Tampereen yliopisto

Otto Thitz

GRAAFITIETOKANTOJEN SOVELLUKSIA

Systemaattinen kirjallisuuskatsaus

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Huhtikuu 2023

TIIVISTELMÄ

Otto Thitz: Graafitietokantojen sovelluksia: systemaattinen kirjallisuuskatsaus
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Huhtikuu 2023

Tässä työssä kartoitetaan akateemisessa tutkimuksessa esiintyviä graafitietokantoja, niiden sovellusaloja sekä niihin liitettyjä hyötyjä ja haittoja. Tutkimusmenetelmänä on systemaattinen kirjallisuuskatsaus, jossa tunnistettiin 111 kriteerit täyttävää artikkelia vuosilta 2017–2021. Artikkeleja analysoitiin sisällönanalyysin keinoin.

Graafitietokantojen sovellusaloja tunnistettiin 25. Sovellusaloilla tieto on tyypillisesti mallinnettavissa kompleksisina verkkoina. Yleisimpiä aloja olivat bioinformatiikka, sosiaaliset verkostot, tietoverkot ja geografinen tieto. Yksittäisistä graafitietokannoista ylivoimaisesti käytetyin oli Neo4j: se oli käytössä valtaosassa artikkelien sovelluksista. Muut graafitietokannat olivat edustettuna vähäisessä määrin aineistossa.

Graafitietokantojen käytölle tunnistettiin kymmenen hyötyä. Yleisimmin mainitut hyödyt olivat graafikyselyiden ja -algoritmien hyödyntäminen sekä graafitietokantojen soveltuvuus verkottuneelle datalle. Näiden jälkeen yleisimpinä hyötyinä tulivat selitysvoima erilaisissa analyyseissa, suorituskyky, visualisointiominaisuudet, tietokantakaavion joustavuus ja graafitietomallin ymmärrettävyys.

Eri haittoja puolestaan tunnistettiin yhdeksän: haittoja mainittiin kuitenkin ylipäänsä huomattavasti hyötyjä harvemmin. Yleisimmin mainitut haitat olivat suorituskyky ja graafitietokantojen opettelu: molemmat oli mainittu kohtalaisen usein myös hyötynä. Tätä voi selittää sillä, että graafitietokantojen suorituskyvyssä on eroja eri sovellusten välillä: graafitietokantojen ja -kyselykielten koettu vaikeustaso taas riippuu tutkijoiden näkemyksistä. Lisäksi harvemmin mainittuja haittoja olivat muun muassa graafitietokantojen soveltumattomuus tietynlaiselle datalle ja alempi kypsyysaste verrattuna relaatiotietokantoihin.

Avainsanat: graafitietokanta, tietokannat, tietomallit, kyselykielet, NoSQL, systemaattinen kirjallisuuskatsaus

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

1 Johdanto

Graafit ovat luontainen tapa esittää tietoa sovellusaloilla, jossa tieto on luonteeltaan verkottunutta [Angles & Gutierrez 2018]. Viime vuosina käsiteltävän datan määrä on kasvanut suuresti erityisesti tällaisilla aloilla: usein mainittu esimerkki tästä on sosiaalisten verkostojen analyysi [Tian 2023]. Tätä myötä myös kysyntä graafien käsittelyyn soveltuvia teknologioita kohtaan on lisääntynyt [Besta *et al.* 2019]. Tässä työssä käsitellään *graafitietokantoja* (graph database), jotka ovat yksi keskeinen teknologia graafimaisen datan hallintaan ja analysointiin [Larriba-Pey *et al.* 2014].

Graafitietokannat ovat tietokantoja, jotka on järjestetty *graafitietomallin* (graph database model) mukaisesti [Angles & Gutierrez 2018]. Graafitietomalli pohjautuu *graafiteoriaan* (graph theory): graafeissa olevat entiteetit esitetään *solmuina* (node) ja niiden väliset suhteet *kaarina* (edge) [Angles *et al.* 2017]. Graafitietomallissa tietorakenteet on mallinnettu graafeina, datan manipulointi tehdään graafisuuntautuneilla operaatioilla *graafikyselykielellä* (graph query language) ja graafin rakenne varmistetaan eheysrajoitteilla [Angles & Gutierrez 2008].

Graafitietomalli soveltuu erityisesti tilanteisiin, jossa datan sisäinen linkittyneisyys tai topologia on kiinnostuksen kohteena [Angles & Gutierrez 2018]. Graafitietomallin hyötyinä on muun muassa esitetty, että se mahdollistaa verkkomaisen datan luonnollisen mallinnuksen, kyselyitä voidaan kohdistaa suoraan graafirakenteisiin ja sen myötä voidaan käyttää tehokkaasti erilaisia graafiteorian algoritmeja [Angles & Gutierrez 2008]. Graafitietomallia on sovellettu menestyksekkäästi esimerkiksi sosiaalisen median, semanttisen webin ja biologian aloilla [Barcelo *et al.* 2011].

Tässä työssä tarkastellaan akateemisessa tutkimuksessa esitettyjä graafitietokantojen sovelluksia sekä graafitietokantoihin liitettyjä hyötyjä ja haittoja *systemaattisen kirjallisuuskatsauksen* (systematic literature review) kautta. Tutkimuskysymykset ovat:

1. Millä sovellusaloilla graafitietokantoja käytetään?
2. Mitä graafitietokantoja sovelluksissa käytetään?
3. Mitä hyötyjä graafitietokantoihin liitetään?
4. Mitä haittoja graafitietokantoihin liitetään?

Työn rakenne on seuraava: johdannon (luku 1) jälkeen käydään läpi graafitietokantoja yleisesti aiemman kirjallisuuden pohjalta (luku 2). Luvussa 3 tarkastellaan tutkimus-asetelmaa ja kuvataan tutkimusprosessi. Tämän jälkeen esitellään tutkimuksen tulokset tutkimuskysymyksittäin (luku 4). Luvussa 5 esitetään työn pohdinta: ensin verrataan tuloksia aiempaan tutkimukseen, sitten pohditaan työn rajoitteita ja esitetään jatko-tutkimusideoita. Lopuksi luvussa 6 esitetään tutkimuksen yhteenveto.

2 Tutkimuksen viitekehys

Tässä luvussa käsitellään graafitietokantoja aiemman kirjallisuuden valossa: tavoitteena on raamittaa tälle tutkimukselle viitekehys. Kohdassa 2.1 käsitellään graafitietomallia: ensin käsitellään tietomalleja yleisesti ja sitten graafitietomallin määritelmää. Graafitietomalleista käsitellään tarkemmin kahta suosittua mallia: *ominaisuusgraafi-tietomallia* (property graph model) ja *RDF-tietomallia* (RDF model). Kohdassa 2.2 käsitellään graafidatan hallintajärjestelmiä ja niiden osalta graafitietokantoja sekä RDF-tietokantoja. Kohdassa 2.3 käsitellään graafitietokantojen sovellusaloja aiemman tutkimuksen sekä graafitietokantojen valmistajien materiaalien pohjalta. Lopuksi kohdassa 2.4 käsitellään graafitietokantojen ominaisuuksia aiemman tutkimuksen perusteella.

2.1 Graafitietomalli

Tietokantojen toteutus sisältää useimmille loppukäyttäjille tarpeettoman yksityiskohtaista tietoa: toteutus abstrahoidaan ymmärrettävämpään muotoon *tietomallin* (data model) avulla [Elmasri & Navathe 2004, s. 10]. Silberschatz ja muut [1996] määrittelevät tietomallin olevan kokoelma käsitteellisiä työkaluja, joilla kuvataan tosimaailman entiteettejä ja niiden välisiä suhteita tietokannassa. Coddin [1980] mukaan tietomallin voidaan nähdä koostuvan kolmesta komponentista: tietorakennetyyppien, operaattorien ja eheysrajoitteiden kokoelmista.

Varhaiset tietomallit, kuten hierarkkinen tietomalli [Tsichritzis & Lochovsky 1976] ja verkkotietomalli [Taylor & Frank 1976], keskittyivät datan mallintamiseen fyysisellä tasolla. Näistä malleista puuttui kunnollinen abstraktiotaso: tietomallia oli vaikea erottaa varsinaisesta toteutuksesta. Tietorakenteet eivät olleet joustavia eivätkä mahdollistaneet kuin perinteisiä sovelluksia. Tietokantojen navigointi tapahtui *tietueiden* (record) tasolla ja niistä pystyttiin johtamaan abstraktimpaa tietoa matalan tason operaatioiden avulla. [Angles & Gutierrez 2008]

Coddin [1970] esittelemä *relaatiotietomalli* (relational data model) on merkittävä taitekohta tietomallien kehityksessä, koska se erottelee datan fyysisen varastoinnin ja käsitteellisen esittämisen [Angles & Gutierrez 2008]. Relaatiotietomalli on uraauurtava myös siinä mielessä, että se perustuu matematiikkaan ja esittää datan relaatioina, jotka voidaan havainnollistaa taulukkomuodossa [Silberschatz *et al.* 1991]. Ajan myötä relaatiotietokannoista on tullut dominoiva teknologia perinteisissä tietokantasovelluksissa [Elmasri & Navathe 2004, s. 21], mitä ne ovat edelleen myös vuonna 2023 [Solid IT 2023]. Elmasrin ja Navathen [2004, s. 207] mukaan yksi keskeinen relaatiotietokantojen voittokulkuun vaikuttanut syy on niille kehitetty standardoitu kyselykieli SQL (Structured Query Language), jonka ensimmäisen version esittivät Chamberlin ja Boyce [1974].

Varhaiset tietomallit (mukaan lukien relaatiotietomalli) eivät kuitenkaan ole kaikin tavoin optimaalisia kompleksisen datan mallintamiseen. Tähän vastauksena 80-luvulla alettiin tutkia esimerkiksi *oliopohjaisia* (object-oriented) tietomalleja sekä graafirakenteita käyttäviä tietokantaratkaisuita. Oliopohjaisen paradigman ideana oli mallintaa data kokoelmana olioita, jotka ovat luokkien jäseniä ja pystyvät tallentamaan runsaasti tietoa. Oliopohjaiset tietomallit ovat graafitietomalleille sukua siinä mielessä, että myös niissä hyödynnetään graafirakenteita määritelmissä. Oliopohjaisissa tietomalleissa painoarvo on kuitenkin monimutkaista tietoa sisältävissä olioissa (entiteeteissä), kun taas graafitietomalleissa ollaan kiinnostuneita etenkin suhteista ja pyritään tallentamaan datan verkkomainen luonne. [Angles & Gutierrez 2008]

Anglesin ja Gutierrezin [2008] mukaan ensimmäisiä graafien sovelluksia tietokannoissa olivat Roussopoulosin ja Mylopoulosin [1975] kehittämä semanttinen verkko, Shipmanin [1981] kehittämä Functional Data Model, jossa käytettiin graafirakenteita implisiittisesti, sekä Kuperin ja Vardin [1984] kehittämä Logical Data Model, joka puolestaan hyödynsi graafirakenteita eksplisiittisesti ja jota voi kutsua jo graafitietomalliksi. Yhtenä varhaisena graafitietomallina voidaan mainita myös Kuniin [1987] kehittämä tietomalli kompleksisen datan esittämiseen G-BASE-tietokannassa. Sittemmin graafitietomalleja ja graafitietokantoja kehitettiin useissa tutkimuksissa erityisesti 90-luvun alkupuolella [Angles & Gutierrez 2008]. Kuitenkin 90-luvun puolivälissä kiinnostus graafitietokantoja kohtaan väheni luultavasti sen takia, että sen aikaisille tietokoneille isojen graafien prosessointi oli liian raskasta [Angles & Gutierrez 2018].

Anglesin ja Gutierrezin [2018] mukaan graafitietokantojen toinen aalto alkoi 2010-luvulla: ensinnäkin sen mahdollisti tietokonelaitteistojen suorituskyvyn kehittyminen [Angles & Gutierrez 2018], ja toisaalta sen tajuaminen, että monella alalla graafitietomallin avulla tietoa voidaan käsitteellistää intuitiivisemmin kuin relaatiotietomallilla [Angles *et al.* 2017]. Toisen aallon aikana useat kaupalliset graafitietokannat ovat vakiinnuttaneet asemansa [Amazon.com 2022; Neo4j 2019; TigerGraph 2023e]. Akateemisesta tutkimusta on myös tehty runsaasti: siinä on keskitytty etenkin graafikyselykielten analysointiin ja kehittämiseen [Angles & Gutierrez 2018].

Graafitietomallin määritelmä

Graafitietokannan takana oleva tietomalli on graafitietomalli. Coddin [1980] määritelmää seuraten Angles ja Gutierrez [2008] määrittelevät graafitietomallin olevan tietomalli, jossa tietorakenteet on mallinnettu graafeina, datan manipulointi tehdään graafisuuntautuneina operaatioilla graafikyselykielellä ja graafin rakenne varmistetaan eheysrajoitteilla.

Ensimmäisenä graafitietomallin piirteenä Angles ja Gutierrez [2018] mainitsevat tietorakenteiden mallintamisen graafeina. Tarkemmin sanottuna tämä tarkoittaa, että datan ja/tai tietokantakaavion tietorakenteet on esitetty graafeina tai graafien yleistyksinä (kuten *hypergraafeina* tai *hypernoodeina*). Graafitietomalleissa tietokantakaavion ja

datan (instanssien) erotteleminen ei ole niin selvää kuin perinteisessä relaatiotietomallissa. Graafirakenteen ansiosta strukturoimattoman datan käsittely on kohtuullisen yksinkertaista. [Angles & Gutierrez 2018]

Toisekseen graafitietomallia määrittää se, että datan manipulointi ja kyselyt tehdään graafisuuntautuneilla operaatioilla, jotka muodostavat graafikyselykielen. Operaatiot voivat hyödyntää graafien ominaisuuksia kuten *polkuja* (paths), *naapurustoja* (neighborhoods) ja *aligraafeja* (subgraphs) sekä graafien tunnuslukuja kuten *halkaisijaa* (diameter) ja *keskeisyyttä* (centrality). Anglesin ja Gutierrezin mukaan kyselykieli antaa yleensä tietomallille sen tunnusomaiset piirteet. Eri graafitietomalleissa kyselykielten väliset erot voivat olla merkittäviä, kun taas rakenteiden väliset erot ovat yleensä pieniä. [Angles & Gutierrez 2018]

Kolmas graafitietomallia määrittävä tekijä ovat eheysrajoitteet, jotka varmistavat datan *johdonmukaisuuden* (consistency). Näitä rajoitteita ovat esimerkiksi *tietokantakaavion ja instanssien välinen johdonmukaisuus* (schema-instance consistency) sekä *identiteetti-* ja *viite-eheys* (identity and referential integrity). Esimerkkejä näistä ovat *nimiöiden* (label) ainutlaatuiset arvot sekä solmujen tyyppirajoitteet. [Angles & Gutierrez 2018]

Erilaisia graafitietomalleja on useita: kaikki kuitenkin perustuvat graafien matemaattiseen määritelmään. Graafitietomallien välisiä eroja ovat esimerkiksi seuraavat seikat: ovatko graafit suunnattuja vai suuntaamattomia, ovatko kaaret ja solmut nimiöityjä vai nimiöimättömiä ja voiko kaarilla ja solmuilla olla ominaisuuksia vai ei. Seuraavaksi käydään läpi tarkemmin kaksi suosittua graafitietomallia: ominaisuusgraafitietomalli ja RDF-tietomalli. Tämän työn tarkastelun ulkopuolelle jäävät esimerkiksi hypergraafeihin ja hypernoodeihin perustuvat tietomallit. [Angles & Gutierrez 2018]

Ominaisuusgraafitietomalli

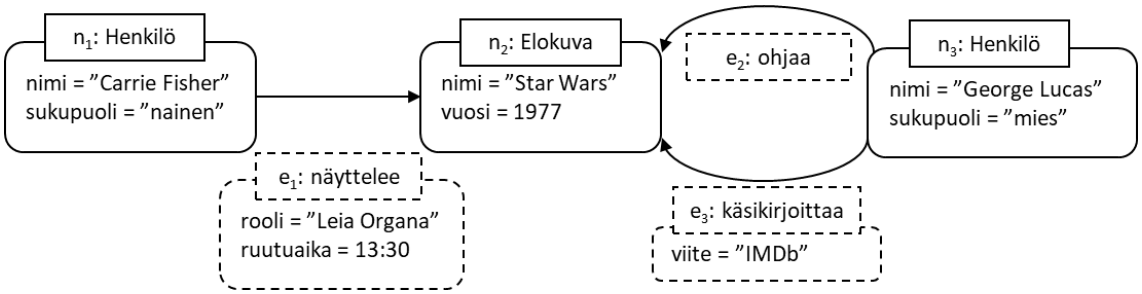
Ominaisuusgraafi (property graph) on suunnattu, nimiöity, attribuutillinen multigraafi [Angles & Gutierrez 2018]. Tämä tarkoittaa, että kaaret ovat suunnattuja, solmuilla ja kaarilla on nimiöt, solmuilla ja kaarilla voi olla mielivaltainen määrä ominaisuuksia (attribuutteja) ja kahden solmun välillä voi olla mielivaltainen määrä kaaria [Rodriguez & Neubauer 2010]. Ominaisuudet ovat avain-arvopareja, jotka esittävät solmujen ja kaarten metadataa [Angles & Gutierrez 2018].

Anglesin [2018] mukaan ominaisuusgraafitietomalli on eniten käytetty graafitietomalli: ominaisuusgraafeja käytetään yleensäkin tietojenkäsittelyssä runsaasti, sillä ne ovat ilmaisuvoimaisempia kuin yksinkertaisemmat matemaattiset objektit [Angles & Gutierrez 2018]. Ominaisuusgraafimalli pystyykin esittämään muut graafitietomallit hylkäämällä ja lisäämällä tarvittavia osia [Rodriguez & Neubauer 2010].

Angles [2018] esittää ominaisuusgraafin formaalina määritelmänä, että ominaisuusgraafi on monikko $G = (N, E, \rho, \lambda, \sigma)$, jossa

1. N on graafin solmujen joukko;
2. E on graafin kaarten joukko;
3. $\rho : E \rightarrow (N \times N)$ on funktio, joka määrittää kaarten suunnat liittämällä jokaisen kaaren joukossa E solmupariin, joka on muodostettu joukosta N;
4. $\lambda : (N \cup E) \rightarrow SET^{+}(\boldsymbol{L})$ on funktio, joka määrittää solmujen ja kaarten nimiöt graafin nimiöiden joukosta $\boldsymbol{L}$;
5. $\sigma : (N \cup E) \times \boldsymbol{P} \rightarrow SET^{+}(\boldsymbol{V})$ on funktio, joka määrittää solmujen ja kaarten attribuutit graafin attribuuttien nimien joukosta $\boldsymbol{P}$ ja arvojen joukosta $\boldsymbol{V}$.

Kuvassa 2.1 on esitetty esimerkki ominaisuusgraafista: graafissa on kolme solmua $\{n_1, n_2, n_3\}$ sekä kolme kaarta $\{e_1, e_2, e_3\}$. Kahden solmun välillä voi olla useita kaaria: tässä graafissa esimerkkinä on kaaret e_2 ja e_3 solmujen n_2 ja n_3 välillä. Esimerkkigraafin kaarilla on suunnat $\rho(e_1) = (n_1, n_2)$, $\rho(e_2) = (n_3, n_2)$ ja $\rho(e_3) = (n_3, n_2)$. Kaikki solmut ja kaaret on nimiöity: solmujen nimiöitä ovat esimerkiksi $\lambda(n_1) = "Henkilö"$ ja $\lambda(n_2) = "Elokuva"$, kaarien taas esimerkiksi $\lambda(e_1) = "näyttelee"$ ja $\lambda(e_3) = "käsikirjoittaa"$. Nimiöitä voidaan tarvittaessa käyttää solmun tai kaaren tyyppinä, joiden avulla voidaan määrätä niille tietokantakaavioon perustuvia rajoitteita [Angles & Gutierrez 2018]. Nimiöiden lisäksi kuvan graafin solmuilla on attribuutteja, esimerkiksi solmulla n_1 attribuutit $\sigma(n_1, "nimi") = "\text{Carrie Fisher}"$ ja $\sigma(n_1, "sukupuoli") = "\text{nainen}"$. Myös kaarilla voi olla attribuutteja, kuten kaarella e_1 attribuutit $\sigma(e_1, "rooli") = "\text{Leia Organa}"$ ja $\sigma(e_1, "ruutuaika") = 13{:}30$.



