

AGH WIET Niestacjonarne Semestr 7

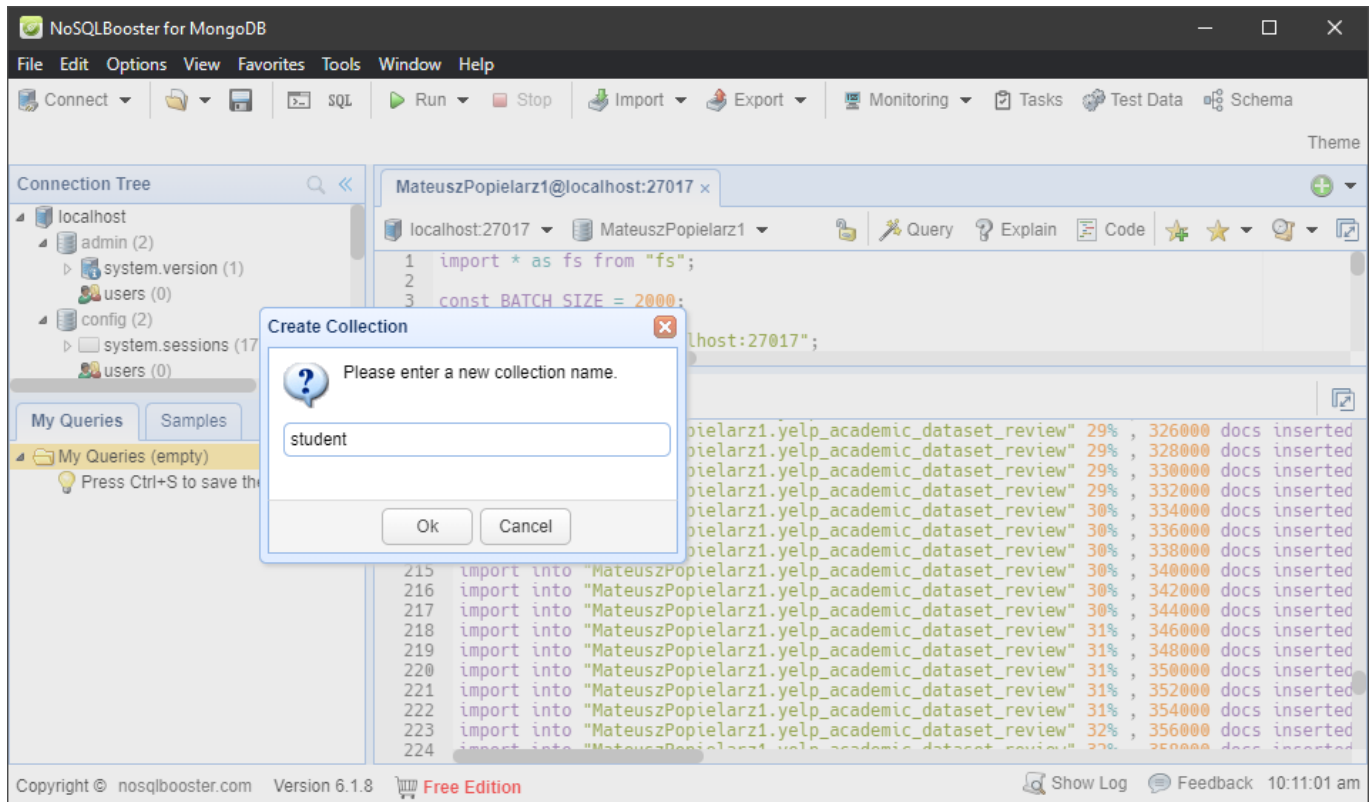
Mateusz Popielarz

Lab MongoDB - Raport

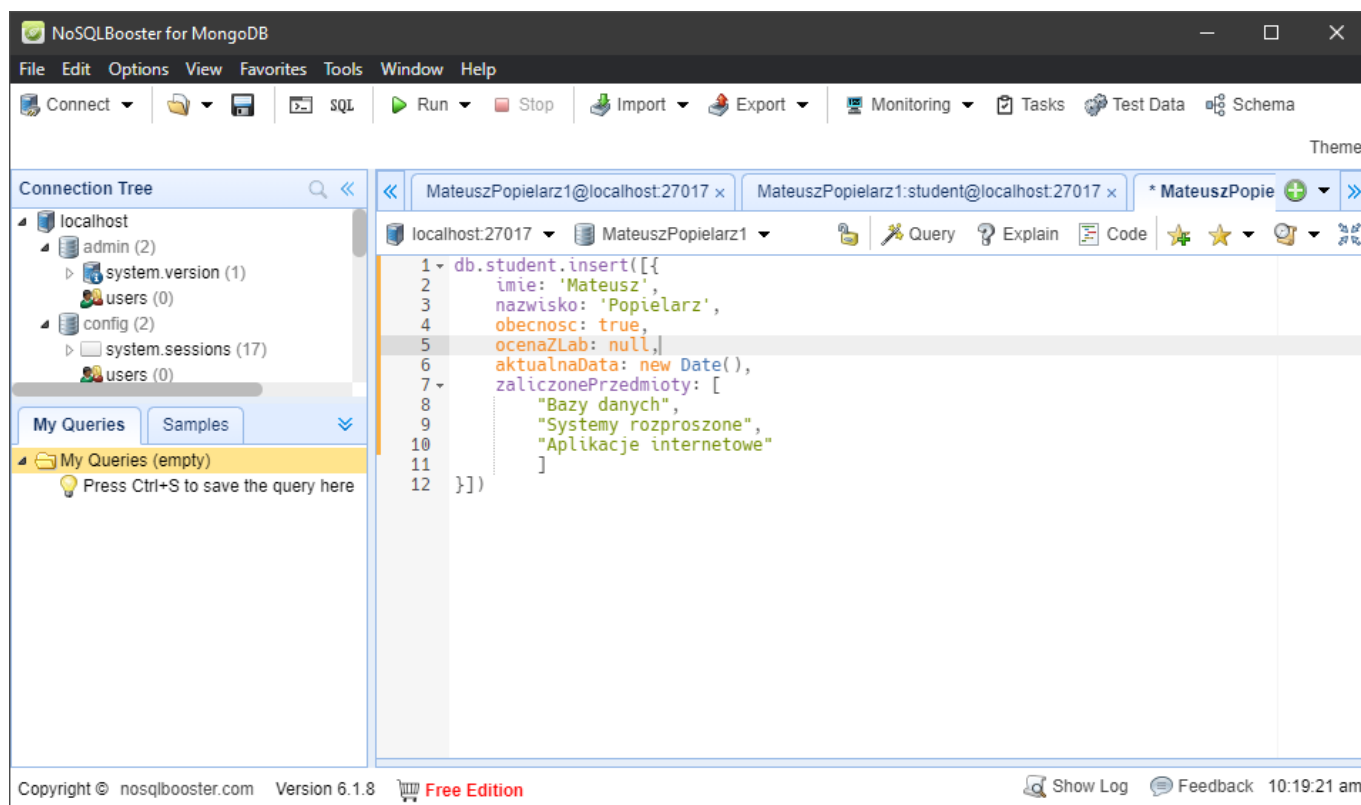
Zadania

