



# MongoDB

*Lukáš Matějů*

26.3.2024 | DPB

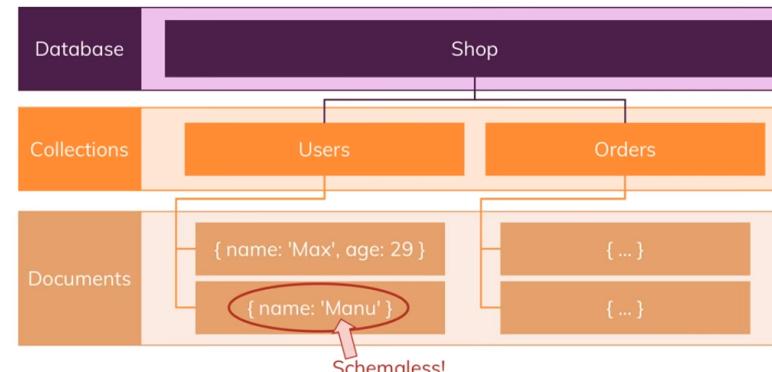


# ČÁST I.: OPAKOVÁNÍ



# OPAKOVÁNÍ

- MongoDB
  - humongous data
    - ukládání spousty a spousty dat
  - dokumentová NoSQL databáze
    - open source
    - multiplatformní
  - v aktivním vývoji
  - v databázi data ukládána v kolekcích dokumentů
    - kolekce
      - seznam dokumentů
    - dokument
      - obsahuje data
      - reprezentován pomocí vnořených objektů / map
      - ve formátu BSON



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

# OPAKOVÁNÍ

- dokument
  - základní stavební prvek
  - maximum 16 MB
  - volné schéma
  - pole \_id
    - primární klíč
    - automaticky přidán
    - ObjectId
      - 12 bytů
      - unikátní, rostoucí
  - pole comments
    - obsahuje pole dalších vnořených dokumentů
      - embedded dokumenty
      - až 100 úrovní
    - data patřící k sobě uložena v jednom dokumentu

Looks like JSON. Example:

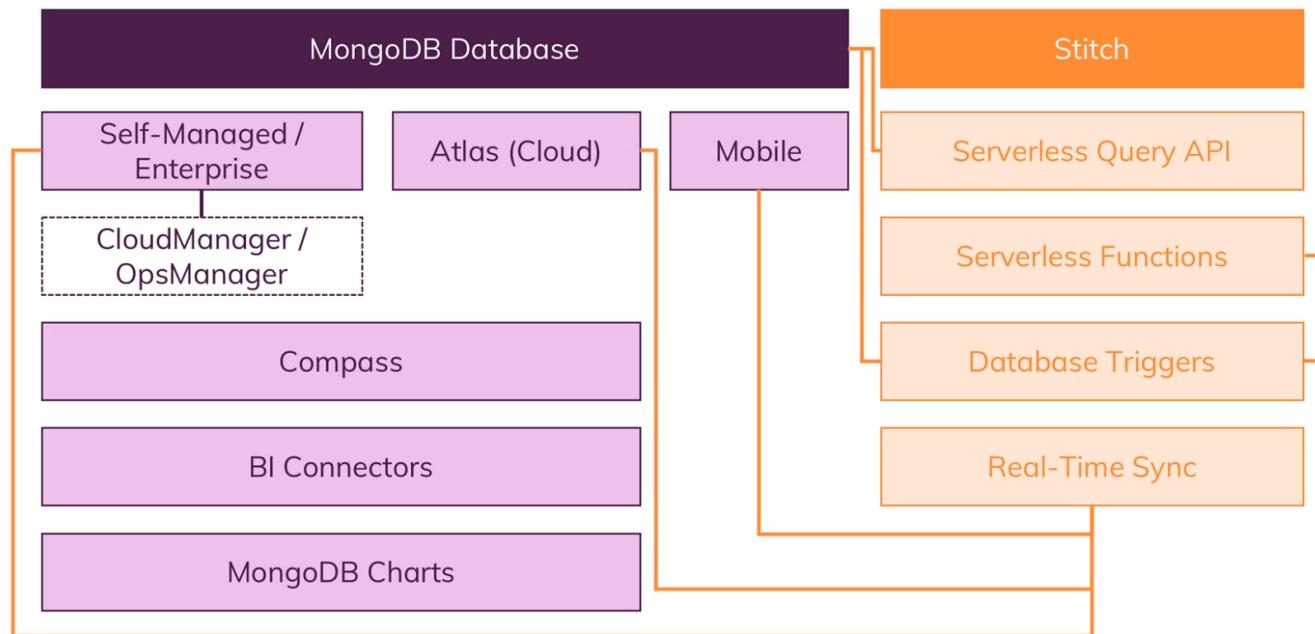
```
{  
    "_id" : ObjectId("7b33e366ae32223aee34fd3"),  
    "title" : "A blog post about MongoDB",  
    "content" : "This is a blog post about MongoDB",  
    "comments": [  
        {  
            "name" : "Frank",  
            "email" : fkane@sundog-soft.com,  
            "content" : "This is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

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



# OPAKOVÁNÍ

- MongoDB ekosystém



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



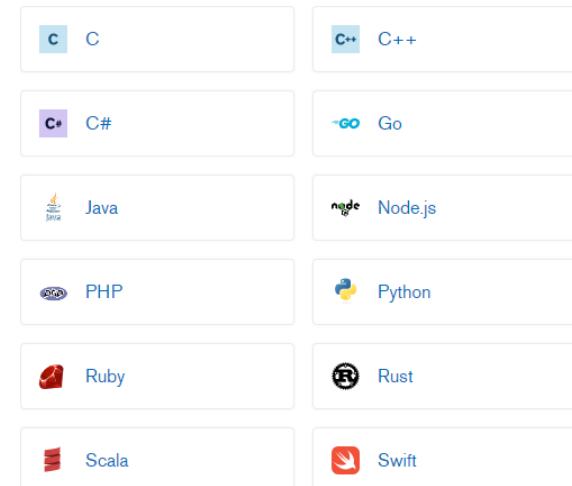
# OPAKOVÁNÍ

- přístupové metody
  - MongoDB shell
    - interaktivní JS interface k MongoDB
    - dotazy, updaty
    - administrativní operace
    - kompletní obsluha
  - MongoDB Compass
    - GUI nadstavba
  - 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](#).

# OPAKOVÁNÍ

- MongoDB prakticky
  - CRUD operace

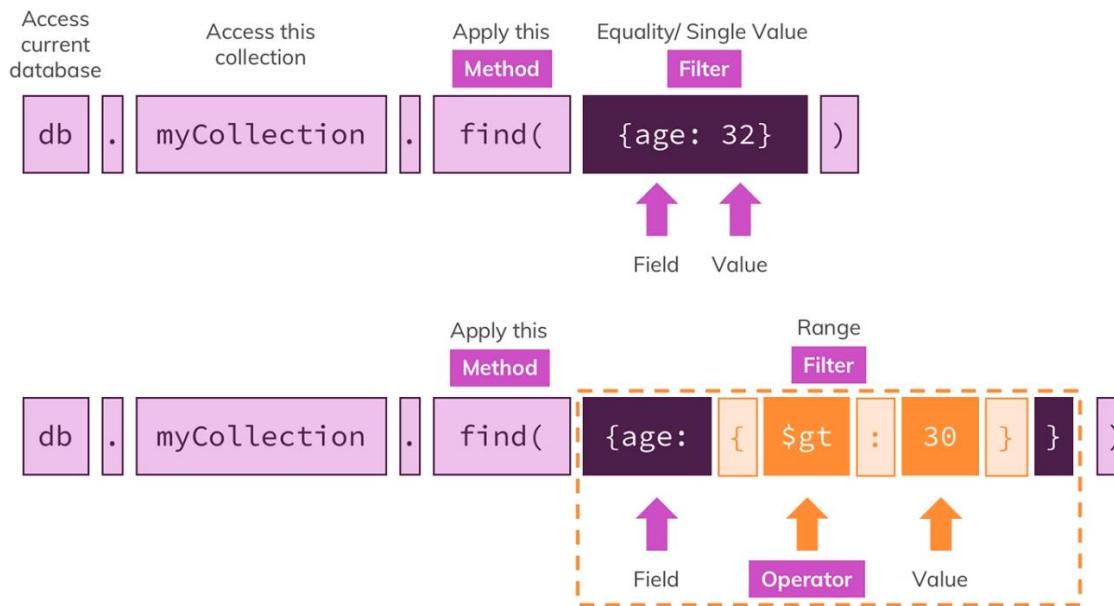
Create	Update
insertOne(data, options)	updateOne(filter, data, options)
insertMany(data, options)	updateMany(filter, data, options)
Read	Delete
find(filter, options)	deleteOne(filter, options)
findOne(filter, options)	deleteMany(filter, options)

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



# OPAKOVÁNÍ

- MongoDB prakticky

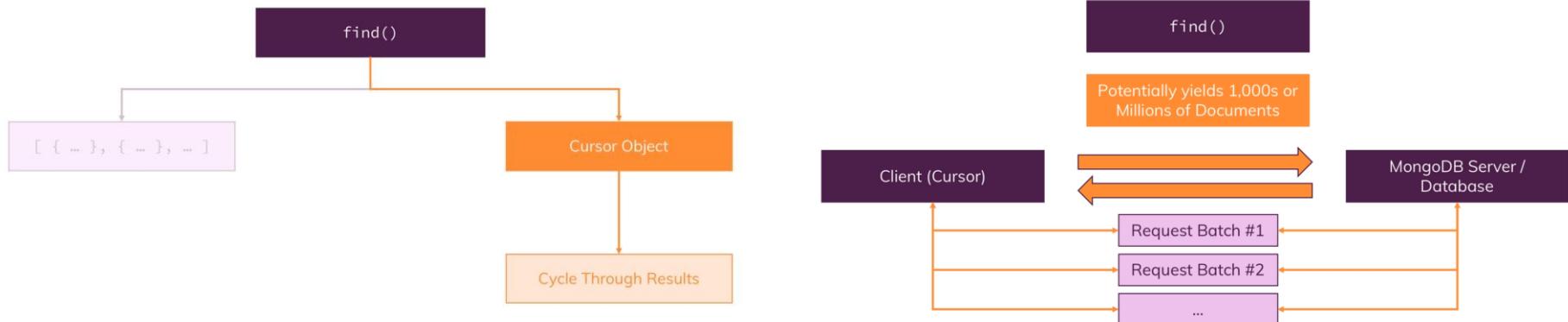


<https://www.udemy.com/course/mongodb-the-complete-developers-guide>



# OPAKOVÁNÍ

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



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



# ČÁST II.: MongoDB





# FUNKCE / VLASTNOSTI

- volné schéma
- ad-hoc dotazy
- indexování
- transakce
- replikace
- sharding
- agregace
- souborový systém
- zabezpečení



# VOLNÉ SCHÉMA

