

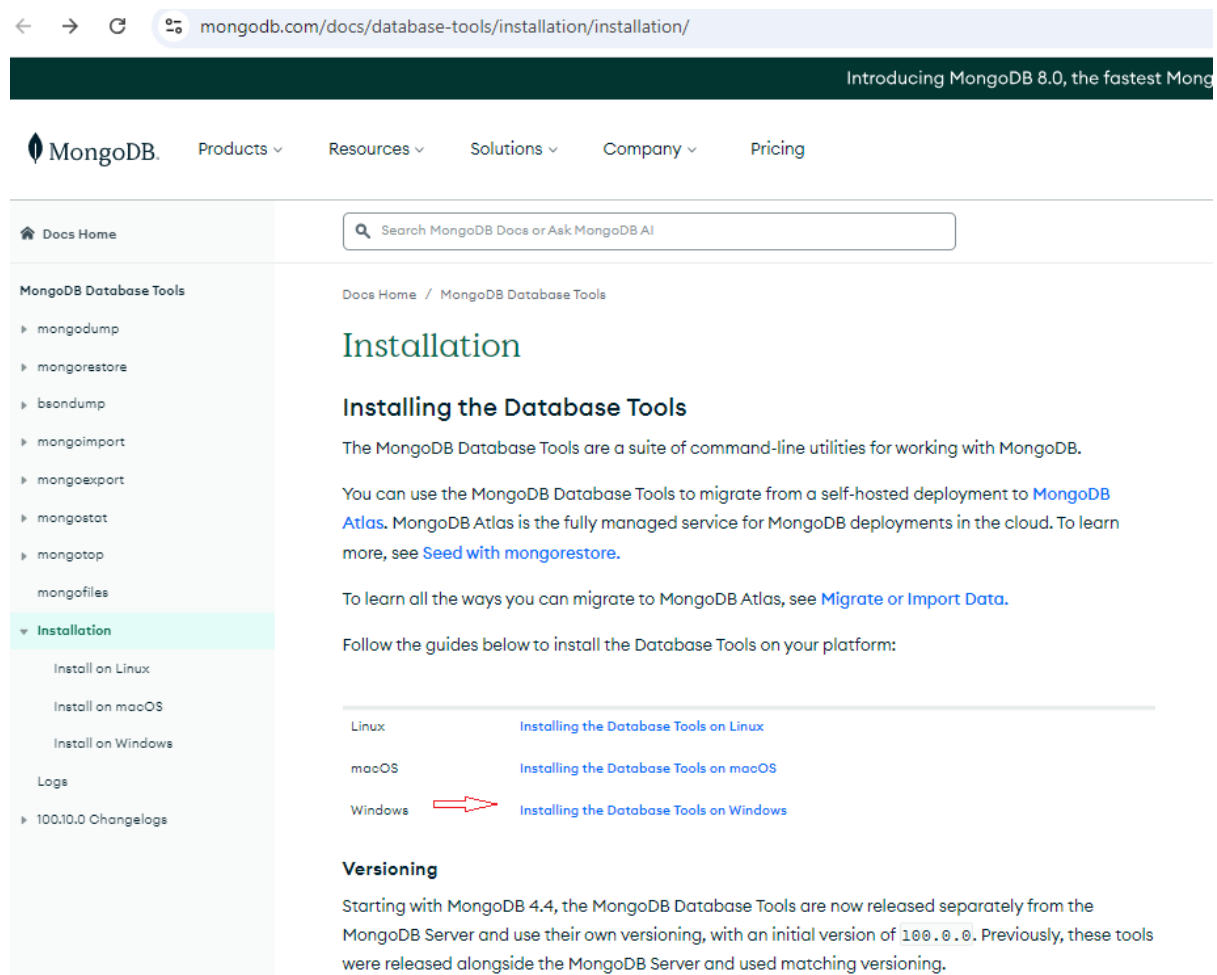
Če želimo večjo bazo z vzorčnimi podatki za MongoDB, so Atlas vzorčne baze dober vir:

<https://atlas-education.s3.amazonaws.com/sampledata.archive>

Uvozimo jih z: **`mongorestore --archive=sampledata.archive`**

Program Mongorestore dobimo v mongodb database tools:

<https://www.mongodb.com/docs/database-tools/installation/installation/>



The screenshot shows the MongoDB documentation page for installing the Database Tools. The browser address bar displays `mongodb.com/docs/database-tools/installation/installation/`. The page header includes the MongoDB logo and navigation links for Products, Resources, Solutions, Company, and Pricing. A search bar is located at the top right of the page content. The left sidebar contains a navigation menu with categories like MongoDB Database Tools, Installation, and 100.10.0 Changelogs. The main content area is titled "Installation" and includes a sub-section "Installing the Database Tools". This section explains that the tools are command-line utilities for working with MongoDB and can be used to migrate from a self-hosted deployment to MongoDB Atlas. It provides links to "Seed with mongorestore" and "Migrate or Import Data". Below this, there are three links for installing the tools on different operating systems: Linux, macOS, and Windows. The Windows link is highlighted with a red arrow. The "Versioning" section explains that starting with MongoDB 4.4, the tools are released separately from the MongoDB Server and use their own versioning, with an initial version of 100.0.0.

Introducing MongoDB 8.0, the fastest MongoDB

MongoDB. Products ▾ Resources ▾ Solutions ▾ Company ▾ Pricing

Docs Home

Search MongoDB Docs or Ask MongoDB AI

Docs Home / MongoDB Database Tools

Installation

Installing the Database Tools

The MongoDB Database Tools are a suite of command-line utilities for working with MongoDB.

You can use the MongoDB Database Tools to migrate from a self-hosted deployment to [MongoDB Atlas](#). MongoDB Atlas is the fully managed service for MongoDB deployments in the cloud. To learn more, see [Seed with mongorestore](#).

To learn all the ways you can migrate to MongoDB Atlas, see [Migrate or Import Data](#).

Follow the guides below to install the Database Tools on your platform:

Linux	Installing the Database Tools on Linux
macOS	Installing the Database Tools on macOS
Windows	Installing the Database Tools on Windows

Versioning

Starting with MongoDB 4.4, the MongoDB Database Tools are now released separately from the MongoDB Server and use their own versioning, with an initial version of 100.0.0. Previously, these tools were released alongside the MongoDB Server and used matching versioning.

Search MongoDB Docs or Ask MongoDB AI

- Windows Server 2012 and later

Installation

The MongoDB Database Tools can be installed with an MSI installer, or downloaded as a ZIP archive. Select the tab below depending on your desired installation method:

MSI Installer Zip Archive

1 Download the Database Tools MSI installer.

Open the [MongoDB Download Center](#). Using the drop-down menu on the right-hand side of the page:

1. Select the `Windows x86_64` Platform
2. Select the `msi` Package
3. Click the **Download** button

2 Run the MSI installer.

Double-click the downloaded MSI installer to install the Database Tools. During the install you may customize the installation directory if desired.

3 Make the DB Tools available in your PATH.

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

Tools

MongoDB Shell

MongoDB Compass (GUI)

Atlas CLI

Atlas Kubernetes Operator

MongoDB CLI for Cloud Manager and Ops Manager

MongoDB Cluster-to-Cluster Sync

Relational Migrator

MongoDB Database Tools

MongoDB Connector for BI

App Services CLI

Atlas SQL Interface

Mobile Apps



TOOLS

MongoDB Command Line Database Tools Download

The MongoDB Database Tools are a collection of command-line utilities for working with a MongoDB deployment. These tools release independently from the MongoDB Server schedule enabling you to receive more frequent updates and leverage new features as soon as they are available. See the [MongoDB Database Tools](#) documentation for more information.

