



Laboratorijska vežba br. 9.

Agregacije I šeme

1. Cilj vežbe

- **Transformaciju podataka:** Razumevanje rada agregacionog okvira kroz faze (pipeline), gde se podaci dinamički obrađuju, filtriraju i preoblikuju iz sirovog formata u korisne izveštaje.
- **Implementacija relacionih veza:** Ovladavanje \$lookup operatorom za spajanje različitih kolekcija, čime se dokazuje da MongoDB može efikasno da rešava složene upite koji zahtevaju povezivanje podataka (poput SQL JOIN-a).
- **Osiguravanje integriteta baze:** Primena JSON šeme i validacije nad kolekcijama kako bi se uspostavila stroga pravila unosa, čime se sprečavaju greške i osigurava visok kvalitet podataka u NoSQL okruženju.

2. Standardizacija podataka (Priprema)

- Izvršiti komande da bi i ostali učenici (naziv vaše kolekcije) dobili ocene u istom formatu:

```
// Dodajemo ispite Jeleni
```

```
db.ucenici.updateOne({ ime: 'Jelena' }, { $set: {  
  ispiti: [ { predmet: 'Baze podataka', ocena: 9 }, { predmet: 'Programiranje', ocena:  
  8 } ]  
}});
```

```
// Dodajemo ispite Ivanu
```

```
db.ucenici.updateOne({ ime: 'Ivan' }, { $set: {  
  ispiti: [ { predmet: 'Baze podataka', ocena: 10 }, { predmet: 'Mreže', ocena: 9 } ]  
}});
```



3. Upiti

1. Korišćenjem agregacionog okvira i \$project operatora, transformisati dokumente tako da se u izlazu zadrže samo specifične kolone: ime, prezime i ispiti. Sve ostale kolone iz originalne kolekcije se odbacuju.

Rešenje:

```
db.ucenici.aggregate([
  { $project: { _id: 0, ime: 1, prezime: 1, ispiti: 1 } }
]);
```

2. Korišćenjem agregacionog okvira i \$match operatora, izdvojiti samo one dokumente iz kolekcije koji ispunjavaju specifične kriterijume: da je smer učenika "IT" i da je grad stanovanja "Beograd".

Rešenje:

```
db.ucenici.aggregate([
  { $match: { smer: "IT", grad: "Beograd" } }
]);
```

3. Korišćenjem agregacionog okvira i \$unwind operatora, transformisati dokumente tako da se niz ispiti dekonstruiše.

Rešenje:

```
db.ucenici.aggregate([
  { $unwind: "$ispiti" }
]);
```

4. Korišćenjem agregacionog okvira, izvršiti višefaznu obradu podataka kako bi se izračunala prosečna ocena za svakog učenika. Prvo je potrebno dekonstruisati niz ispita sa \$unwind, a zatim grupisati podatke po imenu učenika sa \$group i primeniti matematičku funkciju za prosek (\$avg) nad izdvojenim ocenama..

Rešenje:

```
db.ucenici.aggregate([
  { $unwind: "$ispiti" },
  { $group: { _id: "$ime", prosek: { $avg: "$ispiti.ocena" } } }
]);
```



5. Korišćenjem agregacionog okvira, proći kroz tri faze obrade podataka kako bi se izračunalo koliko je ukupno učenika polagalo predmet "Baze podataka". Potrebno je prvo razbiti nizove svih ispita, zatim izolovati samo traženi predmet i na kraju prebrojati pojavljivanja.

Rešenje:

```
db.ucenici.aggregate([
  { $unwind: "$ispiti" },
  { $match: { "ispiti.predmet": "Baze podataka" } },
  { $group: { "_id": "$ispiti.predmet", ukupnoPolagalo: { $sum: 1 } } }
]);
```

6. Korišćenjem agregacionog okvira, proći kroz proces dekonstrukcije nizova, sortiranja i ograničavanja rezultata kako bi se identifikovao i prikazao dokument sa najvećom pojedinačnom ocenom iz bilo kog predmeta.

