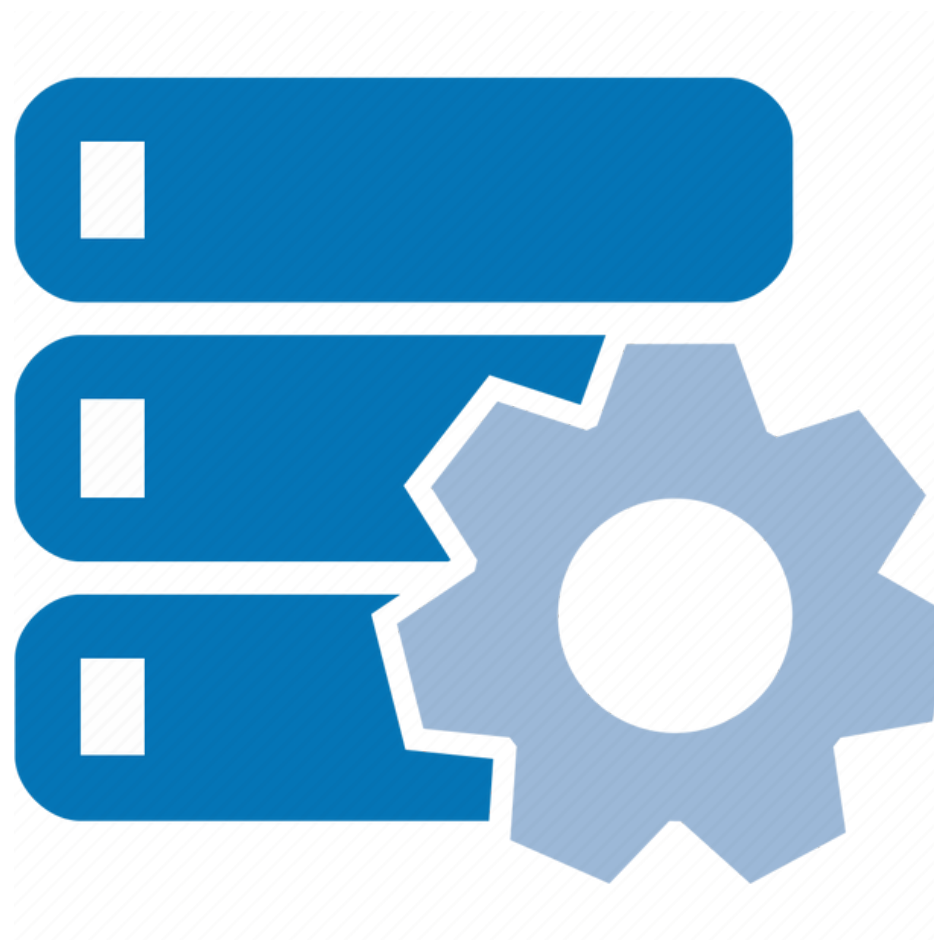


Administriranje baza podataka

Dokument baza podataka



Smerovi: SRT, KOT

Predmetni profesor: Dr Dušan Stefanović, prof. struk. studija

Predmetni asistenti: Nevena Minić, Nikola Vukotić

Cilj vežbe

1. **Razumevanje fleksibilnosti strukture podataka u dokument bazi:** Upoznavanje sa prednostima i izazovima u vezi sa nestrukturiranim podacima u dokument bazama, uključujući modeliranje podataka koji mogu imati varijabilnu strukturu i nestandardizovane attribute.
2. **Kreiranje i upravljanje kolekcijama i dokumentima:** Razumevanje kako kreirati i organizovati kolekcije u dokument bazi podataka, kao i upisivanje, ažuriranje, i brisanje dokumenata u kolekcijama.
3. **Razumevanje prednosti i kada koristiti dokument baze podataka:** Upoznavanje sa slučajevima kada je dokument baza pogodnija od relacionih baza, kao i razumevanje kada koristiti fleksibilnost koju ona nudi za rad sa nestandardizovanim i dinamičkim podacima.

