

Fakultet tehničkih nauka, DRA, Novi Sad

Predmet:
Organizacija podataka

MongoDB

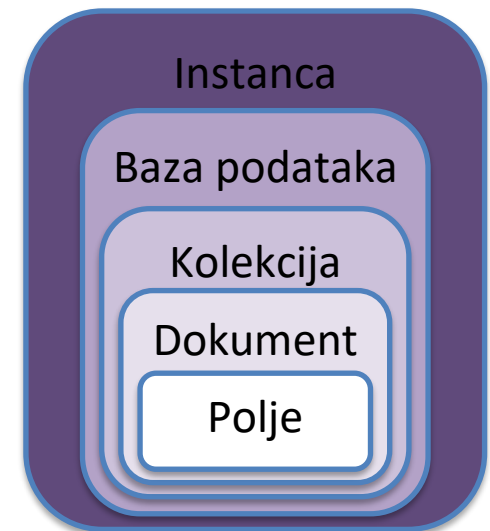
Dokument-orijentisana baza
podataka zasnovana na JSON-u

Uvod

- *MongoDB* predstavlja *open-source* dokument-orijentisanu bazu podataka
- Oblasti primene:
 - Pri radu sa nestrukturiranim i polustrukturiranim podacima
 - Kada nije unapred poznato kakva će biti struktura podataka koji će se skladištiti
 - Link ka dodatnim informacijama:
<https://www.mongodb.com/use-cases>
- Zapise u MongoDB čine dokumenti, čija je struktura predstavljena u formatu parova naziv-vrednost
 - *MongoDB* dokumenti vrlo su slični JSON objektima po formatu

MongoDB – hijerarhija objekata

- Jedna MongoDB instanca
 - Može imati 0 ili više baza podataka
- Jedna baza podataka
 - Može imati 0 ili više kolekcija
- Jedna kolekcija
 - Može imati 0 ili više dokumenata
- Jedan dokument
 - Može imati 0 ili više polja



Šema podataka

- MongoDB ne zahteva postojanje predefinisane šeme podataka
 - Ne postoji jezik za definisanje šeme podataka (engl. *Data Definition Language*)
 - Svaki dokument u okviru jedne kolekcije može imati različit skup polja
- Aplikacijom-vođena šema (engl. *Application-Driven Schema*)
 - Naglasak se stavlja na šablone pristupa podacima (engl. *Data Access Patterns*), odnosno, na način na koji aplikacija koristi podatke
 - Podaci se skladište tako da se podrži njihovo efikasno korišćenje – podaci koji se često koriste zajedno se čuvaju zajedno
 - Iako se šema u MongoDB ne deklarira eksplicitno, aplikacije u najvećem broju slučajeva koriste podatke na način kao da šema postoji

Rad sa bazom u učionici

- Pokretanje instance: *mongod --dbpath /var/lib/mongo*
 - **Za rad kod kuće – nakon instalacije bi trebalo da je instanca već pokrenuta**
- Popunjavanje baze podacima
 1. Skinuti i otpakovati materijale pripremljene za termin
 2. Otvoriti terminal na lokaciji na kojoj se vidi *dump* folder
 3. Pokrenuti naredbu *mongorestore* kako bi se „uvezla“ *video* baza podataka
- Pregled sadržaja baze – koristiti alat *MongoDB Compass*
 - Kreirati konekciju sa podrazumevanim parametrima
- Pokretanje naredbi MongoDB upitnog jezika
 - Koristiti *Mongo* CLI klijentsku aplikaciju (**kod kuće koristiti *Mongosh***)
 - Radi jednostavnijeg rada, definisati *javascript* datoteku koja će sadržati naredbe koje je potrebno pokrenuti
 - U tom slučaju, pokretanje: *mongo --quiet < commands.js*
 - Za *Windows Powershell*: *Get-Content commands.js | mongosh --quiet*