4. Za pomocą narzędzia Robo 3T wykonaj polecenie dodające do stworzonej bazy kolekcję „student”:

W MongoDB booster wybrałem 'Add collection' i wpisałem

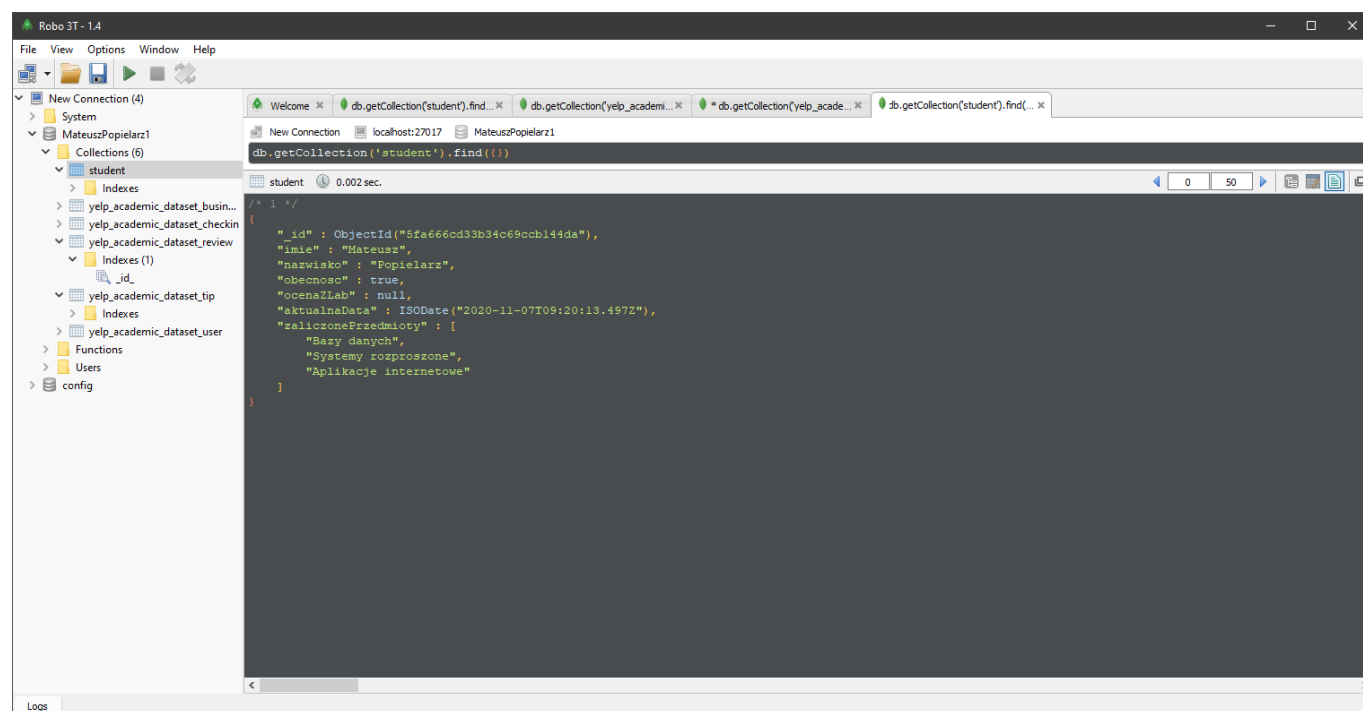


a) wprowadź własne dane do kolekcji: imię, nazwisko, obecność (typ bool), ocena z lab. (null), aktualna data, zaliczone przedmioty (min 3 przykładowe).