Kuva 2.1. Esimerkki ominaisuusgraafista, jossa attribuutteihin on säilötty tietoa [mukaillen Angles *et al.* 2017].

Ominaisuusgraafeista on olemassa erilaisia variaatioita. Esimerkiksi joissakin sovelluksissa on hyödyllistä, että nimiöllä tai attribuutilla voi olla useita eri arvoja: tällaiset ominaisuusgraafit ovat *moniarvoisia* (multi-valued). Graafitietokannoissa onkin erilaisia järjestelmäkohtaisia linjauksia: esimerkiksi ominaisuusgraafitietomalliin perustuva graafitietokanta Neo4j sallii yhden nimiön yhdelle kaarelle ja useita nimiöitä yhdelle solmulle. [Angles *et al.* 2017]

Ominaisuusgraafeille ei ole standardoitua kyselykieltä, joskin sellaista (Graph Query Language, GQL) ollaan kehittämässä parhaillaan tämän työn kirjoitushetkellä [JCC Consulting 2022]. Tunnetuin ominaisuusgraafien deklaratiivinen kyselykieli on Neo4j:n kehittämä Cypher [Pokorný 2015], josta on olemassa myös avoimen lähdekoodin versio openCypher [Neo4j 2015]. Muita tunnettuja kyselykieliä ovat deklaratiiviset kielet GSQL [TigerGraph 2023c] ja PGQL [Oracle 2023] sekä Apachen kehittämä funktionaalinen graafikyselykieli Gremlin [The Apache Software Foundation 2022].

Cypherin syntaksi muistuttaa suuresti SQL:ää ja sillä pystyy helposti ilmaisemaan graafikuvioita ja *polkukyselyitä* (path query) [Angles & Gutierrez 2018]. Yksinkertainen Cypher-kysely, jolla haettaisiin kuvan 2.1 esimerkistä henkilöt, jotka näyttelevät ”Star Wars”-nimisessä elokuvassa, muodostettaisiin seuraavasti:

```
MATCH (x1:Henkilö)-[:näyttelee]->(:Elokuva {nimi: 'Star Wars'})
RETURN x1
```

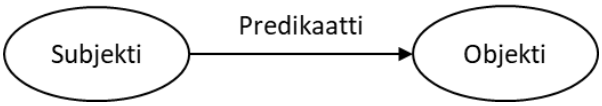
Cypherin syntaksissa on huomattavaa, että MATCH-osa sisältää haettavan *perusgraafikuvion* (basic graph pattern) ja RETURN-osassa määritellään mitä kysely palauttaa. Solmut kirjoitetaan sulkujen "( )" ja kaaret hakasulkujen ”[ ]” sisään. Kaaren suunta on ilmaistu ASCII-merkkien nuolinotaatiolla ”-[kaaren_nimiö]->”. Määritteet solmulle tai kaarelle voi kirjoittaa kaksoispisteen jälkeen. Esimerkiksi $(x1{:}\,Henkilö)$ tarkoittaa, että solmun tulee olla Henkilö-tyyppinen (solmun nimiö on Henkilö): $x1$ puolestaan on muuttujan nimi, jota käytetään tässä palauttamaan haluttu tulos. Ominaisuuksien arvot määritetään aaltosulkeiden ”{ }” väliin kuten esimerkissä on tehty elokuvan osalta $(:Elokuva\ \{nimi{:}\,'Star\ Wars'\})$. [Angles *et al.* 2017]

Cypherissa on monia SQL:n kaltaisia piirteitä: kyselyihin voi esimerkiksi lisätä ehtoja määreellä WHERE, tuloksia voi järjestää määreellä ORDER BY ja alikyselyitä voi käyttää EXISTS-, COUNT- ja CALL-määreiden yhteydessä [Neo4j 2023a; 2023e]. Cypherissa on myös paljon graafitietokannalle ominaisia piirteitä: Cypher on esimerkiksi laajennettavissa Neo4j:n [2023b] laajalla graafialgoritmikirjastolla.

RDF-tietomalli

The Resource Description Framework (RDF) on W3C:n [2014] suunnittelema standardi alun perin webin metadatan esittämiseen. RDF-tietomallissa data esitetään RDF-graafeina, joissa solmut ja kaaret on nimiöity [Angles & Gutierrez 2018]. Ominaisuusgraafien ja RDF-graafien selkein ero on se, että RDF-graafissa ei ole attribuutteja: Anglesin ja muiden [2017] mukaan RDF-graafit pystyvät silti säilömään kompleksista tietoa. RDF-tietomalli myös mahdollistaa kaarien käsittelyn noodeina, tehden informaatio-objektien kohtelusta yhdenmukaista [Angles & Gutierrez 2018].

RDF-graafi muodostuu *RDF-kolmikoista* (RDF triple). RDF-kolmikot puolestaan koostuvat kolmesta elementistä: subjektista, predikaatista ja objektista (kuva 2.2). Jokainen RDF-kolmikko esittää loogisen lausekkeen subjektin ja objektin välisestä suhteesta. RDF-kolmikkojen subjektit ja objektit ovat solmuja, predikaatit puolestaan kaaria. Täten RDF-graafin solmujen joukko on RDF-kolmikkojen subjektien ja objektien joukkojen unioni. [W3C 2014]

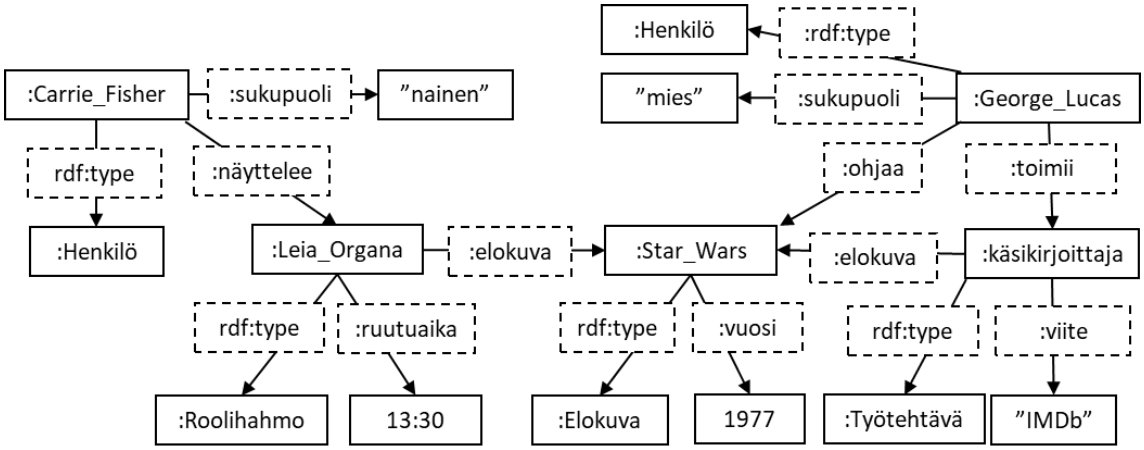


Kuva 2.2. Yksinkertainen RDF-graafi, jossa on yksi RDF-kolmikko [W3C 2014].

RDF-graafissa on kolmenlaisia solmuja: *IRI-nimiä* (Internatiolized Resource Identifier), literaaleja ja tyhjiä solmuja. IRI-nimet ja literaalit kuvaavat *resursseja* eli entiteettejä, jotka voivat olla mitä tahansa: esimerkiksi fyysisiä asioita, dokumentteja, abstrakteja käsitteitä, numeroita tai merkkijonoja. Literaalilla on tietotyyppi (esimerkiksi merkkijono tai numero), joka määrittää sen mahdolliset arvot. IRI-nimet puolestaan ovat RDF-graafeissa käytettyjä merkkijonotunnisteita, jotka noudattavat niille asetettua syntaksia. IRI-nimi voi kuulua *RDF-sanastoon* (RDF vocabulary), jolloin IRIn alkuun tulee usein nimiavaruus-etuliite, joka erotetaan kaksoispisteellä lopusta. RDF-graafissa subjektit voivat olla IRI-nimiä tai tyhjiä solmuja, predikaatit pelkästään IRI-nimiä ja objektit IRI-nimiä, literaaleja tai tyhjiä solmuja. [W3C 2014]

Kuvassa 2.3 on esitetty yksi tapa muuntaa kuvan 2.1 ominaisuusgraafin tiedot RDF-graafiksi: muunnos on tehty käyttäen Anglesin ja muiden [2017] esimerkkejä sekä Stardog-tietokannan dokumentaatiota [Stardog Union 2023]. Solmujen ja kaarien attribuutit on muunnettu uusiksi RDF-kolmikoiksi, koska RDF-graafissa ei ole attribuutteja. Syntyneessä RDF-graafissa solmut ovat joko IRI-nimiä (esimerkiksi $:Star\_Wars$ ja $:George\_Lucas$) tai literaaleja ("$IMDb$" ja 1977). Kaaret puolestaan ovat aina IRI-nimiä (esimerkiksi $:näyttelee$ ja $:vuosi$). IRI-nimien nimiavaruutta ei ole tässä määritetty muille kuin $rdf:type$-predikaateille, joka viittaa W3C:n [2019] RDF-sanastoon.

Tässä muunnoksessa ominaisgraafin solmujen nimi-attribuutin arvo on muunnettu IRI:ksi yksinkertaisuuden vuoksi. Tarkemmin mallinnettuna esimerkiksi solmusta, jonka IRI on $:Carrie\_Fisher$ voisi lähteä $:nimi$-predikaatti (IRI) "$Carrie\ Fisher$"-objektiin (literaali). Kuvasta 2.3 voidaan nähdä, että rakenne on huomattavasti monimutkaisemmin näköinen kuin ominaisuusgraafilla kuvassa 2.1.



Kuva 2.3. Esimerkki RDF-graafista.

RDF:n standardikyselykieli on deklaratiivinen kyselykieli SPARQL [W3C 2013]. SPARQL-kyselyt noudattavat perinteistä SELECT-FROM-WHERE-rakennetta, jossa FROM-osa kertoo datan lähteet, WHERE-osa *graafikuvion* (graph pattern) ja SELECT-osa mitä kysely palauttaa [Angles & Gutierrez 2018]. Yksinkertaisin graafikuvio (*kolmikkokuvio*, triple pattern) on RDF-kolmikko, jossa subjektina, predikaattina tai objektina voi olla muuttuja [Angles *et al.* 2017]. Kolmikkokuvioita pystyy ketjuttamaan perusgraafikuvioiksi [Angles *et al.* 2017] ja kolmikkokuvioita pystyy myös yhdistelemään ja rajoittamaan operaattoreilla kuten AND, UNION ja FILTER [Angles & Gutierrez 2018].

Alle on muodostettu yksinkertainen SPARQL-kysely [mukaillen Angles & Gutierrez 2008 ja W3C 2013], jolla haettaisiin kuvan 2.3 RDF-graafista $:Henkil\ddot{o}$-tyyppiset solmut:

```
PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

SELECT ?X
FROM http://esimerkki.org/data.rdf
WHERE { ?X rdf:type :Henkilö }
```

Kyselyssä on ensin määritetty PREFIX-määreellä rdf-etuliitteen nimiavaruus [W3C 2019]. Tämän jälkeen SELECT-osassa on määritetty palautettavat muuttujat: SPARQL-kyselyssä muuttujien eteen laitetaan tyypillisesti kysymysmerkki [Angles *et al.* 2017]. FROM-osaan on määritetty datan lähde ja lopulta WHERE-osassa on täsmäytettävä kolmikkokuvio (tässä RDF-kolmikko, jonka subjekti on $?X$, predikaatti $:rdf\_type$ ja objekti $:Henkil\ddot{o}$). Kysely palauttaisi solmut $:Carrie\_Fisher$ ja $:George\_Lucas$.

2.2 Graafidatan hallintajärjestelmät

Graafidatan hallintajärjestelmät (graph data management systems) voidaan jakaa kahteen pääkategoriaan: graafitietokantoihin sekä *graafien prosessointiviitekehyksiin* (graph processing frameworks). Vaikka molemmat vastaavat samankaltaisiin ongelmiin, niillä on erilainen lähestymistapa graafidatan säilytykseen ja kyselyihin. Graafitietokannat ovat tietokantajärjestelmiä, jotka pyrkivät graafidatan pidempiaikaiseen säilytykseen ja tarjoavat transaktionaalisen pääsyn dataan. Graafien prosessointiviitekehykset puolestaan prosessoivat graafeja eräajojen kautta yleensä hajautetussa ympäristössä. Tässä työssä keskitytään graafitietokantoihin ja rajataan tarkastelusta graafien prosessointiviitekehykset pois. [Angles & Gutierrez 2018]

Lisäksi eräänlaisia graafitietokantoja ovat RDF-standardin mukaisen datan käsittelyyn tarkoitetut *RDF-tietokannat* (RDF database, RDF triple store) [Angles & Gutierrez 2018]. Niiden voidaan katsoa muodostavan graafitietokantojen joukossa oman ryhmänsä: tässä työssä RDF-tietokannat lasketaan mukaan graafitietokantoihin, vaikka ne käsitellään tässä kohdassa erikseen.

Graafitietokannat

Graafitietokannat ovat tietokantoja, jotka on suunniteltu varta vasten graafidatan hallintaan noudattaen tietokantajärjestelmien yleisiä periaatteita. Näitä periaatteita ovat *tiedon varastoimisen pysyvyys* (persistent data storage), *fyysisen ja loogisen datan itsenäisyys* (physical/logical data independence), *datan ehjyys* (data integrity) sekä *johdonmukaisuus* (consistency). [Angles & Gutierrez 2018]

Markkinoilla on saatavilla useita eri graafitietokantoja, joiden kehitysvaiheet vaihtelevat suuresti. Nämä tietokannat tarjoavat suurimman osan tärkeimmistä tietokannanhallintajärjestelmien komponenteista: *tietokantamoottorin* (database engine), kyselykielen, indeksoinnin, kyselyjen optimoinnin, transaktiot, rinnakkaisuuden hallinnan sekä ulkoisen rajapinnan järjestelmän hallintaan (käyttöliittymän tai ohjelmointirajapinnan). [Angles & Gutierrez 2018]

Graafitietokannat luokitellaan NoSQL-tietokannoiksi, jotka ovat ryhmä erilaisia *ei-relaatiotietokantoja* (non-relational database) [Davoudian *et al.* 2017]. NoSQL-tietokannat on suunniteltu vastaamaan relationaalisten tietokantojen jäykkyyteen: NoSQL-tietokannat painottavat joustavuutta, horisontaalista skaalautuvuutta ja tiedon nopeaa saatavuutta [Roy-Hubara & Sturm 2020]. Graafitietokantojen lisäksi muita tunnettuja NoSQL-tietokantoja ovat *avain-arvoparivarastot* (key-value stores), *dokumenttivarastot* (document stores) ja *sarakeperhevarastot* (wide-column stores) [Davoudian *et al.* 2017].

Graafitietokannat voidaan jakaa sisäisen toteutuksen mukaan *natiiveihin* (native) ja *ei-natiiveihin* (non-native) graafitietokantoihin: taulukossa 2.1 on esitetty Anglesin ja Gutierrezin [2018] kokoama listaus näistä. Natiivit graafitietokannat käyttävät varta

vasten kehitettyjä tietorakenteita ja indeksejä graafidatan tallentamiseen ja kyselyihin. Ei-natiivit graafitietokannat puolestaan käyttävät muita tietokantajärjestelmiä (kuten toista NoSQL-tietokantaa tai relaatiotietokantaa) graafidatan säilömiseen ja tarjoavat graafikyselyjen tekemistä varten rajapinnan taustalla olevaa tietokantaa vasten. [Angles & Gutierrez 2018]

Natiiveja graafitietokantoja	Ei-natiiveja graafitietokantoja
AllegroGraph	ArangoDB
Bitsy	FlockDB
Cailey	Horton
Dgraph*	MariaDB
GraphBase	OQGraph
Graphd	OrientDB
HyperGraphDB	Titan
IBM System G	VelocityGraph
imGraph	
InfinityGraph	
InfoGrid	
Neo4j	
Sparksee	
TigerGraph*	
Trinity	
TurboGraph	

Taulukko 2.1. Listaus graafitietokannoista [mukaillen Angles & Gutierrez 2018]. (*) Listaan on täydennetty TigerGraph [2023d] ja Dgraph [Dgraph Labs 2022].

Tunnetuin graafitietokanta on Neo4j [Solid IT 2022], joka on natiivi ominaisuusgraafitietokanta ja sen oma kyselykieli on Cypher [Neo4j 2015]. Neo4j:n [2019] ensimmäinen julkisesti saatavilla oleva versio julkaistiin 2010. Neo4j:lle on sittemmin kehitetty runsaasti työkaluja ja kirjastoja: Neo4j [2023d] pystyy esimerkiksi käsittelemään RDF-muotoista dataa laajennuksen avulla.

Muista graafitietokannoista voidaan mainita merkittävänä AllegroGraph, joka on yksi ensimmäisistä nykyisen graafitietokantojen sukupolven edustajista. Vaikka alun perin AllegroGraph kehitettiin graafitietokannaksi, nykyinen toteutus vastaa RDF-tietokantaa. [Angles & Gutierrez 2018]

Muita natiiveja ominaisuusgraafitietokantoja ovat esimerkiksi TigerGraph [2023d] ja Dgraph [Dgraph Labs 2022]. Paremmin tunnetuista ei-natiiveista graafitietokannoista voidaan mainita esimerkiksi dokumenttivarastot ArandoDB ja OrientDB [Angles & Gutierrez 2018] sekä relaatiotietokanta MariaDB [MariaDB Foundation 2023].

RDF-tietokannat

RDF-tietokannat perustuvat RDF-tietomalliin, joka on kuvattu tämän työn kohdassa 2.1. Muiden graafitietokantojen tavoin myös RDF-tietokannat voidaan jakaa natiiveihin ja ei-

natiiveihin RDF-tietokantoihin. Natiiveja RDF-tietokantoja ovat esimerkiksi Jena, RDF-3X, BlazeGraph ja Stardog: ei-natiiveja puolestaan OpenLink Virtuoso ja DB2RDF, jotka on tehty pohjautuen relaatiotietokantaan. Natiivit RDF-tietokannat voidaan jakaa vielä tarkemmin sisäisen toteutuksen perusteella neljään kategoriaan: *kolmikkotauluihin* (triples table), *ominaisuustauluihin* (property table), *vertikaalista ositusta käyttäviin sarakevarastoihin* (column store with vertical partitioning) ja *RDF-perusteisiin graafivarastoihin* (RDF based graph store). [Angles & Gutierrez 2018]

Kolmikkotauluissa RDF-data säilötään 3-sarakkeisiin tauluun, jossa jokainen rivi vastaa yhtä RDF-kolmikkoa. Näin ollen SPARQL-kyselyjen suoritus vaatii taulun liittämistä itseensä useita kertoja: taulun indeksointi onkin olennaisessa asemassa tässä lähestymistavassa. Kolmikkotauluja käyttäviä RDF-tietokantoja ovat esimerkiksi RDF-3X, Sesame ja 3store. [Angles & Gutierrez 2018]

Toinen lähestymistapa säilöä RDF-dataa on käyttää ominaisuustauluja. Tällöin taulun ensimmäinen sarake esittää subjektin ja loput sarakkeet predikaatteja, joita voi olla hyvin runsaasti. Ominaisuustaulu sisältää näin ollen luultavasti paljon tyhjäarvoja ja on hyvin harva. Jena ja Oracle ovat ominaisuustauluja käyttäviä RDF-tietokantoja. [Angles & Gutierrez 2018]

Kolmas tapa RDF-datan säilömiseen on käyttää useita kaksisarakkeisia tauluja, joista jokainen esittää yhtä ainutlaatuista predikaattia. Taulujen ensimmäiseen sarakkeeseen tallennetaan subjekti ja toiseen objekti. Tätä menetelmää kutsutaan vertikaalista ositusta käyttäväksi sarakevarastoksi: C-Store on tällainen RDF-tietokanta. [Angles & Gutierrez 2018]

Neljäs tapa on säilöä RDF-data graafina: tällöin RDF-kolmikot mallinnetaan graafin solmuina ja kaarina. SPARQL-kyselyt tulee myös tällöin muuntaa graafikyselyiksi. Tätä toteutusta käyttäviä RDF-tietokantoja ovat Ontotext GraphDB, Stardog ja BlazeGraph [Angles & Gutierrez 2018].

