

Bartosz Szar

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

```
/* 1 */  
[  
    "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",
```

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

```
db.review.find({date: {$gte: '2011-01-01'}}).count()
```

```
880318
```

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

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

```
/* 1 */
{
  "_id" : ObjectId("5e822c2597ce0b5a8ceda4a0"),
  "full_address" : "4156 County Rd B\nMc Farland, WI 53558",
  "name" : "Charter Communications",
  "stars" : 1.5
}

/* 2 */
{
  "_id" : ObjectId("5e822c2597ce0b5a8ceda4a5"),
  "full_address" : "6401 University Ave\nMiddleton, WI 53562",
  "name" : "Crandalls Carryout & Catering",
  "stars" : 4.0
}

/* 3 */
{
  "_id" : ObjectId("5e822c2597ce0b5a8ceda4b2"),
  "full_address" : "6230 University Ave\nMiddleton, WI 53562",
  "name" : "Mi Cocina",
  "stars" : 3.0
}
```

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.

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

```
/* 1 */  
{  
  "_id" : ObjectId("5e822d704ba7d565f2c7b8a9"),  
  "yelping_since" : "2009-08",  
  "votes" : {  
    "funny" : 0,  
    "useful" : 0,  
    "cool" : 0  
  },  
  "review_count" : 1,  
  "name" : " Bernard",  
  "user_id" : "xP3SPgfgW2vc5Zj5uV8SEA",  
  "friends" : [],  
  "fans" : 0,  
  "average_stars" : 5.0,  
  "type" : "user",  
  "compliments" : {},  
  "elite" : []  
}  
  
/* 2 */  
{  
  "_id" : ObjectId("5e822d744ba7d565f2c8b6d0"),  
  "yelping_since" : "2011-12",  
  "votes" : {  
    "funny" : 0,  
    "useful" : 2,  
    "cool" : 2  
  },  
  "review_count" : 10,  
  "name" : "'Anastacia",  
  "user_id" : "qJLc0rYytqzeVBiTpfTfSA",  
  "friends" : [],  
  "fans" : 0,  
  "average_stars" : 4.5,  
  "type" : "user",  
  "compliments" : {},  
  "elite" : []  
}
```

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

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

```
/* 1 */  
{  
  "_id" : "VImQLS1lPys8n-EqYVyhMQ",  
  "count" : 1.0  
}  
  
/* 2 */  
{  
  "_id" : "CNTijY6PVNFe69Gs7jBddQ",  
  "count" : 1.0  
}  
  
/* 3 */  
{  
  "_id" : "VtwMVVgeAOCosAO_I19WYg",  
  "count" : 1.0  
}
```

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.

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

```
/* 1 */  
{  
  "_id" : "LRKJF43s9-3jG9Lgx4zODg",  
  "avg_rating" : 4.375  
}  
  
/* 2 */  
{  
  "_id" : "Hxc0MC5GUTQg13I029S7Ow",  
  "avg_rating" : 5.0  
}  
  
/* 3 */  
{  
  "_id" : "KZlgESEeTmNeKaPAbTY51Q",  
  "avg_rating" : 5.0  
}
```

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

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

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

Zadanie 2

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

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

```
addReview('userId', 5, 'wonderful place!', 'businessId')
```

```
Inserted 1 record(s) in 14ms
```

```
/* 1 */
{
  "_id" : ObjectId("5e91842f36ab4f3b7bc2745a"),
  "votes" : {
    "funny" : 0.0,
    "useful" : 0.0,
    "cool" : 0.0
  },
  "user_id" : "userId",
  "review_id" : ObjectId("5e91842f36ab4f3b7bc27459"),
  "stars" : 5.0,
  "date" : ISODate("2020-04-11T08:47:43.013Z"),
  "text" : "wonderful place!",
  "type" : "review",
  "business_id" : "businessId"
}
```

Zadanie 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 findBusiness(category){
  return db.business.find({
    categories: category
  })
}
```

