

## Bazy danych – NoSQL MongoDB – zadania

Imię i nazwisko: Uładzislau Tumilovich

Tydzień B, godz lab.: 11:15 – 12:45

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

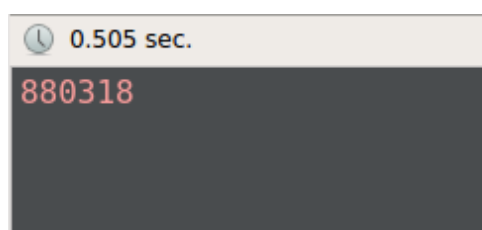
- a) Zwróć bez powtórzeń wszystkie nazwy miast w których znajdują się firmy (*business*). Wynik posortuj na podstawie nazwy miasta alfabetycznie.

```
db.getCollection('business').distinct('city').sort()
```

Key	Value	Type
(1)	[122 elements]	Array
[0]	Ahwatukee	String
[1]	Anthem	String
[2]	Apache Junction	String
[3]	Arcadia	String
[4]	Atlanta	String
[5]	Avondale	String
[6]	Black Canyon City	String
[7]	Bonnyrigg	String
[8]	Boulder City	String
[9]	Buckeye	String
[10]	C Las Vegas	String
[11]	Cambridge	String
[12]	Carefree	String
[13]	Casa Grande	String
[14]	Cave Creek	String
[15]	Centennial Hills	String
[16]	Central City Village	String
[17]	Central Henderson	String
[18]	Chandler	String
[19]	Chandler-Gilbert	String
[20]	City of Edinburgh	String
[21]	Clark County	String
[22]	Columbus	String
[23]	Coolidge	String
[24]	Cottage Grove	String
[25]	Cramond	String
[26]	Dalkeith	String
[27]	Dane	String
[28]	De Forest	String
[29]	DeForest	String
[30]	DeForest	String
[31]	Eagan	String
[32]	Edinburgh	String
[33]	El Mirage	String
[34]	Enterprise	String
[35]	Fitchburg	String
[36]	Florence	String
[37]	Fort Kinnaird	String
[38]	Fort McDowell	String
[39]	Fort McDowell	String
[40]	Fountain Hills	String
[41]	Fountain Hls	String
[42]	Gila Bend	String
[43]	Gilbert	String

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

```
db.getCollection('review').find( {  
  date: { $gte: new Date("2011/01/01").toISOString() }  
} )  
.count()
```



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

```

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

```

business 0.002 sec.		
Key	Value	Type
(1) ObjectId("5e9067e24bbdc1a9bb7d1a18")	{ 4 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a18")	ObjectId
full_address	4156 County Rd B Mc Farland, WI 53558	String
name	Charter Communications	String
stars	1.5	Double
(2) ObjectId("5e9067e24bbdc1a9bb7d1a26")	{ 4 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a26")	ObjectId
full_address	6401 University Ave Middleton, WI 53562	String
name	Crandalls Carryout & Catering	String
stars	4.0	Double
(3) ObjectId("5e9067e24bbdc1a9bb7d1a27")	{ 4 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a27")	ObjectId
full_address	6230 University Ave Middleton, WI 53562	String
name	Mi Cocina	String
stars	3.0	Double
(4) ObjectId("5e9067e24bbdc1a9bb7d1a59")	{ 4 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a59")	ObjectId
full_address	1901 Cayuga St Middleton, WI 53562	String
name	Soup Factory	String
stars	3.0	Double
(5) ObjectId("5e9067e24bbdc1a9bb7d1a73")	{ 4 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a73")	ObjectId
full_address	1632 W Main St Sun Prairie, WI 53590	String
name	Deli Roma	String
stars	4.0	Double

- d) 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('user').find({
  $or:
  [
    {'votes.funny': {$eq: 0}},
    {'votes.useful': {$eq: 0}}
  ]
})
.sort({'name': 1})

```

user 0.002 sec.		
Key	Value	Type
_id	ObjectId("5e9068c3c30c1ce8610eb402")	ObjectId
yelping_since	2009-08	String
votes	{ 3 fields }	Object
review_count	1	Int32
name	Bernard	String
user_id	xP3SPgfgW2vc5Zj5uV8SEA	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	5.0	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array
(2) ObjectId("5e9068c4c30c1ce8610fb228")	{ 12 fields }	Object
_id	ObjectId("5e9068c4c30c1ce8610fb228")	ObjectId
yelping_since	2011-12	String
votes	{ 3 fields }	Object
review_count	10	Int32
name	'Anastacia	String
user_id	qJLc0rYytqzeVBtPFTfSA	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	4.5	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array

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