Dokument baze podataka

- **Document-oriented baza podataka** predstavlja kompjuterski softver čija je osnovna namena skladištenje, pribavljanje i upravljanje **document-oriented** ili **polustrukturanim** podacima.
- Document-oriented baza podataka = **Document store**
- Document store predstavlja jednu od kategorije **NoSQL baza podataka**.
- Za razliku od relacionih baza podataka koje su organizovane oko koncepta relacije (tabele), document-oriented baze podataka su organizovane oko apstraktnog koncepta **DOKUMENT**.
- Svaka document-oriented baza podataka implementira koncept dokumenta sa **različitim nivoima detaljnosti**.
- Document-oriented baza podataka predstavlja **kolekciju dokumenata**.
- Sve informacije iz dokumenta se nalaze u samom dokumentu, nema povezanih tabela.
- **Nema potrebe za striktnim definisanje šeme baze podataka!**
- Za razliku od relacione baze podataka gde sve vrste imaju iste kolone, kod document store baza podataka **ne moraju svi dokumenti da imaju istu strukturu**.
- Ukoliko postoji potreba da se u dokument doda novo polje, novo polje se dodaje bez uticaja na prethodno uskladištene dokumente.
- Dokumenti ne čuvaju praznu vrednost za polje čija je vrednost nepoznata ili nedefinisana

Dokument baze podataka

- **Objekti se čuvaju kao dokumenti** – Objektni model se snima u dokument i smešta u bazu podataka.
- **Dokumenti mogu da budu kompleksni**– čitav objektni model može da se učitava/snimi odjednom. Nema potrebe za velikim brojem atomičnih CRUD operacija.
- **Dokumenti su nezavisni**– Povećavaju se performanse i smanjuju problemi vezani za konkurentan pristup podacima.
- **Otvoreni formati**– Podaci se najčešće čuvaju korišćenjem otvorenih formata: XML, JSON, BSON (Binary JSON), YAML. Postoje i baze podataka koje čuvaju i dokumenta u binarnim formatima (PDF ili Office dokumenti)
- **Ne postoji striktna šema baze podataka** – Povećava se fleksibilnost sistema koji se razvija. Novi podaci mogu da se dodaju bez potrebe da se menjaju postojeća dokumenta u bazi podataka:

```
{  
  Ime:"Petar",  
  Prezime:"Petar",  
  Adresa:"Voždova 23"  
}
```



```
{  
  Ime:"Goran",  
  Prezime:"Stojanović",  
  Adresa:"Dušanova 23"  
  Deca: [  
    {Ime:"Vera", Pol: "Ž"},  
    {Ime:"Todor", Pol: "M"}  
  ]  
}
```

Primer dokumenta i dodavanja novih podataka u postojeći dokument

MongoDB

MongoDB je open-source NoSQL baza podataka, razvijena za čuvanje i obradu velikih količina fleksibilnih, nestrukturisanih podataka. Umesto klasičnih tabela i redova (kao u SQL bazama), koristi **dokumente** slične JSON-u, organizovane u kolekcije.

Karakteristike:

- Razvila kompanija 10gen (danas MongoDB Inc.)
- 📅 Godina nastanka: 2007
- 🌐 Prvi javni *release*: 2009
- 📍 Sedište: New York, SAD
- 🧪 Ime "Mongo" dolazi od "humongous" – što znači "ogroman", jer je dizajnirana za **Big Data**.
- 📄 Dokument-orientisana (BSON format – binarni JSON)
- 🧠 Nema striktne šeme (schema-less)
- ⚙️ Upiti u JSON stilu
- 🔁 Replikacija, sharding (horizontalno skaliranje)
- 🔍 Moćna agregacija i pretraga
- 🌐 Odlična za real-time aplikacije
- 🌐 Zvanični sajt: <https://www.mongodb.com>
- 📖 Dokumentacija: <https://www.mongodb.com/docs/>

mongoDB®



Zadatak - Katalog proizvoda

Potrebno je projektovati i kreirati sistem za prikaz kataloga proizvoda za online prodavnicu. Za svaki proizvod u katalogu pamte se ID, naziv (name), kategorija (category) i atributi (attributes) koji se razlikuju od proizvoda do proizvoda. Na primer:

- Elektronika može imati attribute kao što su "baterija", "garancija", "model", itd.
- Odeća može imati attribute kao što su "veličina", "boja", "materijal", itd.
- Nameštaj može imati attribute kao što su "dimenzije", "materijal", "boja", itd.