Version
100.10.0

Platform
Windows x86_64

Package
msi

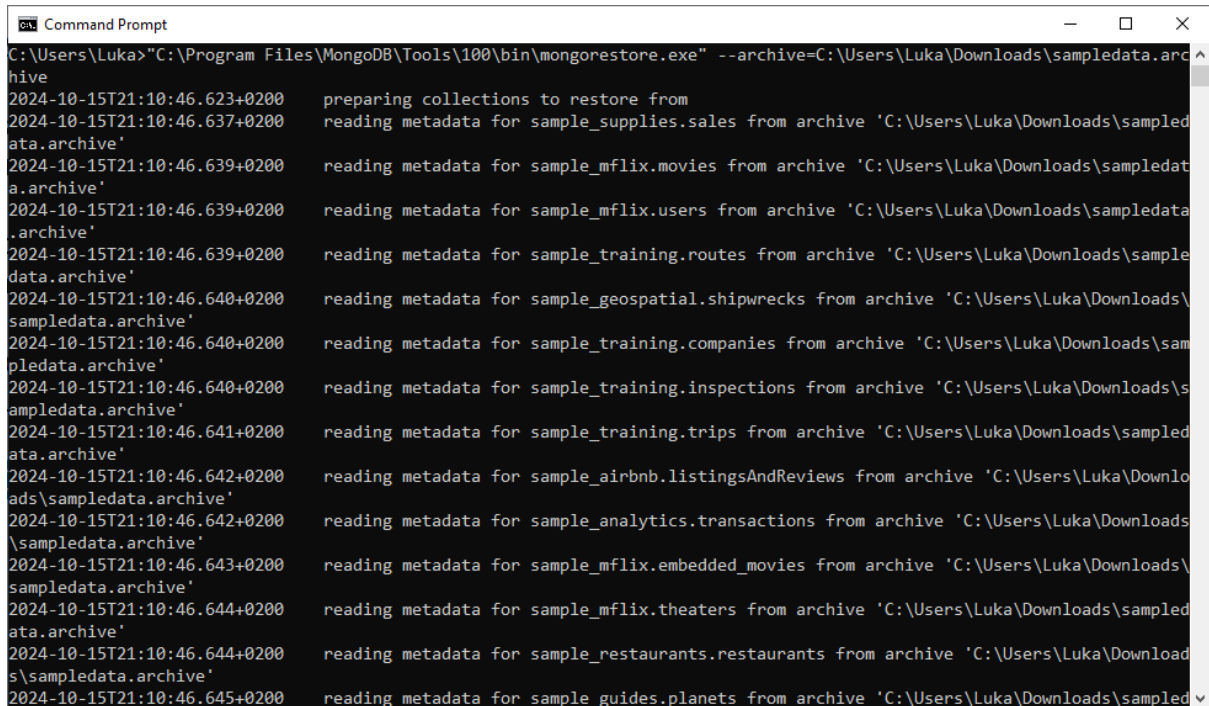
Download

Copy link

More Options

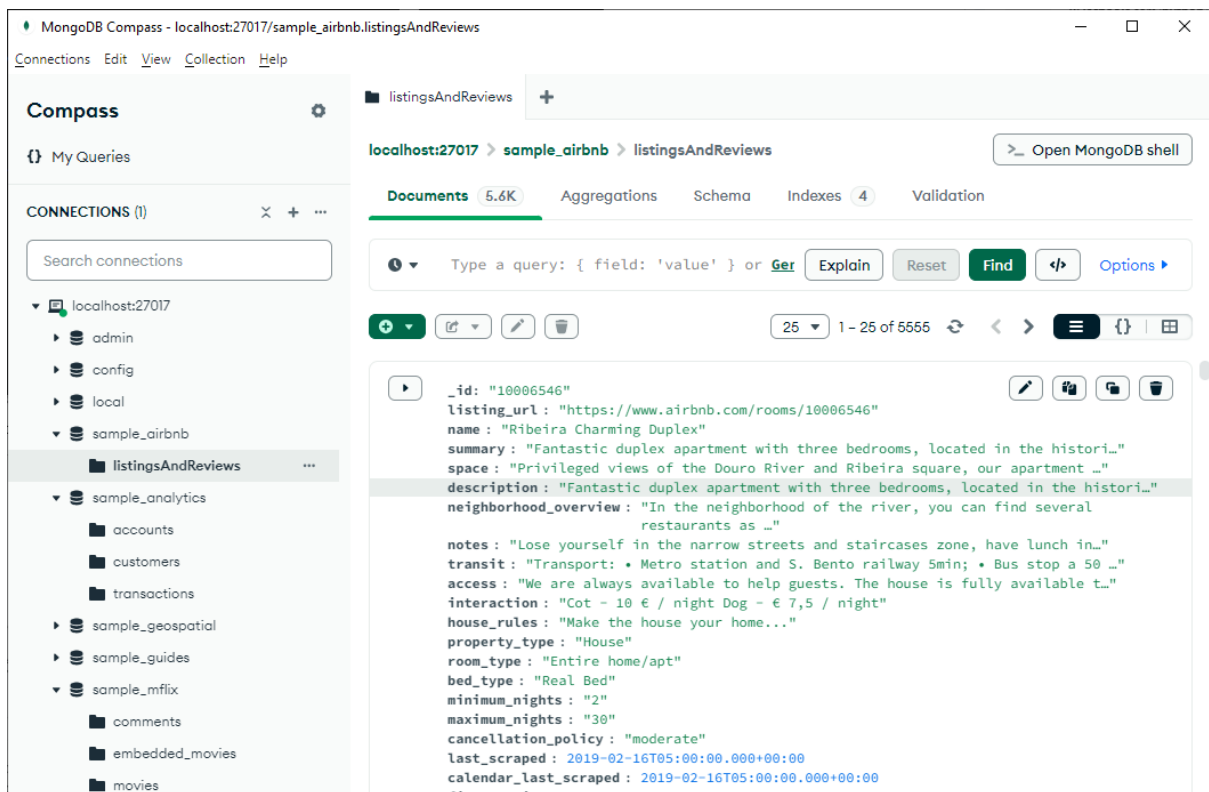
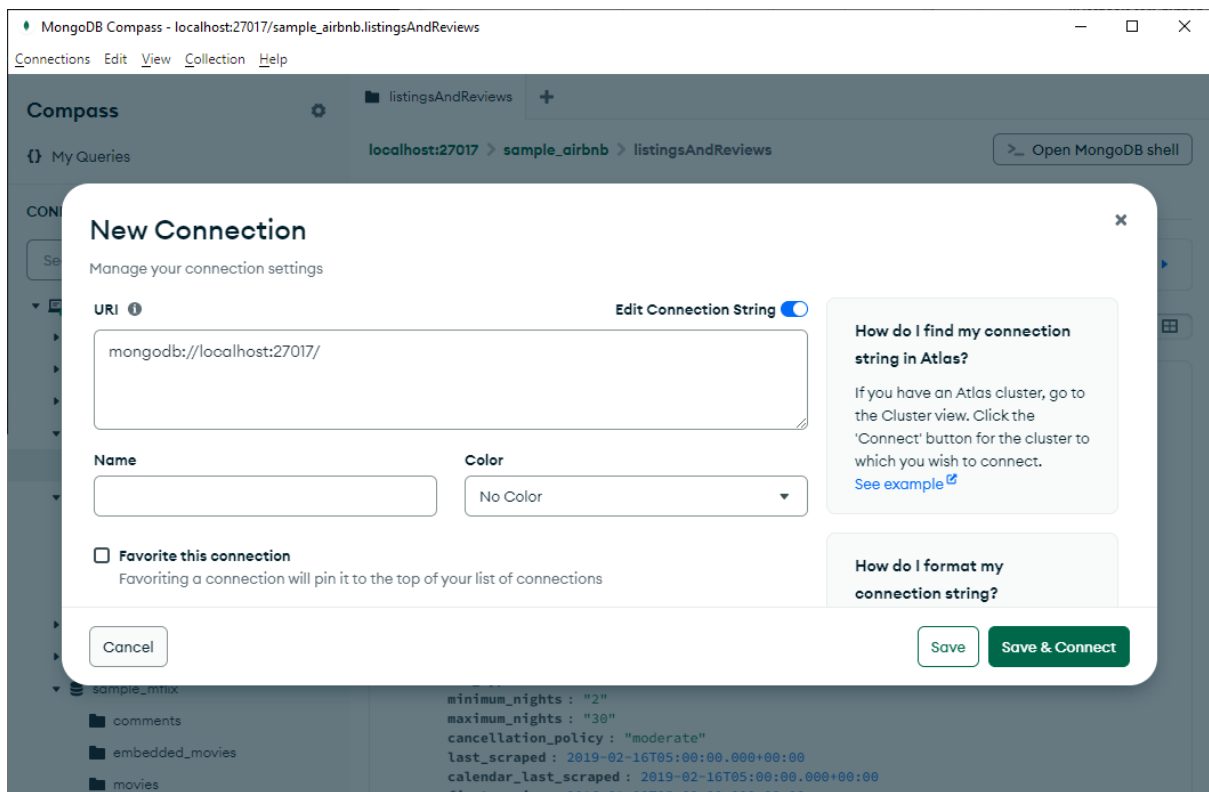
Tako dobimo mongorestore.exe, s katerim uvozimo sampledata.archive, ki smo ga na začetku prenesli.

```
"C:\Program Files\MongoDB\Tools\100\bin\mongorestore.exe" --archive=C:\Users\Luka\Downloads\sampledata.archive
```



```
Command Prompt
C:\Users\Luka>"C:\Program Files\MongoDB\Tools\100\bin\mongorestore.exe" --archive=C:\Users\Luka\Downloads\sampledata.archive
2024-10-15T21:10:46.623+0200    preparing collections to restore from
2024-10-15T21:10:46.637+0200    reading metadata for sample_supplies.sales from archive 'C:\Users\Luka\Downloads\sampled
ata.archive'
