HOMEWORK 3

APRIL 16, 2019

XIANG YU
CS 5513 Gia-Loi Gruenwald

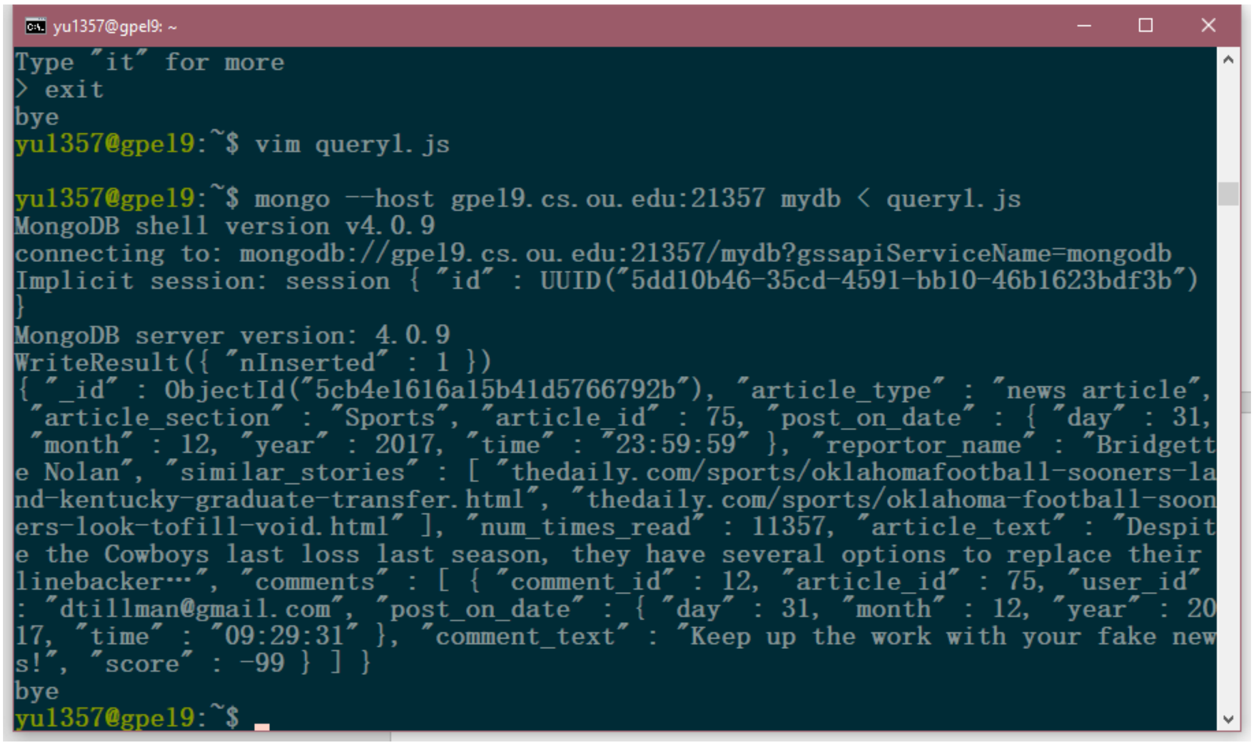
Contents

Query 1 Insert an (news or feature) article into the archive

Query1.js file:

```
db.article.insert({
        "article_type": "news article",
        "article_section": "Sports",
        "article_id": 75,
        "post_on_date": {
                                        "day": 31,
                                        "month": 12,
                                        "year": 2017,
                                        "time": '23:59:59'
                                },
        "reportor_name": "Bridgette Nolan",
        "similar_stories": ["thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html",
                        "thedaily.com/sports/oklahoma-football-sooners-look-tofill-void.html"],

        "num_times_read": 11357,
        "article_text": "Despite the Cowboys last loss last season, they have several options to replace their linebacker…",
        "comments": [
                                {
                                        "comment_id": 12,
                                        "article_id": 75,
                                        "user_id": "dtillman@gmail.com",
                                        "post_on_date": {
                                                                "day": 31,
                                                                "month": 12,
                                                                "year": 2017,
                                                                "time": "09:29:31"
                                                        },
                                        "comment_text": "Keep up the work with your fake news!",
                                        "score": -99
                                }
                        ]
});
db.article.find({"article_id": 75});
```

Query1 output result



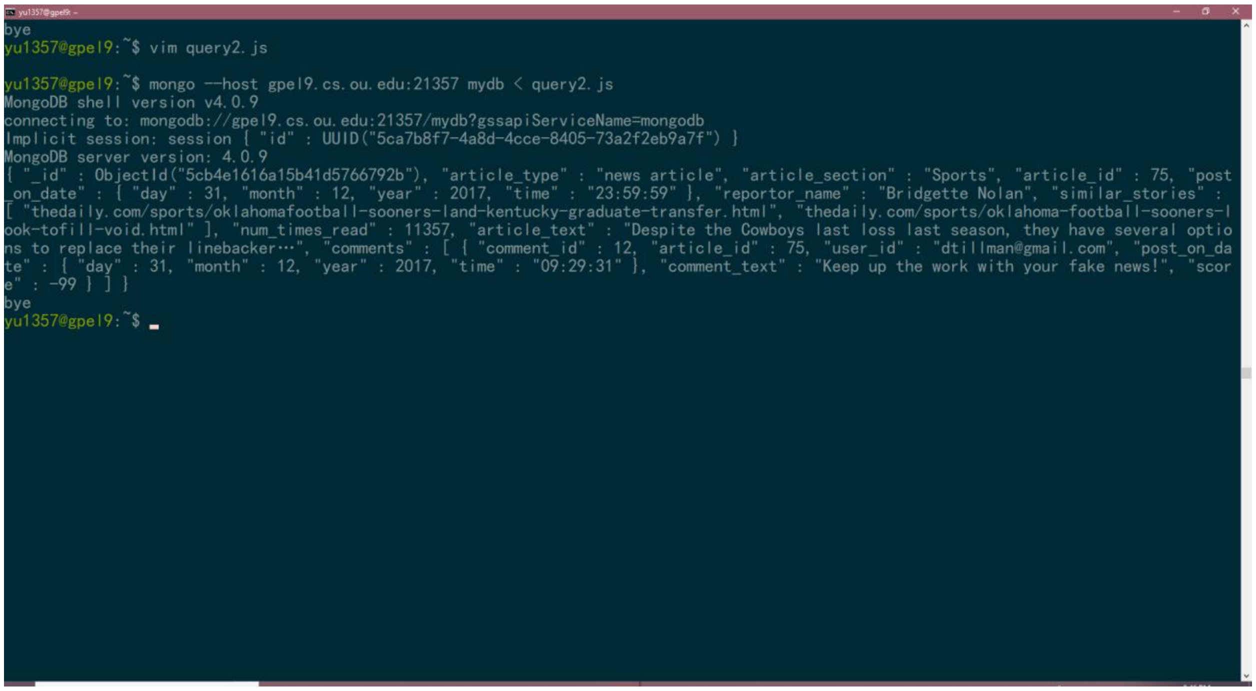
```
Type "it" for more
> exit
bye
yu1357@gpel9:~$ vim query1.js

yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < query1.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5dd10b46-35cd-4591-bb10-46b1623bdf3b")
}
MongoDB server version: 4.0.9
WriteResult({ "nInserted" : 1 })
{ "_id" : ObjectId("5cb4e1616a15b41d5766792b"), "article_type" : "news article",
 "article_section" : "Sports", "article_id" : 75, "post_on_date" : { "day" : 31,
 "month" : 12, "year" : 2017, "time" : "23:59:59" }, "reportor_name" : "Bridgett
e Nolan", "similar_stories" : [ "thedaily.com/sports/oklahomafootball-sooners-la
nd-kentucky-graduate-transfer.html", "thedaily.com/sports/oklahoma-football-soon
ers-look-tofill-void.html" ], "num_times_read" : 11357, "article_text" : "Despit
e the Cowboys last loss last season, they have several options to replace their
linebacker…", "comments" : [ { "comment_id" : 12, "article_id" : 75, "user_id"
: "dtillman@gmail.com", "post_on_date" : { "day" : 31, "month" : 12, "year" : 20
17, "time" : "09:29:31" }, "comment_text" : "Keep up the work with your fake new
s!", "score" : -99 } ] }
bye
yu1357@gpel9:~$
```

Query 2 Retrieve the latest news from the archive.

Query 2.js:

```
db.article.find({}).sort({"post_on_date.year":-1, "post_on_date.month":-1, "post_on_date.date":-1,
"post_on_date.time":-1,}).limit(1)
```

Query 2 Output:



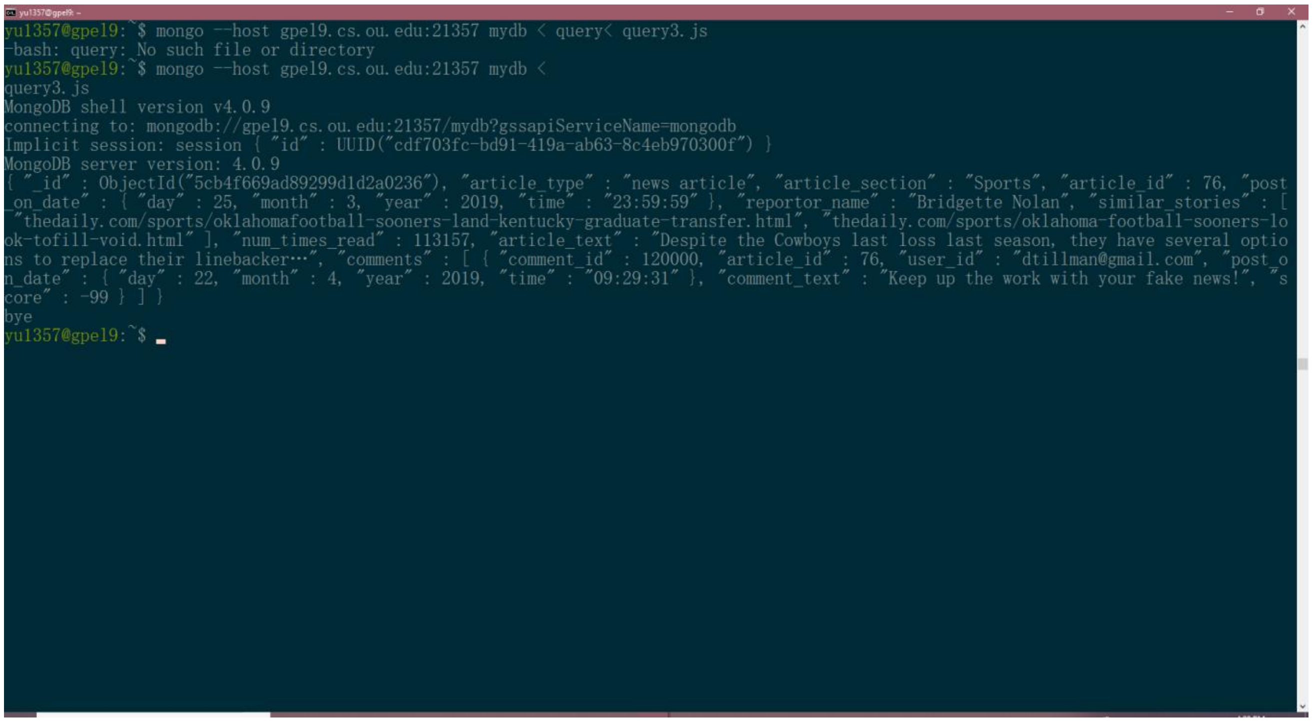
```
bye
yu1357@gpel9:~$ vim query2.js

yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < query2.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5ca7b8f7-4a8d-4cce-8405-73a2f2eb9a7f") }
MongoDB server version: 4.0.9
{ "_id" : ObjectId("5cb4e1616a15b41d5766792b"), "article_type" : "news article", "article_section" : "Sports", "article_id" : 75, "post
_on_date" : { "day" : 31, "month" : 12, "year" : 2017, "time" : "23:59:59" }, "reportor_name" : "Bridgette Nolan", "similar_stories" :
[ "thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html", "thedaily.com/sports/oklahoma-football-sooners-l
ook-tofill-void.html" ], "num_times_read" : 11357, "article_text" : "Despite the Cowboys last loss last season, they have several optio
ns to replace their linebacker…", "comments" : [ { "comment_id" : 12, "article_id" : 75, "user_id" : "dtillman@gmail.com", "post_on_da
te" : { "day" : 31, "month" : 12, "year" : 2017, "time" : "09:29:31" }, "comment_text" : "Keep up the work with your fake news!", "scor
e" : -99 } ] }
bye
yu1357@gpel9:~$
```

