

ЗАДАЦИ

Дате су следеће релације базе података ZDRAVSTVO.

- **PACIJENTI:** Садржи основне податке, низ алергија и низ историје прегледа
- **LEKARI:** Садржи податке о лекаrima, њиховој специјалности и лиценци

Колекција lekari [_id, ime, prezime, specijalnost, licenca, staz] садржи:

- 101, Dragan, Petrovic, pedijatar, 5566, 12
- 102, Maja, Jovanovic, Hirurg, 8899, 15
- 103, Stefan, Kostic, opsta_praksa, 2233, 5
- 104, Sara, Lazić, Pedijatar, 4411, 8

Колекција pacijenti [ime, prezime, grad, kategorija, alergije, pregledi (lekar_id, dijagnoza, cena)] садржи:

- Marko, Markovic, Beograd, odrasli, [Polen], [{ 102, Upala, 3500 }, { 103, Kontrola, 1200 }]
- Milica, Ilic, Nis, decije, [Penicilin, Dust], [{ 101, Prehlada, 2000 }]
- Slobodan, Stankovic, Nis, penzioneri, [], [{ 103, Pritisak, 1500 }]
- Ana, Nikolic, Kragujevac, odrasli, [], []

Задатак 1. [15 поена]

Написати MongoDB Shell наредбе за креирање описане базе, колекција и за додавање докумената у колекције.

```
use ZDRAVSTVO
```

```
db.createCollection('lekari')
```

```
db.createCollection('pacijenti')
```

```
db.lekari.insertMany([
  { _id: 101, ime: "Dragan", prezime: "Petrovic", specijalnost: "Pedijatar", licenca: 5566, staz: 12 },
  { _id: 102, ime: "Maja", prezime: "Jovanovic", specijalnost: "Hirurg", licenca: 8899, staz: 15 },
  { _id: 103, ime: "Stefan", prezime: "Kostic", specijalnost: "opsta_praksa", licenca: 2233, staz: 5 },
  { _id: 104, ime: "Sara", prezime: "Lazić", specijalnost: "Pedijatar", licenca: 4411, staz: 8 }
])
```

```
db.pacijenti.insertMany([
  {
    ime: "Marko", prezime: "Markovic", grad: "Beograd", kategorija: "odrasli",
    alergije: [ "Polen" ],
    pregledi: [ { lekar_id: 102, dijagnoza: "Upala", cena: 3500 }, { lekar_id: 103, dijagnoza: "Kontrola",
cena: 1200 } ]
  },
  {
    ime: "Milica", prezime: "Ilic", grad: "Nis", kategorija: "decije",
    alergije: [ "Penicilin", "Dust" ],
```

```
pregledi: [ { lekar_id: 101, dijagnoza: "Prehlada", cena: 2000 } ]
},
{
  ime: "Slobodan", prezime: "Stankovic", grad: "Nis", kategorija: "penzioneri",
  alergije: [ ],
  pregledi: [ { lekar_id: 103, dijagnoza: "Pritisak", cena: 1500 } ]
},
{
  ime: "Ana", prezime: "Nikolic", grad: "Kragujevac", kategorija: "odrasli",
  alergije: [ ],
  pregledi: [ ]
}
]
```

Задатак 2. [15 поена]

Написати mongosh упите за следеће операције над колекцијама:

а) [8 поена]

Приказати име, презиме и град свих пацијената који су из Ниша или Београда, а чије име почиње на 'М' или 'S'. Такође, пацијент не сме имати "Penicilin" у низу алергија (користити за проверу припадности низу оператор **\$nin** - Not In).

```
db.pacijenti.find({
  grad: { $in: ["Nis", "Beograd"] },
  ime: { $regex: '^[MS]' },
  alergije: { $nin: ["Penicilin"] }
}, { ime: 1, prezime: 1, grad: 1 })
```

б) [8 поена]

Свим пацијентима из града "Kragujevac" који имају поље алергије (проверити постојање поља са **\$exists: true**), додати нову алергију "Polen" у низ, али тако да се не стварају дупликати унутар низа.

```
db.pacijenti.updateMany(
  { grad: "Kragujevac", alergije: { $exists: true } },
  { $addToSet: { alergije: "Polen" } }
)
```

в) [8 поена]

Пацијенту са именом "Marko" и презименом "Markovic" поставити статус на "prioritetan" и увећати цену његовог последњег прегледа (приступ елементу низа са прегледима на позицији N: pregledi.N) за 500 динара (користити **\$inc**).

```
db.pacijenti.updateOne(
  { ime: "Marko", prezime: "Markovic" },
```

Срећан рад!