Joissakin RDF-tietokannoissa käytetään RDF-kolmikkojen sijaan RDF-nelikkoja, joissa subjektin, predikaatin ja objektin lisäksi neljäntenä jäsenenä on graafin nimi [Pokorný 2015]. Esimerkiksi Amazon Neptune on tällainen tietokanta: se käyttää RDF-nelikkoja, joita kutsutaan Neptune-nelikoiksi [Amazon Web Services 2023b]. AllegroGraph puolestaan käyttää RDF-viisikkoja, joissa viidenteen jäseneen voidaan tallentaa lisäksi vielä metadataa [Pokorný 2015].

2.3 Graafitietokantojen sovelluksia

Tässä kohdassa käydään läpi tutkijoiden ja graafitietokantojen valmistajien esittämiä listauksia graafitietokantojen sovellusaloista. Listausten esittämisen jälkeen käydään läpi listauksien perusteella merkittävimmät sovellusalat ja kuvaillaan niitä lyhyesti.

Graafitietokantojen sovellusaloja on tutkittu aiempaa tutkimusta yhteen vetäen kirjallisuushakujen perusteella [Liu *et al.* 2021; Rizaldy *et al.* 2021]. Sahu ja muut [2020]

tutkivat graafitietokantojen käyttökohteita graafitietokantojen valmistajien materiaalien pohjalta. Lisäksi Angles ja Gutierrez [2008] esittävät kirjallisuuskatsauksessaan graafitietomallin sovelluksia ennen vuotta 2006.

Angles ja Gutierrez [2008] jakavat graafitietomallin sovellukset klassisiin sovelluksiin ja kompleksisiin verkkoihin. Klassiset sovellukset liittyivät 80- ja 90-luvuilla esimerkiksi hypertekstin käsittelyyn [Tompa 1989; Watters & Shepherd 1990], tietämyksen esittämiseen [Kunii 1987], geografisen tiedon esittämiseen [Maingeunaud 1995] ja oliotietokantoihin [Gyssens *et al.* 1990]. 2000-luvulle tultaessa kiinnostus erilaisia kompleksisia verkkoja kohtaan oli kasvanut suureksi: näitä pystyttiin käsittelemään graafitietomallin avulla [Angles & Gutierrez 2008]. Yksi tapa jakaa kompleksisia verkkoja on jaotella ne sosiaalisiin verkostoihin, informaatioverkkoihin, teknologisiin verkkoihin ja biologisiin verkkoihin [Newman 2003, s. 174].

Liu ja muut [2021] tekivät osana tutkimustaan kirjallisuushaun, jossa he löysivät 145 graafitietokantojen sovellusta. Heidän luokituksessaan *tietämysgraafit* (knowledge graph) olivat yleisin sovelluskohde ja se sisälsi eri alojen, kuten lääketieteen, biologian ja rahoituksen, sovelluksia. Tämän jälkeen suurimpina aloina heidän luokituksessaan olivat edustettuina biolääketiede ja IoT-laitteiden hallinta. [Liu *et al.* 2021]

Rizaldy ja muut [2021] puolestaan tekivät systemaattisen kirjallisuuskatsauksen, jonka yhtenä tutkimuskysymyksenä oli mitkä toimialat käyttävät graafitietokantoja. Heidän aineistonaan oli 60 artikkelia. He jakoivat graafitietokantojen sovellukset IT-alan ja IT-alan ulkopuolisiin sovelluksiin. Yleisimmät IT-alan ulkopuoliset sovellusalat olivat yleiset alasta riippumattomat sovellukset, kuljetukset, sosiaaliset verkostot ja akateeminen tutkimus. [Rizaldy *et al.* 2021]

Sahu ja muut [2020] tutkivat graafitietokantojen valmistajien tekemiä *white paper* -dokumentteja, joissa kuvaillaan usein markkinoinnillisessa mielessä graafitietokantojen käyttökohteita. He luokittelivat 89 dokumentille sovelluskohteen sekä toimialan. Yleisin sovelluskohde oli datan integrointi: tällä Sahu ja muut tarkoittavat sovelluksia, joissa päätehtävänä on rakentaa keskitetty, usein heterogeeninen graafi useasta lähteestä. Toiseksi yleisin sovelluskohde olivat suositukset ja personalisointi, ja kolmanneksi yleisin petosten sekä uhkien tunnistaminen. [Sahu *et al.* 2020]

Edellä esitettyjen perusteellisempien kirjallisuushakujen ja -katsausten lisäksi tutkimusartikkeleissa on monesti listattu vähemmän systemaattisesti erilaisia graafitietokantojen sovellusaloja. Tällaisia listauksia löytyy esimerkiksi Francisin ja muiden [2018], Patilin ja muiden [2014], Tianin [2023] ja Woodin [2012] artikkeleista.

Myös graafitietokantojen valmistajat [ArangoDB 2023; Neo4j 2023c; OpenLink Software 2019; Oracle 2021; TigerGraph 2023b] ovat esittäneet listauksia graafitietokantojen sovelluskohteista. Näitä listauksia on käytetty ennen kaikkea tietokantojen markkinointimateriaalina.

Yhteenveto näistä 13 eri listauksesta on esitetty taulukossa 2.2. Taulukkoon on merkattu kussakin lähteessä mainitut sovellusalat ja laskettu mainintojen määrä yhteen.

	Angles & Gutierrez 2008	ArangoDB 2023	Francis *et al.* 2018	Liu *et al.* 2021	Neo4j 2023c	OpenLink Software 2019	Oracle 2021	Patil *et al.* 2014	Rizaldy *et al.* 2021	Sahu *et al.* 2020	Tian 2023	TigerGraph 2023b	Wood 2012	Mainintoja yhteensä
Akateeminen tutkimus	x								x					2
Biotieteet	x		x	x	x	x		x	x	x		x	x	10
Datanhallinta			x		x		x	x	x	x		x	x	8
Datatiede					x				x					2
Elintarvikkeet										x				1
Energia	x			x							x	x		4
Esineiden internet				x										1
Geografinen tieto	x								x			x		3
Hakukoneet								x			x			2
Henkilöstöhallinto						x								1
Journalismi			x							x				2
Julkishallinto				x	x		x			x	x		x	6
Koulutus										x				1
Kuljetukset ja liikenne	x	x			x				x	x	x	x	x	8
Kulttuuri										x				1
Kyberturvallisuus		x	x		x		x	x		x		x		7
Markkinointi		x					x			x	x	x		5
Petosten tutkinta		x	x	x	x	x	x			x	x	x		9
Rahoitus				x	x		x			x	x	x		6
Riskinhallinta					x					x		x		3
Sosiaaliset verkostot	x		x	x	x	x	x	x	x	x	x		x	11
Suosittelujärjestelmät		x	x	x	x	x	x			x		x		8
Tekoäly ja koneoppiminen					x		x			x	x	x		5
Telekommunikaatio					x					x				2
Terveydenhuolto											x			1
Tietojärjestelmät	x		x						x	x			x	5
Tietoverkot	x	x	x		x	x		x	x	x		x		9
Tietämysgraafit				x	x	x				x				4
Tuotannonohjaus							x			x	x			3
Vähittäiskauppa					x					x	x			3
Web ja hyperteksti	x								x				x	3

Taulukko 2.2. Graafitietokantojen sovellusaloja tutkijoiden ja valmistajien listaamina.

Erilaisia graafitietokantojen sovellusaloja tunnistettiin 31. Taulukosta 2.2 on nähtävissä, että eri sovellusalojen mainintamäärien välillä on selkeitä eroja. Lähteistä tunnistettiin muutamia sovellusaloja (kuten sosiaaliset verkostot, biotieteet ja petosten tutkinta), jotka mainittiin useimmissa lähteissä, kun taas jotkin alat saivat vain vähän mainintoja. Seuraavaksi kuvaillaan sovellusaloja ja niillä esiintyviä erilaisia sovelluksia lyhyesti.

Graafitietokantojen sovellusalojen kuvauksia

Sosiaaliset verkostot (social network) mainittiin useimmin (11 lähteessä) graafitietokantojen sovellusaloista. Sosiaalinen verkosto on verkko, joka esittää yksilöitä ja heidän välisiä suhteitaan [Merriam-Webster 2023b]. Sosiaalinen verkosto voidaan mallintaa graafina, jolloin solmut tyypillisesti esittävät henkilöitä ja kaaret näiden välisiä yhteyksiä kuten ystävyyttä tai kirjoituskumppanuutta [Larriba-Pey *et al.* 2014]. Mainittuja sosiaalisten verkostojen sovelluksia olivat esimerkiksi erilaiset sosiaalisten verkostojen analyysit [Larriba-Pey *et al.* 2014], sosiaalisen median palvelut kuten Facebook ja LinkedIn [Patil *et al.* 2014], tutkijaverkostot [Angles & Gutierrez 2008] sekä kontaktien jäljitys epidemioissa [Liu *et al.* 2021].

Biotieteet (life sciences) mainittiin 10 lähteessä. Bioteet ovat tieteenaloja, joissa tutkitaan eläviä organismeja: niitä ovat esimerkiksi biologia ja lääketiede [Merriam-Webster 2023a]. Graafit ovat käytännöllisiä biotieteissä, koska monet biologiset systeemit voidaan biologisina verkkoina: tärkeitä sovelluskohteita ovat esimerkiksi aineenvaihduntatiet, geneettisen säätelyn verkot, ravintoverkot ja hermoverkot [Newman 2003, ss. 179–180]. Tutkituissa lähteissä erityisen usein mainittiin bioinformatiikka ja biolääketiede. Biolääketieteen ala onkin ollut yksi graafitietokantojen aikainen omaksuja datan linkittyneisyyden vuoksi: graafitietokantojen sovelluksia löytyy alalta esimerkiksi systeemibiologian, bio- ja lääkekemian sekä biologisten tietämysgraafien alueilta [Timón-Reina *et al.* 2021].

Petosten tutkinta (fraud detection) mainittiin 9 lähteessä. Graafitietokannat voivat auttaa petosten tutkinnassa analysoimalla vaikeasti havaittavia rakenteita datassa [ArangoDB 2023]. Tällaiset rakenteita ovat esimerkiksi rahanpesuverkostot [Oracle 2021], petosringit [Neo4j 2023c], salakuljetusverkostot [Oracle 2021] sekä veroparatiisien käyttö veronkierrossa [Cabra 2016].

Tietoverkot mainittiin 9 lähteessä. Tietoverkot ovat teknologisia verkkoja: hyvin tunnettu esimerkki tietoverkoista on internet [Newman 2003, ss. 178–179]. Graafitietokanta soveltuu tietoverkkojen mallintamiseen, koska verkkolaitteet ja niiden väliset yhteydet muodostavat luonnostaan graafin [ArangoDB 2023]. Graafitietokantojen avulla voidaan esimerkiksi analysoida tietoverkkoja [Patil *et al.* 2014], hallita tietoverkon asetuksia [OpenLink Software 2019] ja visualisoida tietoverkkoja [ArangoDB 2023].

Datanhallinta (data management) mainitaan 8 lähteessä. Graafitietokantoja voidaan käyttää monenlaisen datan hallintaan, esimerkiksi tuotetietojen tai taloudellisten tietojen tallentamiseen [Rizaldy *et al.* 2021]. Yksi graafitietokantojen tärkeä sovelluskohde datanhallinnassa on yritysten *ydintietojen hallinta* (master data management) [Neo4j 2023c; Oracle 2021]. Ydintietojen hallinnalla tarkoitetaan prosessia, jolla liiketoiminta-johto ja tietohallinto pyrkivät yhdessä varmistamaan tiedon yhdenmukaisuuden, tarkkuuden, johdonmukaisuuden, omistajuuden sekä vastuut [Gartner 2023].

Kuljetukset ja liikenne mainittiin 8 lähteessä. Kuljetus- ja liikennealalla esimerkiksi lentokoneiden reitit, rautatiet ja jakeluverkot ovat teknologisia verkostoja [Newman 2003, s. 178]. Erityisesti *toimitusketjun hallinta* (supply chain management) mainittiin usein [ArangoDB 2023; Neo4j 2023c]. ArangoDB:n [2023] mukaan graafitietokannan avulla esimerkiksi voidaan esittää varastotilanteeseen, toimittajiin ja toimituksiin liittyvää tietoa graafina ja selvittää täten häiriöiden syitä.

Suosittelujärjestelmät (recommender systems) mainittiin 8 lähteessä. Suosittelujärjelmät ovat ohjelmistoja, jotka pyrkivät tarjoamaan käyttäjälle hänen mieltymyksiinsä räätälöityjä suosituksia [Giabelli *et al.* 2021]. Graafitietokannat soveltuvat hyvin suosittelujärjestelmiin, koska graafitietokantojen avulla asiakkaiden ja tuotteiden välisiä suhteita voi analysoida nopeasti algoritmeilla suositusten luomiseksi [Oracle 2021].

Kyberturvallisuuden alalla (7 mainintaa) puolestaan graafitietokantoja käytetään monessa sovelluksessa *pääsynhallinnassa* (access management). Pääsynhallinta voi olla monimutkaista, jos organisaatiossa on useita rooleja, ryhmiä ja tuotteita: tällöin graafitietokanta voi auttaa havainnollistamaan kenellä tulisi olla pääsy mihinkin tietoon [Neo4j 2023c]. Lisäksi graafitietokantoja voidaan käyttää kyberhyökkäysten tunnistamisessa [TigerGraph 2023b] ja datan alkuperän selvittämisessä [Oracle 2021].

Julkishallinto (government) mainittiin 6 lähteessä. Viranomaiset voivat käyttää graafitietokantoja esimerkiksi tiedustelupalveluissa [Neo4j 2023c] tai rikosten tutkinnassa [Oracle 2021; Wood 2012] kun selvitetään eri tekijöiden välisiä yhteyksiä. Muita mahdollisia sovelluskohteita ovat veropetosten tutkiminen [Neo4j 2023c; Oracle 2021] ja kontaktien jäljitys [Oracle 2021], mitkä liittyvät aiemmin esiteltyihin sovellusaloihin.

Rahoitus mainittiin 6 lähteessä. Monet lähteissä mainitut graafitietokantojen sovellukset rahoitusalalla liittyvät jo aiemmin käsiteltyyn petosten tutkintaan [mm. Neo4j 2023c; Oracle 2021; TigerGraph 2023b]. Lisäksi rahoitusalalla voidaan käyttää graafitietokantoja esimerkiksi luottoriskien hallinnassa ja maksuliikenteen analyysissa [TigerGraph 2023a].

Markkinointi mainittiin 5 lähteessä. Erityisen usein [ArangoDB 2023; Neo4j 2023c; Oracle 2021; TigerGraph 2023b] mainittiin asiakkaan *360 asteen arviointi* (customer 360° analysis), jolla pyritään saamaan täydellinen käsitys asiakkaasta ja asiakkaan tarpeista vetämällä yhteen eri lähteistä saatavaa dataa [Horwitz 2015]. Graafitietokanta

sopii 360 asteen arviointeihin hyvin, koska graafissa yhteen entiteettiin (asiakkaaseen) on helppo yhdistää eri lähteistä tulevaa tietoa (kuten ydintietoja, transaktioiden tietoja, big dataa ja tekoälyn ennusteita) [Oracle 2021].

Tekoäly ja koneoppiminen mainittiin sovellusalana 5 lähteessä. *Tekoälyllä* (artificial intelligence) tarkoitetaan tietojärjestelmiä, jotka kykenevät suorittamaan älykkäinä pidettyjä toimintoja [McCarthy 1999]. Tarkemmin määriteltynä tekoäly on tietojärjestelmän kyky tulkita ulkoisia tietoja oikein, oppia näistä tiedoista ja käyttää opittua tietoa suoriutumaan tehtävistä paremmin [Kaplan & Haenlain 2019]. *Koneoppiminen* (machine learning) puolestaan on ala, jossa tutkitaan järjestelmiä, jotka parantavat suorituskykyään tietyssä tehtävässä kokemuksen tai datan kertyessä [Mitchell 1997, s. 2]. Koneoppiminen on olennainen osa tekoälyä, mutta tekoäly on käsitteenä laajempi ja kattaa myös esimerkiksi järjestelmän kyvyn havainnoida ja manipuloida ympäristöään [Kaplan & Haenlain 2019]. Graafitietokantoja voidaan hyödyntää koneoppimisen alalla esimerkiksi neuroverkoissa, jotka voidaan esittää graafimuodossa [Amazon Web Services 2023a]. Graafimuotoisen neuroverkon pystyy tallentamaan graafitietokantaan sellaisenaan ja siihen pystyy potentiaalisesti tallentamaan runsaasti tietoa graafitietomallin joustavuuden ansiosta [Oracle 2021].

Tietojärjestelmät mainittiin sovellusalana viidessä lähteessä: niissä graafitietokantoja sovellettiin esimerkiksi tietojärjestelmien analysoinnissa [Francis *et al.* 2018] ja integroinnissa [Angles & Gutierrez 2008]. Neljässä lähteessä mainittiin sovellusalana **energia**, jossa graafitietokanta esimerkiksi pystyy esittämään intuitiivisesti sähköverkon rakenteen ja resurssien käytön [Liu *et al.* 2021]. **Tietämysgraafit** mainittiin myös neljässä lähteessä. Tietämysgraafit ovat tietoa sisältäviä graafeja, joilla ilmaistaan ja kartutetaan oikean maailman tietämystä [Hogan *et al.* 2021].

Geografinen tieto mainittiin sovellusalana kolmessa lähteessä. Geografista tietoa on maantieteeseen perustuvat data, kuten tie- ja rautatieverkostot sekä jokijärjestelmät: ne voidaan nähdä teknologisena verkkona [Newman 2003, s. 178]. *Paikkatietojärjestelmät* (geographic information system, GIS) ovat yksi tärkeä graafitietokantojen sovelluskohde tällä alalla [Angles & Gutierrez 2008]. **Web ja hyperteksti** (3 mainintaa) ovat yksi klassinen graafitietokantojen sovellusala [Angles & Gutierrez 2008]. World Wide Web ja hyperteksti ovat informaatioverkkoja: web rakentuu informaatiota sisältävistä websivuista, jotka linkittyvät toisiinsa hyperlinkkien välityksellä [Newman 2003, ss. 176–177]. Muita kolme mainintaa saaneita sovellusaloja olivat **riskinhallinta**, **tuotannonohjaus** ja **vähittäiskauppa**.

Kahdessa lähteessä mainittuja sovellusaloja olivat **akateeminen tutkimus**, **datatiede**, **hakukoneet** (erityisesti hakualgoritmit), **journalismi** ja **telekommunikaatio**. Vain yhden maininnan saaneita sovellusaloja olivat **elintarvikkeet, esineiden internet**, **henkilöstöhallinto**, **koulutus**, **kulttuuri** ja **terveydenhuolto**.

2.4 Graafitietokantojen ominaisuuksia

Tässä kohdassa tarkastellaan erilaisia graafitietokantojen ominaisuuksia ja miten niitä on aiemmassa tutkimuksessa käsitelty. Tarkasteltavia ominaisuuksia ovat graafimaisen datan mallintaminen, tietokantakaavion joustavuus, graafikyselykielten käyttäminen, graafitietokantojen suorituskyky, visualisointi, turvallisuus sekä rinnakkaisuus.

Graafimaisen datan mallintaminen

Anglesin ja Gutierrezin [2008] mukaan graafitietokantoja hyödynnetään etenkin sovelluksissa, joissa komponenttien keskinäiset yhteydet ovat avainasemassa. Datan mallintaminen graafeina tuo monia hyötyjä näissä tapauksissa: tällöin graafit mahdollistavat datan mallintamisen luonnollisemmassa muodossa. Graafirakenteet tulevat käyttäjälle näkyviksi ja ne mahdollistavat luonnollisen tavan käsitellä sovellusdataa, esimerkiksi hypertekstin tai geografisen datan kohdalla. [Angles & Gutierrez 2008]

Patilin ja muiden [2014] mukaan graafitietokannat tarjoavat intuitiivisen ja ilmaisuvoimaisen tavan datan esittämiseen: graafitietokannoissa data esitetään objekteina, jotka on yhdistetty toisiinsa suhteilla ja joista jokaisella on oma joukkonsa kuvailevia ominaisuuksia. Myös Liun ja muiden [2021] mukaan todellisen maailman kompleksisten suhteiden esittäminen on graafitietomallissa suorempaa, mikä auttaa käyttäjiä ymmärtämään dataa paremmin.