Query 3 Retrieve the top 10 most read news in the past month

Query3.js:

```
db.article.find({"post_on_date.year" : 2019, "post_on_date.month":3}).sort({"num_times_read":-
1}).limit(10)
```

Query3 output:



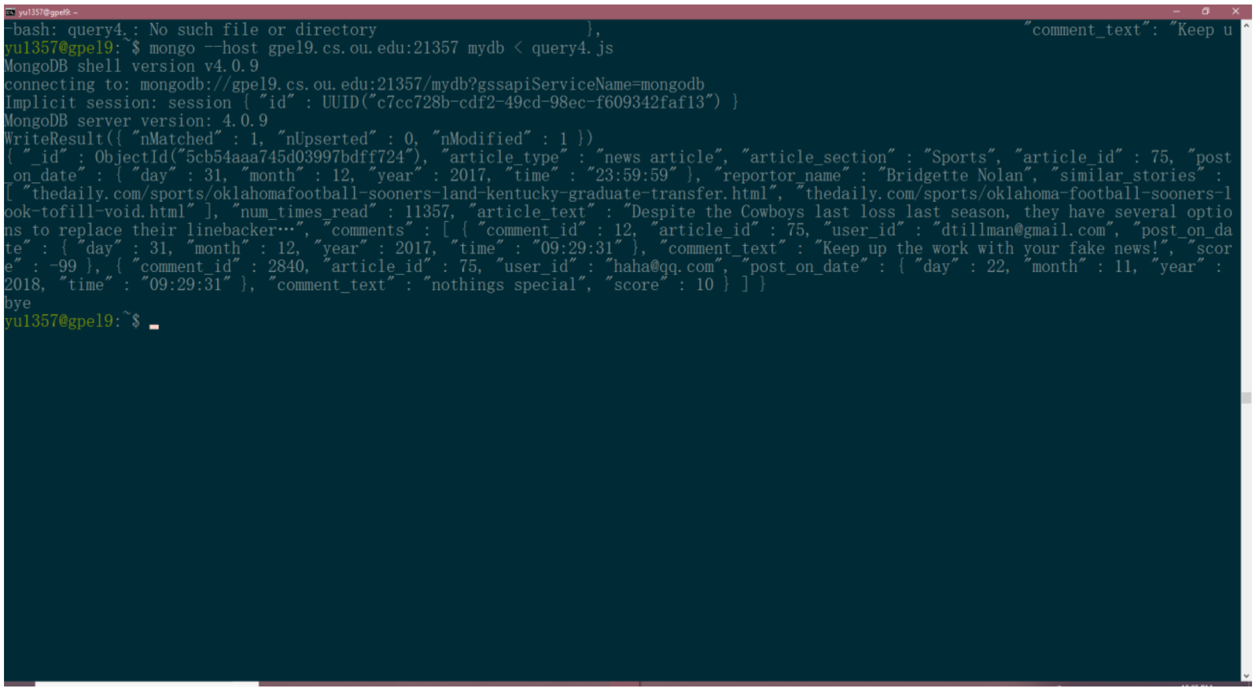
```
yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < query< query3.js
-bash: query: No such file or directory
yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb <
query3.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("cdf703fc-bd91-419a-ab63-8c4eb970300f") }
MongoDB server version: 4.0.9
{ "_id" : ObjectId("5cb4f669ad89299d1d2a0236"), "article_type" : "news article", "article_section" : "Sports", "article_id" : 76, "post
_on_date" : { "day" : 25, "month" : 3, "year" : 2019, "time" : "23:59:59" }, "reportor_name" : "Bridgette Nolan", "similar_stories" : [
 "thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html", "thedaily.com/sports/oklahoma-football-sooners-lo
ok-tofill-void.html" ], "num_times_read" : 113157, "article_text" : "Despite the Cowboys last loss last season, they have several optio
ns to replace their linebacker…", "comments" : [ { "comment_id" : 120000, "article_id" : 76, "user_id" : "dtillman@gmail.com", "post_o
n_date" : { "day" : 22, "month" : 4, "year" : 2019, "time" : "09:29:31" }, "comment_text" : "Keep up the work with your fake news!", "s
core" : -99 } ] }
bye
yu1357@gpel9:~$
```

Query 4 Add a comment to an article.

Query4.js:

```
db.article.update({"article_id": 76},{$push: {"comments": {

                    "comment_id": 2840,

                    "article_id":75,

                    "user_id": "haha@qq.com",

                    "post_on_date":{

                                        "day":22,

                                        "month": 11,

                                        "year": 2018,

                                        "time": "09:29:31"

                               },

                    "comment_text": "nothings special",

                    "score": 10

                    }
                                                                     }
                                                      });


db.article.find({"article_id": 76});
```

Query 4 output:



```
-bash: query4.: No such file or directory                          },                                           "comment_text": "Keep u
yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < query4.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c7cc728b-cdf2-49cd-98ec-f609342faf13") }
MongoDB server version: 4.0.9
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{ "_id" : ObjectId("5cb54aaa745d03997bdff724"), "article_type" : "news article", "article_section" : "Sports", "article_id" : 75, "post
_on_date" : { "day" : 31, "month" : 12, "year" : 2017, "time" : "23:59:59" }, "reportor_name" : "Bridgette Nolan", "similar_stories" :
[ "thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html", "thedaily.com/sports/oklahoma-football-sooners-l
ook-tofill-void.html" ], "num_times_read" : 11357, "article_text" : "Despite the Cowboys last loss last season, they have several optio
ns to replace their linebacker…", "comments" : [ { "comment_id" : 12, "article_id" : 75, "user_id" : "dtillman@gmail.com", "post_on_da
te" : { "day" : 31, "month" : 12, "year" : 2017, "time" : "09:29:31" }, "comment_text" : "Keep up the work with your fake news!", "scor
e" : -99 }, { "comment_id" : 2840, "article_id" : 75, "user_id" : "haha@qq.com", "post_on_date" : { "day" : 22, "month" : 11, "year" :
2018, "time" : "09:29:31" }, "comment_text" : "nothings special", "score" : 10 } ] }
bye
yu1357@gpel9:~$
```

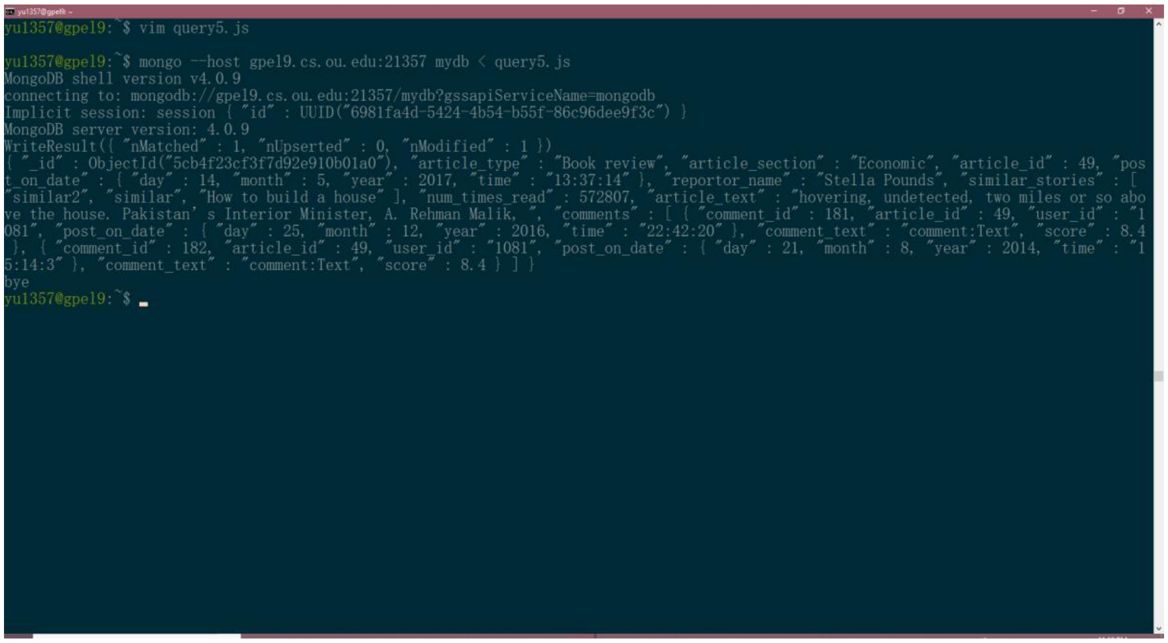
Query 5 Add a story to the ‘similar stories’ section of all articles that contain a given word and that have been published within the past 2 years.

Query5.js :

Give a word: “house”. **Add story “*How to build* a house” to all article that contain a word “house” and that published from 2017-2019 (in past 2 years)**

```
db.article.update(
        {
                "post_on_date.year" :   {$gt: 2016},
                "article_text" :        {$regex : ".*house.*"}
        },
        {
                $addToSet : {
                "similar_stories" : {
                                        $each: ["How to build a house"]
                                  }
                            }
        });
db.article.find({"post_on_date.year" :  {$gt: 2016} ,"article_text" : {$regex : ".*house.*"}});
```

Query 5 output:



```
yu1357@gpel9:~$ vim query5.js

yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < query5.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6981fa4d-5424-4b54-b55f-86c96dee9f3c") }
MongoDB server version: 4.0.9
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{ "_id" : ObjectId("5cb4f23cf3f7d92e910b01a0"), "article_type" : "Book review", "article_section" : "Economic", "article_id" : 49, "pos
t_on_date" : { "day" : 14, "month" : 5, "year" : 2017, "time" : "13:37:14" }, "reportor_name" : "Stella Pounds", "similar_stories" : [
"similar2", "similar", "How to build a house" ], "num_times_read" : 572807, "article_text" : "hovering, undetected, two miles or so abo
ve the house. Pakistan's Interior Minister, A. Rehman Malik, ", "comments" : [ { "comment_id" : 181, "article_id" : 49, "user_id" : "1
081", "post_on_date" : { "day" : 25, "month" : 12, "year" : 2016, "time" : "22:42:20" }, "comment_text" : "comment:Text", "score" : 8.4
 }, { "comment_id" : 182, "article_id" : 49, "user_id" : "1081", "post_on_date" : { "day" : 21, "month" : 8, "year" : 2014, "time" : "1
5:14:3" }, "comment_text" : "comment:Text", "score" : 8.4 } ] }
bye
yu1357@gpel9:~$ 
```

