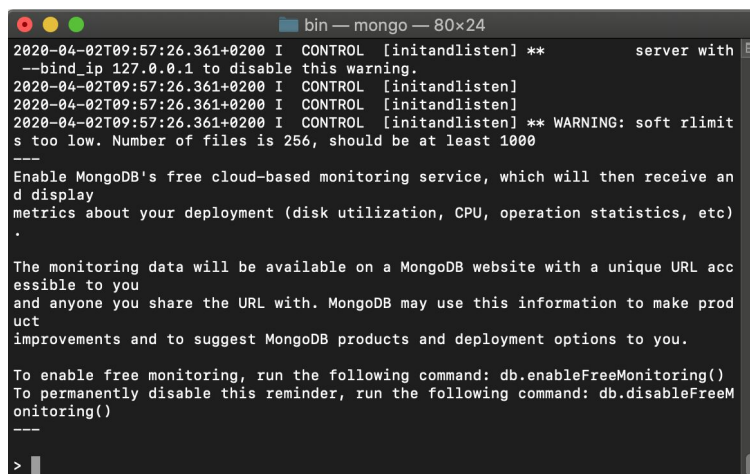


MongoDB

Błażej Kustra

1. Uruchom usługę MongoDB:

a. zweryfikuj poprawnie działający proces przez uruchomienie MongoDB shell w terminalu.



```
bin — mongo — 80x24
2020-04-02T09:57:26.361+0200 I CONTROL [initandlisten] ** server with
--bind_ip 127.0.0.1 to disable this warning.
2020-04-02T09:57:26.361+0200 I CONTROL [initandlisten]
2020-04-02T09:57:26.361+0200 I CONTROL [initandlisten]
2020-04-02T09:57:26.361+0200 I CONTROL [initandlisten] ** WARNING: soft rlimit
s too low. Number of files is 256, should be at least 1000
---
Enable MongoDB's free cloud-based monitoring service, which will then receive an
d display
metrics about your deployment (disk utilization, CPU, operation statistics, etc)
.
The monitoring data will be available on a MongoDB website with a unique URL acc
essible to you
and anyone you share the URL with. MongoDB may use this information to make prod
uct
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeM
onitoring()
---
>
```

2. Zainportuj do MongoDB pliki yelp_academic_data:

a. wykorzystaj komendę mongoimport oraz przyporządkuj kolekcjom opartych na importowanych plikach. json. odpowiednie nazwy.