REŠENJE: Korišćenje dokument baze podataka (pr. MongoDB) za skladištenje - *fleksibilnost u strukturi podataka*.

- MongoDB koristi dokumentski model, gde svaki proizvod može biti predstavljen kao JSON dokument. Svaki dokument može imati različite attribute, a struktura tih atributa nije fiksna, što znači da nije potrebno unapred definisati šablon za sve proizvode. Primer:
 - Proizvod iz kategorije elektronike može imati atribut "baterija" i "model", dok proizvod iz kategorije odeće ne mora imati ove attribute.

*Dokument u
MongoDB - jedan
objekat, proizvod iz
kategorije
elektronika*

```
{
  "product_id": "E123",
  "name": "Pametani telefon",
  "category": "Elektronika",
  "attributes": {
    "model": "XPhone 2025",
    "battery": "4000mAh",
    "warranty": "2 godine",
    "price": 399.99
  }
}
```

*Dokument u
MongoDB - jedan
objekat, proizvod iz
kategorije odeća*

```
{
  "product_id": "C456",
  "name": "T-shirt",
  "category": "Odeća",
  "attributes": {
    "size": "M",
    "color": "Crna",
    "material": "Pamuk",
    "price": 19.99
  }
}
```

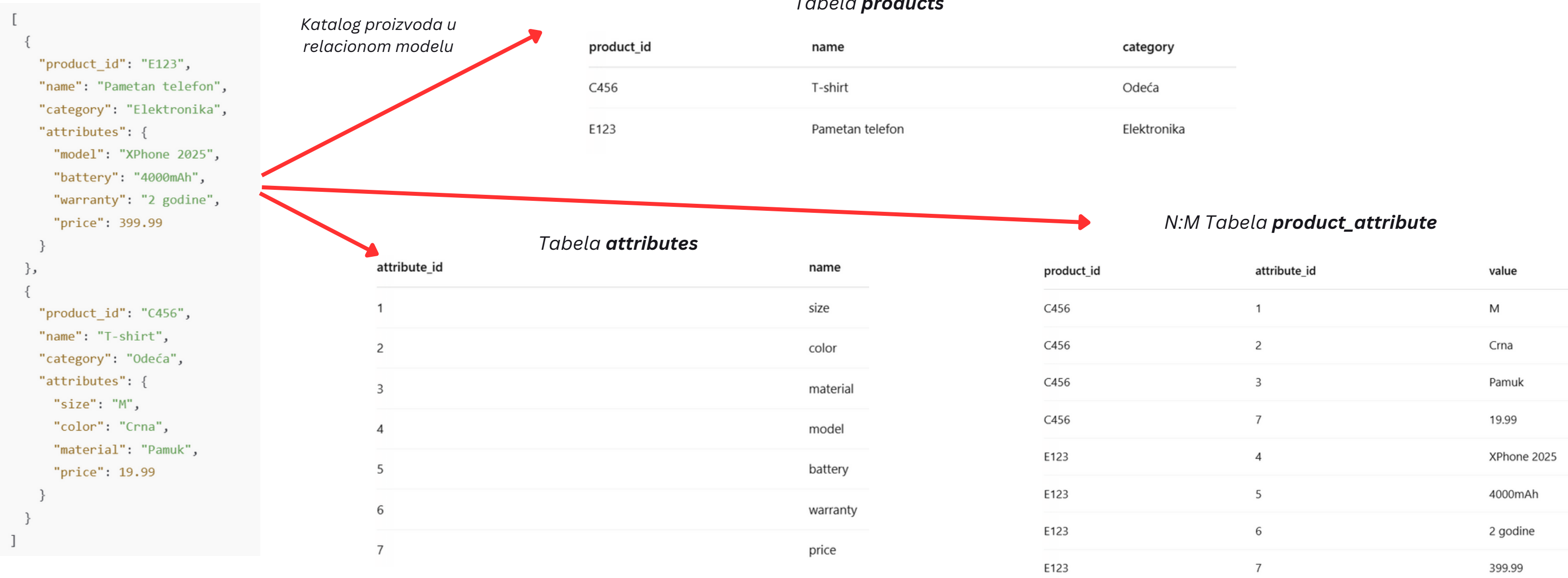
*Kolekcija
dokumenata (svih
objekata, proizvoda)
u MongoDB*

```
[
  {
    "product_id": "E123",
    "name": "Pametani telefon",
    "category": "Elektronika",
    "attributes": {
      "model": "XPhone 2025",
      "battery": "4000mAh",
      "warranty": "2 godine",
      "price": 399.99
    }
  },
  {
    "product_id": "C456",
    "name": "T-shirt",
    "category": "Odeća",
    "attributes": {
      "size": "M",
      "color": "Crna",
      "material": "Pamuk",
      "price": 19.99
    }
  }
]
```