Rešenje:

```
db.ucenici.aggregate([
  { $unwind: "$ispiti" },
  { $sort: { "ispiti.ocena": -1 } },
  { $limit: 1 }
]);
```

7. Korišćenjem koncepta validacione šeme, kreirati novu kolekciju pod nazivom predmeti. Cilj je uspostaviti stroga pravila za unos podataka kako bi se osigurao integritet baze. Svaki dokument u ovoj kolekciji mora imati tačno definisane kolone i tipove podataka, čime se sprečava unos neispravnih informacija.

- Pravila (Ograničenja):
- Obavezne kolone: naziv, espb i katedra moraju postojati u svakom dokumentu.
- Tipovi podataka: Naziv mora biti tekst (string), dok ESPB mora biti cijeli broj (int).
- Vrednosna ograničenja: ESPB bodovi moraju biti u rasponu od 2 do 8.
- Fiksne opcije (Enum): Katedra je ograničena isključivo na vrednosti "IT" ili "Dizajn".

Rešenje:

```
db.createCollection("predmeti", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
```



```
required: ["naziv", "espb", "katedra"],
properties: {
  naziv: {
    bsonType: "string",
    description: "Naziv predmeta mora biti tekst"
  },
  espb: {
    bsonType: "int",
    minimum: 2,
    maximum: 8,
    description: "ESPB mora biti ceo broj između 2 i 8"
  },
  katedra: {
    enum: ["IT", "Dizajn"],
    description: "Katedra može biti samo IT ili Dizajn"
  }
}
});
```

8. Izvršiti naredni kod:

```
db.predmeti.insertMany([
  { naziv: "Baze podataka", espb: 6, katedra: "IT" },
  { naziv: "Programiranje", espb: 8, katedra: "IT" },
  { naziv: "Mreže", espb: 6, katedra: "IT" },
  { naziv: "Grafički dizajn", espb: 5, katedra: "Dizajn" }
]);
```

9. Korišćenjem agregacionog okvira, izvršiti kompleksno povezivanje kolekcije učenici sa kolekcijom predmeti. Cilj je generisati izveštaj koji pored osnovnih podataka o studentu i oceni, prikazuje i ESPB bodove koji se nalaze u drugoj kolekciji. Ovaj proces simulira JOIN operaciju iz relacionih baza podataka.

- Faza 1 (\$unwind): Razbija niz ispiti kod učenika kako bi svaki položen predmet postao poseban dokument spreman za uparivanje.
- Faza 2 (\$lookup): Ključna faza koja vrši "pretragu" u drugoj kolekciji (predmeti). Povezuje polje ispiti.predmet iz prve kolekcije sa poljem naziv iz druge.
- Faza 3 (\$unwind): Rezultat lookup-a je uvek niz. Ovdje ga ponovo razbijamo kako bismo podatke o predmetu izvukli iz niza u običan objekat.



- Faza 4 (\$project): Finalno oblikovanje kolona. Kreiramo nove nazive (student, bodovi) i spajamo podatke iz obe kolekcije u jedan jasan prikaz.

Rešenje:

```
db.ucenici.aggregate([
  { $unwind: "$ispiti" }, // Razbijamo niz ispita da dobijemo pojedinačne nazive
  {
    $lookup: {
      from: "predmeti", // Kolekcija sa kojom spajamo
      localField: "ispiti.predmet", // Polje iz ucenici (nakon unwind)
      foreignField: "naziv", // Polje iz predmeti
      as: "detalji_predmeta" // Gde smeštamo rezultat
    }
  },
  { $unwind: "$detalji_predmeta" }, // Pretvaramo niz detalja u objekat radi lakše
  {
    $project: {
      _id: 0,
      student: "$ime",
      polozen_ispit: "$ispiti.predmet",
      ocena: "$ispiti.ocena",
      bodovi: "$detalji_predmeta.espb" // Podatak povučen iz druge kolekcije
    }
  }
]);
```