





























































## Bazy danych –MongoDB – zadanie

Imię i nazwisko: Barbara Wojtarowicz

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

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

```
db.getCollection("yelp_academic_dataset_business").find({open: false}, {name: 1, full_address: 1, stars: 1})
```

yelp_academic_dataset_business			
_id	full_address	name	stars
 608b32d1f13a50...	 3935 E Thomas R...	 Clicks Billiards	 3.0
 608b32d1f13a50...	 4041 E Thomas R...	 Olympic Flame	 4.0
 608b32d1f13a50...	 3933 E Indian Sc...	 Coffee Lady	 4.0
 608b32d1f13a50...	 3215 E Thomas R...	 Pat Murphy's	 4.0
 608b32d1f13a50...	 4502 E Thomas R...	 Elie's Deli	 4.5
 608b32d1f13a50...	 4046 E Thomas R...	 Fazoli's	 2.5
 608b32d1f13a50...	 3939 E Campbell ...	 Petite Chateau	 3.5
 608b32d1f13a50...	 4410 N 40th St\n...	 Tammie Coe Cak...	 4.0
 608b32d1f13a50...	 3619 E Indian Sc...	 Mama Java's Coff...	 4.0
 608b32d1f13a50...	 4402 E Indian Sc...	 Arcadia Car Wash	 4.0
 608b32d1f13a50...	 3222 E Indian Sc...	 Chuy's Mesquite ...	 3.5
 608b32d1f13a50...	 4225 E Camelbac...	 Havana Cafe	 3.5
 608b32d1f13a50...	 5053 N 44th St\n...	 Taylor's Cafe	 3.5
 608b32d1f13a50...	 4228 E Indian Sc...	 The Great Escape ...	 2.5
 608b32d1f13a50...	 4221 E Indian Sc...	 La Fontaine's Bar	 2.5

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

```
db.getCollection("yelp_academic_dataset_user").find({
  $or:
  [
```

```

    {"votes.funny": 0},
    {"votes.useful": 0}
  ]

```

```

}).sort({"name": 1})

```

yelp_academic_dataset_user							
_id	yelping_since	votes	review_count	name	user_id	friends	fans
608b32d4f13a50...	2009-08	{ 3 fields }	1	Bernard	xP3SPgfgW2vc5Z...	[ 0 elements ]	0
608b32d7f13a50...	2011-12	{ 3 fields }	10	'Anastacia	qJLc0rYytqzeVBiT...	[ 0 elements ]	0
608b32d3f13a50...	2012-03	{ 3 fields }	2	'Brandon	CfWeBCqVvHze8...	[ 1 elements ]	0
608b32daf13a50...	2011-10	{ 3 fields }	5	'David	M5IQGr6omqqm...	[ 33 elements ]	0
608b32dcf13a506...	2011-04	{ 3 fields }	6	'Jenelle	lI5yKFJdqKSTf1b...	[ 0 elements ]	0
608b32d6f13a50...	2013-11	{ 3 fields }	4	'Persian	bSoPY5UR-fXSk...	[ 0 elements ]	0
608b32d6f13a50...	2013-03	{ 3 fields }	4	'Virydiana	wP-nekpfzE2prz...	[ 5 elements ]	1
608b32dcf13a506...	2013-03	{ 3 fields }	1	,Maria	Os5f3TNpM7_A8...	[ 0 elements ]	0
608b32d4f13a50...	2009-06	{ 3 fields }	10	.rick	28s2JRyWuRG6...	[ 0 elements ]	0
608b32def13a50...	2014-06	{ 3 fields }	3	006969123	FtTid0SF_puLOQ...	[ 0 elements ]	0
608b32d1f13a50...	2012-10	{ 3 fields }	2	A	MjLcGNLsSUz2eN...	[ 0 elements ]	0
608b32d1f13a50...	2013-01	{ 3 fields }	2	A	aHuT-eiSBLORLJT...	[ 0 elements ]	0
608b32d1f13a50...	2013-01	{ 3 fields }	1	A	kanjlBDaadQ_E6...	[ 0 elements ]	0
608b32d1f13a50...	2012-01	{ 3 fields }	8	A	UuV8IZ90PdLma...	[ 0 elements ]	0

c) Określ, ile każde przedsiębiorstwo otrzymało wskazówek/napiwków (*tip*) w 2012. Wynik posortuj alfabetycznie według nazwy.

```

db.getCollection("yelp_academic_dataset_tip").aggregate([
  {$match: {"date": /2012/ }},
  {$group: {_id: "$business_id", count: {$sum: 1}}},
  {$sort: {"count": 1}}
])