```

db.getCollection('tip').aggregate([
  { $match: {"date": /2012/} },

```

```
{ $group:
  {
    _id: "$business_id",
    tips: {$sum: 1}
  }
},
{ $sort: {tips: 1} }
}]
```

tip 0.261 sec.		
Key	Value	Type
_id	Gtcq8csNoko-L7SwMLVqA	String
tips	1.0	Double
(2) J66qFJely2dWxlpMECCA_g	{ 2 fields }	Object
_id	J66qFJely2dWxlpMECCA_g	String
tips	1.0	Double
(3) PAgCM-p-VqXic5iVtTpx1g	{ 2 fields }	Object
_id	PAgCM-p-VqXic5iVtTpx1g	String
tips	1.0	Double
(4) 1WFZMZz2-jEuKl7jsKj7DQ	{ 2 fields }	Object
_id	1WFZMZz2-jEuKl7jsKj7DQ	String
tips	1.0	Double
(5) ikcyah81MTF4ZzOivmPitw	{ 2 fields }	Object
_id	ikcyah81MTF4ZzOivmPitw	String
tips	1.0	Double
(6) uuT-uF66XOrqWwsSlrFG2A	{ 2 fields }	Object
_id	uuT-uF66XOrqWwsSlrFG2A	String
tips	1.0	Double
(7) M9BJRlfzly-xQe0cr1bPHA	{ 2 fields }	Object
_id	M9BJRlfzly-xQe0cr1bPHA	String
tips	1.0	Double
(8) Nc6GQOf3sr6zaOm-E0v4_g	{ 2 fields }	Object
_id	Nc6GQOf3sr6zaOm-E0v4_g	String
tips	1.0	Double
(9) hhC02OKU6BWRvrNJQIEIUw	{ 2 fields }	Object
_id	hhC02OKU6BWRvrNJQIEIUw	String

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

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

review 0.925 sec.		
Key	Value	Type
_id	Ko14MmmrA-FYcYkoXSrCpQ	String
avg_stars	4.666666666666667	Double
(2) 8poYe1NpFINNJ5xiN9lpqw	{ 2 fields }	Object
_id	8poYe1NpFINNJ5xiN9lpqw	String
avg_stars	4.75	Double
(3) mXr-1rBHjaGGnqX4MgLtA	{ 2 fields }	Object
_id	mXr-1rBHjaGGnqX4MgLtA	String
avg_stars	4.333333333333333	Double
(4) RUZvUPOn90ScX60eETwcCw	{ 2 fields }	Object
_id	RUZvUPOn90ScX60eETwcCw	String
avg_stars	4.33673469387755	Double
(5) 8ylgPPiXsfOa1wD44CFQ2w	{ 2 fields }	Object
_id	8ylgPPiXsfOa1wD44CFQ2w	String
avg_stars	4.0	Double
(6) nWm5m-O88qprGsX4sjo0Uw	{ 2 fields }	Object
_id	nWm5m-O88qprGsX4sjo0Uw	String
avg_stars	4.25	Double
(7) -glghHqTCCuLsvLGXaoPg	{ 2 fields }	Object
_id	-glghHqTCCuLsvLGXaoPg	String
avg_stars	4.166666666666667	Double
(8) a12jKFiqBVyaUFTWNaAzQQ	{ 2 fields }	Object
_id	a12jKFiqBVyaUFTWNaAzQQ	String
avg_stars	4.33884297520661	Double
(9) 9p1o6jCY4fkQAJRmEjgSMw	{ 2 fields }	Object
_id	9p1o6jCY4fkQAJRmEjgSMw	String

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

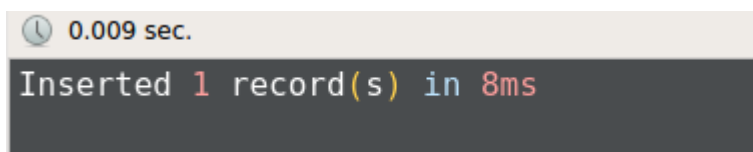
```
db.getCollection('business').deleteMany({stars: 2.0})
```

0.075 sec.		
Key	Value	Type
(1)	{ 2 fields }	Object
acknowledged	true	Boolean
deletedCount	1576.0	Double

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