ПРЕДМЕТНИ НАСТАВНИК И ПРЕДМЕТНИ АСИСТЕНТИ

```
{
  $set: { status: "prioritetan" },
  $inc: { "pregledi.1.cena": 500 }
}
```

Задатак 3. [15 поена]

Креирати колекцију опрема са валидацијом (\$jsonSchema). Обавезна поља су naziv, tip и inventarski_broj [8 поена].

- naziv мора бити string,
- tip може бити само једна од следећих вредности: "dijagnostika", "hirurgija" или "laboratorija",
- inventarski_broj мора бити цео број већи од 5000.

```
db.createCollection("oprema", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["naziv", "tip", "inventarski_broj"],
      properties: {
        naziv: { bsonType: "string" },
        tip: { enum: ["dijagnostika", "hirurgija", "laboratorija"] },
        inventarski_broj: { bsonType: "int", minimum: 5001 }
      }
    }
  }
})
```

Написати скрипту (упит) која [8 поена]:

- Проналази сву опрему чији је тип "dijagnostika" и чија је вредност (nabavna_cena) већа од 150.000.
- Прескаче прва 2 резултата и приказује наредна 3.
- Из финалног приказа избацује поље _id.

```
db.oprema.find(
  { tip: "dijagnostika", nabavna_cena: { $gt: 150000 } },
  { _id: 0 }
).skip(2).limit(3)
```

Задатак 4. [15 поена]

Написати комплексне агрегационе оквири (pipelines) за следеће потребе:

а) [5 поена]

Израчунати просечну потрошњу (цена прегледа) по сваком пацијенту. Потребно је:

- "Разбити" низ pregledi на појединачне документе.

- Груписати податке по имену пацијента и израчунати просечну вредност свих његових прегледа.
- Резултате сортирати тако да они са највећим просеком буду први.

```
db.pacijenti.aggregate([
  { $unwind: "$pregledi" },
  { $group: {
    _id: "$ime",
    prosecna_cena: { $avg: "$pregledi.cena" }
  }},
  { $sort: { prosecna_cena: -1 } }
])
```

b) [7 поена]

Извршити спајање (\$lookup) пацијената са лекарима. Приказати:

- Име пацијента.
- Дијагнозу са прегледа.
- Име и презиме лекара који је извршио преглед (повући из колекције lekari на основу lekar_id).
- Филтрирати резултате тако да се приказују само прегледи код лекара чија је специјалност "Pedijatar".

```
db.pacijenti.aggregate([
  { $unwind: "$pregledi" },
  { $lookup: {
    from: "lekari",
    localField: "pregledi.lekar_id",
    foreignField: "_id",
    as: "lekar_info"
  }},
  { $unwind: "$lekar_info" },
  { $match: { "lekar_info.specijalnost": 'Pedijatar' } },
  { $project: {
    _id: 0,
    ime_pacijenta: "$ime",
    dijagnoza: "$pregledi.dijagnoza",
    ime_lekara: "$lekar_info.ime",
    prezime_lekara: "$lekar_info.prezime"
  } }
])
```

c) [7 поена]

Пронаћи најскупљи појединачни преглед икада урађен у систему. Приказати само име пацијента и износ цене.

```
db.pacijenti.aggregate([
  { $unwind: "$pregledi" },
  { $sort: { "pregledi.cena": -1 } },
  { $limit: 1 },
  {
    $project: {
      _id: 0,
      ime_pacijenta: "$ime",
      iznos_cene: "$pregledi.cena"
    }
  }
])
```