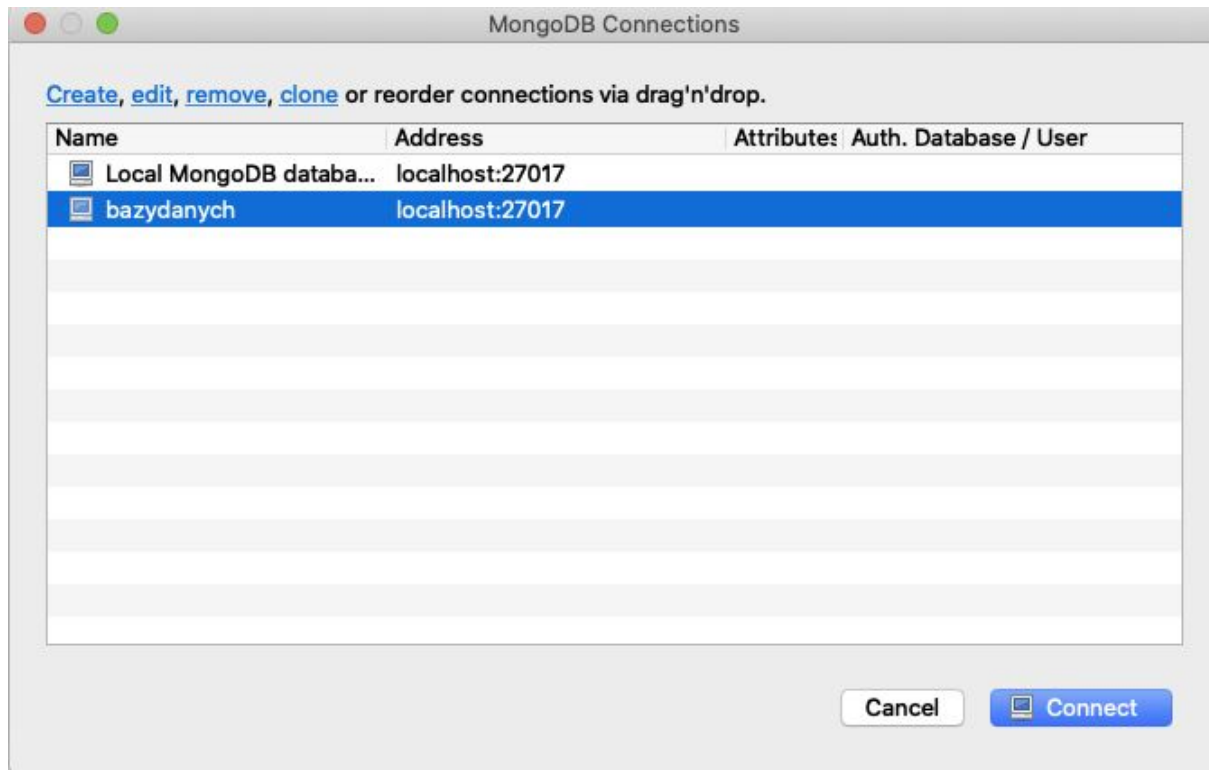
```
/Users/blazejkustra/mongodb/bin/mongoimport --db bazydanych --collection yelp-XXX --type json --file
/Users/blazejkustra/Downloads/yelp_dataset/yelp_academic_dataset_XXX.json
```

w miejsce "XXX" wpisałem odpowiednio:

- business
- checkin
- review
- user
- tip

3. Połącz się z bazą z użyciem narzędzia Robo 3T:

- uruchom narzędzie `/bin/robo3t` umieszczone w folderze `files`
- wykorzystaj `localhost`, port: `27017`.

