

Mikko Kukkonen

# MONGODB MODERNISSA SOVELLUS- KEHITYKSESSÄ

Kandidaatintyö  
Informaatioteknologian ja viestinnän tiedekunta  
Tarkastaja: Mikko Nurminen  
Joulukuu 2022

# TIIVISTELMÄ

Mikko Kukkonen: MongoDB modernissa sovelluskehityksessä  
Kandidaatintyö  
Tampereen yliopisto  
Tieto- ja sähkötekniikan kandidaatin tutkinto-ohjelma, Tietotekniikka  
Joulukuu 2022

---

Kahden viime vuosikymmenen aikana internetin ja mobiililaitteiden käytön räjähdysmäinen kasvu on lisännyt valtavasti tallennettavan datan määrää. Tämä datan määrän kasvu on lisännyt tarvetta uusien innovatiivisten tietokantaratkaisujen kehittämiseen. Yksi näistä uuden sukupolven tietokantajärjestelmistä on MongoDB.

Tässä kirjallisuuskatsauksessa tutustutaan MongoDB:n käyttöön modernissa sovelluskehityksessä. Aiheesta julkaistun kirjallisuuden avulla pyritään selvittämään, miksi MongoDB on noussut viime vuosikymmenen aikana yhdeksi suosituimmista tietokantajärjestelmistä. Työn alussa esitellään yleisesti tietokantojen perusominaisuuksia, sekä yleisimpiä tietokantatyyppejä. Julkaistun kirjallisuuden avulla tutustutaan MongoDB:n ominaisuuksiin ja heikkouksiin, sekä selvitetään MongoDB:n suosioon vaikuttavia tekijöitä. MongoDB:n suorituskykyä vertaillaan muihin tietokantaratkaisuihin aiheesta julkaistujen vertaisarvioitujen tutkimusten avulla.

Tämän kirjallisuuskatsauksen tuloksina havaittiin, että MongoDB:n dokumenttipohjainen datamalli tarjoaa yksinkertaisuutensa takia nopeaa datan saatavuutta ja tietokannan skaalautuvuutta. MongoDB:n ongelmiksi havaittiin, että luontaisesti relaatiomallisen datan muuntaminen MongoDB:lle sopivaan muotoon aiheuttaa ylimääräistä työtä sovelluskehittäjälle. Myös MongoDB:n skeemattomuuden havaittiin aiheuttavan ongelmia, koska datamallin yhtenäisyys jää sovelluskehittäjän vastuulle ja aiheuttaa näin ylimääräistä työtä. MongoDB:n suorituskykyä käsittelevien tutkimusten tulokset osoittivat, että MongoDB on erityisen tehokas suoritettaessa haku- ja lisäysoperaatioita. Havaittiin, että MongoDB voi olla suorituskyvyltään hyvin tehokas tietokanta, mutta saatutettavat tehokkuusedut vaihtelevat käsiteltävän datan ja sovelluksen käyttötarkoituksen mukaan. Tutkimuksen johdolla havaittiin, että MongoDB soveltuu horisontaalisen skaalautuvuutensa ansiosta suuria datamääriä käsitteleville sovelluksille. Havaittiin myös, että skeemattomuuden ansiosta MongoDB sopii hyvin tietokannaksi moderneille sovelluksille, joiden kehitys on nopeaa ja tietokannan arkkitehtuuria joudutaan muuttamaan useita kertoja kehityksen eri vaiheissa. Näiden ominaisuuksien ansiosta MongoDB on noussut viimeisen vuosikymmenen aikana yhdeksi suosituimmista tietokantajärjestelmistä.

Avainsanat: Tietokanta, NoSQL, dokumenttitietokanta, MongoDB

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. TIETOKANNAT .....	3
2.1 Relaatiotietokannat .....	3
2.2 NoSQL-tietokannat .....	4
2.3 Dokumenttitietokannat .....	5
3. MONGODB .....	7
3.1 Ominaisuudet .....	8
3.2 Heikkoudet .....	9
4. SUORITUSKYKY .....	11
4.1 MongoDB verrattuna NoSQL-tietokantoihin .....	11
4.2 MongoDB verrattuna relaatiotietokantoihin .....	14
5. YHTEENVETO .....	18
LÄHTEET .....	20

# LYHENTEET JA MERKINNÄT

BSON

JSON

SQL

XML

Binary JSON

JavaScript Object Notation

Structured Query Language

eXtensible Markup Language

# 1. JOHDANTO

Kahden viime vuosikymmenen aikana internetin ja mobiililaitteiden käyttö on kasvanut räjähdysmäisesti, mikä on lisännyt valtavasti generoidun datan määrää. Kaikki tämä data säilötään erilaisten tietokantajärjestelmien avulla. Jatkuvasti kasvava datan määrä on toiminut kannustimena kehittää uusia innovatiivisia tietokantaratkaisuja.

1970-luvulta lähtien relaatiotietokannat ovat olleet yleisin tietokantaratkaisu. Nykypäivänä kymmenen käytetyimmän tietokantajärjestelmän listasta seitsemän on relaatiotietokantoja. Relatiotietokantoja ei kuitenkaan suunniteltu käsittelemään tehokkaasti suuria datamääriä, joita modernit sovellukset keräävät. Ongelman ratkaisuksi kehitettiin NoSQL tietokannat. (Fraczek & Plechawska-Wojcik 2017)

Kehitetyistä NoSQL-tietokannoista suosituimmaksi on noussut dokumenttitietokanta MongoDB. MongoDB:n kehityksen aloitti yhdysvaltalainen yritys 10gen (myöhemmin MongoDB Inc.) vuonna 2007. (Dechev et al. 2019) MongoDB on kehitetty alusta asti vastaamaan modernin web-kehityksen synnyttämiin datanhallinnan haasteisiin. Se on tehokas ja skaalautuva moniulotteiseen käyttöön suunniteltu tietokanta. (Jia & Shen 2015)

Tässä työssä perehdytään MongoDB-tietokantaan sovelluskehityksen näkökulmasta. Tutkielmassa tutustutaan aiheita käsittelevien tutkimusten avulla MongoDB:n ominaisuuksiin. Tavoitteena on selvittää miksi MongoDB on viime vuosikymmenen aikana noussut yhdeksi käytetyimmistä tietokantajärjestelmistä. Tarkastelukohteena ovat myös MongoDB:n heikkoudet ja rajoitteet. Nämä rajoitteet huomioon ottaen tutkitaan myös sovelluskohteita, joissa MongoDB ei kykene korvaamaan perinteisiä relaatiotietokantoja.

Tutkielma on toteutettu kirjallisuuskatsauksena. Lähteinä on käytetty aiheeseen liittyvää kirjallisuutta, kuten kirjoja, artikkeleita ja konferenssijulkaisuja. Kirjallisuutta on haettu Tampereen yliopiston Andor-palvelusta, sekä Google Scholar-palvelusta. Kirjallisuutta haettiin hakusanoilla "MongoDB", "document-oriented database", "NoSQL" ja "web-development". Näistä hakusanoista muodostettiin hakulauseita AND- ja OR-operaattoreiden avulla. Löytyneistä lähteistä suosittiin vertaisarvioituja julkaisuja. Myös julkaisuvuoden kiinnitettiin huomiota ja mahdollisimman uusia julkaisuja suosittiin. Saadut hakutulokset käytiin läpi ja lähteiksi valittiin ne tulokset, joiden otsikko tai abstrakti sisälsi tutkielman kannalta relevanttia tietoa.

