# 33. Dokumentové databáze- koncept, srovnání s key-value úložišti, pojem dokument, výhody a nevýhody. MongoDB- charakteristika a architektura.

## Dokumentové databáze- Koncept, srovnání s key-value úložišti a pojem dokument

- 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
- 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

- základní stavební prvky dokumentových databází
  - 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

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

  - 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

Ukázka v XML

```xml
<artist>
    <artistname>Iron Maiden</artistname>
    <albums>
     <album>
        <albumname>The Book of Souls</albumname>
        <datereleased>2015</datereleased>
        <genre>Hard Rock</genre>
     </album>
     <album>
        <albumname>Killers</albumname>
        <datereleased>1981</datereleased>
        <genre>Hard Rock</genre>
     </album>
     <album>
        <albumname>Powerslave</albumname>
        <datereleased>1984</datereleased>
        <genre>Hard Rock</genre>
     </album>
     <album>
        <albumname>Somewhere in Time</albumname>
        <datereleased>1986</datereleased>
        <genre>Hard Rock</genre>
     </album>
    </albums>
</artist>
```

Ukázka v JSON:

```json
{
    "_id" : 1,
    "artistName" : "Iron Maiden",
    "albums" : [
        {
            "albumname" : "The Book of Souls",
            "datereleased" : 2015,
            "genre" : "Hard Rock"
        }, {
            "albumname" : "Killers",
            "datereleased" : 1981,
            "genre" : "Hard Rock"
        }, {
            "albumname" : "Powerslave",
            "datereleased" : 1984,
            "genre" : "Hard Rock"
        }, {
            "albumname" : "Somewhere in Time",
            "datereleased" : 1986,
            "genre" : "Hard Rock"
        }
    ]
}
```

- ukázka volného schématu ve formátu JSON


```
_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()"
```

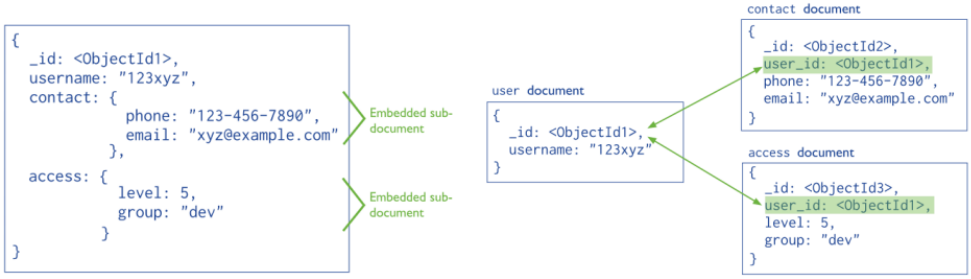
```
_id: ObjectId("5f9026800cbc092824d7e420")
source: "id2"
title: "Unknown"
user: Object
  name: "Jane Doe"
  gender: "female"
```


## Charakteristiky

- 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
    - není ale pravidlem

https://medium.com/@hau12a1/golang-http-crud-l-the-create-part-ae42c962c557

  - obecně se vyhýbají vazbám mezi dokumenty
    - případně dvě varianty řešení
      - embedded dokumenty
      - reference

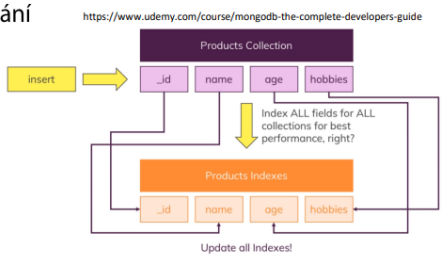


# INDEXOVÁNÍ

- vylepšuje rychlost vyhledávání
  - bez indexování se provádí sken celé kolekce
  - vhodný index výrazně omezuje dokumenty, které je potřeba skenovat
- index
  - speciální datová struktura definovaná na úrovni kolekcí
  - ukládá hodnotu specifického pole v seřazené formě
    - snadné procházení a porovnávání
    - obsahuje také pointer na celý dokument pro snadný přístup
  - možnost definovat nad libovolným polem ale i nad jejich kombinací
    - v základu index _id
  - využití i pro rychlé řazení
  - základ pro sharding
    - jen jeden může být použit