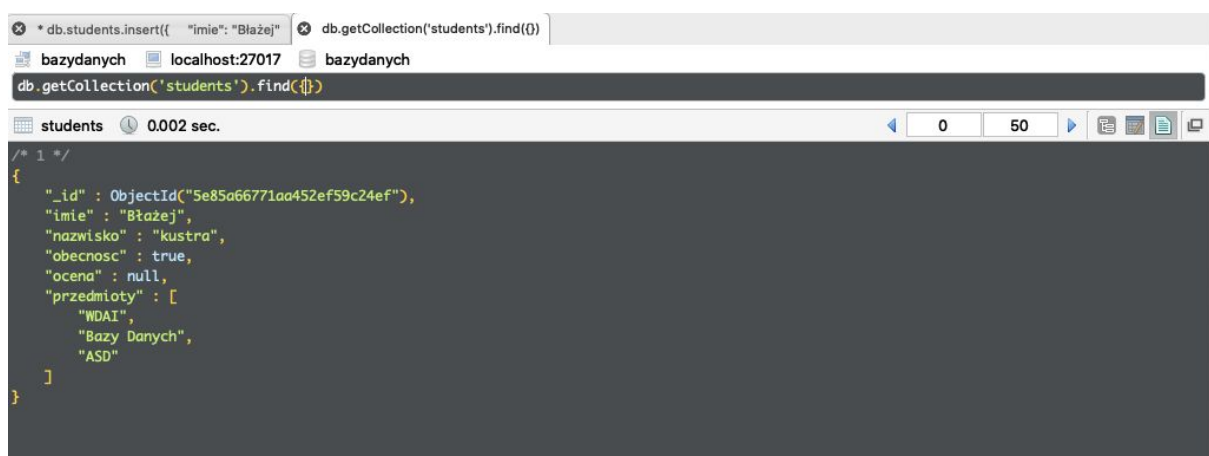


4. Za pomocą narzędzia Robo 3T wykonaj polecenie dodające do stworzonej bazy kolekcję „student”:

a. wprowadź własne dane do kolekcji: imię, nazwisko, obecność (typ bool), ocena z lab. (null), aktualna data, zaliczone przedmioty (min 3 przykładowe).

```
db.students.insert({
  "imie": "Błażej",
  "nazwisko": "kustra",
  "obecność": true,
  "ocena": null,
  "przedmioty": [
    "WDAI",
    "Bazy Danych",
    "ASD"
  ]
})
```

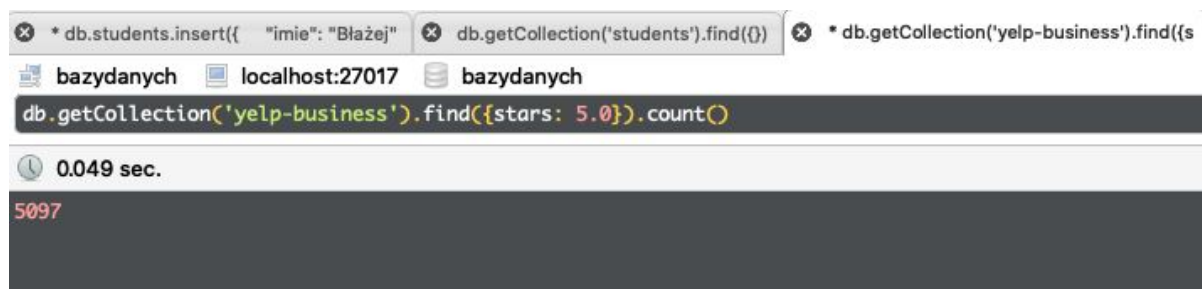
b. wyświetl wynik dodania danej w formie. json txt.



```
/* 1 */
{
  "_id" : ObjectId("5e85a66771aa452ef59c24ef"),
  "imie" : "Błażej",
  "nazwisko" : "kustra",
  "obecność" : true,
  "ocena" : null,
  "przedmioty" : [
    "WDAI",
    "Bazy Danych",
    "ASD"
  ]
}
```

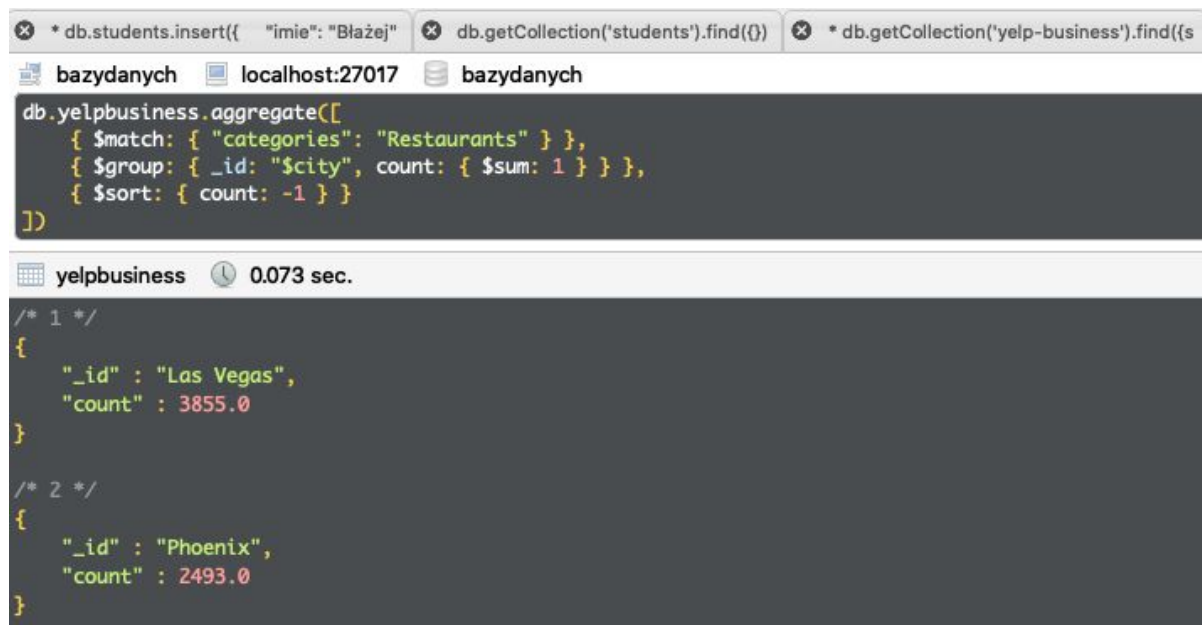
5. Za pomocą narzędzia Robo 3T wykonaj zapytania, które pozwolą uzyskać następujące wyniki:

a. ilość miejsc ocenianych na 5 gwiazdek (pole stars, kolekcja business)



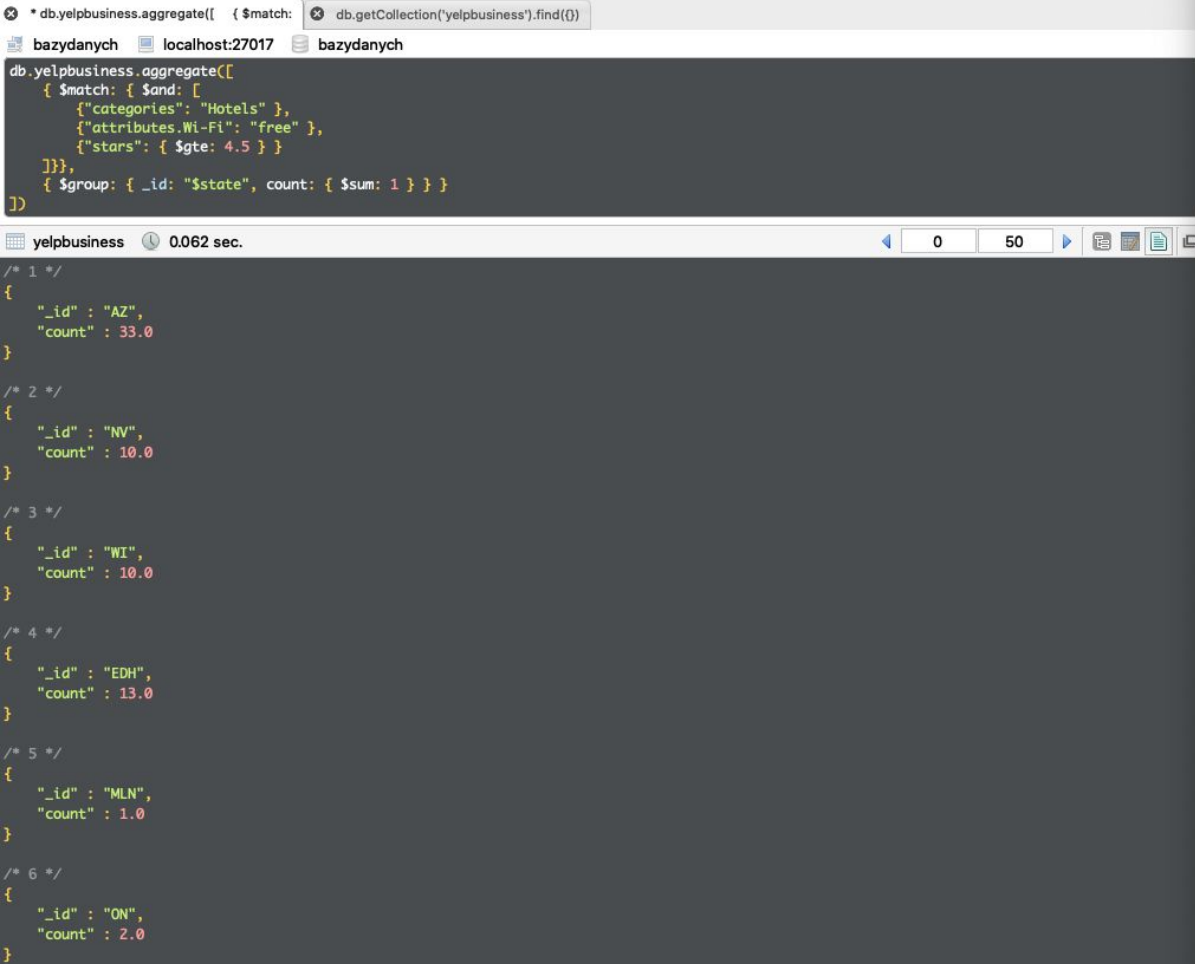
The screenshot shows the Robo 3T interface with three tabs at the top: `* db.students.insert({ "imie": "Błażej" })`, `db.getCollection('students').find({})`, and `* db.getCollection('yelp-business').find({s`. The active tab is `db.getCollection('yelp-business').find({stars: 5.0}).count()`. The interface shows the connection to `bazydanych` at `localhost:27017`. The execution time is `0.049 sec.` and the result is `5097`.

