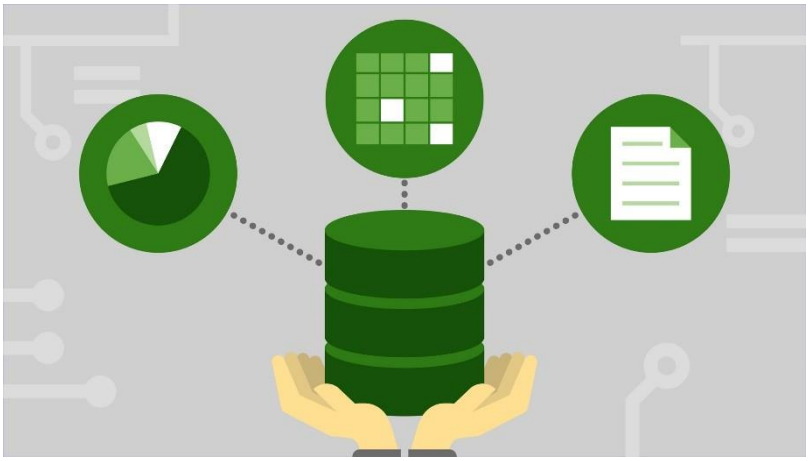
به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره ۶

نام و نام خانوادگی
معین شیردل ۸۱۰۱۹۷۵۳۵

آبان ماه ۱۴۰۰

گام اول- import کردن داده در دیتابیس Mongo:

ابتدا به صورت دستی و با دستور db.tweets.save(tweet_i) ۱۰ تا توییت به صورت دستی ذخیره شد و سپس، به خاطر نیاز به داشتن دقیقا ۵۰۰ توییت، این ۱۰ توییت با ۵۰۰ توییت موجود حاصل از کد پایتونی زیر جایگزین شد.

tweet_extraction.py

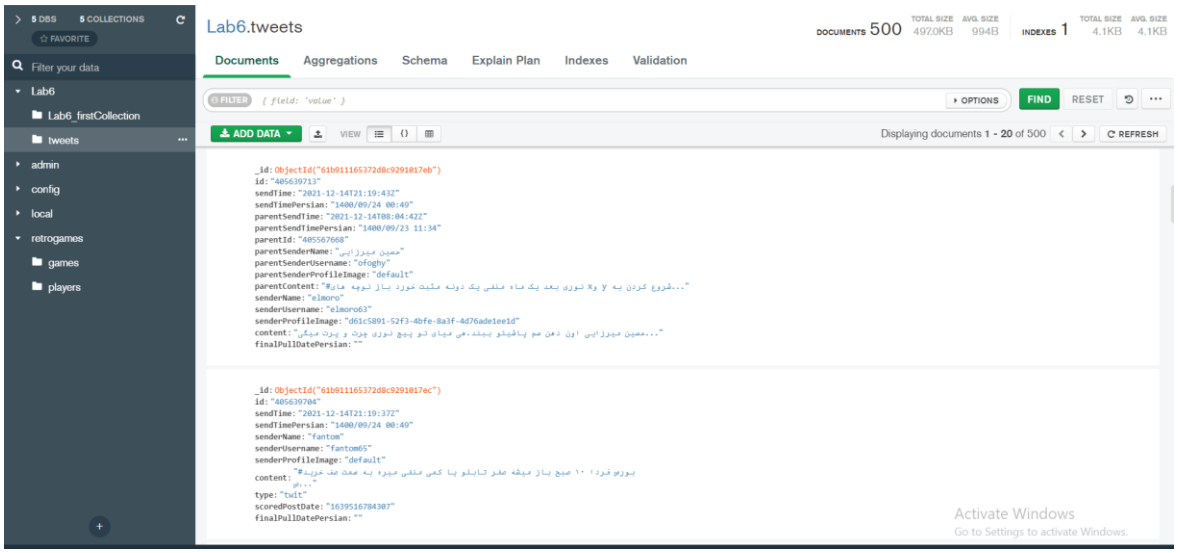
```python
from pymongo import MongoClient
import requests, time

tweets_url = "https://www.sahamyab.com/guest/twiter/list?v=1.0"
client = MongoClient('localhost', 27017)
db = client.Lab6
tweets = db.tweets

while (tweets.count_documents({}) < 500):
    response = requests. request ('GET', tweets_url, headers={ 'User-Agent': 'Chrome/61'})
    if response.status_code == requests.codes.ok:
        crawled_tweets = response.json()['items']
        for tw in crawled_tweets:
            print(tw)
            try:
                tweets.replace_one({"id": tw["id"]}, tw, upsert=True)
                print(tw)
            except Exception as e:
                print("Error occured: ", str(e))
        time.sleep(60)

print("Count:" % tweets.count_documents({}))
```

