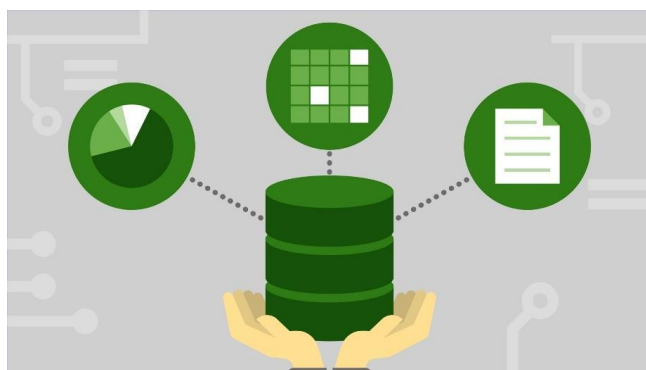


به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



### آزمایشگاه پایگاه داده

دستور کار شماره هشت (کار با مونگودبی)

امیرعلی رایگان

۸۱۰۱۹۷۶۲۳

بهار ۱۴۰۱

## گزارش دستورکار انجام شده

## کار با مונگو دی بی

در ابتدا با استفاده از api سایت randomuser.me به تعداد ۱۰۰ هزار داده ی کاربر تصادفی می گیریم و آنها را با استفاده از کتابخانه ی pymongo به پایگاه داده ای که داریم انتقال می دهیم. عکس از کدی که برای این کار نوشته شد در ادامه آمده است.

```

1 import pymongo
2 import requests, json
3 import time
4 url = "https://randomuser.me/api/?nat=ir"
5 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
6 mydb = myclient['metadate']
7 mycol = mydb['Users']
8 for i in range(1,21):
9     querystring = {"nat":"ir", "results":"5000"}
10    response = requests.request("GET", url, params=querystring)
11    data = json.loads(response.text)
12    print(data)
13    x = mycol.insert_many(data['results'])
14    print(i*5000,"users added to database.")
15    time.sleep(70)
16

```

## مرحله صفرم.

در ابتدا از تعداد داکومننت هایی که به کالکشن users در دیتابیس metadate اضافه شده است اطمینان حاصل می کنیم. تعداد داده ها حدود ۱۳۰ هزار تا می باشد.

```

18 # x = mycol.insert_many(data['results'])
19 # print(i*5000,"users added to database.")
20 # except:
21 #     print('Wait')
22 #     time.sleep(70)
23 Client = MongoClient()
24 myclient = MongoClient('localhost', 27017)
25
26 my_database = myclient['metadate']
27 my_collection = my_database['Users']
28
29 # number of documents in the collection
30 total_count = my_collection.count_documents({})
31 print("Total number of documents : ", total_count)
32

```

## مرحله اول (کوئری روی داده ها).

- نام و نام خانوادگی کاربرانی که سن بالای ۷۸ سال دارند و اهل اهواز نیز هستند.

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure for 'localhost:27017', including 'Databases' and 'Collections'. The 'Users' collection is selected. The main panel shows the 'Documents' tab for 'metadata.Users'. A filter is applied: `{ "location.city": "اهواز", "dob.age" : { "$gte": 78 } }`. The results show two documents, both with 'nat: "IR"'. The first document is for a male user with a phone number '011-58937101' and a cell number '0922-201-4473'. The second document is for a female user with a phone number '055-90821042' and a cell number '0998-726-9532'.

- تمام کاربرانی که در سال ۲۰۱۸ در سایت ما ثبت نام کرده اند.

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure for 'localhost:27017', including 'Databases' and 'Collections'. The 'Users' collection is selected. The main panel shows the 'Documents' tab for 'metadata.Users'. A filter is applied: `{ "registered.date": { "$gte": "2018-01-01T14:15:23.821+00:00", "$lte": "2018-12-28T14:15:23.821+00:00" } }`. The results show two documents, both with 'nat: "IR"'. The first document is for a female user with a phone number '055-90821042' and a cell number '0998-726-9532'. The second document is for a male user with a phone number '055-90821042' and a cell number '0998-726-9532'.

- تمام کسانی که در کوچه ای با نام آیت الله طالقانی ساکن هستند و اسمشان امیرحسین است.

