MongoDB

OBSAH

Ciele

• Pochopiť kľúčové vlastnosti MongoDB
• Zvládnuť techniky a operácie na riešenie dopytovania a základných úloh MongoDB

T1) Základy MongoDB, Modelovanie

1a) Model MongoDB

1b) Inštalácia a jemný úvod

1c) Mongo Shell

1d) Dokumentový model

1e) Modelovanie dát

1a) Model MongoDB

MongoDB z hľadiska SQL

Úvod do MongoDB a dátového modelu, štruktúra dokumentu

MongoDB (humongous - obrovský)

MongoDB je systém pre správu **dokument**-orientovaných/dokumentových databáz, ktorý je viacplatformový (Linux, Windows, Mac), bezplatný, open-source a zaraďuje sa medzi NoSQL databázy.

MongoDB bol navrhovaný tak, aby poskytol vysoký výkon a automatickú škálovateľnosť. MongoDB dve najzákladnejšie požiadavky pri práci s big datami, rýchlosť a dostupnosť, dosahuje na úkor takých vlastností relačných DB, ako konzistentnosť a transakcia.

MongoDB	db	kolekcia	dokument	kľúč/pole	vnorenie, odkazovanie	find
RDB	db	tabuľka	riadok	stĺpec	JOIN	select

Na rozdiel od relačných DB, ktoré využívajú relácie,tabuľky, MongoDB je založená na dokumentoch s dynamickými schémami. Práve táto vlastnosť, že dokumenty môžu mať premenlivú štruktúru, premenlivý počet polí/fieldov/key, uľahčuje integráciu rôznorodých dát.

Dokumenty pozostávajú z key-value párov, kľúč-hodnota. Na key sa odvoláva aj ako field/pole, ale my pojem pole necháme na pomenovanie polí, array.

Z dokumentov sa tvoria pomenované kolekcie - MongoDB ukladá dokumenty do kolekcií. Databázy obsahujú kolekcie. Kolekcie zodpovedajú tabuľkám v relačných databázach. Na rozdiel od tabuľky, kolekcia nevyžaduje, aby jej dokumenty mali rovnakú schému. Na pridanie dokumentov do kolekcie MongoDB používa metódu insert (). Ak sa pokúsime pridať dokumenty do kolekcie, ktorá neexistuje, MongoDB automaticky vytvorí kolekciu.

1b) Inštalácia a jemný úvod

Server MongoDB a klient spúštame v dvoch príkazových riadkoch pomocou mongod.exe a mongo.exe.

Win + R -- cmd -- cd C:\Program Files\MongoDB\Server\3.0\bin -- mongod.exe

Win + R -- cmd -- cd C:\Program Files\MongoDB\Server\3.0\bin -- mongo.exe

Zoznam všetkých DB získame príkazom (Alt+Space – e – p)

```
> show dbs
```

respectívne príkazom

```
> show databases
```

```
> show databases
blog   0.078GB
local  0.078GB
test   0.078GB
```

```
> version()
```

3.0.4

Poznamenáme, že príkazy sú case-sensitive a je odporúčané ukončiť ich bodkočiarkou.

Ktorá DB je aktuálna zistíme príkazom

```
> db;
```

Aktuálnu databázu zastupuje db. Zoznam všetkých kolekcií aktuáalnej databázy získame pomocou

```
> show collections;
```

Zoznam všetkých dokumentov v kolekcii kolMaz vráti

```
> db.kolMaz.find( );
```

Novú databázu vytvoríme, resp. prepneme sa na novú databázu, pomocou use. Po vykonaní

```
> use dbMaz;
> show databases;
```

v obdržanom zozname nenájdeme dbMaz. Aby sme ju uvideli, musíme do nej vložiť kolekciu

```
db.dropDatabase()
use dbMaz;
show databases;
// { meno : "Fero", vaha : 82 } je dokument
db.Tab1.insert({ meno : "Fero", vaha : 82 }) ;
db.Tab1.insert({ meno : "Jano", vaha : 88 });
show databases;
db.Tab1.find( );  // zoznam dvoch dokumentov

show collections ; // ⇔ // show tables // vratia stlpcovy zoznam
db.getCollectionNames() ;                  // vrati pole!!
db.Tab1.drop();

//db.dropDatabase();
```

V skutočnost DB a kolekcia sa vytvorí po insert.