Tutkielmassa käsitellään ensin tietokantoihin liittyviä peruskäsitteitä. Luvussa 2 esitellään erilaiset tietokantaluokat ja niiden suurimmat erot. Luvussa 3 käsitellään MongoDB:n ominaisuuksia ja heikkouksia, sekä selvitetään syitä MongoDB:n suosiolle. Luvussa 4 tutkitaan ja verrataan MongoDB:n suorituskykyä muihin tietokantajärjestelmiin aiheesta julkaistujen tutkimusten avulla. Luvussa 5 saadut tulokset ja havainnot esitetään yhteenvetona.

## 2. TIETOKANNAT

Tietokoneohjelmistot ovat aina vaatineet tavan tallentaa dataa. Datan tallennuksesta vastaa tietokanta. Tietokanta on organisoitu kokoelma informaatiota eli dataa. Data on yleensä tallennettuna sähköisesti tietokonejärjestelmässä. Tietokantaa hallitaan yleensä tietokannanhallintajärjestelmän avulla. Data, tietokannanhallintajärjestelmä ja näiden kanssa toimivat sovellukset muodostavat kokonaisuuden, jota kutsutaan tietokantajärjestelmäksi, tai lyhyemmin vain tietokannaksi. Tässä tutkimuksessa tietokanta-termillä viitataan tähän kokonaisuuteen. (Oracle 2022)

Yleisimpiä tietokantaan kohdistuvia operaatioita ovat uuden datan lisääminen, datan lukeminen, datan päivittäminen ja datan poistaminen. Myös datan varmuuskopiointi, tietoturvan ylläpito ja virheiden raportointi ovat tärkeitä tietokantaoperaatioita. Käyttäjä suorittaa näitä operaatioita tietokannanhallintajärjestelmän avulla. Tietokannanhallintajärjestelmä tarjoaa siis käyttöliittymän käyttäjän ja tietokannan välille. (Oracle 2022)

Monien tietokantatyyppeiden kuten relaatiotietokantojen sisältämää dataa käsitellään ja muokataan tätä tarkoitusta varten kehitetyn kielen avulla. *Structured Query Language* (SQL) on tietokantojen käsittelyä varten kehitetty standardoitu kyselykieli. SQL-kielestä on useita erilaisia versioita. (Jose & Abraham 2020)

Modernit yritykset vaativat tietokannoilta paljon. Datan hakukyselyt ovat monimutkaisia ja vastausajan täytyy olla lähes välitön. Modernit yritykset keräävät todella paljon dataa useista eri lähteistä. Tämä vaikeuttaa huomattavasti datan organisointia tietokannassa. Yritykset myös kasvavat nopeaa tahtia, jolloin tietokannan skaalautuvuus on tärkeää. Tietokannan tulee kaiken tämän lisäksi ylläpitää virheetöntä tietoturvaa. (Oracle 2022)

Vuosikymmenien saatossa tietokannat ovat kehittyneet muuttuvien käyttötarkoitusten ja vaatimusten mukaan. Tämä on johtanut useiden eri tietokantatyyppeiden kehittämiseen. Erilaisia tietokantatyyppejä on nykyisin useita kymmeniä. Tässä tutkielmassa keskitytään MongoDB-tietokantaan, joka luokitellaan dokumenttitietokannaksi. Dokumenttitietokannat itsessään kuuluvat NoSQL-tietokantoihin. (Dechev et al. 2019)

### 2.1 Relaatiotietokannat

Relaatiotietokanta on tietokanta, joka tallentaa dataa ja tarjoaa tavan käsitellä erilaista dataa perustuen datan välisiin suhteisiin. Relaatiotietokannat perustuvat relaatiomalliin, joka on intuitiivinen ja suoraviivainen tapa esittää dataa taulukkomuodossa. Relaatiomallissa jokaisella taulukon rivillä on uniikki id-tunniste, jota kutsutaan avaimeksi. Taulukon

kolumnit sisältävät datan attribuutteja. Kuvassa 1 on esitettyä esimerkki relaatiomallin taulukosta. (Oracle - RDBMS 2022)

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Kuva 1. Esimerkkidataa tallennettuna relaatiomallin taulukkoon (W3Schools 2022)

Relaatiomallin taulukot muodostavat keskinäisiä suhteita yhteisten kolumnien avulla. Suhteen muodostamiseen riittää, että kahdella taulukolla on vain yksi yhteinen ID eli avainkolumni. Relaatiotietokantojen taulukkomalli tarjoaa intuitiivisen tavan tallentaa ja käsitellä rakenteellista informaatiota. (Oracle - RDBMS 2022)

Relaatiomalli on paras tapa ylläpitää datan yhtenäisyyttä usean tietokantakopion välillä, sillä relaatiotietokanta päivittää yhteen tietokantakopioon tehdyn muutoksen välittömästi muihin kopioihin. Relaatiotietokantojen vahvuudet tulevat parhaiten esille SQL-kyselykieltä käytettäessä. Relaatiotietokannat tukevat täydellistä SQL-liitosoperaatiota. Tämän avulla voidaan suorittaa loogisia operaatioita useille tietokannan taulukoille yhdellä kyselyllä hyödyntäen taulukoiden välisiä suhteita. (Oracle - RDBMS 2022)

## 2.2 NoSQL-tietokannat

NoSQL-tietokannat ovat epärelaatiollisia ja horisontaalisesti skaalautuvia tietokantoja. MongoDB kuuluu NoSQL-tietokantoihin eli sillä on myös kaikki edellä mainitut NoSQL-tietokannoille tyypilliset ominaisuudet. NoSQL-tietokannat kehitettiin vastaamaan modernin web-sovelluskehityksen tarpeisiin. (Dechev et al. 2019) Tilastojen mukaan erilaisia NoSQL-tietokantoja on yli 225 kappaletta (Hostingdata.co.uk 2022).

Relaatiomallia hyödyntävät tietokannat joutuvat usein suorittamaan SQL-liitosoperaatioita, jotka liittävät toisiinsa useita taulukoita niiden välisten suhteiden mukaan. Tämä hidastaa kyselyoperaatioita etenkin suurissa ja monimutkaisissa tietokannoissa. Tästä



syystä monet yritykset ovat siirtyneet käyttämään NoSQL-tietokantoja. (Mishra et al. 2018)

Terminä NoSQL on harhaanjohtava, sillä sopivassa tilanteessa myös NoSQL-tietokannat käyttävät SQL-kyselykieltä. NoSQL-tietokannat eivät noudata relaatiomallia. Ne eivät myöskään tue SQL-liitosoperaatioita. NoSQL-tietokannoissa kaikki toisiinsa liittyvä data on tallennettu yhden dokumentin sisälle. Tämä parantaa datan käsittelynopeutta, koska liitosoperaatioita ei suoriteta. (Chen & Lee 2019)

NoSQL-tietokannoissa data on usein tallennettuna eri palvelimilla. Datan tallennuspaikkaa hallitaan metadatan avulla. Monet NoSQL-tietokannat ovat avoimen lähdekoodin ohjelmia ja näin ollen ilmaiseksi ladattavissa. NoSQL-tietokantojen horisontaalinen skaalautuvuus mahdollistaa palvelimien määrän lisäämisen tai vähentämisen. Tämän ominaisuuden avulla pystytään joustavasti vastaamaan muuttuvaan prosessointikapasiteetin tarpeeseen. (Chen & Lee 2019)

NoSQL-tietokannat ovat skeemattomia. NoSQL-tietokannat eivät siis vaadi valmiiksi määriteltyä mallia ennen datan lisäämistä. Tämän lisäksi NoSQL-tietokannat tarjoavat useita ohjelmointirajapintoja moneen eri tarkoitukseen. (Chen & Lee 2019)

NoSQL-tietokannat pystyvät toimimaan, vaikka osa tietokantajärjestelmän osista kaatuisi teknisen virheen takia. On tyypillistä, että tietokanta pitää useita kopioita datasta usealla eri palvelimella. Tämän avulla tietokanta kykenee vastaamaan kaikkiin kyselyihin, vaikka osa palvelimista ei olisi käytettävissä. (Diogo et al. 2019)