b. ilość restauracji w każdym mieście, wynik posortuj malejąco na podstawie liczby. Pole categories w dokumencie business musi zawierać wartość Restaurants. Wykorzystaj operator group i funkcje aggregate.



The screenshot shows the Robo 3T interface with three tabs at the top: `* db.students.insert({ "imie": "Błażej" })`, `db.getCollection('students').find({})`, and `* db.getCollection('yelp-business').find({s`. The active tab is `db.yelpbusiness.aggregate([{ $match: { "categories": "Restaurants" } }, { $group: { _id: "$city", count: { $sum: 1 } } }, { $sort: { count: -1 } }])`. The interface shows the connection to `bazydanych` at `localhost:27017`. The execution time is `0.073 sec.` and the result is a JSON array with two objects: `{ "_id": "Las Vegas", "count": 3855.0 }` and `{ "_id": "Phoenix", "count": 2493.0 }`.

c. ilość hoteli (atrybut categories powinien mieć wartość Hotels) w każdym stanie/okręgu (state), które posiadają darmowe Wi-fi (pole attributes, klucz-wartość 'Wi-Fi': 'free') oraz ocenę co najmniej 4.5 gwiazdki. Wykorzystaj operator group i funkcje aggregate.



The screenshot shows a MongoDB command prompt window with the following content:

```
* db.yelpbusiness.aggregate([ { $match: { $and: [ { "categories": "Hotels" }, { "attributes.Wi-Fi": "free" }, { "stars": { $gte: 4.5 } } ] }, { $group: { _id: "$state", count: { $sum: 1 } } } ] )
```

The results of the query are displayed below the command:

```
/* 1 */
{
  "_id" : "AZ",
  "count" : 33.0
}

/* 2 */
{
  "_id" : "NV",
  "count" : 10.0
}

/* 3 */
{
  "_id" : "WI",
  "count" : 10.0
}

/* 4 */
{
  "_id" : "EDH",
  "count" : 13.0
}

/* 5 */
{
  "_id" : "MLN",
  "count" : 1.0
}

/* 6 */
{
  "_id" : "ON",
  "count" : 2.0
}
```

The interface also shows the database name 'yelpbusiness' and the execution time '0.062 sec.'.

