

Darbas su duomenimis

NoSQL

SQL vs NoSQL

- SQL - struktūrizuoti duomenys
 - duomenų modelis atitinka tikrovę
 - tenkina 3NF
 - duomenų gavimui naudojami lentelių join'ai.
 - neleidžia įdėti nestruktūrizuotų duomenų praleidžiant reikšmes
 - neleidžia lengvai pridėti naujų duomenų laukų
- NoSQL - nestruktūrizuoti duomenys
 - duomenų modelis labai lankstus
 - galima nest'inti duomenis

SQL Pros

- Data is nicely organized in appropriate tables, which reduces redundant information.
- The split structure allows us to join data in any way.. Numerous join types are available and can be done with any number of tables in any way.
- Data model requires some thought. This ensures data is consistent and easy to work with.
- Prevents developers from sloppily entering data with the intent to organize it later.

NoSQL Pros

- Data can easily be nested, allowing everything related to a particular entity to be in one spot.
- With nested documents, we don't have to worry about joining documents in most situations.
- The data model is flexible, allowing for easy iterations through the development process and ease-of-mind if the structure needs to change.
- The flexibility of MongoDB removes the headache of focusing so much on the database. Developers are free to focus on coding.

Kas yra MongoDB?

MongoDB yra dokumentų duomenų bazė skirta kurti interneto aplikacijas, kurios lengvai prieinamos ir greitai skeilinasi.

MongoDB apie save

MongoDB is a **document database used to build highly available and scalable internet applications**. With its flexible schema approach, it's popular with development teams using agile methodologies.

MongoDB žodynas

Jeigu palygintume MongoDB su MySQL, tai laisvai interpretuojant galime teigti, kad

- database = database
- collection ~ table
- document ~ row
- dokumentus sudaro {key: value} poros. Lygiai taip pat atrodo JSON formatas ir python žodynas.

Pasileiskime MongoDB

- https://blog.jeremylikness.com/blog/2018-12-27_mongodb-on-windows-in-minutes-with-docker/

1. paleiskite Docker Desktop (Docker Desktop 4.6.1)
2. paleiskite terminalą. Suveskite komandas
 - `docker volume create --name=mongodata`
 - `docker run --name mongodb -v mongodata:/data/db -d -p 27017:27017 mongo`
3. Jūs jau turite MongoDB.
 - `docker ps`
4. Įeikite į konteinerį
 - `docker exec -it mongodb bash`
 - `mongosh`
5. Turite matyti MongoDB serverio versiją

```
C:\Users\justa>docker ps
CONTAINER ID   IMAGE     COMMAND
1e3e1004f03c   mongo    "docker-entrypoint.s..."

C:\Users\justa>docker exec -it mongodb bash
root@1e3e1004f03c:/# mongosh
Current Mongosh Log ID: 626ae25332301f5ad0a65c01
Connecting to:      mongodb://127.0.0.1:27017/
Using MongoDB:      5.0.8
Using Mongosh:      1.3.1
```

Darbas su mongosh

- `show dbs` - parodyti visas use duomenų bazines.
- `db` - parodyti duomenų bazę, kurioje esate.
- `use <database>` - pereiti į <database> arba sukurti naują duomenų bazę
`use MyDatabase`
- `db.MyCollection.insertOne({x:1});`
Sukuria naują kolekciją MyCollection, jeigu jina neegzistuoja, ir įdėda dokumentą {x:1} į kolekciją. x - raktas, 1 - reikšmė.

