

Laboratorium 3

MongoDB

Mateusz Łopaciński

1. Zadanie 7

1.1. Tworzenie bazy danych i kolekcji

```
use ml
db.createCollection('student')
```

1.2. Struktura dokumentów

Dokumenty wstawiane pojedynczo:

```
student1 = {
  firstName: "Mateo",
  lastName: "Turner",
  sex: "m",
  email: "mateo@email.com",
  phone: "345345345",
  courses: [
    "Computer Science"
  ],
  subjects: {
    Databases: {
      grades: [5, 4.5, 5]
    },
    Java: {
      grades: [4, 3, 4.5]
    },
    Python: {
      grades: [5, 3.5, 4.5]
    },
  },
},
}

student2 = {
  firstName: "Lucious",
  lastName: "Fisher",
  sex: "m",
  email: "lucious@email.com",
  phone: "123454321",
  courses: [
    "Computer Science",
    "Ceramics"
  ],
  subjects: {
    ASD: {
      "grades": [3, 3, 2]
    },
    Kotlin: {
      "grades": [4, 3, 4.5]
    },
    JavaScript: {
      "grades": [4.5, 3.5, 5]
    },
    Pottery: {
      "grades": [5, 5, 5, 5, 5]
    },
  },
},
}
```

Lista dokumentów wstawianych równocześnie:

```
students = [
  {
    firstName: "Eusebio",
    lastName: "Gorczany",
    sex: "m",
    email: "eusebio@email.com",
    phone: "111222111",
    courses: [
      "Robotics"
    ],
    subjects: {
      Programming: {
        grades: [3, 4, 5, 4.5]
      },
      Physics: {
        grades: [3, 3, 3, 4]
      },
      Algebra: {
        grades: [3, 4]
      },
    },
  },
  {
    firstName: "Gracie",
    lastName: "Reichert",
    sex: "f",
    email: "gracie@email.com",
    phone: "909890989",
    courses: [
      "Computer Science"
    ],
    subjects: {
      Databases: {
        grades: [4, 5, 4, 5]
      },
      Cpp: {
        grades: [5, 5, 5, 5]
      },
    },
  },
  {
    firstName: "Elaina",
    lastName: "Wyman",
    sex: "f",
    email: "elaina@email.com",
    phone: "794794794",
    courses: [
      "Geodesy"
    ],
    subjects: {
      SoilScience: {
        grades: [5]
      },
      Mining: {
        grades: [3, 3, 4, 5]
      },
    },
  },
]
```

1.3. Wstawianie dokumentów do kolekcji

Wstawianie dokumentów pojedynczo:

```
db.student.insertOne(student1)
db.student.insertOne(student2)
```

Wstawianie kilku dokumentów równocześnie:

```
db.student.insertMany(students)
```

1.4. Wyszukiwanie dokumentów

Przykładowe wyszukiwanie dokumentów:

`db.student.find()` - zwróci wszystkie dokumenty, które znajdują się w kolekcji *student*,

```
db.student.find({ age: { $gt: 23 } }).pretty()
{
  "_id": ObjectId("626306a4fd695fbd7ad85fa2"),
  "firstName": "Elaina",
  "lastName": "Wyman",
  "sex": "f",
  "email": "elaina@email.com",
  "phone": "794794794",
  "courses": [
    "Geodesy"
  ],
  "subjects": {
    "SoilScience": {
      "grades": [
        5
      ]
    },
    "Mining": {
      "grades": [
        3,
        3,
        4,
        5
      ]
    }
  }
}
```

```
db.student.find({ age: { $gt: 23 } }, { firstName: 1 })
{ "_id": ObjectId("626306a4fd695fbd7ad85fa2"), "firstName": "Elaina" }
```

```
db.student.find({}, { firstName: 1, lastName: 1, yearOfStudies: 1 })
{ "_id": ObjectId("62630694fd695fbd7ad85f9e"), "firstName": "Mateo", "lastName": "Turner" }
{ "_id": ObjectId("62630699fd695fbd7ad85f9f"), "firstName": "Lucious", "lastName": "Fisher" }
{ "_id": ObjectId("626306a4fd695fbd7ad85fa0"), "firstName": "Eusebio", "lastName": "Gorczyński" }
{ "_id": ObjectId("626306a4fd695fbd7ad85fa1"), "firstName": "Gracie", "lastName": "Reichert" }
{ "_id": ObjectId("626306a4fd695fbd7ad85fa2"), "firstName": "Elaina", "lastName": "Wyman" }
```

