

# Dokumentowe bazy danych – MongoDB

## Ćwiczenie 2 - zadanie do samodzielnego wykonania

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>

Yelp Dataset

[www.yelp.com](http://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),
}
```

## Zadania

### 1. Operacje wyszukiwania danych

Dla zbioru Yelp wykonaj następujące zapytania

W niektórych przypadkach może być potrzebne wykorzystanie mechanizmu Aggregation Pipeline

<https://www.mongodb.com/docs/manual/core/aggregation-pipeline/>

- a) Zwróć dane wszystkich restauracji (kolekcja `business`, pole `categories` musi zawierać wartość `Restaurants`), które są otwarte w poniedziałki (pole `hours`) i mają ocenę co najmniej 4 gwiazdki (pole `stars`). Zapytanie powinno zwracać: nazwę firmy, adres, kategorię, godziny otwarcia i gwiazdki. Posortuj wynik wg nazwy firmy.

Wybieram z kolekcji `business` wszystkie takie restauracje, które spełniają powyższe wymagania oraz sortuję je po nazwie.

```
1  use('KC');
2
3  db.getCollection('business')
4    .find(
5      {
6        "hours.Monday": {$exists: true},
7        categories: "Restaurants",
8        stars: {$gte: 4}
9      },
10     {
11       _id: 0,
12       name: 1,
13       full_address: 1,
14       categories: 1,
15       hours: 1,
16       stars: 1
17     }
18   )
19   .sort({
20     ⚡ | name: 1
21   });
```

```

1  [
2    {
3      "full_address": "67 Nicolson Street\nNewington\nEdinburgh EH8 9BZ",
4      "hours": {
5        "Monday": {
6          "close": "22:00",
7          "open": "10:00"
8        },
9        "Tuesday": {
10         "close": "22:00",
11         "open": "10:00"
12       },
13       "Friday": {
14         "close": "22:00",
15         "open": "10:00"
16       },
17       "Wednesday": {
18         "close": "22:00",
19         "open": "10:00"
20       },
21       "Thursday": {
22         "close": "22:00",
23         "open": "10:00"
24       },
25       "Sunday": {
26         "close": "22:00",
27         "open": "10:00"
28       },
29       "Saturday": {
30         "close": "22:00",
31         "open": "10:00"
32       }
33     },
34     "categories": [
35       "Food",
36       "Desserts",
37       "Coffee & Tea",
38       "Indian",
39       "Restaurants"
40     ],
41     "name": "10-to-10 In Delhi",
42     "stars": 4.5
43   },
44   {
45     "full_address": "4180 S Jones Blvd\nLas Vegas, NV 89103",
46     "hours": {
47       "Monday": {
48         "close": "22:00",
49         "open": "11:00"
50       },
51       "Friday": {
52         "close": "22:00",
53         "open": "11:00"
54       },
55       "Wednesday": {
56         "close": "22:00",
57         "open": "11:00"
58       },
59       "Thursday": {
60         "close": "22:00",
61         "open": "11:00"
62       },

```

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

- b) Ile hoteli znajduje się w każdym mieście. (pole *categories* musi zawierać wartość *Hotels & Travel* lub *Hotels*).  
Wynik powinien zawierać nazwę miasta, oraz liczbę hoteli. Posortuj wynik malejąco wg liczby hoteli.

Wybieram wszystkie takie dokumenty, które spełniają powyższy warunek, a następnie je grupuję według miasta.

```
1  use('KC');
2
3  db.getCollection('business')
4    .aggregate([
5      {$match:{
6        categories: { $in: ["Hotels & Travel", "Hotels"]},
7      }},
8      {$group:{
9        _id: "$city",
10       numOfHotels: { $count: {} }
11      }}
12    ])
13    .sort({
14      numOfHotels: -1
15    })
```

