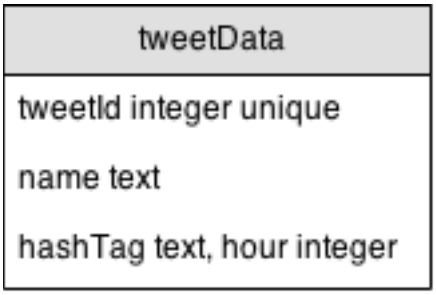
W205: Organizing Tweets

Chula Watugala

1.



2.

Key/Value:

i. Tweet data for each day is stored in separate folders as files
ii. The JSON files can be stored in S3 as is
iii.
   a. Pull in the raw data from S3 using a server connection setup to access the database
   b. Parse only the key/value pairs required (name, hashtag, created_at) in Python
   c. Store relevant data in dictionary format and resolve answers

NoSQL:

i. Tweet data for each day is inserted into MongoDB as a single collection
ii. Since the files are already in JSON format and Mongo supports JSON, no manipulation is required. The file can be loaded into Mongo DB using the mongoimport command
iii.
   a. Pull in the raw data from MongoDB using a collection load command
   b. Parse the required fields (name, hashtag, created_at) in Javascript
   c. Resolve answers using loops and dictionary variables

Relational:

i. Relevant fields are identified and parsed, stored in an SQLite table with a simple schema
ii. Each JSON file is loaded into a dictionary variable using the simplejson module built-in functions, then relevant fields are parsed and stored into a previously created SQLite DB table and committed to storage.

iii.

a. Query the table for all tweets between the hours required (with -1h adjustment for timezone)
b. Loop through the relevant fields and store name-to-tweetCount, hashtag-to-tweetCount and hour-to-tweetCount values in dictionary form.
c. Check whether a certain key for each of the dictionaries above are already present, if so, add to the count, if not, create a new key and set count to 1.
d. Identify the answers to the specific questions using the generated dictionaries.

3.

Implementation with Python-SQLite:

```
import simplejson
import sqlite3

conn = sqlite3.connect('tweet.db')
c = conn.cursor()

#c.execute('drop table tweetInfo')
c.execute('''create table tweetInfo (tweetId integer unique, date text, name text,
hashTag text, hour integer)''')

dates = ('2015-02-14', '2015-02-15')
tweetId = 1

for date in dates:
	with open("prague-"+date+".json") as f:
		data = simplejson.load(f)
	for i in range(0, len(data)):
		tweetData = data[i]
		name = tweetData["user"]["screen_name"]
		hashTag = tweetData["entities"]["hashtags"][0]["text"].lower()
		splitFullTime = tweetData["created_at"].split(" ")
		splitTime = splitFullTime[3].split(":")
		hour = int(splitTime[0])
		c.execute('insert or ignore into tweetInfo (tweetId, date, name, hashTag,
hour) values (?, ?, ?, ?, ?)', (tweetId, date, name, hashTag, hour))
		tweetId += 1
conn.commit()

name2TweetCount = {}
hashTagCount = {}
hour2TweetCount = {}
startHour = 8
endHour = 15

for date in dates:
	print "\n" + date
	c.execute('select name, hashTag, hour from tweetInfo where date=? and hour>=?
and hour<?',(date,startHour,(endHour+1)))
	for tweet in c.fetchall():
		name = tweet[0]
		hashTag = tweet[1]
		hour = tweet[2]
		if hashTag in hashTagCount:
			hashTagCount[hashTag] += 1
```

```
            else:
                  hashTagCount[hashTag] = 1
            if name in name2TweetCount:
                  name2TweetCount[name] += 1
            else:
                  name2TweetCount[name] = 1
            if hour in hour2TweetCount:
                  hour2TweetCount[hour] += 1
            else:
                  hour2TweetCount[hour] = 1
      print max(name2TweetCount, key=name2TweetCount.get)
      print sorted(hashTagCount, key=hashTagCount.get, reverse=True)[:10]
      print hour2TweetCount

conn.close()
```

4.

i. 2015-02-14: xmlprague
2015-02-15: xmlprague

ii. 2015-02-14: xmlprague, xproc, xslt, rdfa, existdb, justsaying, oxygenxml, edupub, conference, xprocathon
2015-02-15: xmlprague, xproc, thetransformationsong, rdfa, oxygenxml, fuckyeah, xslt, existdb, xml, edupub

iii. 2015-02-14 (-1h timezone adjustment):
- 8-9: 46
- 9-10: 55
- 10-11: 19
- 11-12: 42
- 12-13: 9
- 13-14: 24
- 14-15: 24
- 15-16: 29

2015-02-15 (-1h timezone adjustment):
- 8-9: 67
- 9-10: 111
- 10-11: 35
- 11-12: 108
- 12-13: 22
- 13-14: 37
- 14-15: 65
- 15-16: 73