

Grupisanje:

```
Fakultet> db.studenti.aggregate([{$group: {_id: '$godUpisa', avgProsjek: {$avg: '$prosjek'}}}])
```

```
[
  { _id: 2024, avgProsjek: 7.15 },
  { _id: 2023, avgProsjek: 9.100000000000001 }
]
```

```
Fakultet> db.studenti.aggregate([{$group: {_id: '$smjer', avgProsjek: {$avg: '$prosjek'}}}])
```

```
[
  { _id: 'C', avgProsjek: 7.15 },
  { _id: 'B', avgProsjek: 8.8 },
  { _id: 'D', avgProsjek: 9.4 }
]
```

Brojanje (analogno count(*) agregatnoj funkciji iz SQL-a):

```
Fakultet> db.studenti.aggregate([{$group: {_id: '$smjer', avgProsjek: {$avg: '$prosjek'}, brStudenata: {$sum: 1}}}])
```

```
[
  { _id: 'C', avgProsjek: 7.15, brStudenata: 2 },
  { _id: 'D', avgProsjek: 9.4, brStudenata: 1 },
  { _id: 'B', avgProsjek: 8.8, brStudenata: 1 }
]
```

\$bucket - formiranje histograma (npr. koliko studenata ima prosjeke u opsegu od 6 do 8, a koliko od 8 do 10):

```
Fakultet> db.studenti.aggregate([{$bucket: {groupBy: '$prosjek', boundaries: [6, 8, 10]}}])
```

```
[ { _id: 6, count: 2 }, { _id: 8, count: 2 } ]
```

Proširivanje \$bucket faze dodatnim agregacijama:

```
Fakultet> db.studenti.aggregate([{$bucket: {groupBy: '$prosjek', boundaries: [6, 8, 10], output: {'avgProsjek': {$avg: '$prosjek'}}}}])
```

```
[
  { _id: 6, avgProsjek: 7.15 },
  { _id: 8, avgProsjek: 9.100000000000001 }
]
```

```
Fakultet> db.studenti.aggregate([{$bucket: {groupBy: '$prosjek', boundaries: [6, 8, 10], output: {'count': {$sum: 1}, 'avgProsjek': {$avg: '$prosjek'}}}}])
```

```
[
  { _id: 6, count: 2, avgProsjek: 7.15 },
  { _id: 8, count: 2, avgProsjek: 9.100000000000001 }
]
```

\$bucketAuto - umjesto eksplisitnog zadavanja granica, zadaje se broj intervala:

```
Fakultet> db.studenti.aggregate([{$bucketAuto: {groupBy: '$prosjek', buckets: 3,
output: {'count': {$sum: 1}, 'avgProsjek': {$avg: '$prosjek'}}}}])
[
  { _id: { min: 6.8, max: 7.5 }, count: 1, avgProsjek: 6.8 },
  { _id: { min: 7.5, max: 8.8 }, count: 1, avgProsjek: 7.5 },
  {
    _id: { min: 8.8, max: 9.4 },
    count: 2,
    avgProsjek: 9.1000000000000001
  }
]
```

Faza sortiranja za agregacioni pipeline:

```
Fakultet> db.studenti.aggregate([{$bucketAuto: {groupBy: '$prosjek', buckets: 2,
output: {'count': {$sum: 1}, 'avgProsjek': {$avg: '$prosjek'}}}}, {$sort:
{avgProsjek: -1}}])
[
  {
    _id: { min: 8.8, max: 9.4 },
    count: 2,
    avgProsjek: 9.1000000000000001
  },
  { _id: { min: 6.8, max: 8.8 }, count: 2, avgProsjek: 7.15 }
]
```

Ugnježdjeni objekti/dokumenti:

```
Fakultet> db.predmeti.insertMany([{'ime': 'Baze podataka', ects: 5, profesor:
{'ime': 'Nikola Nikolic', titula: 'Docent'}}, {'ime': 'Masinsko ucenje', ects: 6,
profesor: {'ime': 'Ivan Ivanovic', titula: 'Vanredni profesor'}}, {'ime': 'Paralelni
algoritmi', ects: 4, profesor: {'ime': 'Nikola Nikolic', titula: 'Docent'}}])
```

Pretraga po ugnježđenom polju - navodnici su obavezni:

```
Fakultet> db.predmeti.find({'profesor.titula': 'Docent'})
[
  {
    _id: ObjectId('6720fef127ad648ab2964033'),
    ime: 'Baze podataka',
    ects: 5,
    profesor: { ime: 'Nikola Nikolic', titula: 'Docent' }
  },
  {
    _id: ObjectId('6720fef127ad648ab2964035'),
    ime: 'Paralelni algoritmi',
    ects: 4,
    profesor: { ime: 'Nikola Nikolic', titula: 'Docent' }
  }
]
```

