

Mongo DB

NoSQL dokumentna baza

A screenshot of a Google search for 'mongodb'. The search results include a Udemy course 'MongoDB Fundamentals For Developers-Learn By Exercises' and the official MongoDB website. The website snippet describes MongoDB as 'the most popular database for modern apps' and lists links for documentation and manual. On the right, there is a sidebar with 'Razvijalci: MongoDB Inc.', 'Napisano v jeziku: C++, Go, JavaScript, Python', and a 'Drugi iščejo tudi' section featuring logos for PostgreSQL, MySQL, and Redis. Below that, a 'Več rezultatov o' section shows a stock price for MongoDB (MDB) at 196,28 \$ with a 1.01% increase.

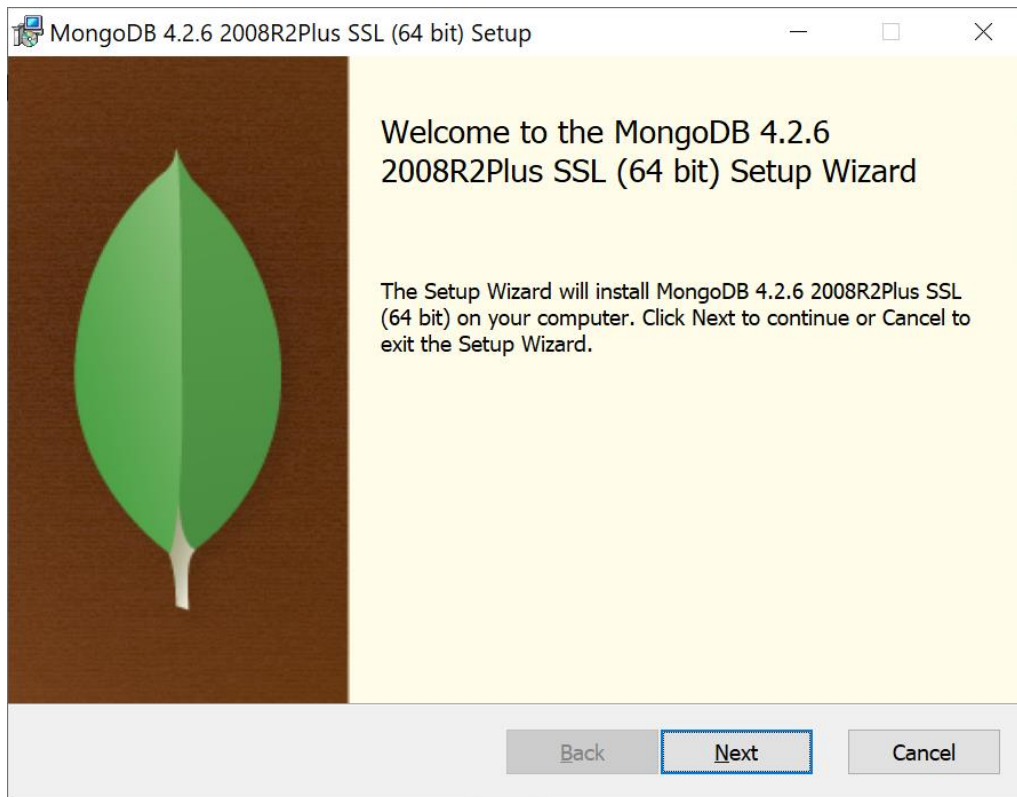
Kliknemo download center

Izberemo Server

A screenshot of the MongoDB download center page. The page title is 'mongodb.com/download-center/community'. It features a navigation bar with 'Cloud', 'Server', and 'Tools' tabs, with 'Server' currently selected. Below the navigation, there is a section titled 'Select the server you would like to run:' with two options: 'MongoDB Community Server' (labeled 'FEATURE RICH. DEVELOPER READY.') and 'MongoDB Enterprise Server' (labeled 'ADVANCED FEATURES. PERFORMANCE GRADE.'). Under the 'MongoDB Community Server' option, there are dropdown menus for 'Version' (4.2.6 (current release)) and 'OS' (Windows x64), and a 'Package' dropdown (MSI). A prominent green 'Download' button is located below these options. To the right, there is a list of links: 'Release notes', 'Changelog', 'All version binaries', and 'Installation instructions'. A chat icon is visible in the bottom right corner.

Izberemo OS in Download

Dobimo MSI 264MB, ki ga namestimo v CMD oknu z administratorskimi pravicami!



Naredimo direktorij za bazo npr.: C:\MongoDB\DB

In za log: C:\MongoDB\Log

To običajno ne deluje – glej <https://stackoverflow.com/questions/52877759/service-mongodb-servermongodb-failed-to-start-verify-that-you-have-sufficien>

Lahko pa po inštalaciji skopiramo tja, če želimo in stvar začuda deluje.

