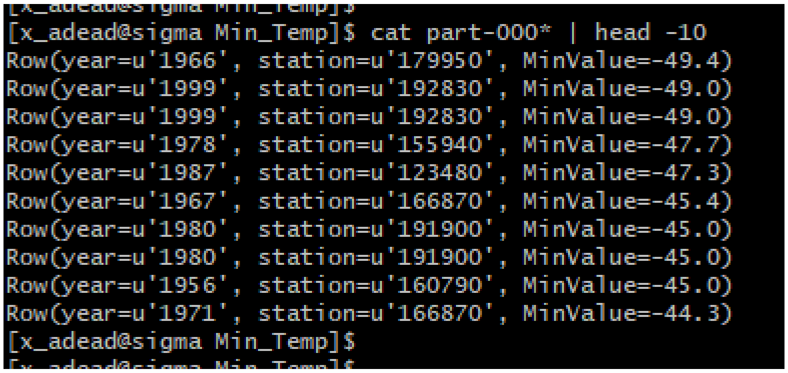
Mohammed Ali{mohal954} and Adesijibomi Aderinto{adead268}

5/10/2022

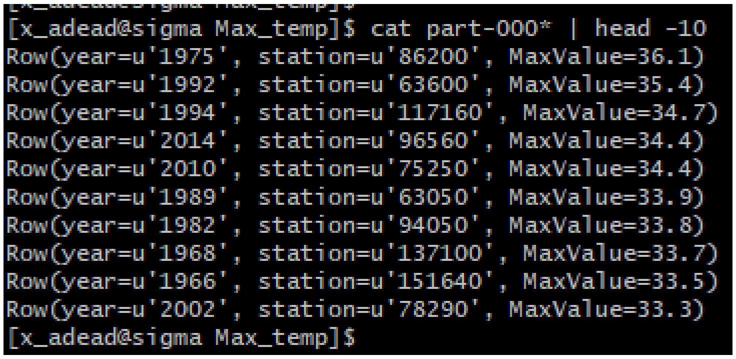
Question 1

```
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
sc =SparkContext()
sqlContext=SQLContext(sc)

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
parts = temperature_file.map(lambda l:l.split(";"))

tempReadings = parts.map(lambda p: Row(station=p[0],  date=p[1], year=p[1].split("-")[0],
time=p[2],  Temp=float(p[3]), quality=p[4]))

TableReadings = sqlContext.createDataFrame(tempReadings)
TableReadings.registerTempTable("tempReadings")

maximum_temperature = TableReadings.select(["year","station","Temp"])
.filter((TableReadings['year'] <=2014)& (TableReadings["year"]>=1950))
.groupby(['year','station']).agg(F.max('Temp')).groupby("year")
.agg(F.max("max(Temp)").alias("Temp")).orderBy('Temp',ascending=False)

maximum_temperature1 = maximum_temperature.join(TableReadings,["Temp","year"],"inner")

maximum_temperature2 = maximum_temperature1.select(["year","station","Temp"])
.orderBy("Temp",ascending= False).withColumnRenamed("Temp","MaxValue")

minimum_temperature = TableReadings.select(["year","station","Temp"])
.filter((TableReadings['year'] <=2014)& (TableReadings["year"]>=1950))
.groupby(['year','station']).agg(F.min('Temp')).groupby("year")
.agg(F.min("min(Temp)").alias("Temp")).orderBy('Temp',ascending=True)

minimum_temperature1 = minimum_temperature.join(TableReadings,["Temp","year"],"inner")

minimum_temperature2 = minimum_temperature1.select(["year","station","Temp"])
.orderBy("Temp",ascending= True).withColumnRenamed("Temp","MinValue")

maximum_temperature2_rdd = maximum_temperature2.rdd
minimum_temperature2_rdd = minimum_temperature2.rdd
```

```
maximum_temperature2_rdd.saveAsTextFile("BDA/output/Max_temp")
minimum_temperature2_rdd.saveAsTextFile("BDA/output/Min_Temp")
```



```
[x_adead@sigma Min_Temp]$
[x_adead@sigma Min_Temp]$ cat part-000* | head -10
Row(year=u'1966', station=u'179950', MinValue=-49.4)
Row(year=u'1999', station=u'192830', MinValue=-49.0)
Row(year=u'1999', station=u'192830', MinValue=-49.0)
Row(year=u'1978', station=u'155940', MinValue=-47.7)
Row(year=u'1987', station=u'123480', MinValue=-47.3)
Row(year=u'1967', station=u'166870', MinValue=-45.4)
Row(year=u'1980', station=u'191900', MinValue=-45.0)
Row(year=u'1980', station=u'191900', MinValue=-45.0)
Row(year=u'1956', station=u'160790', MinValue=-45.0)
Row(year=u'1971', station=u'166870', MinValue=-44.3)
[x_adead@sigma Min_Temp]$
[x_adead@sigma Min_Temp]$
```

