



TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies



# MongoDB

*Lukáš Matějů*

20.3.2024 | DPB



# ČÁST I.: OPAKOVÁNÍ



# OPAKOVÁNÍ

- dokumentové databáze
  - označovány také jako
    - dokumentově orientované databáze
    - úložiště dokumentů
  - v současnosti asi nejpoužívanější typ NoSQL databází
  - v principu podobné key-value úložištím
    - zachován princip key-value (klíč-hodnota)
      - key jednoznačným identifikátorem value
    - ale value obsahují strukturovaná nebo částečně strukturovaná data
      - tzv. dokumenty
    - samotná data mohou být indexována a dotazována
      - indexy nad atributy dat
      - dotazy na strukturu dat i na prvky v této struktuře
    - je možné získat jen požadované části dokumentů
  - na úrovni databáze bez schématu

# OPAKOVÁNÍ

- dokumenty
  - základní stavební prvky dokumentových databází
  - zapouzdřují a kódují data v definovaném formátu (kódování)
    - implementace se liší databázi od databáze
  - používaná kódování
    - textová forma
      - XML, YAML, JSON
    - binární forma
      - BSON, PDF, MS Office dokumenty
  - identifikovány jednoznačným identifikátorem (klíčem)
    - typicky řetězec, URI nebo cesta
    - slouží pro přístup k dokumentům
      - ale i pro vkládání
    - často indexovány
      - rychlejší přístup k dokumentům

```
_id: ObjectId("5f8ef175c43ece2db0230f88")
title: "Post Four"
category: "Entertainment"
```

```
_id: ObjectId("5f8fe759c43ece2db0230f8a")
title: "Post Five"
likes: 81
user: Object
  name: "John Doe"
  status: "author"
  date: "Date()"
```

```
_id: ObjectId("5f9026800cbc092824d7e420")
source: "id2"
title: "Unknown"
user: Object
  name: "Jane Doe"
  gender: "female"
```



# OPAKOVÁNÍ

- dokumenty
  - koncepčně odpovídají objektům v OOP
  - volné schéma
    - dokumenty mohou být velmi komplexní
    - mohou obsahovat vnořené (embedded) dokumenty
    - nemusí obsahovat stejné oddíly, atributy, části nebo klíče
    - podobně jako objekty
    - vysoká míra flexibility
  - obecně data patřící k sobě ukládána do jednoho dokumentu
    - na rozdíl od relačních databází
    - usnadňuje přístup a práci s daty
  - k dokumentům často přidružena a uložena metadata



<https://expert.data.blog/category/bigdata/>

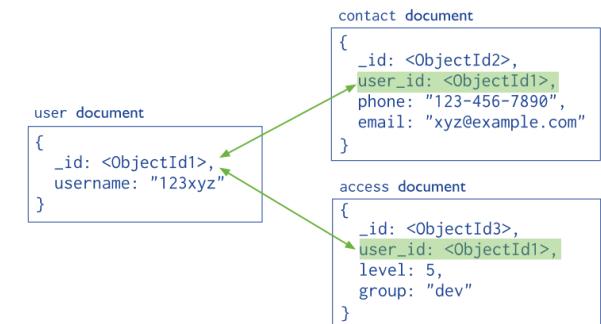


# OPAKOVÁNÍ

- dokumentové databáze
  - podporují standardní operace s daty (dokumenty)
    - operace CRUD
    - implementace se liší databázi od databáze
    - vytvoření (vložení) [creation]
    - čtení (dotazování, vyhledávání) [retrieval]
      - kromě vyhledávání podle klíče také podpora dotazovacího jazyka
        - vyhledávání v závislosti na obsahu (nebo metadatech)
    - aktualizace [update]
      - i jen části dokumentu
    - smazání [deletion]
    - mohou podporovat transakce
      - ACID
  - obecně se vyhýbají vazbám mezi dokumenty
    - případně dvě varianty řešení
      - embedded dokumenty
      - reference

```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```

Embedded sub-document  
Embedded sub-document

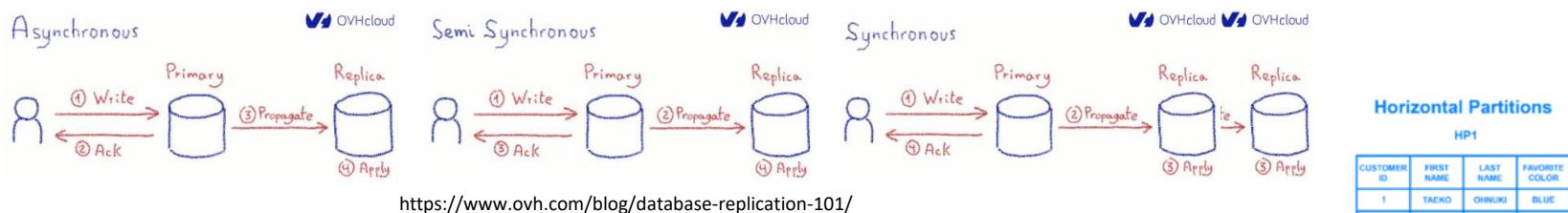


<https://docs.mongodb.com/manual/core/data-model-design/>



# OPAKOVÁNÍ

- distribuované dokumentové databáze
  - škálování
    - vlastnost systému zvládat rostoucí množství požadavků přidáváním dalších prostředků
    - horizontální škálování
  - replikace
    - automatická distribuce změn v originále do jeho kopií
    - u dokumentových databází nejčastěji typu master-slave



- sharding
  - databázový vzor pro horizontální škálování na více serverů
  - rozdělení záznamů na části (partitions, shards) umístěné na různých serverech

<https://www.digitalocean.com/community/tutorials/understanding-database-sharding>



# ČÁST II.: MongoDB



# MongoDB

- **humongo**us data
  - ukládání spousty a spousty dat
- dokumentová databáze
  - NoSQL
  - open source
  - multiplatformní



invision

ebay

Adobe

Google

SQUARESPACE

coinbase

SEGA

intuit

eharmony

EA

verizon<sup>®</sup>

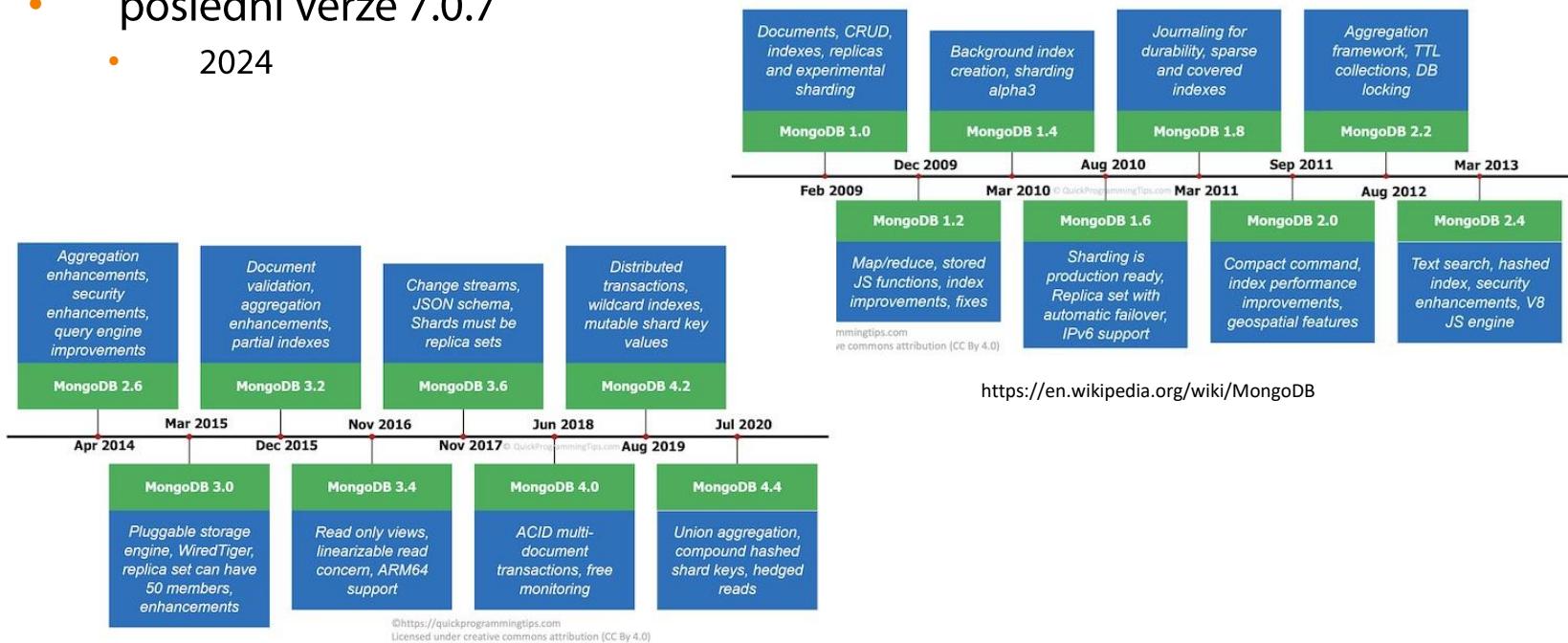
shutterfly

GOV.UK

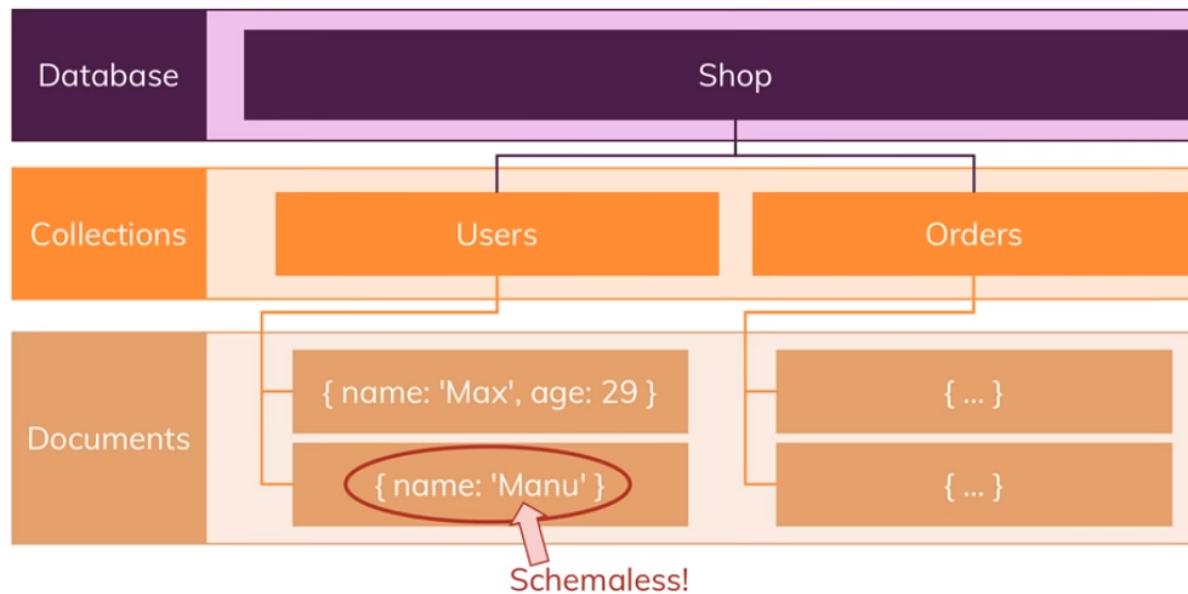
SAP

# MongoDB

- v aktivním vývoji
  - MongoDB Inc.
  - poslední verze 7.0.7
    - 2024



# MongoDB



<https://www.youtube.com/watch?v=VELru-FCWDM>



# MongoDB

- v databázi data ukládána v kolekcích dokumentů
  - kolekce
    - seznam dokumentů
    - odpovídá relaci (tabulce)
  - dokument
    - obsahuje data
    - reprezentován pomocí vnořených objektů / map
    - ve formátu BSON
      - binární JSON
      - přidané datové typy
    - odpovídá záznamu v relaci (řádku)
  - pohled
    - pouze ke čtení
    - zdrojem je kolekce nebo jiný pohled



# BSON DOKUMENT

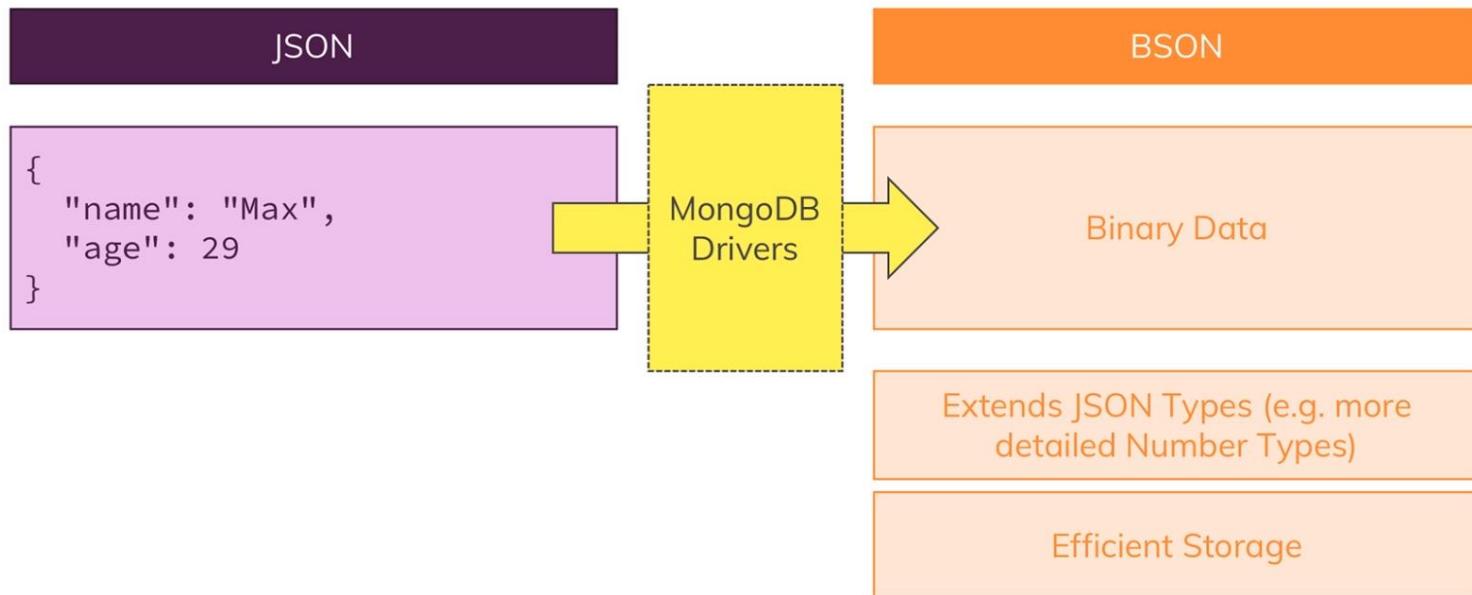
Looks like JSON. Example:

```
{  
    "_id" : ObjectId("7b33e366ae32223aee34fd3"),  
    "title" : "A blog post about MongoDB",  
    "content" : "This is a blog post about MongoDB",  
    "comments": [  
        {  
            "name" : "Frank",  
            "email" : fkane@sundog-soft.com,  
            "content" : "This is the best article ever written!"  
            "rating" : 1  
        }  
    ]  
}
```

<https://www.udemy.com/course/the-ultimate-hands-on-hadoop-tame-your-big-data>



# JSON VS. BSON



<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

# DOKUMENT

- maximum 16 MB
- volné schéma
- pole `_id`
  - primární klíč
  - automaticky přidán
  - `ObjectId`
    - 12 bytů
    - unikátní, rostoucí
- pole `comments`
  - obsahuje pole dalších vnořených dokumentů

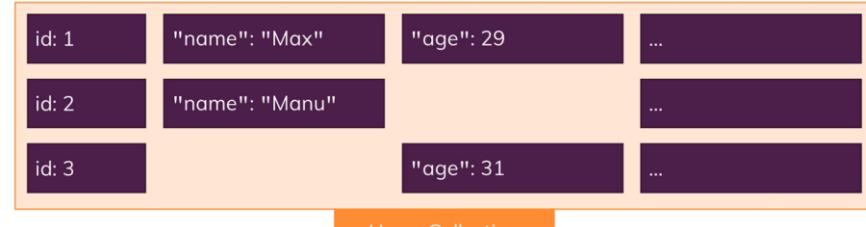
Looks like JSON. Example:

```
{  
  "_id" : ObjectId("7b33e366ae32223aee34fd3"),  
  "title" : "A blog post about MongoDB",  
  "content" : "This is a blog post about MongoDB",  
  "comments": [  
    {  
      "name" : "Frank",  
      "email" : fkane@sundog-soft.com,  
      "content" : "This is the best article ever written!",  
      "rating" : 1  
    }  
  ]  
}
```

<https://www.udemy.com/course/the-ultimate-hands-on-hadoop-tame-your-big-data>

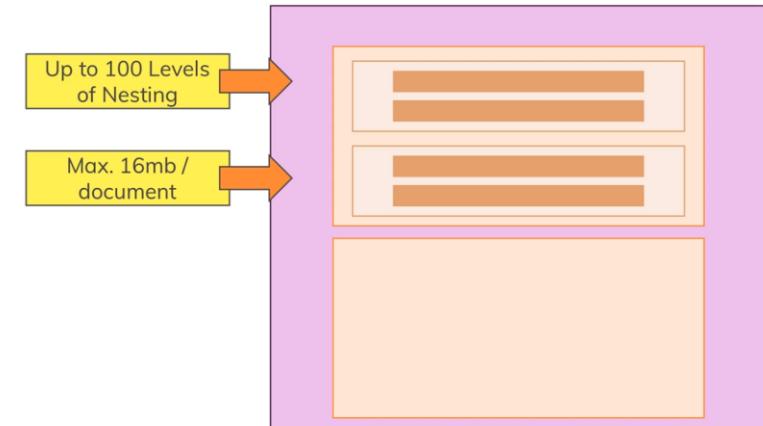
<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

No Schema!



# DOKUMENTY

- data patřící k sobě uložena v jednom dokumentu
  - na rozdíl od relačních databází
    - data v různých relacích propojena přes cizí klíče
    - přístup k datům přes náročné join dotazy
  - usnadňuje přístup a práci s daty
  - vazby mezi kolekcemi ideálně nejsou
    - možné ale jsou
    - ale je nutné je sloučit manuálně
      - dotaz na první dokument v první kolekci
      - dotaz na základě prvního dokumentu na druhou kolekci
- podpora vnořených dokumentů
  - embedded dokumenty
  - až 100 úrovní

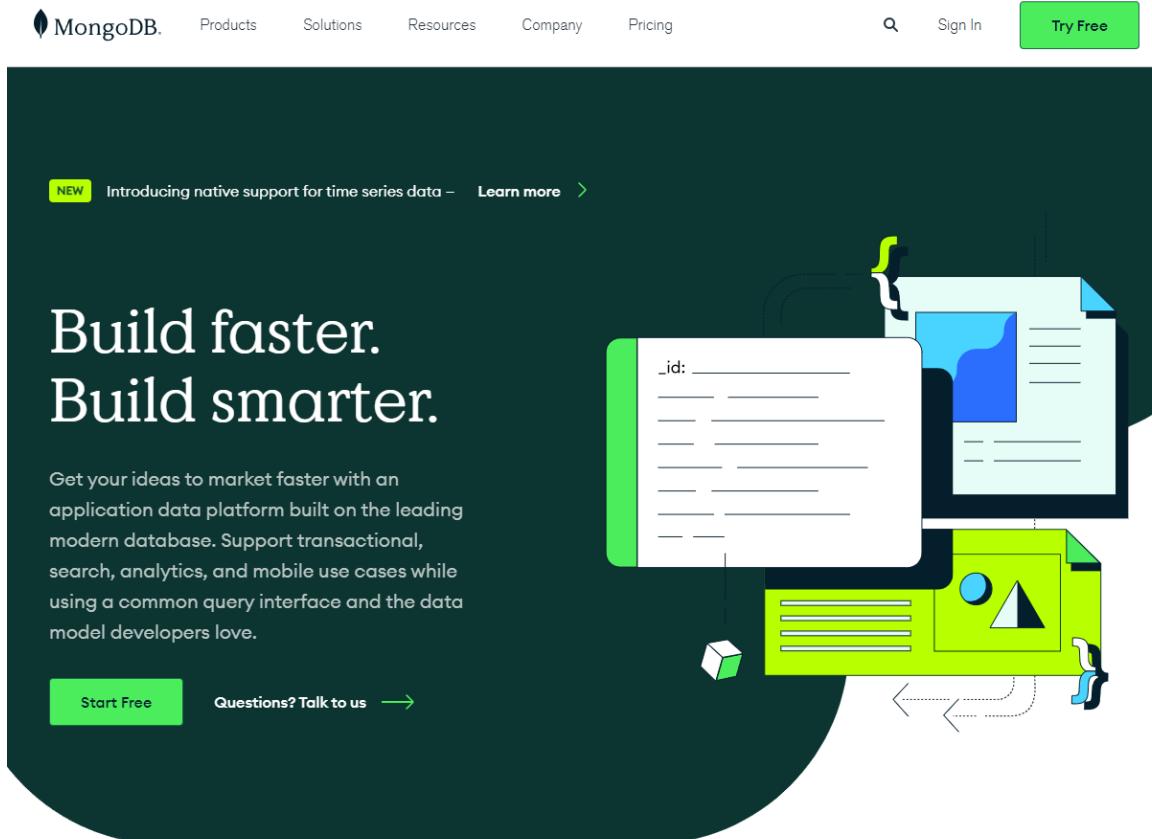


<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

# ČÁST III.: MongoDB PRAKTICKY



# MongoDB

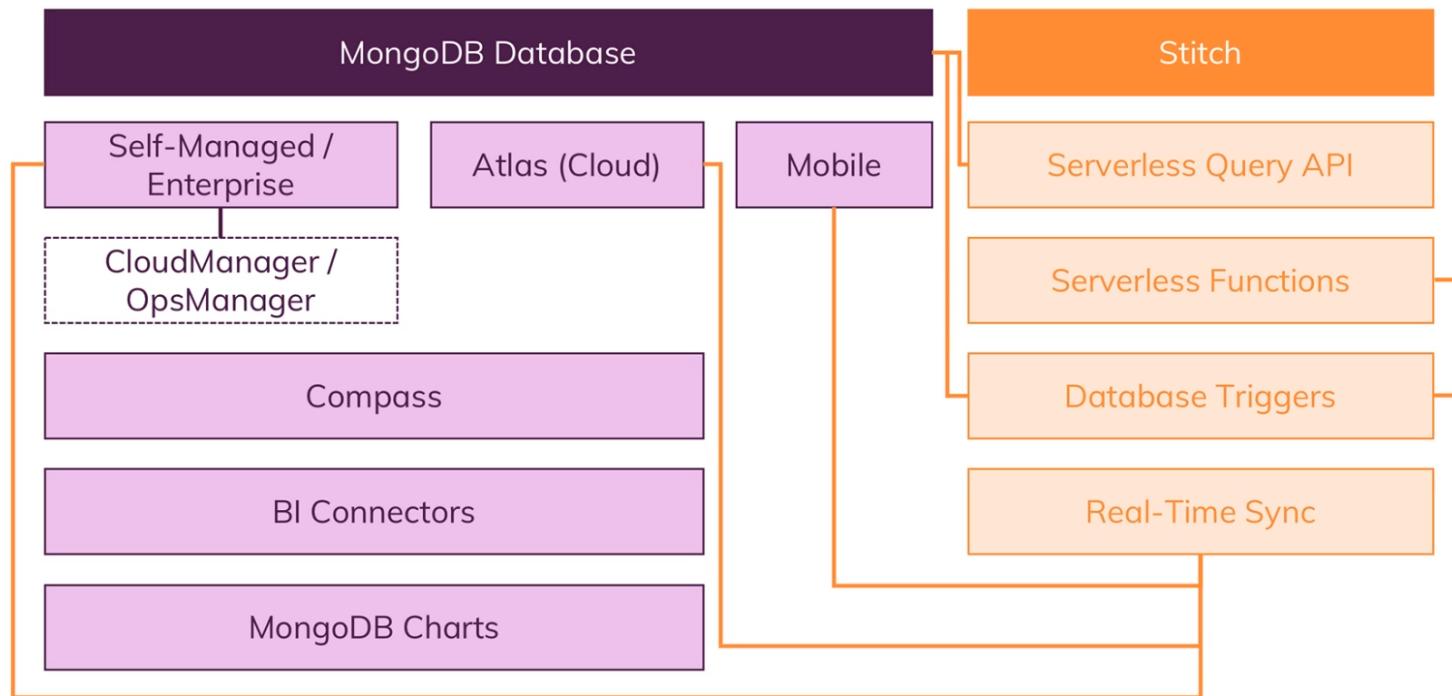


The screenshot shows the MongoDB homepage. At the top, there is a navigation bar with the MongoDB logo, a search icon, and links for Products, Solutions, Resources, Company, Pricing, Sign In, and Try Free. A green banner at the top right announces "Introducing native support for time series data" with a "Learn more" link. The main headline is "Build faster. Build smarter." Below it, a paragraph explains the benefits of using MongoDB for various use cases. Two call-to-action buttons are present: "Start Free" and "Questions? Talk to us". To the right, there is a stylized illustration of a document with fields like "\_id:", a blue chart, and a green cube, all connected by dashed lines to a central green base unit featuring a triangle and a circle.

<https://www.mongodb.com/>



# MongoDB EKOSYSTÉM

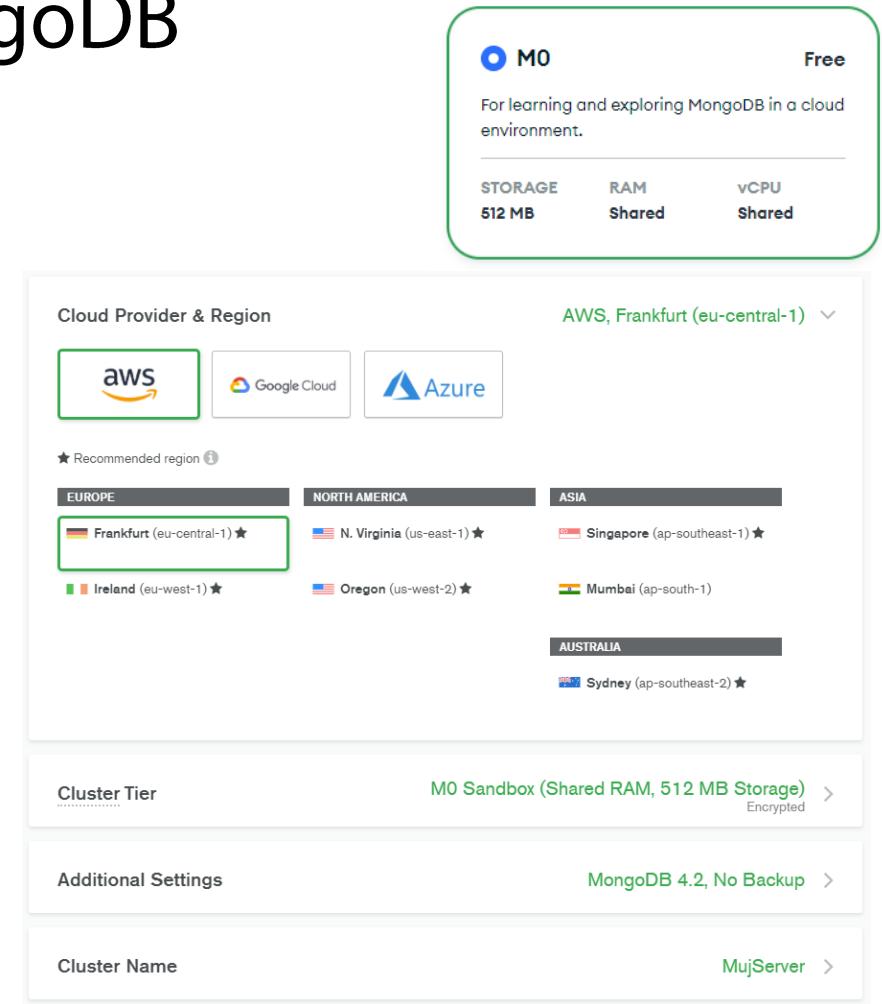


<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# MongoDB

- MongoDB Community Server
  - lokální server
- MongoDB Enterprise Server
  - komerční licence
- MongoDB [Atlas](#)
  - cloudová MongoDB služba
    - globální
  - dostupná na
    - Amazon Web Services
    - Microsoft Azure
    - Google Cloud Platform
  - základní (demo) plán zdarma
    - odpovídá reálnému nasazení



The screenshot shows the MongoDB Atlas cluster creation interface. At the top right, the 'M0' plan is selected with a green border, labeled 'Free'. Below it, the cluster details are shown: 'For learning and exploring MongoDB in a cloud environment.', 'STORAGE 512 MB', 'RAM Shared', and 'vCPU Shared'. The 'Cloud Provider & Region' section shows 'AWS, Frankfurt (eu-central-1)' selected. It lists regions across four continents: Europe (Frankfurt (eu-central-1)★, Ireland (eu-west-1)★), North America (N. Virginia (us-east-1)★, Oregon (us-west-2)★), Asia (Singapore (ap-southeast-1)★, Mumbai (ap-south-1)), and Australia (Sydney (ap-southeast-2)★). The 'Cluster Tier' is set to 'M0 Sandbox (Shared RAM, 512 MB Storage) Encrypted'. Under 'Additional Settings', the MongoDB version is '4.2, No Backup'. The 'Cluster Name' is 'MujServer'. At the bottom right, there is a small navigation icon.



# MongoDB ATLAS

- po založení serveru
  - nastavení přístupu (IP adresy)
  - vytvoření uživatelů
  - výběr přístupové metody
    - MongoDB shell
    - vlastní aplikace
      - Python, Node.js, Java, ...
    - MongoDB Compass

✓ [Setup connection security](#)   [Choose a connection method](#)   [Connect](#)

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.

[!\[\]\(3ea0b4bbc25be831ea423abcf9c7ab1a\_img.jpg\) Connect with the mongo shell](#)  
 Interact with your cluster using MongoDB's interactive Javascript interface

[!\[\]\(5a7834be8d7ea3819f4485bc9eadb9ae\_img.jpg\) Connect your application](#)  
 Connect your application to your cluster using MongoDB's native drivers

[!\[\]\(719091b56bfb10df498b48cfa84b753c\_img.jpg\) Connect using MongoDB Compass](#)  
 Explore, modify, and visualize your data with MongoDB's GUI

**Clusters**

[Create a New Cluster](#)

Find a cluster...

SANDBOX	Operations R: 0 W: 0 100.0/s	Logical Size 0.0 B 512.0 MB max
<b>MujServer</b> Version 4.2.10 <a href="#">CONNECT</a> <a href="#">METRICS</a> <a href="#">COLLECTIONS</a> ...	Last 6 Hours	Last 6 Hours
CLUSTER TIER M0 Sandbox (General)	Operations R: 0 W: 0 100.0/s	Logical Size 0.0 B 512.0 MB max
REGION AWS / Frankfurt (eu-central-1)	Connections 0 500 max	Enhance Your Experience For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!
TYPE Replica Set - 3 nodes	Last 6 Hours	<a href="#">Upgrade</a>
LINKED REALM APP None Linked		
ATLAS SEARCH <a href="#">Create Index</a>		

- připojení přes MongoDB shell

