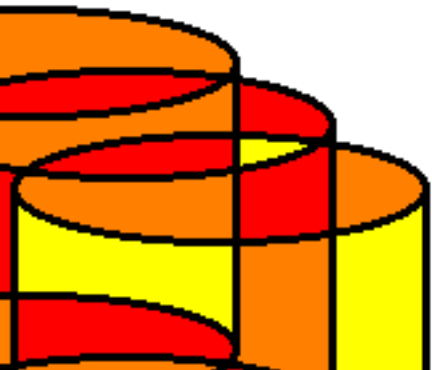


Polustrukturirane JSON baze podataka



- Uz pojam NoSQL (engl. Not only SQL) često se veže podatkovni format JSON (engl. JavaScript Object Notation)
- Postoje mnoge baze podataka koje pohranjuju ovaj format (ili njegov binarni ekvivalent BSON), a podržavaju ga gotovo svi programski jezici u široj upotrebi
- Jedan od najpoznatijih sustava za upravljanje tzv. dokument-baziranim bazama podataka je MongoDB



Priprema

- U slučaju da nije instaliran potrebno je instalirati mongodb
sudo apt install mongodb
- I pokrenuti servis
sudo service mongodb start
- Primjere u nastavku prikazat ćemo u MongoDB konzoli koju pokrećemo s naredbom:

mongo

Upute

- Rezultate isprobavanja (kopiju konzole) pospremite u datoteku ime_prezime.txt

Uvod

- Isprobajte sljedeće naredbe te opišite čemu služe:

db.help()

db.stats()

Uvod

- Na MongoDB sustavu može biti više baza podataka, svaka baza podataka može imati više kolekcija, a svaka kolekcija više dokumenata od kojih je svaki JSON dokument
- Shema podataka je proizvoljna i promjenjiva, tj. za razliku od relacijskih baza podataka nije potrebno unaprijed odrediti shemu

Uvod

- MongoDB ljuska uvijek ima jednu trenutnu bazu podataka s kojom se radi i pohranjena je u varijabli **db**
- Da bi se promijenila trenutna baza podataka koristi se naredba **use**, npr.:

use mojabp

Primjer

- Isprobajte sljedeće naredbe:

use mojabp

db

show dbs

Kreiranje baze podataka

- Kako bismo kreirali našu bazu podataka moramo dodati barem jedan dokument, npr.
`db.proba.insert({ "ključ":"vrijednost" })`
`show dbs`

Brisanje baze podataka

- Za brisanje trenutne baze podataka koristimo metodu `dropDatabase()`

```
db.dropDatabase()
```

```
show dbs
```

Kreiranje kolekcija

- Za kreiranje kolekcija koristi se metoda `createCollection()`

```
db.createCollection( "kolekcija" )  
show collections
```

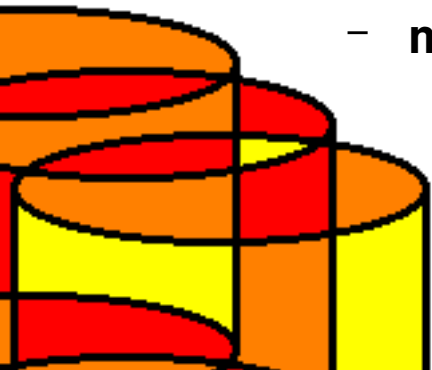
Opcije kreiranja



- Opcionalno se mogu podesiti opcije kreiranja kolekcije, npr.:

```
db.createCollection( "druga", { capped : true,  
size : 6142800, max : 10000 } )
```

- Opcije imaju sljedeće značenje:
 - **capped** (bool) - kolekcija ima fiksiranu maksimalnu veličinu, ako se popuni, automatski se počinju brisati najstariji zapisi. Ako je true obavezno treba navesti size
 - **size** (int) - maksimalna veličina u bajtovima
 - **max** (int) - maksimalan broj dokumenata u kolekciji (također ide uz capped, ali opcionalno)



Brisanje kolekcija

- Za brisanje trenutne kolekcije koristi se njezina metoda `drop()`

`db.druga.drop()`

Tipovi podataka

- Podržani su sljedeći tipovi podataka:
 - **String** – znakovni nizovi (UTF-8)
 - **Integer** – cijeli brojevi
 - **Boolean** – istinitosne vrijednosti
 - **Double** – decimalni brojevi
 - **Min / Max keys** – tip podataka koji se koristi za uporedbu s najmanjim / najvećim BSON elementom.
 - **Arrays** – polja
 - **Timestamp / ctimestamp** - temporalni podaci (trenutak u vremenu)
 - **Object** – ugnijježđeni dokumenti
 - **Null** – striktno nije tip već null vrijednosti
 - **Symbol** – u načelu isto kao i String, koristi se za jezike koji koriste specifični simbolički tip
 - **Date** – temporalni podaci (datum)
 - **Object ID** – identiteti objekata
 - **Binary data** – binarni podaci
 - **Code** – pohrana JavaScript programskog koda
 - **Regular expression** – pohrana pravilnih izraza

Primjer



- Za primjer ćemo kreirati bazu podataka za Weblog, koja će imati dvije kolekcije, korisnici i zapisi:

```
db.dropDatabase()
```

```
use blog
```

```
db.createCollection( "korisnici" )
```

```
db.createCollection( "zapisi" )
```

