

# Raportti: KuntosaliDB - MongoDB-tietokannan suunnittelu ja toteutus –

## *Vika sivu kuvat shellistä.*

---

### 1. Projektin tarkoitus

Projektin tavoitteena on rakentaa MongoDB Atlas -tietokanta, joka hallitsee kuntosalin jäsenyyksiä, tunteja, ohjaajia ja varauksia. Tietomalli tukee laajennettavuutta ja noudattaa dokumenttietokantojen parhaita käytäntöjä

#### Käyttäjät ja oikeudet

MongoDB Atlas -käyttäjä on luotu seuraavasti:

- **Käyttäjäimi:** e2302129
- **Oikeudet:** *readWrite* vain tietokantaan **KuntosaliDB**

#### Connection string:

`mongodb+srv://e2302129:<salasana>@farhio00.p8ecg.mongodb.net/KuntosaliDB`

Salasana jos siihen tarvitaan: *me3hoAme2302129*

### 2. Tietokannan suunnittelu

#### 2.1 Kokoelmat ja dokumenttien rakenne

##### 2.1.1 Jasenet

- Tämä kokoelma tallentaa jäsenten tiedot, kuten etunimen, sukunimen, sähköpostiosoitteen, puhelinnumeron ja liittymispäivämäärän.

```
{  
  "etunimi": "Gabby",  
  "sukunimi": "Doy",  
  "sahkoposti": "gabby@example.com",  
  "puhelin": "0401234567",  
  "liittymispaiva": ISODate("2024-01-01T00:00:00Z")  
}
```

Toka jäsen:

```
{  
  "etunimi": "Fari",  
  "sukunimi": "Ibr",  
  "sahkoposti": "fari@example.com",  
  "puhelin": "0401234567",  
  "liittymispaiva": ISODate("2024-01-01T00:00:00Z")  
}
```

### 2.1.2 Ohjaajat

- Tämä kokoelma tallentaa ohjaajien tiedot.

```
{  
  "etunimi": "Gabriela",  
  "sukunimi": "Doykova",  
  "puhelin": "0501234567",  
  "sahkoposti": "Gabriela@kuntosali.fi",  
  "erikoisala": "Pilates"  
}
```

Toka Ohjaaja:

```
{  
  "etunimi": "Farhio",  
  "sukunimi": "Ibrahim",  
  "puhelin": "0501234567",  
  "sahkoposti": "Gabriela@kuntosali.fi",  
  "erikoisala": "Jooga"  
}
```

### 2.1.3 Tunnit

- Tämä kokoelma sisältää tuntien tiedot.

```
{  
  "tuntinimi": "Jooga",  
  "ajankohta": ISODate("2024-06-10T18:00:00Z"),  
  "ohjaaja_id": ObjectId("..."),  
  "maksimi_osallistujat": 20  
}
```

Toka tunti

```
{  
  "tuntinimi": "Pilates",  
  "ajankohta": ISODate("2024-06-10T18:00:00Z"),  
  "ohjaaja_id": ObjectId("..."),  
  "maksimi_osallistujat": 20  
}
```

#### 2.1.4 Varaukset

- Tämä kokoelma liittää jäsenet ja tunnit toisiinsa.

```
{  
  "jasen_id": ObjectId("..."),  
  "tunti_id": ObjectId("..."),  
  "varattu_pvm": ISODate("2024-06-01T12:00:00Z")  
}
```

### 2.2 Suunnitteluperiaatteet

Tietokannan suunnittelussa noudatettiin MongoDB:n parhaita käytäntöjä:

- Normaalisointi ja denormalisoointi: yksinkertaiset dokumentit, jotka yhdistetään tunnistilla.
- Joustavuus: rakenne mahdollistaa helpon laajentamisen.
- Indeksit: suorituskyvyn parantamiseksi luotiin indeksejä, esim. sähköpostille.

## 3. Käytetyt MongoDB-toiminnot

### 3.1 CRUD-toiminnot

Tietokannan käyttöön liittyvät CRUD-toiminnot suoritettiin MongoDB:n perustoiminoilla, kuten insertOne, updateOne ja deleteOne.

Esimerkkejä:

```
db.Jasenet.insertOne({...})  
db.Jasenet.find({...})  
db.Jasenet.updateOne({...}, {...})  
db.Jasenet.deleteOne({...})
```

### 3.2 Aggregointi

MongoDB:n aggregointia käytettiin yhdistämään tietoja useista kokoelmista:

```
db.Varaukset.aggregate([
  { $lookup: { from: "Jasenet", localField: "jasen_id", foreignField: "_id", as: "jasen_tiedot" } },
  { $unwind: "$jasen_tiedot" },
  { $match: { "tunti_id": ObjectId("...") } }
])
```

### 3.3 Indeksit ja suorituskyky

Tietokannan suorituskyvyn parantamiseksi luotiin indeksejä:

```
db.Jasenet.createIndex({ sahkoposti: 1 })
```

## 4. Yhteenveto ja päätelmät

KuntosaliDB on suunniteltu MongoDB:n dokumenttipohjaisen mallin mukaisesti. Se tukee hyvin kuntosalin jäsen- ja varausjärjestelmää ja mahdollistaa tehokkaan datan hallinnan. Aggregointien ja indeksien avulla saavutettiin hyvä suorituskyky ja skaalautuvuus.