```
1  [
2    {
3      "_id": "Las Vegas",
4      "numOfHotels": 485
5    },
6    {
7      "_id": "Phoenix",
8      "numOfHotels": 250
9    },
10   {
11     "_id": "Edinburgh",
12     "numOfHotels": 161
13   },
14   {
15     "_id": "Scottsdale",
16     "numOfHotels": 122
17   },
18   {
19     "_id": "Madison",
20     "numOfHotels": 67
21   },
22   {
23     "_id": "Tempe",
24     "numOfHotels": 57
25   },
26   {
27     "_id": "Mesa",
28     "numOfHotels": 53
29   },
30   {
31     "_id": "Henderson",
32     "numOfHotels": 41
33   },
34   {
35     "_id": "Chandler",
36     "numOfHotels": 30
37   },
38   {
39     "_id": "Glendale",
40     "numOfHotels": 20
41   },
42   {
43     "_id": "North Las Vegas",
44     "numOfHotels": 12
45   },
46   {
47     "_id": "Peoria",
48     "numOfHotels": 12
49   },
50   {
51     "_id": "Surprise",
52     "numOfHotels": 10
53   },
54   {
55     "_id": "Goodyear",
56     "numOfHotels": 9
57   },
58   {
59     "_id": "Middleton",
60     "numOfHotels": 7
61   },
62 ]
```

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

- c) Ile każda firma otrzymała ocen/wskazówek (kolekcja *tip* ) w 2012. Wynik powinien zawierać nazwę firmy oraz liczbę ocen/wskazówek Wynik posortuj według liczby wskazówek (*tip*).

Dla każdego *business\_id* wyliczam ilość *tip*ów z 2012, sortuję i dopasowuję do nazwy firmy.

```
50 db.getCollection('tip')
51   .aggregate([
52     {$match: {
53       date: { $regex: '2012'}
54     }},
55     {$group: {
56       _id: "$business_id",
57       numOfTips: {$count: {}}
58     }},
59     {$sort: {
60       numOfTips: -1
61     }},
62     {$lookup: {
63       from: "business",
64       localField: "_id",
65       foreignField: "business_id",
66       as: "business_info"
67     }},
68     {$project: {
69       _id: 0,
70       name: {$arrayElemAt: ["$business_info.name", 0]},
71       numOfTips: 1
72     }}
73   ])
```



```
1  [
2    {
3      "numOfTips": 1084,
4      "name": "McCarran International Airport"
5    },
6    {
7      "numOfTips": 622,
8      "name": "Phoenix Sky Harbor International Airport"
9    },
10   {
11     "numOfTips": 430,
12     "name": "Earl of Sandwich"
13   },
14   {
15     "numOfTips": 374,
16     "name": "Las Vegas Athletic Club Southwest"
17   },
18   {
19     "numOfTips": 351,
20     "name": "The Cosmopolitan of Las Vegas"
21   },
22   {
23     "numOfTips": 347,
24     "name": "Wicked Spoon"
25   },
26   {
27     "numOfTips": 258,
28     "name": "Sushi House Goyemon"
29   },
30   {
31     "numOfTips": 252,
32     "name": "Pho Kim Long"
33   },
34   {
35     "numOfTips": 239,
36     "name": "Secret Pizza"
37   },
38   {
39     "numOfTips": 227,
40     "name": "Marquee Nightclub & Dayclub"
41   },
42   {
43     "numOfTips": 221,
44     "name": "Monta Ramen"
45   },
46   {
47     "numOfTips": 209,
48     "name": "AMC Town Square 18"
49   },
50   {
51     "numOfTips": 208,
52     "name": "Aria Hotel & Casino"
53   },
54 ]
```

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

- d) Recenzje mogą być oceniane przez innych użytkowników jako *cool*, *funny* lub *useful* (kolekcja *review*, pole *votes*, jedna recenzja może mieć kilka głosów w każdej kategorii). Napisz zapytanie, które zwraca dla każdej z tych kategorii, ile sumarycznie recenzji zostało oznaczonych przez te kategorie (np. recenzja ma kategorię *funny* jeśli co najmniej jedna osoba zagłosowała w ten sposób na daną recenzję)

Zliczam ilość głosów dla każdej recenzji.

```
1 use('KC');
2
3 db.getCollection('review')
4   .aggregate([
5     {$project:{
6       _id: "$review_id",
7       funny: { $cond: [{ $gt: [ "$votes.funny", 1 ]}, 1, 0 ]},
8       useful: { $cond: [{ $gt: [ "$votes.useful", 1 ]}, 1, 0 ]},
9       cool: { $cond: [{ $gt: [ "$votes.cool", 1 ]}, 1, 0 ]}
10    }},
11    {$group:{
12      _id: "votes",
13      numberOfFunny: { $sum: "$funny"},
14      numberOfUseful: { $sum: "$useful"},
15      numberOfCool: { $sum: "$cool"}
16    }},
17  ])
```

```
1 [
2   {
3     "_id": "votes",
4     "numberOfFunny": 112471,
5     "numberOfUseful": 272753,
6     "numberOfCool": 142465
7   }
8 ]
```

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

- e) Zwróć dane wszystkich użytkowników (kolekcja *user*), którzy nie mają ani jednego pozytywnego głosu (pole *votes*) z kategorii (*funny* lub *useful*), wynik posortuj alfabetycznie według nazwy użytkownika.

Wybieram użytkowników, którzy spełniają podane warunki i sortuję po nazwie.

