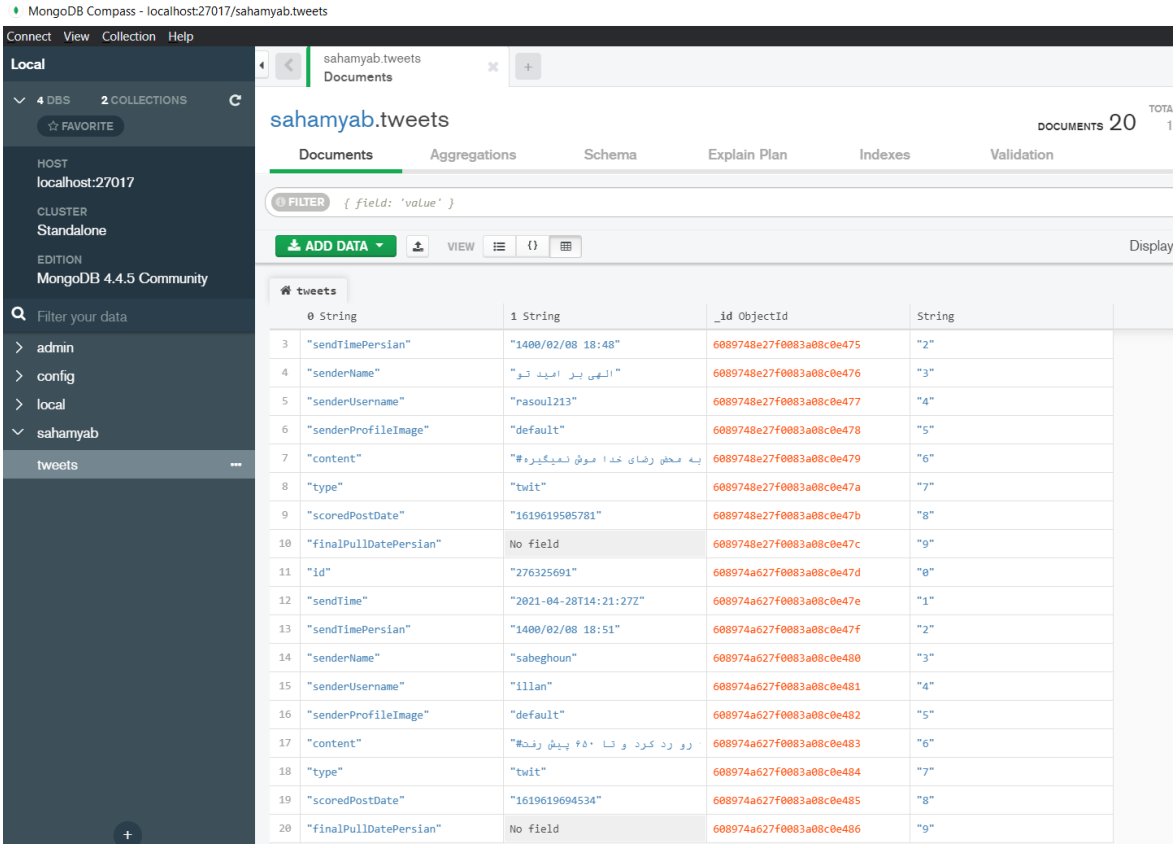
بخش اول

گام اول

نحوه ورود دستی داده ها در مانگو و فیلد های اضافه شده:

ابتدا به کمک کد پایتون صورت سوال، تعداد ۱۰ توئیت دانلود می‌شود و سپس آن‌ها را با فرمت csv. در هارد ذخیره می‌کنیم. سپس به کمک mongo compass داده ها را تکتک با فرمت csv. وارد می‌کنیم.



همان‌طور که می‌بینیم، فیلد objectId به داده ها توسط mongodb اضافه شده است.

درج هزار توییت:

در این قسمت ابتدا از طریق pymongo به کالکشن tweets در mongodb وصل می‌شویم و سپس با استفاده از request داده ها را دانلود کرده و در mongodb اظافه می‌کنیم. جهت اطمینان از درج هزار توئیت از دستور series_collection.count() استفاده

می‌کنیم که به ما تعداد توئیت‌های موجود در کالکشن tweets را می‌دهد. زمان صرف شده : 17478.95367026329 ثانیه

```
import pandas as pd
import requests
import json
from pymongo import MongoClient
import time

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
while series_collection.count() < 1000:
    # Getting last 10 twiits
    response = requests.get('https://www.sahamyab.com/guest/twiter/list?v=0.1',
    data = json.loads(response.text)
    twiit_10 = data['items']
    # insert data into mongodb
        ### cheching if twitt is not in collection
    for element in twiit_10 :
        if series_collection.find_one(element) == None:
            result = series_collection.insert_one(element)

    time.sleep(60 - time.time() % 60)

end_time = time.time()
delta_time = end_time - start_time

#print(series_collection.count())


```

گام دوم

در ابتدا سعی کردم به کمک pymongo این قسمت را پیاده‌سازی کنم. اما در قسمت regexFindAll$ مشکل داشتم و هشتگ‌هایی که در ابتدای جمله قرار داشتند را نمی‌توانستم به عنوان هشتگ استخراج کنم. به همین دلیل با توجه گفته‌ی جناب بنایی، در این قسمت از امکانات پایتون استفاده کردم.

عکس زیر کد پیاده‌سازی به کمک pymongo است.

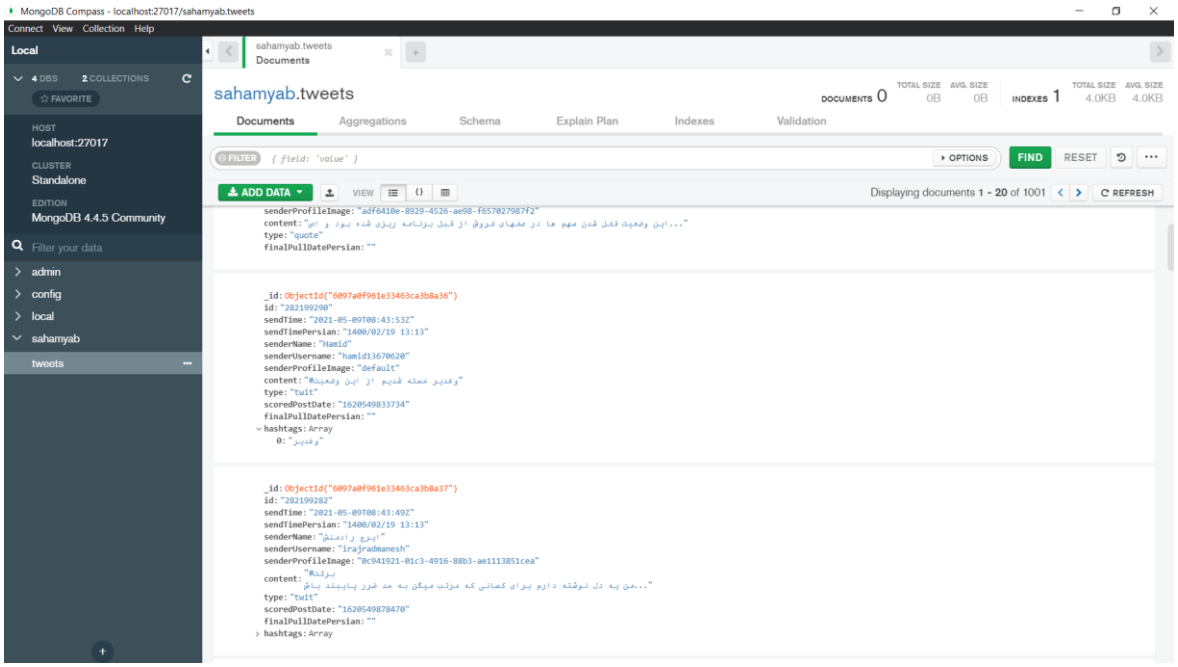
```
##################### Implementing in pymongo ###################
db.tweets.update_many(
  {'content':{'$regex': '.*#.*'}},
  [{'$set':{'hashtags': {'$regexFindAll': { 'input': '$content' , 'regex': '#.*'}}}
  }])


```

کد نوشته شده در پایتون برای این بخش:

زمان اجرا: 0.7917428016662598 ثانیه

```python
from pymongo import MongoClient
import time
import re

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
for document in series_collection.find({'content':{'$regex':'.*#.*'}}):
    hashtags_sharp = re.findall('#\w+', document['content'])
    hashtags = [o.split('#')[1] for o in hashtags_sharp]  #removing shashtag si
    new_dict = {"hashtags": hashtags }
    db.tweets.update(document, {"$set": new_dict })

end_time = time.time()
delta_time = end_time - start_time


```

پس از اجرای کد فوق، در mongo compass فیلد هشتگ اضافه می‌شود. عکس زیر، فیلد اظافه شده به توئیت‌های شامل هشتگ را نشان می‌دهد.



البته قبل از اجرای کد فوق، حروف عربی با حروف فارسی به کمک کد زیر جایگزین شد.

game 1.py | game 2.py | game2_arabic_replacement.py

```python
from pymongo import MongoClient
import re
import time


client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
for document in series_collection.find({'content':{'$regex':'.*ي*.|*.ك.*'}}):
    new_dict = document.copy()
    new_dict['content'] = re.sub('ى' ,'ي', new_dict['content'])
    new_dict['content'] = re.sub('ک' ,'ك', new_dict['content'])
    db.tweets.update(document, new_dict)

end_time = time.time()
delta_time = end_time - start_time
print(delta_time)
```

گام سوم

بخش ۱

دستور نوشته شده:

زمان اجرا: خیلی سریع بود و تقریبا صفر ثانیه

game3_1.py

```python
from pymongo import MongoClient
import time
import pprint

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
res = series_collection.find({
    '$and':[
            {'mediaContentType':'image/jpeg'}, {'parentId':{ '$exists': True }
        ]}, {"senderName": 1})


end_time = time.time()
delta_time = end_time - start_time
print(delta_time)

lis=[]
for i in res:
    lis.append(i['senderName'])
    pprint.pprint(i)
```

خروجی:

```
{'_id': ObjectId('6097a0f961e33463ca3b8a39'), 'senderName': 'فرشاد'}
{'_id': ObjectId('6097a0f961e33463ca3b8a3c'), 'senderName': 'محب'}
{'_id': ObjectId('6097a3dc61e33463ca3b8a7e'), 'senderName': 'مهدی'}
{'_id': ObjectId('6097a49061e33463ca3b8a8d'), 'senderName': 'مهدی'}
{'_id': ObjectId('6097abd661e33463ca3b8b2c'), 'senderName': 'tazekar'}
{'_id': ObjectId('6097b6a961e33463ca3b8bc6'), 'senderName': 'Wave Rider'}
{'_id': ObjectId('6097b80761e33463ca3b8bd8'), 'senderName': 'حرفه ای😁'}
{'_id': ObjectId('6097b87d61e33463ca3b8be0'), 'senderName': 'وحید'}
{'_id': ObjectId('6097b9e561e33463ca3b8bf5'), 'senderName': 'وحید'}
{'_id': ObjectId('6097b9e661e33463ca3b8bf7'), 'senderName': 'وحید'}
{'_id': ObjectId('6097bf0d61e33463ca3b8c3c'), 'senderName': 'مید پوینت'}
{'_id': ObjectId('6097c07561e33463ca3b8c54'), 'senderName': 'fati'}
{'_id': ObjectId('6097cc2d61e33463ca3b8cf3'), 'senderName': 'Amir ahj'}
{'_id': ObjectId('6097d0dd61e33463ca3b8d39'), 'senderName': 'Masoud.RE'}
{'_id': ObjectId('6097dc5a61e33463ca3b8dc6'), 'senderName': 'حمید'}
{'_id': ObjectId('6097e18261e33463ca3b8df1'), 'senderName': 'Vahid2727'}
{'_id': ObjectId('6097e50661e33463ca3b8e1c'), 'senderName': 'علیرضا محمدی'}
```

بخش ۲

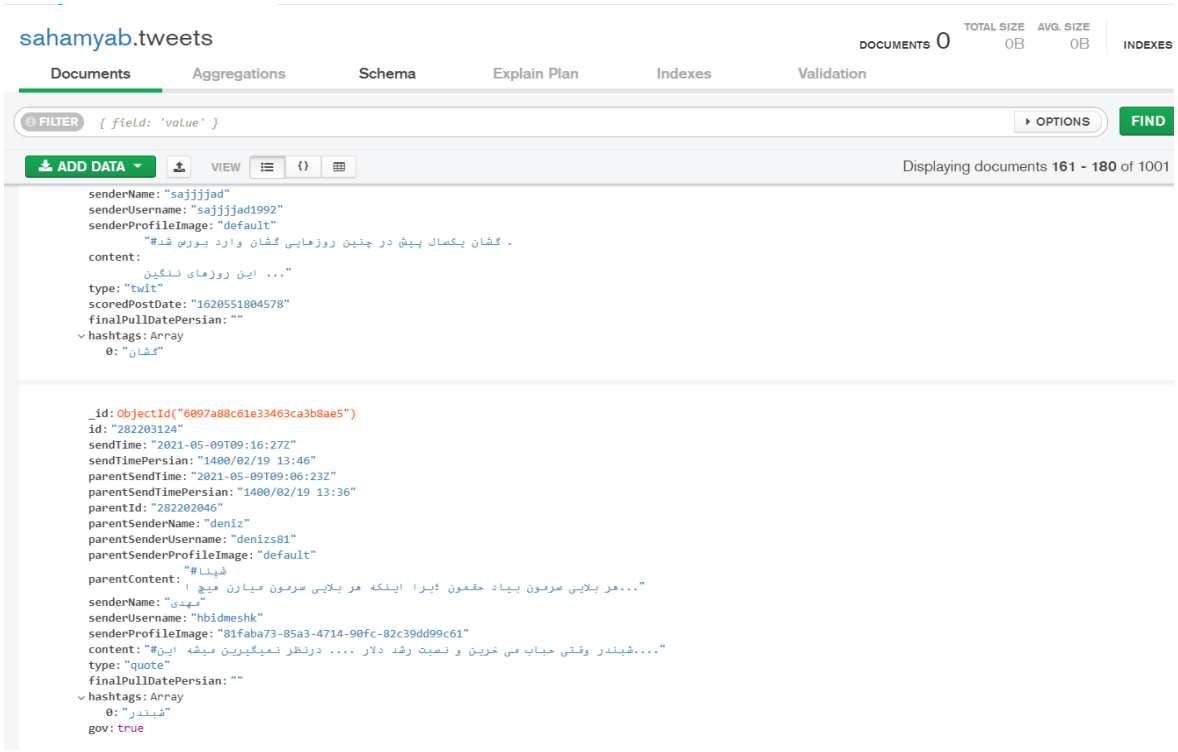
برای این کار، از update_many استفاده می‌کنیم. همینطور چک میکنیم که آیا یکی از المان‌های شبندر، شستا، فولاد در ارائه هشتگ وجود دارد یا خیر. زمان اجرا : 0.033175706863 ثانیه

کد نوشته شده:

game3_2.py game3_3.py

```
1 from pymongo import MongoClient
2 import time
3
4 client = MongoClient('localhost', 27017)
5 db = client['sahamyab']
6 series_collection = db['tweets']
7
8 start_time = time.time()
9 series_collection.update_many(
10              {'hashtags':{'$in': ['شبندر' ,'شستا' ,'فولاد'] }},
11              {'$set':{'gov': True }})
12
13 end_time = time.time()
14 delta_time = end_time - start_time
15 print(delta_time)
16
```

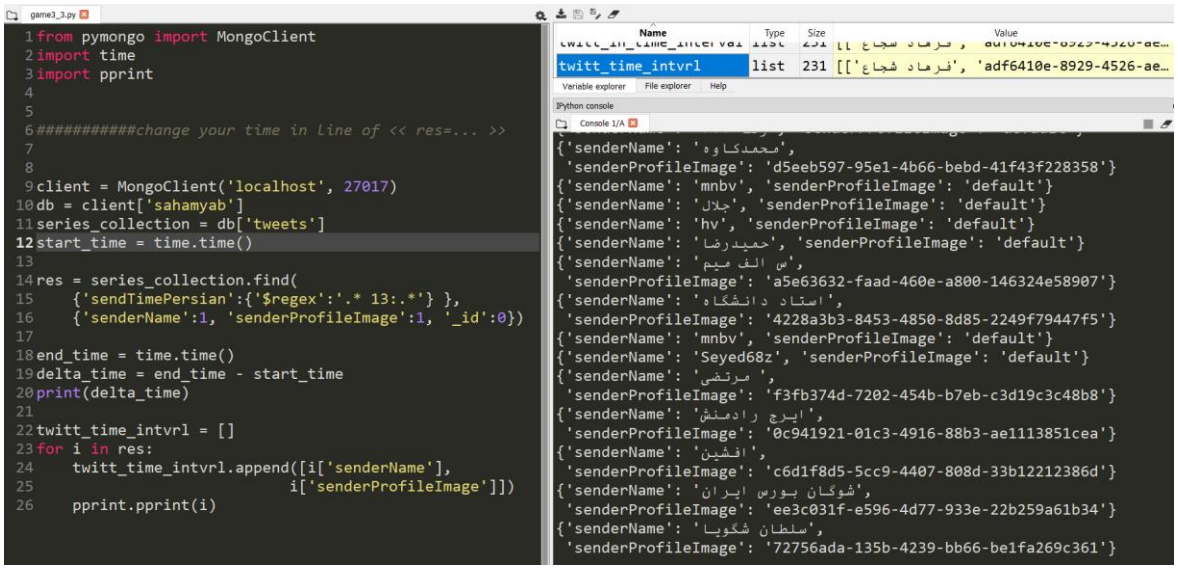
همان‌طور که می‌بینیم فیلد gov با مقدار true به هشتگ‌های حاوی شبندر، شستا، فولاد اظافه شده است



بخش ۳

افرادی که در بازه‌ی ۱۳ الی ۱۴ توئیت زده بودند را پیدا کردم. تعداد افراد ۲۳۱ نفر بود. زمان اجرا تقریبا صفر ثانیه. کد نوشته شده و قسمتی از خروجی، به شکل زیر است.