### Lähteet

- MongoDB Dokumentaatio
- Tekoäly: OpenAI ChatGPT avustanut luonnostelussa ja syntaksin varmistuksessa

### Tekijät:

Farhio Ibrahim (e2302129) & Gabriela Doykova (e2302140)

Päivämäärä: 16.4.2025

## Kuvat



```
> use KuntosaliDB
< switched to db KuntosaliDB
> db.Jasenet.find()
< [
    {
        _id: ObjectId('67ff6e48b453b769d195200a'),
        etunimi: 'Farho',
        sukunimi: 'Ibrahim',
        sahkoposti: 'matti@example.com',
        puhelin: '0401234567',
        liittymispaiva: 2024-01-01T00:00:00.000Z
    },
    {
        _id: ObjectId('67ff6e7bb453b769d195200b'),
        etunimi: 'Gabriela',
        sukunimi: 'Doykova',
        sahkoposti: 'matti@example.com',
        puhelin: '0401234567',
        liittymispaiva: 2024-01-01T00:00:00.000Z
    }
]
> db.Jasenet.drop()
< true
> db.Jasenet.find()
<
> show collections
< Ohjaajat
    Tunnit
    Varaukset
```

```
> db.createCollection("Jasenet")
< { ok: 1 }
> show collections
< Jasenet
  Ohjaajat
  Tunnit
  Varaukset
> db.Jasenet.insertOne({
  etunimi: "Gabby",
  sukunimi: "Doy",
  sahkoposti: "gabby@example.com",
  puhelin: "0401234567",
  liittymispaiva: ISODate("2024-01-01T00:00:00Z")
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff740249c0dbcad6b6ab9f')
}
> db.Jasenet.insertOne({
  etunimi: "Fari",
  sukunimi: "Ibr",
  sahkoposti: "fari@example.com",
  puhelin: "0401234567",
  liittymispaiva: ISODate("2024-01-01T00:00:00Z")
})
```

```
littymispaiva: ISODate("2024-01-01T00:00:00Z")
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff741e49c0dbcad6b6aba0')
}
> db.Ohjaajat.insertOne({
  etunimi: "Gabriela",
  sukunimi: "Doykova",
  puhelin: "0501234567",
  sahkoposti: "Gabriela@kuntosali.fi",
  erikoisala: "Pilates"
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff745a49c0dbcad6b6aba1')
}
> db.Ohjaajat.insertOne({
  etunimi: "Farhio",
  sukunimi: "Ibrahim",
  puhelin: "0501234567",
  sahkoposti: "Farhio@kuntosali.fi",
  erikoisala: "Jooga"
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff747149c0dbcad6b6aba2')
```

```
    acknowledged: true,
    insertedId: ObjectId('67ff747149c0dbcad6b6aba2')
}
> db.Tunnit.insertOne({
  tuntinimi: "Jooga",
  ajankohta: ISODate("2024-06-10T18:00:00Z"),
  ohjaaja_id: ObjectId("67ff747149c0dbcad6b6aba2"), // Muista hakea oikea id aiemmin lisätystä ohjaajasta
  maksimi_osallistujat: 20
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff74a349c0dbcad6b6aba3')
}
> db.Tunnit.insertOne({
  tuntinimi: "Pilates",
  ajankohta: ISODate("2024-06-10T18:00:00Z"),
  ohjaaja_id: ObjectId("67ff745a49c0dbcad6b6aba1"), // Muista hakea oikea id aiemmin lisätystä ohjaajasta
  maksimi_osallistujat: 20
})
< {
  acknowledged: true,
  insertedId: ObjectId('67ff74c649c0dbcad6b6aba4')
}
> db.Varaukset.insertOne({
  jasen_id: ObjectId("67ff747149c0dbcad6b6aba2"), // Hae oikea id
  tunti_id: ObjectId("67ff74a349c0dbcad6b6aba3"),
  varattu_pvm: ISODate("2024-06-01T12:00:00Z")
})*
```

```
        tunti_id: ObjectId("67ff74a349c0dbcad6b6aba3"),
        varattu_pvm: ISODate("2024-06-01T12:00:00Z")
    })
< {
    acknowledged: true,
    insertedId: ObjectId('67ff756449c0dbcad6b6aba5')
}
> db.Varaukset.deleteOne({
    jasen_id: ObjectId("67ff747149c0dbcad6b6aba2"),
    tunti_id: ObjectId("67ff74a349c0dbcad6b6aba3"),
    varattu_pvm: ISODate("2024-06-01T12:00:00Z")
})
< {
    acknowledged: true,
    deletedCount: 1
}
> db.Varaukset.find().pretty()
<
> db.Varaukset.insertOne({