```
function addReview(user_id, review_id, text, business_id) {
  db.getCollection('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
  })
}
```

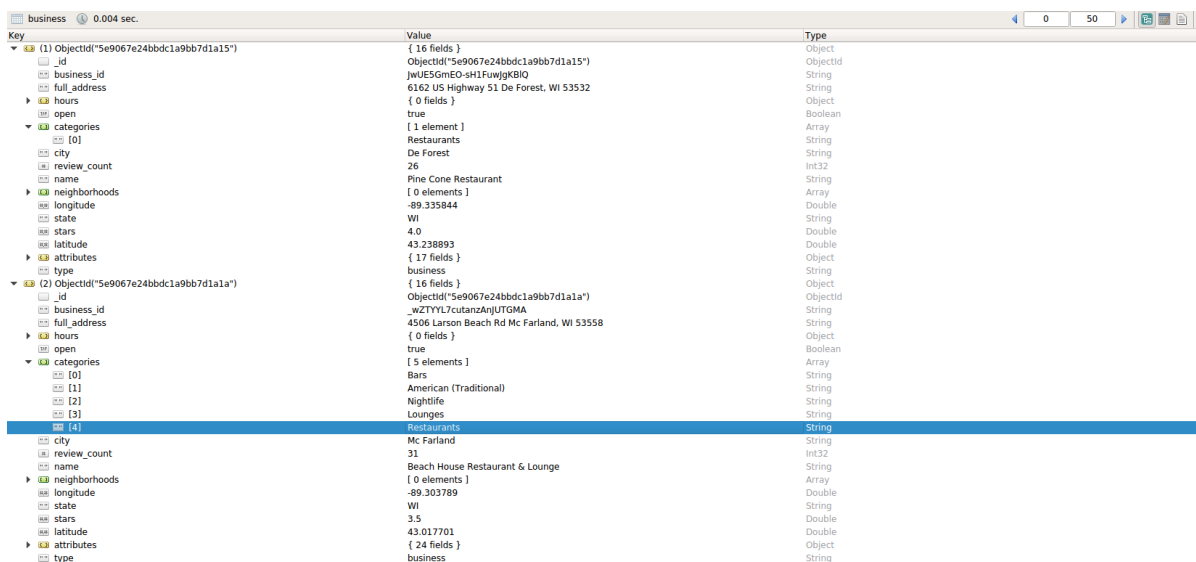
```
addReview('QnhQ8G51XbUpVEyWY2Km-A', 'FRTCsZJWkJonDAZx3yr8FA', 'Text review', 'JwUE5GmEO-sH1FuwJgKB1Q')
```



3. Zdefiniuj funkcję (*MongoDB*), 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 showBusinessByCategory(category) {
  return db.getCollection('business').find({ categories: category })
}

showBusinessByCategory("Restaurants")
```



Key	Value	Type
(1) ObjectId("5e9067e24bbdc1a9bb7d1a15")	{ 16 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a15")	ObjectId
business_id	JwUE5GmEO-sH1FuwJgKB1Q	String
full_address	6162 US Highway 51 De Forest, WI 53532	String
hours	{ 0 fields }	Object
open	true	Boolean
categories	[ 1 element ]	Array
[0]	Restaurants	String
city	De Forest	String
review_count	26	Int32
name	Pine Cone Restaurant	String
neighborhoods	[ 0 elements ]	Array
longitude	-89.335844	Double
state	WI	String
stars	4.0	Double
latitude	43.238893	Double
attributes	{ 17 fields }	Object
type	business	String
(2) ObjectId("5e9067e24bbdc1a9bb7d1a1a")	{ 16 fields }	Object
_id	ObjectId("5e9067e24bbdc1a9bb7d1a1a")	ObjectId
business_id	_wZTYYL7cutanzAnjUTGMA	String
full_address	4506 Larson Beach Rd Mc Farland, WI 53558	String
hours	{ 0 fields }	Object
open	true	Boolean
categories	[ 5 elements ]	Array
[0]	Bars	String
[1]	American (Traditional)	String
[2]	Nightlife	String
[3]	Lounges	String
[4]	Restaurants	String
city	Mc Farland	String
review_count	31	Int32
name	Beach House Restaurant & Lounge	String
neighborhoods	[ 0 elements ]	Array
longitude	-89.303789	Double
state	WI	String
stars	3.5	Double
latitude	43.017701	Double
attributes	{ 24 fields }	Object
type	business	String

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

```
function changeUserName(user_id, user_name) {
    db.getCollection('user').findOneAndUpdate(
        { _id: new ObjectId(user_id) },
        { $set: { name: user_name } }
    )
}

changeUserName('5e9068c2c30c1ce8610df3eb', 'Alexandra')
```

before:

Key	Value	Type
(1) ObjectId("5e9068c2c30c1ce8610df3eb")	{ 12 fields }	Object
_id	ObjectId("5e9068c2c30c1ce8610df3eb")	ObjectId
yelping_since	2012-01	String
votes	{ 3 fields }	Object
review_count	1	Int32
name	Alex	String
user_id	5Xh4Qc3rxhAQ_NcNtxLssQ	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	1.0	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array

after:

Key	Value	Type
(1) ObjectId("5e9068c2c30c1ce8610df3eb")	{ 12 fields }	Object
_id	ObjectId("5e9068c2c30c1ce8610df3eb")	ObjectId
yelping_since	2012-01	String
votes	{ 3 fields }	Object
review_count	1	Int32
name	Alexandra	String
user_id	5Xh4Qc3rxhAQ_NcNtxLssQ	String
friends	[ 0 elements ]	Array
fans	0	Int32
average_stars	1.0	Double
type	user	String
compliments	{ 0 fields }	Object
elite	[ 0 elements ]	Array

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

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

var reduce_fun = function(business_id, likes) {
    return Array.avg(likes);
};

db.getCollection('tip').
mapReduce(
    map_fun,
    reduce_fun,
    { out: "averageBusinessTip" }
)

