



Analiza i optimizacija MongoDB šeme za analitičke upite u lancu kafea

Finalni projekat iz predmeta Sistemi baza podataka

Fakultet tehničkih nauka, Novi Sad

Tim: Tamara Stanković IN 1/2021, Nikola Stojičić IN 11/2021

Datum: 23.9.2025.

O projektu i skupu podataka



Tema: Sveobuhvatna analiza transakcija u lancu kafea sa fokusom na razumevanje kupovnih navika, performansi prodavnica i efikasnosti programa lojalnosti.

Cilj: Demonstrirati kako pravilan odabir MongoDB šeme utiče na performanse analitičkih upita i dokazati superiornost denormalizovane šeme za potrebe analize.

Izvor: Kaggle - "G Coffee Shop Transaction Dataset".

Obuhvat: Analiziramo kompletan set podataka za jul 2023. godine.

Ključne metrike:

- 7 povezanih CSV datoteka
- ~343 MB ukupne veličine
- 620.000+ transakcija
- 1.2 miliona+ prodatih stavki

Inicijalna logička šema (V1): Normalizovan pristup

- Sastoji se iz **7 odvojenih kolekcija**, prateći principe normalizacije slične relacionim bazama.
- **Statičke kolekcije (Šifarnici)**: *stores, menu_items, payment_methods, vouchers*
- **Dinamičke kolekcije (Podaci)**: *users, transactions, transaction_items*
- **Problem**: Da bi se dobila kompletna slika jedne transakcije, neophodno je spajanje podataka iz više kolekcija korišćenjem skupe *\$lookup* operacije, što je fundamentalno "usko grlo" za analitiku.

Rešenje: Optimizovana šema (V2)

Cilj:

Eliminacija \$lookup-a i prebacivanje kompleksnih kalkulacija sa čitanja na upis.

Strategija:

Denormalizacija podataka u **jednu centralnu *transactions* kolekciju** primenom tri ključna MongoDB dizajn šablonata.

Struktura jednog dokumenta

Jedan dokument sada sadrži sve informacije o transakciji, eliminajući potrebu za spajanjem (`$lookup`).

```
{  
    "_id": "a1b2c3d4-e5f6...",  
    "created_at": "2023-07-15T14:45:10.000Z",  
    "amounts": { "final": 14.18 },  
  
    // --- Primjenjeni Dizajn Šabloni ---  
  
    // 1. Šablon Proračunavanja (Computed Pattern)  
    "createdAtDetails": {  
        "year": 2023,  
        "month": 7,  
        "dayOfWeek": 6  
    },  
  
    // 2. Šablon Proširene Reference (Extended Reference)  
    "store": {  
        "id": 1,  
        "name": "G Coffee Downtown"  
    },  
    "user": {  
        "id": 123,  
        "gender": "Female",  
        "age_group": "25-34"  
    },  
  
    // 3. Šablon Ugradivanja (Embedding Pattern)  
    "items": [  
        {  
            "menu_item_id": 101,  
            "name": "Espresso",  
            "category": "coffee",  
            "quantity": 1  
        },  
        {  
            "menu_item_id": 205,  
            "name": "Croissant",  
            "category": "non-coffee",  
            "quantity": 1  
        }  
    ]  
}
```

Primenjeni dizajn šabloni

1. Ugrađivanje (Embedding Pattern):

- **Šta:** Umesto posebne *transaction_items* kolekcije, niz *items* je sada **ugrađen** unutar svakog dokumenta transakcije.
- **Zašto:** Omogućava dobavljanje svih podataka o transakciji u **jednom čitanju sa diska**, potpuno eliminišući najskuplje spajanje.

2. Proširena referenca (Extended Reference Pattern):

- **Šta:** Umesto da čuvamo samo *store_id*, u dokument transakcije kopiramo i najčešće potrebne podatke kao što su *store.name* i *store.city*.
- **Zašto:** Eliminiše potrebu za *\$lookup* operacijama za dobijanje osnovnih informacija.

3. Proračunavanje (Computed Pattern):

- **Šta:** Vrednosti koje se često koriste za analizu, kao što su *dayOfWeek*, *hour* ili *age_group*, izračunavaju se *jednom prilikom upisa* i čuvaju direktno u dokumentu.
- **Zašto:** Eliminiše skupa izračunavanja "u letu" koja se ponavljaju svaki put kada se upit izvrši.

Strategija indeksiranja u V2

Kreirali smo namenske indekse koji podržavaju naše analitičke upite:

1. Složeni indeks (Compound Index):

- { "created_at": 1, "amounts.final": -1 }
- Omogućava brzo filtriranje po datumu i sortiranje po vrednosti, idealno za upite o "top transakcijama".

2. Retki indeks (Sparse Index):

- { "user.id": 1, sparse: true } i { "voucher.id": 1, sparse: true }
- Indeksira **samo** dokumente koji sadrže dato polje. Omogućava bazi da trenutno pronađe sve transakcije registrovanih korisnika (ili sa vaučerom), ignorišući 98% kolekcije.

3. Višeključni indeks (Multikey Index):

- { "items.category": 1 }
- Indeksira vrednosti unutar niza *items*. Omogućava brzo pronalaženje svih transakcija koje sadrže određenu kategoriju proizvoda.