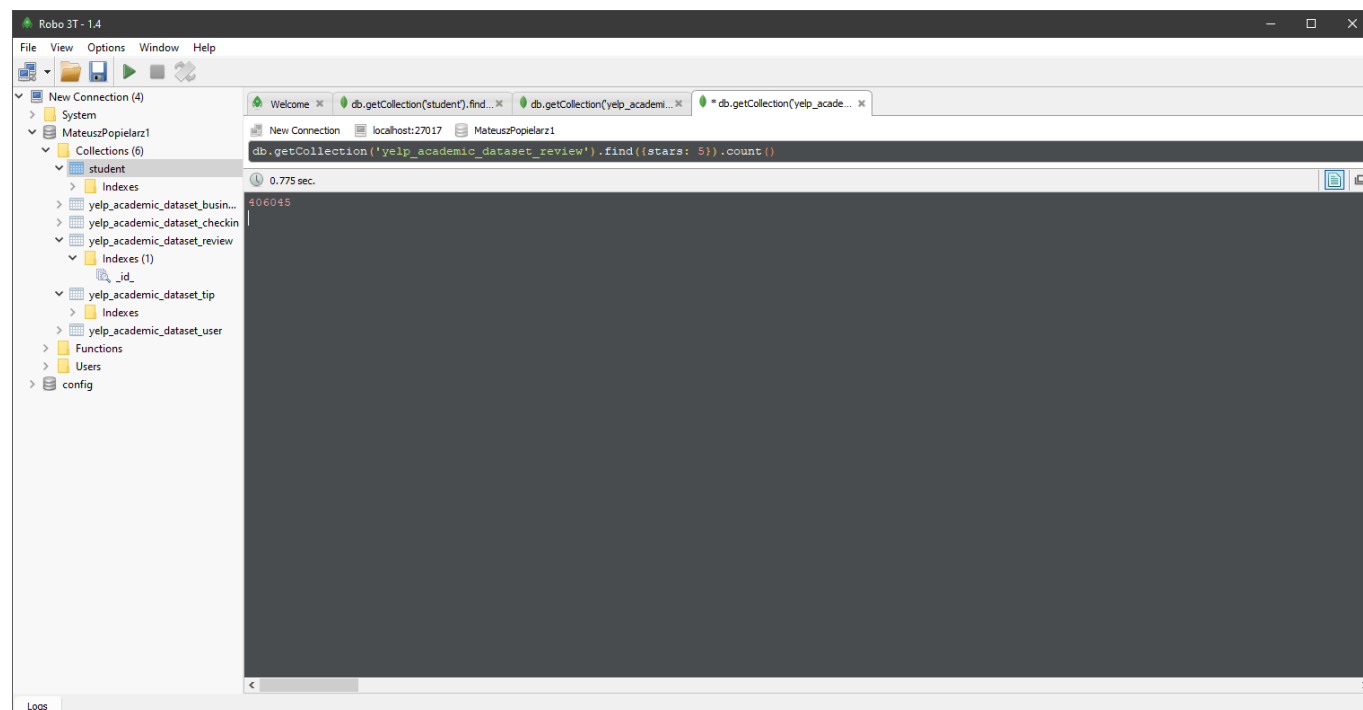
Dane wprowadzone według instrukcji

b) wyświetl wynik dodania danej w formie. json txt.



5. Za pomocą narzędzia Robo 3T wykonaj zapytania, które pozwolą uzyskać następujące wyniki:

a. ilość miejsc ocenianych na 5 gwiazdek (pole stars, kolekcja business)



b. ilość restauracji w każdym mieście, wynik posortuj malejąco na podstawie liczby. Pole categories w dokumencie business musi zawierać wartość Restaurants.

The screenshot shows the Robo 3T 1.4 application. The left sidebar displays a project tree with collections like 'student', 'yelp_academic_dataset_business', 'yelp_academic_dataset_checkin', 'yelp_academic_dataset_review', 'yelp_academic_dataset_tip', 'yelp_academic_dataset_user', 'Functions', 'Users', and 'config'. The main window shows a MongoDB aggregation query:

```
db.getCollection('yelp_academic_dataset_business')
.aggregate([
  { $match: { categories: { $in: ["Restaurants"] } } },
  { $group: { "_id": "$city", "count": { $sum: 1 } } },
  { $sort: { "count": -1 } }
])
```

Below the query, the results are displayed in a table for the 'yelp_academic_dataset_business' collection. The table has three columns: Key, Value, and Type. The results are sorted by count in descending order.

Key	Value	Type
(1) Las Vegas	{ 2 fields }	Object
_id	Las Vegas	String
count	3855.0	Double
(2) Phoenix	{ 2 fields }	Object
_id	Phoenix	String
count	2493.0	Double
(3) Edinburgh	{ 2 fields }	Object
_id	Edinburgh	String
count	1049.0	Double
(4) Scottsdale	{ 2 fields }	Object
_id	Scottsdale	String
count	1023.0	Double
(5) Mesa	{ 2 fields }	Object
_id	Mesa	String
count	693.0	Double
(6) Madison	{ 2 fields }	Object
_id	Madison	String
count	679.0	Double
(7) Tempe	{ 2 fields }	Object
(8) Henderson	{ 2 fields }	Object
(9) Chandler	{ 2 fields }	Object

```
db.getCollection('yelp_academic_dataset_business')
.aggregate([
  { $match: { categories: { $in: ["Restaurants"] } } },
  { $group: { "_id": "$city", "count": { $sum: 1 } } },
  { $sort: { "count": -1 } }
])
```

