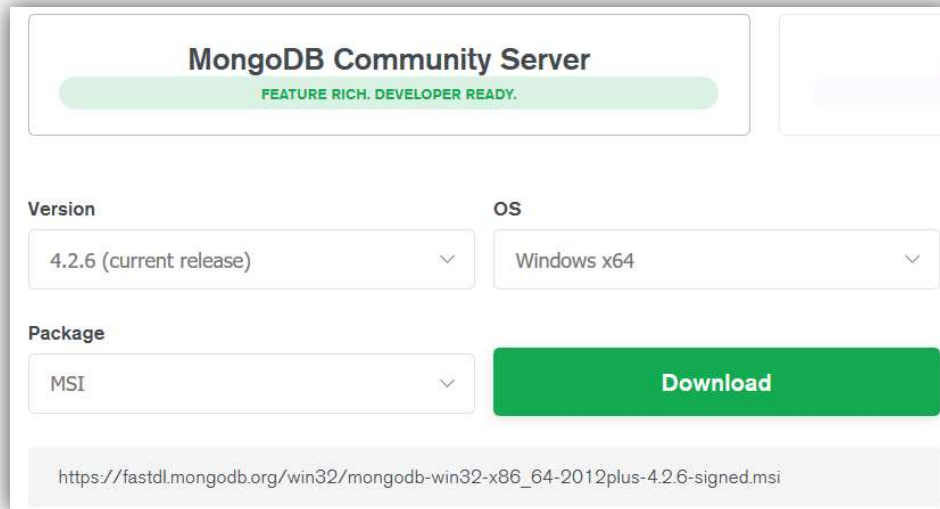


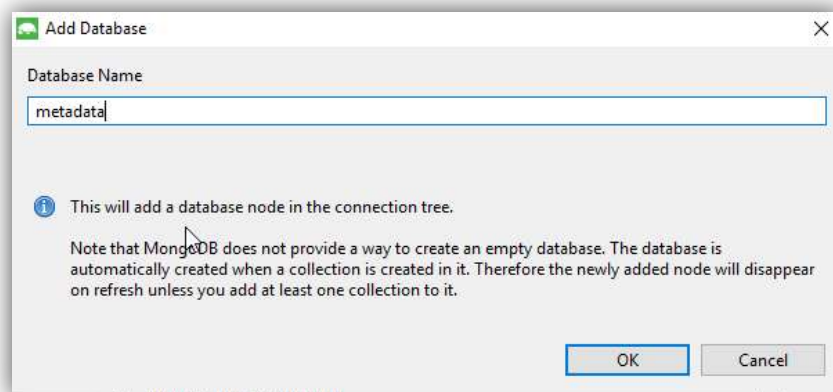
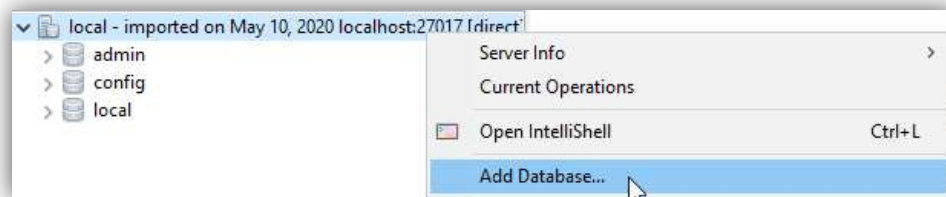
بخش اول: ذخیره اطلاعات بدون ساختار / کار با MongoDB

مراحل دانلود، نصب و آماده‌سازی کار با MongoDB به‌صورت زیر می‌باشد:

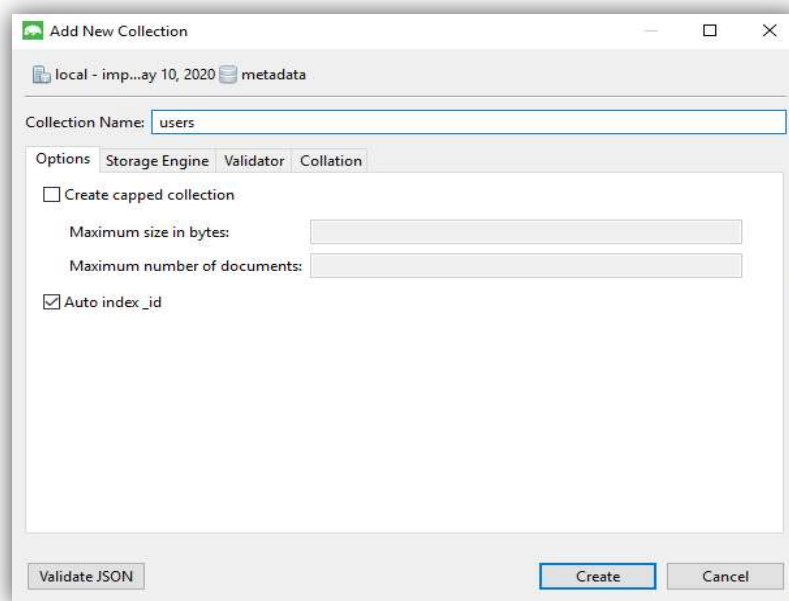
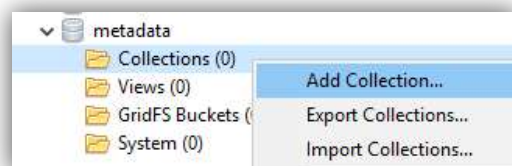
- 1- ابتدا از آدرس <https://www.mongodb.com/download-center/community>، مطابق با تصویر زیر، نسخه‌ی 4.2.6 مانگودی‌بی را به‌صورت بسته‌بندی MSI برای سیستم‌عامل ویندوز دانلود می‌کنیم:



- 2- سپس در آدرس پیش‌فرض خود که همان **C:\Program Files\MongoDB\Server\4.2** است، آنرا نصب می‌کنیم.
- 3- به‌طور پیش‌فرض داده‌های مانگو در آدرس **C:\data\db** ذخیره خواهند شد، به‌همین دلیل، در درایو C یک فولدر به نام data را ایجاد می‌کنیم و سپس در داخل آن یک فولدر به نام db می‌سازیم تا داده‌های مانگو بدون مشکل در این فولدر ذخیره شوند.
- 4- سپس آدرس **C:\Program Files\MongoDB\Server\4.2\bin** را به Path های سیستم اضافه می‌کنیم.
- 5- در نهایت با اجرای دستور mongod.exe در cmd، مانگودی‌بی را اجرا می‌کنیم. در ضمن اگر بخواهیم که از خط فرمان مانگودی‌بی (mongo shell) استفاده کنیم، دستور mongo.exe را در cmd اجرا می‌کنیم.
- 6- در اینجا به‌جای خط فرمان مانگودی‌بی از ابزار گرافیکی بسیار قدرتمند Studio 3T استفاده می‌کنیم که نسخه‌ی قدرتمندتر ابزار Robo 3T می‌باشد.
- 7- در Studio 3T، ابتدا برای اتصال به DB Server به آدرس **localhost:27017** متصل می‌شویم. سپس دیتابیس metadata را به‌صورت زیر ایجاد می‌کنیم:




پس از ایجاد دیتابیس metadata، کالکشن users را به دیتابیس metadata اضافه می‌نماییم:



گام اول تمرین – درج داده‌ها



ابتدا گزینه‌ی  را انتخاب کرده تا به خط فرمان مانگودی‌بی دسترسی داشته باشیم. سپس آدرس <https://randomuser.me/api/?nat=ir> را فراخوانی کرده، فایل JSON آنرا کپی می‌کنیم و سپس اطلاعات موجود در فیلد results را با دستور زیر در کالکشن users وارد می‌کنیم:

```
db.getCollection("users").insertOne({"gender":"female","name":{"title":"Ms","first":  
"الینا","last":"نکو نظر"},"location":{"street":{"number":288,"name":"شهید محمد  
منتظری"},"city":"اهواز","state":"مرکزی","country":"Iran","postcode":32571,"coordinates"  
":{"latitude":45.7234,"longitude":100.7914},"timezone":{"offset":"-  
11:00"},"description":"Midway Island,  
Samoa"}}, {"email":"lyn.nkwnzr@example.com","login":{"uuid":"3bd0072a-4a54-47da-b025-  
a26f3eb9a074","username":"goldenmouse258","password":"boat","salt":"OvrEY09c","md5":  
"0eb8b489793ee4c7dd78dd0dbefb833","sha1":"2313446429c10d1a494ddc8e7b363843fb8de37e"  
,"sha256":"f0119ba3f1166cdbf1e53305cc9131d7d06bd9f9a893ef1ae9e41e3ce4d39861"},"dob":  
{"date":"1967-03-23T19:18:51.750Z","age":53},"registered":{"date":"2019-07-  
16T03:50:04.010Z","age":1},"phone":"086-56042437","cell":"0930-454-  
9056","id":{"name":"","value":null},"picture":{"large":"https://randomuser.me/api/po  
rtraits/women/18.jpg","medium":"https://randomuser.me/api/portraits/med/women/18.jpg",  
,"thumbnail":"https://randomuser.me/api/portraits/thumb/women/18.jpg"},"nat":"IR"})
```

همین عمل را سه بار تکرار کرده و در نهایت از کالکشن users خروجی می‌گیریم (در صفحه‌ی بعد). همانطور که مشاهده می‌شود، به ازای هر insertی که داشته‌ایم، یک ObjectId با نام _id به‌عنوان کلید به مشخصات کاربر اضافه می‌شود.

Key	Value	Type
▼ (1) { _id : 5ec59f1f0a06be7297040af7 }	{ 13 fields }	Document
_id	5ec59f1f0a06be7297040af7	ObjectId
gender	female	String
> name	{ 3 fields }	Object
> location	{ 7 fields }	Object
email	lyn.nkwnzr@example.com	String
> login	{ 7 fields }	Object
> dob	{ 2 fields }	Object
> registered	{ 2 fields }	Object
phone	086-56042437	String
cell	0930-454-9056	String
> id	{ 2 fields }	Object
> picture	{ 3 fields }	Object
nat	IR	String
▼ (2) { _id : 5ec59f490a06be7297040af8 }	{ 13 fields }	Document
_id	5ec59f490a06be7297040af8	ObjectId
gender	female	String
> name	{ 3 fields }	Object
> location	{ 7 fields }	Object
email	ysmyn.hsny@example.com	String
> login	{ 7 fields }	Object
> dob	{ 2 fields }	Object
> registered	{ 2 fields }	Object
phone	019-67880586	String
cell	0989-573-9004	String
> id	{ 2 fields }	Object
> picture	{ 3 fields }	Object
nat	IR	String
▼ (3) { _id : 5ec59f6f0a06be7297040af9 }	{ 13 fields }	Document
_id	5ec59f6f0a06be7297040af9	ObjectId
gender	female	String
> name	{ 3 fields }	Object
> location	{ 7 fields }	Object
email	mhy.khmrw@example.com	String
> login	{ 7 fields }	Object
> dob	{ 2 fields }	Object
> registered	{ 2 fields }	Object
phone	047-82905330	String
cell	0909-153-3197	String
> id	{ 2 fields }	Object
> picture	{ 3 fields }	Object
nat	IR	String

