

# Sistemi baza podataka

---

*MongoDB – CRUD*

# Sadržaj

- Pretraga dokumenata
- Kreiranje dokumenata
- Ažuriranje dokumenata
- Brisanje dokumenata

# Pretraga dokumenata

---

# Pretraga dokumenata

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

```
db.<naziv_kolekcije>.find(  
    [<dokument_za_filtriranje_upita>  
    [,<dokument_za_projekciju>])
```

- Dokument za filtriranje upita specificira uslove kojima se određuje koje zapise u kolekciji bi trebalo odabrati **pri operacijama čitanja, ažuriranja ili brisanja**
  - Predstavlja *JSON* dokument, sledeće strukture:

```
{  
    <field1>: <value1>,  
    <field2>: { <operator>: <value> },  
    ...  
}
```

# Dokument za filtriranje upita

- Navođenjem polja ovog dokumenta, specificiraju se uslovi jednakosti i izrazi operatora za upite
  - Operatori za upite mogu biti: relacioni operatori, logički operatori, operatori nad elementima, operatori nad nizovima, projekcioni operatori, operatori za evaluaciju, *geospatial* operatori...
- **Navođenje više polja u okviru dokumenta za filtriranje upita ima efekat primene logičkog operatora „i“**
- Primer dokumenta za filtriranje upita:

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

# Dokument za projekciju

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

```
{  
  <field1>: <value>,  
  <field2>: <value>,  
  ...}
```

- <value> 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

# Dokument za projekciju

- Primeri dokumenata za projekciju:

```
{ "name" : 1, "email" : 1 }
```

```
{ "date" : 0 }
```

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

## Pretraga dokumenata

- Izlistavanje sadržaja kolekcije:

```
db.movieDetails.find()
```

```
db.movieDetails.find({})
```

- Izlistavanje naziva filmova iz 2015. godine:

```
db.movieDetails.find({"year": 2015}, {"title": 1})
```

```
db.movieDetails.find({"year": 2015}, {"title": 1, "_id": 0})
```

# Pretraga dokumenata

- Pretraga po zadatoj vrednosti polja `_id`:

```
db.movieDetails.find({"_id": ObjectId("5692a14f24de1e0ce2dfcf60")})
```

- Kada se koristi drajver nekog programskog jezika, koristi se samo string
- 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 dokumenata

- Prikazati podatke o filmu „2001: A Space Odyssey“ iz kolekcije *movieDetails*.
- Prikazati naziv, godinu snimanja i žanrove za sve filmove rangirane kao „R“.
- Prikazati film čija je vrednost polja `_id` „569190cf24de1e0ce2dfcd71“. Iz prikaza isključiti podatke o nagradama.
- Prikazati koliko filmova snimljenih 2015. godine ima rejting „PG-13“.

# Pretraga po poljima ugnježenih dokumenata

- Pristup poljima ugnježenih 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 po poljima ugnježenih dokumenata

- Prikazati naziv, trajanje i polje *awards* za sve filmove iz 2014. godine koji nisu osvojili nijednu nagradu.
- Prikazati naziv i polje rating iz *imdb* za filmove sa *imdb* ocenom od 8.5.

# Pretraga po sadržaju nizova

- Pretraga po sadržaju nizova može se vršiti na 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

# Pretraga po sadržaju celog niza

- 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 kojem članu niza

- Pretraga po bilo kojem 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).
- Primer:

```
db.movieDetails.find({"writers": "Joel Coen"})
```

- Pokušati:

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

# Pretraga bazirana na poziciji člana niza

- 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 po sadržaju nizova

- Prikazati broj filmova u čijem pisanju je učestvovao *Sergio Leone*.
- Prikazati filmove najvećim delom snimljene u Velikoj Britaniji (*UK*). *Imdb* i *tomato* podatke ne treba prikazati korisniku.
- Prikazati podatke o filmovima čiji je jedini pisac *George Lucas*.

