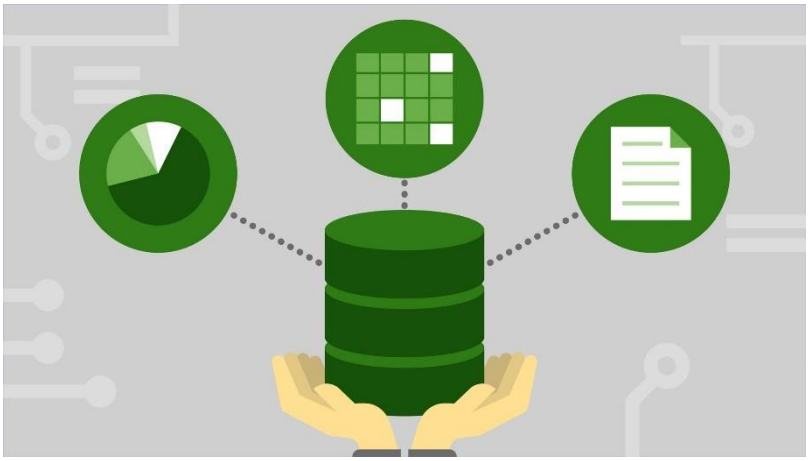
به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره ۶

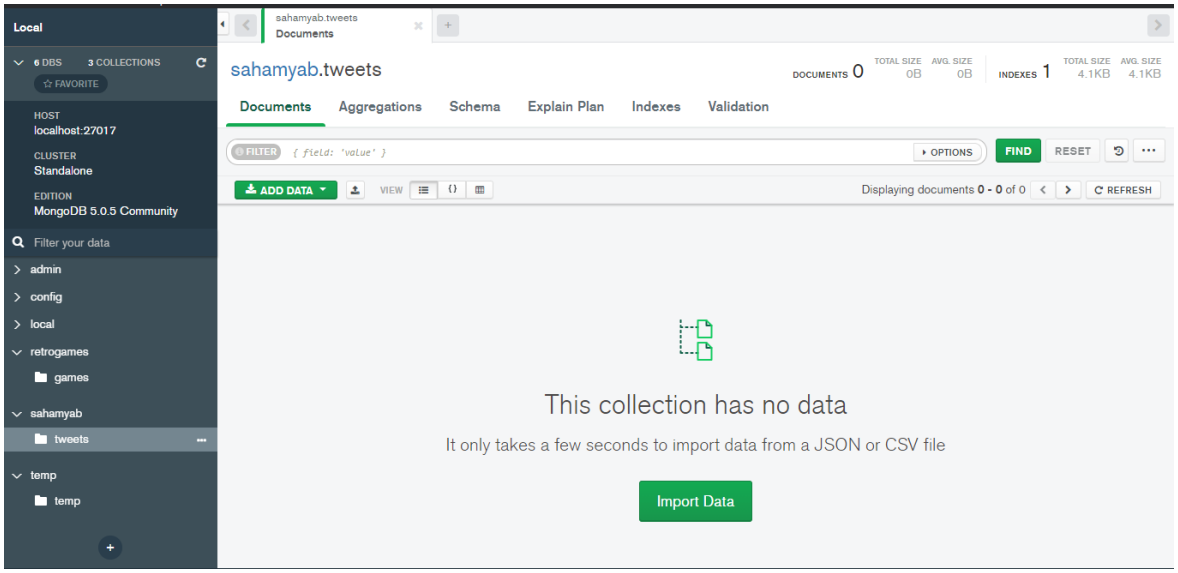
سجاد علی‌زاده
۸۱۰۱۹۷۵۴۷

آذرماه ۱۴۰۰

گزارش دستورکار انجام شده

نصب مانگو و ساختن کالکشن توییت‌ها

با استفاده از نرم‌افزار mongoDB compass دیتابیس sahamyab و کالکشن tweets را ساختیم.



گام اول تمرین

ابتدا ده توییت اول را با استفاده از کد زیر دریافت می‌کنیم:

```
import requests, json
response = requests.get('https://www.sahamyab.com/guest/twiter/list?v=0.1', headers={'User-Agent': 'Chrome/61'})
data = json.loads(response.text)
print(data)
with open('data.json', 'w', encoding='utf-8') as f:
    json.dump(data['items'], f, indent=2, ensure_ascii = False)
```

این توییت‌ها را در فایلی با نام data.json ذخیره می‌کنیم. حال این فایل را در دیتابیس ایمپورت می‌کنیم.

tweets

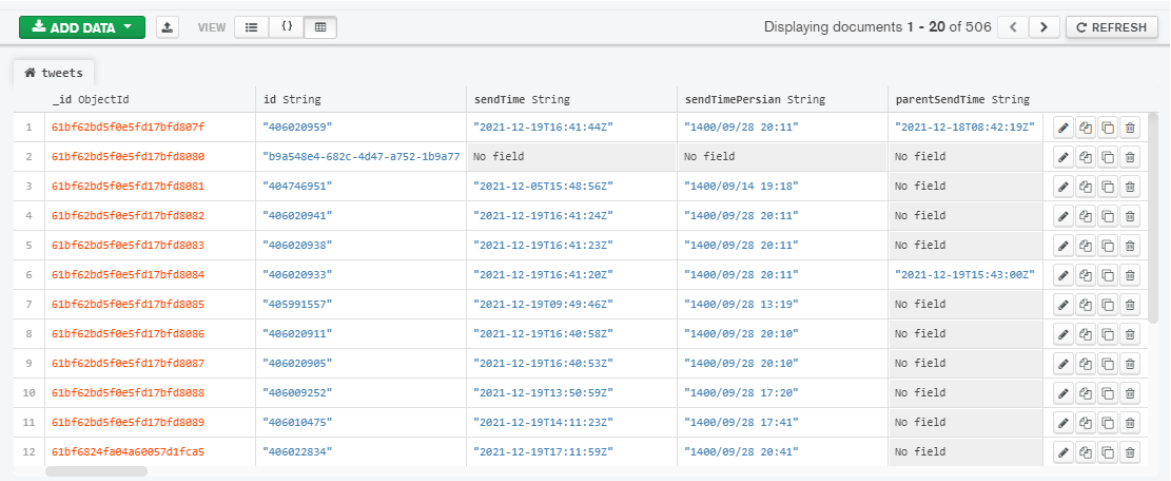
	_id ObjectId	id String	sendTime String	sendTimePersian String	parentSendTime String
1	61bf62bd5f0e5fd17bfd807f	"406020959"	"2021-12-19T16:41:44Z"	"1400/09/28 20:11"	"2021-12-18T08:42:19Z"
2	61bf62bd5f0e5fd17bfd8080	"b9a548e4-682c-4d47-a752-1b9a77	No field	No field	No field
3	61bf62bd5f0e5fd17bfd8081	"404746951"	"2021-12-05T15:48:56Z"	"1400/09/14 19:18"	No field
4	61bf62bd5f0e5fd17bfd8082	"406020941"	"2021-12-19T16:41:24Z"	"1400/09/28 20:11"	No field
5	61bf62bd5f0e5fd17bfd8083	"406020938"	"2021-12-19T16:41:23Z"	"1400/09/28 20:11"	No field
6	61bf62bd5f0e5fd17bfd8084	"406020933"	"2021-12-19T16:41:20Z"	"1400/09/28 20:11"	"2021-12-19T15:43:00Z"
7	61bf62bd5f0e5fd17bfd8085	"405991557"	"2021-12-19T09:49:46Z"	"1400/09/28 13:19"	No field
8	61bf62bd5f0e5fd17bfd8086	"406020911"	"2021-12-19T16:40:58Z"	"1400/09/28 20:10"	No field
9	61bf62bd5f0e5fd17bfd8087	"406020905"	"2021-12-19T16:40:53Z"	"1400/09/28 20:10"	No field
10	61bf62bd5f0e5fd17bfd8088	"406009252"	"2021-12-19T13:50:59Z"	"1400/09/28 17:20"	No field

