

## 1. Tehtävä

## 2. Tehtävä

Hae Harppu-nimisen lemmikin tiedot. Näytä kaikki kentät.

```
>_MONGOSH
<
> db.rekisteri.find({nimi:"Harppu"})
<
> db.lemmikit.find({nimi:"Harppu"})
< { _id: ObjectId("62681be8ff8e87faf2766ec2"),
  nimi: 'Harppu',
  laji: 'Koira',
  rotu: 'Saksanpaimenkoira',
  syntaika: 2021-10-17T00:00:00.000Z,
  kasvattaja:
    { nimi: 'Kasvatus Kananen',
      yhteys: 'Kalevi Kananen',
      kunta: 'Laukaa' },
  rokotukset:
    [ { nimi: 'Parvo', annoskoko: 200 },
      { nimi: 'Penikkatauti', annoskoko: 500 },
      { nimi: 'Rabies', annoskoko: 100 } ],
  omistaja: { nimi: 'Tiro Immonen', kunta: 'Helsinki' },
  nayttely: false,
  laakarikaynnit: 4 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

Hae kaikki lemmikit, joiden omistaja asuu Hankasalmissa. Näytä tuloksissa vain lemmikin nimi ja omistajan tiedot.

```
> db.lemmikit.find({"omistaja.kunta":"Hankasalmi"},{nimi:1, omistaja:1})
< { _id: ObjectId("62681be8ff8e87faf2766ebf"),
  nimi: 'Ransu',
  omistaja: { nimi: 'Sami Saarikko', kunta: 'Hankasalmi' } }
{ _id: ObjectId("62681be8ff8e87faf2766ec0"),
  nimi: 'Rekku',
  omistaja: { nimi: 'Martti Manninen', kunta: 'Hankasalmi' } }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

Hae lemmikit, jotka ovat saaneet Penikkatauti-rokotuksen ja joiden kasvattajan kunta on joko Laukaa tai Liperi. Näytä tuloksissa lemmikin nimi, kasvattajan tiedot ja rokotustiedot.

```
it.find({$and:[{"rokotukset.nimi":"Penikkatauti"},{$or:[{"kasvattaja.kunta":"Laukaa"}, {"kasvattaja.kunta":"Liperi"}]}]}, {nimi:1, kasvattaja:1, rokotukset:1})
```

```

> db.lemmikit.find({$and:[{"rokotukset.nimi":"Penikkatauti"},{$or:[{"kasvattaja.kunta":"Laukaa"}, {"kasvattaja.kunta":"Liperi"}]}]}, (nimi:1, kasvattaja:1, rokot
< { _id: ObjectId("62681be8ff8e87faf2766ec1"),
  nimi: 'Haukku',
  kasvattaja:
    { nimi: 'Kasvatus Kananen',
      yhteys: 'Kalevi Kananen',
      kunta: 'Laukaa' },
  rokotukset:
    [ { nimi: 'Parvo', annoskoko: 200 },
      { nimi: 'Penikkatauti', annoskoko: 500 },
      { nimi: 'Rabies', annoskoko: 100 } ] ]
{ _id: ObjectId("62681be8ff8e87faf2766ec2"),
  nimi: 'Harppu',
  kasvattaja:
    { nimi: 'Kasvatus Kananen',
      yhteys: 'Kalevi Kananen',
      kunta: 'Laukaa' },
  rokotukset:
    [ { nimi: 'Parvo', annoskoko: 200 },
      { nimi: 'Penikkatauti', annoskoko: 500 },
      { nimi: 'Rabies', annoskoko: 100 } ] ]
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>

```

Lisää yhdellä lauseella lemmikkien laakarikaynnit-avaimen arvoon 1. Mikäli lemmikillä ei ole laakarikaynnit-avainta, ei dokumenttia saa päivittää.

```

> db.lemmikit.updateMany({"laakarikaynnit":true},{ $inc:{"laakarikaynnit":1}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0 }

> db.lemmikit.updateMany({"nayttely":true},{ $inc:{"laakarikaynnit":1}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0 }

Atlas atlas-y4vje7-shard-0 [primary] rekisteri>

```

### 3. Tehtävä

a) Hae kuinka monta yksilöä kutakin lemmikkilajia lemmikit-collectionissa on?

```
> db.lemmikit.aggregate([{$group:{_id:"$laji", "maara":{$sum:1}}}] );
< { _id: 'Kissa', maara: 2 }
   { _id: 'Koira', maara: 7 }
```

b) Lisää \$project-vaihe, jonka jälkeen lajitieto näkyy avaimessa "laji" ja \_id-avainta ei näytetä lainkaan.

```
> db.lemmikit.aggregate([
  {
    $group:{_id:"$laji", "maara":{$sum:1}}
  },
  {
    $project:{_id:0, laji:"$_id", maara:1}
  }
]);
< { maara: 7, laji: 'Koira' }
   { maara: 2, laji: 'Kissa' }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

c) Nimeä maara-sarake \$project-vaiheessa nimelle lkm (älä siis muuta \$group-vaiheen koodia, vaan hoida asia \$project-vaiheessa).