Insert Documents

- `db.collection.insertOne()`
 - `db.MyCollection.insertOne({x:2})`
- `db.collection.insertMany()`
 - `db.MyCollection.insertMany([{x:3}, {x:4}])`

Query Documents

- `db.collection.find()`
 - `db.MyCollection.find()` - gražina visus dokumentus.
 - ši operacija yra ekvivalenti `select * from MyCollection`
- `db. MyCollection.find({x:1})`
 - ši operacija yra ekvivalenti `select * from MyCollection where x=1`
- `db.MyCollection.find({x: {$in: [1,2] } })`
 - ši operacija yra ekvivalenti `select * from MyCollection where x in (1, 2)`

Query Documents

- Pridėkime papildomų dokumentų į mūsų kolekciją
- `db.MyCollection.insertMany([{x:5, y:1}, {x:6, y:2}, {x:7, y:3}])`
- and, or - logikos operatoriai
 - and operatorius
 - `db.MyCollection.find({x:5, y:1})` - kableliu atskiriame reikalavimus, kurie yra jungiami kaip su and operatoriumi.
 - `db.MyCollection.find({x:{$gte:4}, y:{$lte:2}})` - gte - greater then or equal, lte - less than or equal.
 - or operatorius
 - `db.MyCollection.find({ $or: [{ x: {$lte:6}} , {y:{$lte:2}}] })`

Update Documents

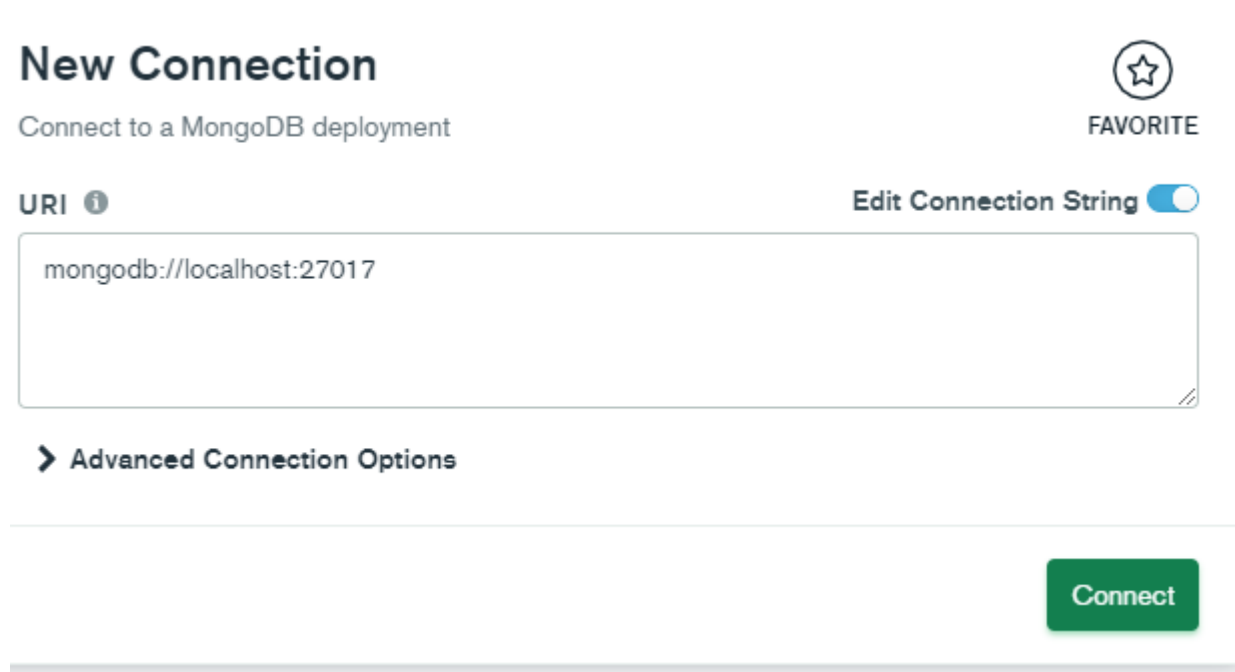
- `db.collection.updateOne()`
 - `db.MyCollection.updateOne({x:1}, {$set: { z:2 } })`
- `db.collection.updateMany()`
 - `db.MyCollection.updateMany({x:{$gt:2}}, {$set:{z:3}})`
- `db.collection.replaceOne()`
 - `db.MyCollection.replaceOne({x:2}, {x:2, y:2, z:2})`

