

Dokumentowe bazy danych – MongoDB

ćwiczenie 1

Imiona i nazwiska autorów:

1. Bartłomiej Szubiak
2. Szymon Kubiczek
3. Konrad Armatys

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:

przykłady, kod, zrzuty ekranów, komentarz ...

```
// stworzenie bazy danych
use students_db

// stworzenie kolekcji students
db.createCollection("students")

// wstawienie dokumentu do kolekcji
db.students.insertOne({imie: "Jan", nazwisko: "Kowalski"})
```

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

```
mongoimport <data-dump-folder>
```

np.

```
mongoimport 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

baza sample_mflix

- movies:
 - zawiera bardzo duży zagnieżdżony obiekt tomatoes
 - zagnieżdżone obiekty mają również swoje podobiekty
- embedded_movies:
 - bardzo długa tabela
 - W relacyjnych bazach danych unikałoby się tabel tej długości
- theaters: -zagnieżdżony obiekt stanowi większość zawartości tej kolekcji
- comments:
 - przypomina tabelę SQL
- users:
 - również przypomina tabelę SQL

baza sample_airbnb:

- listingsAndReviews
 - kolekcja składa się głównie z jednego obiektu

baza Yelp:

- business:
 - niektóre pola są niezależnymi obiektami
 - w SQLu należałoby podzielić tę kolekcję na kilka mniejszych tabel
- checkin:
 - posiada dużo pól z informacjami
 - w SQLu byłyby to 2 tabele z relacją jeden do wielu
- review:
 - poza tym że zawiera obiekt przypomina tabelę SQLową
- tip:
 - wygląda identycznie jak tabela z SQLa
- user
 - zawiera duże zagnieżdżone obiekty

mongodump: Narzędzie do tworzenia kopii zapasowych danych z bazy danych MongoDB w formacie BSON.

mongoexport: Narzędzie do eksportowania danych z bazy danych MongoDB do plików JSON lub CSV.

do wykonania zadania 2 użyliśmy komend podanych w poleceniu

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

Zakładamy bazę danych `BS_orders` z kolekcjami `customers`, `products`, `orders`

- kolekcja **customers** przechowuje dane klientów w postaci dokumentów z polami: **name: str**, **address: AdressObj**, **email: str**, gdzie address to obiekt z polami: **street: str**, **zip_code:str**, **city:str**
- kolekcja **products** przechowuje dane produktów w postaci dokumentów z polami: **name:str**, **price:number**, **category:str**
- kolekcja **orders** przechowuje dane zamówień w postaci dokumentów z polami: **customerId: ObjectId**, **products: [ObjectId]**, **total:number**

```
// Stworzenie nowej bazy danych
use BS_orders

// Stworzenie kolekcji 'customers'
db.createCollection("customers")

// Stworzenie kolekcji 'products'
db.createCollection("products")

// Stworzenie kolekcji 'orders'
db.createCollection("orders")

// Wstawienie przykładowych dokumentów do kolekcji 'customers'
db.customers.insertMany([
  { name: "Jan Kowalski", address: {street: "Kwiatowa 5" , zip_code:"00-000", city:"Warszawa"}, email: "jan.kowalski@example.com" },
  { name: "Anna Nowak", address: {street: "Maja 5" , zip_code:"01-250", city:"Warszawa"}, email: "anna.nowak@example.com" }
])

// Wstawienie przykładowych dokumentów do kolekcji 'products'
db.products.insertMany([
  { name: "Apple", price: 10, category: "fruit" },
  { name: "Tomato", price: 20, category: "vegetable" },
  { name: "Carrot", price: 5, category: "vegetable" },
  { name: "Orange", price: 15, category: "fruit" }
])

// Wstawienie przykładowych dokumentów do kolekcji 'orders'
db.orders.insertMany([
  { customerId: db.customers.findOne({ name: "Jan Kowalski" })._id, products: [db.products.findOne({ name: "Apple" })._id], total: 100 },
  { customerId: db.customers.findOne({ name: "Anna Nowak" })._id, products: [db.products.findOne({ name: "Tomato" })._id], total: 200 },
  { customerId: db.customers.findOne({ name: "Anna Nowak" })._id, products: [db.products.findOne({ name: "Tomato" })._id ,
db.products.findOne({name: "Orange"})._id]}
])

// Przetestowanie operacji modyfikacji dokumentów
db.customers.updateOne({ name: "Jan Kowalski" }, { $set: { email: "new_email@email.pl" } })
db.products.updateOne({ name: "Apple" }, { $set: { price: 150 } })

// Przetestowanie operacji usuwania dokumentów
db.customers.deleteOne({ name: "Anna Nowak" })
db.products.deleteOne({ name: "Tomato" })

// Przetestowanie operacji wyszukiwania dokumentów
db.customers.find({ name: "Jan Kowalski" })
db.products.find({ price: { $gt: 15 } })
db.orders.find({ total: { $lt: 10 } })
```

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