

سوال اول

می‌دانیم تنها زمانی در MongoDB، سازگاری (Consistency) بر دسترس پذیری (Availability) ارجحیت دارد که ویژگی افراز را داشته باشیم و این افراز به این صورت است که باید بیش از نیمی از سرور های مجموعه Replica-Sets به یکدیگر متصل باشند و همچنین داده‌ها بین همه Node سازگار است و تحمل پارتیشن را حفظ می‌کند (جلوگیری از همگام سازی داده‌ها) با در دسترس نبودن یک Node و تا زمانی که یک Node اصلی انتخاب نشود (به روز رسانی نشود) آن Node موقتا از دسترس خارج می‌شود و response نمی‌دهد. و از لحاظ خواندن و نوشتن نیز اگر Majority Not Connected باشد برای جلوگیری از سیستم های جدا و ناسازگار، فقط دسترسی خواندن را داریم.

سوال دوم

این قسمت نیز همانند قسمت قبل است با این تفاوت که در این قسمت می‌خواهیم پایگاه داده MongoDB فقط نهایتا سازگار (Eventual Consistency) باشد و ویژگی سازگاری را نداشته باشد و حالت سیستم زمانی در نهایت سازگار خواهد شد که دیگر سازگاری سیستم در انتهای هر تراکنش چک نشود چرا که مقادیر اقلام داده‌ای دیر یا زود منتشر و نسخه‌های مختلف نهایتا سازگار می‌شوند و به دلیل CP بودن خود پایگاه داده MongoDB ویژگی دسترس پذیری (Availability) امکان پذیر نیست.

از لحاظ خواندن و نوشتن اگر Majority Connected باشد نوشتن سازگاری را از دست می‌دهد و Node اصلی قبلی در نظر گرفته نمی‌شود اما در نهایت می‌تواند بعد از نوشتن ویژگی سازگاری را کسب کند.

سوال سوم)

(A) نگهداری Session کاربران در Cache :

برای این مورد پایگاه داده غیر رابطه‌ای از نوع کلید-مقدار مناسب است و دلیل استفاده از این نوع پایگاه داده این است که این داده نه ساختاری است و نه شمای ثابتی دارد و هیچ ارتباطی هم در آن مطرح نیست و همچنین توزیع داده‌ها در پایگاه داده کلید-مقدار با استفاده از $h(k)$ (مقدار درهم سازی شده کلید) انجام می‌شود و ذخیره سازی این Session کاربران در Cache به این صورت انجام می‌شود که بسته به این که $h(k)$ در چه بازه‌ای بیفتد، گره مربوطه بر روی حلقه (در جهت عقربه های ساعت) برای ذخیره سازی (k, v) استفاده می‌شود و در صورت تکرار این نشست‌ها، کلید-مقدار مورد نظر بر روی گره مربوط به خود و چند گره متوالی بعدی بر روی حلقه (در جهت عقربه‌های ساعت) ذخیره می‌شود. یکی از پایگاه داده‌های کلید-مقدار نیز که می‌توانیم برای این مورد استفاده کنیم، Redis است.

(B) مدیریت مطالب یک وبلاگ:

برای مدیریت مطالب یک وبلاگ از پایگاه داده رابطه‌ای استفاده می‌کنیم و دلیل استفاده از آن هم برای این است که مطالب یک وبلاگ شامل تعدادی پست و تصویر و ... که همان RDBMS است.

(C) Recommendation System و یا یک شبکه اجتماعی:

برای این مورد پایگاه داده غیر رابطه‌ای مناسب‌تر است، مانند FACEBOOK و TWITTER که هر دو از پایگاه داده غیر رابطه‌ای از نوع ستونی استفاده می‌کنند و دلیل استفاده از این نوع پایگاه داده هم این است که می‌توان برای یک شبکه اجتماعی به طور مثال خانواده ستون‌های اطلاعات پروفایل هر فرد، پست‌های هر فرد و ... را در نظر گرفت که نام یکی از این پایگاه داده‌های غیر رابطه‌ای ستونی که هم FACEBOOK و هم TWITTER از آن استفاده می‌کنند، Cassandra می‌باشد.

