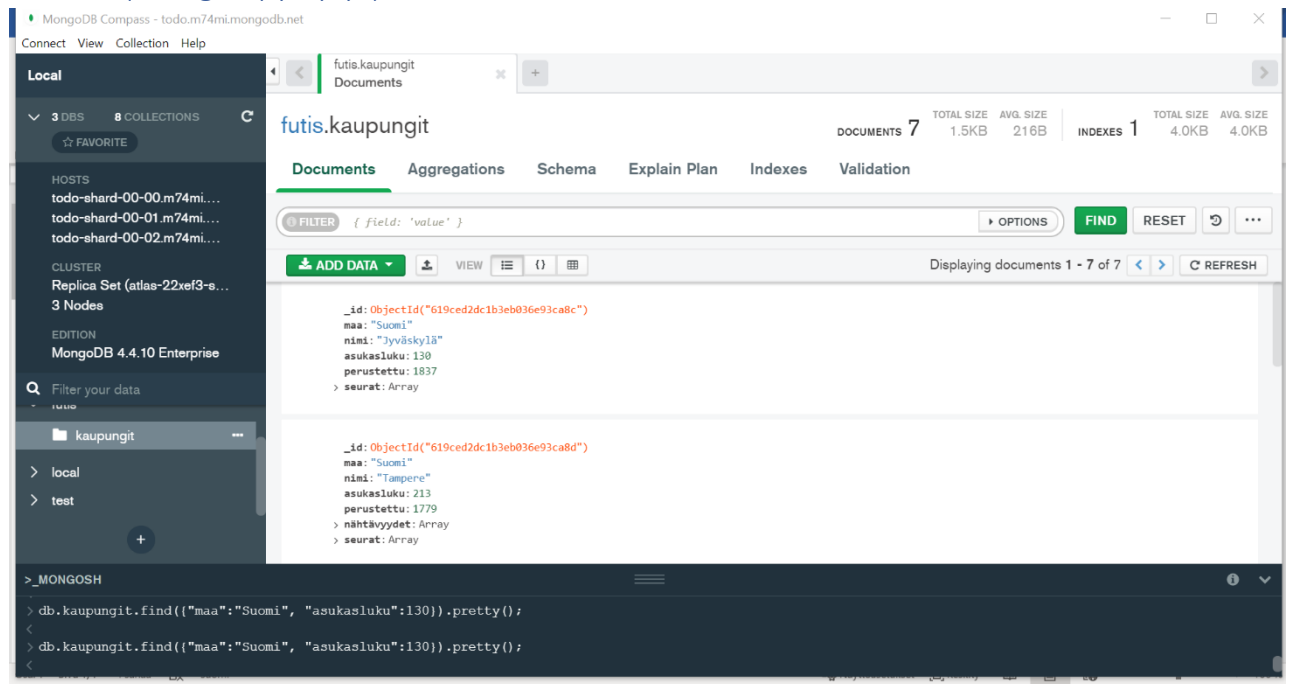


1. tehtävä (Mongon pystytys)



2. tehtävä (tutustuminen Mongo-palveluun ja MongoExpressiin)

tehty!

3. tehtävä (suunnittelu: levy-case)

levy.json

```
[
{
  "Yhtiönnimi": "Yhtiö1",
  "Levynnimi":
  [
    {
      "nimi":"Koodinvääntäjä",
      "julkaisuvuosi":"2013",
      "kappale":[
        {"kappalenimi": "Koodaa kuin mies", "säveltäjä": "Tommi", "sanoittaja": "Tommi"},
        {"kappalenimi": "Ikuinen silmukka", "säveltäjä": "Jakke", "sanoittaja": "Jakke"},
        {"kappalenimi": "Iisi ostaa verkosta", "säveltäjä": "Jakke", "sanoittaja": "Niko"}
      ],
      "kategoria":[
        {"kategoriannimi": "huumorimusiikki"},
        {"kategoriannimi": "progressiivinen hälinä"}
      ]
    }
  ]
}
```

```

    },
    {
      "nimi": "Syntax Error",
      "julkaisuvuosi": "2019",
      "kappale": [
        { "kappalenimi": "Pistetään pistetään omenaa",
          "säveltäjä": "Jakke", "sanoittaja": "Jakke" },
        { "kappalenimi": "Testaajahumppa", "säveltäjä": "Niko",
          "sanoittaja": "Jakke" }
      ],
      "kategoria": [
        { "kategoriannimi": "huumorimusiikki" },
        { "kategoriannimi": "progressiivinen hälinä" }
      ]
    }
  ]
}
]

