Advanced Database Systems (CS60113)

Assignment 6. NoSQL/MongoDB (Due Date: **November 14** 2020)

NOTE: This is an individual assignment.

MongoDB shell query assignments

Assume that the **collection** is "**tutorial**" in your answers (e.g., db.tutorial.find({...})). Expected output formats are shown after each question. To assist you with the queries, relevant field names are given in parentheses. However, it is not compulsory to use these.

1. Get the number of tweets posted from each country ('place.country_code') in the collection sorted by number of tweets in descending order. Do not consider null or empty 'country_codes'.

Country_code	number_of_tweets
IN	100
US	61
DE	9
…	

2. For each country, get the number of tweets written in the most frequently used language ('lang'), sorted in descending order. Do not consider null or empty 'country_codes'.

Country_code	number_of_tweets_in_lang
IN	52
US	44
DEN	6

3. Get the list of users ('user.id') who have posted more than 1 and less than 5 tweets on or after the date 01-Jan-2016 sorted by number of tweets in descending order. ('created_at_iso')

_id	num_tweets
12345	8
12346	3
12347	2

MongoDB python assignments

Assume that the database is "**twitter**" and the collection is "**tutorial**" in your program.

1. Write a python program which takes as input a file containing user id pairs in each line like

1, 2
1, 3
2, 1
4, 5

Where the pair 1, 2 denotes that twitter user with user id 1 follows twitter user with user id 2. In other words, user id 1 is a follower of user id 2. Similarly in Twitter lingo, user id 2 is a friend of user id 1. In the above example pairs, friend and follower counts of user id 1 are 2 and 1, respectively.

Add two new fields “user.new_friends_count” and “user.new_followers_count” (i.e., nested inside the field “user”) to the documents in the collection and set the values appropriately as explained above. Do not update a document if the corresponding user id does not appear in the input file.

2. Some documents in the collection are retweets. These are no different than tweets except that these have a special field named “retweeted_status” which contains the original tweet. Try these to get an idea-

```
> db.tutorial.count({'retweeted_status':{$exists:1}})
> db.tutorial.findOne({'retweeted_status':{$exists:1}})
```

Write a program to print the top 5 case insensitive hashtags (‘entities.hashtags’) for retweets in the collection.