1.5. Modyfikowanie i usuwanie dokumentów

Przykładowe modyfikowanie dokumentów:

```
db.student.update({ _id: ObjectId("62637c72659c2d470027cc32") }, { $set: { phone: "123123123" } })
db.student.find({ phone: "123123123" }, { _id: 1 })
  { "_id" : ObjectId("62637c72659c2d470027cc32") }

db.student.updateMany({}, { $unset: { phone: 1 } })
  { "_id" : ObjectId("62637c3f659c2d470027cc31") }
  { "_id" : ObjectId("62637c72659c2d470027cc32") }
  { "_id" : ObjectId("62637c80659c2d470027cc33") }
  { "_id" : ObjectId("62637c80659c2d470027cc34") }
  { "_id" : ObjectId("62637c80659c2d470027cc35") }

db.student.updateMany({}, { $set: { courseRetakeFees: 0 } })
db.student.updateMany({}, { $inc: { courseRetakeFees: 100 } })
db.student.find({}, { _id: 1, courseRetakeFees: 1 })
  { "_id" : ObjectId("62637c3f659c2d470027cc31"), "courseRetakeFees" : 100 }
  { "_id" : ObjectId("62637c72659c2d470027cc32"), "courseRetakeFees" : 100 }
  { "_id" : ObjectId("62637c80659c2d470027cc33"), "courseRetakeFees" : 100 }
  { "_id" : ObjectId("62637c80659c2d470027cc34"), "courseRetakeFees" : 100 }
  { "_id" : ObjectId("62637c80659c2d470027cc35"), "courseRetakeFees" : 100 }

db.student.update({ _id: ObjectId("62637c72659c2d470027cc32") }, { $pull: { courses: "Ceramics" } })
db.student.update({ _id: ObjectId("62637c72659c2d470027cc32") }, { $push: { courses: "Bricklayer Plasterer Acrobat" } })
db.student.find({ courses: "Bricklayer Plasterer Acrobat" }, { _id: 1, courses: 1 }).pretty()
  {
    "_id" : ObjectId("62637c72659c2d470027cc32"),
    "courses" : [
      "Computer Science",
      "Bricklayer Plasterer Acrobat"
    ]
  }
}
```

Przykładowe usuwanie dokumentów:

```
db.student.deleteOne({ _id: ObjectId("62637c80659c2d470027cc34") })
  { "_id" : ObjectId("62637c3f659c2d470027cc31") }
  { "_id" : ObjectId("62637c72659c2d470027cc32") }
  { "_id" : ObjectId("62637c80659c2d470027cc33") }
  { "_id" : ObjectId("62637c80659c2d470027cc35") }

db.student.find().count()
  4
db.student.deleteMany({ sex: 'm' })
db.student.find().count()
  1
```

1.6. Usuwanie bazy danych

```
db.dropDatabase()
```

2. Zadanie 8

W tym zadaniu skorzystałem ze zbioru danych Yelp Academic Dataset.

a) Zwróć dane wszystkich zamkniętych (*open*) firm (*business*). Zapytanie powinno zwracać dane z pól: nazwa, adres, gwiazdki (*stars*).

```
db.business.find({is_open: 0}, {name: 1, address: 1, stars: 1}).pretty()

{
  "_id" : ObjectId("62639f1e98daa6d2dc108d07"),
  "name" : "Target",
  "address" : "5255 E Broadway Blvd",
  "stars" : 3.5
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d09"),
  "name" : "Tsevi's Pub And Grill",
  "address" : "8025 Mackenzie Rd",
  "stars" : 3
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d0f"),
  "name" : "Zio's Italian Market",
  "address" : "2575 E Bay Dr",
  "stars" : 4.5
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d11"),
  "name" : "Abby Rappoport, LAC, CMQ",
  "address" : "1616 Chapala St, Ste 2",
  "stars" : 5
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d15"),
  "name" : "Super Dog",
  "address" : "1160 Gallatin Pike S",
  "stars" : 4
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d1a"),
  "name" : "The Green Pheasant",
  "address" : "215 1st Ave S",
  "stars" : 4
}
{
  "_id" : ObjectId("62639f1e98daa6d2dc108d1d"),
  "name" : "Bar One",
  "address" : "767 S 9th St",
```

```

    "stars" : 4
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d22"),
    "name" : "DeSandro on Main",
    "address" : "4105 Main St",
    "stars" : 3
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d24"),
    "name" : "Cheeseburger In Paradise",
    "address" : "116 N Pottstown Pike",
    "stars" : 2.5
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d28"),
    "name" : "Roast Coffeehouse and Wine Bar",
    "address" : "10359 104 Street NW",
    "stars" : 4
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d2e"),
    "name" : "China Dragon Restaurant",
    "address" : "1625 W Valencia Rd, Ste 101-103",
    "stars" : 3
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d33"),
    "name" : "Gold's Gym",
    "address" : "203 - 38th Ave N",
    "stars" : 3
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d37"),
    "name" : "Caviar & Bananas",
    "address" : "2031 Broadway",
    "stars" : 3.5
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d39"),
    "name" : "Paws The Cat Cafe",
    "address" : "10588 109 Street",
    "stars" : 5
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d3a"),
    "name" : "Pier 1",
    "address" : "1014 Baltimore Pike",
    "stars" : 2.5
  }
  {