MongoDB 4.2.6 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

Install MongoDB as a Service

Run service as Network Service user

Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

Data Directory:

Log Directory:

< Back Next > Cancel

V command prompt z ukazom Whoami: dobimo domena\uporabnik

MongoDB Compass

Install MongoDB Compass

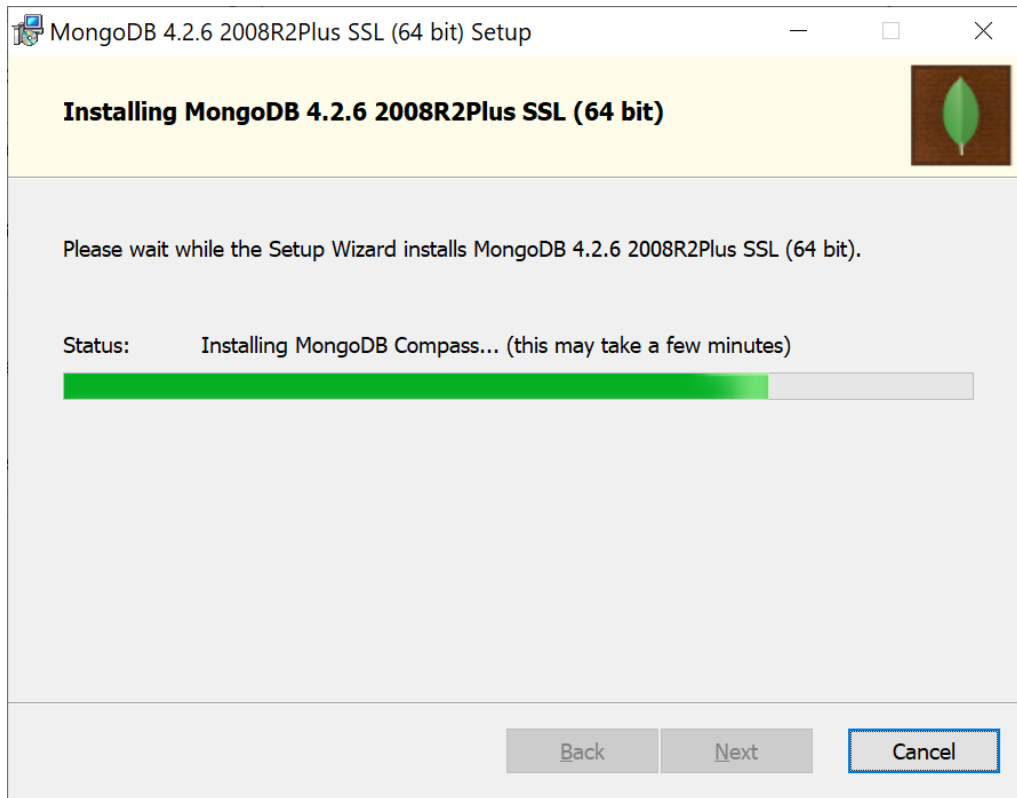
MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: <https://www.mongodb.com/products/compass>

Install MongoDB Compass

Back Next Cancel

Kljukica Install MongoDB Compass – GUI za urejanje in pregled baze



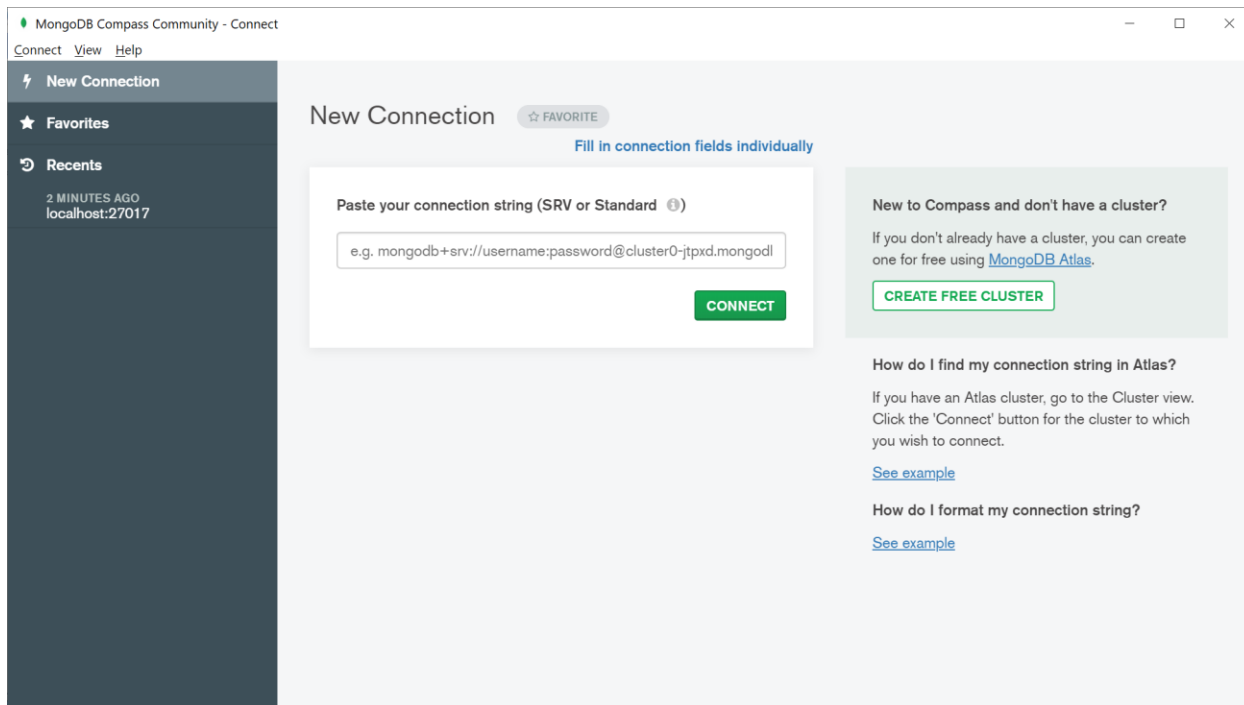
Traja precej dolgo

Potem se namesti MongoDB Compass

Čeprav mu poveš, kam naj se namesti, se on seveda po svoje: C:\Program Files\MongoDB\Server\4.2