```

Raw Shell Output		Aggregate Query (line 1) ×	
← →		50	Documents 1 to 50
1	{		
2	"_id" : "PDrn1ShvACVge3X-Q24I1g",		
3	"count" : 1.0		
4	}		
5	{		
6	"_id" : "PB1CpTP1r6B6xSjHZctbHA",		
7	"count" : 1.0		
8	}		
9	{		
10	"_id" : "ctbzSbfDymOVGObyDJK1qw",		
11	"count" : 1.0		
12	}		
13	{		
14	"_id" : "pz-jypT4fEPQKr9RZtVdAw",		
15	"count" : 1.0		
16	}		
17	{		
18	"_id" : "mAsBgwSv1VprhnUVasGuRg",		
19	"count" : 1.0		
20	}		
21	{		
22	"_id" : "EPez0rU3_CtHfRcvGrgT9Q",		

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

```
db.getCollection("yelp_academic_dataset_review").aggregate([
  {$group: {_id: "$business_id", avgStars: {$avg: "$stars"}}},
  {$match: {"avgStars": {$gte: 4}}}
])
```

Output > avgStars	
_id	avgStars
"-7kSfrNsb-RUUD0...	4.666666666666...
"D34hbrWuKMp7j...	4.888888888888...
"r-xrEWxlwUgrApxt...	5.0
"Y0KbOC-2R6NYX...	4.5
"qnJgp2Ezm2ipM6...	4.25
"ics35glmd7JXtY3...	4.052631578947...
"XdNuJo7kuspVp6...	4.25
"-_Ke8q969OAweE...	4.333333333333...
"PzZvKWJfoY9YvIX...	5.0
"b-Sc_mRnCmaEzS...	4.166666666666...
"WAjGxck4e-ELBvq...	4.0
"DeUheTWpllaU3O...	4.785714285714...
"vkMX3xZJlcc70th...	5.0
"rXiYH8FU2Z0tGdp...	4.333333333333...
"eYwTptN9P3D_qa...	5.0
"zslW3YVSMnhluh	4.25

e) Usuń wszystkie firmy (*business*), które posiadają ocenę (*stars*) równą 2.0.

```
db.getCollection("yelp_academic_dataset_business").deleteMany({"stars": 2.0})
```

```
31 db.getCollection("yelp_academic_dataset_business").deleteMany({"stars": 2.0})
32
```

Result > acknowledged		
_id	acknowledged	deletedCount
	true	1576.0

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

```
function insertReview(user_id, text, business_id, review_id){
  db.yelp_academic_dataset_review.insert({
    votes: {
      funny: 0,
      useful: 0,
      cool: 0
    },
    user_id: user_id,
    review_id: review_id,
    stars: 0,
    date: new Date(),
    text: text,
    type: "review",
    business_id: business_id
  })
}
```

```
insertReview("MWhR9LvOdRbqtu1I_DRFBg", "Sample review",
"gxuVySgACHDqJlwmelFHLA", "gxuVySgACHDqJlwmelFHLA")

db.getCollection("yelp_academic_dataset_review").find({text:
"Sample review"}, {})
```

yelp_academic_dataset_review							
_id	votes	user_id	review_id	stars	date	text	type
609c35dd8c9e1f7...	{ 3 fields }	MWhR9LvOdRbq...	gxuVySgACHDqJl...	0.0	2021-05-12T20:0...	Sample review	review

3. Zdefiniuj funkcję, która zwróci wszystkie biznesy (*business*), w których w kategorii znajduje się podana przez użytkownika cecha. Wartość kategorii należy przekazać do funkcji jako parametr. Wykonaj przykładowe wywołanie zdefiniowanej funkcji.

```
function getBusinessWithCategory(category){
    return
    db.getCollection("yelp_academic_dataset_business").find({"categories": category})
}
```

```
getBusinessWithCategory("Restaurants")
```