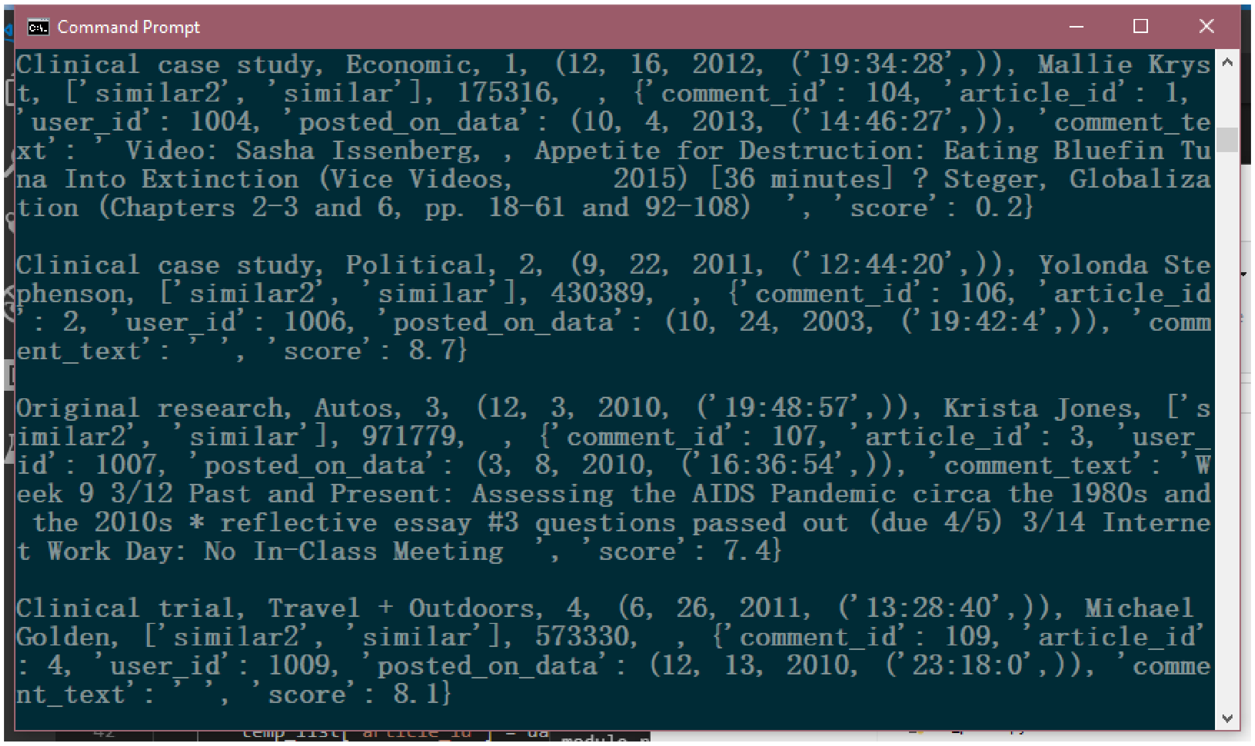
Dataset Generation

articleData.py:

```
import random

import names

from requests.exceptions import HTTPError

import codecs


article_type = ['Original research', 'Review article', 'Clinical case study',
'Clinical case study', 'Clinical trial', 'Perspective opinion and commentary',
'Book review']

article_section = ['Sports', 'Political', 'Economic', 'Entertainment', 'World',
'Technology', 'Faith & Values','Autos', 'Travel + Outdoors', 'Food +
Drink','House + Home', 'Fitness + Well-being']

def new_article_data(number):


    data_list = []

    j = 1

    for i in range(number):

        data = {}

        while True:

            try:


                j = j + 1

                break

            except HTTPError:

                j = j + 1

        data['article_type'] = random.choice(article_type)

        data['article_section'] = random.choice(article_section)

        data['article_id'] = i
        data['post_time'] = (random.randrange(1, 31), random.randrange(1, 13),
random.randrange(2000, 2019), (str(random.randrange(0, 24)) + ':' +
str(random.randrange(0, 60)) + ':' + str(random.randrange(0,60))))

        data['reporter_name'] =  names.get_full_name()

        data['similar article'] = []

        data['similar article'].append('similar2')

        data['similar article'].append('similar')


        data['number_times_read'] = random.randrange(0,1000000)

        lines = open('file.txt').read().splitlines()

        data['article_text'] = random.choice(lines)




        #comments

        temp_list = []

        for k in range(2):

            temp = ()

            commentID = 100 + j + k

            articleID = data['article_id']

            userID = 1000 + j

            postedDate = (random.randrange(1, 30), random.randrange(1, 13),
random.randrange(2000, 2019),)

            time = (str(random.randrange(0, 24)) + ':' + str(random.randrange(0,
60)) + ':' + str(random.randrange(0,60)),)

            commentText = 'comment:Text'

            score = random.randrange(2, 100)/10

            temp += (commentID, articleID, userID) + (postedDate + time, ) +
(commentText,) + (score,)

            temp_list.append(temp)

        data['comments'] = temp_list
```

```python
        data_list.append(data)




    # write to file: articleData.txt
    with codecs.open('./articleData.txt', 'a') as f:
        for data in data_list:
            keys = list(data.keys())
            for i in range(len(keys)):
                value = data[keys[i]]
                f.write(str(value))
                if i != len(keys) - 1:
                    f.write(', ')
                else:
                    pass
            f.write('\n')

    # a) print the first 5 entries
    with codecs.open('articleData.txt', 'r') as f:
        for i in range(5):
            print(f.readline())

if __name__ == "__main__":
    # create 50 article datas
    new_article_data(50)
```

Dataset Generation Output:



```
Clinical case study, Economic, 1, (12, 16, 2012, ('19:34:28',)), Mallie Krys
t, ['similar2', 'similar'], 175316,  , {'comment_id': 104, 'article_id': 1,
'user_id': 1004, 'posted_on_data': (10, 4, 2013, ('14:46:27',)), 'comment_te
xt': ' Video: Sasha Issenberg, , Appetite for Destruction: Eating Bluefin Tu
na Into Extinction (Vice Videos,       2015) [36 minutes] ? Steger, Globaliza
tion (Chapters 2-3 and 6, pp. 18-61 and 92-108)  ', 'score': 0.2}

Clinical case study, Political, 2, (9, 22, 2011, ('12:44:20',)), Yolonda Ste
phenson, ['similar2', 'similar'], 430389,  , {'comment_id': 106, 'article_id
': 2, 'user_id': 1006, 'posted_on_data': (10, 24, 2003, ('19:42:4',)), 'comm
ent_text': ' ', 'score': 8.7}

Original research, Autos, 3, (12, 3, 2010, ('19:48:57',)), Krista Jones, ['s
imilar2', 'similar'], 971779,  , {'comment_id': 107, 'article_id': 3, 'user_
id': 1007, 'posted_on_data': (3, 8, 2010, ('16:36:54',)), 'comment_text': 'W
eek 9 3/12 Past and Present: Assessing the AIDS Pandemic circa the 1980s and
 the 2010s * reflective essay #3 questions passed out (due 4/5) 3/14 Interne
t Work Day: No In-Class Meeting  ', 'score': 7.4}

Clinical trial, Travel + Outdoors, 4, (6, 26, 2011, ('13:28:40',)), Michael
Golden, ['similar2', 'similar'], 573330,  , {'comment_id': 109, 'article_id'
: 4, 'user_id': 1009, 'posted_on_data': (12, 13, 2010, ('23:18:0',)), 'comme
nt_text': ' ', 'score': 8.1}
```