Sahu ja muut [2020] tutkivat muun muassa kyselyn ja haastattelujen avulla graafitietokantojen käyttöä. Heidän tutkimuksessaan tuli esiin, että usein graafitietokantoja käytetään mallintamaan dataa, jolla ei ole luonnollista jakoa solmuihin ja kaariin. Erityisesti he nostivat esiin, että graafitietokantojen yleinen sovelluskohde on perinteinen yritysdata (joka koostuu tuotteista, tilauksista ja transaktioista), jolle relaatiotietokannat sopisivat täydellisesti. Tämä on ristiriidassa olemassa olevan tutkimuksen kanssa: tehdyt tutkimukset eivät anna tukea sille, että graafitietokantoja kannattaisi soveltaa perinteiselle relaatiomallin mukaiselle datalle. Sahu ja muut esittävät löydökselleen selityksenä, että yritykset saattavat pitää hyödyllisenä myös perinteisen yritysdatan suhteiden analysointia graafitietokantojen avulla. He kuitenkin pitivät ongelmana, että graafitietokannoille ei ole kehitetty suorituskykytestejä perinteisellä datalla (RDF-tietokantoja lukuun ottamatta). [Sahu *et al.* 2020]

Tietokantakaavion joustavuus

Tietokantakaavio (database schema) on yksityiskohtainen kuvaus tietokannan rakenteesta. Tietokantakaavio sisältää tietokannalle sallitut rakenteet ja rajoitteet: tietokannan tulee kaikkina hetkinä toteuttaa tietokantakaavio ollakseen validissa tilassa. Tietokantakaavion laatiminen onkin hyvin tärkeä tehtävä. Perinteisesti tietokantakaavio on määritetty suunnitteluvaiheessa ja sen ei odoteta muuttuvan säännöllisesti: muutoksia

tietokantakaavioon joudutaan kuitenkin usein tekemään tietokantasovelluksen vaatimusten muuttuessa. [Elmasri & Navathe 2004, ss. 27–29]

Relaatiotietomalli soveltuu hyvin sellaisen datan mallintamiseen, jonka rakenne tunnetaan melko hyvin ennalta (kuten lentojen varaukset, kirjanpitodata ja varastojen saldot) ja on täten esimerkki perinteisestä melko muuttumattomasta tietokantakaaviosta [Angles & Gutierrez 2018]. Graafitietokantojen kaaviot ovat puolestaan yleensä joustavia: näin tietokantakaaviota on helppo laajentaa ajan kuluessa eikä kaikkea tarvitse vielä tietää alussa [Fernandes & Bernanrdino 2018; Liu *et al.* 2021]. Graafitietokannat soveltuvat täten hyvin strukturoimattoman datan mallintamiseen [Angles & Gutierrez 2018].

Patilin ja muiden [2014] mukaan tietokantaa suunnitellessa ei välttämättä tiedä vielä minkä muotoista ja kuinka monimutkaista todellinen data on: yksityiskohtainen tietokantakaavion mallintaminen suunnitteluvaiheessa voi olla erittäin vaikea tehtävä. Graafitietokannat mahdollistavat kaavion kehittämisen asteittain kasvavan ymmärryksen kanssa. Graafit ovat luonnostaan additiivisia, mikä tarkoittaa, että niihin voi lisätä uuden tyyppisiä suhteita (kaaria), uusia solmuja ja uusia aligraafeja häiritsemättä jo olemassa olevaa toiminnallisuutta. Sallimalla muutokset ajan kanssa graafitietokannat lisäävät kehittäjien tuottavuutta ja vähentävät projektien riskiä. [Patil *et al.* 2014]

Graafitietokantojen kaavion joustavuus tukee myös ketterän ohjelmistotuotannon periaatteita [Fernandes & Bernanrdino 2018; Patil *et al.* 2014]. Ketterässä ohjelmistotuotannossa sovelluksia kehitetään inkrementaalisesti ja iteratiivisesti: graafitietokanta sopii tämän kanssa hyvin yhteen, koska tietokannan rakennetta voidaan muuttaa sovelluksen ja liiketoimintavaatimusten mukaan [Patil *et al.* 2014]. Graafitietokannat tukevat myös *testivetoista kehittämistä* (test-driven development) [Fernandes & Bernardino 2018]: Patilin ja muiden [2014] mukaan graafitietokantojen ohjelmointirajapinnat ja kyselykielet ovat luonteeltaan sopivia testattaviksi.

Graafikyselykielten käyttäminen

Graafikyselykielet ovat olennainen osa graafitietomallia ja graafitietokantoja: graafikyselykielissä hyödynnetään graafioperaatioita ja kyselyt voidaan kohdistaa suoraan graafirakenteeseen. Eksplisiittiset graafit ja graafioperaatiot mahdollistavat käyttäjien esittää kyselyt korkealla abstraktiotasolla. [Angles & Gutierrez 2008]

Graafitietokannoissa hyödynnetään myös usein erityisiä graafirakenteita ja tehokkaita graafialgoritmeja [Angles & Gutierrez 2018]. Graafikyselykielten ja graafianalytiikan välillä ei ole selvää rajaa: graafikyselykielet sisältävät paljon päällekkäisyyksiä graafianalytiikan kanssa [Angles *et al.* 2017].

Monia graafikyselykieliä on kuvailtu ymmärrettäviksi ja ilmaisuvoimaisiksi graafien käsittelyssä: Fernandes ja Bernardino [2018] kuvailevat näin Neo4j:n Cypheriä ja ArangoDB:n AQL-kyselykieltä. Holzschuher ja Peinl [2016] puolestaan vertailivat eri

graafikyselykieliä ja heidän mukaansa Cypherilla on hyvä luettavuus ja Neo4j pystyy suorittamaan sitä kohtuullisen tehokkaasti. Myös Vicknair ja muut [2010] kuvailevat graafikulkujen ilmaisemisen olevan Cypherilla melko yksinkertaista. Samaan tapaan Kallava [2018] arvioi Cypherin syntaksia tiiviiksi ja intuitiiviseksi.

Toisaalta graafikyselykielten käyttöön liitetään myös ongelmia. Sahun ja muiden [2020] tutkimuksen yhtenä löydöksenä oli graafikyselykielten käytön ongelmat esimerkiksi ohjelmointirajapintojen, alikyselyjen ja useista graafeista tietoa hakevien kyselyjen suhteen. RDF-tietokantojen ongelmana oli joissain tapauksissa, että ne eivät toteuta täysin SPARQL-standardia. Osittaisena ratkaisuna graafikyselykielten käytön ongelmiin Sahu ja muut esittävät graafikyselykielten standardoinnin. [Sahu *et al.* 2020]

Bhowmick ja muut [2017] pitivät graafikyselykieliä haastavina asiaan perehtymättömille käyttäjille: ratkaisuna he ehdottavat visuaalisten kyselyjen kehittämistä pidemmälle. Erilaisia visuaalisia ratkaisuita graafitietokantojen kyselyjen tekemiseen onkin esiintynyt akateemisessa tutkimuksessa: näitä ovat esimerkiksi visuaalinen kyselyjärjestelmä GraphTQL [Pabón *et al.* 2019], visuaalinen RDF-kyselykieli RDF-GL [Hogenboom *et al.* 2010] sekä Query-by-Example -kyselykielen soveltaminen ominaisuusgraafitietokannoille [Lay 2022].

Suorituskyky

Graafitietokantojen suorituskykyä verrataan usein relaatiotietokantoihin. Relaatiotietokannat joutuvat tekemään taulujen välisiä liitoksia hakiessaan linkittynyttä dataa: niiden suorituskyky laskee huomattavasti taulujen koon ja kyselyiden syvyyden (liitosten määrän) kasvaessa [Patil *et al.* 2014]. Graafitietokantojen suorituskykyä linkittyneen datan käsittelyssä pidetään usein hyvänä: kyselyjen suoritusaika pysyy melko vakiona datan kasvaessa [Fernandes & Bernardino 2018].

Graafitietokantojen parempi suorituskyky perustuu siihen, että graafikyselyssä tarvitsee kulkea läpi vain kyselyyn vastaava osa koko graafin sijaan [Patil *et al.* 2014]. Lisäksi graafitietokantoihin toteutettu indeksointi ja varta vasten graafirakenteita varten suunnitellut operaatiot datan tekevät hakemisesta nopeaa [Liu *et al.* 2021].

Monissa tutkimuksissa on verrattu relaatio- ja graafitietokantojen suorituskykyä kokeellisesti. Esimerkiksi Batran ja Tyagin [2012], Khanin ja muiden [2017] sekä Don ja muiden [2022] tutkimuksissa graafitietokanta oli relaatiotietokantaa tehokkaampi. Joissakin kokeissa taas graafitietokantojen suorituskyky oli relaatiotietokantoja huonompi [mm. Kotiranta *et al.* 2022; Mhedbi *et al.* 2021]. Kotirannan ja muiden [2022] tutkimuksessa relaatiotietokanta oli graafitietokantaa nopeampi kompleksisissa ja rekursiivisissa kyselyissä: he nostivat esiin löydöksenä myös sen, että relaatiotietokannat ovat kehittyneet huomattavasti ja modernien relaatiotietokantojen suorituskyky on huomattavasti parempi kuin edeltäjiensä.

Sahu ja muut [2020] nostavat esiin graafitietokantojen suorituskykyongelmana huonon skaalautuvuuden. Käytännössä monet prosessoitavat graafit ovat erittäin isoja (sisältävät yli miljardi kaarta) ja sisältävät paljon erilaisia entiteettejä. Erittäin isoja graafeja käsittelevät kaikenkokoiset yritykset, mikä haastaa sen käsityksen, että isot graafit olisivat vain isojen yritysten kuten Googlen, Facebookin (nykyisin Metan) ja Twitterin ongelma. Skaalautuvuus oli Sahun ja muiden tutkimuksessa useimmin mainittu graafitietokantojen ongelma: ongelmia oli niin isojen graafien lataamisessa, päivittämisessä kuin niihin liittyvän laskennan suorittamisessa. [Sahu *et al.* 2020]

Tian [2023] muistuttaa, että vaikka suorituskyky onkin hyvin tärkeä graafitietokantojen osa-alue, graafitietokantoja tulisi tarkastella myös muista näkökulmista. Hänen mukaansa akateeminen tutkimus on painottunut paljolti graafialgoritmien ja -kyselyiden tehokkuuden tutkimiseen. Kuitenkin graafitietokantojen käyttäjille graafitietokanta on yleensä vain osa suurempaa kokonaisuutta: käyttäjillä on tietokannan suorituskyvyn lisäksi muitakin tarpeita, esimerkiksi analytiikkaan ja tulosten esittämiseen liittyen. [Tian 2023]

Visualisointi

Graafit ovat tehokas työkalu suhteiden havainnollistamiseen: graafien visualisointia voidaan hyödyntää laajasti useissa erilaisissa sovelluskohteissa [Zhang 2010]. Graafien visualisointi onkin usein graafitietokantojen käyttäjille hyvin tärkeää [Sahu *et al.* 2020; Tian 2023]. Useimpiin graafitietokantoihin on rakennettu sisään visualisointityökalu, kun taas jotkut turvautuvat kolmannen osapuolen tekemiin työkaluihin [Tian 2023]. Graafien visualisointia käytetään apuna esimerkiksi datan eksploratiivisessa tutkimisessa, *virheenkorjauksessa* (debugging), kyselyjen muodostamisessa ja esitysvälineenä [Sahu *et al.* 2020].

Graafien visualisointiin liittyy kuitenkin erilaisia käytännön ongelmia: esimerkiksi entiteettien välisten suhteiden esittäminen ymmärrettävällä ja selkeällä tavalla on usein hankalaa [Tian 2023]. Sahu ja muut [2020] mainitsevat visualisoinnin ongelmina räätälöinnin hankaluuden, graafien sijoittelun käyttäjän haluamalla tavalla sekä suurten ja dynaamisten graafien visualisoinnin haasteet.

Turvallisuus

Boza ja Muñoz [2017] tutkivat internettiä skannaten julkisesti löydettäviä Neo4j- ja OrientDB-tietokantoja. Heidän löydöksenään oli, että monissa tietokannoissa oli kyberturvallisuuden näkökulmasta puutteellisia konfiguraatioita. Osassa Neo4j-tietokannoista tunnistautumisessa käytettävien avainten hallintaa tulisi parantaa ja lisäksi niissä tarvittaisiin toimenpiteitä haitallisten kyselyiden estämisen suhteen. OrientDB:n kohdalla ongelmana oli etenkin se, että palvelinta alustettaessa se altistaa itsensä internetiin

oletusarvoisesti. OrientDB loi myös oletusarvoisesti kolme käyttäjää, jotka oli erikseen poistettava asiattoman pääsyn estämiseksi. [Boza & Muñoz 2017]

Hurlburt [2015] puolestaan nosti esiin graafitietokantoihin liittyvänä huolenaiheena yksityisyydensuojan: graafitietokannat ovat tehokkaita löytämään tietoa niin hyvässä kuin pahassa. Mikäli esimerkiksi graafitietokannassa olevien entiteettien identiteettejä ei ole suojattu asianmukaisesti, saattaa niiden kautta paljastua arkaluonteista tietoa. Tietokannassa olevan arkaluonteisen tiedon suojaamisen lisäksi myös graafitietokanta pitää myös suojata tunkeutumista vastaan. [Hurlburt 2015]

Rinnakkaisuus

Rinnakkaisuudella (concurrency, parallelism) tarkoitetaan useita samaan aikaan tapahtuvia laskentaprosesseja, oli kyse sitten monen tietokoneen, saman tietokoneen eri prosessorien tai ohjelman eri säikeiden rinnakkaisuudesta [Amarasinghe *et al.* 2014]. Rinnakkaisuus on käsitteenä lavea ja toisinaan se jaetaankin erikseen *samanaikaisuuteen* (concurrency) ja rinnakkaisuuteen (parallelism): samanaikaisuus tarkoittaa jaettujen resurssien käyttämistä oikein ja tehokkaasti, kun taas rinnakkaisuus on useiden laskentaresurssien käyttämistä tietyn ongelman ratkaisemiseksi nopeammin [Grossman & Anderson 2012]. Graafitietokantojen on usein mainittu tukevan rinnakkaisuutta, mutta monesti suorituskyky ei käytännössä vastaa toivottua [Zhou *et al.* 2018].

Besta ja muut [2019] käyttävät jakoa kyselyjen *samanaikaiseen suorittamiseen* (concurrent execution) ja *rinnakkaiseen suorittamiseen* (parallel execution) tutkiessaan eri graafitietokantojen ominaisuuksia. Samanaikainen suorittaminen tarkoittaa usean kyselyn samanaikaisesta suorittamisesta: se voi johtaa tietokannan korkeampaan tehokkuuteen. Rinnakkainen suorittaminen puolestaan tarkoittaa yhden kyselyn jakamista useammalle laskentaresurssille (esimerkiksi prosessorille): se voi johtaa kyseisen kyselyn nopeampaan suoritusaikaan. Bestan ja muiden yhtenä löydöksenä oli, että useimmat graafitietokannat tukevat monen kyselyn samanaikaista suorittamista, mutta vain harvat graafitietokannat mahdollistavat yhden kyselyn rinnakkaisen suorittamisen. [Besta *et al.* 2019]

3 Tutkimusasetelma

Tässä luvussa kerrotaan työn tutkimusasetelmasta: ensin käsitellään työn tavoitteet ja tutkimuskysymykset (kohta 3.1). Tämän jälkeen käsitellään tutkimusmenetelmäksi valittua systemaattista kirjallisuuskatsausta yleisesti ja esitellään Finkin [2014, ss. 3–5] systemaattisen kirjallisuuskatsauksen prosessi (kohta 3.2). Seuraavissa kohdissa käydään läpi vaiheittain, miten Finkin prosessia on sovellettu tässä työssä: ensin kuvaillaan aineiston hakua (kohta 3.3), sitten aineiston seulontaa (kohta 3.4) ja lopuksi kerrotaan aineiston analyysista (kohta 3.5).

3.1 Tutkimuskysymykset

Tämän tutkimuksen tavoitteena on kartoittaa akateemisesta tutkimuksesta millä sovellusaloilla graafitietokantoja käytetään sekä mitä hyötyjä (positiivisia puolia) ja haittoja (negatiivisia puolia) graafitietokantoihin liitetään. Lisäksi tavoitteena on kartoittaa mitä graafitietokantoja sovelluksissa käytetään. Tavoitteiden pohjalta muodostetut tutkimuskysymykset ovat:

1. Millä sovellusaloilla graafitietokantoja käytetään?
2. Mitä graafitietokantoja sovelluksissa käytetään?
3. Mitä hyötyjä graafitietokantoihin liitetään?
4. Mitä haittoja graafitietokantoihin liitetään?

Työn tutkimusmenetelmänä on systemaattinen kirjallisuuskatsaus. Tutkimuksen aineistona on täten systemaattisella prosessilla valikoitunut joukko tieteellisiä artikkeleita. Valikoitunutta aineistoa käsitellään laadullisen sisällönanalyysin keinoin, jossa aineistoa koodataan ja luokitellaan. Tämän pohjalta tehdään kuvaileva katsaus, jossa vastataan tutkimuskysymyksiin syntynyttä luokitusta apuna käyttäen. [Vuori 2021]

Apuna tulosten käsittelyssä käytetään aineiston kvantifiointia, jolla pyritään havainnollistamaan ilmiöiden esiintymistiheyttä aineistossa [Seitamaa-Hakkarainen 2014]. Tutkimus on luonteeltaan enimmäkseen laadullinen, vaikkakin mukana on määrällisen tutkimuksen elementtejä kvantifioinnin myötä [Saaranen-Kauppinen & Puusniekka 2006].

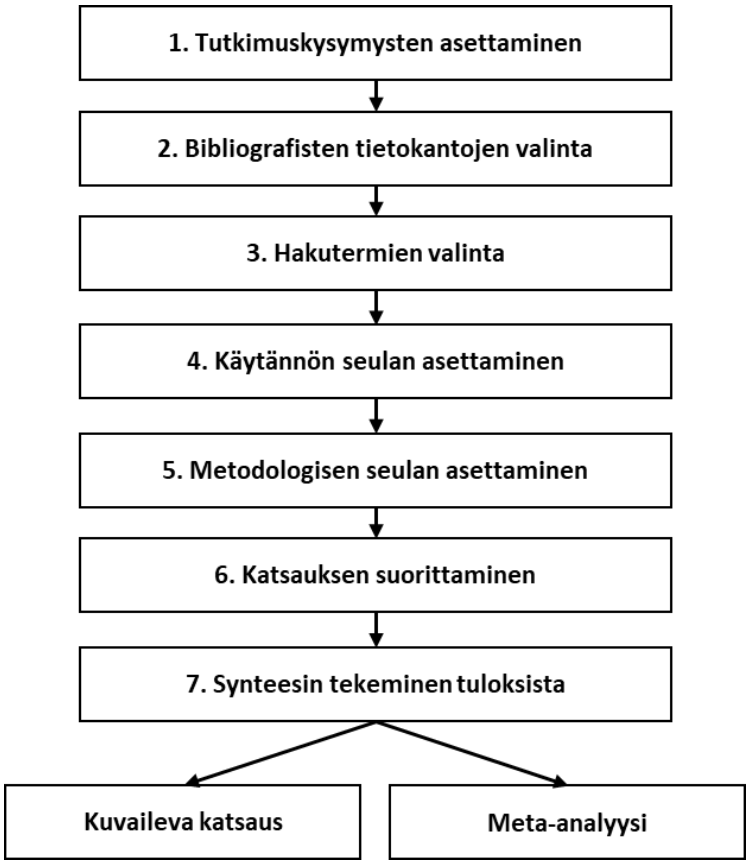
3.2 Systemaattinen kirjallisuuskatsaus

Tieteellisessä tutkimuksessa on oleellista rakentaa ja asemoida se suhteessa jo olemassa olevaan tutkimukseen kirjallisuuskatsauksen avulla. Laveasti määriteltynä kirjallisuuskatsaus on enemmän tai vähemmän systemaattinen tapa kerätä ja syntetisoida aiempaa tutkimustietoa. Lähes kaikki tutkimukset sisältävätkin jonkinlaisen kirjallisuuskatsauksen, mutta tutkimusmetodina käytettynä kirjallisuuskatsauksen tulisi noudattaa katsaustyyppinsä standardeja ja käytäntöjä. [Snyder 2019]

Salminen [2011, s. 6] jakaa kirjallisuuskatsaukset kolmeen perustyyppiin: kuvailevaan kirjallisuuskatsaukseen, systemaattiseen kirjallisuuskatsaukseen ja meta-analyysiin: tässä työssä tutkimusmenetelmäksi päädyttiin valitsemaan systemaattinen kirjallisuuskatsaus. Fink [2014, s. 3] määrittelee systemaattisen kirjallisuuskatsauksen olevan systemaattinen, täsmällinen ja toistettavissa oleva menetelmä, jonka avulla voidaan tunnistaa, arvioida ja tiivistää aihepiirin aiempi tutkimus. Systemaattiseen kirjallisuuskatsaukseen valittuja yksittäisiä tutkimuksia kutsutaan *primääritutkimuksiksi* (primary study), kun taas itse systemaattinen kirjallisuuskatsaus on *sekundaaritutkimus* (secondary study) [Kitchenham & Charters 2007, s. 3].

