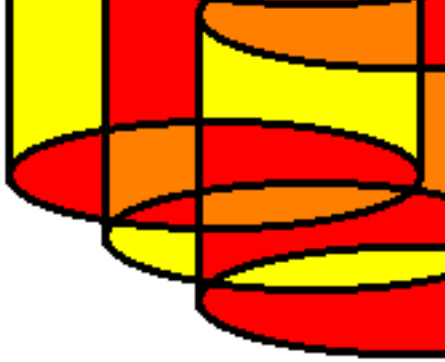
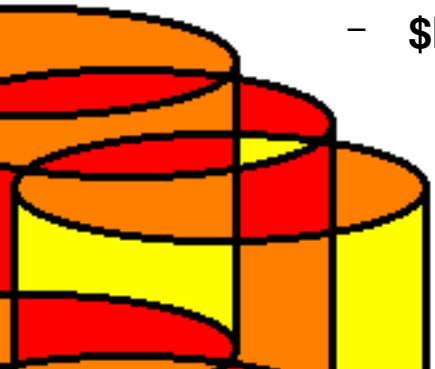
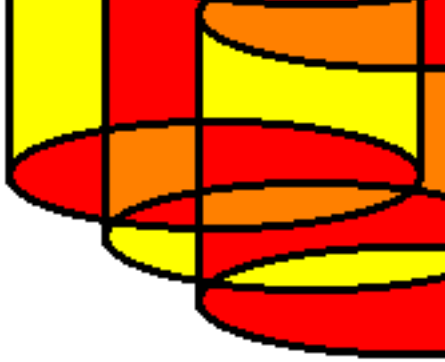
Agregacije



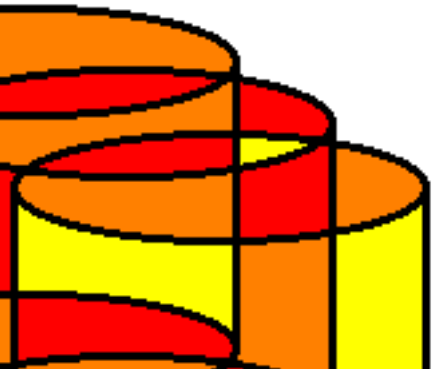
- Koristi se metoda **aggregate()**. Podržane su sljedeće zbirne funkcije:
  - **$sum** - zbraja grupu vrijednosti (može se koristiti i kao ekvivalent count() funkciji za prebrojavanje)
  - **$avg** - računa prosjek grupe vrijednosti
  - **$min** - pronalazi minimalnu vrijednost
  - **$max** - pronalazi maksimalnu vrijednosti
  - **$push** - "gura" vrijednosti u polje koje će biti izlaz iz agregacije
  - **$addToSet** - ubazuje vrijednosti u polje ali eliminira duplikate
  - **$first** - pronalazi prvi element u grupi (u pravilu se koristi s nekom vrstom sortiranja)
  - **$last** - pronalazi zadnji element u grupi (isto kao i prethodni)





Upute

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

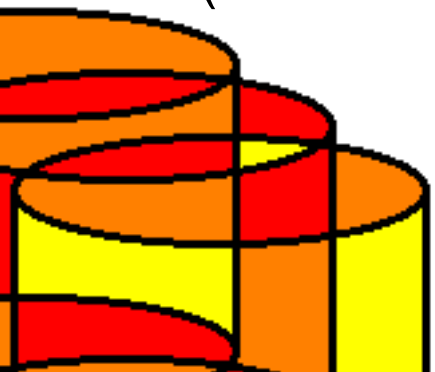


Primjeri

- Upit koji vraća broj zapisa po autoru:

```
db.zapisi.aggregate( [ {
   $group : {
      _id : "$autor",
     "broj_zapisa" :
       { $sum:1 }
   }
} ] )
```

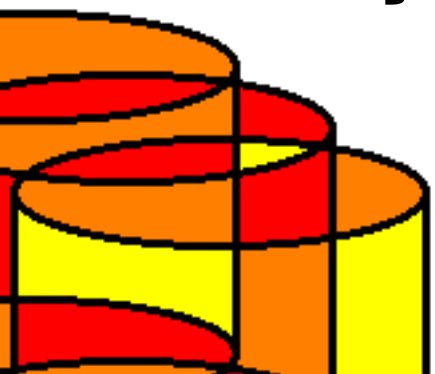
- Grupira se po autoru, a za svaku stavku u grupi sumira se broj 1 (konstanta) zbog čega se $sum ponaša kao count().