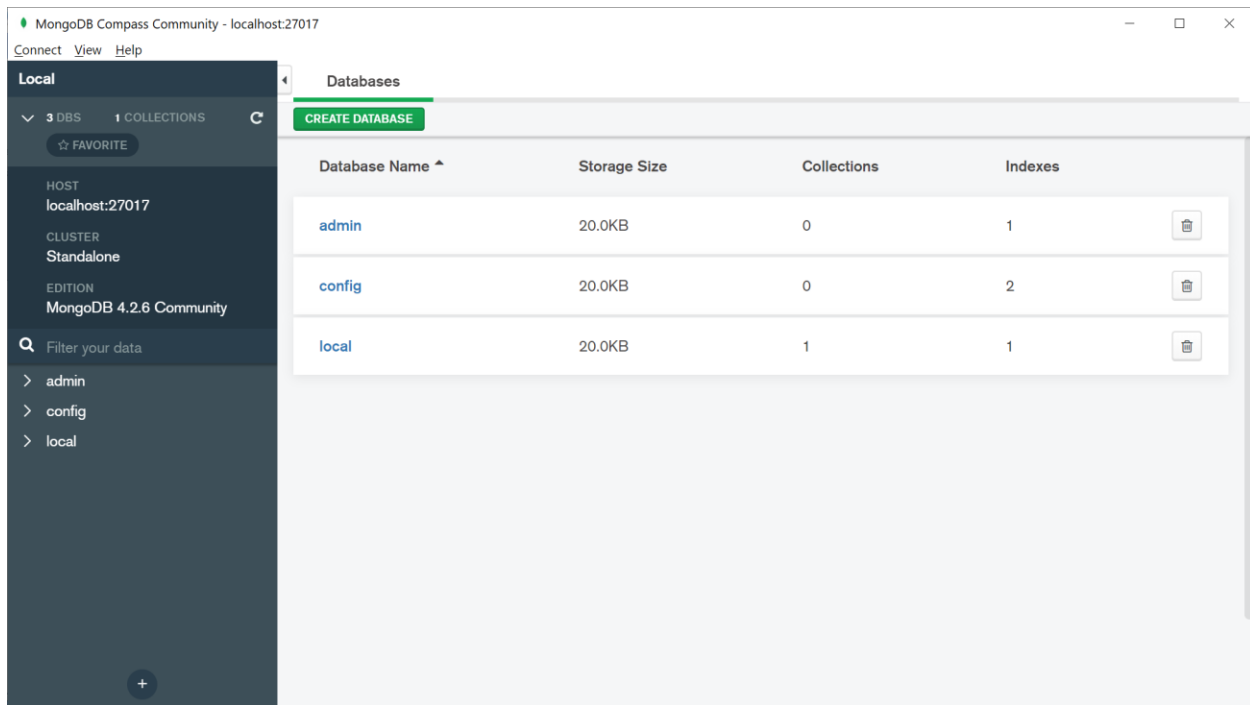
Tam je direktorij bin, kjer so exe datoteke: C:\Program Files\MongoDB\Server\4.2\bin

Ko se Compass odpre, kliknemo samo Connect



Še malo iz predavanj:

SQL	MongoDB
database (podatkovna baza)	database (podatkovna baza)
table (tabela)	collection (zbirka)
row (vrstica)	dokument JSON
column (stolpec)	JSON field (polje v dokumentu JSON)
primary key (primarni ključ)	polje <code>_id</code> v dokumentu JSON
indeks	indeks
group by	agregacija



Novo bazo lahko odpremo tukaj, je pa pametneje delati vse v ukazni vrstici.

Torej poženemo `C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe`

Odpre se ukazno okno:

```

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
Server has startup warnings:
2020-05-16T19:43:28.830+0200 I CONTROL [initandlisten]
2020-05-16T19:43:28.830+0200 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-05-16T19:43:28.830+0200 I CONTROL [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-05-16T19:43:28.831+0200 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc)
.

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>

```

Sedaj smo v lokalni bazi. Z ukazom `show dbs` pogledamo katere baze/sheme se tu nahajajo.

```
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
```

