Query: Find out the latest date of each tweet (i.e find out when it was last retweeted, if this tweet is original tweet then return its created date.

In our mongodb database, the information of the tweets is stored in a collection named **coll**. The query follows multiple steps:

- Identify all tweets which do not have any retweeted_status field and extract useful fields–

  -
    ```
    var notReTweets = db.coll.find( { retweeted_status : { $exists : false } }, {id:1,
    "retweeted_status.id":1, "retweeted_status.retweet_count":1,"favorite_count":1
    ,"created_at":1} )
    ```

- Create a new collection **collOriginal**, add these tuples to it and update the database with the original id as a copy of the id.

  -
    ```
    db.createCollection('collOriginal')
    ```

  -
    ```
    notReTweets.forEach(function(x){ db.collOriginal.insertOne(x) })
    ```

  -
    ```
    db.collOriginal.find().snapshot().forEach(
     function (e) {
      e.orig_id = e.id;
       e.ret_count = 0;
       db.collOriginal.save(e);
     })
    ```

- Identify all tweets which are retweets and add them along with the id of the original tweet (from the retweeted_status.id) into the **collOriginal** collection.

  -
    ```
    var reTweets = db.coll.find( { retweeted_status : { $exists : true } }, {id:1,
    "retweeted_status.id":1, "retweeted_status.retweet_count":1,
    "retweeted_status.favorite_count":1, "created_at":1} )
    ```

  -
    ```
    reTweets.forEach(function(x){ db.collOriginal.insertOne({'id': x.id, 'orig_id':
    x.retweeted_status.id,'ret_count':x.retweeted_status.retweet_count
    ,'favorite_count':x.retweeted_status.favorite_count,'created_at':x.created_at})
    })
    ```

- Now, we have a collection **collOriginal** which has the tweet id and original tweet id for each tweet. Here, we use the group by and reduce function in mongodb to create a new collection **temp** containing the latest retweet count, latest favorite count, latest date and id of the latest retweet corresponding to each original tweet id.

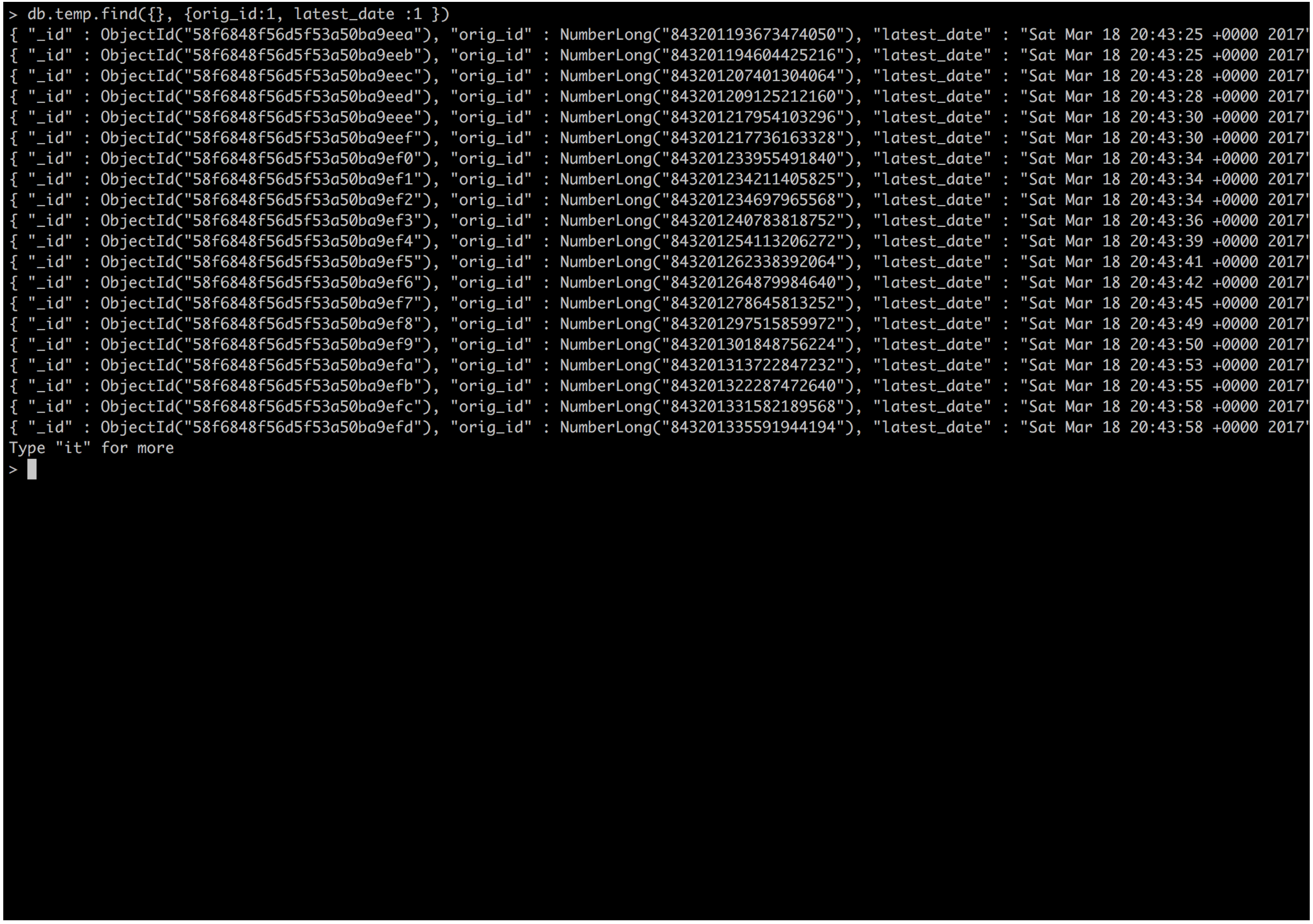
  -
    ```
    var arr = db.collOriginal.group({key: {orig_id:true},
    ```