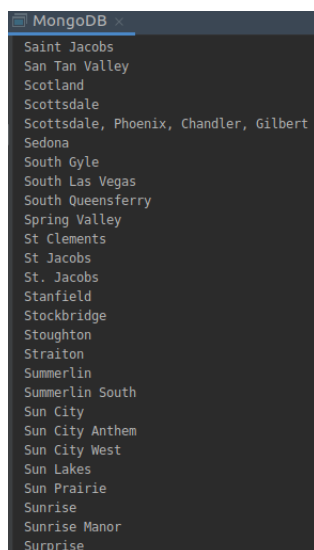
db.averageBusinessTip.find()
```

averageBusinessTip 0.001 sec.		
Key	Value	Type
▼ (1) --1emggGHgoG6ipd_RMb-g	{ 2 fields }	Object
_id	--1emggGHgoG6ipd_RMb-g	String
value	0.0	Double
▼ (2) --5jkZ3-nUPZxUvtcbr8Uw	{ 2 fields }	Object
_id	--5jkZ3-nUPZxUvtcbr8Uw	String
value	0.0625	Double
▼ (3) --BlvDO_RG2yElKu9XA1_g	{ 2 fields }	Object
_id	--BlvDO_RG2yElKu9XA1_g	String
value	0.0	Double
▼ (4) --Dl2rW_xO8GuYBomlg9zw	{ 2 fields }	Object
_id	--Dl2rW_xO8GuYBomlg9zw	String
value	0.0	Double
▼ (5) --Y_2lDOTvDioX5bwF6GIw	{ 2 fields }	Object
_id	--Y_2lDOTvDioX5bwF6GIw	String
value	0.0	Double
▼ (6) --jFTZmywe7StuZ2hEjxyA	{ 2 fields }	Object
_id	--jFTZmywe7StuZ2hEjxyA	String
value	0.0	Double
▼ (7) --qeSYxyn62mMjWvznNTdg	{ 2 fields }	Object
_id	--qeSYxyn62mMjWvznNTdg	String
value	0.0	Double
▼ (8) --sSW-WY3vyAsh_eVPGUaw	{ 2 fields }	Object
_id	--sSW-WY3vyAsh_eVPGUaw	String
value	0.0	Double
▼ (9) -0Ackw6MF82PX09f9Jh_Kg	{ 2 fields }	Object
_id	-0Ackw6MF82PX09f9Jh_Kg	String
value	0.03333333333333333	Double

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

```
private List<String> getBusinessCities() {
    List<String> cities = db
        .getCollection("business")
        .distinct("city", String.class)
        .into(new ArrayList<>());
    Collections.sort(cities);
    return cities;
}
```



```
MongoDB >
Saint Jacobs
San Tan Valley
Scotland
Scottsdale
Scottsdale, Phoenix, Chandler, Gilbert
Sedona
South Gyle
South Las Vegas
South Queensferry
Spring Valley
St Clements
St Jacobs
St. Jacobs
Stanfield
Stockbridge
Stoughton
Straiton
Summerlin
Summerlin South
Sun City
Sun City Anthem
Sun City West
Sun Lakes
Sun Prairie
Sunrise
Sunrise Manor
Surprise
```

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

```
private long getNumberOfReviews() {
    return db
        .getCollection("review")
        .countDocuments(new Document("date",
            new Document("$gte", "2011-01-01")));
}
```

```
}
```

```
880318
```

```
Process finished with exit code 0
```

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

```
private List<Document> getClosedBusinessesInfo() {
    return db
        .getCollection("business")
        .find(eq("open", false))
        .projection(fields(include("name", "stars", "full_address"), excludeId()))
        .into(new ArrayList<>());
}
```

```
MongoDB x
Las Vegas, NV 89123, name=Sin City Spray Tans, stars=5.0}}
Document{{full_address=12645 N Saguaro Blvd
Fountain Hills, AZ 85268, name=Smokey's BBQ, stars=3.0}}
Document{{full_address=119 Rose Street
New Town
New Town EH2 3DT, name=El Barrio, stars=3.5}}
Document{{full_address=2185 E Windmill Ln
Ste 200
Southeast
Las Vegas, NV 89123, name=Vid Noodles, stars=4.0}}
Document{{full_address=10430 S Eastern Ave
Anthem
Henderson, NV 89052, name=Carmine's, stars=3.0}}
Document{{full_address=3000 W Ann Rd
Ste 109
North Las Vegas, NV 89031, name=Super No1 Chinese Restaurant, stars=3.5}}
Document{{full_address=13014 N Saguaro Blvd
Fountain Hills, AZ 85268, name=Moxie Modern Fare, stars=3.5}}

Process finished with exit code 0
```

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

```
private List<Document> getUserInfo() {
    return db
        .getCollection("user")
        .find(or(eq("votes.funny", 0), eq("votes.useful", 0)))
        .sort(ascending("name"))
        .into(new ArrayList<>());
}
```