```
1  use('KC');
2
3  db.getCollection('user')
4    .find({
5      $and: [
6        {"votes.funny": 0},
7        {"votes.useful": 0}
8      ]
9    },
10   {
11     _id: 0,
12     name: 1,
13     votes: 1
14   })
15   .sort({
16     name: 1
17   })
```

```
1  [
2    {
3      "votes": {
4        "funny": 0,
5        "useful": 0,
6        "cool": 0
7      },
8      "name": " Bernard"
9    },
10   {
11     "votes": {
12       "funny": 0,
13       "useful": 0,
14       "cool": 0
15     },
16     "name": ",Maria"
17   },
18   {
19     "votes": {
20       "funny": 0,
21       "useful": 0,
22       "cool": 0
23     },
24     "name": "006969123"
25   },
26   {
27     "votes": {
28       "funny": 0,
29       "useful": 0,
30       "cool": 0
31     },
32     "name": "A"
33   },
34   {
35     "votes": {
36       "funny": 0,
37       "useful": 0,
38       "cool": 0
39     },
40     "name": "A"
41   },
42   {
43     "votes": {
44       "funny": 0,
45       "useful": 0,
46       "cool": 0
47     },
48     "name": "A"
49   },
50   {
51     "votes": {
52       "funny": 0,
53       "useful": 0,
54       "cool": 0
55     },
56     "name": "A"
57   },
58 ]
```

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

- f) Wyznacz, jaką średnią ocenę uzyskała każda firma na podstawie wszystkich recenzji (kolekcja *review*, pole *stars*). Ogranicz do firm, które uzyskały średnią powyżej 3 gwiazdek.

przypadek 1: Wynik powinien zawierać id firmy oraz średnią ocenę. Posortuj wynik wg id firmy.

przypadek 2: Wynik powinien zawierać nazwę firmy oraz średnią ocenę. Posortuj wynik wg nazwy firmy.

Opinię grupuję, dla każdej firmy i wyliczam średnią.

```
3 db.collection('review')
4   .aggregate([
5     {$group:{
6       _id: "$business_id",
7       avgStars: {$avg: "$stars"}
8     }},
9     {$match:{
10      avgStars:{$gt: 3}
11    }},
12    {$sort:{
13      _id: 1
14    }}
15  ])
```

```
1  [
2    {
3      "_id": "--1emggGHgoG6ipd_RMb-g",
4      "avgStars": 3.75
5    },
6    {
7      "_id": "--5jkZ3-nUPZxUvtcbr8Uw",
8      "avgStars": 4.615384615384615
9    },
10   {
11     "_id": "--BlvD0_RG2yElKu9XA1_g",
12     "avgStars": 3.9696969696969697
13   },
14   {
15     "_id": "--Dl2rW_x08GuYBomlg9zw",
16     "avgStars": 4.166666666666667
17   },
18   {
19     "_id": "--0l5mVSMaW8ExtmWRUmKA",
20     "avgStars": 5
21   },
22   {
23     "_id": "--XBxRlD92RaV6TyUnP80w",
24     "avgStars": 3.6666666666666665
25   },
26   {
27     "_id": "--jFTZmywe7StuZ2hEjxyA",
28     "avgStars": 4.333333333333333
29   },
30   {
31     "_id": "--m1g9P1wxNblrLANfVqlA",
32     "avgStars": 4.25
33   },
34   {
35     "_id": "--qeSYxyn62mMjWvznNTdg",
36     "avgStars": 4
37   },
38   {
39     "_id": "--05qMWbhAtrD6EiV-UElPg",
40     "avgStars": 5
41   },
42   {
43     "_id": "--0Ackw6MF82PX09f9Jh_Kg",
44     "avgStars": 3.3173076923076925
45   },
46   {
47     "_id": "--0D_CYhlD2ILkmLR0pBmnA",
48     "avgStars": 3.3
49   },
50 ]
```

Opinię grupuję, dla każdej firmy i wyliczam średnią, a potem dopisuję nazwę firmy i sortuję.

