Olusola Ogun

Assignment #5

Section 1

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) FROM TWEETS WHERE SOURCE = '%TWITTER%FOR%IPHONE%'
   ```

   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 RETWEETCOUNT AS SELECT ID_STR, TEXT, SOURCE FROM TWEETS WHERE RETWEET_COUNT >=5
   ```

Section 2

1. 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 (http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Assignment5.txt)

   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

In [ ]:

```
import sqlite3

conn=sqlite3.connect('CSC_455_Assignment5.db')
c=conn.cursor()
user='''CREATE TABLE USER
    (
    ID NUMBER(20),
    NAME VARCHAR(100)
    SCREEN_NAME VARCHAR(100)
    DESCRIPTION VARCHAR(1000)
    FRIENDS_COUNT NUMBER(20)

    CONSTRAINT PK_USER

    PRIMARY KEY(ID)
    ):'''
c.execute('DROP TABLE IF EXISTS USER');
c.execute(user)

tweet='''CREATE TABLE TWEET
(
CREATED_AT DATE,
ID_STR CHAR(30)
TEXT VARCHAR(1000),
SOURCE VARCHAR(500),
IN_REPLY_TO_USER_ID NUMBER(20),
IN_REPLY_TO_SCREEN_NAME VARCHAR(50),
IN_REPLY_TO_STATUS_ID NUMBER(30),
RETWEET_COUNT NUMBER(100),
CONTRIBUTORS VARCHAR(200),
USER_ID NUMBER(50)

CONSTRAINT PK_TWEET PRIMARY KEY(ID_STR),
CONSTRAINT FK_TWEET FORIEGN KEY(USER_ID) REFERENCES USER(ID)
);'''

c.execute('DROP TABLE IF EXISTS TWEET');
c.execute(TWEET)
conn.commit()
```

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

In [ ]:

```
import json
import urllib

allTweets = urllib.request("http://rasinsrv07.cstcis.cti.depaul.edu/CSC455/Assignment5.txt").read()
allTweets = allTweets.split("\n")

for tweet in allTweets:
    try:
        tdict = json.loads(tweet.decode('utf8'))
    except ValueError:
        newtweet=tweet.encode('utf8')
        outfile=open("Assignment5_errors.txt","w")
        outfile.write(newtweet)
        outfile.close()
```

Section 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

1. SELECT id,name FROM tweetuser WHERE friends_count (SELECT max(friends_count) as friends_count FROM tweetuser)

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

1. SELECT * FROM tweet where user_id=null