1c) Mongo Shell

A) Prehľad JavaScript metód, funkcií.

Príkaz, definujúci premennú, sa nevytlačí, ak ho deklarujeme pomocou var, porovnaj dt a y.

```
11*11;

dt = new Date("2016/3/3");
dt.getFullYear();  // js
dt.toString();        //

var y=Math.cos(Math.PI);
y;

//
function faktMDB (n) {
  if (n <= 1) return 1;
  return n * faktMDB(n - 1);
};
faktMDB(5);
```

B) Mongo Shell metódy

- Kolekcia
  - insert, insertOne, insertMany
  - find, findOne, distinct, findOneAndUpdate
  - update, updateOne, updateMany, replaceOne
  - drop, dropIndex
  - remove, deleteOne, deleteMany
  - validate
  - aggregate, count, group
  - mapReduce

- Kurzor
  - count, min, max
  - forEach, next, hasNext
  - limit, skip, size,
  - map, sort, toArray
- DB
  - copyDatabase, cloneDatabase, cloneCollection
  - dropDatabase, createCollection
  - getCollection, getCollectionNames, getCollectionInfos, getName runCommand
- Ďalších 10 tém, ako Sharding

Agregácia (pozri aj tretiu prednášku k tejto téme)

```
db.Tab1.count();          // kolekcia
// ⇔
db.Tab1.find().count();   // kurzor

db.Tab1.find( { vaha: { $gt: 81 } } ).count();
```

1d) Dokumentový model

- Dokumenty

- BSON Typy

Dokumenty

- Document Format
- Document Structure
- Field Names
- Field Value Limit
- Document Limitations
- The _id Field
- Dot Notation
- Additional Resources

Štruktúra dokumentu

```
{
   field1: value1,
   field2: { fieldA: valueA, fieldB: valueB},  // vnoreny dokument
   field: [val1, val2],   // pole hodnot
   ...
   fieldN: valueN
}
```

Kolekcia **`Autori`** s poľom vnorených dokumentov [{},{}] namiesto dvoch kolekcií *Spisovatelia* a *Knihy*.

Kolekcia **`Autori`** s dvomi dokumentami, kde dva vnorené dokumenty tvoria pole.

```
db.Autori.insert(
[{
      meno: "Sano",
```

```
      adresa: "KE",
      dat_nar: "1980",
      knihy: [
            { nazov: "Financie",
              zaner: "Ekonomika",
              rok: 2008,
              cena: 45},
            { nazov: " Algoritmy ",
              zaner: "PC",
              rok: 2012,
              cena: 40}]
},
{
      meno: "Imro",
      adresa: "AL",
      dat_nar: "1990",
      knihy: [
            { nazov: "RDBS",
              zaner: "PC",
              rok: 2010,
              cena: 45},
            { nazov: "NoSQL",
              zaner: "PC",
              rok: 2011,
              cena: 40}]
}]
)

db.Autori.findOne();

db.Autori.find();
```

Validácia

```
db.Autori.validate( {
$and: [
      {dat_nar: {$lte: 2016}},
      {$or: [
          {meno: { $type: "string" }},
          {adresa: { $type: "string" }}
            ]}
       ],full:true});
```

...

```
},
"valid" :
"errors"
"ok" : 1
```

https://docs.mongodb.com/manual/reference/command/validate/ https://docs.mongodb.com/manual/reference/command/collMod/#dbcmd.collMod https://docs.mongodb.org/manual/core/document-validation/

BSON Typy

MongoDB je založená na BSON/JSON dokumentoch s rozšíreným počtom typov a dynamickými schémami. JSON ponúka iba šesť typov

boolean, numeric, string, array, object, null

BSON poskytuje ďalšie typy ako **`int, long, double`**, **`date`** alebo **`javascript`** a **`regex.`**

1e) Modelovanie dát

MongoDB nepodporuje JOIN. Dáta sú v ňom

- denormalizované, uložené spolu so súvisiacimi dátami alebo
- normalizované, uloženie v samostatných dokumentoch rôznych kolekcií.