Zadatak - Katalog proizvoda

Mane relacionog SQL modela za ovaj use-case:

- **Fiksna struktura i zahtev za normiranjem podataka:** U SQL bazi morala bi da se koristi struktura sa više tabela:
 - products (product_id, name, category)
 - attributes (attribute_id, name) - npr. *model, boja, veličina, materijal*
 - product_attribute (product_id, attribute_id, value)
- N:M veza između proizvoda i atributa
- ❌ Upiti postaju kompleksni (potreban JOIN za svaki atribut), ❌ Loša čitljivost, ❌ Održavanje i izmene šeme postaju komplikovane (dodavanje novih atributa), ❌ Nema prirodne hijerarhije i ugneždenosti kao u dokumentnom modelu.



- **Ograničenja u fleksibilnosti:** SQL baze podataka se oslanjaju na striktne šeme, što znači da svaka promena u strukturi proizvoda može zahtevati migraciju baze ili promene u tabelama. MongoDB omogućava fleksibilnost jer je moguće dodati nove attribute bez potrebe za promenama u celokupnoj strukturi baze.

Upiti - Katalog proizvoda

MongoDB komande:

- **Mongo shell:** Interaktivni JavaScript interfejs za rad sa MongoDB. Omogućava izvršavanje administrativnih operacija i upita nad podacima. Pokretanje shell-a komandom *mongosh*

```
PS C:\Users\neven> mongosh
Current Mongosh Log ID: 67a87871f33e9e8ea24d7941
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.9
Using MongoDB:      7.0.12
Using Mongosh:      2.3.9

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2025-02-07T16:29:20.408+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
```

- **Prikaz svih baza:** *show dbs*
- **Prebacivanje/kreiranje nove baze:** *use products*
- **Prikaz svih kolekcija unutar baze:** *show collections*

```
test> show dbs
admin    40.00 KiB
config   60.00 KiB
local    88.00 KiB
```

```
test> use products
switched to db products
```

```
products> show collections
products
```

Upiti - Katalog proizvoda

- Dodavanje novog dokumenta/proizvoda u kolekciju

products: `db.products.insertOne`

```
products> db.products.insertOne({
...   "product_id": "E123",
...   "name": "Pametn telefon",
...   "category": "Elektronika",
...   "attributes": {
...     "model": "XPhone 2025",
...     "battery": "4000mAh",
...     "warranty": "2 godine",
...     "price": 399.99
...   }
... })
{
  acknowledged: true,
  insertedId: ObjectId('67a878fcf33e9e8ea24d7942')
}
```

U MongoDB-u, svaki dokument u kolekciji mora imati polje `_id` koje služi kao jedinstveni identifikator dokumenta – nešto kao PRIMARY KEY u SQL bazama.

Ako se ne postavi `_id` ručno pri unosu, MongoDB automatski dodeljuje vrednost tipa `ObjectId`.

- Dodavanje više dokumenata/proizvoda u kolekciju products: `db.products.insertMany`

```
products> db.products.insertMany([
...   {
...     "product_id": "E124",
...     "name": "Pametn telefon",
...     "category": "Elektronika",
...     "attributes": {
...       "model": "XPhone 2025 Pro",
...       "battery": "5000mAh",
...       "warranty": "3 godine",
...       "price": 499.99
...     }
...   },
...   {
...     "product_id": "C456",
...     "name": "T-shirt",
...     "category": "Odeća",
...     "attributes": {
...       "size": "M",
...       "color": "Crna",
...       "material": "Pamuk",
...       "price": 19.99
...     }
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67a87986f33e9e8ea24d7943'),
    '1': ObjectId('67a87986f33e9e8ea24d7944')
  }
}
```