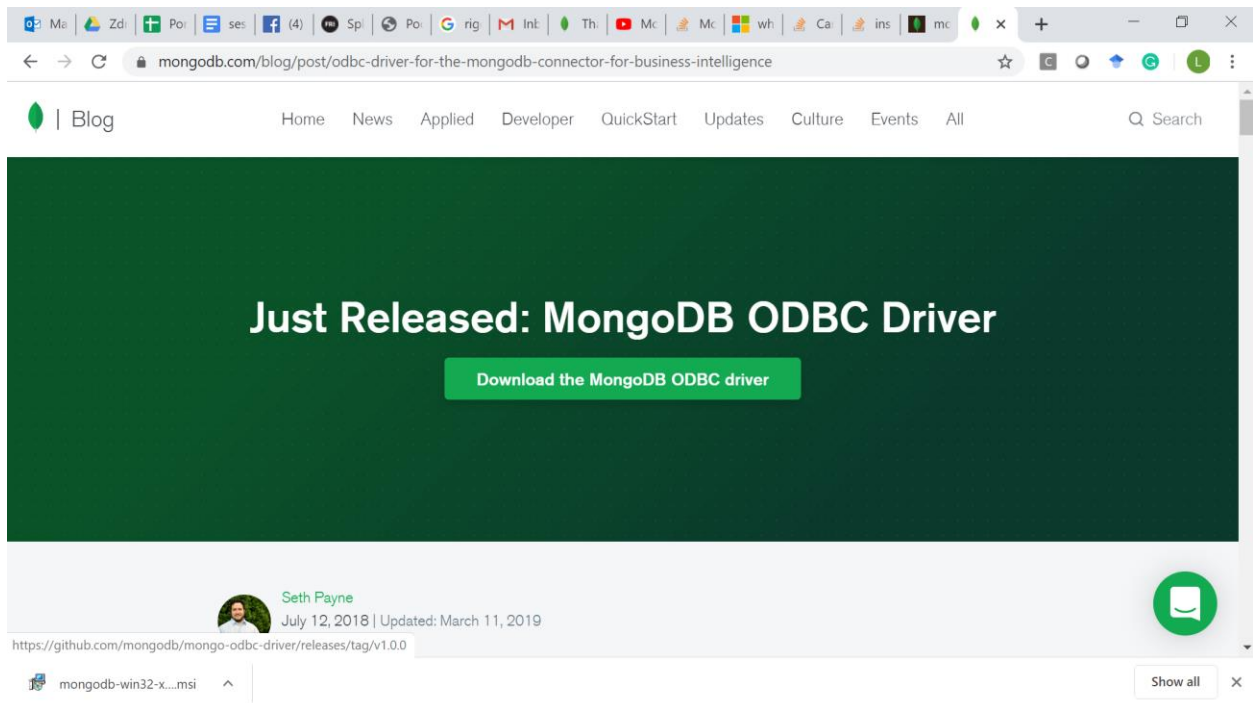
>

Novo shemo naredimo kar tako, da napišemo use abc. Abc baza je dejansko ustvarjena, ko dobi kakšno vsebino.

!!! velike in male črke so pomembne !!!

za komentar uporabimo //

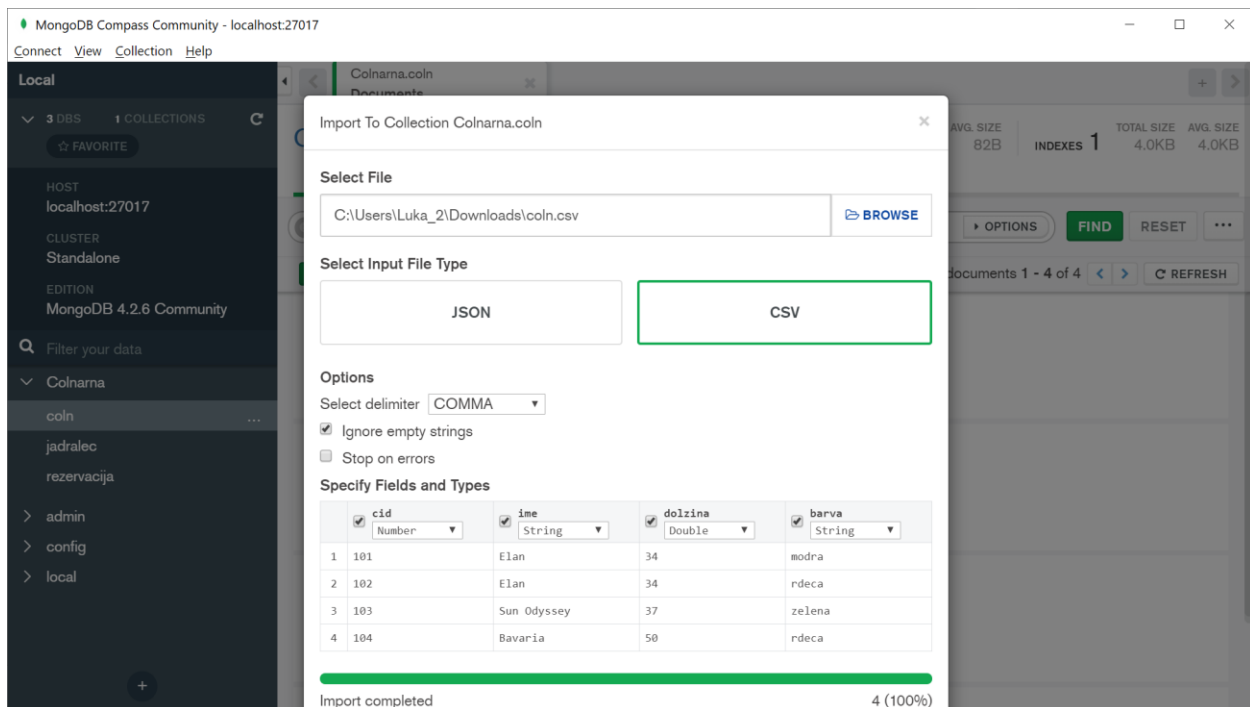
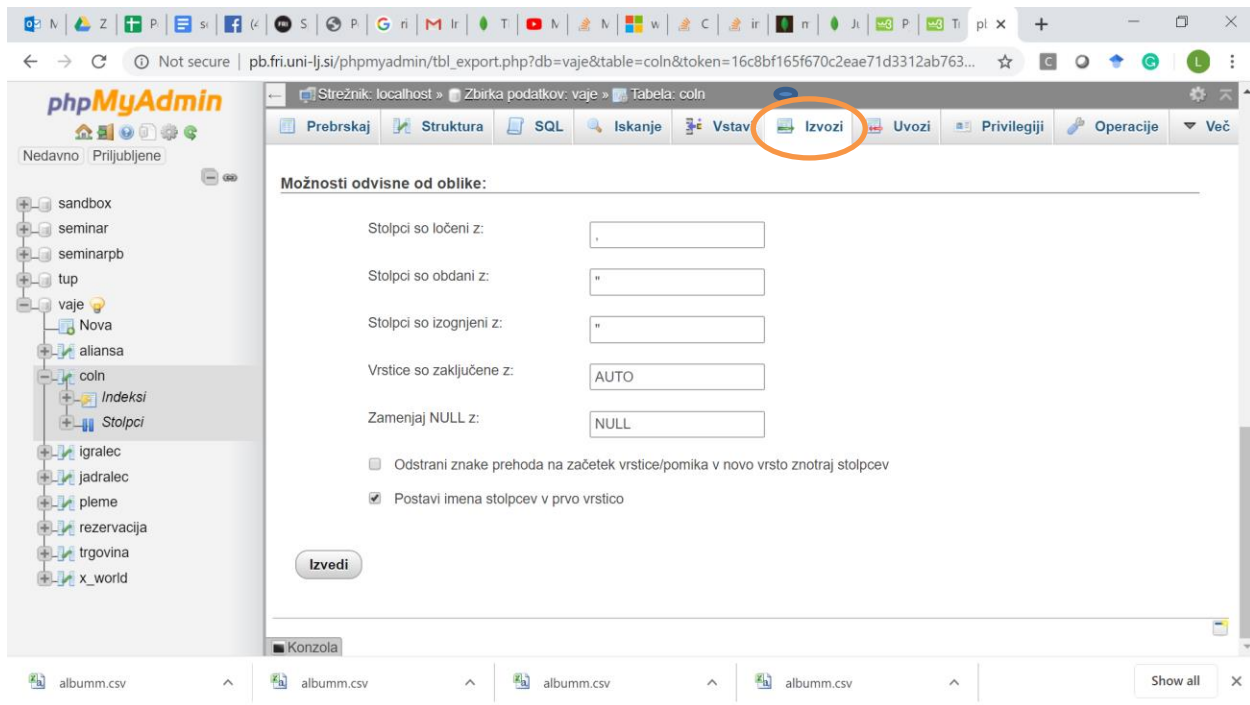
Za uvoz/izvoz podatkov lahko uporabimo ODBC