6. Wykonaj zadania punktu 5 w Pythonie, każde z zadań wykonaj z oddzielnej metodzie.

```
1 import pymongo
2
3 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
4 mydb = myclient["bazydanych"]
5 business = mydb["yelpbusiness"]
6
7 fivestar = business.find({"stars": 5.0}).count()
8
9 print("\nilość miejsc ocenianych na 5 gwiazdek:", fivestar)
10
11 print()#####
12
13 restaurants = business.aggregate([
14     {"$match": {"categories": "Restaurants"}},
15     {"$group": {"_id": "$city", "count": {"$sum": 1}}},
16     {"$sort": {"count": -1}}
17 ])
18
19 print("ilość restauracji w każdym mieście:")
20 count = 0
21 for item in restaurants:
22     print(item)
23     if count > 5:
24         break
25     count += 1
26
27 print()#####
28
29 hotels = business.aggregate([
30     {"$match": {"$and": [
31         {"categories": "Hotels"},
32         {"attributes.Wi-Fi": "free"},
33         {"stars": {"$gte": 4.5}}
34     ]}},
35     {"$group": {"_id": "$state", "count": {"$sum": 1}}}
36 ])
37
38 print("ilość hoteli w każdym stanie, które posiadają darmowe Wi-fi, itd:")
39 for item in hotels:
40     print(item)
```

```
ilość miejsc ocenianych na 5 gwiazdek: 5097
ilość restauracji w każdym mieście:
{'_id': 'Las Vegas', 'count': 3855}
{'_id': 'Phoenix', 'count': 2493}
{'_id': 'Edinburgh', 'count': 1049}
{'_id': 'Scottsdale', 'count': 1023}
{'_id': 'Mesa', 'count': 693}
{'_id': 'Madison', 'count': 679}
{'_id': 'Tempe', 'count': 672}
eg
a
V
e
ilość hoteli w każdym stanie, które posiadają darmowe Wi-fi, itd:
{'_id': 'ON', 'count': 2}
{'_id': 'MLN', 'count': 1}
{'_id': 'WI', 'count': 10}
{'_id': 'EDH', 'count': 13}
{'_id': 'AZ', 'count': 33}
{'_id': 'NV', 'count': 10}
bk-3:untitled blazejkustra$
```

(nie wypisałem wszystkich restauracji, tylko 7 pierwszych dla widoczności)

7. Napisz kod w języku Python, który zwróci użytkownika (nazwa użytkownika) o największej liczbie pozytywnych recenzji (ocena co najmniej 4.5).

```
1 import pymongo
2
3 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
4 mydb = myclient["bazydanych"]
5 users = mydb["yelpuser"]
6 reviews = mydb["yelpreview"]
7
8 user_id_query = reviews.aggregate([
9     {"$match": {"stars": {"$gte": 4.5}}},
10    {"$group": {"_id": "$user_id", "count": {"$sum": 1}}},
11    {"$sort": {"count": -1}},
12    {"$limit": 1}
13 ])
14
15 user_id = list(user_id_query)[0]["_id"]
16
17 users_query = users.find({"user_id": user_id})
18
19 name = list(users_query)[0]["name"]
20
21 print("Użytkownik o największej liczbie pozytywnych recenzji: ", name)
```

```
bk-3:untitled blazejkustra$ python3 gui.py
Użytkownik o największej liczbie pozytywnych recenzji: Rand
bk-3:untitled blazejkustra$
```

8. Napisz kod w języku Python, który zwróci ile recenzji posiadają oceny z każdej kategorii: funny, cool, useful. Przypisanie recenzji do kategorii oznacza, że przynajmniej jedna osoba zagłosowała na recenzje w tej kategorii).

```
1 import pymongo
2
3 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
4 mydb = myclient["bazydanych"]
5 reviews = mydb["yelpreview"]
6
7 funny_count = reviews.count_documents({"votes.funny": {"$gt": 0}})
8 useful_count = reviews.count_documents({"votes.useful": {"$gt": 0}})
9 cool_count = reviews.count_documents({"votes.cool": {"$gt": 0}})
10
11 print("funny:", funny_count)
12 print("useful:", useful_count)
13 print("cool:", cool_count)
```

```
bk-3:untitled blazejkustra$ python3 gui.py
funny: 269256
useful: 549519
cool: 346519
```