c. ilość hoteli (atrybut categories powinien mieć wartość Hotels) w każdym stanie/okręgu (state), które posiadają darmowe Wi-fi (pole attributes, klucz-wartość 'Wi-Fi': 'free') oraz ocenę co najmniej 4.5 gwiazdki. Wykorzystaj funkcję group.

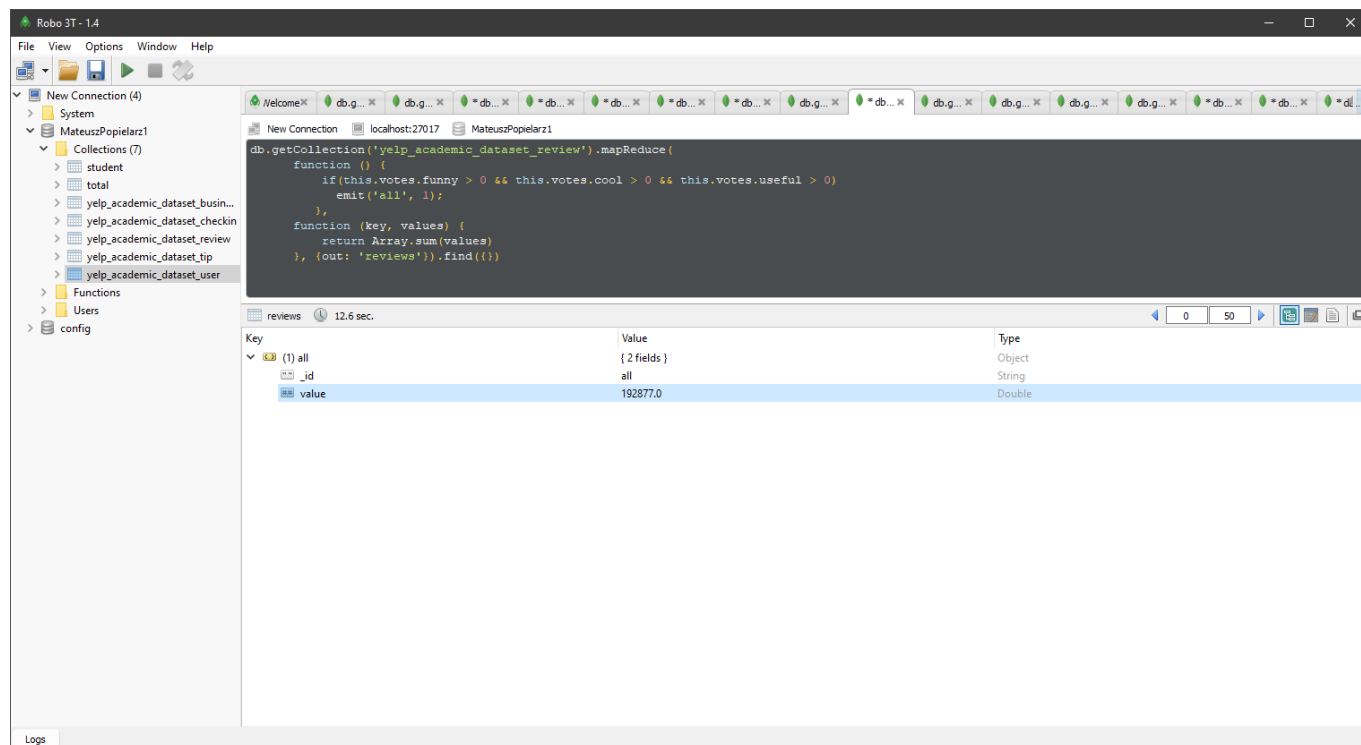
The screenshot shows the Robo 3T-1.4 application. The left sidebar displays a project structure with collections like 'student', 'yelp_academic_dataset_business', 'yelp_academic_dataset_checkin', 'yelp_academic_dataset_review', 'yelp_academic_dataset_tip', 'yelp_academic_dataset_user', and 'config'. The main window shows a MongoDB aggregation query for the 'yelp_academic_dataset_business' collection. The query filters for businesses with categories containing 'Hotels', free Wi-Fi, and a rating greater than 4.5. It then groups the results by state and sorts them by count in descending order. The results pane at the bottom shows a table with 5 rows of data.