MongoDB Compass - localhost:27017/metadata.Users

Connect View Collection Help

localhost:27017

4 DBS 4 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

admin

config

local

startup\_log

metadata

Users

Documents metadata.Users

130.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"name.first":"امیرحسین","location.street.name":"آیت الله طالقانی"}

ADD DATA VIEW

Displaying documents 1 - 7 of 7

```

{
  "_id": ObjectId("62a08ff2c4c3b2f37c46eb11"),
  "gender": "male",
  "name": Object,
  "location": Object,
  "email": "myrhsyn.mwsvy@example.com",
  "login": Object,
  "dob": Object,
  "registered": Object,
  "phone": "015-64779351",
  "cell": "0986-036-3558",
  "id": Object,
  "picture": Object,
  "nat": "IR"
}

```

>\_MONGOSH

8:31 PM 6/8/2022

- در جهت تشخیص کسانی که رمز ساده برای اکانتشان انتخاب کرده اند، تمام کسانی که رمز ساده ی ۱۲۳۴۵۶۷۸ برای اکانتشان برگزیده اند مشخص شده است.

MongoDB Compass - localhost:27017/metadata.Users

Connect View Collection Help

localhost:27017

4 DBS 4 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

admin

config

local

startup\_log

metadata

Users

Documents metadata.Users

130.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"login.password":"12345678"}

ADD DATA VIEW

Displaying documents 1 - 20 of 22

```

{
  "_id": ObjectId("62a08ff2c4c3b2f37c46f136"),
  "gender": "female",
  "name": Object,
  "location": Object,
  "email": "mrl.zraay@example.com",
  "login": Object,
  "dob": Object,
  "registered": Object,
  "phone": "088-52968043",
  "cell": "0999-340-7118",
  "id": Object,
  "picture": Object,
  "nat": "IR"
}

```

>\_MONGOSH

8:37 PM 6/8/2022

- تمام خانم های مجردی که در تهران زندگی می کنند و پیش شماره تلفنشان ۰۹۱۲ است.

MongoDB Compass - localhost:27017/metadata.Users

Connect View Collection Help

localhost:27017

4 DBS 4 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

admin

config

local

startup\_log

metadata

Users

Documents metadata.Users

130.0k 1 DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"gender":"female","cell":{"regex":"/^0912/i"}, "location.city":"تهران", "name.title":"Miss"} OPTIONS FIND RESET

ADD DATA VIEW

Displaying documents 1 - 4 of 4 REFRESH

```

{
  "_id": ObjectId("62a09030e4fef54f3c419a48"),
  "gender": "female",
  "name": Object,
  "location": Object,
  "email": "khymy.nkwnzr@example.com",
  "login": Object,
  "dob": Object,
  "registered": Object,
  "phone": "098-46262307",
  "cell": "0912-381-5612",
  "id": Object,
  "picture": Object,
  "nat": "IR"
}

```

> \_MONGOSH

## مرحله دوم (اضافه کردن فیلدی برای تاریخ شمسی).

برای این کار ابتدا تاریخ میلادی را که به صورت string ذخیره شد تبدیل به فرمت date می کنیم تا بتوانیم عملیات ریاضی روی آن انجام دهیم تا تبدیل به تاریخ شمسی شود.

سپس تاریخ به دست آمده را به اندازه ۲۲۶۸۹۹ روز به عقب می کشیم تا تاریخ شمسی به دست آید. در نهایت عدد جدید به دست آمده را در فیلدی جدید ذخیره می کنیم.

MongoDB Compass - localhost:27017/metadata.Users

Connect View Collection Help

localhost:27017

4 DBS 4 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

admin

config

local

startup\_log

metadata

Users

Aggregations metadata.Users

130.0k 1 DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Pipeline \$set Explain Export Run More Options

persian year - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

AUTO PREVIEW

```

1 * {
2   "$set": {
3     "year_persian": {
4       "$subtract": [
5         {"$dateFromString": [{"dateString": "$dob.date"}]},
6         {"$dateFromString": [{"dateString": "$registered.date"}]}
7       ]
8     },
9     "year_persian": {
10      "$subtract": [
11        {"$dateFromString": [{"dateString": "$year_persian.date"}]},
12        {"$dateFromString": [{"dateString": "$registered.date"}]}
13      ]
14    }
15  }
16 }