سپس اطلاعات قبل را پاک کرده و با استفاده از کتابخانه‌ی pymongo و به‌صورت زیر، اطلاعات 100000 کاربر تصادفی را در مانگو ذخیره می‌کنیم:

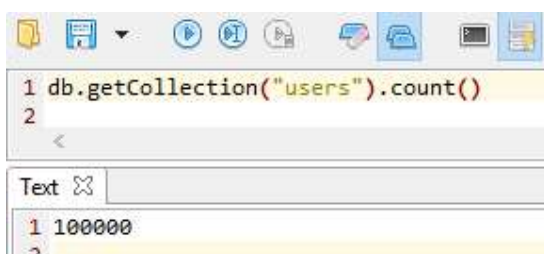
```
from pymongo import MongoClient
import requests, json, time
url = "https://randomuser.me/api/"

client = MongoClient("mongodb://localhost:27017/")
database = client["metadata"]
collection = database["users"]

def get_random_users(n):
    for i in range(0,n//5000):
        querystring = {"nat": "ir", "results": "5000"}
        response = None
        while True:
            response = requests.request("GET", url, params=querystring)
            if response.status_code==503:
                print("120 secs wait...")
                time.sleep(120)
            else:
                break
        data = json.loads(response.text)
        collection.insert_many(data['results'])
        print(i+1)
    print("finished.")

get_random_users(n=100000)
```

در نهایت، با دستور count مطمئن می‌شویم که 100000 کاربر تصادفی در کالکشن users ذخیره شده باشد:



گام دوم تمرین – دستورات اصلی

1. نام و نام خانوادگی کاربرانی که بالای 50 سال سن داشته و ساکن نیشابور باشند:

کوئری:

```
1 db.getCollection("users").find(
2   {
3     "dob.age":{
4       "$gte": 50
5     },
6     "location.city": "نیشابور"
7   },
8   {
9     "name.first": 1,
10    "name.last" : 1,
11    "_id" : 0
12  }
13 )
```

خروجی:

Tree		JSON
Key	Value	
✓ (1) { _id : }	{ 1 fields }	1 {
✓ (1) name	{ 2 fields }	2 " name" : {
"-" first	پارسا	3 "first" : "پارسا",
"-" last	حسینی	4 "last" : "حسینی"
(2) { _id : }	{ 1 fields }	5 }
✓ (1) name	{ 2 fields }	6 }
"-" first	مهرسا	7 {
"-" last	کامیاران	8 " name" : {
(3) { _id : }	{ 1 fields }	9 "first" : "مهرسا",
✓ (1) name	{ 2 fields }	10 "last" : "کامیاران"
"-" first	آرمین	11 }
"-" last	کوئی	12 }
		13 {
		14 " name" : {
		15 "first" : "آرمین",
		16 "last" : "کوئی"
		17 }
		18 }

* لیست کامل خروجی که شامل 995 مورد است، در آدرس [Code\Part 1- Mongodb\Step 2 - main commands](#) قرار گرفته است.

0.191s

زمان اجرا:

2. نام خانوادگی، آدرس و شماره موبایل کاربرانی که بیش از 20 سال است که در سایت ما ثبت نام کرده اند:

نکته: با توجه به اینکه کاربران این سایت، حداکثر 18 سال پیش در این سایت ثبت نام کرده اند، در نتیجه برای اینکه سوال ذکر شده، خروجی داشته باشد، به جای 20، از 18 استفاده خواهیم کرد.

کوئری:

```
1 db.getCollection("users").find(
2   {
3     "registered.age":{
4       "$gte": 18
5     }
6   },
7   {
8     "name.last" : 1,
9     "location": 1,
10    "cell": 1,
11    "_id": 0
12  }
13 )
```

خروجی:

Tree		JSON
Key	Value	
▼ (1) { _id : }	{ 3 fields }	1 {
▼ (1) name	{ 1 fields }	2 "name" : {
"-" last	کریمی	3 "last" : "کریمی"
▼ (1) location	{ 7 fields }	4 },
▼ (1) street	{ 2 fields }	5 "location" : {
"-" number	8352	6 "street" : {
"-" name	پارک لاله	7 "number" : NumberInt(8352),
"-" city	بابل	8 "name" : "پارک لاله"
"-" state	یزد	9 },
"-" country	Iran	10 "city" : "بابل",
"-" postcode	69540	11 "state" : "یزد",
▼ (1) coordinates	{ 2 fields }	12 "country" : "Iran",
"-" latitude	-25.4863	13 "postcode" : NumberInt(69540),
"-" longitude	-116.6241	14 "coordinates" : {
▼ (1) timezone	{ 2 fields }	15 "latitude" : "-25.4863",
"-" offset	-11:00	16 "longitude" : "-116.6241"
"-" description	Midway Island, Samoa	17 },
"-" cell	0938-081-3563	18 "timezone" : {
		19 "offset" : "-11:00",
		20 "description" : "Midway Island, Samoa"
		21 },
		22 "cell" : "0938-081-3563"
		23 }
		24 }

* لیست کامل خروجی که شامل 4399 مورد است، در آدرس [Code\Part 1- MongoDB\Step 2 - main commands](#) قرار گرفته است.