Key	Value	Type
(1) EDH	{ 2 fields }	Object
_id	EDH	String
count	3.0	Double
(2) NV	{ 2 fields }	Object
_id	NV	String
count	3.0	Double
(3) AZ	{ 2 fields }	Object
_id	AZ	String
count	2.0	Double
(4) WI	{ 2 fields }	Object
_id	WI	String
count	2.0	Double
(5) MLN	{ 2 fields }	Object
_id	MLN	String
count	1.0	Double

```
db.getCollection('yelp_academic_dataset_business')
  .aggregate(
    [
      {
        $match: {
          $and: [
            { categories: { $in: ["Hotels"] } },
            { "attributes.Wi-Fi": "free" },
            { stars: { $gt: 4.5 } }
          ]
        }
      },
      { $group: { "_id": "$state", "count": { $sum: 1 } } },
      { $sort: { "count": -1 } }
    ]
  )
```

d. zwróć, ile recenzji posiadają oceny z każdej kategorii: funny, cool, useful. Przypisanie recenzji do kategorii oznacza, że przynajmniej jedna osoba zagłosowała na recenzję w tej kategorii). Wykorzystaniem mechanizmu map-reduce.

W funkcji `map` emituje wartość 1, tylko jeżeli jakkolwiek recenzja została dodana. W funkcji `reduce` zostaje mi tylko proste sumowanie.



The screenshot shows the Robo 3T 1.4 interface. The left sidebar displays the database structure, including the 'yelp_academic_dataset_review' collection. The main window shows a MongoDB map-reduce operation being executed on the 'yelp_academic_dataset_review' collection. The map function emits 'all' if any vote is greater than 0. The reduce function sums the values. The result shows a single document with 'all' as a string and 'value' as 192877.0.

```
db.getCollection('yelp_academic_dataset_review').mapReduce(  
  function () {  
    if(this.votes.funny > 0 || this.votes.cool > 0 || this.votes.useful > 0)  
      emit('all', 1);  
  },  
  function (key, values) {  
    return Array.sum(values)  
  }, {out: 'reviews'}).find({})
```

Key	Value	Type
(1) all	{ 2 fields }	Object
_id	all	String
value	192877.0	Double

6. Wykonaj zadania punktu 5 z poziomu języka (TypeScript)

Użycie Node.js w dialekcie **typescript** pozwala mi natywnie użyć składni mongodb.

Kod bootstrpera

```
import * as mongodb from 'mongodb';

const url = 'mongodb://localhost:27017';
const dbName = 'MateuszPopielarz1';
declare function emit(k, v);

// Use connect method to connect to the server
mongodb.connect(url, async function (err, client) {
  console.error(err);
  console.log("Connected successfully to server");
  const db = client.db(dbName);

  await excerciseA(db);
  await excerciseB(db);
  await excerciseC(db);
  await excerciseD(db);
  await excercise7(db);

  client.close();
});
```

emit musi zostać zadeklarowany, bo nie istnieje w globalnym scope TS

a. a. ilość miejsc ocenianych na 5 gwiazdek (pole stars, kolekcja business)

```
async function excerciseA(db: mongodb.Db) {  
  const count = await db.collection('yelp_academic_dataset_review').find({ stars:  
5 }).count();  
  console.log(`Zadanie A: Count: ${count}`);  
}
```