سوال چهارم)

تمامی کدهای مربوط همراه همین گزارش کار آپلود شده است. (کوئری های کامل شده در محیط ژوپیتتر به اسم Queries.ipynb و کدهای پایتون آن با شماره دانشجویی هر دو عضو به همراه فایل JSON)

همچنین تصاویر کد مربوط به هر سوال به همراه خروجی آن در ادامه آپلود شده است:

۱. نام داروخانه هایی که شماره تلفن آنها با ۱+ شروع می شود:

```
# نام داروخانه هایی که شماره تلفن آنها با 1+ شروع می شود
print("##1##")
a1 = list(
    client.assignment.pharmacy.find(
        filter={'telephone':{'$regex': '\\+1*'}}, # Complete the filter
        projection={"name": 1, "_id": 0},
    )
)
print(a1)

##1##
[{'name': 'Rogers-Cardenas'}, {'name': 'Jones, Fox and Romero'}, {'name': 'Lee, Moore and Ferrell'}, {'name': 'Ellison-Arias'}, {'name': 'Henry Inc'},
{'name': 'Lopez-Buchanan'}, {'name': 'Weber-Zamora'}, {'name': 'Kennedy Inc'}, {'name': 'Ramirez Group'}, {'name': 'Fry-Douglas'}, {'name': 'Smith, Ke
mp and Hughes'}, {'name': 'Nicholson, Smith and Hernandez'}]
```

۲. شماره ملی افرادی که بعد از تاریخ datetime.datetime(2000, 1, 1, 0, 0) متولد شده اند:

```
# شماره ملی افرادی که بعد از تاریخ datetime.datetime(2000, 1, 1, 0, 0) متولد شده اند
print("##2##")
a2 = list(
    client.assignment.patient.find(
        filter={"birthdate" : {'$gt' : datetime.datetime(2000, 1, 1, 0, 0)}}, # Complete the filter
        projection={"national_id": 1, "_id": 0},
    )
)
print(a2)

##2##
[{'national_id': '7510799118'}, {'national_id': '8069901627'}, {'national_id': '7059640165'}, {'national_id': '9599534283'}, {'national_id': '59738320
60'}, {'national_id': '5899441715'}, {'national_id': '9812387800'}, {'national_id': '4271739627'}, {'national_id': '2091241862'}, {'national_id': '024
9724399'}, {'national_id': '0242403381'}, {'national_id': '9074260880'}, {'national_id': '3578243477'}, {'national_id': '0961701595'}, {'national_id':
'8114578870'}, {'national_id': '8816502544'}, {'national_id': '1123274897'}, {'national_id': '4815269255'}, {'national_id': '8519537749'}, {'national_
id': '5700412918'}, {'national_id': '2546862728'}, {'national_id': '8864078802'}, {'national_id': '0186074290'}, {'national_id': '2365401814'}]
```

۳. تعداد نسخه هایی که دارای حداقل ۱۵ دارو هستند:

```
# تعداد نسخه هایی که دارای حداقل 15 دارو هستند
print("##3##")
a3 = client.assignment.prescription.find(
    filter={'items': {'$exists': True}, '$where': 'this.items.length > 15'} # Complete the filter
).count()
print(a3)

##3##
20
```

۴. کد ملی بیمارانی که اسم پزشک آنها "Robert" است:

```
# کد ملی بیمارانی که اسم پزشک آنها "Robert" است
print("##4##")
a4 = list(
    client.assignment.patient.aggregate(
        [
            {
                "$lookup": {
                    "from": "doctor",
                    "localField": "doctor_id",
                    "foreignField": "_id",
                    "as": "doctor",
                },
            },
            {"$match": {"doctor.first_name": "Robert"}},
            {'$project': {'_id': 0, 'national_id': 1}}
        ]
    )
)
print(a4)

##4##
[{'national_id': '3389083863'}, {'national_id': '0405411526'}, {'national_id': '9267389975'}, {'national_id': '0242403381'}, {'national_id': '0961701595'}, {'national_id': '2611231261'}, {'national_id': '3153888245'}, {'national_id': '4991004202'}, {'national_id': '0259364502'}]
```