```
MongoDB
Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-08, votes=Document{{funny=0, useful=0, review_count=1, name=yennib, user_id=6w6b28l6wJ2T7f64y40, friends=[], fans=0, average_stars=0.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-01, votes=Document{{funny=0, useful=0, review_count=1, name=yiran, user_id=jWLVuX4qfs-uPQ5-g, friends=[], fans=0, average_stars=2.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2009-03, votes=Document{{funny=0, useful=1, review_count=3, name=yolanda, user_id=BMjot9WqyA18T23pgg, friends=[], fans=0, average_stars=3.33, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2007-11, votes=Document{{funny=0, useful=1, review_count=3, name=yolanda, user_id=PaqC7u7uWuM4s5G5w, friends=[], fans=0, average_stars=2.5, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2008-03, votes=Document{{funny=0, useful=0, review_count=2, name=yitan, user_id=3p2p2qC7F7uvcf7r, friends=[], fans=0, average_stars=1.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-01, votes=Document{{funny=0, useful=3, review_count=5, name=your, user_id=Vancpu_8Mq9M-Luy4w, friends=[], fans=0, average_stars=3.2, type=user, compliments=Document({note=1}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2010-06, votes=Document{{funny=0, useful=1, review_count=1, name=your-nom, user_id=6w6BEC2W05BM2Q5e90w, friends=[], fans=0, average_stars=4.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-03, votes=Document{{funny=0, useful=5, review_count=3, name=yoyz, user_id=Hq5ShGJ7F6vKXQy40, friends=[], fans=0, average_stars=4.52, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2010-03, votes=Document{{funny=0, useful=0, review_count=3, name=yumey, user_id=4Y49y6b0u9C7QyCj(CPS100, friends=[], fans=0, average_stars=3.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-08, votes=Document{{funny=0, useful=7, review_count=22, name=yvonneandscott, user_id=wgChtA-Pa007hucxkm, friends=[], fans=1, average_stars=3.45, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2010-03, votes=Document{{funny=0, useful=6, review_count=5, name=zeus, user_id=4wYVJ3C0y3K5EYLMg, friends=[], fans=0, average_stars=2.4, type=user, compliments=Document({cool=1}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-10, votes=Document{{funny=0, useful=2, review_count=2, name=zama, user_id=8WJ27X1v1r1230, friends=[], fans=0, average_stars=1.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2010-07, votes=Document{{funny=0, useful=2, review_count=3, name=zoo, user_id=QmbyYlKqipPhciq3Mw, friends=[], fans=0, average_stars=5.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2011-10, votes=Document{{funny=0, useful=0, review_count=3, name=zb1rd, user_id=Qd6w0FRBX0NC1161A, friends=[], fans=0, average_stars=4.5, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2014-01, votes=Document{{funny=0, useful=0, review_count=1, name=Angelina, user_id=crp2FYfYy4c9PmBqf, friends=[], fans=0, average_stars=1.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2009-07, votes=Document{{funny=0, useful=0, review_count=1, name=Izandro, user_id=GE_5wH0p0C2Bq72220, friends=[], fans=0, average_stars=0.0, type=user, compliments=Document({}), elite=({})}, Document{{_id=5e9868c3c3b1ce881b8c1b, yelping_since=2014-07, votes=Document{{funny=0, useful=0, review_count=1, name=I8r, user_id=6Hb260K73M nxCBq7g, friends=[], fans=0, average_stars=1.0, type=user, compliments=Document({}), elite=({})},

Process finished with exit code 0
```

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

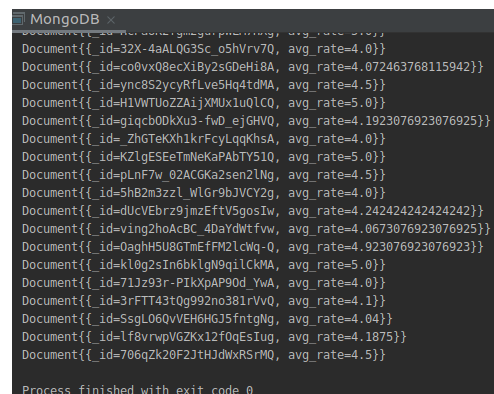
```
private List<Document> getBusinessesTipNumber() {  
    return db  
        .getCollection("tip")  
        .aggregate(Collections.singletonList(  
            Aggregates.group("$business_id",  
                Accumulators.sum("tips", 1)  
            )  
        ))  
        .into(new ArrayList<>());  
}
```



```
MongoDB x  
Document{{_id=QSztn91NziilWwEhh0B94A, tips=4}}  
Document{{_id=k8ka4-hVR-uxNlCvDt5Vaw, tips=2}}  
Document{{_id=F0l8mp0U8BZDF2IBewZduw, tips=1}}  
Document{{_id=IX0zkb2L13-Q3EGWvTem0w, tips=1}}  
Document{{_id=iQEX5MJ0WfFcUqdXl--6UA, tips=181}}  
Document{{_id=H0gLsxmzxpH2w40If-DmBg, tips=2}}  
Document{{_id=0ttAiIRJcxXmMK7ClLYHkw, tips=1}}  
Document{{_id=C2WHc4srB9Yj6VoYq5pH_g, tips=1}}  
Document{{_id=6fgu-TSXWRVSkdvL3eNkYQ, tips=6}}  
Document{{_id=0vS1mJR0cz94TiGP_j3cFw, tips=3}}  
Document{{_id=7ckguqzYBLAYaqme-qmrWg, tips=2}}  
Document{{_id=a3VX8WpAVt29VnLGgYJxrA, tips=134}}  
Document{{_id=LSIyCvaWZl-Ex9y3r2nDPg, tips=1}}  
Document{{_id=4xIfRXmnbsj996Jm8HoUgw, tips=3}}  
Document{{_id=ej65W-XSc1PL_ilssBJe2A, tips=4}}  
Document{{_id=wfdXhG4KY8qgPKJ36x0Z2Q, tips=3}}  
Document{{_id=SL04b5w6WvEmLgQX-FDu9A, tips=4}}  
Document{{_id=F0fHEC6rN23RQaUlsR2byw, tips=27}}  
Document{{_id=5Rtds_pTigLiIE0m5KSAPA, tips=13}}  
  
Process finished with exit code 0
```

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

```
private List<Document> getHighRatedBusinesses() {  
    return db  
        .getCollection("review")  
        .aggregate(Arrays.asList(  
            Aggregates.group("$business_id",  
                Accumulators.avg("avg_rate", "$stars")  
            ), Aggregates.match(Filters.gte("avg_rate", 4.0))  
        ))  
        .into(new ArrayList<>());  
}
```



```
MongoDB x  
Document{{_id=32X-4aALQ63Sc_o5Hvrv7Q, avg_rate=4.0}}  
Document{{_id=co0vxQ8ecXiBy2sG0eHi8A, avg_rate=4.072463768115942}}  
Document{{_id=ync8S2ycyRfLve5Hq4tdMA, avg_rate=4.5}}  
Document{{_id=H1WUoZZaijXMuXiu0lCO, avg_rate=5.0}}  
Document{{_id=giqcb00kXu3-fw0_ejGHV0, avg_rate=4.1923076923076925}}  
Document{{_id=ZhGTeKXh1krFcyLqQkhsA, avg_rate=4.0}}  
Document{{_id=KZlgE5EetmNeKaPAbTY51Q, avg_rate=5.0}}  
Document{{_id=PlnF7w_02ACGKa2sen2lNg, avg_rate=4.5}}  
Document{{_id=ShB2m3zzl_WlGr9bJVCY2g, avg_rate=4.0}}  
Document{{_id=duCvEbrz9jnzEftV5gosIw, avg_rate=4.242424242424242}}  
Document{{_id=ving2hoAcBC_4DaYdWtfvW, avg_rate=4.0673076923076925}}  
Document{{_id=0aghH5U8GTmEfFM2lWq-Q, avg_rate=4.923076923076923}}  
Document{{_id=k10g2sIn6bk1gN9qilCkMA, avg_rate=5.0}}  
Document{{_id=71jz93r-PikXpAP90d_YwA, avg_rate=4.0}}  
Document{{_id=3rFTT43t0g992no381rVvQ, avg_rate=4.1}}  
Document{{_id=SsgL06QvVEH6HGJ5fntgNg, avg_rate=4.04}}  
Document{{_id=l8vrvpV6ZKx12f0qEsIug, avg_rate=4.1875}}  
Document{{_id=706qZk20F2jtHJDWxRSrMQ, avg_rate=4.5}}  
  
Process finished with exit code 0
```



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

```
private void deleteLowRatedBusinesses() {  
    db.getCollection("business").deleteMany(eq("stars", 2.0));  
}
```

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

```
// Clients  
function addClient(firstname, lastname, contact) {  
    db.getCollection('Clients').insert({  
        firstname: firstname,  
        lastname: lastname,  
        contact: contact,  
        purchases: []  
    });  
};  
  
addClient('Uladzislau', 'Tumilovich', 'tumich@example.com');  
addClient('Kamil', 'Nowak', 'kamilexample.com');  
addClient('Kasia', 'Andrukiewicz', '4343453453');  
addClient('Mark', 'Krembik', '1435345234');  
  
db.getCollection('Clients').find();  
  
// Items  
function addItem(name, price, category, origin_country) {  
    db.getCollection('Items').insert({  
        name: name,  
        price: price,  
        category: category,  
        origin_country: origin_country  
    });  
};  
  
addItem('Biuro', 399.99, 'Mebla', 'Poland');  
addItem('Krzeslo', 299.99, 'Mebla', 'Belarus');  
addItem('Monitor', 649.99, 'Elektronika', 'Germany');  
addItem('Herbata', 4.29, 'Produkty', 'China');  
  
db.getCollection('Items').find();  
  
// Purchases  
function addPurchase(seller, address) {  
    db.getCollection('Purchases').insert({  
        seller: seller,  
        address: address,  
        date: new Date(),  
        items: []  
    });  
};  
  
addPurchase('Ikea', 'ul. GdziesTam, 32, Krakow');  
addPurchase('Lewiatan', 'ul. GdziesTam, 32, Warszawa');  
addPurchase('MediaMarket', 'ul. GdziesTam, 32, Gdansk');  
  
db.getCollection('Purchases').find();  
  
// add Purchase to Client  
function addPurchaseToClient(client_id, purchase_id) {  
    db.getCollection('Clients').update(  
        { '_id': client_id, 'purchases': purchase_id },  
        { '$push': 'purchases', '$addToSet': 'purchases.items', '$set': 'purchases.items.$addToSet.date', '$set': 'purchases.items.$addToSet.seller', '$set': 'purchases.items.$addToSet.address' },  
        { 'upsert': true },  
        { 'multi': true }
```

```

        { _id: new ObjectId(client_id) },
        { $addToSet: { purchases: { $ref: "Purchases", $id: new
ObjectId(purchase_id) } } }
    );
};

addPurchaseToClient("5e9c5be22cd52bf242f0e9b7", "5e9c65a62cd52bf242f0e9c6");
addPurchaseToClient("5e9c5be22cd52bf242f0e9b8", "5e9c65a62cd52bf242f0e9c7");
addPurchaseToClient("5e9c5be22cd52bf242f0e9b8", "5e9c65a62cd52bf242f0e9c8");

db.getCollection('Clients').find();

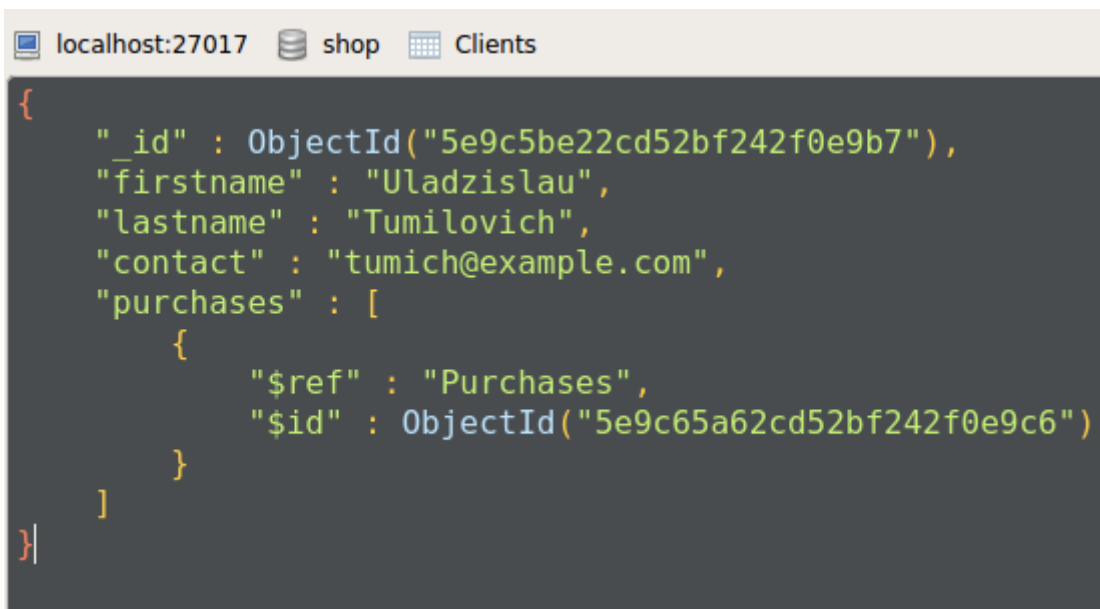
// add Item to Purchase
function addItemToPurchase(purchase_id, item_id) {
    db.getCollection("Purchases").update(
        { _id: new ObjectId(purchase_id) },
        { $addToSet: { items: { $ref: 'Items', $id: new ObjectId(item_id) } } }
    );
};

addItemToPurchase('5e9c65a62cd52bf242f0e9c6', '5e9c5ea52cd52bf242f0e9bf');
addItemToPurchase('5e9c65a62cd52bf242f0e9c7', '5e9c5ea52cd52bf242f0e9c0');
addItemToPurchase('5e9c65a62cd52bf242f0e9c8', '5e9c5ea52cd52bf242f0e9c1');
addItemToPurchase('5e9c65a62cd52bf242f0e9c6', '5e9c5ea52cd52bf242f0e9c2');

db.getCollection('Purchases').find()

```

Struktura kolekcji Clients przedstawiona na danych przykładowych:

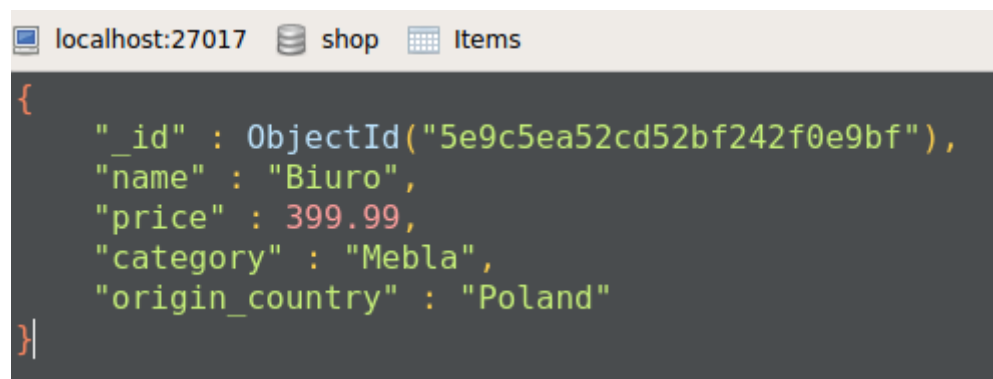