```
findBusiness('Hotels & Travel')
```

```
/* 1 */
{
  "_id" : ObjectId("5e822c2597ce0b5a8ceda49e"),
  "business_id" : "XrzTVrJAjs0yYLvhFI9vkQ",
  "full_address" : "5508 Broadhead St\nMc Farland, WI 53558",
  "hours" : {},
  "open" : true,
  "categories" : [
    "Hotels & Travel",
    "Bed & Breakfast",
    "Event Planning & Services",
    "Hotels"
  ],
  "city" : "Mc Farland",
  "review_count" : 9,
  "name" : "Parsonage Bed & Breakfast the",
  "neighborhoods" : [],
  "longitude" : -89.285206,
  "state" : "WI",
  "stars" : 5.0,
  "latitude" : 43.015931,
  "attributes" : {
    "Accepts Credit Cards" : true,
    "Wi-Fi" : "free",
    "Price Range" : 2
  },
  "type" : "business"
}
```

Zadanie 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 , name){  
  db.user .update(  
    {user_id : user_id},  
    {$set: {name: name}}  
  )  
}
```

```
changeUsername('TUf5iMrBPXb13sunWMdyeQ', 'Jerzy Kiler')
```

```
Updated 1 existing record(s) in 2ms
```

```
db.user .find({user_id : 'TUf5iMrBPXb13sunWMdyeQ'})
```

```
/* 1 */  
{  
  "_id" : ObjectId("5e822d6d4ba7d565f2c6f8a5"),  
  "yelping_since" : "2013-01",  
  "votes" : {  
    "funny" : 0,  
    "useful" : 0,  
    "cool" : 0  
  },  
  "review_count" : 5,  
  "name" : "Jerzy Kiler",  
  "user_id" : "TUf5iMrBPXb13sunWMdyeQ",  
  "friends" : [  
    "uLPY0c3jxd6vi6fUxbv4UA",  
    "uSYAjQws1Ot2ZcOObMOxGA",  
    "pPB99MmfTARB4s6_XB5vzA"  
  ],  
  "fans" : 0,  
  "average_stars" : 4.4,  
  "type" : "user",  
  "compliments" : {  
    "plain" : 1  
  },  
  "elite" : []  
}
```

Zadanie 5

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

```
db.tip.mapReduce(  
  function() {emit(this.business_id, 1);},  
  function(key, values) {return Array.sum(values);},  
  {  
    out: 'tips_per_business'  
  }  
);  
  
db.tips_per_business.aggregate([  
  {$group:  
    {  
      _id: '',  
      avg_tips_per_business: {$avg: '$value'}  
    }  
  }  
]);
```

```
/* 1 */  
{  
  "result" : "tips_per_business",  
  "timeMillis" : 3501.0,  
  "counts" : {  
    "input" : 403210,  
    "emit" : 403210,  
    "reduce" : 23618,  
    "output" : 29909  
  },  
}
```

```
/* 1 */  
{  
  "_id" : "",  
  "avg_tips_per_business" : 13.4812263867063  
}
```


Zadanie 6

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

```
protected void zad_6_1a(){
    ArrayList<String> cities = db.getCollection("business")
        .distinct("city", String.class).into(new ArrayList<>());
    Collections.sort(cities);

    for(String city: cities)
        System.out.println(city);
}
```

```
Ahwatukee
Anthem
Apache Junction
Arcadia
Atlanta
Avondale
Black Canyon City
Bonnyrigg
Boulder City
Buckeye
C Las Vegas
Cambridge
Carefree
Casa Grande
```

```
protected void zad_6_1b() {
    MongoCollection<Document> review = db.getCollection("review");
    System.out.println(review.countDocuments(Filters.gte("date",
        "2011-01-01")));
}
```