MongoDB upitni jezik

- Eng. *MongoDB Query Language (MQL)*
- Zvanični upitni jezik za rad sa MongoDB sistemima za upravljanje bazama podataka
- Obuhvata naredbe za:
 - Pretragu dokumenata
 - Kreiranje dokumenata
 - Ažuriranje dokumenata
 - Brisanje dokumenata

Pretraga – opšte

- Za pretragu dokumenata koristi se naredba *find*
- Struktura naredbe:

```
db.<naziv_kolekcije>.find([<dokument_za_filtriranje>  
                           [,<dokument_za_projekciju>])
```
- Dokument za filtriranje upita:
 - Specificira uslove kojima se određuje koje zapise u kolekciji bi trebalo odabrati
 - Predstavlja JSON dokument, sledeće strukture:

```
{ <naziv1>:<vrednost1>, <naziv2>: {<operator>: <vrednost2> }, ...}
```
 - Polja dokumenta specificiraju uslove jednakosti i izraze operatora za upite
 - Operatori za upite mogu biti: relacioni, logički, operatori nad elementima, operatori nad nizovima, projekcioni operatori...
 - Navođenje više polja u okviru dokumenta ima efekat primene logičkog operatora „i“
 - Primer:

```
{ "status" : "A", "quantity": { $lt: 30 } }
```

Pretraga – opšte

- Dokument za projekciju:
 - Specificira koja će polja biti vraćena za svaki dokument koji je rezultat upita
 - Predstavlja JSON dokument, sledeće strukture:

```
{ <naziv1>:<vrednost1>, <naziv2>:<vrednost2> }
```
 - <vrednost> može biti:
 - **0**, odnosno, **false** → ukoliko je polje potrebno isključiti iz rezultata upita (preostala polja će biti uključena u rezultat) ili
 - **1**, odnosno, **true** → ukoliko je polje potrebno uključiti u rezultat (preostala polja će biti isključena iz rezultata)
 - Nije dozvoljeno istovremeno korišćenje true i false vrednosti u okviru istog dokumenta
 - Izuzetak je `_id` polje, koje se uvek mora eksplicitno isključiti
 - Primeri:

```
{ "name" : 1, "email" : 1 }  
{ "date" : 0 }  
{ "name" : 1, "_id" : 0 }
```

Pretraga – opšte

- Primeri jednostavnih naredbi:
 - Izlistavanje sadržaja kolekcije:
`db.movieDetails.find()`
 - Izlistavanje naziva filmova iz 2015. godine:
`db.movieDetails.find({"year": 2015}, {"title": 1, "_id": 0})`
 - Izlistavanje filmova iz 2015. godine, sa rangiranjem PG-13:
`db.movieDetails.find({"year": 2015, "rated": "PG-13"})`
 - Prikaz broja dokumenata koji ispunjava zadati kriterijum pretrage:
`db.movieDetails.find({"year": 2015}).count()`

Pretraga – zadaci 1

- Prikazati podatke o filmu „2001: A Space Odyssey“ iz kolekcije movieDetails:

```
db.movieDetails.find({ "title": "2001: A Space Odyssey"})
```
- Prikazati naziv, godinu snimanja i žanrove za sve filmove rangirane kao „R“:

```
db.movieDetails.find(  
  { "rated": "R"},  
  { "title": 1, "year": 1, "genres":1, "_id": 0})
```
- Prikazati koliko filmova snimljenih 2015. godine ima rejting „PG-13“:

```
db.movieDetails.find(  
  { "year": 2015, "rated": "PG-13"}).count()
```

Pretraga – ugnježdeni dokumenti

- Pretraga po poljima ugnježdenih dokumenata:
 - Pristup poljima ugnježdenih dokumenata vrši se pomoću *dot* notacije:
 - Ovakav pristup dokumentima važi za bilo koji nivo dubine ugnježdavanja
 - Dot notacija se može koristiti i u okviru dokumenata za projekciju
 - Primeri:

```
db.movieDetails.find({"tomato.meter": 100}).count()
```

```
db.movieDetails.find({"year": 1995, "tomato.meter": 100},  
{"year": 1, "tomato.meter": 1})
```