```
_id: ObjectId("5f8ef175c43ece2db0230f85")
title: "Post One"
body: "Body of post one"
category: "News"
likes: 4
tags: Array
  0: "news"
  1: "events"
user: Object
  name: "John Doe"
  status: "author"
  date: "Date()"

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

_id: ObjectId("5f8ef175c43ece2db0230f87")
source: "id1"
title: "Post Three"
views: "80"

_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"
```



# VOLNÉ SCHÉMA

- žádné schéma není nutné
- každé pole v každém dokumentu může být jiné...
  - i datové typy se mohou lišit
- není nutný jediný klíč jako v jiných databázích
  - indexy mohou být vytvořeny na různá pole
- flexibilita
  - ad-hoc dotazy
  - i tak je dobré si návrh dobře promyslet...
    - které indexy budou potřeba pro dotazy, co se budou používat?
    - vazby mezi daty?
  - join stále neefektivní
    - NoSQL databáze...

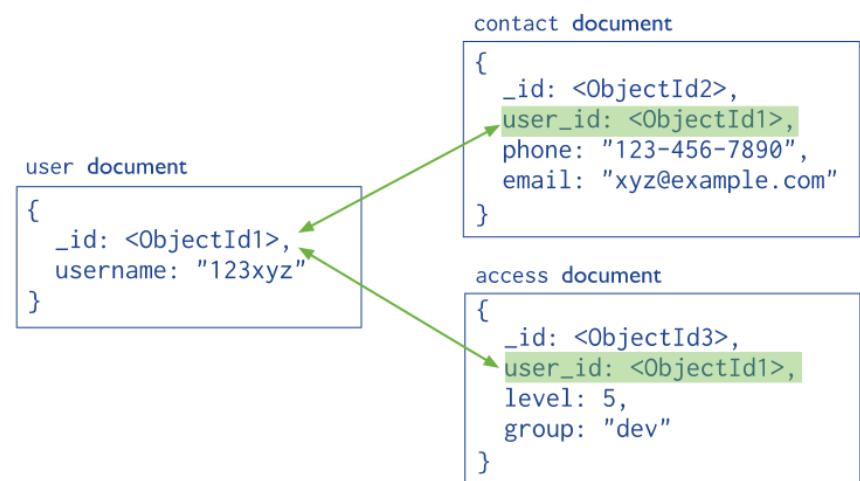
# VOLNÉ SCHÉMA

- a co s vazbami mezi daty?
  - vnořené (embedded) dokumenty
  - reference
- závisí na konkrétních případech a použití

```
{  
  _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/>



# VOLNÉ SCHÉMA

- **embedded dokumenty**
  - nenormalizovaný model
  - veškerá relevantní data v jednom dokumentu
  - kdy použít?
    - vazby 1:1
    - vazby 1:N
  - **výhody**
    - lepší výkon pro operace čtení
    - přístup k relevantním datům v rámci jedné db operace
    - update dat v rámci jedné atomické operace zápisu
  - **přístup přes tečkovou notaci**

```
db.users.find({"addresses.zip": '12345'})
```

```
{  
  "_id": "joe",  
  "name": "Joe Bookreader",  
  "address": {  
    "street": "123 Fake Street",  
    "city": "Faketon",  
    "state": "MA",  
    "zip": "12345"  
  }  
}
```

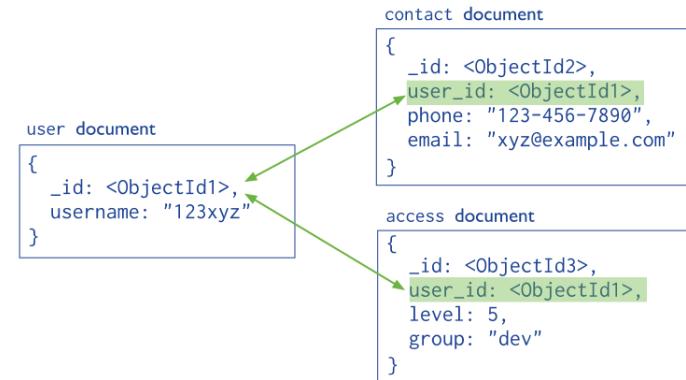
```
{  
  "_id": "joe",  
  "name": "Joe Bookreader",  
  "addresses": [  
    {  
      "street": "123 Fake Street",  
      "city": "Faketon",  
      "state": "MA",  
      "zip": "12345"  
    },  
    {  
      "street": "1 Some Other Street",  
      "city": "Boston",  
      "state": "MA",  
      "zip": "12345"  
    }  
  ]  
}
```

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



# VOLNÉ SCHÉMA

- reference
  - potřeba normalizovaného datového modelu
  - relevantní data v různých dokumentech (kolekcích)
  - kdy použít?
    - vnořené dokumenty tvoří velké množství duplikátů a zároveň neurychlují přístup k datům
    - vazby M:N
    - velké hierarchické datasety
- join kolekcí pomocí
  - \$lookup
    - left outer join
    - agregace
  - \$graphLookup
    - rekurzivní vyhledávání



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



# VOLNÉ SCHÉMA

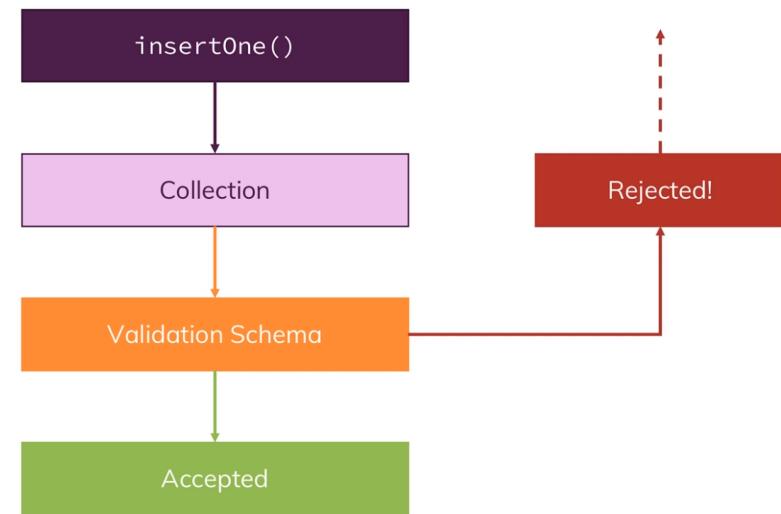
```
{  
    title: "MongoDB: The Definitive Guide",  
    author: [ "Kristina Chodorow", "Mike Dirolf" ],  
    published_date: ISODate("2010-09-24"),  
    pages: 216,  
    language: "English",  
    publisher: {  
        name: "O'Reilly Media",  
        founded: 1980,  
        location: "CA"  
    }  
}  
  
{  
    title: "50 Tips and Tricks for MongoDB Developer",  
    author: "Kristina Chodorow",  
    published_date: ISODate("2011-05-06"),  
    pages: 68,  
    language: "English",  
    publisher: {  
        name: "O'Reilly Media",  
        founded: 1980,  
        location: "CA"  
    }  
}  
  
{  
    name: "O'Reilly Media",  
    founded: 1980,  
    location: "CA",  
    books: [123456789, 234567890, ...]  
}  
  
{  
    _id: 123456789,  
    title: "MongoDB: The Definitive Guide",  
    author: [ "Kristina Chodorow", "Mike Dirolf" ],  
    published_date: ISODate("2010-09-24"),  
    pages: 216,  
    language: "English"  
}  
  
{  
    _id: 234567890,  
    title: "50 Tips and Tricks for MongoDB Developer",  
    author: "Kristina Chodorow",  
    published_date: ISODate("2011-05-06"),  
    pages: 68,  
    language: "English"  
}  
  
{  
    _id: "oreilly",  
    name: "O'Reilly Media",  
    founded: 1980,  
    location: "CA"  
}  
  
{  
    _id: 123456789,  
    title: "MongoDB: The Definitive Guide",  
    author: [ "Kristina Chodorow", "Mike Dirolf" ],  
    published_date: ISODate("2010-09-24"),  
    pages: 216,  
    language: "English",  
    publisher_id: "oreilly"  
}  
  
{  
    _id: 234567890,  
    title: "50 Tips and Tricks for MongoDB Developer",  
    author: "Kristina Chodorow",  
    published_date: ISODate("2011-05-06"),  
    pages: 68,  
    language: "English",  
    publisher_id: "oreilly"  
}
```