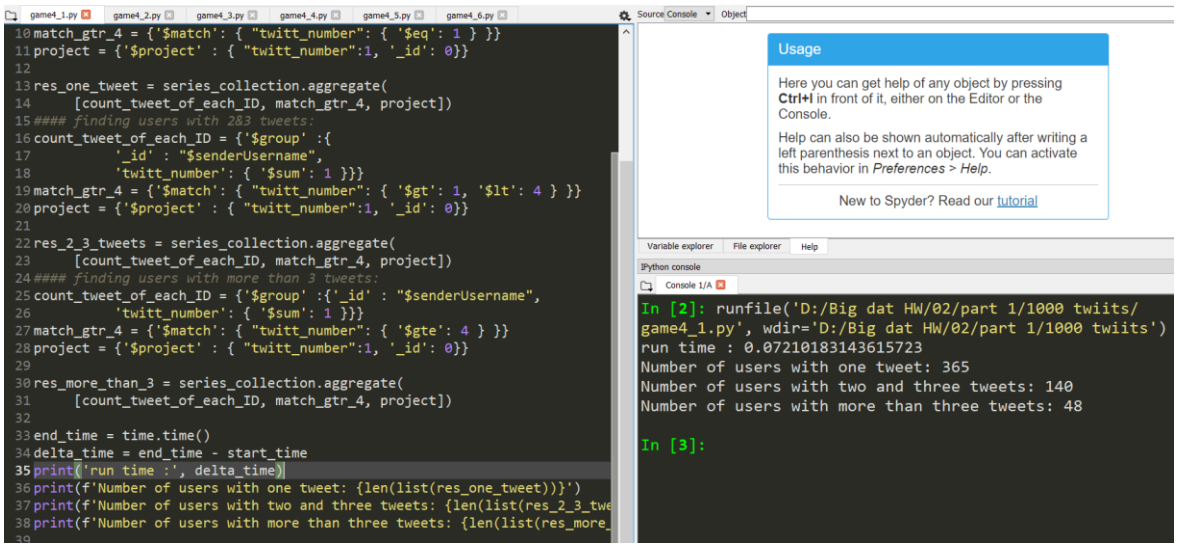
چون تعداد افراد زیاد است، یکسری از آنها را اینجا می‌آورم.



گام چهارم

بخش ۱

دستور نوشته شده، خروجی و زمان اجرا در عکس زیر قابل مشاهده است.

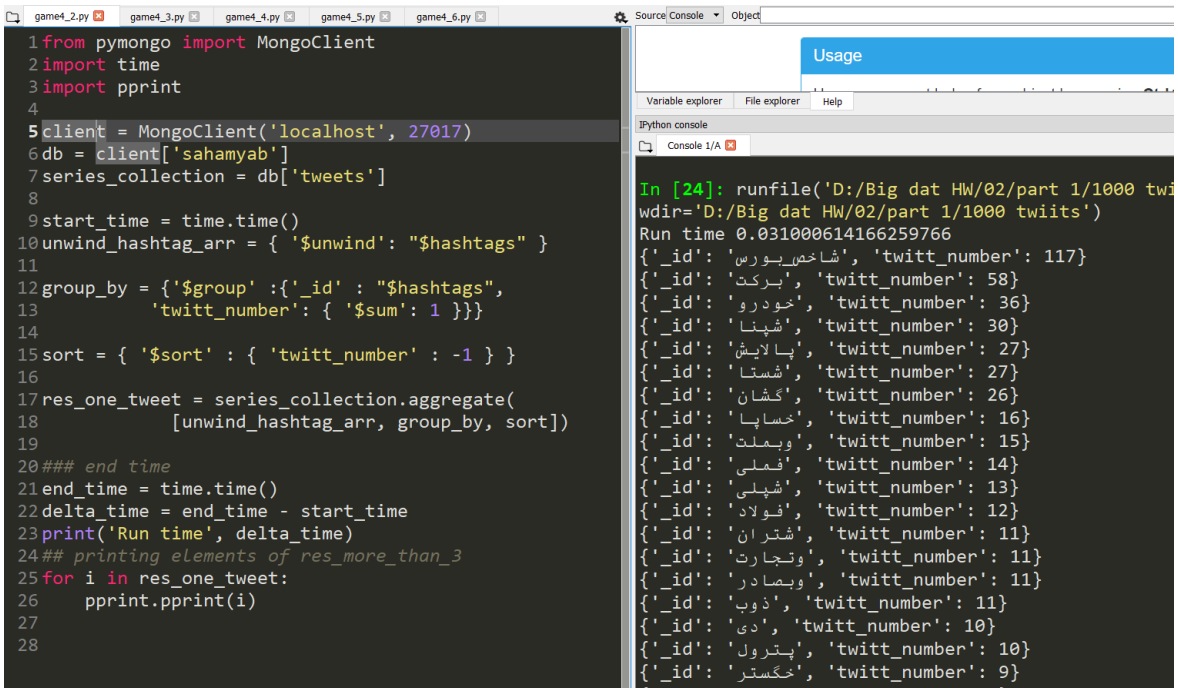


```
match_gtr_4 = {'$match': { "twitt_number": { '$eq': 1 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_one_tweet = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])
#### finding users with 2&3 tweets:
count_tweet_of_each_ID = {'$group' :{
        '_id' : "$senderUsername",
        'twitt_number': { '$sum': 1 }}}
match_gtr_4 = {'$match': { "twitt_number": { '$gt': 1, '$lt': 4 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_2_3_tweets = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])
#### finding users with more than 3 tweets:
count_tweet_of_each_ID = {'$group' :{'_id' : "$senderUsername",
        'twitt_number': { '$sum': 1 }}}
match_gtr_4 = {'$match': { "twitt_number": { '$gte': 4 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_more_than_3 = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])

end_time = time.time()
delta_time = end_time - start_time
print('run time :', delta_time)
print(f'Number of users with one tweet: {len(list(res_one_tweet))}')
print(f'Number of users with two and three tweets: {len(list(res_2_3_twe
print(f'Number of users with more than three tweets: {len(list(res_more_
```

```
In [2]: runfile('D:/Big dat HW/02/part 1/1000 twiits/
game4_1.py', wdir='D:/Big dat HW/02/part 1/1000 twiits')
run time : 0.07210183143615723
Number of users with one tweet: 365
Number of users with two and three tweets: 140
Number of users with more than three tweets: 48

In [3]:
```

بخش ۲

دستور نوشته شده، بخشی از خروجی و زمان اجرا در عکس زیر قابل مشاهده است.



```
from pymongo import MongoClient
import time
import pprint

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
unwind_hashtag_arr = { '$unwind': "$hashtags" }

group_by = {'$group' :{'_id' : "$hashtags",
          'twitt_number': { '$sum': 1 }}}

sort = { '$sort' : { 'twitt_number' : -1 } }

res_one_tweet = series_collection.aggregate(
            [unwind_hashtag_arr, group_by, sort])

### end time
end_time = time.time()
delta_time = end_time - start_time
print('Run time', delta_time)
## printing elements of res_more_than_3
for i in res_one_tweet:
    pprint.pprint(i)
```

```
In [24]: runfile('D:/Big dat HW/02/part 1/1000 twi
wdir='D:/Big dat HW/02/part 1/1000 twiits')
Run time 0.031000614166259766
{'_id': 'شاخص‌بورس', 'twitt_number': 117}
{'_id': 'برکت', 'twitt_number': 58}
{'_id': 'خودرو', 'twitt_number': 36}
{'_id': 'شپنا', 'twitt_number': 30}
{'_id': 'پالایش', 'twitt_number': 27}
{'_id': 'شستا', 'twitt_number': 27}
{'_id': 'گشان', 'twitt_number': 26}
{'_id': 'خساپا', 'twitt_number': 16}
{'_id': 'وبملت', 'twitt_number': 15}
{'_id': 'فملی', 'twitt_number': 14}
{'_id': 'شپلی', 'twitt_number': 13}
{'_id': 'فولاد', 'twitt_number': 12}
{'_id': 'شتران', 'twitt_number': 11}
{'_id': 'وتجارت', 'twitt_number': 11}
{'_id': 'وبصادر', 'twitt_number': 11}
{'_id': 'ذوب', 'twitt_number': 11}
{'_id': 'دی', 'twitt_number': 10}
{'_id': 'پترول', 'twitt_number': 10}
{'_id': 'خگستر', 'twitt_number': 9}
```

بـخش ۳

run time: 0.0319676399230957

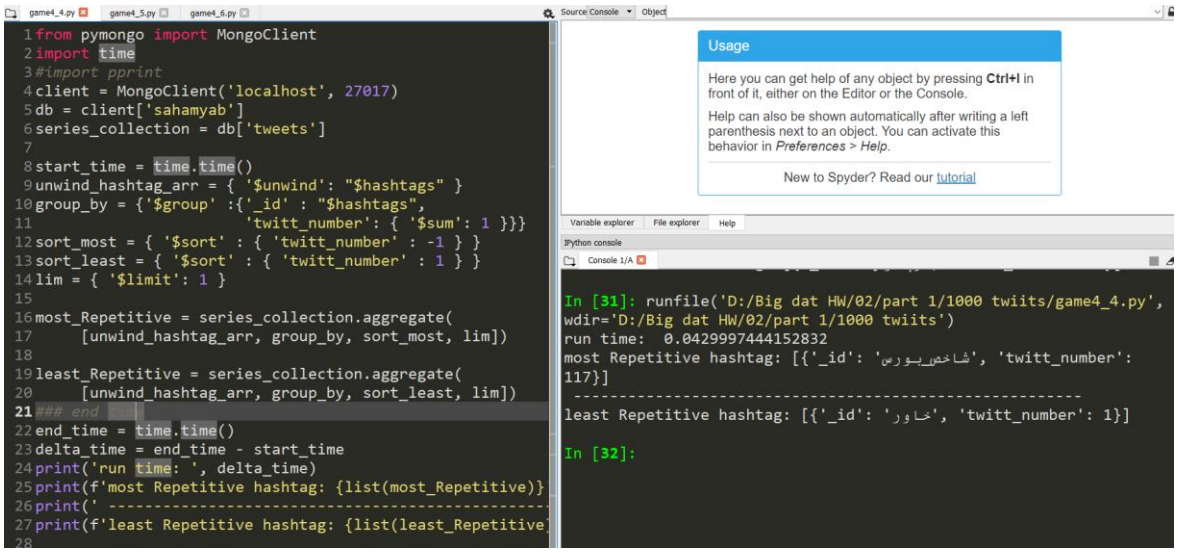
کـد نـوشـتـه شـده :

game4_3.py | game4_4.py | game4_5.py | game4_6.py

```python
from pymongo import MongoClient
import time

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
series_collection.update_many(
                {'parentId':True},
                {'$unset':{'type': '' }})

end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
```

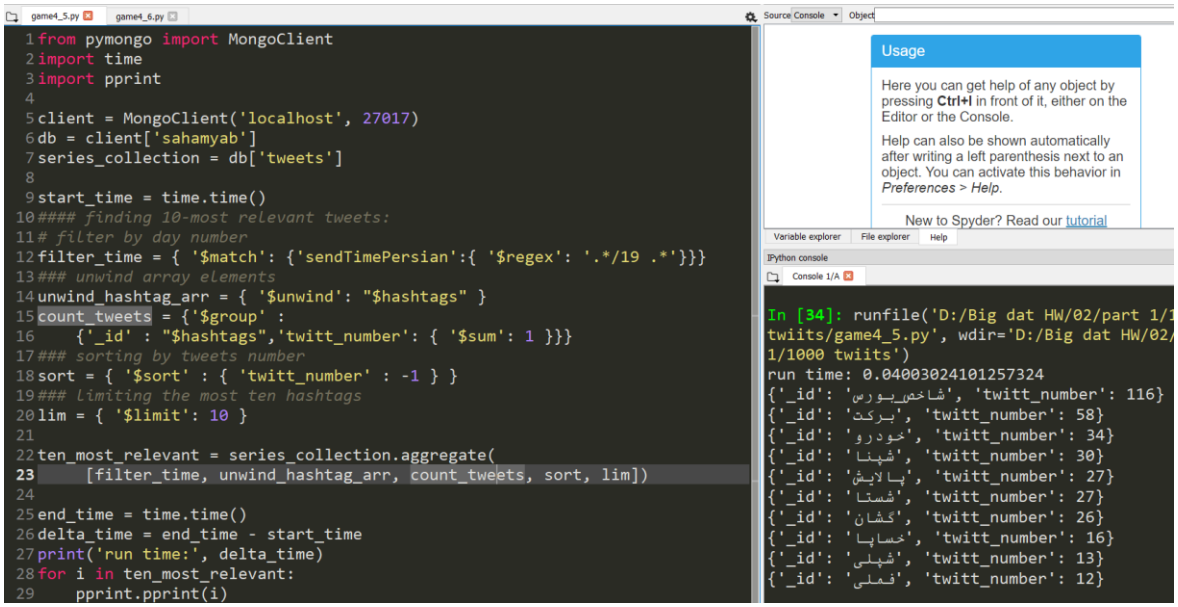
بـخش ۴

دستـور نـوشـتـه شـده، خـروجـی و زمـان اجـرا در عکس زیـر قـابـل مـشـا هـده است.



بخش ۵

دستور نوشته شده، خروجی و زمان اجرا در عکس زیر قابل مشاهده است.

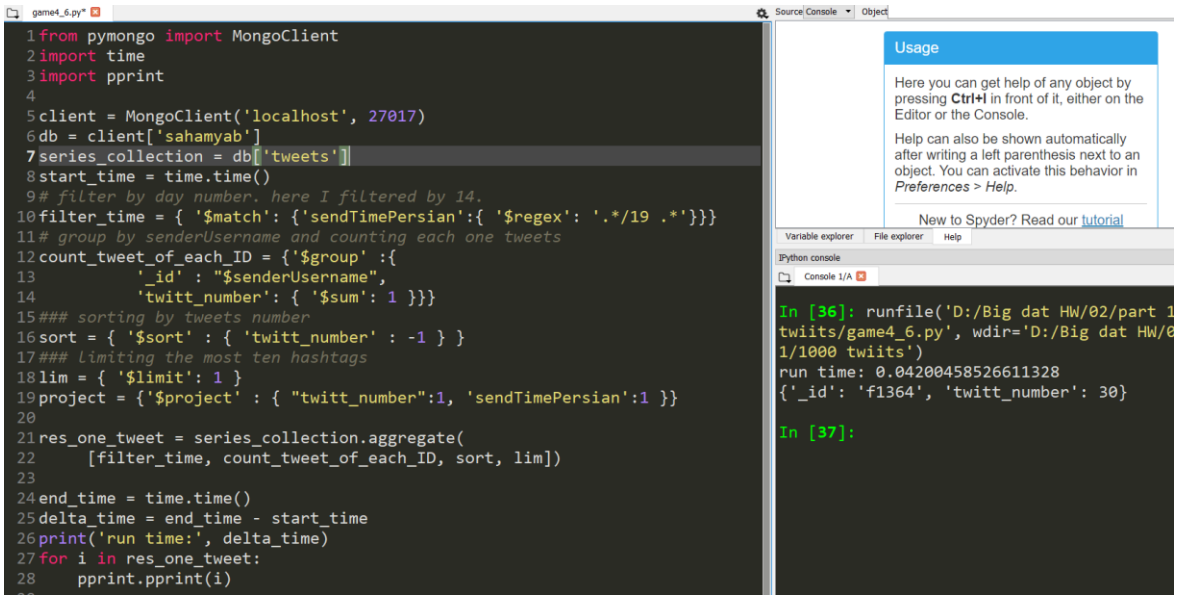


```
from pymongo import MongoClient
import time
import pprint

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

start_time = time.time()
#### finding 10-most relevant tweets:
# filter by day number
filter_time = { '$match': {'sendTimePersian':{ '$regex': '.*/19 .*'}}}
### unwind array elements
unwind_hashtag_arr = { '$unwind': "$hashtags" }
count_tweets = {'$group' :
    {'_id' : "$hashtags",'twitt_number': { '$sum': 1 }}}
### sorting by tweets number
sort = { '$sort' : { 'twitt_number' : -1 } }
### limiting the most ten hashtags
lim = { '$limit': 10 }

ten_most_relevant = series_collection.aggregate(
    [filter_time, unwind_hashtag_arr, count_tweets, sort, lim])

end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
for i in ten_most_relevant:
    pprint.pprint(i)
```

```
In [34]: runfile('D:/Big dat HW/02/part 1/1
twiits/game4_5.py', wdir='D:/Big dat HW/02/
1/1000 twiits')
run time: 0.04003024101257324
{'_id': 'شاخص_بورس', 'twitt_number': 116}
{'_id': 'برکت', 'twitt_number': 58}
{'_id': 'خودرو', 'twitt_number': 34}
{'_id': 'شپنا', 'twitt_number': 30}
{'_id': 'پالایش', 'twitt_number': 27}
{'_id': 'فستا', 'twitt_number': 27}
{'_id': 'گشان', 'twitt_number': 26}
{'_id': 'خساپا', 'twitt_number': 16}
{'_id': 'شپلی', 'twitt_number': 13}
{'_id': 'فملی', 'twitt_number': 12}
```

بخش ۶

دستور نوشته شده، خروجی و زمان اجرا در عکس زیر قابل مشاهده است.