۵. نام داروخانه‌ای که دارویی به گرانترین قیمت به آن فروخته شده است:

```
# نام داروخانه‌ای که دارویی به گرانترین قیمت به آن فروخته شده است
print("##5##")
a5 = client.assignment.sale.aggregate(
    [
        {
            '$sort': {'price': -1}
        },
        {
            '$lookup': {
                'from': 'pharmacy',
                'localField': 'pharmacy_id',
                'foreignField': '_id',
                'as': 'pharmacy'
            },
        },
        {
            '$unwind': '$pharmacy'
        },
        {
            '$replaceRoot': {'newRoot': '$pharmacy'}
        },
        {
            '$project': {
                '_id': 0,
                'name': 1
            }
        }
    ]
).next()
print(a5)

##5##
{'name': 'Clarke and Sons'}
```

۶. نام و فرمول پنج دارویی که گران ترین قیمت برای آنها ثبت شده است:

```
# نام و فرمول پنج دارویی که گران ترین قیمت برای آنها ثبت شده است
print("##6##")
a6 = list(
    client.assignment.sale.aggregate(
        [
            {
                '$sort': {'price': -1}
            },
            {
                '$lookup': {
                    'from': 'drug',
                    'localField': 'drug_id',
                    'foreignField': '_id',
                    'as': 'drug'
                }
            },
            {
                '$unwind': '$drug'
            },
            {
                '$replaceRoot': {'newRoot': '$drug'}
            },
            {
                '$project': {
                    '_id': 0,
                    'name': 1,
                    'formula': 1
                }
            },
            { '$limit': 5 }
        ]
    )
)
print(a6)

##6##
[{'name': 'Cyanocobalamin', 'formula': 'H2O'}, {'name': 'Trihexyphenidyl', 'formula': 'C4H8O2'}, {'name': 'Phenytoin', 'formula': 'Al'}, {'name': 'Est
radiol', 'formula': 'C3H7NO2'}, {'name': 'Florinef', 'formula': 'KF'}]
```

۷. نام تمام داروهایی که در تاریخ `datetime.datetime(2020, 9, 23, 0, 0)` تجویز شده اند:

```
# نام تمام داروهایی که در تاریخ datetime.datetime(2020, 9, 23, 0, 0) تجویز شده اند
print("##7##")
a7 = list(
    client.assignment.prescription.aggregate(
        [
            {
                '$match': {'date': datetime.datetime(2020, 9, 23, 0, 0)}
            },
            {
                '$lookup': {
                    'from': 'drug',
                    'localField': 'items.drug_id',
                    'foreignField': '_id',
                    'as': 'drug'
                }
            },
            {
                '$unwind': '$drug'
            },
            {
                '$replaceRoot': {'newRoot': '$drug'}
            },
            {
                '$project': {
                    '_id': 0,
                    'name': 1,
                }
            },
        ]
    )
)
print(a7)

##7##
[{'name': 'Oxycodone HCl'}, {'name': 'Demadex'}, {'name': 'Ceftin'}, {'name': 'Etrafon'}]
```

۸. نام تمام کارخانه هایی که داروی با فرمول "C2H6Na4O12" را تولید می کنند:

```
# نام تمام کارخانه هایی که داروی با فرمول "C2H6Na4O12" را تولید می کنند
print("##8##")
a8 = list(
    client.assignment.drug.aggregate(
        [
            {
                '$match': {'formula': 'C2H6Na4O12'}
            },
            {
                '$lookup': {
                    'from': 'company',
                    'localField': 'company_id',
                    'foreignField': '_id',
                    'as': 'company'
                }
            },
            {
                '$unwind': '$company'
            },
            {
                '$replaceRoot': {'newRoot': '$company'}
            },
            {
                '$project': {
                    '_id': 0,
                    'name': 1,
                }
            }
        ]
    )
)
print(a8)

##8##
[{'name': 'Williams, Parker and Cross'}, {'name': 'Walker Ltd'}, {'name': 'Moore, Davis and Rodriguez'}, {'name': 'Lopez-Johnson'}, {'name': 'Davis Group'}, {'name': 'Brock-White'}]
```