```

C:\Users\Rimmer>mongosh "mongodb+srv://mujserver.1vd1s.mongodb.net/myFirstDatabase" --apiVersion 1 --username lshark1
Enter password: *****
Current Mongosh Log ID: 62307c3477c62fdc9f78ff1f
Connecting to: mongodb+srv://mujserver.1vd1s.mongodb.net/myFirstDatabase?appName=mongosh+1.2.3
Using MongoDB: 5.0.6 (API Version 1)
Using Mongosh: 1.2.3

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
Atlas atlas-12nfzk-shard-0 [primary] myFirstDatabase>

```





# PŘÍSTUPOVÉ METODY

- MongoDB shell
  - interaktivní JS interface k MongoDB
  - dotazy, updaty
  - administrativní operace
  - kompletní obsluha
- MongoDB Compass
  - GUI nadstavba
- v praxi shell užitečnější
  - práce na dálku
  - terminálová obsluha rychlejší...



# PŘÍSTUPOVÉ METODY

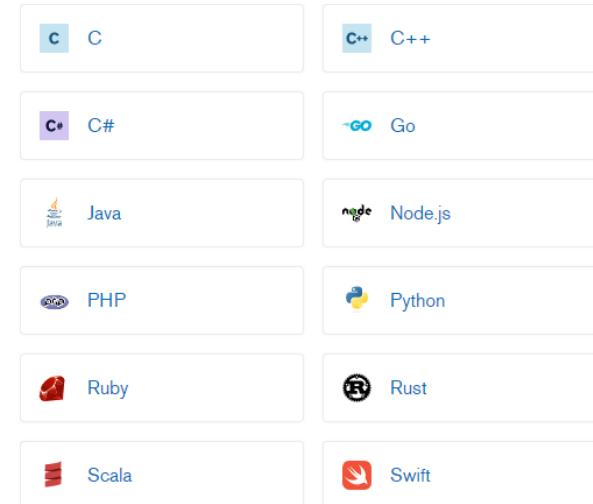
- MongoDB [Drivers](#)

- ovladače pro přístup z vlastní aplikace k serveru MongoDB
  - přemostují komunikaci
- podpora celé škály jazyků
  - podobná syntaxe příkazů
  - přizpůsobení jazykům
  - např. PyMongo pro Python

## Start Developing with MongoDB

Connect your application to your database with one of our official libraries.

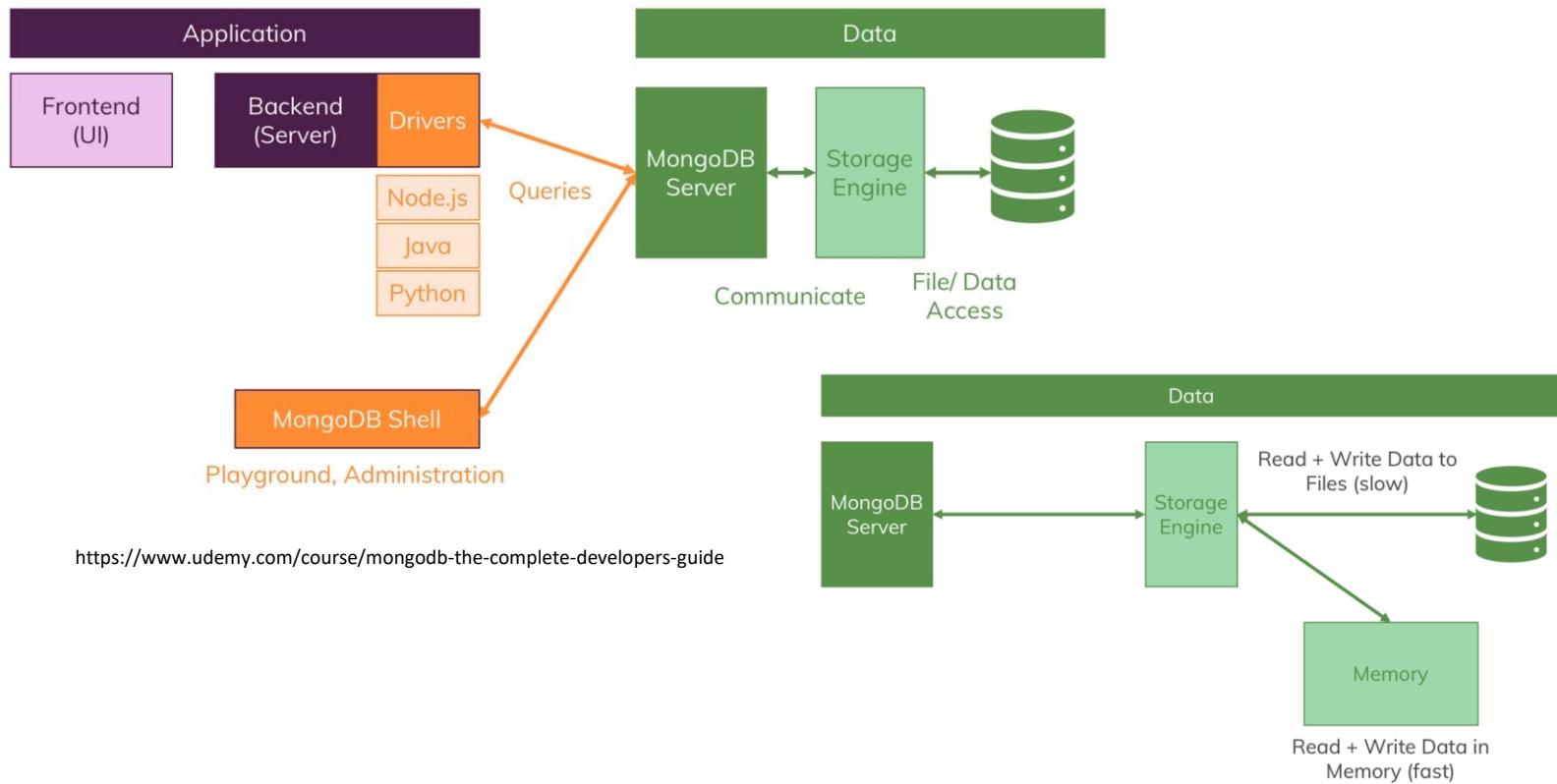
The following libraries are officially supported by MongoDB. They are actively maintained, support new MongoDB features, and receive bug fixes, performance enhancements, and security patches.



Don't see your desired language? Browse a list of [community supported libraries](#).



# MongoDB





# MongoDB PRAKTICKY

- zobrazení existujících databází

show dbs

- vytvoření databáze

use database-name

- aby se databáze objevila v seznamu, musí obsahovat kolekci

- přepnutí databáze

use database-name

- současná databáze

db

- odstranění databáze

db.dropDatabase()



# PRAKTICKY

- CRUD operace

## Create

```
insertOne(data, options)  
insertMany(data, options)
```

## Update

```
updateOne(filter, data, options)  
updateMany(filter, data, options)  
replaceOne(filter, data, options)
```

## Read

```
find(filter, options)  
findOne(filter, options)
```

## Delete

```
deleteOne(filter, options)  
deleteMany(filter, options)
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKY

- vytvoření kolekce „příspěvky“

```
db.createCollection('posts')
```

- seznam kolekcí

```
show collections
```

- vložení dokumentu

```
db.posts.insertOne({  
    title: 'Post One',  
    body: 'Body of post one',  
    category: 'News',  
    likes: 4,  
    tags: ['news', 'events'],  
    user: {  
        name: 'John Doe',  
        status: 'author'  
    },  
    date: Date()  
})
```

- pokud kolekce posts neexistuje, bude vytvořena
- vnořený dokument



# PRAKTICKÝ

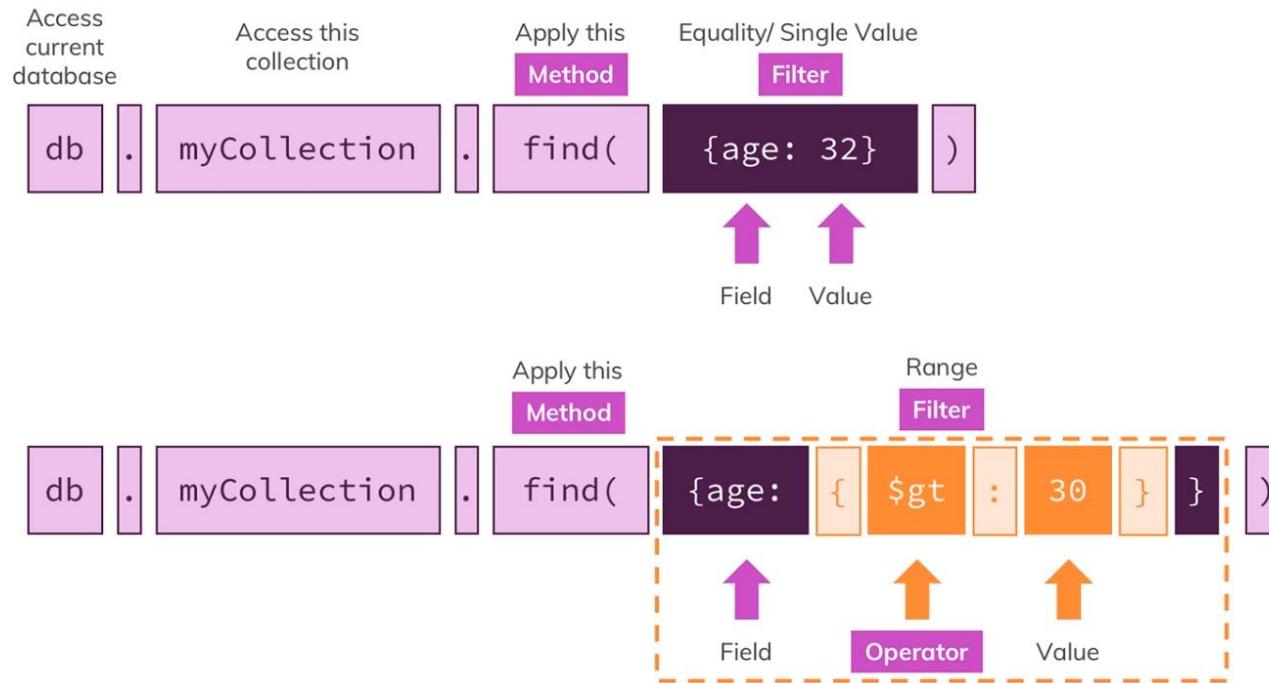
- vložení více dokumentů najednou

```
db.posts.insertMany([
  {
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: Date()
  },
  {
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: Date()
  },
  {
    title: 'Post Four',
    body: 'Body of post three',
    category: 'Entertainment',
    date: Date()
  }
])
```

- při chybě dojde k přerušení operace, ale není proveden rollback
  - např. existující index
  - dokumenty vložené před chybou zůstávají v databázi
  - dokumenty po chybě vloženy nejsou
- možnost parametru ordered na false
  - vloží veškeré dokumenty, které neskončí chybou



# PRAKTICKY



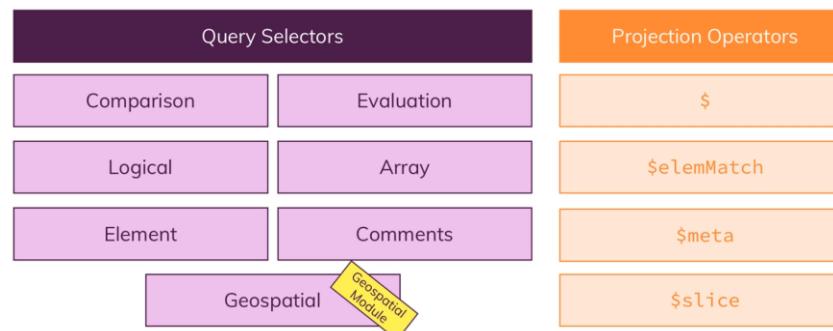
<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKÝ

- rozdělení operátorů

Type	Purpose	Changes Data?	Example
Query Operator	Locate Data	🚫	\$eq
Projection Operator	Modify data presentation	🚫	\$
Update Operator	Modify + add additional data	✓	\$inc



<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKÝ

- vypsání všech dokumentů  
`db.posts.find()`
- vypsání jednoho dokumentu  
`db.posts.findOne()`
- formátované vypsání
  - dnes již automatické, dříve `.pretty`
- vypsání všech news dokumentů  
`db.posts.find({ category: 'news' })`
- vypsání dokumentů s více jak 5 likes  
`db.posts.find({ likes: { $gt: 5 } })`
- vypsání dokumentů s méně jak 5 likes  
`db.posts.find({ likes: { $lt: 5 } })`

```
Atlas atlas-12nfzk-shard-0 [primary] dpb4> db.prispevky.find()  
[  
  {  
    _id: ObjectId("623082e3645157e3da631a12"),  
    nazev: 'Muj první článek'  
  },  
  {  
    _id: ObjectId("623082e3645157e3da631a13"),  
    nazev: 'Prispevek 2',  
    body: 'Toto je muj prispevek',  
    kategorie: 'zpravy',  
    tagy: [ 'zpravy', 'zesveta' ],  
    vlozil: { jmeno: 'Lukas', prijmeni: 'Mateju' },  
    datum: 'Tue Mar 15 2022 13:13:23 GMT+0100 (Středoevropský standardní čas)'  
  },  
  > db.posts.find().pretty()  
{  
  "_id" : ObjectId("5d2cdcf6f222de7f3b9d281a"),  
  "title" : "Post One",  
  "body" : "Body of post one",  
  "category" : "News",  
  "likes" : 4,  
  "tags" : [  
    "news",  
    "events"  
  ],  
  "user" : {  
    "name" : "John Doe",  
    "status" : "author"  
  },  
  "date" : "Mon Jul 15 2019 16:07:18 GMT-0400 (EDT)"  
}  
{  
  "_id" : ObjectId("5d2cdd2af222de7f3b9d281b"),  
  "title" : "Post Two",  
  "body" : "Body of post two",  
  "category" : "Technology",  
  "date" : "Mon Jul 15 2019 16:08:10 GMT-0400 (EDT)"  
}
```



# PRAKTICKÝ

- vypsání všech dokumentů, které mají 15, 30 nebo 45 likes  
`db.posts.find({likes: {$in: [15, 30, 45]} })`
- vypsání všech dokumentů, které nemají 15, 30 nebo 45 likes  
`db.posts.find({likes: {$nin: [15, 30, 45]} })`
- dokumenty, které mají likes nižší než 15 nebo vyšší než 45  
`db.posts.find({$or: [{likes: {$lt: 15}}, {likes: {$gt: 45}}] })`
- dokumenty, které mají likes vyšší než 15 a jsou v kategorii news
  - `db.posts.find({$and: [{likes: {$lt: 15}}, {category: 'News'}]} )`  
`db.posts.find({likes: {$gt: 15}, category: 'News' })`
  - \$and nutné použít v případě podmínky týkající se stejných polí

# PRAKTICKY

- vypsání všech dokumentů, které obsahují tag news

```
db.posts.find({tags: 'news'})
```

- vypsání všech dokumentů, které obsahují jen tag news

```
db.posts.find({tags: ['news']})
```

- vypsání všech dokumentů, které obsahují jen tagy news a tech

- v libovolném pořadí

```
db.posts.find({tags: {$all: ['news', 'tech']}})
```

- vypsání všech dokumentů se třemi tagy

```
db.posts.find({tags: {$size: 3}})
```



# PRAKTICKÝ

- vypsání všech dokumentů, které jsou od John Doe
  - vnořené dokumenty
  - tečková notace

```
db.posts.find({ "user.name": 'John Doe' })
```

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({ "user.name": 'John Doe' }).pretty()
{
    "_id" : ObjectId("5f8ef175c43ece2db0230f85"),
    "title" : "Post One",
    "body" : "Body of post one",
    "category" : "News",
    "likes" : 4,
    "tags" : [
        "news",
        "events"
    ],
    "user" : {
        "name" : "John Doe",
        "status" : "author"
    },
    "date" : "Date()"
}
{
    "_id" : ObjectId("5f8fe759c43ece2db0230f8a"),
    "title" : "Post Five",
    "likes" : 81,
    "user" : {
        "name" : "John Doe",
        "status" : "author"
    },
    "date" : "Date()"
}
```



# PRAKTICKÝ

- vypsání všech dokumentů, které obsahují počet likes

```
db.posts.find({likes: {$exists: true, $ne: null}})
```

- vypsání všech dokumentů, kde likes je číslo

```
db.posts.find({likes: {$type: 'number'}})
```

- vypsání všech dokumentů, kde počet sdílení je vyšší než likes

- operátor expr
- \$ pro označení polí dokumentu

```
db.posts.find({$expr: {$gt: ['$shares', '$likes']}})
```

# PRAKTICKÝ

- vyhledání prvku v poli
  - např. všechny dokumenty, kde komentoval uživatel X
  - \$elemMatch