NoSQL-tietokannat eivät toteuta vahvaa johdonmukaisuutta. Vahva johdonmukaisuus on ominaisuus, joka takaa, että jokaisen tietokantaan suoritettujen transaktioiden jälkeen tietokanta on aina johdonmukaisessa tilassa. NoSQL-tietokanta takaa johdonmukaisen tilan vasta tietyn aikamäärän jälkeen. (Dechev et al. 2019) Tämä tarkoittaa, että tietokanta saattaa olla joskus epäjohdonmukaisessa tilassa. Esimerkiksi tilanne, jossa tietokanta pitää useita kopioita samasta datasta usealla eri palvelimella. Tällaisessa järjestelmässä esiintyy aikaikkunoita, joissa kopiot datasta ovat epäjohdonmukaisia, kun yhtä kopiota datasta on päivitetty, mutta muut kopiot sisältävät vanhaa, päivittämätöntä dataa. Tietyn ajan kuluttua NoSQL-tietokantajärjestelmä päivittää kaikki kopiot johdonmukaiseen tilaan. (Chen & Lee 2019)

### **2.3 Dokumenttitietokannat**

Dokumenttitietokannat ovat NoSQL-tietokantojen alaluokka. Dokumenttitietokannat ovat joustavia. Tämä joustavuus tekee dokumenttitietokannoista hyvän ratkaisun sovelluk-

sille, joissa datan muoto vaihtelee. Yleisesti käytettyjä dokumenttimuotoja ovat *eXtensible Markup Language* (XML) sekä *JavaScript Object Notation* (JSON). (Jia & Shen 2015; json.org 2022) MongoDB käyttää JSON-muotoista dokumenttiformaattia. Dokumenttitietokannoilla on neljä ydinoperaatiota. Nämä operaatiot ovat luontioperaatio, hakuoperaatio, päivitysoperaatio ja poisto-operaatio. (Mishra et al. 2018)

Dokumenttitietokannat korvaavat muissa tietokantamalleissa, kuten relaatiotietokannoissa käytetyn rivikonseptin dokumenttimallilla. Dokumenttitietokannat sallivat datan säilömisen dokumentti- tai listamuodossa. Tämän ansiosta myös monimutkaisia hierarkisia suhteita voidaan säilöä yksittäisen solun sisällä. Dokumenttitietokannoissa ei ole valmiiksi määriteltyjä tiukkoja kaaviomalleja eli ne ovat skeemattomia. Dokumentin avain-arvo-pareilla ei ole valmiiksi määriteltyä ja pysyvää tyyppiä tai kokoa. Tämä tekee kenttien lisäämisestä ja poistamisesta joustavampaa kuin relaatiotietokantamalleissa. (Bradshaw et al. 2019)

Edellä mainittiin, että dokumenttitietokannat eivät vaadi tarkasti määriteltyä skeemaa. Tämä tarjoaa tietokannan suunnittelijoille joustavuutta, mutta aiheuttaa myös ylimääräistä työtä. Skeemattomuus voi johtaa tietokannan tilaan, jossa saman kokoelman eri paikallisissa instansseissa on käytetty erilaisia skeemoja. Tämä voi aiheuttaa ongelmia skeemojen yhteensopivuuden kanssa etenkin suuremmissa tietokannoissa. Dokumenttitietokantojen skeemattomuus vaatii tietokannan suunnittelijalta tarkempaa dokumenttiota kokoelman skeemojen rakenteesta. (Gallinucci et al. 2018)

### 3. MONGODB

MongoDB käyttää erilaista termistöä perinteisiin relaatiotietokantoihin verrattuna. MongoDB:ssä datan perusyksikkö on nimeltään dokumentti. Dokumentti on verrattavissa relaatiotietokantojen riviin. Suurimpana erona dokumentti on paljon ilmaisuvoimaisempi. ”Kokoelma” vastaa relaatiotietokantojen taulua. MongoDB ryhmittelee kokoelmat tietokantoihin. Yksi MongoDB:n instanssi voi sisältää useita itsenäisiä tietokantoja, jotka kaikki sisältävät omat kokoelmansa. (Dechev et al. 2019)

MongoDB ei käytä perinteistä SQL-kyselykieltä, vaan omaa MongoDB-kyselykieltä. MongoDB hyödyntää Mongo shell-työkalua. Mongo shell:n avulla MongoDB:n instansseja voidaan tarkastella ja MongoDB:n sisältämää dataa voidaan muokata käyttäen MongoDB:n kyselykieltä. Mongo shell toimii myös täysin itsenäisenä JavaScript-tulkkina. MongoDB:llä on virallinen pilvipalvelualusta MongoDB Atlas, jonka avulla MongoDB:tä voidaan käyttää pilvipohjaisena palveluna. (Bradshaw et al. 2019)

MongoDB:n ydin on dokumentti. Dokumentti on järjestetty kokoelma avain–arvo-pareja. Kuvassa 2 on esitettyä itse toteutettu esimerkki MongoDB:n dokumenttirakenteesta. Esimerkin dokumentti sisältää blogisovelluksen tallentamaa esimerkkidataa. Dokumenttien sisältämät arvot voivat olla useita eri tietotyyppisiä. Dokumentin arvo voi olla myös toinen dokumentti. (Bradshaw et al. 2019) MongoDB:n kokoelmilla ei ole ennalta määritettyjä skeemoja. Tämän ansiosta yhden kokoelman sisällä olevilla dokumenteilla voi olla useita erilaisia muotoja. Tämä skeemattomuus lisää merkittävästi MongoDB-tietokannan joustavuutta, mutta vaatii myös, että tietokannan spesifinen toteutus rakenne valitaan jo kehityksen aikaisessa vaiheessa. (Klein et al. 2015)

```
{
  _id: ObjectId('62dcd20tt04958376ebdd45')
  title: "test blog"
  author: "tester"
  url: "test.url"
  user: ObjectId('62dcd20tt04978426ebdd6t')
  __v: 0
  likes: 5
  comments: {
    0: "test comment"
    1: "test comment2"
  }
}
```

Kuva 2. MongoDB:n dokumenttirakenne käyttäjien blogeja tallentavalle sovellukselle.

Useiden erityyppisten dokumenttien säilyminen saman kokoelman sisällä vaikeuttaa sovelluskehittäjän työtä. Tällaista tietokannan arkkitehtuuria käytettäessä sovelluskehittäjän on varmistettava, että jokainen kysely palauttaa ainoastaan tietyn skeeman mukaisia dokumentteja. Vaihtoehtoisesti voidaan varmistaa, että sovelluksen koodi kykenee käsittelemään erimuotoisia dokumentteja. (Bradshaw et al. 2019)

### 3.1 Ominaisuudet

MongoDB on suunniteltu toteuttamaan horisontaalista skaalautumista. Dokumenttipohjainen datamalli helpottaa datan hajottamista usealle palvelimelle. MongoDB myös hoi-taa datan tasapainottamisen usealle dataklusterille. Näin MongoDB tarjoaa horisontaa-lista skaalautuvuutta. Yksinkertaistettujen datamallien avulla MongoDB tarjoaa myös no-peaa datan saatavuutta. (Klein et al. 2015; Dechev et al. 2019)

Toisin kuin perinteiset relaatiotietokannat, MongoDB ei tue liitosoperaatioita. Liitosepe-raatioiden toteuttaminen tehokkaasti on haastavaa jaetuissa järjestelmissä. Tämän takia liitosepe-raatioiden sisällytys jätettiin tietoisesti pois MongoDB:n suunnitteluvaiheessa, jotta skaalautuvuutta parannettaisiin. (Bradshaw et al. 2019)