```

user.json

```

[
  {
    "ktunnus": "terhoz",
    "salasana": "password123",
    "nimi": "Terho"
  }
]

```

4.tehtävä (suunnittelu: IKK-case)

```

[
  {
    "Nimi": "IKK",
    "Sovellus": [
      {
        "nimi": "PingPong",
        "luokka": "pele",
        "tagit": [
          "3D",
          "urheilu"
        ],
        "group": [
          {
            "membername": "Keijo-Antero",
            "membersurname": "Kuronen",
            "tehtävä": "tekniikka ja testaus"
          }
        ]
      }
    ]
  }
]

```

```

    },
    {
      "membername": "Sanna",
      "membersurname": "Suomalainen",
      "tehtävä": "sisalto ja Scrum Master"
    }
  ],
  "meta": [
    {"katsottu": "100",
      "ladattu": "21"}
  ]
},
{
  "nimi": "Tiko-Karate",
  "luokka": "peli",
  "tagit": [
    "2D",
    "urheilu",
    "Tappelu"
  ],
  "group": [
    {
      "membername": "Raimo",
      "membersurname": "Ruuskanen",
      "tehtävä": "tekniikka ja sisalto"
    },
    {
      "membername": "Pentti",
      "membersurname": "Puustinen",
      "tehtävä": "sisalto ja Scrum Master"
    }
  ],
  "meta": [
    {"katsottu": "55",
      "ladattu": "23"}
  ]
},
{
  "nimi": "Naamakirja",
  "luokka": "hyötysovellus",
  "tagit": [
    "Sosiaalinen media",
    "Yhteisopalvelu"
  ],
  "group": [
    {
      "membername": "Ilmari",
      "membersurname": "Innanen-Immonen",

```

```

        "tehtävä": "tekniikka ja business"
    },
    {
        "membername": "Raija-Irmeli",
        "membersurname": "Rantanen",
        "tehtävä": "tekniikka, sisälto, Scrum Master"
    }
],
"meta": [
    { "katsottu": "133",
      "ladattu": "16" }
]
}
]

```

5.tehtävä (importointi)

MongoDB Compass - todo.m74mi.mongodb.net

Connect View Collection Help

Local

3 DBS 8 COLLECTIONS

☆ FAVORITE

HOSTS

- todo-shard-00-00.m74mi...
- todo-shard-00-01.m74mi...
- todo-shard-00-02.m74mi...

CLUSTER

Replica Set (atlas-22xf3-s...

3 Nodes

EDITION

MongoDB 4.4.10 Enterprise

Filter your data

kaupungit

local

test

+

futis.kaupungit Documents

DOCUMENTS 7 TOTAL SIZE 1.5KB AVG. SIZE 216B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 7 of 7 REFRESH

```

{
  "_id": ObjectId("619ced2dc1b3eb036e93ca8c"),
  "maa": "Suomi",
  "nimi": "Jyväskylä",
  "asukasluku": 130,
  "perustettu": 1837,
  "seurat": Array
}

{
  "_id": ObjectId("619ced2dc1b3eb036e93ca8d"),
  "maa": "Suomi",
  "nimi": "Tampere",
  "asukasluku": 213,
  "perustettu": 1779,
  "nähtävyydet": Array,
  "seurat": Array
}

```

> _MONGOSH

```

> db.kaupungit.find({"maa": "Suomi", "asukasluku": 130}).pretty();
<
> db.kaupungit.find({"maa": "Suomi", "asukasluku": 130}).pretty();
<

```

6.tehtävä (hakeminen)