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



# VOLNÉ SCHÉMA

- co když potřebuji pevné schéma?
  - finální aplikace očekává vždy stejná data
    - validace schématu
  - validační úroveň
    - které dokumenty jsou validovány?
    - striktní
      - vložení a aktualizace všech dokumentů
    - mírná
      - vložení všech dokumentů
      - aktualizace jen validních dokumentů
  - validační akce
    - co se stane, když validace selže?
    - chyba
    - varování



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

# VOLNÉ SCHÉMA

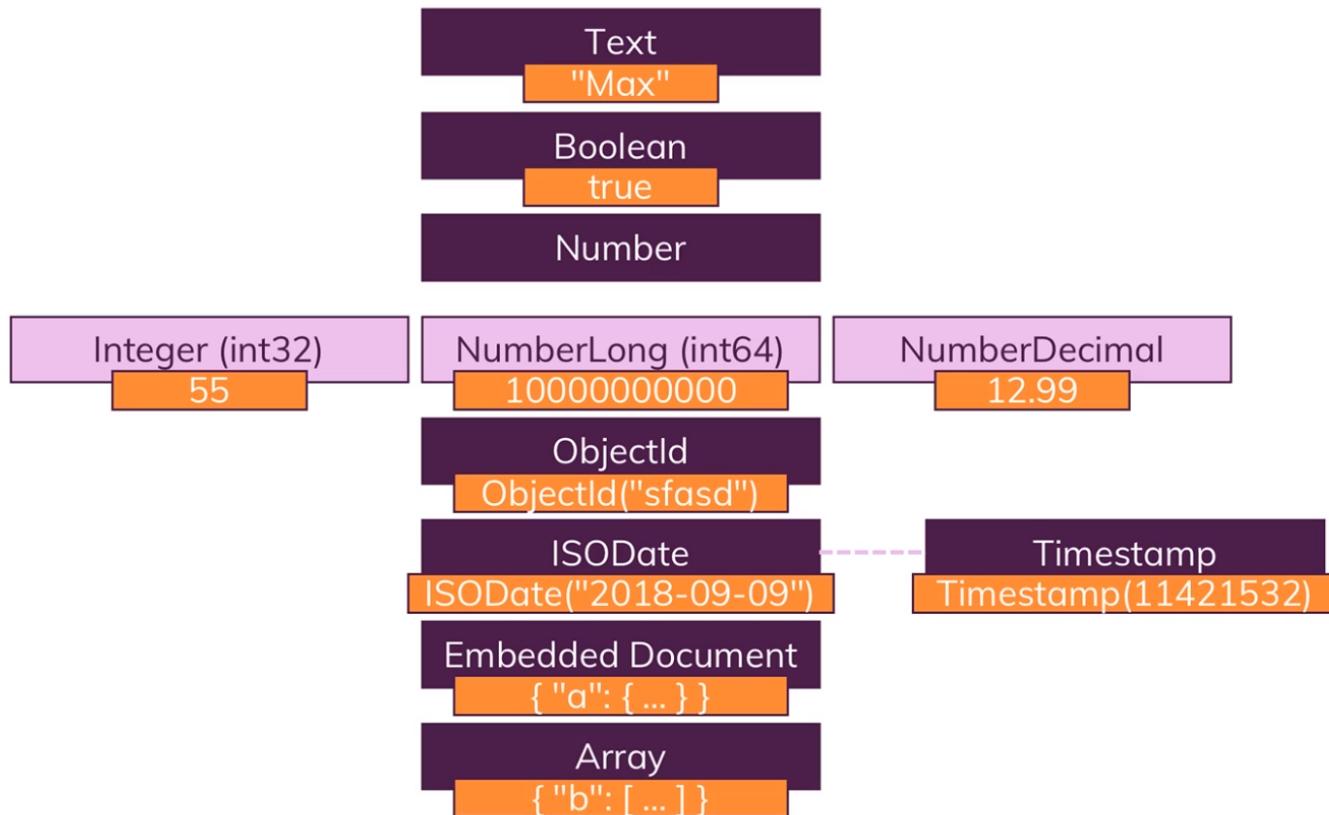
- co když potřebuji pevné schéma?
  - vytvoření validačního schématu
    - při vytvoření kolekce
    - za běhu přes runCommand()

```
db.runCommand({
  collMod: 'posts',
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      required: ['title', 'text', 'creator'],
      properties: {
        title: {
          bsonType: 'string',
        },
        text: {
          bsonType: 'string',
        },
        creator: {
          bsonType: 'objectId',
        },
      }
    }
  },
  validationAction: 'warn'
});
```

```
db.createCollection('posts', {
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      required: ['title', 'text', 'creator', 'comments'],
      properties: {
        title: {
          bsonType: 'string',
          description: 'must be a string'
        },
        text: {
          bsonType: 'string',
        },
        creator: {
          bsonType: 'objectId',
        },
        comments: {
          bsonType: 'array',
          items: {
            bsonType: 'object',
            required: ['text', 'author'],
            properties: {
              text: {
                bsonType: 'string',
              },
              author: {
                bsonType: 'objectId',
              }
            }
          }
        }
      }
    }
  }
});
```



# DATOVÉ TYPY



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





# AD-HOC DOTAZY

```
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",
  "body" : "Body of post three",
  "category" : "News",
  "likes" : 81,
  "tags" : [
    "news",
    "events"
  ],
  "user" : {
    "name" : "John Doe",
    "status" : "author"
  },
  "date" : "Date()"
}

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",
  "body" : "Body of post five",
  "category" : "Technology",
  "date" : "Mon Jul 15 2019 16:08:10 GMT-0400 (EDT)"
}

> db.posts.find().sort({ title: -1 }).limit(2)
{
  "_id" : ObjectId("5d2bdd2af222de7f3b9d281b"),
  "title" : "Post Two",
  "body" : "Body of post two",
  "category" : "Technology",
  "date" : "Mon Jul 15 2019 16:08:10 GMT-0400 (EDT)"
}
{
  "_id" : ObjectId("5d2bdd2af222de7f3b9d281c"),
  "title" : "Post Three",
  "body" : "Body of post three",
  "category" : "News",
  "date" : "Mon Jul 15 2019 16:08:10 GMT-0400 (EDT)"
}
```



