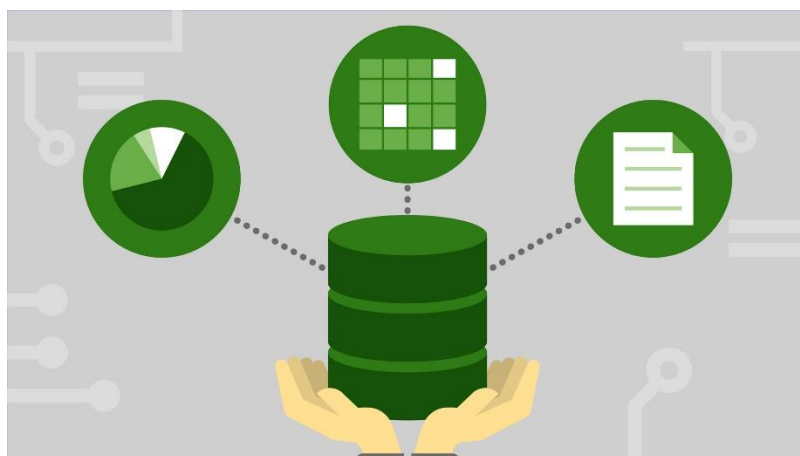


به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



## آزمایشگاه پایگاه داده

دستورکار شماره ۷

شماره دانشجویی

۸۱۰۱۹۵۵۲۶

دی ۹۹

پارسا صدری سینکی

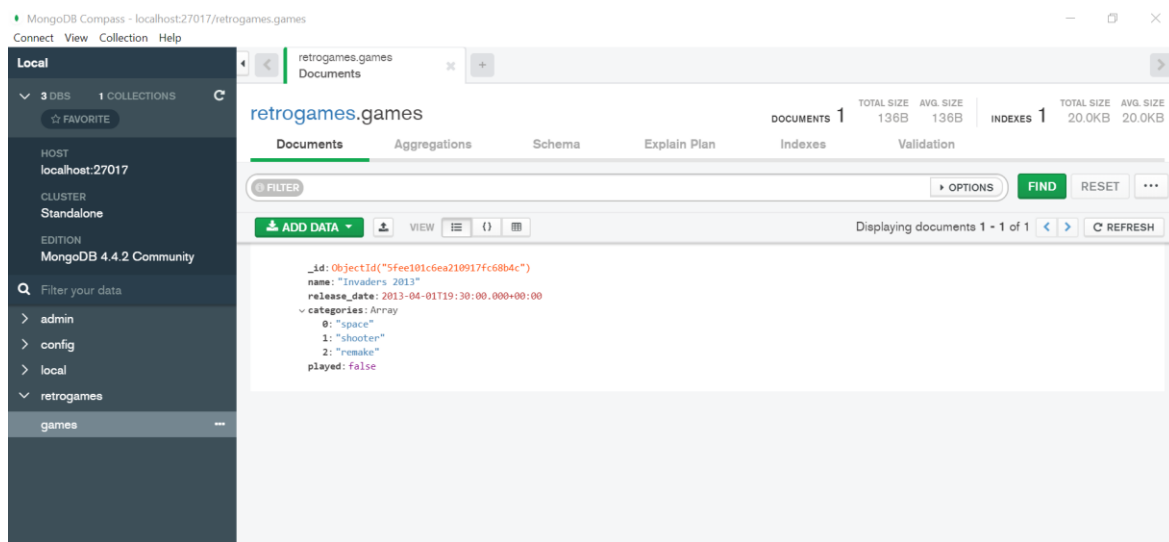
## گزارش فعالیت‌های انجام شده

ابتدا دیتابیس بازی را در پاورشل می‌سازیم.

تصویر زیر مربوط به ساخت دیتابیس بازی هاست.

```
Windows PowerShell
> use retrogames
switched to db retrogames
> game1 =
...
{ name: "Invaders 2013",
  release_date: new Date(2013, 03, 02),
  categories: ["space", "shooter", "remake"],
  played: false
}
...
{
  "name" : "Invaders 2013",
  "release_date" : ISODate("2013-04-01T19:30:00Z"),
  "categories" : [
    "space",
    "shooter",
    "remake"
  ],
  "played" : false
}
> db.games.save(game1)
writeResult({ "ninserted" : 1 })
>
```

نتیجه دستور بالا را در تصویر زیر می‌توانیم مشاهده کنیم.