Delete Documents

- Norint ištrinti visus dokumentus, kurie tenkina sąlygą
 - `db.MyCollection.deleteMany({x:1})`
- Norint ištrinti pirmą dokumentą, kuris tenkina sąlygą
 - `db.MyCollection.deleteOne({x:2})`
- Norint ištrinti visus dokumentus, reikia naudoti komandą
 - `db.MyCollection.deleteMany({})`

MongoDB UI

- <https://www.mongodb.com/products/compass>
- Atsisiųskite failą ir suinstaliuokite
- Paleidus programą jums pasiūlys prisijungti prie MongoDB



The screenshot shows the 'New Connection' dialog box in MongoDB Compass. At the top left, it says 'New Connection' with a subtitle 'Connect to a MongoDB deployment'. To the right is a star icon labeled 'FAVORITE'. Below this, there is a 'URI' label with an information icon and an 'Edit Connection String' toggle switch which is currently turned on. A text input field contains the URI 'mongodb://localhost:27017'. At the bottom left, there is a link 'Advanced Connection Options' with a right-pointing chevron. At the bottom right, there is a green 'Connect' button.

New Connection
Connect to a MongoDB deployment

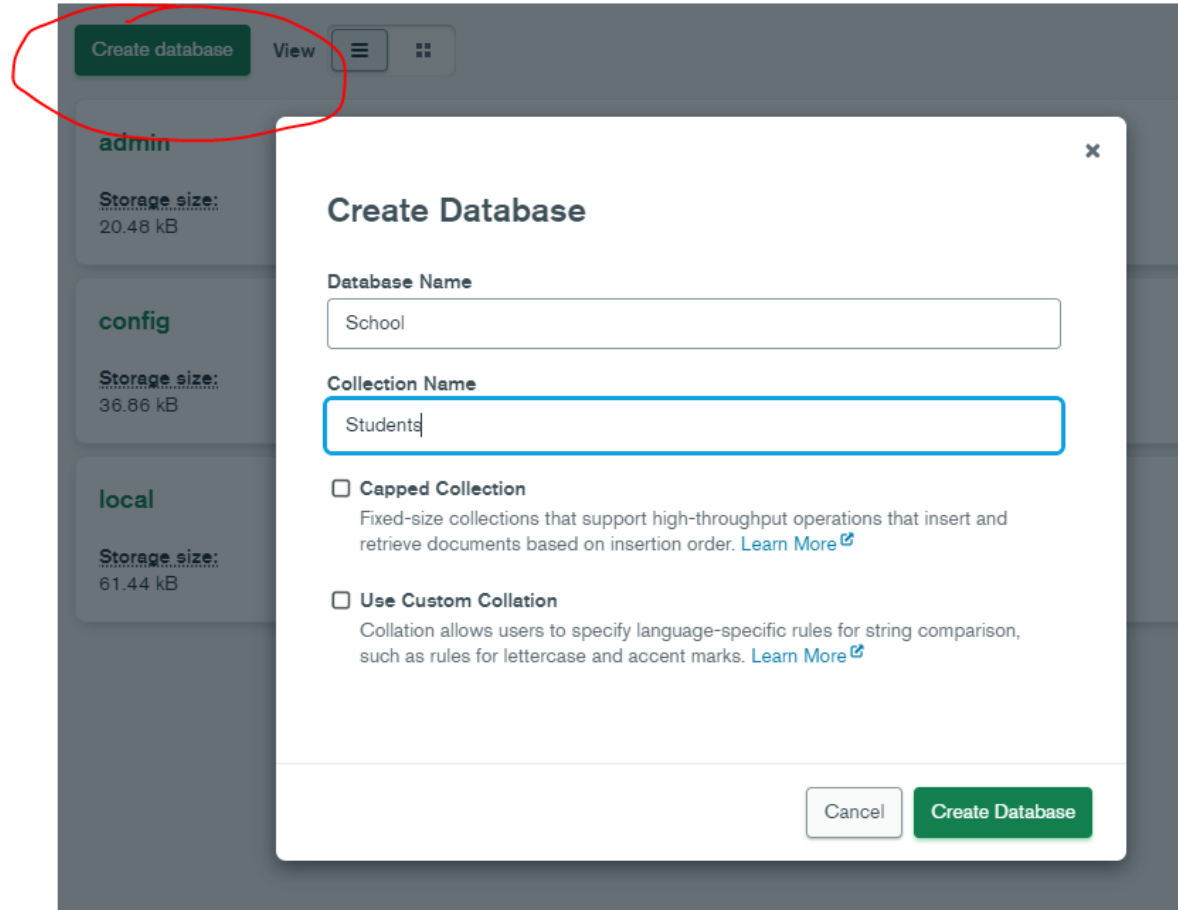
URI ⓘ Edit Connection String ☒

mongodb://localhost:27017

➤ Advanced Connection Options

Connect

Sukurkime paprastą DB

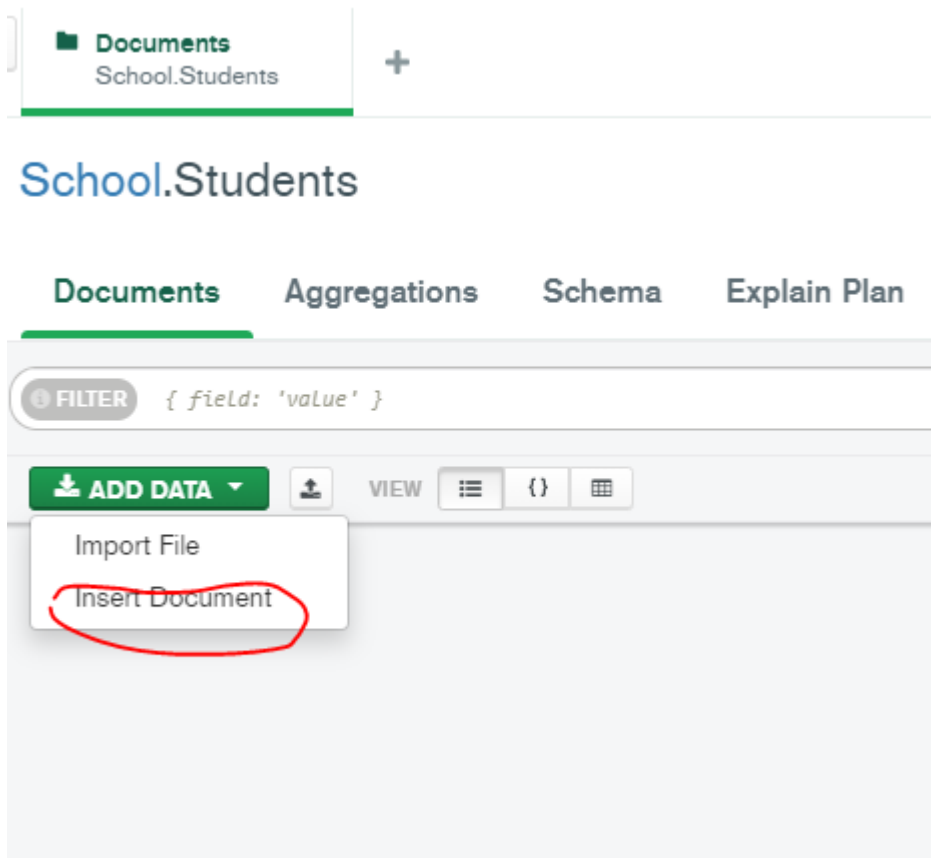


The screenshot shows the MongoDB Compass interface. In the top left, the 'Create database' button is highlighted with a red circle. A modal dialog titled 'Create Database' is open in the center. The dialog has a close button (x) in the top right corner. It contains the following fields and options:

- Database Name:** A text input field containing 'School'.
- Collection Name:** A text input field containing 'Students'.
- ☐ **Capped Collection**
Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [Learn More](#)
- ☐ **Use Custom Collation**
Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [Learn More](#)

At the bottom right of the dialog are two buttons: 'Cancel' and 'Create Database'.