```

```

    "_id" : ObjectId("62639f1e98daa6d2dc108d3d"),
    "name" : "Joe's Pizza",
    "address" : "2038 N Dale Mabry Hwy",
    "stars" : 4
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d3e"),
    "name" : "Roman Forum",
    "address" : "10440 N Dale Mabry Hwy",
    "stars" : 4
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d41"),
    "name" : "Village Tap Room",
    "address" : "838 Broad Ripple Ave",
    "stars" : 2.5
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d42"),
    "name" : "Doc Magrogan's Oyster House - West Chester",
    "address" : "117 E Gay St",
    "stars" : 3
  }
  {
    "_id" : ObjectId("62639f1e98daa6d2dc108d46"),
    "name" : "Publix Super Market",
    "address" : "12101 Little Rd",
    "stars" : 3.5
  }
}

```

b) Ile miejsc ocenianych na 5 gwiazdek (pole *stars*, kolekcja *business*).

```

db.business.find({stars: 5}).count()

16307

```

d) Zwróć bez powtórzeń wszystkie nazwy miast w których znajdują się firmy (*business*)

```

db.business.distinct('city')

[
  ...
  "riverview",
  "saint ann",
  "saint petersburg",
  "santa Barbara",
  "sewell",
  "sparks",
  "spring city",
  "spring hill",
  "tampa",
  "telford",

```



```

        "tucson",
        "wesley chapel",
        "wilmington",
        "wimauma",
        "Clayton",
        "Largo",
        "Lithia"
    ]

```

Ponieważ w wyniku otrzymujemy kolekcję składającą się z 1416 miast, powyżej umieściłem tylko końcowy fragment wyniku. Liczbę miast sprawdziłem, przy pomocy poniższego polecenia:

```

db.business.distinct('city').length

1416

```

e) Ile każda firma otrzymała wskazówek (*tip*) w 2012. Wynik posortuj według liczby wskazówek (*tip*).

```

db.tip.aggregate([
    {$match: {date: /2012/}},
    {$group: {_id: '$business_id', count: {$sum: 1}}},
    {$sort: {count: 1}}
])

{ "_id": "aWbhsVKavSCpxalbJkTaUQ", "count": 1 }
{ "_id": "0KULrKCxl5RbRwTlylyHEw", "count": 1 }
{ "_id": "642lgLN1a8kjDleAkaps1A", "count": 1 }
{ "_id": "PzuFGj34vjWoLlo3uNeRRA", "count": 1 }
{ "_id": "QpgszBy9G1KqXt6QHwQEAQ", "count": 1 }
{ "_id": "Q8Y3g7GEHNSBfc_5Kr7hug", "count": 1 }
{ "_id": "snVWzHOfJF-q4PEI7dmAqg", "count": 1 }
{ "_id": "jfpjVab1JSCQQQyBpQ7aAw", "count": 1 }
{ "_id": "2Sr44BJmYJ4PDMyoW0cePw", "count": 1 }
{ "_id": "RfKB6YGI0YUrt_ECXr7lxA", "count": 1 }
{ "_id": "uPQVKBptNqOxiYfXQyFxA", "count": 1 }
{ "_id": "BdHadrVCRSsBaTYrBUpBgQ", "count": 1 }
{ "_id": "_ozCIOzZtluaewuMYI_OqQ", "count": 1 }
{ "_id": "h7ICnU21LboMGoqTj3gAzg", "count": 1 }
{ "_id": "xXdZl-e8OWZdpsLsgL5aqq", "count": 1 }
{ "_id": "5VrMyTPhmRTgnjldrpMuvA", "count": 1 }
{ "_id": "CAnV18mOhhpQYIOqF40-Dw", "count": 1 }
{ "_id": "uiBnQv48kXMcqMNtfEC2kA", "count": 1 }
{ "_id": "YvBGiNf-IPJg68RyKBsq2g", "count": 1 }
{ "_id": "-ee5czr21mfEJGrbZ-572Q", "count": 1 }

