

Dokumentowe bazy danych – MongoDB

ćwiczenie 1

Imiona i nazwiska autorów: Łukasz Kluza, Mateusz Sacha

Zadanie 1 - połączenie z serwerem bazy danych

Połącz się serwerem MongoDB

Można skorzystać z własnego/lokanego serwera MongoDB Można stworzyć własny klaster/bazę danych w serwisie MongoDB Atlas

- <https://www.mongodb.com/atlas/database>

Połącz za pomocą konsoli mongosh

Ewentualnie zdefiniuj połączenie w wybranym przez siebie narzędziu

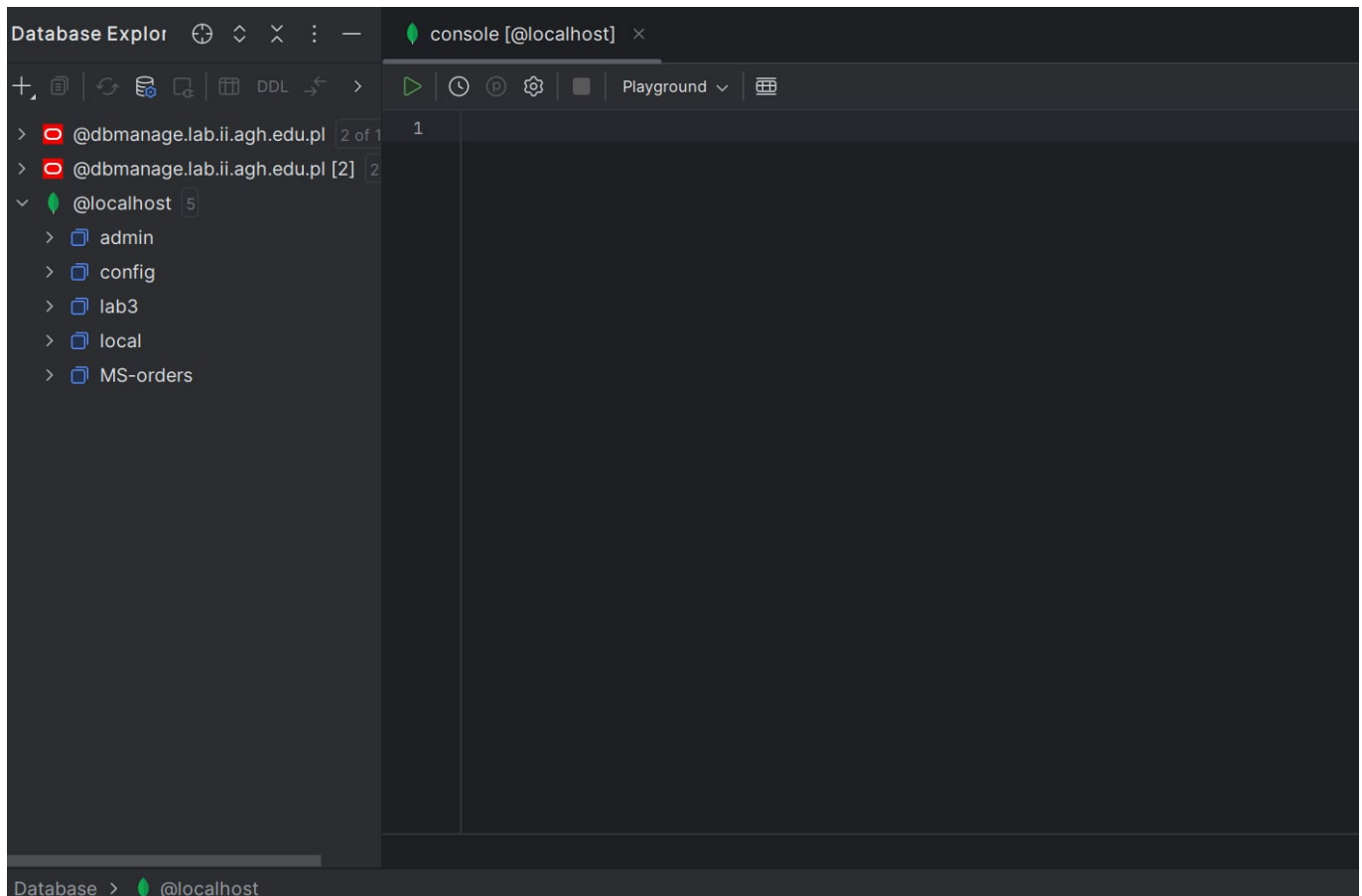
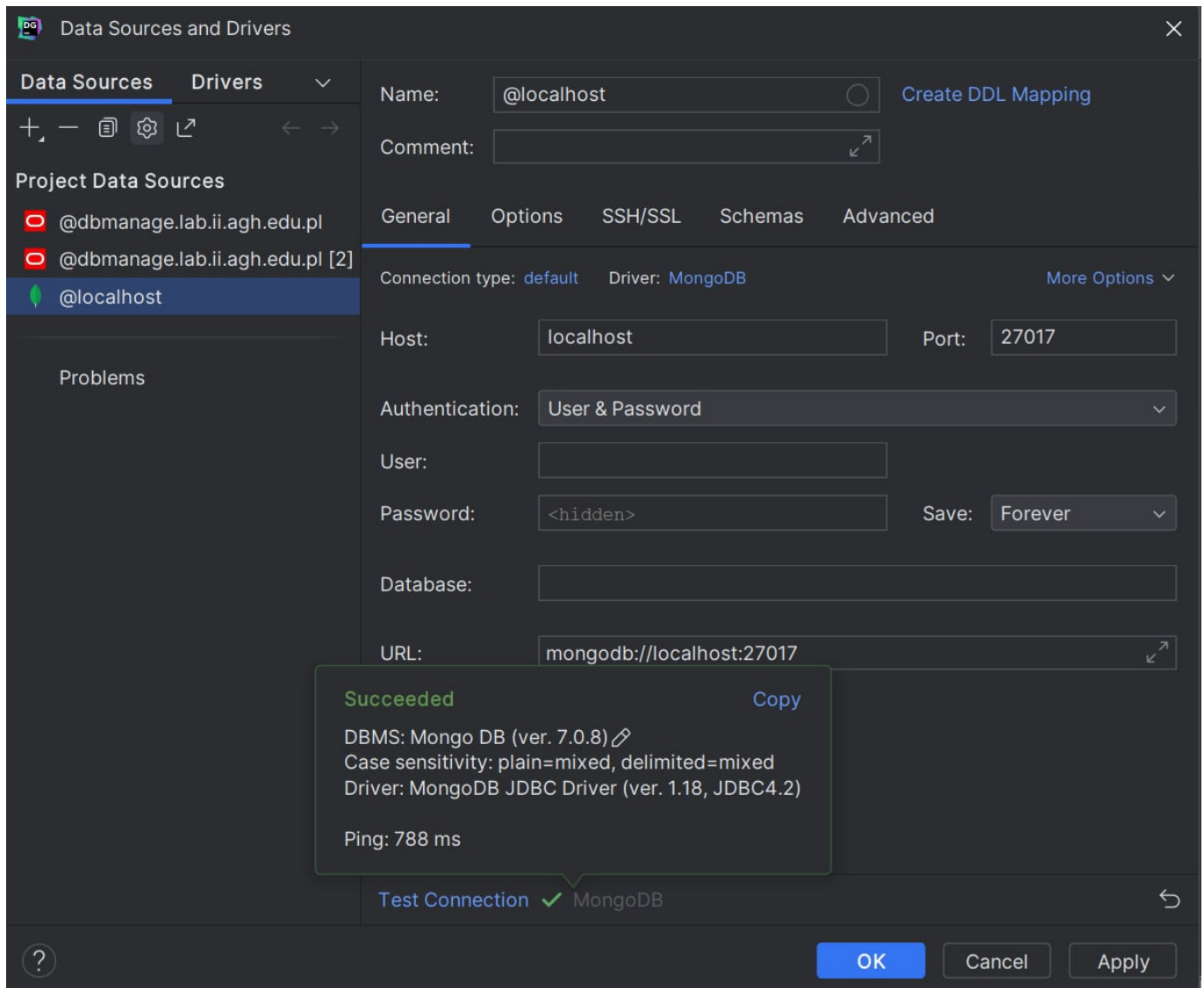
Stwórz bazę danych/kolekcję/dokument

- może to być dowolna kolekcja, dowolny dokument – o dowolnej strukturze, chodzi o przetestowanie działania połączenia

Zadanie 1 - rozwiązanie

Wyniki:

Użyliśmy DataGrip do połączenia się z bazą danych



```
db.collection.insertMany([{name: "Lukasz", lastname: "Kluza"},{name: "Mateusz",  
lastname: "Sacha"}])  
db.collection.find({})
```