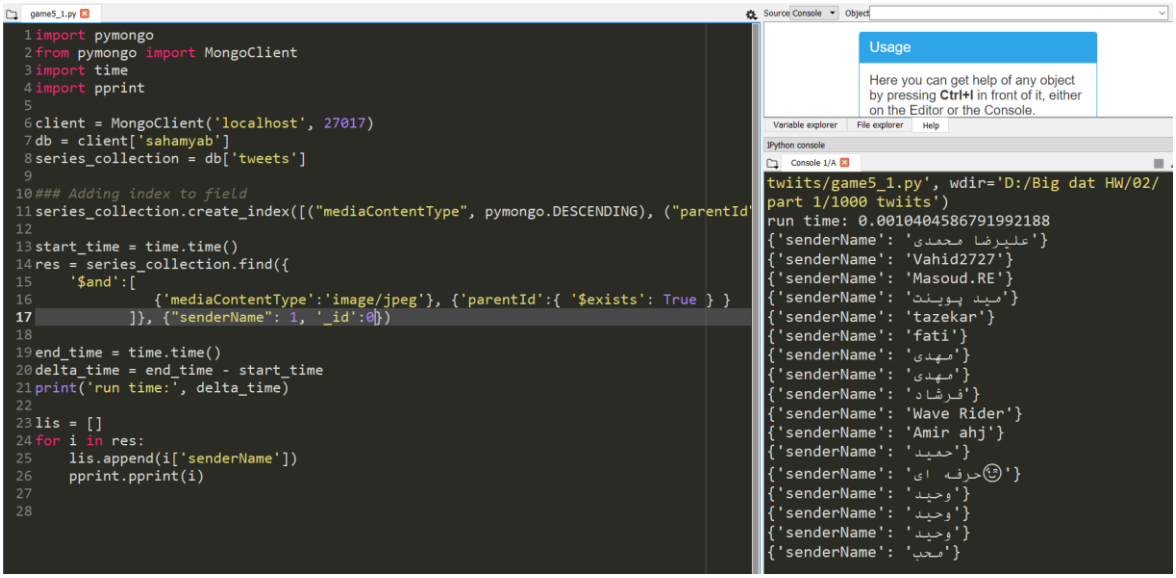


```
from pymongo import MongoClient
import time
import pprint

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']
start_time = time.time()
# filter by day number. here I filtered by 14.
filter_time = { '$match': {'sendTimePersian':{ '$regex': '.*/19 .*'}}}
# group by senderUsername and counting each one tweets
count_tweet_of_each_ID = {'$group' :{
          '_id' : "$senderUsername",
         'twitt_number': { '$sum': 1 }}}
### sorting by tweets number
sort = { '$sort' : { 'twitt_number' : -1 } }
### limiting the most ten hashtags
lim = { '$limit': 1 }
project = {'$project' : { "twitt_number":1, 'sendTimePersian':1 }}

res_one_tweet = series_collection.aggregate(
    [filter_time, count_tweet_of_each_ID, sort, lim])

end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
for i in res_one_tweet:
    pprint.pprint(i)
```

```
In [36]: runfile('D:/Big dat HW/02/part 1
twiits/game4_6.py', wdir='D:/Big dat HW/0
1/1000 twiits')
run time: 0.04200458526611328
{'_id': 'f1364', 'twitt_number': 30}

In [37]:
```

گام پنجم

گام سوم، سوال ۱:



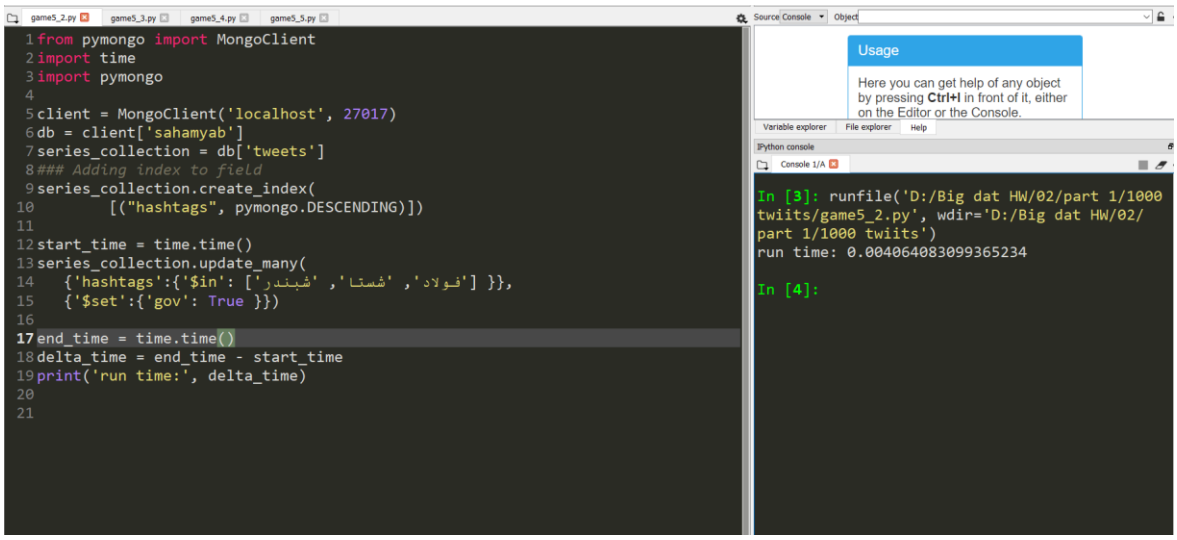
```
import pymongo
from pymongo import MongoClient
import time
import pprint

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

### Adding index to field
series_collection.create_index([("mediaContentType", pymongo.DESCENDING), ("parentId"
start_time = time.time()
res = series_collection.find({
    '$and':[
            {'mediaContentType':'image/jpeg'}, {'parentId':{ '$exists': True } }
          ]}, {"senderName": 1, '_id':0})
end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)

lis = []
for i in res:
    lis.append(i['senderName'])
    pprint.pprint(i)
```

```
twiits/game5_1.py', wdir='D:/Big dat HW/02/
part 1/1000 twiits')
run time: 0.0010404586791992188
{'senderName': 'علیرضا محمدی'}
{'senderName': 'Vahid2727'}
{'senderName': 'Masoud.RE'}
{'senderName': 'مید پوینت'}
{'senderName': 'tazekar'}
{'senderName': 'fati'}
{'senderName': 'مهدی'}
{'senderName': 'مهدی'}
{'senderName': 'فرشاد'}
{'senderName': 'Wave Rider'}
{'senderName': 'Amir ahj'}
{'senderName': 'حمید'}
{'senderName': 'حرفه ای😉'}
{'senderName': 'وحید'}
{'senderName': 'وحید'}
{'senderName': 'وحید'}
{'senderName': 'محب'}
```

run time : بدون استفاده از indexing

run time : 0.00104 با استفاده از indexing

گام سوم، سوال ۲:



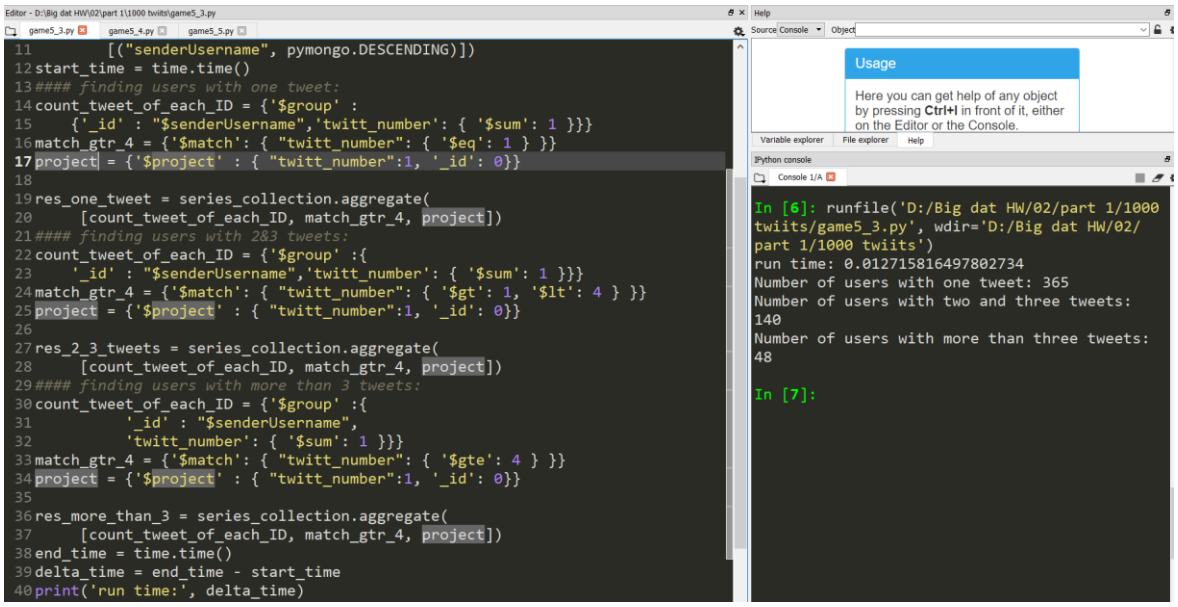
```
from pymongo import MongoClient
import time
import pymongo

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']
### Adding index to field
series_collection.create_index(
        [("hashtags", pymongo.DESCENDING)])

start_time = time.time()
series_collection.update_many(
   {'hashtags':{'$in': ['شبندر' ,'شستا' ,'فولاد'] }},
   {'$set':{'gov': True }})

end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
```

```
In [3]: runfile('D:/Big dat HW/02/part 1/1000
twiits/game5_2.py', wdir='D:/Big dat HW/02/
part 1/1000 twiits')
run time: 0.004064083099365234

In [4]:
```

run time : 0.03317 بدون استفاده از indexing

run time : 0.00406 با استفاده از indexing

گام چهارم، سوال ۱:



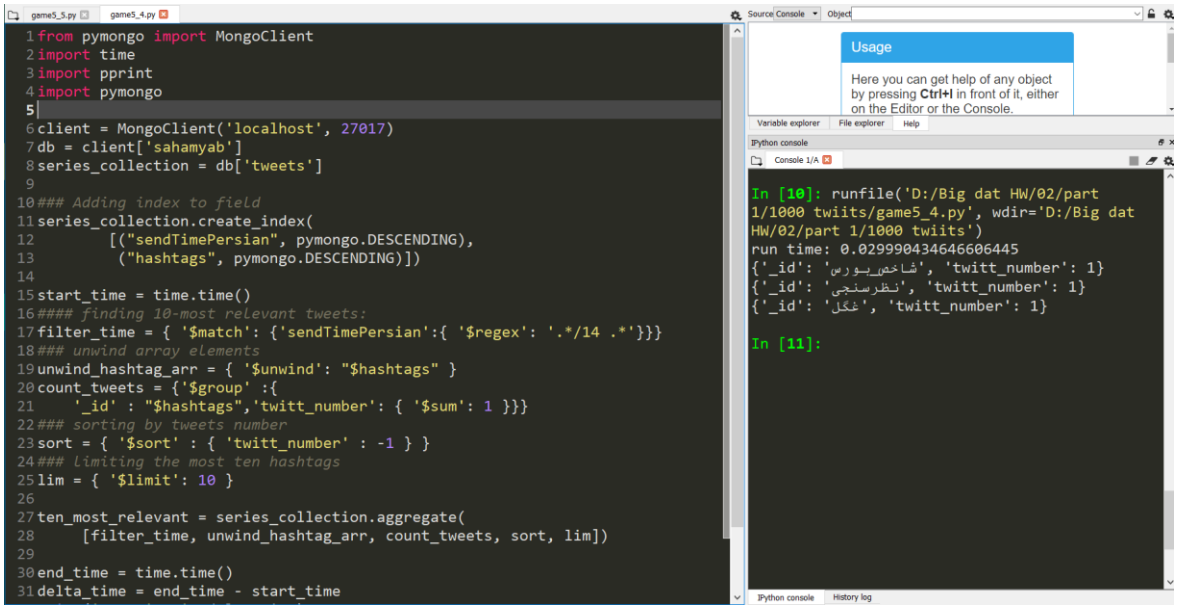
```
        [("senderUsername", pymongo.DESCENDING)])
start_time = time.time()
#### finding users with one tweet:
count_tweet_of_each_ID = {'$group' :
    {'_id' : "$senderUsername",'twitt_number': { '$sum': 1 }}}
match_gtr_4 = {'$match': { "twitt_number": { '$eq': 1 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_one_tweet = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])
#### finding users with 2&3 tweets:
count_tweet_of_each_ID = {'$group' :{
    '_id' : "$senderUsername",'twitt_number': { '$sum': 1 }}}
match_gtr_4 = {'$match': { "twitt_number": { '$gt': 1, '$lt': 4 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_2_3_tweets = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])
#### finding users with more than 3 tweets:
count_tweet_of_each_ID = {'$group' :{
          '_id' : "$senderUsername",
         'twitt_number': { '$sum': 1 }}}
match_gtr_4 = {'$match': { "twitt_number": { '$gte': 4 } }}
project = {'$project' : { "twitt_number":1, '_id': 0}}

res_more_than_3 = series_collection.aggregate(
    [count_tweet_of_each_ID, match_gtr_4, project])
end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
```

```
In [6]: runfile('D:/Big dat HW/02/part 1/1000
twiits/game5_3.py', wdir='D:/Big dat HW/02/
part 1/1000 twiits')
run time: 0.012715816497802734
Number of users with one tweet: 365
Number of users with two and three tweets:
140
Number of users with more than three tweets:
48

In [7]:
```

indexing از استفاده بدون run time : 0.0721

indexing از استفاده با run time : 0.012715

گام چهارم، سوال ۵:



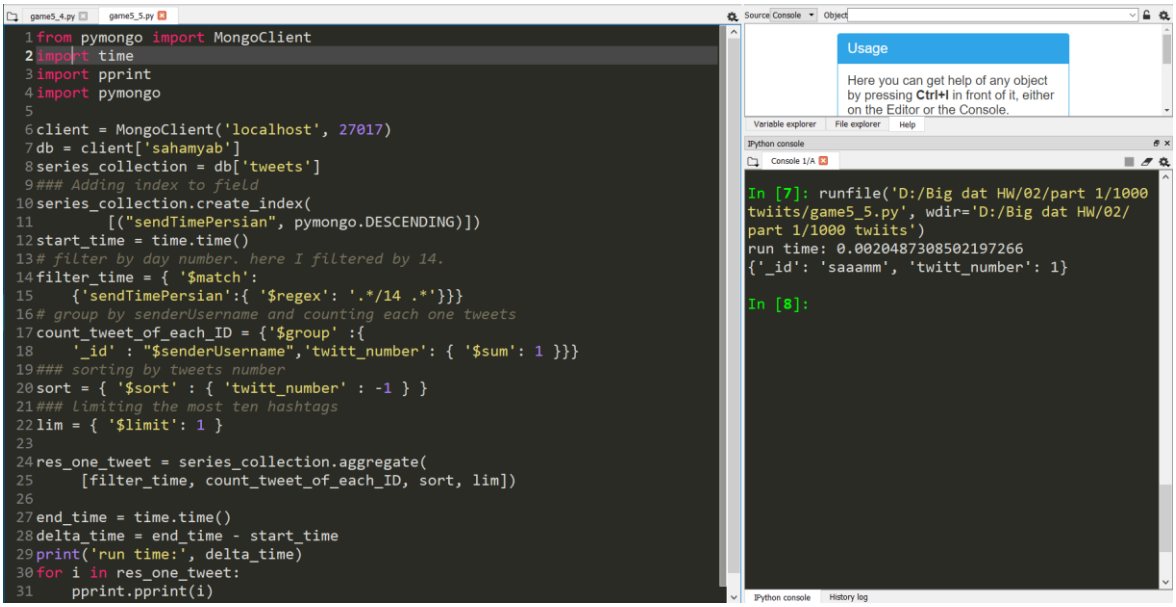
```
from pymongo import MongoClient
import time
import pprint
import pymongo

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']

### Adding index to field
series_collection.create_index(
        [("sendTimePersian", pymongo.DESCENDING),
         ("hashtags", pymongo.DESCENDING)])

start_time = time.time()
#### finding 10-most relevant tweets:
filter_time = { '$match': {'sendTimePersian':{ '$regex': '.*/14 .*'}}}
### unwind array elements
unwind_hashtag_arr = { '$unwind': "$hashtags" }
count_tweets = {'$group' :{
    '_id' : "$hashtags",'twitt_number': { '$sum': 1 }}}
### sorting by tweets number
sort = { '$sort' : { 'twitt_number' : -1 } }
### limiting the most ten hashtags
lim = { '$limit': 10 }

ten_most_relevant = series_collection.aggregate(
    [filter_time, unwind_hashtag_arr, count_tweets, sort, lim])

end_time = time.time()
delta_time = end_time - start_time
```

```
In [10]: runfile('D:/Big dat HW/02/part
1/1000 twiits/game5_4.py', wdir='D:/Big dat
HW/02/part 1/1000 twiits')
run time: 0.029990434646606445
{'_id': 'شاخص بورس', 'twitt_number': 1}
{'_id': 'نظرسنجی', 'twitt_number': 1}
{'_id': 'غگل', 'twitt_number': 1}

In [11]:
```

indexing از استفاده بدون run time : 0.040003

indexing از استفاده با run time : 0.02999

گام چهارم، سوال ۶:



```
from pymongo import MongoClient
import time
import pprint
import pymongo

client = MongoClient('localhost', 27017)
db = client['sahamyab']
series_collection = db['tweets']
### Adding index to field
series_collection.create_index(
        [("sendTimePersian", pymongo.DESCENDING)])
start_time = time.time()
# filter by day number. here I filtered by 14.
filter_time = { '$match':
    {'sendTimePersian':{ '$regex': '.*/14 .*'}}}
# group by senderUsername and counting each one tweets
count_tweet_of_each_ID = {'$group' :{
    '_id' : "$senderUsername",'twitt_number': { '$sum': 1 }}}
### sorting by tweets number
sort = { '$sort' : { 'twitt_number' : -1 } }
### limiting the most ten hashtags
lim = { '$limit': 1 }

res_one_tweet = series_collection.aggregate(
     [filter_time, count_tweet_of_each_ID, sort, lim])

end_time = time.time()
delta_time = end_time - start_time
print('run time:', delta_time)
for i in res_one_tweet:
    pprint.pprint(i)
```

```
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

In [7]: runfile('D:/Big dat HW/02/part 1/1000 twiits/game5_5.py', wdir='D:/Big dat HW/02/part 1/1000 twiits')
run time: 0.0020487308502197266
{'_id': 'saaamm', 'twitt_number': 1}

In [8]:
```

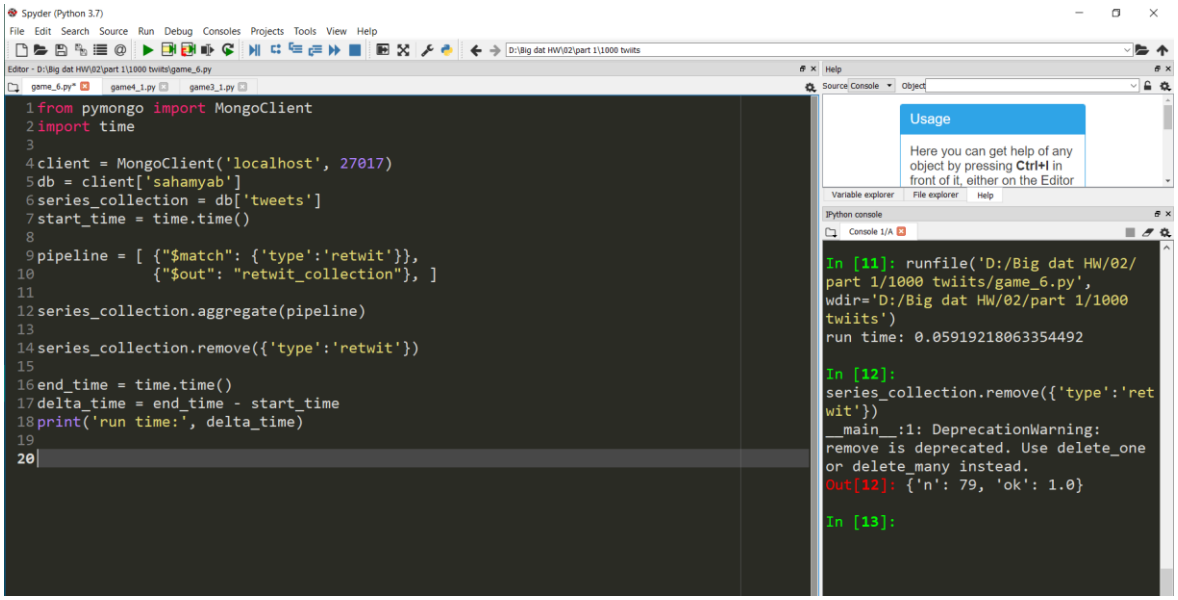
run time : 0.042 بدون استفاده از indexing

run time : 0.00204 با استفاده از indexing

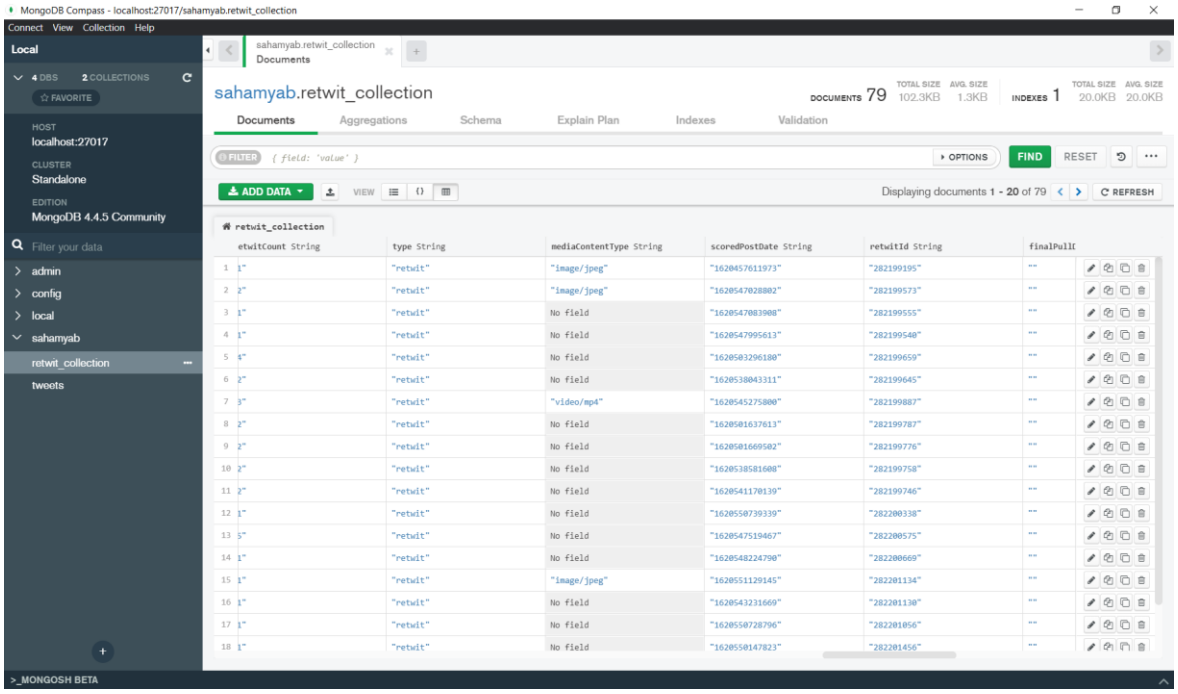
همان طور که مشاهده می‌کنیم، در همه‌ی موارد زمان اجرا حدود 0.1 برابر شده است و این مزیت indexing است.