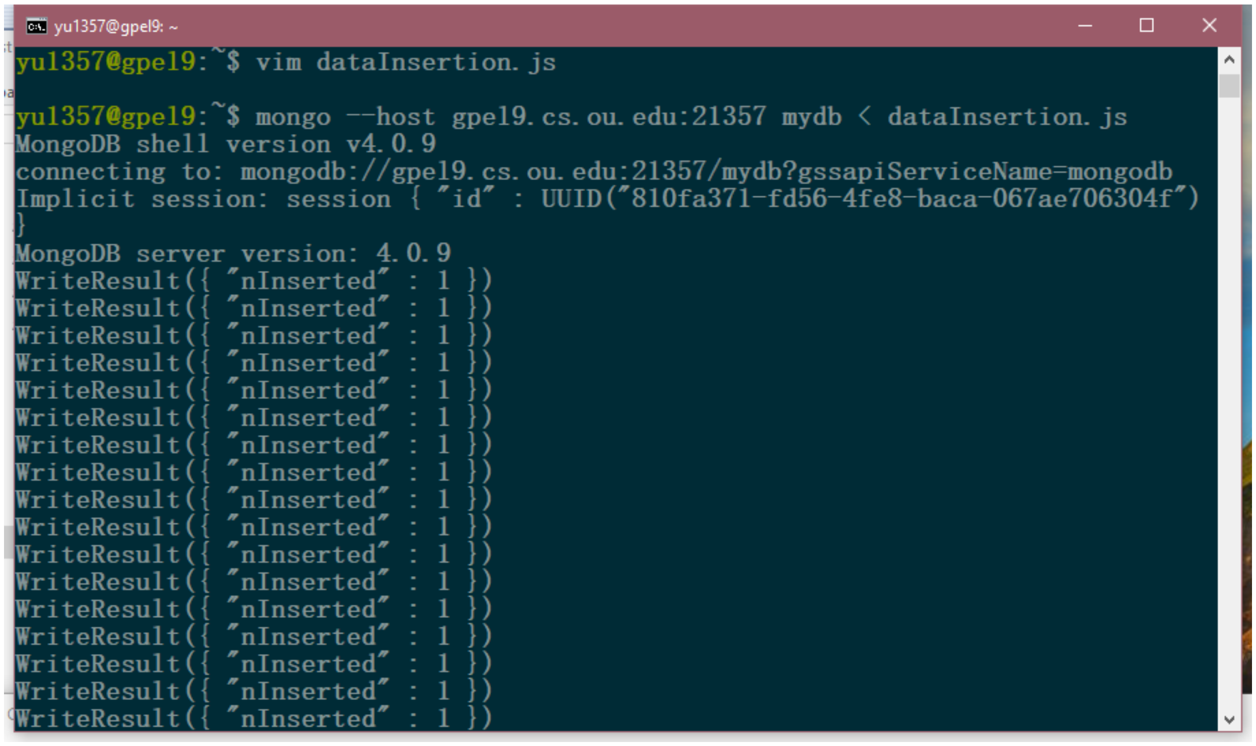
Data Insertion

dataInsertion.js: (First five insertions display)

```
connection = new Mongo()

db = connection.getDB('mydb')

db.article.insert({"article_type": "Clinical case study", "article_section": "Entertainment", "article_id": 0, "post_on_date": {"day": 15, "month": 4, "year": 2011, "time": '4:17:15'}, "reportor_name": "Robert Cox", "similar_stories": ['similar2', 'similar'], "num_times_read": 641088, "article_text": "father-in-law, his motherin-law, a lieutenant, and seven bodyguard", "comments": [{"comment_id": 103, "article_id": 0, "user_id": "1003", "post_on_date": {"day": 26, "month": 5, "year": 2014, "time": '2:34:0'}, "comment_text": 'comment:Text', "score": 6.1}, {"comment_id": 104, "article_id": 0, "user_id": "1003", "post_on_date": {"day": 8, "month": 10, "year": 2006, "time": '2:53:32'}, "comment_text": 'comment:Text', "score": 6.6}]})

db.article.insert({"article_type": "Clinical case study", "article_section": "Travel + Outdoors", "article_id": 1, "post_on_date": {"day": 28, "month": 5, "year": 2004, "time": '16:46:26'}, "reportor_name": "Crystal Dole", "similar_stories": ['similar2', 'similar'], "num_times_read": 22995, "article_text": "told me recently that Mehsud was resting on his back. Malik, using his hands to make a picture frame, explained that ", "comments": [{"comment_id": 104, "article_id": 1, "user_id": "1004", "post_on_date": {"day": 18, "month": 11, "year": 2006, "time": '12:49:44'}, "comment_text": 'comment:Text', "score": 2.0}, {"comment_id": 105, "article_id": 1, "user_id": "1004", "post_on_date": {"day": 19, "month": 5, "year": 2014, "time": '6:35:11'}, "comment_text": 'comment:Text', "score": 9.7}]})

db.article.insert({"article_type": "Perspective opinion and commentary", "article_section": "Entertainment", "article_id": 2, "post_on_date": {"day": 13, "month": 3, "year": 2008, "time": '10:11:6'}, "reportor_name": "Katherine Ruiz", "similar_stories": ['similar2', 'similar'], "num_times_read": 46472, "article_text": "Mehsud\u2019s death is instant. Nor, described unambiguously as a terrorist, does he seem undeserving ", "comments": [{"comment_id": 106, "article_id": 2, "user_id": "1006", "post_on_date": {"day": 13, "month": 1, "year": 2015, "time": '14:16:25'}, "comment_text": 'comment:Text', "score": 7.7}, {"comment_id": 107, "article_id": 2, "user_id": "1006", "post_on_date": {"day": 4, "month": 4, "year": 2006, "time": '23:7:58'}, "comment_text": 'comment:Text', "score": 1.7}]})

db.article.insert({"article_type": "Perspective opinion and commentary", "article_section": "Fitness + Well-being", "article_id": 3, "post_on_date": {"day": 7, "month": 10, "year": 2014, "time": '3:45:23'}, "reportor_name": "Carrie Robinson", "similar_stories": ['similar2', 'similar'], "num_times_read": 252900, "article_text": "Study. It is hard to imagine an environment that is more stimulating or more congenial", "comments": [{"comment_id": 107, "article_id": 3, "user_id": "1007", "post_on_date": {"day": 7, "month": 9, "year": 2001, "time": '21:16:46'}, "comment_text": 'comment:Text', "score": 5.1}, {"comment_id": 108, "article_id": 3, "user_id": "1007", "post_on_date": {"day": 2, "month": 12, "year": 2011, "time": '13:41:12'}, "comment_text": 'comment:Text', "score": 9.3}]})
```

db.article.insert({"article_type": "Perspective opinion and commentary", "article_section": "Entertainment", "article_id": 4, "post_on_date": {"day": 11, "month": 5, "year": 2006, "time": '2:59:16'}, "reportor_name": "Lucy Harper", "similar_stories": ['similar2', 'similar'], "num_times_read": 411832, "article_text": "This book was written when I spent a year at the Princeton Institute for Advanced", "comments": [{"comment_id": 109, "article_id": 4, "user_id": "1009", "post_on_date": {"day": 20, "month": 5, "year": 2003, "time": '18:54:32'}, "comment_text": 'comment:Text', "score": 7.6}, {"comment_id": 110, "article_id": 4, "user_id": "1009", "post_on_date": {"day": 22, "month": 11, "year": 2014, "time": '4:15:50'}, "comment_text": 'comment:Text', "score": 3.4}]})

Data Insertion Output:



```
yu1357@gpel9:~$ vim dataInsertion.js

yu1357@gpel9:~$ mongo --host gpel9.cs.ou.edu:21357 mydb < dataInsertion.js
MongoDB shell version v4.0.9
connecting to: mongodb://gpel9.cs.ou.edu:21357/mydb?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("810fa371-fd56-4fe8-baca-067ae706304f")
}
MongoDB server version: 4.0.9
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
```

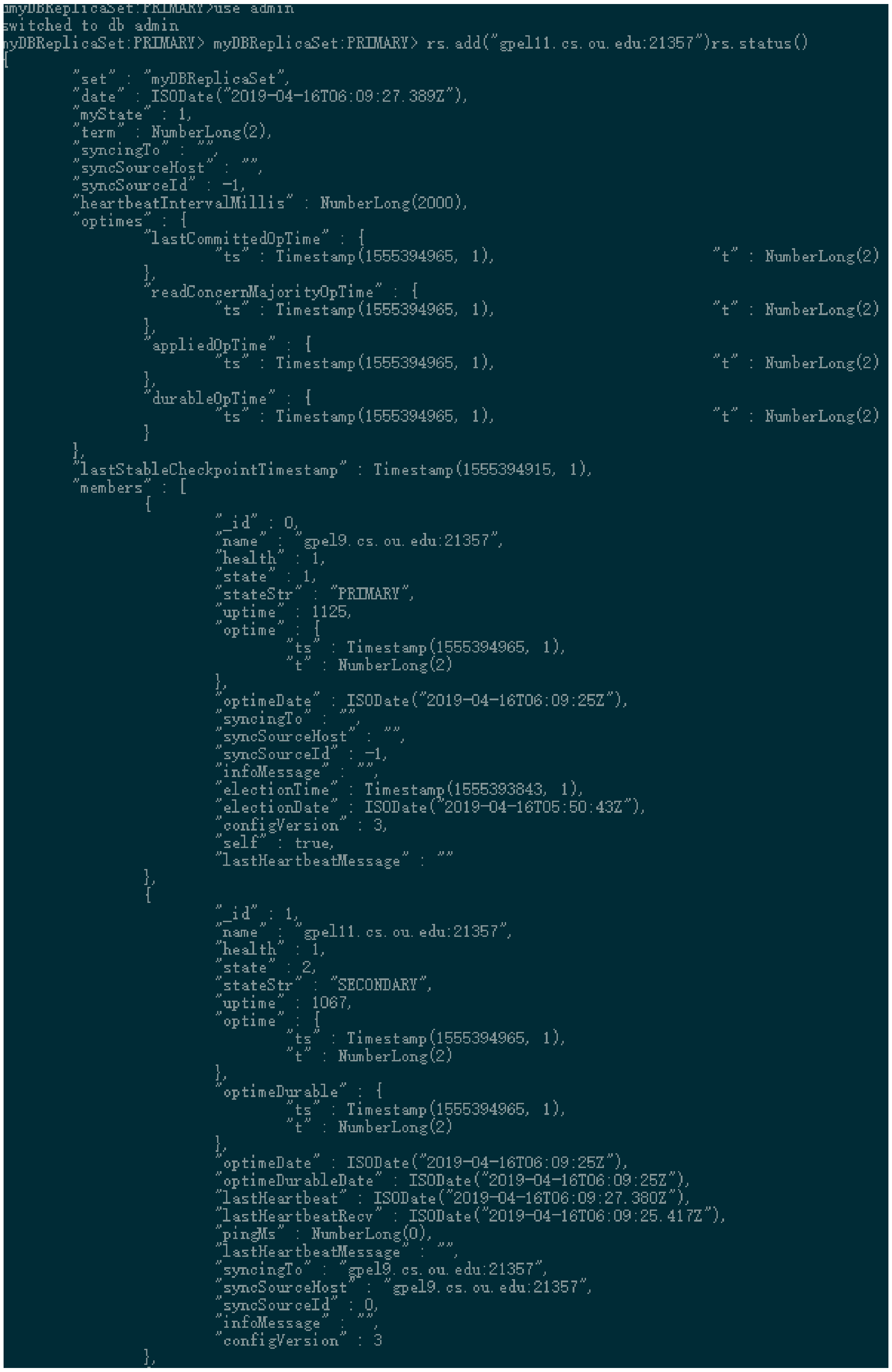
Txt parser

```python
import codecs
import json

def txt_parser():
    with codecs.open('./articleData.txt', 'r') as f1:
        with codecs.open('./dataInsertion.js', 'a') as f2:
            f2.write('connection = new Mongo()\n')
            f2.write('db = connection.getDB(\'mydb\')\n')

            lines = f1.readlines()
            for line in lines:
                lbrackets = [pos for pos, char in enumerate(line) if char == '[']
                rbrackets = [pos for pos, char in enumerate(line) if char == ']']

                post_on_date = {}
                left = line.find(', (')
                right = line[left:].find(')') + left
                post_on_date_list = [word.strip() for word in line[left + 3:
right].split(',')]
                post_on_date['day'] = int(post_on_date_list[0])
                post_on_date['month'] = int(post_on_date_list[1])
                post_on_date['year'] = int(post_on_date_list[2])
                post_on_date['time'] = post_on_date_list[3]

                similars = line[lbrackets[0]: rbrackets[0] + 1]
                commentRaw = line[lbrackets[1] : rbrackets[1]]
                commentRaw = commentRaw.replace('[', '')
                commentRaw = commentRaw.replace(']', '')
                commentRaw = commentRaw.replace('(', '')
                commentRaw = commentRaw.replace(')', '')
                comments_list = [word.strip() for word in commentRaw.split(',')]
                comments = []
                numberOfcomments = len(comments_list)//9
                for j in range(numberOfcomments):
                   comment = {}
                   comment['comment_id'] = int(comments_list[j*9 + 0])
                   comment['article_id'] = int(comments_list[j*9 + 1])
                   comment['user_id'] = comments_list[j*9 + 2]
                   comment['post_on_date'] = {}
                   comment['post_on_date']['day'] = int(comments_list[j*9 + 3])
                   comment['post_on_date']['month'] = int(comments_list[j*9 + 4])
                   comment['post_on_date']['year'] = int(comments_list[j*9 + 5])
                   comment['post_on_date']['time'] = comments_list[j*9 + 6]
                   comment['comment_text'] = comments_list[j*9 + 7]
                   comment['score'] = float(comments_list[j*9 + 8])
                   comments.append(comment)

                lbrackets = [pos for pos, char in enumerate(line) if char == '[']
                rbrackets = [pos for pos, char in enumerate(line) if char == ']']
                restString = line[0: left]
                restString += line[right : lbrackets[0] - 2]
                restString = restString.replace(')','')
                restString = restString.replace('(','')
                words = [word.strip() for word in restString.split(',')]
                readStartIndex = rbrackets[0] + 3
                readEndIndex = line[readStartIndex:].find(',') + readStartIndex

                record = {}
                record['article_type'] = words[0]
                record['article_section'] = words[1]
                record['article_id'] = int(words[2])
                record['post_on_date'] = post_on_date
                record['reportor_name'] = words[3]
                record['similar_stories'] =  similars
                record['num_times_read'] = int(line[readStartIndex:readEndIndex])
                record['article_text'] = line[readEndIndex + 2 : lbrackets[1] -
2]
                record['comments'] = comments

                recordRaw = json.dumps(record)
                recordRaw = recordRaw.replace('"[', '[')
                recordRaw = recordRaw.replace(']"', ']')
                recordRaw = recordRaw.replace('"\'', '\'')
                recordRaw = recordRaw.replace('\'"', '\'')
                record = 'db.article.insert(' + recordRaw + ')\n'

                f2.write(record)

if __name__ == '__main__':
    txt_parser()
```