1. Hae kaikki dokumentit (find) ja näytä tulos helpommin luettavassa muodossa (pretty). Tutki ja kokeile myös findOne-metodia. Mikä käytännön ero näillä on?

```
> db.kaupungit.find().pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca91"),
  maa: 'Espanja',
  nimi: 'Barcelona',
  asukasluku: 1621,
  'nähtävyydet': [ 'La Rambla', 'Tuomiokirkko' ],
  seurat:
    [ { nimi: 'FC Barcelona', stadion: 'Camp Nou', sarja: 'La Liga' },
      { nimi: 'RCD Espanyol',
        stadion: 'Estadi Cornellà-El Prat',
        sarja: 'La Liga' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca92"),
  maa: 'Espanja',
  nimi: 'Sevilla',
  asukasluku: 703,
  seurat:
    [ { nimi: 'Real Betis',
        stadion: 'Estadio Benito Villamarín',
```

findOne palauttaa yhden dokumentin nätisti formatoidu ja ei toimi pretty() yhdessä

```
> db.kaupungit.findOne();
< { _id: ObjectId("619ced2dc1b3eb036e93ca91"),
  maa: 'Espanja',
  nimi: 'Barcelona',
  asukasluku: 1621,
  'nähtävyydet': [ 'La Rambla', 'Tuomiokirkko' ],
  seurat:
    [ { nimi: 'FC Barcelona', stadion: 'Camp Nou', sarja: 'La Liga' },
      { nimi: 'RCD Espanyol',
        stadion: 'Estadi Cornellà-El Prat',
        sarja: 'La Liga' } ] }
```

2. Hae kaikki espanjalaiset kaupungit. Älä näytä tuloksessa _id-kenttää eikä seurat-taulukkoa.

```
> db.kaupungit.find({maa:"Espanja"},{_id:0,seurat:0});
< { maa: 'Espanja',
    nimi: 'Barcelona',
    asukasluku: 1621,
    'nähtävyydet': [ 'La Rambla', 'Tuomiokirkko' ] }
  { maa: 'Espanja', nimi: 'Sevilla', asukasluku: 703 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

3. Hae kaikki suomalaiset kaupungit, joissa on alle 135 000 asukasta. Näytä vain nimi ja asukasluku.

```
> db.kaupungit.find({$and:[{maa:"Suomi"},{asukasluku:{$lte:135}}]},{nimi: 1,_id:0,asukasluku:1}).pretty();
< { nimi: 'Jyväskylä', asukasluku: 130 }
  { nimi: 'Varkaus', asukasluku: 25 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

4. Hae kaikki suomalaiset kaupungit, joissa on 150 000 - 200 000 asukasta.

```
> db.kaupungit.find({$and:[{maa:"Suomi"},{asukasluku:{$lte:200}},{asukasluku:{$gte:150}}]}.pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca8e"),
    maa: 'Suomi',
    nimi: 'Turku',
    asukasluku: 177,
    'nähtävyydet': [ 'Turun linna' ],
    seurat:
      [ { nimi: 'TPS', stadion: 'Kupittaa', sarja: 'Veikkausliiga' },
        { nimi: 'FC Inter', stadion: 'Kupittaa', sarja: 'Veikkausliiga' } ] }
  { _id: ObjectId("619ced2dc1b3eb036e93ca8f"),
    maa: 'Suomi',
    nimi: 'Oulu',
    asukasluku: 192,
    perustettu: 1605,
    seurat:
      [ { nimi: 'AC Oulu', sarja: 'Ykkönen' },
        { nimi: 'OPS', sarja: 'Ykkönen' } ] }
```

