

Dokumentowe bazy danych – MongoDB

ćwiczenie 1

Imiona i nazwiska autorów: Tomasz Furgala, Łukasz Zegar, Konrad Tendaj

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

Tworzymy serwer w mongoDB Atlas

W konsoli z zainstalowanym mongosh wpisujemy:

```
mongosh "mongodb+srv://cluster1.oj7zkb.mongodb.net/" --apiVersion 1 --username <nasza_nazwa>
```

Podajemy hasło: <nasze_hasło>

Z danymi użytkownika utworzonemu w bazie Atlas.

Korzystamy z `use my_db`, które utworzy nam bazę danych jeżeli taka nie istnieje.

Dodajemy przykładowe dane:

```
db.students.insertOne(
  {
    nr_albumu: 111,
    Imie: "Jan",
    Nazwisko: "Kowalski"
  }
)

db.students.insertMany([
  {nr_albumu: 112, Imie: "Natalia", Nazwisko: "Robak"},
  {nr_albumu: 113, Imie: "Kamil", Nazwisko: "Nowak"},
  {nr_albumu: 114, Imie: "Julia", Nazwisko: "Mazowiecka"}
]);
```

Korzystamy z find'a na kolekcji students:

```
Atlas atlas-13u8tm-shard-0 [primary] my_db> db.students.find({})
[
  {
    _id: ObjectId('660d5c0affd05271ef9f990a'),
    nr_albumu: 111,
    Imie: 'Jan',
    Nazwisko: 'Kowalski'
  },
  {
    _id: ObjectId('660d5d2cffd05271ef9f990e'),
    nr_albumu: 112,
    Imie: 'Natalia',
    Nazwisko: 'Robak'
  },
  {
    _id: ObjectId('660d5d2cffd05271ef9f990f'),
    nr_albumu: 113,
    Imie: 'Kamil',
    Nazwisko: 'Nowak'
  },
  {
    _id: ObjectId('660d5d2cffd05271ef9f9910'),
    nr_albumu: 114,
    Imie: 'Julia',
    Nazwisko: 'Mazowiecka'
  }
]
```

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:

- Do wykonania importu Atlas Sample wykorzystujemy komendy w formie

```
mongorestore --uri="mongodb+srv://username:password@clustername" --archive=path/to/folder
```

- Analizujemy baze sample_mflix z atlas samples

▼ sample_mflix

comments

embedded_movies

movies

sessions

theaters

users

▶ sample_weatherdata

- W bazie mflix istnieją powiązania między różnymi kolekcjami. Na przykład, recenzja jest powiązana z filmem poprzez identyfikator filmu.
- poniżej wyświetlony fragment kolekcji movies

```
use sample_mflix
db.movies.find()
```

```
{
  _id: ObjectId("573a1390f29313caabcd516c"),
  plot: 'Original advertising for the film describes it as a drama of primitive life on the shores of the
North Pacific...',
  genres: [ 'Drama', 'History' ],
  runtime: 65,
  cast: [
    'Stanley Hunt',
    'Sarah Constance Smith Hunt',
    'Mrs. George Walkus',
    "Paddy 'Malid"
  ],
  num_mflix_comments: 1,
  poster: 'https://m.media-
amazon.com/images/M/MV5BMjE3MjAyNzZM5NV5BM15BanBnXkFtZTgwMjA5OTg5NjE@._V1_SY1000_SX677_AL_.jpg',
```

```
title: 'In the Land of the Head Hunters',
lastupdated: '2015-09-16 12:11:37.770000000',
languages: [ 'English' ],
released: ISODate("1914-12-07T00:00:00.000Z"),
directors: [ 'Edward S. Curtis' ],
writers: [ 'Edward S. Curtis (story)' ],
awards: { wins: 1, nominations: 0, text: '1 win.' },
year: 1914,
imdb: { rating: 5.8, votes: 223, id: 4150 },
countries: [ 'USA' ],
type: 'movie',
tomatoes: {
  viewer: { rating: 2.7, numReviews: 64, meter: 18 },
  dvd: ISODate("2000-08-15T00:00:00.000Z"),
  website: 'http://www.milestonefilms.com/',
  production: 'World Film Corporation',
  lastUpdated: ISODate("2015-08-18T19:21:12.000Z")
}
},
```