Pretraga – zadaci 2

- Prikazati naziv, trajanje i polje awards za sve filmove iz 2014. godine koji nisu osvojili nijednu nagradu:

```
db.movieDetails.find(  
  { "year": 2014, "awards.wins": 0},  
  { "title": 1, "runtime": 1, "awards":1, "_id": 0})
```

- Prikazati naziv i polje rating iz imdb za filmove sa imdb ocenom od 8.5:

```
db.movieDetails.find(  
  { "imdb.rating": 8.5},  
  { "title": 1, "imdb.rating": 1, "_id": 0})
```

Pretraga – nizovi

- Pretraga po sadržaju nizova – četiri načina:
 - pretraga po sadržaju celog niza,
 - pretraga po bilo kojem članu niza
 - pretraga bazirana na poziciji člana niza i
 - napredna pretraga nizova korišćenjem operatora (kasnije)

Pretraga – nizovi

- Pretraga po sadržaju celog niza:
 - Podrazumeva da se pretraga vrši po sadržaju konkretnog niza, **pri čemu se pozicije elemenata u nizu uzimaju u obzir**
 - Primeri:

```
db.movieDetails.find({"writers": ["Ethan Coen", "Joel Coen"]})  
db.movieDetails.find({"writers": ["Joel Coen", "Ethan Coen"]})
```
- Pretraga po bilo kom članu niza:
 - Podrazumeva da niz sadrži navedeni element, **pri čemu pozicija navedenog elementa u nizu nije bitna**
 - Sintaksa za pretragu je ista kao i kod skalarnih vrednosti polja
 - Napomena: obratiti pažnju na to da bi korišćenje uglastih zagrada vratilo neželjene rezultate (moguć izvor grešaka)
 - Primeri:

```
db.movieDetails.find({"writers": "Joel Coen"})  
db.movieDetails.find({"writers": ["Joel Coen"]}) // pokušati
```

Pretraga – nizovi

- Pretraga bazirana na poziciji člana niza:
 - Podrazumeva da niz sadrži navedeni element, **pri čemu se on nalazi navedenoj poziciji u nizu**
 - Korisno kada su elementi u nizu pozicionirani po prioritetu
 - Koristi se *dot* notacija – iza imena niza stavlja se tačka nakon koje se navodi pozicija (indeks) traženog elementa
 - Indeksiranje počinje od nule
 - Primer:

```
db.movieDetails.find({"actors.0": "Tom Hanks"})
```

Pretraga – zadaci 3

- Prikazati broj filmova u čijem pisanju je učestvovao Sergio Leone:

```
db.movieDetails.find({ "writers": "Sergio Leone"},  
  { "title": 1, "writers": 1, "_id": 0}).count()
```
- Prikazati filmove najvećim delom snimljene u Velikoj Britaniji (UK). Imdb i tomato podatke ne treba prikazati korisniku:

```
db.movieDetails.find({ "countries.0": "UK" },  
  { "imdb": 0, "tomato": 0})
```
- Prikazati podatke o filmovima čiji je jedini pisac George Lucas:

```
db.movieDetails.find({ "writers": ["George Lucas"] }, {})
```

Pretraga – relacioni operatori

- Opšti oblik sintakse:

`{<naziv>:{<operator>: <vrednost>[, <operator>:<vrednost>]}}`

- Podržani su sledeći relacioni operatori:
 - **\$eq** – ekvivalent operatoru ==,
 - **\$ne** – ekvivalent operatoru !=,
 - **\$gt** – ekvivalent operatoru >,
 - **\$gte** – ekvivalent operatoru >=,
 - **\$lt** – ekvivalent operatoru <,
 - **\$lte** – ekvivalent operatoru <=,
 - **\$in** – navodi se lista poželjnih vrednosti za polje i
 - **\$nin** – navodi se lista nepoželjnih vrednosti za polje