MongoDB tarjoaa aggregointiviitekehyksen, jonka avulla käyttäjä voi suorittaa erilaisia analytiikkaoperaatioita kokoelmille. MongoDB:n aggregointi vastaa toiminnallisuudel-taan SQL:n GROUP BY -lausetta. Tämän toiminnon avulla voidaan siis yhdistää arvoja useammilta eri dokumenteilta, jonka jälkeen näille arvoille voidaan suorittaa erilaisia ag-gregointioperaatioita. (Bradshaw et al. 2019)

Vaikka MongoDB ei noudatakaan relaatiomallia, on sillä siitä huolimatta monia relaatio-tietokantojen ominaisuuksia. MongoDB kykenee järjestämiseen, toissijaiseen indeksointiin ja sisäkkäisten dokumenttien kyselyihin. MongoDB tukee yleisimpiä tietokantojen operaatioita, kuten luonti-, poisto-, luku- ja päivitysoperaatioita. (Makris et al. 2020)

Datan muuttumattomuudella tarkoitetaan tietokantajärjestelmän ominaisuutta, joka takaa, että tietokantaan tallennetut kirjoitusoperaatiot säilyvät siellä pysyvästi. Otetaan esimerkiksi järjestelmä, joka varaa käyttäjille teatterilippuja. Tämän järjestelmän täytyy pysyä säilyttämään tieto lippujen varauksesta tietokannassa, vaikka osa varausjärjestelmästä kaatuisi väliaikaisesti. On muutamia tilanteita, joissa MongoDB ei kykene takaamaan datan muuttumattomuutta. Tällaisia tilanteita ovat laitteistoviat, kuten kovalevyn korruptoituminen. MongoDB ei myöskään kykene suojaamaan dataa tiedostojärjestelmän virheiltä. Datan replikaatio ja varmuuskopiointi on ainoa vaihtoehto järjestelmävirheiltä suojautumiseen. (Bradshaw et al. 2019)

## 3.2 Heikkoudet

MongoDB on kerännyt olemassaolonsa aikana myös kritiikkiä sovelluskehittäjiltä. Osa MongoDB:n puutteista ja ongelmista on korjattu uusimmissa versioissa, mutta osa ongelmista on säilynyt. Yksi suurimpia MongoDB:n puutteita on täydellisen SQL-liitosoperaation puuttuminen. MongoDB tukee vain osittaista liitosoperaatiota luvussa 3.1 esitellyn aggregointiominaisuuden avulla. MongoDB:n aggregoinnin avulla ei voida suorittaa päivitysoperaatioita datalle. Sovelluskehittäjän täytyy siis itse toteuttaa sovelluslogiikka, kun dataa halutaan päivittää datan välisten suhteiden perusteella. (Bradshaw et al. 2019; Spencer 2020)

Toinen merkittävä MongoDB:n ongelma on MongoDB:n dokumenteille asettama oletus ID. Tämä ID on muodoltaan *Binary JSON* (BSON) Object ID. BSON on tapa esittää tietorakenteita binäärimuodossa. Kuvassa 3 on esitettyä esimerkki BSON Object ID:stä. BSON ID:n järjestyslogiikka ei ole ohjelmoijalle selkeästi hahmottuva, joten se on vaikealukuisempi kuin tavallinen nouseva kokonaisluku. BSON ID on myös vaikea muistaa ja sen luettavuus on heikko. BSON ID:t aiheuttavat ongelmia myös datan yhdenvertaisuuden tarkastelussa. BSON ID:t eivät ole merkkijonoja vaan täysin oma tyyppinsä. BSON ID:t kuitenkin sarjallistuvat merkkijonoksi. Tämän takia sovelluskehittäjän täytyy BSON ID:tä käsiteltäessä tarkistaa, että merkkijonoja vertaillaan merkkijonoihin ja olioita vertaillaan olioihin. (Bradshaw et al. 2019; Spencer 2020)

```
_id: ObjectId('62dcd20tt04958376ebdd45')
```

### Kuva 3. MongoDB:n asettama BSON Object ID

Yksi MongoDB:n ongelmista liittyy sen dokumenttimalliin. Suuri osa sovellusten säilö-  
mästä datasta sisältää luontaisesti relaatioita. Datarivit kuuluvat suurempiin kokonai-  
suuksiin ja niillä on erimuotoisia suhteita toisiinsa. Tämän kaltaisen luontaisesti relaatio-  
muodossa olevan datan mukauttaminen MongoDB:n dokumenttimalliin on työlästä so-  
velluskehittäjän näkökulmasta. (Spencer 2020)

Myös MongoDB:n skeemattomuus on aiheuttanut ylimääräistä työtä sovelluskehittäjille.  
Kun tietokantaan tallennetulle datalle halutaan suorittaa loogisia operaatioita, tarvitaan  
yhtenäinen malli, jota sekä data, että sovelluslogiikka noudattavat. Tämän avulla välty-  
tään ajonaikaisilta virheiltä. Koska MongoDB ei pakota tietyn muotoista skeemaa, joutuu  
sovelluskehittäjä joko itse toteuttamaan sovellusdatan yhtenäisen mallintamisen sovel-  
lustasolla tai käyttämään sovellusdatan mallintamiseen kehitettyä kirjastoa. (Gallinucci  
et al. 2018; Spencer 2020)

MongoDB:n kehittäjä MongoDB Inc. on saanut jonkin verran kritiikkiä avoimen lähdekoo-  
din kehittäjäyhteisöltä MongoDB Atlas-pilvitietokantapalvelun takia. MongoDB-tietokan-  
tajärjestelmä on ollut kehityksen varhaisesta vaiheesta asti ilmainen avoimen lähdekoo-  
din sovellus. MongoDB Atlas on kuitenkin palveluna tiukan maksullisen lisenssimuurin  
takana. Tämä lisenssimuuri on rajoittanut muita alan toimijoita tarjoamasta omia pilvitie-  
tokantapalveluitaan. Osa avoimen lähdekoodin kehittäjäyhteisön jäsenistä on tämän ta-  
kia kyseenalaistanut, ansaitseeko oman projektinsa käyttöä rajoittava MongoDB Inc. tu-  
kea kehittäjäyhteisöltä. (Krazit 2020)

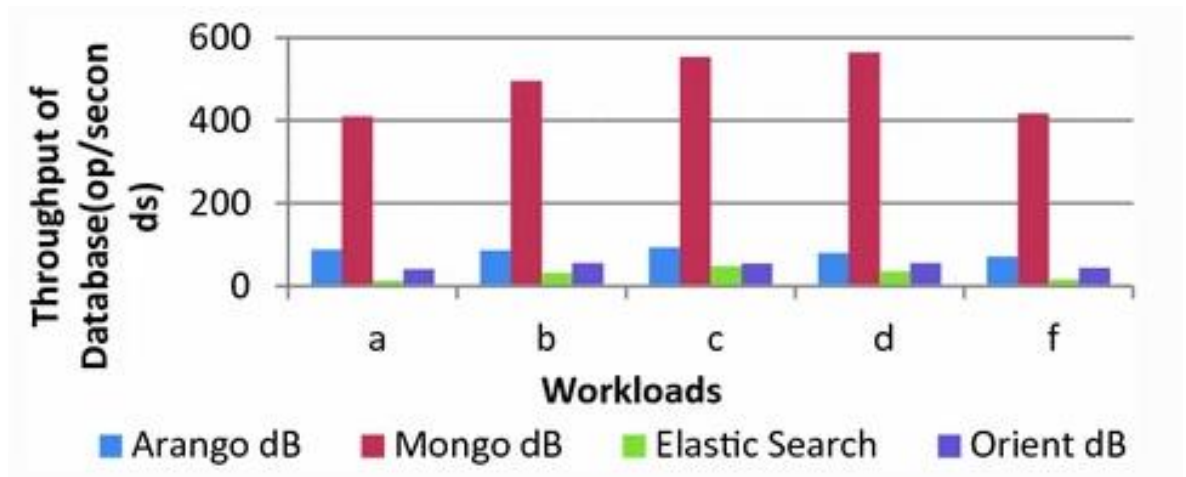
## 4. SUORITUSKYKY

Modernit sovellukset käsittelevät jatkuvalla syötteellä suuria määriä dataa. Tätä dataa täytyy pystyä käsittelemään tehokkaasti ja nopeasti. Organisaation tulee valita tietokanta, joka on tehokas ja yhteensopiva organisaation vaatimiin tarkoituksiin. Oikean tietokannan valinnassa tulee huomioida ja tutkia tietokannan suorituskykyä. MongoDB:n suorituskyvystä on tehty useita tutkimuksia, joihin tässä luvussa tutustutaan. (Mishra et al. 2018)