5. Hae kaikki suomalaiset kaupungit, joissa on alle 100 000 tai yli 200 000 asukasta. Tarvitset \$and-operaattoria, jonka operandit (maa-vertailu) ja asukaslukuvertailu tulevat taulukkoon. Asukaslukuveratailuun tarvitset \$or-operaattoria, jonka operandit tulevat omaan taulukkoonsa.

```
> db.kaupungit.find({$and:[{"maa":"Suomi"},{$or:[{"asukasluku":{$lte:100}}, {"asukasluku":{$gte:200}}]}]).pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca8d"),
  maa: 'Suomi',
  nimi: 'Tampere',
  asukasluku: 213,
  perustettu: 1779,
  'nähtävyydet': [ 'Huvipuisto' ],
  seurat:
    [ { nimi: 'Ilves', stadion: 'Ratina', sarja: 'Ykkönen' },
      { nimi: 'TPV', stadion: 'Tammela' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca90"),
  maa: 'Suomi',
  nimi: 'Varkaus',
  asukasluku: 25 }
```

6. Listaa suomalaiset kaupungit, joissa on alle 50 000 asukasta tai joissa pelaa Veikkausliiga-seura.

```
> db.kaupungit.find({$and:[{"maa":"Suomi"},{$or:[{"asukasluku":{$lte:50}}, {"seurat.sarja":{$in:["Veikkausliiga"]}]}]}]).pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca8e"),
  maa: 'Suomi',
  nimi: 'Turku',
  asukasluku: 177,
  'nähtävyydet': [ 'Turun linna' ],
  seurat:
    [ { nimi: 'TPS', stadion: 'Kupittaa', sarja: 'Veikkausliiga' },
      { nimi: 'FC Inter', stadion: 'Kupittaa', sarja: 'Veikkausliiga' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca90"),
  maa: 'Suomi',
  nimi: 'Varkaus',
  asukasluku: 25 }
```

7. Hae kaikki suomalaiset kaupungit, joiden tiedoissa on määritelty perustamisvuosi. Tähän sopii \$exists-operaattori.

```
> db.kaupungit.find({"maa":"Suomi","perustettu":{"$exists:true"}}).pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca8c"),
  maa: 'Suomi',
  nimi: 'Jyväskylä',
  asukasluku: 130,
  perustettu: 1837,
  seurat:
    [ { nimi: 'JJK', stadion: 'Harjun stadion', sarja: 'Ykkönen' },
      { nimi: 'JPS', stadion: 'Viitaniemi' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca8f"),
  maa: 'Suomi',
  nimi: 'Oulu',
  asukasluku: 192,
  perustettu: 1605,
  seurat:
    [ { nimi: 'AC Oulu', sarja: 'Ykkönen' },
      { nimi: 'OPS', sarja: 'Ykkönen' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca8d"),
  maa: 'Suomi',
  nimi: 'Tampere',
  asukasluku: 213,
  perustettu: 1779,
  'nähtävyydet': [ 'Huvipuisto' ],
  => 1555 - 1650 kääpistetty => 5518 - 1600 kääpistetty => koki: 5512 k
```


8. Hae kaikki kaupungit, joissa on seura jonka nimi alkaa "FC". Käytä \$regex-operaattoria. Alidokumenteista voidaan hakea pistenotaatiolla.

```
> db.kaupungit.find({"seurat.nimi":{"regex":"^FC"}}).pretty();
< { _id: ObjectId("619ced2dc1b3eb036e93ca91"),
  maa: 'Espanja',
  nimi: 'Barcelona',
  asukasluku: 1621,
  'nähtävyydet': [ 'La Rambla', 'Tuomiokirkko' ],
  seurat:
    [ { nimi: 'FC Barcelona', stadion: 'Camp Nou', sarja: 'La Liga' },
      { nimi: 'RCD Espanyol',
        stadion: 'Estadi Cornellà-El Prat',
        sarja: 'La Liga' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca8e"),
  maa: 'Suomi',
  nimi: 'Turku',
  asukasluku: 177,
  'nähtävyydet': [ 'Turun linna' ],
  seurat:
    [ { nimi: 'TPS', stadion: 'Kupittaa', sarja: 'Veikkausliiga' },
      { nimi: 'FC Inter', stadion: 'Kupittaa', sarja: 'Veikkausliiga' } ] }
Atlas: atlas-22xof3-shard-0 [primary] futis\
```

1. Listaa kolme suurinta kaupunkia asukasluvun mukaisessa järjestyksessä suurimmasta pienimpään siten että jätät suurimman huomioimatta (siis todellisuudessa haet asukasluvun mukaan suurimmat kaupugit 2-4). Tähän sopii \$sort, \$limit ja \$skip. Näytä vain kaupungin nimi ja asukasluku.