0.219s

زمان اجرا:

3. افزودن فیلد year_persian به فیلد dob و registered:

کوئری:

```
1 db.getCollection("users").update(
2   {
3     },
4   [
5     {
6       "$set": {
7         "dob.year_persian": {
8           "$year": {
9             "$subtract": [ {"$dateFromString": {"dateString": "$dob.date"}}, 226899*24*60*60000 ]
10          }
11        },
12        "registered.year_persian": {
13          "$year": {
14            "$subtract": [ {"$dateFromString": {"dateString": "$registered.date"}}, 226899*24*60*60000 ]
15          }
16        }
17      }
18    }
19  ],
20  {
21    "multi": true
22  }
23 )
24 )
```

توضیح: ابتدا نوع فیلد dob.date که به صورت رشته ذخیره شده است را به Date تغییر می دهیم. سپس با توجه به اینکه اختلاف تاریخ شمسی و میلادی 226899 روز می باشد، 226899 روز از این تاریخ کم می کنیم تا به تاریخ شمسی دست یابیم. در نهایت با استفاده از \$year، سال را از تاریخ شمسی حاصل استخراج می کنیم و آنرا به فیلد جدیدی به نام dob.year_persian اختصاص می دهیم. ما این عمل را برای registered.date نیز انجام می دهیم.

خروجی کالکشن users به صورت Tree:

Key	Value
<ul style="list-style-type: none"> <ul style="list-style-type: none"> { 13 fields } _id: 5ec5aec0ab89eef4d3aaa3c2 gender: female name: { 3 fields } location: { 7 fields } email: lyn.hmdy@example.com login: { 7 fields } dob: { 3 fields } <ul style="list-style-type: none"> date: 1988-07-07T06:26:12.914Z age: 32 year_persian: 1367 registered: { 3 fields } <ul style="list-style-type: none"> date: 2005-12-13T16:08:51.103Z age: 15 year_persian: 1384 phone: 031-23157485 cell: 0945-033-0974 id: { 2 fields } picture: { 3 fields } nat: IR 	

* به همین صورت تمام ردیف های کالکشن users ویرایش شده، حاوی زیرفیلد year_persian در فیلد dob و registered هستند. با توجه به اینکه خروجی این سوال، حدود 180 مگابایت حجم دارد، از ارائه ی آن صرف نظر می کنیم.

12.03s

زمان اجرا:

4. دستوری بنویسید که با اجرا شدن آن در هر روز، نام و نام خانوادگی و ایمیل افرادی که در آن روز تولدشان است، به ما برگردانده شود:

کوئری:

```
1 db.getCollection("users").find(
2   {
3     "$expr":{
4       "$and":[
5         {
6           "$eq":[
7             {"$month":{"date":{"$dateFromString":{"dateString":"$dob.date"}}}},
8             {"$month":{"date:new Date()}}
9           ]
10        },
11        {
12          "$eq":[
13            {"$dayOfMonth":{"date":{"$dateFromString":{"dateString":"$dob.date"}}}},
14            {"$dayOfMonth":{"date:new Date()}}
15          ]
16        }
17      ]
18    }
19  },
20  {
21    "name.first": 1,
22    "name.last": 1,
23    "email":1,
24    "_id":0
25  }
26 )
```

توضیح: در کوئری بالا، نام و نام خانوادگی افرادی که روز و ماه تولدشان با روز و ماه جاری برابر است، برگردانده می شود. لازم به ذکر است که میتوانستیم بجای \$month و \$dayOfMonth از \$dayOfYear استفاده کنیم، ولی چونکه با این روش، سال کبیسه در نظر گرفته نمی شود، به همین دلیل از روش اول استفاده نمودیم.

خروجی:

Tree		JSON
Key	Value	
▼ (1) { _id : }	{ 2 fields }	1 {
▼ { name }	{ 2 fields }	2 "name" : {
{ "first" }	مهدی	3 "first" : "مهدی",
{ "last" }	نجاتی	4 "last" : "نجاتی"
{ "email" }	mhdy.njty@example.com	5 },
{ "email" }	mhdy.njty@example.com	6 "email" : "mhdy.njty@example.com"
▼ (2) { _id : }	{ 2 fields }	7 }
▼ { name }	{ 2 fields }	8 {
{ "first" }	محمدپارسا	9 "name" : {
{ "last" }	سلطانی نژاد	10 "first" : "محمدپارسا",
{ "email" }	mhmdprs.sltnynjd@example.com	11 "last" : "سلطانی نژاد"
		12 },
		13 "email" : "mhmdprs.sltnynjd@example.com"
		14 }

* لیست کامل خروجی که شامل 251 مورد است، در آدرس [Code\Part 1- MongoDB\Step 2 - main commands](#) قرار گرفته است.

3.041s

زمان اجرا:

5. ذخیره پسورد به شکل خام در اطلاعات کاربر، کاری غیر حرفه‌ای است. می‌خواهیم این مشکل را برطرف کنیم. چه راه حلی برای حل مساله پیشنهاد میکنید؟
برای حل این مسئله، می‌توان بجای پسورد از hex_md5 آن استفاده کرد.