A screenshot of a web browser displaying a MongoDB blog post. The browser's address bar shows the URL 'mongodb.com/blog/post/odbc-driver-for-the-mongodb-connector-for-business-intelligence'. The page features a dark green header with the MongoDB logo and navigation links: Home, News, Applied, Developer, QuickStart, Updates, Culture, Events, All, and a search bar. The main content area has a dark green background with the text 'Just Released: MongoDB ODBC Driver' in white, followed by a green button that says 'Download the MongoDB ODBC driver'. Below this, the author's name 'Seth Payne' is shown with a profile picture, along with the date 'July 12, 2018' and 'Updated: March 11, 2019'. A link to the GitHub release page is provided: 'https://github.com/mongodb/mongo-odbc-driver/releases/tag/v1.0.0'. At the bottom, a download bar shows a file named 'mongodb-win32-x...msi' with a 'Show all' button.

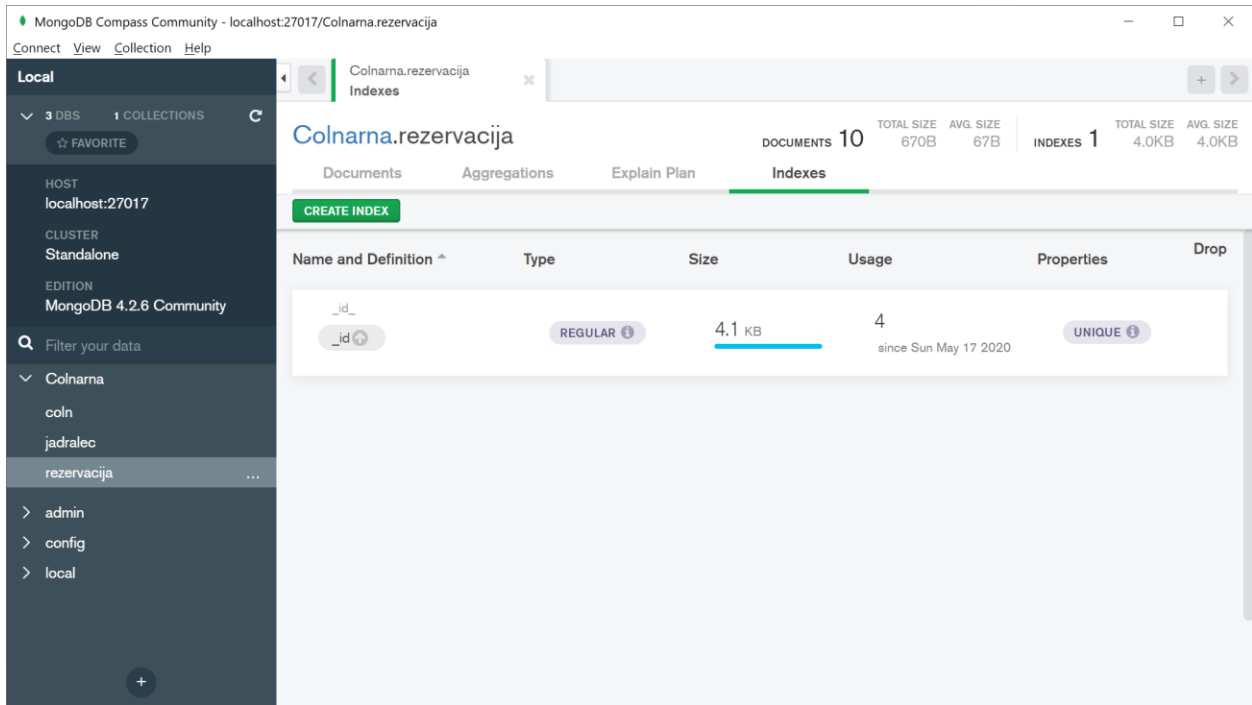
Tudi s pythonom se lahko povežete na MongoDB (pymongo):