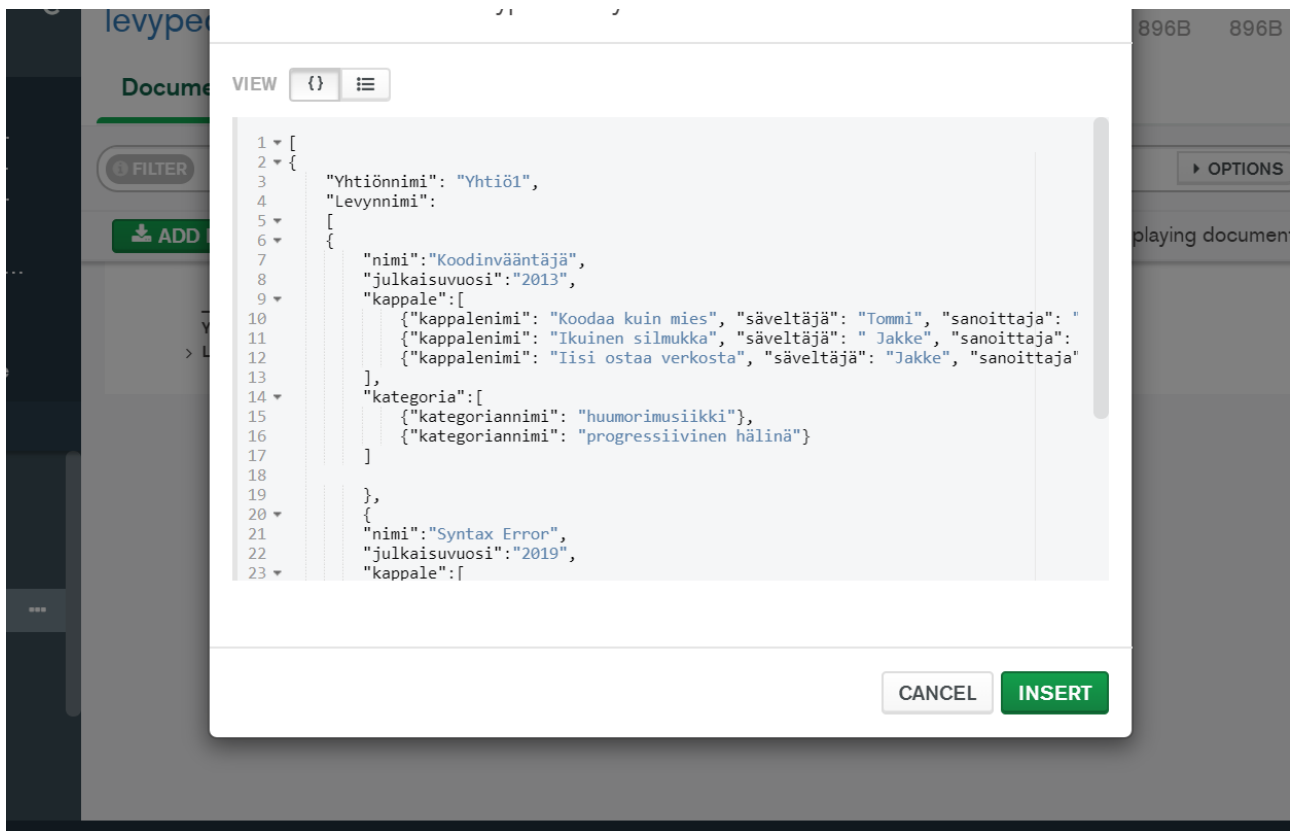
```
> db.kaupungit.find({}, {_id:0,nimi:1,asukasluku:1}).sort({"asukasluku":-1}).skip(1).limit(3).pretty();
< { nimi: 'Sevilla', asukasluku: 703 }
  { nimi: 'Tampere', asukasluku: 213 }
  { nimi: 'Oulu', asukasluku: 192 }
```

2. Näytä suomalaisten kaupunkien lukumäärä.

```
> db.kaupungit.count({"maa":"Suomi"});
< 5
```

7. tehtävä (find-komentoja levypediaan)

Aiemmin suunniteltiin levytietokanta ja kirjoitettiin esimerkkitaupaukseen liittyvät json-tiedostot. Importoi tiedot kantaan "levypedia". Kirjoita komennot, joilla importoit tiedot Mongooon.