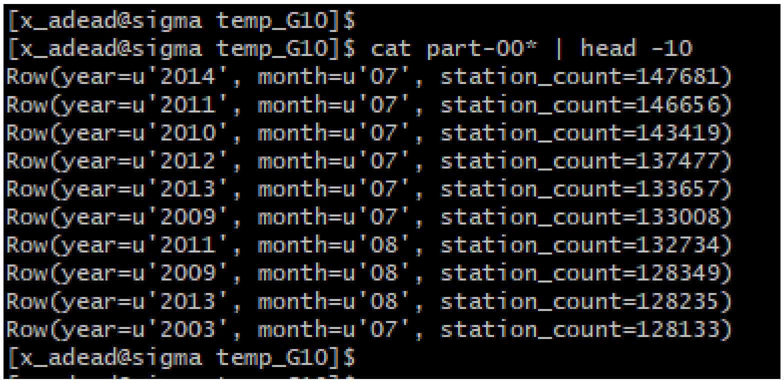


```
[x_adead@sigma Max_temp]$
[x_adead@sigma Max_temp]$ cat part-000* | head -10
Row(year=u'1975', station=u'86200', MaxValue=36.1)
Row(year=u'1992', station=u'63600', MaxValue=35.4)
Row(year=u'1994', station=u'117160', MaxValue=34.7)
Row(year=u'2014', station=u'96560', MaxValue=34.4)
Row(year=u'2010', station=u'75250', MaxValue=34.4)
Row(year=u'1989', station=u'63050', MaxValue=33.9)
Row(year=u'1982', station=u'94050', MaxValue=33.8)
Row(year=u'1968', station=u'137100', MaxValue=33.7)
Row(year=u'1966', station=u'151640', MaxValue=33.5)
Row(year=u'2002', station=u'78290', MaxValue=33.3)
[x_adead@sigma Max_temp]$
```

Question 2

Question 2a

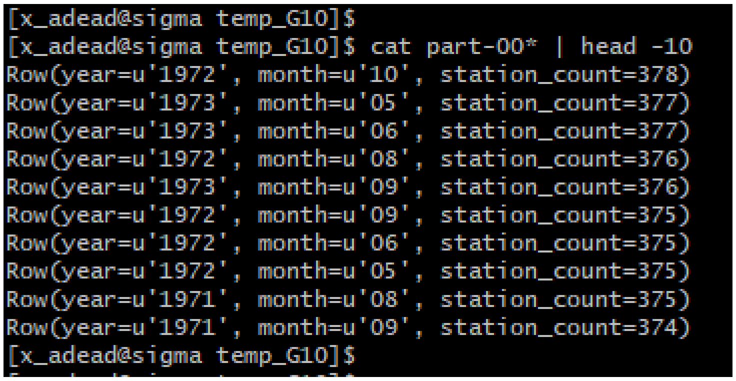
```
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
sc =SparkContext(appName = "exercise 1")
sqlContext=SQLContext(sc)

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
parts = temperature_file.map(lambda l:l.split(";"))

tempReadings = parts.map(lambda p: Row(station=p[0],date=p[1],year=p[1].split("-")[0]
                                       ,month=p[1].split("-")[1],time=p[2],
                                       Temp=float(p[3]),quality=p[4]))

TableReadings = sqlContext.createDataFrame(tempReadings)
TableReadings.registerTempTable("tempReadings")

TempAbove10 = TableReadings.select(["year","month","station"])
.filter((TableReadings['year'] <=2014) & (TableReadings["year"]>=1950)
& (TableReadings["Temp"] > 10)).groupby(["year","month"])
.agg(F.count("station").alias("station_count")).orderBy("station_count",ascending=False)

TempAbove10.show()

TempAbove10.rdd.saveAsTextFile("BDA/output/temp_G10")
```



```
[x_adead@sigma temp_G10]$
[x_adead@sigma temp_G10]$ cat part-00* | head -10
Row(year=u'2014', month=u'07', station_count=147681)
Row(year=u'2011', month=u'07', station_count=146656)
Row(year=u'2010', month=u'07', station_count=143419)
Row(year=u'2012', month=u'07', station_count=137477)
Row(year=u'2013', month=u'07', station_count=133657)
Row(year=u'2009', month=u'07', station_count=133008)
Row(year=u'2011', month=u'08', station_count=132734)
Row(year=u'2009', month=u'08', station_count=128349)
Row(year=u'2013', month=u'08', station_count=128235)
Row(year=u'2003', month=u'07', station_count=128133)
[x_adead@sigma temp_G10]$
```

