

Dokumentowe bazy danych – MongoDB

Ćwiczenie 1 - do wykonania podczas zajęć

Imię i nazwisko: Kacper Ćwiertnia

Materiały:

Książki

Np.

- Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, MongoDB: The Definitive Guide. Powerful and Scalable Data Storage, O'Reilly 2019
- Alex Giamas, Mastering MongoDB 4.x., Pact 2019

Dokumentacja

- <https://www.mongodb.com/docs/manual/reference/program/mongo/>

MongoDB University Courses

- <https://university.mongodb.com/courses/catalog>
- MongoDB Basics
 - <https://university.mongodb.com/courses/M001/about>
- The MongoDB Aggregation Framework
 - <https://university.mongodb.com/courses/M121/about>
- Data Modeling
 - <https://university.mongodb.com/courses/M320/about>

Zbiory danych

The MongoDB Atlas Sample Datasets

<https://www.mongodb.com/developer/article/atlas-sample-datasets/>

Yelp Dataset

www.yelp.com - serwis społecznościowy – informacje o miejscach/lokalach

- restauracje, kluby, hotele itd. (*businesses*),
- użytkownicy piszą recenzje (*reviews*) o miejscach i wystawiają oceny,
- użytkownicy odwiedzają te miejsca - "meldują się" (*check-in*)
- Przykładowy zbiór danych zawiera dane z 5 miast: Phoenix, Las Vegas, Madison, Waterloo i Edinburgh.

Kolekcje:

- **42,153** businesses
- **320,002** business attributes
- **31,617** check-in sets
- **252,898** users
- **955,999** edge social graph
- **403,210** tips
- **1,125,458** reviews

business

```
{
  'type': 'business',
  'business_id': (encrypted business id),
  'name': (business name),
  'neighborhoods': [(hood names)],
  'full_address': (localized address),
  'city': (city),
  'state': (state),
  'latitude': latitude,
  'longitude': longitude,
  'stars': (star rating, rounded to half-stars),
  'review_count': review count,
  'categories': [(localized category names)]
  'open': True / False (corresponds to closed, not business hours),
  'hours': {
    (day_of_week): {
      'open': (HH:MM),
      'close': (HH:MM)
    },
    ...
  },
  'attributes': {
    (attribute_name): (attribute_value),
    ...
  },
}
```

review

```
{
  'type': 'review',
  'business_id': (encrypted business id),
  'user_id': (encrypted user id),
  'stars': (star rating, rounded to half-stars),
  'text': (review text),
  'date': (date, formatted like '2012-03-14'),
  'votes': {(vote type): (count)},
}
```

user

```
{
  'type': 'user',
  'user_id': (encrypted user id),
  'name': (first name),
  'review_count': (review count),
  'average_stars': (floating point average, like 4.31),
  'votes': {(vote type): (count)},
  'friends': [(friend user_ids)],
  'elite': [(years_elite)],
  'yelping_since': (date, formatted like '2012-03'),
  'compliments': {
    (compliment_type): (num_compliments_of_this_type),
    ...
  },
  'fans': (num_fans),
}
```

check-in

```
{
  'type': 'checkin',
  'business_id': (encrypted business id),
  'checkin_info': {
    '0-0': (number of checkins from 00:00 to 01:00 on all Sundays),
    '1-0': (number of checkins from 01:00 to 02:00 on all Sundays),
    ...
    '14-4': (number of checkins from 14:00 to 15:00 on all Thursdays),
    ...
    '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays)
  }, # if there was no checkin for a hour-day block it will not be in the dict
}
```

tip

```
{
  'type': 'tip',
  'text': (tip text),
  'business_id': (encrypted business id),
  'user_id': (encrypted user id),
  'date': (date, formatted like '2012-03-14'),
  'likes': (count),
}
```

Narzędzia

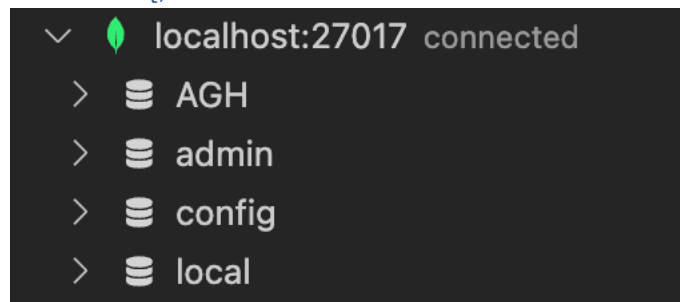
- Konsola: mongosh, MongoDB Database Tools
- Studio3T
- MongoDB Compass
- DataGrip
- VS Code