تصاویر بعدی کوئری برای یافتن داکيومنت‌های داخل دیتابیس است.

```
Windows PowerShell
> db.games.find()
{ "_id" : ObjectId("5fee101c6ea210917fc68b4c"), "name" : "Invaders 2013", "release_date" : ISODate("2013-04-01T19:30:00Z"), "categories" : [ "space", "shooter", "remake" ], "played" : false }
>

> db.games.findOne({ name: "Invaders 2013"})
{
  "_id" : ObjectId("5fee101c6ea210917fc68b4c"),
  "name" : "Invaders 2013",
  "release_date" : ISODate("2013-04-01T19:30:00Z"),
  "categories" : [
    "space",
    "shooter",
    "remake"
  ],
  "played" : false
}
```

تصویر بعدی ساخت دیتابیس بازیکن ها است.

```
> player1 =
...   { name: "PUZZLEGAMESMASTER",
...     gender: "male",
...     scores: [
...       { game_id: new ObjectId("5fee101c6ea210917fc68b4c"),
...         game_name: "Invaders 2013",
...         score: 10500,
...         score_date: new Date(2013, 03, 02)
...       }
...     ]
...   }
... }
... {
...   "name" : "PUZZLEGAMESMASTER",
...   "gender" : "male",
...   "scores" : [
...     {
...       "game_id" : ObjectId("5fee101c6ea210917fc68b4c"),
...       "game_name" : "Invaders 2013",
...       "score" : 10500,
...       "score_date" : ISODate("2013-04-01T19:30:00Z")
...     }
...   ]
... }
> db.players.save(player1)
writeResult({ "nInserted" : 1 })
>
```