Upiti - Katalog proizvoda

- Prikaz svih dokumenata na osnovu id-ja:

`db.products.find({ product_id: "E123" })`

```
products> db.products.find({product_id: "E123"})
[
  {
    _id: ObjectId('67a878fcf33e9e8ea24d7942'),
    product_id: 'E123',
    name: 'Pametn telefon',
    category: 'Elektronika',
    attributes: {
      model: 'XPhone 2025',
      battery: '4000mAh',
      warranty: '2 godine',
      price: 299
    }
  }
]
```

- Ažuriranje jednog proizvoda na osnovu id-ja a:

`db.products.updateOne`

```
products> db.products.updateOne(
... { product_id: "E123"},
... {$set: { "attributes.price": 299.00} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Upiti - Katalog proizvoda

- Ažuriranje svih proizvoda iz kategorije Elektronika:

db.products.updateMany

```
products> db.products.updateMany(
...   { category: "Elektronika" },
...   { $set: { "attributes.price": 399.99 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```



```
products> db.products.find()
[
  {
    _id: ObjectId('67a878fcf33e9e8ea24d7942'),
    product_id: 'E123',
    name: 'Pametn telefon',
    category: 'Elektronika',
    attributes: {
      model: 'XPhone 2025',
      battery: '4000mAh',
      warranty: '2 godine',
      price: 399.99
    }
  },
  {
    _id: ObjectId('67a87986f33e9e8ea24d7943'),
    product_id: 'E124',
    name: 'Pametn telefon',
    category: 'Elektronika',
    attributes: {
      model: 'XPhone 2025 Pro',
      battery: '5000mAh',
      warranty: '3 godine',
      price: 399.99
    }
  },
  {
    _id: ObjectId('67a87986f33e9e8ea24d7944'),
    product_id: 'C456',
    name: 'T-shirt',
    category: 'Odeća',
    attributes: { size: 'M', color: 'Crna', material: 'Pamuk', price: 19.99 }
  }
]
```

Upiti - Katalog proizvoda

- Brisanje proizvoda na osnovu id-ja:

`db.products.deleteOne( {product_id: "C456"})`

```
products> db.products.deleteOne( {product_id: "C456"})  
{ acknowledged: true, deletedCount: 1 }
```



```
products> db.products.find()  
[  
  {  
    _id: ObjectId('67a878fcf33e9e8ea24d7942'),  
    product_id: 'E123',  
    name: 'Pametani telefon',  
    category: 'Elektronika',  
    attributes: {  
      model: 'XPhone 2025',  
      battery: '4000mAh',  
      warranty: '2 godine',  
      price: 399.99  
    }  
  },  
  {  
    _id: ObjectId('67a87986f33e9e8ea24d7943'),  
    product_id: 'E124',  
    name: 'Pametani telefon',  
    category: 'Elektronika',  
    attributes: {  
      model: 'XPhone 2025 Pro',  
      battery: '5000mAh',  
      warranty: '3 godine',  
      price: 399.99  
    }  
  }  
]
```

Agregacija - MongoDB

Agregacija u MongoDB predstavlja način obrade i analize većih skupova podataka kroz niz faza koje transformišu dokumente.

Služi za:

- **filtriranje**
- **grupisanje**
- **sabiranje**
- **računanje proseka, minimuma, maksimuma**
- **pravljenje zbirnih izveštaja**
- Baš kao što se u SQL-u koristi GROUP BY, SUM(), AVG() itd, u MongoDB-u se to radi preko *agregacionog pipeline-a* (lanca faza).

MongoDB koristi aggregate() metodu, u koju se prosleđuje niz faza, a svaka faza je objekt sa određenim operatorom (npr. \$match, \$group, \$sort...).