Įdėkime dokumentą



Insert to Collection School.Students

VIEW

```
1  /**
2   * Paste one or more documents here
3   */
4  {
5    "_id": {
6      "$oid": "626836302d62fa4437dc589d"
7    },
8    "name": {"first_name": "John", "last_name": "Paul"},
9    "age": 17
10 }
```





Cancel

Insert

Find

Documents Aggregations Schema Explain





FILTER {age:17}

ADD DATA  **VIEW**   

```
_id: ObjectId('626836302d62fa4437dc589d')
  name: Object
    first_name: "John"
    last_name: "Paul"
  age: 17
```

Documents Aggregations Schema Explain

FILTER {"name.first_name": "John"}

ADD DATA  **VIEW**   

```
_id: ObjectId('626836302d62fa4437dc589d')
  name: Object
    first_name: "John"
    last_name: "Paul"
  age: 17
```

Sukurkime didelę DB

Duomenų įkėlimas

Įkelkime Lietuvos įmonių katalogą.

1. sukurkite nauja db lietuvos_imones
2. sukurkite naujas kolekcijas imones, kategorijos
3. Išskeiskite lietuvos_imones.zip failą
4. į kolekcijas suimportuokite atitinkamus failus

Find

lietuvas_imones.imones

253.6k

DOCUMENTS

1

INDEXES

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER [{"imonės_amžius": "9 metai 10 mėnesių 1 diena"}]

► OPTIONS

FIND

RESET



ADD DATA



VIEW



Displaying documents 1 - 20 of 32



REFRESH

```
_id: ObjectId('5fb07731261973556a11202b')
pavadinimas: "UAB "OLAN LT""
> kategorijos: Array
imonės_kodas: "302583298"
```

Text indeksas

- Jeigu norime ieškoti tekste reikia sukurti teksto indeksą
- Tai turime padaryti Mongo shell
 - `docker exec -it mongodb bash`
 - `mongosh`
 - `use lietuvas_imones`
 - `db.imones.createIndex({"pavadinimas":"text"})`
- Dabar galime naudoti komandą `$text`, kuri leidžia efektyviai ieškoti tekste

Find

lietuvos_imones.imones

253.6k 2
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER `{ $text: { $search: "maxima" } }`

► OPTIONS

FIND

RESET

↺

...

ADD DATA

VIEW

VIEW

VIEW

VIEW

VIEW

VIEW

Displaying documents 1 - 7 of 7

<

>

REFRESH

```
_id: ObjectId('5fbc977e025fc0398f7f8638')
pavadinimas: "Maxima, UAB"
imonės_kodas: "132477256"
vadovas: "Vilius Rimkus, direktorius"
adresas: "Naugarduko g. 84, LT-03160 Vilnius"
telefonas: "+370 5 2744337"
darbuotojai: "1 darbuotojas (apdraustasis)"
sd_draudėjo_kodas: "448311"
creditinfo_co: "Creditinfo CO sistemoje skolų neregistruota"
pardavimo_pajamos: "2015: 0 - 5 000 € be PVM Pajamų istorija »"
imonės_amžius: "29 metai 9 mėnesiai 18 dienų"
įvertinimas: "5.5 / 10 (įvertino 11)"
meta: Object
```

```
_id: ObjectId('5fbc27eb1a11a488fa67e3c3')
pavadinimas: "Maxima pecunia, MB"
imonės_kodas: "303476388"
vadovas: "Kristina Vodnickytė-Astrakova, likvidatorius"
adresas: "Kaunas"
mobilus_telefonas: "+370 620 11930"
tinklapis: "http://www.print4print.lt"
sd_draudėjo_kodas: "2309620"
creditinfo_co: "Creditinfo CO sistemoje skolų neregistruota"
```

MongoDB naudojimas kartu su Python

- Naudosime Pymongo biblioteką.
- Pvz. yra pateiktas `mongo_db_ir_python.ipynb`