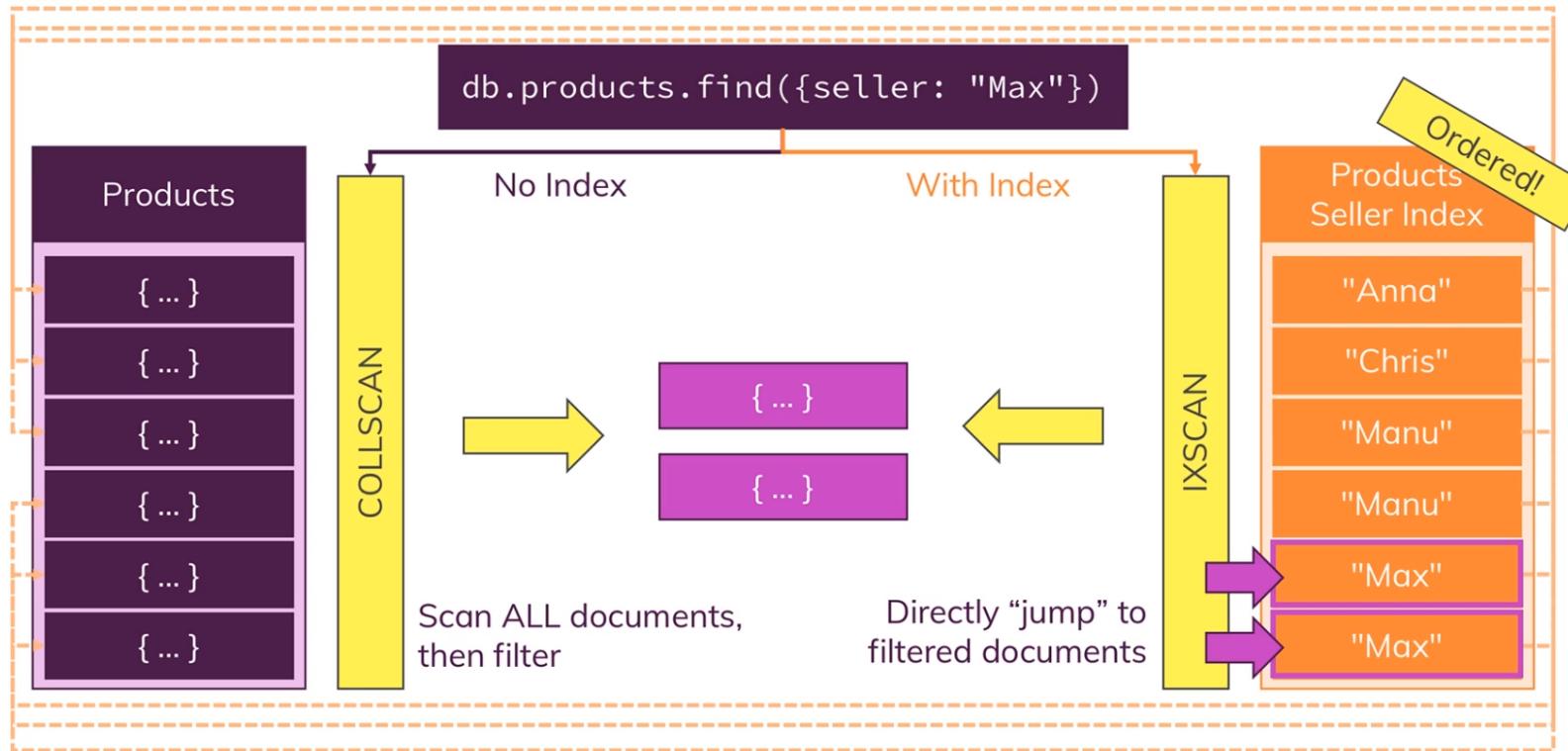


# AD-HOC DOTAZY

- dotazy neznámé při návrhu databáze
  - vytvořené až při potřebě získat určitou informaci
- podpora
  - hledání podle pole dokumentu
  - rozsahové dotazy
  - hledání podle regulárních výrazů
- dotazy mohou vracet
  - specifická pole dokumentu
  - náhodný vzorek výsledků o dané velikosti
  - a také obsahovat uživatelem definované JS funkce



# INDEXOVÁNÍ



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



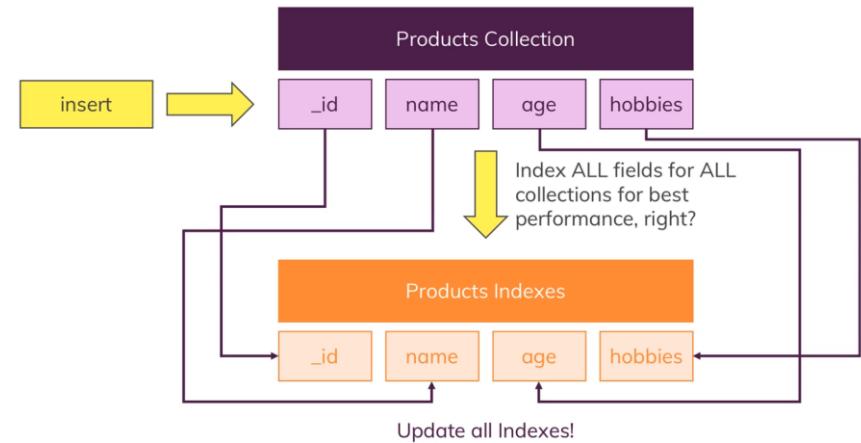
# INDEXOVÁNÍ

- vylepšuje rychlosť vyhľadávania
  - bez indexovania sa provádzí sken celé kolekcia
  - vhodný index výrazne omezuje dokumenty, ktoré je potreba skenovať
- index
  - speciálna datová struktúra definovaná na úrovni kolekcií
  - ukladá hodnotu specifického pole v seřazené formě
    - snadné procházení a porovnávaní
    - obsahuje také pointer na celý dokument pre snadný prístup
  - možnosť definovať nad libovolným polem ale i nad jejich kombináciami
    - v základe index \_id
  - využitie i pre rýchle řazení
  - základ pre sharding
    - len jeden môže byť použit

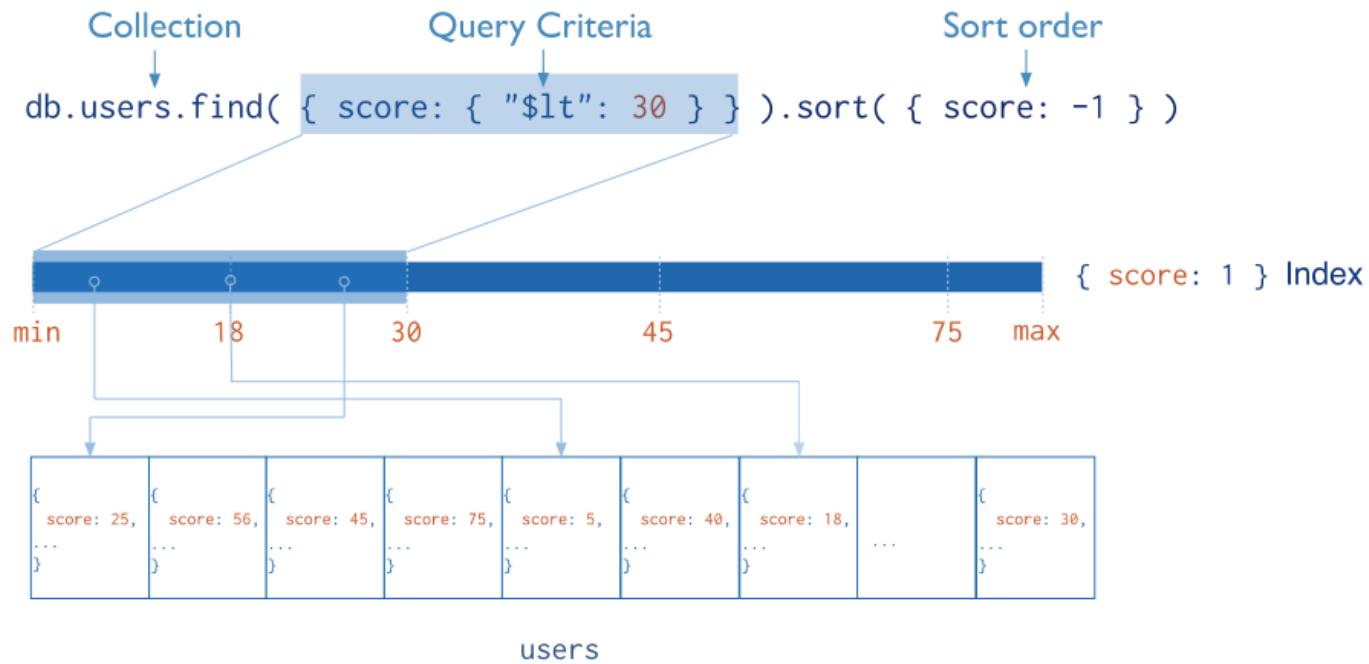
# INDEXOVÁNÍ

- vylepšuje rychlosť vyhľadávania
- proč tedy nedefinovať indexy nad všemi polí?
  - teoreticky vylepší veškeré vyhľadávania
  - v praxi ale index není zadarmo
- cenou je vkládanie / aktualizácia
  - pri každém vložení je potreba vložiť prvek i do seřazeného indexu
    - potreba vložiť do všech indexov
    - pomalé?
- indexy je potreba řádně promyslet
- indexování je pomalejší, když dotaz vrací velkou část kolekce
  - způsobeno přechodem index – pointer – dokument
  - naopak při kompletním skenu už jsou dokumenty načteny v paměti