Zadania

1. Połącz się serwerem MongoDB

- Można skorzystać z własnego/lokanego serwera MongoDB
- Można skorzystać własny klaster/bazę danych w serwisie MongoDB Atlas
 - <https://www.mongodb.com/atlas/database>
 - Serwis umożliwia założenie darmowego konta
- Połącz za pomocą konsoli mongosh
- Ewentualnie zdefiniuj połączenie w wybranym przez siebie narzędziu

Połączyłem się lokalnie z bazą,



.... Wyniki, zrzuty ekranów, kod, komentarz

2. Stwórz bazę danych/kolekcję/dokument

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

Utworzyłem bazę i kolekcję.

```
const database = 'test';
const collection = 'Test';

// Create a new database.
use(database);

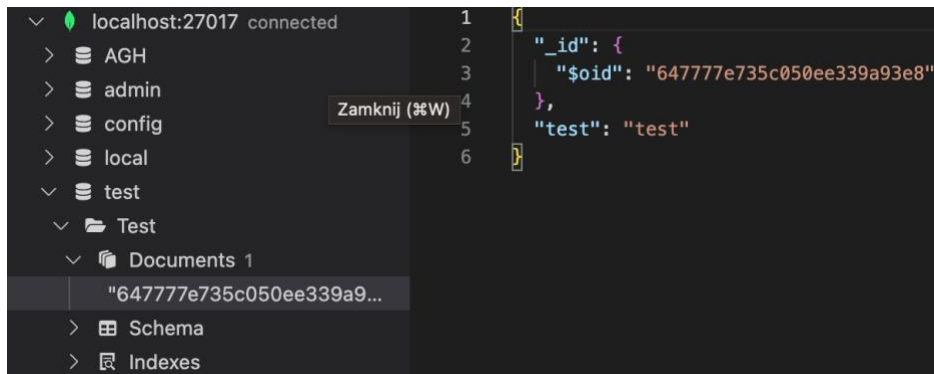
// Create a new collection.
db.createCollection(collection);
```

Tworzę przykładowy dokument.

```
use('test');
```

```
// Create a new document in the collection.
```

```
db.getCollection('Test').insertOne({test:"test"});
```