2024-10-15T21:10:46.639+0200    reading metadata for sample_mflix.movies from archive 'C:\Users\Luka\Downloads\sampl
ata.archive'
2024-10-15T21:10:46.639+0200    reading metadata for sample_mflix.users from archive 'C:\Users\Luka\Downloads\sampl
data.archive'
2024-10-15T21:10:46.639+0200    reading metadata for sample_training.routes from archive 'C:\Users\Luka\Downloads\sampl
data.archive'
2024-10-15T21:10:46.640+0200    reading metadata for sample_geospatial.shipwrecks from archive 'C:\Users\Luka\Downloa
ds\sampled
ata.archive'
2024-10-15T21:10:46.640+0200    reading metadata for sample_training.companies from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.640+0200    reading metadata for sample_training.inspections from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.641+0200    reading metadata for sample_training.trips from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.642+0200    reading metadata for sample_airbnb.listingsAndReviews from archive 'C:\Users\Luka\Downlo
ads\sampl
ata.archive'
2024-10-15T21:10:46.642+0200    reading metadata for sample_analytics.transactions from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.643+0200    reading metadata for sample_mflix.embedded_movies from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.644+0200    reading metadata for sample_mflix.theaters from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.644+0200    reading metadata for sample_restaurants.restaurants from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
2024-10-15T21:10:46.645+0200    reading metadata for sample_guides.planets from archive 'C:\Users\Luka\Downloa
ds\sampl
ata.archive'
```