- Kolekcja ta zawiera dokumenty reprezentujące pojedyncze filmy. Każdy dokument zawiera różne pola opisujące film, takie jak tytuł, rok produkcji, gatunek, lista obsady, reżyser itp.
- Struktura jest elastyczna i można ją rozszerzyć o dodatkowe pola. Zawiera zagnieżdżone dokumenty i tablice.
- Analizujemy "business" z bazy Yelp
- Przykładowy dokument prezentuje się następująco:

```
{
  "_id": "ObjectId",
  "name": "Business Name",
  "address": {
    "street": "Street Name",
    "city": "City Name",
    "state": "State Name",
    "zip": "Zip Code"
  },
  "categories": ["Category 1", "Category 2"],
  "stars": 4.5,
  "review_count": 100
}
```

- W bazie Yelp dane są przechowywane w jednym dokumencie, co może utrudniać aktualizację danych, ale przyspiesza odczyty.
- Dokumenty zawierają bogate informacje o firmach, takie jak adres, kategorie, gwiazdki i liczba recenzji.
- Zawiera zagnieżdżone dokumenty i tablice.

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

Tworzymy bazę danych poleceniem: `use TF_orders`

Dodajemy dane o klientach:

```
db.clients.insertOne({
  Client_id: 1,
  FirstName: "John",
  LastName: "Doe",
  Address: ""
})

db.clients.insertMany([
  {
    Client_id: 2,
    FirstName: "Alice",
    LastName: "Pope",
    Address: ""
  },
  {
    Client_id: 3,
    FirstName: "John",
    LastName: "Luke",
    Address: ""
  }
])
```

```
Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.clients.find()
[
  {
    _id: ObjectId('660d61b8ffd05271ef9f9912'),
    Client_id: 1,
    FirstName: 'John',
    LastName: 'Doe',
    Address: ''
  },
  {
    _id: ObjectId('660d6287ffd05271ef9f9913'),
    Client_id: 2,
    FirstName: 'Alice',
    LastName: 'Pope',
    Address: ''
  },
  {
    _id: ObjectId('660d6287ffd05271ef9f9914'),
    Client_id: 3,
    FirstName: 'John',
    LastName: 'Luke',
    Address: ''
  }
]
```

Dodajemy dane zamówień

```
db.orders.insertOne({
  Order_id: 1,
  Client_id: 1,
  Order_date: new Date("2024-04-03"),
  Products: [ 1 ]
})

db.orders.insertMany([
  {
    Order_id: 2,
    Client_id: 2,
    Order_date: new Date("2024-04-02"),
    Products: [ 1, 2, 5 ]
  },
  {
    Order_id: 3,
    Client_id: 1,
    Order_date: new Date("2024-04-10"),
    Products: [ 3, 6 ]
  }
])
```

```
},
{
  Order_id: 4,
  Client_id: 3,
  Order_date: new Date("2024-02-25"),
  Products: [ 2, 6 ]
}])
```