تفکیک توئیت‌ها و ریتوئیت‌ها

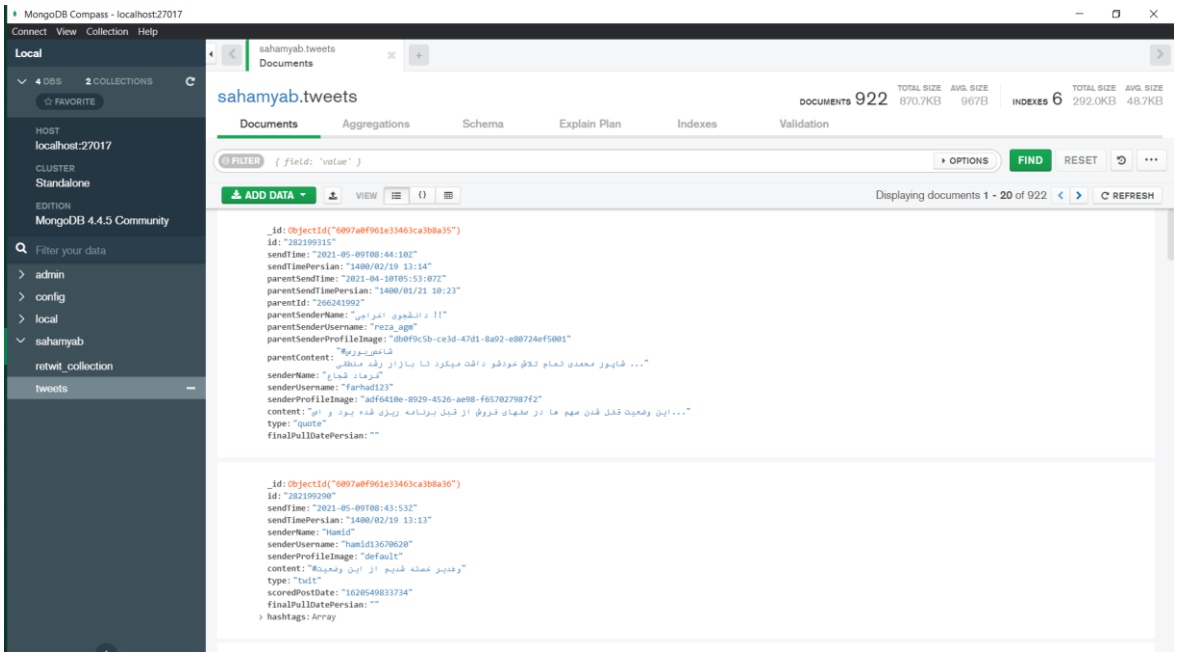
به کمک کد زیر، retwit ها را به کالکشن جدید منتقل می‌کنیم و از کالکشن فعلی حذف می‌کنیم.



عکس زیر کالکشن جدید شامل retwit ها را نشان می‌دهد.



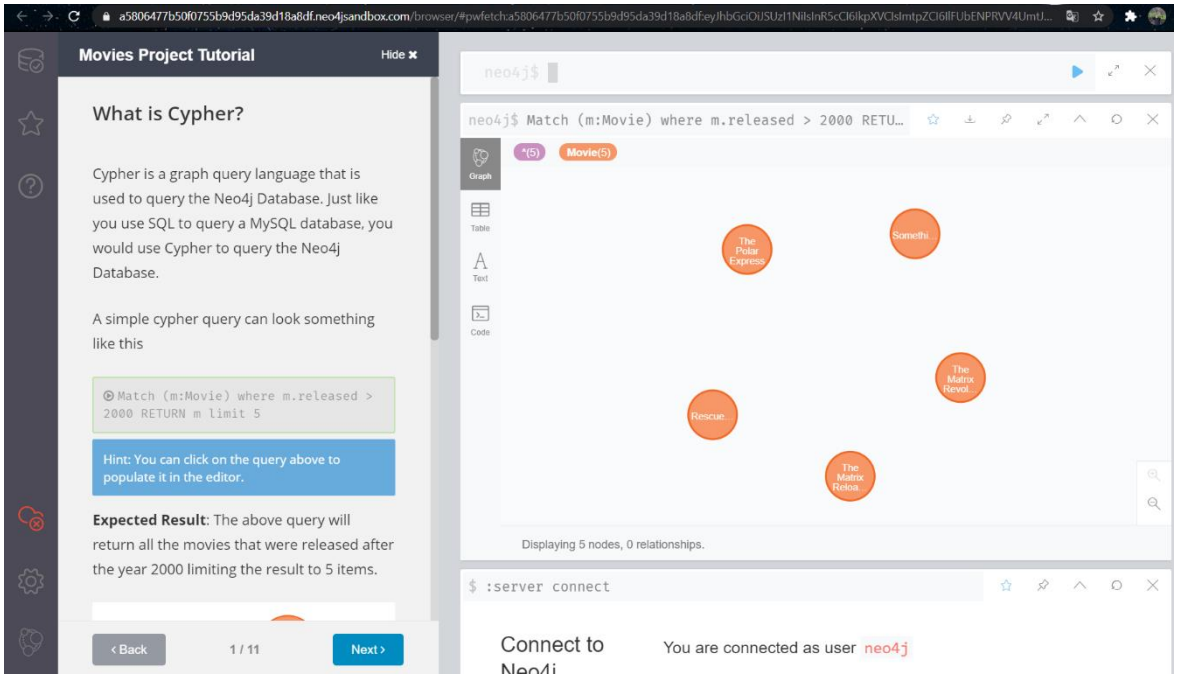
در عکس زیر مشاهده می‌کنیم document هایی که type آن‌ها retwit است از کاکشن قدیمی حذف شده اند.



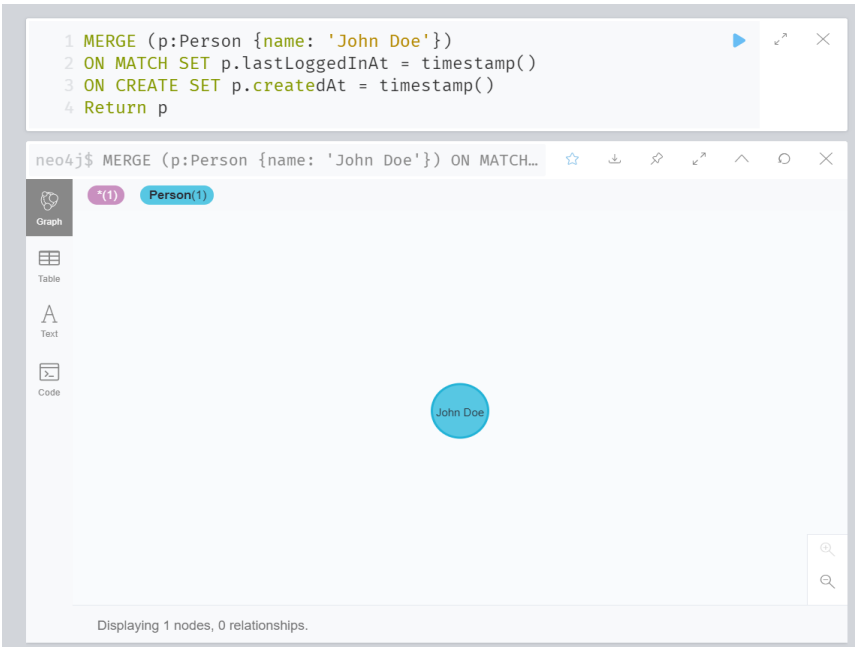
بخش دوم

گام اول

جهت آشنایی با زبان سایفر، به movie project tutorial که به صورت پیش فرض داده های محدودی در آن وجود دارد. عکس زیر محیط سندباکس مربوط به movie را نشان می‌دهد.



توضح چند کوئری:



این کوئری می‌گوید اگر نود person وجود دارد، ویژگی current timestamp آن را برابر زمان فعلی قرار دهد. اگر نود person وجود نداشته باشد، آن را می‌سازد و createdAt آن را برابر زمان فعلی قرار می‌دهد.

کوئری دوم:

ابتدا برای نودی که از نوع person است و name آن "Tom Hanks" است ارتباط ACTED_IN با فیلم‌ها را بررسی می‌کند و سپس افراد دیگری که با همان فیلم‌ها ارتباط ACTED_IN دارند را هم پیدا می‌کند (در اینجا این افراد را p نام‌گذاری کرده است). در واقع این کوئری، بازیگران همکار Tom Hanks را معرفی می‌کند.

کوئری سوم:

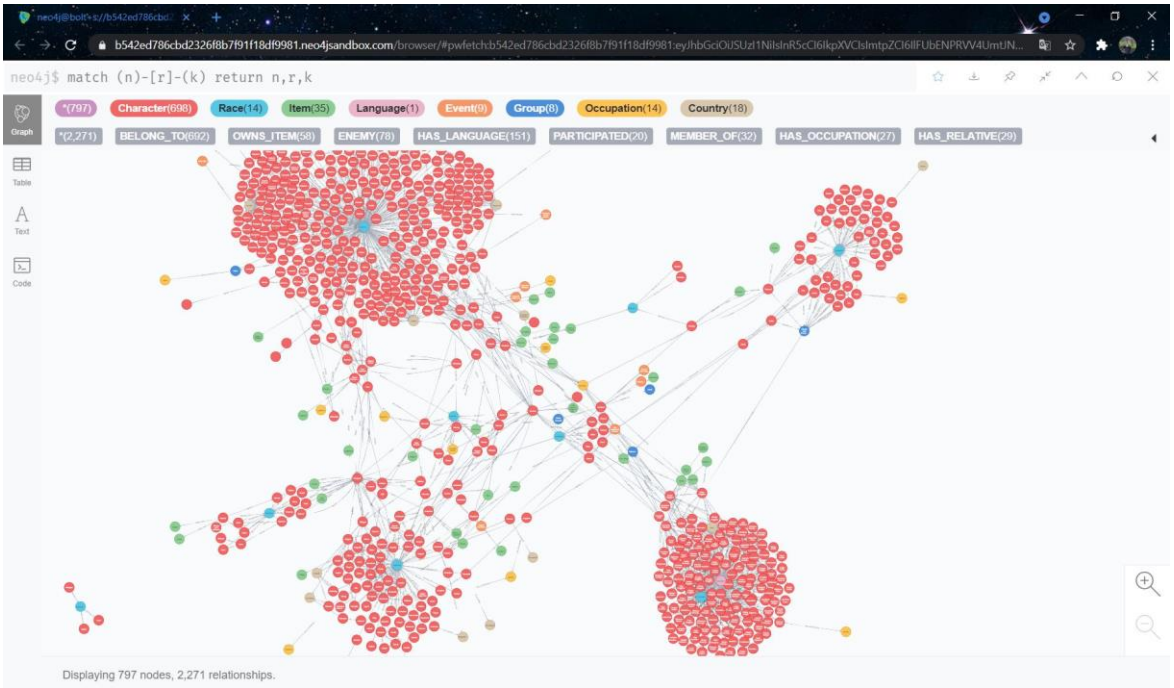
بین نودی شخصی که نام آن Tom Hanks و نود از نوع فیلم که نام آن Cloud Atlas هست، ارتباط جهت دار WATCHED را ایجاد میکند.

گام دوم

Bolt username و password را از سندباکس خالی خودم کپی کردم و در فایل نوت بوکی که در اختیارمان قرار داده شده بود، کپی کردم و سپس همان فایل نوت بوک را در گوگل کولب اجرا کردم. در نتیجه داده ها بارگذاری شد.

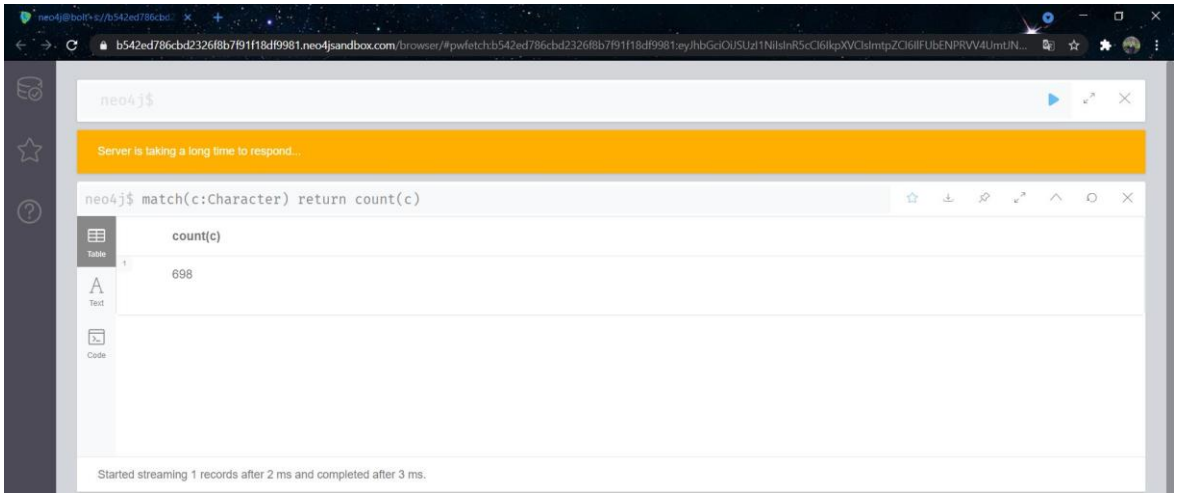
گام سوم

()



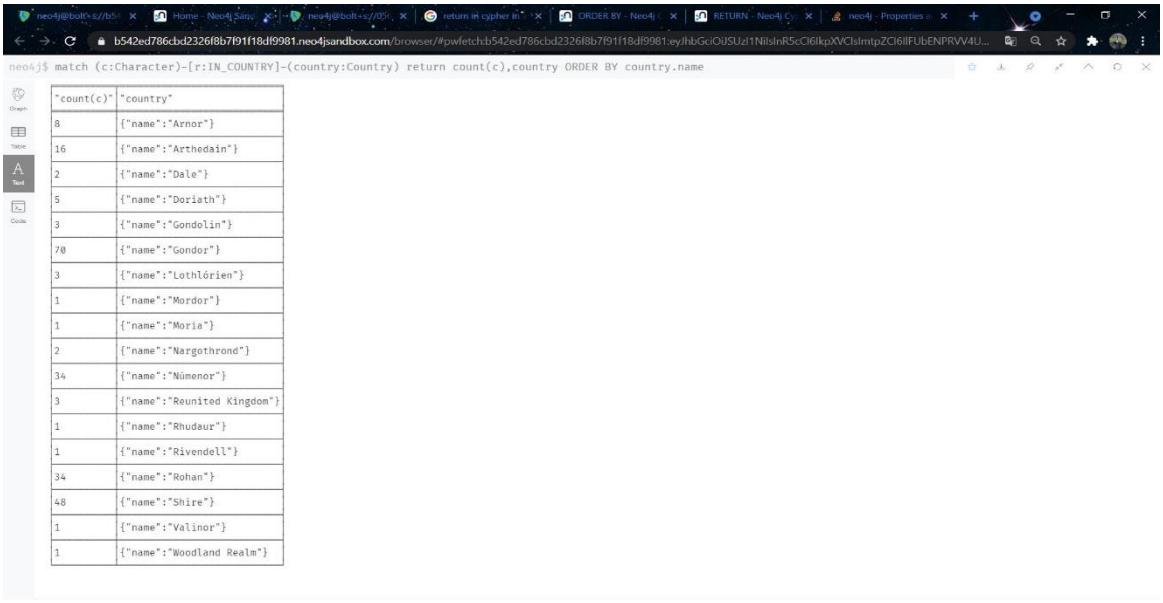
در این کوئری، همه‌ی نودهایی که با نودهای دیگر ارتباطی دارند، آورده شده است.

۲) الف) در این کوئری نودهای با لیبل "Character" را گرفتیم و تعداد این نودها را به عنوان خروجی دریافت کردیم



ب) در این قسمت علاوه بر آنچه در قسمت الف اجرا شد، از distinct استفاده می‌کنیم تا کاراکترهایی که نام یکتا دارند در خروجی ظاهر شوند.