Wynik wykonania

```
> ts-node index.ts  
Debugger listening on ws://127.0.0.1:64771/97a2573c-dff4-4ba0-9347-aa259e242a82  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
(node:28426) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
(Use "node --trace-deprecation ..." to show where the warning was created)  
null  
Connected successfully to server  
Zadanie A: Count: 485845  
Waiting for the debugger to disconnect...  
PS D:\Vigh\Semestr6\mongo>
```

b. ilość restauracji w każdym mieście, wynik posortuj malejąco na podstawie liczby. Pole categories w dokumencie business musi zawierać wartość Restaurants.

```
async function excerciseB(db: mongodb.Db) {  
  
  const data = await db.collection('yelp_academic_dataset_business').aggregate(  
    [  
      { $match: { categories: { $in: ["Restaurants"] } } },  
      { $group: { "_id": "$city", "count": { $sum: 1 } } },  
      { $sort: { "count": -1 } }  
    ]  
  ).toArray();  
  
  console.log("Zadanie B", data);  
}
```

Rezultat:

```
> ts-node index.ts  
  
Debugger listening on ws://127.0.0.1:64789/041e0614-948d-4f84-8d4b-7e43d68a4fa1  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
(node:19212) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated  
(Use `node --trace-deprecation ...` to show where the warning was created)  
null  
Connected successfully to server  
Zadanie B [  
  { _id: 'Las Vegas', count: 3855 },  
  { _id: 'Phoenix', count: 2493 },  
  { _id: 'Edinburgh', count: 1049 },  
  { _id: 'Scottsdale', count: 1023 },  
  { _id: 'Mesa', count: 693 },  
  { _id: 'Madison', count: 679 },  
  { _id: 'Tempe', count: 672 },  
  { _id: 'Henderson', count: 564 },  
  { _id: 'Chandler', count: 548 },  
  { _id: 'Glendale', count: 422 },  
  { _id: 'Gilbert', count: 317 },  
  { _id: 'Peoria', count: 221 },  
  { _id: 'North Las Vegas', count: 198 },  
  { _id: 'Surprise', count: 144 },  
  { _id: 'Goodyear', count: 119 },  
  { _id: 'Waterloo', count: 117 },  
  { _id: 'Avondale', count: 100 },  
  { _id: 'Kitchener', count: 96 },  
  { _id: 'Queen Creek', count: 82 },  
  { _id: 'Middleton', count: 66 },  
  { _id: 'Cave Creek', count: 63 },  
  { _id: 'Casa Grande', count: 61 },  
  { _id: 'Fountain Hills', count: 47 },  
  { _id: 'Apache Junction', count: 44 },  
  { _id: 'Buckeye', count: 42 },  
  { _id: 'Sun Prairie', count: 39 },  
  { _id: 'Fitchburg', count: 38 },  
  { _id: 'Maricopa', count: 37 },  
  { _id: 'Monona', count: 32 },  
  { _id: 'Sun City', count: 31 },
```

c. ilość hoteli (atrybut categories powinien mieć wartość Hotels) w każdym stanie/okręgu (state), które posiadają darmowe Wi-fi (pole attributes, klucz-wartość 'Wi-Fi': 'free') oraz ocenę co najmniej 4.5 gwiazdki. Wykorzystaj funkcję group.

```
async function excerciseC(db: mongodb.Db) {
  const data = await db.collection('yelp_academic_dataset_business')
    .aggregate([
      [
        {
          $match: {
            $and: [
              { categories: { $in: ["Hotels"] } },
              { "attributes.Wi-Fi": "free" },
              { stars: { $gt: 4.5 } }
            ]
          }
        },
        { $group: { "_id": "$state", "count": { $sum: 1 } } },
        { $sort: { "count": -1 } }
      ]
    ])
    .toArray();

  console.log("Zadanie C", data);
}
```

