CSC 455: Database Processing for Large-Scale Analytics

Assignment 5

Due Tuesday, March 6th

Supplemental reading: Python for Data Analytics, Chapter 4, 5 and 7

1. Use the tweet table we created in the class and write the follow SQL queries”

   a. Count the number of iPhone users (based on “source” attribute)

   ```
   SELECT Count(source) AS the number of iPhone users
   FROM tweets
   WHERE source = ‘<a href=\”http://twitter.com/download/iphone\"
   rel=\"nofollow\">Twitter for iPhone</a>’;
   ```

   b. Create a view that contains only “id_str”, “text” and “source” from each tweet that has a “retweet_count” of at least 5

   ```
   CREATE VIEW idstr_text_source_view AS
   SELECT id_str, text, source
   FROM tweets
   WHERE retweet_count >= 5;
   ```

2. In this part of the assignment we are going to work with a larger collection of tweets (10,000) that are available here:
   http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Assignment5.txt

   Please see attached Python file

   The tweets are all on separate lines, but some of the tweets are intentionally damaged and will not parse properly. You will need to store these tweets in a separate “error” file. At the bottom of the page you can find python code that will let you skip over badly formed tweets.

   c. Create a new SQL table for the user dictionary. It should contain the following attributes “id”, “name”, “screen_name”, “description” and “friends_count”. Modify your SQL table from the class we did in class to include “user_id” which will be a foreign key referencing the user table.

   d. Write python code that is going to read and load the Assignment5.txt file from the web and populate both of your tables (Tweet table from class example and User table from this assignment).
   For tweets that could not parse, simply store them in Assignment5_errors.txt file

   You can gracefully catch JSON errors using the following code:

```
for tweet in allTweets:
  try:
    tdict = json.loads(tweet.decode('utf8'))
  except ValueError:
    # Handle the problematic tweet, which in your case would require writing it to another file
    # This code imply prints the first 50 characters
    print (tweet[:50])
```

As discussed in class, you can access the contents of the user dictionary after it was parsed by json like this:

```
dict['user']   # user dictionary
dict['user']['id']  # user's ID
```

3.

a. Write and execute a SQL query to do the following: Find the user (“id” and “name”) with the highest “friend_count” in the database

```
SELECT id,name FROM User
WHERE friends_count= (SELECT max(friends_count) FROM USER);
```

b. Write and execute SQL query to do the following: Find the tweets without associated user id entry.

```
SELECT * FROM Tweet2 T
WHERE NOT EXISTS(SELECT * FROM User U where U.id = T.User_id);
```

Be sure that your name and “Assignment 5” appear at the top of your submitted file.