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'Reily 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 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),
}
```

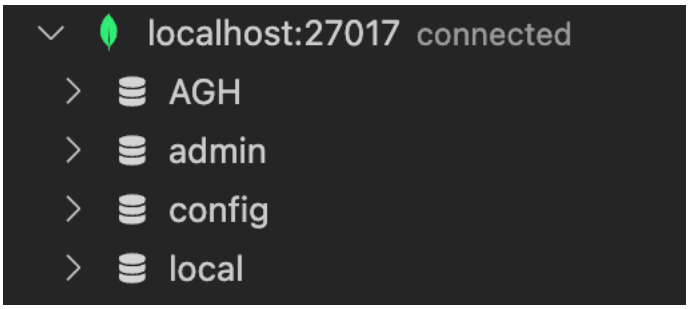
Narzędzia

- 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 mongsh
   - 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

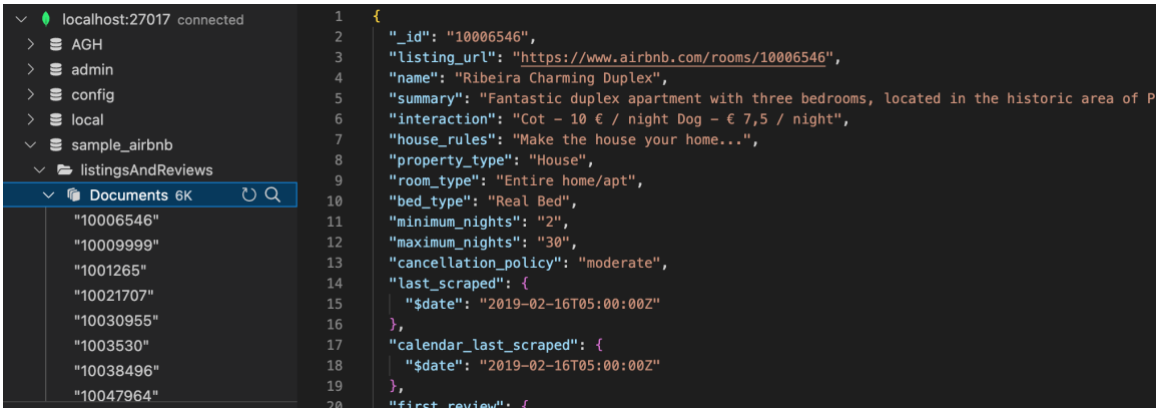
3. Zaimportuj przykładowe zbory danych
   - MongoDB Atlas Sample Dataset
     - https://docs.atlas.mongodb.com/sample-data
     - w przypadku importu z lokalnych plików można wykorzystać polecenie mongorestore
     - mongorestore <data-dump-folder>
     - np.
       `mongorestore samples`
       - Oczywiście, 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.