در نهایت، شکل کلی کالکشن توییت ها به شکل زیر درآمد که قرار داشتن ۵۰۰ توییت در آن مشخص است.



گام دوم- پیش‌پردازش داده:

```
from pymongo import MongoClient
import requests, time, re                    # re for regex

mongodb_cli = MongoClient('localhost', 27017)

db = mongodb_cli.Lab6
tweets = db.tweets
start = time.time()

update_count = 0
for tw in tweets.find({}):
    tweet_content = tw['content']
    hashtags = list(re.findall(r'#(\w+)', tweet_content))
    tweets.update_one({"id": tw['id']}, {"$set": {"hashtags": hashtags}})
    update_count += 1

end = time.time()
print("{} rows updated in {} seconds".format(update_count, end - start))
```

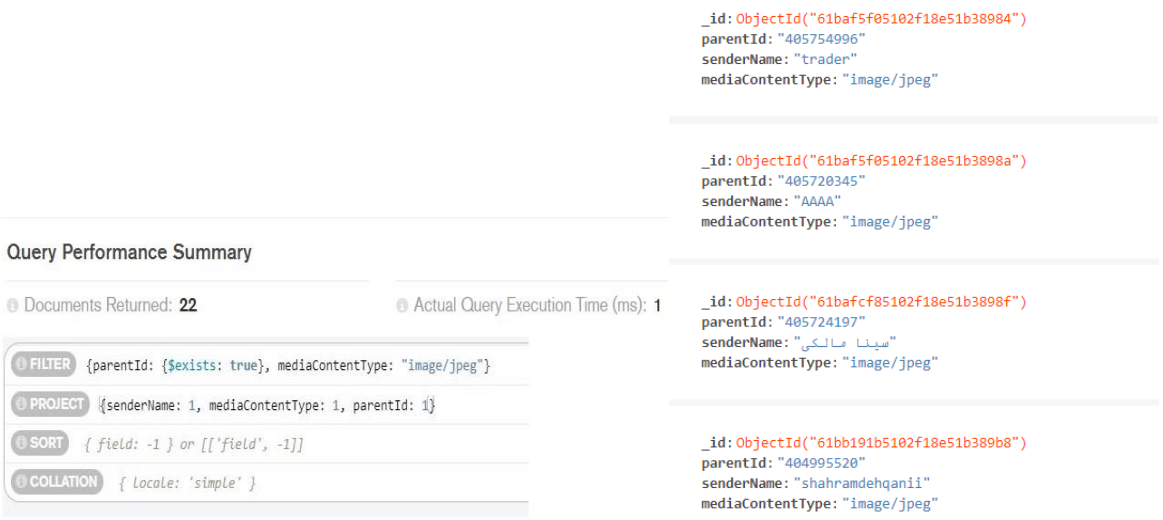
```
500 rows updated in 0.34645795822143555 seconds
```

در این مرحله، به کمک تکه کد بالا برای هر توییت از فیلد content آن هشتگ ها استخراج شدند و همانطور که خواسته شده‌بود، در فیلدی به نام hashtags ذخیره شدند. عمل اصلی در خط مشخص شده در این تکه کد انجام میگیرد که به کمک regex تمامی هشتگ های موجود را در یک لیست می‌ریزد. البته زمان اجرای اولیه کمی بیشتر از این مقدار بود و ۰/۷۵ ثانیه به طول انجامید که از آن تصویر تهیه نشد و این اجرای دوم، به علت تکراری بودن سریع تر انجام شد.