Systemaattisen kirjallisuuskatsauksen tekemiseen on erilaisia perusteita: Kitchenham ja Charters [2007, s. 3] nostavat esiin näistä yleisimpiä. Ensinnäkin tavoitteena voi olla yhteenvedon tekeminen olemassa olevasta tutkimuksellisesta näytöstä esimerkiksi jonkin teknologian tai hoitomenetelmän suhteen. Toinen mahdollinen tavoite on aukkojen tunnistaminen nykyisessä tutkimuksessa. Kolmanneksi tavoitteena voi olla viitekehyksen tarjoaminen uusien tutkimusaiheiden asemointiin. [Kitchenham & Charters 2007, s. 3]

Systemaattisen kirjallisuuskatsauksen tekeminen koostuu erillisistä vaiheista [Brereton *et al.* 2007]. Fink [2014, ss. 3–5] esittelee nämä vaiheet prosessina, jota sovelletaan tässä tutkimuksessa. Prosessin vaiheet on esitetty kuvassa 3.1.



Kuva 3.1. Systemaattisen kirjallisuuskatsauksen vaiheet.
[mukaillen Fink 2014, s. 4; suomennokset mukaillen Salminen 2011, s. 11].

Ensimmäisessä vaiheessa valitaan tutkimuskysymykset: ne tulee muotoilla tarkkaan, koska ne ohjaavat koko katsauksen tekoa. Tämän jälkeen valitaan bibliografiset tietokannat, joihin haku kohdistuu. Kolmannessa vaiheessa valitaan hakutermit, joilla pyritään rajaamaan aineisto sellaiseksi, että sillä voidaan vastata tutkimuskysymyksiin. [Fink 2014, s. 3].

Neljäs ja viides vaihe ovat aineiston seulontaa. Ensin asetetaan käytännön seula, jolla aineistoa rajataan esimerkiksi julkaisuajankohdan, kielen ja artikkelin tyypin perusteella. Seuraavaksi seulotaan hakutuloksia metodologisesti: tällöin arvioidaan aineiston tieteellistä laatua ja pyritään valikoimaan katsaukseen laadukasta materiaalia. [Fink 2014, ss. 4–5]

Kuudennessa vaiheessa tehdään varsinainen kirjallisuuskatsaus. Jotta katsaus olisi luotettava ja pätevä, tulee aineiston käsittely tehdä standardoidussa muodossa ja prosessin laatua tulee valvoa. Lisäksi ennen varsinaisen katsauksen aloittamista prosessi olisi hyvä pilotoida ja mahdolliset avustajat tulisi perehdyttää tehtäviinsä. [Fink 2014, ss. 4–5]

Seitsemäs ja viimeinen vaihe on tulosten syntetisointi. Tähän vaiheeseen kuuluu tämänhetkisen tiedon raportointi ja löydösten selittäminen. Synteesi voidaan tehdä laadullisesti kuvailevana katsauksena tai tilastollisena meta-analyysina. Kuvailevat katsaukset ovat relevantteja etenkin, jos tilastotieteellisiä tutkimuksia ei ole saatavilla. Meta-analyysi puolestaan käyttää aineistonaan juuri korkealaatuisia tilastollisia tutkimuksia (kontrolloituja satunnaisotantaan perustuvia kokeita ja täsmällisiä havainnointitutkimuksia). [Fink 2014, ss. 4–5, 199].

Tässä tutkimuksessa sovelletaan Finkin mallin lisäksi myös muita ohjeita aineiston analyysissa prosessin kahdessa viimeisessä vaiheessa. Muiden kirjoittajien ohjeiden soveltamisesta mainitaan erikseen asianmukaisissa paikoissa.

3.3 Aineiston haku

Tutkimusta suunnitellessa tunnistettiin aiempien tietojenkäsittelytieteen eri tutkimusalojen systemaattisten kirjallisuuskatsausten kautta [mm. Dias *et al.* 2021; Yasuda *et al.* 2020; Voronkov *et al.* 2017], että artikkelihaun tulisi sisältää ainakin ACM:n (Association for Computing Machinery) ja IEEE:n (Institute of Electrical and Electronics Engineers) tietokannat. Tietokannaksi valikoitui Elsevierin Scopus, koska se kattaa julkaisuita laajasti ja sisältää viitteet myös ACM:n ja IEEE:n lehtiin [Elsevier 2023].

Hakutermejä valittaessa pyrittiin muodostamaan haku niin, että saataisiin mukaan artikkeleita, joissa graafitietokantasovellus on tärkeässä osassa. Heti alussa keskeiseksi hakutermiksi tunnistettiin ”graph database”. Tällä haettaessa otsikoista, tiivistelmistä ja avainsanoista saatiin tuloksia runsaasti, mutta suuri osa artikkeleista ei ollut relevantteja tämän tutkimuksen kannalta. Kokeilun kautta haku rajattiin vain artikkelien otsikoihin.

Lisäksi tässä vaiheessa tunnistettiin, että joidenkin graafitietokantasovelluksia käsittelevien artikkelien kohdalla termi ”graph database” ei esiinny otsikossa, vaan

nimessä on mukana jonkin graafitietokannan nimi. Erityisesti graafitietokanta Neo4j esiintyi tällä tavoin otsikoissa, esimerkiksi artikkelissa *"Modeling and Processing Big Data of Power Transmission Grid Substation Using Neo4j"* [Perçuku *et al.* 2017]. Tämän perusteella hakutermeihin päätettiin ottaa mukaan hakuhetkellä kahdeksan suosituimman graafitietokannan nimi [Solid IT 2022]. Testihauissa kokeiltiin myös suurempaa määrää tietokantojen nimiä, mutta suurempi määrä nimiä ei enää tuottanut lisäosumia.

Hakuun käytettiin Scopusin *Advanced search* -toimintoa. Scopusin hakuun sai määritettyä otsikon (TITLE) lisäksi hakuehtoina julkaisuvuoden (PUBYEAR), julkaisun tyypin (SRCTYPE), dokumenttityypin (DOCTYPE) ja kielen (LANGUAGE). Näistä kerrotaan tarkemmin kohdassa 3.4 käytännön seulomisen yhteydessä. Lopulliseksi hakulausekkeeksi muodostui (hakukenttien arvot lihavoitu):

```
TITLE ("graph database" OR "Neo4j" OR "Cosmos DB"
   OR "ArangoDB" OR "OrientDB" OR "GraphDB"
   OR "Amazon Neptune" OR "JanusGraph" OR "TigerGraph")
AND PUBYEAR > 2016 AND PUBYEAR < 2022
AND (LIMIT-TO(SRCTYPE, "p") OR LIMIT-TO(SRCTYPE, "j"))
AND (LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ar"))
AND (LIMIT-TO(LANGUAGE, "English"))
```

Haku suoritettiin 4.6.2022 ja se tuotti 411 artikkelia tulokseksi. Haun tulokset ladattiin csv-formaatissa Scopusista ja saatu tiedosto muunnettiin taulukkolaskenta-ohjelmalla taulukkoformaattiin.

3.4 Aineiston seulonta

Aineiston seulonta tapahtuu kahdessa vaiheessa: käytännön ja metodologisen seulan kautta. Näistä ensimmäinen vaihe, käytännön seula, tarkoittaa aineiston seulomista käytännöllisten kriteerien kuten tutkimuksen sisällön, julkaisuvuoden, julkaisutyypin, kielen, rahoituslähteen sekä tutkimusasetelman ja -metodien perusteella. Metodologinen seula puolestaan rajaa aineistoa yrittäen löytää tutkimukset, jotka noudattavat hyviä tieteellisiä käytäntöjä parhaiten. Tieteellisten käytäntöjen noudattamista voi arvioida tutkimuksen eri osa-alueiden kautta: miten tutkimus on suunniteltu, miten otanta on tehty, miten data on kerätty ja analysoitu, millaista on tulosten tulkinta ja miten tutkimus on raportoitu. [Fink 2014, s. 48]

Käytännön seulonnassa käytetään sisäänotto- ja poissulkukriteerejä: jokaisen mukaan otettavan artikkelin tulee täyttää jokainen sisäänottokriteeri eivätkä ne saa täyttää yhtäkään poissulkukriteereistä [Fink 2014, s. 53]. Tässä tutkimuksessa tavoitteena oli saada aineistoksi uudehkoja, tarpeeksi laajoja graafitietokantojen sovelluksia käsitteleviä artikkeleita, joilla saataisiin vastauksia tutkimuskysymyksiin (mikä on sovellusala; mitä graafitietokantaa on käytetty; mitä hyötyjä ja haasteita graafitietokannan käyttöön liittyy). Tämän tavoitteen perusteella luotiin sisäänottokriteerit, jotka on kuvattu taulukossa 3.1.

Sisäänottokriteerit
Artikkelin julkaisuvuosi on välillä 2017–2021.*
Artikkelin julkaisun tyyppi on vertaisarvioitu tieteellinen lehti tai konferenssijulkaisu.*
Artikkelin dokumenttityyppi on lehtiartikkeli tai konferenssiartikkeli.*
Artikkeli on englanninkielinen.*
Artikkeli on vähintään 6 sivua pitkä.
Artikkelissa oleva tietokanta on graafitietokanta tai RDF-tietokanta.
Artikkelissa käsitellään graafitietokantojen käyttöä jollain sovellusalalla.
Artikkelissa on esitelty jonkinlainen graafitietokantoja hyödyntävä sovellus.
Artikkelissa käsitellään graafitietokannan käytön hyötyjä ja/tai haasteita.
Artikkeli on primäärinen tutkimus.

Taulukko 3.1. Tutkimuksen sisäänottokriteerit. Hakulausekkeessa huomioidut kriteerit on merkitty tähdellä (*).

Neljä ensimmäistä kriteeriä (julkaisuvuosi, julkaisun tyyppi, dokumenttityyppi, ja artikkelin kieli) sisällytettiin hakulausekkeeseen. Julkaisuvuosiksi valittiin 2017–2021, koska aineisto haluttiin rajata uudehkoon materiaaliin ja julkaisuvuosi 2022 oli vielä hakua tehtäessä kesken. Julkaisun tyypiksi asetettiin *vertaisarvioidut tieteelliset lehdet* (journal) ja *konferenssijulkaisut* (conference proceedings). Lisäksi artikkelin dokumenttityypiksi asetettiin *lehtiartikkeli* (journal article) tai *konferenssiartikkeli* (conference paper), sillä vertaisarvioiduissa tieteellisissä lehdissä ja konferenssijulkaisuissa voi esiintyä muunkin tyyppisiä tekstejä (esimerkiksi esipuheita ja konferenssien työpajojen raportteja). Kieleksi valittiin englanti, koska suurin osa relevantista tutkimuksesta on sen kielistä.

Viides sisäänottokriteeri on artikkelin pituus. Artikkelin minimipituudeksi asetettiin 6 sivua, jotta mukaan otettavissa artikkeleissa olisi riittävästi sisältöä arvioitavaksi. Tämä pituusrajoitus rajaa pois joitakin lyhyitä konferenssiartikkeleita (short paper) ja mahdollisesti myös lyhyitä lehtiartikkeleita.

Taulukon 3.1 viisi alinta sisäänottokriteeriä ottavat kantaa artikkelien sisältöön. Ensinnäkin artikkelien tulee käsitellä graafitietokantoja tai RDF-tietokantoja: muun tyyppiset tietokannat rajattiin pois, mukaan lukien graafien prosessointiviitekehykset. Toisekseen artikkelissa tuli olla jokin sovellusala määritettynä, jotta se voi vastata ensimmäiseen tutkimuskysymykseen. Tällä kriteerillä yleisen tason artikkelit rajautuvat pois. Kolmanneksi artikkeleissa tulee esitellä jokin sovellus, josta on toteutettu vähintään jonkinlainen prototyyppi. Tämä kriteeri rajaa pois artikkelit, jotka ovat puhtaasti teoreettisia tai keskittyvät esimerkiksi tietokannan mallintamiseen.

Neljäs sisältöön liittyvä sisäänottokriteeri on, että artikkelien tulee käsitellä graafitietokantojen käytön hyötyjä ja/tai haittoja, jotta se voisi vastata kolmanteen ja

neljänteen tutkimuskysymykseen. Viimeisenä sisältökriteerinä on, että artikkelin tulee olla primäärinen tutkimus, joten kirjallisuuskatsaukset rajautuvat pois.

Sisäänottokriteerien lisäksi aineistoa rajattiin poissulkukriteereillä, jotka on esitetty taulukossa 3.2. Ensiksi aineistosta poistettiin Scopusin hakutuloksiin sisältyneet saman artikkelin duplikaatit. Toiseksi tarkasteltiin kuuluuko tutkimuksesta aineistoon toinen versio: tämä on tyypillinen tilanne, kun samasta tutkimuksesta on tehty konferenssi- ja lehtiartikkeli. Tällöin mukaan valittiin täydellisin versio eli käytännössä lehtiartikkeli. Kolmanneksi aineistosta rajattiin pois artikkelit, johon ei ole pääsyä (eivät sisälly yliopiston tilaukseen eikä niistä ole saatavilla *open access* -versiota). Viimeiseksi aineistosta rajattiin pois artikkelit, joissa käytettyä tietokantaa ei ollut nimetty tai tietokannan toteutuksesta ei ollut tarpeeksi tietoa, jotta se voitaisiin luokitella graafitietokannaksi.

Poissulkukriteerit
Artikkeli sisältyy jo aineistoon (duplikaattien poisto).
Artikkelista sisältyy jo aineistoon täydellisempi versio.
Artikkeliin ei ole pääsyä.
Artikkelissa käsiteltyä tietokantaa ei ole nimetty tai tietokannan toteutuksesta ei ole tarpeeksi tietoa.

Taulukko 3.2. Tutkimuksen poissulkukriteerit.

Kaikki aineistohaun löytämät 411 artikkelia käytiin läpi sisäänotto- ja poissulkukriteerien osalta. Artikkeleja tarkasteltiin ensisijaisesti haun tuottamien bibliografisten tietojen, otsikon ja tiivistelmän perusteella. Mikäli näiden perusteella ei pystytty tekemään päätöstä sisäänotosta tai poissulkemisesta, artikkelin sisältöön tutustuttiin tarkemmin. Jokaisen artikkelin kohdalle lisättiin aineistotaulukkoon tieto siitä, onko se mukana aineistossa vai ei. Mikäli artikkeli jätettiin pois, sille kirjattiin ensimmäinen vastaan tullut poissulkemisen aiheuttanut syy: nämä on esitetty taulukossa 3.3. Artikkelilla olisi toki saattanut olla useita syitä poissulkemiseen, mutta näitä ei kirjattu tarkemmin ylös.

Käytännön seula rajasi pois 293 artikkelia ja jäljelle jäi 118 artikkelia. Huomionarvoista poisjätetyissä artikkeleissa on, että merkittävä osa (n = 183) artikkeleista jätettiin pois, koska sisällölliset kriteerit eivät täyttyneet. Toiseksi yleisin syy poissulkemiseen oli artikkelien pituus (n = 69) ja kolmanneksi yleisin se, ettei artikkeliin ollut pääsyä (n = 16). Lisäksi erikseen voi mainita, että 12 artikkelia oli tullut sisällytetyksi haun tuloksiin Scopusin virheellisten bibliografisten tietojen vuoksi. Näistä yhdeksän julkaisuvuosi ei kuulunut välille 2017–2021 ja kolmen artikkelin tyyppi oli muu kuin konferenssi- tai lehtiartikkeli.

Sisäänottokriteerit eivät täyty	**Frekvenssi**
Julkaisuvuosi ei välillä 2017–2021	9
Ei ole konferenssi- tai lehtiartikkeli	3
On alle 6 sivua pitkä	69
Ei käsittele graafitietokantoja	3
Sisällölliset kriteerit eivät täyty	183
Poissulkukriteerit täyttyvät	
Duplikaatit	4
Löytyy täydellisempi versio	3
Ei saatavilla	16
Tietokantaa ei nimetty tai toteutus tuntematon	3
Yhteensä	293

Taulukko 3.3. Aineistosta poisjätetyt artikkelit kriteereittäin.

Seuraavaksi tarkasteltiin artikkeleja metodologisessa seulonnassa. Tässä tutkimuksessa metodologinen seula pidettiin tarkoituksella varsin sallivana, koska tutkimuksessa haluttiin kerätä laajasti tietoa. Toisekseen tämä tutkimus on luonteeltaan laadullinen ja kartoittava, mikä ei edellytä yhtä tiukkaa seulontaa kuin esimerkiksi tilastotieteellisiin menetelmiin perustuva meta-analyysi.

Metodologissa seulonnassa suljettiin pois 7 artikkelia, jotka olivat jollain tavoin merkittävästi puutteellisia tai käsittelivät keskeneräistä työtä. Artikkeleissa olevia puutteita olivat esimerkiksi hyvin lyhyet tai kokonaan puuttuvat tieteellisen raportin osiot. Lisäksi joissain artikkeleissa oli merkittäviä ongelmia kielen ymmärrettävyyden ja oikeinkirjoituksen kanssa.

Näin ollen lopulliseksi aineistoksi muodostui 111 artikkelia, jotka on listattu viiteluettelossa kohdassa 7.2. Artikkelit jakautuivat melko tasaisesti vuosille 2017–2021, lukuun ottamatta vuotta 2019, jonka ajalta vain 11 artikkelia päätyi mukaan katsaukseen. Artikkelien jakauma julkaisuvuosittain on esitetty taulukossa 3.4.

Julkaisuvuosi	**Frekvenssi**
2017	27
2018	23
2019	11
2020	21
2021	29

Taulukko 3.4. Artikkelien määrä julkaisuvuosittain.

Kirjallisuuskatsauksen artikkeleista 52 on vertaisarvioituja lehtiartikkeleita ja 59 konferenssiartikkeleita. Eri julkaisunimikkeitä on runsaasti: mukana on sekä tietojenkäsittelytieteen että sovellusalojen julkaisuita.

3.5 Aineiston analyysi

Finkin [2014, ss. 4–5] prosessin seuraava vaihe on itse kirjallisuuskatsauksen tekeminen, jonka keskeinen osa on aineiston analysointi. Tässä tutkimuksessa aineiston analyysimenetelmänä on laadullinen sisällönanalyysi ja sitä täydennetään kvantifioinnin keinoin [Juhila 2021a]. Aineiston analyysin jälkeen prosessin viimeisenä vaiheena on synteesin tekeminen tuloksista: tässä työssä synteesi tehdään kuvailevana katsauksena, joka on etupäässä laadullinen tapa esittää tulokset [Fink 2014, ss. 4–5].

Kuten edellä todettiin, analyysimenetelmäksi valikoitui laadullinen sisällönanalyysi. Laadullisessa sisällönanalyysissa keskitytään nimensä mukaisesti analysoimaan sisältöä eli sitä mistä asioista ja aiheista aineistossa kerrotaan: kielellistä tai ilmaisullista muotoa ei oteta tällöin analyysin kohteeksi. Laadullista sisällönanalyysia voi käyttää sekä kirjallisen että ei-kirjallisen aineiston (ääni, kuva) analyysiin. [Vuori 2021]

Laadullinen sisällönanalyysi perustuu tutkijan tekemään koodaukseen. Juhilan [2021a] mukaan koodauksessa on kyse siitä, että aineisto järjestetään ja luokitellaan ennen varsinaista analyysia. Juhila jatkaa, että koodauksessa aineiston osia (esimerkiksi katkelmia litteroiduista haastatteluista tai artikkeleista) yhdistellään ja erotellaan jonkin ominaisuuden mukaan. Tämän jälkeen samankaltaiset osat luokitellaan yhteen ja syntynyt luokka nimetään yhteisen ominaisuuden mukaan. Luokittelu muotoutuu iteratiivisesti koodausprosessissa tutkijan jäsennystyön tuloksena. Prosessin aikana luokittelu elää ja joustaa: luokkia saatetaan esimerkiksi nimetä uusiksi ja luokat voivat yhdistyä tai jakautua useammaksi luokaksi. [Juhila 2021a]

Koodaus voi olla aineisto- tai teorialähtöistä. Teorialähtöisessä koodauksessa aiemmasta tutkimuksesta syntyneet kategoriat ja käsitteet ohjaavat koodausta, kun taas aineistolähtöisessä koodauksessa tutkija pyrkii muodostamaan käsitteet ja luokittelun aineiston perusteella [Juhila 2021a]. Tässä tutkimuksessa koodaus ja luokittelu on pyritty muodostamaan aineistolähtöisesti, mutta siihen on vaikuttanut aineistona olevien artikkelien kautta myös teoria. Tällainen aineiston ja teoreettisen käsitteellistämisen yhteistyönä syntynyt luokittelu onkin melko yleinen laadullisessa sisällönanalyysissa [Seitamaa-Hakkarainen 2014].