```

Ponieważ w wyniku uzyskujemy kolekcję, składającą się z 28477 elementów, powyżej umieściłem tylko początkowy fragment wynikowej kolekcji. Liczbę elementów wynikowej kolekcji sprawdziłem, przy pomocy poniższego polecenia:

```

db.tip.aggregate([
    {$match: {date: /2012/}},
    {$group: {_id: '$business_id', count: {$sum: 1}}},
    {$sort: {count: 1}}
]).toArray().length

28477

```

3. Zadanie 9

Modelowanie danych przeprowadziłem dla zagadnienia B (*wycieczki*).

3.1. Sposoby modelowania danych

3.1.1. Podejście tabelaryczne (*referencing*)

▪ Zalety

- Większa wydajność podczas robienia zapytań do każdego z dokumentów z osobna (dobre podejście, gdy często interesują nas jedynie poszczególne dokumenty, a nie cały dokument wraz z zagnieżdżonymi dokumentami),
- Pozwala na uporządkowanie dokumentów w bardziej przejrzysty sposób (pogrupowanie w kolekcje, odpowiadające różnym kategoriom dokumentów),
- Pozwala zachować mniejszy rozmiar dokumentów, co jest szczególnie przydatne, gdy mamy bardzo dużo danych w bazie i istnieje ryzyko przekroczenia limitu (16MB) wielkości pojedynczego dokumentu,
- Pozwala na szybsze aktualizacje danych, zawartych w dokumentach (jest to lepsze podejście od zagnieżdżania dokumentów w sytuacji, gdy dużo częściej zapisujemy dane do bazy danych niż je odczytujemy)

▪ Wady

- Konieczność łączenia danych z dokumentów znajdujących się w różnych kolekcjach, w sytuacji, w której potrzebujemy dokumentu wraz z dokumentami, które są z nim w relacji (dokł. z dokumentami, do których referencje posiada ten dokument),
- Konieczność tworzenia skomplikowanych zapytań, jeżeli na wynikowy dokument składa się wiele dokumentów z różnych kolekcji,
- Dłuższy czas odczytu dokumentu, wymagającego złączenia dokumentów z kilku kolekcji (podejście tabelaryczne często jest gorszym wyborem, gdy bardzo często odczytujemy dane z bazy danych)

▪ Przykład

W zamieszczonym niżej fragmencie dokumentu, widzimy, że atrybuty `company` oraz `guides` zawierają referencje do innych dokumentów w bazie danych. Firma (`company`), a także każdy z przewodników (`guides`) jest osobnym dokumentem w odpowiedniej kolekcji (`companies` lub `people`), a taki podział przypomina rozmieszczenie danych podobne do grupowania danych w tabelę.

```
{
  "_id": {
    "$oid": "5c88fa8cf4afda39709c2955"
  },
  "company": {
    "$oid": "6263e1123ca1655ba6a4dd8f"
  },
  "name": "Mountains Hiking",
  ...
  "guides": [
    {
      "$oid": "5c8a22c62f8fb814b56fa18b"
    },
    {
      "$oid": "5c8a1f4e2f8fb814b56fa185"
    }
  ]
}
```

3.1.2. Podejście dokumentowe (*embedding*)

▪ Zalety

- Większa wydajność podczas gdy potrzebujemy całego dokumentu wraz z zagnieżdżonymi w nim dokumentami (wystarczy jedno zapytanie bez konieczności łączenia dokumentów z kilku kolekcji),
- Pozwala na łatwiejsze zdefiniowanie relacji między dokumentami, poprzez zagnieżdżanie dokumentów, będących w relacji,
- Pozwala na szybki odczyt danych o całym dokumencie (dobre podejście, jeżeli więcej czytamy z bazy danych niż do niej zapisujemy)