.... Wyniki, zrzuty ekranów, kod, komentarz

3. Zaimportuj przykładowe zbory danych

- MongoDB Atlas Sample Dataset

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

Importuję wszystkie bazy z foldera samples.

```
kacperciw@MacBook-Air-Kacper-2 lab5 % mongorestore samples
2023-05-31T19:08:51.625+0200 preparing collections to restore from
2023-05-31T19:08:51.630+0200 reading metadata for sample_restaurants.neighborhoods from samples/sample_restaurants/neighborhoods.metadata.json
2023-05-31T19:08:51.633+0200 reading metadata for sample_restaurants.restaurants from samples/sample_restaurants/restaurants.metadata.json
2023-05-31T19:08:51.633+0200 reading metadata for sample_supplies.sales from samples/sample_supplies/sales.metadata.json
2023-05-31T19:08:51.633+0200 reading metadata for sample_training.inspections from samples/sample_training/inspections.metadata.json
2023-05-31T19:08:51.633+0200 reading metadata for sample_training.posts from samples/sample_training/posts.metadata.json
2023-05-31T19:08:51.633+0200 reading metadata for sample_training.trips from samples/sample_training/trips.metadata.json
2023-05-31T19:08:51.634+0200 reading metadata for sample_geospatial.shipwrecks from samples/sample_geospatial/shipwrecks.metadata.json
2023-05-31T19:08:51.634+0200 reading metadata for sample_mflix.comments from samples/sample_mflix/comments.metadata.json
2023-05-31T19:08:51.634+0200 reading metadata for sample_mflix.sessions from samples/sample_mflix/sessions.metadata.json
2023-05-31T19:08:51.635+0200 reading metadata for sample_mflix.theaters from samples/sample_mflix/theaters.metadata.json
2023-05-31T19:08:51.635+0200 reading metadata for sample_training.companies from samples/sample_training/companies.metadata.json
2023-05-31T19:08:51.635+0200 reading metadata for sample_training.routes from samples/sample_training/routes.metadata.json
2023-05-31T19:08:51.635+0200 reading metadata for sample_training.zips from samples/sample_training/zips.metadata.json
2023-05-31T19:08:51.635+0200 reading metadata for sample_mflix.movies from samples/sample_mflix/movies.metadata.json
2023-05-31T19:08:51.636+0200 reading metadata for sample_mflix.users from samples/sample_mflix/users.metadata.json
2023-05-31T19:08:51.636+0200 reading metadata for sample_training.grades from samples/sample_training/grades.metadata.json
2023-05-31T19:08:51.636+0200 reading metadata for sample_weatherdata.data from samples/sample_weatherdata/data.metadata.json
2023-05-31T19:08:51.636+0200 reading metadata for sample_airbnb.listingsAndReviews from samples/sample_airbnb/listingsAndReviews.metadata.json
2023-05-31T19:08:51.696+0200 restoring sample_mflix.movies from samples/sample_mflix/movies.bson
2023-05-31T19:08:51.703+0200 restoring sample_training.grades from samples/sample_training/grades.bson
2023-05-31T19:08:51.722+0200 restoring sample_airbnb.listingsAndReviews from samples/sample_airbnb/listingsAndReviews.bson
2023-05-31T19:08:51.755+0200 restoring sample_training.companies from samples/sample_training/companies.bson
2023-05-31T19:08:52.292+0200 finished restoring sample_training.companies (9500 documents, 0 failures)
2023-05-31T19:08:52.314+0200 restoring sample_training.inspections from samples/sample_training/inspections.bson
2023-05-31T19:08:52.600+0200 finished restoring sample_mflix.movies (23530 documents, 0 failures)
2023-05-31T19:08:52.600+0200 restoring sample_training.posts from samples/sample_training/posts.bson
2023-05-31T19:08:52.774+0200 finished restoring sample_training.posts (500 documents, 0 failures)
2023-05-31T19:08:52.818+0200 restoring sample_weatherdata.data from samples/sample_weatherdata/data.bson
2023-05-31T19:08:52.882+0200 finished restoring sample_airbnb.listingsAndReviews (5555 documents, 0 failures)
2023-05-31T19:08:52.925+0200 restoring sample_mflix.comments from samples/sample_mflix/comments.bson
2023-05-31T19:08:53.045+0200 finished restoring sample_weatherdata.data (10000 documents, 0 failures)
2023-05-31T19:08:53.085+0200 restoring sample_training.routes from samples/sample_training/routes.bson
2023-05-31T19:08:53.287+0200 finished restoring sample_training.grades (100000 documents, 0 failures)
2023-05-31T19:08:53.317+0200 restoring sample_restaurants.restaurants from samples/sample_restaurants/restaurants.bson
2023-05-31T19:08:53.597+0200 finished restoring sample_training.inspections (80047 documents, 0 failures)
2023-05-31T19:08:53.702+0200 restoring sample_training.trips from samples/sample_training/trips.bson
2023-05-31T19:08:53.702+0200 finished restoring sample_mflix.comments (50303 documents, 0 failures)
2023-05-31T19:08:53.748+0200 restoring sample_supplies.sales from samples/sample_supplies/sales.bson
2023-05-31T19:08:53.785+0200 finished restoring sample_restaurants.restaurants (25359 documents, 0 failures)
2023-05-31T19:08:53.829+0200 restoring sample_geospatial.shipwrecks from samples/sample_geospatial/shipwrecks.bson
2023-05-31T19:08:53.829+0200 finished restoring sample_training.trips (10000 documents, 0 failures)
2023-05-31T19:08:53.852+0200 restoring sample_restaurants.neighborhoods from samples/sample_restaurants/neighborhoods.bson
2023-05-31T19:08:53.870+0200 finished restoring sample_supplies.sales (5000 documents, 0 failures)
2023-05-31T19:08:53.870+0200 finished restoring sample_restaurants.neighborhoods (195 documents, 0 failures)
```

```
localhost:27017 connected
> AGH
> admin
> config
> local
> sample_airbnb
  > listingsAndReviews
    > Documents 6K
      "10006546"
      "10009999"
      "1001265"
      "10021707"
      "10030955"
      "1003530"
      "10038496"
      "10047964"
      {
        "id": "10006546",
        "listing_url": "https://www.airbnb.com/rooms/10006546",
        "name": "Ribeira Charming Duplex",
        "summary": "Fantastic duplex apartment with three bedrooms, located in the historic area of P",
        "interaction": "Cot - 10 € / night Dog - € 7,5 / night",
        "house_rules": "Make the house your home...",
        "property_type": "House",
        "room_type": "Entire home/apt",
        "bed_type": "Real Bed",
        "minimum_nights": "2",
        "maximum_nights": "30",
        "cancellation_policy": "moderate",
        "last_scraped": {
          "$date": "2019-02-16T05:00:00Z"
        },
        "calendar_last_scraped": {
          "$date": "2019-02-16T05:00:00Z"
        },
        "first_review": {
```