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



# INDEXOVÁNÍ



<https://docs.mongodb.com/manual/indexes/>



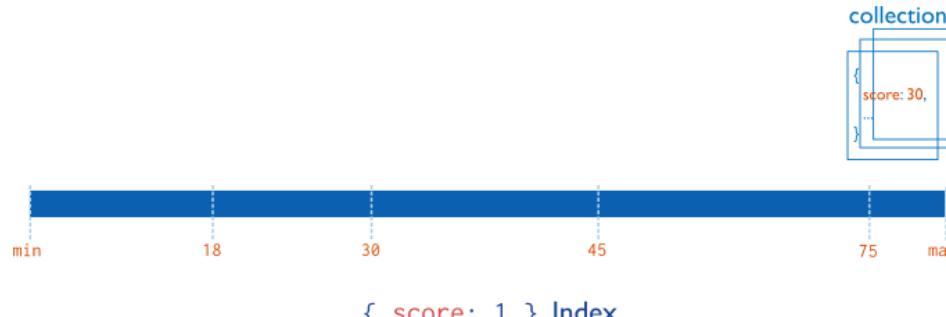
# INDEXOVÁNÍ

- vytvoření / aktualizace indexu dočasně uzamkne kolekci
  - možnost ovlivnit parametrem background
- typy indexů
  - single field
    - index nad jedním polem
  - compound
    - index nad více polí
  - multikey
    - index pro prvky v poli (array)
  - text
    - index pro vyhledávání textových řetězců
  - geospatial
  - hashed



# INDEXOVÁNÍ

- single field index
  - index nad jedním polem
  - sestupný / vzestupný
    - ve výsledku ale není podstatné, MongoDB podporuje oboustranné procházení
  - možnost definovat i nad vnořeným polem

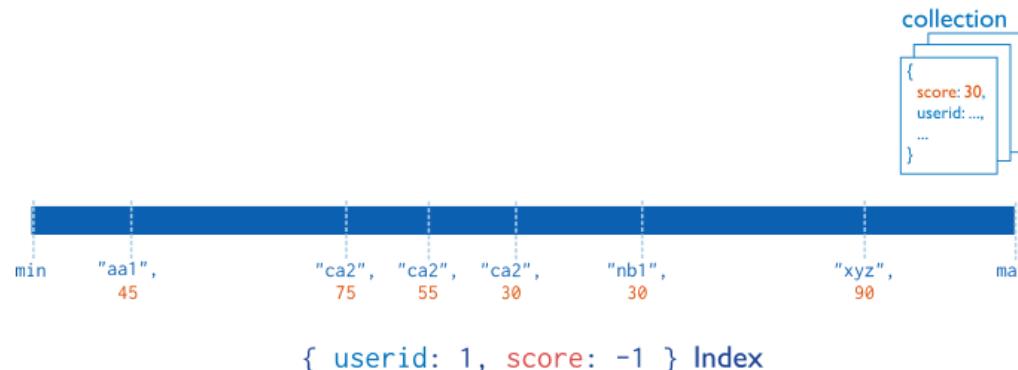


<https://docs.mongodb.com/manual/indexes/>



# INDEXOVÁNÍ

- compound index
  - index nad více polí
  - závisí na pořadí polí
    - řadí se postupně zleva (první pole, druhé pole, ... pole N)
    - lze použít zleva i pro méně polí



<https://docs.mongodb.com/manual/indexes/>



# INDEXOVÁNÍ

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

db.contacts.createIndex({"dob.age": -1})

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

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



# INDEXOVÁNÍ

- `explain()` zobrazí podrobnosti o provedeném dotazu
  - zobrazí např. i index, který byl použit, a dobu zpracování

```
db.contacts.explain("executionStats").find({"dob.age": {$gt: 60}})

db.contacts.createIndex({"dob.age": 1})

"executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1222,
    "executionTimeMillis" : 5,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 5000,
    "executionStages" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "dob.age" : {
                "$gt" : 60
            }
        },
        "nReturned" : 1222,
        "executionTimeMillisEstimate" : 0,
        "works" : 5002,
        "advanced" : 1222,
        "needTime" : 3779,
        "needYield" : 0,
        "saveState" : 39,
        "restoreState" : 39,
        "isEOF" : 1,
        "invalidates" : 0,
        "direction" : "forward",
        "docsExamined" : 5000
    }
},
"executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1222,
    "executionTimeMillis" : 3,
    "totalKeysExamined" : 1222,
    "totalDocsExamined" : 1222,
    "executionStages" : [
        {
            "stage" : "FETCH",
            "nReturned" : 1222,
            "executionTimeMillisEstimate" : 0,
            "works" : 1223,
            "advanced" : 1222,
            "needTime" : 0,
            "needYield" : 0,
            "saveState" : 9,
            "restoreState" : 9,
            "isEOF" : 1,
            "invalidates" : 0,
            "docsExamined" : 1222,
            "alreadyHasObj" : 0,
            "inputStage" : {
                "stage" : "IXSCAN",
                "nReturned" : 1222,
                "executionTimeMillisEstimate" : 0,
                "works" : 1223,
                "advanced" : 1222,
```

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



# INDEXOVÁNÍ

- nastavitelné vlastnosti indexů
  - unikátní (unique)
    - duplikátní hodnoty v indexu nejsou povoleny
    - \_id je unikátní
  - částečný (partial)
    - indexace jen podmnožiny dokumentů, které vyhovují zadanému filtru
    - např. filtr ženy pro pole pohlaví -> indexovány budou jen ženy
    - zabírá méně místa, nižší nároky na vytvoření a správu indexu
  - TTL
    - time to live
    - indexy sloužící k automatickému odstranění dokumentů po uplynutí určitého času
  - řídký (sparse)
  - skrytý (hidden)



# INDEXOVÁNÍ

- multikey index
  - index pro prvky v poli (array)
  - každý prvek z pole je zaindexován samostatně
  - pozor na výkon
    - při vložení dokumentu s velkým počtem prvků v poli musí být všechny zařazeny
  - možnost i pro compound indexy,  
ale jen jedno pole (array)



<https://docs.mongodb.com/manual/indexes/>



# INDEXOVÁNÍ

- **text index**
  - index pro vyhledávání textových řetězců
    - možnost definovat i nad více polí najednou
  - přemění text na seznam slov a ukládá je tak
    - odstraňuje stop slova a provádí stematizaci (podpora různých jazyků)
  - možnost vyhledávat slova i fráze
  - výsledky dotazu jsou skórovány
    - možnost vrátit nejlepší

This product is a must-buy for all fans of modern fiction!



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



# INDEXOVÁNÍ

```
MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.createIndex({ title: 'text'})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1,
    "$clusterTime" : {
        "clusterTime" : Timestamp(
            "signature" : {
                "hash" : BinData(0
                    "keyId" : NumberLo
                )
            }
        ),
        "operationTime" : Timestamp(160328
    }
}

MongoDB Enterprise atlas-12nfzk-shard-0:PRIMARY> db.posts.find({$text: {$search: "\"Post o Ničem\""}).pretty()
{
    "_id" : ObjectId("5f9026800cbc092824d7e420"),
    "source" : "id2",
    "title" : "Post o Ničem",
    "user" : {
        "name" : "Jane Doe",
        "gender" : "female"
    }
}

{
    "_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()"
}

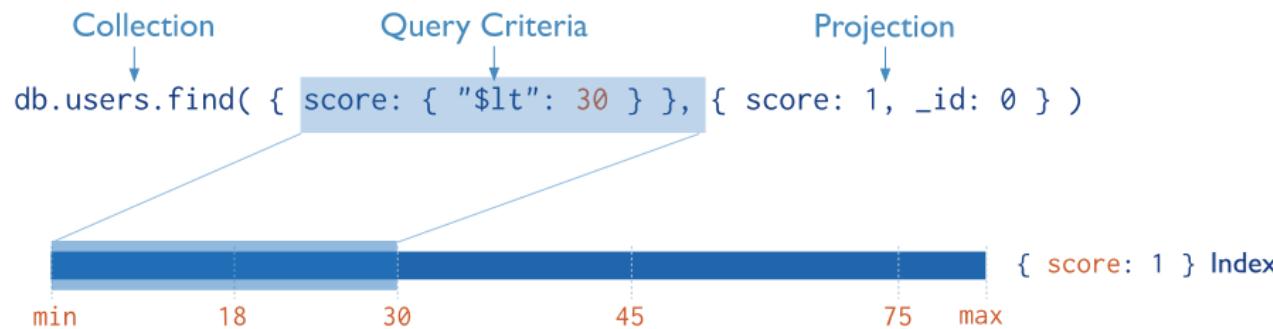
> db.products.find({$text: {$search: "awesome t-shirt"}}, {score: {$meta: "textScore"}}).sort({score: {$meta: "textScore"}}).pretty()
db.products.find({$text: {$search: "awesome -t-shirt"}}).pretty()
```

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



# INDEXOVÁNÍ

- skryté (covered) dotazy
  - dotazy, které získají hledanou odpověď jen v indexu a nemusí přistupovat ke konkrétním dokumentům
  - nejfektivnější