▪ Wady

- Brak możliwości tworzenia zapytań, które bezpośrednio dotyczą zagnieżdżonych dokumentów (trzeba „dostać się” do nich poprzez dokument, w którym są zagnieżdżone),
- Konieczność tworzenia bardziej skomplikowanych zapytań, gdy potrzebujemy odwołać się do głęboko zagnieżdżonego dokumentu,
- Bardziej długotrwałe i skomplikowane niż w podejściu tabelarycznym aktualizacje dokumentów (wymagane jest przejście przez kolejno zagnieżdżone dokumenty)

▪ Przykład

Poniższy fragment dokumentu, przechowującego dane jednej z wycieczek, prezentuje podejście dokumentowe. Widzimy, że do atrybutu `locations` przypisana jest lista obiektów.

```
{
  "_id": {
    "$oid": "5c88fa8cf4afda39709c2955"
  },
  ...
  "name": "Mountains Hiking",
  "duration": 4,
  ...
  "locations": [
    {
      "description": "Aspen Highlands",
      "day": 1
    },
    {
      "description": "Beaver Creek",
      "day": 3
    }
  ]
  ...
}
```

3.2. Kolekcje oraz przykładowe dokumenty

Dla każdej z kolekcji zamieściłem niżej po jednym przykładowym dokumencie.

3.2.1. Wycieczki (*tours*)

Dokument, przechowujący dane dotyczące wycieczki, zawiera następujące pola:

- `_id` – id wycieczki w bazie danych,
- `company` – id firmy, która jest organizatorem danej wycieczki,
- `name` – nazwa wycieczki,
- `duration` – czas trwania wycieczki,
- `maxGroupSize` – maksymalna liczebność grupy osób, która może pojechać na wycieczkę,

- difficulty – poziom trudności wycieczki,
- ratingsAverage – średnia ocena wycieczki,
- ratingsQuantity – liczba recenzji (ocen) wycieczki,
- price – cena wycieczki,
- summary – skrótowne podsumowanie wycieczki,
- description – szczegółowy opis przebiegu wycieczki,
- createdAt – data dodania wycieczki do bazy danych,
- startDates – daty rozpoczęcia wycieczki (daty wyjazdu),
- locations – kolejne lokalizacje odwiedzane w trakcie wycieczki,
- guides – lista przewodników

```
{
  "_id": {
    "$oid": "5c88fa8cf4afda39709c2955"
  },
  "company": {
    "$oid": "6263e1123ca1655ba6a4dd8f"
  },
  "name": "Mountains Hiking",
  "duration": 4,
  "maxGroupSize": 10,
  "difficulty": "difficult",
  "ratingsAverage": 4.83,
  "ratingsQuantity": 6,
  "price": 999,
  "summary": "Extreme mountain climbing with magnificent views and loads of
adrenaline",
  "description": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorem
qui dolore earum sed quo obcaecati enim unde tenetur, odio dolor dignissimos vitae
labore et iste dolores nesciunt debitis aperiam assumenda aut voluptatem
reprehenderit blanditiis libero? Veritatis, ullam. Dolorem facere nobis
cupiditate, eveniet voluptatum voluptates minus libero iusto placeat, reiciendis
odio!",
  "createdAt": {
    "$date": "2022-01-29T01:03:28.180Z"
  },
  "startDates": [
    {
      "$date": "2022-12-12T12:35:03.795Z"
    },
    {
      "$date": "2022-12-28T05:52:32.724Z"
    }
  ],
  "locations": [
    {
      "description": "Aspen Highlands",
      "day": 1
    },
    {
      "description": "Beaver Creek",
      "day": 3
    }
  ],
  "guides": [
    {
      "$oid": "5c8a22c62f8fb814b56fa18b"
    },
    {
      "$oid": "5c8a1f4e2f8fb814b56fa185"
    }
  ]
}
```

3.2.2. Firmy (*companies*)

Dokument, przechowujący dane dotyczące firmy, będącej organizatorem niektórych wycieczek:

- `_id` – id firmy w bazie danych,
- `name` – nazwa firmy,
- `tours` – lista wycieczek organizowanych przez firmę

```
{
  "_id": {
    "$oid": "6263e1123ca1655ba6a4dd8f"
  },
  "name": "Travellio",
  "tours": [
    {
      "$oid": "5c88fa8cf4afda39709c2955"
    },
    {
      "$oid": "5c88fa8cf4afda39709c295d"
    }
  ]
}
```

