

# Ćwiczenie 2 MongoDB

Karol Hamielec

3/10/2020

## zad.1

Wykorzystując bazę danych yelp dataset wykonaj zapytanie i komendy MongoDB, aby uzyskać następujące rezultaty:

- a) Zwróć bez powtórzeń wszystkie nazwy miast w których znajdują się firmy (business). Wynik posortuj na podstawie nazwy miasta alfabetycznie.

New Connection localhost:27017 KarolHamielecCzw1250

```
db.getCollection('business').aggregate([ { $group: { _id: "$city" } }, { $sort: { _id: 1 } } ])
```

business 0.058 sec. 0 50

Key	Value	Type
▶ (15) Cave Creek	{ 1 field }	Object
▶ (16) Centennial Hills	{ 1 field }	Object
▶ (17) Central City Village	{ 1 field }	Object
▶ (18) Central Henderson	{ 1 field }	Object
▶ (19) Chandler	{ 1 field }	Object
▶ (20) Chandler-Gilbert	{ 1 field }	Object
▶ (21) City of Edinburgh	{ 1 field }	Object
▶ (22) Clark County	{ 1 field }	Object
▶ (23) Columbus	{ 1 field }	Object
▶ (24) Coolidge	{ 1 field }	Object
▶ (25) Cottage Grove	{ 1 field }	Object
▶ (26) Cramond	{ 1 field }	Object
▶ (27) Dalkeith	{ 1 field }	Object
▶ (28) Dane	{ 1 field }	Object
▶ (29) De Forest	{ 1 field }	Object
▶ (30) DeForest	{ 1 field }	Object
▶ (31) Deforest	{ 1 field }	Object
▶ (32) Eagan	{ 1 field }	Object
▶ (33) Edinburgh	{ 1 field }	Object
▶ (34) El Mirage	{ 1 field }	Object
▶ (35) Enterprise	{ 1 field }	Object
▶ (36) Fitchburg	{ 1 field }	Object
▶ (37) Florence	{ 1 field }	Object
▶ (38) Fort Kinnaird	{ 1 field }	Object
▶ (39) Fort McDowell	{ 1 field }	Object
▶ (40) Fort McDowell	{ 1 field }	Object
▶ (41) Fountain Hills	{ 1 field }	Object
▶ (42) Fountain Hls	{ 1 field }	Object
▶ (43) Gila Bend	{ 1 field }	Object
▶ (44) Gilbert	{ 1 field }	Object
▶ (45) Glendale	{ 1 field }	Object
▶ (46) Glendale Az	{ 1 field }	Object
▶ (47) Gold Canyon	{ 1 field }	Object
▶ (48) Goldfield	{ 1 field }	Object
▶ (49) Goodyear	{ 1 field }	Object
▶ (50) Green Valley	{ 1 field }	Object

```
db.getCollection('business').aggregate([
  { $group: { _id: "$city" } },
  { $sort: { _id: 1 } }
])
```

- b) Zwróć liczbę wszystkich recenzji, które pojawiły się po 2011 roku (włącznie).

New Connection localhost:27017 KarolHamielecCzw1250

```

1 db.getCollection('review').aggregate([{$addFields:
2   {year: {$year: {$toDate: "$date"}}}},
3   {$match: { year: {$gte: 2011}}},
4   {$group: { _id: null, total: { $sum: 1}}},
5   {$project: { _id: 0}}])
6

```

review 2.19 sec. 0 50

Key	Value	Type
▼ (1)	{ 1 field }	Object
total	880318.0	Double

```

db.getCollection('review').aggregate([
{$addFields: {year: {$year: { $toDate: "$date"}}}},
{$match: {year: {$gte: 2011}}},
{$group: { _id: null, total: {$sum: 1}}},
{$project: { _id: 0}}
])

```

- c) Zwróć dane wszystkich zamkniętych (open) firm (business) z pól: nazwa, adres, gwiazdki (stars).

New Connection localhost:27017 KarolHamielecCzw1250

```

1 getCollection('business').find({open: true},{ name: 1,
2   full_address: 1,
3   stars: 1})

```

business 0.002 sec. 0 50

Key	Value	Type
▶ (1) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (2) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▼ (3) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
_id	ObjectId("5e7c7ad3399090e241...)	ObjectId
full_address	4910 County Rd V De Forest, WI ...	String
name	Culver's	String
stars	4.5	Double
▶ (4) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (5) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▼ (6) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
_id	ObjectId("5e7c7ad3399090e241...)	ObjectId
full_address	4709 Triangle St Mc Farland, WI ...	String
name	Air Quality Systems	String
stars	2.0	Double
▶ (7) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▼ (8) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
_id	ObjectId("5e7c7ad3399090e241...)	ObjectId
full_address	4412 Siggelkow Rd Mc Farland, ...	String
name	Green Lantern Restaurant	String
stars	3.5	Double
▶ (9) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (10) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (11) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (12) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (13) ObjectId("5e7c7ad339...)	{ 4 fields }	Object
▶ (14) ObjectId("5e7c7ad339...)	{ 4 fields }	Object

```

db.getCollection('business').find({
  {open : true},
  {name: 1, full_address: 1, stars: 1}})

```

- d) Zwróć dane wszystkich użytkowników (user), którzy nie uzyskali ani jednego pozytywnego głosu z kategorii (funny lub useful), wynik posortuj alfabetycznie według imienia użytkownika.

New Connection localhost:27017 KarolHamielecCzw1250

```
1 letCollection('user').find({ $or:
2 [ {"votes.funny": 0}, {"votes.useful": 0}]}).sort({ name: 1})
```

user 2.67 sec. 0 50

Key	Value	Type
▶ (1) ObjectId("5e7c7b974fb...)	{ 12 fields }	Object
▼ (2) ObjectId("5e7c7b9a4fb...)	{ 12 fields }	Object
_id	ObjectId("5e7c7b9a4fb379eff31c...)	ObjectId
yelping_since	2011-12	String
▼ votes	{ 3 fields }	Object
funny	0	Int32
useful	2	Int32
cool	2	Int32
review_count	10	Int32
name	'Anastacia	String
user_id	qjLc0rYytqzeVBiTPftfSA	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	4.5	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array
▶ (3) ObjectId("5e7c7b974fb...)	{ 12 fields }	Object
▶ (4) ObjectId("5e7c7b9c4fb...)	{ 12 fields }	Object
▼ (5) ObjectId("5e7c7b9e4fb...)	{ 12 fields }	Object
_id	ObjectId("5e7c7b9e4fb379eff31e...)	ObjectId
yelping_since	2011-04	String
▼ votes	{ 3 fields }	Object
funny	0	Int32
useful	5	Int32
cool	3	Int32
review_count	6	Int32
name	'Jenelle	String
user_id	lI5yKFJdqKSTft1bxsD1fQ	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	5.0	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array
▶ (6) ObjectId("5e7c7b994fb...)	{ 12 fields }	Object

```
db.getCollection('user').find({ $or:
  [ {"votes.funny": 0}, {"votes.useful": 0}]}).sort({ name: 1})
```

- e) Określ, ile każde przedsiębiorstwo otrzymało wskazówek/napiwków (tip) w 2012. Wynik posortuj alfabetycznie według liczby (tip).

Welc... x \* db.getC... x \* db.getC... x db.getCol... x db.getCol... x \* db.getCo... x

New Connection localhost:27017 KarolHamielecCzw1250

```
1 db.collection('tip').aggregate([
2   { $match: { date: { $regex: "^2012.*" } } },
3   { $group: { _id: "$business_id", count: { $sum: 1 } } },
4   { $sort: { count: 1 } }
5 ])
```

tip 0.307 sec. 0 50

Key	Value	Type
(1) 0MOK0dElhzTAqeaola...	{ 2 fields }	Object
(2) n_wZriakBT7sVn87dpc...	{ 2 fields }	Object
_id	n_wZriakBT7sVn87dpcEA	String
count	1.0	Double
(3) OrXzZB0M0DbPXx3F_X...	{ 2 fields }	Object
(4) DVppg4NQceZw7b4KM...	{ 2 fields }	Object
(5) alzLOo6dk92DDts8Ypy...	{ 2 fields }	Object
(6) 4Io8yGKduNHZofQCzNi...	{ 2 fields }	Object
(7) g_kOqG2Zl8ygx46Xkt...	{ 2 fields }	Object
(8) NNjr2Wa08L1xwCnWL...	{ 2 fields }	Object
(9) N-18zbUr2ueTVyxuElb...	{ 2 fields }	Object
(10) t9EWq6HxmGH77YvC...	{ 2 fields }	Object
(11) OLV_KmtKklcx3QJIXk...	{ 2 fields }	Object
(12) GVerH-7DIZTDUSci3...	{ 2 fields }	Object
(13) HO9rGljsiXkO05JLncz...	{ 2 fields }	Object
(14) fTVI61ab7m4nlDafhL...	{ 2 fields }	Object
(15) fOxHbgmFwzIAC6cJaF...	{ 2 fields }	Object
(16) 2x7IRhCWPLJOTcT2e...	{ 2 fields }	Object
(17) jdi5ZRMINSHW0f3RYh...	{ 2 fields }	Object
(18) ZQm2rGL-dFfr15vfwY...	{ 2 fields }	Object
(19) SBmbCfkqbG5KkFE8s...	{ 2 fields }	Object
(20) yq4NDtkYiNv_Qik0T...	{ 2 fields }	Object
(21) 76t_NE9GQthoryLZgD...	{ 2 fields }	Object
(22) FSKnfbSeXYyF-uHv1t...	{ 2 fields }	Object
(23) 1CNjeljdT4Ap7B-8ddb...	{ 2 fields }	Object
(24) Exvv1ltxRhnoB1fKCh...	{ 2 fields }	Object
(25) 0CFBeDV58e3uH4vsK...	{ 2 fields }	Object
(26) RGFKaBRV_HO-ZJb_e...	{ 2 fields }	Object
(27) 59kTtQDwjH-z01r1Kix...	{ 2 fields }	Object
(28) mlendFH3mc2E4dW0...	{ 2 fields }	Object
(29) y8XDNU5xCUwRcBEF...	{ 2 fields }	Object
(30) n-4bt8aB282PEG4KM...	{ 2 fields }	Object
(31) B6sT5YGBYB9KfhkiZt...	{ 2 fields }	Object

```
db.getCollection('tip').aggregate([
  { $match: { date: { $regex: "^2012.*" } } },
  { $group: { _id: "$business_id", count: { $sum: 1 } } },
  { $sort: { count: 1 } }
])
```

- f) Wyznacz, jaką średnia ocen (stars) uzyskała każda firma (business) na podstawie wszystkich recenzji. Wynik ogranicz do recenzji, które uzyskały min 4.0 gwiazdki.

Welc... x \* db.g... x \* db.g... x db.get... x db.get... x \* db.g... x \* db.ge... x

New Connection localhost:27017 KarolHamielecCzw1250

```
1 db.getCollection('review').aggregate([
2   {$group: { _id: "$business_id", avg: {$avg: "$stars" }}},
3   {$match: { avg: { $gt: 4.0}}}
4 ])
```

review 1.32 sec. 0 50

Key	Value	Type
▶ (1) sgbqQH4DXohaApOO5...	{ 2 fields }	Object
▶ (2) v9htJ8R0Ctvykmqjz7e...	{ 2 fields }	Object
▶ (3) 9MT5gEtM2S4zRkPbK...	{ 2 fields }	Object
▶ (4) qgATg-mdvMYqRZR1p...	{ 2 fields }	Object
▶ (5) Urt6I634ngW4YB9H5S...	{ 2 fields }	Object
▼ (6) qQNjHyoFL3_0xugLf8s...	{ 2 fields }	Object
_id	qQNjHyoFL3_0xugLf8sM2A	String
avg	4.32258064516129	Double
▶ (7) _EmzBbDf2QdFDixuXA...	{ 2 fields }	Object
▶ (8) stHnGL53WVo34kWdo...	{ 2 fields }	Object
▶ (9) m0qh4NO0fD16eVS2n...	{ 2 fields }	Object
▶ (10) nmiZKlvldDiPatQwc_P...	{ 2 fields }	Object
▶ (11) 3Tc5brEpaVnl7R4bm...	{ 2 fields }	Object
▶ (12) jM-vUiQqkHDanbUezY...	{ 2 fields }	Object
▶ (13) 4kNuUwXyfeH-E78Ts...	{ 2 fields }	Object
▶ (14) CDY7AE5HWBy0TJXC...	{ 2 fields }	Object
▶ (15) mvAAgNQaypArsoGX...	{ 2 fields }	Object
▶ (16) 2ZIsC_iepL_VbP2JjyfYzQ	{ 2 fields }	Object
▶ (17) 8Gh4a0-MCLPRKbln5...	{ 2 fields }	Object
▶ (18) ODxHdTy2VypDL103j...	{ 2 fields }	Object
▶ (19) hGwnLUZ10za8Mt1ID...	{ 2 fields }	Object
▶ (20) A3BxYKWLW0olBtvaR...	{ 2 fields }	Object
▶ (21) OoXySPyQuSHstMF2z...	{ 2 fields }	Object
▶ (22) nq8hjP6NfykQmC6lCt...	{ 2 fields }	Object
▶ (23) jsVlv9NRRBXdmCe8b...	{ 2 fields }	Object
▶ (24) OFE_1lMnq3uvtCbxMr...	{ 2 fields }	Object
▼ (25) uj8Dalyig3oJLQyslyg9...	{ 2 fields }	Object
_id	uj8Dalyig3oJLQyslyg99w	String
avg	4.333333333333333	Double
▶ (26) xU8OyNDCFGaELqRf...	{ 2 fields }	Object
▶ (27) Ls2ZYnTrESmH8t9Tp...	{ 2 fields }	Object
▶ (28) eEQ065Hf3fGaiha0T	{ 2 fields }	Object

```
db.getCollection('review').aggregate([
  {$group: { _id: "$business_id", avg: {$avg: "$stars" }}},
  {$match: { avg: { $gt: 4.0}}}
])
```

- g) Usuń wszystkie firmy (business), które posiadają ocenę (stars) równą 2.0.



```
db.getCollection('business').remove({stars: { $eq: 2.0}})
```

## zad.2

Zdefiniuj funkcję (MongoDB) umożliwiającą dodanie nowej recenzji (review). Wykonaj przykładowe wywołanie.

```
function addReview(votes, user_id, review_id, stars, date, text, type, business_id) {  
  // write your code here  
  db.review.insert({  
    votes: votes,  
    user_id: user_id,  
    review_id: review_id,  
    stars: stars,  
    date: date,  
    text: text,  
    type: type,  
    business_id: business_id  
  });  
}  
addReview({ funny: 2}, 3, 3, 3, "2020-04-14", "testsetest", "xd", 211213)
```



```

* function addReview(v... x db.system.js.find() x * db.getCollection('revie... x
New Connection localhost:27017 KarolHamielecCzw1250
1 function addReview(votes, user_id, review_id, stars, date, text, type, bu
2 | // write your code here
3 db.review.insert({
4 votes: votes,
5 user_id: user_id,
6 review_id: review_id,
7 stars: stars,
8 date: date,
9 text: text,
10 type: type,
11 business_id: business_id
12 });
13 }
14 addReview({ funny: 2}, 3, 3, 3, "2020-04-14", "testsetest", "xd", 211213)
0.169 sec.
1 Inserted 1 record(s) in 168ms
2

```

```

* function addReview(v... x db.system.js.find() x * db.getCollection('revie... x
New Connection localhost:27017 KarolHamielecCzw1250
1 db.getCollection('review').find({type: "xd"})
review 1.57 sec. 0 50

```

Key	Value	Type
▼ (1) ObjectId("5e95e618e37d77... { 9 fields }		Object
_id	ObjectId("5e95e618e37d77b057290b...)	ObjectId
votes	{ 1 field }	Object
user_id	3.0	Double
review_id	3.0	Double
stars	3.0	Double
date	2020-04-14	String
text	testsetest	String
type	xd	String
business_id	211213.0	Double

## zad.3

Zdefiniuj funkcję (MongoDB), która zwróci wszystkie biznesy (business), w których w kategorii znajduje się podana przez użytkownika cecha. Wartość kategorii należy

```
function getBusinessWithCategory(cat) {
  // write your code here
  return db.business.find({ categories: cat});
}
getBusinessWithCategory("Food");
```

function getBusiness... x db.business.find({c... x db.getCollection('re... x db.getCollection('busi... x

New Connection localhost:27017 KarolHamielecCzw1250

```
1 function getBusinessWithCategory(cat) {
2   // write your code here
3   return db.business.find({ categories: cat});
4 }
5 getBusinessWithCategory("Food");
```

business 0.001 sec. 0 50

Key	Value	Type
(1) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(2) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
_id	ObjectId("5e7c7ad3399090e24164c...)	ObjectId
business_id	1-EldeHewfEVcpoZm38N8A	String
full_address	1901 Cayuga St Middleton, WI 53562	String
hours	{ 7 fields }	Object
open	true	Boolean
categories	[ 3 elements ]	Array
[0]	Food	String
[1]	Beer, Wine & Spirits	String
[2]	Coffee & Tea	String
city	Middleton	String
review_count	22	Int32
name	Barriques Wine & Spirits	String
neighborhoods	[ 0 elements ]	Array
longitude	-89.513737	Double
state	WI	String
stars	4.0	Double
latitude	43.096359	Double
attributes	{ 6 fields }	Object
type	business	String
(3) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(4) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(5) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(6) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(7) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(8) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(9) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(10) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(11) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(12) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(13) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object
(14) ObjectId("5e7c7ad339909...)	{ 16 fields }	Object

## zad.4

Zdefiniuj funkcję (MongoDB), która umożliwi modyfikację nazwy użytkownika (user) na podstawie podanego id. Id oraz nazwa mają być przekazywane jako parametry.

```

* function modi... x * db.business.f... x * db.getCollect... x db.getCollectio... x db.getCollectio... x
New Connection localhost:27017 KarolHamielecCzw1250
1 function modifyName(id, name) {
2   // write your code here
3   db.user.update({_id: id},{ $set: { name:name}});
4 }
5 modifyName(ObjectId("5e7c7b964fb379eff31af9b3"), "Karol");

0.133 sec.
1 Updated 1 existing record(s) in 133ms
2

```

New Connection localhost:27017 KarolHamielecCzw1250

```
1 db.getCollection('user').find({})
```

user 0.004 sec.

Key	Value	Type
▶ (1) ObjectId("5e7c7b964fb379... { 12 fields }		Object
▼ (2) ObjectId("5e7c7b964fb379... { 12 fields }		Object
_id	ObjectId("5e7c7b964fb379eff31af9b...)	ObjectId
yelping_since	2010-09	String
votes	{ 3 fields }	Object
review_count	22	Int32
name	Karol	String
user_id	0vscrHoajVRa1Yk19XWdwA	String
friends	[ 4 elements ]	Array
fans	1	Int32
average_stars	5.0	Double
type	user	String
compliments	{ 5 fields }	Object
elite	[ 0 elements ]	Array
▶ (3) ObjectId("5e7c7b964fb379... { 12 fields }		Object
▶ (4) ObjectId("5e7c7b964fb379... { 12 fields }		Object

```

function modifyName(id, name) {
  // write your code here
  db.user.update({_id: id},{ $set: { name:name}});
}
modifyName(ObjectId("5e7c7b964fb379eff31af9b3"), "Karol");

```

## zad.5

Zwróć średnią ilość wszystkich wskazówek/napiwków dla każdego z biznesów, wykorzystaj map reduce.

```

var mapF = function(){
  emit (this.business_id, this.likes)
};

var reduceF = function( business_id, valuesLikes){
  return Array.avg(valuesLikes);
};

db.tip.mapReduce(mapF, reduceF, { out: "avg_likes_per_business"})

```

db.g... x db.g... x \* d... x db.g... x db.g... x \* d... x \* d... x \* va... x

New Connection localhost:27017 KarolHamielecCzw1250

```

1 var mapF = function(){
2   emit (this.business_id, this.likes)
3 };
4
5 var reduceF = function( business_id, valuesLikes){
6   return Array.avg(valuesLikes);
7 };
8
9 db.tip.mapReduce(mapF, reduceF, { out: "avg_likes_per_business"})

```

4.09 sec.