# Relacioni operatori

- Opšti oblik sintakse:

```
{ <field>: { <operator>: <value>[, <operator>: <value> ] } }
```

- U okviru *MongoDB* upitnog jezika postoje 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

## Relacioni operatori

- 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"]}})
```

# Relacioni operatori

- 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.
- Prikazati filmove nastale nakon 1980. godine, a koji su osvojili makar pet nagrada.
- 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.

# Logički operatori

- Opšti oblik sintakse:

```
{ operator: [ { <expression1> }, { <expression2> } , ... , {<expressionN> } ] }
```

- U okviru *MongoDB* upitnog jezika postoje sledeći logički operatori:
  - **\$and** – logičko „i“,
  - **\$or** – logičko „ili“,
  - **\$not** – logičko „ne“ – sintaksa malo drugačija,
  - **\$nor** – logičko „nili“
- Logičko „i“ je neophodno koristiti u situacijama kada se više puta postavlja upit nad istim poljem
  - Dosad korišćenom sintaksom to nije moguće jer naziv polja u dokumentu mora biti jedinstven

# Logički operatori

- Izlistavanje filmova koji ili imaju *imdb* rejting veći ili jednak 8 ili *tomato* metriku veću od 95:

```
db.movieDetails.find({$or: [  
  {"imdb.rating": {$gte: 8.0}},  
  {"tomato.meter": {$gt: 95}}  
]})
```

- Izlistati imena filmova čiji *imdb* rejting je veći od 8, a manji od 8.6:

```
db.movieDetails.find({$and: [  
  {"imdb.rating": {$gt: 8.0}},  
  {"imdb.rating": {$lt: 8.6}}]}, {"title": 1})
```

- Izlistati filmove koji nisu snimljeni 2014. godine

```
db.movieDetails.find({ "year": {$not: { $eq: 2014}}})
```

# Logički operatori

- Prikazati naziv, godinu prikazivanja i nagrade filmova koji su osvojili između 5 i 10 nagrada na festivalima.
- Prikazati filmove koji imaju *imdb* ocenu veću od 8 ili imaju više od sto hiljada *imdb* glasova.
- Prikazati filmove koji nisu rangirani sa *R*, a imaju više od pet osvojenih nagrada.

# Operator `$exists`

- Omogućava pretragu dokumenata po tome da li sadrže neko polje ili ne
- Sintaksa:

```
{ field: { $exists: <boolean> } }
```

- Ukoliko se za `<boolean>` stavi vrednost **true**, vraća sve dokumente koji sadrže navedeno polje, čak iako je vrednost tog polja *null*
- Ukoliko se za `<boolean>` stavi vrednost **false**, vraća dokumente koji ne sadrže navedeno polje
- Primer:

```
db.movieDetails.find({"tomato": {$exists: false}})
```

# Operator \$exists

- Prikazati broj filmova za koje postoje *metacritic* podaci.

# Operatori za upite nad nizovima

- U okviru MongoDB upitnog jezika postoje 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 }})
```

# Operatori za upite nad nizovima

- Prikazati filmove koji su snimani i u *USA* i u *UK*.
- Prikazati filmove koje je napisala tačno jedna osoba.

## Zadaci za vežbu

- 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})
```

## Zadaci za vežbu

- 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()
```

## Zadaci za vežbu

- 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  
})
```

## Zadaci za vežbu

- 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

---

# Kreiranje dokumenata

- Postoje tri načina za kreiranje dokumenata u okviru *MongoDB* kolekcije:
  - korišćenjem naredbe **insertOne**,
  - korišćenjem naredbe **insertMany** i
  - korišćenjem naredbe **update**
- Ukoliko kolekcija u koju pokušamo da unesemo dokument ne postoji, biće kreirana
- Strukture naredbi:

```
db.<naziv_kolekcije>.insertOne(<dokument>)
```

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

# insertOne

- Primer korišćenja:

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

- Odziv sistema:

```
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5aec3f5a5362a6f42ff499e1")
}
```

# insertOne

- Primer korišćenja uz ručno postavljanje vrednosti `_id` polja:

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

- Odziv sistema:

```
{ "acknowledged" : true, "insertedId" : "AvgIW" }
```

# insertMany

- Primer korišćenja:

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

- Odziv sistema:

```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5aed40196543a5b58268c758"),
    ObjectId("5aed40196543a5b58268c759"),
    ObjectId("5aed40196543a5b58268c75a")
  ]
}
```

## Zadaci za vežbu

- 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.
- U *movies* kolekciju uneti podatke o sledećim filmovima:
  - *Dunkirk*, 2017
  - *Solo*, 2018
  - *Logan*, 2017

# Ažuriranje dokumenata

---

# Ažuriranje dokumenata

- Postoje četiri načina za ažuriranje dokumenata u okviru *MongoDB* kolekcije:
  - korišćenjem naredbe **updateOne**,
  - korišćenjem naredbe **updateMany**,
  - korišćenjem naredbe **replaceOne** i
  - korišćenjem naredbe **save**

# Ažuriranje dokumenata

- **updateOne** i **updateMany** naredbe imaju sledeću strukturu:

```
db.<naziv_kolekcije>.<naziv_naredbe>(
  <dokument_za_filtriranje_upita>,
  <dokument_za_specifikaciju_ažuriranja>
  [,<dokument_za_specifikaciju_opcija>])
```

- Ukoliko je pri ažuriranju potrebno odabrati sve dokumente, dokument za filtriranje upita bi trebalo da bude prazan – { }

# Dokument za specifikaciju ažuriranja

- Predstavlja dokument koji specificira promene koje je potrebno primeniti nad odabranim dokumentom/dokumentima
- Predstavlja *JSON* dokument sledeće strukture:

```
{  
  <operator1>: { <field1>: <value1>, ... },  
  <operator2>: { <field2>: <value2>, ... },  
  ...  
}
```

- 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!

# Operatori za rad nad skalarnim poljima

- U okviru dokumenta za specifikaciju ažuriranja mogu se naći sledeći 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,
  - **\$inc** – odabrano numeričko polje će biti uvećano za specificiranu vrednost,
  - **\$mul** – odabrano numeričko polje će biti pomnoženo specificiranom vrednošću,
  - **\$min** – polje će biti ažurirano ukoliko je trenutna vrednost manja od specificirane,
  - **\$max** – polje će biti ažurirano ukoliko je trenutna vrednost manja od specificirane.

# Operatori za rad sa skalarnim poljima

- Ažurirati vreme trajanja filma „*The Amazing Spider-Man*” iz 2012. godine na vrednost 145.

```
db.movieDetails.updateOne({"title": "The Amazing Spider-Man", "year": 2012 },  
                           {$set: {"runtime": 145}})
```

- U svim dokumentima promeniti naziv polja *metacritic* u *meta-critic*.

```
db.movieDetails.updateMany({ }, {$rename: {"metacritic": "meta-critic"}})
```

- Iz svih dokumenata obrisati polje *type*.

```
db.movieDetails.updateMany({}, {$unset: {"type": ""}})
```

# Operatori za rad sa skalarnim poljima

- Uvećati vreme trajanja filma „The Amazing Spider-Man” iz 2012. godine za 13 minuta.

```
db.movieDetails.updateOne({"title": "The Amazing Spider-Man", "year": 2012 },  
  {$inc: {"runtime": 13}})
```

- Ukoliko je vreme trajanja filma “The Amazing Spider-Man” manje od 185, postaviti ga na tu vrednost.