همانطورکه مشاهده می‌شود این داده‌ها در دیتابیس ذخیره شده‌اند. همانطورکه مشاهده میشود فیلد _id ObjectId به آن اضافه شده است.

برای اضافه کردن ۵۰۰ توییت از قطعه کد زیر استفاده میکنیم:

```python
import time
import requests, json
from pymongo import MongoClient

client = MongoClient()
db = client.sahamyab
url = 'https://www.sahamyab.com/guest/twiter/list?v=0.1'
max_count = 500
interval = 1
counter = db.tweets.count_documents({})
while counter < max_count:
    response = requests.request('GET', url, headers={'User-Agent':'Chrome/61'})
    result = response.status_code
    data = response.json()['items']
    for d in data:
        try:
            db.tweets.insert_one(d)
        except Exception as e:
            print("Exception: " + str(e))
    counter = db.tweets.count_documents({})
    print(f'Fetched tweets: {counter}')
    time.sleep(interval)
```

در این قطعه کد تا جایی که تعداد توییت‌ها به بیش از ۵۰۰ برسد دریافت ادامه می‌یابد و در دیتابیس ذخیره می‌شود. در نهایت برای اینکه چک کنیم چه تعداد توییت در دیتابیس وجود دارد از compass نگاه میکنیم:



	_id ObjectId	id String	sendTime String	sendTimePersian String	parentSendTime String
1	61bf62bd5f0e5fd17bfd807f	"406020959"	"2021-12-19T16:41:44Z"	"1400/09/28 20:11"	"2021-12-18T08:42:19Z"
2	61bf62bd5f0e5fd17bfd8080	"b9a548e4-682c-4d47-a752-1b9a77	No field	No field	No field
3	61bf62bd5f0e5fd17bfd8081	"404746951"	"2021-12-05T15:48:56Z"	"1400/09/14 19:18"	No field
4	61bf62bd5f0e5fd17bfd8082	"406020941"	"2021-12-19T16:41:24Z"	"1400/09/28 20:11"	No field
5	61bf62bd5f0e5fd17bfd8083	"406020938"	"2021-12-19T16:41:23Z"	"1400/09/28 20:11"	No field
6	61bf62bd5f0e5fd17bfd8084	"406020933"	"2021-12-19T16:41:20Z"	"1400/09/28 20:11"	"2021-12-19T15:43:00Z"
7	61bf62bd5f0e5fd17bfd8085	"405991557"	"2021-12-19T09:49:46Z"	"1400/09/28 13:19"	No field
8	61bf62bd5f0e5fd17bfd8086	"406020911"	"2021-12-19T16:40:58Z"	"1400/09/28 20:10"	No field
9	61bf62bd5f0e5fd17bfd8087	"406020905"	"2021-12-19T16:40:53Z"	"1400/09/28 20:10"	No field
10	61bf62bd5f0e5fd17bfd8088	"406009252"	"2021-12-19T13:50:59Z"	"1400/09/28 17:20"	No field
11	61bf62bd5f0e5fd17bfd8089	"406010475"	"2021-12-19T14:11:23Z"	"1400/09/28 17:41"	No field
12	61bf6824fa04a60057d1fca5	"406022834"	"2021-12-19T17:11:59Z"	"1400/09/28 20:41"	No field

مشاهده میکنیم ۵۰۶ داده در دیتابیس وجود دارد. از طریق کامند لاین نیز میتوانیم ببینیم:

```
> db.tweets.count()
506
```

البته در کد نوشته شده مشکلی وجود دارد و این است که چک نمیکنیم آیا توییتی که گرفته شده از قبل وجود داشته یا خیر. بنابراین مقدار زیادی توییت تکراری ذخیره شده است. در صورتی که میخواستم توییت‌های تکراری را ایگنور کنم همواره ۱۱ توییت دریافت میشد و مثلا هر ۳۰ دقیقه یک بار ۱۰ توییت جدید اضافه میشد. بنابراین برای ادامه سوالات فایلی که یکی از دوستان به اشتراک گذاشت را ایمپورت کردم.

گام دوم تمرین

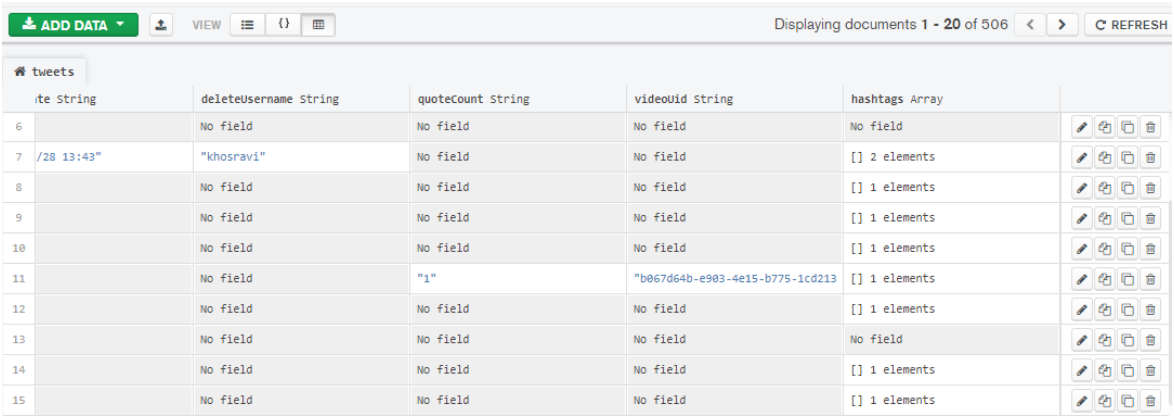
با استفاده از کد زیر کار گفته شده را انجام می‌دهیم:

```
import time
import requests, json
from pymongo import MongoClient
from bson.objectid import ObjectId

client = MongoClient()
db = client.sahamyab

tic = time.time()
result = db.tweets.find({"content": { "$regex" : "#" }})
for tweet in result:
    hashtags = []
    splited_tweet = tweet['content'].split()
    for word in splited_tweet:
        if word[0] == '#':
            hashtags.append(word)
    db.tweets.update_one(
        {'_id': tweet['_id']},
        {'$set': {'hashtags': hashtags}}
    )
toc = time.time()
print("time:", toc - tic)
```