دستور بعدی برای ثبت امتیاز بازیکن در بازی مورد نظر هست

```
Windows PowerShell
> db.players.find(
... { "_id" : ObjectId("5fee29a86ea210917fc68b4d"), "name" : "PUZZLEGAMESMASTER", "gender" : "male", "scores" : [ { "game_id" : ObjectId("5fee101c6ea210917fc68b4c"), "game_name" : "Invaders 2013", "score" : 10500, "score_date" : ISODate("2013-04-01T19:30:00Z") } ] }
... )
> db.players.update(
... { "_id" : new ObjectId("5fee29a86ea210917fc68b4d") },
... { $push: { scores: { game_id: new ObjectId("51e10c50085977bc3cd92a65"),
...   game_name: "Invaders 2013",
...   score: 30250,
...   score_date: new Date(2013, 03, 03)
... } }
... },
... { upsert: true }
... )
writeResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

در بخش دوم ما دیتابیس های قبلی را توسط برنامه Studio 3T می سازیم.  
ساخت دیتابیس بازی:

The screenshot shows the Studio 3T for MongoDB interface. On the left, the 'Collections' pane shows a database named 'retrogames2' with a collection named 'games'. The main window displays the MongoDB Shell with the following commands:

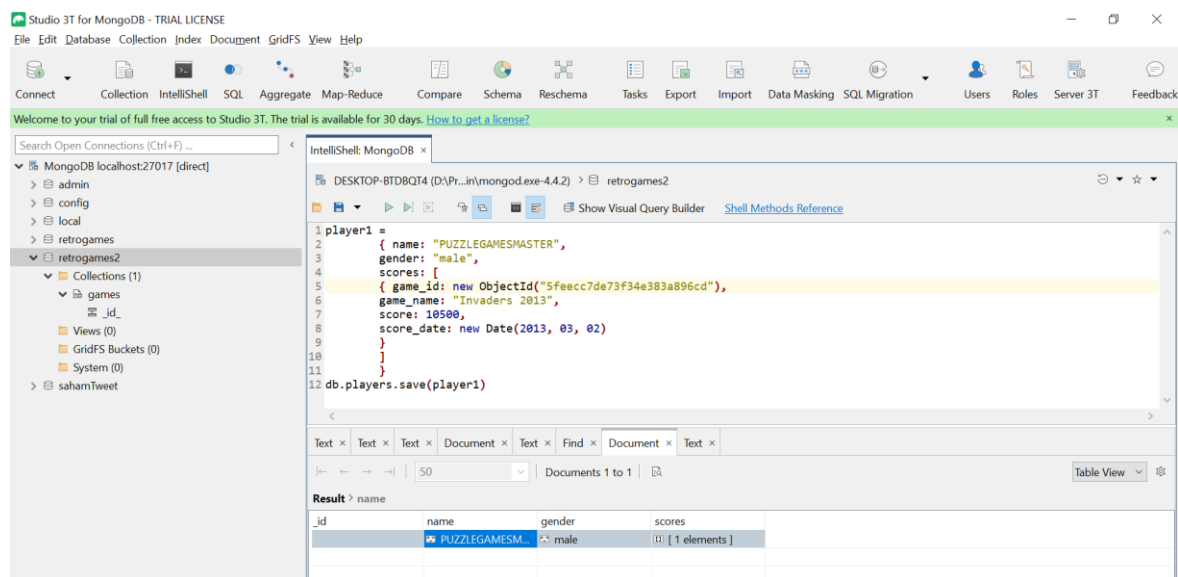
```
1 use retrogames2
2 game1 =
3   { name: "Invaders 2013",
4     release_date: new Date(2013, 03, 02),
5     categories: ["space", "shooter", "remake"],
6     played: false
7   }
8
9 db.games.save(game1)
```

Below the shell, the 'Result' pane shows a table with the following data:

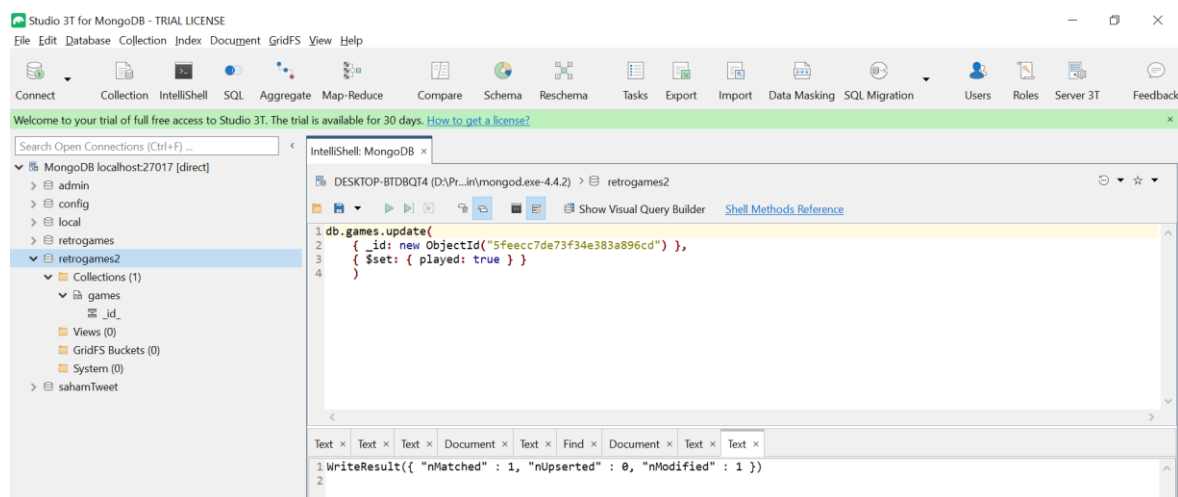
_id	name	release_date	categories	played
51e10c50085977bc3cd92a65	Invaders 2013	2013-04-01T19:30:00Z	[ 3 elements ]	false

The bottom status bar indicates '1 document selected'.

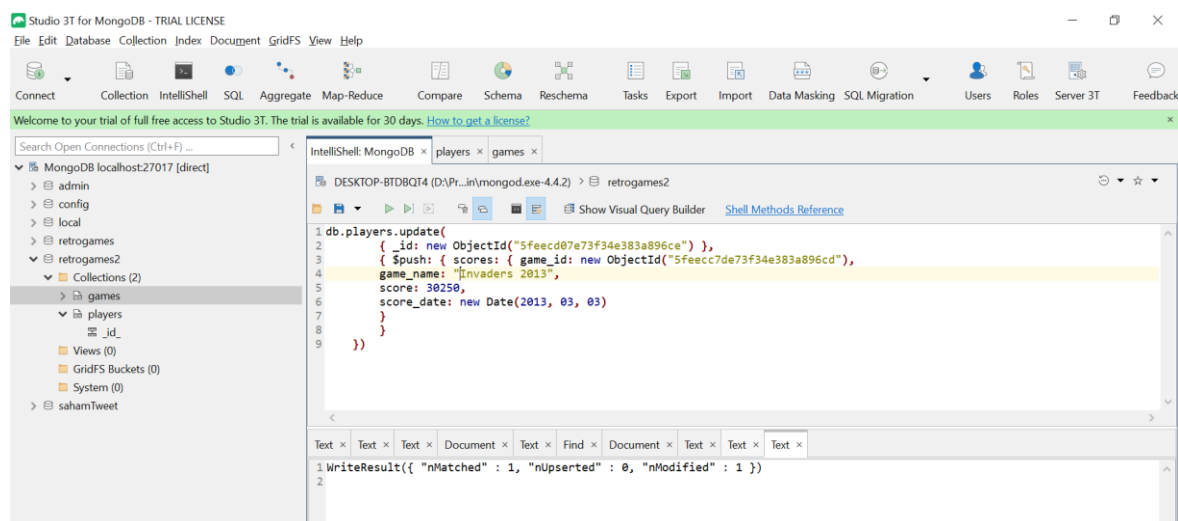
## ساخت دیتابیس بازیکن:



ثبت بازی کردن بازی مورد نظر توسط بازیکن:



و در نهایت ثبت امتیاز گرفته شده بازیکن در بازی مورد نظر:



## بخش سوم

در این قسمت ما ۱۰۰۰ توییت مربوط به بورس را از API سایت sahamyab دریافت می کنیم.  
برای این کار از کد پایتون مقابل استفاده می کنیم.

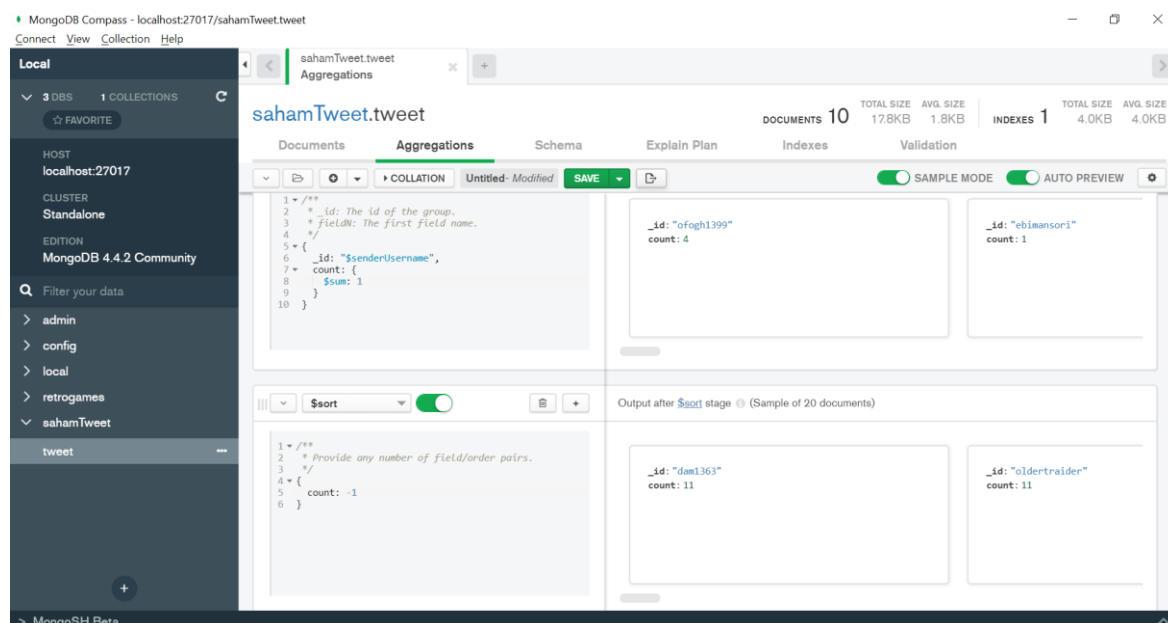
```
import requests
import json
import time
from pymongo import MongoClient