	{_id}	{lastname}	{name}
3	6616acb7eb7d9575da676485	Kluza	Lukasz
4	6616acb7eb7d9575da676486	Sacha	Mateusz

Zadanie 2 - przykładowe zbiory danych

Zaimportuj przykładowe zbiory danych

MongoDB Atlas Sample Dataset

- <https://docs.atlas.mongodb.com/sample-data>
- w przypadku importu z lokalnych plików można wykorzystać polecenie `mongorestore`
 - <https://www.mongodb.com/docs/database-tools/mongorestore/>

```
mongorestore <data-dump-folder>
```

np.

```
mongorestore samples
```

- Oczywiście, w przypadku łączenia się zdalnym serwerem należy podać parametry połączenia oraz dane logowania

Yelp Dataset

- wykorzystaj komendę `mongoimport`
- <https://www.mongodb.com/docs/database-tools/mongoimport>

```
mongoimport --db <db-name> --collection <coll-name> --type json --file <file>
```

np.

```
mongoimport --db yelp --collection business --type json --file  
./yelp_academic_dataset_business.json
```

- można też wykorzystać np. narzędzie MongoDB Compass

Zapoznaj się ze strukturą przykładowych zbiorów danych/kolekcji

- W bazach danych: MongoDB Atlas Sample Dataset
 - Skomentuj struktury użyte w dokumentach dla dwóch wybranych zbiorów (takich które wydają ci się najciekawsze)
 - np. Sample Analytics Dataset i Sample Training Dataset
- W bazie Yelp
 - Skomentuj struktury użyte w dokumentach bazy Yelp

Przetestuj działanie operacji

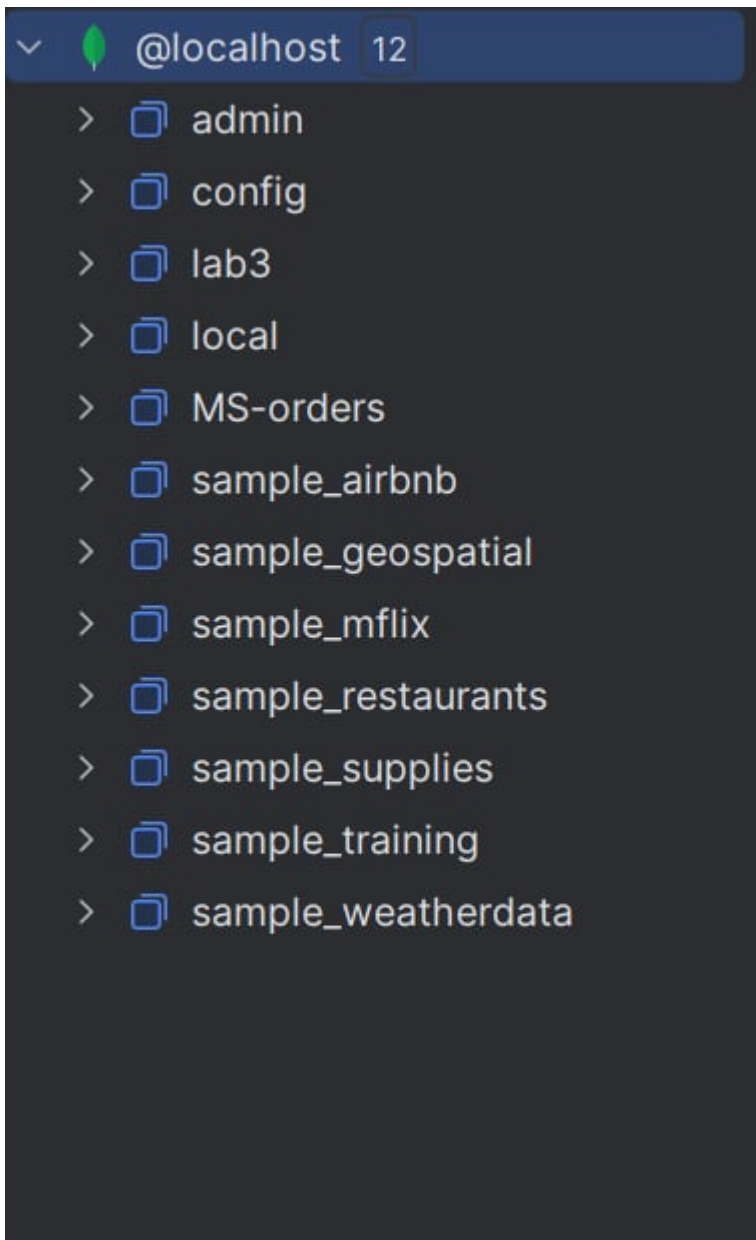
- `mongodump`
 - <https://www.mongodb.com/docs/database-tools/mongodump/>
- `mongoexport`
 - <https://www.mongodb.com/docs/database-tools/mongoexport/>