```
1  use('KC');
2
3  db.getCollection('review')
4    .aggregate([
5      {$group:{
6        _id: "$business_id",
7        avgStars: {$avg: "$stars"}
8      }},
9      {$match:{
10       avgStars:{$gt: 3}
11     }},
12     {$lookup:{
13       from: "business",
14       localField: "_id",
15       foreignField: "business_id",
16       as: "business_info"
17     }},
18     {$project:{
19       _id: 0,
20       name: {$arrayElemAt: ["$business_info.name",0]},
21       avgStars: 1
22     }},
23     {$sort:{
24       name: 1
25     }}
26   ])
```

```
1  [
2  {
3    "avgStars": 4.125,
4    "name": "#1 Brothers Pizza"
5  },
6  {
7    "avgStars": 4.846153846153846,
8    "name": "$25 Chiropractic"
9  },
10 {
11   "avgStars": 3.125,
12   "name": "1 Brother's Pizza"
13 },
14 {
15   "avgStars": 3.44,
16   "name": "1 Brothers Pizza"
17 },
18 {
19   "avgStars": 3.4827586206896552,
20   "name": "1 Eastern Super Buffet"
21 },
22 {
23   "avgStars": 4.846153846153846,
24   "name": "1 Hr Photo Shack"
25 },
26 {
27   "avgStars": 5,
28   "name": "1-800-GOT-JUNK?"
29 },
30 {
31   "avgStars": 4.666666666666667,
32   "name": "1-800-GOT-JUNK?"
33 },
34 {
35   "avgStars": 4.318181818181818,
36   "name": "10-to-10 In Delhi"
37 },
38 {
39   "avgStars": 3.6363636363636362,
40   "name": "100% Natural Mexican Grill"
41 },
42 {
43   "avgStars": 3.081081081081081,
44   "name": "101 Asian Buffet"
45 },
46 {
47   "avgStars": 3.25,
48   "name": "101 Bistro"
49 },
50 {
51   "avgStars": 4.75,
52   "name": "103 Hot Pilates & Yoga"
53 },
```



#### .... 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

## 2. Modelowanie danych

- Zaproponuj strukturę bazy danych dla wybranego/przykładowego zagadnienia/problemu
- Należy wybrać jedno zagadnienie/problem (A lub B)

### Przykład A

- Wykładowcy, przedmioty, studenci, oceny
- Wykładowcy prowadzą zajęcia z poszczególnych przedmiotów
- Studenci uczęszczają na zajęcia
- Wykładowcy wystawiają oceny studentom
- Studenci oceniają zajęcia

### Przykład B

- Firmy, wycieczki, osoby
  - Firmy organizują wycieczki
  - Osoby rezerwują miejsca/wykupują bilety
  - Osoby oceniają wycieczki
- 
- a) Warto zaproponować/rozważyć różne warianty struktury bazy danych i dokumentów w poszczególnych kolekcjach oraz przeprowadzić dyskusję każdego wariantu (wskazać wady i zalety każdego z wariantów)
  - b) Kolekcje należy wypełnić przykładowymi danymi
  - c) W kontekście zaprezentowania wad/zalet należy zaprezentować kilka przykładów/zapytań/zadań/operacji oraz dla których dedykowany jest dany wariantów

Dla przykładu B wariant 1, w którym mam 3 kolekcje: firmy, wycieczki i użytkownicy.

```
1  use('travel');
2
3  db.createCollection('company');
4  db.createCollection('trip');
5  db.createCollection('user');
6
7  db.getCollection('company').insertMany([
8      {
9          business_id: 0,
10         name: "itaka",
11         address: "45-072 Opole ul. Reymonta 39",
12         trips: [{trip_id: 0},
13                 {trip_id: 2},
14                 {trip_id: 3},
15                 {trip_id: 6}]
16     },
17     {
18         business_id: 1,
19         name: "r",
20         address: "90-361 Łódź ul. Piotrkowska 270",
21         trips: [{trip_id: 1},
22                 {trip_id: 2},
23                 {trip_id: 4},
24                 {trip_id: 5}]
25     },
26     {
27         business_id: 2,
28         name: "tui",
29         address: "02-675 Warszawa ul. Wołoska 22a",
30         trips: [{trip_id: 0},
31                 {trip_id: 1},
32                 {trip_id: 2},
33                 {trip_id: 3}]
34     },
35 ])
```