Upit 1 (N): Analiza performansi po kategorijama

Svrha upita: Za svaku kategoriju proizvoda (*coffee, non-coffee*) izračunava ukupan prihod, ukupan broj prodatih artikala i prosečnu cenu po artiklu.

Kod (V1):

```
db.getCollection('transaction_items').aggregate([
  { $lookup: { from: "menu_items", localField: "menu_item_id", foreignField: "_id", as: "details" } },
  { $unwind: "$details" },
  {
    $group: {
      _id: "$details.category",
      revenue: { $sum: "$subtotal" },
      quantity: { $sum: "$quantity" }
    }
  },
  { $sort: { revenue: -1 } },
  {
    $project: {
      _id: 0,
      category: "$_id",
      total_revenue: { $round: ["$revenue", 2] },
      total_items_sold: "$quantity",
      average_price_per_item: { $round: [{ $divide: ["$revenue", "$quantity"] }, 2] }
    }
  }
], { allowDiskUse: true })
```

Upit 1 (N): Analiza performansi po kategorijama

Kod (V2 - Optimizovano):

```
db.transactions.aggregate([
  // FAZA 1: "Rasturi" (unwind) niz ugrađenih stavki 'items'.
  {
    $unwind: "$items"
  },
  // FAZA 2: Grupiši po kategoriji koja je sada direktno dostupna unutar svake stavke.
  {
    $group: {
      _id: "$items.category",
      totalRevenue: { $sum: "$items.subtotal" },
      totalQuantitySold: { $sum: "$items.quantity" }
    }
  },
  // FAZA 3: Sortiraj rezultate po ukupnom prihodu.
  {
    $sort: {
      totalRevenue: -1
    }
  },
  // FAZA 4: Finalna projekcija i izračunavanje prosečne cene.
  {
    $project: {
      _id: 0,
      category: "$_id",
      total_revenue: { $round: ["$totalRevenue", 2] },
      total_items_sold: "$totalQuantitySold",
      average_price_per_item: {
        $round: [
          { $divide: ["$totalRevenue", "$totalQuantitySold"] },
          2
        ]
      }
    }
  }
])
```

Metrika	V1 šema	V2 šema	Promena
Vreme	~4 min (236 s)	~9 sec	~96% brže
Pregledanih dokumenata	~2.5 miliona	~1.2 miliona	50% manje

Zaključak: Eliminacija `$lookup`-a ugradnjom `items` niza je pretvorila neupotrebljiv upit u praktičan alat.

Upit 2 (N): Analiza načina plaćanja po danu

Svrha upita: Analizira i upoređuje procenat korišćenja gotovine (cash) u jutarnjim (pre 12h) i popodnevnim (posle 12h) satima.

Kod (V1):

```
db.getCollection('transaction_items').aggregate([
  {
    $lookup: {
      from: "menu_items",
      localField: "menu_item_id",
      foreignField: "_id",
      as: "details"
    }
  },
  { $unwind: "$details" },
  {
    $group: {
      _id: "$details.category",
      revenue: { $sum: "$subtotal" },
      quantity: { $sum: "$quantity" }
    }
  },
  {
    $project: {
      _id: 0,
      category: "$_id",
      total_revenue: { $round: ["$revenue", 2] },
      total_items_sold: "$quantity",
      average_price_per_item: { $round: [{ $divide: ["$revenue", "$quantity"] }, 2] }
    }
  },
  { $sort: { total_revenue: -1 } }
], { allowDiskUse: true })
```

Upit 2 (N): Analiza načina plaćanja po danu

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~94 sec	~2 sec	~98% brže
Pregledanih dokumenata	~2.5 miliona	~1.2 miliona	50% manje

Zaključak: Eliminisanje `$lookup`-a i korišćenje pre-kalkulisanog polja `createdAtDetails.hour` donelo je ogromno ubrzanje i smanjilo opterećenje diska za 50%.

```
db.transactions.aggregate([
  {
    $project: {
      period: {
        $cond: {
          if: { $lt: ["$createdAtDetails.hour", 12] },
          then: "Morning (AM)",
          else: "Afternoon (PM)"
        }
      },
      isCash: { $eq: ["$payment_method.name", "cash"] }
    }
  },
  {
    $group: {
      _id: "$period",
      totalTransactions: { $sum: 1 },
      cashTransactions: { $sum: { $cond: { if: "$isCash", then: 1, else: 0 } } }
    }
  },
  {
    $project: {
      _id: 0,
      period: "$_id",
      cash_usage_percentage: {
        $round: [
          { $multiply: [{ $divide: ["$cashTransactions", "$totalTransactions"] }, 100] },
          2
        ]
      }
    }
  },
  {
    $sort: { period: 1 }
  }
])
```