    jasen_id: ObjectId("67ff741e49c0dbcad6b6aba0"),    // Hae oikea id
    tunti_id: ObjectId("67ff74a349c0dbcad6b6aba3"),
    varattu_pvm: ISODate("2024-06-01T12:00:00Z")
})
< {
    acknowledged: true,
    insertedId: ObjectId('67ff75ed49c0dbcad6b6aba6')
}
> db.Varaukset.insertOne({
    }
> db.Varaukset.insertOne({
    jasen_id: ObjectId("67ff740249c0dbcad6b6ab9f"),    // Hae oikea id
    tunti_id: ObjectId("67ff74c649c0dbcad6b6aba4"),
    varattu_pvm: ISODate("2024-06-01T12:00:00Z")
})
< {
    acknowledged: true,
    insertedId: ObjectId('67ff761149c0dbcad6b6aba7')
}
> db.Jasenet.find()
< [
    {
        _id: ObjectId('67ff740249c0dbcad6b6ab9f'),
        etunimi: 'Gabby',
        sukunimi: 'Doy',
        sahkoposti: 'gabby@example.com',
        puhelin: '0401234567',
        liittymispaiva: 2024-01-01T00:00:00.000Z
    },
    {
        _id: ObjectId('67ff741e49c0dbcad6b6aba0'),
        etunimi: 'Fari',
        sukunimi: 'Ibr',
        sahkoposti: 'fari@example.com',
        puhelin: '0401234567',
        liittymispaiva: 2024-01-01T00:00:00.000Z
    }
]
```

```
        liittymispaiva: 2024-01-01T00:00:00.000Z
    }
}
db.Ohjaajat.find()
{
    _id: ObjectId('67ff6e93b453b769d195200c'),
    etunimi: 'Gabby',
    sukunimi: 'Doy',
    puhelin: '0501234567',
    sahkoposti: 'laura@kuntosalifi',
    erikoisala: 'Jooga'
}
{
    _id: ObjectId('67ff745a49c0dbcad6b6aba1'),
    etunimi: 'Gabriela',
    sukunimi: 'Doykova',
    puhelin: '0501234567',
    sahkoposti: 'Gabriela@kuntosalifi',
    erikoisala: 'Pilates'
}
{
    _id: ObjectId('67ff747149c0dbcad6b6aba2'),
    etunimi: 'Farhio',
    sukunimi: 'Ibrahim',
    puhelin: '0501234567',
    sahkoposti: 'Farhio@kuntosalifi',
    erikoisala: 'Jooga'
}
```

```
    erikoisala: 'Jooga'
  }
> db.Tunnit.find()
< [
  {
    _id: ObjectId('67ff74a349c0dbcad6b6aba3'),
    tuntinimi: 'Jooga',
    ajankohta: 2024-06-10T18:00:00.000Z,
    ohjaaja_id: ObjectId('67ff747149c0dbcad6b6aba2'),
    maksimi_osallistujat: 20
  },
  {
    _id: ObjectId('67ff74c649c0dbcad6b6aba4'),
    tuntinimi: 'Pilates',
    ajankohta: 2024-06-10T18:00:00.000Z,
    ohjaaja_id: ObjectId('67ff745a49c0dbcad6b6aba1'),
    maksimi_osallistujat: 20
  }
> db.Varaukset.find()
< [
  {
    _id: ObjectId('67ff75ed49c0dbcad6b6aba6'),
    jasen_id: ObjectId('67ff741e49c0dbcad6b6aba0'),
    tunti_id: ObjectId('67ff74a349c0dbcad6b6aba3'),
    varattu_pvm: 2024-06-01T12:00:00.000Z
  },
  {
    _id: ObjectId('67ff761149c0dbcad6b6aba7'),
    jasen_id: ObjectId('67ff740249c0dbcad6b6ab9f'),
    tunti_id: ObjectId('67ff745a49c0dbcad6b6aba4'),
    varattu_pvm: 2024-06-01T12:00:00.000Z
  }
]
```

```
> db.Jasenet.insert()
< [
  {
    _id: ObjectId('67ff75ed49c0dbcad6b6aba6'),
    jasen_id: ObjectId('67ff741e49c0dbcad6b6aba0'),
    tunti_id: ObjectId('67ff74a349c0dbcad6b6aba3'),
    varattu_pvm: 2024-06-01T12:00:00.000Z
  }
]
{
  _id: ObjectId('67ff761149c0dbcad6b6aba7'),
  jasen_id: ObjectId('67ff740249c0dbcad6b6ab9f'),
  tunti_id: ObjectId('67ff74c649c0dbcad6b6aba4'),
  varattu_pvm: 2024-06-01T12:00:00.000Z
}
> db.Jasenet.updateOne(
  { sahkoposti: "matti@example.com" },
  { $set: { puhelin: "0412345678" } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
> db.Jasenet.createIndex({ sahkoposti: 1 })
< sahkoposti_1
Atlas atlas-z5vv0g-shard-0 [primary] KuntosaliDB>
```



Kuntosali

EDIT



Varaukset

Farhio&Gabriela



Next refresh in an hour

Valittiin tollainen

chart koska ei ollut enemmän kuin kaksi varasta, jäsentä, tunteja yms