Dodavanje ugnježenih polja kroz update upit:

```
Fakultet> db.predmeti.updateMany({'profesor.ime': 'Nikola Nikolic'}, {$set:
{'profesor.godIskustva': 4}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

```
Fakultet> db.predmeti.updateMany({'profesor.ime': 'Ivan Ivanovic'}, {$set:
{'profesor.godIskustva': 10}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Pretraga po ugnježenom polju sa operatorom poređenja:

```
Fakultet> db.predmeti.find({'profesor.godIskustva': {$gt: 5}})
[
  {
    _id: ObjectId('6720fef127ad648ab2964034'),
    ime: 'Masinsko ucenje',
    ects: 6,
    profesor: {
      ime: 'Ivan Ivanovic',
      titula: 'Vanredni profesor',
      godIskustva: 10
    }
  }
]
```

Nizovi vrijednosti:

```
Fakultet> db.studenti.updateOne({brIndeksa: 10}, {$set: {ocjene: [7, 8, 9]}})
Fakultet> db.studenti.updateOne({brIndeksa: 2}, {$set: {ocjene: [6, 7, 8]}})
Fakultet> db.studenti.updateOne({brIndeksa: 3}, {$set: {ocjene: [8, 10, 8]}})
Fakultet> db.studenti.updateOne({brIndeksa: 5}, {$set: {ocjene: [9, 7, 7, 8]}})
```

Pretraga po nizu vrijednosti:

```
Fakultet> db.studenti.find({ocjene: [8, 10, 8]})
[
  {
    _id: ObjectId('67054201c63d988a95964035'),
    ime: 'Petar Petrovic',
    brIndeksa: 3,
    godUpisa: 2024,
  }
]
```

```
    smjer: 'C',
    prosjek: 6.8,
    ocjene: [ 8, 10, 8 ]
  }
]
```

Da li sve vrijednosti iz niza pripadaju datom skupu:

```
Fakultet> db.studenti.find({ocjene: {$all: [8, 10]}})
[
  {
    _id: ObjectId('67054201c63d988a95964035'),
    ime: 'Petar Petrovic',
    brIndeksa: 3,
    godUpisa: 2024,
    smjer: 'C',
    prosjek: 6.8,
    ocjene: [ 8, 10, 8 ]
  }
]
```

Da li niz sadrži datu vrijednost:

```
Fakultet> db.studenti.find({ocjene: 9})
[
  {
    _id: ObjectId('6705404cc63d988a95964034'),
    ime: 'Marko Markovic',
    brIndeksa: 10,
    godUpisa: 2023,
    smjer: 'D',
    lastModified: ISODate('2024-10-22T13:52:17.767Z'),
    prosjek: 9.4,
    ocjene: [ 7, 8, 9 ]
  },
  {
    _id: ObjectId('67054201c63d988a95964036'),
    ime: 'Jovan Jovanovic',
    brIndeksa: 5,
    godUpisa: 2023,
    smjer: 'B',
    lastModified: ISODate('2024-10-22T13:49:52.689Z'),
    prosjek: 8.8,
    ocjene: [ 9, 7, 7, 8 ]
  }
]
```

Da li niz sadrži bar jedan element veći ili jednak od date vrijednosti:

```
Fakultet> db.studenti.find({ocjene: {$gte: 9}})
[
  {
    _id: ObjectId('6705404cc63d988a95964034'),
    ime: 'Marko Markovic',
    brIndeksa: 10,
```

```
    godUpisa: 2023,  
    smjer: 'D',  
    lastModified: ISODate('2024-10-22T13:52:17.767Z'),  
    prosjek: 9.4,  
    ocjene: [ 7, 8, 9 ]  
  },  
  {  
    _id: ObjectId('67054201c63d988a95964035'),  
    ime: 'Petar Petrovic',  
    brIndeksa: 3,  
    godUpisa: 2024,  
    smjer: 'C',  
    prosjek: 6.8,  
    ocjene: [ 8, 10, 8 ]  
  },  
  {  
    _id: ObjectId('67054201c63d988a95964036'),  
    ime: 'Jovan Jovanovic',  
    brIndeksa: 5,  
    godUpisa: 2023,  
    smjer: 'B',  
    lastModified: ISODate('2024-10-22T13:49:52.689Z'),  
    prosjek: 8.8,  
    ocjene: [ 9, 7, 7, 8 ]  
  }  
]
```