880318

```
protected void zad_6_1c() {
    ArrayList<Document> business = db.getCollection("business")
        .find(new Document("open", false))
        .projection(fields(include("name", "full_address",
            "stars"), excludeId()))
        .into(new ArrayList<Document>());

    for(Document document: business)
        System.out.println(document.toJson());
}
```

```
{"full_address": "4568 E Cactus Rd\nPhoenix, AZ 85032", "name": "Persian King", "stars": 3.5}
{"full_address": "2606 S Durango Dr\nSte 213\nWestside\nLas Vegas, NV 89117", "name": "Premier Party Tours", "stars": 5.0}
{"full_address": "15745 N Hayden Road\nSuite D2\nScottsdale, AZ 85260", "name": "Bedmart", "stars": 4.0}
{"full_address": "7380 S Eastern Ave\nSte 109C\nSoutheast\nLas Vegas, NV 89123", "name": "Sin City Spray Tans", "stars": 5.0}
{"full_address": "12645 N Saguro Blvd\nFountain Hills, AZ 85268", "name": "Smokey's BBQ", "stars": 3.0}
{"full_address": "119 Rose Street\nNew Town\nNew Town EH2 3DT", "name": "El Barrio", "stars": 3.5}
{"full_address": "2185 E Windmill Ln\nSte 200\nSoutheast\nLas Vegas, NV 89123", "name": "Vid Noodles", "stars": 4.0}
{"full_address": "10430 S Eastern Ave\nAnthem\nHenderson, NV 89052", "name": "Carmines", "stars": 3.0}
{"full_address": "3000 W Ann Rd\nSte 109\nNorth Las Vegas, NV 89031", "name": "Super No1 Chinese Restaurant", "stars": 3.5}
{"full_address": "13014 N Saguro Blvd\nFountain Hills, AZ 85268", "name": "Moxie Modern Fare", "stars": 3.5}
```

```
protected void zad_6_1d() {
    ArrayList<Document> users = db.getCollection("user")
        .find(or(eq("votes.funny", 0), eq("votes.useful", 0)))
        .sort(ascending("name"))
        .into(new ArrayList<>());

    for(Document document: users)
        System.out.println(document.toJson());
}
```

```
{"_id": {"$oid": "5e822d7c4ba7d565f2cab20c"}, "yelping_since": "2011-10", "votes": {"funny": 0, "useful": 1, "cool": 0}, "review_count": 1,
{"_id": {"$oid": "5e822d6f4ba7d565f2c77dc9"}, "yelping_since": "2009-03", "votes": {"funny": 0, "useful": 1, "cool": 1}, "review_count": 3,
{"_id": {"$oid": "5e822d7b4ba7d565f2ca351c"}, "yelping_since": "2007-11", "votes": {"funny": 0, "useful": 1, "cool": 0}, "review_count": 3,
{"_id": {"$oid": "5e822d6f4ba7d565f2c76e22"}, "yelping_since": "2008-03", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 2,
{"_id": {"$oid": "5e822d704ba7d565f2c7c692"}, "yelping_since": "2011-01", "votes": {"funny": 0, "useful": 3, "cool": 0}, "review_count": 5,
{"_id": {"$oid": "5e822d704ba7d565f2c789d9"}, "yelping_since": "2010-06", "votes": {"funny": 0, "useful": 1, "cool": 0}, "review_count": 1,
{"_id": {"$oid": "5e822d7b4ba7d565f2ca6d94"}, "yelping_since": "2011-03", "votes": {"funny": 0, "useful": 5, "cool": 0}, "review_count": 13,
{"_id": {"$oid": "5e822d6d4ba7d565f2c6fdcf"}, "yelping_since": "2010-10", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 3,
{"_id": {"$oid": "5e822d7c4ba7d565f2ca9a08"}, "yelping_since": "2011-08", "votes": {"funny": 0, "useful": 7, "cool": 1}, "review_count": 22,
{"_id": {"$oid": "5e822d784ba7d565f2c9ae6d"}, "yelping_since": "2010-03", "votes": {"funny": 0, "useful": 6, "cool": 0}, "review_count": 5,
{"_id": {"$oid": "5e822d754ba7d565f2c8f96e"}, "yelping_since": "2011-10", "votes": {"funny": 0, "useful": 2, "cool": 0}, "review_count": 2,
{"_id": {"$oid": "5e822d754ba7d565f2c8deca"}, "yelping_since": "2010-07", "votes": {"funny": 0, "useful": 2, "cool": 0}, "review_count": 3,
{"_id": {"$oid": "5e822d754ba7d565f2c8dcc3"}, "yelping_since": "2011-10", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 3,
{"_id": {"$oid": "5e822d734ba7d565f2c885ab"}, "yelping_since": "2014-01", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 1,
{"_id": {"$oid": "5e822d744ba7d565f2c88aa5"}, "yelping_since": "2009-07", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 1,
{"_id": {"$oid": "5e822d764ba7d565f2c90ba5"}, "yelping_since": "2014-07", "votes": {"funny": 0, "useful": 0, "cool": 0}, "review_count": 1,
```

```
protected void zad_6_1e() {
    ArrayList<Document> tips = db.getCollection("tip")
        .aggregate(Arrays.asList(
            Aggregates.group("$business_id",
                Accumulators.sum("tips", 1))
        )).into(new ArrayList<>());

    for(Document document: tips)
        System.out.println(document.toJson());
}
```

```
{ "_id": "ZNb84wf_9xDQtOvM5UKxEQ", "tips": 17 }
{ "_id": "8PNgikkR6rEBB2Mms5qXJA", "tips": 3 }
{ "_id": "3jq0v6re-xPY0g7srmi7tg", "tips": 35 }
{ "_id": "-5RN56jH78MV2oquLV_G8g", "tips": 34 }
{ "_id": "n6oNY8L8iEWkRksJkmwCqg", "tips": 87 }
{ "_id": "21SLVH51ShKUIEEVsXJwyg", "tips": 5 }
{ "_id": "xx37S9SerTIaSZLKfpZZjg", "tips": 7 }
{ "_id": "uluM1I_Pf_d9egCBXH4oqQ", "tips": 1 }
{ "_id": "QhEJX1nZZnCBM_Y3BnB9zw", "tips": 5 }
{ "_id": "dl5K9Hc2kUcGcrVIQAOhwQ", "tips": 29 }
{ "_id": "6JaonzYu24vAR8xe3bKe6w", "tips": 7 }
```

```
protected void zad_6_1f() {
    ArrayList<Document> reviews = db.getCollection("review")
        .aggregate(Arrays.asList(
            Aggregates.group("$business_id",
                Accumulators.avg("avg_rating", "$stars")),
            Aggregates.match(Filters.gte("avg_rating", 4.0))
        )).into(new ArrayList<>());

    for(Document document: reviews)
        System.out.println(document.toJson());
}
```

```
{ "_id": "9QE1yP6sWlmwBvpolg5zItw", "avg_rating": 5.0 }
{ "_id": "yRhDoKG2calmtU_7MAt90g", "avg_rating": 4.4 }
{ "_id": "qdDLtRi-OiZZKKSxyz_fzQ", "avg_rating": 4.7 }
{ "_id": "uDJNCyWuTl3RCfGxJtEo1g", "avg_rating": 5.0 }
{ "_id": "L-SpHlHt_id-U28gFbnhKA", "avg_rating": 4.0 }
{ "_id": "RZS7pdeB3xOyHwYgT7VYGw", "avg_rating": 4.25 }
```

```
protected void zad_6_1g() {
    db.getCollection("business").deleteMany(eq("stars", 2.0));
}
```

