**Malini Mittal**

**W205**

**Assignment 3**

**Task 2 - Relational Database (sqlite):**

**Data Cleaning:**

Each tweet that gets read in will be cleaned to store only the required fields:

for tweet in tweets:

d = {}

d['created\_at'] = parser.parse(tweet['created\_at']).isoformat()

d['userhashtags'] = [h[u'text'] for h in tweet['entities']['hashtags']]

d['userscreenname'] = tweet['user']['screen\_name'].encode('utf-8')

d['username'] = tweet['user']['name'].encode('utf-8')

d[‘tweet\_id’] = tweet[‘id’]

# … any other operation … #

**Data Organization:**

The following four tables will be created, with schemas as shown:

1. Hashtags:

CREATE TABLE Hashtags(

id INT PRIMARY KEY,

name TEXT NOT NULL

);

1. Tweets:

CREATE TABLE Tweets(

id INT PRIMARY KEY,

date TEXT,

user\_id INT,

FOREIGN KEY(user\_id) REFERENCES Users(id)

);

1. Tweets\_Hashtags:

CREATE TABLE Tweets\_Hashtags(

tweet\_id INT,

hashtag\_id INT,

FOREIGN KEY(tweet\_id) REFERENCES Tweets(id),

FOREIGN KEY(hashtag\_id) REFERENCES Hashtags(id)

);

1. Users:

CREATE TABLE Users(

id INT PRIMARY KEY,

name TEXT,

screen\_name TEXT NOT NULL

);

**Storage:**

Iterating through the tweets after loading the json files, the relevant data is added to the appropriate tables.

**Retrieval:**

1. Who tweeted the most during the conference?

def printUserWithMostTweets(cursor, num):

query = 'Select user\_id, Count(\*) from Tweets Group by user\_id ORDER by Count(\*) DESC'

cursor.execute(query)

cursor2 = conn.cursor()

for i in range(0, num):

row = cursor.fetchone()

query = ''.join(['Select \* from Users where id=', str(row[0])])

cursor2.execute(query)

print ("total = {0}, screen\_name = {1}").format(row[1], cursor2.fetchone()[2])

1. What were the top ten hashtags used?

def printMostUsedHashtags(cursor, num):

query = 'Select hashtag\_id, Count(\*) from Tweets\_Hashtags Group by hashtag\_id

ORDER by Count(\*) DESC'

cursor.execute(query)

cursor2 = conn.cursor()

for i in range(0, num):

row = cursor.fetchone()

query = ''.join(['Select \* from Hashtags where id=', str(row[0])])

cursor2.execute(query)

print ("total = {0}, hashtag = {1}").format(row[1], cursor2.fetchone()[1])

1. How many tweets were produced each hour?

def getNumTweets(cursor, starttime, endtime):

query = ''.join(['Select Count(\*) from Tweets where date >= "', starttime, '" and date

< "', endtime, '"'])

cursor.execute(query)

data = cursor.fetchone()

if data == None:

return 0

return data[0]