راه حل را ابتدا روی یک سند خاص، امتحان کنید و مطمئن شوید بعد از اعمال تغییرات، آن کاربر خاص را با داشتن یوزنیم و پسورد، میتوانید پیدا کنید. سپس تمام کاربران را به روز رسانی کنید.

باتوجه به اینکه hex_md5 یک تابع جاوا اسکریپتی است، در نتیجه می‌توان از این تابع در خط فرمان مانگودی‌بی استفاده کرد. ولی بایستی ورودی این تابع، پسورد کاربر مورد نظر باشد. ولی مشکلی که وجود دارد این است که در یک تابع جاوا اسکریپتی، نمی‌توان با "\$login.password"، مقدار پسورد کاربر مورد نظر را دریافت و از آن به‌عنوان ورودی تابع hex_md5 استفاده کرد. به‌همین دلیل ما مجبور شدیم که از foreach بر روی find استفاده کنیم تا بتوانیم با دستورات جاوا اسکریپتی، پسورد کاربر(های) مورد نیاز را استخراج و از آنها به‌عنوان ورودی تابع hex_md5 استفاده کنیم. لازم به ذکر است که با توجه به اینکه استفاده از دستورات و توابع جاوا اسکریپتی در مانگودی‌بی مجاز است، در نتیجه تمام دستورات زیر در خط فرمان مانگودی‌بی قابل اجرا است.

ابتدا با استفاده از کوئری زیر، به‌جای پسورد کاربری با نام کاربری smallrabbit981، مقدار hex_md5 آنرا قرار می‌دهیم:

```
db.getCollection("users").find({"login.username":"smallrabbit981"}).forEach(function(data){
    data.login.password = hex_md5(data.login.password)
    db.getCollection("users").save(data)
});
```

پسورد خام کاربر smallrabbit981 ، daytona بود. حال اگر بخواهیم که این کاربر را با داشتن نام کاربری و پسورد خام پیدا کنیم، کافیت از کوئری زیر استفاده کنیم:

```
1 db.getCollection("users").find(
2   {
3     "login.username":"smallrabbit981",
4     "login.password":hex_md5("daytona")
5   }
6 )
```

که خروجی آن به‌صورت زیر می‌باشد:

Key	Value
▼ (1) { _id : 5ec5aec0ab89eef4d3aaa3a4 }	{ 13 fields }
_id	5ec5aec0ab89eef4d3aaa3a4
gender	male
name	{ 3 fields }
location	{ 7 fields }
email	prs.sly@example.com
login	{ 7 fields }
uuid	e4e3ff7f-d6e5-487b-84dc-f8e3c921eebb
username	smallrabbit981
password	5d1fb338b351a63a1720b837cc9010ff
salt	2x5wGESI
md5	7057c28225012a5df1d82ba343b5dcc0
sha1	ea9ee870538e7e7bf5ea85af770d2713937672
sha256	6a4ec63fba4177a656cc53d9e5683009f13367
dob	{ 3 fields }
registered	{ 3 fields }
phone	075-11897642
cell	0919-951-5198
id	{ 2 fields }
picture	{ 3 fields }
nat	IR

با توجه به اینکه مطمئن شدیم که عملکرد روش ما درست است، در نتیجه با کوئری زیر، پسورد تمام کاربران را به روز رسانی می‌کنیم (البته قبل از آن، پسورد کاربر بالا را به صورت قبل برمی‌گردانیم):

```
db.getCollection("users").find().forEach(function(data){
  data.login.password = hex_md5(data.login.password)
  db.getCollection("users").save(data)
});
```

همانطور که از خروجی username و password چند کاربر اول کالکشن users به صورت زیر مشخص است، پسورد کاربران تغییر کرده است:

```
1 {
2   "login" : {
3     "username" : "smallrabbit981",
4     "password" : "5d1fb338b351a63a1720b837cc9010ff"
5   }
6 }
7 {
8   "login" : {
9     "username" : "beautifulmeercat896",
10    "password" : "507fcbb50bf9e88168dcd0d083600c0b"
11  }
12 }
13 {
14   "login" : {
15     "username" : "whiterabbit388",
16     "password" : "7bf4a177e86772277c63b6d5352741"
17   }
18 }
```

زمان اجرای کوئری به روز رسانی پسورد تمام کاربران نیز 2 دقیقه و 2 ثانیه و 50 صدم ثانیه به طول انجامید.

گام سوم تمرین – دستورات تجمعی و آماری

1. ابتدا بایستی کاربران را به سه گروه سنی کمتر از 16 (نوجوان) / بین 16 تا 30 (جوان) / بالاتر از 30 (میانسال به بالا) تقسیم کنید، سپس دستوری بنویسید که تعداد هر گروه را به ما برگرداند:

کوئری:

```
1 db.getCollection("users").aggregate(
2   [
3     {
4       "$project" : {
5         "AgeGroup" : {
6           "$cond" : {
7             "if" : { "$lt" : [ "$dob.age", 16.0 ] },
8             "then" : "teenager",
9             "else" : {
10              "$cond" : {
11                "if" : { "$lte" : [ "$dob.age", 30.0 ] },
12                "then" : "young",
13                "else" : "middle-aged"
14              }
15            }
16          }
17        }
18      },
19    ],
20    {
21      "$group" : {
22        "_id" : "$AgeGroup",
23        "count" : { "$sum" : 1.0 }
24      }
25    },
26    {
27      "$project" : {
28        "_id" : 0.0,
29        "age-group" : "$_id",
30        "count" : { "$toInt" : "$count" }
31      }
32    }
33  ]
34 )
```