Käytännössä koodaus tehtiin tässä tutkimuksessa kolmella kierroksella. Artikkelit käytiin läpi ja jokaiselle laadittiin oma artikkelikorttinsa, johon poimittiin tutkimuskysymyksiin vastaavia tietoja. Artikkelikorttiin koodattiin tietoina tutkimuksen sovellusala, vapaamuotoinen kuvaus tutkimuksen sisällöstä, käytetyt graafitietokannat sekä mitä graafitietokantojen hyötyjä ja haittoja artikkelissa on mainittu. Ensimmäisellä kierroksella koodaus tehtiin melko vapaasti pyrkien mahdollisuuksien mukaan käyttämään jo luetuissa artikkeleissa esiintyneitä luokkia. Toisella kierroksella pyrittiin yhdistelemään eri luokkia ja kolmas kierros oli luonteeltaan ennen kaikkea tarkistuskierros.

Jokaiselle artikkelille määritettiin yksi sovellusala: se määritettiin tutkimuksen aiheen, julkaisun, avainsanojen sekä tarvittaessa sisällön kautta. Jotkin artikkeleista olisivat saattaneet kuulua kahteen eri luokkaan: tällöin valittiin toinen luokista. Esimerkiksi Hall ja muut [2017] käsittelivät biokemian suosittelujärjestelmää, joten artikkeli olisi voinut kuulua sekä *biokemia*- että *suosittelujärjestelmät*-luokkaan. Tässä tapauksessa päädyttiin biokemiaan, koska se oli pääasiallinen teema artikkelissa.

Artikkeleille laadittiin sovellusalan lisäksi lyhyt vapaamuotoinen kuvaus. Tätä kuvausta käytettiin apuna, kun sovellusaloille muodostettiin alakategorioita. Alakategoriat muodostettiin varsinaisen koodauksen valmistuttua.

Käytetyt graafitietokannat poimittiin artikkelin sisällöstä: usein näitä oli vain yksi, mutta joissain tapauksissa useampia. Muun tyyppisiä tietokantoja ei koodattu, vaikka esimerkiksi relaatiotietokanta esiintyi monessa artikkelissa vertailukohteena. Terminologian suhteen voidaan huomauttaa, että termillä graafitietokanta käsitetään sekä varsinaiset graafitietokannat että RDF-tietokannat, ellei jakoa ei näiden välille erikseen tehdä.

Graafitietokannan käytön hyödyt ja haitat poimittiin artikkelin sisällöstä. Hyödyt ovat graafitietokantojen käyttöön liittyviä positiivisia asioita; haasteet puolestaan negatiivisia asioita. Molemmat koodattiin jonkinlaisen väittämän tai toteamuksen perusteella: ei vaadittu, että hyötyä tai haastetta olisi empiirisesti todennettu, tekijöiden maininta asiasta riitti. Kuitenkaan hyötyä tai haastetta ei koodattu, jos viittaus oli suoraan muuhun kirjallisuuteen ja sen yhteyttä tarkasteltuun tutkimukseen ei avattu mitenkään. Sama hyöty tai haitta joko esiintyi tai ei esiintynyt artikkelissa: koodaus ei ota kantaa hyödyn tai haitan suuruuteen tai yleisyyteen yhden artikkelin sisällä.

Tyypillisesti hyödyt löytyivät artikkeleista niistä kohdista, joissa perusteltiin graafitietokannan valintaa. Näin on esimerkiksi seuraavassa viittauksessa, joka koodattiin *soveltuvuus verkottuneen datan käsittelyyn* -nimiseksi hyödyksi:

> *”Compared with the traditional relational database, the graph database achieves data storage and relationship expression in the form of vertices and edges, whose representation has a natural advantage in describing physical networks and communication networks.”* [Wang *et al.* 2017]

Hyötyjä ja haittoja löytyi usein myös artikkelien tuloksista ja johtopäätöksistä. Esimerkiksi seuraava viittaus koodattiin hyödyksi *suorituskyky*-luokan alle:

> *”Experiments compare two system prototypes, that are respectively based on Python and Neo4j, showing that the latter presents better performance in terms of processing time guaranteeing the same accuracy.”* [Celesti *et al.* 2020]

Koodauksen valmistuttua artikkelikorttien tiedot koottiin tulostaulukkoon, joka on esitetty liitteessä 1 (taulukko 7.1). Aineiston koko (n = 111) on sen verran suuri, että laadullista analyysia päätettiin täydentää kvantifioimalla aineistoa. Kvantifioimalla voidaan esimeriksi laskea sitä, kuinka paljon jotain ominaisuutta sisältäviä elementtejä aineistoon sisältyy. Kvantifiointi soveltuu tilanteisiin, joissa halutaan tehdä yhteenveto rikkaasta ja moninaisesta aineistosta. [Seitamaa-Hakkarainen 2014]

Saaranen-Kauppisen ja Puusniekan mukaan laskemisella voidaan havainnollistaa tutkimustuloksia ja systematisoida analyysia. He huomauttavat, että kyseessä on kuitenkin tutkijan luoma konstruktio. Lisäksi he huomattavat, että kun laadulliseen tutkimukseen otetaan mukaan määrällisiä elementtejä, tulisi unohtaa tilastotieteelliset vaatimukset yleistämisestä ja tunnuslukujen laskemisesta. [Saaranen-Kauppinen & Puusniekka 2006] Näissä on riskinä, että tutkimusta alettaisiin arvioida määrällisen tutkimuksen käsitteillä, vaikka tutkimusasetelma on laadullinen [Mäkelä 1990, viitattu lähteessä Saaranen-Kauppinen & Puusniekka 2006].

Tässä tutkimuksessa kvantifiointia käytettiin kaikkien tutkimuskysymysten kohdalla apuna havainnollistamisessa. Laskettavia asioita olivat sovellusalat, käytetyt graafi-tietokannat sekä graafitietokantojen hyödyt ja haitat. Näihin liittyvät taulukot on esitetty tulosten yhteydessä.

Analyysin valmistuttua päästiin tekemään kuvaileva katsaus, joka on esitetty tämän työn luvuissa 4 (tulokset) ja 5 (pohdinta). Kuvailevassa katsauksessa tutkija tulkitsee oman tietämyksensä ja kokemuksensa kautta aineistoa yrittäen löytää tutkimusten tarkoituksista, löydöksistä ja metodeista samankaltaisuuksia ja erilaisuuksia. Kuvailevan katsauksen validiteetti riippuu tutkijan tietämyksestä tutkittavan ja kriittisestä ajattelusta aiheen suhteen sekä tutkittavan aineiston laadusta. [Fink 2014, s. 199]

4 Tulokset

Tässä luvussa esitetään systemaattisen kirjallisuuskatsauksen tulokset. Ensin käsitellään sovellusalojen luokittelua (kohta 4.1), sitten käytettyjä graafitietokantoja (kohta 4.2) ja näiden jälkeen artikkeleissa mainittuja graafitietokantojen hyötyjä (kohta 4.3) ja haittoja (kohta 4.4).

4.1 Sovellusalojen luokittelu

Ensimmäisenä tutkimuskysymyksenä oli ”Millä sovellusaloilla graafitietokantoja käytetään?”. Kaiken kaikkiaan artikkelien sovellusaloista muodostettiin eri luokkia 25: syntynyt luokittelu on esitetty taulukossa 4.1. Artikkeleista määritettiin sovellusalan lisäksi myös tarkempi aihe. Artikkelien aiheiden tarkempi luokittelu tekijätietojen kera on esitetty kokonaisuudessaan liitteessä 2 esiintyvässä taulukossa 7.2.

Sovellusala	**Frekvenssi**
Bioinformatiikka	16
Sosiaaliset verkostot	15
Tietoverkot	11
Geografinen tieto	10
Energia	7
Suosittelujärjestelmät	7
Koneoppiminen	5
Ohjelmistotuotanto	5
Esineiden internet	4
Tietämysgraafit	4
Biokemia	3
Palveluntuotanto	3
Rakentaminen	3
Turvallisuus	3
Akateeminen tutkimus	2
Epidemiologia	2
Kulttuuri	2
Tuotannonohjaus	2
Lainsäädäntö	1
Maatalous	1
Politiikka	1
Rahoitus	1
Robotiikka	1
Terveydenhuolto	1
Tähtitiede	1
Yhteensä	111

Taulukko 4.1. Kirjallisuuskatsauksen artikkelien sovellusalat luokiteltuna.

Bioinformatiikan alan artikkelit (n = 16) muodostavat melko selvärajaisen joukon. Bioinformatiikan voidaan määritellä olevan ala, jossa käsitellään laskennallisten menetelmien avulla biologisiin makromolekyyleihin (kuten DNA:han ja aminohappo-sekvensseihin) liittyvää tietoa [Luscombe *et al.* 2001]. Bioinformatiikassa graafitietokannoille löytyy aineistosta sovelluskohteiksi biologisen datan tietokannat, erilaiset bioinformatiikan analyysit, biologisen datan visualisointi, syöpätutkimus ja bioinformatiikan työnkulkujen kuvaus.

Sosiaalisten verkostojen sovelluksia oli 15 artikkelissa: myös ne muodostavat selvärajaisen joukon. Yleisin sovelluskohde oli yhteisöjen tunnistaminen sosiaalisista verkostoista, joka esiintyi viidessä artikkelissa. Myös muunlaiset analyysit olivat suosittuja: sitaattiverkkojen analysointi, sosiaalisen maineen analysointi, Twitter-sisältöjen analysointi, trollien analysointi ja tutkiva jäljittäminen. Näiden lisäksi aiheina oli kahdessa artikkelissa sosiaalisen verkoston tietokanta, jotka keskittyivät datan säilyttämiseen, ja yhdessä artikkelissa reaaliaikaisen viestinnän sovellus.

Tietoverkkojen alan artikkelit (n = 11) käsittelevät tietoverkkojen hallintaa ja kyberturvallisuutta. Kyberturvallisuuteen liittyviä artikkeleita oli yhteensä 8: aiheina näissä oli verkkoliikenteen analyysi, verkon pääsynhallinta, kyberrikosten tutkinta sekä kyberhyökkäysten simulointi. Tietoverkkojen hallintaa käsiteltiin kolmessa artikkelissa. Näissä aiheina oli tietoverkon konfigurointidatan hallinta, virtualisoidun verkon infrastruktuuri sekä ohjelmallisesti määritettyä verkko.

Geografista tietoa käsitteleviä artikkeleja oli 10. Monet sovellukset liittyivät paikkatietojärjestelmiin. Artikkelien aiheina oli kaupunkien mallintaminen, *rakennuksen tietomallin* (building information model, BIM) ja paikkatietojärjestelmän datan integrointi, älykaupunkipalvelut, kognitiivinen kaupunki, julkisen liikenteen reitinsuunnittelu, infrastruktuuri- ja IoT-mittaridatan yhdistäminen, jokijärjestelmän tietokanta, paikkatietojen kuvaukset ja liikkuvien objektien analyysi.

Energia-alan sovelluksia käsitteleviä artikkeleja oli 7. Näistä neljässä käsiteltiin sähköverkkojen mallintamista tietokantoihin. Kahdessa artikkelissa käsiteltiin sähköverkkojen analysointia: sähköverkon turvallisuuden arviointia sekä voimavirran analyysia. Lisäksi yhden artikkelin aiheena oli tuulivoima-alan tietokanta.

Suosittelujärjestelmiä käsiteltiin 7 artikkelissa: aiheina olivat kirjojen, lääkärien, ruokareseptien, sosiaalisten suositusten, fyysisten tuotteiden ja työnhaun suosittelujärjestelmät. Lisäksi yksi artikkeli keskittyi suosittelujärjestelmän laskennan nopeutukseen graafitietokannan avulla.

Koneoppimista käsiteltiin viidessä artikkelissa. Näistä kahdessa käsiteltiin luonnollisen kielen prosessointia. Muiden artikkelien aiheina olivat neuroverkot, puheentunnistus sekä etäkuntoutuspalvelun koneoppimisalgoritmin parantaminen.

Ohjelmistotuotannon alan artikkeleja oli 5. Ohjelmistokehityksessä graafeja voidaan hyödyntää monella eri tasolla ohjelmakoodin mallintamisessa. Yhden artikkelin aiheena oli staattinen PHP-koodin analyysi, jossa ohjelmakoodi oli kuvattu *abstraktina syntaksipuuna* (abstract syntaxt tree). Kahdessa artikkelissa hyödynnettiin ohjelman *ohjausvuograafeja* (control flow graph): toisessa sitä hyödynnettiin plagiarismin havainnointiin ja toisessa mittaamaan staattisen koodianalyysin tehokkuutta. Muissa artikkeleissa kiinnostus oli ohjelmistoprojektien hallinnassa: näissä käsiteltiin automaattista bugikorjausten osoitusta sekä koodin jäljitettävyyden säilömistä repositorioihin.

Esineiden internetiä (IoT, Internet of things) käsiteltiin neljässä artikkelissa. Esineiden internetillä tarkoitetaan hajautettua verkkoa, joka yhdistää fyysisiä objekteja, joilla on kyky havainnoida tai vaikuttaa ympäristöönsä, ja jotka pystyvät kommunikoimaan muiden laitteiden tai tietokoneiden kanssa [Davies 2015]. Aiheina esineiden internetiä käsittelevissä artikkeleissa oli IoT-laitteiden tuottaman datan analyysi, IoT-laitteiden hallinta, IoT-laitteiden hyödyntäminen taistelukentän hallintajärjestelmissä sekä langattomat kyberfyysiset järjestelmät.

Tietämysgraafeja käsitteli 4 artikkelia: näistä kahdessa käsiteltiin ontologioita. Hoganin ja muiden [2021] mukaan ontologiat ovat formaaleja esityksiä tietämyksestä, jotka voidaan esittää graafina: Ehrlinger ja Wöß [2016] huomauttavat, että ontologiat voidaan myös nähdä tietämysgraafeina. Ontologioiden lisäksi muina aiheina oli tietämysgraafien yhdistäminen ja Wikidatan graafimuotoinen tietokanta, joita käsiteltiin kumpaakin yhdessä artikkelissa.

Biokemian alan artikkeleja oli kolme: näissä käsiteltiin biokemiallista analyysia, biokemiallisen datan tietokantaa sekä biokemiallista suosittelujärjestelmää. **Palvelun-tuotannon** (n = 3) artikkelit käsittelivät palvelun koostumuksen määritystä, palvelutoimittajien suosittelua sekä palveluiden tuotekatalogeja. **Rakentamisen** alan artikkeleissa (n = 3) puolestaan aiheina olivat rakennuksen tietomalli sekä uudenlainen tietomalli elementtitaloille. **Turvallisuuden** alan artikkeleissa (n = 3) käsiteltiin fyysisen pääsyn kontrollointijärjestelmää, petosten havaitsemista sekä rikollisiin liittyvien dokumenttien tietokantaa. Erona tietoverkkojen alla oleviin kyberturvallisuuden sovelluksiin, näissä artikkeleissa turvallisuus sovellusalana oli yleisempi eikä pelkästään tietotekninen tai tietoverkkoihin liittyvä.

Akateemisen tutkimuksen (n = 2) sovelluksissa aiheena oli tiedon metamalli ja tutkimusten semanttinen indeksointi. **Epidemiologiassa** (n = 2) käsiteltiin tautikontaktien jäljitystä. **Kulttuurin** (n = 2) alan artikkeleissa käsiteltiin elokuvien vaikuttavuutta ja kulttuurisen perinnön tietokantaa. **Tuotannonohjauksen** (n = 2) artikkelit käsittelivät tuotantolaitteiden suorituskyvyn analyysia ja tuotantotehtävien aikataulutusta.

Loput luokista sisälsivät yhden artikkelin kukin. Tällaisia luokkia olivat **lainsäädäntö** (artikkelin aiheena yleisemmän tapauksen löytäminen), **maatalous**

(viljelyskasvien tietokanta), **politiikka** (vaalidatan analysointi), **rahoitus** (konkurssien ennustaminen), **robotiikka** (mobiilirobotiikan tietokanta), **terveydenhuolto** (sähköiset potilastiedot) ja **tähtitiede** (avaruusobjektien rekisteröintidata).

4.2 Sovelluksissa käytetyt graafitietokannat

Toisena tutkimuskysymyksenä tarkasteltiin käytettyjä graafitietokantoja. Suurimmassa osassa artikkeleita käsiteltiin yhtä graafitietokantaa. Tämän lisäksi kuudessa artikkelissa käytettiin (usein vertailua varten) kahta tai useampaa graafitietokantaa. Tulokset on esitetty taulukossa 4.2.

Käytetty graafitietokanta	Lukumäärä
Neo4j	93
Useita graafitietokantoja	6
BlazeGraph	2
OrientDB	2
TigerGraph	2
ArangoDB	1
Dgraph	1
JasmineGraph	1
Nepal	1
S2JGraph	1
SeQuery	1

Taulukko 4.2. Artikkeleissa käytetyt graafitietokannat.

Ylivoimaisesti käytetyin graafitietokanta oli Neo4j (n = 93). Suosioon vaikuttavia tekijöitä ovat muun muassa, että Neo4j:stä on saatavilla ilmainen *community*-versio [Geepalla & Asharif 2020], sen käyttöliittymää on kuvailtu helppokäyttöiseksi [Aung & Nyunt 2020], siihen on tarjolla laaja graafialgoritmien kirjasto [Bollen *et al.* 2021] ja sen kyselykieltä Cypheria pidetään helposti käytettävänä [Fabregat *et al.* 2018].

Muita käytettyjä graafitietokantoja olivat ArangoDB, Dgraph, OrientDB ja TigerGraph. Kahdessa artikkelissa tietokantana oli RDF-tietokanta BlazeGraph. Perusteluina näiden graafitietokantojen valinnalle esitettiin esimerkiksi:

- ArangoDB:n suorituskyky on hyvä ja sillä on monipuolisia ominaisuuksia [Lorincz *et al.* 2020];
- BlazeGraph käsittelee isoja graafeja tehokkaasti [Bedmar *et al.* 2017];
- Dgraph mahdollistaa isojen datamäärien käsittelyn tehokkaasti, joka olisi hankalaa Neo4j:llä [Cermak & Sramkova 2021];
- OrientDB täyttää sovellusalueen vaatimukset ja sillä on kehittyneitä ominaisuuksia ja hyvä suorituskyky [D'Onofrio *et al.* 2017];
- TigerGraph tarjoaa hyvän tuen samanaikaiseen laskentaan [Miranda *et al.* 2021].

Lisäksi neljässä artikkelissa esiteltiin uusi tekijöiden kehittämä graafitietokanta: näitä olivat JasmineGraph, S2JGraph, Nepal, ja SeQuery. Näistä JasmineGraph on hajautettu graafitietokantapalvelin neuroverkkoihin liittyvää laskentaa varten [Karunarathna *et al.* 2020] ja S2JGraph graafitietokanta, joka on kehitetty suosittelujärjestelmää varten [Giabelli *et al.* 2021]. Nepal puolestaan on graafitietokanta, joka on kehitetty erityisesti *malliohjattuun tietoverkon hallintaan* (model-driven networking): Nepalissa on vahvasti tyypitetty tietokantakaavio, se mahdollistaa polkujen käytön ensimmäisen luokan kansalaisina kyselyissä ja se tukee menneisyyteen suuntautuvia kyselyitä [Jamkhedkar *et al.* 2018]. Viimeisenä SeQuery on web-pohjainen graafitietokanta, joka yhdistelee usean eri bioinformatiikan tietokannan sisältöjä ja painottaa toteutuksessaan visualisointia ja interaktiivisuutta [Hu *et al.* 2019].