```
37 db.getCollection('trip').insertMany([
38     {
39         trip_id: 0,
40         name: "Whala!Bavaro",
41         country: "Dominikana",
42         reviews: [{user_id: 0, stars: 3, comment: "Tak średnio bym powiedział"},
43                   |         {user_id: 3, stars: 4, comment: "Ok"}]
44     },
45     {
46         trip_id: 1,
47         name: "Rixos Sungate",
48         country: "Turcja",
49         reviews: [{user_id: 2, stars: 2, comment: "Słabo"},
50                   |         {user_id: 1, stars: 1, comment: "Dramat"}]
51     },
52     {
53         trip_id: 2,
54         name: "Saint Nicholas",
55         country: "Grecja",
56         reviews: [{user_id: 1, stars: 5, comment: "Super"},
57                   |         {user_id: 0, stars: 4, comment: "Git"}]
58     },
59     {
60         trip_id: 3,
61         name: "Nest Style Zanzibar",
62         country: "Zanzibar",
63         reviews: [{user_id: 1, stars: 5, comment: "Piękne miejsce"},
64                   |         {user_id: 3, stars: 2, comment: "Straszne miejsce"}]
65     },
66     {
67         trip_id: 4,
68         name: "Koggala Beach",
69         country: "Sri Lanka",
70         reviews: [{user_id: 2, stars: 2, comment: "Brzydko"},
71                   |         {user_id: 3, stars: 3, comment: "Może być"}]
72     },
73     {
74         trip_id: 5,
75         name: "Plaza Real",
76         country: "Algarve",
77         reviews: [{user_id: 2, stars: 2, comment: "Nie mój klimat"},
78                   |         {user_id: 0, stars: 2, comment: "Słabo"}]
79     },
80     {
81         trip_id: 6,
82         name: "Port de Soller",
83         country: "Majorka",
84         reviews: [{user_id: 2, stars: 3, comment: "Tak średnio bym powiedział"},
85                   |         {user_id: 3, stars: 4, comment: "Ok"}]
86     },
87 ])
```

```
89 db.getCollection('user').insertMany([
90     {
91         user_id: 0,
92         name: "Jan",
93         surname: "Nowak",
94         trips: [{trip_id: 0},
95                 |      {trip_id: 2},
96                 |      {trip_id: 5}]
97     },
98     {
99         user_id: 1,
100        name: "Marek",
101        surname: "Kowalski",
102        trips: [{trip_id: 1},
103                |      {trip_id: 2},
104                |      {trip_id: 3}]
105    },
106    {
107        user_id: 2,
108        name: "Janusz",
109        surname: "Jop",
110        trips: [{trip_id: 1},
111                |      {trip_id: 4},
112                |      {trip_id: 5},
113                |      {trip_id: 6}]
114    },
115    {
116        user_id: 3,
117        name: "Anna",
118        surname: "Polak",
119        trips: [{trip_id: 0},
120                |      {trip_id: 3},
121                |      {trip_id: 4},
122                |      {trip_id: 6}]
123    },
124    ])
```

Do 1 wariantu zapytanie o wszystkie firmy Warszawy.

```
1 use('travel');
2
3 db.getCollection('company')
4   .find({
5     address: {$regex: 'Warszawa'}
6   })
```

```
1 [
2   {
3     "_id": {
4       "$oid": "6482e18cfe473caf0583b058"
5     },
6     "business_id": 2,
7     "name": "tui",
8     "address": "02-675 Warszawa ul. Wołoska 22a",
9     "trips": [
10      {
11        "trip_id": 0
12      },
13      {
14        "trip_id": 1
15      },
16      {
17        "trip_id": 2
18      },
19      {
20        "trip_id": 3
21      }
22    ]
23  }
24 ]
```

Do 1 wariantu zapytanie o wszystkie firmy i ich wycieczki.