خروجی:

Tree		JSON
Key	Value	
▼ (1) { _id : } "-" age-group count	{ 2 fields } young 16006	<pre>1 { 2 "age-group" : "young", 3 "count" : NumberInt(16006) 4 } 5 {</pre>
▼ (2) { _id : } "-" age-group count	{ 2 fields } middle-aged 83994	<pre>6 "age-group" : "middle-aged", 7 "count" : NumberInt(83994) 8 }</pre>

0.389s

زمان اجرا:

2. تعداد کاربران هر استان را به تفکیک تولید کنید:

کوئری:

```
1 db.getCollection("users").aggregate(
2   [
3     {
4       "$group" : {
5         "_id" : "$location.state",
6         "count" : { "$sum" : 1.0 }
7       }
8     },
9     {
10      "$project" : {
11        "_id" : 0.0,
12        "state" : "$_id",
13        "count" : { "$toInt" : "$count" }
14      }
15    }
16  ]
17 )
```

خروجی:

Tree		JSON
Key	Value	
▼ (1) {_id: }	{ 2 fields }	1 {
state	قم	2 "state" : "قم",
count	3253	3 "count" : NumberInt(3253)
▼ (2) {_id: }	{ 2 fields }	4 }
state	آذربایجان غربی	5 {
count	3186	6 "state" : "آذربایجان غربی",
▼ (3) {_id: }	{ 2 fields }	7 "count" : NumberInt(3186)
state	سیستان و بلوچستان	8 }
count	3239	9 {
▼ (4) {_id: }	{ 2 fields }	10 "state" : "سیستان و بلوچستان",
state	خراسان رضوی	11 "count" : NumberInt(3239)
count	3324	12 }
▼ (5) {_id: }	{ 2 fields }	13 {
state	اردبیل	14 "state" : "خراسان رضوی",
count	3187	15 "count" : NumberInt(3324)
▼ (6) {_id: }	{ 2 fields }	16 }
state	خراسان شمالی	17 {
count	3196	18 "state" : "اردبیل",
▼ (7) {_id: }	{ 2 fields }	19 "count" : NumberInt(3187)
state	لرستان	20 }
count	3205	21 {
▼ (8) {_id: }	{ 2 fields }	22 "state" : "خراسان شمالی",
state	اصفهان	23 "count" : NumberInt(3196)
count	3244	24 }
		25 {
		26 "state" : "لرستان",
		27 "count" : NumberInt(3205)
		28 }
		29 {
		30 "state" : "اصفهان",
		31 "count" : NumberInt(3244)
		32 }

▼ { } (9) { _id : }	{ 2 fields }	33 {	"state" : "کهگیلویه و بویراحمد",
state	کهگیلویه و بویراحمد	34	"count" : NumberInt(3116)
count	3116	35 }	
▼ { } (10) { _id : }	{ 2 fields }	36 {	
state	یزد	37	"state" : "یزد",
count	3262	38	"count" : NumberInt(3262)
▼ { } (11) { _id : }	{ 2 fields }	39 }	
state	گلستان	40 {	
count	3297	41	"state" : "گلستان",
▼ { } (12) { _id : }	{ 2 fields }	42	"count" : NumberInt(3297)
state	کرمانشاه	43 }	
count	3178	44 {	
▼ { } (13) { _id : }	{ 2 fields }	45	"state" : "کرمانشاه",
state	هرمزگان	46	"count" : NumberInt(3178)
count	3261	47 }	
▼ { } (14) { _id : }	{ 2 fields }	48 {	
state	ایلام	49	"state" : "هرمزگان",
count	3356	50	"count" : NumberInt(3261)
▼ { } (15) { _id : }	{ 2 fields }	51 }	
state	مازندران	52 {	
count	3290	53	"state" : "ایلام",
▼ { } (16) { _id : }	{ 2 fields }	54	"count" : NumberInt(3356)
state	چهارمحال و بختیاری	55 }	
count	3206	56 {	
▼ { } (17) { _id : }	{ 2 fields }	57	"state" : "مازندران",
state	خراسان جنوبی	58	"count" : NumberInt(3290)
count	3205	59 }	
▼ { } (18) { _id : }	{ 2 fields }	60 {	
state	گیلان	61	"state" : "چهارمحال و بختیاری",
count	3268	62	"count" : NumberInt(3206)
▼ { } (19) { _id : }	{ 2 fields }	63 }	
state	مرکزی	64 {	
count	3205	65	"state" : "خراسان جنوبی",
▼ { } (20) { _id : }	{ 2 fields }	66	"count" : NumberInt(3205)
state	همدان	67 }	
count	3207	68 {	
▼ { } (21) { _id : }	{ 2 fields }	69	"state" : "گیلان",
state	آذربایجان شرقی	70	"count" : NumberInt(3268)
count	3185	71 }	
▼ { } (22) { _id : }	{ 2 fields }	72 {	
state	بوشهر	73	"state" : "مرکزی",
count	3170	74	"count" : NumberInt(3205)
▼ { } (23) { _id : }	{ 2 fields }	75 }	
state	کردستان	76 {	
count	3241	77	"state" : "همدان",
▼ { } (24) { _id : }	{ 2 fields }	78	"count" : NumberInt(3207)
state	خوزستان	79 }	
count	3222	80 {	
		81	"state" : "آذربایجان شرقی",
		82	"count" : NumberInt(3185)
		83 }	
		84 {	
		85	"state" : "بوشهر",
		86	"count" : NumberInt(3170)
		87 }	
		88 }	