در این کد ابتدا با استفاده از regex توییت‌هایی که در آن‌ها # است را می‌یابیم و در result میریزیم. سپس روی این توییت‌ها پیمایش می‌کنیم و به ازای هر توییت آن را split میکنیم و دوباره روی کلمات آن پیمایش می‌کنیم. حال اگر ابتدای یک کلمه # باشد یعنی آن کلمه هشتگ است و به لیست هشتگ‌ها اضافه می‌کنیم. در نهایت نیز این لیست را به عنوان یک فیلد به آن توییت اضافه میکنیم. زمان کد به شکل زیر است:

```
PS C:\> & "c:/program files/python/python39/python.exe" c:/Users/Home/Desktop/DBLAB/LAB6/3.py
time: 0.4280688762664795
```

اگر به داده‌ها نیز نگاهی بیندازیم مشاهده می‌کنیم فیلد hashtags به آن اضافه شده است:



	te String	deleteUsername String	quoteCount String	videoUid String	hashtags Array
6		No field	No field	No field	No field
7	/28 13:43"	"khosravi"	No field	No field	[] 2 elements
8		No field	No field	No field	[] 1 elements
9		No field	No field	No field	[] 1 elements
10		No field	No field	No field	[] 1 elements
11		No field	"1"	"b067d64b-e903-4e15-b775-1cd213	[] 1 elements
12		No field	No field	No field	[] 1 elements
13		No field	No field	No field	No field
14		No field	No field	No field	[] 1 elements
15		No field	No field	No field	[] 1 elements

به طور خاص اگر به یکی از توییت‌ها نگاهی بیندازیم:


```
_id: ObjectId("61bf62bd5f0e5fd17bfd8081")
id: "404746951"
sendTime: "2021-12-05T15:48:56Z"
sendTimePersian: "1400/09/14 19:18"
retwitSendTime: "2021-12-19T16:41:40Z"
retwitSendTimePersian: "1400/09/28 20:11"
retwitSenderName: "جعفر حسنخانی"
retwitSenderUsername: "zq10585"
retwitSenderProfileIm… : "default"
senderName: "جعفر حسنخانی"
senderUsername: "zq10585"
senderProfileImage: "default"
content: "#تهران"
lastLikeNickName: "مهرجو"
likeCount: "1"
retwitCount: "1"
type: "retwit"
imageUid: "2fec2401-8351-4f9f-bee9-b95052498774"
mediaContentType: "image/jpeg"
scoredPostDate: "1638719987370"
retwitId: "406020954"
finalPullDatePersian: ""
hashtags: Array
  0: "#تهران"
```


مشاهده میشود لیست هشتگ‌های آن به درستی آپدیت شده است.

گام سوم تمرین:

سوال اول)

```
import time
import requests, json
from pymongo import MongoClient
from bson.objectid import ObjectId

client = MongoClient()
db = client.sahamyab

tic = time.time()
result = db.tweets.find({ 'mediaContentType': 'image/jpeg', 'parentId': {'$ne': None} })
for tweet in result:
    print(tweet['senderName'])
toc = time.time()
print("time:", toc - tic)
```

در قسمت find شروط گفته شده را اعمال کردیم. یکی از شروط این است که نوع مدیا jpeg باشد و دومی این است که مقدار parentId برای آن غیر None باشد (با استفاده از ne). بعد از آن برای توییت‌های به دست آمده senderName را چاپ کردیم. نتیجه به شرح زیر است:

```
PS C:\> & "c:/program files/python/python39/python.exe" c:/Users/Home/Desktop/DBLAB/LAB6/4.py
یفطمم
سروب 97 یدورو
ن ابرقن اج ن امیپ
Mohsen
بارمم
m
دمحم
شویراد
trader
AAAA
یکدام انیهن
shahramdehqanii
رگبامرح
SM.SADAT
AAAA
Mster majid
ن ایدومحم
Mahsa
زیزع ن اریا یم اج
/o^o\
یمیرک اقیر
سیون مم انزب یدمم
time: 0.03599977493286133
PS C:\>
```

مشاهده میشود ۲۲ نتیجه در زمان ۰/۰۳ ثانیه به دست آمده است.

من متوجه شدم از خود compass نیز میتوان زمان اجرا را دید. بنابراین از این به بعد به جای کد پایتون در compass کوئری‌ها را اجرا میکنم.

سوال دوم)

کوئری را به شکل زیر مینویسیم:

کوئری‌هایی را میخواهیم که sendTime آنها در یک بازه ۱۵ دقیقه‌ای باشند. مقادیر senderName و Type را برای این توییت‌ها چاپ می‌کنیم:

tweets

	_id ObjectId	senderName String	type String
1	61bf93f55f0e5fd17bfd808c	" بینا"	"quote"
2	61bf93f55f0e5fd17bfd808f	"حمایت بکاب"	"twit"
3	61bf93f55f0e5fd17bfd8090	"milad"	"twit"
4	61bf93f55f0e5fd17bfd8091	"دکتر مرادی اتحاد"	"quote"
5	61bf93f55f0e5fd17bfd8093	"حبیب"	"twit"
6	61bf93f55f0e5fd17bfd8094	"Moments"	"twit"

مشاهده می‌شود در این بازه شش نفر توییت زده‌اند. زمان اجرای آن به شرح زیر است:

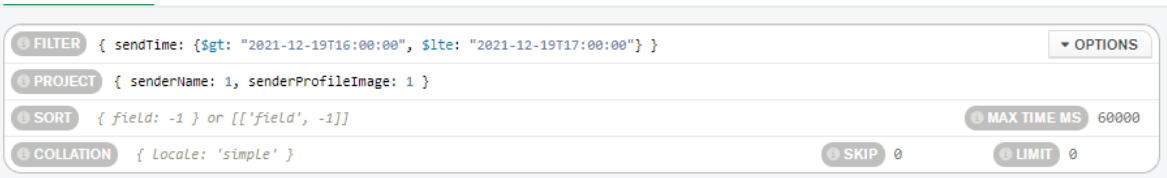
Query Performance Summary

- Documents Returned: **6**
- Actual Query Execution Time (ms): **1**
- Index Keys Examined: **0**
- Sorted in Memory: **no**
- Documents Examined: **543**
- ⚠ No index available for this query.

در ۱ میلی ثانیه این کوئری اجرا شده است.