```
db.posts.find({  
    comments: {  
        $elemMatch: {  
            user: 'Mary Williams'  
        }  
    }  
})
```



# PRAKTICKÝ

- vypsání názvů článků u všech dokumentů (projekce)

```
db.posts.find({}, {title:1})
```

```
db.posts.find({}, {_id:0, title:1})
```

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({}, {title:1}).pretty()
{
  "_id" : ObjectId("5f8ef175c43ecc2db0230f85"),
  "title" : "Post One"
}
{
  "_id" : ObjectId("5f8ef175c43ecc2db0230f86"),
  "title" : "Post Two"
}
{
  "_id" : ObjectId("5f8ef175c43ecc2db0230f87"),
  "title" : "Post Three"
}
{
  "_id" : ObjectId("5f8ef175c43ecc2db0230f88"),
  "title" : "Post Four"
}
{
  "_id" : ObjectId("5f8ef759c43ecc2db0230f8a"),
  "title" : "Post Five"
}
{
  "_id" : ObjectId("5f9026800cbc092824d7e420"),
  "title" : "Post o Ničem"
}
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({}, {_id:0, title:1}).pretty()
{
  "title" : "Post One"
}
{
  "title" : "Post Two"
}
{
  "title" : "Post Three"
}
{
  "title" : "Post Four"
}
{
  "title" : "Post Five"
}
{
  "title" : "Post o Ničem"
}
```

- vypsání názvů článků od John Doe

```
db.posts.find({"user.name": 'John Doe'}, {_id:0, title:1})
```

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({"user.name": 'John Doe'}, {_id:0, title:1, user:1}).pretty()
{
  "title" : "Post One",
  "user" : {
    "name" : "John Doe",
    "status" : "author"
  }
}
{
  "title" : "Post Five",
  "user" : {
    "name" : "John Doe",
    "status" : "author"
  }
}
```



# PRAKTICKÝ

- vypsání názvů článků končících na písmeno m

```
db.posts.find({title: /m$/})
```

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({title: /m$/}).pretty()
{
    "_id" : ObjectId("5f9026800cbc092824d7e420"),
    "source" : "id2",
    "title" : "Post o Ničem",
    "user" : {
        "name" : "Jane Doe",
        "gender" : "female"
    }
}
{
    "_id" : ObjectId("5f917a74a07d4d360f716ee9"),
    "title" : "Nevim"
}
{
    "_id" : ObjectId("5f917a7aa07d4d360f716eea"),
    "title" : "Neznam"
}
```

## ➤ použití regulárních výrazů

- \$ značí konec řetězce
- jak najít názvy začínající na N?
- a co třeba názvy obsahující čísla?

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({title: /^N/}).pretty()
{
    "_id" : ObjectId("5f917a74a07d4d360f716ee9"),
    "title" : "Nevim"
}
{
    "_id" : ObjectId("5f917a7aa07d4d360f716eea"),
    "title" : "Neznam"
}
{
    "_id" : ObjectId("5f917a7fa07d4d360f716eeb"),
    "title" : "Nechci"
}
```



# PRAKTICKÝ

- seřazení dokumentů podle názvu

- vzestupně

```
db.posts.find().sort({ title: 1 })
```

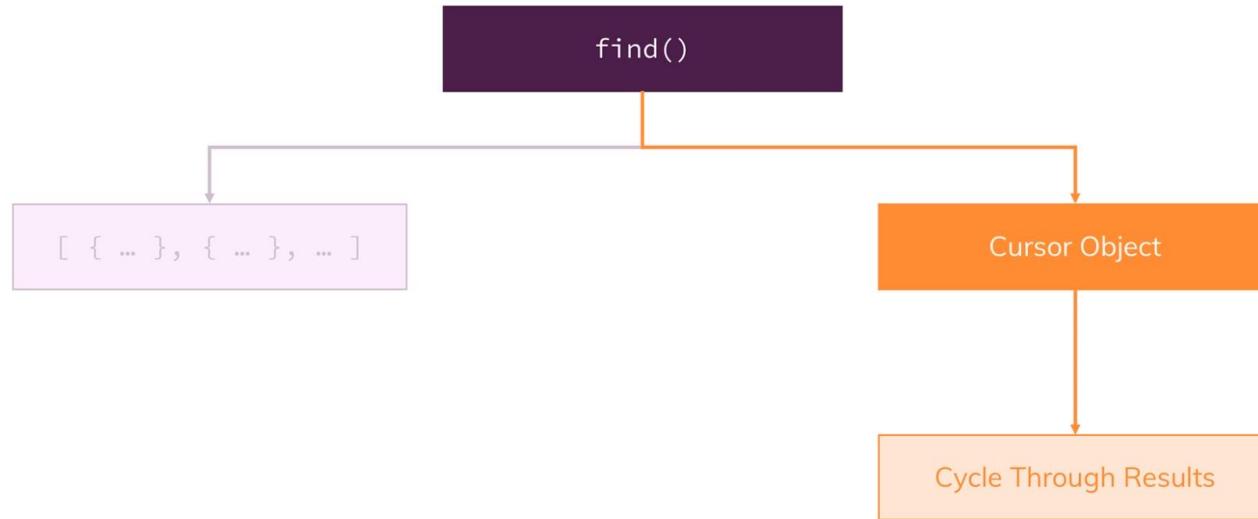
- sestupně

```
db.posts.find().sort({ title: -1 })
```

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find().sort({ title: 1 }).pretty()
{
    "_id" : ObjectId("5f8ef175c43ece2db0230f88"),
    "title" : "Post Four",
    "body" : "Body of post three",
    "category" : "Entertainment",
    "date" : "Date()"
}
{
    "_id" : ObjectId("5f8ef175c43ece2db0230f85"),
    "title" : "Post One",
    "body" : "Body of post one",
    "category" : "News",
    "likes" : 4,
    "tags" : [
        "news",
        "events"
    ],
    "user" : {
        "name" : "John Doe",
        "status" : "author"
    },
    "date" : "Date()"
}
{
    "_id" : ObjectId("5f8ef175c43ece2db0230f87"),
    "title" : "Post Three",
```



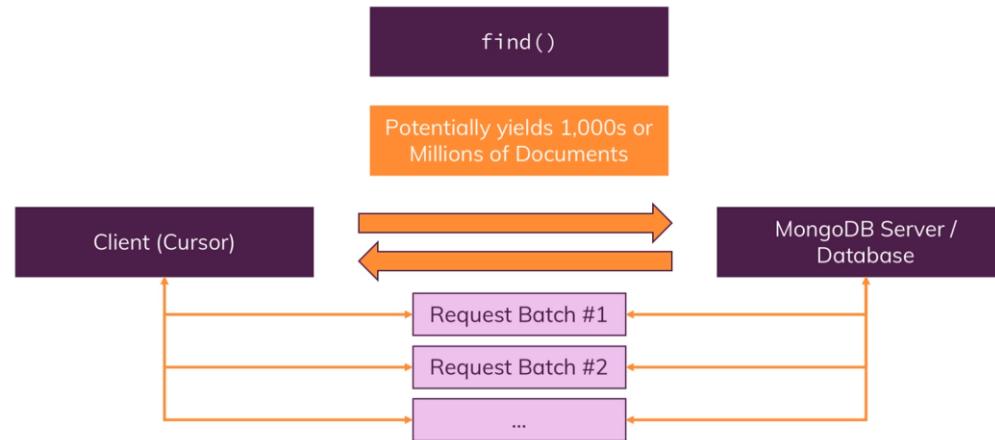
# PRAKTICKY



<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

# PRAKTICKÝ

- find v praxi nevrací všechny dokumenty, ale kurzor
  - pointer na data



<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

- **toArray()**
  - vrátí všechny dokumenty v poli
- **forEach()**
  - umožňuje postupnou aplikaci na všechny dokumenty



# PRAKTICKÝ

- řetězení na kurzoru

```
db.posts.find().sort({title: 1}).pretty()
```

- počet dokumentů

```
db.posts.find().count()
```

- v kategorii news

```
db.posts.find({ category: 'news' }).count()
```

- limit

```
db.posts.find().limit(2)
```

- vrátí první dva dokumenty

- skip

```
db.posts.find().skip(1)
```

- přeskočí první dokument

```
> db.posts.find().sort({ title: -1 }).limit(2)
{ "_id" : ObjectId("5d2cdd2af222de7f3b9d281b"), "title" : "Post Two", "body" :
"Body of post two", "category" : "Technology", "date" : "Mon Jul 15 2019 16:08:10
GMT-0400 (EDT)" }
{ "_id" : ObjectId("5d2cdd2af222de7f3b9d281c"), "title" : "Post Three", "body" :
"Body of post three", "category" : "News", "date" : "Mon Jul 15 2019 16:08:10
GMT-0400 (EDT)" }
```



# PRAKTICKÝ

- **foreach**

```
db.posts.find().forEach(function(doc) {  
    print("Blog Post: " + doc.title)  
})
```

- např. pro formátované výpisy

```
> db.posts.find().forEach(function(doc) { print('Blog Post: ' + doc.title) })  
Blog Post: Post One  
Blog Post: Post Two  
Blog Post: Post Three  
Blog Post: Post Four
```

# PRAKTICKÝ

- **updateOne()** vs. **replaceOne()** / **updateMany()** / **update()**
  - **updateOne()**
    - slouží k aktualizaci jednoho dokumentu
    - parametr \$set udává, která pole budou změněna
    - neslouží k přepsání dokumentu
  - **replaceOne()**
    - slouží k přepsání jednoho dokumentu
  - **updateMany()**
    - slouží k aktualizaci více dokumentu podle zadaného filtru
    - parametr \$set udává, která pole budou změněna
    - neslouží k přepsání dokumentu
  - **update**
    - dnes již deprecated
    - sloužilo pro aktualizaci i nahrazení



# PRAKTICKÝ

- **update jednoho dokumentu**

```
db.posts.updateOne({ title: 'Post Two' },
{ $set: { title: 'Post Two New' } },
{ upsert: true }
```