Upit 3 (N): Analiza potrošačkih navika po polu

Svrha upita: Pruža detaljnu analizu i poređenje potrošačkih navika između muških i ženskih korisnika.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  { $match: { user_id: { $ne: null } } },
  { $lookup: { from: "users", localField: "user_id", foreignField: "_id", as: "userDetails" } },
  { $unwind: "$userDetails" },
  {
    $group: {
      _id: "$userDetails.gender",
      totalSpend: { $sum: "$amounts.final" },
      totalTransactions: { $sum: 1 },
      totalUsers: { $addToSet: "$user_id" }
    }
  },
  {
    $project: {
      _id: 0,
      gender: "$_id",
      total_spend: { $round: ["$totalSpend", 2] },
      total_transactions: "$totalTransactions",
      number_of_users: { $size: "$totalUsers" },
      average_transaction_value: { $round: [{ $divide: ["$totalSpend", "$totalTransactions"] }, 2] },
      average_transactions_per_user: { $round: [{ $divide: ["$totalTransactions", { $size: "$totalUsers" }] }, 2] }
    }
  },
  { $sort: { gender: 1 } }
])
```

Upit 3 (N): Analiza potrošačkih navika po polu

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~7 sec	~0.1 sec	~98% brže
Pregledanih dokumenata	~1.2 miliona	~20,500	98.3% manje

```
db.transactions.aggregate([
  // FAZA 1: Brzo filtriranje samo transakcija registrovanih korisnika pomoću sparse indeksa.
  {
    $match: {
      "user.id": { $ne: null }
    }
  },
  // FAZA 2: Grupisanje po polu. Svi podaci (pol, ID korisnika) su direktno dostupni.
  {
    $group: {
      _id: "$user.gender",
      totalSpend: { $sum: "$amounts.final" },
      totalTransactions: { $sum: 1 },
      totalUsers: { $addToSet: "$user.id" }
    }
  },
  // FAZA 3: Projekcija i izračunavanje finalnih metrika.
  {
    $project: {
      _id: 0,
      gender: "$_id",
      total_spend: { $round: [ "$totalSpend", 2 ] },
      total_transactions: "$totalTransactions",
      number_of_users: { $size: "$totalUsers" },
      average_transaction_value: { $round: [ { $divide: [ "$totalSpend", "$totalTransactions" ] }, 2 ] },
      average_transactions_per_user: { $round: [ { $divide: [ "$totalTransactions", { $size: "$totalUsers" } ] }, 2 ] }
    }
  },
  {
    $sort: { gender: 1 }
  }
])
```

Zaključak: Sparse indeks na *user.id* je ključan - baza ignoriše 98.3% kolekcije i čita samo relevantne podatke, što dovodi do drastičnog smanjenja I/O operacija.

Upit 4 (N): Demografska analiza potrošnje

Svrha upita: Segmentira transakcije po polu i starosnim grupama korisnika, a zatim izračunava prosečnu vrednost transakcije za svaki segment.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  { $match: { user_id: { $ne: null } } },
  { $lookup: { from: "users", localField: "user_id", foreignField: "_id", as: "u" } },
  { $unwind: "$u" },
  {
    $group: {
      _id: {
        gender: "$u.gender",
        ageGroup: {
          $switch: {
            branches: [
              { case: { $lt: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 25] }, then: "1) < 25" },
              { case: { $and: [
                  { $gte: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 25] },
                  { $lt: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 35] }
                ]}, then: "2) 25-34" },
              { case: { $and: [
                  { $gte: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 35] },
                  { $lt: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 45] }
                ]}, then: "3) 35-44" },
              { case: { $gte: [{ $subtract: [2024, { $year: "$u.birthdate" }] }, 45] }, then: "4) 45+" }
            ],
            default: "Nepoznato"
          }
        }
      },
      avgAmount: { $avg: "$amounts.final" },
      count: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      gender: "$_id.gender",
      age_group: "$_id.ageGroup",
      average_transaction_value: { $round: ["$avgAmount", 2] },
      transaction_count: "$count"
    }
  },
  { $sort: { gender: 1, age_group: 1 } }
])
```

Upit 4 (N): Demografska analiza potrošnje

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~9 sec	~0.15 sec	~98% brže
Pregledanih dokumenata	~1.2 miliona	~20,500	98.3% manje

Zaključak: Kombinacija **sparse indeksa** i pre-kalkulisanog polja *user.age_group* eliminiše *COLLSCAN*, *\$lookup* i skupa izračunavanja.

```
db.transactions.aggregate([
  // FAZA 1: Brzo filtriranje transakcija registrovanih korisnika pomoću sparse indeksa.
  {
    $match: {
      "user.id": { $ne: null }
    }
  },
  // FAZA 2: Direktno grupisanje po pre-kalkulisanim i ugrađenim podacima.
  {
    $group: {
      _id: {
        gender: "$user.gender",
        ageGroup: "$user.age_group"
      },
      averageTransactionValue: { $avg: "$amounts.final" },
      transactionCount: { $sum: 1 }
    }
  },
  // FAZA 3: Finalna projekcija.
  {
    $project: {
      _id: 0,
      gender: "_id.gender",
      age_group: "_id.ageGroup",
      average_transaction_value: { $round: ["$averageTransactionValue", 2] },
      transaction_count: "$transactionCount"
    }
  },
  {
    $sort: {
      gender: 1,
      age_group: 1
    }
  }
])
```