Najčešće MongoDB agregacione faze

1. **\$match** – filtrira dokumente (kao WHERE u SQL-u)

```
{ $match: { category: "Elektronika" } }
```

2. **\$group** – grupiše dokumente i primenjuje agregacione funkcije

```
{
  $group: {
    _id: "$category",
    total_price: { $sum: "$attributes.price" }
  }
}
```

3. **\$sort** – sortira rezultat (-1 opadajuće, 1 rastuće)

```
{ $sort: { "attributes.price": -1 } } // Po ceni opadajuće
```

3. **\$project** – bira koje polja da prikaže (kao SELECT ime, prezime)

```
{
  $project: {
    name: 1,
    price: "$attributes.price"
  }
}
```

Operatori - MongoDB

MongoDB	Značenje	SQL ekvivalent
<code>\$gt</code>	Greater than	<code>></code>
<code>\$lt</code>	Less than	<code><</code>
<code>\$gte</code>	Greater than or equal	<code>>=</code>
<code>\$lte</code>	Less than or equal	<code><=</code>
<code>\$eq</code>	Equal	<code>=</code>
<code>\$ne</code>	Not equal	<code>!=</code> ili <code><></code>
<code>\$in</code>	Vrednost u listi	<code>IN (...)</code>
<code>\$and</code> , <code>\$or</code>	Logički operatori	<code>AND</code> , <code>OR</code>

Zadatak: Analiza proizvoda po kategoriji

U katalogu proizvoda prikazati:

- samo proizvode proizvedene posle 2021. godine
- za svaku kategoriju izračunati prosečnu cenu
- rezultate sortirati po prosečnoj ceni (rastuće)
- i na kraju prikazati samo naziv kategorije i prosečnu cenu, bez _id i drugih polja:

```
year: { $gt: 2021 }
```

\$gt: 2021, samo godine veće od 2021 (greater than)

```
$project: {  
  _id: 0,  
  category: "$_id",  
  avg_price: 1  
}
```

- *_id: 0 → sakrij (isključi) "_id" iz rezultata*
- *avg_price: 1 → prikaži "avg_price" kako jeste*
- *category: "\$_id" → koristi vrednost iz "_id" (koji je u \$group fazi bio npr. "Elektronika"), ali ga nazovi "category" u izlaznom rezultatu*

```
products> db.products.aggregate([  
  ... // 1. MATCH - filtriranje po godini  
  ... {  
  ...   $match: {  
  ...     year: { $gt: 2021 }  
  ...   },  
  ... },  
  ... // 2. GROUP - grupisanje po kategoriji i računanje prosečne cene  
  ... {  
  ...   $group: {  
  ...     _id: "$category",  
  ...     avg_price: { $avg: "$attributes.price" }  
  ...   }  
  ... },  
  ... // 3. SORT - sortiranje po prosečnoj ceni rastuće  
  ... {  
  ...   $sort: {  
  ...     avg_price: 1  
  ...   }  
  ... },  
  ... // 4. PROJECT - prikaz samo potrebnih polja  
  ... {  
  ...   $project: {  
  ...     _id: 0,  
  ...     category: "$_id",  
  ...     avg_price: 1  
  ...   }  
  ... },  
  ... ])  
[  
  { avg_price: 14.99, category: 'Knjige' },  
  { avg_price: 19.99, category: 'Odeća' },  
  { avg_price: 299.99, category: 'Nameštaj' }  
]
```

Zadaci - Katalog proizvoda

1. Ukupna cena svih proizvoda proizvedenih
posle 2021:

SQL paralela:

```
products> db.products.aggregate([
...   { $match: { year: { $gt: 2021 } } },
...   { $group: {
...     _id: null,
...     total_price: { $sum: "$attributes.price" }
...   }}
... ])
...
[ { _id: null, total_price: 334.97 } ]
```

```
SELECT SUM(price) AS total_price
FROM products
WHERE year > 2021;
```

*\$group: { _id: null } - grupisati sve dokumente zajedno,
ne na osnovu nekog polja. MongoDB ne može sabrati
sve dokumente bez group faze*

Zadaci - Katalog proizvoda

2. Pronaći proizvod sa najmanjom cenom:

```
products> db.products.find().sort({ "attributes.price": -1 }).limit(1)
[
  {
    _id: ObjectId('67a878fcf33e9e8ea24d7942'),
    product_id: 'E123',
    name: 'Pametn telefon',
    category: 'Elektronika',
    attributes: {
      model: 'XPhone 2025',
      battery: '4000mAh',
      warranty: '2 godine',
      price: 399.99
    }
  }
]
```