در قسمت زیر، یک سند نمونه را می‌بینیم که هر ۳ هشتگ موجود در متن آن در لیست قرار دارند.

گام سوم- دستورات اصلی:

۱) پیدا کردن نام کاربرانی که parentId دارند و mediaContentType شان image/jpeg است:



به کمک فیلتر هایی که روی فیلدهای parentId و mediaContentType شد و projection روی سه فیلد مد نظر، نتیجه به شکلی که در تصویر سمت چپ مشخص است به نمایش در آمد. در مجموع ۲۲ رکورد این شرایط را دارا بودند.

---

۲) مقدار senderUsername و type از توییت‌های ارسالی در یک بازه ۱۵ دقیقه ای دلخواه:

برای این سوال، بازه زمانی ساعت ۹:۰۰ تا ۹:۱۵ تاریخ ۱۴ دسامبر استفاده شد که طبق تصویر، ۱۱ توییت در این بازه زمانی منتشر شده بودند. مقادیر sendTime آن ها باید در این بازه باشد چون sendTime حاصل تبدیل DateTime به استرینگ است و برای مقایسه مناسب تر است.

۳) کاربرانی که از ساعت ۹ تا ۱۰ یک روز دلخواه بیش از یک توییت کرده اند:

برای این سوال، از آنجایی که به نظر باید بدون استفاده از aggregation و گروه‌بندی روی username کاربران اسامی را پیدا می‌کردیم، ساعت ۹ تا ۱۰ صبح روز ۱۶ دسامبر انتخاب شد که ۲۰ توییت در این بازه موجود بود. سپس برای راحت تر متوجه شدن تعداد توییت هر کاربر، یک sorting روی senderUsername انجام گرفت تا username های مشابه در کنار هم قرار گیرند و تعداد توییت های هر کس راحت تر به چشم آید. به طور مثال، در این ۲۰ توییت دو کاربر بودند که بیش از یک توییت داشتند و می‌توانند جایزه دریافت کنند:

گام چهارم- دستورات تجمعی و آماری (Aggregate Functions):

۱) یافتن تعداد کاربران بر اساس گروه‌های مربوط به تعداد توییت (یک توییت، ۲ یا ۳ توییت و بیش از ۳ توییت):

به کمک محیط گرافیکی و با aggregation pipeline با ۲ stage این مورد انجام شد.

در اولین استیج، تعداد توییت‌های هر کاربر در فیلد tweet_count ذخیره شد. نمونه‌ای از نتایج پس از این مرحله نیز در سمت راست تصویر قابل مشاهده است.

تکه‌کد بالا نیز مربوط به استیج دوم از پایپ لاین است و خروجی آن در تصویر دوم مشخص است. کد پایتونی این دستورها نیز در ادامه آمده‌است.

```
aggregation_res = db.tweets.aggregate([
    {
        '$group': {
            '_id': {
                'username': '$senderUsername'
            },
            'tweet_count': {
                '$sum': 1
            }
        }
    }, {
        '$group': {
            '_id': {
                '$cond': {
                    'if': {
                        '$eq': [
                            '$tweet_count', 1
                        ]
                    },
                    'then': 'one_tweet',
                    'else': {
                        '$cond': {
                            'if': {
                                '$lt': [
                                    '$tweet_count', 4
                                ]
                            },
                            'then': 'two_or_three_tweets',
                            'else': 'more_than_three_tweets'
                        }
                    }
                }
            },
            'tweet_group_count': {
                '$sum': 1
            }
        }
    }
]
)
end = time.time()
for res in aggregation_res:
    print(res)

print("Completed in {} seconds".format(end - start))
```

```
{'_id': 'more_than_three_tweets', 'tweet_group_count': 10}
{'_id': 'one_tweet', 'tweet_group_count': 307}
{'_id': 'two_or_three_tweets', 'tweet_group_count': 66}
Completed in 0.017939329147338867 seconds
```

در تصویر بالا، نتایج و تعداد کاربران هر دسته مشخص است و طول زمان اجرای کد نیز آورده شده‌است.