<div> <div> <div></div> <div>(25) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>کرمان</div> </div> </div> <div> <div>count</div> <div>3284</div> </div> </div>	<div> <div>89 {</div> <div>90 "state" : "کرمان",</div> <div>91 "count" : NumberInt(3241)</div> <div>92 }</div> </div>
<div> <div> <div></div> <div>(26) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>تهران</div> </div> </div> <div> <div>count</div> <div>3203</div> </div> </div>	<div> <div>93 {</div> <div>94 "state" : "تهران",</div> <div>95 "count" : NumberInt(3222)</div> <div>96 }</div> </div>
<div> <div> <div></div> <div>(27) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>فارس</div> </div> </div> <div> <div>count</div> <div>3215</div> </div> </div>	<div> <div>97 {</div> <div>98 "state" : "کرمان",</div> <div>99 "count" : NumberInt(3284)</div> <div>100 }</div> </div>
<div> <div> <div></div> <div>(28) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>زنجان</div> </div> </div> <div> <div>count</div> <div>3225</div> </div> </div>	<div> <div>101 {</div> <div>102 "state" : "تهران",</div> <div>103 "count" : NumberInt(3203)</div> <div>104 }</div> </div>
<div> <div> <div></div> <div>(29) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>قزوین</div> </div> </div> <div> <div>count</div> <div>3187</div> </div> </div>	<div> <div>105 {</div> <div>106 "state" : "فارس",</div> <div>107 "count" : NumberInt(3215)</div> <div>108 }</div> </div>
<div> <div> <div></div> <div>(30) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>البرز</div> </div> </div> <div> <div>count</div> <div>3196</div> </div> </div>	<div> <div>109 {</div> <div>110 "state" : "زنجان",</div> <div>111 "count" : NumberInt(3225)</div> <div>112 }</div> </div>
<div> <div> <div></div> <div>(31) {_id: }</div> </div> <div> <div>{ 2 fields }</div> <div> <div>state</div> <div>سمنان</div> </div> </div> <div> <div>count</div> <div>3187</div> </div> </div>	<div> <div>113 {</div> <div>114 "state" : "قزوین",</div> <div>115 "count" : NumberInt(3187)</div> <div>116 }</div> <div>117 {</div> <div>118 "state" : "البرز",</div> <div>119 "count" : NumberInt(3196)</div> <div>120 }</div> <div>121 {</div> <div>122 "state" : "سمنان",</div> <div>123 "count" : NumberInt(3187)</div> <div>124 }</div> </div>

0.247s

زمان اجرا:

3. فیلد شماره موبایل افرادی که آفست timezone آنها برابر +5:00 را حذف نمایید، سپس تعداد افرادی را بیابید که شماره موبایل ندارند.

کوئری حذف شماره موبایل:

```
1 db.getCollection("users").update(
2   {
3     "location.timezone.offset": "+5:00"
4   },
5   [{
6     $unset: "cell"
7   }],
8   {
9     "multi": true
10  }
11 )
```

کوئری جست و جوی تعداد افراد:

```
1 db.getCollection("users").aggregate(
2   [
3     {
4       "$match" : {
5         "cell" : { "$exists" : false }
6       }
7     },
8     {
9       "$group" : {
10        "_id" : null,
11        "count" : { "$sum" : 1 }
12      }
13    },
14    {
15      "$project" : {
16        "_id" : 0,
17        "count" : { "$toInt" : "$count" }
18      }
19    }
20  ]
21 )
```

خروجی:

Tree		JSON
Key	Value	1 {
▼ (1) { _id : }	{ 1 fields }	2 "count" : NumberInt(3398)
count	3398	3 }

0.096s

زمان اجرا:

4. میانگین سن کاربران استان تهران را با میانگین سن کاربران سایر استان‌ها مقایسه کنید:

کوئری:

درک اینجانب از سوال، این است که میانگین سن کاربران استان تهران را با میانگین سن دیگر کاربران مقایسه کنیم. به همین دلیل، کوئری ما به صورت زیر می‌باشد:

```
1 db.getCollection("users").aggregate(  
2   [  
3     {  
4       "$project" : {  
5         "state" : {  
6           "$cond" : {  
7             "if" : { "$eq" : [ "$location.state", "تهران" ] },  
8             "then" : "تهران",  
9             "else" : "سایر"  
10          }  
11        },  
12        "age" : "$dob.age"  
13      }  
14    },  
15    {  
16      "$group" : {  
17        "_id" : "$state",  
18        "avgAge" : { "$avg" : "$age" }  
19      }  
20    },  
21    {  
22      "$project" : {  
23        "_id" : 0.0,  
24        "state" : "$_id",  
25        "avgAge" : "$avgAge"  
26      }  
27    }  
28  ]  
29 )
```

خروجی:

Tree		JSON
Key	Value	
✓ (1) { _id : }	{ 2 fields }	1 {
"state"	سایر	2 "state" : "سایر",
avgAge	48.87177288552331	3 "avgAge" : 48.87177288552331
✓ (2) { _id : }	{ 2 fields }	4 }
"state"	تهران	5 {
avgAge	48.39993755853887	6 "state" : "تهران",
		7 "avgAge" : 48.39993755853887
		8 }

همانطور که مشاهده می‌شود، میانگین سن کاربران استان تهران از میانگین سن دیگر کاربران کمتر است.