```

Output after \$set stage (Sample of 10 documents)

```

{
  "date": "1946-11-03T01:06:57.843Z",
  "age": 76,
  "year_persian": 1325-08-11T01:06:57.843+00:00,
  "persian_year": Object,
  "registered": Object,
  "date": "2004-08-28T11:36:10.054Z",
  "age": 18,
  "year_persian": 1383-06-06T11:36:10.054+00:00,
  "phone": "072-13816968",
  "cell": "0928-953-2316"
}

```

ADD STAGE

> \_MONGOSH

## مرحله سوم (ذخیره پسورد).

چون ذخیره کردن پسورد کاربران به صورت خام در پایگاه داده کار غیر حرفه ای است و امکان لو رفتن داده ها را فراهم می کند باید از راه جایگزین استفاده کنیم. راه جایگزین استفاده از توابع یک طرفه رمز نگاری است. در این تمرین ابتدا امتحان می کنیم که اگر برای یک کاربر در فیلد password به جای رمز کاربر تابع هش sha256 آن را نگه داریم ، همچنان می توانیم آن کاربر را پیدا کنیم یا خیر. اگر موفقیت آمیز بود همین کار را برای تمام کاربران انجام می دهیم.

برای این کار پایگاه داده را به پایتون متصل می کنیم و با استفاده از کتابخانه ی hashlib و تابع sha256 داخل آن رمز کاربران را جایگزین می کنیم.

ابتدا برای یک کاربر امتحان می کنیم.

```

33 password = 'pass1234'
34 mydoc = my_collection.find({"login.username":username, "login.password":password})
35 print(hashlib.sha256(password.encode()).hexdigest())
36 hashed = hashlib.sha256(password.encode()).hexdigest()
37 print("Login successful")
38 print("User id is", mydoc[0]['_id'])
39 print("User name is", mydoc[0]['name'])
40 print("User id is", mydoc[0]['email'])
41 user_id = mydoc[0]['_id']
42 mydoc = my_collection.update_one({"_id":user_id}, {"$set":{"login.password":hashed}})
43 mydoc = my_collection.find({"login.username":username, "login.password":hashlib.sha256(password.encode()).hexdigest()})
44 print("Login successful")
45 print("User id is", mydoc[0]['_id'])
46 print("User name is", mydoc[0]['name'])
47 print("User id is", mydoc[0]['email'])
48 print("User password is", mydoc[0]['login']['password'])

pData\Local\Programs\Python\Python310\python.exe "c:\Users\amira\.vscode\extensions\ms-python.python-2022.6.3\pythonFiles\
lib\python\debugpy\launcher" '9673' '-' 'd:\University\DataBase Lab\Lab 7,8\lab8.py'
bd94dcda26fccb4e68da31f9b5aac0b571ae266d822620e901ef7ebe3a11d4f
Login successful!
User id is 62a08ff2c4c3b2f37c46eb12
User name is {'title': 'Mr', 'first': 'میرزا', 'last': 'ردی'}
User id is shyn.sdr@example.com
Login successful!
User id is 62a08ff2c4c3b2f37c46eb12
User name is {'title': 'Mr', 'first': 'میرزا', 'last': 'ردی'}
User id is shyn.sdr@example.com
User password is bd94dcda26fccb4e68da31f9b5aac0b571ae266d822620e901ef7ebe3a11d4f
PS D:\University\DataBase Lab\Lab 7,8>
  