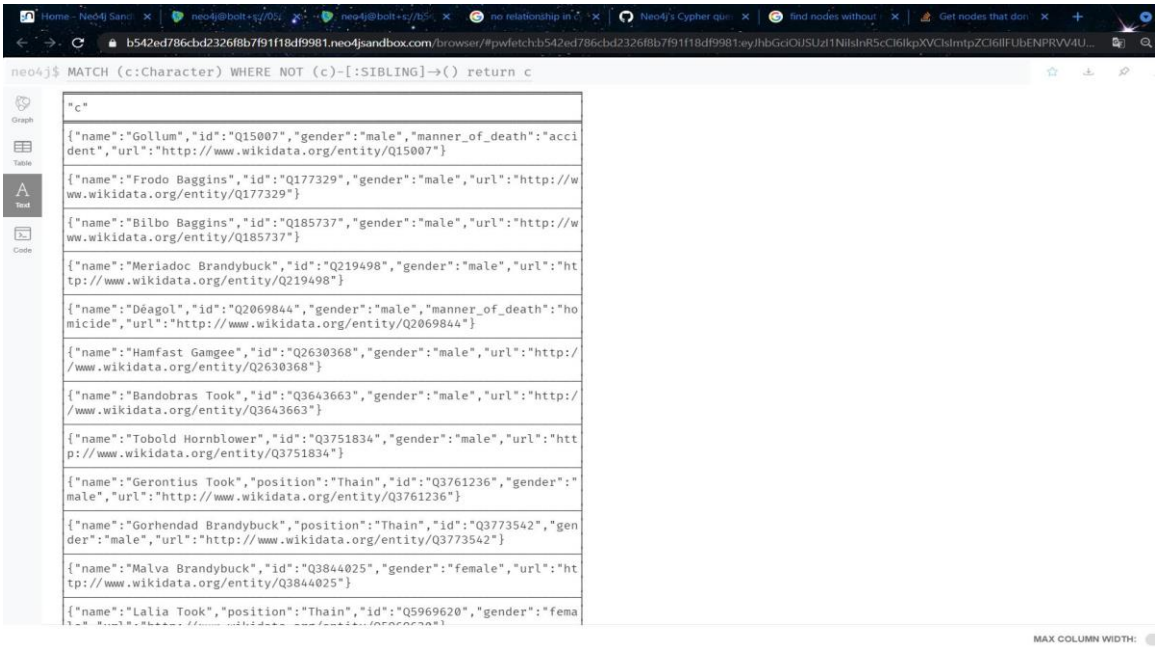
۳) نودهای با لیبل "Character" که با نودهای با لیبل "country" رابطه "IN_COUNTRY" دارند را پیدا می‌کنیم. سپس تعداد افراد و نام کشور را برمی‌گردانیم. از "order by" برای برگرداندن نتیجه برحسب حروف الفبا استفاده می‌کنیم.



```
neo4j$ match (c:Character)-[r:IN_COUNTRY]-(country:Country) return count(c),country ORDER BY country.name
```

"count(c)"	"country"
8	{"name":"Arnor"}
16	{"name":"Arthedain"}
2	{"name":"Dale"}
5	{"name":"Doriath"}
3	{"name":"Gondolin"}
70	{"name":"Gondor"}
3	{"name":"Lothlórien"}
1	{"name":"Mordor"}
1	{"name":"Moria"}
2	{"name":"Nargothrond"}
34	{"name":"Númenor"}
3	{"name":"Reunited Kingdom"}
1	{"name":"Rhudaur"}
1	{"name":"Rivendell"}
34	{"name":"Rohan"}
48	{"name":"Shire"}
1	{"name":"Valinor"}
1	{"name":"Woodland Realm"}

۴) الف) نودهای با لیبل "Character" که با نودهای دیگر رابطه‌ی "SIBLINGS" ندارد را می‌یابیم.

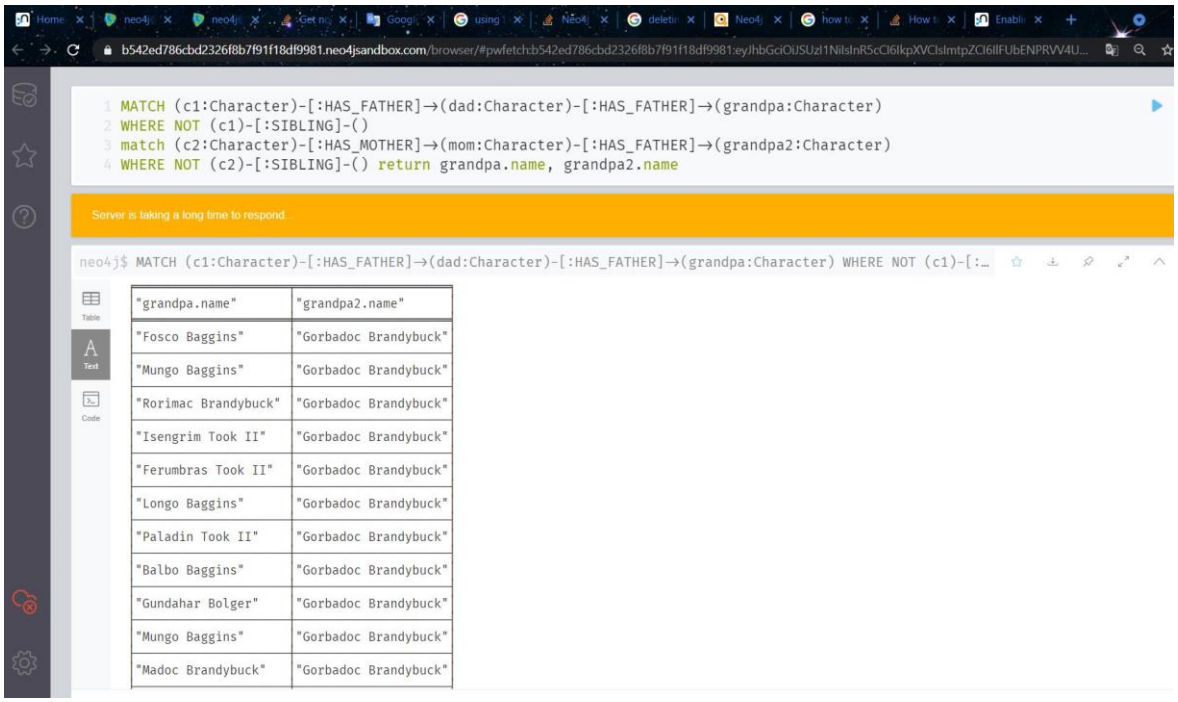


```
neo4j$ MATCH (c:Character) WHERE NOT (c)-[:SIBLING]→() return c
```

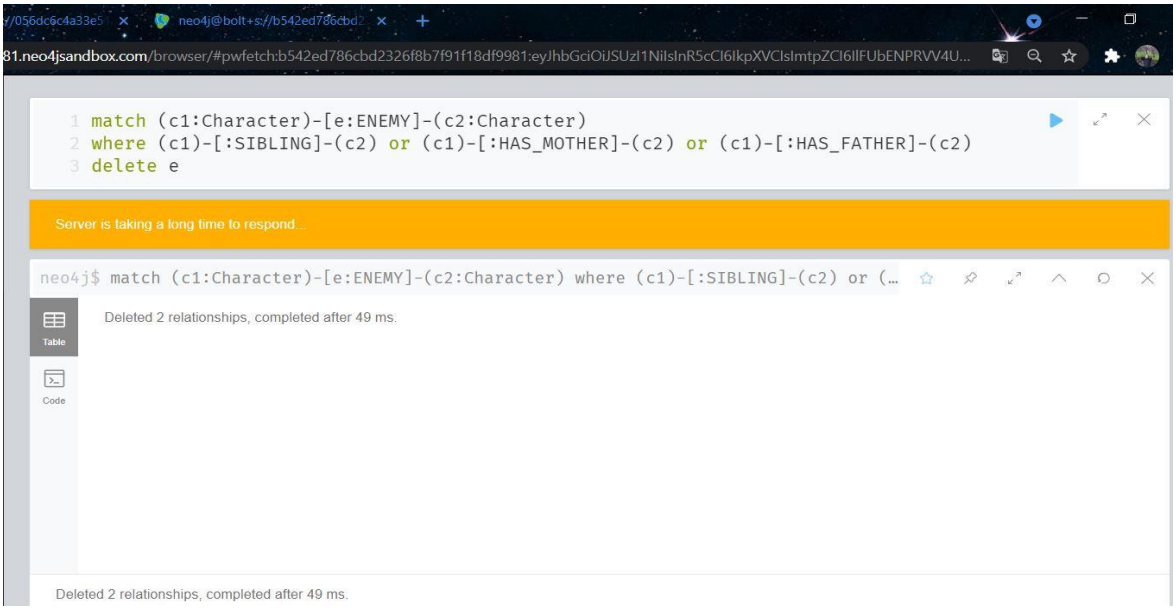
"c"
{"name":"Gollum","id":"Q15007","gender":"male","manner_of_death":"accident","url":"http://www.wikidata.org/entity/Q15007"}
{"name":"Frodo Baggins","id":"Q177329","gender":"male","url":"http://www.wikidata.org/entity/Q177329"}
{"name":"Bilbo Baggins","id":"Q185737","gender":"male","url":"http://www.wikidata.org/entity/Q185737"}
{"name":"Meriadoc Brandybuck","id":"Q219498","gender":"male","url":"http://www.wikidata.org/entity/Q219498"}
{"name":"Déagol","id":"Q2069844","gender":"male","manner_of_death":"homicide","url":"http://www.wikidata.org/entity/Q2069844"}
{"name":"Hamfast Gamgee","id":"Q2630368","gender":"male","url":"http://www.wikidata.org/entity/Q2630368"}
{"name":"Bandobras Took","id":"Q3643663","gender":"male","url":"http://www.wikidata.org/entity/Q3643663"}
{"name":"Tobold Hornblower","id":"Q3751834","gender":"male","url":"http://www.wikidata.org/entity/Q3751834"}
{"name":"Gerontius Took","position":"Thain","id":"Q3761236","gender":"male","url":"http://www.wikidata.org/entity/Q3761236"}
{"name":"Gorhendad Brandybuck","position":"Thain","id":"Q3773542","gender":"male","url":"http://www.wikidata.org/entity/Q3773542"}
{"name":"Malva Brandybuck","id":"Q3844025","gender":"female","url":"http://www.wikidata.org/entity/Q3844025"}
{"name":"Lalia Took","position":"Thain","id":"Q5969620","gender":"fema

MAX COLUMN WIDTH:

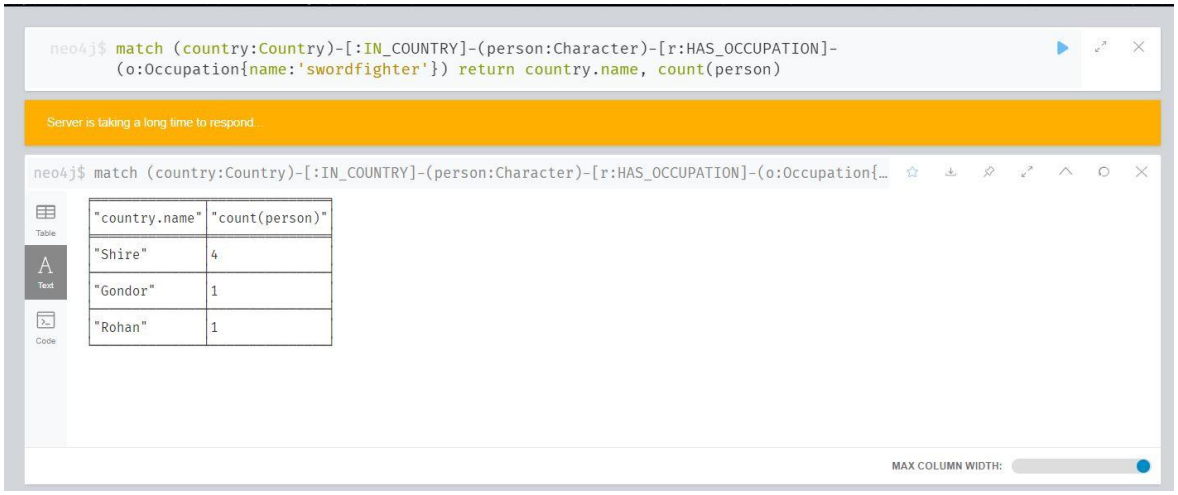
ب) نودهای با لیبل "Character" به نام "c1" (نام‌گذاری توسط خودمان دل‌به‌خواه صورت می‌گیرد) که با نودهای لیبل "Character" به نام "dad" رابطه "HAS_FATHER" دارند را می‌یابیم. سپس یه همین طریق پدربزرگ نودهای "dad" را می‌یابیم. در نهایت شرط عدم وجود رابطه‌ی "SIBLINGS" برای نودهای "c1" را قرار می‌دهیم.



۵) نودهایی با لیبل "Character" که باهم رابطه‌ی دشمنی دارند را می‌یابیم. سپس اگر رابطه خواهر/برادری یا مادر یا پدری داشته باشند، رابطه‌ی دشمنی را حذف می‌کنیم.



۶) الف) نودهایی با لیبل "Character" که هم با نودهایی با لیبل "country" رابطه‌ی "IN_COUNTRY" دارند و هم با نودهای با لیبل "Occupation" و نام "swordfighter" رابطه‌ی "HAS_OCCUPATION" دارند را می‌یابیم



ب)

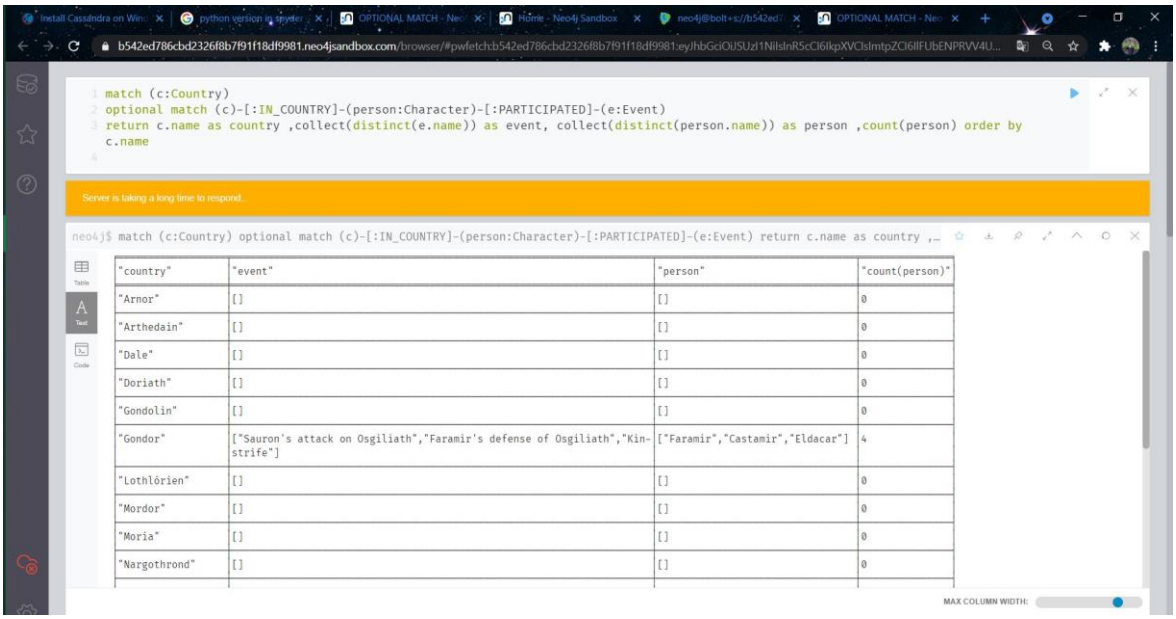
دراین قسمت علاوه بر قسمت الف، خواهر برادر افراد را می‌یابیم. همین‌طور رابطه‌ی "IN_COUNTRY" برای خواهر/برادر را می‌یابیم. سرزمین شمشیرزن‌هایی که رابطه‌ی "IN_COUNTRY" ندارند را از روی سرزمین خواهر/برادر آن‌ها مشخص می‌کنیم.

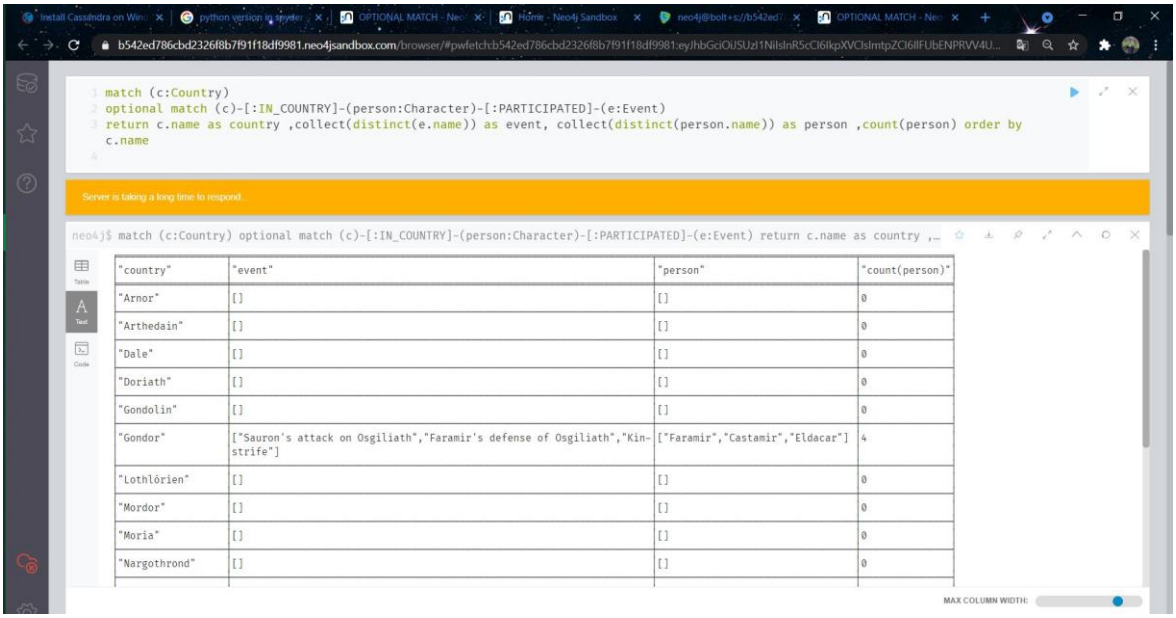
۷)

در خط اول، تعداد شروندان shire را پیدا کردم و سپس آن را در متغیر total قرار می‌دهم (چون از دو تا match استفاده می‌کنیم، باید این کار را انجام داد). سپس در خط دوم، تعداد انهایی که در رخداد خاص شرکت کردند را می‌یابیم.