---

۲) یافتن پرتکرار ترین هشتگ‌ها:

```
aggregation_res = tweets.aggregate([
        {"$unwind": "$hashtags"},
        {'$group':
            {
                '_id': {'hashtag': '$hashtags'},
                'hashtag_count': {'$sum': 1}
            }
        },
        {'$sort': {"hashtag_count": -1}}
    ]
)

end = time.time()
print("Completed in {} seconds. \n\n".format(end - start))
for res in aggregation_res:
    print(res)
```

```
Completed in 0.007529735565185547 seconds.


{'_id': {'hashtag': 'شاخص_بورس'}, 'hashtag_count': 44}
{'_id': {'hashtag': 'خودرو'}, 'hashtag_count': 24}
{'_id': {'hashtag': 'برکت'}, 'hashtag_count': 19}
{'_id': {'hashtag': 'شهلی'}, 'hashtag_count': 17}
{'_id': {'hashtag': 'شستا'}, 'hashtag_count': 15}
{'_id': {'hashtag': 'کسرا'}, 'hashtag_count': 13}
{'_id': {'hashtag': 'پالایش'}, 'hashtag_count': 10}
{'_id': {'hashtag': 'تپکو'}, 'hashtag_count': 8}
{'_id': {'hashtag': 'کشما'}, 'hashtag_count': 7}
```

ابتدا لیست هشتگ های هر توییت را unwind می کنیم. این عمل معادل با explode کردن یک لیست است که به ازای هر مقدار از لیست هشتگ‌ها، یک سند جدید ایجاد می‌کند و به عبارتی لیست را با المان هایش در سندهای مختلف جایگزین می‌کند. سپس به روش گروه‌بندی کردن روی نام هشتگ‌ها، تعداد تکرار هریک را می‌شماریم که نتایج بالا حاصل می‌شود. پر تکرار ترین، مطابق انتظار هشتگ شاخص_بورس است. در تصویر خروجی کامل قرار داده نشده است و تعداد از پرتکرار ترین ها را می‌بینیم.

---

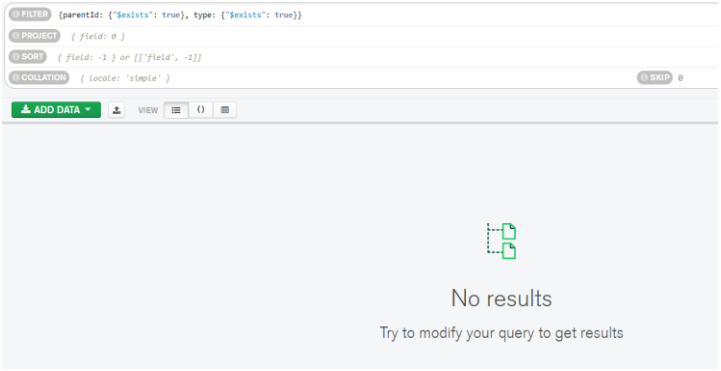
۳) حذف فیلد type از توییت های دارای parent-id:

```
res = tweets.update_many(
    {"parentId": {"$exists": "true"}},
    {"$unset": {"type": ""}}
)

end = time.time()
print("Completed in {} seconds. \n\n".format(end - start))
```

```
Completed in 0.007972002029418945 seconds.
```

به کمک دستور unset برای رکورد های مورد نظر فیلد type را حذف می‌کنیم. سپس برای اطمینان از حذف شدن آن، در تصویر پایین میبینیم که آیا رکوردی هست که هم parentId داشته باشد و هم type داشته باشد یا نه. که همانگونه که در تصویر پایین مشخص است، این کوئری نتیجه‌ای در بر نداشته و کار ما به درستی انجام شده است.