client = MongoClient()
db = client['sahamTweet']
url = "https://www.sahamyab.com/guest/twiter/list"
fetched = 0
total = 1000
api_sleep = 2
while fetched < total:
    print(fetched)
    response = requests.request('GET', url, params={'v':'0.1'})
    if response.status_code == requests.codes.ok:
        data = response.json()['items']
        for tweet in data:
            db.tweet.replace_one({"id" : tweet['id']}, tweet, upsert=True)
        fetched = db.tweet.estimated_document_count()
    time.sleep(api_sleep)
```

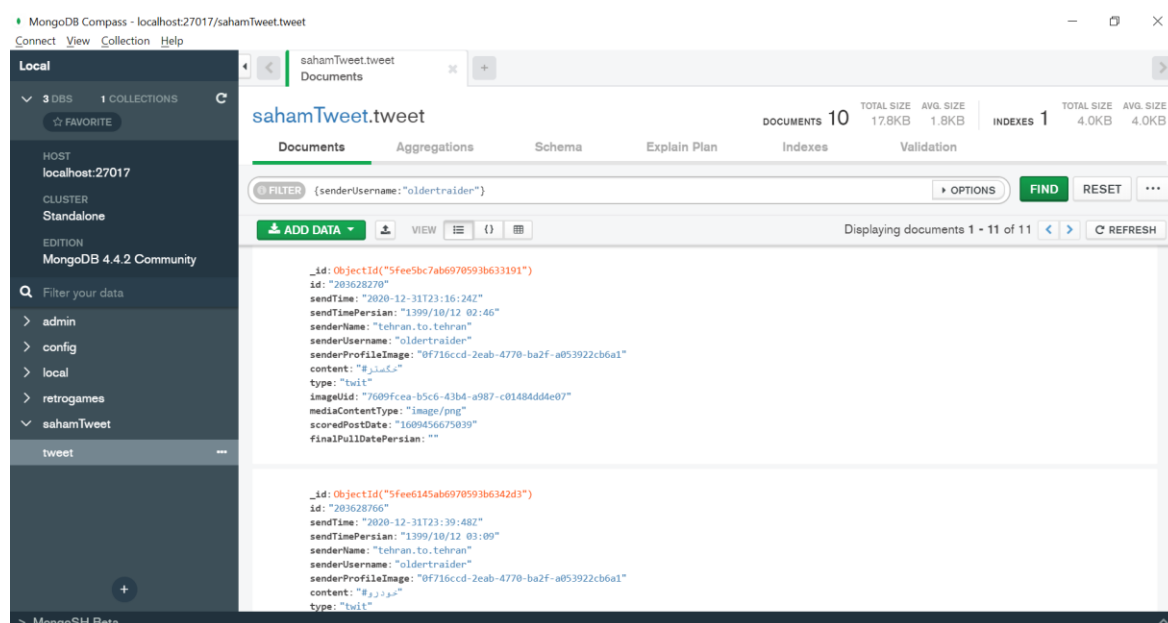
توسط دستور `replace_one` و `upsert` ما هر توییتی که تکراری باشد را دوباره روی خودش می نویسیم و اگر از قبل وجود نداشته باشد آن را اضافه می کنیم.

در مرحله بعد ما یوزری که بیشترین تعداد توییت را انجام داده پیدا می کنیم.

این کار را ابتدا از محیط گرافیکی مونگو دی بی انجام می دهیم.



و در مرحله بعد تویت های کاربر oldertraider را دریافت می کنیم.



دو کوئری بالا را می توان توسط کد پایتون هم انجام داد.

from pymongo import MongoClient

client

MongoClient('mongodb://localhost:27017/?readPreference=primary&appName=MongoDB%20Compass&ssl=false')

result = client['sahamTweet']['tweet'].aggregate([