Question 2B

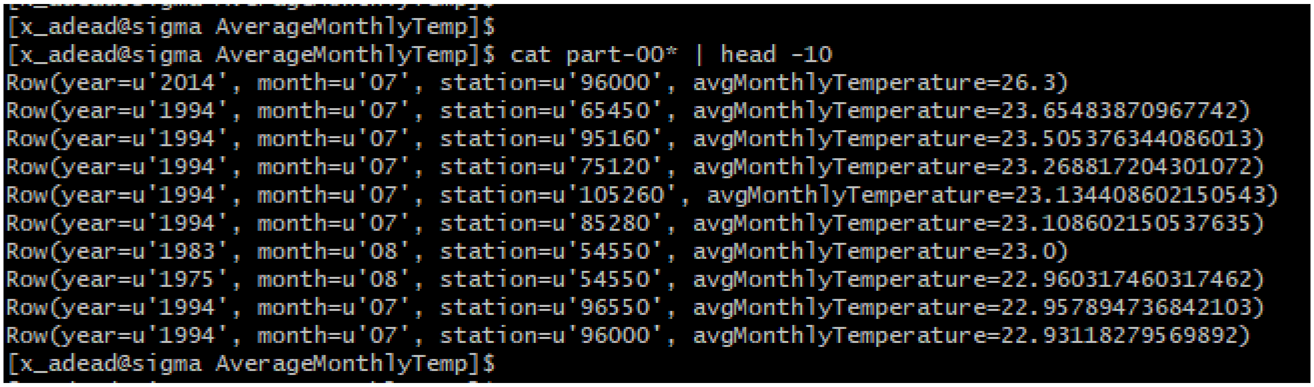
```
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
sc =SparkContext(appName = "exercise 1")
sqlContext=SQLContext(sc)

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
parts = temperature_file.map(lambda l:l.split(";"))

tempReadings = parts.map(lambda p: Row(station=p[0],date=p[1],year=p[1]
                                       .split("-")[0],month=p[1].split("-")[1],
                                       time=p[2],  Temp=float(p[3]),quality=p[4]))

TableReadings = sqlContext.createDataFrame(tempReadings)
TableReadings.registerTempTable("tempReadings")

TempAbove10Distinct = TableReadings.select(["year","month","station"])
.filter((TableReadings['year'] <=2014) & (TableReadings["year"]>=1950)
& (TableReadings["Temp"] > 10)).groupby(["year","month"])
.agg(F.countDistinct("station").alias("station_count"))
.orderBy("station_count",ascending=False)

TempAbove10Distinct.rdd.saveAsTextFile("BDA/output/TempAbove10Distinct")
```



```
[x_adead@sigma temp_G10]$
[x_adead@sigma temp_G10]$ cat part-00* | head -10
Row(year=u'1972', month=u'10', station_count=378)
Row(year=u'1973', month=u'05', station_count=377)
Row(year=u'1973', month=u'06', station_count=377)
Row(year=u'1972', month=u'08', station_count=376)
Row(year=u'1973', month=u'09', station_count=376)
Row(year=u'1972', month=u'09', station_count=375)
Row(year=u'1972', month=u'06', station_count=375)
Row(year=u'1972', month=u'05', station_count=375)
Row(year=u'1971', month=u'08', station_count=375)
Row(year=u'1971', month=u'09', station_count=374)
[x_adead@sigma temp_G10]$
```

Question 3

```
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)

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
parts = temperature_file.map(lambda l:l.split(";"))

tempReadings = parts.map(lambda p: Row(station=p[0],date=p[1],year=p[1].split("-")[0],
                                       month=p[1].split("-")[1],day=p[1].split("-")[2],time=p[2],
                                       Temp=float(p[3]),quality=p[4]))
TableReadings = sqlContext.createDataFrame(tempReadings)
TableReadings.registerTempTable("tempReadings")

AverageTempDaily = TableReadings.select(["year","month","day","station","Temp"])
.filter((TableReadings["year"]<=2014) & (TableReadings["year"]>=1960))
.groupby(["year","month","day","station"])
.agg((F.avg('Temp')).alias("daily_avg_temp"))
.orderBy(['year','month','station',"daily_avg_temp"],ascending=False)

AverageTempMonthly= AverageTempDaily.select(["year","month","station","daily_avg_temp"])
.groupBy(["year","month","station"]).agg((F.avg('daily_avg_temp')).alias("avgMonthlyTemperature"))
.orderBy("avgMonthlyTemperature",ascending=False)

AverageTempMonthly_rdd = AverageTempMonthly.rdd

AverageTempMonthly.rdd.saveAsTextFile("BDA/output/AverageMonthlyTemp")
```