---

۴) پرتکرارترین و کم‌تکرار ترین هشتگ‌ها:

```
aggregation_res = tweets.aggregate([
        {"$match": {
                "sendTime": {
                    "$gte": start_datetime,
                    "$lte": end_datetime
                }
            }
        },
        {"$unwind": "$hashtags"},
        {'$group':
            {
                '_id': {'hashtag': '$hashtags'},
                'hashtag_count': {'$sum': 1}
            }
        },
        {'$sort': {"hashtag_count": -1}}
    ]
)

end = time.time()
print("Completed in {} seconds. \n\n".format(end - start))
aggregation_res = list(aggregation_res)

max_hashtag_count = aggregation_res[0]["hashtag_count"]
min_hashtag_count = aggregation_res[len(aggregation_res) - 1]["hashtag_count"]

print("Max hashtag count: ")
i = 0
while aggregation_res[i]["hashtag_count"] == max_hashtag_count:
    print(aggregation_res[i]["_id"], max_hashtag_count)
    i += 1

print("\n\nMin hashtag count: ")
i = len(aggregation_res) - 1
while aggregation_res[i]["hashtag_count"] == min_hashtag_count:
    print(aggregation_res[i]["_id"], min_hashtag_count)
    i -= 1
```

```
Completed in 0.007975339889526367 seconds.


Max hashtag count:
{'hashtag': 'شاخص_بورس'} 25


Min hashtag count:
{'hashtag': 'ویاسار'} 1
{'hashtag': 'خاور'} 1
{'hashtag': 'هگمتر'} 1
{'hashtag': 'تیتزر'} 1
{'hashtag': 'پتایر'} 1
{'hashtag': 'تلیسه'} 1
{'hashtag': 'کیمیون'} 1
{'hashtag': 'فملی'} 1
{'hashtag': 'غشهداب'} 1
{'hashtag': 'پلوله'} 1
{'hashtag': 'فروی'} 1
{'hashtag': 'فایرا'} 1
{'hashtag': 'البرز'} 1
{'hashtag': 'شفارس'} 1
{'hashtag': 'توسن'} 1
{'hashtag': 'وبصادر'} 1
{'hashtag': 'کمعدی'} 1
{'hashtag': 'ثرود'} 1
{'hashtag': 'فولاد'} 1
{'hashtag': 'بوعلی'} 1
{'hashtag': 'شپنا'} 1
{'hashtag': 'نظرسنجی'} 1
{'hashtag': 'کماسه'} 1
```

```
{'hashtag': 'کماسه'} 1
{'hashtag': 'سصوفی'} 1
{'hashtag': 'کقپارس'} 1
{'hashtag': 'سصفها'} 1
{'hashtag': 'سمگا'} 1
{'hashtag': 'یکام'} 1
{'hashtag': 'فجر'} 1
{'hashtag': 'خبهمن'} 1
{'hashtag': 'خزامیا'} 1
{'hashtag': 'سپید'} 1
{'hashtag': 'حریل'} 1
{'hashtag': 'حسینا'} 1
{'hashtag': 'غکورش'} 1
{'hashtag': 'قاسم'} 1
{'hashtag': 'سکارون'} 1
{'hashtag': 'تتوسا'} 1
{'hashtag': 'تاخت'} 1
{'hashtag': 'رافزا'} 1
{'hashtag': 'جهرم'} 1
{'hashtag': 'وصندوق'} 1
{'hashtag': 'حفاری'} 1
{'hashtag': 'تورپل'} 1
{'hashtag': 'وایران'} 1
{'hashtag': 'کالا'} 1
{'hashtag': 'ختور'} 1
{'hashtag': 'شبندر'} 1
{'hashtag': 'والبر'} 1
{'hashtag': 'بکهنوج'} 1
```

```
{'hashtag': 'سنیر'} 1
{'hashtag': 'زبینا'} 1
{'hashtag': 'خکاوه'} 1
{'hashtag': 'شپاکسا'} 1
{'hashtag': 'نوب'} 1
{'hashtag': 'وبانک'} 1
{'hashtag': 'ولسایا'} 1
{'hashtag': 'وآیند'} 1
{'hashtag': 'امید'} 1
{'hashtag': 'بورس'} 1
{'hashtag': 'ستران'} 1
{'hashtag': 'غپینو'} 1
{'hashtag': 'کیمیا'} 1
{'hashtag': 'سبجنو'} 1
{'hashtag': 'بالاس'} 1
{'hashtag': 'فخوز'} 1
{'hashtag': 'تتران'} 1
{'hashtag': 'کازرو'} 1
{'hashtag': 'وسدید'} 1
{'hashtag': 'وغدیر'} 1
```

مشابه کد استفاده شده در قسمت دوم از گام چهارم، تعداد تکرار هر هشتگ را حساب می‌کنیم. سپس مقدار کمترین و بیشترین تکرار را با تبدیل حاصل به لیست استخراج می‌کنیم. در مرحله آخر برای چاپ، از ابتدای لیست شروع می‌کنیم و ماکسیمم ها را چاپ می‌کنیم و سپس از انتهای لیست مینیمم ها را.