3.2.3. Osoby (*people*)

Dokument, przechowujący dane dotyczące osoby (uczestnika wycieczki lub przewodnika):

- `_id` – id osoby w bazie danych,
- `name` – imię i nazwisko osoby,
- `email` – adres email osoby,
- `role` – rola (uczestnik wycieczki / przewodnik),
- `active` – informacja, czy konto użytkownika jest aktywne

```
{
  "_id": {
    "$oid": "5c8a22c62f8fb814b56fa18b"
  },
  "name": "Miyah Myles",
  "email": "miyah@example.com",
  "role": "lead-guide",
  "active": true
}
```

3.2.4. Recenzje (*reviews*)

Dokument, przechowujący dane dotyczące recenzji wycieczek:

- `_id` – id recenzji w bazie danych,
- `review` – treść recenzji,
- `rating` – ocena wycieczki,
- `user` – id użytkownika, wystawiającego daną recenzję,
- `tour` – id ocenianej wycieczki

```
{
  "_id": {
    "$oid": "5c8a34ed14eb5c17645c9108"
  },
  "review": "Cras mollis nisi parturient mi nec aliquet suspendisse sagittis eros condimentum scelerisque taciti mattis praesent feugiat eu nascetur a tincidunt",
  "rating": 5,
  "user": {
```

```

    "$oid": "5c8a1dfa2f8fb814b56fa181"
  },
  "tour": {
    "$oid": "5c88fa8cf4afda39709c2955"
  }
}

```

3.3. Przykładowe operacje na danych

3.3.1. Pobieranie statystyk wycieczek, których średnia ocena wynosi przynajmniej 4.6

Statystyki zawierają informacje o:

- liczbie wycieczek spełniających kryteria,
- łącznej liczbie ocen wszystkich wycieczek spełniających kryteria,
- średniej ocenie tych wycieczek,
- minimalnej cenie wycieczki spośród wycieczek spełniających kryteria,
- maksymalnej cenie wycieczki spośród wycieczek spełniających kryteria

(kryterium jest jedynie ocena wycieczki (`{$match: {ratingsAverage: {$gte: 4.6}}}`))

```

db.tours.aggregate([
  {$match: {ratingsAverage: {$gte: 4.6}}},
  {
    $group: {
      _id: null,
      noTours: {$sum: 1},
      noRatings: {$sum: '$ratingsQuantity'},
      avgRating: {$avg: '$ratingsAverage'},
      avgPrice: {$avg: '$price'},
      minPrice: {$min: '$price'},
      maxPrice: {$max: '$price'},
    }
  }
])
.pretty()
{
  "_id": null,
  "noTours": 2,
  "noRatings": 11,
  "avgRating": 4.715,
  "avgPrice": 1249,
  "minPrice": 999,
  "maxPrice": 1499
}

```

3.3.2. Pobieranie listy wycieczek, które rozpoczną się we wakacje

```

db.tours.aggregate([
  {
    $match: {
      startDate: {
        $gte: new Date('2022-07-01'),
        $lte: new Date('2022-08-31')
      }
    }
  },
  {
    $project: {
      name: 1,

```

```

        startDates: 1
    }
}
1).pretty()
{
  "_id" : ObjectId("5c88fa8cf4afda39709c295a"),
  "name" : "Pleasurable Sunbathing",
  "startDates" : [
    ISODate("2022-06-11T12:35:03.795Z"),
    ISODate("2022-07-21T05:52:32.724Z")
  ]
}
{
  "_id" : ObjectId("5c88fa8cf4afda39709c295d"),
  "name" : "Under the sea",
  "startDates" : [
    ISODate("2022-05-14T12:35:03.795Z"),
    ISODate("2022-07-18T05:52:32.724Z")
  ]
}

```

3.3.3. Sortowanie użytkowników malejąco według liczby wystawionych recenzji

```

db.reviews.aggregate([
  {
    $group: {
      _id: "$user",
      count: { $sum: 1 }
    }
  },
  {
    $sort: { count: -1 }
  }
])

```

3.4. Zrzut bazy danych

W przesłanym osobno pliku .zip znajduje się zrzut bazy danych wykonanej w ramach tego zadania.