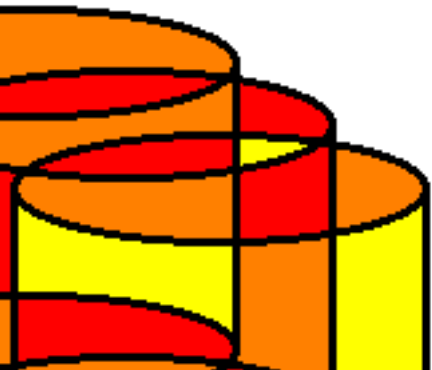


Unos podataka



- Unos podataka obavlja se metodom `insert()` kolekcije u koju se želi unesti jedan ili više dokumenata
- Ako želimo unesti više dokumenata unosimo ih u listi koja se proslijeđuje kao argument metodi `insert()`
- Naredbe za unos u našu bazu podataka nalaze se na adresi:

<https://tinyurl.com/mongodb-unos>



Pretraživanje baze podataka

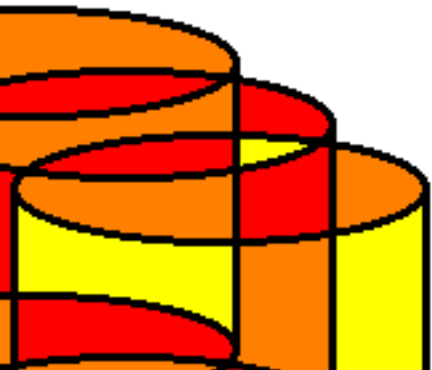


- Koristi se metoda **find()**. Isprobajte sljedeće naredbe:

```
db.korisnici.find()
```

```
db.korisnici.find().pretty()
```

```
db.zapisi.find().pretty()
```



Filtriranje

- Za provjeru jednakosti koriste se JSON dokumenti, npr. upit koji vraća sve korisnike iz Hrvatske:

```
db.korisnici.find( { "država":"Hrvatska" } ).pretty()
```

Filtriranje

- Kako bismo provjerili je li postoji ili ne postoji neki atribut koristimo ključnu riječ `$exists`:

```
db.korisnici.find( { "spol":{ $exists:true } } ).pretty()
```

```
db.korisnici.find( { "spol":{ $exists:false } } ).pretty()
```

Filtriranje

- Za binarne usporedbe koristimo sljedeće ključne riječi:

< \$lt

> \$gt

<= \$lte

>= \$gte

!= \$ne

```
db.zapisi.find( { "lajkova":{ $gt:1000 } } ).pretty()
```

Logički veznici

- Primjer za logičko "i":

```
db.korisnici.find( {  
  $and: [  
    { "spol":{ $exists:false } },  
    { "država":"Hrvatska" }  
  ]  
} ).pretty()
```

Logički veznici

- Primjer za logičko “ili”:

```
db.korisnici.find( {  
  $or:[  
    { "spol":{ $exists:false } },  
    { "država":"Hrvatska" }  
  ]  
} ).pretty()
```

Logički veznici

- Kombinaciji “i” i “ili”

```
db.zapisi.find( {  
  "komentari":{ $exists:true },  
  $or:[  
    { "lajkova":{ $gt:1000 } },  
    { "naslov":"Odojak na lički punjen vrganjima" }  
  ]  
} ).pretty()
```

Projekcija

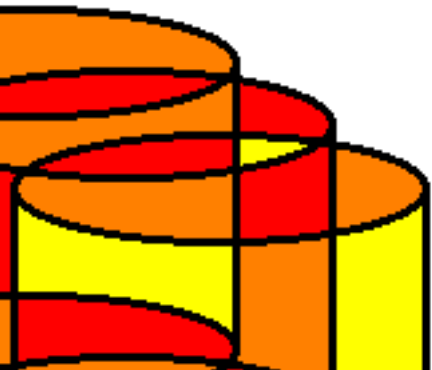


- Kako bismo prikazali samo neke attribute koristimo drugi argument metode find():

```
db.zapisi.find( { "oznake" : "gluten free" }, { "naslov":true,  
"vrijeme":true, "_id":false } ).pretty()
```

- ili kraće

```
db.zapisi.find( { "oznake" : "gluten free" }, { "naslov":1,  
"vrijeme":1, "_id":0 } ).pretty()
```



Ograničavanje izlaza

- Za ograničavanje izlaza koristimo metode `limit()` i `skip()`:

```
db.korisnici.find().limit( 2 ).pretty()
```

```
db.korisnici.find().limit( 2 ).skip( 1 ).pretty()
```


Sortiranje izlaza

- Kako bismo sortirali izlaz koristimo metodu `sort()` navodeći attribute po kojima želimo sortirati pri čemu je vrijednost 1 za silazno, a vrijednost -1 za uzlazno sortiranje:

```
db.zapisi.find( {}, { "naslov":1, "lajkova":1, "_id":0 } ).sort( { "lajkova":-1 } )
```

Pravilni izrazi (RegEx)

- Moguće je pretraživati i putem pravilnih izraza (engl. regular expressions) na sljedeći način:

```
db.zapisi.find( { "naslov":{ "$regex":/0dojak/ } } ).pretty()
```

Zadatak

- Kreirajte novu bazu podataka na proizvoljnu temu te pri tome:
 - Kreirajte barem dvije kolekcije
 - U svaku kolekciju unesite barem 10 dokumenata
 - U svaki dokument unesite barem 5 atributa
 - Kreirajte barem 2 upita s filtriranjem
 - Kreirajte barem 2 upita s logičkim veznicima
 - Kreirajte barem 2 upita s projekcijom (ukupno minimalno 6 upita)
- Sve korištene naredbe pohranite u datoteku ime_prezime.js