۹. کاربرانی که در سبد آنها حداقل ده وجود دارد BasketItem وجود دارد:

```
# کاربرانی که در سبد آنها حداقل ده BasketItem وجود دارد
print("##9##")
a9 = list(
    client.assignment.user.find(
        filter={'basket': {'$size': 10}}, # Complete the filter
        projection={"email": 1, "_id": 0},
    )
)
print(a9)

##9##
[{'email': 'walkerlindsey@hotmail.com'}, {'email': 'terriramirez@yahoo.com'}, {'email': 'lawrence49@gmail.com'}, {'email': 'jonesdeborah@hotmail.com'}, {'email': 'kyle88@hotmail.com'}, {'email': 'wangdavid@yahoo.com'}, {'email': 'osimmons@yahoo.com'}, {'email': 'coxkarl@yahoo.com'}, {'email': 'williamsmichelle@gmail.com'}, {'email': 'castrokaitlin@gmail.com'}, {'email': 'hernandezkatherine@yahoo.com'}, {'email': 'carolmoore@hotmail.com'}, {'email': 'kathleen67@yahoo.com'}, {'email': 'hornejasmine@hotmail.com'}]
```

۱۰. میزان موجودی (sum quantity) که اجناس سایز "XL" هستند:

```
# میزان موجودی (sum quantity) اجناس که سایز "XL" هستند
print("##10##")
a10 = client.assignment.product_item.aggregate(
    [
        {
            '$match': {'size': 'XL'}
        },
        {
            '$group': {'_id': None, 'sum': {'$sum': '$quantity'}}
        },
        {
            '$project': {'_id': 0}
        }
    ]
).next()
print(a10)

##10##
{'sum': 15593}
```

۱۱. شماره ملی رانندگانی که پلاک آنها به ۲۵ ختم می شود:

```
# شماره ملی رانندگانی که پلاک آنها به 25 ختم می شود
print("##11##")
a11 = list(
    client.assignment.driver.find(
        filter={'license_plate': {'$regex': '.*+25$'}}, # Complete the filter
        projection={"_id": 0, "national_id": 1},
    )
)
print(a11)

##11##
[{'national_id': '1956964607'}, {'national_id': '8157341591'}]
```

۱۲. متن کامنت هایی که کلمه "great" و یا "good" در آن وجود دارد:

```
# متن کامنت هایی که کلمه "good" و یا "great" در آن وجود دارد
print("##12##")
a12 = list(
    client.assignment.comment.find(
        filter={'text': {'$regex': '.*(good|great).*'}},
        projection={"_id": 0, "text": 1},
    )
)
print(a12)

##12##
[{'text': 'Church certainly question share that job ever capital great what mean continue.'}, {'text': 'Piece good reason too evening probably low pul l political.'}]
```

۱۳. تعداد نظراتی که ریتینگ آن‌ها ۵ است:

```
# تعداد نظراتی که ریتینگ آن‌ها ۵ است
print("##13##")
a13 = client.assignment.comment.aggregate(
    [
        { "$match": {
            "rating": 5
        }
        },
        {
            "$count": "count"
        }
    ]
).next()
print(a13)

##13##
{'count': 12}
```

تصویر خروجی کد قسمت JSON:

```
correct = 0
for i in range(1, 14):
    if answers["a{}".format(i)] == target["a{}".format(i)]:
        print("Query {:2d} Correct!".format(i))
        correct += 1
    else:
        print("Query {:2d} Wrong!".format(i))
print(correct)
```

```
Query  1 Correct!
Query  2 Correct!
Query  3 Correct!
Query  4 Correct!
Query  5 Correct!
Query  6 Correct!
Query  7 Correct!
Query  8 Correct!
Query  9 Correct!
Query 10 Correct!
Query 11 Correct!
Query 12 Correct!
Query 13 Correct!
13
```