Assignment5_Haque

CSC 4760/6760 Big Data Programming
Assignment 5
Due Date: 11:59 pm, April 7, 2021

1. (100 points) (Counting Tweets)

Input Datasets:

```
Tweets (tweets.json):
+-------------+----------+
|      Atlanta|   Georgia|
|       Athens|   Georgia|
|        Miami|   Florida|
|      Orlando|   Florida|
|   Birmingham|   Alabama|
|       Auburn|   Alabama|
|  Log Angeles|California|
|San Francisco|California|
|    San Diego|California|
+-------------+----------+
```

```
City and State lookup table (cityStateMap.json):
+-------------+-------------------+-------+
|      Atlanta|  It is a sunny day!|    Bob|
|       Athens|We have a footbal...|  Susan|
|      Atlanta|      Today is cold.|  David|
|       Auburn|I love Auburn Uni...|   Lisa|
|   Birmingham|I will go to Atla...|    Ben|
|San Francisco|We watch a movie ...|   Paul|
|    San Diego|It is hot today. ...|  Smith|
|  Log Angeles|Oscar ceremony is...|  Ethan|
|  Log Angeles|I love Oscar cere...|   Emma|
|      Orlando|I will go to the ...|Rolando|
|        Miami|          Sunny Day!|    Mia|
+-------------+-------------------+-------+
```

Problem and Output Data:

- Print only tweets from Atlanta.

```
1st solution
+-------+------------------+
|    geo|             tweet|
+-------+------------------+
|Atlanta|It is a sunny day!|
|Atlanta|    Today is cold.|
+-------------+-----------+
```

```
2nd solution
+------------------+
|             tweet|
+------------------+
|It is a sunny day!|
|    Today is cold.|
+------------------+
```

- Print only tweets that contain the word “today”.

```
1st solution
+-------------+--------------------+
|          geo|               tweet|
+-------------+--------------------+
|       Athens|We have a footbal...|
|   Birmingham|I will go to Atla...|
|San Francisco|We watch a movie ...|
|    San Diego|It is hot today. ...|
+-------------+--------------------+
```

```
2nd solution
+--------------------+
|               tweet|
+--------------------+
|We have a footbal...|
|I will go to Atla...|
|We watch a movie ...|
|It is hot today. ...|
+--------------------+
```

- Print only tweets from California.

```
1st solution
+----------+--------------------+
|     state|               tweet|
+----------+--------------------+
|California|We watch a movie ...|
|California|It is hot today. ...|
|California|Oscar ceremony is...|
|California|I love Oscar cere...|
+----------+--------------------+
```

```
2nd solution
+--------------------+
|               tweet|
+--------------------+
|We watch a movie ...|
|It is hot today. ...|
|Oscar ceremony is...|
|I love Oscar cere...|
+--------------------+
```

- We want to count the number of tweets published in each state. The following table shows the desired results.

```
Sample output
+----------+-----+
|     state|count|
+----------+-----+
|   Georgia|    3|
```

```
Code output
+----------+-----+
|     state|count|
+----------+-----+
|   Georgia|    3|
|   Alabama|    2|
|   Florida|    2|
|California|    4|
+----------+-----+
```

```
|   Alabama|    2|
|   Florida|    2|
|California|    4|
+----------+-----+
```

1. Implementation:
   a. Design and implement a PySpark program to solve the problems. We did not provide any template python file this time. You may want to create one python file from scratch.
      i. You are required to use Spark Dataframe to implement this function.
      ii. Report:
      Please write a report illustrating your experiments. You need to explain your basic idea about how Spark Dataframe is used for each problem. You may add comments to the source code such that the source code can be read and understood by the graders.
         1. In the report, you should include the answers to the following questions.
2. Explanation of the source code.

   The source contains code to print out the information in the datfilterased on the query type. Each line is a query designed to get the correct information using filter, join. and select statements.
3. Experimental Results.

   The result outputted is the correct based on the query type and designed data on the specific query.

   - The first question asked to query data from Atlanta, so a filter was used to filter the column name: “geo” by Atlanta.

     ```
     tweets_from_atlanta = DF2.filter("geo == 'Atlanta'").select("geo", "tweet").show()
     ```

   - The second question asked to query the tweets that had the word “today” in it. So the like operator was used to search today inside of a filter.