Key	Value	Type
(1)	{ 8 fields }	Object
result	avg_likes_per_business	String
timeMillis	4073.0	Double
counts	{ 4 fields }	Object
ok	1.0	Double
_o	{ 4 fields }	Object
_keys	[ 4 elements ]	Array
_db	{ 3 fields }	Object
_coll	{ 4 fields }	Object

db.g... x db.g... x \* d... x db.g... x db.g... x \* d... x \* d... x

New Connection localhost:27017 KarolHamielecCzw1250

```
1 db.avg_likes_per_business.find({ value: { $gt: 0.0}})
```

avg\_likes\_per\_business 0.004 sec. 0 50

Key	Value	Type
(1) --5jkZ3-nUPZxUvtcbr8...	{ 2 fields }	Object
_id	--5jkZ3-nUPZxUvtcbr8Uw	String
value	0.0625	Double
(2) -0Ackw6MF82PX09f9J...	{ 2 fields }	Object
(3) -1ERbsOf9XOC9wGbZ...	{ 2 fields }	Object
(4) -1tQu_yKCOgOj-WtXm...	{ 2 fields }	Object
_id	-1tQu_yKCOgOj-WtXmMYIA	String
value	0.0123456790123457	Double
(5) -2gyHPSL8pjjkSe5Ozz...	{ 2 fields }	Object
(6) -2n-OHbbwlv3SY8RP9...	{ 2 fields }	Object
(7) -584fn2GxYe9sLsgN2...	{ 2 fields }	Object
_id	-584fn2GxYe9sLsgN2WeQA	String
value	0.032258064516129	Double
(8) -5XAZC1hC6G0NDkm...	{ 2 fields }	Object
(9) -63VfA2tnYyzwRt81B1...	{ 2 fields }	Object
(10) -73hx2hut_2KPcGpC...	{ 2 fields }	Object
(11) -7Gvh1JVEZnEZ1DkE...	{ 2 fields }	Object
(12) -7smFzvh2E2fO0Ys-...	{ 2 fields }	Object

## zad.6

Odwzoruj wszystkie zadania z punktu 1 w języku programowania (np. JAVA) z pomocą API do MongoDB. Wykorzystaj dla każdego zadania odrębną metodę

```

import pymongo
from bson.son import SON
from bson.code import Code

def zada( client):
    return client['business'].aggregate([ { "$group": { "_id": "$city"}}, { "$sort":
{ "_id": 1}} ]])

def zadb(client):
    return client['review'].aggregate([
        {'$addFields': { 'year': { '$year': {'$toDate': "$date"}}}},
        {'$match': {'year': {'$gte': 2011 } } },
        {'$group': { '_id': 'null', 'total': { '$sum': 1}}},
        {'$project': {'_id': 0}}
    ])

def zadc(client):
    return client['business'].find({'open': False}, { 'name': 1, 'full_address': 1, '
stars': 1})

def zadd(client):
    return client['user'].find({ '$or': [{'votes.funny': 0}, {'votes.useful': 0}] }).
sort([( 'name', 1)])

def zade(client):
    return client['tip'].aggregate([
        { '$match': { 'date': { '$regex': "^2012.*"} }},
        { '$group': { "_id": "$business_id", "count": { '$sum': 1}}},
        { '$sort': { "count": 1}}
    ])

def zadf(client):
    return client['review'].aggregate([
        { '$group': { '_id': "$business_id", 'avg': { '$avg': "$stars" } }},
        { '$match': { 'avg': { '$gt': 4.0}}}]])

def zadg(client):
    return client['business'].remove({ 'stars': { '$eq': 2.0 } })

if __name__ == '__main__':
    host = 'localhost'
    port = 27017
    db = 'KarolHamielecCzw1250'
    cl = pymongo.MongoClient( host, port)

    l2 = list(zadg(cl[db]))
    print(l2)

```

## zad.7

Kolekcja kienci:

```
{
  name: "",
  email: "",
  phone_number: 123456789
  address: {
    city: ""
    street: ""
    postal_code: ""
  },
  created: "2020-02-02"
}
```

Kolekcja przedmioty:

```
{
  name: "",
  price: 10.0
  avaiability: true
}
```

Kolekcja zakupy:

```
{
  user_id: "",
  cart: [ { item_id: "", qty: 3 },
          { item_id: "", qty: 23 },
          { item_id: "", qty: 16 } ],
  date: "2020-02-02",
  payment_made: "2020-02-03"
}
```

normal