0.425s

زمان اجرا:

کوئری: ولی اگر هدف سوال از مقایسه این است که مشخص کنیم که چه استان‌هایی، میانگین سن کمتر، چه استان‌هایی، میانگین سن برابر و چه استان‌هایی، میانگین سن بیشتر از استان تهران را دارند، کوئری آن به صورت زیر می‌باشد:

```
1 db.getCollection("users").aggregate(  
2   [  
3     {  
4       "$group" : {  
5         "_id" : "$location.state",  
6         "avgAge" : { "$avg" : "$dob.age" }  
7       }  
8     },  
9     {  
10      "$facet" : {  
11        "tehran" : [  
12          {  
13            "$match" : { "$expr" : { "$eq" : [ "$_id", "تهران" ] }}  
14          },  
15          {  
16            "$project" : { "avg" : "$avgAge" }  
17          }  
18        ],  
19        "states" : [  
20          {  
21            "$match" : { "$expr" : { "$ne" : [ "$_id", "تهران" ] }}  
22          },  
23          {  
24            "$project" : { "_id" : 1.0, "avgAge" : 1.0 }  
25          }  
26        ]  
27      }  
28    },  
29    {  
30      "$facet" : {  
31        "less" : [  
32          {  
33            "$match" : { "$expr" : { "$lt" : [ "$states.avgAge", "$tehran.avg" ] }}  
34          },  
35          {  
36            "$group" : { "_id" : "$states", "states" : { "$addToSet" : "$states._id" }}  
37          }  
38        ],  
39        "eq1" : [  
40          {  
41            "$match" : { "$expr" : { "$eq" : [ "$states.avgAge", "$tehran.avg" ] }}  
42          },  
43          {  
44            "$group" : { "_id" : "$states", "states" : { "$addToSet" : "$states._id" }}  
45          }  
46        ],  
47        "grt" : [  
48          {  
49            "$match" : { "$expr" : { "$gt" : [ "$states.avgAge", "$tehran.avg" ] }}  
50          },  
51          {  
52            "$group" : { "_id" : "$states", "states" : { "$addToSet" : "$states._id" }}  
53          }  
54        ]  
55      }  
56    },  
57    {  
58      "$project" : {  
59        "less_cities" : { "$arrayElemAt" : [{ "$arrayElemAt": [ "$less.states", 0.0 ] }, 0.0 ] },  
60        "equal_cities" : { "$arrayElemAt" : [{ "$arrayElemAt": [ "$eq1.states", 0.0 ] }, 0.0 ] },  
61        "greater_cities" : { "$arrayElemAt" : [{ "$arrayElemAt": [ "$grt.states", 0.0 ] }, 0.0 ] }  
62      }  
63    }  
64  ]  
65 )
```

خروجی:

Tree		JSON
Key	Value	
▼ { } (1) { _id : }	{ 3 fields }	1 {
less_cities	null	2 "less_cities" : null,
equal_cities	null	3 "equal_cities" : null,
▼ [] greater_cities	[30 elements]	4 "greater_cities" : [
0	سمنان	5 "سمنان",
1	البرز	6 "البرز",
2	قزوین	7 "قزوین",
3	فارس	8 "فارس",
4	زنجان	9 "زنجان",
5	کرمان	10 "کرمان",
6	خوزستان	11 "خوزستان",
7	بوشهر	12 "بوشهر",
8	آذربایجان شرقی	13 "آذربایجان شرقی",
9	همدان	14 "همدان",
10	کردستان	15 "کردستان",
11	گیلان	16 "گیلان",
12	مرکزی	17 "مرکزی",
13	خراسان جنوبی	18 "خراسان جنوبی",
14	چهارمحال و بختیاری	19 "چهارمحال و بختیاری",
15	مازندران	20 "مازندران",
16	هرمزگان	21 "هرمزگان",
17	ایلام	22 "ایلام",
18	کرمانشاه	23 "کرمانشاه",
19	گلستان	24 "گلستان",
20	یزد	25 "یزد",
21	کهگیلویه و بویراحمد	26 "کهگیلویه و بویراحمد",
22	اصفهان	27 "اصفهان",
23	لرستان	28 "لرستان",
24	خراسان شمالی	29 "خراسان شمالی",
25	خراسان رضوی	30 "خراسان رضوی",
26	اردبیل	31 "اردبیل",
27	سیستان و بلوچستان	32 "سیستان و بلوچستان",
28	آذربایجان غربی	33 "آذربایجان غربی",
29	قم	34 "قم",
		35]
		36 }

همانطور که مشاهده می‌شود، میانگین سن کاربران استان تهران از میانگین سن کاربران تمام استان‌های دیگر کمتر است.

0.317s

زمان اجرا:

کوئری:

در اینجا، ابتدا تعداد کاربران هر استان را محاسبه کرده، سپس یک بار، استان‌ها را بر اساس تعداد کاربرانشان به‌صورت نزولی مرتب کرده و اولین استان را برمی‌داریم، و یک بار، استان‌ها را بر اساس تعداد کاربرانشان به‌صورت صعودی مرتب کرده و اولین استان را برمی‌داریم. سپس این دو استان را به‌صورت مناسب در خروجی نمایش می‌دهیم.

خروجی:

0.250s

زمان اجرا: