

Bazy danych – NoSQL MongoDB – zadania

Imię i nazwisko: Błażej Kuśtra

Tydzień B, czwartek 12:50

Zadania:

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.yelpbusiness.distinct("city").sort()
```

```
/* 1 */
[
  "Ahwatukee",
  "Anthem",
  "Apache Junction",
  "Arcadia",
  "Atlanta",
  "Avondale",
  "Black Canyon City",
  "Bonnyrigg",
  "Boulder City",
  "Buckeye",
  "C Las Vegas",
  "Cambridge",
  "Carefree",
```

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

```
db.yelpreview.countDocuments({"date": { $gte: '2011-01-01' }})
```

1.55 sec.

880318

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

```
db.yelpbusiness.find({open: false},{name: 1, full_address: 1, stars: 1, _id: 0})
```

yelpbusiness 0.002 sec.

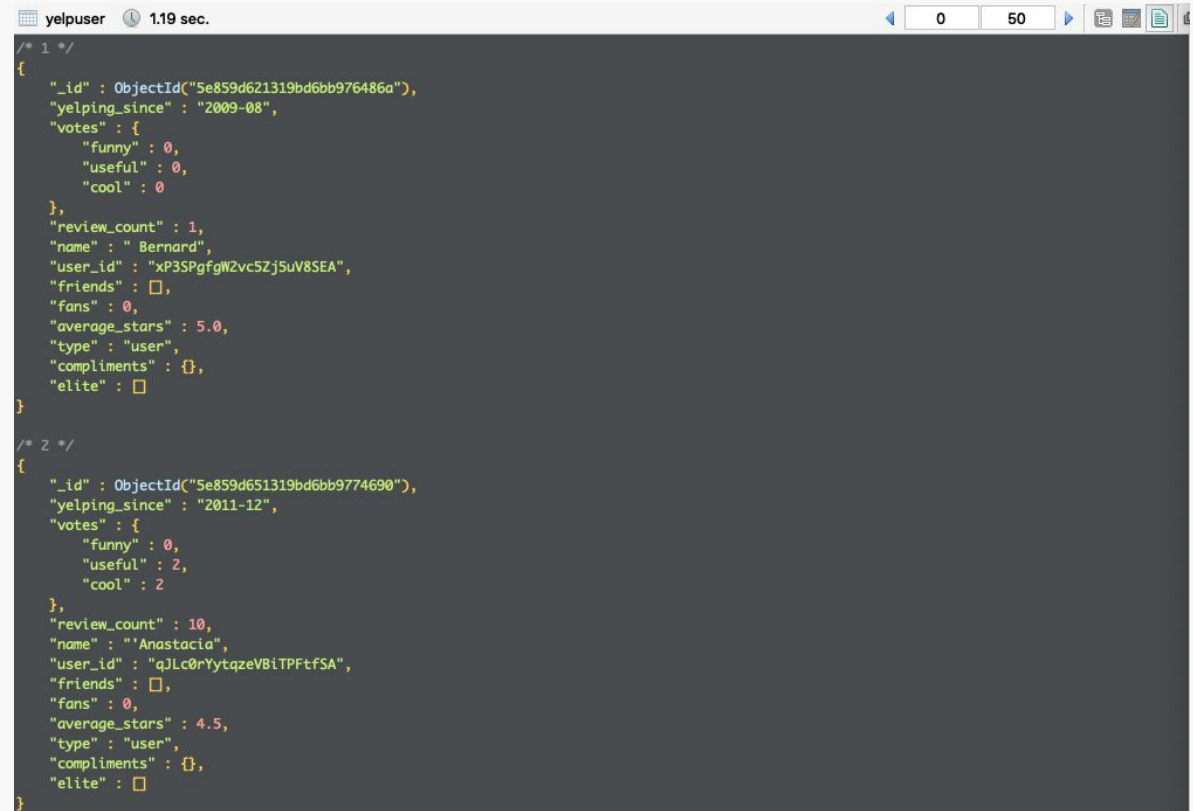
```
{
  "full_address": "4156 County Rd B\nMc Farland, WI 53558",
  "name": "Charter Communications",
  "stars": 1.5
}

/* 2 */
{
  "full_address": "6401 University Ave\nMiddleton, WI 53562",
  "name": "Crandalls Carryout & Catering",
  "stars": 4.0
}

/* 3 */
{
  "full_address": "6230 University Ave\nMiddleton, WI 53562",
  "name": "Mi Cocina",
  "stars": 3.0
}
```

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

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

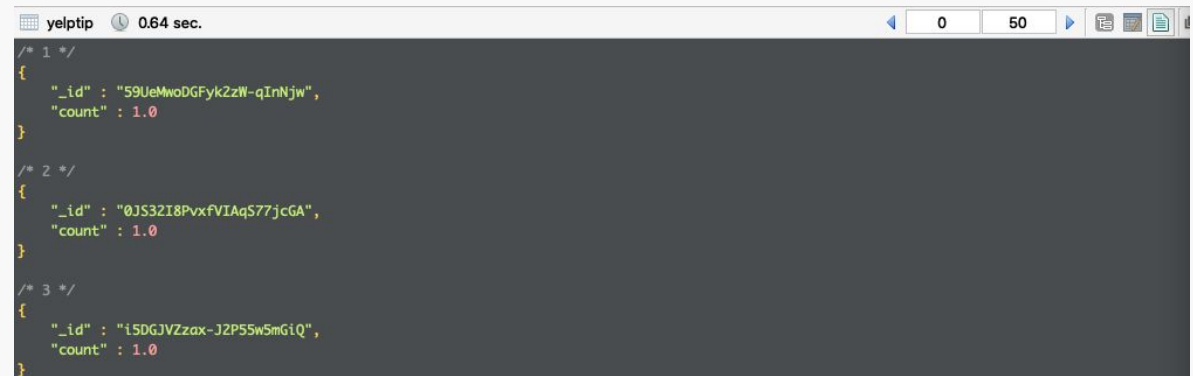


```
/* 1 */
{
  "_id" : ObjectId("5e859d621319bd6bb976486a"),
  "yelping_since" : "2009-08",
  "votes" : {
    "funny" : 0,
    "useful" : 0,
    "cool" : 0
  },
  "review_count" : 1,
  "name" : "Bernard",
  "user_id" : "xP3SPgfgW2vc5ZjSuV8SEA",
  "friends" : [],
  "fans" : 0,
  "average_stars" : 5.0,
  "type" : "user",
  "compliments" : {},
  "elite" : []
}

/* 2 */
{
  "_id" : ObjectId("5e859d651319bd6bb9774690"),
  "yelping_since" : "2011-12",
  "votes" : {
    "funny" : 0,
    "useful" : 2,
    "cool" : 2
  },
  "review_count" : 10,
  "name" : "Anastacia",
  "user_id" : "qJLc0rYytqzeVBtPftfSA",
  "friends" : [],
  "fans" : 0,
  "average_stars" : 4.5,
  "type" : "user",
  "compliments" : {},
  "elite" : []
}
```

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

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



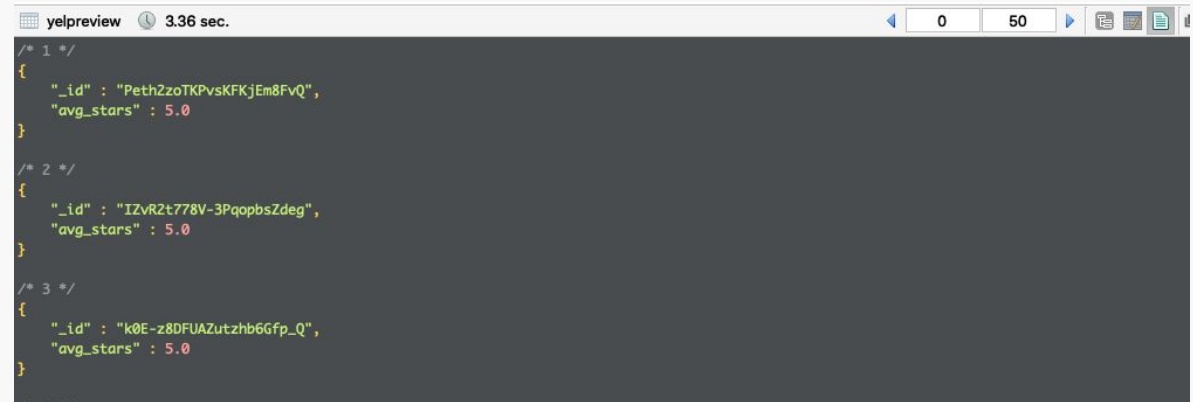
```
/* 1 */
{
  "_id" : "59UeMwoDGfYkZzW-qInNjw",
  "count" : 1.0
}

/* 2 */
{
  "_id" : "0JS32I8PvxvFVIAqS77jcGA",
  "count" : 1.0
}

/* 3 */
{
  "_id" : "i5DGJVZzax-J2P55wSmGiQ",
  "count" : 1.0
}
```

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

```
db.yelpreview.aggregate([
  { $group: { _id: "$business_id", avg_stars: { $avg: "$stars" } } },
  { $match: { avg_stars: { $gte: 4 } } },
  { $sort: { avg_stars: -1 } }
])
```



```
/* 1 */
{
  "_id" : "PethZzoTKPvsKFKjEm8FvQ",
  "avg_stars" : 5.0
}

/* 2 */
{
  "_id" : "IZvRZt778V-3PqopbsZdeg",
  "avg_stars" : 5.0
}

/* 3 */
{
  "_id" : "k0E-z8DFUAZutzhb6Gfp_Q",
  "avg_stars" : 5.0
}
```

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

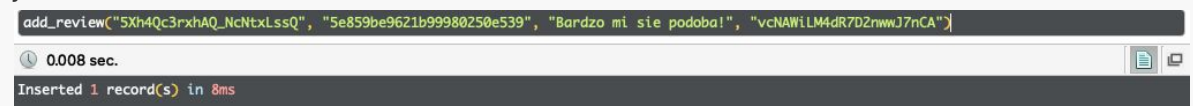
```
db.yelpbusiness.deleteMany({stars:2.0})
```



```
/* 1 */
{
  "acknowledged" : true,
  "deletedCount" : 1576.0
}
```

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

```
function add_review(user_id, review_id, text, business_id) {
  db.yelpreview.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
  })
}
```



```
add_review("5Xh4Qc3rxhAQ_NcNtxLssQ", "5e859be9621b99980250e539", "Bardzo mi sie podoba!", "vcNAWILM4dR7D2nwwJ7nCA")

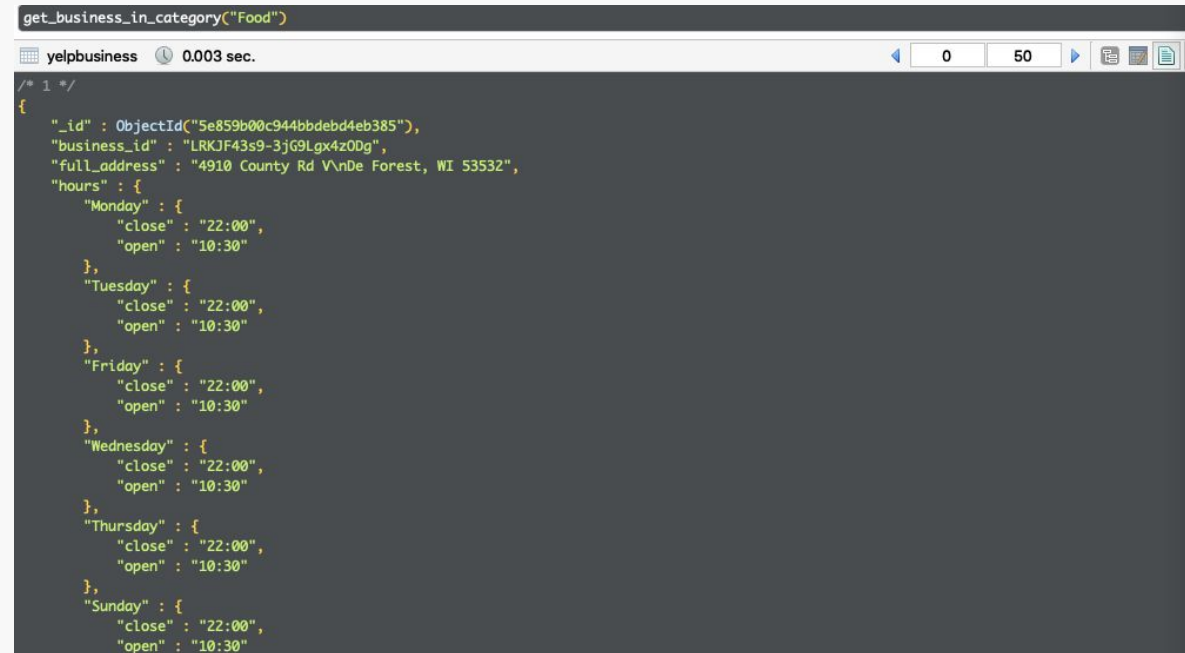
/* 1 */
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5f0000000000000000000000")
}
```

0.008 sec.

Inserted 1 record(s) in 8ms

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 get_business_in_category(category) {  
    return db.yelpbusiness.find({  
        categories: category  
    })  
}
```



```
get_business_in_category("Food")  
  
yelpbusiness 0.003 sec. 0 50  
  
/* 1 */  
{  
  "_id" : ObjectId("5e859b00c944bbdeb4eb385"),  
  "business_id" : "LRKJF43s9-3jG9Lgx4z0Dg",  
  "full_address" : "4910 County Rd WnDe Forest, WI 53532",  
  "hours" : {  
    "Monday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    },  
    "Tuesday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    },  
    "Friday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    },  
    "Wednesday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    },  
    "Thursday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    },  
    "Sunday" : {  
      "close" : "22:00",  
      "open" : "10:30"  
    }  
  }  
}
```

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

```
function modify_user(user_id, new_name) {  
    db.yelpuser.findOneAndUpdate({_id: new ObjectId(user_id) }, {$set: {name: new_name}})  
}
```

```

modify_user("5e859d5f1319bd6bb9758852", "Blazej")
db.yelpuser.find({_id: new ObjectId("5e859d5f1319bd6bb9758852")})

```

yelpuser 0.001 sec.

```

/* 1 */
{
  "_id" : ObjectId("5e859d5f1319bd6bb9758852"),
  "yelping_since" : "2011-08",
  "votes" : {
    "funny" : 2,
    "useful" : 6,
    "cool" : 1
  },
  "review_count" : 6,
  "name" : "Blazej",
  "user_id" : "4dJLZvpYRcjQ6qDR5i7WsA",
  "friends" : [],
  "fans" : 0,
  "average_stars" : 4.33,
  "type" : "user",
  "compliments" : {},
  "elite" : []
}

```

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

```

var map = function() { emit(this.business_id, { count: 1 }) }
var reduce = function(key, values){
  var counter = 0
  values.forEach(function(val){ counter+=val.count })
  return counter
}

```

```

db.yelptip.mapReduce(map,reduce, {out: "number_of_tips" })
db.count.find()

```

```

/* 1 */
{
  "_id" : "--1emggGHgoG6ipd_RMb-g",
  "value" : 6.0
}

/* 2 */
{
  "_id" : "--5jkZ3-nUPZxUvtcbr8Uw",
  "value" : 16.0
}

/* 3 */
{
  "_id" : "--B1vD0_RGZyE1Ku9XA1_g",
  "value" : 21.0
}

/* 4 */
{
  "_id" : "--D12rW_x08GuYBomlg9zw",
  "value" : 2.0
}

```

6. Odwzoruj wszystkie zadania z punktu 1 w Pythonie z pomocą API do MongoDB. Wykorzystaj dla każdego zadania odrębną metodę.

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["bazydanych"]

reviews = mydb["yelpreview"]
business = mydb["yelpbusiness"]
tips = mydb["yelptip"]
users = mydb["yelpuser"]
```

Połączenie pythona z mongodb.

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

```
def get_all_cities():
    return sorted(business.distinct("city"))

['Ahwatukee', 'Anthem', 'Apache Junction', 'Arcadia', 'Atlanta', 'Avondale', 'Black Canyon City', 'Bonnyrigg', 'Boulder City',
' Buckeye', 'C Las Vegas', 'Cambridge', 'Carefree', 'Casa Grande', 'Cave Creek', 'Centennial Hills', 'Central City Village',
' Central Henderson', 'Chandler', 'Chandler-Gilbert', 'City of Edinburgh', 'Clark County', 'Columbus', 'Coolidge', 'Cottage G
rove', 'Cramond', 'Dalkeith', 'Dane', 'De Forest', 'Deforest', 'Eagan', 'Edinburgh', 'El Mirage', 'Enterprise', 'F
itchburg', 'Florence', 'Fort Kinnaird', 'Fort McDowell', 'Fort McDowell', 'Fountain Hills', 'Fountain Hls', 'Gila Bend', 'Gi
lbert', 'Glendale', 'Glendale Az', 'Gold Canyon', 'Goldfield', 'Goodyear', 'Green Valley', 'Guadalupe', 'Heidelberg', 'Hender
son', 'Henderson ', 'Henderson (Green Valley)', 'Henderson (Stephanie)', 'Henderson and Las vegas', 'Henderston', 'Higley',
' Inverkeithing', 'Juniper Green', 'Kitchener', 'Lake Las Vegas', 'Las Vegas', 'Las Vegas ', 'Las Vegas East', 'Las Vegas, NV
89147', 'Las Vegas', 'Lasswade', 'Laveen', 'Leith', 'Litchfield Park', 'Litchfield Park ', 'Loanhead', 'London', 'Madison',
' Maricopa', 'Mc Farland', 'McFarland', 'McFarland', 'Mesa', 'Mesa ', 'Middleton', 'Midlothian', 'Monona', 'Morristown', 'Muss
elburgh', 'N E Las Vegas', 'N Las Vegas', 'N W Las Vegas', 'N. Las Vegas', 'NELLIS AFB', 'Nellis AFB', 'Nellis Afb', 'Nellis
Air Force Base', 'New River', 'New Town', 'New York', 'Newberry Springs', 'Newbridge', 'Newington', 'North Las Vegas', 'North
```

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

```
def count_reviews_after_2011():
    return reviews.count_documents({"date": {"$gte": "2011-01-01"}})
```

```
blazejkustra@bk-3 untitled % python3 program.py
880318
```

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

```
def get_all_closed_companies():
    return business.find({"open": False},
                          {"name": 1, "full_address": 1, "stars": 1, "_id": 0})

{'full_address': '818 N Central Avenue\nPhoenix, AZ 85004', 'name': 'Panorama Wigs', 'stars': 4.0}
{'full_address': '3 S 2nd St\nSte 205\nPhoenix, AZ 85004', 'name': 'Paradise Bakery', 'stars': 4.0}
{'full_address': '1030 NW Grand Ave\nPhoenix, AZ 85007', 'name': 'Paisley Violin', 'stars': 3.5}
{'full_address': '50 N Central Ave\nPhoenix, AZ 85004', 'name': 'Baja Fresh Mexican Grill', 'stars': 3.5}
{'full_address': '455 N 3rd St\nSte 121\nPhoenix, AZ 85004', 'name': 'AT & T Wireless', 'stars': 1.0}
{'full_address': '1 E Jackson St\nPhoenix, AZ 85004', 'name': 'The Brickhouse', 'stars': 3.5}
{'full_address': '333 E Jefferson St\nSte 102\nPhoenix, AZ 85004', 'name': 'Karim's Cobbler Shop & Deli', 'stars': 3.5}
{'full_address': '2601 N Central Ave\nPhoenix, AZ 85004', 'name': 'Schlotzskys Deli', 'stars': 2.5}
{'full_address': '1326 N Central Ave\nPhoenix, AZ 85004', 'name': 'Cheuvront', 'stars': 3.5}
{'full_address': '2022 N 7th St\nPhoenix, AZ 85006', 'name': 'That's A Wrap', 'stars': 4.0}
{'full_address': '1331 N 7th St\nPhoenix, AZ 85006', 'name': 'Center For Spinal Disorders', 'stars': 2.5}
{'full_address': '2201 N 7th St\nPhoenix, AZ 85006', 'name': 'Coronado Cafe', 'stars': 4.0}
{'full_address': '2201 E McDowell Rd\nPhoenix, AZ 85006', 'name': 'Stanley's Homemade Polish Sausage', 'stars': 4.5}
{'full_address': '668 N 44th St\nSte 188\nPhoenix, AZ 85008', 'name': '99 Ranch Market', 'stars': 4.0}
{'full_address': '1515 N 44th St\nPhoenix, AZ 85008', 'name': 'Embassy Suites Hotel', 'stars': 2.5}
{'full_address': '5122 E McDowell Rd\nPhoenix, AZ 85008', 'name': 'Baja Red's Cantina', 'stars': 4.0}
{'full_address': 'Cofo Chinese Cultural Ctr\n668 N 44th St\nPhoenix, AZ 85008', 'name': 'Lao Ching Hing', 'stars': 3.0}
{'full_address': '4420 N Central Ave\nPhoenix, AZ 85012', 'name': 'Music Brokers', 'stars': 4.5}
```

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

```
def get_all_users_with_no_votes():
    return users.find({"$or": [{"votes.funny": 0}, {"votes.useful": 0}]})

{'_id': ObjectId('5e859d6d1319bd6bb9796417'), 'yelping_since': '2013-10', 'votes': {'funny': 1, 'useful': 0, 'cool': 0},
'review_count': 1, 'name': 'Anamika', 'user_id': 'CYKWssZpMJYE6uXzydKGUA', 'friends': [], 'fans': 0, 'average_stars': 5.0,
'type': 'user', 'compliments': {}, 'elite': []}
{'_id': ObjectId('5e859d6d1319bd6bb979641a'), 'yelping_since': '2013-07', 'votes': {'funny': 0, 'useful': 3, 'cool': 0},
'review_count': 3, 'name': 'Jeanine', 'user_id': 'UEoCuejDwMmK4KyF88Bew', 'friends': [], 'fans': 0, 'average_stars': 4.3,
'type': 'user', 'compliments': {}, 'elite': []}
{'_id': ObjectId('5e859d6d1319bd6bb979641b'), 'yelping_since': '2009-12', 'votes': {'funny': 0, 'useful': 0, 'cool': 0},
'review_count': 1, 'name': 'Ali', 'user_id': 'Sn_bI9-2uz2zAsSSbdfgXw', 'friends': [], 'fans': 0, 'average_stars': 5.0, 't
ype': 'user', 'compliments': {}, 'elite': []}
{'_id': ObjectId('5e859d6d1319bd6bb9796421'), 'yelping_since': '2010-11', 'votes': {'funny': 0, 'useful': 1, 'cool': 0},
'review_count': 3, 'name': 'Allister', 'user_id': 'ZRzhH3_33EkimjsWYHRNw', 'friends': [], 'fans': 0, 'average_stars': 4.
0, 'type': 'user', 'compliments': {}, 'elite': []}
{'_id': ObjectId('5e859d6d1319bd6bb9796423'), 'yelping_since': '2011-02', 'votes': {'funny': 0, 'useful': 2, 'cool': 1},
'review_count': 6, 'name': 'Holly', 'user_id': 'dTRxrgIcYz1vbbb_P_rv7g', 'friends': ['32_9e5jRUayEINGoRARAATQ'], 'fans': 1
, 'average_stars': 4.17, 'type': 'user', 'compliments': {'more': 1}, 'elite': []}
{'_id': ObjectId('5e859d6d1319bd6bb9796424'), 'yelping_since': '2012-02', 'votes': {'funny': 0, 'useful': 0, 'cool': 0},
'review_count': 1, 'name': 'Derik', 'user_id': 'CNUBWijlMihjN_2ipIauBg', 'friends': [], 'fans': 0, 'average_stars': 5.0,
'type': 'user', 'compliments': {}, 'elite': []}
```

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

```
def list_business_with_their_tips():
    return tips.aggregate([
        {"$match": {"date": {"$regex": "2012"} }},
        {"$group": {"_id": "$business_id", "count": {"$sum": 1}}},
        {"$sort": {"count": 1}}
    ])

{'_id': 'LbVBhtk7yMVgDAuoiknBFg', 'count': 1}
{'_id': '5EeGGEFbc2KlrBYhL80ciA', 'count': 1}
{'_id': 'MlRBLmH9JwUMnJ8yvTpc0Q', 'count': 1}
{'_id': 'dHXoP7bZYJAV005-uAQadg', 'count': 1}
{'_id': 'gxuVySgACHDqJlwmelFHLA', 'count': 1}
{'_id': 'PtJdFzFitBa6rXQWx3v7_A', 'count': 1}
{'_id': '0HDkCZtCvua-Wahio1-pBw', 'count': 1}
{'_id': '_zBGLuqsrqpTGT09M1Q19Q', 'count': 1}
{'_id': 'U-tiHSh0P38H-6-yDUHrZA', 'count': 1}
{'_id': '274DgOpS524g_tzcfaKYog', 'count': 1}
{'_id': 'aPu5wZKYyro4Jhw0JJYAgQ', 'count': 1}
{'_id': 'M7AMdpRP1r0o15JMYeg3dg', 'count': 1}
{'_id': 'ZVmZaTHC5bB1A9ihtBDlww', 'count': 1}
{'_id': '4XuQ-oAsPQo5xwh-DPZovg', 'count': 1}
{'_id': 'JtmzJl8FXNTBbWcXwroZuw', 'count': 1}
{'_id': 'apGVTRZRCQ9-89hu2qW-vw', 'count': 1}
{'_id': 'RkZIVN-Wyii8he3o07GdgQ', 'count': 1}
{'_id': 'bNmsfLXaKKcDigEMidEcGg', 'count': 1}
```

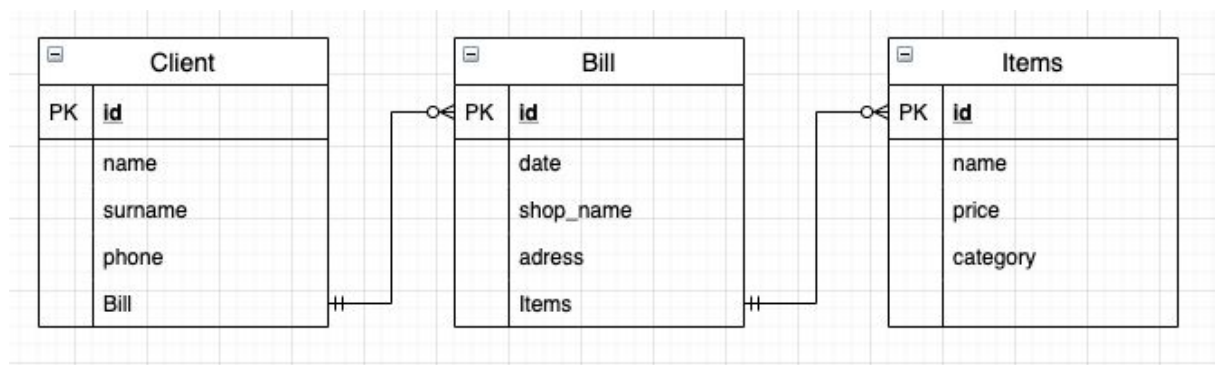

f. Wyznacz, jaką średnią ocen (*stars*) uzyskała każda firma (*business*) na podstawie wszystkich recenzji. Wynik ogranicz do recenzji, które uzyskały min 4.0 gwiazdki.

```
def average_rate():
    return reviews.aggregate([
        { "$group": { "_id": "$business_id", "avg_stars": { "$avg": "$stars" } } },
        { "$match": { "avg_stars": { "$gte": 4 } } },
        { "$sort": { "avg_stars": -1 } }
    ])
```

```
{ '_id': 'gWM0GN05r-KUJapfuCYzA', 'avg_stars': 4.0 }
{ '_id': 'nQgcnd4NoNe9IJqAuc0nUA', 'avg_stars': 4.0 }
{ '_id': 'fo7GgOSKdizF0gNn1MYjRg', 'avg_stars': 4.0 }
{ '_id': 'C6nZkmtRpUQ2eXZ1Jkb1GA', 'avg_stars': 4.0 }
{ '_id': 'KPFw26qAxJlYMyAZ8bb3Lg', 'avg_stars': 4.0 }
{ '_id': '23CYkAUCRd4JbTrvRR4I-Q', 'avg_stars': 4.0 }
{ '_id': 'nx_mbm1L1GRHmzQ0wYdIxg', 'avg_stars': 4.0 }
{ '_id': 'j1BASG5Kvt6dqfjCmtjtuA', 'avg_stars': 4.0 }
{ '_id': 'OzCdLOH89sT9Q776JCuIYg', 'avg_stars': 4.0 }
{ '_id': 'mMFw9cZU1gQYv1uWREzg_w', 'avg_stars': 4.0 }
{ '_id': 'BrIHPtnJC6gXDEbnTuEupA', 'avg_stars': 4.0 }
{ '_id': 'BN036HERQ7id8OMXJZYVvQ', 'avg_stars': 4.0 }
{ '_id': 'HELT9itN1_A23pKxAZ2UAQ', 'avg_stars': 4.0 }
{ '_id': 'fib0PyPJ2yJ07i8TLnuqXA', 'avg_stars': 4.0 }
{ '_id': 'cgNuDdc1_zo8p0LkrSVFUG', 'avg_stars': 4.0 }
{ '_id': '475vsfEBdBYL_pSKgRoBMw', 'avg_stars': 4.0 }
{ '_id': 'SvfSwVswCDo1tRzkgghREpw', 'avg_stars': 4.0 }
{ '_id': 'KUXjxPnSQ-5r23IfMzs3ow', 'avg_stars': 4.0 }
```

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

a. Schemat:



b. Uzasadnienie:

Klient posiada swoje dane osobowe oraz numer telefonu wraz z wszystkimi rachunkami, rachunki posiadają datę zakupów, nazwa sklepu, adres sklepu oraz listę przedmiotów. Każdy przedmiot posiada nazwę, cenę i kategorie. Struktura jest prosta, oraz brak w niej redundancji.

c. Client:

```
function add_client(name,surname,phone){
  db.clients.insert({
    name: name,
    surname: surname,
    phone: phone,
    bill: []
  })
}

add_client("Blazej","Kustra", 734467906);
add_client("Marcin","Zielony", 123456789);
add_client("Mateusz","Monada", 987654321);
```

d. Bill:

```
function add_bill(shop_name,shop_address){
  db.bill.insert({
    date: new Date(),
    shop_name: shop_name,
    shop_address: shop_address,
    items: []
  })
}

add_bill("Żabka","Myczkowskiego 21");
add_bill("Auchan","Merna 99");
```

e. Items:

```
function add_item(name,price,category){
  db.items.insert({
    name: name,
    price: price,
    category: category
  })
}

add_item("Woda",2.0,"Napoje");
add_item("Czekolada",4.99,"Słodyczne");
add_item("Sok",3.19,"Napoje");
add_item("Szpinak",5.99,"Zdrowe");
add_item("Chleb",4.0,"Pieczywo");
```

f. Dodanie przedmiotu do rachunku:

```
function add_to_cart(bill_id,item_id){
  db.bill.update(
    { _id: new ObjectId(bill_id)},
    {$addToSet: { items: {$ref: "item", $id: new ObjectId(item_id)}}}
  );
}

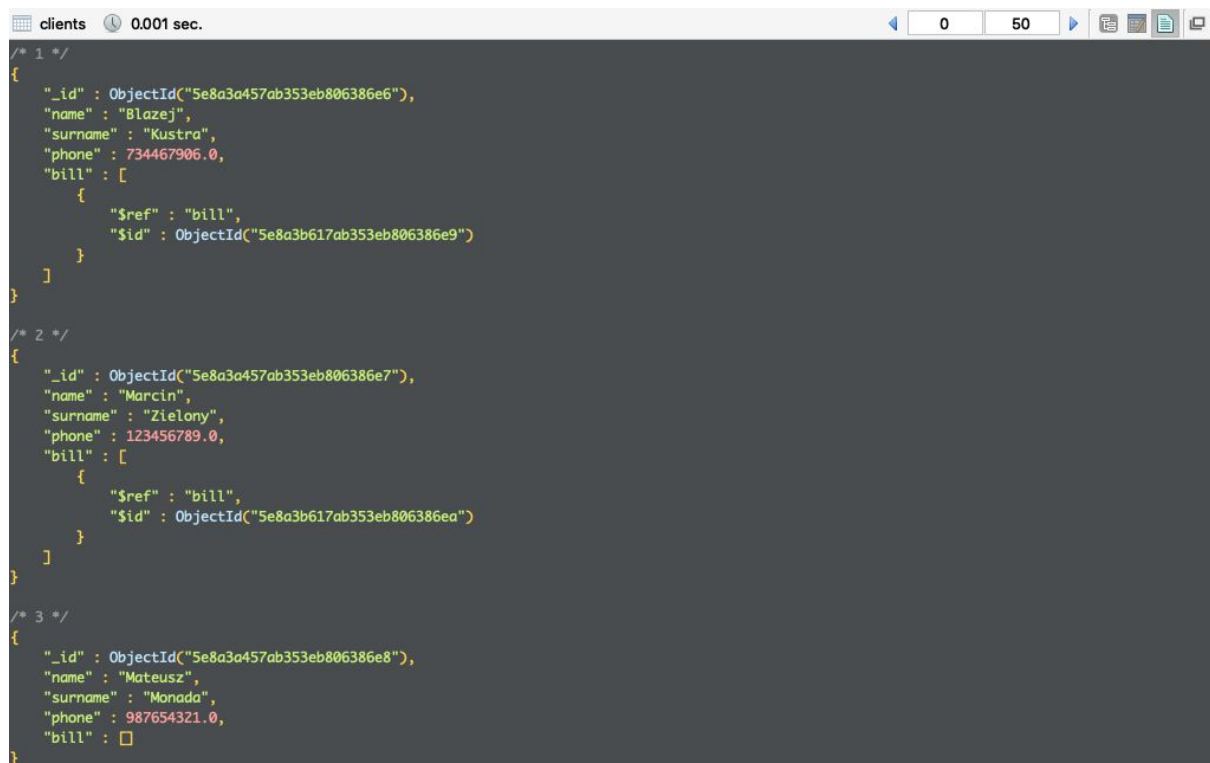
add_to_cart("5e8a3b617ab353eb806386e9","5e8a3c5e7ab353eb806386f0");
add_to_cart("5e8a3b617ab353eb806386ea","5e8a3c5e7ab353eb806386f2");
add_to_cart("5e8a3b617ab353eb806386ea","5e8a3c5e7ab353eb806386f4");
```

g. Dodanie rachunku do klienta:

```
function add_cart_to_client(client_id, bill_id){
  db.clients.update(
    { _id: new ObjectId(client_id)},
    {$addToSet: { bill: {$ref: "bill", $id: new ObjectId(bill_id)}}}
  );
}

add_cart_to_client("5e8a3a457ab353eb806386e6","5e8a3b617ab353eb806386e9");
add_cart_to_client("5e8a3a457ab353eb806386e7","5e8a3b617ab353eb806386ea");
```

h. Baza danych:



```
clients 0.001 sec.
0 50

/* 1 */
{
  "_id" : ObjectId("5e8a3a457ab353eb806386e6"),
  "name" : "Blażej",
  "surname" : "Kustra",
  "phone" : 734467906.0,
  "bill" : [
    {
      "$ref" : "bill",
      "$id" : ObjectId("5e8a3b617ab353eb806386e9")
    }
  ]
}

/* 2 */
{
  "_id" : ObjectId("5e8a3a457ab353eb806386e7"),
  "name" : "Marcin",
  "surname" : "Zielony",
  "phone" : 123456789.0,
  "bill" : [
    {
      "$ref" : "bill",
      "$id" : ObjectId("5e8a3b617ab353eb806386ea")
    }
  ]
}

/* 3 */
{
  "_id" : ObjectId("5e8a3a457ab353eb806386e8"),
  "name" : "Mateusz",
  "surname" : "Monada",
  "phone" : 987654321.0,
  "bill" : []
}
```

```
bill 0.005 sec.
/* 1 */
{
  "_id" : ObjectId("5e8a3b617ab353eb806386e9"),
  "date" : ISODate("2020-04-05T20:11:13.928Z"),
  "shop_name" : "Żabka",
  "shop_address" : "Myczkowskiego 21",
  "items" : [
    {
      "$ref" : "item",
      "$id" : ObjectId("5e8a3c5e7ab353eb806386f0")
    }
  ]
}

/* 2 */
{
  "_id" : ObjectId("5e8a3b617ab353eb806386ea"),
  "date" : ISODate("2020-04-05T20:11:13.994Z"),
  "shop_name" : "Auchan",
  "shop_address" : "Merna 99",
  "items" : [
    {
      "$ref" : "item",
      "$id" : ObjectId("5e8a3c5e7ab353eb806386f2")
    },
    {
      "$ref" : "item",
      "$id" : ObjectId("5e8a3c5e7ab353eb806386f4")
    }
  ]
}
```

```
items 0.003 sec.
/* 1 */
{
  "_id" : ObjectId("5e8a3c5e7ab353eb806386f0"),
  "name" : "Woda",
  "price" : 2.0,
  "category" : "Napoje"
}

/* 2 */
{
  "_id" : ObjectId("5e8a3c5e7ab353eb806386f1"),
  "name" : "Czekolada",
  "price" : 4.99,
  "category" : "Słodycze"
}

/* 3 */
{
  "_id" : ObjectId("5e8a3c5e7ab353eb806386f2"),
  "name" : "Sok",
  "price" : 3.19,
  "category" : "Napoje"
}

/* 4 */
{
  "_id" : ObjectId("5e8a3c5e7ab353eb806386f3"),
  "name" : "Szpinak",
  "price" : 5.99,
  "category" : "Zdrowe"
}

/* 5 */
{
  "_id" : ObjectId("5e8a3c5e7ab353eb806386f4"),
  "name" : "Chleb",
  "price" : 4.0,
  "category" : "Pieczywo"
}
```