سوال سوم)

کوئری این سوال را به شکل زیر می‌نویسیم:



داده‌هایی را میخواهیم که تایم آنها در یک بازه یک ساعته در یک روز باشد. ما بازه ۱۶ تا ۱۷ را انتخاب کردیم. از این داده‌ها senderName و senderProfileImage را نشان میدهیم. نتیجه به شکل زیر است:

tweets

	_id ObjectId	senderName String	senderProfileImage String
1	61bf93f55f0e5fd17bfd808c	" بینا"	"default"
2	61bf93f55f0e5fd17bfd808f	"حمایت بکاب"	"default"
3	61bf93f55f0e5fd17bfd8090	"milad"	"default"
4	61bf93f55f0e5fd17bfd8091	"دکتر مرادی اتحاد"	"1976fa08-72f8-47e4-bcea-b8c379
5	61bf93f55f0e5fd17bfd8093	"حبیب"	"default"
6	61bf93f55f0e5fd17bfd8094	"Moments"	"dfdcd759-8452-421b-835c-5d5e21

همانطور که مشاهده می‌شود شش داده به دست آمده است. البته صورت سوال برای من ابهام داشت. اگر منظور حداقل داشتن حداقل یک توییت است که نتیجه بالا پاسخ مسئله است. اما اگر منظور داشتن بیشتر مساوی ۲ توییت است باید گفت این کوئری جواب ندارد زیرا هیچ دو نامی از نتیجه بالا یکی نیست. زمان اجرا نیز به شکل زیر است:

Query Performance Summary

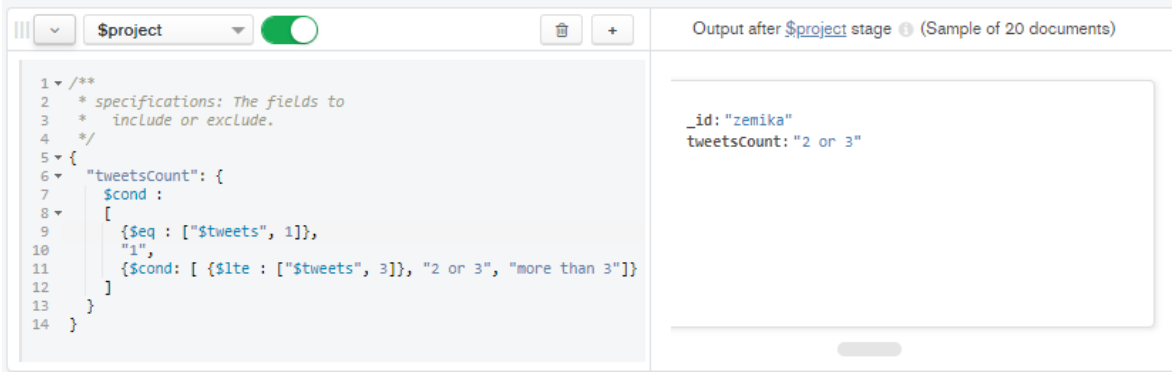
- Documents Returned: **6**
- Actual Query Execution Time (ms): **1**
- Index Keys Examined: **0**
- Sorted in Memory: **no**
- Documents Examined: **543**
- ⚠ No index available for this query.

یعنی این کوئری ۱ میلی ثانیه طول کشیده است.

گام چهارم تمرین:

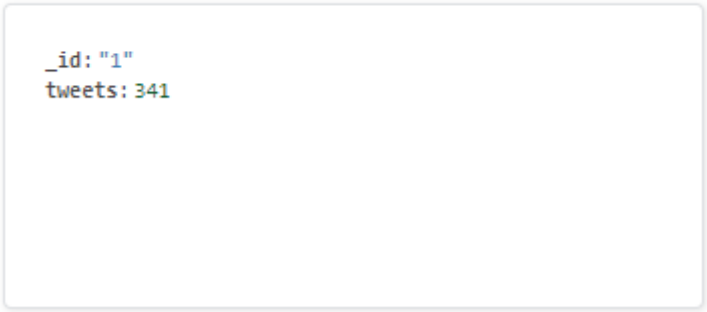
سوال اول) این مسئله را با روند زیر حل می‌کنیم:

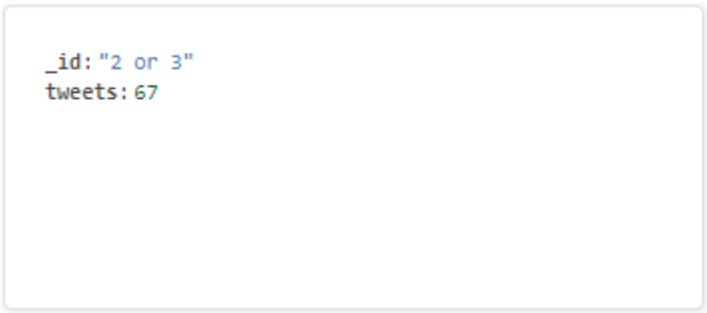
در این استیج با یک group به ازای هر یوزرنیم تعداد توییت‌های آن یوزر را به دست می‌آوریم.



در این استیج با استفاده از cond یک بازنویسی انجام میدهیم. اگر مقدار tweets که در استیج قبل به دست آوردیم یک بود مقدار tweetsCount برابر ۱ میشود. در غیر این صورت اگر مقدار tweets کمتر از ۳ باشد (که از آنجایی که بیشتر از یک نیز هست پس مقادیر ۲ و ۳ قابل قبول می‌شود) مقدار tweetsCount برابر 2 or 3 می‌شود. در نهایت نیز اگر بیشتر از ۳ باشد مقدار tweetsCount برابر more than 3 می‌شود.

در نهایت یک گروه‌بندی روی tweetsCount انجام می‌دهیم تا تعداد هر گروه به دست آید. نتیجه به شرح زیر است:

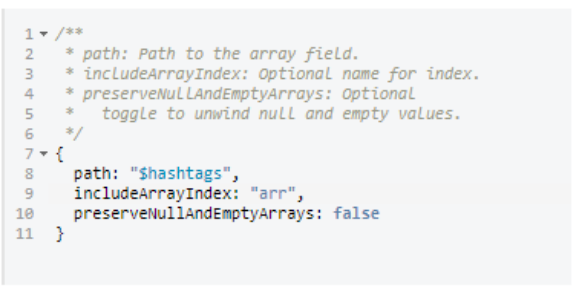




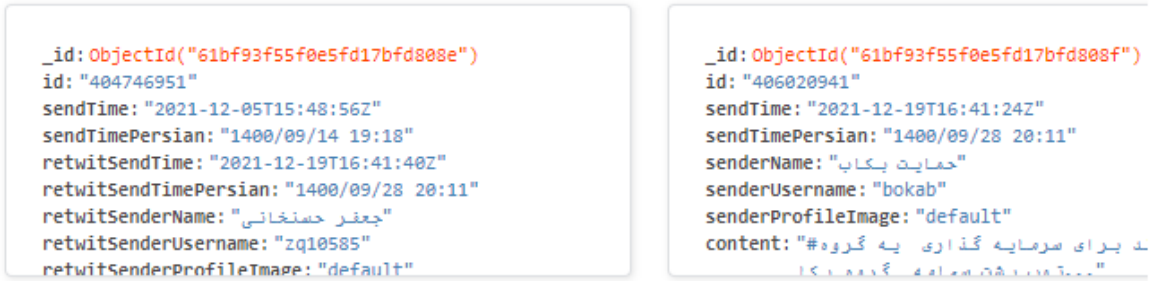
یعنی ۳۴۱ اکانت وجود دارند که یک توییت دارند. ۶۷ اکانت ۲ یا ۳ توییت دارند و ۱۱ اکانت بیش از ۳ توییت دارند.