```
8 db.getCollection('company')
9   .aggregate([
10     {$lookup: {
11       from: 'trip',
12       localField: 'trips.trip_id',
13       foreignField: 'trip_id',
14       as: 'trips_info'
15     }},
16     {$project: {
17       _id: 0,
18       name: 1,
19       trips_info: 1
20     }}
21   ])
```

```

1  [
2    {
3      "name": "itaka",
4      "trips_info": [
5        {
6          "_id": {
7            "$oid": "6482e18cfe473caf0583b059"
8          },
9          "trip_id": 0,
10         "name": "Whala!Bavaro",
11         "country": "Dominikana",
12         "reviews": [
13           {
14             "user_id": 0,
15             "stars": 3,
16             "comment": "Tak średnio bym powiedział"
17           },
18           {
19             "user_id": 3,
20             "stars": 4,
21             "comment": "Ok"
22           }
23         ]
24       },
25       {
26         "_id": {
27           "$oid": "6482e18cfe473caf0583b05b"
28         },
29         "trip_id": 2,
30         "name": "Saint Nicholas",
31         "country": "Grecja",
32         "reviews": [
33           {
34             "user_id": 1,
35             "stars": 5,
36             "comment": "Super"
37           },
38           {
39             "user_id": 0,
40             "stars": 4,
41             "comment": "Git"
42           }
43         ]
44       },

```

Dla 1 wariantu zapytanie o wszystkie wycieczki, którym recenzje wystawił użytkownik o id 0.

```

23  db.getCollection('trip')
24    .find({
25      |   "reviews.user_id": 0
26    })

```

```
1  [
2    {
3      "_id": {
4        "$oid": "6482e18cfe473caf0583b059"
5      },
6      "trip_id": 0,
7      "name": "Whala!Bavaro",
8      "country": "Dominikana",
9      "reviews": [
10       {
11         "user_id": 0,
12         "stars": 3,
13         "comment": "Tak średnio bym powiedział"
14       },
15       {
16         "user_id": 3,
17         "stars": 4,
18         "comment": "Ok"
19       }
20     ]
21   },
22   {
23     "_id": {
24       "$oid": "6482e18cfe473caf0583b05b"
25     },
26     "trip_id": 2,
27     "name": "Saint Nicholas",
28     "country": "Grecja",
29     "reviews": [
30       {
31         "user_id": 1,
32         "stars": 5,
33         "comment": "Super"
34       },
35       {
36         "user_id": 0,
37         "stars": 4,
38         "comment": "Git"
39       }
40     ]
41   },
```

Dla 1 wariantu zapytanie o wszystkie wycieczki i ich średnią ocenę.

```
29 db.collection('trip')
30   .aggregate([
31     {$project:{
32       _id: 0,
33       name: 1,
34       avgStars: { $avg: "$reviews.stars" }
35     }},
36   ])
```

```
1  [
2    {
3      "name": "Whala!Bavaro",
4      "avgStars": 3.5
5    },
6    {
7      "name": "Rixos Sungate",
8      "avgStars": 1.5
9    },
10   {
11     "name": "Saint Nicholas",
12     "avgStars": 4.5
13   },
14   {
15     "name": "Nest Style Zanzibar",
16     "avgStars": 3.5
17   },
18   {
19     "name": "Koggala Beach",
20     "avgStars": 2.5
21   },
22   {
23     "name": "Plaza Real",
24     "avgStars": 2
25   },
26   {
27     "name": "Port de Soller",
28     "avgStars": 3.5
29   }
30 ]
```



Dla 1 wariantu zapytanie o wszystkich użytkowników, którzy w nazwisku mają frazę 'owa'.

```
39 db.collection('user')
40   .find({
41     surname: {$regex: 'owa'}
42   })
```

```
1  [
2    {
3      "_id": {
4        "$oid": "6482e18cfe473caf0583b060"
5      },
6      "user_id": 0,
7      "name": "Jan",
8      "surname": "Nowak",
9      "trips": [
10       {
11         "trip_id": 0
12       },
13       {
14         "trip_id": 2
15       },
16       {
17         "trip_id": 5
18       }
19     ]
20   },
21   {
22     "_id": {
23       "$oid": "6482e18cfe473caf0583b061"
24     },
25     "user_id": 1,
26     "name": "Marek",
27     "surname": "Kowalski",
28     "trips": [
29       {
30         "trip_id": 1
31       },
32       {
33         "trip_id": 2
34       },
35       {
36         "trip_id": 3
37       }
38     ]
39   }
40 ]
```

Dla 1 wariantu zapytanie o wszystkich użytkowników i ich wycieczki.