```

سپس برای تمام کاربران این کار را انجام می دهیم. زمانی در حدود چند دقیقه برای آپدیت اطلاعات تمام کاربران صرف شد.

```

44 # print("Login successful")
45 # print("User id is", mydoc[0]['_id'])
46 # print("User name is", mydoc[0]['name'])
47 # print("User id is", mydoc[0]['email'])
48 # print("User password is", mydoc[0]['login']['password'])
49
50 mydoc = my_collection.find({})
51 for i in mydoc:
52     user = my_collection.find({"_id":i['_id']})
53     hashed = hashlib.sha256(user[0]['login']['password'].encode()).hexdigest()
54     my_collection.update_one({"_id":i['_id']}, {"$set":{"login.password":hashed}})
55
56 #
57 # bd_date = x['dob']['date']
58 # year = int(bd_date[:4])
59 # month = int(bd_date[5:7])
60 # day = int(bd_date[8:10])

PS D:\University\DataBase Lab\Lab 7,8> d:; cd 'd:\University\DataBase Lab\Lab 7,8'; & 'C:\Users\amira\AppData\Local\Progra
ms\Python\Python310\python.exe' 'c:\Users\amira\.vscode\extensions\ms-python.python-2022.6.3\pythonFiles\lib\python\debugpy
\launcher' '10546' '-' 'd:\University\DataBase Lab\Lab 7,8\lab8.py'
  
```

## مرحله چهارم (دستورات تجمعی/آماری).

می‌خواهیم افراد را بر اساس سنشان به ۳ دسته قسمت کنیم. قسمت اول کمتر از ۲۵ سال سن دارند. (چون در داده‌هایی که داریم سن زیر ۲۴ سال کلاً نداریم). قسمت دوم بین ۲۴ تا ۳۰ و قسمت آخر بزرگ‌تر از ۳۰ سال. برای این کار از توابع aggregate استفاده می‌کنیم.

MongoDB Compass - localhost27017/metadata.Users

Aggregations metadata.Users

130.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Pipeline \$project \$group \$project

persian year - modified SAVE + CREATE NEW EXPORT TO LANGUAGE AUTO PREVIEW

Output after \$project stage (Sample of 10 documents)

```

1 {
2   "age_groups": {
3     "$cond": {
4       "if": { "$lt": [ "$dob.age", 25.0 ] },
5       "then": "teenager",
6       "else": {
7         "$cond": {
8           "if": { "$lte": [ "$dob.age", 30.0 ] },
9           "then": "young",
10          "else": "middle-aged"
11        }
12      }
13    }
14  }
15 }
  
```

Sample Document 1:

```

{ "_id": ObjectId('62a08ff2c4c3b2f37c46eb15'), "age_groups": "middle-aged" }
  
```

Sample Document 2:

```

{ "_id": ObjectId('62a08ff2c4c3b2f37c46eb15'), "age_groups": "middle-aged" }
  
```

MongoDB Compass - localhost27017/metadata.Users

Aggregations metadata.Users

130.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Pipeline \$project \$group \$project Edit