Result						
_id	business_id	full_address	hours	open	categories	city
608b32d1f13a50605f204f7f	gxuVySgACHDqj...	5619 E Indian Sc...	{ 0 fields }	true	[ 4 elements ]	Phoenix
608b32d1f13a50605f204fd5	cvb_rYMk48Mh...	4041 E Thomas R...	{ 0 fields }	false	[ 3 elements ]	Phoenix
608b32d1f13a50605f204fdd	VpSiOW1PapVm...	3930 E Camelbac...	{ 0 fields }	true	[ 2 elements ]	Phoenix
608b32d1f13a50605f204fee	Upru5VE0jZ81ZQ...	3330 E Thomas R...	{ 0 fields }	true	[ 3 elements ]	Phoenix
608b32d1f13a50605f204ff5	oc0Wl6ktfMLb5V...	3930 E Camelbac...	{ 7 fields }	true	[ 4 elements ]	Phoenix
608b32d1f13a50605f205032	_ODHYjCmrb3W...	3239 E Indian Sc...	{ 0 fields }	true	[ 3 elements ]	Phoenix
608b32d1f13a50605f205056	CvjNcUtmNubDF...	3945 E Thomas R...	{ 7 fields }	true	[ 3 elements ]	Phoenix
608b32d1f13a50605f205074	MLLWQyd...	3923 E Thomas R...	{ 7 fields }	true	[ 2 elements ]	Phoenix
608b32d1f13a50605f205074	created: 2021-04-29T22:27:29.000Z machine: 15809104 pid: 24671 counter: 2117748	hILdh8S8L...	{ 0 fields }	true	[ 2 elements ]	Phoenix
608b32d1f13a50605f205074	2PFYBzn5...	13610 N Scottsd...	{ 6 fields }	true	[ 2 elements ]	Scottsdale
608b32d1f13a50605f205074	91vaYDV...	4539 E Thomas R...	{ 0 fields }	true	[ 2 elements ]	Phoenix

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

```
function modifyUserName(user_id, new_name){
    db.yelp_academic_dataset_user.update(
        {user_id: user_id},
        {$set: {name: new_name}}
    )
}
```

```
modifyUserName("MWhR9LvOdRbqtu1I_DRFBg", "New name")
db.yelp_academic_dataset_user.find({name: "New name"})
```

yelp_academic_dataset_user							
_id	yelping_since	votes	review_count	name	user_id	friends	fans
608b32d0f13a50...	2011-12	{ 3 fields }	1	New name	MWhR9LvOdRbq...	[ 13 elements ]	0

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

W pierwszym kroku zwrócenie średniej liczby wskazówek/napiwków dla każdego z biznesów:

```

var mapFun = function(){
    var key = this.business_id;
    var value = 1;
    emit(key, value);
};

var reduceFun = function(key, values){
    return Array.sum(values);
};

db.getCollection("yelp_academic_dataset_review").mapReduce(
    mapFun,
    reduceFun,
    {
        out: "count_tips"
    }
);

```

count_tips > value	
_id	value
"_id" qpsej-hlcCcRR8nL...	7.0
"_id" vSVWon_i19O3j-K...	11.0
"_id" ZHFQZ3M8H_xE0I...	17.0
"_id" 12ivXGvkBqVTpWl...	16.0
"_id" 0AsJNgbhmkY5Z0...	62.0
"_id" m-vf7COWx0qHC...	20.0
"_id" rlxuJ8LizdvrgUBB...	5.0
"_id" tqIFZtb873aJM_4...	6.0
"_id" O_ltzZej0aMVy4D...	3.0
"_id" B5y4yh38nK26cUi...	5.0
"_id" 33C2tEB2IUiKFJ_o...	3.0
"_id" ai4e0PDmLLrlieV1...	6.0

W drugim kroku zwrócenie średniej z wszystkich napiwków:

```

var avgTips = function(tips){
    sum = 0;
    for (i = 0; i < tips.length; i++){
        sum += tips[i].value;
    }
    return sum / tips.length;
};

```

```

    }
    return sum/tips.length;
}

avgTips(db.count_tips.find().toArray())

```

```

22 { "result" : "count_tips", "ok" : 1 }
23 26.823466323466324

```

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

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

```

public void getClosedBusinessInfo(){
    MongoClient<Document> business =
db.getCollection("yelp_academic_dataset_business");
    AggregateIterable aggRes = business.aggregate(Arrays.asList(
        Aggregates.match(Filters.and(Filters.eq("open",
false))),

Aggregates.project(Projections.fields(Projections.include("name",
"full_address", "stars"))));

    for (Object res: aggRes) {
        System.out.println(res);
    }
}

public static void main(String[] args) throws
UnknownHostException {
    MongoLab mongoLab = new MongoLab();
    mongoLab.getClosedBusinessInfo();
}