https://www.w3schools.com/python/python_mongodb_getstarted.asp

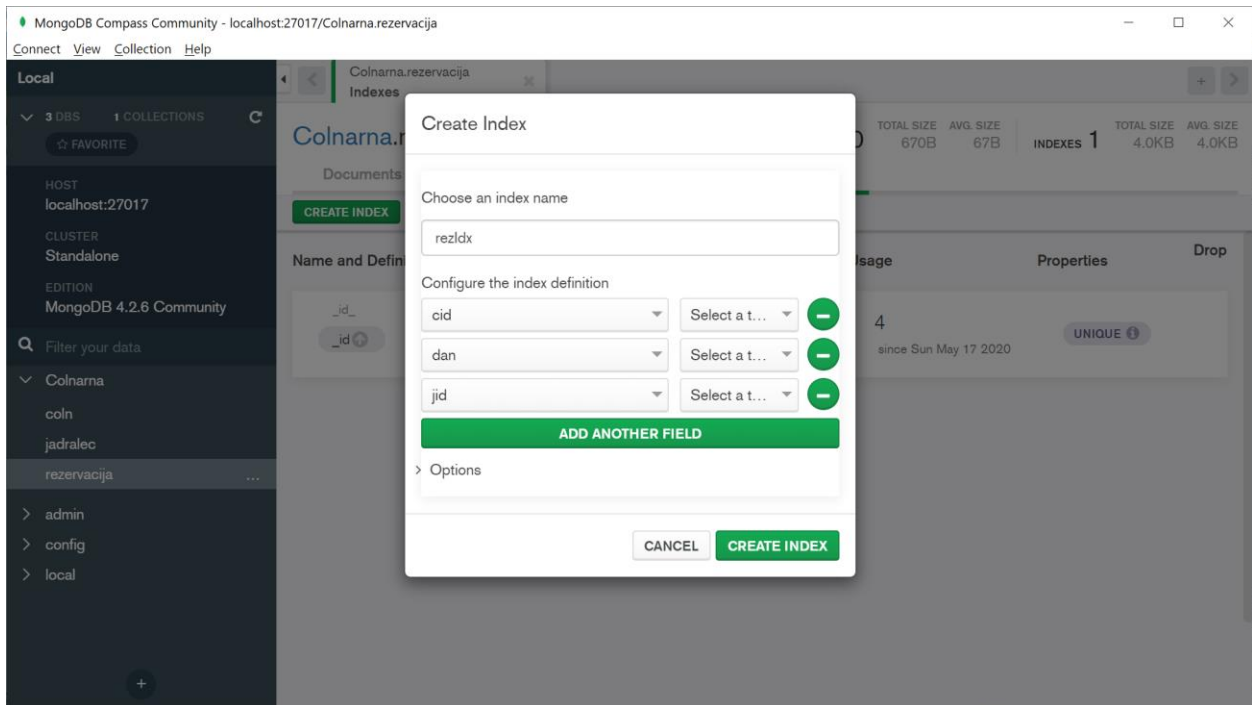
Ali pa navadno CSV datoteko. Za jadrance lahko izvozimo kar iz phpMyAdmin-a.