Primjeri

- Upit koji vraća broj lajkova po autoru:

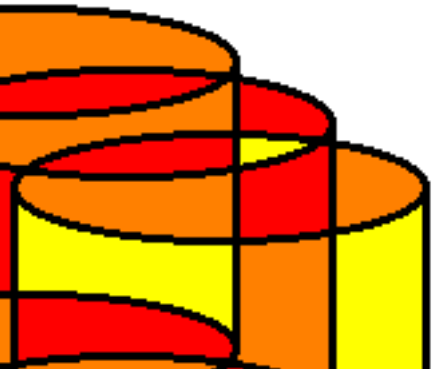
```
db.zapisi.aggregate( [ {
   $group : {
      _id : "$autor",
      "broj_lajkova" : {
        $sum:"$lajkova"
      }
   }
} ] )
```



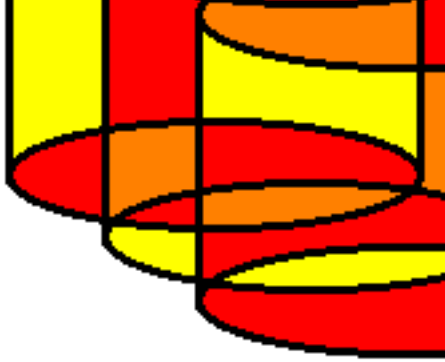
Primjeri

- Upit koji vraća listu (polje) naslova po autoru:

```
db.zapisi.aggregate( [ {
    $group : {
        _id : "$autor",
        "naslovi" :
            { $push:"$naslov" }
    }
} ] )
```



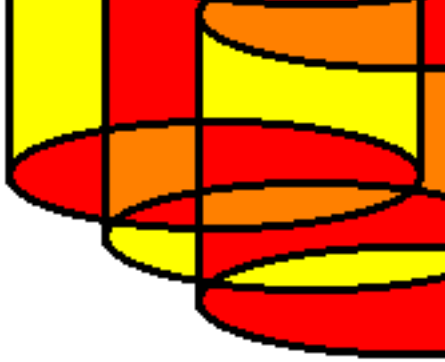
Cjevovod (engl. pipline)



- Slično kao UNIX shell (npr. operatorom |) MongoDB dopušta da rezultat jedne agregirajuće operacije bude ulaz u sljedeću.
- Primjer - korištene oznake prema autoru

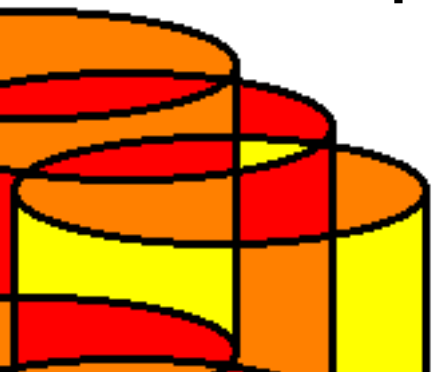
```
db.zapisi.aggregate( [
  { $unwind : "$oznake" },
  { $group : {
    _id : "$autor",
    "broj_lajkova" : { $addToSet:"$oznake" }
  } }
] ).pretty()
```

Cjevovod (engl. pipline)



Moguće su sljedeće operacije:

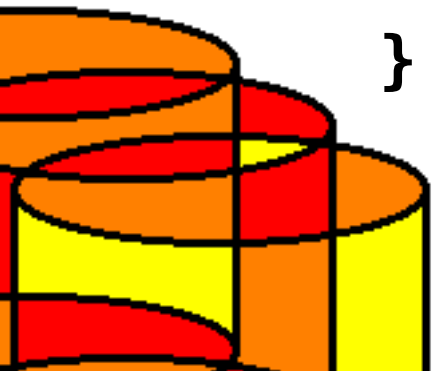
- **$project** – projekcija na samo određene vrijednosti u dokumentu
- **$match** – filtriranje
- **$group** – agregacija uz neku od funkcija (kao što je dano u prethodnim primjerima)
- **$sort** – sortiranje
- **$skip** – preskakanje određenog broja zapisa
- **$limit** – ograničavanje na određeni broj zapisa
- **$unwind** – pretvaranje polja u pojedinačne elemente
- **$lookup** – spajanje s drugim kolekcijama (slično JOIN u SQL-u)