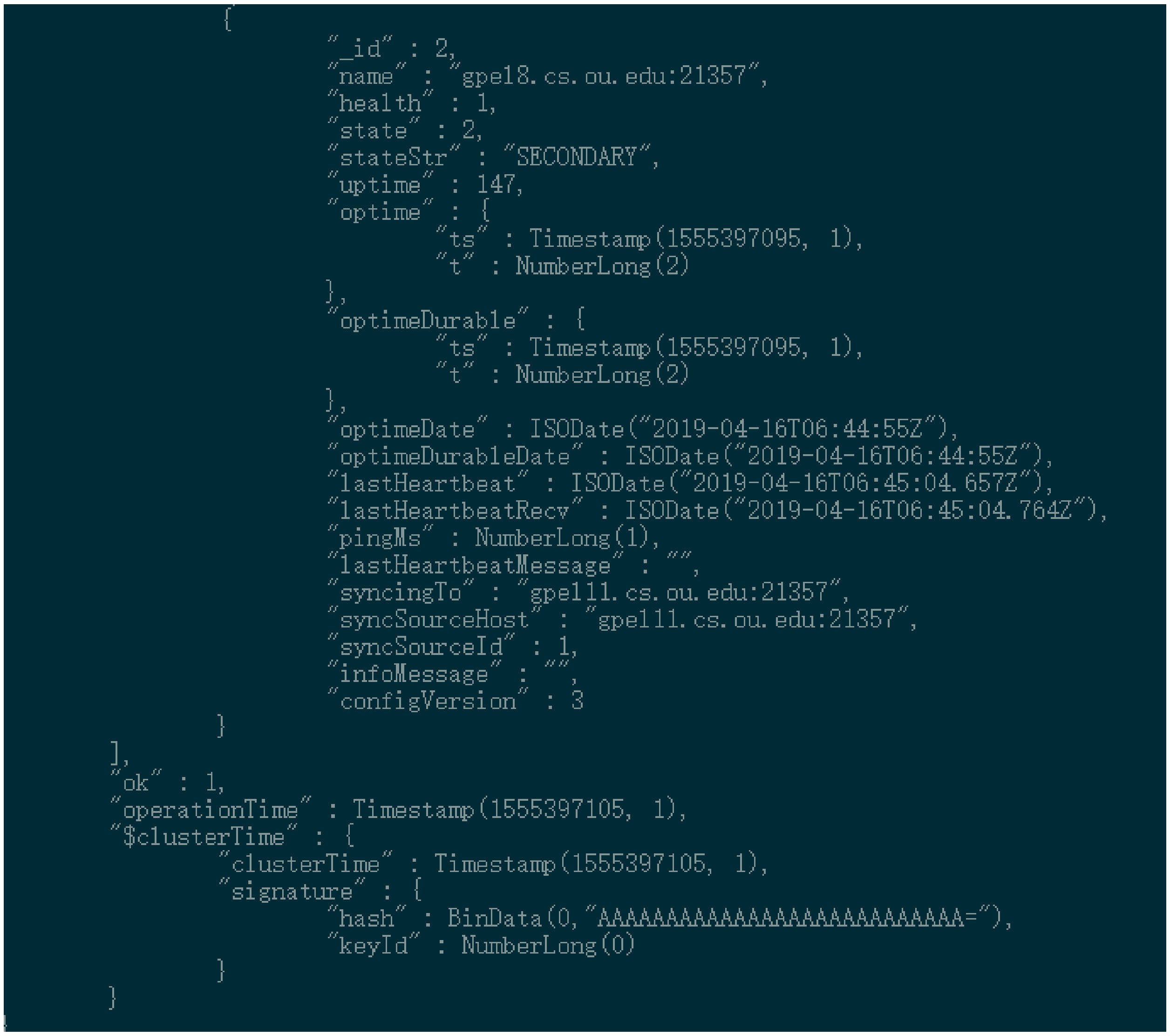
Task f. Set Replicates

```
mkdir -p /home/yu1357/data/gpel8
mongod --fork --dbpath ~/data/gpel8/ --port 21357 --logpath mongodb.log --replSet
myDBReplicaSet --bind_ip_all

mkdir -p /home/yu1357/data/gpel9
mongod --fork --dbpath ~/data/gpel9/ --port 21357 --logpath mongodb.log --replSet
myDBReplicaSet --bind_ip_all

mkdir -p /home/yu1357/data/gpel11
mongod --fork --dbpath ~/data/gpel11/ --port 21357 --logpath mongodb.log --
replSet myDBReplicaSet --bind_ip_all

HOST:
mongo –host gpel9.cs.ou.edu --port 21357 mydb

rs.initiate()
myDBReplicaSet:PRIMARY> rs.add("gpel8.cs.ou.edu:21357")
myDBReplicaSet:PRIMARY> rs.add("gpel11.cs.ou.edu:21357")
use admin
rs.status()
```

rs.status() output



```
mmyDBReplicaSet:PRIMARY>use admin
switched to db admin
myDBReplicaSet:PRIMARY> myDBReplicaSet:PRIMARY> rs.add("gpel11.cs.ou.edu:21357")rs.status()
{
        "set" : "myDBReplicaSet",
        "date" : ISODate("2019-04-16T06:09:27.389Z"),
        "myState" : 1,
        "term" : NumberLong(2),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1555394965, 1),                         "t" : NumberLong(2)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1555394965, 1),                         "t" : NumberLong(2)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1555394965, 1),                         "t" : NumberLong(2)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1555394965, 1),                         "t" : NumberLong(2)
                }
        },
        "lastStableCheckpointTimestamp" : Timestamp(1555394915, 1),
        "members" : [
                {
                        "_id" : 0,
                        "name" : "gpel9.cs.ou.edu:21357",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 1125,
                        "optime" : {
                                "ts" : Timestamp(1555394965, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:09:25Z"),
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1555393843, 1),
                        "electionDate" : ISODate("2019-04-16T05:50:43Z"),
                        "configVersion" : 3,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                },
                {
                        "_id" : 1,
                        "name" : "gpel11.cs.ou.edu:21357",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 1067,
                        "optime" : {
                                "ts" : Timestamp(1555394965, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1555394965, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:09:25Z"),
                        "optimeDurableDate" : ISODate("2019-04-16T06:09:25Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:09:27.380Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:09:25.417Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "gpel9.cs.ou.edu:21357",
                        "syncSourceHost" : "gpel9.cs.ou.edu:21357",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 3
                },
```



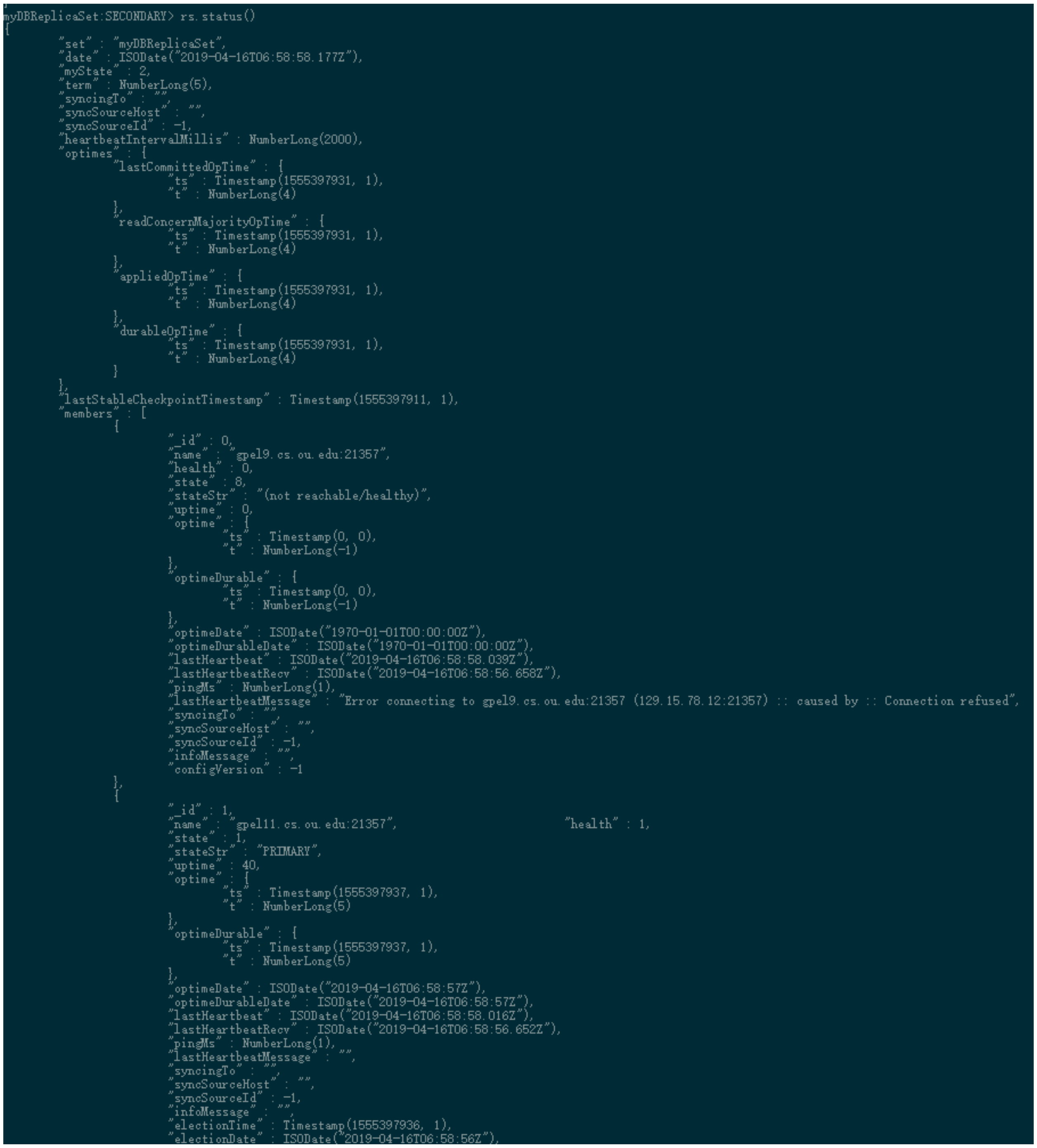
```
                {
                        "_id" : 2,
                        "name" : "gpel8.cs.ou.edu:21357",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 147,
                        "optime" : {
                                "ts" : Timestamp(1555397095, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1555397095, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:44:55Z"),
                        "optimeDurableDate" : ISODate("2019-04-16T06:44:55Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:45:04.657Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:45:04.764Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "gpel11.cs.ou.edu:21357",
                        "syncSourceHost" : "gpel11.cs.ou.edu:21357",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 3
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1555397105, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1555397105, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
```

Replication Design Explanation:

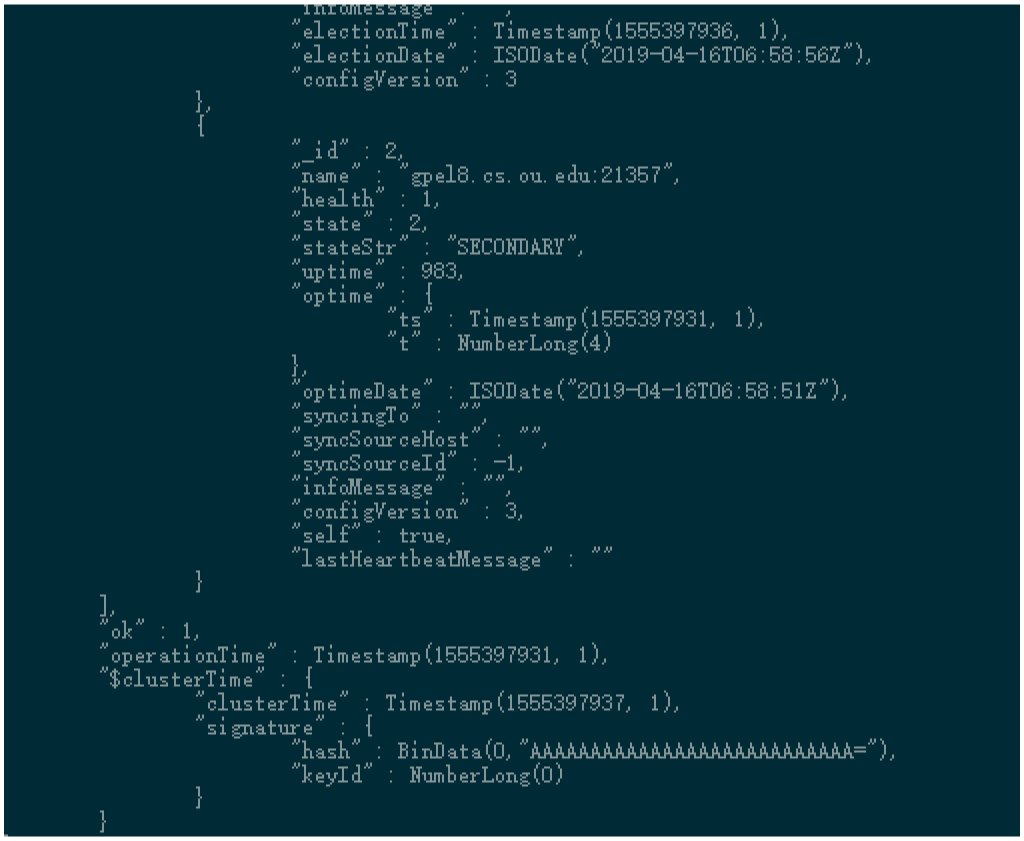
I have created multiple mongod processes sharing same data. If there is one node fail, for example if primary node gpel9 fails, and then the replica set will detect this situation and elect a new primary node, therefore, user could continue access data he/she needs. What is more, if secondary node fails like gpel8, the database still can be accessible. This replica design could increase availability of database.

Task G

Immediately connect to another active MongoDB server in your replica set and run the command rs.status(). Take a screenshot that shows that no primary has yet been elected.

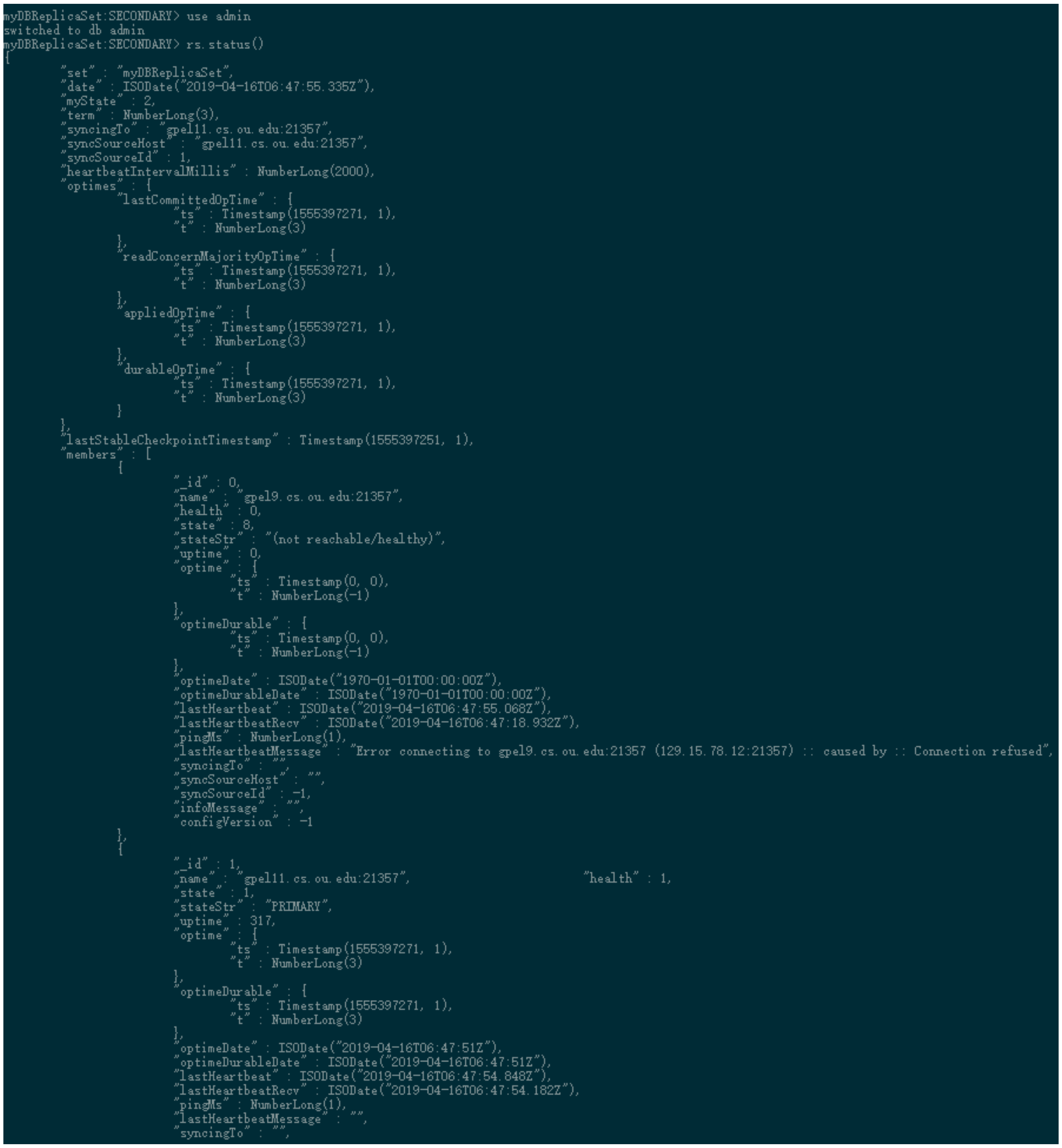


```
myDBReplicaSet:SECONDARY> rs.status()
{
        "set" : "myDBReplicaSet",
        "date" : ISODate("2019-04-16T06:58:58.177Z"),
        "myState" : 2,
        "term" : NumberLong(5),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1555397931, 1),
                        "t" : NumberLong(4)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1555397931, 1),
                        "t" : NumberLong(4)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1555397931, 1),
                        "t" : NumberLong(4)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1555397931, 1),
                        "t" : NumberLong(4)
                }
        },
        "lastStableCheckpointTimestamp" : Timestamp(1555397911, 1),
        "members" : [
                {
                        "_id" : 0,
                        "name" : "gpel9.cs.ou.edu:21357",
                        "health" : 0,
                        "state" : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        "optime" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
                        "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:58:58.039Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:58:56.658Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "Error connecting to gpel9.cs.ou.edu:21357 (129.15.78.12:21357) :: caused by :: Connection refused",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : -1
                },
                {
                        "_id" : 1,
                        "name" : "gpel11.cs.ou.edu:21357",                          "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 40,
                        "optime" : {
                                "ts" : Timestamp(1555397937, 1),
                                "t" : NumberLong(5)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1555397937, 1),
                                "t" : NumberLong(5)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:58:57Z"),
                        "optimeDurableDate" : ISODate("2019-04-16T06:58:57Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:58:58.016Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:58:56.652Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1555397936, 1),
                        "electionDate" : ISODate("2019-04-16T06:58:56Z"),
```

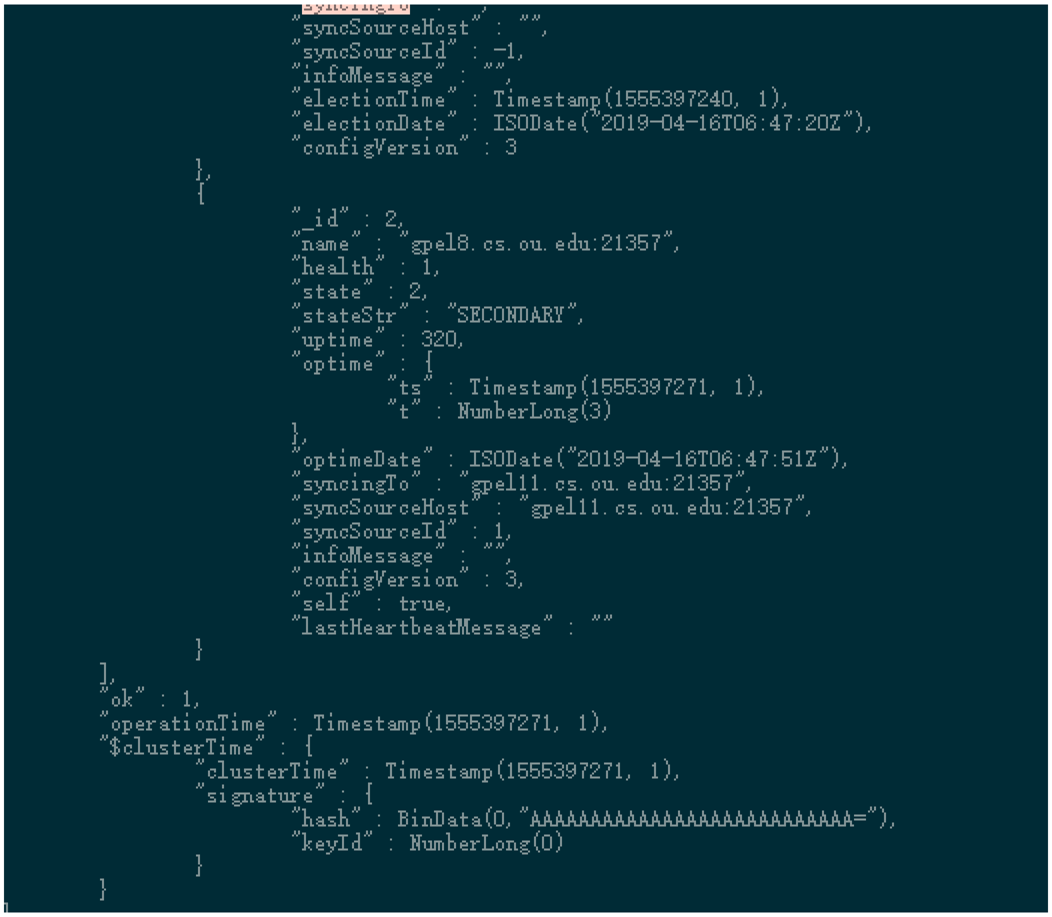


```
                "infoMessage" : "",
                "electionTime" : Timestamp(1555397936, 1),
                "electionDate" : ISODate("2019-04-16T06:58:56Z"),
                "configVersion" : 3
        },
        {
                "_id" : 2,
                "name" : "gpel8.cs.ou.edu:21357",
                "health" : 1,
                "state" : 2,
                "stateStr" : "SECONDARY",
                "uptime" : 983,
                "optime" : {
                        "ts" : Timestamp(1555397931, 1),
                        "t" : NumberLong(4)
                },
                "optimeDate" : ISODate("2019-04-16T06:58:51Z"),
                "syncingTo" : "",
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "configVersion" : 3,
                "self" : true,
                "lastHeartbeatMessage" : ""
        }
],
"ok" : 1,
"operationTime" : Timestamp(1555397931, 1),
"$clusterTime" : {
        "clusterTime" : Timestamp(1555397937, 1),
        "signature" : {
                "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                "keyId" : NumberLong(0)
        }
}
```

Wait until a primary has been elected, and then run your queries one time each again with the remaining nodes. Provide screenshots for the output of each run