```

```

Document{{_id=608b32d3f13a50605f233cd9, full_address=2375 E Tropicana Ave
Ste 4
Southeast
Las Vegas, NV 89119, name=Optionz Fashion Lounge, stars=5.0}}
Document{{_id=608b32d4f13a50605f235e07, full_address=6605 Las Vegas Blvd S
Southeast
Las Vegas, NV 89119, name=Fresh52 Farmers Market, stars=3.5}}
Document{{_id=608b32d4f13a50605f236350, full_address=255 King Street N
Waterloo, ON N2J 4V2, name=YoYo's Yogurt Cafe, stars=3.5}}
Document{{_id=608b32d3f13a50605f233d02, full_address=450 Fremont St
Downtown
Las Vegas, NV 89101, name=Luna Rossa, stars=3.5}}
Document{{_id=608b32d3f13a50605f233d08, full_address=4700 N Central Ave
Phoenix, AZ 85012, name=Citizen Espresso Bar, stars=3.5}}
Document{{_id=608b32d3f13a50605f233d15, full_address=1020 S Mill Ave
Ste C
Tempe, AZ 85281, name=Meat Market Vintage, stars=4.5}}
Document{{_id=608b32d3f13a50605f233d37, full_address=10989 E Dynamite Blvd
Scottsdale, AZ 85262, name=Bella Notte Ristorante, stars=4.0}}
Document{{_id=608b32d3f13a50605f233d4b, full_address=7125 E 5th Ave
Ste 31
Scottsdale, AZ 85251, name=Baratin, stars=4.5}}
Document{{_id=608b32d4f13a50605f236389, full_address=3900 Paradise Rd
Ste N
Eastside
Las Vegas, NV 89169, name=Las Vegas Hookah, stars=4.5}}
Document{{_id=608b32d3f13a50605f233d55, full_address=8763 E Bell Rd
Scottsdale, AZ 85260, name=Sports Grill 101, stars=3.5}}
Document{{_id=608b32d4f13a50605f235e3f, full_address=6316 N Scottsdale Rd
Scottsdale, AZ 85253, name=Davanti Enoteca, stars=4.0}}
Document{{_id=608b32d4f13a50605f235e43, full_address=6803 N Durango Dr
<

```

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

```

public void getUsers(){
    MongoClient<Document> users =
db.getCollection("yelp_academic_dataset_user");
    AggregateIterable aggRes = users.aggregate(Arrays.asList(
Aggregates.match(Filters.or(Filters.eq("votes.funny", 0),
Filters.eq("votes.useful", 0))),
Aggregates.sort(Sorts.ascending("name"))));

    for (Object res: aggRes) {
        System.out.println(res);
    }
}

public static void main(String[] args) throws

```



```

UnknownHostException {
    MongoLab mongoLab = new MongoLab();
    mongoLab.getUsers();
}