```
44 db.collection('user')
45   .aggregate([
46     {$lookup: {
47       from: 'trip',
48       localField: 'trips.trip_id',
49       foreignField: 'trip_id',
50       as: 'trips_info'
51     }},
52     {$project: {
53       _id: 0,
54       name: 1,
55       surname: 1,
56       trips: "$trips_info.name"
57     }},
58   ])
```

```
1  [
2    {
3      "name": "Jan",
4      "surname": "Nowak",
5      "trips": [
6        "Whala!Bavaro",
7        "Saint Nicholas",
8        "Plaza Real"
9      ]
10   },
11   {
12     "name": "Marek",
13     "surname": "Kowalski",
14     "trips": [
15       "Rixos Sungate",
16       "Saint Nicholas",
17       "Nest Style Zanzibar"
18     ]
19   },
20   {
21     "name": "Janusz",
22     "surname": "Jop",
23     "trips": [
24       "Rixos Sungate",
25       "Koggala Beach",
26       "Plaza Real",
27       "Port de Soller"
28     ]
29   },
30   {
31     "name": "Anna",
32     "surname": "Polak",
33     "trips": [
34       "Whala!Bavaro",
35       "Nest Style Zanzibar",
36       "Koggala Beach",
37       "Port de Soller"
38     ]
39   }
40 ]
```

Wadą wariantu 1 na pewno jest niewygodne zagnieżdżenie recenzji w wycieczkach. Dla 2 wariantu stworzę osobną kolekcję na recenzje.

```
1  use('travel2');
2
3  db.createCollection('company');
4  db.createCollection('trip');
5  db.createCollection('user');
6  db.createCollection('review');
7
8  db.getCollection('company').insertMany([
9      {
10         business_id: 0,
11         name: "itaka",
12         address: "45-072 Opole ul. Reymonta 39",
13         trips: [{trip_id: 0},
14                 {trip_id: 2},
15                 {trip_id: 3},
16                 {trip_id: 6}]
17     },
18     {
19         business_id: 1,
20         name: "r",
21         address: "90-361 Łódź ul. Piotrkowska 270",
22         trips: [{trip_id: 1},
23                 {trip_id: 2},
24                 {trip_id: 4},
25                 {trip_id: 5}]
26     },
27     {
28         business_id: 2,
29         name: "tui",
30         address: "02-675 Warszawa ul. Wołoska 22a",
31         trips: [{trip_id: 0},
32                 {trip_id: 1},
33                 {trip_id: 2},
34                 {trip_id: 3}]
35     },
36 ])
```

```
38 db.getCollection('trip').insertMany([
39     {
40         trip_id: 0,
41         name: "Whala!Bavaro",
42         country: "Dominikana",
43         reviews: [{review_id: 0},
44                   |   |   | {review_id: 1}]
45     },
46     {
47         trip_id: 1,
48         name: "Rixos Sungate",
49         country: "Turcja",
50         reviews: [{review_id: 2},
51                   |   |   | {review_id: 3}]
52     },
53     {
54         trip_id: 2,
55         name: "Saint Nicholas",
56         country: "Grecja",
57         reviews: [{review_id: 4},
58                   |   |   | {review_id: 5}]
59     },
60     {
61         trip_id: 3,
62         name: "Nest Style Zanzibar",
63         country: "Zanzibar",
64         reviews: [{review_id: 6},
65                   |   |   | {review_id: 7}]
66     },
67     {
68         trip_id: 4,
69         name: "Koggala Beach",
70         country: "Sri Lanka",
71         reviews: [{review_id: 8},
72                   |   |   | {review_id: 9}]
73     },
74     {
75         trip_id: 5,
76         name: "Plaza Real",
77         country: "Algarve",
78         reviews: [{review_id: 10},
79                   |   |   | {review_id: 11}]
80     },
81     {
82         trip_id: 6,
83         name: "Port de Soller",
84         country: "Majorka",
85         reviews: [{review_id: 12},
86                   |   |   | {review_id: 13}]
87     },
88 ])
```

```
90 db.getCollection('user').insertMany([
91     { user_id: 0,
92     ⚡ name: "Jan",
93       surname: "Nowak",
94       trips: [{trip_id: 0},
95               {trip_id: 2},
96               {trip_id: 5}]
97     },
98     { user_id: 1,
99       name: "Marek",
100       surname: "Kowalski",
101       trips: [{trip_id: 1},
102               {trip_id: 2},
103               {trip_id: 3}]
104     },
105     { user_id: 2,
106       name: "Janusz",
107       surname: "Jop",
108       trips: [{trip_id: 1},
109               {trip_id: 4},
110               {trip_id: 5},
111               {trip_id: 6}]
112     },
113     { user_id: 3,
114       name: "Anna",
115       surname: "Polak",
116       trips: [{trip_id: 0},
117               {trip_id: 3},
118               {trip_id: 4},
119               {trip_id: 6}]
120     },
121 ])
```