سوال دوم)

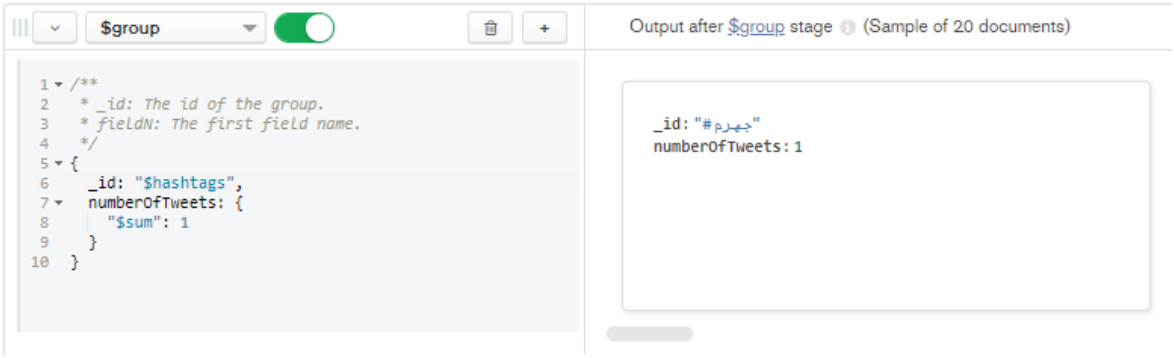
این مسئله را با روند زیر حل می‌کنیم:



از unwind استفاده کردیم. در اینجا چون آرایه‌ای به نام hashtags داریم با unwind کردن به ازای هر کدام از مقادیر این آرایه یک آبجکت جدید تولید می‌شود. البته توجه کنید آیدی تمام این مقادیر تولید شده یکسان است. همچنین مقدار preserveNullEmptyArrays را فالس کردیم تا اگر آرایه‌ای خالی یا نال بود بسط داده نشود. به عنوان مثال:

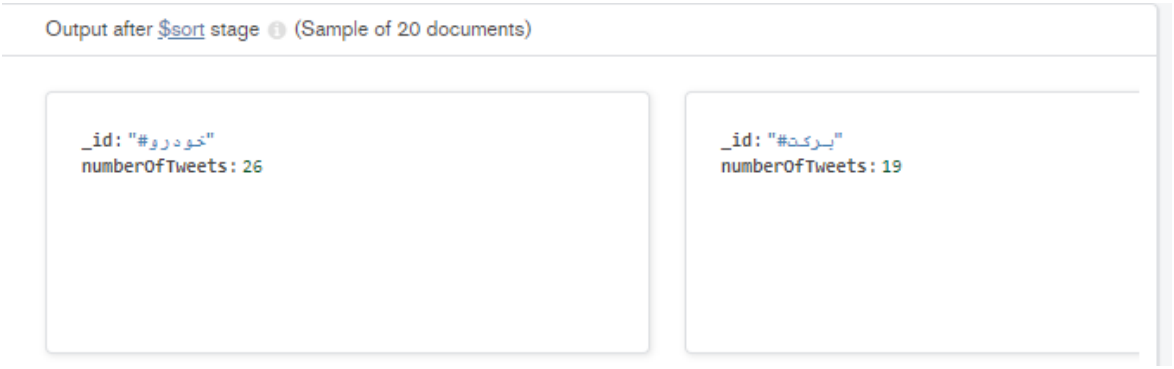


همانطورکه میبینید آیدی دو مقدار برابر است. حال به ادامه حل میپردازیم:

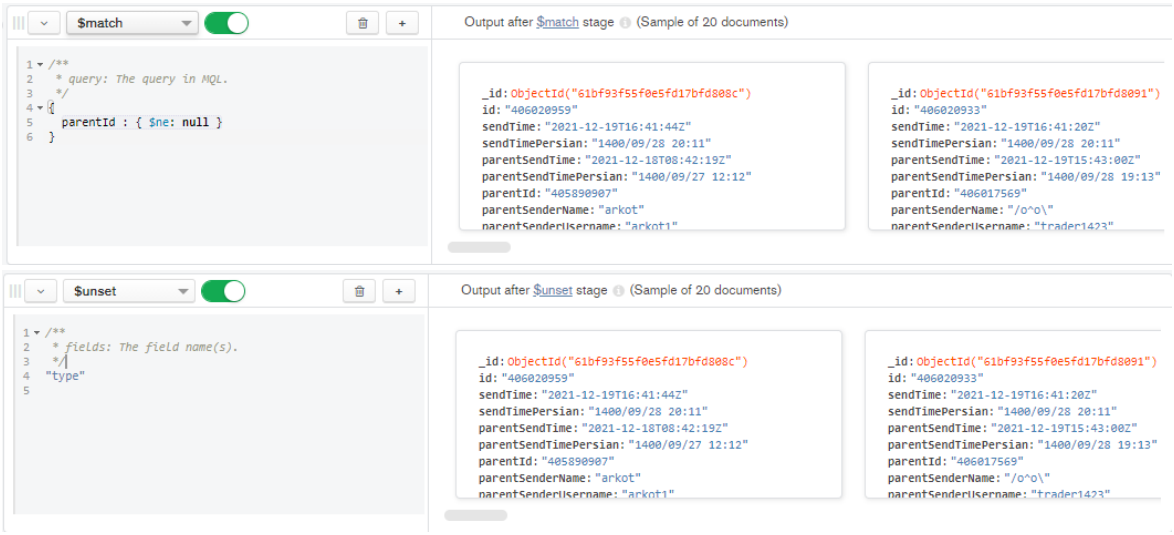


حال بر اساس مقدار هشتگ گروه‌بندی می‌کنیم. به ازای هر گروه مجموع آن را به دست می‌آوریم. یعنی در اصل به ازای هر هشتگ مجموع تعداد آن را به دست آوردیم و در numberOfTweets ریختیم.

در نهایت نیز بر اساس مقدار numberOfTweets به صورت نزولی مرتب میکنیم. همانطورکه مشاهده میشود بیشترین توییت مربوط به #شاخص_بورس است. دو مقدار بعدی به شکل زیر است:



سوال سوم)



در ابتدا با استفاده از match کوئری‌هایی را پیدا کردیم که مقدار parentId برای آنها غیرمساوی با null باشد. سپس در استیج بعدی مقدار type را برای آن unset می‌کنیم.