```

```

Document{{_id=608b32dcf13a50605f2c363d, yelping_since=2011-07, votes=Document{{funny=0, usef=6, cool=1}}, review_count=6, name=sharon, user_id=viPKbRIYf
Document{{_id=608b32d5f13a50605f25208b, yelping_since=2008-11, votes=Document{{funny=0, usef=3, cool=0}}, review_count=2, name=shaun, user_id=eIzSWbYlCV
Document{{_id=608b32d3f13a50605f22b694, yelping_since=2011-10, votes=Document{{funny=0, usef=0, cool=0}}, review_count=8, name=shavon, user_id=j6-TXWL4q
Document{{_id=608b32d4f13a50605f23b673, yelping_since=2011-04, votes=Document{{funny=0, usef=1, cool=0}}, review_count=1, name=shawn, user_id=0pK7NiT-SC
Document{{_id=608b32dbf13a50605f2b6f6c, yelping_since=2011-06, votes=Document{{funny=0, usef=0, cool=0}}, review_count=1, name=shawna, user_id=5WuKTuRZ7
Document{{_id=608b32d6f13a50605f266578, yelping_since=2011-03, votes=Document{{funny=0, usef=4, cool=1}}, review_count=13, name=shawnee, user_id=ENR8W_G
Document{{_id=608b32d8f13a50605f284e1c, yelping_since=2009-02, votes=Document{{funny=0, usef=3, cool=0}}, review_count=7, name=shawnika, user_id=dR_-A23
Document{{_id=608b32def13a50605f2e918a, yelping_since=2008-03, votes=Document{{funny=0, usef=12, cool=2}}, review_count=18, name=shay, user_id=Vo3lnt68
Document{{_id=608b32dbf13a50605f2eb807, yelping_since=2011-02, votes=Document{{funny=0, usef=1, cool=1}}, review_count=1, name=shellie, user_id=KVibIBPs
Document{{_id=608b32d4f13a50605f23b20c, yelping_since=2011-08, votes=Document{{funny=0, usef=2, cool=0}}, review_count=5, name=shelly, user_id=xxyq_0N8Dm
Document{{_id=608b32dbf13a50605f2bd1a9, yelping_since=2009-03, votes=Document{{funny=0, usef=6, cool=2}}, review_count=2, name=shelly, user_id=KO3GscF3H
Document{{_id=608b32dbf13a50605f2d081c, yelping_since=2011-03, votes=Document{{funny=0, usef=2, cool=2}}, review_count=3, name=sherri, user_id=cb-YshgEc
Document{{_id=608b32d6f13a50605f258cb9, yelping_since=2009-12, votes=Document{{funny=0, usef=1, cool=0}}, review_count=11, name=sherwin, user_id=2GV1xro
Document{{_id=608b32d8f13a50605f28a803, yelping_since=2011-08, votes=Document{{funny=0, usef=1, cool=0}}, review_count=1, name=shinobu, user_id=jB4ZXWs2
Document{{_id=608b32d8f13a50605f28ca9a, yelping_since=2011-08, votes=Document{{funny=0, usef=2, cool=0}}, review_count=1, name=shirley, user_id=nzNP0LP
Document{{_id=608b32dbf13a50605f2bc52d, yelping_since=2008-11, votes=Document{{funny=0, usef=9, cool=2}}, review_count=9, name=shirley, user_id=Zfv2KrwL
Document{{_id=608b32ddf13a50605f2d54ab, yelping_since=2010-09, votes=Document{{funny=0, usef=4, cool=2}}, review_count=7, name=shiro, user_id=MU7GDS68iI
Document{{_id=608b32d6f13a50605f2a5f33, yelping_since=2011-07, votes=Document{{funny=0, usef=1, cool=0}}, review_count=4, name=siamac, user_id=2akfVQOiq
Document{{_id=608b32dbf13a50605f2bdaaf, yelping_since=2009-07, votes=Document{{funny=0, usef=1, cool=0}}, review_count=1, name=silke, user_id=fkrKmyqFhm
Document{{_id=608b32d1f13a50605f2098dc, yelping_since=2011-08, votes=Document{{funny=0, usef=3, cool=1}}, review_count=3, name=silvia, user_id=JKU070T0
Document{{_id=608b32d3f13a50605f22867b, yelping_since=2009-04, votes=Document{{funny=0, usef=0, cool=0}}, review_count=4, name=simon, user_id=swhd5EcSP
Document{{_id=608b32d4f13a50605f235851, yelping_since=2011-08, votes=Document{{funny=0, usef=0, cool=0}}, review_count=2, name=simone, user_id=Amo5Rugw
Document{{_id=608b32d7f13a50605f27468d, yelping_since=2009-01, votes=Document{{funny=0, usef=2, cool=0}}, review_count=4, name=siohban, user_id=w_7uUYXR
Document{{_id=608b32d3f13a50605f226573, yelping_since=2010-02, votes=Document{{funny=0, usef=4, cool=2}}, review_count=8, name=sk, user_id=JjX-Lx3Vj4Kdp
Document{{_id=608b32d5f13a50605f21e770, yelping_since=2011-08, votes=Document{{funny=0, usef=2, cool=1}}, review_count=2, name=skybrin, user_id=IOWM0wEx
Document{{_id=608b32d5f13a50605f2592e, yelping_since=2009-05, votes=Document{{funny=0, usef=8, cool=0}}, review_count=2, name=sleighand, user_id=v1Tb01
Document{{_id=608b32ddf13a50605f2d9338, yelping_since=2009-07, votes=Document{{funny=0, usef=1, cool=0}}, review_count=8, name=snighda, user_id=P-xvT439
Document{{_id=608b32ddf13a50605f2e1e3a, yelping_since=2011-02, votes=Document{{funny=0, usef=14, cool=2}}, review_count=13, name=snoog, user_id=ICxs2c5
Document{{_id=608b32ddf13a50605f2dac89, yelping_since=2010-03, votes=Document{{funny=0, usef=2, cool=0}}, review_count=17, name=sodany, user_id=s-R6KUM
Document{{_id=608b32d5f13a50605f24ffa8, yelping_since=2009-06, votes=Document{{funny=0, usef=1, cool=0}}, review_count=5, name=sol, user_id=l69gLOioQIMM

```

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

```

public void getTips(){
    MongoCollection<Document> tips =
    db.getCollection("yelp_academic_dataset_tip");
    AggregateIterable aggRes = tips.aggregate(Arrays.asList(
    Aggregates.match(Filters.and(Filters.regex("date", "2012"))),
    Aggregates.group("$business_id",
    Accumulators.sum("count", 1))));

    for (Object res: aggRes) {
        System.out.println(res);
    }
}

public static void main(String[] args) throws
UnknownHostException {
    MongoLab mongoLab = new MongoLab();
    mongoLab.getTips();
}