### 4.1 MongoDB verrattuna NoSQL-tietokantoihin

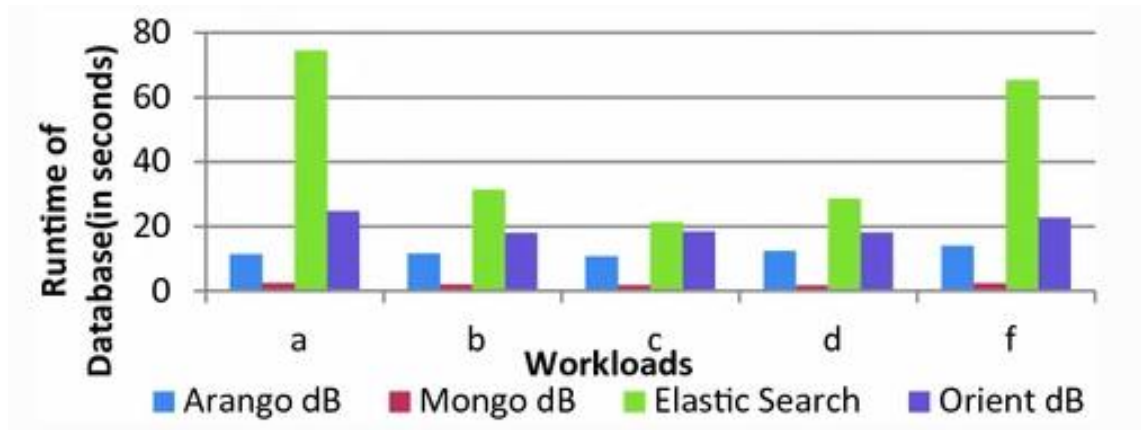
Mishra et al. (2018) vertailivat tutkimuksessaan neljän eri NoSQL-tietokannan suorituskykyä suurten datamäärien käsittelyssä. Tutkimuksen kohteena olivat MongoDB, ArangoDB, Elastic search ja OrientDB. Ensimmäisessä vertailussa tietokantojen suoritustehoa ja suoritusaikaa vertailtiin. Vertailut suoritettiin viidellä eri työkuormalla, jotka painottivat erilaisia tietokannan operaatioita. Osa työkuormista painotti lukuoperaatioita ja osa kirjoitusoperaatioita. (Mishra et al. 2018)

NoSQL-tietokantojen suorituskykymittauksen tulokset näkyvät kuvassa 4. Kuvan taulukossa y-akselilla on tietokannan suorituskyky (Throughput of Database) ja x-akselilla eri työkuormat (Workloads). Suorituskykyä mitattiin yksiköllä operaatiota sekunnissa (op/seconds). Tuloksista havaitaan, että MongoDB:llä on korkein suoritusteho jokaisella eri työkuormalla. MongoDB kykenee käsittelemään suuria määriä kaikenlaisia operaatioita muita vertailussa olevia NoSQL-tietokantoja paremmin. (Mishra et al. 2018)



Kuva 4. Tietokantojen suoritusnopeus mitattuna operaatioina sekuntia kohden. (Mishra et al. 2018)

Kuvassa 5 näkyy tietokantojen suoritusajamittauksessa saadut tulokset. Kuvan taulukossa y-akselilla on tietokannan suoritusajaksi sekunteina (Runtime of Database) ja x-akselilla eri työkuormat (Workloads). Tuloksista näkyy, että MongoDB suoriutui muita tietokantoja paremmin lyhyimmällä suoritusajalla kaikilla eri työkuormilla. MongoDB kykenee siis suorittamaan operaatioita kaikilla eri työkuormilla muita tutkimuksessa olleita tietokantoja tehokkaammin. (Mishra et al. 2018)



Kuva 5. Tietokantojen suoritusajaksi mitattuna sekunteina. (Mishra et al. 2018)

Mishra et al. (2018) suorittamassa tutkimuksessa tutkittiin myös MongoDB:n suorituskykyä, kun käytössä on useampi prosessorin säie. Tutkimuksessa MongoDB:tä verrattiin ArangoDB-tietokantaan. Vertailussa havaittiin, että MongoDB:llä on parempi suoritusnopeus ja suoritusajaksi, ja on näin parempi vaihtoehto myös silloin kun tietokanta käyttää useita säikeitä. (Mishra et al. 2018)

Mishra et al. (2018) tutkimuksessa MongoDB:n suorituskykyä verrattiin ArangoDB-tietokantaan luku- ja päivitysoperaatioita suoritettaessa. Tämän vertailun tuloksina havaittiin,



että MongoDB suoriutuu selvästi ArangoDB:tä paremmin, kun suoritetaan raskaita päivitysoperaatioita. Myös lukuoperaatioiden suorituskyky mittauksessa MongoDB osoittautui tehokkaammaksi tietokannaksi. Viimeisessä mittauksessa testattiin tehokkuutta luku-, muokkaus- ja kirjoitusoperaatioita suoritettaessa. Tässäkin mittauksessa MongoDB osoittautui tehokkaammaksi tietokannaksi lyhyemmän suoritusajan ja paremman suoritusnopeuden ansiosta. (Mishra et al. 2018)

Lopputuloksena Mishra et al. (2018) tutkimukselle havaittiin, että tutkituista NoSQL-tietokannoista MongoDB suoriutui parhaiten kaikilla eri työkuormilla. MongoDB on siis suorituskyvyn pohjalta hyvä tietokanta sovelluksille, joiden täytyy suorittaa nopeasti paljon luku- ja -kirjoitusoperaatioita. Tällaisia sovelluksia ovat esimerkiksi sosiaalisen median alustat, selaimen sessiomuisti, viestisovellukset tai rahansiirtopalvelut. (Mishra et al. 2018)

Toisessa NoSQL-tietokantojen suorituskyvyn vertailuun keskittyvässä tutkimuksessa Pereira et al. (2018) vertailivat MongoDB:n, Couchbase:n ja RethinkDB:n suorituskykyä. Tutkimuksessa tietokantojen suorituskykyä testattiin lisäys-, päivitys-, poisto- ja hakuoperaatioiden avulla käyttäen samaa datamallia kaikille tietokannoille. Tietokantojen suorituskykyä arvioitiin sekä yhden säikeen, että useamman säikeen skenaariossa. (Pereira et al. 2018)

Lisäysoperaation testeissä dokumentteja lisättiin tietokantaa. MongoDB suoriutui vertailussa mukana olleista tietokannoista parhaiten yhden säikeen ja usean säikeen testeissä. Päivitysoperaation testeissä tietokantaan tallennettuja dokumentteja muokattiin. Dokumentit haettiin niille asetetun uniikin tunnisteiden avulla. Päivitysoperaation testeissä MongoDB suoriutui parhaiten yhden säikeen vertailussa. Useamman säikeen testeissä MongoDB kuitenkin suoriutui muita tietokantoja heikommin. (Pereira et al. 2018)