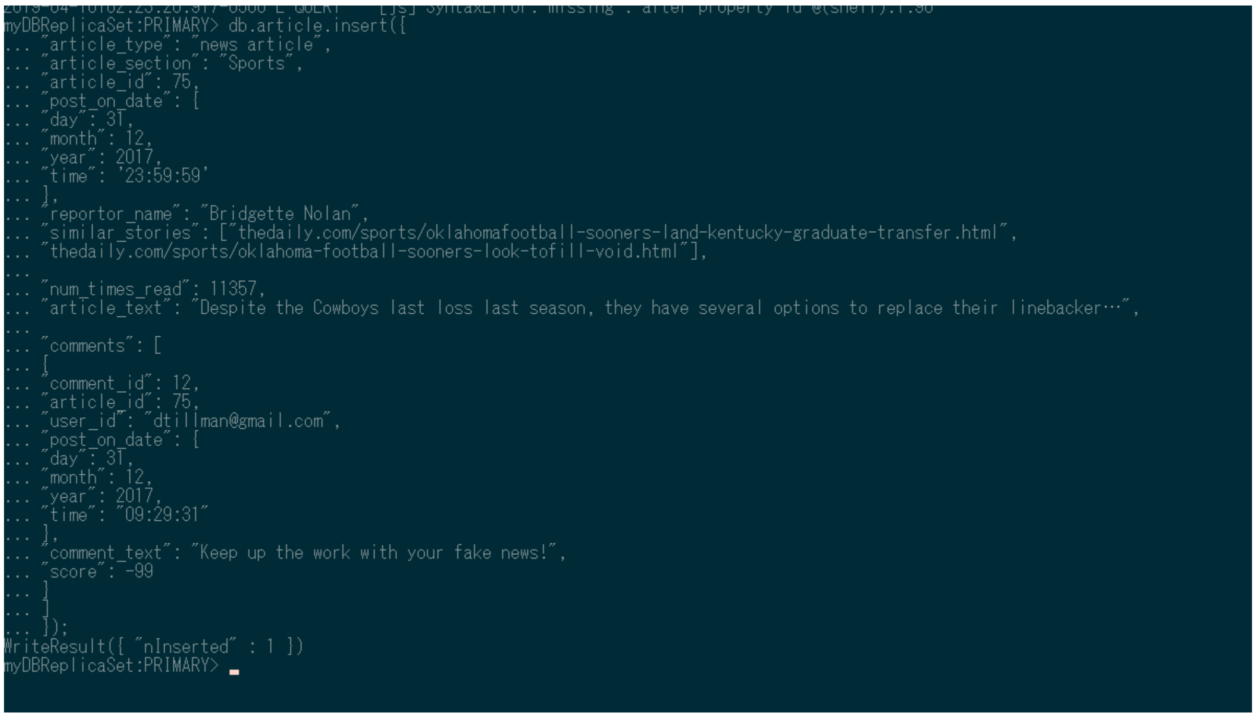


```
myDBReplicaSet:SECONDARY> use admin
switched to db admin
myDBReplicaSet:SECONDARY> rs.status()
{
        "set" : "myDBReplicaSet",
        "date" : ISODate("2019-04-16T06:47:55.335Z"),
        "myState" : 2,
        "term" : NumberLong(3),
        "syncingTo" : "gpel11.cs.ou.edu:21357",
        "syncSourceHost" : "gpel11.cs.ou.edu:21357",
        "syncSourceId" : 1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1555397271, 1),
                        "t" : NumberLong(3)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1555397271, 1),
                        "t" : NumberLong(3)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1555397271, 1),
                        "t" : NumberLong(3)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1555397271, 1),
                        "t" : NumberLong(3)
                }
        },
        "lastStableCheckpointTimestamp" : Timestamp(1555397251, 1),
        "members" : [
                {
                        "_id" : 0,
                        "name" : "gpel9.cs.ou.edu:21357",
                        "health" : 0,
                        "state" : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        "optime" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
                        "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:47:55.068Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:47:18.932Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "Error connecting to gpel9.cs.ou.edu:21357 (129.15.78.12:21357) :: caused by :: Connection refused",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : -1
                },
                {
                        "_id" : 1,
                        "name" : "gpel11.cs.ou.edu:21357",                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 317,
                        "optime" : {
                                "ts" : Timestamp(1555397271, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1555397271, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:47:51Z"),
                        "optimeDurableDate" : ISODate("2019-04-16T06:47:51Z"),
                        "lastHeartbeat" : ISODate("2019-04-16T06:47:54.848Z"),
                        "lastHeartbeatRecv" : ISODate("2019-04-16T06:47:54.182Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
```



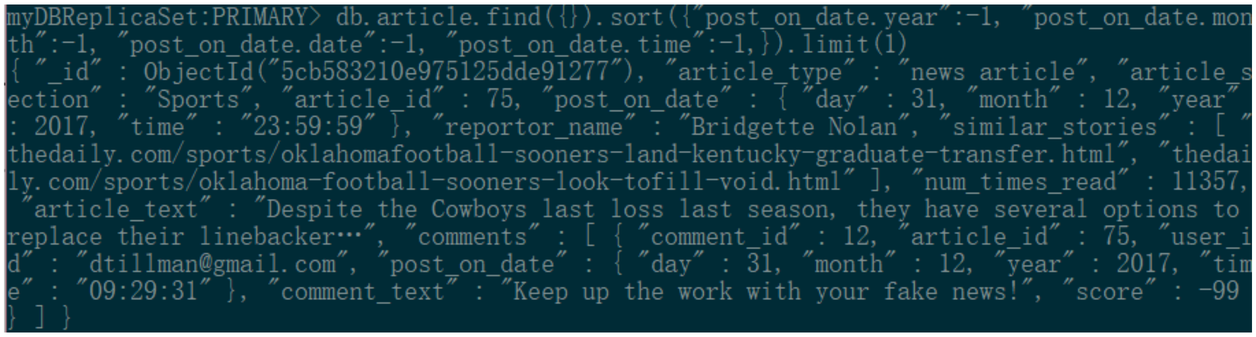
```
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1555397240, 1),
                        "electionDate" : ISODate("2019-04-16T06:47:20Z"),
                        "configVersion" : 3
                },
                {
                        "_id" : 2,
                        "name" : "gpel8.cs.ou.edu:21357",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 320,
                        "optime" : {
                                "ts" : Timestamp(1555397271, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2019-04-16T06:47:51Z"),
                        "syncingTo" : "gpel11.cs.ou.edu:21357",
                        "syncSourceHost" : "gpel11.cs.ou.edu:21357",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 3,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1555397271, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1555397271, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
```

Run query1



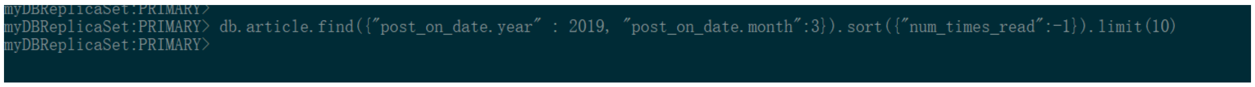
```
2019-04-10T02:23:20.917-0500 E QUERY    [js] SyntaxError: missing : after property id @(shell):1:30
myDBReplicaSet:PRIMARY> db.article.insert([
... "article_type": "news article",
... "article_section": "Sports",
... "article_id": 75,
... "post_on_date": [
... "day": 31,
... "month": 12,
... "year": 2017,
... "time": '23:59:59'
... ],
... "reportor_name": "Bridgette Nolan",
... "similar_stories": ["thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html",
... "thedaily.com/sports/oklahoma-football-sooners-look-tofill-void.html"],
...
... "num_times_read": 11357,
... "article_text": "Despite the Cowboys last loss last season, they have several options to replace their linebacker…",
...
... "comments": [
... [
... "comment_id": 12,
... "article_id": 75,
... "user_id": "dtillman@gmail.com",
... "post_on_date": [
... "day": 31,
... "month": 12,
... "year": 2017,
... "time": "09:29:31"
... ],
... "comment_text": "Keep up the work with your fake news!",
... "score": -99
... ]
... ]
... ]);
WriteResult([ "nInserted" : 1 ])
myDBReplicaSet:PRIMARY>
```

Run Query 2:



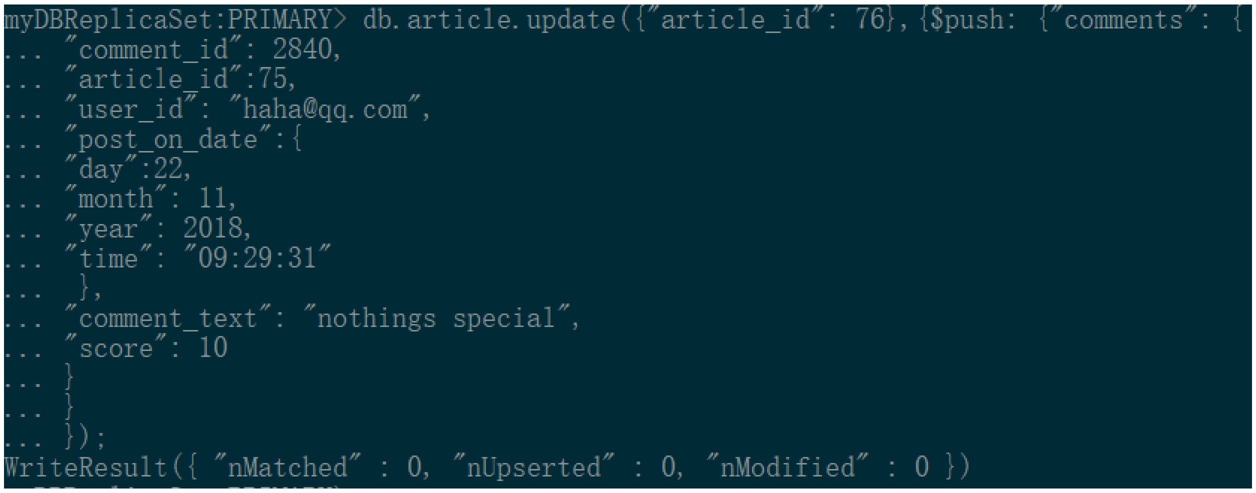
```
myDBReplicaSet:PRIMARY> db.article.find({}).sort({"post_on_date.year":-1, "post_on_date.mon
th":-1, "post_on_date.date":-1, "post_on_date.time":-1,}).limit(1)
{ "_id" : ObjectId("5cb583210e975125dde91277"), "article_type" : "news article", "article_s
ection" : "Sports", "article_id" : 75, "post_on_date" : { "day" : 31, "month" : 12, "year"
: 2017, "time" : "23:59:59" }, "reportor_name" : "Bridgette Nolan", "similar_stories" : [ "
thedaily.com/sports/oklahomafootball-sooners-land-kentucky-graduate-transfer.html", "thedai
ly.com/sports/oklahoma-football-sooners-look-tofill-void.html" ], "num_times_read" : 11357,
 "article_text" : "Despite the Cowboys last loss last season, they have several options to
replace their linebacker…", "comments" : [ { "comment_id" : 12, "article_id" : 75, "user_i
d" : "dtillman@gmail.com", "post_on_date" : { "day" : 31, "month" : 12, "year" : 2017, "tim
e" : "09:29:31" }, "comment_text" : "Keep up the work with your fake news!", "score" : -99
} ] }
```

Run Query 3:



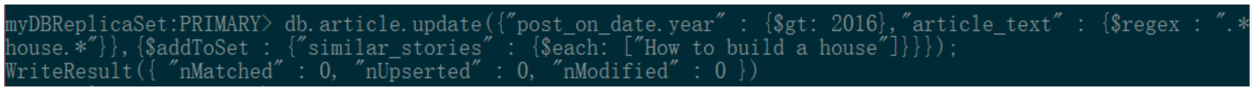
```
myDBReplicaSet:PRIMARY>
myDBReplicaSet:PRIMARY> db.article.find({"post_on_date.year" : 2019, "post_on_date.month":3}).sort({"num_times_read":-1}).limit(10)
myDBReplicaSet:PRIMARY>
```