     ```
     tweets_with_word_today = DF2.filter(col("tweet").like("%today%")).Californiao", "tweet").show()
     ```

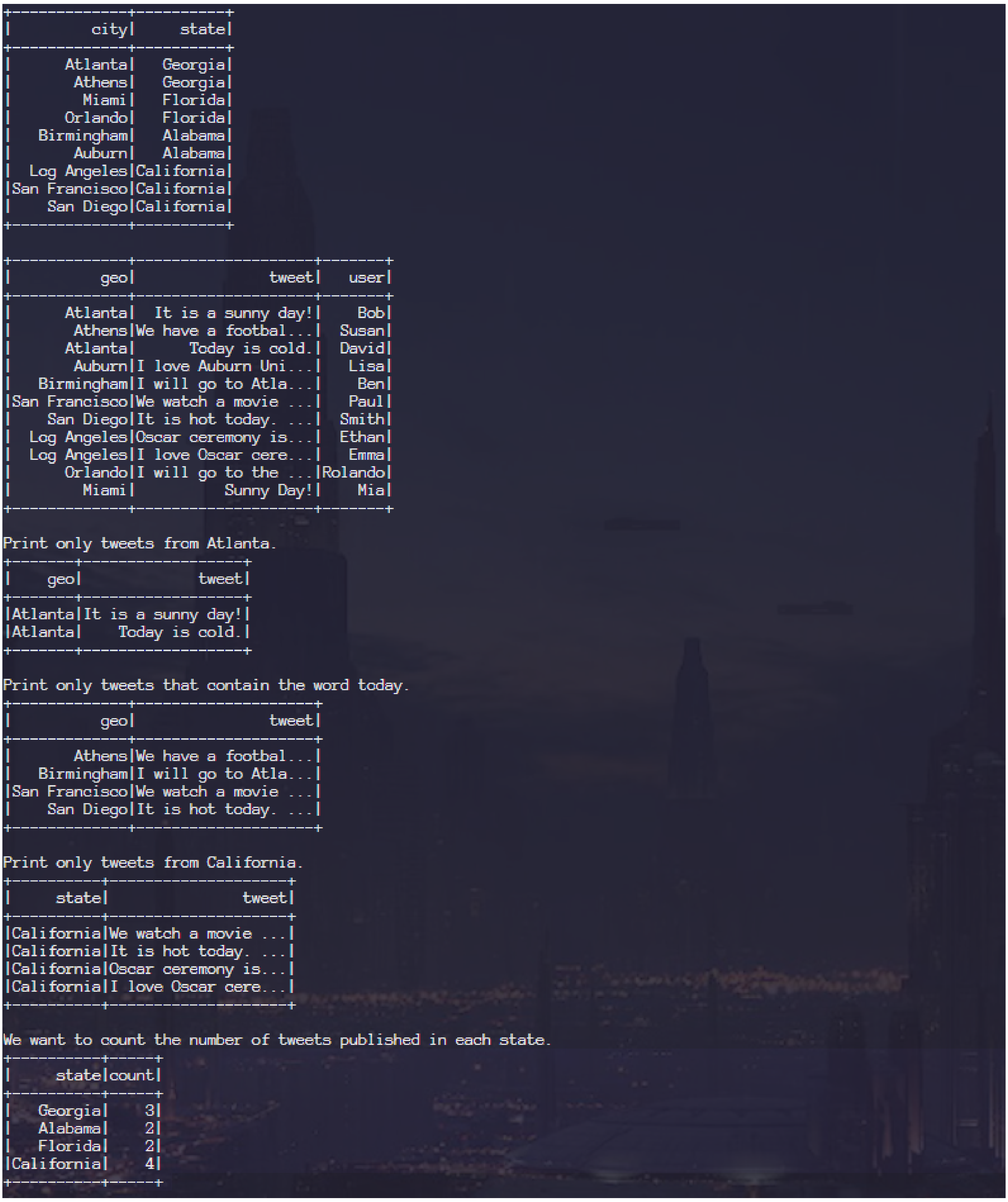
   - The third question asked to query tweets from California. The state information about each tweet was not provided in current table. So a join was used based on the city(”geo” == city). Once that was done we were free to use the filer to filter by state where the state name was California.

     ```
     tweet_geo = DF2.join(DF1, DF1.city == DF2.geo)
     tweets_from_california = tweet_geo.filter(DF1.state == "California").select("state", "tweet").show()
     ```

   - The final question askes to count the number of tweets by each state. We already have a joined table containing the states. So we can just use that to grouBy by state and use the count aggregate function