.... Wyniki, zrzuty ekranów, kod, komentarz

4. 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

sample_airbnb

```
localhost:27017 connected
> AGH
> admin
> config
> local
> sample_airbnb
  > listingsAndReviews
    > Documents 6K
      "10006546"
      "10009999"
      "1001265"
      "10021707"
      "10030955"
      "10035330"
      "10038496"
      "10047964"

1  {
2  "id": "10006546",
3  "listing_url": "https://www.airbnb.com/rooms/10006546",
4  "name": "Ribeira Charming Duplex",
5  "summary": "Fantastic duplex apartment with three bedrooms, located in the historic area of Porto, Ribeira (Cube) - UNESCO Wor
6  "space": "Privileged views of the Douro River and Ribeira square, our apartment offers the perfect conditions to discover the
7  "description": "Fantastic duplex apartment with three bedrooms, located in the historic area of Porto, Ribeira (Cube) - UNESCO
8  "neighborhood_overview": "In the neighborhood of the river, you can find several restaurants as varied flavors, but without fo
9  "notes": "Lose yourself in the narrow streets and staircases zone, have lunch in pubs and typical restaurants, and find the re
10 "transit": "Transport: • Metro station and S. Bento railway 5min; • Bus stop a 50 meters; • Lift Guindais (Funicular) 50 meters
11 "access": "We are always available to help guests. The house is fully available to guests. We are always ready to assist guests
12 "interaction": "Cot - 10 € / night Dog - € 7,5 / night",
13 "house_rules": "Make the house your home...",
14 "property_type": "House",
15 "room_type": "Entire home/apt",
16 "bed_type": "Real Bed",
17 "minimum_nights": "2",
18 "maximum_nights": "30",
19 "cancellation_policy": "moderate",
20 "last_scraped": /
```

sample_geospatial

```
localhost:27017 connected
> AGH
> admin
> config
> local
> sample_airbnb
> sample_geospatial
  > shipwrecks
    > Documents 11K
      "578f6fa2df35c7fbdba8c4"
      "578f6fa2df35c7fbdba8c4"
      "578f6fa2df35c7fbdba8c4"
      "578f6fa2df35c7fbdba8c7"
      "578f6fa2df35c7fbdba8c4"
      "578f6fa2df35c7fbdba8c4"
      "578f6fa2df35c7fbdba8ca"

1  {
2  "_id": {
3  "$oid": "578f6fa2df35c7fbdba8c4"
4  },
5  "recrd": "",
6  "vesselterms": "",
7  "feature_type": "Wrecks - Visible",
8  "chart": "US,U1,graph,DNC H1409860",
9  "latdec": 9.3547792,
10 "londec": -79.9081268,
11 "gp_quality": "",
12 "depth": "",
13 "sounding_type": "",
14 "history": "",
15 "quasou": "",
16 "watlev": "always dry",
17 "coordinates": [
18   -79.9081268,
19   9.3547792
20 ]
21 }
```

Podane bazy używają wielu typów prostych: string, number, boolean ale też typów bardziej skomplikowanych: obiekty, tablice. Jak widać między tabelami nie ma relacji, a pojedyncze rekordy są przedstawione jako obiekty JSowe.

.... Wyniki, zrzuty ekranów, kod, komentarz

5. Operacje CRUD

- <https://www.mongodb.com/docs/manual/crud/>
- stwórz nową bazę danych, jako nazwy bazy danych użyj swoich inicjałów
- stwórz kolekcję "student"
 - informacje o studentach, przedmiotach ocenach z przedmiotów itp.
 - zaproponuj strukturę dokumentu
 - wykorzystaj typy proste/podstawowe, dokumenty zagnieżdżone, tablice itp.
- wprowadź kilka przykładowych dokumentów
 - przetestuj operacje wstawiania, modyfikacji/usuwania dokumentów
- przetestuj operacje wyszukiwania dokumentów

