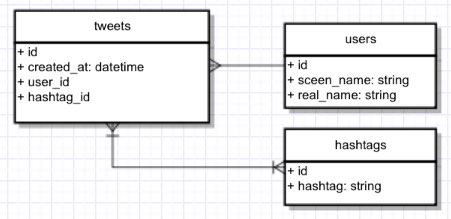
**Task #1: UML ERD**

This is the general UML ERD for how I would model the information extracted:

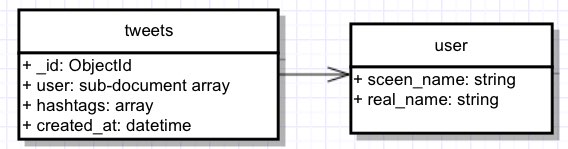


The idea is that there is three main categories of information: the tweet, the user, and the hashtag. The ‘id’, ‘user\_id’, and ‘hashtag\_id’ implementation can be omitted depending on how the database is structured.

**Task #2A: NoSQL**

(i) Implementation Model

Below is the collection structure for modeling information extracted from the raw tweets:



There will only be one collection of “tweets”. No foreign key is used as “user” displayed above is not a separate collection. “User” is actually embedded in sub-document array within the “tweets” collection. Similarly, “hashtags” is also embedded within the “tweets” collection as a standard array.

The above collection structure is based on using MongoDB with the example JSON structure below:

//tweets collection

[

{

"\_id": ObjectId(xxxxx),

"user": {

screen\_name: "evlist",

real\_name: "Eric van der Vlist"

},

"hashtags": ["xmlprague", "marklogic",...],

"created\_at": ISODate()

},

…

]

(ii) Process for Storing Information

1. Create a python script that can load the 2 JSON files in-memory to form dictionary format so that python can easily manipulate
2. For each JSON file, the python script extracts only the needed information (tweet created time, user’s screen name, user’s real name, and hashtags) and format the data in the JSON structure described above. Special attention should be paid on formatting the date
3. In the same python script using "pymongo” package to connect to localhost and to load the dictionary data into MongoDB.
4. Run “mongod” command to initiate mongo server and run the python script that contains step 1-3.

(iii) Pseudo Code / Procedures for Answering Questions

Q1: Who tweeted the most during the conference?

* Install node module “mongodb”
* Create nodejs app that connects to the database at localhost
* Run the following query inside the nodejs app:

Run “aggregate” function for mongotDB with the following conditions:

1. Match this condition:

(

(“created\_at” >= day 1’s start time of 9am CET Feb 14)

and (“created\_at” <= day 1’s end time of 9am CET Feb 14)

)

or

(

(“created\_at” >= day 2’s start time of 9am CET Feb 15)

and (“created\_at” <= day 2’s end time of 9am CET Feb 15)

)

)

2. Group by screen name of each tweet and count no. of tweet per screen name

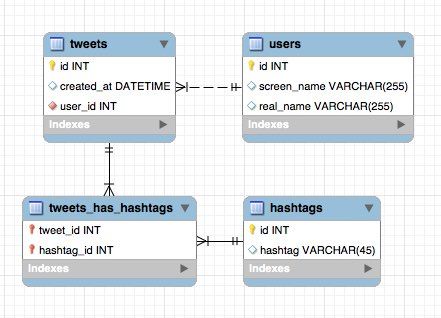
3. Sort the result by descending order

Since the output is an array of JSON with the following format

Q2: What were the top 10 hash tags used?

Q3L For a particular hour, how many tweets were produced?

**Task #2B: SQL**



s3://ConferenceTweetData/dates/<datetime>/<user\_screen\_name>/<tweet\_id>.json

s3://ConferenceTweetData/hashtags/<hashtag>/<tweet\_id>

Who tweeted the most during the conference?

XMLPrague 68 times

What were the top 10 hash tags used?

#1 Hashtag with 746 mentions: xmlprague

#2 Hashtag with 38 mentions: XMLPrague

#3 Hashtag with 27 mentions: thetransformationsong

#4 Hashtag with 19 mentions: oxygenxml

#5 Hashtag with 18 mentions: XProc

#6 Hashtag with 15 mentions: RDFa

#7 Hashtag with 14 mentions: BRILLIANT

#8 Hashtag with 14 mentions: XML

#9 Hashtag with 14 mentions: FUCKYEAH

#10 Hashtag with 12 mentions: xproc

1. xmlprague 746

2. XMLPrague 38

3. thetransformationsong 27

4. oxygenxml 19

5. XProc 18

6. RDFa 15

7. BRILLIANT 14

8. FUCKYEAH 14

9. XML 14

10. JSON 12

For a particular hour, how many tweets were produced?

**Task #3**

Instructions:

* Run the command “python load.py” to load the 2 JSON files into MongoDB
* Run “app.js” to conduct the 3 queries for the 3 questions asked. Below is the output that answer the 3 questions after running app.js:

alexmilowski tweeted the most with 59 tweets between 9am to 4pm CET on Feb 14 and Feb 15

These are the top 10 most used hashtags between 9am and 4pm CET on Feb 14 and Feb 15:

{ hashtag: 'xmlprague', count: 441 }

{ hashtag: 'thetransformationsong', count: 22 }

{ hashtag: 'XMLPrague', count: 20 }

{ hashtag: 'XProc', count: 15 }

{ hashtag: 'xproc', count: 9 }

{ hashtag: 'existdb', count: 8 }

{ hashtag: 'RDFa', count: 7 }

{ hashtag: 'FUCKYEAH', count: 6 }

{ hashtag: 'CreditWhereDue', count: 6 }

{ hashtag: 'BRILLIANT', count: 6 }

Hourly tweets in chronological order:

On Feb 14 2015 between 09:00 and 10:00 (CET), there are 46 tweets

On Feb 14 2015 between 10:00 and 11:00 (CET), there are 55 tweets

On Feb 14 2015 between 11:00 and 12:00 (CET), there are 19 tweets

On Feb 14 2015 between 12:00 and 13:00 (CET), there are 42 tweets

On Feb 14 2015 between 13:00 and 14:00 (CET), there are 9 tweets

On Feb 14 2015 between 14:00 and 15:00 (CET), there are 24 tweets

On Feb 14 2015 between 15:00 and 16:00 (CET), there are 24 tweets

On Feb 15 2015 between 09:00 and 10:00 (CET), there are 21 tweets

On Feb 15 2015 between 10:00 and 11:00 (CET), there are 56 tweets

On Feb 15 2015 between 11:00 and 12:00 (CET), there are 16 tweets

On Feb 15 2015 between 12:00 and 13:00 (CET), there are 66 tweets

On Feb 15 2015 between 13:00 and 14:00 (CET), there are 13 tweets

On Feb 15 2015 between 14:00 and 15:00 (CET), there are 13 tweets

On Feb 15 2015 between 15:00 and 16:00 (CET), there are 41 tweets