4.3 Graafitietokantojen hyötyjä

Kolmantena tutkimuskysymyksenä oli selvittää mitä graafitietokantojen hyötyjä artikkeleissa mainitaan. Kaikissa artikkeleissa mainittiin joitain hyötyjä, monissa useita. Hyödyistä muodostettiin aineistolähtöisesti 10 eri luokkaa, jotka on esitetty taulukossa 4.3.

Mainittu hyöty	Frekvenssi	Osuus
Graafikyselyt ja -algoritmit	108	97,3 %
Soveltuvuus verkottuneelle datalle	105	94,6 %
Selitysvoima	87	78,4 %
Suorituskyky	74	66,7 %
Tietomallin ymmärrettävyys	50	45,0 %
Tietokantakaavion joustavuus	47	42,3 %
Visualisointi	47	42,3 %
Kyselykielen ymmärrettävyys	35	31,5 %
Soveltuvuus heterogeeniselle datalle	26	23,4 %
Rinnakkaisuus	12	10,8 %

Taulukko 4.3. Graafitietokantoihin liitettyjä hyötyjä artikkeleissa.

Graafikyselyt- ja algoritmit ovat olennainen osa graafitietokantoja. Niiden hyödyntäminen mainittiin lähes kaikissa (n = 108) artikkeleissa. Tunnetut graafiteorian algoritmit kuten lyhimmän polun laskenta, klusterointi ja yhteisöjen etsintä ovat helposti saatavilla graafitietokantoja käytettäessä [Mei *et al.* 2020; Tripathi *et al.* 2017]. Graafitietokanta ja graafitietomalli ovatkin hyvä valinta silloin kun sovellus perustuu graafin kulkujen ja siihen pohjautuvien algoritmien hyödyntämiseen [Chen *et al.* 2018]. Monessa artikkelissa mainittiin myös graafikyselykielten olevan ilmaisuvoimaisia graafien

käsittelyssä [mm. Bukhari *et al.* 2021; Hor *et al.* 2018; Shi *et al.* 2021]. Yksinkertaistettuna ilmaisuvoima tarkoittaa miten helppo jokin laskennallinen toimenpide on ilmaista tarkastellulla kielellä [Davidson & Michaelson 2018].

Toinen lähes kaikissa artikkeleissa (n = 105) mainittu hyöty on **soveltuvuus verkottuneelle datalle**. Useassa artikkeleissa kuvaillaan datan muodostavan graafin luonnostaan [mm. Barakat *et al.* 2017; Constantinov *et al.* 2018; Mezzanzanica *et al.* 2018]. Graafitietokanta mahdollistaa graafimaisen datan tallentamisen alkuperäisessä muodossaan, toisinkuin relaatiotietokanta, jossa data muunnetaan taulupohjaiseen muotoon [Bollen *et al.* 2021]. Monissa artikkeleissa myös todetaan graafitietorakenteen soveltuvan hyvin, kun datassa esiintyy paljon suhteita [mm. Maxwell *et al.* 2021; Mondal *et al.* 2020; Yuan *et al.* 2020]. Simpsonin ja Gnadin [2020] mukaan suhteiden analysointi ja hakeminen kyselyillä on graafitietokannoissa nopeaa ja intuitiivista, koska graafitietokannat esittävät linkittyneen datan solmuina, kaarina ja ominaisuuksina.

Selitysvoimalla tarkoitetaan tässä graafitietokannan kykyä tuottaa vastauksia kysymyksiin ja käyttökelpoista informaatiota. Se mainittiin suuressa osassa artikkeleita (n = 87). Selitysvoima muodostuu kahdesta tekijästä: ensinnäkin graafitietokannan mahdollistamasta eksploratiivisesta tutkimisesta ja toisaalta graafitietokannan tarjoamista laskennallisista ja algoritmisista työkaluista. Eksploratiivista tutkimista auttaa esimerkiksi se, että graafisessa käyttöliittymässä solmujen väliset suhteet voi nähdä suoraan yhdistävistä kaarista [Mei *et al.* 2020]. Chienin ja Hsiehin [2020] mukaan graafilla on luonnostaan kyky paljastaa ja selittää suhteita datan sisällä, koska se muodostuu solmuista ja kaarista. Allen ja muut [2019] lisäävät, että suhteiden analysointi graafiteorian ja verkkoanalyysin avulla paljastaa usein piilossa olevia suhteita ja rakenteita. Graafitietokantojen laskennallista selitysvoimaa hyödynnettiin useissa erilaisissa analyyseissa, mm. yhteisöjen havaitsemisessa tutkijaverkostoissa [Aung & Nyunt 2020], fyysisen kulunvalvonnan anomalioiden tunnistuksessa [Geepalla & Asharif 2020], kyberhyökkäysten simuloinnissa [Yuan *et al.* 2020] sekä tuotantotehtävien aikataulutuksessa [Zhu *et al.* 2019].

Suorituskyky mainittiin 74 artikkelissa. Sen voi jakaa teoreettiseen ja mitattuun suorituskykyyn. Teoreettisessa mielessä graafitietokantoja voi pitää relaatiotietokantoja tehokkaampana linkittyneen datan kohdalla, koska relaatiotietokannassa joudutaan tyypillisesti tekemään useita rekursiivisia taulun liitoksia itseensä, minkä suunnittelu on monimutkaista ja kyselyt ovat laskennallisesti kalliita [Simpson & Gnad 2020]. Padayachyn ja muiden [2018] mukaan graafitietokannassa kyselyjen nopeus pysyy jotakuinkin vakiona datamäärän kasvaessa, toisinkuin relaatiotietokannassa. Myös esimerkiksi graafin kulkuja laskiessa graafitietokanta on tehokkaampi, koska yhteyden voi muodostaa suoraan solmujen välille [Balaur *et al.* 2017]. Mitattu suorituskyky esiintyi myös useassa artikkelissa [mm. El Helou *et al.* 2019; Gorawski & Grochla 2020; Ngueyen

& Do 2017; Tao *et al.* 2018]: näissä tutkimuksissa graafitietokannan kyselyjen suoritusnopeus oli parempi verrattuna relaatiotietokannan kyselyihin. Toisaalta joissakin artikkelissa [mm. Mattioli & Gubitoso 2018] tietokantojen suorituskyky vaihteli kyselyittäin: osan kyselyistä suoritti nopeammin graafitietokanta, osan relaatiotietokanta.

Tietomallin ymmärrettävyys mainittiin vähän alle puolessa artikkeleista (n = 50). Lehotay-Kéry ja Kiss [2020] perustelevat, että graafirakenteet mahdollistavat heidän sovelluksessaan verkkomaisen datan (tietoliikenneverkkojen) esittämisen ihmiselle intuitiivisella tavalla. Tripathi ja muut [2017] tarkastelevat datan mallinnusta esineiden internetin kontekstissa: heidän mukaansa datan mallintaminen graafina on helppoa ja luonnollista, ja siinä on etuna, että loppukäyttäjät ymmärtävät helpommin dataa. Myös El Helou ja muut [2019] kertovat datan ja graafitietomallin välisen semanttisen vastaavuuden helpottaneen suuresti sovelluksen kehitystyötä.

Tietokantakaavion joustavuus mainittiin 47 artikkelissa. Graafitietokannat sopivat hyvin strukturoimattoman datan säilömiseen, koska niihin ei ole pakko määrittää tietokantakaaviota: relaatiotietokannoille strukturoimattoman datan käsittely olisi hankalampaa [Carnaz *et al.* 2021]. Joustavan tietokantakaavionsa ansiosta graafitietokanta myös tukee inkrementaalisesta datavaraston kehitystä, kun uusia datatyyppejä voi lisätä ajan myötä [Le May *et al.* 2020]. Myös Faralli ja muut [2020] nostavat esiin graafitietokannan hyötyinä skaalautuvuuden sekä kyvyn lisätä sovellukseen uusia datalähteitä ja uusia ominaisuuksia solmuille ja kaarille.

Visualisointi mainittiin 47 artikkelissa. Useassa artikkelissa graafitietokannan hallintajärjestelmän visualisointiominaisuuksia käytettiin suoraan [esim. Bajaj *et al.* 2018; Padayachy *et al.* 2018; Swainston *et al.* 2017], joissain taas hyödynnettiin kolmannen osapuolen työkaluja kuten Cytoscape.js-kirjastoa [Hu *et al.* 2019; Messina *et al.* 2018]. Ravikumar ja Khaperde [2017] nostivat esiin, että Neo4j pystyy tuomaan ja viemään graafidataa graafimerkkauskielen GraphML:n formaatissa, jonka he pystyivät edelleen hyödyntämään sovelluksessaan. Joissakin sovelluksissa graafin visualisointi on koko sovelluksen idea: tästä esimerkkinä on Hun ja muiden [2019] kehittämä genomiverkoston visualisointiin tarkoitettu sovellus. Visualisointi ja datan näyttäminen käyttäjille oli tärkeässä roolissa myös Messinan ja muiden [2018] sekä Bukharin ja muiden [2017] bioinformatiikan tietokannoissa sekä Swainstonin ja muiden [2017] biokemiallisen datan tietokannassa.

Kyselykielen ymmärrettävyys mainittiin 35 artikkelissa. Bollenin ja muiden [2021] mukaan graafikyselykielen kyselyt ovat yksinkertaisempia, ytimekkäämpiä ja intuitiivisempia graafien käsittelyyn kuin niiden SQL-vastineet. Monissa artikkeleissa kuvailtiin Neo4j:n graafikyselykieli Cypher helpoksi oppia [mm. Di Maro *et al.* 2017; Perçuku *et al.* 2017; Thirupathi & Padmanabhuni 2021]. Fabregat ja muut [2018] sekä Gómez ja muut [2019] lisäävät, että Cypher-kyselyiden oppiminen onnistuu suhteellisen

hyvin myös muilta kuin tietokanta-asiantuntijoilta. Cypherin omaksumista helpottaa se, että sen kyselyt muistuttavat rakenteeltaan SQL:n kyselyitä [Omasa & Inoue 2019] ja siinä käytetään ASCII-symboleita kuvaamaan graafirakenteiden välisiä yhteyksiä [Allen *et al.* 2019].

Graafitietokannan **soveltuvuus heterogeeniselle datalle** mainittiin 26 artikkelissa. Graafitietokanta pystyy yhdistämään eri lähteistä dataa yhdeksi resurssiksi [Le May *et al.* 2020; Origlia *et al.* 2021; Shoshi *et al.* 2018]. Data voi olla heterogeenista eri lähteiden lisäksi myös formaattien tai muodostumisnopeuden suhteen [Kashef *et al.* 2021]. Esimerkkinä älykaupunkisovelluksen keräämästä heterogeenisesta datasta Bellini ja Nesi [2018] kertovat, että RDF-tietokantoihin voidaan tallentaa niin staattista dataa (kuten tiestöstä ja palveluista) kuin reaaliaikaista dataa (säätilasta, liikennesensoreista, joukkoliikenteen ajoneuvoista, hätätilanteista ja tapahtumista).

Graafitietokantojen hyödyistä harvimmin mainittu oli **rinnakkaisuus**, joka esiintyi 12 artikkelissa. Graafitietokantojen kohdalla rinnakkaisuus mahdollistaa usean samanaikaisen käyttäjän toiminnan [Lukács *et al.* 2020; Quaeghebeur *et al.* 2020] ja eri säikeitä käyttävän rinnakkaislaskennan [Celesti *et al.* 2020; Yuan *et al.* 2018]. Allenin ja muiden [2019] mukaan graafitietokannat skaalautuvat hyvin ja pystyvät hyödyntämään moniytimisiä prosessoreita: niiden suorituskyky paranee lineaarisesti, kun laskentaa jaetaan useille ytimille.

4.4 Graafitietokantojen haittoja

Neljäntenä tutkimuskysymyksenä tarkasteltiin mitä haittoja graafitietokantoihin liitetään. Haittoja mainittiin artikkeleissa kohtuullisen vähän: vähintään yksi haitta mainittiin 28 artikkelissa, joten suuressa osassa artikkeleita haittoja ei mainittu lainkaan. Tulokset ovat nähtävillä taulukossa 4.4.

Mainittu haitta	Frekvenssi	Osuus
Suorituskyky	13	11,7 %
Opettelu	9	8,1 %
Soveltumattomuus tietynlaiselle datalle	5	4,5 %
Epästabiilius	2	1,8 %
Levytilan käyttö	2	1,8 %
Kypsyys	2	1,8 %
Puutteelliset turvallisuusominaisuudet	1	0,9 %
Algoritmien toteutus	1	0,9 %
Suhteiden visualisointi	1	0,9 %

Taulukko 4.4. Artikkeleissa mainittuja graafitietokantojen haittoja.

Graafitietokantojen **suorituskyky** mainittiin useimmin haitoista (n = 13), toki sekin kohtalaisen harvoin. Tämä tuli esiin esimerkiksi Dharmawanin ja Sarnon [2017] sekä

Pacacin ja muiden [2017] artikkeleissa, joissa relaatiotietokantaan kohdistuneet kyselyt suoritettiin graafitietokantaan kohdistuneita kyselyitä nopeammin. Mezzanzanican ja muiden [2018] sovelluksessa puolestaan graafitietokannan kyselyjen suoritusnopeus vaihteli voimakkaasti analysoitavien aligraafien mukaan. El Helou ja muut [2019] huomasivat, että graafitietokannassa kyselyiden suoritus kesti ensimmäisellä kertaa huomattavasti kauemmin kuin seuraavalla: he pitivät tätä ongelmana, mikäli tietokantaan kohdistuu paljon vain yhden kerran suoritettavia kyselyitä. Gorawskin ja Grochlan [2020] sovelluksessa graafitietokannan pystyttäminen kesti pitkään, mutta he pitivät sitä hyväksyttävänä, koska tyypillisesti se tehdään vain kerran.

Graafitietokantojen vaatima **opettelu** mainittiin haittana yhdeksässä artikkelissa: erityisesti graafikyselykielet vaativat opettelua. Esimerkiksi Swainstonin ja muiden [2017] mukaan Cypher-kyselykieli ei ole intuitiivinen noviisikäyttäjille: heidän biokemiallisen tietokantasovelluksensa tulevat versiot eivät tule vaatimaan käyttäjiltä kykyä kirjottaa kyselyitä suoraan Cypherilla. Myös Ismailin ja muiden [2017] sekä Horin ja muiden [2018] mukaan monimutkaisempien kyselyiden muodostaminen graafikyselykielellä vaatii sekä aihealueen että graafitietokantojen asiantuntijuutta. Cermakin ja Stramkovan [2021] mukaan haasteena on graafikyselykielen opettelun lisäksi laajemmin graafitietomallin vaatima ajattelutavan muutos, jotta graafianalyysia voisi hyödyntää tehokkaasti. Pujante ja muut [2021] nostavat esiin, että graafien visualisoinnista on suuri apu kliinisille asiantuntijoille päätöksenteossa, mutta käytännössä kyselyiden tekeminen ja graafien tulkinta voi olla heille hyvin hankalaa.

Soveltumattomuus tietynlaiselle datalle mainittiin viidessä artikkelissa. Guian ja muiden [2017] mukaan graafitietokannat eivät ole erityisen tehokkaita esimerkiksi relaatiotietomallin mukaisen datan käsittelyssä eivätkä ne täten toimi yleisesti relaatiotietokantojen korvaajina. D'Onofrio ja muut [2017] puolestaan käsittelivät artikkelissaan sumeita kognitiivisia karttoja ja totesivat, että vain 6 graafitietokantaa 29:sta täyttää näille asetetut vaatimukset. Myös Pujanten ja muiden [2021] mukaan graafitietokantojen kyky esittää sumeita suhteita ja todennäköisyyksiä kaipaisi parannusta. Cermak ja Stramkova [2021] nimesivät graafitietokantojen haitaksi puutteellisen kyvyn käsitellä aikakontekstia.

Graafitietokantojen **epästabiilius** mainittiin kahdessa artikkelissa. Guia ja muut [2017] kohtasivat ongelmana Neo4j:n kohdalla suurten aineistojen käsittelyn: järjestelmä kaatui usein suuria aineistoja ladattaessa, mikä pakotti aloittamaan uudelleen datan lataamisen. Padayachy ja muut [2018] puolestaan löysivät Neo4j:stä Cypher-kyselykielen käsittelyssä joitakin ohjelmavirheitä, jotka pakottivat muotoilemaan kyselyt uusiksi toisella tapaa ja aiheuttivat näin lisätyötä.

Graafitietokantojen **levytilan käyttö** mainittiin haittana kahdessa artikkelissa. Nguyenin ja Don [2017] kehittämä graafitietokantaan perustuva sovellus kuluttaa

enemmän levytilaa kuin relaatiotietokantaan perustuva sovellus, mutta he eivät pidä tätä ongelmana, sillä levytilan käytöllä ei ole enää nykyään niin suurta merkitystä kuin aiemmin. Mirandan ja muiden [2021] mukaan graafitietokannat saattavat luoda redundanttia dataa tallentaessaan strukturoimatonta dataa, jolloin solmun tiedot voivat tallentua useaan tiedostoon.

Graafitietokantojen alempi **kypsyys** mainittiin haittana kahdessa artikkelissa. Pacacin ja muiden [2017] mukaan relaatiotietokantoja on suosittu graafianalytiikassa, koska ne ovat vakaampia ja niitä käytetään runsaasti data-analyyttisissä ekosysteemeissä. Myös Yuanin ja muiden [2020] mukaan relaatiotietokannat ovat yleisesti ottaen graafitietokantoja kypsempiä ja täydellisempiä pitkän kehityksen ansiosta.

Yuan ja muut [2020] nostivat esiin graafitietokantojen haittana myös **puutteelliset turvallisuusominaisuudet**. Relaatiotietokannoissa turvallisuusominaisuuksia on toteutettu esimerkiksi luomalla salaisia avaimia, eritasoisia pääsyoikeuksia käyttäjille ja erilaisia tietokannan kerroksia. Yuanin ja muiden mukaan esimerkiksi lähes kaikista graafitietokannoista puuttuu mekanismi samanaikaisten käyttäjien hallintaan ja rajoittamiseen. Heidän mukaansa graafitietokantojen turvaominaisuuksia tulisikin parantaa. [Yuan *et al.* 2020]

Pujanten ja muiden [2021] artikkelissa mainittiin haittana **algoritmien toteutus**. Heidän mukaansa joissain tapauksissa tarvittaisiin kompleksisempaa graafianalyysia kuin mitä graafitietokantoihin toteutetut algoritmit pystyvät tekemään. Uusien algoritmien implementointi tietokannassa voi olla vaikeampaa kuin ohjelmakoodissa: mikäli algoritmi toteutetaan ohjelmallisesti, saattaisi graafitietokanta jäädä tarpeettomaksi. [Pujante *et al.* 2021]

Suhteiden visualisointi mainittiin haittana Cermakin ja Stramkovan [2021] artikkelissa: suuri solmumäärä aiheutti sen, että tulosgraafista tuli epäselvä. Ratkaisuna tähän he pitivät sitä, että solmut tulisi levittää visualisoinnissa tarpeeksi laajalle tai ryhmitellä samankaltaiset solmut yhteen. Lisäksi suuren solmumäärän visualisointi oli laskennallisesti raskasta. [Cermak & Stramkova 2021]

5 Pohdinta

Tässä luvussa verrataan ensin tutkimuksen tuloksia aiempaan kirjallisuuteen. Ensin tarkastellaan graafitietokantojen sovellusaloja (kohta 5.1) ja tämän jälkeen käytettyjä graafitietokantoja sekä niiden hyötyjä ja haittoja (kohta 5.2). Luvun lopussa pohditaan työn rajoitteita ja mahdollisia jatkotutkimusaiheita (kohta 5.3).

5.1 Graafitietokantojen sovellusalat

Tutkimuksen ensimmäinen tavoite oli tarkastella millä sovellusaloilla graafitietokantoja käytetään. Tähän vastauksena muodostettiin systemaattisen kirjallisuuskatsauksen (SKK) perusteella luokitus, johon kuuluu 25 eri sovellusalaa (kohta 4.1). Tätä luokitusta verrataan *aiemman kirjallisuuden* (AK) perusteella tehtyyn luokitukseen, jolla tarkoitetaan aiemman tutkimuksen ja graafitietokantojen valmistajien tietojen perusteella tehtyä sovellusalojen luokitusta (kohta 2.3). Täydellinen vertailu näistä kahdesta luokituksesta on esitetty liitteessä 3 (taulukko 7.3). Joitakin luokkia on muokattu paremman verrattavuuden aikaansaamiseksi: nämä luokat on merkattu tähdellä (*). Vertailuissa tulee huomata, että niissä verrataan kahta eri asiaa: systemaattisessa kirjallisuuskatsauksessa laskettiin yksittäisiä artikkeleita, kun taas aiemman kirjallisuuden kohdalla laskettiin mainintoja listoilla. Tässä esitetään merkityksellisemmät luokat ja suurimmat erot luokitusten välillä.