<https://docs.mongodb.com/manual/indexes/>

# TRANSAKCE

- operace nad jedním dokumentem jsou atomické
  - vazby mezi daty často zachyceny pomocí vnořených dokumentů místo normalizovaných dokumentů a kolekcí
  - multi-dokumentové transakce často nejsou potřeba
    - v určitých případech ale chybí
- podpora multi-dokumentových transakcí od verze 4.0
  - transakce mohou být provedeny nad
    - databázemi
    - shardy
    - kolekcemi
    - dokumenty
    - operacemi
  - i tak ale pomalé a ne ideální

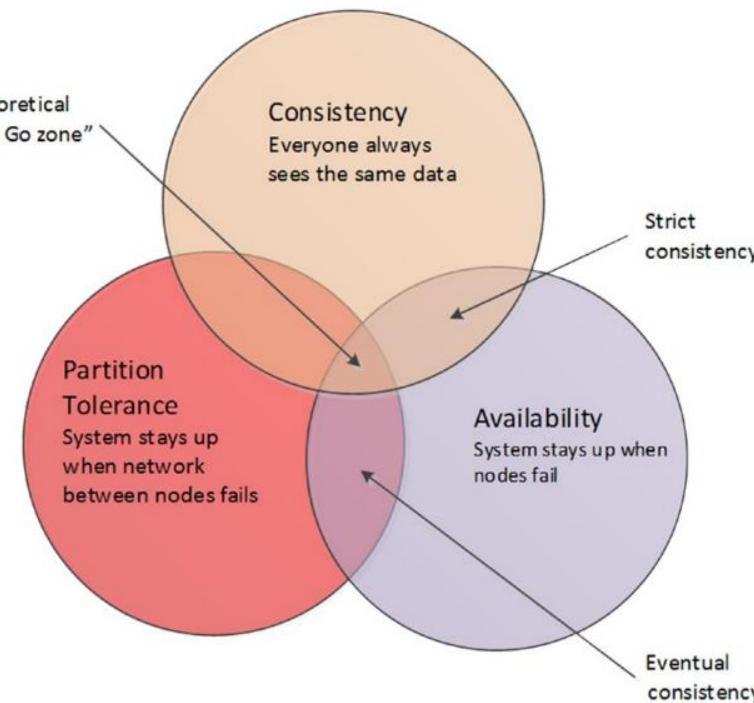
# CAP TEORÉM

- vychází z teoretické informatiky (1999)
- říká, že pro distribuovaný datový sklad nelze zajistit více jak dvě z následujících tří vlastnosti
  - konzistence – consistency
    - všechny uzly mají stejná data
    - každé čtení vrací poslední výsledek, nebo chybu
  - dostupnost – availability
    - na každý dotaz je vrácena (nechybová) odpověď
  - odolnost k přerušení – partition tolerance
    - systém funguje dál i při zdržení či ztrátě zprávy





# CAP TEORÉM

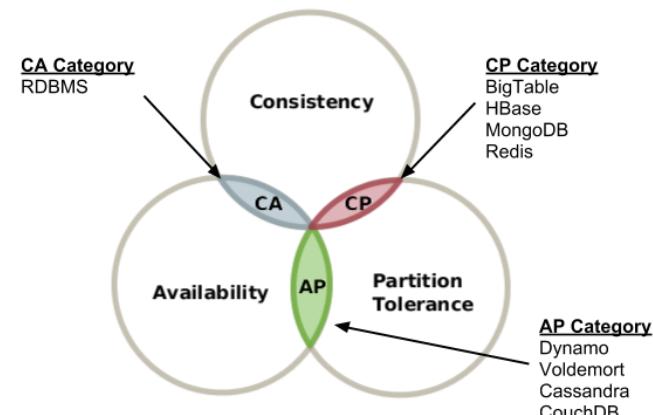


HARRISON, Guy. *Next generation databases: NoSQL, NewSQL, and Big Data*. ISBN 978-1484213308.

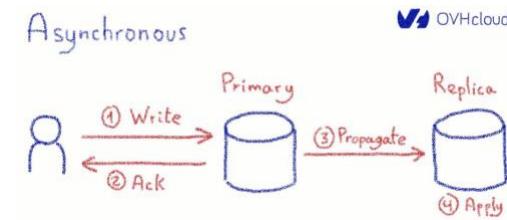


# CAP TEORÉM

- CP z CAP v MongoDB
- odolnost k přerušení (P)
  - partition tolerance
  - nezbytná pro big data
- MongoDB preferuje konzistenci (C) oproti dostupnosti (A)
  - jeden master (primární databáze)
  - selhání mastera způsobí chvílkovou nedostupnost
    - než dojde k nahrazení
- replikace



<https://changeaas.wordpress.com/tag/nosql/>

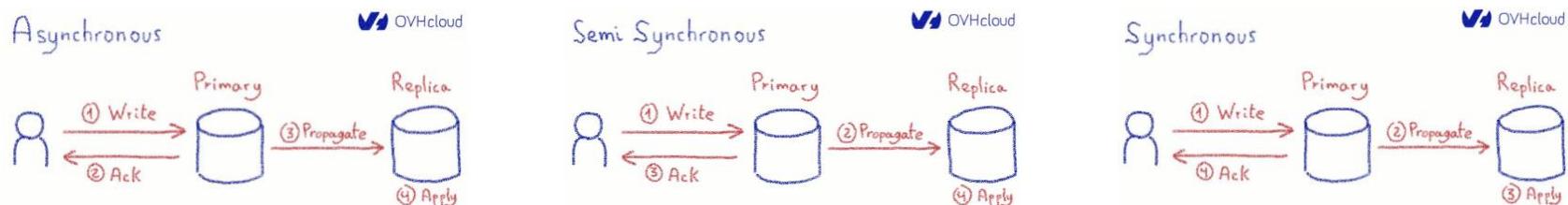


<https://www.ovh.com/blog/database-replication-101/>



# REPLIKACE

- automatická distribuce změn v originále do jeho kopií
- u dokumentových databází nejčastěji typu master-slave
  - obvyklá vazba mezi originálem a kopíemi
  - master zaznamenává změny, které předává slaves
  - slaves potvrdí přijetí změn, čímž umožní další aktualizace
  - asynchronní (eventuálně k.), semi synchronní, synchronní (striktně k.)

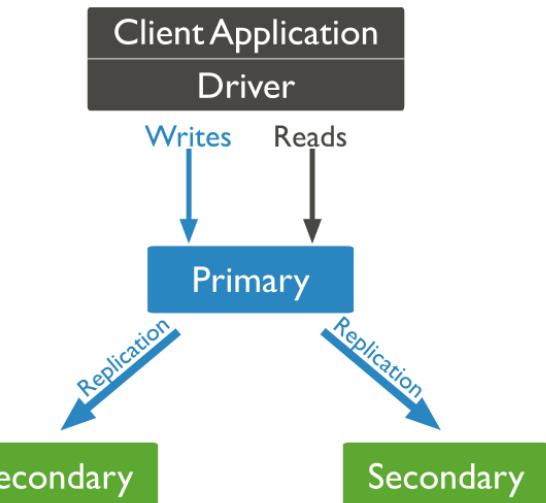


<https://www.ovh.com/blog/database-replication-101/>

- zajišťuje vysokou dostupnost

# REPLIKACE

- MongoDB
  - master – slave architektura
    - 1 master -> CP
  - primární – sekundární uzly
  - asynchronní
- sada replik (replica sets)
  - skupina mongod procesů obsahující stejná data
    - datové uzly (primární a sekundární)
    - volitelně arbiter (max 1)
  - poskytuje redundanci dat a vysokou dostupnost
    - vyšší odolnost vůči výpadkům a ztrátám serverů
    - v určitých případech zrychlení čtení
    - možnost kopií pro specifické účely
      - ochrana proti lidským chybám, disaster recovery, logování, záloha, ...

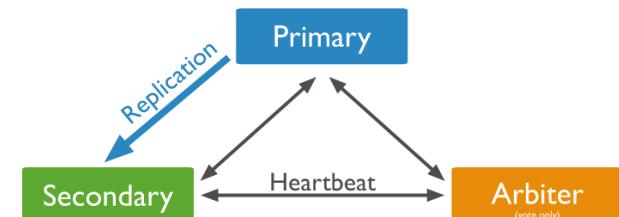
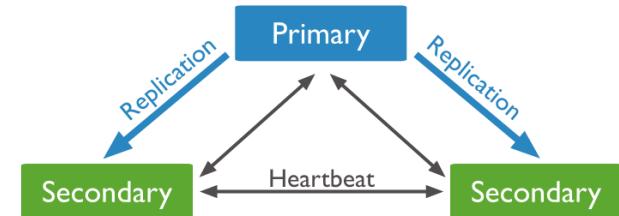


<https://docs.mongodb.com/manual/replication/>



# REPLIKACE

- primární uzel
  - existuje jen jeden
  - obstarává veškeré operace zápisu
  - zaznamenává operace do operačního logu
    - oplog
- sekundární uzly
  - obsahují stejná data jako primární uzel
  - replikují operační log a aplikují operace na svá data
    - aby data opět odpovídala primárnímu uzlu
    - aplikováno asynchronně
- arbiter
  - speciální uzel, který neobsahuje data
  - využití při hlasování o novém primárním uzlu