Upit 5 (N): Analiza najvrednijih transakcija

Svrha upita: Prikazuje 5 najvrednijih transakcija u julu 2023. godine, sa svim detaljima.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  { $match: { created_at: { $gte: ISODate("2023-07-01T00:00:00.000Z"), $lt: ISODate("2023-08-01T00:00:00.000Z") } } },
  { $sort: { "amounts.final": -1 } },
  { $limit: 5 },
  { $lookup: { from: "stores", localField: "store_id", foreignField: "_id", as: "store_details" } },
  { $lookup: { from: "payment_methods", localField: "payment_method_id", foreignField: "_id", as: "payment_details" } },
  { $lookup: { from: "transaction_items", localField: "_id", foreignField: "transaction_id", as: "items_sold" } },
  { $unwind: "$items_sold" },
  { $lookup: { from: "menu_items", localField: "items_sold.menu_item_id", foreignField: "_id", as: "menu_item_details" } },
  { $group: {
      _id: "$_id",
      final_amount: { $first: "$amounts.final" },
      store_name: { $first: { $arrayElemAt: ["$store_details.name", 0] } },
      payment_method: { $first: { $arrayElemAt: ["$payment_details.method_name", 0] } },
      products_purchased: { $push: { $arrayElemAt: ["$menu_item_details.name", 0] } }
    }
  },
  { $sort: { final_amount: -1 } }
], { allowDiskUse: true })
```

Upit 5 (N): Analiza najvrednijih transakcija

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~15 sec	~2.3 sec	~85% brže
Pregledanih dokumenata	~1.2 miliona	samo 5	Drastično smanjenje

Zaključak: Eliminacija **četiri** uzastopna *\$lookup-a* i korišćenje indeksa znači da baza, nakon filtriranja, dohvata samo 5 potrebnih dokumenata sa diska..

```
db.transactions.aggregate([
    // FAZA 1: Brzo filtriranje po datumu
    {
        $match: {
            created_at: {
                $gte: ISODate("2023-07-01T00:00:00.000Z"),
                $lt: ISODate("2023-08-01T00:00:00.000Z")
            }
        },
        // FAZA 2: Sortiranje po finalnoj vrednosti
        {
            $sort: {
                "amounts.final": -1
            }
        },
        // FAZA 3: Ograničavanje na prvih 5 rezultata
        {
            $limit: 5
        },
        // FAZA 4: Projekcija i formatiranje finalnog izlaza
        {
            $project: {
                _id: 1,
                final_amount: "$amounts.final",
                store_name: "$store.name",
                payment_method: "$payment_method.name",
                products_purchased: "$items.name"
            }
        }
    }
])
```

Upit 1 (T): Plaćanje po tipu vaučera

Svrha upita: Prikazuje najčešće korišćene načine plaćanja za transakcije sa vaučerima.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  { $match: { voucher_id: { $ne: null } } },
  { $lookup: { from: "vouchers", localField: "voucher_id", foreignField: "_id", as: "voucherDetails" } },
  { $unwind: "$voucherDetails" },
  { $lookup: { from: "payment_methods", localField: "payment_method_id", foreignField: "_id", as: "paymentDetails" } },
  { $unwind: "$paymentDetails" },
  { $group: { _id: { voucherType: "$voucherDetails.discount_type", paymentMethod: "$paymentDetails.method_name" }, count: { $sum: 1 } } },
  { $sort: { "_id.voucherType": 1, count: -1 } },
  { $group: { _id: "$_id.voucherType", mostUsedPaymentMethods: { $push: { method: "$_id.paymentMethod", count: "$count" } } } },
  { $project: { _id: 0, voucher_discount_type: "$_id", top_payment_methods: { $slice: ["$mostUsedPaymentMethods", 3] } } },
  { $sort: { voucher_discount_type: 1 } }
])
```

Upit 1 (T): Plaćanje po tipu vaučera

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~8 sec	~0.3 sec	~96% brže
Pregledanih dokumenata	~1.2 miliona	~42,000	96.6% manje

Zaključak: Sparse indeks na *voucher.id* i denormalizacija su ključni za ubrzanje i smanjenje broja pročitanih dokumenata.

```
db.transactions.aggregate([
    // 1. Brzo filtriranje transakcija sa vaučerom (koristi sparse indeks)
    { $match: { "voucher": { $ne: null } } },
    // 2. Grupisanje po tipu vaučera i načinu plaćanja (NEMA lookup-a)
    {
        $group: {
            _id: {
                voucherType: "$voucher.discount_type",
                paymentMethod: "$payment_method.name"
            },
            count: { $sum: 1 }
        }
    },
    // 3. Sortiranje radi pripreme za pronaalaženje top 3
    { $sort: { "_id.voucherType": 1, count: -1 } },
    // 4. Ponovno grupisanje da se kreira niz metoda plaćanja
    {
        $group: {
            _id: "$_id.voucherType",
            paymentMethods: { $push: { method: "$_id.paymentMethod", count: "$count" } }
        }
    },
    // 5. Finalna projekcija koja uzima samo top 3
    {
        $project: {
            _id: 0,
            voucher_discount_type: "$_id",
            top_payment_methods: { $slice: ["$paymentMethods", 3] }
        }
    },
    { $sort: { voucher_discount_type: 1 } }
])
```