```
> db.lemmikit.aggregate([
  {
    $group:{_id:"$laji", "maara":{$sum:1}}
  },
  {
    $project:{_id:0, laji:"$_id", lkm:"$maara"}
  }
]);
< { laji: 'Koira', lkm: 7 }
   { laji: 'Kissa', lkm: 2 }
```

#### 4.Tehtävä

a) Listaa lemmikkien määrät kasvattajan kotikunnan mukaisesti.

```
> db.lemmikit.aggregate([
  {$group: {_id: "$kasvattaja.kunta", lkm: {$sum: 1}}}
]);
< { _id: 'Kuopio', lkm: 1 }
  { _id: 'Muurame', lkm: 1 }
  { _id: 'Uurainen', lkm: 1 }
  { _id: 'Toivakka', lkm: 2 }
  { _id: 'Liperi', lkm: 1 }
  { _id: 'Keuruu', lkm: 1 }
  { _id: 'Laukaa', lkm: 2 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri >
```

b) Lajittele tulosjoukko ensisijaisesti lukumäärän mukaan laskevaan järjestykseen ja toissijaisesti kunnan mukaan nousevaan järjestykseen.

```
> db.lemmikit.aggregate([
  {$group: {_id: "$kasvattaja.kunta", lkm: {$sum: 1}}},
  {$sort: {lkm: -1}}
]);
< { _id: 'Laukaa', lkm: 2 }
  { _id: 'Toivakka', lkm: 2 }
  { _id: 'Muurame', lkm: 1 }
  { _id: 'Kuopio', lkm: 1 }
  { _id: 'Liperi', lkm: 1 }
  { _id: 'Keuruu', lkm: 1 }
  { _id: 'Uurainen', lkm: 1 }
```

```
> db.lemmikit.aggregate([
  {$group: {_id: "$kasvattaja.kunta", lkm: {$sum: 1}}},
  {$sort: {_id: 1}}
]);
< { _id: 'Keuruu', lkm: 1 }
  { _id: 'Kuopio', lkm: 1 }
  { _id: 'Laukaa', lkm: 2 }
  { _id: 'Liperi', lkm: 1 }
  { _id: 'Muurame', lkm: 1 }
  { _id: 'Toivakka', lkm: 2 }
  { _id: 'Uurainen', lkm: 1 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri >
```

c) Näytä tulosjoukosta sijoilla 3-5 olevat kunnat.

```
> db.lemmikit.aggregate([
  {$group: {_id: "$kasvattaja.kunta", lkm: {$sum: 1}}},
  {$sort: {lkm: 1, _id: 1}},
  {"$limit": 3}
]);
< { _id: 'Keuruu', lkm: 1 }
  { _id: 'Kuopio', lkm: 1 }
  { _id: 'Liperi', lkm: 1 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

## 5.Tehtävä

a) Hae lemmikkien saamat rokotusannosten yhteissummat rokotteittain.

```
> db.lemmikit.aggregate([
  {$unwind:"$rokotukset"},
  {$group:{_id:"$rokotukset.nimi", annoskoko:{$sum:"$rokotukset.annoskoko"}}}
]);
< { _id: 'Kolmoisrokote', annoskoko: 0 }
  { _id: 'Parvo', annoskoko: 1000 }
  { _id: 'Rabies', annoskoko: 300 }
  { _id: 'Penikkatauti', annoskoko: 1500 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

b) Näytä vain rokotteet, joita on annettu vähintään 500 yksikköä.

```
> db.lemmikit.aggregate([
  {$unwind:"$rokotukset"},
  {$group:{_id:"$rokotukset.nimi", annoskoko:{$sum:"$rokotukset.annoskoko"}}},
  {$match:{annoskoko:{$gt:500}}}
]);
< { _id: 'Penikkatauti', annoskoko: 1500 }
  { _id: 'Parvo', annoskoko: 1000 }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

c) Nimeä edellisen \_id-sarake nimelle "rokote" ja muunna rokotteen nimi suuraakkosille.

```
> db.lemmikit.aggregate([
  {$unwind:"$rokotukset"},
  {$group:{_id:"$rokotukset.nimi", annoskoko:{$sum:"$rokotukset.annoskoko"}}},
  {$match:{annoskoko:{$gt:500}}},
  {$project:{yhteensa:"$annoskoko", rokote:{$toUpper:"$_id"}, _id:0}}
]);
< { yhteensa: 1000, rokote: 'PARVO' }
  { yhteensa: 1500, rokote: 'PENIKKATAUTI' }
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

## 6.Tehtävä

Selvitä aluksi, millaisia indeksejä lemmikit-collectioniin on määritelty.