```
db.movieDetails.updateOne({"title": "The Amazing Spider-Man"},  
  {$max: {"runtime": 185}})
```

# Operatori za rad sa skalarnim poljima

- Ažurirati filmove kojima je polje *rated* jednako *null* tako da im to polje bude jednako *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.
- Ukoliko film „*The Woman in Black 2: Angel of Death*“ ima *imdb* ocenu veću od 4.5, dodeliti mu tu ocenu.

# Operatori za rad sa nizovima

- U okviru dokumenta za specifikaciju ažuriranja mogu se naći sledeći operatori za rad nad 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.

# Operatori za rad sa 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}})
```

- Ukoliko glumica *Amber Rose Revah* ne postoji u spisku glumaca za film „*From Paris with Love*“, dodati je.

```
db.movieDetails.updateOne({"title": "From Paris with Love"},  
  {$addToSet: {"actors": "Amber Rose Revah"}})
```

# Operatori za rad sa nizovima

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

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

- Iz spiska glumaca filma „*Butch Cassidy and the Sundance Kid*“ ukloniti glumce *Katharine Ross*, *Strother Martin* i *Butch Cassidy*.

```
db.movieDetails.updateOne({"title": "Butch Cassidy and the Sundance Kid"},  
  {$pullAll: {"actors": ["Katharine Ross", "Strother Martin", "Butch Cassidy"]}})
```

# Operatori za rad sa nizovima

- 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*.
- Iz svih filmova kreiranih, između ostalog, i u *USA*, a koji na *imdb* imaju ocenu između 1 i 4, obrisati vrednost *USA* iz polja *countries*.
- Svaki film čiji je spisak zemalja u kom je sniman prazan niz ažurirati tako da se u niz doda *USA*.

# Dokument za specifikaciju opcija pri ažuriranju

- Pri operaciji ažuriranja dokumenta moguće je specificirati različite opcije, a najčešće korišćena opcija jeste **upsert**
  - **upsert** – boolean vrednost; kada se vrednost ovog polja postavi na **true**, dokument će prilikom operacije ažuriranja, ukoliko ne postoji, biti dodat.

# Ažuriranje dokumenata

- **replaceOne** – naredba pomoću koje je moguće jedan dokument, odabran korišćenjem dokumenta za filtriranje upita, zameniti drugim dokumentom
- Struktura naredbe:

```
db.<naziv_kolekcije>.replaceOne(  
  <dokument_za_filtriranje_upita>,  
  <dokument_sa_novim_sadržajem>  
  [,<dokument_za_specifikaciju_opcija>])
```

# Ažuriranje dokumenata

- **save** – naredba sa sledećom strukturom:

```
db.<naziv_kolekcije>.save(  
  <dokument>  
  [,<dokument_za_specifikaciju_opcija>])
```

- Ukoliko dokument koji se prosledi **sadrži** polje `_id`, dokument sa tim `_id` poljem će biti zamenjen prosleđenim dokumentom
- Ukoliko dokument koji se prosledi **ne sadrži** polje `_id`, biće unesen novi dokument
- Pri korišćenju naredbe, `upsert` opcija je podrazumevana

# Brisanje dokumenata

---

# Brisanje dokumenata

- Postoje dva načina za brisanje dokumenata u okviru *MongoDB* kolekcije:
  - Korišćenjem naredbe **deleteOne** i
  - Korišćenjem naredbe **deleteMany**
- Strukture naredbi:

```
db.<naziv_kolekcije>.deleteOne(<dokument_za_filtriranje_upita>)
```

```
db.<naziv_kolekcije>.deleteMany(<dokument_za_filtriranje_upita>)
```

- Za brisanje svih dokumenata pri korišćenju **deleteMany** naredbe, dokument za filtriranje upita **treba da bude prazan**

# DeleteOne

- Primer korišćenja:

```
db.movies.deleteOne(  
  { "year": 2018 }  
)
```

- Odziv sistema:

```
{  
  "acknowledged" : true,  
  "deletedCount" : 1.0  
}
```

# DeleteMany

- Primer korišćenja:

```
db.movies.deleteMany(  
  { "year": 2018 }  
)
```

- Odziv sistema:

```
{  
  "acknowledged" : true,  
  "deletedCount" : 6.0  
}
```

# Kraj!

Hvala na pažnji!