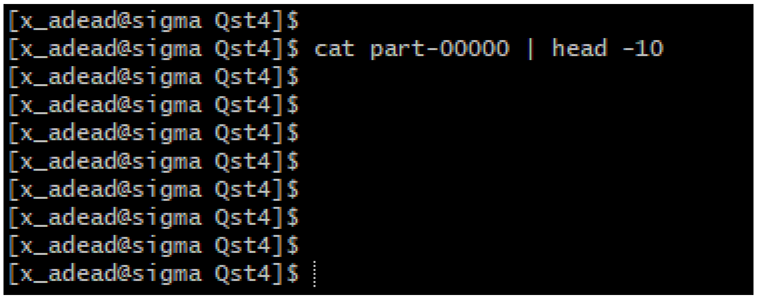


```
[x_adead@sigma AverageMonthlyTemp]$
[x_adead@sigma AverageMonthlyTemp]$ cat part-00* | head -10
Row(year=u'2014', month=u'07', station=u'96000', avgMonthlyTemperature=26.3)
Row(year=u'1994', month=u'07', station=u'65450', avgMonthlyTemperature=23.65483870967742)
Row(year=u'1994', month=u'07', station=u'95160', avgMonthlyTemperature=23.505376344086013)
Row(year=u'1994', month=u'07', station=u'75120', avgMonthlyTemperature=23.268817204301072)
Row(year=u'1994', month=u'07', station=u'105260', avgMonthlyTemperature=23.134408602150543)
Row(year=u'1994', month=u'07', station=u'85280', avgMonthlyTemperature=23.108602150537635)
Row(year=u'1983', month=u'08', station=u'54550', avgMonthlyTemperature=23.0)
Row(year=u'1975', month=u'08', station=u'54550', avgMonthlyTemperature=22.960317460317462)
Row(year=u'1994', month=u'07', station=u'96550', avgMonthlyTemperature=22.957894736842103)
Row(year=u'1994', month=u'07', station=u'96000', avgMonthlyTemperature=22.931182795698920)
[x_adead@sigma AverageMonthlyTemp]$
```

Question 4

```
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)

temp_data = sc.textFile("BDA/input/temperature-readings.csv")
precip_data = sc.textFile("BDA/input/precipitation-readings.csv")

lines_temp = temp_data.map(lambda lines: lines.split(";"))
lines_precip = precip_data.map(lambda lines: lines.split(";"))

tempReadings = lines_temp.map(lambda p: Row(station=p[0],date=p[1],year=p[1].split("-")[0],
month=p[1].split("-")[1],time=p[2],  Temp=float(p[3]),quality=p[4]))

precipReadings = lines_precip.map(lambda p: Row(station=p[0], date=p[1], year=p[1].split("-")[0],
month=p[1].split("-")[1], day=p[1].split("-")[2], time=p[2], precip=float(p[3]), quality=p[4]))

TableTempReadings = sqlContext.createDataFrame(tempReadings)
TableTempReadings.registerTempTable("tempReadings")

TablePrecipReadings = sqlContext.createDataFrame(precipReadings)
TablePrecipReadings.registerTempTable("precipReadings")

maximum_temperature = TableTempReadings.select(["station","Temp"])
.groupby("station").agg(F.max('Temp').alias("MaxTemp")).orderBy('MaxTemp',ascending=False)
maximum_temperature = maximum_temperature.select(["station","MaxTemp"])
.filter((maximum_temperature["MaxTemp"] >= 25) & (maximum_temperature["MaxTemp"] <= 30))

maximum_precipitation_daily = TablePrecipReadings.select(["station","year","month","day","precip"])
.groupby(["station","year","month","day"])
.agg(F.sum("precip").alias("daily_precip_sum"))
.orderBy("daily_precip_sum",ascending=False)maximum_precipitation =maximum_precipitation_daily
.select(["station","daily_precip_sum"])
.groupby("station").agg(F.max("daily_precip_sum").alias("maxDailyPrecipitation"))
.orderBy("station",ascending=False)
```

```
maximum_precipitationFiltered = maximum_precipitation.
filter((maximum_precipitation["maxDailyPrecipitation"] > 100)
       & (maximum_precipitation["maxDailyPrecipitation"] < 200))

Result = maximum_temperature.join(maximum_precipitationFiltered,"station","inner")
.orderBy("station",ascending=False)

Result.rdd.saveAsTextFile("BDA/output/Qst4")
```