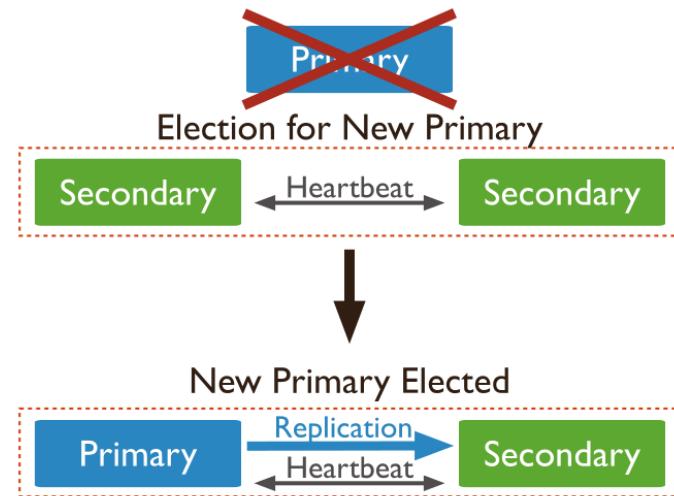


<https://docs.mongodb.com/manual/replication/>



# REPLIKACE

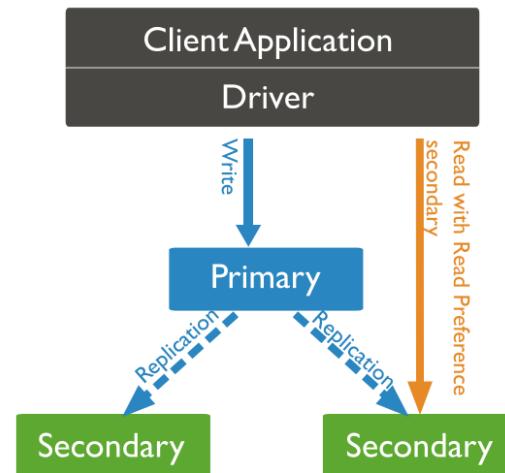
- co se stane při výpadku primárního uzlu?
  - primární uzel nekomunikuje déle než stanovený limit
    - chybějící reakce na heartbeats (ping) posílané mezi uzly každé 2 sekundy
    - v základu 10 sekund
  - jeden ze sekundárních uzlů zažádá o zvolení nového primárního uzlu
    - většina sekundárních uzlů se musí shodnout
      - lichý počet serverů
      - volitelně arbiter
    - v základním nastavení by nemělo trvat déle než 12 sekund
  - i tak je ale porušena dostupnost
    - operace zápisu nejsou možné, dokud není zvolen nový primární uzel
    - čtení je možné, pokud je povoleno ze sekundárních uzlů



<https://docs.mongodb.com/manual/replication/>

# REPLIKACE

- operace čtení
  - v základu z primárního uzlu
  - je ale možné i ze sekundárních
    - asynchronní replikace nezaručuje, že budou přečtena stejná data jako z primárního uzlu
    - není možné pro multi-dokumentové transakce

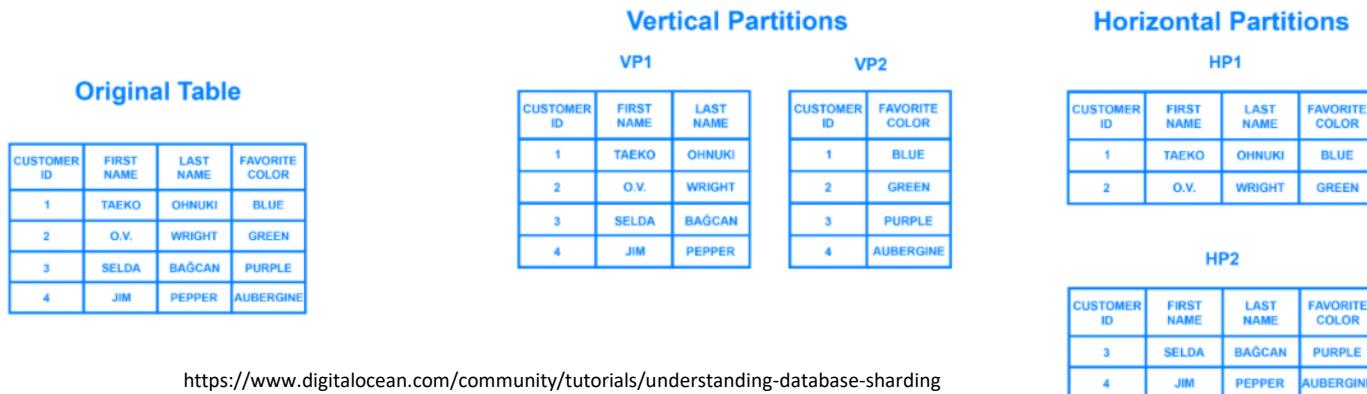


<https://docs.mongodb.com/manual/replication/>



# SHARDING

- databázový vzor pro horizontální škálování na více serverů
- rozdělení záznamů na části (partitions, shards) umístěné na různých serverech
  - např. u relačních databází rozdělení tabulky podle řádků, ne sloupců
  - např. u dokumentových databází rozdělení podle dokumentů, ne atributů

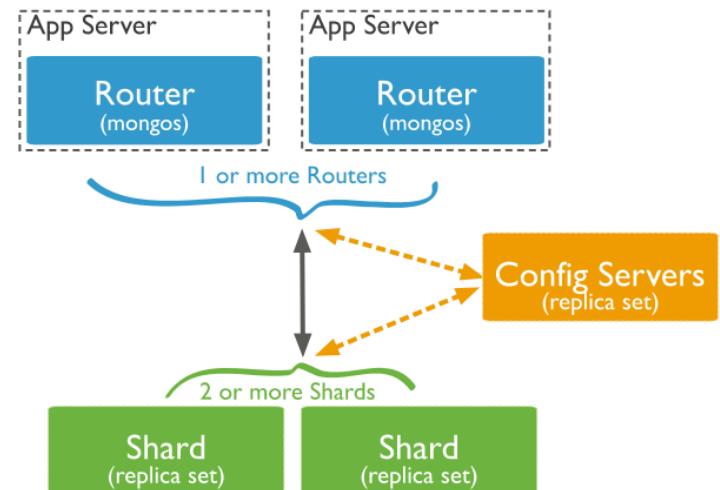


- data mezi shardy nejsou sdílena



# SHARDING

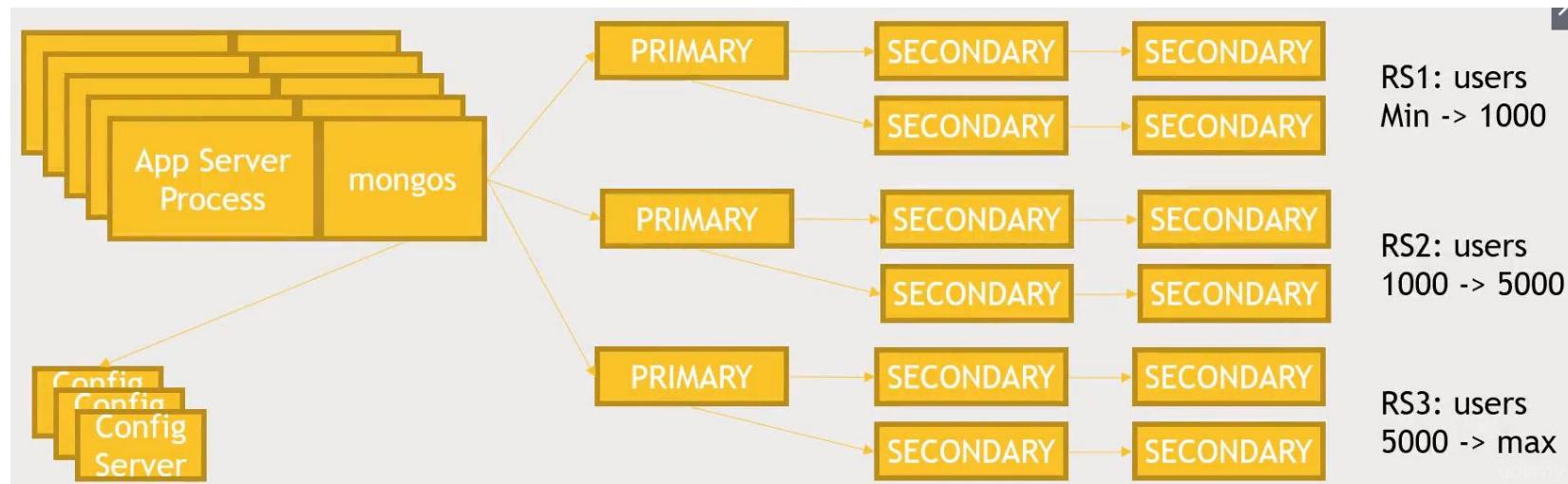
- „konečně“ big data
- shardy – více sad replik (replica sets)
  - každá sada odpovídá za rozsah hodnot
  - musí být definovaný index nad kolekcí
    - slouží k určení rozsahu hodnot a vyvážení zátěže
    - shard key
- služba mongos
  - v serveru komunikujícím s databází
  - komunikuje se 3 konfiguračními servery
    - poskytuje informaci, jak jsou data rozdělena
  - vybere odpovídající sadu replik