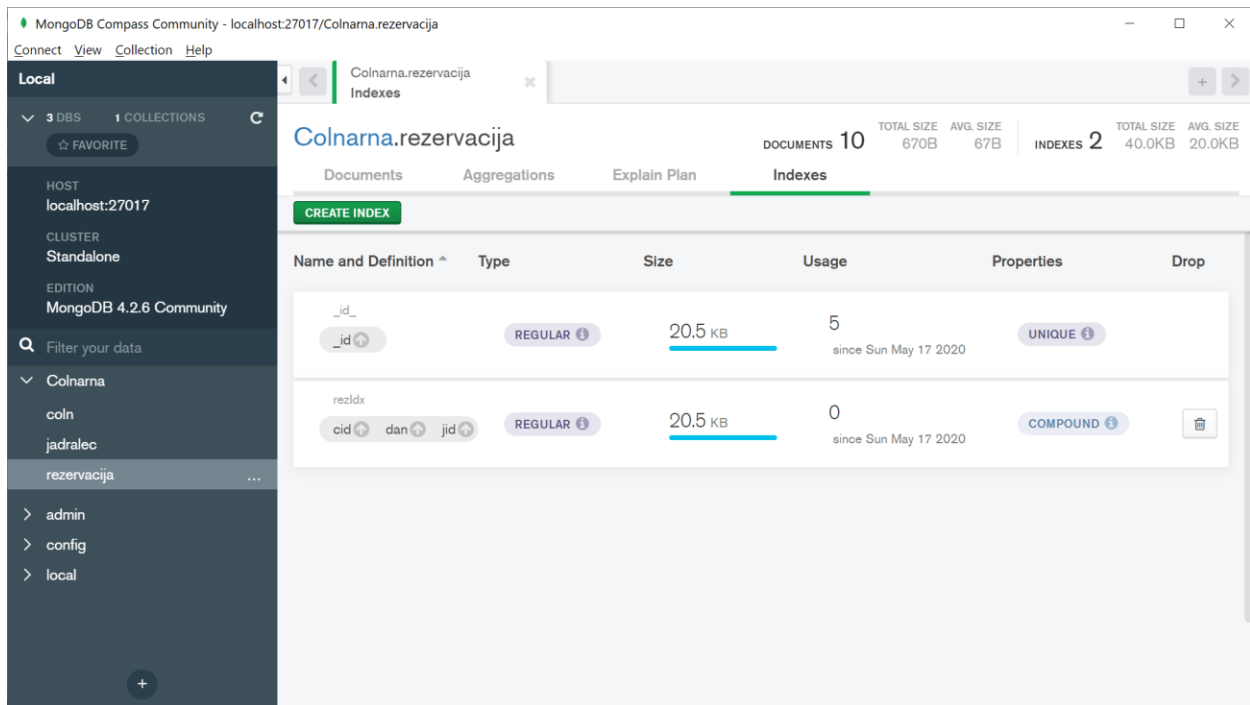


Lahko naredimo tudi indekse (_id je vedno indeksiran in avtomatsko dobi svoj GUID):



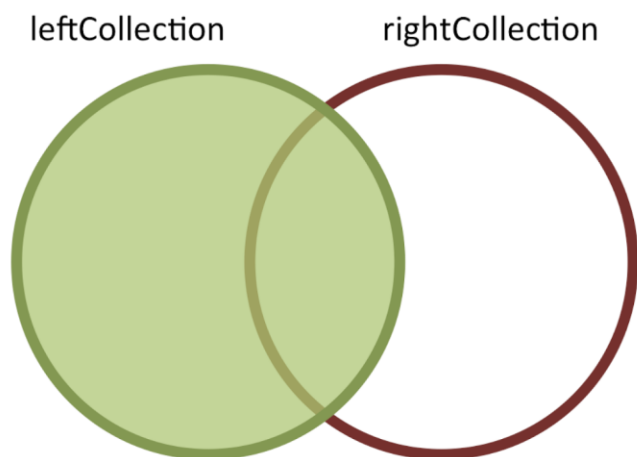
Lahko določimo indeks na skupek atributov:





Možni so tudi joini s funkcijo \$lookup, je pa dosti bolj zamudno (<https://www.mongodb.com/blog/post/joins-and-other-aggregation-enhancements-coming-in-mongodb-3-2-part-1-of-3-introduction>):

\$lookup



```
db.leftCollection.aggregate(
  [{
    $lookup:
      {
        from: "rightCollection",
        localField: "leftVal",
        foreignField: "rightVal",
        as: "embeddedData"
      }
  }
])
```

Recimo:

```
db.coln.aggregate(
  [{
```

```
        $lookup:
          {
            from: "rezervacija",
            localField: "cid",
            foreignField: "cid",
            as: "stik"
          }
        ]]
      )
```

V datoteki C:\Program Files\MongoDB\Server\4.2\log\mongod.log imamo vso zgodovino dela z bazo:

...

```
2020-05-17T17:08:41.023+0200 I NETWORK [conn21] received client metadata from 127.0.0.1:51187
conn21: { driver: { name: "nodejs", version: "3.5.6" }, os: { type: "Windows_NT", name: "win32",
architecture: "x64", version: "10.0.18362" }, platform: "'Node.js v12.4.0, LE (unified)", application: {
name: "MongoDB Compass Community" } }
```

```
2020-05-17T17:08:41.073+0200 I INDEX [conn20] index build: done building index _id_ on ns
Colnarna.Coln
```

```
2020-05-17T17:08:41.084+0200 I NETWORK [listener] connection accepted from 127.0.0.1:51188 #22
(5 connections now open)
```

```
2020-05-17T17:08:41.084+0200 I NETWORK [listener] connection accepted from 127.0.0.1:51189 #23
(6 connections now open)
```

```
2020-05-17T17:08:41.085+0200 I NETWORK [listener] connection accepted from 127.0.0.1:51190 #24
(7 connections now open)
```

```
2020-05-17T17:08:41.090+0200 I NETWORK [conn22] received client metadata from 127.0.0.1:51188
conn22: { driver: { name: "nodejs", version: "3.5.6" }, os: { type: "Windows_NT", name: "win32",
architecture: "x64", version: "10.0.18362" }, platform: "'Node.js v12.4.0, LE (unified)", application: {
name: "MongoDB Compass Community" } }
```

```
2020-05-17T17:08:41.093+0200 I NETWORK [conn23] received client metadata from 127.0.0.1:51189
conn23: { driver: { name: "nodejs", version: "3.5.6" }, os: { type: "Windows_NT", name: "win32",
architecture: "x64", version: "10.0.18362" }, platform: "'Node.js v12.4.0, LE (unified)", application: {
name: "MongoDB Compass Community" } }
```

```
2020-05-17T17:08:41.094+0200 I NETWORK [conn24] received client metadata from 127.0.0.1:51190
conn24: { driver: { name: "nodejs", version: "3.5.6" }, os: { type: "Windows_NT", name: "win32",
architecture: "x64", version: "10.0.18362" }, platform: "'Node.js v12.4.0, LE (unified)", application: {
name: "MongoDB Compass Community" } }
```

2020-05-17T17:08:44.679+0200 I SHARDING [conn21] Marking collection Colnarna.Coln as collection version: <unsharded>

2020-05-17T17:09:21.978+0200 I STORAGE [conn23] createCollection: Colnarna.jadralec with generated UUID: fa88cac4-0450-4e93-bc8c-740b2d17cd5d and options: {}

2020-05-17T17:09:22.011+0200 I INDEX [conn23] index build: done building index _id_ on ns Colnarna.jadralec

2020-05-17T17:09:22.091+0200 I SHARDING [conn24] Marking collection Colnarna.jadralec as collection version: <unsharded>

2020-05-17T17:09:40.425+0200 I STORAGE [conn22] createCollection: Colnarna.rezervacija with generated UUID: f2fc170a-6890-467d-a937-842db42c4d3a and options: {}

...