سوال چهارم)

از قسمت دوم استفاده می‌کنیم. ابتدا تعداد تکرار تمام هشتگ‌ها را به دست می‌آوریم و سپس آن را به صورت نزولی سورت میکنیم. هشتگ اول پرتکرارترین است. دو استیج اول که مثل سوال اول است:

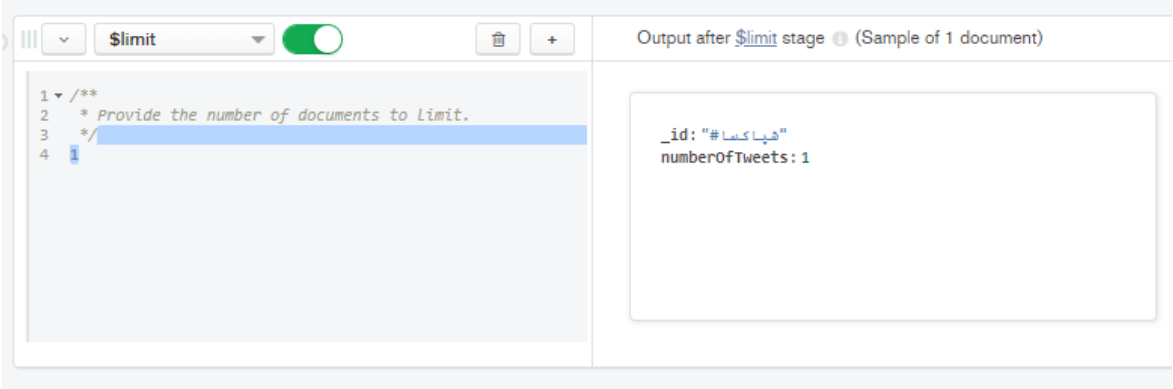
حال داده‌ها را به صورت نزولی مرتب می‌کنیم:

در نهایت با استفاده از limit فقط داده اول را نمایش می‌دهیم:

مشاهده می‌شود پرتکرارترین توییت #شاخص_بورس با تعداد تکرار ۴۵ است. برای به دست آوردن کم‌تکرارترین هشتگ داده‌ها را به صورت صعودی مرتب میکنیم در این صورت کم تکرارترین هشتگ در مکان اول قرار میگیرد.

یعنی بعد از به دست آوردن تعداد توییت‌ها آن‌ها را صعودی مرتب می‌کنیم:

در نهایت با استفاده از limit فقط داده اول را نمایش می‌دهیم:

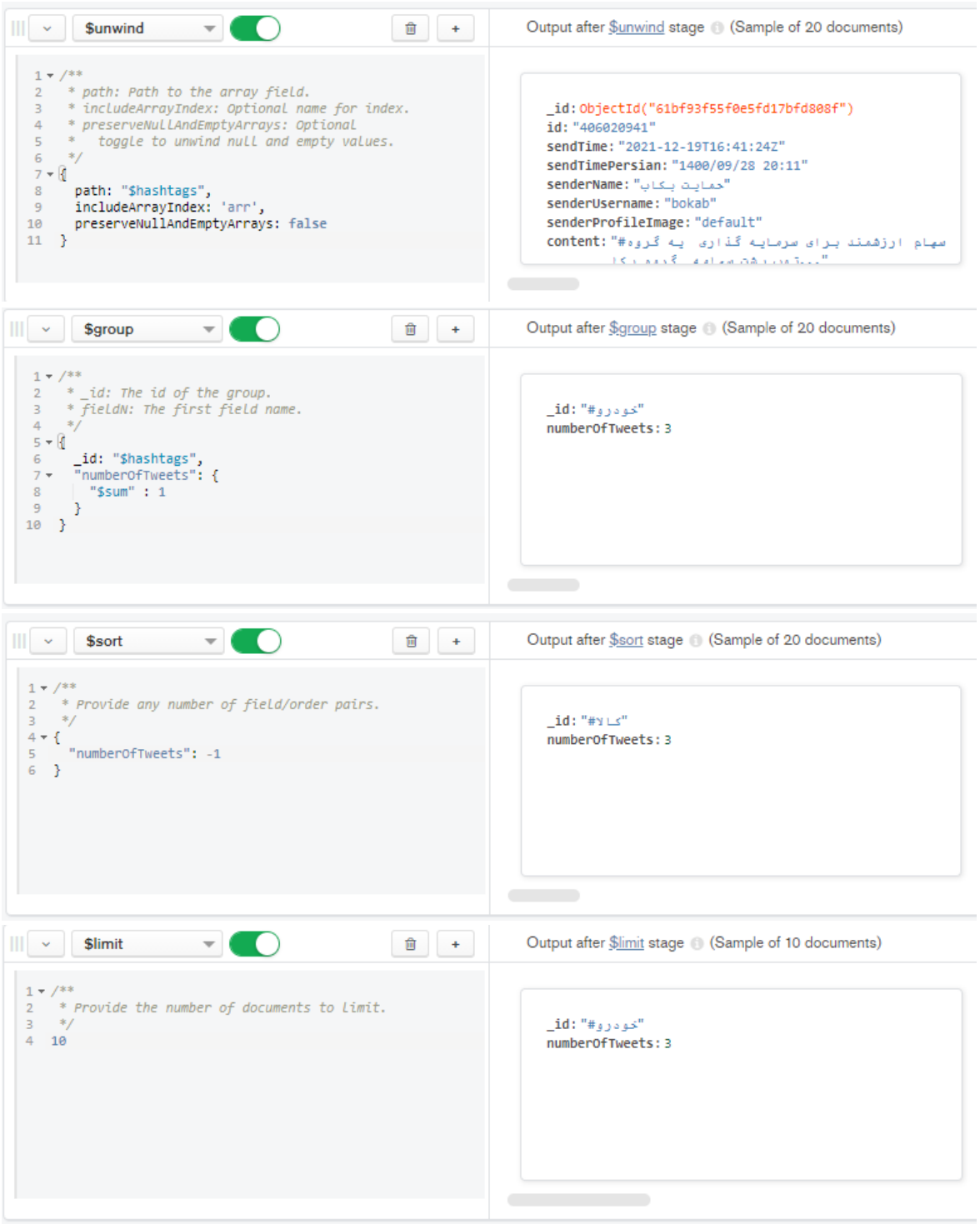


مشاهده می‌شود #شپاکنسا کم‌تکرارترین هشتگ با تعداد ۱ تکرار است.