<https://docs.mongodb.com/manual/sharding/>



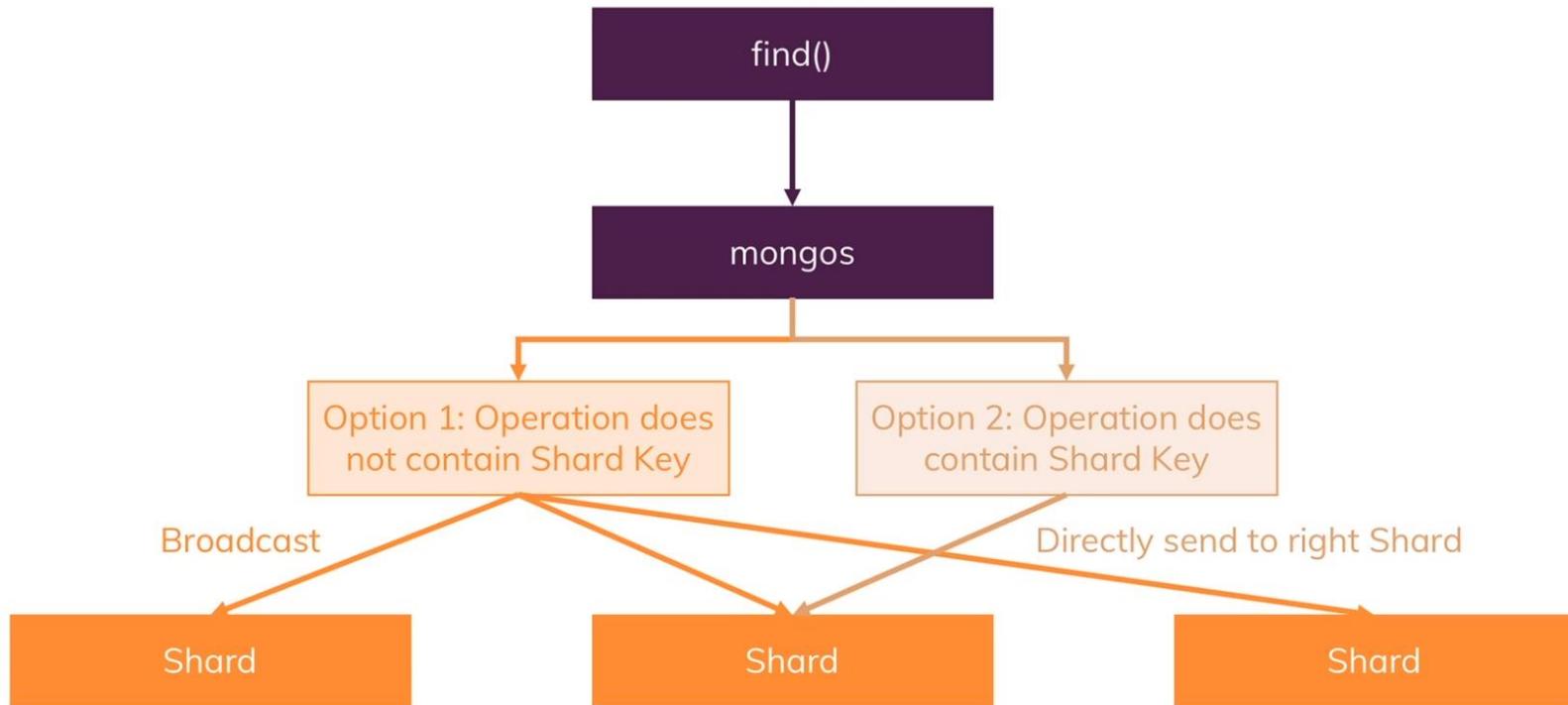
# SHARDING



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

- v praxi relativně pomalé a komplikované
  - závisí na správné volbě indexů
  - Cassandra vhodnější pro horizontální škálování

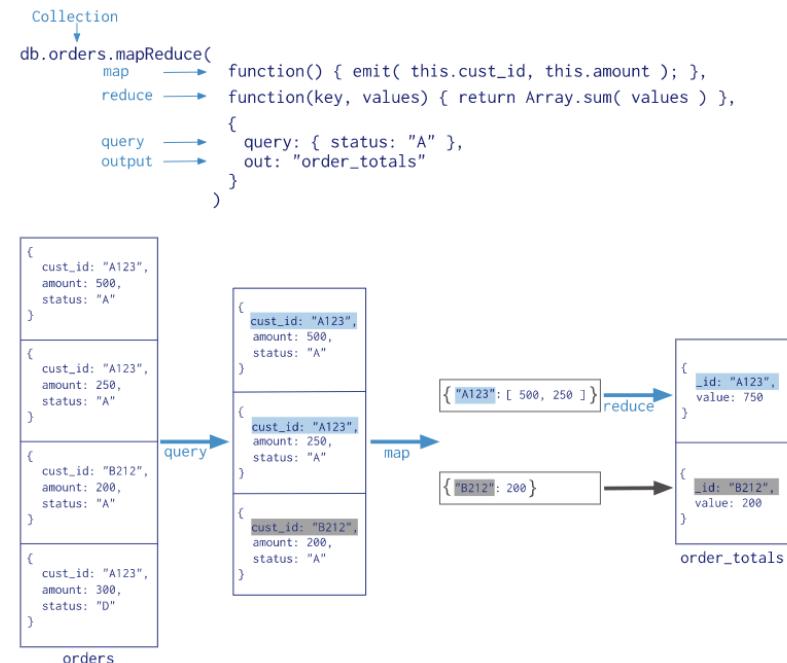
# SHARDING



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

# AGREGACE

- dávkové zpracování dokumentů vracející jeden výsledek i po provedení celé řady operací
- tři možnosti v MongoDB
  - agregační roura (pipeline)
    - dokumenty jsou zpracovávány postupně v krocích až do konečného výsledku
  - jednoúčelová agregace
    - pro dokumenty v jedné kolekci
  - map-reduce (deprecated)
    - dvě fáze
      - map – mapování
        - zpracování dokumentů do objektů odpovídajících vstupním dokumentům
      - reduce – redukce
        - zkombinování výstupů z mapování



<https://docs.mongodb.com/manual/aggregation/>



# SOUBOROVÝ SYSTÉM

- GridFS
  - specifikace pro ukládání a načítání souborů překračujících limit 16 MB
  - místo celého dokumentu jsou ukládány tzv. chunky
    - každý chunk je uložen jako samostatný dokument
    - v základu velikost 255 kB
  - využívá dvě kolekce
    - první kolekce ukládá chunky
    - druhá metadata
  - v případě dotazu jsou potřebné chunky sestaveny do původního stavu
    - nemusí být sestaven celý soubor, ale jen jeho potřebná část
      - např. video od určitého času
      - vhodné, pokud je často zájem jen o část velkého dokumentu
  - nevhodné pro atomické operace nad celým dokumentem

# ZABEZPEČENÍ

- podpora různých forem zabezpečení
  - autentifikace, nastavení přístupů, šifrování, ...



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



Authentication	Authorization	TLS/SSL
Authentication	Role-Based Access Control	TLS/SSL (Transport Encryption)
SCRAM	Enable Access Control	Configure mongod and mongo for TLS/SSL
x.509	Manage Users and Roles	TLS/SSL Configuration for Clients
Enterprise Only	Encryption	
Kerberos Authentication	Client-Side Field Level Encryption	
LDAP Proxy Authentication		
Encryption at Rest		
Auditing		



# VÝHODY

- flexibilní databáze
  - není pevně dané schéma
- ad-hoc dotazování
- vestavěna agregace
- horizontální škálování & sharding
- vysoká dostupnost
  - minimalizace nedostupnosti pomocí replikace
- rychlosť
  - pokud jsou data v paměti
  - závislá na správně zvolených indexech
    - snadný přístup k dokumentům pomocí indexování
- geospatial data



# OMEZENÍ

- **joiny**
  - možnost doprogramovat, i tak ale pomalejší
- **rychlosť**
  - může být výrazně pomalejší než u relačních databází
  - špatně zvolené indexy & data na disku
  - horizontální škálování & sharding
- **paměťová náročnost**
  - data v paměti vs. indexy v paměti
- **komplexní transakce**
- **velikost dokumentu (16 MB)**
- **vnořování dokumentů (max 100)**



# A PŘÍŠTĚ?

- agregace a MongoDB





Děkuji za pozornost.  
Otázky?