```
                    reduce: function(obj,prev) {
                    if (prev.latest_ret_count < obj.ret_count) {
                    prev.latest_ret_count = obj.ret_count;
                    prev.maxId = obj.id
                    prev.latest_date = obj.created_at
                    prev.latest_fav_count = obj.favorite_count
                }  },initial: { latest_ret_count: -1, maxId: -1, latest_date:""}});
```

  - db.createCollection('temp')
  - db.temp.insert(arr)

- Finally, to get the latest retweet date for each tweet, we just query this new collection **temp** to get the results :
  - db.temp.find({}, {orig_id:1, latest_date :1 })

<u>The below screenshot shows a few results. Each result has an orig_id (tweet id of the original tweet) and latest_date (date of last retweet of that tweet).</u>



```
> db.temp.find({}, {orig_id:1, latest_date :1 })
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eea"), "orig_id" : NumberLong("843201193673474050"), "latest_date" : "Sat Mar 18 20:43:25 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eeb"), "orig_id" : NumberLong("843201194604425216"), "latest_date" : "Sat Mar 18 20:43:25 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eec"), "orig_id" : NumberLong("843201207401304064"), "latest_date" : "Sat Mar 18 20:43:28 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eed"), "orig_id" : NumberLong("843201209125212160"), "latest_date" : "Sat Mar 18 20:43:28 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eee"), "orig_id" : NumberLong("843201217954103296"), "latest_date" : "Sat Mar 18 20:43:30 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9eef"), "orig_id" : NumberLong("843201217736163328"), "latest_date" : "Sat Mar 18 20:43:30 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef0"), "orig_id" : NumberLong("843201233955491840"), "latest_date" : "Sat Mar 18 20:43:34 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef1"), "orig_id" : NumberLong("843201234211405825"), "latest_date" : "Sat Mar 18 20:43:34 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef2"), "orig_id" : NumberLong("843201234697965568"), "latest_date" : "Sat Mar 18 20:43:34 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef3"), "orig_id" : NumberLong("843201240783818752"), "latest_date" : "Sat Mar 18 20:43:36 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef4"), "orig_id" : NumberLong("843201254113206272"), "latest_date" : "Sat Mar 18 20:43:39 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef5"), "orig_id" : NumberLong("843201262338392064"), "latest_date" : "Sat Mar 18 20:43:41 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef6"), "orig_id" : NumberLong("843201264879984640"), "latest_date" : "Sat Mar 18 20:43:42 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef7"), "orig_id" : NumberLong("843201278645813252"), "latest_date" : "Sat Mar 18 20:43:45 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef8"), "orig_id" : NumberLong("843201297515859972"), "latest_date" : "Sat Mar 18 20:43:49 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9ef9"), "orig_id" : NumberLong("843201301848756224"), "latest_date" : "Sat Mar 18 20:43:50 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9efa"), "orig_id" : NumberLong("843201313722847232"), "latest_date" : "Sat Mar 18 20:43:53 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9efb"), "orig_id" : NumberLong("843201322287472640"), "latest_date" : "Sat Mar 18 20:43:55 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9efc"), "orig_id" : NumberLong("843201331582189568"), "latest_date" : "Sat Mar 18 20:43:58 +0000 2017'
{ "_id" : ObjectId("58f6848f56d5f53a50ba9efd"), "orig_id" : NumberLong("843201335591944194"), "latest_date" : "Sat Mar 18 20:43:58 +0000 2017'
Type "it" for more
> 
```