- ideální použít `_id` místo názvu
- `upsert true`
- `update` nebo `insert`
- v případě nenalezení dokumentu dojde k jeho vytvoření

```
Atlas atlas-12nfzk-shard-0 [primary] dpb4> db.prispevky.updateOne({ nazev: 'Zpravy' }, { $set: { nazev: 'Moje Zpravy' } }, { upsert: true })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

- **update více dokumentů**

```
db.posts.updateMany({ tag: 'News' },
{ $set: { tag: 'World' } },
{ upsert: true }
```



# PRAKTICKÝ

- nahrazení jednoho dokumentu

```
db.posts.replaceOne({ title: 'Post Two' },
{ title: 'Post Two New', likes: 60 },
{ upsert: true }
```

- ideální použít `_id` místo názvu
- veškerá ostatní pole původního dokumentu jsou ztracena

```
Atlas atlas-12nfzk-shard-0 [primary] dpb4> db.prispevky.replaceOne({ nazev: 'Moje Zpravy' },
{ nazev: 'Zpravy' }, { upsert: true })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



# PRAKTICKÝ

- **inkrementace**

```
db.posts.updateMany({ title: 'Post Two' },
{
  $inc: {
    likes: 5
  }
})
• inkrementace o 5
```

- **minimum / maximum mezi uloženou a novou hodnotou**

```
db.posts.updateMany({ title: 'Post Two' },
{
  $max: {
    likes: 10
  }
})
```



# PRAKTICKÝ

- odstranění pole

```
db.posts.updateMany({ title: 'Post Two' },
{
  $unset: {
    category: ''
  }
})
```

- přejmenování pole

```
db.posts.updateMany({ title: 'Post Two' },
{
  $rename: {
    likes: 'views'
  }
})
```

# PRAKTICKÝ

- vnořené dokumenty

```
db.posts.updateOne({ title: 'Post One' },
{
  $set: {
    comments: [
      {
        body: 'Comment One',
        user: 'Mary Williams',
        date: Date()
      },
      {
        body: 'Comment Two',
        user: 'Harry White',
        date: Date()
      }
    ]
  }
})
```



# PRAKTICKÝ

- úprava vyfiltrovaných prvků pole

```
> db.users.updateMany({hobbies: {$elemMatch: {title: "Sports", frequency: {$gte: 3}}}}, {$set: {"hobbies.$.highFrequency": true}})
```

- první část dotazu vybírá uživatele, kteří mají koníček sport s frekvencí více než 2
  - `$elemMatch`
- druhá část k těmto dokumentům přidává pole `highFrequency` `true`
  - operátor `$` slouží pro přístup k prvku pole odpovídajícího vyhledávání

```
        "_id" : ObjectId("5b9f652dc62d13aa2e81c29d"),
        "name" : "Max",
        "hobbies" : [
            {
                "title" : "Sports",
                "frequency" : 3
            },
            {
                "title" : "Cooking",
                "frequency" : 6
            }
        ],
        "isSporty" : true
    }

    "_id" : ObjectId("5b9f65d0c62d13aa2e81c29f"),
    "name" : "Anna",
    "hobbies" : [
        {
            "title" : "Sports",
            "frequency" : 2
        },
        {
            "title" : "Yoga",
            "frequency" : 3
        }
    ],
    "isSporty" : false
}
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKÝ

- úprava všech prvků pole

```
db.users.updateMany({totalAge: {$gt: 30}}, {$inc: {"hobbies.$[].frequency": -1}})
```

- první část dotazu vybírá uživatele starší 30 let
- druhá část snižuje frekvenci u všech koníčků o 1
  - operátor \$[]
  - \$ vrací první odpovídající prvek
    - snížení by bylo jen u prvního koníčku
  - \$[] všechny

```
        "_id" : ObjectId("5b9f652dc62d13aa2e81c29d"),
        "name" : "Max",
        "hobbies" : [
            {
                "title" : "Sports",
                "frequency" : 3
            },
            {
                "title" : "Cooking",
                "frequency" : 6
            }
        ],
        "isSporty" : true
    },
    "_id" : ObjectId("5b9f65d0c62d13aa2e81c29f"),
    "name" : "Anna",
    "hobbies" : [
        {
            "title" : "Sports",
            "frequency" : 2
        },
        {
            "title" : "Yoga",
            "frequency" : 3
        }
    ],
    "isSporty" : false
}
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKÝ

- přidání prvku do pole
  - \$push / \$addToSet (jen unikátní hodnoty)

```
> db.users.updateOne({name: "Maria"}, {$push: {hobbies: {title: "Sports", frequency: 2}}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

- přidání více prvků do pole
  - \$push a \$each

```
> db.users.updateOne({name: "Maria"}, {$push: {hobbies: {$each: [{"title: "Good Wine", frequency: 1}, {title: "Hiking", frequency: 2}]}}})
```

- odstranění prvku z pole
  - \$pull

```
> db.users.updateOne({name: "Maria"}, {$pull: {hobbies: {title: "Hiking"}}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKÝ

- smazání dokumentu

```
db.posts.deleteOne({ title: 'Post Four' })
```

```
db.posts.deleteMany({ title: 'Unknown' })
```

➤ většinou samozřejmě přes \_id

```
> db.users.deleteMany({age: {$gt: 30}, isSporty: true})
{ "acknowledged" : true, "deletedCount" : 0 }
> db.users.deleteMany({totalAge: {$gt: 30}, isSporty: true})
{ "acknowledged" : true, "deletedCount" : 0 }
> db.users.deleteMany({totalAge: {$exists: false}, isSporty: true})
{ "acknowledged" : true, "deletedCount" : 2 }
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# PRAKTICKY

- vytvoření indexu

```
db.posts.createIndex({ title: 1 })
```

```
> db.contacts.createIndex({"dob.age": 1})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1
}
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>

- vyhledání funguje automaticky
  - MongoDB využije index, pokud je to vhodné

# PRAKTICKÝ

- vytvoření text indexu

```
db.posts.createIndex({ title: 'text' })
```

- textové hledání

- nad indexem

```
db.posts.find({  
    $text: {  
        $search: "\"Post o\""  
    }  
})
```

➤ vrátí **post one**

- dokumentace

- <https://docs.mongodb.com/>

# PRAKTICKÝ

- import dat
  - pomocí mongoimport
  - -d – cílová databáze
  - -c – cílová kolekce
  - --jsonArray – specifikuje pole dokumentů
  - --drop – nahradí původní kolekce
    - bez --drop jsou data přidána do původní kolekce

```
Maximilians-MBP:Datasets mschwarzmueller$ mongoimport tv-shows.json -d movieData -c movies --jsonArray
--drop
2018-09-17T09:15:01.282+0200      connected to: localhost
2018-09-17T09:15:01.282+0200      dropping: movieData.movies
2018-09-17T09:15:01.339+0200      imported 240 documents
Maximilians-MBP:Datasets mschwarzmueller$
```

<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# MongoDB COMPASS

- lokální GUI nadstavba
- propojení s MongoDB Atlas

Connect to MujServer

✓ Setup connection security   ✓ Choose a connection method   Connect

I do not have MongoDB Compass   I have MongoDB Compass

1 Select your operating system and download MongoDB Compass

Windows 64-bit (7+)   Download Compass (1.22.1) or Copy download URL

2 Copy the connection string, then open MongoDB Compass.

```
mongodb+srv://1shark1:<password>@mujserver.1vd1s.mongodb.net/test
```

Copy

You will be prompted for the password for the **1shark1** user's (Database User) username.  
When entering your password, make sure that any special characters are [URL encoded](#).



New Connection   FAVORITE

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

e.g. mongodb+srv://username:password@cluster

CONNECT

New to Compass and don't have a cluster? If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#). CREATE FREE CLUSTER

How do I find my connection string in Atlas? If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