Hakuoperaatioiden testeissä tietokannoista haettiin joukko dokumentteja. Ensimmäisessä testissä dokumentteja haettiin uniikin tunnisteiden perusteella. Toisessa hakuoperaatiotestissä dokumentteja haettiin satunnaisesti ilman tunnistetta. Kun dokumentteja haettiin uniikin tunnisteiden perusteella, yhden säikeen testissä tietokannat olivat suorituskyvyltään yhtä tehokkaita. Useamman säikeen testissä MongoDB osoittautui suorituskyvyltään selvästi muita tietokantoja heikommaksi. Kun dokumentteja haettiin satunnaisesti ilman tunnistetta, suoriutui MongoDB selvästi muita tietokantoja tehokkaammin sekä yhden, että useamman säikeen testeissä. (Pereira et al. 2018)

Poisto-operaatioiden testeissä tietokannasta poistettiin dokumentteja. Poistettavat dokumentit haettiin uniikin tunnisteiden avulla yksitellen. Yhden säikeen testeissä MongoDB ja

Couchbase suoriutuivat yhtä tehokkaasti. RethinkDB suoriutui selkeästi kahta muuta tietokantaa heikommin. Useamman säikeen testeissä tietokantojen suorituskyyvyissä ei ollut suuria eroja. Couchbase:n suoritusteho oli hieman muita tietokantoja parempi. (Pereira et al. 2018)

Yhden säikeen vertailuissa huomattiin, että MongoDB ja Couchbase olivat tehokkuudeltaan ja nopeudeltaan parempia kuin RethinkDB. MongoDB ja Couchbase olivat suorituskyyvyiltään lähes samantasoisia. Suurin ero tietokantojen välillä näkyi haettaessa yksittäistä dokumenttia. Kun luettiin vain yhtä dokumenttia pyyntöä kohden Couchbase oli nopeampi. Kun taas luettiin useaa dokumenttia pyyntöä kohden MongoDB oli huomattavasti Couchbasea nopeampi. (Pereira et al. 2018)

Useamman säikeen testeissä testattiin tietokantojen suorituskyykyä vaativassa tilanteessa. Näissä testeissä huomattiin, että MongoDB suoriutui parhaiten dokumenttien lisäysoperaatioissa. Päivitysoperaatioissa MongoDB suoriutui huomattavasti muita tietokantoja heikommin. MongoDB:n vastausaika oli lähes kaksi kertaa suurempi kuin Couchbasen. Hakuoperaatioissa havaittiin myös eroavaisuuksia. Yhtä dokumenttia haettaessa MongoDB suoriutui heikommin kuin kaksi muuta tietokantaa, mutta haettaessa useita dokumentteja kerralla MongoDB:n suoritusteho ja vastausaika oli parempi kuin muilla tietokannoilla. Poisto-operaatioissa Couchbase suoriutui paremmin kuin MongoDB. (Pereira et al. 2018)

Pereira et al. (2018) saamien tulosten perusteella voidaan todeta, että Couchbase suoriutuu MongoDB:tä paremmin suurimmassa osassa operaatioita. Poikkeuksina ovat useamman dokumentin hakuoperaatio ja lisäysoperaatio, joissa MongoDB suoriutui huomattavasti paremmin. MongoDB sopii siis paremmin tietokannaksi sovelluksille, jotka suorittavat paljon usean dokumentin hakuoperaatioita ja lisäysoperaatioita. (Pereira et al. 2018)

## 4.2 MongoDB verrattuna relaatiotietokantoihin

Makris et al. (2020) tutkimus vertasi MongoDB:n suorituskyykyä PostgreSQL:ään, joka on suosittu relaatiotietokanta. Tietokantojen suorituskyykyä testattiin laivojen olinpaikkoja jäljittävän järjestelmän paikkatietodatan ja oikeisiin käyttöympäristöihin perustuvien kyselyiden avulla mittaamalla kyselyihin kuluva vastausaika. Tutkimuksessa huomattiin, että PostgreSQL suoriutuu MongoDB:tä paremmin lähes jokaisessa kyselyssä. Tutkimuksessa havaittiin myös, että indeksoinnin käyttö MongoDB:n kanssa laskee vastausai-

kaa merkittävästi. Indeksoinnilla ei saavutettu yhtä suurta etua PostgreSQL:n tapauksessa. Merkittävänä huomiona oli myös testidatan vaatima tila eri tietokannoissa. Testidatan koko oli neljä kertaa pienempi PostgreSQL:ää käytettäessä. (Makris et al. 2020)

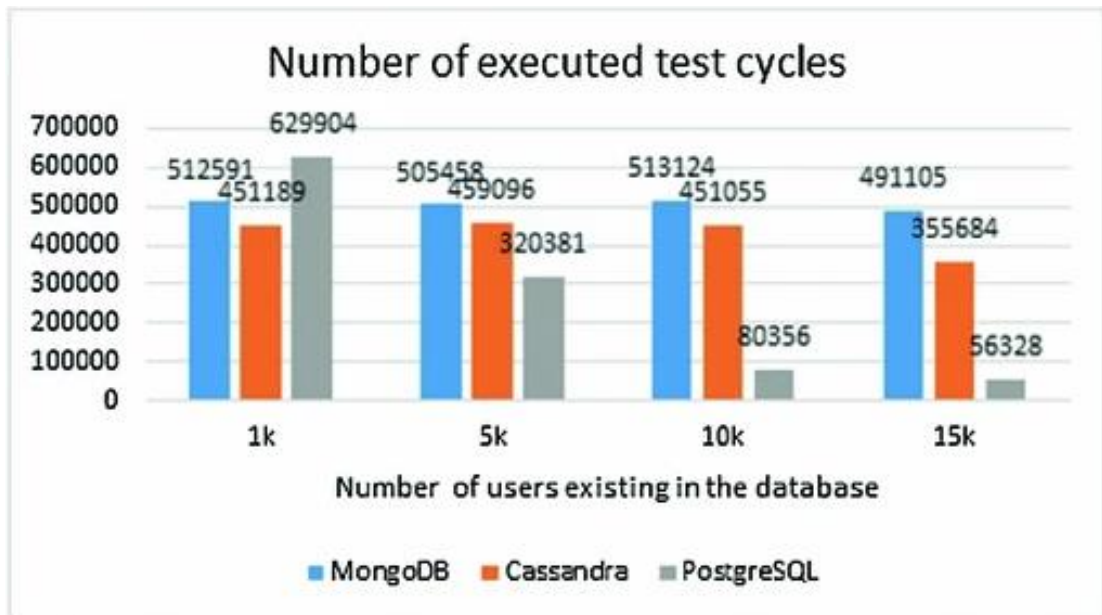
Jose & Abraham (2020) vertailivat tutkimuksessaan MongoDB:n suorituskykyä MySQL-relaatiotietokantaa. Vertailussa tietokannoille suoritettiin haku-, lisäys- ja päivitysopeeraatioita. Vertailussa molempia tietokantoja testattiin kahdella erikokoisella datajoukolla. Suorituskykyä mitattiin tutkimalla operaatioiden suoritukseen kuluva suoritusajaa. (Jose & Abraham 2020)

Yksinkertaisessa hakuoperaatiossa MongoDB suoriutui tehokkaammin molemmilla datajoukoilla testattaessa. MongoDB:n suorituskyky oli parempi myös silloin kun hakuoperaatiolle annettiin jokin ehtolause. Päivitysopeeraatioiden testeissä tietokantojen tehokkuutta testattiin kolmella eri kyselylauseella ja kahdella eri datajoukolla. Myös päivitysopeeraatioissa MongoDB suoriutui selvästi tehokkaammin kaikilla eri kyselylauseilla ja molemmilla datajoukoilla. Lisäysoperaatioiden testeissä huomattiin, että MongoDB suorittaa lisäysoperaation nopeammin molemmilla datajoukoilla. (Jose & Abraham 2020)