```

localhost:27017  shop  Clients
{
  "_id" : ObjectId("5e9c5be22cd52bf242f0e9b7"),
  "firstname" : "Uładzislau",
  "lastname" : "Tumilovich",
  "contact" : "tumich@example.com",
  "purchases" : [
    {
      "$ref" : "Purchases",
      "$id" : ObjectId("5e9c65a62cd52bf242f0e9c6")
    }
  ]
}

```

Struktura kolekcji Items przedstawiona na danych przykładowych:

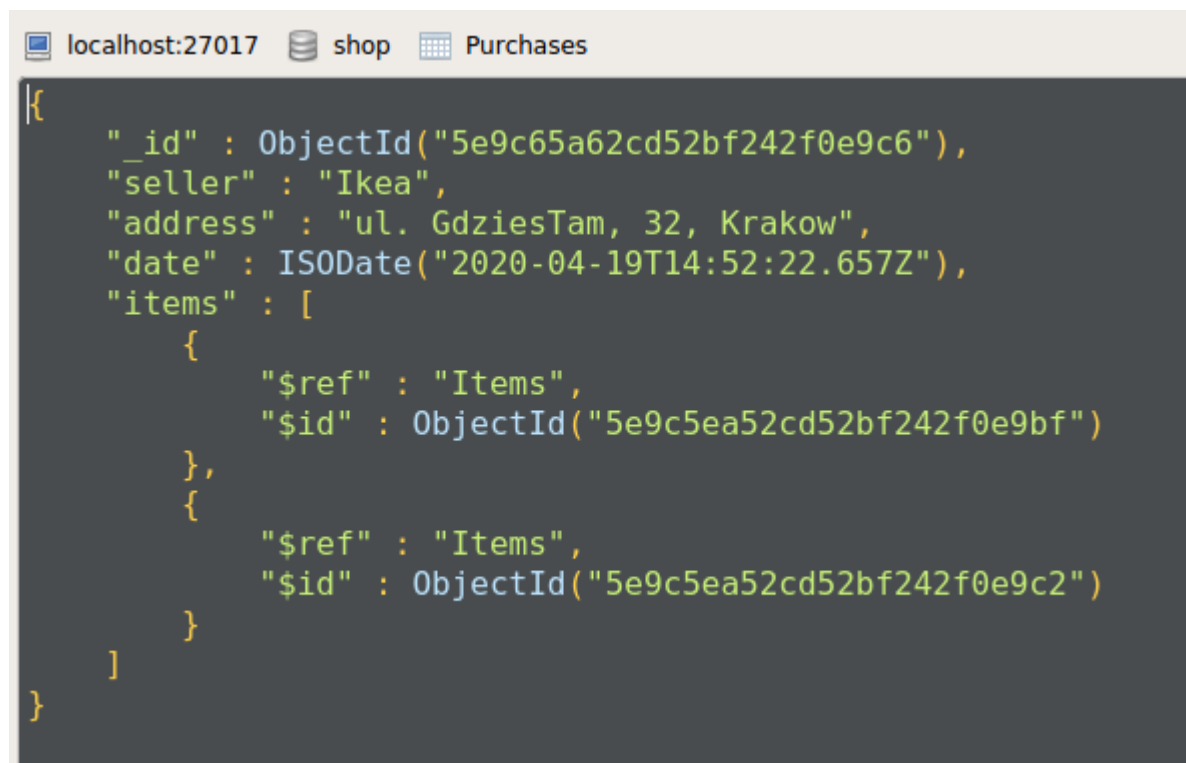


```

localhost:27017  shop  Items
{
  "_id" : ObjectId("5e9c5ea52cd52bf242f0e9bf"),
  "name" : "Biuro",
  "price" : 399.99,
  "category" : "Mebła",
  "origin_country" : "Poland"
}

```

Struktura kolekcji Purchases przedstawiona na danych przykładowych:



```
{
  "_id" : ObjectId("5e9c65a62cd52bf242f0e9c6"),
  "seller" : "Ikea",
  "address" : "ul. GdziesTam, 32, Krakow",
  "date" : ISODate("2020-04-19T14:52:22.657Z"),
  "items" : [
    {
      "$ref" : "Items",
      "$id" : ObjectId("5e9c5ea52cd52bf242f0e9bf")
    },
    {
      "$ref" : "Items",
      "$id" : ObjectId("5e9c5ea52cd52bf242f0e9c2")
    }
  ]
}
```

Przedstawiona powyżej reprezentacja bazy danych wygodna do korzystania.