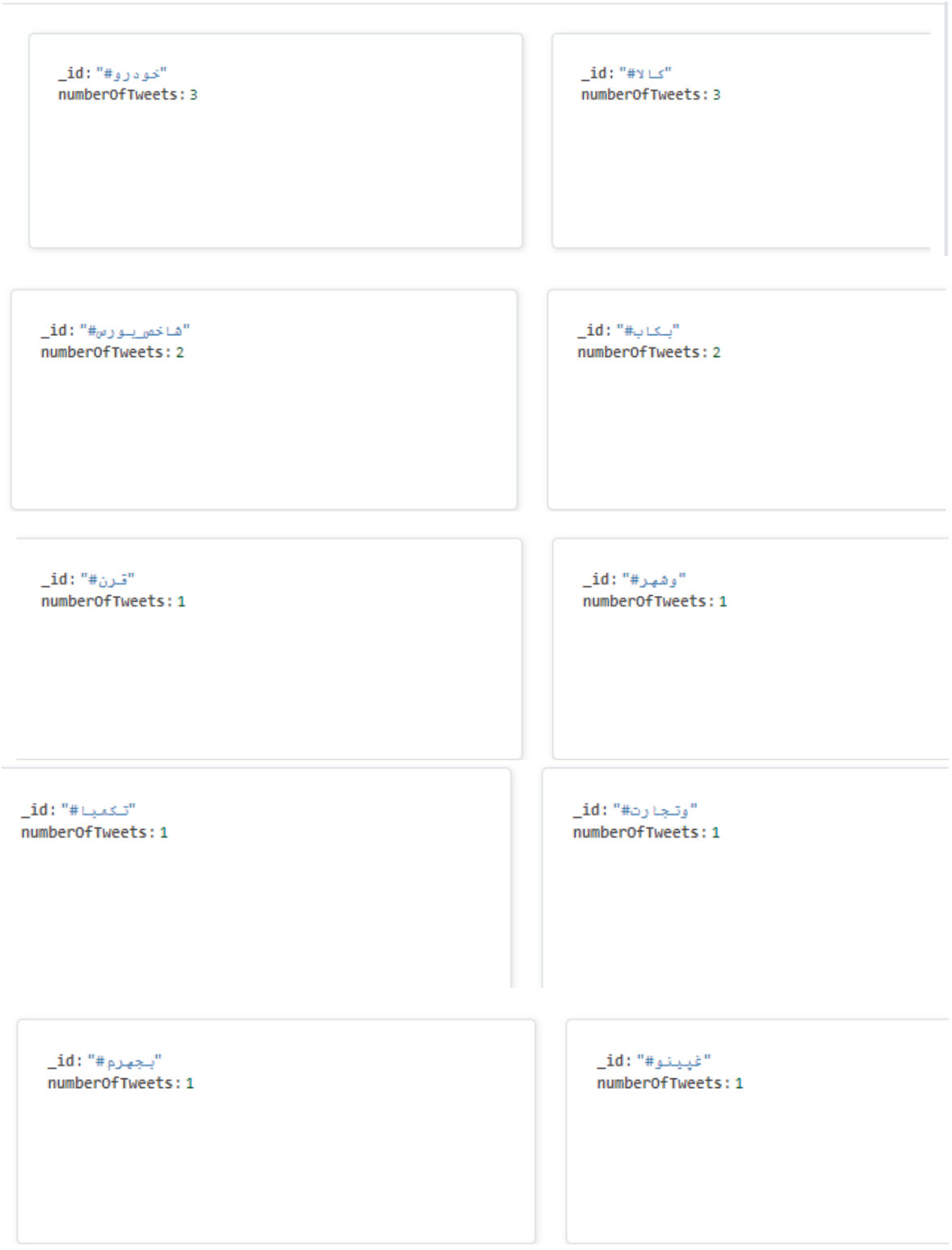
سوال پنجم)

در این مسئله ابتدا توییت‌های بازه زمانی مورد نظر را می‌یابیم:

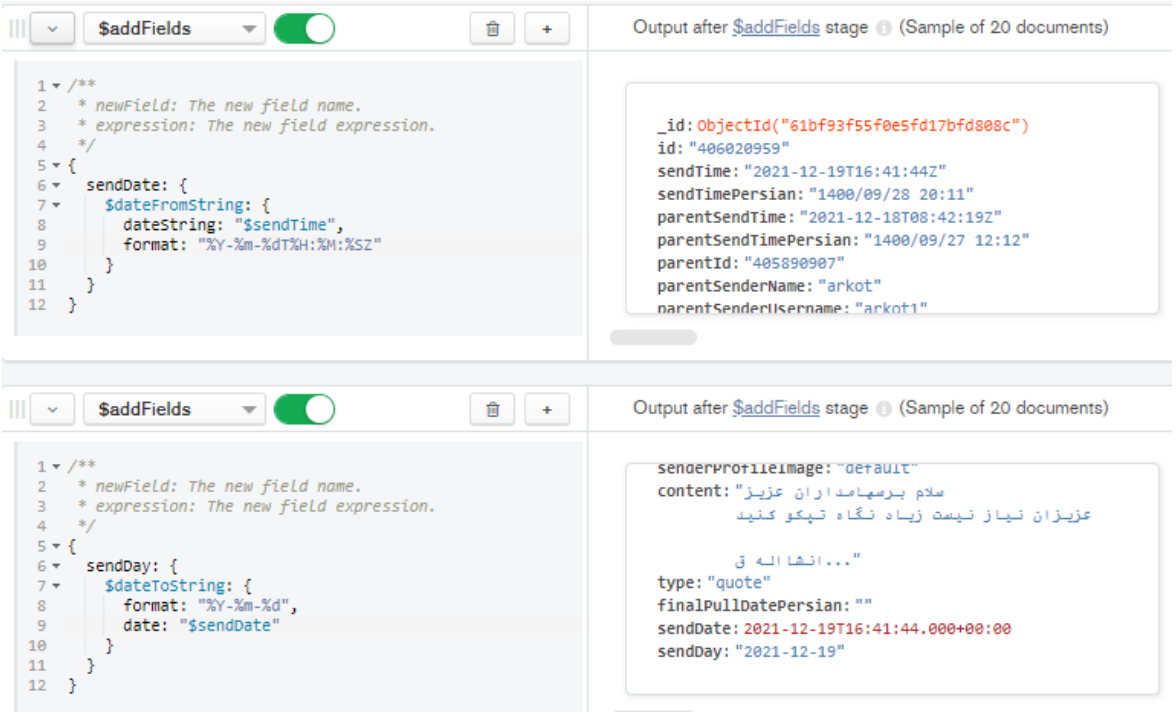
در اینجا توییت‌هایی را انتخاب کرده‌ایم که در روز ۲۰۲۱/۱۲/۱۹ زده شده‌اند. از این مرحله به بعد دقیقا مانند سوال قبل است.