Pretraga – relacioni operatori

- Primeri jednostavnih naredbi:
 - Izlistavanje filmova koji imaju trajanje duže od 90 i manje ili jednako 120 minuta:
`db.movieDetails.find({"runtime": {$gt: 90, $lte: 120}})`
 - Prikaz filmova koje je režirao George Lucas.
`db.movieDetails.find({"director": {$eq: "George Lucas"}})`
 - Izlistavanje filmova koji nisu rangirani kao „PG“ ili „PG-13“
`db.getCollection('movieDetails').find({"rated": {$nin: ["PG", "PG-13"]}})`

Pretraga – zadaci 4

- Prikazati nazive i režisere filmova čiji je imdb rejting manji ili jednak 8.5, pri čemu je bitno da broj glasova za film na imdb bude veći od 400 hiljada:

```
db.movieDetails.find({ "imdb.rating": { $lte: 8.5}, "imdb.votes": {  
  $gt: 400000 } },  
  { "title": 1, "director": 1 })
```
- Prikazati filmove nastale nakon 1980. godine, a koji su osvojili makar pet nagrada:

```
db.movieDetails.find({ "year": { $gt: 1980},  
  "awards.wins": { $gte: 5} }, {})
```
- Prikazati nazive, zemlje i nagrade filmova koji su snimani ili u USA, UK ili Francuskoj. Uzeti u obzir i filmove koji su snimani u više zemalja:

```
db.movieDetails.find({ "countries": { $in: ["US", "UK", "France"]} },  
  { "title": 1, "countries": 1, "awards": 1 })
```

Pretraga – operatori za nizove

- Podržani sledeći operatori za rad nad nizovima:
 - **\$all** – vraća dokumente koji za navedeno polje sadrže sve specificirane elemente
 - Redosled elemenata nije bitan, kao ni to da li niz, pored specificiranih, sadrži i druge elemente

```
db.movieDetails.find({"writers":  
                      { $all: ["Joel Coen", "Ethan Coen"] }})
```

- **\$size** – vraća dokumente sa specificiranim brojem elemenata
 - Nije podržan rad sa relacionim operatorima
- ```
db.movieDetails.find({"countries": { $size: 1 }})
```

# Pretraga – zadaci 5

- Prikazati filmove koji su snimani i u USA i u UK:

```
db.movieDetails.find({ "countries": { $all: ["UK", "USA"]} })
```

- Prikazati filmove koje je napisala tačno jedna osoba:

```
db.movieDetails.find({ "writers": { $size: 1} }, {})
```

# Pretraga – zadaci za vežbu 1

- Prikazati naziv filma, glumačku postavu i žanr za filmove u kojima glumi Tom Hanks, a da pritom nisu u pitanju animirani filmovi.

```
db.movieDetails.find({
 "actors": "Tom Hanks",
 "genres": {$nin: ["Animation"]}},
 {"genres": 1, "title": 1, "actors": 1})
```

- Napisati upit koji će vratiti broj filmova u kojima je glavni glumac Johnny Depp, i koji imaju između 100.000 i 200.000 korisničkih ocena na Rotten Tomatoes sajtu.:

```
db.movieDetails.find(
 {"actors.0": "Johnny Depp",
 $and: [
 {"tomato.userReviews": {$gte: 100000}},
 {"tomato.userReviews": {$lte: 200000}}]
 }).count()
```

# Pretraga – zadaci za vežbu 2

- Prikazati sve filmove u kojima glumi Robert De Niro, snimljene ili 1984. ili 1987. godine, a da je pritom vreme trajanja filma veće od 200 minuta. U prikazu filma treba da se nađu informacije o nazivu filma, glumačkoj postavi i imdb rejtingu filma.

```
db.movieDetails.find({"actors": "Robert De Niro",
 $or: [{"year": 1984}, {"year": 1987}], "runtime": {$gt: 200}},
 {"title": 1, "actors": 1, "imdb.rating": 1})
```