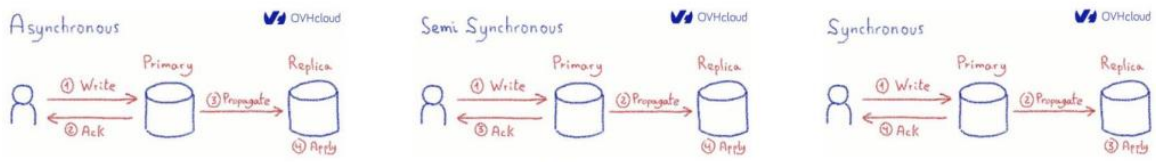
# INDEXOVÁNÍ

- vylepšuje rychlost vyhledávání
- proč tedy nedefinovat indexy nad všemi poli?
  - teoreticky vylepší veškeré vyhledávání
  - v praxi ale index není zadarmo
  - cenou je vkládání / aktualizace
    - při každém vložení je potřeba vložit prvek i do seřazeného indexu
    - potřeba vložit do všech indexů
    - pomalé?
  - indexy je potřeba 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

- u dokumentových databází nejčastěji typu master-slave
  - obvyklá vazba mezi originálem a kopiemi
  - master zaznamenává změny, které předává slaves
  - slaves potvrdí přijetí změn, čímž umožní další aktualizace
  - asynchronní (eventuálně k.), semi synchronní, synchronní (striktně k.)

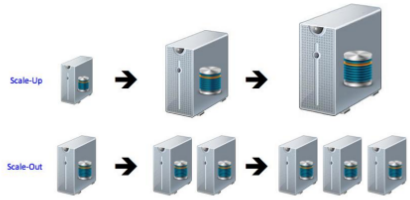



https://www.ovh.com/blog/database-replication-101/

- zajišťuje vysokou dostupnost

## Škálování

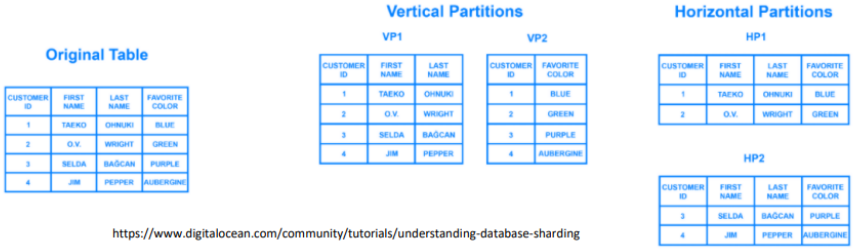
- u dokumentových databází nejčastěji horizontální škálování (ven)
  - přidání (nebo ubrání) prvků
  - ➢ přidání výpočetních uzlů
- přesun k distribuovanému paralelnímu zpracování
  - rozdělení dat mezi uzly
  - ➢ horizontální sharding
- zvýšení kapacity
  - nové komponenty jsou levné
    - základní HW
  - ➢ distribuované clustery
- cloudové služby řeší za uživatele
- ➢ big data




https://medium.com/faun/scalability-248019b918ed

## 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

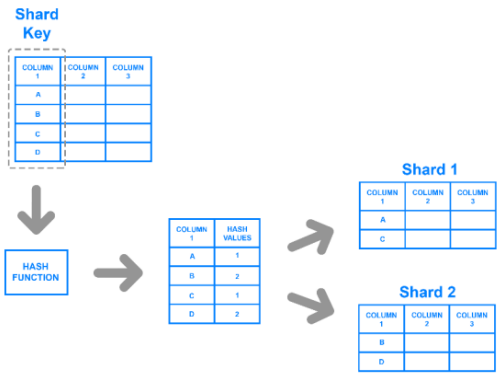



https://www.digitalocean.com/community/tutorials/understanding-database-sharding

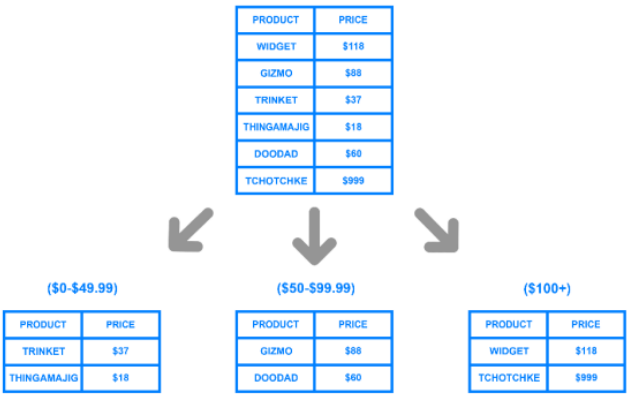
  - data mezi shardy nejsou sdílena

Architektury:

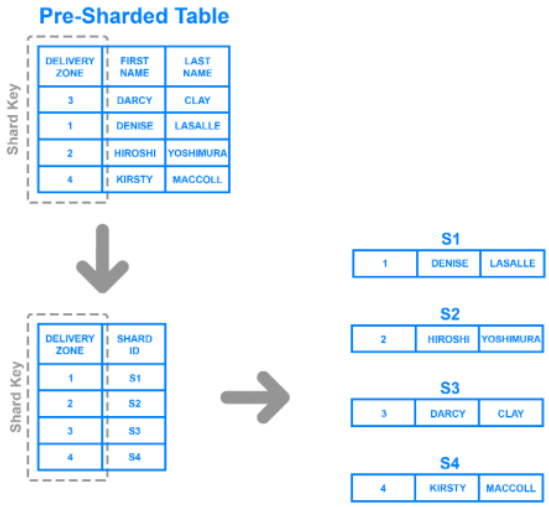
- Založena na klíči (hashi)




- Založena na rozsahu




- Založena na adresářích




## Použití

- široké možnosti použití (ukládání)
  - webové aplikace
    - blogovací platformy, analytická data, nastavení uživatelů, e-reklamy, …
  - data generovaná uživateli
    - chaty, tweety, příspěvky, hodnocení, komentáře, …
  - katalogy
    - uživatelské účty, produkty, preference, …
  - počítačové hry
    - herní statistiky, žebříčky, vestavěné chaty, splněné úkoly, integrace social media, …
  - networking
    - data ze senzorů, logy, real-time analýza, …
  - …

## výhody a nevýhody.

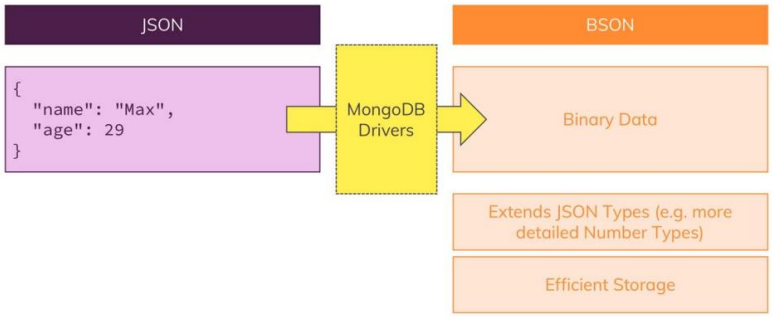
## MongoDB – charakteristika, architektura.

- Dokumenty ve formátu BSON (binární JSON)
- Hlavní funkce:
  - volné schéma
  - ad-hoc dotazy
    - dotazy neznámé v době vytvoření databáze
  - indexování
  - replikace
    - vysoká dostupnost
  - sharding
  - agregace
- 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
                }
        ]
}
```

# JSON VS. BSON

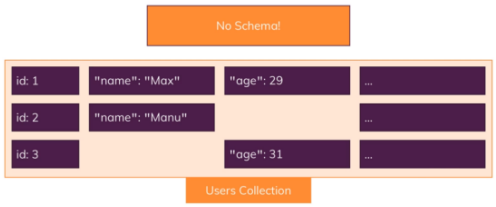



# DOKUMENT

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

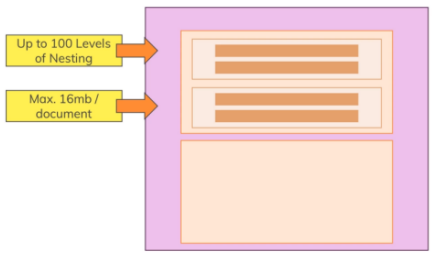

https://www.udemy.com/course/the-ultimate-hands-on-hadoop-tame-your-big-data

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



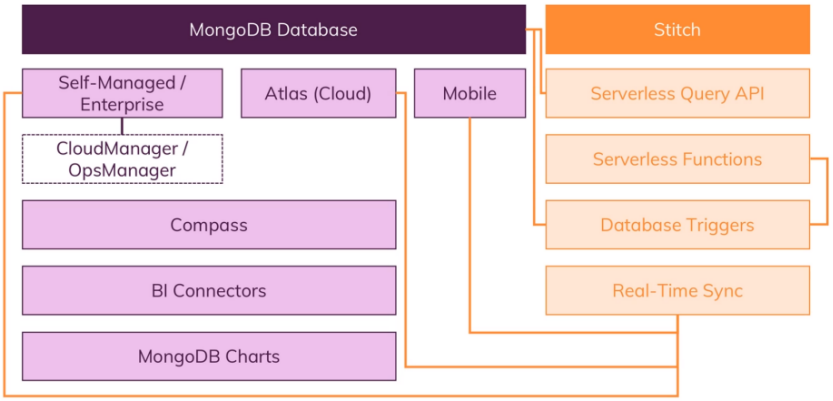

# DOKUMENTY

- 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

# MongoDB EKOSYSTÉM




# 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…