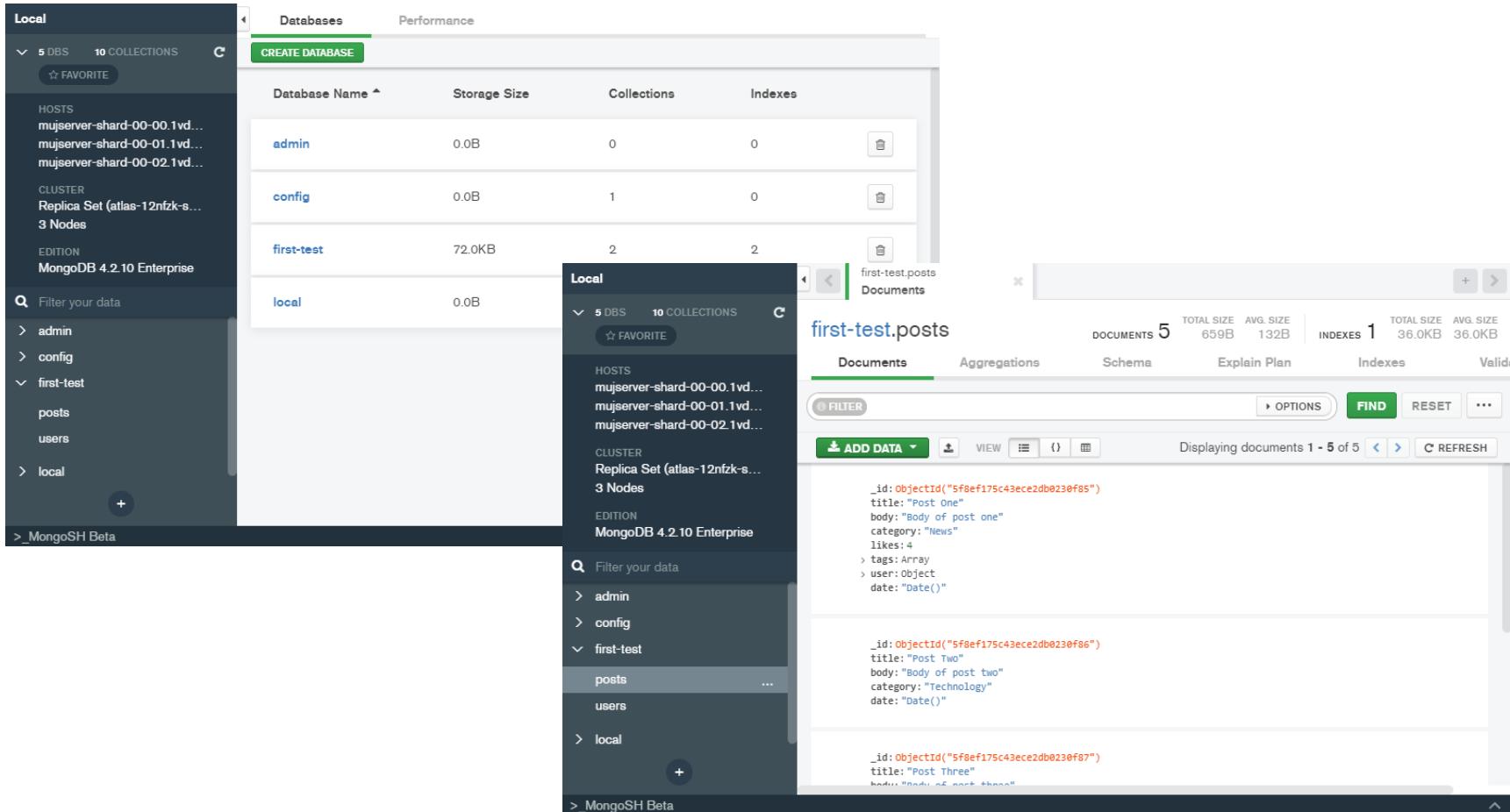
See example

How do I format my connection string? See example

Having trouble connecting? [View our troubleshooting documentation](#)



# MongoDB COMPASS



The screenshot shows the MongoDB Compass interface. On the left, a sidebar displays the 'Local' database with 5 DBs and 10 collections. The 'first-test' database is selected. The main area shows the 'Databases' tab with a list of databases: admin, config, first-test, and local. The 'first-test' database is expanded, showing its 10 collections: posts (selected), users, and local. The 'posts' collection has 5 documents. The document details pane shows three documents:

```
_id: ObjectId("5f8ef175c43ece2db0230f85")
title: "Post One"
body: "Body of post one"
category: "News"
likes: 4
tags: Array
user: Object
date: "Date()"

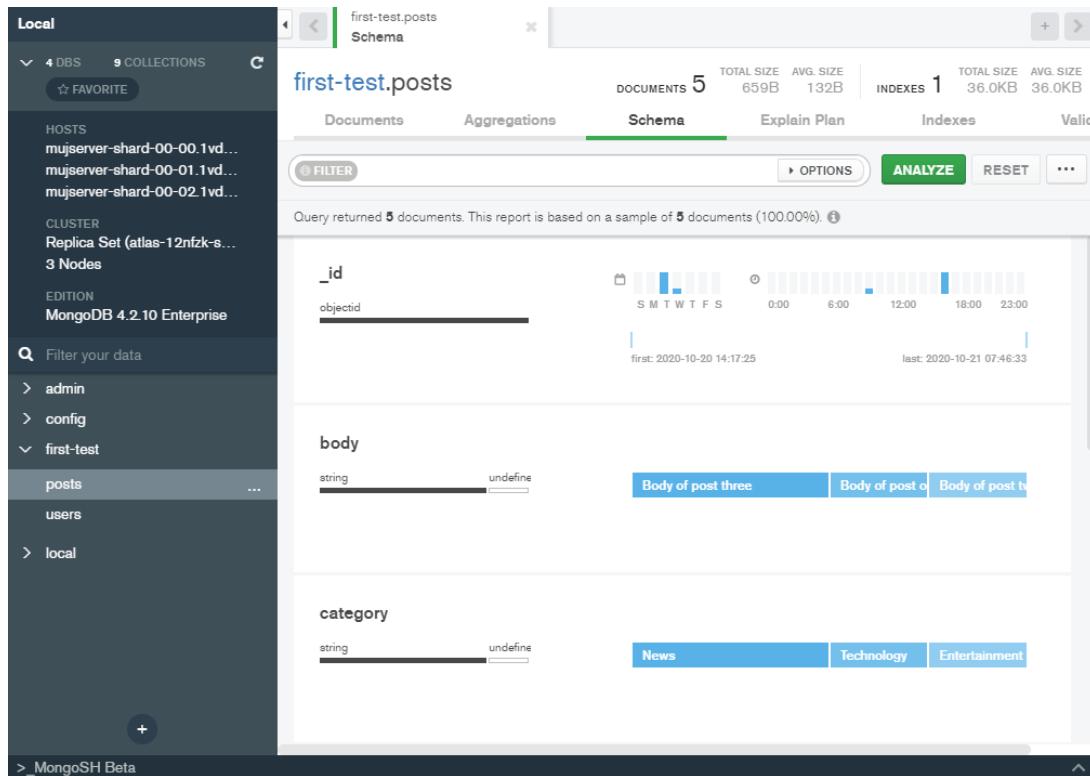
_id: ObjectId("5f8ef175c43ece2db0230f86")
title: "Post Two"
body: "Body of post two"
category: "Technology"
date: "Date()"

_id: ObjectId("5f8ef175c43ece2db0230f87")
title: "Post Three"
body: "Body of post three"
category: "News"
date: "Date()"
```



# MongoDB COMPASS

- vhodné také pro vizualizaci dat

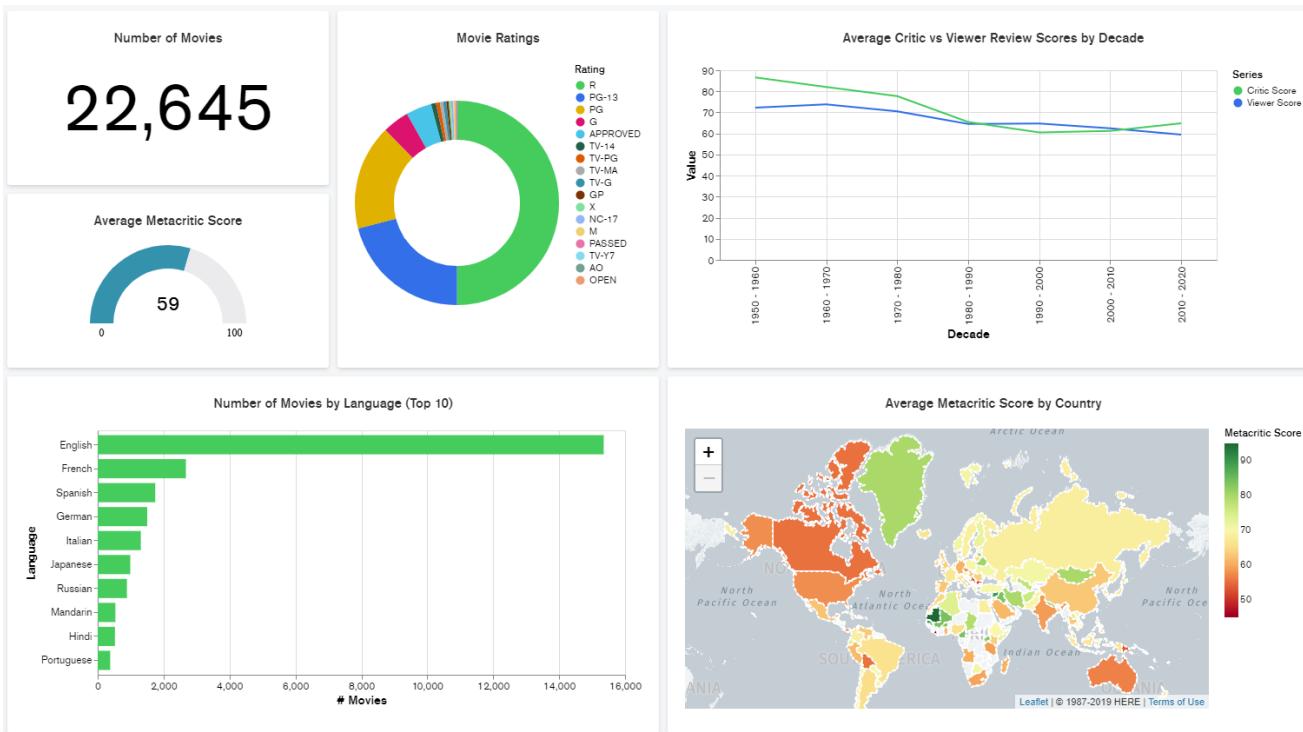


The screenshot shows the MongoDB Compass interface for the 'first-test.posts' collection. The left sidebar lists 'HOSTS' (mujserver-shard-00-00.vd..., mujserver-shard-00-01.vd..., mujserver-shard-00-02.vd...), a 'CLUSTER' (Replica Set (atlas-12nfzk-s...), 3 Nodes), and the 'EDITION' (MongoDB 4.2.10 Enterprise). The main panel shows the 'Schema' tab for the 'first-test.posts' collection, which contains 5 documents, totaling 659B with an average size of 132B, and 1 index, totaling 36.0KB with an average size of 36.0KB. Below the schema, there is a histogram for the '\_id' field, showing a distribution from 0:00 to 23:00. The 'body' field is shown as a string type with three sample values: 'Body of post three', 'Body of post o', and 'Body of post t'. The 'category' field is also shown as a string type with three sample values: 'News', 'Technology', and 'Entertainment'.



# MongoDB CHARTS

- ještě vhodnější pro vizualizaci...



# A PŘÍŠTĚ?

- MongoDB...





Děkuji za pozornost.  
Otázky?