- Napisati upit koji će vratiti filmove koji nisu osvojili nijednu nagradu, a da su pritom snimljeni, između ostalog, i u USA i u UK, pri čemu je bitno da je film snimljen u tačno 3 države. Pri prikazu rezultata upita potrebno je za svaki film prikazati samo naslov, zemlje u kojima je sniman, glumačku postavu i broj nominacija.

```
db.getCollection('movieDetails').find(
 $and: [
 {"countries" : {$size: 3}},
 {"countries": {$all: ["USA", "UK"]}},
 {"awards.wins": 0}],
 {"title": 1, "countries": 1, "actors": 1,
 "awards.nominations": 1, "_id": 0})
```

# Kreiranje dokumenata

- Dva načina za kreiranje dokumenata u okviru MongoDB kolekcije:
  - korišćenjem naredbe *insertOne*,
  - korišćenjem naredbe *insertMany* i
- Ukoliko kolekcija u koju pokušamo da unesemo dokument ne postoji, biće kreirana
- Ukoliko se vrednost polja *\_id* ne zada, biće automatski dodeljena
- Strukture naredbi:

```
db.<naziv_kolekcije>.insertOne(<dokument>)
db.<naziv_kolekcije>.insertMany([<lista_dokumenata>])
```

# Kreiranje dokumenata

- Primer korišćenja *insertOne*:

```
db.movies.insertOne({
 "title": "Avengers - Infinity War",
 "year": 2018,
 "type": "movie"
})
```

- Primer korišćenja *insertMany* uz ručno postavljanje *\_id* polja:

```
db.movies.insertMany([
 {"title": "Deadpool 2", "year": 2018 },
 {"title": "Venom", "year": 2018, "_id": "Vnm_id" },
 {"title": "Incredibles 2", "year": 2018 }
])
```

# Kreiranje dokumenata - zadaci

- U movies kolekciju uneti podatke o filmu Black Panther iz 2018. godine. Pored naziva filma i godine prikazivanja, dodati polje watched boolean tipa, a za vrednost staviti true ukoliko ste pogledali film, odnosno false ukoliko niste.

```
db.movies.insertOne({
 "title": "Black Panther",
 "year": 2018,
 "watched": true
})
```

- Primer korišćenja *insertMany* uz ručno postavljanje *\_id* polja:

```
db.movies.insertMany([
 {"title": "Dunkirk", "year": 2017 },
 {"title": "Solo", "year": 2018 },
 {"title": "Logan", "year": 2017 }
])
```

# Ažuriranje dokumenata

- Dve naredbe za ažuriranje dokumenata u okviru MongoDB kolekcije:
  - korišćenjem naredbe *updateOne*,
  - korišćenjem naredbe *updateMany* i
- Strukture naredbi:

```
db.<naziv_kolekcije>.<naziv_naredbe>(
 <dokument_za_filtriranje_upita>,
 <dokument_za_specifikaciju_ažuriranja>)
```
- Ukoliko je pri ažuriranju potrebno odabrati sve dokumente, dokument za filtriranje bi trebalo da bude prazan

# Ažuriranje dokumenata

- Dokument za specifikaciju ažuriranja:
  - Predstavlja dokument koji specificira promene koje je potrebno primeniti nad odabranim dokumentom/dokumentima
  - Predstavlja JSON dokument sledeće strukture

```
{
 <operator1>: { <naziv1>: <vrednost1>, ... },
 <operator2>: { <naziv2>: <vrednost2>, ... },
 ...
}
```
  - Operatori za ažuriranje mogu biti operatori za rad nad skalarnim poljima i operatori za rad nad nizovima
  - Ukoliko se pri specificiranju dokumenta ne navede operator, već samo par <field>: <value>, desiće se izuzetak!