Primjer za $lookup

- Upit koji uz zapise spaja korisnike

```
db.zapisi.aggregate( [ {
  $lookup:{
    from:"korisnici",
    localField:"autor",
    foreignField:"_id",
    as:"autor_obj"
  }
} ] ).pretty()
```



MapReduce

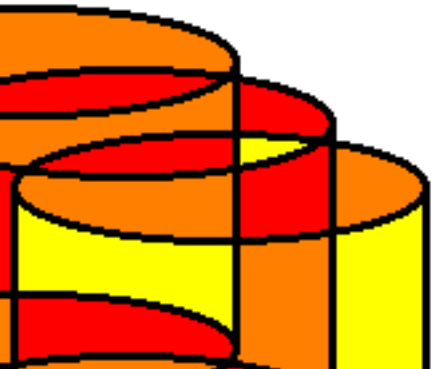
- Postupak mapiranja i reduciranja u MongoDB-u implementira se pomoću funkcija u JavaScript-u.
- Sintaksa je sljedeća:

```
db.[kolekcija].mapReduce(
   function() { // funkcija mapiranja
      emit( kljuc, vrijednost ) ;
   },
   function( kljuc, vrijednosti ) { // funkcija reduciranja
      return [reduceFunkcija]
   },
   {
      out: [kolekcija], // izlazna kolekcija
      query: [dokument], // filter kao u find() (opcionalno)
      sort: [dokument], // sortiranje kao u sort() (opcionalno)
      limit: [broj] // ograničenje izlaza  (opcionalno)
   }
)
```

Primjeri

- Ukupan broj lajkova za svakog korisnika

```
db.zapisi.mapReduce(
    function(){
       emit( this.autor, this.lajkova );
    },
    function( kljuc, vrijednost ){
       return Array.sum( vrijednost )
    },
    { out: "lajkovi ukupno" }
).find().pretty()
```



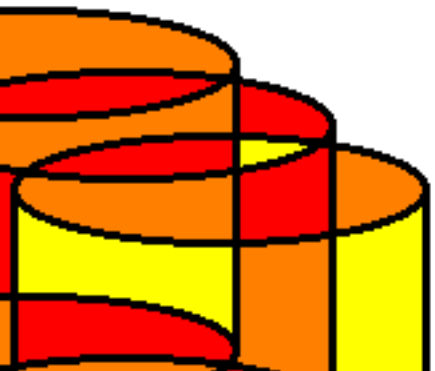
Primjeri

- Ukupan broj lajkova komentara za svakog korisnika:

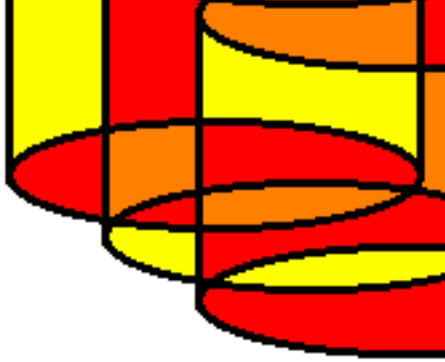
```
var mapiraj =  function(){
    for( var i in this.komentari )
    {
        emit( this.autor, this.komentari[ i ].lajkova );
    }
}

var reduciraj = function( kljuc, vrijednosti ){
    return Array.sum( vrijednosti )
}

db.zapisi.mapReduce( mapiraj, reduciraj, { "out":"lajkovi komentara" }
).find().pretty()
```

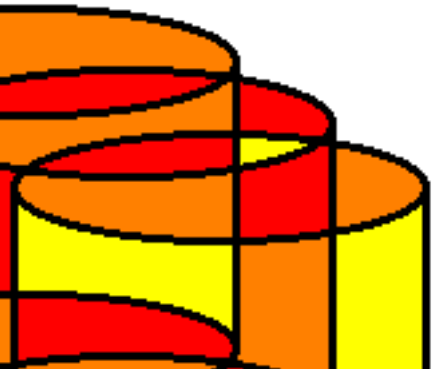


Ažuriranje baze podataka

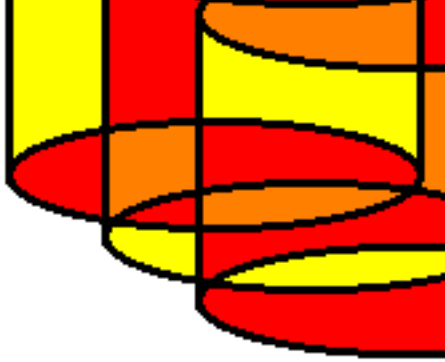


- Dokumente možemo ažurirati metodom update():

```
db.zapisi.update(
   { "naslov":"In vino veritas" },
   { $set:{ "lajkova":3293 } }
)
```