Upit 2 (T): Korisnici sa više metoda plaćanja

Svrha upita: Identificuje korisnike koji koriste više od jedne metode plaćanja.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  {
    $match: { user_id: { $ne: null } } // Fokusira se samo na registrovane korisnike
  },
  {
    $group: {
      _id: "$user_id",
      uniquePaymentMethods: { $addToSet: "$payment_method_id" }, // Sakuplja jedinstvene metode plaćanja
      totalTransactions: { $sum: 1 },
      totalSpend: { $sum: "$amounts.final" }
    }
  },
  {
    $match: { $expr: { $gte: [ { $size: "$uniquePaymentMethods" }, 2 ] } } // Filtrira korisnike sa 2+ metode plaćanja
  },
  {
    $lookup: { from: "users", localField: "_id", foreignField: "_id", as: "userDetails" }
  },
  {
    $unwind: "$userDetails"
  },
  {
    $project: {
      _id: 0,
      user_id: "$_id",
      user_gender: "$userDetails.gender",
      number_of_payment_methods: { $size: "$uniquePaymentMethods" },
      average_transaction_value: { $round: [ { $divide: [ "$totalSpend", "$totalTransactions" ] }, 2 ] },
      total_transactions: "$totalTransactions"
    }
  },
  {
    $sort: { number_of_payment_methods: -1, average_transaction_value: -1 }
  }
])
```

Upit 2 (T): Korisnici sa više metoda plaćanja

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~3 sec	~0.4 sec	~86% brže
Pregledanih dokumenata	~1.2 miliona	~20,500	98.3% manje

Zaključak: Ponovo, **sparse indeks** na *user.id* i denormalizacija eliminišu *COLLSCAN* i *\$lookup*, što drastično smanjuje I/O.

```
db.transactions.aggregate([
  // FAZA 1: Filtriranje samo transakcija registrovanih korisnika korišćenjem sparse indeksa.
  {
    $match: {
      "user.id": { $ne: null }
    }
  },
  // FAZA 2: Grupisanje po korisniku, prikupljanje podataka iz ugrađenih objekata.
  {
    $group: {
      _id: "$user.id",
      user_gender: { $first: "$user.gender" },
      uniquePaymentMethods: { $addToSet: "$payment_method.id" },
      totalTransactions: { $sum: 1 },
      totalSpend: { $sum: "$amounts.final" }
    }
  },
  // FAZA 3: Filtriranje korisnika sa 2 ili više metoda plaćanja.
  {
    $match: {
      $expr: { $gte: [{ $size: "$uniquePaymentMethods" }, 2] }
    }
  },
  // FAZA 4: Projekcija i formatiranje rezultata.
  {
    $project: {
      _id: 0,
      user_id: "$_id",
      user_gender: 1,
      number_of_payment_methods: { $size: "$uniquePaymentMethods" },
      average_transaction_value: { $round: [{ $divide: ["$totalSpend", "$totalTransactions"] }, 2] },
      total_transactions: "$totalTransactions"
    }
  },
  // FAZA 5: Sortiranje.
  {
    $sort: {
      number_of_payment_methods: -1,
      average_transaction_value: -1
    }
  }
])
```

Upit 3 (T): Analiza "Vikend Gužve" po satima

Svrha upita: Analizira ukupan prihod za svaki sat tokom vikenda (subota i nedelja).

Kod (V1):

```
db.getCollection('transactions').aggregate([
    // FAZA 1: Za svaki dokument, izračunaj dan u nedelji i sat.
    {
        $project: {
            dayOfWeek: { $dayOfweek: "$created_at" },
            hourOfDay: { $hour: "$created_at" },
            finalAmount: "$amounts.final"
        }
    },
    // FAZA 2: Filtriraj i zadrži samo one koji su se desili subotom ili nedeljom.
    {
        $match: {
            dayOfWeek: { $in: [6, 0] }
        }
    },
    // FAZA 3: Grupiši po satu i sumiraj prihod.
    {
        $group: {
            _id: "$hourOfDay",
            totalRevenue: { $sum: "$finalAmount" }
        }
    },
    // FAZA 4: Sortiraj po satu.
    {
        $sort: { _id: 1 }
    },
    // FAZA 5: Formatiraj izlaz.
    {
        $project: {
            _id: 0,
            hour: "$_id",
            total_revenue: { $round: ["$totalRevenue", 2] }
        }
    }
])
```

Upit 3 (T): Analiza "Vikend Gužve" po satima

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~11 sec	~1.5 sec	~86% brže
Pregledanih dokumenata	~1.2 miliona	~360,000	71% manje

Zaključak: Korišćenje indeksa na pre-kalkulisanom polju `dayOfWeek` omogućava efikasno "rano" filtriranje, ignorajući 71% kolekcije.