# Ažuriranje dokumenata

- Operatori za rad nad skalarnim poljima:
  - **\$set** – vrednost polja će biti postavljena na specificiranu vrednost,
  - **\$unset** – odabrano polje će biti uklonjeno iz dokumenta,
  - **\$rename** – odabrano polje će biti preimenovano.
- Operatori za rad sa nizovima
  - **\$push** – specificirana vrednost će biti dodata u niz,
  - **\$addToSet** – specificirana vrednost će biti dodata u niz samo ako u nizu ne postoji element sa istom vrednošću,
  - **\$pop** – prvi (vrednost -1) ili poslednji (vrednost 1) element niza će biti uklonjen ,
  - **\$pull** – iz niza će biti uklonjeni svi elementi koji ispunjavaju specificirani kriterijum (mogu se koristiti i relacioni operatori),
  - **\$pullAll** – iz odabranog niza uklanja vrednosti koje se nalaze u zadatom nizu.

# Ažuriranje dokumenata

- Primeri upotrebe operatora nad skalarnim poljima:
  - Ažurirati vreme trajanja filma „The Amazing Spider-Man” iz 2012. godine na vrednost 145.
  - U svim dokumentima promeniti naziv polja metacritic u meta-critic.

```
db.movieDetails.updateOne(
```

```
 {"title": "The Amazing Spider-Man", "year": 2012 },
```

```
 {$set: {"runtime": 145}})
```

```
db.movieDetails.updateMany({},
```

```
 {$rename: {"metacritic": "meta-critic"}})
```

# Ažuriranje dokumenata

- Primeri upotrebe operatora za rad nad nizovima:
  - U spisak glumaca koji su glumili u filmu „*From Paris with Love*“ dodati glumicu Amber Rose Revah.  

```
db.movieDetails.updateOne({"title": "From Paris with Love"},
 {$push: {"actors": "Amber Rose Revah"}})
```
  - Ukloniti prvu osobu iz liste glumaca filma „*From Paris with Love*“.  

```
db.movieDetails.updateOne({"title": "From Paris with Love"},
 {$pop: {"actors": -1}})
```

# Ažuriranje dokumenata – zadaci 1

- Ažurirati filmove kojima je polje rated jednako null tako da im to polje bude jednako NOT RATED.

```
db.movies.updateMany({ "rated": null },
 {$set: {"rated": "NOT RATED"}})
```

- Izbrisati polje consensus, dobijeno sa sajta Rotten Tomatoes, iz dokumenata o filmovima koji imaju manje od 10.000 korisničkih ocena na istom sajtu.

```
db.movies.updateMany({"tomato.userReviews": { $lt:
10000}}, { $unset: { "consensus": "" }})
```

# Ažuriranje dokumenata – zadaci 2

- Iz niza glumaca obrisati poslednjeg glumca sa spiska za svaki film za koji na imdb ima manje od 10.000 glasova, a da je pritom film sniman samo u USA.

```
db.movies.updateMany({"imdb.votes": { $lt: 10000},
"countries": ["USA"]}, {$pop: {"actors": 1}})
```

- Ukloniti glumica Eric Bana iz spiska glumaca za film „*Star Trek*“ .

```
db.movieDetails.updateOne({"title": "Star Trek"},
{$pull: {"actors": "Eric Bana"}})
```

# Brisanje dokumenata

- Dve naredbe za brisanje dokumenata u okviru MongoDB kolekcije:
  - korišćenjem naredbe *deleteOne*,
  - korišćenjem naredbe *deleteMany* i
- Strukture naredbi:

```
db.<naziv_kolekcije>.<naziv_naredbe>(
 <dokument_za_filtriranje_upita>)
```
- Ukoliko je pri brisanju potrebno odabrati sve dokumente, dokument za filtriranje bi trebalo da bude prazan

# Brisanje dokumenata

- Primeri upotrebe:
  - Obrisati jedan nasumično odabran film iz 2018. godine.  
`db.movies.deleteOne(  
 { "year": 2018 }  
)`
  - Obrisati sve filmove snimljene u 2018. godini.  
`db.movies.deleteMany(  
 { "year": 2018 }  
)`

# Brisanje dokumenata – zadaci

- Obrisati film sa nazivom Black Panther

```
db.movies.deleteOne({
 "title": "Black Panther",
})
```