```

```
<terminated> MongoLab [Java Application] C:\Users\basia\.p2\pool\
Document{['_id=nm-s0cj2ehLWpkv0eseoLA', count=2]}
Document{['_id=a8xFB8jhDdlEudss-yMsWg', count=4]}
Document{['_id=x8cxaDdFUY5H1e-XdyM6FA', count=1]}
Document{['_id=-s9Vc4QS91I4DUfR9V4VgA', count=1]}
Document{['_id=HElt9itN1_A23pKxAZ2UAQ', count=7]}
Document{['_id=UAR9XXVVTQ5shxvU80C1dA', count=1]}
Document{['_id=GVerH-7DlZTDUSci38OWRA', count=1]}
Document{['_id=gJ0EC555EYYqFKbEUzEbog', count=5]}
Document{['_id=aXwTNEL63K4jf8bUiTpjSQ', count=1]}
Document{['_id=rDzvwiusMr5MMZxXo1GfMw', count=4]}
Document{['_id=j2HZ_eCJ-I1vsDto9Y8c-w', count=3]}
Document{['_id=50h0EQ20utvIxrMjNm15bA', count=4]}
Document{['_id=6xxa1ZONUbdG5ZZQhAglzw', count=1]}
Document{['_id=YF6Z8PhNWN0ZR33hfvZRjg', count=11]}
Document{['_id=7ei7R9Cen4x4S_fs2lxf7w', count=3]}
Document{['_id=LYLGCiQnQQrpMwOxJ1hlrg', count=2]}
Document{['_id=N-JfaAr4hRdjE5vV0i1dxw', count=1]}
Document{['_id=NzI4pAPrjlCkwcLgjseDtA', count=5]}
Document{['_id=A_vRqMo7HrlnCRFvzWVcfw', count=26]}
Document{['_id=f4T3s1x2u_KPOffIJIJ0uA', count=4]}
Document{['_id=iVtOE7_UGPsxiioGc034tQ', count=1]}
Document{['_id=xS7LC0cJbqh_Y5cIKxqWfQ', count=10]}
Document{['_id=hh-jaG2HPFdwul-yRdd00A', count=3]}
Document{['_id=FBLV0w09Ks4t2U0e0mF7HA', count=3]}
Document{['_id=YxMnfznT3eYya0YV37tE8w', count=12]}
Document{['_id=FbypkXpsdyboIiwhBwSSVA', count=8]}
Document{['_id=UmwUsHeVjUUtDmgB4jZz3w', count=1]}
<
```

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

```
public void getAvgStars(){
    MongoCollection<Document> review =
db.getCollection("yelp_academic_dataset_review");
    AggregateIterable aggRes = review.aggregate(Arrays.asList(
        Aggregates.group("$business_id",
        Accumulators.sum("avgStars", "$stars")),
        Aggregates.match(Filters.gte("avgStars", 4.0))
    ));

    for (Object res: aggRes) {
        System.out.println(res);
    }
}
```

```

    public static void main(String[] args) throws
UnknownHostException {
        MongoLab mongoLab = new MongoLab();
        mongoLab.getAvgStars();
    }

```

```

Document{{_id=Z4UuyC0XJTNQ3bAllu0Y9g, avgStars=7}}
Document{{_id=7hEhc5YSU3W-qLhj-ydU7Q, avgStars=8}}
Document{{_id=13HqRPuLUPyxaWwA9lylQ, avgStars=45}}
Document{{_id=T_-ih2IGAmDQAExnbF8S6g, avgStars=11}}
Document{{_id=bYdniNQEdYolz0onDWj0CA, avgStars=10}}
Document{{_id=j_3KufSCgZE0fOYjJ7ZCNQ, avgStars=71}}
Document{{_id=ky_0S-PRZ3jldfFgzBDmsg, avgStars=193}}
Document{{_id=SmD4SmtuSN_IsDSdix-kgg, avgStars=15}}
Document{{_id=GeLUSRVqE5_EY-TvYH9JnQ, avgStars=15}}
Document{{_id=2HUKoi6pLuKCKmu7v962Iw, avgStars=32}}
Document{{_id=nFIOyQ9-JSnND1T2GZ6Rdw, avgStars=8}}
Document{{_id=5MZrOEUD-L2B_v8X-ZxrHA, avgStars=23}}
Document{{_id=IjxzgcXb-_PCbxdc8xHrKQ, avgStars=11}}
Document{{_id=6QwvrXk-gcDMotRCua21QA, avgStars=12}}
Document{{_id=VHfj2t0Ko6aevNfwiDJzGg, avgStars=17}}
Document{{_id=-cVKo1S1B2jAs2Bgy2Lw_Q, avgStars=18}}
Document{{_id=6gBVMQaifDGUpJW_SpiIZg, avgStars=69}}
Document{{_id=YjeFZOP3wIkxv1pQs-Bb-Q, avgStars=23}}
Document{{_id=gS91Cpl82GwikC3Ig05qWg, avgStars=8}}
Document{{_id=1VHtVCyFJOqAH0ltn7pLuw, avgStars=569}}
Document{{_id=8ZSFR_x20FC1Kc1fm07uUA, avgStars=12}}
Document{{_id=HtI_uDwZ-obk--VW4JjWHw, avgStars=20}}
Document{{_id=rQcYXTFjjXn1qp8XUghrNA, avgStars=9}}
Document{{_id=C1luhYAaVzuoha89uJmr7g, avgStars=9}}
Document{{_id=IJnaMjZA0VecuOnBEYaFKA, avgStars=15}}
Document{{_id=XJONzQAeYpWxEkJvICivMg, avgStars=22}}
Document{{_id=57u7ZVNznPV1yFf0oOdqQw, avgStars=14}}
Document{{_id=D7_jP7nqNbt1E6iNxbg1Pw, avgStars=8}}
Document{{_id=Zj2jZe7E0B2qE0cshQra2A, avgStars=63}}
Document{{_id=vQaWT8qHhCJXuRI4NeBUCQ, avgStars=156}}
Document{{_id=WupCl5odiu_3F2lBrzDpsg, avgStars=144}}