```
> db.lemmikit.getIndexes();  
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]  
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

Tee sitten kysely, joka näyttää kaikki tiedot dokumenteista, joissa kasvattajan kunta on Toivakka.

```
> db.lemmikit.find({"kasvattaja.kunta":"Toivakka"}).explain(true);  
< { explainVersion: '1',  
  queryPlanner:  
    { namespace: 'rekisteri.lemmikit',  
      indexFilterSet: false,  
      parsedQuery: { 'kasvattaja.kunta': { '$eq': 'Toivakka' } },  
      maxIndexedOrSolutionsReached: false,  
      maxIndexedAndSolutionsReached: false,  
      maxScansToExplodeReached: false,  
      winningPlan:  
        { stage: 'COLLSCAN',  
          filter: { 'kasvattaja.kunta': { '$eq': 'Toivakka' } },  
          direction: 'forward' },  
          rejectedPlans: [] },  
      executionStats:  
        { executionSuccess: true,  
          nReturned: 2,  
          executionTimeMillis: 0,  
          totalKeysExamined: 0,  
          totalDocsExamined: 9,  
          executionStages:  
            { stage: 'COLLSCAN',  
              filter: { 'kasvattaja.kunta': { '$eq': 'Toivakka' } },  
              nReturned: 2,  
              executionTimeMillisEstimate: 1,  
              works: 11,  
              advanced: 2,  
              needTime: 8,  
              needYield: 0,  
              saveState: 0,  
              restoreState: 0,  
              isEOF: 1,  
              direction: 'forward',  
              docsExamined: 9 },  
              allPlansExecution: [] },  
          command:  
            { find: 'lemmikit',  
              filter: { 'kasvattaja.kunta': 'Toivakka' },  
              '$db': 'rekisteri' },  
          serverInfo:  
            { host: 'cluster0-shard-00-01.5qkjo.mongodb.net',  
              port: 27017,  
              version: '5.0.8',  
              gitVersion: 'c87e1c23421bf79614baf500fda6622bd90f674e' },  
          serverParameters:  
            { internalQueryFacetBufferSizeBytes: 104857600,  
              internalQueryFacetMaxOutputDocSizeBytes: 104857600,  
              internalLookupStageIntermediateDocumentMaxSizeBytes: 16793600,  
              internalDocumentSourceGroupMaxMemoryBytes: 104857600,  
              internalQueryMaxBlockingSortMemoryUsageBytes: 33554432,  
              internalQueryProhibitBlockingMergeOnMongoS: 0,  
              internalQueryMaxAddToSetBytes: 104857600,  
              internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },  
          ok: 1,  
          '$clusterTime':  
            { clusterTime: Timestamp({ t: 1651599320, i: 13 } ),  
              signature:  
                { hash: Binary(Buffer.from("3299431d98679759e134d2a43282635bcd8cde82", "hex"), 0),  
                  keyId: 7032297490162385000 } },  
          operationTime: Timestamp({ t: 1651599320, i: 13 } )  
        }  
      }  
    }  
  }  
Atlas atlas-y4vje7-shard-0 [primary] rekisteri> db.lemmikit.find({"kasvattaja.kunta":"Toivakka"});
```

Tarkastele, kuinka Mongo suorittaa haun.

Pystyykö Mongo hyödyntämään mitään indeksiä?

```
> db.lemmikit.createIndex({"kasvattaja.kunta":1});
< 'kasvattaja.kunta_1'
> db.lemmikit.getIndexes();
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { 'kasvattaja.kunta': 1 }, name: 'kasvattaja.kunta_1' }
]
Atlas atlas-y4vje7-shard-0 [primary] rekisteri>
```

```
executionStats:
  { executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 1,
    totalKeysExamined: 2,
    totalDocsExamined: 2,
    executionStages:
      { stage: 'FETCH',
        nReturned: 2,
        executionTimeMillisEstimate: 0,
        works: 3,
        advanced: 2,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        docsExamined: 2,
        alreadyHasObj: 0,
        inputStage:
          { stage: 'IXSCAN',
```



## 7.Tehtävä

Muokkaa hakuasi siten, että tulosjoukosta palautetaan vain lemmikin nimi ja kasvattajan kotikunta.

```
> db.lemmikit.find({"kasvattaja.kunta":"Toivakka"}, {"kasvattaja.kunta":1, nimi:1, _id:0});  
< { nimi: 'Tossu', kasvattaja: { kunta: 'Toivakka' } }  
  { nimi: 'Topsu', kasvattaja: { kunta: 'Toivakka' } }  
Atlas-Atlas-Mongo-7-shard-0 [primary]> use test
```

Tarkastele, käytetäänkö edellisen tehtävän indeksä ja pystyykö haku saamaan kaikki tiedot suoraan indeksistä.

Kyllä, haku pystyy edelleen saamaan kaikki tiedot suoraan indeksistä

```
executionStats:  
  { executionSuccess: true,  
    nReturned: 2,  
    executionTimeMillis: 0,  
    totalKeysExamined: 2,  
    totalDocsExamined: 2,  
    executionStages:  
      { stage: 'PROJECTION_DEFAULT',  
        nReturned: 2,  
        executionTimeMillisEstimate: 0,  
        works: 3,  
        advanced: 2,  
        needTime: 0,  
        needYield: 0,  
        saveState: 0,  
        restoreState: 0,  
        isEOF: 1,  
        transformBy: { 'kasvattaja.kunta': 1, nimi: 1 },  
        inputStage:  
          { stage: 'FETCH',  
            nReturned: 2,
```

Lisää indeksi, jota käyttäen koko tulos saadaan indeksistä (totalDocsExamined: 0).

Epäonnistui (((