```
// Napomena: Korišćeni su dani 6 (Subota) i 7 (Nedelja) jer Python-ov isowEEKDAY()
// generiše te vrednosti, koje su sada sačuvane u bazi.
db.transactions.aggregate([
    // FAZA 1: Brzo filtriranje samo dana vikenda pomoću indeksa.
    {
        $match: {
            "createdAtDetails.dayOfWeek": { $in: [6, 7] }
        }
    },
    // FAZA 2: Grupisanje po pre-kalkulisanom satu i sumiranje prihoda.
    {
        $group: {
            _id: "$createdAtDetails.hour",
            totalRevenue: { $sum: "$amounts.final" }
        }
    },
    // FAZA 3: Sortiranje po satu.
    {
        $sort: { _id: 1 }
    },
    // FAZA 4: Formatiranje izlaza.
    {
        $project: {
            _id: 0,
            hour: "$_id",
            total_revenue: { $round: ["$totalRevenue", 2] }
        }
    }
])
```

Upit 4 (T): Prosečan popust po starosnoj grupi

Svrha upita: Izračunava prosečan iznos popusta za svaku starosnu grupu, ali samo za transakcije u kojima je korišćen vaučer.

Kod (V1):

```
db.getCollection('transactions').aggregate([
  { $match: { voucher_id: { $ne: null } } }, // FAZA 1: samo transakcije sa vaučerom
  { $lookup: { from: "users", localField: "user_id", foreignField: "_id", as: "userDetails" } },
  { $unwind: "$userDetails" },
  { $addFields: { age: { $subtract: [2024, { $year: "$userDetails.birthdate" }] } } }, // FAZA 3: izračunaj starost
  {
    $addFields: {
      ageGroup: {
        $switch: {
          branches: [
            { case: { $lt: ["$age", 25] }, then: "1) < 25" },
            { case: { $and: [{ $gte: ["$age", 25] }, { $lt: ["$age", 35] }] }, then: "2) 25-34" },
            { case: { $and: [{ $gte: ["$age", 35] }, { $lt: ["$age", 45] }] }, then: "3) 35-44" },
            { case: { $gte: ["$age", 45] }, then: "4) 45+" }
          ],
          default: "Nepoznato"
        }
      }
    }
  },
  { $group: { _id: "$ageGroup", averageDiscount: { $avg: "$amounts.discount" } } }, // FAZA 5
  { $project: { _id: 0, age_group: "$_id", average_discount_applied: { $round: ["$averageDiscount", 2] } } }, // FAZA 6
  { $sort: { age_group: 1 } }
])
```

Upit 4 (T): Prosečan popust po starosnoj grupi

Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~11 sec	~4 sec	~64% brže
Pregledanih dokumenata	~1.2 miliona	~1.2 miliona	Isto

```
db.transactions.aggregate([
  // FAZA 1: Filtriraj transakcije koje imaju vaučer I definisani starosnu grupu
  {
    $match: {
      voucher: { $ne: null },
      "user.age_group": { $ne: null }
    }
  },
  // FAZA 2: Grupiši po pre-kalkulisanoj starosnoj grupi i izračunaj prosek popusta
  {
    $group: {
      _id: "$user.age_group",
      averageDiscount: { $avg: "$amounts.discount" }
    }
  },
  // FAZA 3: Formatiraj finalni izlaz
  {
    $project: {
      _id: 0,
      age_group: "$_id",
      average_discount_applied: { $round: ["$averageDiscount", 2] }
    }
  },
  { $sort: { age_group: 1 } }
])
```

Zaključak: Iako *COLLSCAN* ostaje, eliminacija *\$lookup-a* i izračunavanja "u letu" značajno smanjuje CPU opterećenje i ubrzava upit.

Upit 5 (T): Najprofitabilniji dani za "Americano"

Svrha upita: Pronalazi 5 dana sa najvećim prihodom od prodaje proizvoda "Americano".

Kod (V1):