Showing 1 - 3 count results VIEW

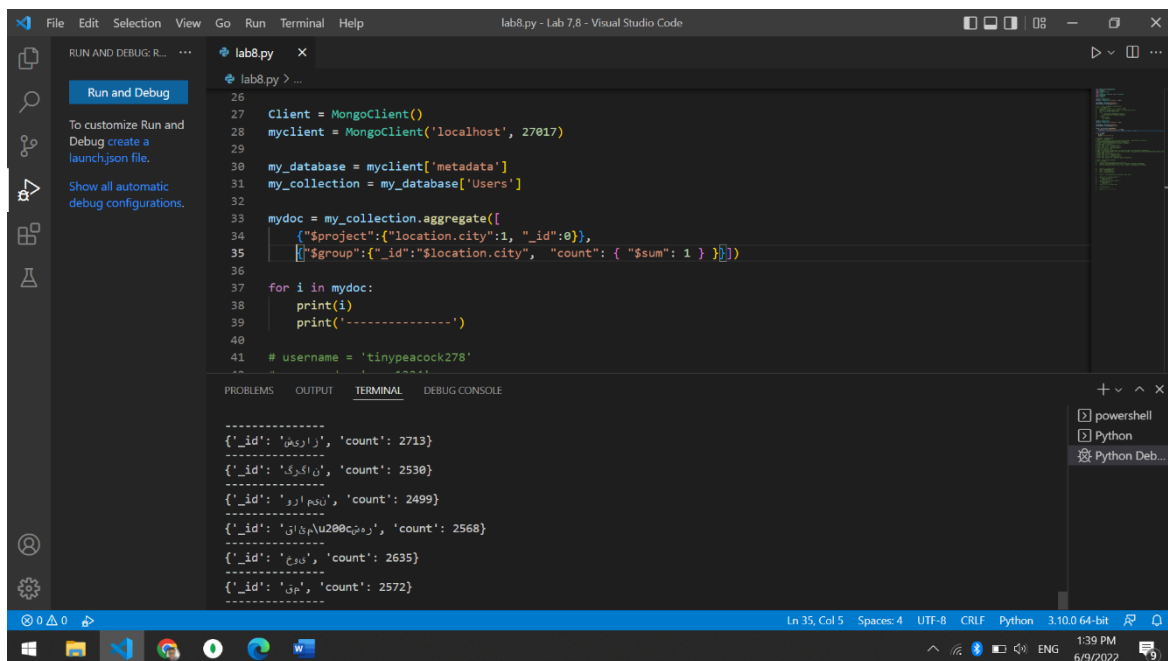
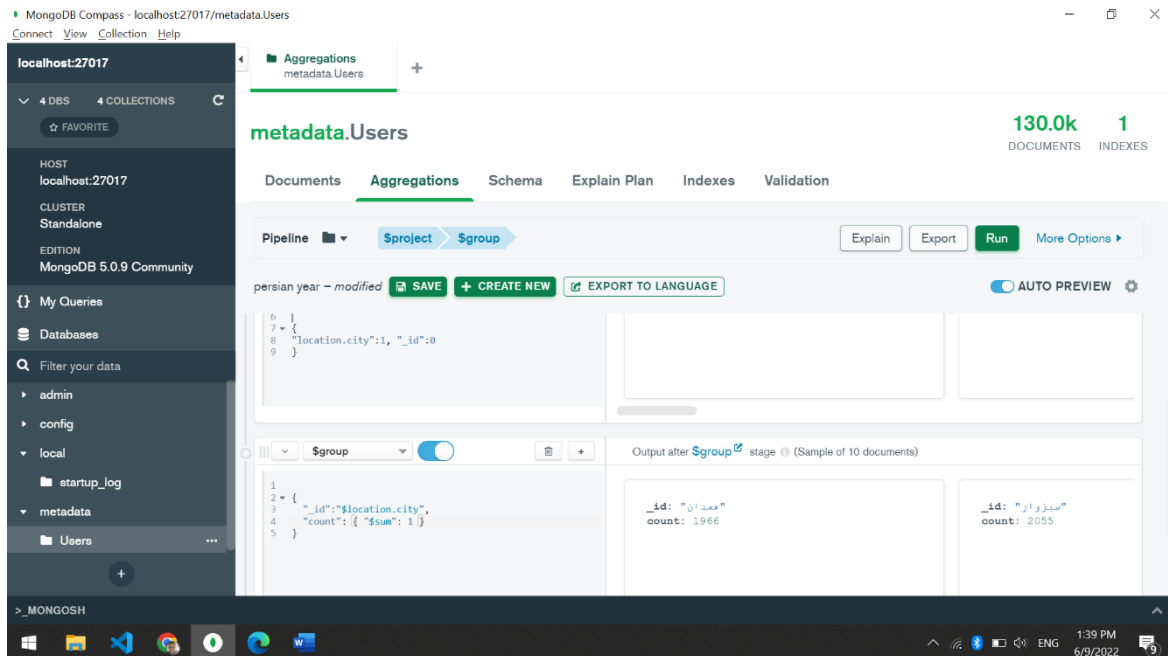
age-group	count
age-group: "teenager"	count: 1792
age-group: "middle-aged"	count: 113602
age-group: "young"	count: 14606



می خواهیم تعداد کاربران هر استان را به تفکیک محاسبه کنیم. برای این کار از دستور زیر استفاده می کنیم.

در این قسمت با استفاده از دستورات aggregate چنین کاری را انجام دادیم. ابتدا شهر ها را انتخاب و سپس گروه بندی و بعد از آن فیلدی برای محاسبه ی تعداد آنها انتخاب کردیم.

این کار هم در mongodb compass و هم در کد پایتون پیاده سازی شد و عکس هایی از آنها در ادامه آمده اند.





## مشکلات و توضیحات تکمیلی

---

مشکلی به وجود نیامد.

## آنچه آموختم / پیشنهادات

---

تجربه خوب از کار با مونگو دی بی.