Dodajemy dokument z produktami:

```
db.products.insertMany([
{
  id: 1,
  name: 'Smartphone',
  description: 'Latest smartphone model',
  price: 599.99,
  stock: 100,
  details: {
    dimensions: { length: 5.5, width: 2.5, height: 0.3 },
    weight: 150,
    manufacturer: 'ABC Electronics'
  },
  reviews: [
    {
      user: 'Alice',
      rating: 4,
      comment: 'Great phone, but the battery life could be better.'
    },
    {
      user: 'Bob',
      rating: 5,
      comment: 'Excellent device, highly recommended!'
    }
  ]
},
{
  "id": 2,
  "name": "Laptop",
  "description": "High-performance laptop with SSD storage",
  "price": 1299.99,
  "stock": 50,
  "details": {
    "dimensions": {
      "length": 13.5,
      "width": 9.75,
      "height": 0.6
    },
    "weight": 1300,
    "manufacturer": "XYZ Electronics"
  },
  "reviews": [
    {
      "user": "Alice",
      "rating": 4,
      "comment": "Great laptop, but a bit heavy."
    }
  ]
},
{
  "id": 3,
  "name": "Headphones",
  "description": "Wireless headphones with noise cancellation",
  "price": 199.99,
  "stock": 100,
  "details": {
    "dimensions": {
      "length": 7,
```

```
        "width": 6,
        "height": 3
    },
    "weight": 250,
    "manufacturer": "AudioTech"
},
"reviews": [
    {
        "user": "Bob",
        "rating": 5,
        "comment": "Excellent sound quality!"
    }
]
},
{
    "id": 4,
    "name": "Tablet",
    "description": "10-inch tablet with retina display",
    "price": 499.99,
    "stock": 30,
    "details": {
        "dimensions": {
            "length": 9.5,
            "width": 7.31,
            "height": 0.3
        },
        "weight": 450,
        "manufacturer": "TabCo"
    },
    "reviews": [
        {
            "user": "Charlie",
            "rating": 3,
            "comment": "Average tablet, could be faster."
        }
    ]
},
{
    "id": 5,
    "name": "Smartwatch",
    "description": "Fitness tracker with heart rate monitor",
    "price": 149.99,
    "stock": 80,
    "details": {
        "dimensions": {
            "length": 1.5,
            "width": 1.5,
            "height": 0.4
        },
        "weight": 50,
        "manufacturer": "FitTech"
    },
    "reviews": [
        {
            "user": "David",
            "rating": 4,
            "comment": "Great for tracking workouts!"
        }
    ]
},
{
    "id": 6,
    "name": "Camera",
    "description": "Mirrorless camera with 4K video recording",
    "price": 899.99,
    "stock": 20,
    "details": {
        "dimensions": {
            "length": 4.75,
```

```

        "width": 3.25,
        "height": 2.75
    },
    "weight": 500,
    "manufacturer": "CamTech"
  },
  "reviews": [
    {
      "user": "Eva",
      "rating": 5,
      "comment": "Love the image quality!"
    }
  ]
}
])

```

Przykładowa agregacja łącząca orders z clients, wyświetlająca dane o kliencie przy zamówieniu:

```

db.orders.aggregate([
  {
    $lookup: {
      from: "clients",
      localField: "Client_id",
      foreignField: "Client_id",
      as: "client_info"
    }
  }
])

```

```

Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.orders.aggregate([
... {
...   $lookup: {
...     from: "clients",
...     localField: "Client_id",
...     foreignField: "Client_id",
...     as: "client_info"
...   }
... }
... ])
[
  {
    _id: ObjectId('660d644cffd05271ef9f9915'),
    Order_id: 1,
    Client_id: 1,
    Order_date: ISODate('2024-04-03T00:00:00.000Z'),
    Products: [ 1 ],
    client_info: [
      {
        _id: ObjectId('660d61b8ffd05271ef9f9912'),
        Client_id: 1,
        FirstName: 'John',
        LastName: 'Doe',
        Address: ''
      }
    ]
  }
]

```

Modyfikowanie danych

```

db.clients.updateOne(
  { Client_id: 1 },
  {
    $set: {
      Address: "New Address"
    }
  }
)

```



```
Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.clients.find({Client_id: 1})
[
  {
    _id: ObjectId('660d749664710778159f990a'),
    Client_id: 1,
    FirstName: 'John',
    LastName: 'Doe',
    Address: 'New Address'
  }
]
```