```
kacpercwiertnia@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)
2023-05-31T19:08:52.643+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)
2023-05-31T19:08:53.638+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)
2023-05-31T19:08:53.733+0200    restoring sample_supplies.sales from samples/sample_supplies/sales.bson
2023-05-31T19:08:53.748+0200    finished restoring sample_restaurants.restaurants (25359 documents, 0 failures)
2023-05-31T19:08:53.785+0200    restoring sample_geospatial.shipwrecks from samples/sample_geospatial/shipwrecks.bson
2023-05-31T19:08:53.799+0200    finished restoring sample_training.trips (10000 documents, 0 failures)
2023-05-31T19:08:53.829+0200    restoring sample_restaurants.neighborhoods from samples/sample_restaurants/neighborhoods.bson
2023-05-31T19:08:53.852+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)
```

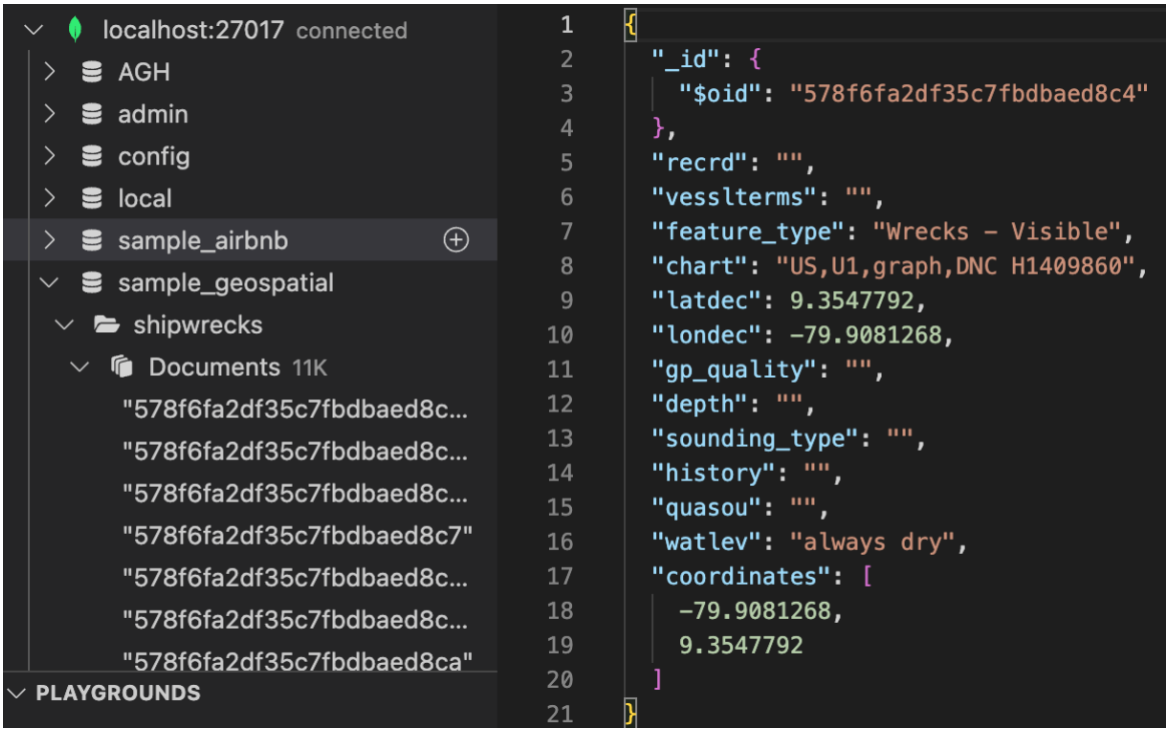


…. Wyniki, zrzuty ekranów, kod, komentarz

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 najcikawsze)
        - Np. Sample Analitics Dataset i Sampe Traning Dataset
    - W bazie Yelp
        - Skomentuj struktury użyte w dokumentach bazy Yelp

sample_airbnb

sample_geospatial



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

…. Wyniki, zrzuty ekranów, kod, komentarz

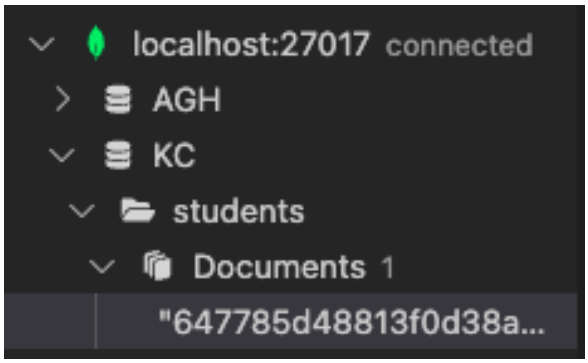
5. Operacje CRUD
   - https://www.mongodb.com/docs/manual/crud/
   - stwórz nową bazę danych, jako nazwy bazy danych użyj swoich inicjałów
   - stwórz kolekcję "student"
     - informacje o studentach, przedmiotach ocenach z przedmiotów itp.
     - zaproponuj strukturę dokumentu
     - wykorzystaj typy 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.

```
const database = "KC"
const collection = 'students'

use (database);

db.createCollection (collection);

db.getCollection (collection).insertMany(
[{  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]
    }]
 },
 {  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]
    }]
 }])
```

Usuwam jeden z dokumentów.

```
use('KC');

db.getCollection('students').deleteOne(
    {
        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]
            }],
});
```



Edytuję jeden z dokumentów.

```
use('KC');

db.getCollection('students').updateOne(
    {   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] }
                ]
    },
    {
        $set:{name: "Arkadiusz"}
    }
);
```


```
{
  "_id": {
    "$oid": "647785d48813f0d38afd3e78"
  },
  "name": "Arkadiusz",
  "surname": "Kowalski",
  "age": 15,
  "subjects": [
    {
      "name": "Biology",
      "ects": 3,
      "marks": [
        2,
        6,
        2,
        4
      ]
    },
    {
      "name": "Physics",
      "ects": 5,
      "marks": [
        3,
        5,
        3,
        2
      ]
    }
  ]
}
```

Wyszukuję jeden z dokumentów.

```
use('KC');

db.getCollection('students').find({name:"Arkadiusz"})
```

```
[
  {
    "_id": {
      "$oid": "647785d48813f0d38afd3e78"
    },
    "name": "Arkadiusz",
    "surname": "Kowalski",
    "age": 15,
    "subjects": [
      {
        "name": "Biology",
        "ects": 3,
        "marks": [
          2,
          6,
          2,
          4
        ]
      },
      {
        "name": "Physics",
        "ects": 5,
        "marks": [
          3,
          5,
          3,
          2
        ]
      }
    ]
  }
]
```

…. 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)