Zadanie 2 - rozwiązanie

Wyniki:

```
Windows PowerShell
PS C:\Users\zsuet\OneDrive\Pulpit\mongo\mongodb-database-tools-windows-x86_64-100.9.4\bin> .\mongorestore samples

2024-04-10T17:36:35.612+0200 no indexes to restore for collection sample_training.inspections
2024-04-10T17:36:35.612+0200 no indexes to restore for collection sample_weatherdata.data
2024-04-10T17:36:35.612+0200 restoring indexes for collection sample_mflix.users from metadata
2024-04-10T17:36:35.613+0200 index: &idx.IndexDocument{Options:primitive.M{"name":"email_1", "unique":true, "v":2}, K
ey:primitive.D{primitive.E{Key:"email", Value:1}}, PartialFilterExpression:primitive.D(nil)}
2024-04-10T17:36:35.613+0200 restoring indexes for collection sample_mflix.sessions from metadata
2024-04-10T17:36:35.613+0200 index: &idx.IndexDocument{Options:primitive.M{"name":"user_id_1", "unique":true, "v":2},
Key:primitive.D{primitive.E{Key:"user_id", Value:1}}, PartialFilterExpression:primitive.D(nil)}
2024-04-10T17:36:35.721+0200 restoring indexes for collection sample_mflix.theaters from metadata
2024-04-10T17:36:35.721+0200 index: &idx.IndexDocument{Options:primitive.M{"2dsphereIndexVersion":3, "name":"geo inde
x", "v":2}, Key:primitive.D{primitive.E{Key:"location.geo", Value:"2dsphere"}}, PartialFilterExpression:primitive.D(nil)}
2024-04-10T17:36:35.771+0200 restoring indexes for collection sample_mflix.movies from metadata
2024-04-10T17:36:35.771+0200 index: &idx.IndexDocument{Options:primitive.M{"default_language":"english", "language_ov
erride":"language", "name":"cast_text_fullplot_text_genres_text_title_text", "textIndexVersion":3, "v":2, "weights":prim
itive.M{"cast":1, "fullplot":1, "genres":1, "title":1}}, Key:primitive.D{primitive.E{Key:"fts", Value:"text"}, primitiv
e.E{Key:"fts_x", Value:1}}, PartialFilterExpression:primitive.D(nil)}
2024-04-10T17:36:35.839+0200 no indexes to restore for collection sample_mflix.comments
2024-04-10T17:36:46.043+0200 429289 document(s) restored successfully. 0 document(s) failed to restore.
PS C:\Users\zsuet\OneDrive\Pulpit\mongo\mongodb-database-tools-windows-x86_64-100.9.4\bin>
```



```
PS C:\Users\zsuet\OneDrive\Pulpit\mongo\mongodb-database-tools-windows-x86_64-100.9.4\bin> mongoimport --db yelp --collection ./business --type json --file ./yelp_academic_dataset_business.json
```

```
PS C:\Users\zsuet\OneDrive\Pulpit\mongo\mongodb-database-tools-windows-x86_64-100.9.4\bin> .\mongoimport --db yelp --collection business --type json --file ./yelp/business.json
2024-04-10T17:42:27.473+0200    connected to: mongodb://localhost/
2024-04-10T17:42:30.473+0200    [#####...] yelp.business          30.1MB/35.5MB (84.8%)
2024-04-10T17:42:31.042+0200    [#####] yelp.business             35.5MB/35.5MB (100.0%)
2024-04-10T17:42:31.042+0200    42153 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\zsuet\OneDrive\Pulpit\mongo\mongodb-database-tools-windows-x86_64-100.9.4\bin>
```

W bazie danych sample_training w kolekcji companies możemy zauważyć takie struktury jak lista oraz zagnieżdżone w struktury, które też posiadają listy.

```
id: 2312,  
round_code: 'c',  
source_url: 'http://www.accel.com/news/news_one_up.php?news_id=185',  
source_description: 'Accel',  
raised_amount: 25000000,  
raised_currency_code: 'USD',  
funded_year: 2008,  
funded_month: 5,  
funded_day: 19,  
investments: [  
  {  
    company: null,  
    financial_org: {  
      name: 'DAG Ventures',  
      permalink: 'dag-ventures'  
    },  
    person: null  
  },  
]
```