```

e) Usuń wszystkie firmy (*business*), które posiadają ocenę (*stars*) równą 2.0.

```

public void deleteCompaniesRate2() {

    db.getCollection("yelp_academic_dataset_business").deleteMany(Filters.eq("stars", 2.0));
}

```

```

public void checkIfDeleted() {
    MongoCollection<Document> business =
db.getCollection("yelp_academic_dataset_business");
    AggregateIterable aggRes = business.aggregate(Arrays.asList(
        Aggregates.match(Filters.eq("stars", 2.0))
    ));

    System.out.println("Documets with 2.0 stars: ");

    for (Object res: aggRes) {
        System.out.println(res);
    }

    System.out.println("None!");
}

```

```

public static void main(String[] args) throws
UnknownHostException {
    MongoLab mongoLab = new MongoLab();
    mongoLab.deleteCompaniesRate2();
    mongoLab.checkIfDeleted();
}

```

Udało się poprawnie usunąć wszystkie dokumenty, w których firmy posiadają ocenę równą 2.0:

```

terminated: mongolab java application: C:\Users\user\AppData\Local\Programs\org.mongodb
maj 13, 2021 12:25:38 AM com.mongodb.diagnostics.logging.JULLogger:INFO: Cluster created with settings {hosts=[127.0.0.1:27017], mode=replica
Documets with 2.0 stars:
maj 13, 2021 12:25:38 AM com.mongodb.diagnostics.logging.JULLogger:INFO: Cluster description not yet available. Waiting for 30000 ms
maj 13, 2021 12:25:38 AM com.mongodb.diagnostics.logging.JULLogger:INFO: Opened connection [connectionId{localValue:1, serverValue:5}
maj 13, 2021 12:25:38 AM com.mongodb.diagnostics.logging.JULLogger:INFO: Monitor thread successfully connected to server with descri
maj 13, 2021 12:25:38 AM com.mongodb.diagnostics.logging.JULLogger:INFO: Opened connection [connectionId{localValue:2, serverValue:5}
None!

```

7. Zaproponuj bazę danych składającą się z 3 kolekcji pozwalającą przechowywać dane dotyczące: klientów, zakupu oraz przedmiotu zakupu. W bazie wykorzystaj: pola proste, złożone i tablice. Zaprezentuj strukturę dokumentów w formie JSON dla przykładowych danych. Uzasadnij swoją propozycję.

Proponowana przeze mnie baza składa się z 3 kolekcji, których strukturę widać na poniższych przykładowych dokumentach:

- Clients - składa się z pól: `_id`, nazwiska, imienia, nr telefonu, adresu e-mail oraz z tablicy przechowującej referencje do dokonanych zakupów (Shoppings).

```
11 {
12   "_id" : ObjectId("609c594a033702ff4cead7ac"),
13   "lastname" : "Nowakowska",
14   "firstname" : "Jolanta",
15   "phone" : "12345678",
16   "email" : "jn@gmail.com",
17   "shoppings" : [
18     DBRef("Shoppings", ObjectId("609c61d3033702ff4cead7b7")),
19     DBRef("Shoppings", ObjectId("609c61d3033702ff4cead7bc"))
20   ]
21 }
```