Tee kolme hakua levypedia-tietokantaan. Kirjoita oma tehtävänantosi ja oikea komento sekä komennon palauttama tuloste.

```

use levypedia
'switched to db levypedia'
db.levy.find({"Levyynnimi.julkaisuvuosi":2019});

db.levy.find({"Levyynnimi.julkaisuvuosi":"2019"});
{ _id: ObjectId("619d23cdclb3eb036e93ca96"),
  'Yhtiönnimi': 'Yhtiö1',
  Levyynnimi:
    [ { nimi: 'Koodinvääntäjä',
        julkaisuvuosi: '2013',
        kappale:
          [ { kappalenimi: 'Koodaa kuin mies',
              'säveltäjä': 'Tommi',
              sanoittaja: 'Tommi' },
            { kappalenimi: 'Ikuinen silmukka',
              'säveltäjä': ' Jakke',
              sanoittaja: ' Jakke' },
            { kappalenimi: 'Iisi ostaa verkosta',
              'säveltäjä': 'Jakke',
              sanoittaja: 'Niko' } ],
        kategoria:
          [ { kategoriannimi: 'huumorimusiikki' },
            { kategoriannimi: 'progressiivinen hälinä' } ] ],
    / nimi: 'Syntax Error'
> db.levy.find().pretty();
< { _id: ObjectId("619d23cdclb3eb036e93ca96"),
  'Yhtiönnimi': 'Yhtiö1',
  Levyynnimi:
    [ { nimi: 'Koodinvääntäjä',
        julkaisuvuosi: '2013',
        kappale:
          [ { kappalenimi: 'Koodaa kuin mies',
              'säveltäjä': 'Tommi',
              sanoittaja: 'Tommi' },
            { kappalenimi: 'Ikuinen silmukka',
              'säveltäjä': ' Jakke',
              sanoittaja: ' Jakke' },
            { kappalenimi: 'Iisi ostaa verkosta',
              'säveltäjä': 'Jakke',
              sanoittaja: 'Niko' } ],
        kategoria:
          [ { kategoriannimi: 'huumorimusiikki' },
            { kategoriannimi: 'progressiivinen hälinä' } ] ],

```

```
> db.levy.count({"Levyynnimi.julkaisuvuosi":"2019"});  
< 1  
Atlas atlas-22xef3-shard-0 [primary] levypedia>
```

8. tehtävä (IKK-CRUD)

Aiemmin suunniteltiin Mongo-kannan rakenne Innokasti Koodavat Kokelaat -yrityksen sovellusten hakemiseen. Luo Mongo-kanta tekemistäsi json-tiedostoista mongoimport-komennolla.

Tee seuraavat kyselyt:

Hae kaikki projektit (find)

db.crud.find();

```
> db.crud.find();  
< { _id: ObjectId("61af768c5de8da9756e716aa"),  
  nimi: 'IKK',  
  Sovellus:  
    [ { nimi: 'PingPong',  
      luokka: 'peli',  
      tagit: [ '3D', 'urheilu' ],  
      group:  
        [ { membername: 'Keijo-Antero',  
          membersurname: 'Kuronen',  
            'tehtävä': 'tekniikka ja testaus' },  
          { membername: 'Sanna',  
            membersurname: 'Suomalainen',  
              'tehtävä': 'sisalto ja Scrum Master' } ],  
      meta: [ { katsottu: '100', ladattu: '21' } ] },  
    { nimi: 'Tiko-Karate',  
      luokka: 'peli',  
      tagit: [ '2D', 'urheilu', 'Tappelu' ],  
      group:  
        [ { membername: 'Raimo',
```