---

۵) ۱۰ هشتگ پرتکرار یک روز خاص:

```
start_datetime = "2021-12-16T00:00:00Z"
end_datetime = "2021-12-16T23:59:59Z"

aggregation_res = tweets.aggregate([
        {"$match": {
                "sendTime": {
                    "$gte": start_datetime,
                    "$lte": end_datetime
                }
            }
        },
        {"$unwind": "$hashtags"},
        {'$group':
            {
                '_id': {'hashtag': '$hashtags'},
                'hashtag_count': {'$sum': 1}
            }
        },
        {'$sort': {"hashtag_count": -1}}
    ]
)

end = time.time()
print("Completed in {} seconds. \n\n".format(end - start))
printed_count = 0
for res in aggregation_res:
    if printed_count == 10:
        break
    print(res)
    printed_count += 1
```

درکد بالا، شروع و پایان روز مدنظر (مثلا ۱۶ دسامبر) در دو متغیر ثابت داده شده اند. سپس sendTime فیلتر شده است و بعد به همان روش استفاده شده در سوال دوم گام چهارم، هشتگ های پرتکرار شمرده شده‌اند. نتیجه در تصویر زیر قابل مشاهده است.

```
Completed in 0.006978273391723633 seconds.


{'_id': {'hashtag': 'شاخص_بورس'}, 'hashtag_count': 25}
{'_id': {'hashtag': 'برکت'}, 'hashtag_count': 14}
{'_id': {'hashtag': 'خودرو'}, 'hashtag_count': 9}
{'_id': {'hashtag': 'شهلی'}, 'hashtag_count': 8}
{'_id': {'hashtag': 'پالایش'}, 'hashtag_count': 8}
{'_id': {'hashtag': 'شستا'}, 'hashtag_count': 5}
{'_id': {'hashtag': 'تشاهد'}, 'hashtag_count': 5}
{'_id': {'hashtag': 'شلرد'}, 'hashtag_count': 4}
{'_id': {'hashtag': 'کدما'}, 'hashtag_count': 4}
{'_id': {'hashtag': 'تپکو'}, 'hashtag_count': 4}
```

---

۶) فعال‌ترین کاربر در یک روز خاص:

```
start_datetime = "2021-12-16T00:00:00Z"
end_datetime = "2021-12-16T23:59:59Z"

aggregation_res = tweets.aggregate([
        {"$match": {
                "sendTime": {
                    "$gte": start_datetime,
                    "$lte": end_datetime
                }
            }
        },
        {'$group':
            {
                '_id': {'senderUsername': '$senderUsername'},
                'tweet_count': {'$sum': 1}
            }
        },
        {'$sort': {"tweet_count": -1}}
    ]
)

end = time.time()
print("Completed in {} seconds. \n\n".format(end - start))

aggregation_res = list(aggregation_res)

max_tweet_count = aggregation_res[0]["tweet_count"]

print("Most active user: ")
i = 0
while aggregation_res[i]["tweet_count"] == max_tweet_count:
    print(aggregation_res[i]["_id"], "\tTweets count:", max_tweet_count)
    i += 1
```

در کد بالا نیز، شروع و پایان روز مدنظر (مثلا ۱۶ دسامبر) در دو متغیر ثابت داده شده اند. سپس sendTime فیلتر شده است و بعد گروه‌بندی روی senderUsername انجام گرفته و تعداد توییت های هر کاربر شمارش شده‌است. در پایان نیز کاربر(ها) با بیشتری تعداد توییت مشخص شده‌است.

```
Completed in 0.008061408996582031 seconds.


Most active user:
{'senderUsername': 'alidsh88'}  Tweets count: 8

----------------------------------------------------------
```