Homework #5 (Hadoop MapReduce & Spark)

Due: 11:59pm, Friday, April 21, 2023

Points: 100

1. [40 points] Write a Hadoop MapReduce program named SQL2MR.java to find answers to the following SQL query on the **aqi.csv** data set (for Air Quality). The file has 10,129 rows, each row has four values: date, country, status, value.

   ```
   select status, avg(value)
   from aqi
   where  date like '2022-08%'
   group by status
   having count(value) >= 100;
   ```

   You are provided with a template for SQL2MR.java where you can provide the missing code to produce a complete program. The template also has some hints that may help you.

   Before you compile and run the program, please complete the following:

   - You should **remove the header of aqi.csv file and save it under a directory called aqi-input on EC2.**
   - You should **comment out OR remove <property>xxx</property> part in core-site.xml**. Here is the path for the core-site.xml file <your hadoop file name>/etc/hadoop/core-site.xml.
   - **Remember to stop your mysql server before running hadoop(sudo service mysql stop).**

   You are reminded of the following steps to compile and run the program.

   - hadoop com.sun.tools.javac.Main SQL2MR.java
   - jar cf sql2mr.jar SQL2MR*.class
   - hadoop jar sql2mr.jar SQL2MR aqi-input aqi-output

   Submission: SQL2MR.java, sql2mr.jar, and part-r-00000 file under aqi-output.

2. [30 points] Using the JSON files in country-db.zip and the aqi.csv file, answer the following questions using Spark DataFrame API. You can use "import pyspark.sql.functions as fc". Note: you should not use Spark SQL in this question.
   Remember to stop your mysql server before running spark (sudo service mysql stop).

   Submission: Copy your Spark DataFrame scripts and outputs into one file and generate **PDF** to submit.

   a. [8 points] Find countries that are in both country.json and aqi.csv.
      i. Using join

ii. Using set operation

b. [8 points] Find (names of) countries that are in aqi.csv but not in country.json. Output the same countries only once.
   i. Using join
   ii. Using set operation

c. [6 points] Find countries that are in country.json but not in aqi.csv.
   i. Using join
   ii. Using set operation

d. [8 points] Find answer to the SQL query in Task 1, copied below:

```
select status, avg(value)
from aqi
where  date like '2022-08%'
group by status
having count(value) >= 100;
```

Note that if the "date" column is a timestamp, you may proceed as follows to extract year and month of the dates:

```
fc.year('date') # this will get year

fc.month('date') # of this will get month
```

3. [30 points] Using the JSON files in country-db.zip and the aqi.csv file, answer the same questions as in Task 2 but using Spark RDD API. Note that you should first convert the dataframe for the entire data set (e.g., aqi for aqi.csv) to rdd (e.g., using aqi.rdd) and work on the RDDs to solve the questions.

   Hint: if date is a datetime, e.g., datetime.datetime(2022, 8, 1, 0, 0), you can use date.year and date.month to get year and month respectively.

   [Row(date=datetime.datetime(2022, 8, 1, 0, 0), country='Albania', …]

   <u>Submission</u>: Copy your Spark **RDD** scripts and outputs into one file and generate **PDF** to submit.

   a. [8 points] Find countries that are in both country.json and aqi.csv.
      i. Using join
      ii. Using set operation

   b. [8 points] Find (names of) countries that are in aqi.csv but not in country.json.
      i. Using join
      ii. Using set operation

   c. [6 points] Find countries that are in country.json but not in aqi.csv.
      i. Using join

ii. Using set operation

d. [8 points] Find answer to the SQL query in Task 1, copied below:

```
select status, avg(value)
from aqi
where  date like '2022-08%'
group by status
having count(value) >= 100;
```

Note: you are required to use aggregateByKey in question d.

Submission:

1. **Please ZIP your files as a whole .zip file(don't use any other compress type such as '.rar') and submit. Otherwise you may not be able to submit because of D2L security reasons.**
2. For Q1: SQL2MR.java, sql2mr.jar, and part-r-00000 file under aqi-output
3. For Q2: Copy your Spark DataFrame scripts and outputs into one file and generate PDF to submit.
4. For Q3: Copy your Spark RDD scripts and outputs into one file and generate PDF to submit.