```
db.getCollection('menu_items').aggregate([
  { $match: { name: "Americano" } }, // FAZA 1: pronađi proizvod
  { $lookup: { from: "transaction_items", localField: "_id", foreignField: "menu_item_id", as: "sales" } },
  { $unwind: "$sales" },
  { $lookup: { from: "transactions", localField: "sales.transaction_id", foreignField: "_id", as: "transactionDetails" } },
  { $unwind: "$transactionDetails" },
  { $project: { saleDate: { $dateToString: { format: "%Y-%m-%d", date: "$transactionDetails.created_at" } }, subtotal: "$sales.subtotal" } },
  { $group: { _id: "$saleDate", dailyRevenue: { $sum: "$subtotal" } } },
  { $sort: { dailyRevenue: -1 } },
  { $limit: 5 },
  { $project: { _id: 0, date: "$_id", total_revenue_for_americano: { $round: ["$dailyRevenue", 2] } } }
])
```

Upit 5 (T): Najprofitabilniji dani za "Americano"

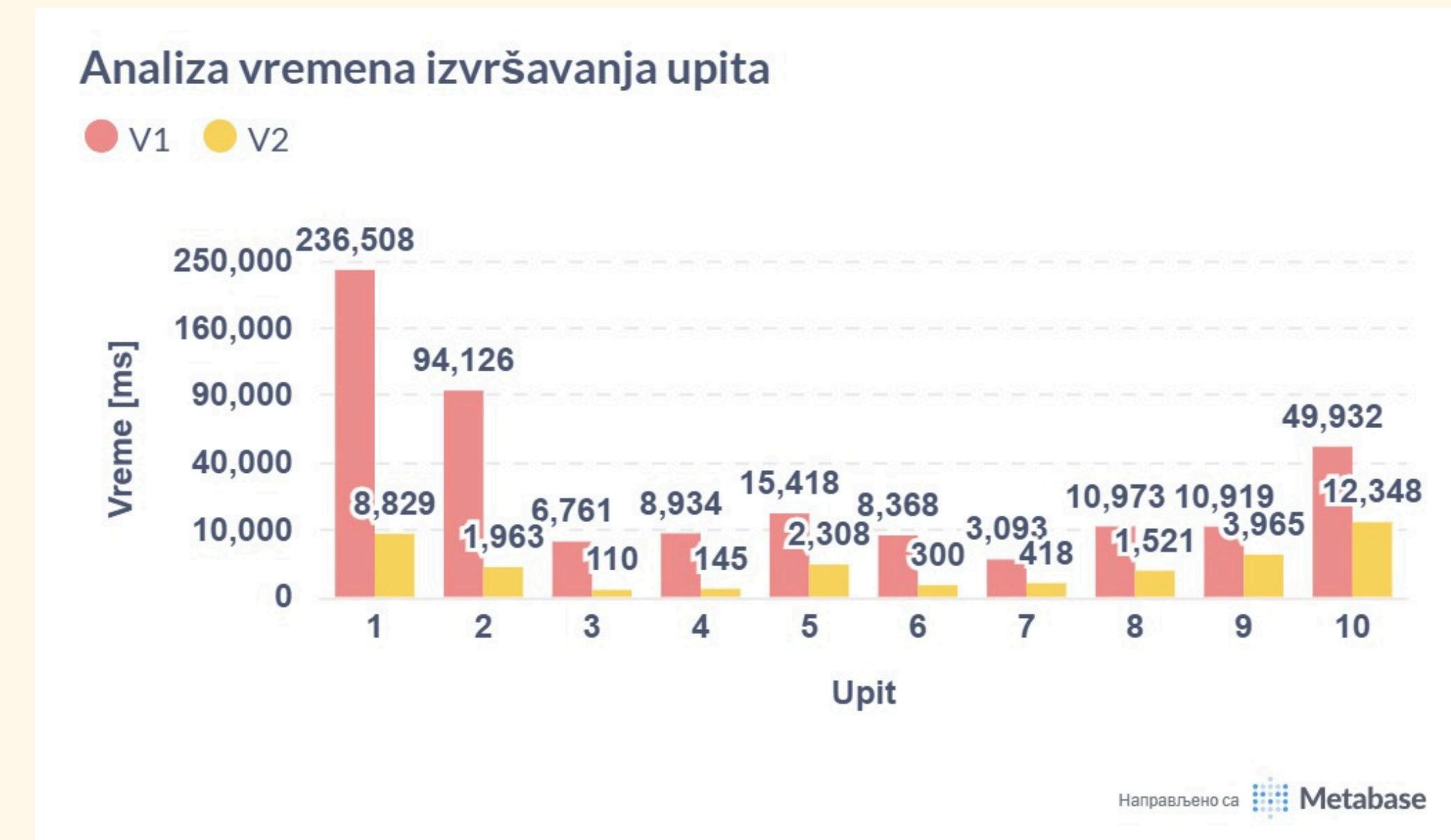
Kod (V2 - Optimizovano):

Metrika	V1 šema	V2 šema	Promena
Vreme	~50 sec	~12 sec	~75% brže
Pregledanih dokumenata	~5.2 miliona	~1.2 miliona	76% manje

Zaključak: Eliminacija dva masivna `$lookup`-a ugradnjom items niza je ključni dobitak, smanjujući ukupan broj pročitanih dokumenata za preko 4 miliona.

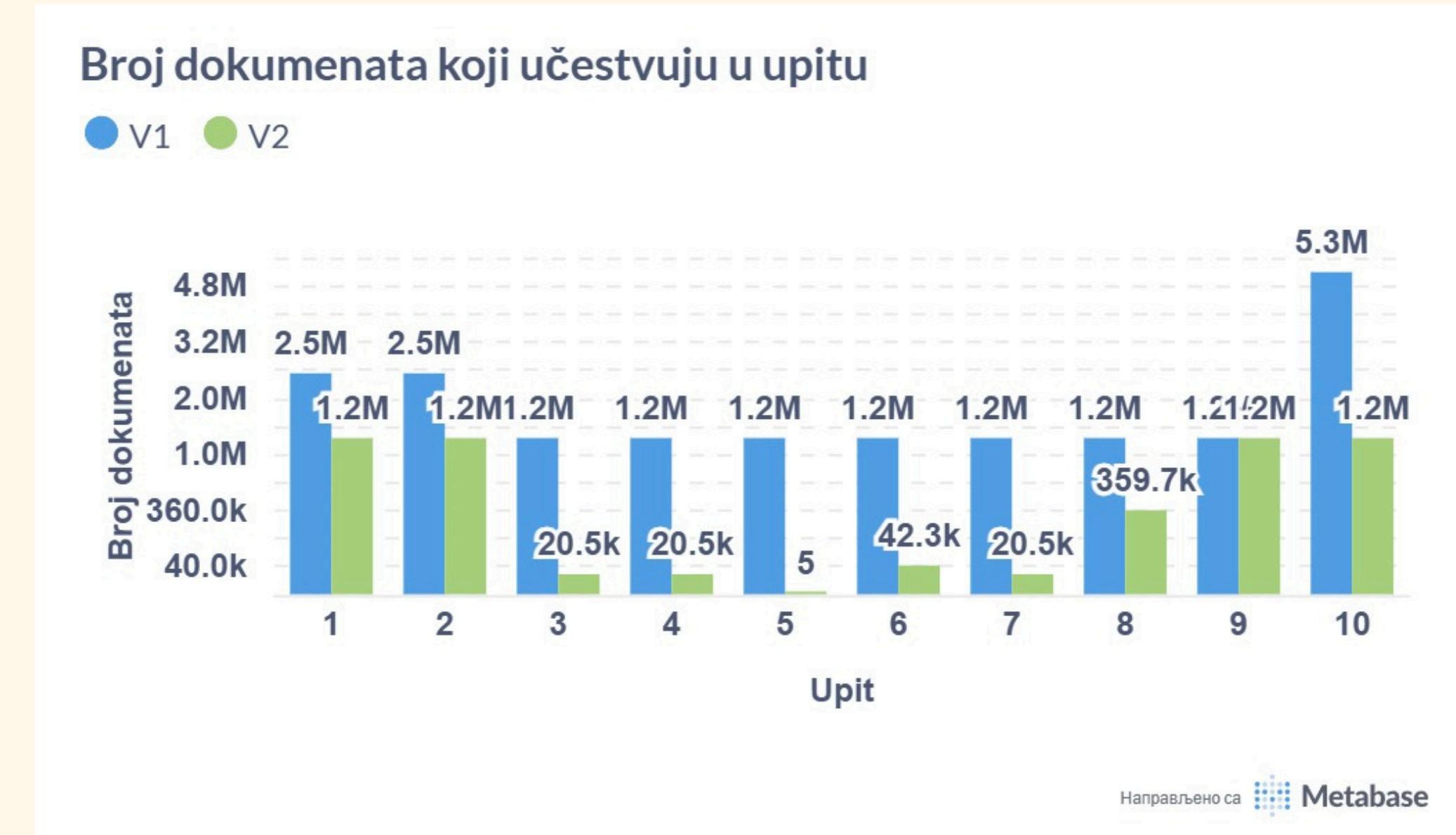
```
db.transactions.aggregate([
  // FAZA 1: Brzo filtriraj samo transakcije koje sadrže "Americano"
  {
    $match: { "items.name": "Americano" }
  },
  // FAZA 2: "Rasturi" niz stavki da bi se radilo sa pojedinačnim prodajama
  {
    $unwind: "$items"
  },
  // FAZA 3: Filtriraj ponovo da bi se izolovale samo "Americano" stavke
  {
    $match: { "items.name": "Americano" }
  },
  // FAZA 4: Formatiraj datum i izdvoj prihod po stavci
  {
    $project: {