SQL paralela:

```
SELECT * FROM products
ORDER BY price ASC
LIMIT 1;
```

Zadaci - Katalog proizvoda

3. Za svaku kategoriju izvući maksimalnu cenu:

```
products> db.products.aggregate([
...   { $group: {
...     _id: "$category",
...     max_price: { $max: "$attributes.price" }
...   }}
... ])
```

```
[
  { _id: 'Elektronika', max_price: 399.99 },
  { _id: 'Knjige', max_price: 14.99 },
  { _id: 'Nameštaj', max_price: 299.99 },
  { _id: 'Odeća', max_price: 29.99 }
]
```

SQL paralela:

```
SELECT category, MAX(price) AS max_price
FROM products
GROUP BY category;
```

Zadaci - Katalog proizvoda

4. Prikaz svih proizvoda čija je cena manja od 100:

```
products> db.products.find({ "attributes.price": { $lt: 100 } })
[
  {
    _id: ObjectId('67fcbdd77b926ebea74d7942'),
    product_id: 'C456',
    name: 'T-shirt',
    category: 'Odeća',
    year: 2022,
    attributes: { size: 'M', color: 'Crna', material: 'Pamuk', price: 19.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7943'),
    product_id: 'C457',
    name: 'Duks',
    category: 'Odeća',
    year: 2021,
    attributes: { size: 'L', color: 'Siva', material: 'Pamuk', price: 29.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7945'),
    product_id: 'F790',
    name: 'Sto za kafu',
    category: 'Nameštaj',
    year: 2020,
    attributes: { material: 'Drvo', color: 'Braon', price: 89.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7946'),
    product_id: 'K333',
    name: 'Knjiga: MongoDB u praksi',
    category: 'Knjige',
    year: 2022,
    attributes: { author: 'Jovan Jović', pages: 312, price: 14.99 }
  }
]
```

SQL paralela:



```
SELECT * FROM products
WHERE price < 100;
```

Zadaci - Katalog proizvoda

5. Prikaz svih proizvoda proizvedenih između 2021. i 2023:

```
products> db.products.find({ year: { $gte: 2021, $lte: 2023 } })
[
  {
    _id: ObjectId('67fcbdd77b926ebea74d7942'),
    product_id: 'C456',
    name: 'T-shirt',
    category: 'Odeća',
    year: 2022,
    attributes: { size: 'M', color: 'Crna', material: 'Pamuk', price: 19.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7943'),
    product_id: 'C457',
    name: 'Duks',
    category: 'Odeća',
    year: 2021,
    attributes: { size: 'L', color: 'Siva', material: 'Pamuk', price: 29.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7944'),
    product_id: 'F789',
    name: 'Fotelja Relax',
    category: 'Nameštaj',
    year: 2023,
    attributes: { material: 'Koža', color: 'Bela', price: 299.99 }
  },
  {
    _id: ObjectId('67fcbdd77b926ebea74d7946'),
    product_id: 'K333',
    name: 'Knjiga: MongoDB u praksi',
    category: 'Knjige',
    year: 2022,
    attributes: { author: 'Jovan Jović', pages: 312, price: 14.99 }
  }
]
```

SQL paralela:

```
SELECT * FROM products
WHERE year BETWEEN 2021 AND 2023;
```

Zadaci - Katalog proizvoda

6. Broj proizvoda po kategoriji:

```
products> db.products.aggregate([  
...   { $group: {  
...     _id: "$category",  
...     count: { $sum: 1 }  
...   }}  
... ])  
...
```

```
[  
  { _id: 'Elektronika', count: 2 },  
  { _id: 'Odeća', count: 2 },  
  { _id: 'Nameštaj', count: 2 },  
  { _id: 'Knjige', count: 1 }  
]
```

SQL paralela:

```
SELECT category, COUNT(*) AS count  
FROM products  
GROUP BY category;
```

Zadaci - Katalog proizvoda

7. Prosečna cena po kategoriji:

```
products> db.products.aggregate([
...   { $group: {
...     _id: "$category",
...     avg_price: { $avg: "$attributes.price" }
...   }}
... ])
```

```
[
  { _id: 'Elektronika', avg_price: 399.99 },
  { _id: 'Odeća', avg_price: 24.99 },
  { _id: 'Nameštaj', avg_price: 194.99 },
  { _id: 'Knjige', avg_price: 14.99 }
]
```

SQL paralela:

```
SELECT category, AVG(price) AS avg_price
FROM products
GROUP BY category;
```