Usuwanie danych

```
db.orders.deleteOne({ Order_id: 4 })
```

Operacje wyszukiwania dokumentów

```
db.orders.find({ Products: 2 })
```

Wyświetla zamówienia które posiadają produkt o id równym 2

```
Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.orders.find({ Products: 2 })
[
  {
    _id: ObjectId('660d665bfffd05271ef9f9916'),
    Order_id: 2,
    Client_id: 2,
    Order_date: ISODate('2024-04-02T00:00:00.000Z'),
    Products: [ 1, 2, 5 ]
  },
  {
    _id: ObjectId('660d665bfffd05271ef9f9918'),
    Order_id: 4,
    Client_id: 3,
    Order_date: ISODate('2024-02-25T00:00:00.000Z'),
    Products: [ 2, 6 ]
  }
]
```

Wyświetlanie zamówień, które zawierają co najmniej 2 produkty

```
db.orders.find({ "Products": { $elemMatch: { $size: { $gte: 2 } } } })
```

```
Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.orders.find({ Products: 2 })
[
  {
    _id: ObjectId('660d665bfffd05271ef9f9916'),
    Order_id: 2,
    Client_id: 2,
    Order_date: ISODate('2024-04-02T00:00:00.000Z'),
    Products: [ 1, 2, 5 ]
  },
  {
    _id: ObjectId('660d665bfffd05271ef9f9918'),
    Order_id: 4,
    Client_id: 3,
    Order_date: ISODate('2024-02-25T00:00:00.000Z'),
    Products: [ 2, 6 ]
  }
]
```

Wyświetla zamówienia wraz z danymi o kliencie

```
db.orders.aggregate([
  {
    $lookup: {
      from: "clients",
      localField: "Client_id",
      foreignField: "Client_id",
```

```

        as: "client_info"
      }
    }
  ])

```

```

[
  {
    _id: ObjectId('660d644cf9d05271ef9f9915'),
    Order_id: 1,
    Client_id: 1,
    Order_date: ISODate('2024-04-03T00:00:00.000Z'),
    Products: [ 1 ],
    client_info: [
      {
        _id: ObjectId('660d749664710778159f990a'),
        Client_id: 1,
        FirstName: 'John',
        LastName: 'Doe',
        Address: 'New Address'
      }
    ]
  },
  {
    _id: ObjectId('660d665bffd05271ef9f9917'),
    Order_id: 3,
    Client_id: 1,
    Order_date: ISODate('2024-04-10T00:00:00.000Z'),
    Products: [ 3, 6 ],
    client_info: [
      {
        _id: ObjectId('660d749664710778159f990a'),
        Client_id: 1,
        FirstName: 'John',
        LastName: 'Doe',
        Address: 'New Address'
      }
    ]
  },
  {
    _id: ObjectId('660d665bffd05271ef9f9918'),
    Order_id: 4,
    Client_id: 3,
    Order_date: ISODate('2024-02-25T00:00:00.000Z'),
    Products: [ 2, 6 ],
    client_info: [
      {
        _id: ObjectId('660d749d64710778159f990c'),
        Client_id: 3,
        FirstName: 'John',
        LastName: 'Luke',
        Address: ''
      }
    ]
  }
]

```

```
db.clients.find().sort({ LastName: 1, FirstName: 1 })
```

Wyświetla posortowanych klientów

```

Atlas atlas-13u8tm-shard-0 [primary] TF_orders> db.clients.find().sort({ LastName: 1, FirstName: 1 })
[
  {
    _id: ObjectId('660d749664710778159f990a'),
    Client_id: 1,
    FirstName: 'John',
    LastName: 'Doe',
    Address: 'New Address'
  },
  {
    _id: ObjectId('660d749d64710778159f990c'),
    Client_id: 3,
    FirstName: 'John',
    LastName: 'Luke',
    Address: ''
  },
  {
    _id: ObjectId('660d749d64710778159f990b'),
    Client_id: 2,
    FirstName: 'Alice',
    LastName: 'Pope',
    Address: ''
  }
]

```

Punktacja:

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