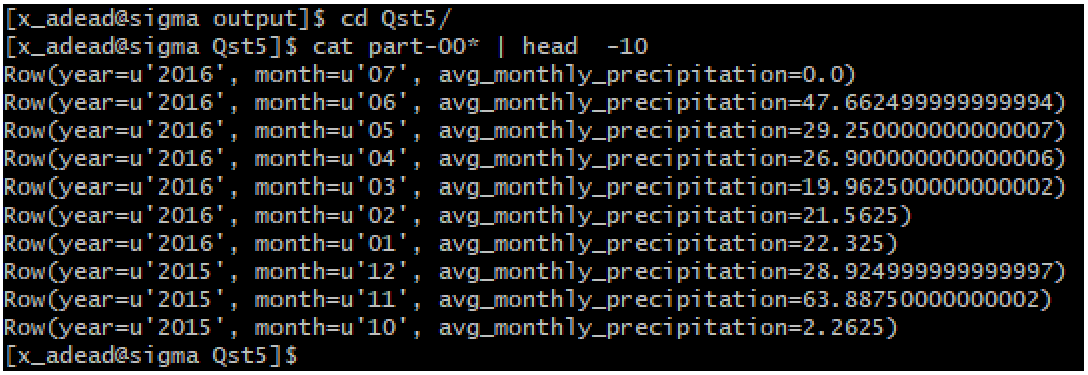


```
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$ cat part-00000 | head -10
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
[x_adead@sigma Qst4]$
```

Question 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)

station_data = sc.textFile("BDA/input/stations-Ostergotland.csv")
precip_data = sc.textFile("BDA/input/precipitation-readings.csv")

stations = station_data.map(lambda line: line.split(";"))

precip = precip_data.map(lambda line: line.split(";"))

precipReadings = precip.map(lambda p: Row(station=p[0], date=p[1], year=p[1].split("-")[0],
month=p[1].split("-")[1], day=p[1].split("-")[2], time=p[2], precip=float(p[3]), quality=p[4]))
TablePrecipReadings = sqlContext.createDataFrame(precipReadings)
TablePrecipReadings.registerTempTable("precipReadings")

stationReadings = stations.map(lambda x: Row(station=x[0], name=x[1]))
TablestationReadings = sqlContext.createDataFrame(stationReadings)
TablestationReadings.registerTempTable("stationReadings")

TablePrecipReadings = TablePrecipReadings.filter((TablePrecipReadings["year"] >= 1992)&
 (TablePrecipReadings["year"] <= 2016))
TablePrecipReadings = TablePrecipReadings.join(TablestationReadings,"station","inner")
TablePrecipReadings = TablePrecipReadings.groupby(["station","year","month"])
.agg(F.sum("precip")).groupby(["year",
"month"]).agg(F.avg("sum(precip)").alias("avg_monthly_precipitation"))
```

```
.orderBy(["year","month"],ascending= False)
TablePrecipReadings.rdd.saveAsTextFile("BDA/output/Qst5")
```



```
[x_adead@sigma output]$ cd Qst5/
[x_adead@sigma Qst5]$ cat part-00* | head  -10
Row(year=u'2016', month=u'07', avg_monthly_precipitation=0.0)
Row(year=u'2016', month=u'06', avg_monthly_precipitation=47.662499999999994)
Row(year=u'2016', month=u'05', avg_monthly_precipitation=29.250000000000007)
Row(year=u'2016', month=u'04', avg_monthly_precipitation=26.900000000000006)
Row(year=u'2016', month=u'03', avg_monthly_precipitation=19.962500000000002)
Row(year=u'2016', month=u'02', avg_monthly_precipitation=21.5625)
Row(year=u'2016', month=u'01', avg_monthly_precipitation=22.325)
Row(year=u'2015', month=u'12', avg_monthly_precipitation=28.924999999999997)
Row(year=u'2015', month=u'11', avg_monthly_precipitation=63.88750000000002)
Row(year=u'2015', month=u'10', avg_monthly_precipitation=2.2625)
[x_adead@sigma Qst5]$
```