Run Query 4:



```
myDBReplicaSet:PRIMARY> db.article.update({"article_id": 76},{$push: {"comments": {
...  "comment_id": 2840,
...  "article_id":75,
...  "user_id": "haha@qq.com",
...  "post_on_date":{
...  "day":22,
...  "month": 11,
...  "year": 2018,
...  "time": "09:29:31"
...   },
...  "comment_text": "nothings special",
...  "score": 10
...  }
...  }
...  });
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

Run Query 5

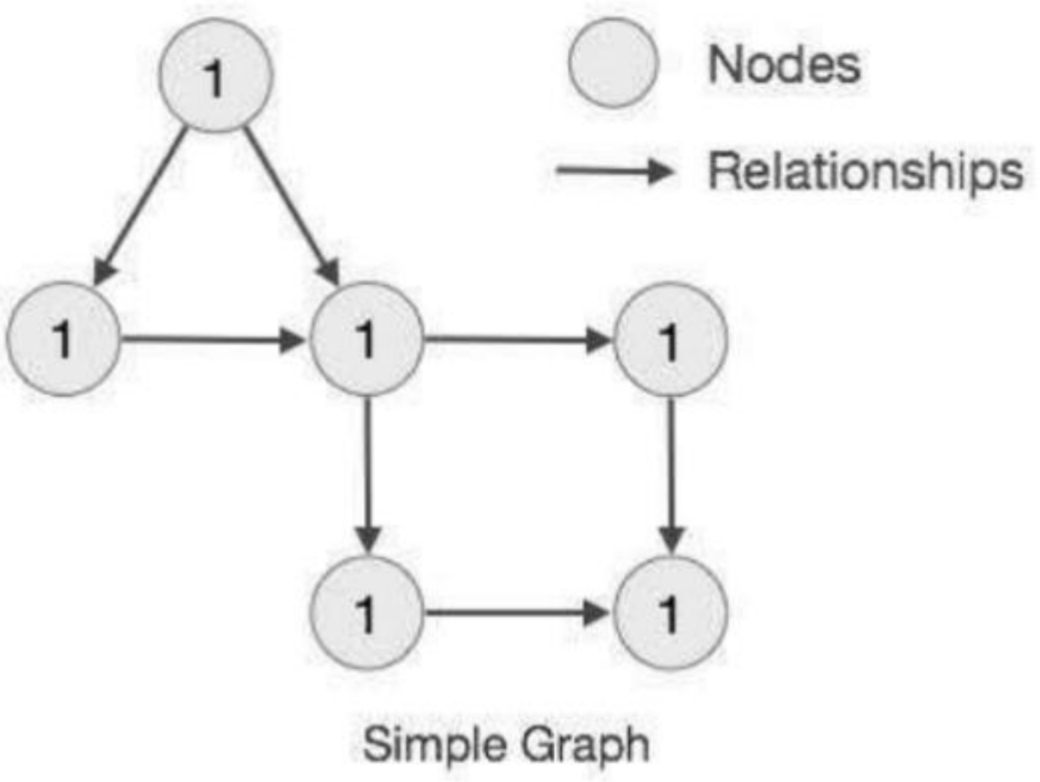


```
myDBReplicaSet:PRIMARY> db.article.update({"post_on_date.year" : {$gt: 2016},"article_text" : {$regex : ".*
house.*"}},{$addToSet : {"similar_stories" : {$each: ["How to build a house"]}}});
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

BONUS PROBLEM

Neo4j graph database system:

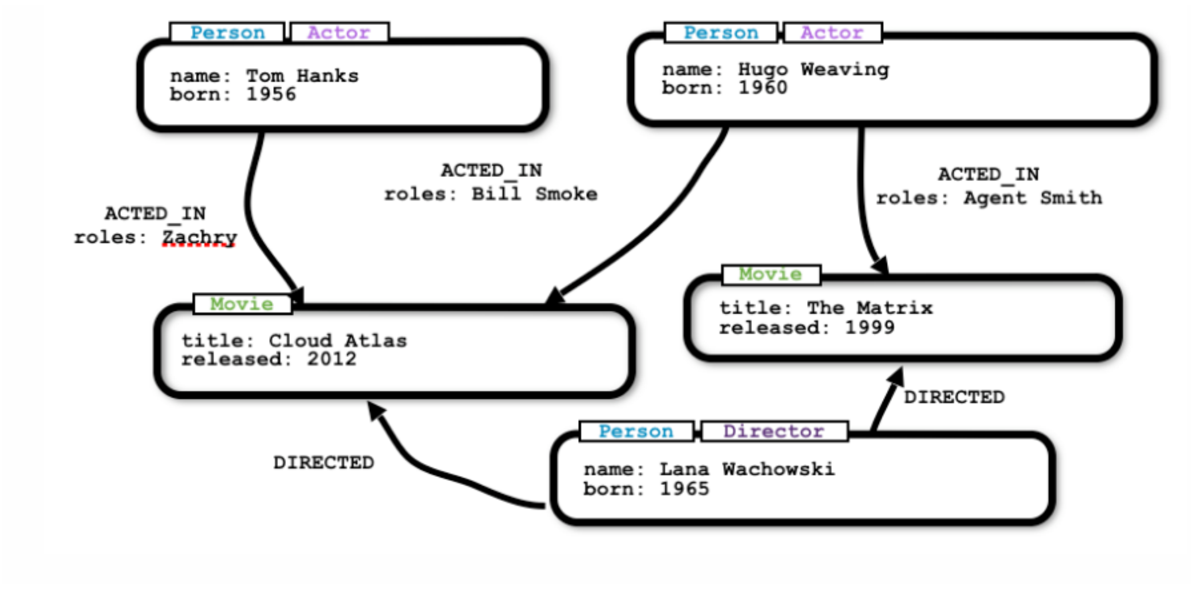
Neo4j graph database system data model is consisted by ***Nodes, Label, Relationships*** and ***Properties***.



From above, the ***nodes*** are represented by graph circles and relationships are represented by arrows. What is more, because using arrows to represent ***relationship***, it means ***relationship*** in this graph database are directional. And ***properties*** (key-value) are represented by ***node's*** data.

Graph Data Model = Whiteboard-Friendly

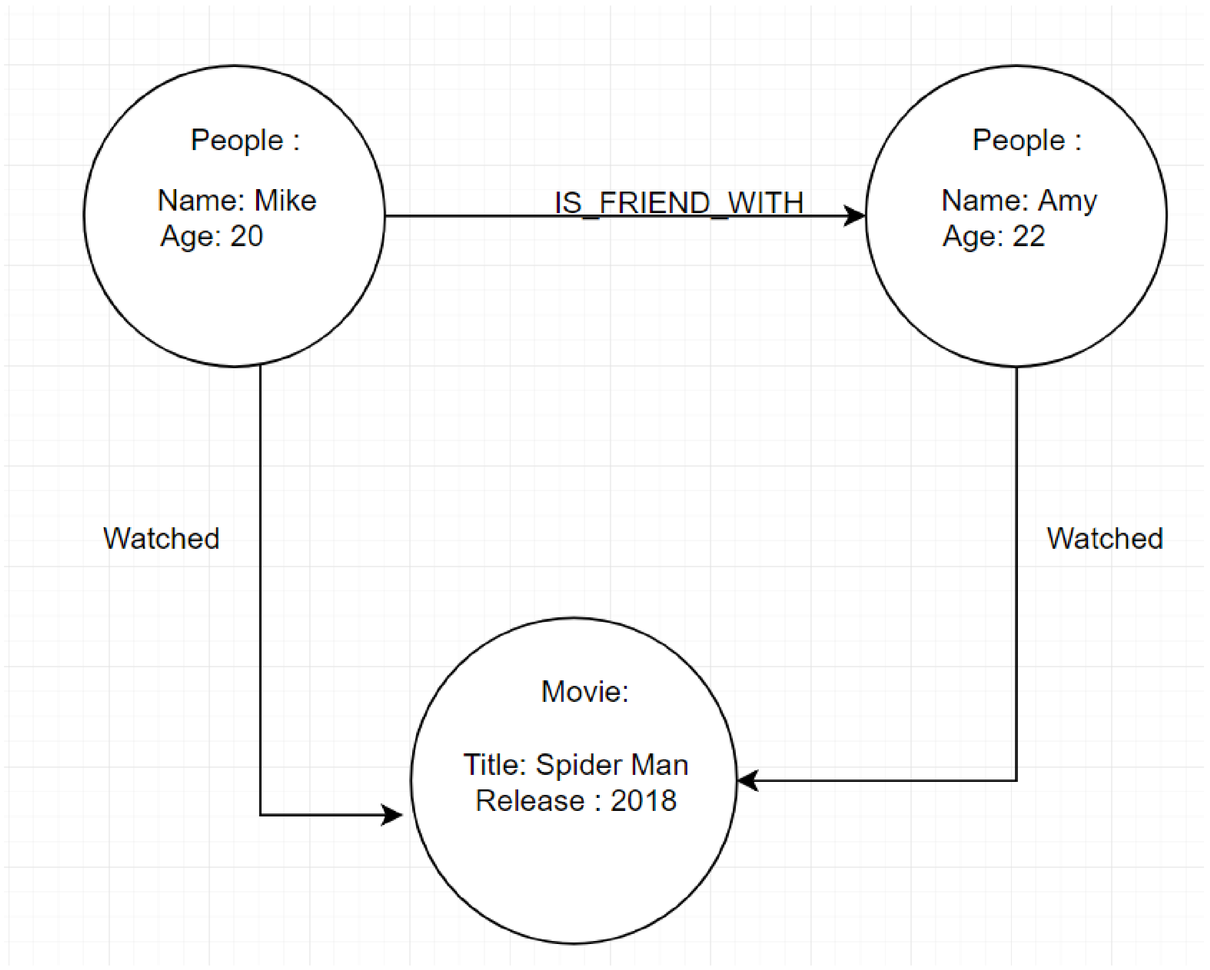
The Whiteboard-Friendly means that when people design data model, they usually create nodes, and then using arrow relationships to connect them instead of modifying the data model to fit a normalized table structure.



Above is the move graph data. ***Label*** is the type of nodes such as Person Actor. The database using directional edges to represents relationships. The set of edges are store in the systems. Once we need to change data, for example, (from above figure) the director Lana Wachowski direct a new movie called “666”, we just need to create a movie ”666” node and use arrow from Lana to “666” to connect them and put DIRECTED on the arrow.

USE CASE

Assume that Mike and Amy are friends, and they have watched movie “Spider Man”. In the graph database, the data model is shown below.



Once we want to add more people who has watched this people, only thing we need to is that create new people nodes, and put the arrow to movie. We also could use people node to represents more relationship, such as that new node people John is teaching of Amy, we could use arrow to represent this relationship. We don’t have to worry about one-one, many to many or one to many those relationship in schema. We also could use graph travel to list people that Mike may know in the database. And relational database may not use friend may know recommendation function. In relational database, if you want to delete a record whose attribute are foreign key of other record, you cannot delete this record. In graph database, you could delete the relationship and nodes to represents remove instead of modify data model.

Bonus Problems referenced from [1] Neo4j documentation

REFERENCE

[1] "Data Modeling Concepts and Techniques | Neo4j." *Neo4j Graph Database Platform*, neo4j.com/developer/guide-data-modeling/.