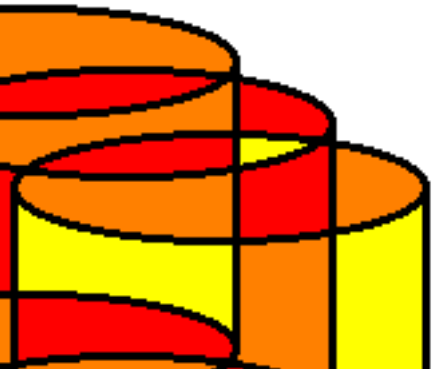


Ažuriranje baze podataka



- U pravilu metoda update() mijenja samo prvi dokument na koji naiđe. Ako želimo mijenjati sve dokumente na koje se odnosi kriterij moramo podesiti parametar multi:

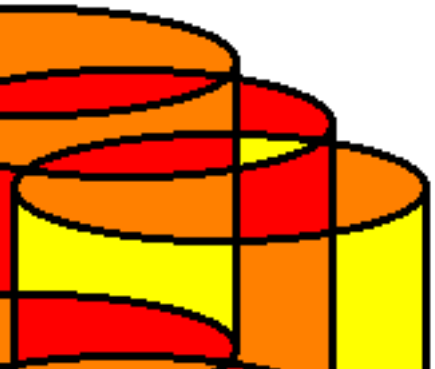
```
db.korisnici.update(
   { "država":"Hrvatska" },
   { $set:{ "na vezi":true } },
   { multi:true }
)
```



Metoda save()

- Prepisuje objekt s određenim identitetom objekta

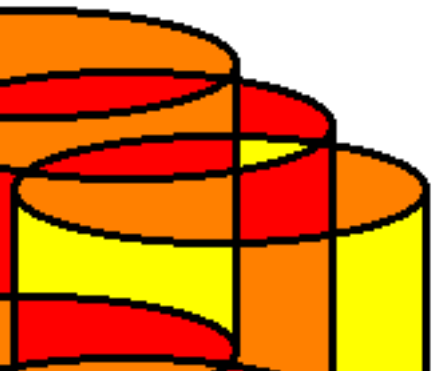
```
db.korisnici.save(
    {
        "_id":ObjectId("5de0232f41459750677cd9f1"),
        "e-mail":"stefa@tmobile.de",
        "ime":"Štefanija Jambrešćak-Prekratki",
        "godina rođenja":1998,
        "grad":"Berlin",
        "država":"Njemačka"
    }
)
```



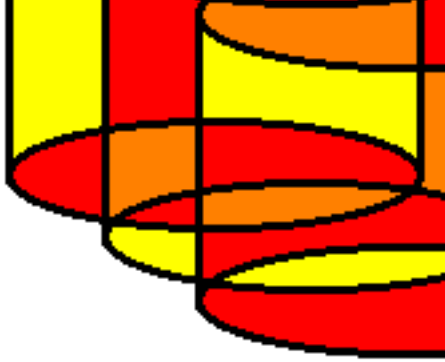
Brisanje dokumenata

- Za brisanje dokumenata koristimo metodu remove():

```
db.zapisi.remove( { "naslov":"Parkour po Dravi" } )
```

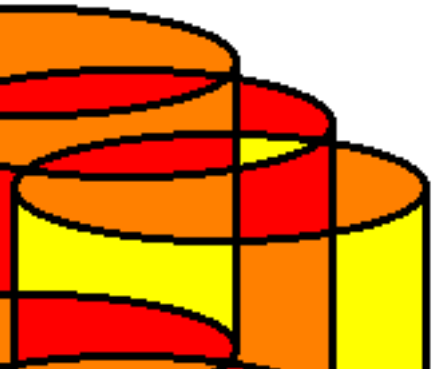


Indeksiranje



- Kako bi se povećala efikasnost pretraživanja prema nekom atributu mogu se koristiti indeksi, npr.

```
db.zapisi.ensureIndex( { "naslov":1 } )
```



Zadatak

- U bazu podataka iz prethodnog zadatka pridodajte:
  - Barem 3 upita s agregacijom
  - Barem 3 upita s MapReduce-om
  - Barem 3 ažuriranja
- Sve korištene naredbe dodajte u datoteku ime_prezime.js iz prethodnog zadatka