Rezultat wykonania:

```
PS D:\Agh\Semestr6\mongo> npm run run
> mongo@1.0.0 run D:\Agh\Semestr6\mongo
> ts-node index.ts

Debugger listening on ws://127.0.0.1:64804/7369f676-bd3f-4ce9-89f1-d675004ab433
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
(node:26384) DeprecationWarning: current Server Discovery and Monitoring engine is
(Use `node --trace-deprecation ...` to show where the warning was created)
null
Connected successfully to server
Zadanie C [
  { _id: 'EDH', count: 3 },
  { _id: 'NV', count: 3 },
  { _id: 'AZ', count: 2 },
  { _id: 'WI', count: 2 },
  { _id: 'MLN', count: 1 }
]
Waiting for the debugger to disconnect...
PS D:\Agh\Semestr6\mongo>
```

D. zwróć, ile recenzji posiadają oceny z każdej kategorii: funny, cool, useful. Przypisanie recenzji do kategorii oznacza, że przynajmniej jedna osoba zagłosowała na recenzję w tej kategorii). Wykorzystaniem mechanizmu map-reduce.

```
async function excerciseD(db: mongodb.Db) {

  const data = await db.collection('yelp_academic_dataset_review').mapReduce(
    function () {
      if (this.votes.funny > 0)
      if (this.votes.cool > 0)
      if (this.votes.useful > 0)
        emit('all', 1);
    },
    function (key, values: number[]) {
      return values.reduce((x, acc) => x + acc);

    }, { out: 'reviews' as {} } as mongodb.MapReduceOptions);

  const result = await db.collection('reviews').find({}).toArray();
  console.log("Zadanie D", result);
}
```

Rezultat wykonania:

```
> mongo@1.0.0 run D:\Agh\Semestr6\mongo
> ts-node index.ts

Debugger listening on ws://127.0.0.1:59768/bd99f98f-6
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
(node:3896) DeprecationWarning: current Server Discover
(Use `node --trace-deprecation ...` to show where the
null
Connected successfully to server
Zadanie D [ { _id: 'all', value: 192877 } ]
Waiting for the debugger to disconnect...
```

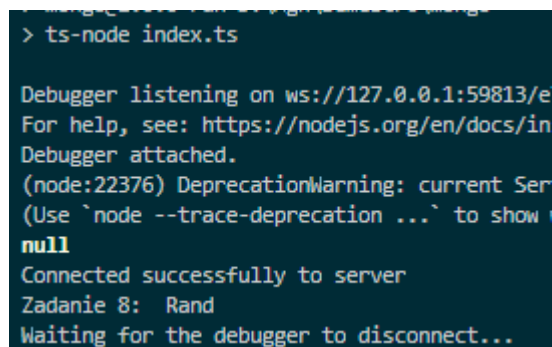
7. Napisz kod w języku Java (metoda), który zwróci użytkownika (nazwa użytkownika) o największej liczbie pozytywnych recenzji (ocena co najmniej 4.5).

```
async function exercise7(db: mongodb.Db) {
  const data = await db.collection('yelp_academic_dataset_review').aggregate(
    [
      { $match: { stars: { $gt: 4.5 } } },
      { $group: { "_id": "$user_id", "count": { $sum: 1 } } },
      { $sort: { "count": -1 } },
    ]
  ).toArray();
  const first = data[0];

  const userId = await db.collection('yelp_academic_dataset_user').findOne({
    user_id: first['_id'] })
  console.log("Zadanie 7: ", userId['name']);
}
```

Do zrealizowania tego zadania użyłem najpierw grupowania po `user_id`, gdzie ilość gwiazdek > 4.5 , posortowałem po ilości, a następnie poszukałem w kolekcji `yelp_academic_dataset_user` odpowiadającego `user_id` i wypisałem

Rezultat wykonania:



```
> ts-node index.ts
Debugger listening on ws://127.0.0.1:59813/e
For help, see: https://nodejs.org/en/docs/in
Debugger attached.
(node:22376) DeprecationWarning: current Ser
(Use `node --trace-deprecation ...` to show
null
Connected successfully to server
Zadanie 8: Rand
Waiting for the debugger to disconnect...
```