```

123 db.getCollection('review').insertMany([
124     {review_id: 0, trip_id: 0, user_id: 0, stars: 3, comment: "Tak średnio bym
powiedział"},
125     {review_id: 1, trip_id: 0, user_id: 3, stars: 4, comment: "Ok"},
126     {review_id: 2, trip_id: 1, user_id: 2, stars: 2, comment: "Słabo"},
127     {review_id: 3, trip_id: 1, user_id: 1, stars: 1, comment: "Dramat"},
128     {review_id: 4, trip_id: 2, user_id: 1, stars: 5, comment: "Super"},
129     {review_id: 5, trip_id: 2, user_id: 0, stars: 4, comment: "Git"},
130     {review_id: 6, trip_id: 3, user_id: 1, stars: 5, comment: "Piękne miejsce"},
131     {review_id: 7, trip_id: 3, user_id: 3, stars: 2, comment: "Straszne miejsce"},
132     {review_id: 8, trip_id: 4, user_id: 2, stars: 2, comment: "Brzydko"},
133     {review_id: 9, trip_id: 4, user_id: 3, stars: 3, comment: "Może być"},
134     {review_id: 10, trip_id: 5, user_id: 2, stars: 2, comment: "Nie mój klimat"},
135     {review_id: 11, trip_id: 5, user_id: 0, stars: 2, comment: "Słabo"},
136     {review_id: 12, trip_id: 6, user_id: 2, stars: 3, comment: "Tak średnio bym
powiedział"},
137     {review_id: 13, trip_id: 6, user_id: 3, stars: 4, comment: "Ok"}
138 ])

```

Teraz wypisanie użytkowników wraz z recenzjami jest znacznie łatwiejsze.

```

1  use('travel2');
2
3  db.getCollection('user')
4      .aggregate([
5      { $lookup: {
6          from: 'review',
7          localField: 'user_id',
8          foreignField: 'user_id',
9          as: 'reviews'
10     }},
11     { $project: {
12         _id: 0,
13         name: 1,
14         surname: 1,
15         reviews: 1
16     }}
17 ])

```

```
1  [
2    {
3      "name": "Jan",
4      "surname": "Nowak",
5      "reviews": [
6        {
7          "_id": {
8            "$oid": "6482fb864eff1d8b0903f526"
9          },
10         "review_id": 0,
11         "trip_id": 0,
12         "user_id": 0,
13         "stars": 3,
14         "comment": "Tak średnio bym powiedział"
15       },
16       {
17         "_id": {
18           "$oid": "6482fb864eff1d8b0903f52b"
19         },
20         "review_id": 5,
21         "trip_id": 2,
22         "user_id": 0,
23         "stars": 4,
24         "comment": "Git"
25       },
26       {
27         "_id": {
28           "$oid": "6482fb864eff1d8b0903f531"
29         },
30         "review_id": 11,
31         "trip_id": 5,
32         "user_id": 0,
33         "stars": 2,
34         "comment": "Słabo"
35       }
36     ]
37   },
38   {
39     "name": "Marek",
40     "surname": "Kowalski",
41     "reviews": [
42       {
43         "_id": {
44           "$oid": "6482fb864eff1d8b0903f529"
45         },
46         "review_id": 3,
47         "trip_id": 1,
48         "user_id": 1,
49         "stars": 1,
50         "comment": "Dramat"
51       },
52       {
53         "_id": {
54           "$oid": "6482fb864eff1d8b0903f52a"
55         },
56         "review_id": 4,
57         "trip_id": 2,
58         "user_id": 1,
```

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

W sprawozdaniu należy zamieścić przykładowe dokumenty w formacie JSON ( pkt a) i b)), oraz kod zapytań/operacji (pkt c)), wraz z odpowiednim komentarzem opisującym strukturę dokumentów oraz polecenia ilustrujące wykonanie przykładowych operacji na danych

Do sprawozdania należy kompletny zrzut wykonanych/przygotowanych baz danych (taki zrzut można wykonać np. za pomocą poleceń mongoexport, mongodump ...) oraz plik z kodem operacji zapytań (załącznik powinien mieć format zip).

**Punktacja za zadanie (razem 2pkt)**