Tutkimuksen tulosten perusteella huomattiin, että suuria datajoukkoja käsiteltäessä MongoDB kykenee hyvään tehokkuuteen ja skaalautuvuuteen suurimmalle osalle suoritettavista operaatioista. Kaikissa tutkimuksen tehokkuustesteissä MongoDB suoriutui MySQL-relaatiotietokantaa tehokkaammin. Tämän tehokkuusedun perusteella tutkimuksessa todettiin MongoDB:n soveltuvan MySQL-tietokantaa paremmin suurien datamäärien käsittelyyn. (Jose & Abraham 2020)

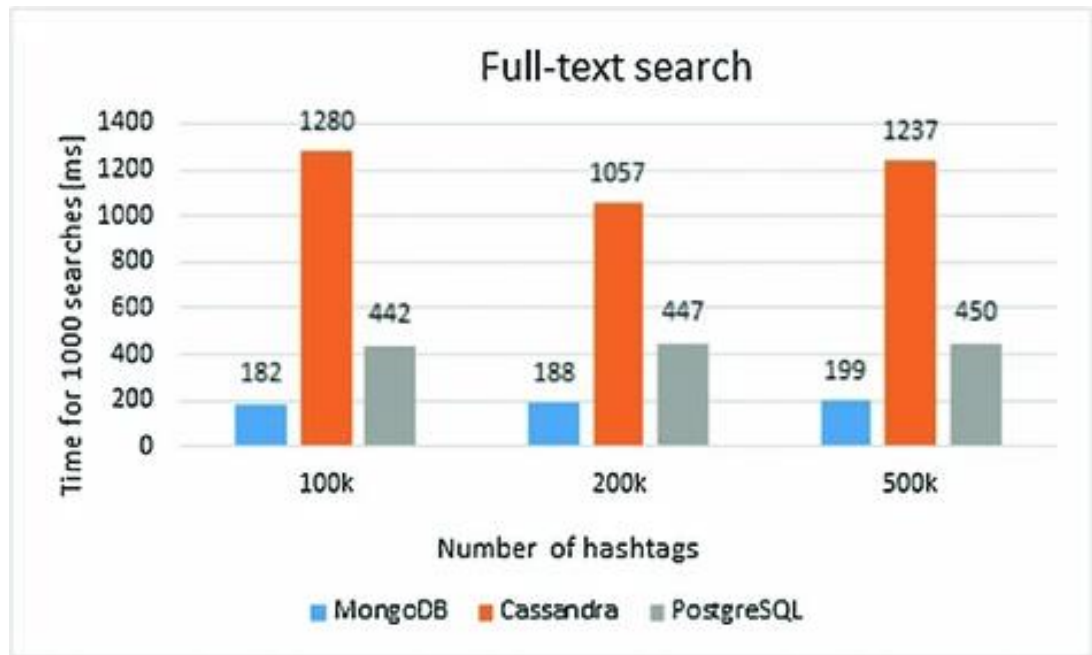
Fraczek & Plechawska-Wojcik (2017) vertasivat tutkimuksessaan PostgreSQL:n, MongoDB:n ja Cassandra-tietokannan suorituskykyä. Suorituskyvyn testaamista varten kehitettiin sosiaalisen median web-sovellus. Jokaiselle tietokannalle kehitettiin omat datamallit. MongoDB:n datamalli oli näistä datamalleista yksinkertaisin sisäkkäisten dokumenttien ja taulukoiden käytön ansiosta. (Fraczek & Plechawska-Wojcik 2017)

Tietokantojen suorituskykyä testattiin simuloimalla sovelluksen käyttöä usealla käyttäjällä samanaikaisesti. Jokainen testi kesti 5 minuuttia. Jokaista tietokantaa testattiin neljällä eri dataryhmällä, jotka sisälsivät eri määrän käyttäjiä ja postauksia. Tehokkuutta mitattiin laskemalla, kuinka monta sykliä kukin tietokanta ehtii suorittaa 5 minuutin aikana. Testin tulokset näkyvät kuvassa 6. Kuvassa y-akselilla on suoritettujen testisykliiden määrä ja x-akselilla eri dataryhmien käyttäjämäärä. Testin tuloksina huomattiin, että pienellä datamäärällä PostgreSQL oli nopein tietokanta. Suurilla datamäärillä MongoDB havaittiin tehokkaimmaksi tietokannaksi. (Fraczek & Plechawska-Wojcik 2017)



Kuva 6. Tietokantojen suorittamat testisyklit eri kokoisilla datamäärillä. (Fraczek & Plechawska-Wojcik 2017)

Tutkimuksessa testattiin myös tietyllä mallilla alkavien hashtagien etsimistä. Jokaiselle tietokannalle tehtiin kolme testiä eri määrällä hashtageja. Kuvassa 7 on esitetty, kuinka kauan kultakin tietokannalta kului aikaa suorittaa 1000 täyttä hakuoperaatiota. Kuvassa y-akselilla on kulunut aika millisekunteina 1000 hakuoperaatiota kohden. Kuvan x-akselilla on hashtagien määrä. Kuvasta havaitaan, että MongoDB on noin kaksi kertaa nopeampi kuin PostgreSQL. (Fraczek & Plechawska-Wojcik 2017)



Kuva 7. Tietokantojen käyttämä aika 1000 hakuoperaatiolle. (Fraczek & Plechawska-Wojcik 2017)

Fraczek & Plechawska-Wojcik (2017) tutkimuksessa havaittiin, että MongoDB suoriutui relaatiomallia hyödyntävää PostgreSQL-tietokantaa paremmin kaikissa muissa operaatioissa paitsi datan kirjoittamisessa, jossa PostgreSQL oli tehokkaampi. Jose & Abraham (2020) havaitsivat tutkimuksessaan, että MongoDB suoriutuu haku-, päivitys- ja lisäysoperaatioissa relaatiomallia edustavaa MySQL-tietokantaa tehokkaammin. Makris et al. (2020) taas havaitsivat omassa tutkielmassaan PostgreSQL:n tehokkaammaksi lähes kaikissa tehokkuustesteissä. Antas et al. (2022) havaitsivat tutkimuksessaan relaatiotietokantojen olevan parempi tietokantaratkaisu, kun data on rakenteellista ja vaatii paljon liitosoperaatioiden suorittamista. MongoDB osoittautui tehokkaimmaksi tietokannaksi, kun datamäärät olivat suuria ja data ei ollut rakenteellista, eikä vaatinut liitosoperaatioiden suorittamista. MongoDB:llä saavutettavat tehokkuusedut verrattaessa relaatiotietokantoihin vaihtelevat siis käyttötarkoituksen ja käsiteltävän datan mukaan.

## 5. YHTEENVETO

Tämän tutkielman tarkoituksena oli perehtyä MongoDB-dokumenttitietokantaan sovel-  
luskehityksen näkökulmasta. Tavoitteena oli löytää syitä MongoDB:n viimeisen vuosi-  
kymmenen aikana kasvaneeseen suosioon. Tutkimuksessa perehdyttiin myös käyttötar-  
koituksiin, joissa MongoDB ei kykene tarjoamaan perinteisiä relaatiomallin tietokantoja  
parempaa ratkaisua.