     ```
     tweet_count = tweet_geo.groupBy("state").count().show()
     ```

   2.1) Screenshots of the output for each problem. Since we plan to use Dataframe in Spark, it is easy to type in “DF.show()” to visualize the table in the terminal. Please do so and take a screenshot of the output in the terminal.



```
+-------------+----------+
|         city|     state|
+-------------+----------+
|      Atlanta|   Georgia|
|       Athens|   Georgia|
|        Miami|   Florida|
|      Orlando|   Florida|
|   Birmingham|   Alabama|
|       Auburn|   Alabama|
|  Log Angeles|California|
|San Francisco|California|
|    San Diego|California|
+-------------+----------+

+-------------+--------------------+-------+
|          geo|               tweet|   user|
+-------------+--------------------+-------+
|      Atlanta|  It is a sunny day!|    Bob|
|       Athens|We have a footbal...|  Susan|
|      Atlanta|      Today is cold.|  David|
|       Auburn|I love Auburn Uni...|   Lisa|
|   Birmingham|I will go to Atla...|    Ben|
|San Francisco|We watch a movie ...|   Paul|
|    San Diego|It is hot today. ...|  Smith|
|  Log Angeles|Oscar ceremony is...|  Ethan|
|  Log Angeles|I love Oscar cere...|   Emma|
|      Orlando|I will go to the ...|Rolando|
|        Miami|          Sunny Day!|    Mia|
+-------------+--------------------+-------+

Print only tweets from Atlanta.
+-------+------------------+
|    geo|             tweet|
+-------+------------------+
|Atlanta|It is a sunny day!|
|Atlanta|    Today is cold.|
+-------+------------------+

Print only tweets that contain the word today.
+-------------+--------------------+
|          geo|               tweet|
+-------------+--------------------+
|       Athens|We have a footbal...|
|   Birmingham|I will go to Atla...|
|San Francisco|We watch a movie ...|
|    San Diego|It is hot today. ...|
+-------------+--------------------+

Print only tweets from California.
+----------+--------------------+
|     state|               tweet|
+----------+--------------------+
|California|We watch a movie ...|
|California|It is hot today. ...|
|California|Oscar ceremony is...|
|California|I love Oscar cere...|
+----------+--------------------+

We want to count the number of tweets published in each state.
+----------+-----+
|     state|count|
+----------+-----+
|   Georgia|    3|
|   Alabama|    2|
|   Florida|    2|
|California|    4|
+----------+-----+
```

2.2) Explain your results. Does your implementation give the right answer?

yes, this out does give use the correct answer based on the query. Further explanation is provided above

Code for solution 1

```
import pyspark
from pyspark.context import SparkContext
from pyspark import SparkConf
from pyspark.sql import SparkSession, SQLContext, Row
from pyspark.sql.functions import *
import json
import os

conf = SparkConf()
sc = SparkContext(conf=conf)
sc.setLogLevel("ERROR")

spark = SparkSession \
    .builder \
    .appName("Phone Book - Country Look up") \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()



DF1 = spark.read.json("/home/nur_haque/HW5/cityStateMap.json")
DF2 = spark.read.json("/home/nur_haque/HW5/tweets.json")

DF1.show()
DF2.show()
# Print only tweets from Atlanta.
print("Print only tweets from Atlanta.")
tweets_from_atlanta = DF2.filter("geo == 'Atlanta'").select("geo", "tweet").show()
# Print only tweets that contain the word “today”.
print("Print only tweets that contain the word today.")
tweets_with_word_today = DF2.filter(col("tweet").like("%today%")).select("geo", "tweet").show()
# Print only tweets from California.
print("Print only tweets from California.")
tweet_geo = DF2.join(DF1, DF1.city == DF2.geo)
tweets_from_california = tweet_geo.filter(DF1.state == "California").select("state", "tweet").show()
# We want to count the number of tweets published in each state.
print("We want to count the number of tweets published in each state.")
tweet_count = tweet_geo.groupBy("state").count().show()
```

Code for solution 2

```
import pyspark
from pyspark.context import SparkContext
from pyspark import SparkConf
from pyspark.sql import SparkSession, SQLContext, Row
from pyspark.sql.functions import *
import json
import os

conf = SparkConf()
sc = SparkContext(conf=conf)
sc.setLogLevel("ERROR")

spark = SparkSession \
    .builder \
    .appName("Phone Book - Country Look up") \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()



DF1 = spark.read.json("/home/nur_haque/HW5/cityStateMap.json")
DF2 = spark.read.json("/home/nur_haque/HW5/tweets.json")

DF1.show()
DF2.show()
# Print only tweets from Atlanta.
print("Print only tweets from Atlanta.")
tweets_from_atlanta = DF2.filter("geo == 'Atlanta'").select("tweet").show()
# Print only tweets that contain the word “today”.
print("Print only tweets that contain the word today.")
tweets_with_word_today = DF2.filter(col("tweet").like("%today%")).select("tweet").show()
# Print only tweets from California.
print("Print only tweets from California.")
tweet_geo = DF2.join(DF1, DF1.city == DF2.geo)
tweets_from_california = tweet_geo.filter(DF1.state == "California").select("tweet").show()
# We want to count the number of tweets published in each state.
print("We want to count the number of tweets published in each state.")
tweet_count = tweet_geo.groupBy("state").count().show()
```