W bazie danych sample_training w kolekcji trips, możemy zauważyć pola napisane w ' ' z powodu tego, że zawierają spacje czyli znaki niedozwolone w normalnej nazwie pola, widzimy także użycie dokumentów zagnieżdżonych 'start station location' oraz kolejne listy.

```
_id: ObjectId('572bb8222b288919b68abf5a'),
tripduration: 379,
'start station id': 476,
'start station name': 'E 31 St & 3 Ave',
'end station id': 498,
'end station name': 'Broadway & W 32 St',
bikeid: 17827,
usertype: 'Subscriber',
'birth year': 1969,
gender: 1,
'start station location': {
  type: 'Point',
  coordinates: [
    -73.97966069,
    40.74394314
  ]
},
```

W bazie yelp w kolekcji business widzimy, np listy zawierjæ stringi oraz puste dokumenty zagnie¿dzone oraz puste listy.


```
_id: ObjectId('6616b4563e496780131f7f23'),
business_id: 'ZW2WeP2Hp20tq0RG1NFkoQ',
full_address: '4709 Triangle St\n' +
  'Mc Farland, WI 53558',
hours: {},
open: true,
categories: [
  'Home Services',
  'Heating & Air Conditioning/HVAC'
],
city: 'Mc Farland',
review_count: 4,
name: 'Air Quality Systems',
neighborhoods: [],
longitude: -89.2985442,
state: 'WI',
stars: 2,
latitude: 43.0303561,
```

W bazie yelp w kolekcji checkin widzimy pole będące mapą/słownikiem.

```
_id: ObjectId('6616b46248002c54481cc783'),
checkin_info: {
  '14-5': 1,
  '10-5': 1,
  '7-5': 1
},
type: 'checkin',
business_id: '95p9Xg358BezJyk1wqzzyg'
}
```


Zadanie 3 - operacje CRUD, operacje wyszukiwania danych

<https://www.mongodb.com/docs/manual/crud/>

Stwórz nową bazę danych

- baza danych będzie przechowywać informacje o klientach, produktach, zamówieniach tych produktów. itp.
- w nazwie bazy danych użyj swoich inicjałów
 - np. **AB-orders**
- zaproponuj strukturę kolekcji/dokumentów (dwie, maksymalnie 3 kolekcje)
 - wykorzystaj typy proste/podstawowe, dokumenty zagnieżdżone, tablice itp.
 - wprowadź kilka przykładowych dokumentów
 - przetestuj operacje wstawiania, modyfikacji/usuwania dokumentów
 - przetestuj operacje wyszukiwania dokumentów

Zadanie 3 - rozwiązanie

Wyniki:

przykłady, kod, zrzuty ekranów, komentarz ...

```
db.createCollection("customers")
db.createCollection("orders")
db.createCollection("products")
```

```
db.customers.insertMany([
  { _id: 1, name: "Ala", lastname: "Kowalska", country: "Poland", city: "Cracow" },
  { _id: 2, name: "Jan", lastname: "Kowalski", country: "Poland", city: "Cracow" },
  { _id: 3, name: "John", lastname: "Cash", country: "France", city: "Paris" },
  { _id: 4, name: "David", lastname: "Messi", country: "Poland", city: "Cracow" }
])
```

	{ _id }	{ city }	{ country }	{ lastname }	{ name }
1	1	Cracow	Poland	Kowalska	Ala
2	2	Cracow	Poland	Kowalski	Jan
3	3	Paris	France	Cash	John
4	4	Cracow	Poland	Messi	David

```

db.products.insertMany([
  {_id: 1, name: "Shirt", in_stock: 12, colors: ["Red", "Blue"], size: { h: 14, w: 21, uom: "cm" }},
  {_id: 2, name: "Dress", in_stock: 100, colors: ["Green", "Blue"], size: { h: 11, w: 41, uom: "cm" }},
  {_id: 3, name: "Trouser", in_stock: 13, colors: ["Yellow", "Blue"], size: { h: 4.5, w: 11, uom: "in" }},
  {_id: 4, name: "Boots", in_stock: 30, colors: ["Red", "Black"], size: { h: 8.5, w: 12, uom: "in" }}
])
db.products.find({})