V konzoli odpremo C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe in lahko delamo z bazo v ukazni vrstici. Npr: show dbs pokaže vse zbirke. Ko smo v neki Novega uporabnika naredimo tako:

```
db.createUser(  
  {  
    user: "accountUser",  
    pwd: passwordPrompt(), // Or "<cleartext password>"  
    roles: [ "readWrite", "dbAdmin" ]  
  }  
)
```

Ukaz

```
use Colnarna  
db.coln.find();
```

izpiše dokumente v zbirki čoln.

Lahko dodamo kar novo zbirko npr.:

```
> db.test.insert({ime:"luka"},{priimek:"Sajn"});  
BulkWriteResult({  
  "writeErrors" : [],  
  "writeConcernErrors" : [],  
  "nInserted" : 2,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,
```

```
    "nRemoved" : 0,  
    "upserted" : [ ]  
  })
```

Tako dobimo novo zbirko test, z dvema dokumentoma:

```
> db.test.find();  
{ "_id" : ObjectId("5ec24169073b16ffddd81041"), "ime" : "luka" }  
{ "_id" : ObjectId("5ec24169073b16ffddd81042"), "priimek" : "Sajn" }
```

Sedaj smo dodali dva dokumenta, v resnici smo hoteli pa enega, za to moramo paziti ali delamo object ali seznam.

```
> db.test.insert({ime:"luka",priimek:"Sajn"})
```

```
BulkWriteResult({  
  "writeErrors" : [ ],  
  "writeConcernErrors" : [ ],  
  "nInserted" : 1,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,  
  "nRemoved" : 0,  
  "upserted" : [ ]  
})
```

```
> db.test.find();  
{ "_id" : ObjectId("5ec24169073b16ffddd81041"), "ime" : "luka" }  
{ "_id" : ObjectId("5ec24169073b16ffddd81042"), "priimek" : "Sajn" }  
{ "_id" : ObjectId("5ec24403073b16ffddd81043"), "ime" : "luka", "priimek" : "Sajn" }
```

Tukaj vidimo tudi, da so ustvarjeni id-ji zaporedni in ne naključni.

Od relacijskih baz smo navajeni, da imajo relacije strogo določeno strukturo preko sheme, tukaj pa to ne velja in lahko dodamo nekaj popolnoma vsebinsko drugačnega:

```
> db.test.insert({  
  ... ime:"Miha",  
  ... priimek:"Mraz",  
  ... seznamPredmetov: ["Modeliranje računalniških omrežij", "Zanesljivost in zmogljivost računalniških sistemov", "Nekonvencionalne platforme in metode procesiranja", "Veterinarska informatika (VF)", "Izbrana poglavja iz računalniških sistemov 1"],  
  ... ObjektNaslov: {  
    ... kabinet: "R3.59",  
    ... lab: "LRSS",  
    ... },  
  ... diplomanti: [{ime:"jan",leto:1995},  
  ... {ime:"anja",leto:1999},  
  ... {ime:"tone",leto:2020}  
  ... ]  
  ... })  
WriteResult({ "nInserted" : 1 })  
>
```

In dobimo:

```
> db.test.find().pretty(); //pretty lepše izpiše
{ "_id" : ObjectId("5ec24169073b16ffddd81041"), "ime" : "luka" }
{ "_id" : ObjectId("5ec24169073b16ffddd81042"), "priimek" : "Sajn" }
{
  "_id" : ObjectId("5ec24403073b16ffddd81043"),
  "ime" : "luka",
  "priimek" : "Sajn"
}
{
  "_id" : ObjectId("5ec24695073b16ffddd81044"),
  "ime" : "Miha",
  "priimek" : "Mraz",
  "seznamPredmetov" : [
    "Modeliranje računalniških omrežij",
    "Zanesljivost in zmogljivost računalniških sistemov",
    "Nekonvencionalne platforme in metode procesiranja",
    "Veterinarska informatika (VF)",
    "Izbrana poglavja iz računalniških sistemov 1"
  ],
  "ObjektNaslov" : {
    "kabinet" : "R3.59",
    "lab" : "LRSS"
  },
  "diplomanti" : [
    {
      "ime" : "jan",
      "leto" : 1995
    },
    {
      "ime" : "anja",
      "leto" : 1999
    },
    {
      "ime" : "tone",
      "leto" : 2020
    }
  ]
}
```

Update atributov lahko naredimo takole:

```
db.test.update({ime: "luka "},{ $set: { spol: "M "}});
```

tako dobimo:

```
> db.test.update({ime: "luka"}, { $set: { spol: "M" }});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.test.find().pretty();
{
  "_id" : ObjectId("5ec24169073b16ffddd81041"),
  "ime" : "luka",
  "spol" : "M"
}
```

Postarajmo Henrika za eno leto:

```
> db.jadralec.update({ime:"Henrik"},{$inc:{starost:1}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.jadralec.find({ime:"Henrik"});
{ "_id" : ObjectId("5ec1580ffcfa690e587540b3"), "jid" : 64, "ime" : "Henrik", "rating" : 7, "starost" : 36 }
{ "_id" : ObjectId("5ec1580ffcfa690e587540b5"), "jid" : 74, "ime" : "Henrik", "rating" : 9, "starost" : 35 }
```

Za uporabo v konzoli več na: <https://www.youtube.com/watch?v=pWbMrx5rVBE>