Hae kaikki projektit nousevassa aakkosjärjestyksessä nimen mukaan (find ja sort)

Hae projektit, joita on katsottu yli 100 kertaa (find, ensimmäiseksi alidokumentiksi \$gt-vertailu)

```
> db.crud.find({"Sovellus.meta.katsottu":{"$gt:100}});
< { _id: ObjectId("61af808d5de8da9756e716b0"),
  Nimi: 'IKK',
  Sovellus:
    [ { nimi: 'PingPong',
      luokka: 'pele',
      tagit: [ '3D', 'urheilu' ],
      group:
        [ { membername: 'Keijo-Antero',
          membersurname: 'Kuronen',
          'tehtävä': 'tekniikka ja testaus' },
          { membername: 'Sanna',
            membersurname: 'Suomalainen',
            'tehtävä': 'sisälto ja Scrum Master' } ],
      meta: [ { katsottu: 100, ladattu: 21 } ] },
    { nimi: 'Tiko-Karate',
      luokka: 'pele' }
```

Lisää kaikkien peliprojektien katsottu-kentän arvoon yksi (\$inc-operaattori). Tee edellinen kysely uudestaan; muuttuiko tulosjoukko?

```
db.crud.updateMany({$inc:{"Sovellus.meta.katsottu":1}});
```

Hae projektit, joita on katsottu yli 100 kertaa, mutta ladattu alle 20 kertaa (find, ensimmäisessä alidokumentissa kaksi ehtoa)

```
db.crud.find({$and:[{"Sovellus.meta.katsottu":{"gt:100}},{"Sovellus.meta.ladattu":{"lt:20}}]});
```

Hae projektit, joita on katsottu alle 60 tai yli 120 kertaa (tarvitset \$or-operaattoria)

```
db.crud.find({$and:[{"Sovellus.meta.katsottu":{"lte:100}},{"Sovellus.meta.ladattu":{"gte:20}}]});
```

Näytä kaikki projektit, joissa on ollut mukana testaaja (\$in-operaattori).

```
db.crud.find({"Sovellus.group.membersurname":{"in:["testaaja"]}});
```

9. tehtävä (ylläpitoa futis-kantaan)

Palataan vielä aiemman tehtävän futis-kantaan. Tee seuraavat toimenpiteet:

1. Päivitä Varkauden asukasluku oikeaksi (22).

```
> db.kaupungit.update({"nimi":"Varkaus"},{$set:{"asukasluku":22}});
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0 }
> db.kaupungit.find("nimi":"Varkaus");
```

✖ **SyntaxError:** Unexpected token, expected "," (1:24)

```
> 1 | db.kaupungit.find("nimi":"Varkaus");
    |                               ^
```

```
> db.kaupungit.find({"nimi":"Varkaus"});
< { _id: ObjectId("619ced2dc1b3eb036e93ca90"),
  maa: 'Suomi',
  nimi: 'Varkaus',
  asukasluku: 22 }
```

Atlas atlas-22xef3-shard-0 [primary] futis>

2. Lisää Varkaus-dokumenttiin kaupungin perustamisvuosi (1929).

```
> db.kaupungit.find({"nimi":"Varkaus"});
< { _id: ObjectId("619ced2dc1b3eb036e93ca90"),
  maa: 'Suomi',
  nimi: 'Varkaus',
  asukasluku: 22 }
> db.kaupungit.updateOne({"nimi":"Varkaus"},{$set:{"perustettu":1929}});
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

Atlas atlas-22xef3-shard-0 [primary] futis>

3. Päivitä suomalaisten kaupunkien asukasluvut lisäämällä kullekin kaupungille 2000 asukasta.