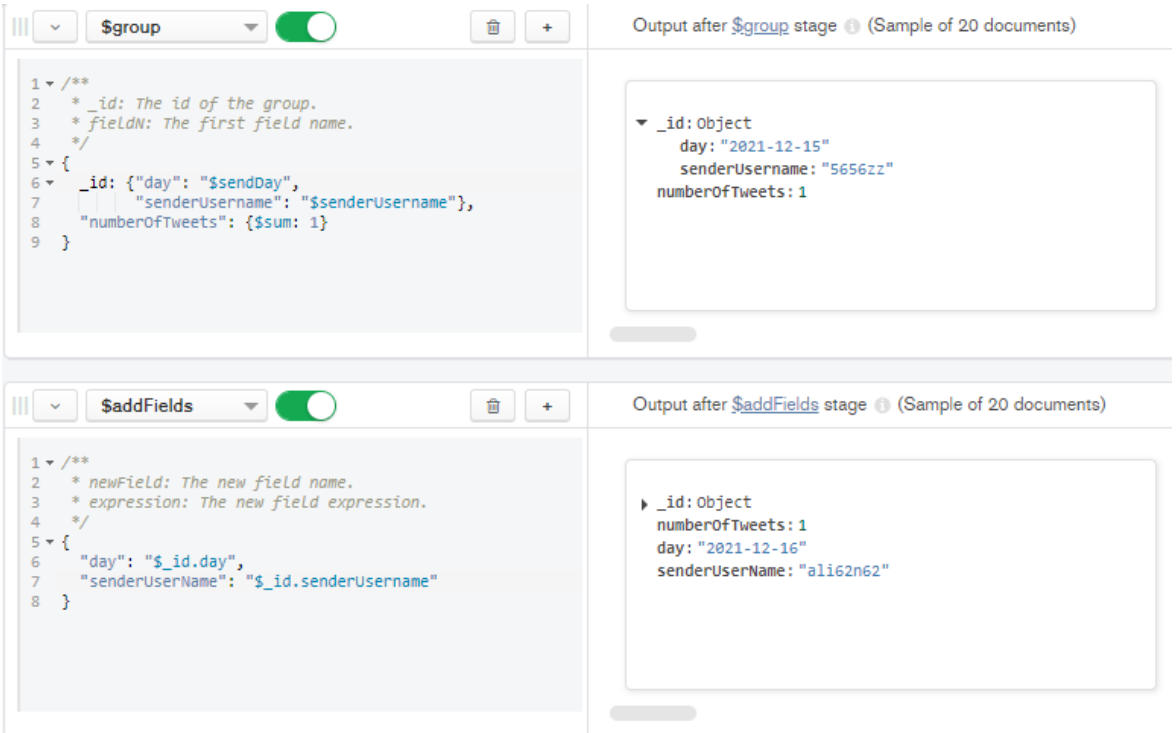
ابتدا با استفاده از unwind آرایه هشتگ‌ها را باز کردیم. سپس با استفاده از group مجموع استفاده از هر هشتگ را محاسبه کردیم. سپس آن‌ها را به صورت نزولی مرتب کردیم و در نهایت ۱۰ مقدار اول را نمایش دادیم. خروجی به شکل زیر است:


_id: "#خودرو"
numberOfTweets: 3
_id: "#کالا"
numberOfTweets: 3
_id: "#شاخص_بورس"
numberOfTweets: 2
_id: "#بکاب"
numberOfTweets: 2
_id: "#قرن"
numberOfTweets: 1
_id: "#وشهر"
numberOfTweets: 1
_id: "#تکمبا"
numberOfTweets: 1
_id: "#وتجارت"
numberOfTweets: 1
_id: "#بجهرم"
numberOfTweets: 1
_id: "#غپینو"
numberOfTweets: 1

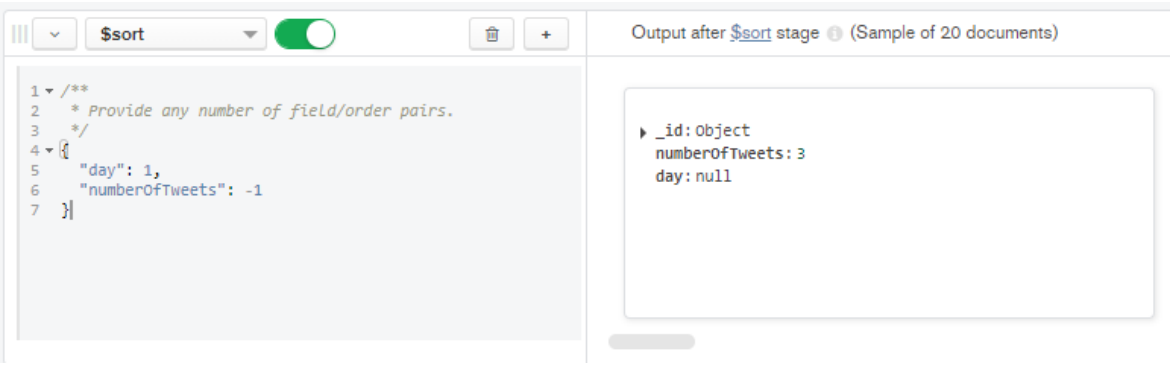
سوال ششم)



همانطور که مشاهده می‌شود ابتدا با دوبار addFields توانستیم تاریخ ارسال توییت را استخراج کنیم و در فیلد sendDay بریزیم.



سپس بر اساس این تاریخ و یوزرنیم ارسال‌کننده توییت‌ها را گروه‌بندی کردیم و به ازای هر گروه تعداد توییت‌ها را محاسبه کردیم. با این کار تعداد توییت‌های هر یوزر در هر روز را داریم. در مرحله بعد مقادیر تاریخ و یوزرنیم را به عنوان فیلد اضافه کردیم تا در مراحل بعد راحت‌تر باشیم.



در استیج بعد داده‌ها را بر اساس روز_تعداد توییت مرتب کردیم. در نتیجه به ازای هر روز، یوزر با بیشترین توییت در ابتدای داده‌های آن روز نشان داده می‌شود. در استیج بعدی داده‌ها را بر اساس روز ارسال توییت گروه بندی کردیم و اولین داده هر روز را با استفاده از $first استخراج کردیم. با این کار، فردی که در داده اول مربوط به آن روز آمده بود، یعنی همان فردی که بیشترین تعداد توییت را داشته، در این فیلد نمایش داده می‌شود که همان هدف ماست. یعنی به ازای هر روز کاربری که بیشترین توییت را در آن روز زده داریم. برخی خروجی‌ها به شکل زیر هستند:

مشکلات و توضیحات تکمیلی

برای گرفتن توییت‌ها مشکل داشتم و فقط ۱۱ توییت جدید دریافت می‌شد. هر سی دقیقه یکبار ده توییت جدید دریافت می‌شد و عملیات دریافت بسیار زمان‌بر بود. برای همین فایل جی‌سان را از یکی از دوستان گرفتم و به انجام گزارش‌کار پرداختم.

آنچه آموختم / پیشنهادات

فوق العاده آموزنده و زیبا بود. سپاس فراوان.