BartoszSzar_wt1115_B localhost:27017 BartoszSzar_wt1115_B

```
db.business.find({stars: {$eq: 2}})
```

0.023 sec.

Fetches 0 record(s) in 23ms

Zadanie 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.

Struktura bazy:

Każdy klient posiada tablicę z paragonami będącą zbiorem wszystkich dokonanych przez niego zakupów w naszym sklepie. Na podobnej zasadzie każdy paragon posiada tablicę przypisanych do niego produktów, które zostały naliczone na dany paragon.

Schemat:



Kolekcja customers:

```
function addCustomer(firstname, secondname){
  db.customers.insert({
    firstname: firstname,
    secondname: secondname,
    receipts: []
  })
}
```

```
addCustomer('Jerzy', 'Kiler')
addCustomer('Ferdynand', 'Lipski')
addCustomer('Stefan', 'Siarzewski')
```

Kolekcja receipts:

```
function addReceipt(){
  db.receipts.insert({
    date: new Date(),
    products: []
  })
}
```

```
addReceipt()
addReceipt()
addReceipt()
```

Kolekcja products:

```
function addProduct(name, price, category){
  db.products.insert({
    name: name,
    price: price,
    category: category
  })
}
```

```
addProduct('pizza kiler', 24.99, 'food')
addProduct('kontener brudnych pieniedzy', 2000000, 'just like that')
addProduct('zurek', 420.69, 'sheraton')
```