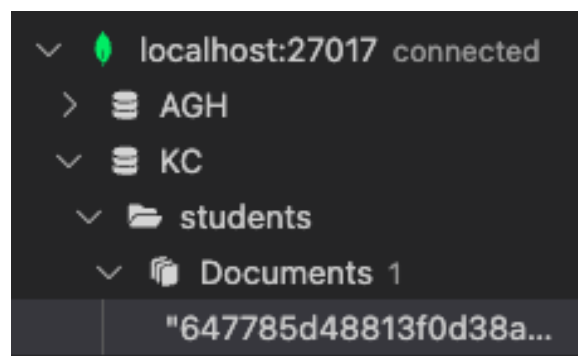
Tworzę bazę, kolekcję oraz dodaję dwa dokumenty.

```
1  const database = "KC"
2  const collection = 'students'
3
4  use (database);
5
6  db.createCollection (collection);
7
8  db.getCollection (collection).insertMany(
9  [{ name: "Jan",
10     surname: "Kowalski",
11     age: 15,
12     subjects: [
13       { name: "Biology",
14         ects: 3,
15         marks: [2, 6, 2, 4]
16       },
17       { name: "Physics",
18         ects: 5,
19         marks: [3, 5, 3, 2]
20       }
21     ],
22     { name: "Mateusz",
23       surname: "Nowak",
24       age: 16,
25       subjects: [
26         { name: "Mathematics",
27           ects: 3,
28           marks: [6, 4, 1, 1]
29         },
30         { name: "Physics",
31           ects: 5,
32           marks: [5, 2, 3, 2]
33         }
34       ]
35     }
36  ])
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure: 'localhost:27017 connected' -> 'AGH' -> 'KC' -> 'students' -> 'Documents 2'. Two documents are listed with their IDs. The main area shows the details of the selected document, which is the first one: { "_id": { "\$oid": "647785e067c30f2c5ff09508" }, "name": "Mateusz", "surname": "Nowak", "age": 16, "subjects": [{ "name": "Mathematics", "ects": 3, "marks": [6, 4, 1, 1] }, { "name": "Physics", "ects": 5, "marks": [5, 2, 3, 2] }] }. The right panel shows the details of the second document: { "_id": { "\$oid": "647785d48813f0d38afd3e78" }, "name": "Jan", "surname": "Kowalski", "age": 15, "subjects": [{ "name": "Biology", "ects": 3, "marks": [2, 6, 2, 4] }, { "name": "Physics", "ects": 5, "marks": [3, 5, 3, 2] }] }.

Usuwać jeden z dokumentów.

```
1  use('KC');
2
3  db.getCollection('students').deleteOne(
4    {
5      name: "Mateusz",
6      surname: "Nowak",
7      age: 16,
8      subjects: [
9        { name: "Mathematics",
10          ects: 3,
11          marks: [6, 4, 1, 1]
12        },
13        { name: "Physics",
14          ects: 5,
15          marks: [5, 2, 3, 2]
16        }
17      ],
18    }
19  );
```



Edytuję jeden z dokumentów.

```
1 use('KC');
2
3 db.collection('students').updateOne(
4   {
5     name: "Jan",
6     surname: "Kowalski",
7     age: 15,
8     subjects: [{
9       name: "Biology",
10      ects: 3,
11      marks: [2, 6, 2, 4]
12    },
13    {
14      name: "Physics",
15      ects: 5,
16      marks: [ 3, 5, 3, 2] }
17  ],
18  {
19    $set: {name: "Arkadiusz"}
20  }
21 );
```

```
1 {
2   "_id": {
3     "$oid": "647785d48813f0d38afd3e78"
4   },
5   "name": "Arkadiusz",
6   "surname": "Kowalski",
7   "age": 15,
8   "subjects": [
9     {
10      "name": "Biology",
11      "ects": 3,
12      "marks": [
13        2,
14        6,
15        2,
16        4
17      ]
18    },
19    {
20      "name": "Physics",
21      "ects": 5,
22      "marks": [
23        3,
24        5,
25        3,
26        2
27      ]
28    }
29  ]
30 }
```

Wyszukuję jeden z dokumentów.

```
1 use('KC');
2
3 db.getCollection('students').find({name:"Arkadiusz"})

2 {
3   "_id": {
4     "$oid": "647785d48813f0d38afd3e78"
5   },
6   "name": "Arkadiusz",
7   "surname": "Kowalski",
8   "age": 15,
9   "subjects": [
10    {
11      "name": "Biology",
12      "ects": 3,
13      "marks": [
14        2,
15        6,
16        2,
17        4
18      ]
19    },
20    {
21      "name": "Physics",
22      "ects": 5,
23      "marks": [
24        3,
25        5,
26        3,
27        2
28      ]
29    }
30  ]
31 }
32 }
```

.... Wyniki, zrzuty ekranów, kod, komentarz

W sprawozdaniu należy umieścić zrzuty ekranów (z kodem poleceń oraz z uzyskanymi wynikami).
Dodatkowo należy dołączyć plik tekstowy (najlepiej z rozszerzeniem .js) zawierający kod poleceń

Punktacja za zadanie (razem 1pkt)