Potem v Compassu kliknemo na + in save&connect in dobimo pogled na uvožene podatke:



Sedaj lahko naredimo kakšne poizvedbe:

MongoDB ponuja bogato zbirko operacij agregacije, ki omogočajo obdelavo in analizo podatkov iz zbirke, kot je tvoja uvožena zbirka iz `sampledata.archive`. Agregacije lahko uporabimo za različne namene, kot so filtriranje, združevanje podatkov, izvajanje matematičnih izračunov in podobno.

Tukaj je nekaj primerov uporabnih agregacij na zbirki podatkov:

1. Združevanje podatkov glede na polje

Če želiš na primer šteti število dokumentov, ki imajo enako vrednost v določenem polju (npr. glede na leto arhiva), lahko uporabiš operacijo `$group` za združevanje in `$count` za štetje.

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$group": {"_id": "$year", "count": {"$sum": 1}}}  
])
```

- Tukaj združimo dokumente glede na leto in preštujemo, koliko dokumentov pripada vsakemu letu.

2. Filtriranje podatkov na podlagi pogojev

Če želiš na primer filtrirati vse arhive, ki so starejši od določenega leta (npr. 2000), lahko uporabiš operacijo `$match`.

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$match": {"year": {"$lt": 2000}}}  
])
```

- To bo vrnilo vse dokumente, ki imajo leto manjše od 2000.

3. Uporaba `$project` za izbiro določenih polj

Operacijo `$project` lahko uporabimo, če želimo izbrati določena polja in pretvoriti obstoječa polja. Na primer, lahko izberemo samo ime dokumenta in leto arhiva:

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$project": {"_id": 0, "document_name": 1, "year": 1}}  
])
```

- Tukaj izberemo samo polji `document_name` in `year`, pri čemer skrijemo `_id` polje.

4. Štetje dokumentov z določenimi značilnostmi

Recimo, da želiš prešteti vse dokumente, ki izpolnjujejo določen pogoj, npr. imajo določeno oznako ali kategorijo.

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$match": {"category": "historical"}},  
    {"$count": "historical_documents"}  
])
```

- To bo preštelo vse dokumente v kategoriji "historical".

5. Združevanje in povprečje določenega polja

Če želiš na primer združiti dokumente po letu in izračunati povprečje števila strani, ki jih imajo dokumenti, lahko uporabiš `$group` in `$avg`.

```
python 📄 Kopiraj kodo  
  
db.archive.aggregate([  
    {"$group": {"_id": "$year", "average_pages": {"$avg": "$number_of_pages"}}}  
])
```

- Tukaj združimo dokumente glede na leto in izračunamo povprečno število strani dokumentov v posameznem letu.

6. Sortiranje dokumentov po določenem polju

Če želiš rezultate sortirati npr. po številu strani ali letnici, lahko uporabiš operacijo `$sort`.

```
python 📄 Kopiraj kodo  
  
db.archive.aggregate([  
    {"$sort": {"number_of_pages": -1}} # Sortiramo padajoče po številu strani  
])
```

- To bo vrnilo vse dokumente, razvrščene po številu strani v padajočem vrstnem redu.

7. Uporaba `$limit` za omejevanje števila rezultatov

Če želiš omejiti število vrnjenih rezultatov, uporabiš `$limit`. To je uporabno, če želiš na primer pridobiti samo prvih 5 dokumentov po številu strani.

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$sort": {"number_of_pages": -1}},  
    {"$limit": 5}  
])
```

8. Izvajanje kompleksnih operacij s pomočjo `$lookup` (join dveh zbirk)

Če imaš dve zbirki (npr. `archive` in `authors`), lahko uporabiš `$lookup`, da pridobiš podatke iz obeh na podlagi skupnega polja.

```
python Kopiraj kodo  
  
db.archive.aggregate([  
    {"$lookup": {  
        "from": "authors",  
        "localField": "author_id",  
        "foreignField": "id",  
        "as": "author_details"  
    }},  
    {"$unwind": "$author_details"}, # Razpakiramo avtorjeve podrobnosti  
    {"$project": {"document_name": 1, "author_details.name": 1, "year": 1}}  
])
```

- Tukaj združujemo podatke iz zbirke `archive` in `authors` na podlagi `author_id`, ter izpišemo ime dokumenta, avtorja in leto.

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>):

```
Recimo:  
db.coln.aggrega  
te(  
    {
```

`$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" : [ ],
  "writeConcernEr
rors" : [ ],
  "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" : [ ],
  "writeConcernEr
rors" : [ ],
  "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",
  "seznamPredmet
  ov" : [
    "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:{spot:
"M "}}); tako dobimo:
```

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

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.test.find().pretty();
```

```

{
  "_id" :
    ObjectId("5ec24169073b16ffddd81041"),
  "ime" : "luka",
  "spot" : "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>