Dodanie produktu do paragonu:

```
function addProductToReceipt(receiptId, productId){
  db.receipts.update(
    {_id: new ObjectId(receiptId)},
    {$addToSet: { products: {$ref: 'products',
                           $id: new ObjectId(productId)}}}
  )
}
```

```
addProductToReceipt('5e91a9370d2d1f50d3c45c81',
'5e91ab200d2d1f50d3c45c84')
addProductToReceipt('5e91a9370d2d1f50d3c45c82',
'5e91ab200d2d1f50d3c45c85')
addProductToReceipt('5e91a9370d2d1f50d3c45c83',
'5e91ab200d2d1f50d3c45c86')
```

Przypisanie paragonu klientowi:

```
function addReceiptToCustomer(customerId, receiptId){
  db.customers.update(
    {_id: new ObjectId(customerId)},
    {$addToSet: {receipts: {$ref: 'receipts',
                           $id: new ObjectId(receiptId)}}}
  )
}
```

```
addReceiptToCustomer('5e91a72236ab4f3b7bc2745b',
'5e91a9370d2d1f50d3c45c81')
addReceiptToCustomer('5e91a7d336ab4f3b7bc2745c',
'5e91a9370d2d1f50d3c45c82')
addReceiptToCustomer('5e91a7d336ab4f3b7bc2745d',
'5e91a9370d2d1f50d3c45c83')
```

Kolekcje w bazie danych:

```

BartoszSzar_wt1115_B  localhost:27017  BartoszSzar_wt1115_B
db.customers.find()

customers  0.001 sec.

/* 1 */
{
  "_id" : ObjectId("5e91a72236ab4f3b7bc2745b"),
  "firstname" : "Jerzy",
  "secondname" : "Kiler",
  "receipts" : [
    {
      "$ref" : "receipts",
      "$id" : ObjectId("5e91a9370d2d1f50d3c45c81")
    }
  ]
}

/* 2 */
{
  "_id" : ObjectId("5e91a7d336ab4f3b7bc2745c"),
  "firstname" : "Ferdynand",
  "secondname" : "Lipski",
  "receipts" : [
    {
      "$ref" : "receipts",
      "$id" : ObjectId("5e91a9370d2d1f50d3c45c82")
    }
  ]
}

/* 3 */
{
  "_id" : ObjectId("5e91a7d336ab4f3b7bc2745d"),
  "firstname" : "Stefan",
  "secondname" : "Siarzewski",
  "receipts" : [
    {
      "$ref" : "receipts",
      "$id" : ObjectId("5e91a9370d2d1f50d3c45c83")
    }
  ]
}

```



```
db.receipts.find()
```

receipts 0.001 sec.

```
/* 1 */
{
  "_id" : ObjectId("5e91a9370d2d1f50d3c45c81"),
  "date" : ISODate("2020-04-11T11:25:43.138Z"),
  "products" : [
    {
      "$ref" : "products",
      "$id" : ObjectId("5e91ab200d2d1f50d3c45c84")
    }
  ]
}

/* 2 */
{
  "_id" : ObjectId("5e91a9370d2d1f50d3c45c82"),
  "date" : ISODate("2020-04-11T11:25:43.155Z"),
  "products" : [
    {
      "$ref" : "products",
      "$id" : ObjectId("5e91ab200d2d1f50d3c45c85")
    }
  ]
}

/* 3 */
{
  "_id" : ObjectId("5e91a9370d2d1f50d3c45c83"),
  "date" : ISODate("2020-04-11T11:25:43.157Z"),
  "products" : [
    {
      "$ref" : "products",
      "$id" : ObjectId("5e91ab200d2d1f50d3c45c86")
    }
  ]
}
```



```
db.products.find()
```

products 0.001 sec.

```
/* 1 */
{
  "_id" : ObjectId("5e91ab200d2d1f50d3c45c84"),
  "name" : "pizza kiler",
  "price" : 24.99,
  "category" : "food"
}

/* 2 */
{
  "_id" : ObjectId("5e91ab200d2d1f50d3c45c85"),
  "name" : "kontener brudnych pieniedzy",
  "price" : 200000.0,
  "category" : "just like that"
}

/* 3 */
{
  "_id" : ObjectId("5e91ab200d2d1f50d3c45c86"),
  "name" : "zurek",
  "price" : 420.69,
  "category" : "sheraton"
}
```