Návrh dátových modelov

https://blog.mongodb.org/post/87200945828/6-rules-of-thumb-for-mongodb-schema-design-part-1 https://docs.mongodb.org/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/ https://www.mongodb.com/blog/post/thinking-documents-part-1

Dva základné dátové modely v MongoDB sú

- vnorené dátové modely (denormalizované)
- normalizované dátové modely (referencie medzi dokumentami) .

Ako modelovať vzťah 1:N?
Pri navrhovaní schém v MongoDB je potrebné, aby sme upresnili svoj 1:N vzťah:
ide o 1-málo, 1:veľa alebo 1:obrovské množstvo?
Podľa toho, o ktorý prípad ide, použijeme inú schému.

Existujú tri základné spôsoby a dve vyspelejšie návrhové schémy

1) Modelovanie 1:málo - vložené pole
2) Modelovanie 1:veľa - odkaz na dieťaťa
3) Modelovanie 1:obrovské množstvo - odkaz na rodiča
4) denormalizácia
5) dvoj/obojsmerné odkazovanie

Základné spôsoby

1) Modelovanie 1:málo - vložené pole

- knihy autori (pozri vyššie) https://docs.mongodb.org/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/#data-modeling-example-one-to-many
- adresy osoby

Vložíme knihy do poľa vnútri objektu spisovateľ.

- Výhodou je, že nemusíme vykonávať ďalší dopyt na získanie vložených údajov
- Nevýhodou je, že nie je štandardný spôsob, ako pristupovať k jednotlivým vloženým údajom ako samostatné objekty.

2) Modelovanie 1:veľa - odkaz na dieťaťa

- pacienti lekára https://docs.mongodb.org/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/
- náhradné súčiastky produktu

```
use Poliklinika;
db.dropDatabase();
use Poliklinika;
db.Pacienti.insert(
[{
    _id: 'AAAAA',
    rodneCislo: '12345678',
    meno: 'Pac1',
    vaha: 87,
    dat_navstevy: ISODate("2016-02-27T08:31:42.273Z"),
    poplatok: '15'
},
{
    _id: 'AAAAB',
    rodneCislo: '12345679',
    meno: 'Pac2',
    vaha: 70,
    dat_navstevy: ISODate("2016-02-27T08:55:45.214Z"),
    poplatok: '11'
}]
);

// Lekari
db.Lekari.insert(
{
    name : 'Imro',
    spec : 'zubar',
    kod: 1234,
    pacienti : ['AAAAA', 'AAAAB' ]  // odkaz, referencia na pacienta
}
);

db.Pacienti.findOne();

// Vráť dokument lekára pomocou kódu
var lekar = db.Lekari.findOne({kod: 1234});
```

```
// Vráť všetkých pacientov toho lekára
var lekarove_pac = db.Pacienti.find({_id: { $in : lekar.pacienti } }
).toArray();

lekar;
{
        "_id" : ObjectId("570eb08fff88a5e44573cdeb"),
        "name" : "Imro",
        "spec" : "zubar",
        "kod" : 1234,
        "pacienti" : [
                "AAAAA",
                "AAAAB"
        ]
}

lekarove_pac;
[
        {
                "_id" : "AAAAA",
                "rodneCislo" : "12345678",
                "meno" : "Pac1",
                "vaha" : 87,
                "dat_navstevy" : ISODate("2016-02-27T08:31:42.273Z"),
                "poplatok" : "15"
        },
        {
                "_id" : "AAAAB",
                "rodneCislo" : "12345679",
                "meno" : "Pac2",
                "vaha" : 70,
                "dat_navstevy" : ISODate("2016-02-27T08:55:45.214Z"),
                "poplatok" : "11"
        }
]
```

- Výhodou tohto modelu je, že Lekari a Pacienti sú samostatné kolekcie a tak dopytovať a modifikovať ich je ľahké.
- Nevýhodou je, že k získaniu pacientov lekára je nutné vykonať ďalší dopyt.