      saleDate: { $dateToString: { format: "%Y-%m-%d", date: "$created_at" } },
      subtotal: "$items.subtotal"
    }
  },
  // FAZA 5: Grupiši po datumu da bi se dobio ukupan dnevni prihod
  {
    $group: {
      _id: "$saleDate",
      dailyRevenue: { $sum: "$subtotal" }
    }
  },
  // FAZA 6: Sortiraj i ograniči na top 5
  { $sort: { dailyRevenue: -1 } },
  { $limit: 5 },
  // FAZA 7: Formatiraj finalni izlaz
  {
    $project: {
      _id: 0,
      date: "$_id",
      total_revenue_for_americano: { $round: ["$dailyRevenue", 2] }
    }
  }
])
```

Poređenje performansi - Vreme izvršavanja



Grafikon nedvosmisleno pokazuje drastično smanjenje vremena potrebnog za izvršavanje upita u V2 šemi. Ubrzanja se kreću od 64% do preko 98%, pretvarajući spore upite u analitičke alate upotrebljive u realnom vremenu.

Poređenje performansi - Broj dokumenata



V1 šema ima daleko veći ukupan broj dokumenata. V2 šema ima značajno manje dokumenata, ali je svaki dokument veći. Za analitiku, manji broj dokumenata znači **manje I/O operacija čitanja sa diska**, što je ključni razlog za ubrzanje.

Zaključak

- **Dokazali smo da modelovanje podataka ima ključan uticaj na performanse.**
- **V1 (Normalizovana šema):** Pogodna za transakcione sisteme sa velikim brojem upisa, ali prespora za kompleksnu analizu zbog zavisnosti od $\$lookup$ -a.
- **V2 (Denormalizovana šema):** Superiorno rešenje za analitičke sisteme. Primenom dizajn šablonu (Embedding, Computed Pattern) i pravilnim indeksiranjem (Sparse, Multikey), postigli smo drastična ubrzanja.
- **Kompromis:** V2 šema zahteva više prostora po dokumentu i kompleksniju logiku pri upisu, ali je dobitak u brzini čitanja i analize višestruko veći i opravdava ovaj pristup.

Hvala na pažnji!

Pitanja?