- Shoppings - składa się z pól: `_id`, pola zagnieżdżonego shop (reprezentującego sklep poprzez jego nazwę oraz adres), daty dokonania zakupu, sposobu płatności oraz z tablicy przechowującej referencję do produktów powiązanych z tym zakupem.

```
1 {
2   "_id" : ObjectId("609c61d3033702ff4cead7b7"),
3   "shop" : {
4     "shopName" : "Auchan",
5     "shopAddress" : "Aleja Mickiewicza 23"
6   },
7   "date" : ISODate("2021-05-13T00:00:00.000+0000"),
8   "payment" : "cash",
9   "Products" : [
10     DBRef("Products", ObjectId("609c5fab033702ff4cead7b1")),
11     DBRef("Products", ObjectId("609c5fab033702ff4cead7b2")),
12     DBRef("Products", ObjectId("609c5fab033702ff4cead7b1")),
13     DBRef("Products", ObjectId("609c5fab033702ff4cead7b1")),
14     DBRef("Products", ObjectId("609c5fab033702ff4cead7b1")),
15     DBRef("Products", ObjectId("609c5fab033702ff4cead7b2")),
16     DBRef("Products", ObjectId("609c5fab033702ff4cead7b2")),
17     DBRef("Products", ObjectId("609c5fab033702ff4cead7b2"))
18   ]
19 }
```

- Products - przedmioty zakupu - składa się z pól: `_id`, nazwy, kategorii, ceny oraz opisu.

```

22 {
23   "_id" : ObjectId("609c5fab033702ff4cead7b3"),
24   "name" : "Kanapka",
25   "category" : "Spożywcze",
26   "price" : 7.99,
27   "description" : "Pyszna!"
28 }
29 {
30   "_id" : ObjectId("609c5fab033702ff4cead7b4"),
31   "name" : "Croissant",
32   "category" : "Spożywcze",
33   "price" : 6.99,
34   "description" : "Chrupiący!"
35 }
36 {

```

Baza obsługiwana jest przy użyciu następujących funkcji:

- AddClient - dodanie klienta do kolekcji

```

function addClient(firstname, lastname, phone, email){
  db.Clients.insert({
    lastname: lastname,
    firstname: firstname,
    phone: phone,
    email: email,
    shoppings: []
  })
}

```

Przykładowe wywołanie:

```

addClient("Aneta", "Lipińska", "12345678", "a.lipinska@onet.pl")

```

- AddProduct - dodaje produkt do kolekcji

```

function addProduct(name, category, price, description){
  db.Products.insert({
    name: name,
    category: category,
    price: price,
    description: description
  })
}

```



```
    })  
  }  
}
```

Przykładowe wywołanie:

```
addProduct("Croissant", "Spożywcze", 6.99, "Chrupiący!");
```

- addShopping - dodaje zakup do kolekcji:

```
function addShopping(shopName, shopAddress, date, payment){  
  db.Shopping.insert({  
    shop: {  
      shopName: shopName,  
      shopAddress: shopAddress  
    },  
    date: ISODate(date),  
    payment: payment,  
    Products: []  
  })  
}
```

Przykładowe wywołanie:

```
addShopping("Auchan", "Aleja Mickiewicza 23", "2021-05-13", "cash")
```

- addProductToShopping - dodaje produkt do zakupów.

```
function addProductToShopping(shoppingID, productID){  
  db.Shoppings.update(  
    {"_id": ObjectId(shoppingID)},  
    {$push: {  
      Products: {$ref: "Products", $id: ObjectId(productID)}}  
    },  
    );  
}
```

Przykładowe wywołanie:

```
addProductToShopping("609c61d3033702ff4cead7b7",
```

```
"609c5fab033702ff4cead7b1")
```

- addShoppingToClient - umożliwia umieszczenie wydarzenia zakupowego w liście zakupów klienta:

```
function addShoppingToClient(clientID, shoppingID){
  db.Clients.update(
    {"_id": ObjectId(clientID)},
    {$addToSet: {
      shoppings: {$ref: "Shoppings", $id:
ObjectId(shoppingID)}
    }}
  );
}
```

Przykładowe wywołanie:

```
addShoppingToClient("609c594a033702ff4cead7ac",
"609c61d3033702ff4cead7b7")
```

Uzasadnienie: Taki wybór kolekcji wydaje się spełniać wymagania. W funkcji addShoppingToClient dodajemy dany zakup poprzez metodę addToSet, która nie pozwala na zduplikowanie wydarzenia zakupu w zbiorze, na czym nam zależy. Z kolei w funkcji addProductToShopping korzystamy z metody push, która umożliwia dodanie kilku tych samych produktów do danego wydarzenia zakupowego. Dzięki temu możemy za pośrednictwem mechanizmu agregacji zliczyć wystąpienia każdego produktu w ramach każdego wydarzenia zakupowego (symulacja paragonu).