```
> db.kaupungit.updateMany({"maa":"Suomi"},{$inc:{"asukasluku":2}});
< { acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

4. Lisää Turun ja Tampereen nähtävyyksiin Tuomiokirkko.

```
> db.kaupungit.updateMany({$or:[{"nimi":"Turku"}, {"nimi":"Tampere"}]},{$push:{"nähtävyydet":"Tuomiokirkko"}});
< { acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

5. Tee haku, joka listaa kaikki kaupungit, joissa on nähtävyys "Tuomiokirkko" (\$in-operaattori).

```
> db.kaupungit.find({"nähtävyydet":{"$in":["Tuomiokirkko"]}});
< { _id: ObjectId("619ced2dc1b3eb036e93ca91"),
  maa: 'Espanja',
  nimi: 'Barcelona',
  asukasluku: 1621,
  'nähtävyydet': [ 'La Rambla', 'Tuomiokirkko' ],
  seurat:
    [ { nimi: 'FC Barcelona', stadion: 'Camp Nou', sarja: 'La Liga' },
      { nimi: 'RCD Espanyol',
        stadion: 'Estadi Cornellà-El Prat',
        sarja: 'La Liga' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca8e"),
  maa: 'Suomi',
  nimi: 'Turku',
  asukasluku: 179,
  'nähtävyydet': [ 'Turun linna', 'Tuomiokirkko' ],
  seurat:
    [ { nimi: 'TPS', stadion: 'Kupittaa', sarja: 'Veikkausliiga' },
      { nimi: 'FC Inter', stadion: 'Kupittaa', sarja: 'Veikkausliiga' } ] }
{ _id: ObjectId("619ced2dc1b3eb036e93ca8d"),
  maa: 'Suomi',
```


6. Lisää/päivitä kaupungin "Nurmes" asukasluvuksi 10, perustamisvuodeksi 1973 ja nähtävyyksiksi "Bomba" ja "Tuomiokirkko".

```
> db.kaupungit.update({"nimi":"Nurmes"},{$set:{"maa":"Suomi","asukasluku":10,"nähtävyydet":["Bomba","Tuomiokirkko"]},{upsert:true});
< { acknowledged: true,
  insertedId: ObjectId("61a66f96fc4cdab68e92c1e5"),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

7. Poista Nurmeksen nähtävyyksistä Tuomiokirkko ja lisää nähtävyyksiin "Vanha kauppala".

```
> db.kaupungit.updateOne({"nimi":"Nurmes"},{$set:{"nähtävyydet":"Vanha kauppala"}});
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }

> db.kaupungit.find({"nimi":"Nurmes.nähtävyydet"}).pretty();
<
> db.kaupungit.find({"nimi":"Nurmes"}).pretty();
< { _id: ObjectId("61a66f96fc4cdab68e92c1e5"),
  nimi: 'Nurmes',
  asukasluku: 10,
  maa: 'Suomi',
  'nähtävyydet': 'Vanha kauppala' }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

8. Poista kaupunki "Nurmes" (remove- tai drop-metodit).

```
db.kaupungit.deleteMany({"nimi":"Sevilla"});
```

```
{ _id: ObjectId("61af68ee5de8da9756e716a2"),
  maa: 'Espanja',
  nimi: 'Sevilla',
  asukasluku: 703,
  seurat:
    [ { nimi: 'Real Betis',
        stadion: 'Estadio Benito Villamarín',
        sarja: 'La Liga' },
      { nimi: 'Sevilla FC', sarja: 'La Liga' } ]
> db.kaupungit.deleteMany({"nimi":"Sevilla"});
< { acknowledged: true, deletedCount: 1 }
Atlas atlas-22xef3-shard-0 [primary] futis>
```

9. Poista koko kaupungit-kokoelma (drop toimii tässäkin).

```
db.kaupungit.drop();
```

10. Poista koko futis-tietokanta (metodi dropDatabase).

```
db.dropDatabase();
```

10. tehtävä (Miten menee?)

Vastaa Miten menee -3? -kyselyyn. Kysely löytyy osa 3 -välilehdeltä palautuslaatikon jälkeen. Tehtävän vastaukseksi riittää maininta, että on vastannut kyselyyn. Voit halutessasi kirjoittaa tähän tarkempia kommentteja.

Olen vastannut kyselyyn!