3) Modelovanie 1:obrovské množstvo

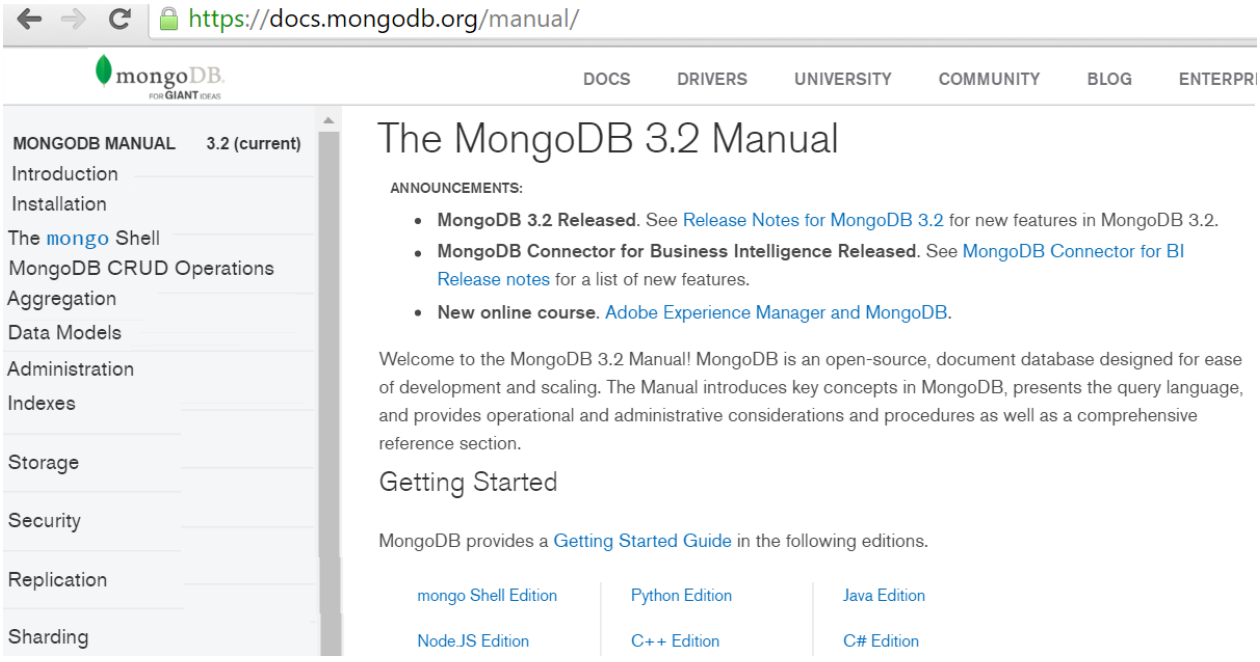
Ďalšie zdroje

Pripomíname, že my postupujeme podľa obsahu

- **Základy a princípy MongoDB, Mongo Shell**
- **CRUD, Indexy**
- **Agregácia, Kurzory**
- **Replikácia a Sharding**

Všetky učebnice, prednášky o MongoDB čerpajú, aj my, zo zdrojov oficiálnej stránky MongoDB, ktorá ponúka dve členenia výkladu materiálu:

Členenie na základe manuálu MongoDB 3.2 Manual Getting Started with MongoDB (MongoDB Shell Edition) Getting Started with MongoDB (C#)	Členenie na báze referencií Pojmy a koncepty MongoDB
	Referencie - Prehľad pojmov a konceptov MongoDB. - Komponenty MongoDB balíka - *Operátory* - Databázové príkazy v tvare db.runCommand({...}) - Mongo Shell metódy



The bios Example Collection

Tutorial