Luvussa 3 tutustuttiin MongoDB:n ominaisuuksiin ja heikkouksiin aiheesta julkaistujen  
tutkimusten ja kirjallisuuden avulla. Havaittiin, että MongoDB:n dokumenttipohjainen da-  
tamalli tarjoaa yksinkertaisuutensa takia nopeaa datan saatavuutta ja tietokannan skaa-  
lautuvuutta. Toisaalta MongoDB:n dokumenttimalli aiheuttaa myös ongelmia, sillä osa  
sovellusten säilömästä datasta on luontaisesti relaatiomallista. Tällaisen datan muunta-  
minen dokumenttimalliin on sovelluskehittäjän kannalta työlästä. Luvussa 3 havaittiin  
myös MongoDB:n skeemattomuuden aiheuttamia ongelmia. Koska MongoDB ei pakota  
yhtenäisen datamallin käyttämiseen, jää datamallin yhtenäisyyden valvominen sovellus-  
kehittäjän vastuulle ylimääräiseksi työksi.

Luvussa 4 MongoDB:n suorituskykyä vertailtiin kilpaileviin NoSQL- ja relaatiotietokantoi-  
hin julkaistujen tutkimusten avulla. NoSQL-tietokantoihin verrattaessa havaittiin, että  
MongoDB on suorituskyvyltään erityisen tehokas suoritettaessa haku- ja lisäysoperaati-  
oita. Tulosten pohjalta pääteltiin, että MongoDB sopii suorituskykynsä puolesta hyvin so-  
velluksille, joiden täytyy suorittaa paljon haku- ja lisäysoperaatioita. Luvussa 4 Mon-  
goDB:n suorituskykyä vertailtiin myös PostgreSQL-tietokantaan ja MySQL-tietokantaan,  
jotka molemmat edustavat relaatiotietokantoja. Kahden PostgreSQL:ään keskittyvän tut-  
kimuksen tulokset olivat ristiriitaisia. Näiden tulosten johdolla pääteltiin, että MongoDB:llä  
saavutettavat tehokkuusedut verrattuna PostgreSQL:ään vaihtelevat käsiteltävän datan  
ja sovelluksen käyttötarkoituksen mukaan. MySQL-tietokantaan keskittyvän tutkimuksen  
tulosten perusteella havaittiin, että MongoDB on tehokkaampi tietokanta haku-, päivitys-  
ja lisäysoperaatioissa.

Tutkimuksen johdolla havaittiin, että MongoDB soveltuu horisontaalisen skaalautuvuu-  
tensa ansiosta suuria datamääriä käsitteleville sovelluksille. MongoDB:n skeemattomu-  
uden tarjoama joustavuus tietokannan arkkitehtuurissa mahdollistaa nopean tietoraken-  
teiden muuttamisen. Tämän ansiosta MongoDB sopii hyvin tietokannaksi sovelluksille,  
joiden kehitys on nopeaa ja tietokannan arkkitehtuuria joudutaan muuttamaan useita

kertoja kehityksen eri vaiheissa. Etenkin modernit sovellukset hyötyvät paljon MongoDB:n tarjoamista ominaisuuksista. Tämän takia MongoDB on noussut viimeisen vuosikymmenen aikana yhdeksi suosituimmista tietokantajärjestelmistä.

# LÄHTEET

- Antas, J., Silva, R. R., & Bernardino, J. (2022). Assessment of SQL and NoSQL Systems to Store and Mine COVID-19 Data. *Computers (Basel)*, 11(2), 29–. <https://doi.org/10.3390/computers11020029>
- Bradshaw, Brazil, E., & Chodorow, K. (2019). *MongoDB: The Definitive Guide, 3rd Edition (3rd edition)*. O'Reilly Media, Inc.
- Chen, & Lee, W.-Z. (2019). An introduction of NoSQL databases based on their categories and application industries. *Algorithms*, 12(5), 106–. <https://doi.org/10.3390/a12050106>
- Dechev, M., Galabov, M., & Georgieva-Trifonova, T. (2019). Application of Non-Relational Databases in the Social Networks. *Brain. Broad Research in Artificial Intelligence and Neuroscience*, 10(4), 56–72. <https://doi.org/10.18662/brain/05>
- Diogo, M., Cabral, B., & Bernardino, J. (2019). Consistency models of NoSQL databases. *Future Internet*, 11(2), 43–. <https://doi.org/10.3390/fi11020043>
- Fraczek, & Plechawska-Wojcik, M. (2017). Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications. *BEYOND DATABASES, ARCHITECTURES AND STRUCTURES: TOWARDS EFFICIENT SOLUTIONS FOR DATA ANALYSIS AND KNOWLEDGE REPRESENTATION*, 716, 153–164. [https://doi.org/10.1007/978-3-319-58274-0\\_13](https://doi.org/10.1007/978-3-319-58274-0_13)
- Gallinucci, Golfarelli, M., & Rizzi, S. (2018). Schema profiling of document-oriented databases. *Information Systems (Oxford)*, 75, 13–25. <https://doi.org/10.1016/j.is.2018.02.007>
- Hostingdata.co.uk. (2022). List of NoSQL Database Management Systems. *Hostingdata.co.uk* <https://hostingdata.co.uk/nosql-database/> (Haettu 13.10.2022).
- Jia, & Shen, G. C. (2015). Research on Relational Database and NoSQL Based on XML Data. *Applied Mechanics and Materials*, 713-715(Mechatronics Engineering and Modern Information Technologies in Industrial Engineering), 2329–2334. <https://doi.org/10.4028/www.scientific.net/AMM.713-715.2329>
- Jose, B., & Abraham, S. (2020). Performance analysis of NoSQL and relational databases with MongoDB and MySQL. *MATERIALS TODAY-PROCEEDINGS*, 24, 2036–2043. <https://doi.org/10.1016/j.matpr.2020.03.634>
- Json.org (2022). Introducing JSON. *json.org* <https://www.json.org/json-en.html> (Haettu 17.10.2022)
- Klein, Gorton, I., Ernst, N., Donohoe, P., Pham, K., & Matser, C. (2015). Performance Evaluation of NoSQL Databases: A Case Study. *PABS 2015 - Proceedings of the 1st ACM/SPEC International Workshop on Performance Analysis of Big Data Systems*, 5–10. <https://doi.org/10.1145/2694730.2694731>
- Krazit, T. (2020). Does MongoDB regret offending the open-source community? Not one bit. *Protocol* <https://www.protocol.com/enterprise/mongodb-open-source-database> (Haettu 21.10.2022)
- Makris, Tserpes, K., Spiliopoulos, G., Zissis, D., & Anagnostopoulos, D. (2020). MongoDB Vs PostgreSQL: A comparative study on performance aspects. *Geoinformatica*, 25(2), 243–268. <https://doi.org/10.1007/s10707-020-00407-w>



Mishra, Lodhi, P., & Mehta, S. (2018). Document Oriented NoSQL Databases: An Empirical Study. *Communications in Computer and Information Science*, 799, 126–136. [https://doi.org/10.1007/978-981-10-8527-7\\_12](https://doi.org/10.1007/978-981-10-8527-7_12)

Oracle (2022). What Is a Database. Oracle <https://www.oracle.com/database/what-is-database/> (Haettu 11.10.2022)

Oracle - RDBMS (2022). What Is a Relational Database (RDBMS)? Oracle <https://www.oracle.com/database/what-is-a-relational-database/> (Haettu 26.10.2022)

Pereira, D. A., Ourique de Moraes, W., & Pignaton de Freitas, E. (2018). NoSQL real-time database performance comparison. *International Journal of Parallel, Emergent and Distributed Systems*, 33(2), 144–156. <https://doi.org/10.1080/17445760.2017.1307367>

Spencer, J. (2020). Why MongoDB is (probably) the Wrong Choice for Your Next App. *spencerjones.blog* <https://spencerjones.blog/why-not-mongo/> (Haettu 19.10 2022)

W3Schools (2022). MySQL RDBMS. W3Schools [https://www.w3schools.com/mysql/mysql\\_rdbms.asp](https://www.w3schools.com/mysql/mysql_rdbms.asp) (Haettu 26.10.2022)