۸)

برای سرزمین‌هایی که افراد آن در هیچ رخدادی شرکت نکردند، باید لیست خالی برگردانیم. بدین جهت از «optional match» استفاده کردم. در عکس صفحه‌ی بعد، نتایج به صورت کامل وجود دارد.




match (c:Country)
optional match (c)-[:IN_COUNTRY]-(person:Character)-[:PARTICIPATED]-(e:Event)
return c.name as country ,collect(distinct(e.name)) as event, collect(distinct(person.name)) as person ,count(person) order by
c.name
Server is taking a long time to respond.
neo4j$ match (c:Country) optional match (c)-[:IN_COUNTRY]-(person:Character)-[:PARTICIPATED]-(e:Event) return c.name as country ,
"country" "event" "person" "count(person)"
"Arnor" [] [] 0
"Arthedain" [] [] 0
"Dale" [] [] 0
"Doriath" [] [] 0
"Gondolin" [] [] 0
"Gondor" ["Sauron's attack on Osgiliath","Faramir's defense of Osgiliath","Kin-strife"] ["Faramir","Castamir","Eldacar"] 4
"Lothlórien" [] [] 0
"Mordor" [] [] 0
"Moria" [] [] 0
"Nargothrond" [] [] 0
MAX COLUMN WIDTH:

بخش سوم

گام دوم

(۱)

برای ساخت table از کوئری زیر استفاده کردم:

```
CREATE TABLE part_one(track_id int, title_track text, date_released text, title_album text, artist text, duration int, genre text, favorites_artist int, listens_track int, favorites_track int, listens_album int, favorites_album int, PRIMARY KEY(title_album , title_track));
```

سپس با دستور کپی داده‌ها را از فایل csv وارد جدول می‌کنیم:

```
COPY part_one(track_id, title_track, date_released, title_album, artist, duration, genre, favorites_artist, listens_track, favorites_track, listens_album, favorites_album) FROM 'fma_dataset.csv' WITH DELIMITER=',' AND HEADER=TRUE;
```

زمان وارد کردن داده‌ها در عکس زیر آمده است:

```
Processed: 106574 rows; Rate:   15505 rows/s; Avg. rate:   15490 rows/s
105159 rows imported from 1 files in 6.882 seconds (0 skipped).
cqlsh:fma>
```

سپس مطلوب سوال و زمان اجرای کوئری به کمک کوئری زیر بدست می‌آید:

```
cqlsh:fma> TRACING ON
Now Tracing is enabled
cqlsh:fma> SELECT title_TRACK FROM part_one where title_album='Rumble, Young Man, Rumble';

 title_track
-------------------------------------------------------------
                    I Listen 2 Classical (Featuring Sleaze)
                I'm Not Lyin' (Featuring Octavion Xcellence)
         No Title For It (Featuring Ohbliv and Gordy Michael)
                          Out Of Here (Featuring Mic Jordan)
                 Primetime Bandit (Remix) (Featuring Draztiq)
 Request (Featuring Barcodez and SamSun of Photosynthesizers)
                          Scream Out (Featuring Braintrust)
                    Some People Never Learn (Featuring NOTE)
                             Take One (Featuring Joey Ripps)
                These Times (Featuring Sleaze and Joey Ripps)
  This Is Madness (Just Plain Black) (Featuring Black Liquid)
                          Way Back When (Featuring Chuck D)

(12 rows)

Tracing session: f33a94f0-bbcb-11eb-8fdd-d9c8370df47b

 activity                                                                                                                                                  | timestamp                  | source    | sou
-----------------------------------------------------------------------------------------------------------------------------------------------------------+----------------------------+-----------+----
                                                                                                                                        Execute CQL3 query | 2021-05-23 18:06:59.842000 | 127.0.0.1 |
 Parsing SELECT title_TRACK FROM part_one where title_album='Rumble, Young Man, Rumble'; [Native-Transport-Requests-1] | 2021-05-23 18:06:59.850000 | 127.0.0.1 |
                                                                                                  Preparing statement [Native-Transport-Requests-1] | 2021-05-23 18:06:59.850000 | 127.0.0.1 |
                                                                                Executing single-partition query on part_one [ReadStage-3] | 2021-05-23 18:06:59.851000 | 127.0.0.1 |
                                                                                          Acquiring sstable references [ReadStage-3] | 2021-05-23 18:06:59.851000 | 127.0.0.1 |
                                                                  Skipped 0/0 non-slice-intersecting sstables, included 0 due to tombstones [ReadStage-3] | 2021-05-23 18:06:59.851000 | 127.0.0.1 |
                                                                                Merged data from memtables and 0 sstables [ReadStage-3] | 2021-05-23 18:06:59.851000 | 127.0.0.1 |
                                                                                  Read 12 live rows and 0 tombstone cells [ReadStage-3] | 2021-05-23 18:06:59.852000 | 127.0.0.1 |
                                                                                                                        Request complete | 2021-05-23 18:06:59.853226 | 127.0.0.1 |
```

جهت دریافت زمان اجرای کوئری از TRACING ON استفاده شده است (مدت زمان 0.011226 ثانیه است).

۲)

در این قسمت کلیدهای اساسی چنین تعریف شده اند:

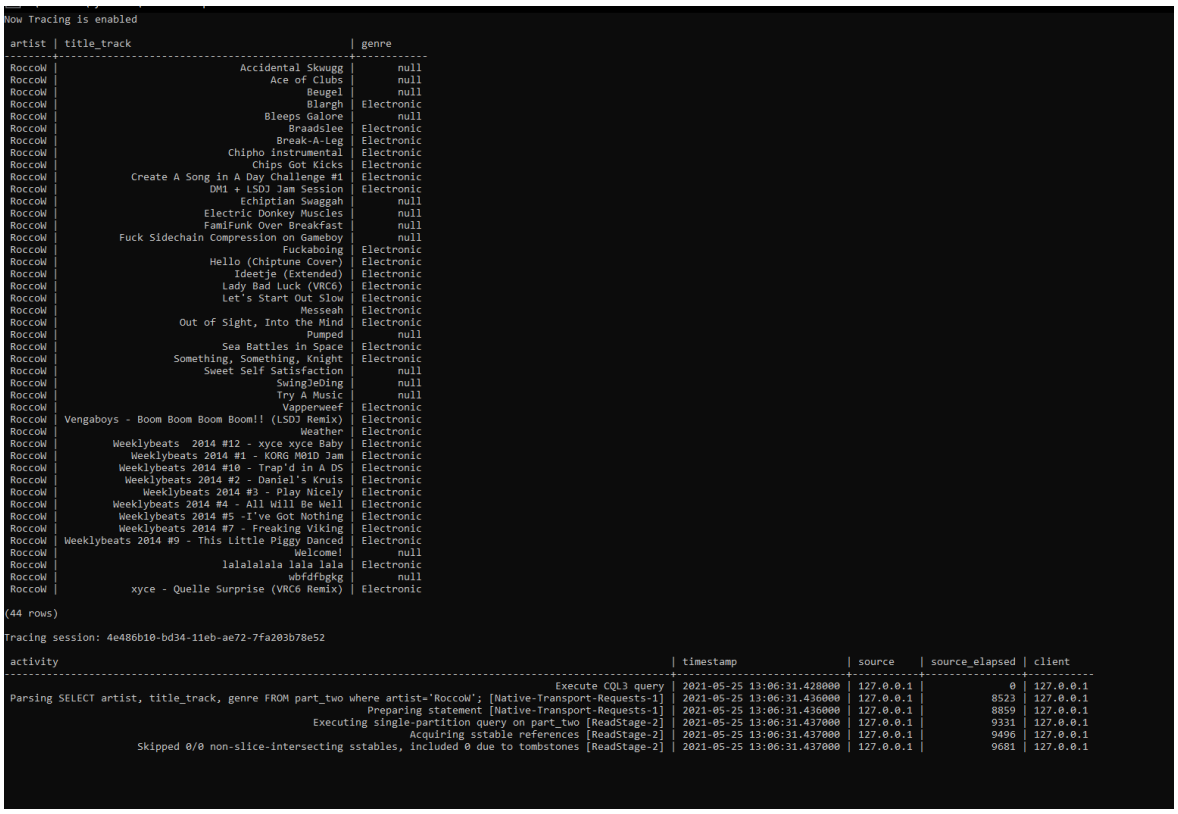
PRIMARY KEY(artist , title_track)

و کوئری زده شده چنین است:

TRACING ON; SELECT artist, title_track, genre FROM part_two where artist='RoccoW';

توجه داریم که در کاساندرا مقدار null وجود ندارد و برای کلیدهایی که مقداری ندارند، به کاربران برای فهم بهتر null نشان می‌دهند. از این باب، کلیدهایی که مقدار null دارند، نمی‌توانند به عنوان primary key تعریف شوند.

خروجی مورد نظر و زمان اجرا:



```
Now Tracing is enabled

 artist | title_track                                  | genre
--------+----------------------------------------------+-----------
 RoccoW |                            Accidental Skwugg |       null
 RoccoW |                                 Ace of Clubs |       null
 RoccoW |                                       Beugel |       null
 RoccoW |                                       Blargh | Electronic
 RoccoW |                                  Bleeps Galore |       null
 RoccoW |                                    Braadslee | Electronic
 RoccoW |                                   Break-A-Leg | Electronic
 RoccoW |                              Chipho instrumental | Electronic
 RoccoW |                                 Chips Got Kicks | Electronic
 RoccoW |           Create A Song in A Day Challenge #1 | Electronic
 RoccoW |                          DM1 + LSDJ Jam Session | Electronic
 RoccoW |                              Echiptian Swaggah |       null
 RoccoW |                         Electric Donkey Muscles |       null
 RoccoW |                          FamiFunk Over Breakfast |       null
 RoccoW |          Fuck Sidechain Compression on Gameboy |       null
 RoccoW |                                  Fuckaboing | Electronic
 RoccoW |                            Hello (Chiptune Cover) | Electronic
 RoccoW |                                Ideetje (Extended) | Electronic
 RoccoW |                              Lady Bad Luck (VRC6) | Electronic
 RoccoW |                              Let's Start Out Slow | Electronic
 RoccoW |                                        Messeah | Electronic
 RoccoW |                       Out of Sight, Into the Mind | Electronic
 RoccoW |                                         Pumped |       null
 RoccoW |                            Sea Battles in Space | Electronic
 RoccoW |                      Something, Something, Knight | Electronic
 RoccoW |                           Sweet Self Satisfaction |       null
 RoccoW |                                      Swing3eDing |       null
 RoccoW |                                     Try A Music |       null
 RoccoW |                                      Vapperweef | Electronic
 RoccoW | Vengaboys - Boom Boom Boom Boom!! (LSDJ Remix) | Electronic
 RoccoW |                                         Weather | Electronic
 RoccoW |          Weeklybeats  2014 #12 - xyce xyce Baby | Electronic
 RoccoW |            Weeklybeats 2014 #1 - KORG M01D Jam | Electronic
 RoccoW |           Weeklybeats 2014 #10 - Trap'd in A DS | Electronic
 RoccoW |           Weeklybeats 2014 #2 - Daniel's Kruis | Electronic
 RoccoW |              Weeklybeats 2014 #3 - Play Nicely | Electronic
 RoccoW |          Weeklybeats 2014 #4 - All Will Be Well | Electronic
 RoccoW |           Weeklybeats 2014 #5 -I've Got Nothing | Electronic
 RoccoW |           Weeklybeats 2014 #7 - Freaking Viking | Electronic
 RoccoW | Weeklybeats 2014 #9 - This Little Piggy Danced | Electronic
 RoccoW |                                       Welcome! |       null
 RoccoW |                              lalalalala lala lala | Electronic
 RoccoW |                                      wbfdfbgkg |       null
 RoccoW |            xyce - Quelle Surprise (VRC6 Remix) | Electronic

(44 rows)

Tracing session: 4e486b10-bd34-11eb-ae72-7fa203b78e52

 activity                                                                                                  | timestamp                  | source    | source_elapsed | client
-----------------------------------------------------------------------------------------------------------+----------------------------+-----------+----------------+-----------
                                                                                        Execute CQL3 query | 2021-05-25 13:06:31.428000 | 127.0.0.1 |              0 | 127.0.0.1
 Parsing SELECT artist, title_track, genre FROM part_two where artist='RoccoW'; [Native-Transport-Requests-1] | 2021-05-25 13:06:31.436000 | 127.0.0.1 |           8523 | 127.0.0.1
                                                  Preparing statement [Native-Transport-Requests-1] | 2021-05-25 13:06:31.436000 | 127.0.0.1 |           8859 | 127.0.0.1
                                       Executing single-partition query on part_two [ReadStage-2] | 2021-05-25 13:06:31.437000 | 127.0.0.1 |           9331 | 127.0.0.1
                                                    Acquiring sstable references [ReadStage-2] | 2021-05-25 13:06:31.437000 | 127.0.0.1 |           9496 | 127.0.0.1
                  Skipped 0/0 non-slice-intersecting sstables, included 0 due to tombstones [ReadStage-2] | 2021-05-25 13:06:31.437000 | 127.0.0.1 |           9681 | 127.0.0.1
```

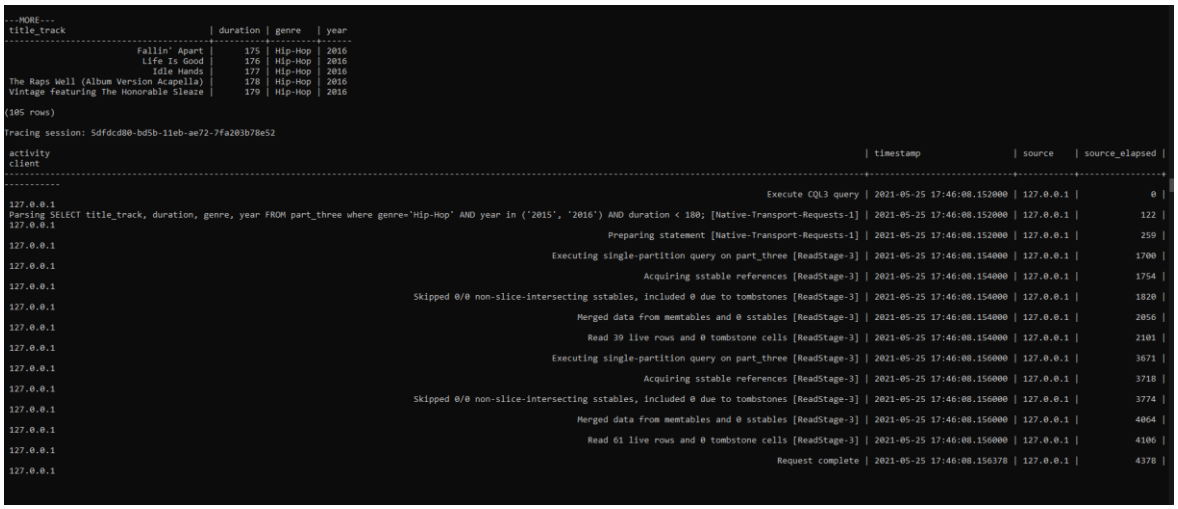
۳) در این قسمت ابتدا به کمک پایتون، ستون year را به فایل csv اظافه کردم و سپس به کمک کوئری زیر خروجی بدست می‌آید.

TRACING ON; SELECT title_track, duration, genre, year FROM part_three where genre='Hip-Hop' AND year in ('2015', '2016') AND duration < 180;

خروجی: ۱۰۵ آهنگ به عنوان خروجی نمایش داده شد.

```
cqlsh:fma> TRACING ON; SELECT title_track, duration, genre, year FROM part_three where genre='Hip-Hop' AND year i
Tracing is already enabled. Use TRACING OFF to disable.

 title_track                                        | duration | genre   | year
----------------------------------------------------+----------+---------+------
                                Gimme That Tape! |        5 | Hip-Hop | 2015
                          Quando O Universo Explode |       27 | Hip-Hop | 2015
                                pathetic at best |       31 | Hip-Hop | 2015
                                     Hey, Guys |       37 | Hip-Hop | 2015
                          Vamos Perguntar ├á Plateia |       38 | Hip-Hop | 2015
                                     Homie Jota |       41 | Hip-Hop | 2015
                                   Aquele Polvo |       57 | Hip-Hop | 2015
                               The 1 Upper (84 Mixx) |       60 | Hip-Hop | 2015
                                   Sara Sampaio |       61 | Hip-Hop | 2015
                         The 1 Upper (BreakBeat Mixx) |       73 | Hip-Hop | 2015
                                   She's a man? |       75 | Hip-Hop | 2015
                               These Dreams And Shit |       93 | Hip-Hop | 2015
                                       Whentro |       97 | Hip-Hop | 2015
                        Afterschool Speshul Bonus Beats |      101 | Hip-Hop | 2015
                                            54% |      109 | Hip-Hop | 2015
                                     Simon Pegg |      116 | Hip-Hop | 2015
                                     prometheus |      122 | Hip-Hop | 2015
                        Hey You Guys (Sloth Cause Mixx) |      131 | Hip-Hop | 2015
                                      Bird Up! |      135 | Hip-Hop | 2015
                                    Footprints |      140 | Hip-Hop | 2015
                                      Pop Tart |      141 | Hip-Hop | 2015
                           Blue Steel (Instrumental) |      143 | Hip-Hop | 2015
                         Spot Rockers (Gotcha Man Mix) |      144 | Hip-Hop | 2015
                          Drag├úo Azul, Drag├úo Vermelho |      148 | Hip-Hop | 2015
                           The 1 Upper (Do Me Remix) |      150 | Hip-Hop | 2015
                                      Penny17 |      152 | Hip-Hop | 2015
                            Hey You Guys (Acapella) |      153 | Hip-Hop | 2015
  Tonight You Belong To Me (Featuring Kat Wahamaa) |      155 | Hip-Hop | 2015
                                     Thrashin |      156 | Hip-Hop | 2015
                        Spot Rockers (Doc's 2-Live Mixx) |      157 | Hip-Hop | 2015
                           Spot Rockers (Blank Mixx) |      158 | Hip-Hop | 2015
                                  PS Love Always |      159 | Hip-Hop | 2015
                                      crumbs |      163 | Hip-Hop | 2015
      Poetikal Refugee ft. Mikey Krumins aka Dyems |      168 | Hip-Hop | 2015
                                  Eat Some Cake |      169 | Hip-Hop | 2015
                                    Homie Stray |      170 | Hip-Hop | 2015
                                    C-N-P Enuff |      175 | Hip-Hop | 2015
                                      goldheart |      176 | Hip-Hop | 2015
                                          Gruta |      177 | Hip-Hop | 2015
                           Welcome To Chi-Raq Intro |       30 | Hip-Hop | 2016
                           Welcome To Chi-Raq Outtro |       42 | Hip-Hop | 2016
                         Drugs By Roman Candlelight |       57 | Hip-Hop | 2016
                                  Phon's Waltz |       62 | Hip-Hop | 2016
                              Dead Homie Intro |       64 | Hip-Hop | 2016
                                         Intro |       67 | Hip-Hop | 2016
                                         Intro |       70 | Hip-Hop | 2016
                                    drunkenCPU |       82 | Hip-Hop | 2016
                                     Fireheart |       84 | Hip-Hop | 2016
                                     Needy Ass |       85 | Hip-Hop | 2016
                                    Cawper Crow |       87 | Hip-Hop | 2016
                                    Mello Hello |      100 | Hip-Hop | 2016
 Money Beets: A Tribute to RHP (Featuring Milkdrop) |      104 | Hip-Hop | 2016
                                      activate |      106 | Hip-Hop | 2016
                                Suggestion Rocks |      109 | Hip-Hop | 2016
                               Pusher For Record |      111 | Hip-Hop | 2016
                           Daddy's Drinkin' Song |      113 | Hip-Hop | 2016
                 Souvenirs, Novelties, Party Tricks |      114 | Hip-Hop | 2016
                           Creepin' On Your Shh |      116 | Hip-Hop | 2016
```

ادامه خروجی و زمان اجرا در عکس زیر قابل مشاهده است:



۴) مشابهه قسمت قبل، ستون month را به فایل csv اضافه می‌کنیم و سپس به کمک کوئری زیر خروجی بدست می‌آید:

```
TRACING ON; SELECT title_track, genre, month, listens_track FROM part_four where genre in ('Electronic', 'Pop') AND month='Apr' AND listens_track > 300;
```

خروجی: ۲۷۳ آهنگ به عنوان خروجی چاپ شد. به دلیل زیاد بودن تعداد خروجی ها، از نمایش همه‌ی آن‌ها در اینجا صرف نظر می‌کنم و تعدادی از آن‌ها را در دو عکس نمایش می‌دهم:

```
cqlsh:fma> TRACING ON; SELECT title_track, genre, month, listens_track FROM part_four where genre in ('Electronic', 'Pop') AND month='Apr' AND listens_track > 300;
Tracing is already enabled. Use TRACING OFF to disable.

 title_track                                        | genre      | month | listens_track
----------------------------------------------------+------------+-------+---------------
                                             wobble | Electronic |   Apr |           301
                                          Rough Sex | Electronic |   Apr |           302
                                         Peloponnes | Electronic |   Apr |           309
                                Off In The Distance | Electronic |   Apr |           315
                                       Low on Ideas | Electronic |   Apr |           317
                                Umbrella Pete Strut | Electronic |   Apr |           322
                                         Candy Funk | Electronic |   Apr |           326
                                   Once Upon A Time | Electronic |   Apr |           329
                                           Marathon | Electronic |   Apr |           332
 Vodka Bitter Smartie Party (Ergo Phizmiz' version) | Electronic |   Apr |           340
                                     So I Went Home | Electronic |   Apr |           343
                                          Slow jam | Electronic |   Apr |           346
                                     Pick Your Brain | Electronic |   Apr |           353
                                      Chocolate Sky | Electronic |   Apr |           356
                               Amorpheus (2b20 rebuild) | Electronic |   Apr |           358
                                        i don't know | Electronic |   Apr |           363
                        Bilboquet (La M├-lodie Jetable) | Electronic |   Apr |           370
                             Mr. Pig & Mr. Chrome | Electronic |   Apr |           380
                                            Goodbye | Electronic |   Apr |           382
                                                5.7 | Electronic |   Apr |           388
                                          Nel Vuoto | Electronic |   Apr |           391
                                             Freaks | Electronic |   Apr |           415
                        Rising in the Wake of Supernova | Electronic |   Apr |           422
             Dismantling Earth Based Authority Via Satellite | Electronic |   Apr |           423
                                            Track 3 | Electronic |   Apr |           428
                                       Nolan Thing 3 | Electronic |   Apr |           438
                                             Nausea | Electronic |   Apr |           443
                                          [Track 2] | Electronic |   Apr |           445
                                             Fabric | Electronic |   Apr |           450
                                              Logue | Electronic |   Apr |           451
                                        mighty nice | Electronic |   Apr |           453
                                 Sleep Just To Dream | Electronic |   Apr |           460
                                   Straight Blimpin' | Electronic |   Apr |           467
                                              Sluni | Electronic |   Apr |           478
                        Vodka Alone (Vernon's version) | Electronic |   Apr |           479
                                               Zerr | Electronic |   Apr |           481
                                      Midget Madness | Electronic |   Apr |           483
                                                Coz | Electronic |   Apr |           485
                                               Spot | Electronic |   Apr |           493
                                          Untitled3 | Electronic |   Apr |           499
                     Time: disabled (Step by Step edit) | Electronic |   Apr |           502
                                           Absolute | Electronic |   Apr |           513
                                      N N R M L Z M | Electronic |   Apr |           518
                              Save, Develop, Protect | Electronic |   Apr |           519
                                            Track 1 | Electronic |   Apr |           521
                                       Janet's Voice | Electronic |   Apr |           523
                                     Sogno D'Autunno | Electronic |   Apr |           527
                                        Santa Claus | Electronic |   Apr |           530
                                          Greenheys | Electronic |   Apr |           531
                                        you often ebb | Electronic |   Apr |           534
                                 I Put A Spell On You | Electronic |   Apr |           536
                                        green butter | Electronic |   Apr |           539
                                              Light | Electronic |   Apr |           540
                                     chicken factory | Electronic |   Apr |           551
                                        Empty Field | Electronic |   Apr |           559
                               Snail! Snail! Rocknroll | Electronic |   Apr |           567
                               Putting Spells On Birds | Electronic |   Apr |           570
                                            Landing | Electronic |   Apr |           573
                                     Face Distortion | Electronic |   Apr |           575
```

```
C:\Windows\System32\cmd.exe - cqlsh
                      Moi Samolety |        Pop |   Apr |           635
                       Our Sorrows |        Pop |   Apr |           663
                   Stinger Sisters |        Pop |   Apr |           703
                             Brute |        Pop |   Apr |           704
       probably digging my own grave |        Pop |   Apr |           710
                           Nevesta |        Pop |   Apr |           714
          When The Sea Called Me Home |        Pop |   Apr |           732
                            gneiss |        Pop |   Apr |           796
                Hey, Predator Drone |        Pop |   Apr |           834
               Nevesta (Flayve Remix) |        Pop |   Apr |           855
                          Giraffes |        Pop |   Apr |           872
                  Betsy On The Roof |        Pop |   Apr |           880
                     Brute (Intro) |        Pop |   Apr |           972
                    Discount Store |        Pop |   Apr |           999
                 Study For A Pop Band |        Pop |   Apr |          1037
                  that face you wear |        Pop |   Apr |          1132
                             Malta |        Pop |   Apr |          1160
                           Mirrors |        Pop |   Apr |          1207
                      Palm Springs |        Pop |   Apr |          1432
                          Almighty |        Pop |   Apr |          2010
                           Trouble |        Pop |   Apr |          2079
                         An Island |        Pop |   Apr |          2093
                            I Feel |        Pop |   Apr |          2582

(273 rows)

Tracing session: 2ed197d0-bd60-11eb-ae72-7fa203b78e52

 activity                                                                                                                                 | timestamp
ce_elapsed | client
------------------------------------------------------------------------------------------------------------------------------------------+------------------
-----------+-----------

                                                                                                                      Execute CQL3 query | 2021-05-25 18:20:36.
         0 | 127.0.0.1
 Parsing SELECT title_track, genre, month, listens_track FROM part_four where genre in ('Electronic', 'Pop') AND month='Apr' AND listens_track > 300; [Native-Transport-Requests-1] | 2021-05-25 18:20:36.
       108 | 127.0.0.1

                                                                             Preparing statement [Native-Transport-Requests-1] | 2021-05-25 18:20:36.
       287 | 127.0.0.1

                                                                  Executing single-partition query on part_four [ReadStage-3] | 2021-05-25 18:20:36.
      1888 | 127.0.0.1

                                                                               Acquiring sstable references [ReadStage-3] | 2021-05-25 18:20:36.
      1925 | 127.0.0.1

                                    Skipped 0/0 non-slice-intersecting sstables, included 0 due to tombstones [ReadStage-3] | 2021-05-25 18:20:36.
      2036 | 127.0.0.1

                                                                   Merged data from memtables and 0 sstables [ReadStage-3] | 2021-05-25 18:20:36.
      2522 | 127.0.0.1

                                                                  Read 100 live rows and 0 tombstone cells [ReadStage-3] | 2021-05-25 18:20:36.
      2641 | 127.0.0.1

                                                                                                        Request complete | 2021-05-25 18:20:36.
      3124 | 127.0.0.1
```

۷) متوسط طول آهنگ‌ها به کمک کوئری زیر محاسبه شده است:

```
SELECT AVG (duration) FROM part_seven;
```

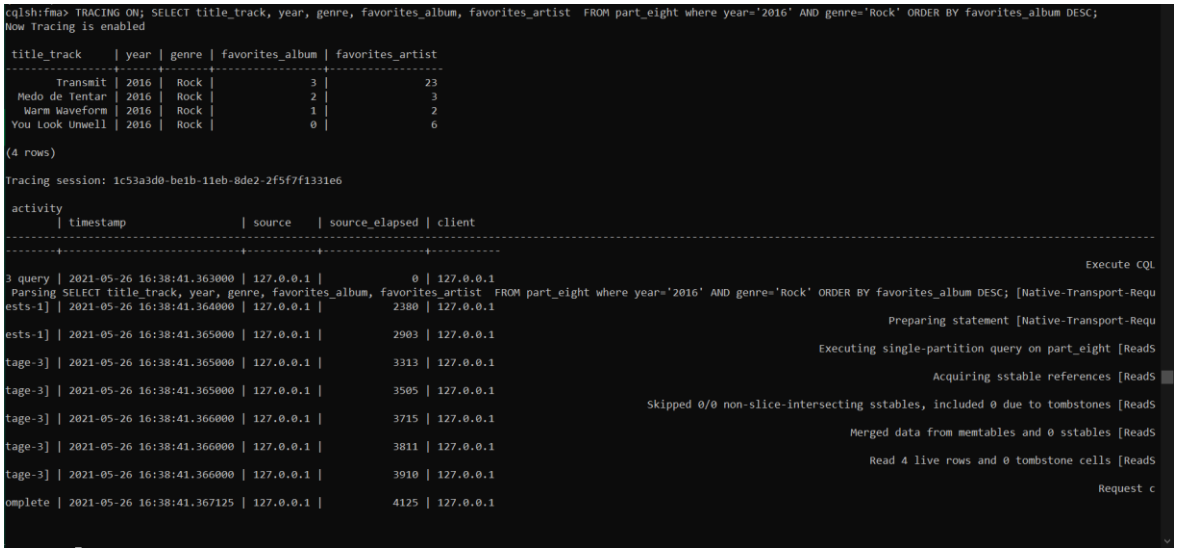
و خروجی بدین شکل است:

```
cqlsh:fma> SELECT AVG (duration) FROM part_seven;

 system.avg(duration)
----------------------
                  279

(1 rows)
```

کوئری و خروجی برای ۱۰ آهگی که بیشترین طول را دارند، از این قرارند:

```
cqlsh:fma> SELECT title_track, duration FROM part_seven_2 where temp=1  ORDER BY duration DESC LIMIT 10;

 title_track                      | duration
----------------------------------+----------
    Harmonia Live at ATP 08 part 1 |    18350
                     In the Garden |    11030
                Adam Are You Free? |    11016
                   RAPE OF NANKING |    10999
                   Orrrrgaaanooone |     7372
                      OceanTapping |     7320
     rad[Incidental Music]ius 0025 |     3718
     Elliott Sharp - ISSUE Podcast |     3717
 Passio Domini Nostri Jesu Christe |     3716
                      To Our Yes |     3714

(10 rows)
cqlsh:fma>
```

۸) کوئری، خروجی و زمان اجرا در عکس قابل مشاهده است.



۹)برای حل این قسمت، دو جدول تعریف کردم. از آنجایی که در کاساندرا امکان سورت کردن روی خروجی counter وجود ندارد، خروجی کوئری table اول را در یک فایل csv موقت ذخیره کردم و سپس یک table دوم تعریف کردم و داده‌ها را از فایل csv موقت وارد table دوم کردم. تمامی کوئری هایی که مورد استفاده قرار گرفتن در عکس زیر قابل مشاهده است:

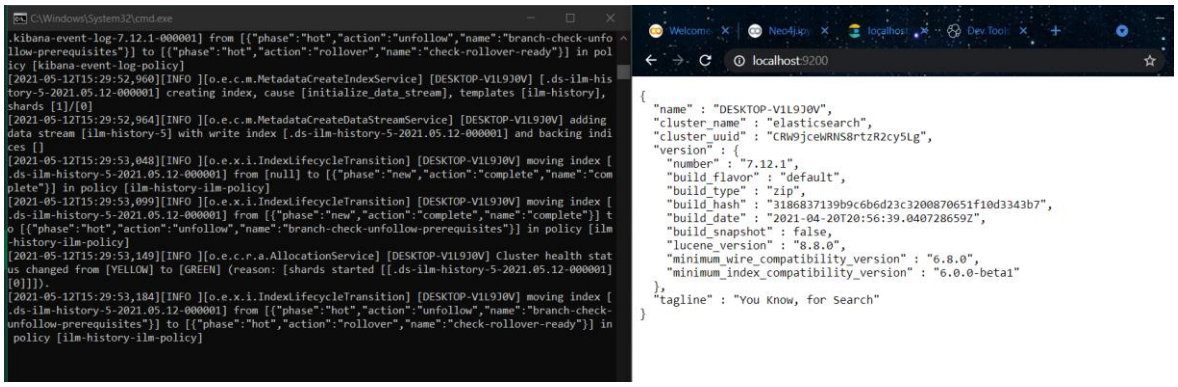
```
CREATE TABLE part_nine(track_id int, title_track text, date_released text, title_album text, artist text, duration int, genre text

COPY part_nine(track_id, title_track, date_released, title_album, artist, duration, genre, favorites_artist, listens_track, favori

#using capture to write outpute as csv file in temp_9:

CAPTURE; CAPTURE 'temp_9.csv'; SELECT genre, COUNT(listens_track) as counter, temp FROM part_nine GROUP BY genre;
CAPTURE off;

# second table:
CREATE TABLE part_nine_2(genre text, counter int, temp int,  PRIMARY KEY(temp, counter ));

COPY part_nine_2(genre, counter, temp ) FROM 'temp_9.csv' WITH DELIMITER='|' AND HEADER=TRUE;

select genre, counter from part_nine_2 WHERE temp=1 ORDER BY counter DESC;
```

خروجی:

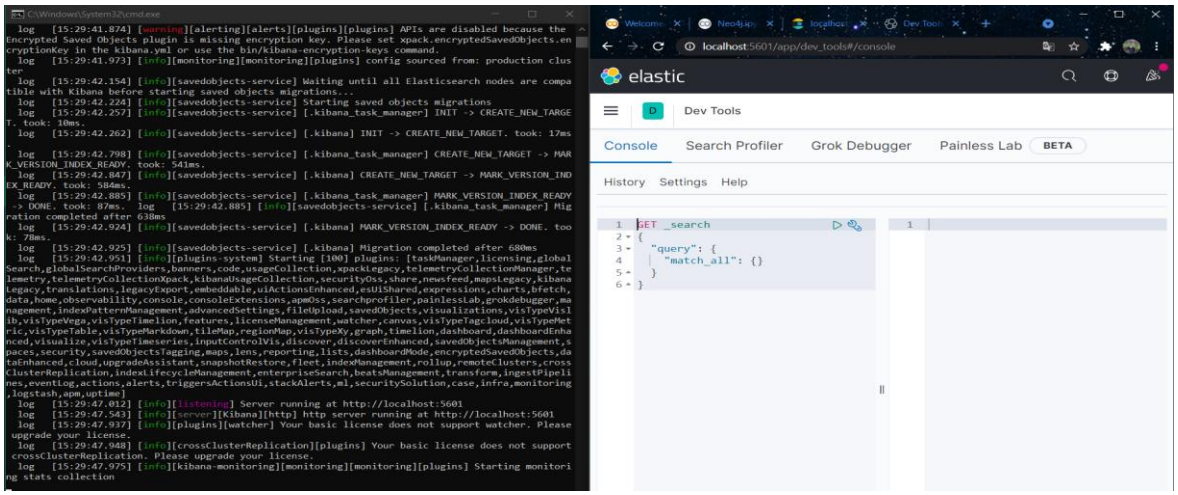
```
cqlsh:fma> select genre, counter from part_nine_2 WHERE temp=1 ORDER BY counter DESC;

 genre                | counter
----------------------+---------
          Electronic  |    3702
                Rock  |    2876
             Hip-Hop  |    1897
        Experimental  |    1873
        Instrumental  |    1789
                Folk  |    1438
                 Pop  |    1399
       International  |    1165
           Classical  |    1072
                Jazz  |     538
  Old-Time / Historic |     521
              Spoken  |     368
             Country  |     193
           Soul-RnB   |     171
               Blues  |      99
      Easy Listening  |      24

(16 rows)
```

بخش چهارم

نصب الستیک سرچ:



نصب کیبانا:



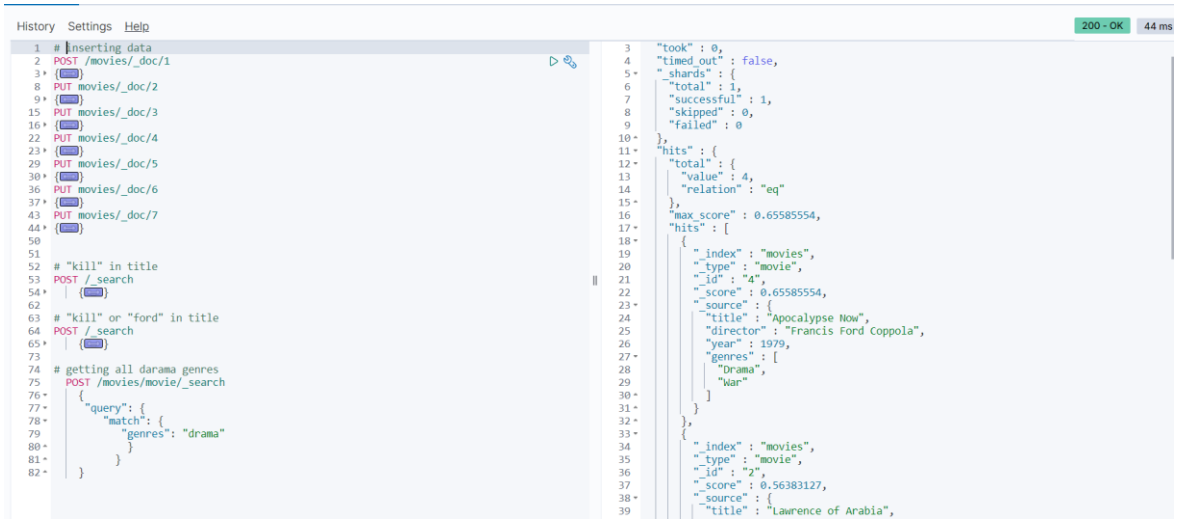
ورود داده ها:

جستجو فیلم‌هایی که کلمه kill در title آن‌ها وجود دارد:

جستجو فیلم‌هایی که کلمه kill یا ford در title آن‌ها وجود دارد:

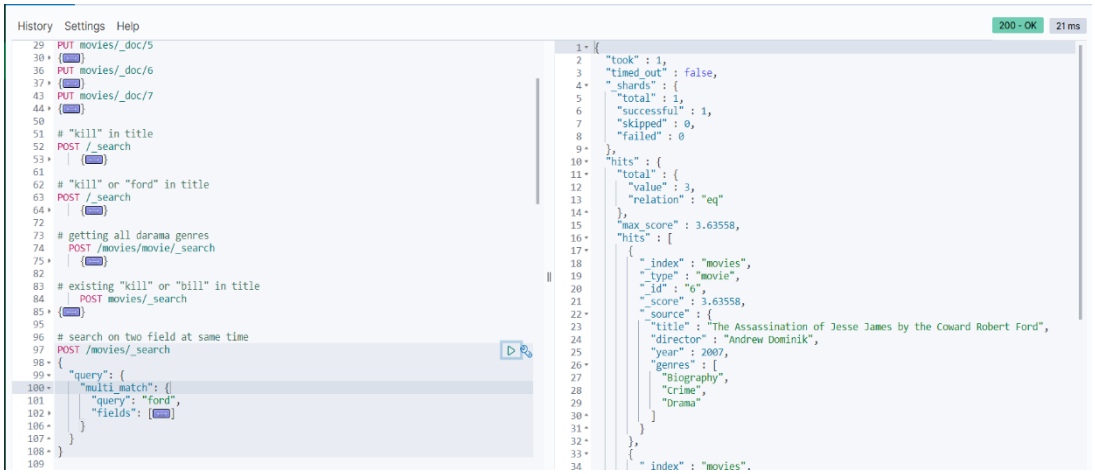
ساخت درخت جستجو:

فیلم‌های با ژانر drama را برمی‌گرداند

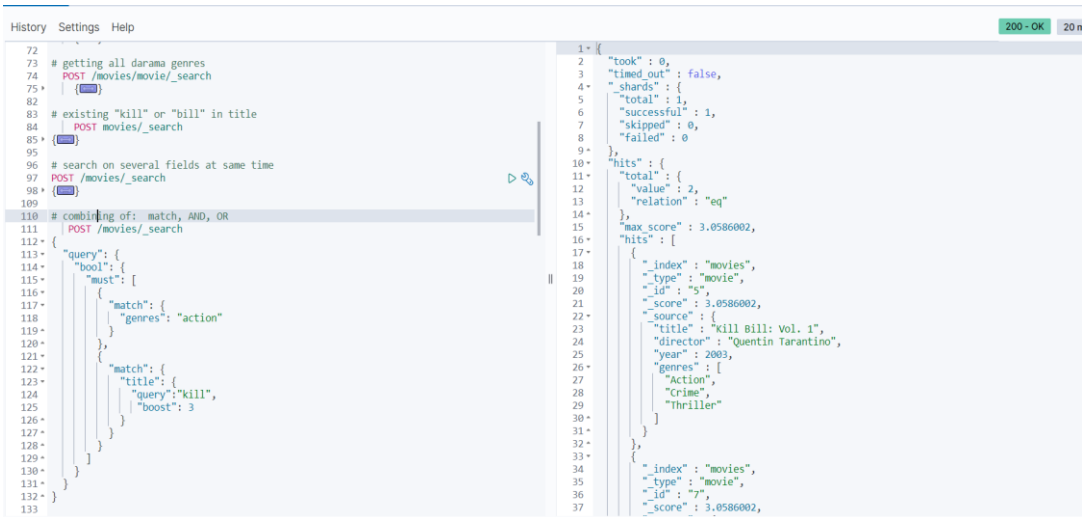


فیلم‌هایی که در title آن‌ها "kill" یا "bill" وجود دارد

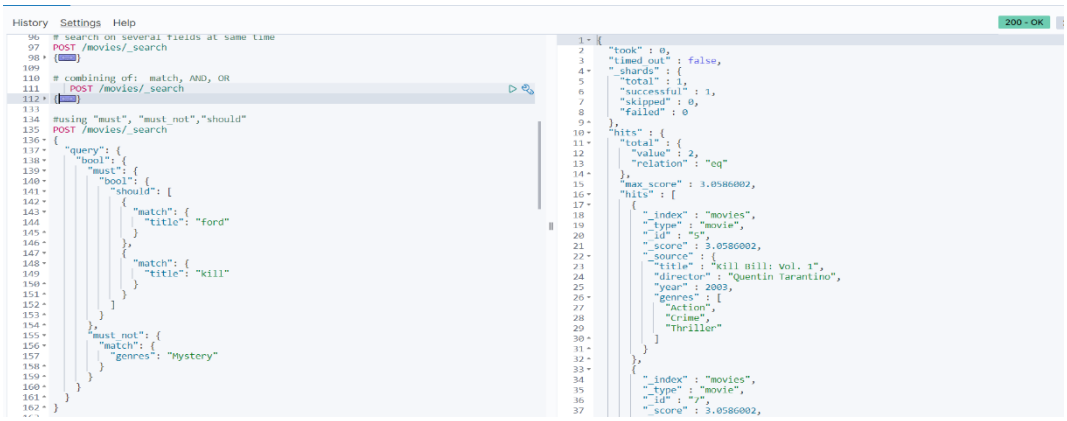
سرچ کردن همزمان روی دو یا چند فیلد:



فیلم‌های اکشن با عنوانی شامل bill برگردانده می‌شود:



ترکیب must, must_not, should:



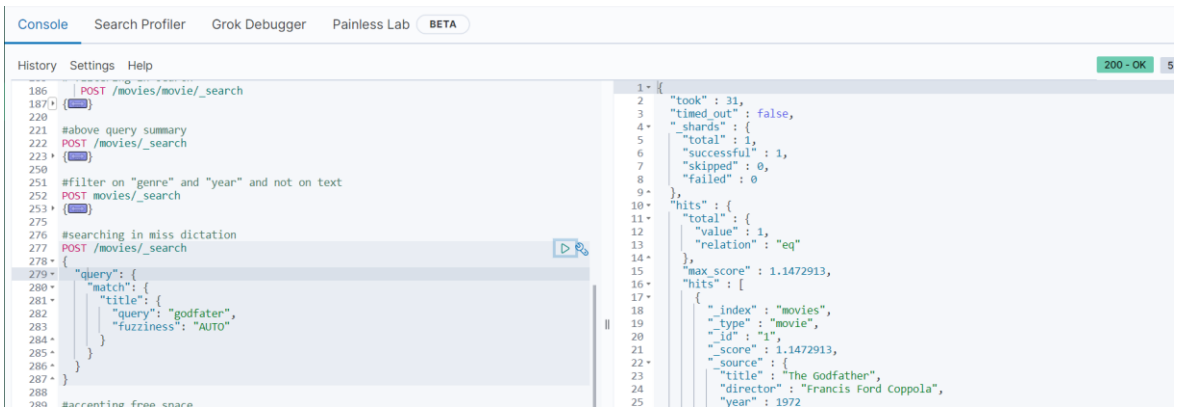
جستجوی فیلدهای غیر متنی:

اعمال فیلتر در جستجو:

کوئری بالا به شکل خلاصه تر:

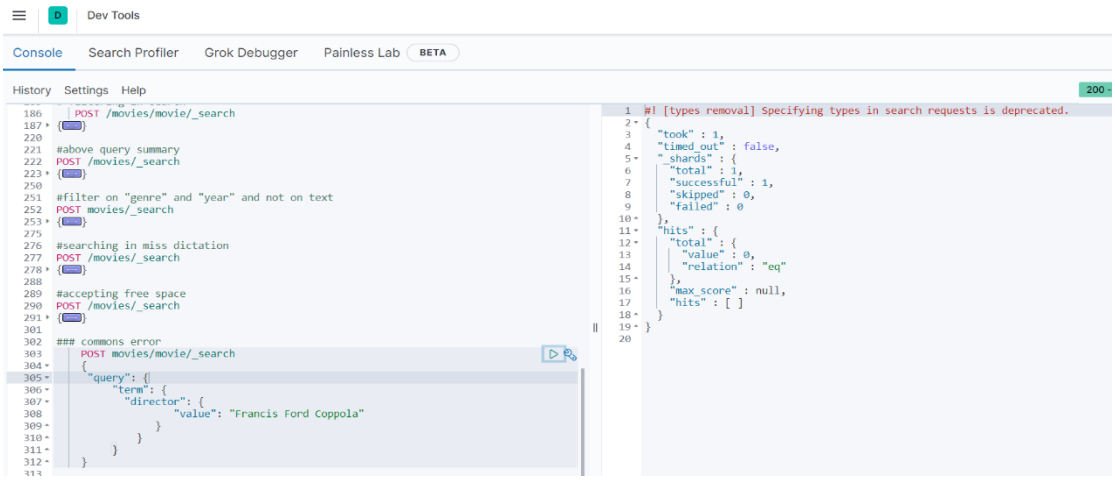
فیلتر کردن روی ژانر و سال بدون جستجوی متن:

سرچ کردن با در نظر گرفتن غلط دیکته ای:



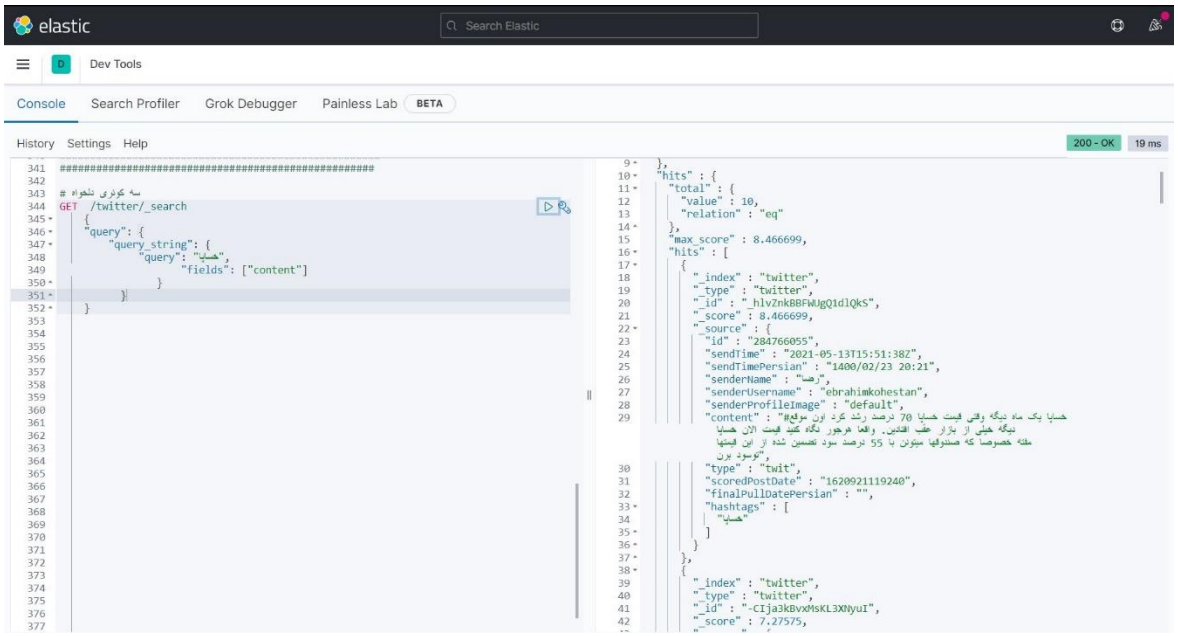
قبول کردن فظای خالی در کوئری:

اشتباه رایج:

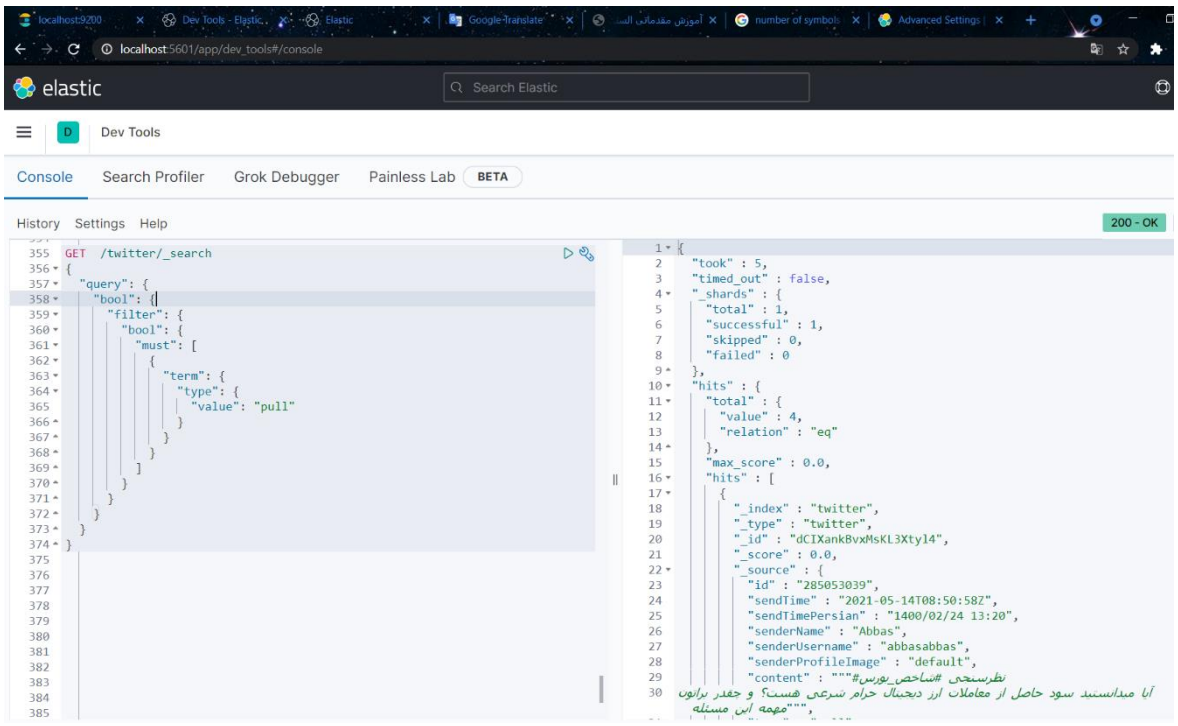


سه کوئری دلخواه روی داده های بورس:

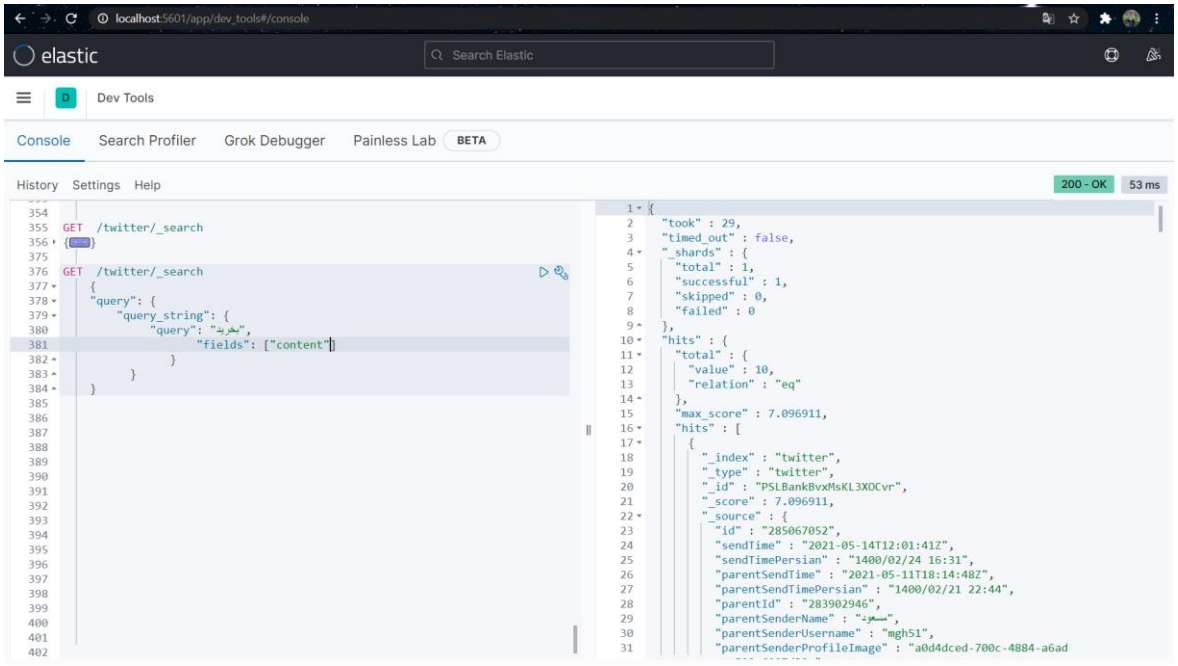
۱) در این کوئری داده هایی که در قسمت content سامل کلمه‌ی «خسابا» هستند را برمی‌گردانیم.



۲) داده هایی که type آن‌ها pull است را برمی‌گردانیم

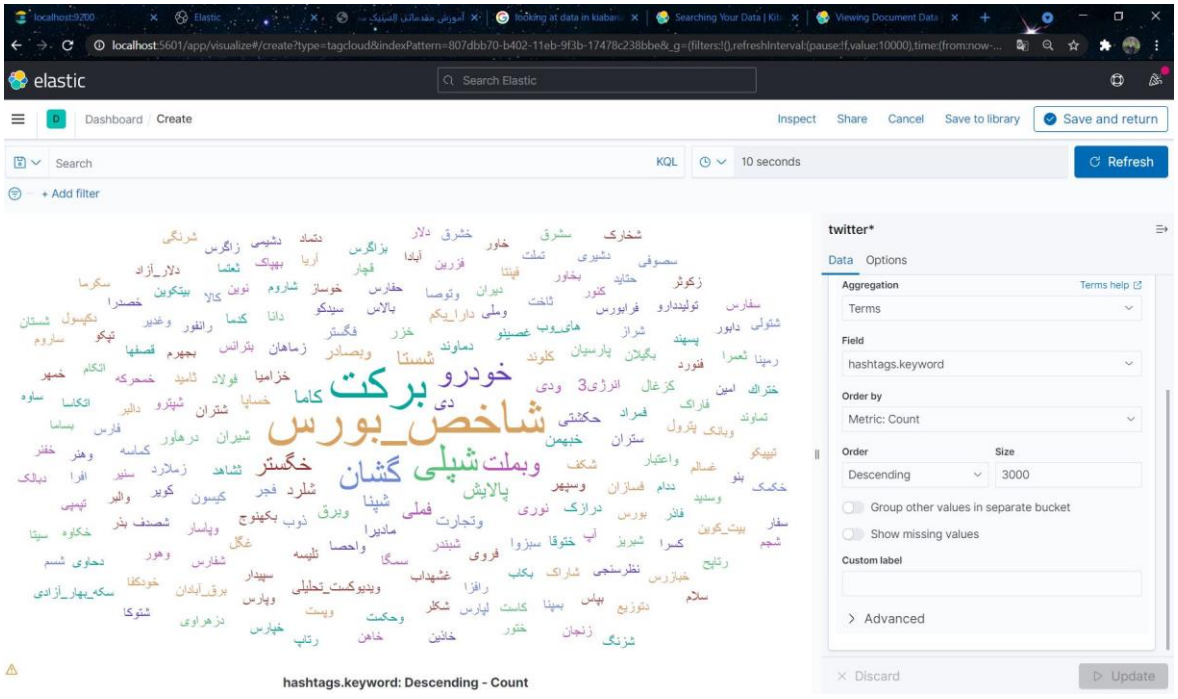


۳) توییت‌های که در content شامل کلمه «بخرید» می‌باشند را بازمی‌گردانیم



: Visualization

۱) ابر کلمات



۲)تعداد نمادهای پرتکرار

۳)تعداد کاربران با بیشترین توئیت