```

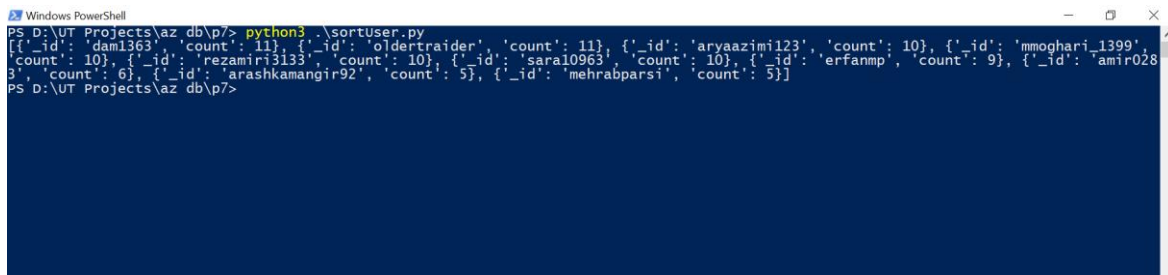
{
  '$group': {
    '_id': '$senderUsername',
    'count': {
      '$sum': 1
    }
  }
}, {
  '$sort': {
    'count': -1
  }
}, {
  '$limit': 10
}
])

print(list(result))

result = client['sahamTweet']['tweet'].find({"senderUsername": "oldertraider"})
print(list(result))

```

و خروجی زیر را دریافت کرد



```

PS D:\UT Projects\az db\p7> python3 .\sortUser.py
[{'_id': 'dami1363', 'count': 11}, {'_id': 'oldertraider', 'count': 11}, {'_id': 'aryaazimi123', 'count': 10}, {'_id': 'mmoghari_1399', 'count': 10}, {'_id': 'rezamir3133', 'count': 10}, {'_id': 'sara10963', 'count': 10}, {'_id': 'erfanmp', 'count': 9}, {'_id': 'amir0283', 'count': 6}, {'_id': 'arashkamangir92', 'count': 5}, {'_id': 'mehrabbarsi', 'count': 5}]
PS D:\UT Projects\az db\p7>

```