```

	{_id}	{ colors }	{ in_stock }	{ name }	{ size }
1	1	["Red", "Blue"]	12	Shirt	{ "h": new NumberInt("14"), "w": new NumberInt("21"), "uom": "cm" }
2	2	["Green", "Blue"]	100	Dress	{ "h": new NumberInt("11"), "w": new NumberInt("41"), "uom": "cm" }
3	3	["Yellow", "Blue"]	13	Trouser	{ "h": 4.5, "w": new NumberInt("11"), "uom": "in" }
4	4	["Red", "Black"]	30	Boots	{ "h": 8.5, "w": new NumberInt("12"), "uom": "in" }

```

db.orders.insertMany([
  {_id: 1, product_id: 1, customer_id: 1, quantity: 2, discount: 0.02, order_date: new Date("2024-04-09")},
  {_id: 2, product_id: 2, customer_id: 3, quantity: 5, discount: 0.12, order_date: new Date("2024-03-10")},
  {_id: 3, product_id: 3, customer_id: 4, quantity: 11, discount: 0.04, order_date: new Date("2024-04-11")},
  {_id: 4, product_id: 1, customer_id: 1, quantity: 6, discount: 0.01, order_date: new Date("2024-04-13")},
])

```

	{_id}	{ customer_id }	{ discount }	{ order_date }	{ product_id }	{ quantity }
1	1	1	0.02	2024-04-09T00:00:00.000Z	1	2
2	2	3	0.12	2024-03-10T00:00:00.000Z	2	5
3	3	4	0.04	2024-04-11T00:00:00.000Z	3	11
4	4	1	0.01	2024-04-13T00:00:00.000Z	1	6

```

db.orders.find({customer_id: 1})

```

	{_id}	{ customer_id }	{ discount }	{ order_date }	{ product_id }	{ quantity }
1	1	1	0.02	2024-04-09T00:00:00.000Z	1	2
2	4	1	0.01	2024-04-13T00:00:00.000Z	1	6

```

db.products.find( {
  $or: [
    {size: { h: 14, w: 21, uom: "cm" } },
    {size: { h: 8.5, w: 12, uom: "in" } }
  ]
});

```

	<div>{} _id</div>	<div>{} colors</div>	<div>{} in_stock</div>	<div>{} name</div>	<div>{} size</div>
1	1	["Red", "Blue"]	12	Shirt	{ "h": new NumberInt("14"), "w": new NumberInt("21"), "uom": "cm" }
2	4	["Red", "Black"]	30	Boots	{ "h": 8.5, "w": new NumberInt("12"), "uom": "in" }

```
db.products.find( { in_stock: { $gt: 20 } } )
```

	<div>{} _id</div>	<div>{} colors</div>	<div>{} in_stock</div>	<div>{} name</div>	<div>{} size</div>
1	2	["Green", "Blue"]	100	Dress	{ "h": new NumberInt("11"), "w": new NumberInt("41"), "uom": "cm" }
2	4	["Red", "Black"]	30	Boots	{ "h": 8.5, "w": new NumberInt("12"), "uom": "in" }

```
db.products.updateOne({_id: 1}, {$set: {in_stock: 14}})
```

	<div>{} _id</div>	<div>{} colors</div>	<div>Add Row Alt+Insert</div>	<div>{} name</div>	<div>{} size</div>
1	1	["Red", "Blue"]	14	Shirt	{ "h": new NumberInt("14"), "w": new NumberInt("21"), "uom": "cm" }

```
db.products.deleteOne({_id: 2})
```

	<div>{} acknowledged</div>	<div>{} deletedCount</div>
1	• true	1

	<div>{} _id</div>	<div>{} colors</div>	<div>{} in_stock</div>	<div>{} name</div>	<div>{} size</div>
1	1	["Red", "Blue"]	14	Shirt	{ "h": new NumberInt("14"), "w": new NumberInt("21"), "uom": "cm" }
2	3	["Yellow", "Blue"]	13	Trouser	{ "h": 4.5, "w": new NumberInt("11"), "uom": "in" }
3	4	["Red", "Black"]	30	Boots	{ "h": 8.5, "w": new NumberInt("12"), "uom": "in" }

Ćwiczenie przeznaczone jest do wykonania podczas zajęć. Pod koniec zajęć należy przesłać wyniki prac

Punktacja:

zadanie	pkt
1	0,1
2	0,2
3	0,7
razem	1