Taulukossa 5.1 on esitetty kahdeksan sovellusalaa, jotka esiintyivät säännöllisesti niin systemaattisessa kirjallisuuskatsauksessa kuin myös aiemmassa kirjallisuudessa. Näitä sovellusaloja voisi kuvailla graafitietokantojen vakiintuneiksi käyttökohteiksi. Yhteistä näille sovellusaloille on, että niissä tieto on luonteeltaan verkkomaista ja se on täten luontevasti mallinnettavissa graafiksi [Angles & Gutierrez 2018]. Newmanin [2003, s. 174] kompleksisten verkkojen jakoa käyttäen sovellukset voi luokitella ylätasolla sosiaalisiin verkostoihin, biologisiin verkkoihin, teknologisiin verkkoihin ja informaatioverkkoihin.

Sovellusala	SKK	AK	Huomioitavaa
Biotieteet	19*	10	Sisältää bioinformatiikan ja biokemian
Sosiaaliset verkostot	17*	11	Sisältää epidemiologian
Tietoverkot	11	10*	Sisältää kyberturvallisuuden
Geografinen tieto	10	8*	Sisältää kuljetukset ja liikenteen
Suosittelujärjestelmät	7	8	
Energia	7	4	
Tekoäly ja koneoppiminen	5*	5	Luokka laajennettu kattamaan tekoälyn
Tietämysgraafit	4	4	

Taulukko 5.1. Systemaattisessa kirjallisuuskatsauksessa (SKK) ja aiemmassa kirjallisuudessa (AK) esiintyneitä sovellusaloja. Tähdellä (*) merkattuja luokkia yhdistelty.

Biotieteet nousivat yleisimmäksi sovellusalaksi: se sisältää systemaattisen kirjallisuuskatsauksen luokista bioinformatiikan ja biokemian. Alaa kutsutaan tässä kattokäsitteellä biotieteet, jotta se kattaisi myös muut näille aloille läheiset tieteenalat. Graafitietokantojen suosio biotieteissä selittynee osin sillä, että luonnossa esiintyy runsaasti erilaisia mikro- ja makrotason biologisia verkkoja [Newman 2003, ss. 179–180; Timón-Reina *et al.* 2021].

Sosiaaliset verkostot olivat toiseksi yleisin sovellusala. Suosiota selittää se, että graafimainen esitystapa soveltuu hyvin ihmisten tai ihmisryhmien ja heidän välisiensä suhteiden kuvaamiseen [Larriba-Pey *et al.* 2014]. Tässä yhteydessä sosiaalisten verkostojen sovelluksiin laskettiin mukaan epidemiologian sovellukset, koska taudinjäljityksessä on myös kyse henkilöiden välisten suhteiden mallintamisesta.

Muita yleisiä graafitietokantojen sovellusaloja ovat tietoverkot, geografinen tieto ja energia-ala. Näillä aloilla graafeilla mallinnetaan teknologisia verkkoja: teknologiset verkot ovat Newmanin [2003, s. 178] mukaan teknologisia, ihmisen tekemiä verkkoja, jotka on suunniteltu yleensä jonkin hyödykkeen (kuten sähkön tai informaation) jakeluun. Tässä yhteydessä tietoverkkoihin laskettiin mukaan kyberturvallisuuden sovellukset. Geografiseen tietoon puolestaan laskettiin mukaan liikenne- ja kuljetusalan sovellukset, sillä ne perustuvat usein paikkatietojen käyttöön.

Kolme jäljellä olevaa yleistä sovellusalaa, suosittelujärjestelmät, tietämysgraafit sekä tekoäly ja koneoppiminen, edustavat Newmanin [2003, s. 176] luokituksessa informaatioverkkoja. Myös näiden alojen sovelluksille ominaista on tiedon verkkomainen olemus. Näitä sovelluksia voidaan hyödyntää kaikilla toimialoilla. Tekoälyn ja koneoppimisen alalla systemaattisen kirjallisuuskatsauksen artikkeleissa käsiteltiin koneoppimista: luokka laajennettiin kattamaan tekoälyn, koska koneoppiminen on tekoälyn osa-alue.

Kaksi sovellusalaa esiintyi systemaattisen kirjallisuuskatsauksen aineistossa, mutta ei juurikaan aiemmassa kirjallisuudessa: nämä alat olivat ohjelmistotuotanto ja esineiden internet (taulukko 5.2). Kyseiset sovellusalat ovat siis esiintyneet viime vuosina akateemisessa tutkimuksessa jossain määrin, mutta ne eivät olleet juurikaan edustettuina aiemman kirjallisuuden listauksissa.

Sovellusala	SKK	AK
Ohjelmistotuotanto	5	0
Esineiden internet	4	1

Taulukko 5.2. Erityisesti systemaattisen kirjallisuuskatsauksen (SKK) artikkeleissa painottuneet sovellusalat.

Ohjelmistotuotannon osalta graafeilla on kaksi sovelluskohdetta: ohjelmistokehitys ja ohjelmistoprojektien mallinnus. Ohjelmistokehityksessä käytetään graafeja ja puita (graafin erityistapaus) yleisesti: ohjelmakoodi voidaan jakaa atomisiin osiin abstraktiksi

syntaksipuuksi [Sadyrin *et al.* 2019] tai *jäsennyspuuksi* (parse tree) [Harsu 2012, s. 41]. Korkeammalla tasolla ohjelman rakenne voidaan kuvata ohjausvuograafina, josta selviää ohjelman kulku ja metodien kutsujärjestys [Lukacs *et al.* 2020]. Ohjelmistoprojekteissa graafeja pystytään hyödyntämään projektien mallintamisessa jakamalla ne pienempiin osiin tai tehtäviin.

Toinen lähinnä tutkimuksissa esiintynyt sovellusala oli esineiden internet. Esineiden internet on yhdenlainen verkko, jolloin graafi on luontainen esitystapa sille [Smidt *et al.* 2018]. Graafitietokanta soveltuu hyvin IoT-sovelluksissa tyypillisten kompleksisten suhteiden mallintamiseen: lisäksi graafitietokanta skaalautuu hyvin isojen datamassojen käsittelyyn [Küçükkeçeci & Yazıcı 2018].

Viimeisenä ryhmänä ovat sovellusalat, jotka painottuivat aikaisemmassa kirjallisuudessa ja graafitietokantojen valmistajien materiaaleissa, mutta joihin liittyen ei ole juurikaan tehty tutkimusta viime vuosina (taulukko 5.3). Nämä sovellusalat eivät syystä tai toisesta ole viime vuosina kiinnostaneet tutkimusaiheina niin paljoa. Graafitietokantojen valmistajien materiaaleissa korostuu markkinoinnillinen näkökulma [Sahu *et al.* 2020]: graafitietokannat pyritään esittämään potentiaaliselle ostajalle hyödyllisinä, edusti hän sitten julkista tai yksityistä puolta.

Luokka	**SKK**	**AK**	**Huomioitavaa**
Petosten tutkinta	1*	9	Jaettu Turvallisuus-sovellusalasta
Datanhallinta	0	8	
Julkishallinto	1*	6	Jaettu Turvallisuus-sovellusalasta
Rahoitus	1	6	
Markkinointi	3*	5	Sisältää palveluntuotannon
Tietojärjestelmät	0	5	

Taulukko 5.3. Erityisesti aiemmassa kirjallisuudessa (AK) painottuneet sovellusalat. Tähdellä (*) merkattuja luokkia jaettu muista luokista tai yhdistelty.

Aiemmassa kirjallisuudessa useimmin mainittu sovellusala oli petosten tutkinta: systemaattisen kirjallisuuskatsauksen tällaisia tutkimuksia oli vain yksi (koodattu alun perin *Turvallisuus*-sovellusalaan). Tianin [2023] mukaan petosten tutkinta on ehkä yleisimmin esitetty esimerkki graafitietokantojen käytöstä: erityisen kuuluisa tapaus on Panaman paperien veroparatiisikytkösten tutkinta Neo4j:n avulla [Cabra 2016].

Petosten tutkinta liittyy listatuista sovellusaloista julkishallintoon sekä rahoitukseen. Näitä oli usein käsitelty lähteissä lähellä toisiaan tai mainittu petosten tutkinnan olevan yksi tärkeä sovellus toimialalla. Nämä kuitenkin tulivat luokitteluun erillisinä sovellusaloina, koska sekä julkishallinto että rahoitus sisälsivät muitakin sovelluksia.

Markkinointi mainittiin kohtuullisen usein graafitietokantojen valmistajien materiaaleissa. Systemaattisessa kirjallisuuskatsauksessa markkinointiin liittyviä sovelluksia löytyi palveluntuotannosta.

Taulukon 5.3 sovellusaloista datanhallinta ja tietojärjestelmät sisälsivät yleisluontoisia graafitietokantojen sovelluksia. Systemaattisessa kirjallisuuskatsauksessa näitä ei esiintynyt ollenkaan, koska yksi vaatimus aineistoon sisällyttämiselle oli, että jokaisella artikkelilla on oltava jokin sovellusala.

Yhteenvetona voidaan vielä todeta, että systemaattisen kirjallisuuskatsauksen artikkelien sovellusalueilla data on tyypillisesti ymmärrettävissä Newmanin [2003, s. 174] kuvaamina kompleksisina verkkoina. Myös aiemmassa kirjallisuudessa graafitietokantojen sovellusten on usein sanottu olevan erilaisia verkkoja [mm. Barcelo *et al.* 2011; Jin *et al.* 2010; Kalyar 2015]. Merkittävin poikkeus tähän on Sahun ja muiden [2020] tutkimus, jossa graafitietokantojen *käytännön soveltajat* (practitioner) ilmoittivat usein mallintavansa graafitietomallilla myös perinteistä yritysdataa, jolle relaatiotietokantojen on todettu soveltuvan erityisen hyvin.

5.2 Graafitietokantojen käytön hyötyjä ja haittoja

Tässä kohdassa tarkastellaan ensin käytettyjä graafitietokantoja (toinen tutkimuskysymys) ja niiden valintaperusteita. Tämän jälkeen pohditaan graafitietokantojen käytön hyötyjä ja haittoja (kolmas ja neljäs tutkimuskysymys) ja verrataan löydöksiä aiempaan kirjallisuuteen. Hyödyt ja haitat käydään läpi tarkastelemalla graafitietokantojen eri osa-alueita: graafikyselykieliä, graafitietomallia, suorituskykyä sekä muita ominaisuuksia.

Käytetyt graafitietokannat

Toisena tutkimuskysymyksenä oli tunnistaa mitä graafitietokantoja on käytetty. Neo4j osoittautui graafitietokannoista ylivoimaisesti suosituimmaksi: 93 sovelluksessa 111:sta käytettiin yksinään sitä. Tämä ei ole suuri yllätys: Neo4j [2019] on ollut yleisesti saatavilla vuodesta 2010 lähtien ja se on DB-Engines-sivuston [Solid IT 2023] mukaan huomattavasti tunnetumpi kuin muut natiivit graafitietokannat. Neo4j koettiin systemaattisen kirjallisuuskatsauksen artikkeleissa esimerkiksi helppokäyttöiseksi [Aung & Nyunt 2020] ja sen laajaa algoritmikirjastoa pidetiin hyvänä [Bollen *et al.* 2021].

Suosiostaan huolimatta Neo4j ei ole aina optimaalinen ratkaisu graafitietokannaksi. Esimerkiksi Cermak ja Sramkova [2021] päätyivät käyttämään Dgraphia, koska heidän mukaansa isojen datamäärien prosessointi ja tallentaminen on sillä helpompaa kuin Neo4j:llä. D'Onofrion ja muiden [2017] vertailussa puolestaan OrientDB täytti parhaiten vaatimukset heidän tutkimilleen sumeille kognitiivisille kartoille. Joissain artikkeleissa päädyttiin myös kehittämään kokonaan uusi, sovelluskohteen tarpeisiin räätälöity graafitietokanta. Näissä tietokannoissa voi olla ominaisuuksia, joita muissa ei ole: esimerkiksi Jamkhedkarin ja muiden [2018] Nepal-graafitietokanta mahdollistaa

menneisyyteen suuntautuvat kyselyt. Tämä vastaa siihen ongelmaan, että aikakontekstin huomioiminen voi olla hankalaa graafitietokannoissa [Cermak & Sramkova 2021].

Graafikyselykielten käyttäminen

Kaikista yleisin systemaattisen kirjallisuuskatsauksen artikkeleissa mainittu graafitietokantojen hyöty oli graafikyselyiden ja -algoritmien hyödyntäminen. Tämä on linjassa aiemman kirjallisuuden kanssa: graafikyselykielet on optimoitu nimenomaan graafirakenteiden käsittelyyn, ja ne mahdollistavat kyselyjen esittämisen korkealla abstraktiotasolla [Angles & Gutierrez 2008]. Systemaattisen kirjallisuuskatsauksen artikkeleissa graafikyselykieliä kuvaillaan ilmaisuvoimaisiksi ja ne mahdollistavat kyselyiden ilmaisen tiiviissä muodossa [Bukhari *et al.* 2021; Hor *et al.* 2018]. Lisäksi niissä nostettiin esiin, että graafitietokantoihin ja graafikyselykieliin voidaan myös toteuttaa erilaisia graafiteorian algoritmeja, jolloin niiden käyttö on varsin helppoa käyttäjälle [Mei *et al.* 2020; Tripathi *et al.* 2017].

Osassa systemaattisen kirjallisuuskatsauksen artikkeleissa mainittiin myös hyötynä graafikyselykielten käytön helppous [Bollen *et al.* 2021; Di Maro *et al.* 2017]. Lisäksi mainittiin, että myös muiden kuin tietokanta-asiantuntijoiden on suhteellisen helppo käyttää graafikyselykieliä [Fabregat *et al.* 2018; Gómez *et al.* 2019]. Toisaalta osassa artikkeleista tuli esiin, että käytännössä monimutkaisempien graafikyselyjen vaatii syvällistä tietokantaosaamista, eikä se täten luultavasti onnistu noviisikäyttäjiltä [Hor *et al.* 2018; Ismail *et al.* 2017; Swainston *et al.* 2017]. Myös nämä löydökset ovat linjassa aiemman kirjallisuuden kanssa: graafikyselykielten käytön helppous mainitaan monesti [Fernandes & Bernardino 2018; Holzschuher & Peinl 2016], mutta myös niiden käytön haasteita on käsitelty [Bhowmick *et al.* 2017; Sahu *et al.* 2020]. Erityinen haaste graafikyselykielien käytössä on yleisen standardin puute (muuten kuin RDF:n osalta) [Sahu *et al.* 2020]. Ratkaisuna graafikyselykielten käytön haasteisiin on esitetty kyselykielten standardointia [JCC Consulting 2022] ja erilaisia visuaalisia ratkaisuja kyselyjen muodostamiseen [Hogenboom *et al.* 2010; Lay 2022; Pabón *et al.* 2019].

Graafitietomallin hyödyt

Lähes kaikissa systemaattisen kirjallisuuskatsauksen artikkeleissa mainittiin graafitietokantojen ja graafitietomallin hyötynä soveltuvuuden graafimaisen datan mallintamiseen. Monessa sovelluksessa [Barakat *et al.* 2017; Constantinov *et al.* 2018] data muodosti graafin luonnostaan, jolloin graafitietokanta mahdollistaa datan tallentamisen lähempänä sen luonnollista olemusta [Bollen *et al.* 2021]. Osassa artikkeleista todettiin graafitietokannan soveltuvan hyvin sellaisen datan tallentamiseen, jossa esiintyy paljon entiteettien välisiä suhteita [Maxwell *et al.* 2021; Mondal *et al.* 2020]. Lisäksi graafitietomallia pidettiin monessa artikkelissa ihmiselle helposti ymmärrettävänä [Tripathi *et al.* 2017] ja intuitiivisena [Lehotay-Kéry & Kiss 2020]. Nämä

löydökset ovat hyvin linjassa aiemman kirjallisuuden kanssa: graafitietomallin on kuvailtu sopivan linkittyneen datan [Angles & Gutierrez 2008] ja kompleksisten verkkojen [Hurlburt *et al.* 2017] mallintamiseen. Lisäksi esimerkiksi Patil ja muut [2014] pitivät graafitietomallia intuitiivisena ja ilmaisuvoimaisena.

Graafitietokannan kaavion joustavuus mainittiin myös monissa systemaattisen kirjallisuuskatsauksen artikkeleissa hyötynä. Joustava tietokantakaavio mahdollistaa strukturoimattoman datan käsittelyn [Carnaz *et al.* 2021] ja graafitietokannan kehityksen inkrementaalisesti [Le May *et al.* 2020]. Joustavan kaavionsa ansiosta graafitietokannat pystyvät yhdistämään dataa heterogeenisesta lähteistä [Origlia *et al.* 2021]: useammassa systemaattisen kirjallisuuskatsauksen tunnistettiin graafitietokantojen soveltuvuus heterogeenisen datan käsittelyyn [Kashef *et al.* 2021; Shoshi *et al.* 2018]. Myös aiemmassa kirjallisuudessa on tunnistettu joustavan tietokantakaavion hyödyt strukturoimattoman datan käsittelyssä [Angles & Gutierrez 2018] sekä tietokannan asteittaisessa kehittämisessä [Patil *et al.* 2014]. Graafitietokantoja on myös pidetty arvokkaana työkaluna heterogeenisen datan käsittelyssä [Timón-Reina *et al.* 2021].

Graafitietokantojen suorituskyky

Aiemmassa kirjallisuudessa graafitietokantoja on pidetty tehokkaina graafikyselyiden suorittamisessa: graafikyselyissä tarvitsee usein kulkea läpi vain osa graafista [Patil *et al.* 2014] ja kyselyjen suoritusaika ei juurikaan kasva datan kasvaessa [Fernandes & Bernardino 2018]. Joissain systemaattisessa kirjallisuuskatsauksen artikkeleista graafitietokantoja verrattiin relaatiotietokantoihin: teoreettisessa mielessä graafitietokanta on nopea graafien käsittelyssä, koska siinä ei tarvita relaatiotietokantojen tavoin rekursiivisia taulujen liitoksia itseensä [Padayachy *et al.* 2018; Simpson & Gnad 2020].

Systemaattisen kirjallisuuskatsauksen kokeellisissa tutkimuksissa sen sijaan tulokset vaihtelevat: osassa tutkimuksista graafitietokanta oli nopeampi [El Helou *et al.* 2019; Gorawski & Grochla 2020], osassa relaatiotietokanta [Dharmawan & Sarno 2017; Pacaci *et al.* 2017] ja osassa tämä vaihteli tilannekohtaisesti [Mattioli & Gubitoso 2018]. Vastaava tilanne on aiemmassa kirjallisuudessa: eri tutkimuksissa niin graafi- kuin relaatiotietokannat ovat olleet vertailuissa nopeampia. Esimerkiksi Kotirannan ja muiden [2022] tutkimuksessa relaatiotietokanta oli nopeampi: he nostivat esiin sen, että relaatiotietokantojen kyky käsitellä graafimaisia kyselyitä on parantunut huomattavasti viime vuosina. Sahu ja muut [2020] nostavat esiin suorituskykyongelmia aiheuttava tekijänä sen, että käytännössä graafit saattavat olla valtavankokoisia.

Lisäksi suorituskykyyn liittyy rinnakkaisuus ja samanaikaisuus: graafitietokantojen tuki rinnakkaisuudelle ja samanaikaisuudelle tuli esiin joissakin systemaattisen kirjallisuuskatsauksen artikkeleissa [Allen *et al.* 2019; Celesti *et al.* 2020; Lukács *et al.* 2020]. Aiemmassa kirjallisuudessa Zhou ja muut [2018] totesivat, että graafitietokantojen sanotaan usein tukevan rinnakkaisuutta, mutta käytännössä suorituskyvyn kanssa voi

esiintyä ongelmia. Suuri osa graafitietokannoista tukee usean kyselyn samanaikaista suorittamista, mutta melko harva saman kyselyn rinnakkaista suorittamista esimerkiksi usealla prosessorilla [Besta *et al.* 2019].

Graafitietokantojen muut ominaisuudet

Aiemmassa kirjallisuudessa on tunnistettu, että visualisointi on usein graafitietokantojen käyttäjille hyvin tärkeää esimerkiksi datan tutkimisessa ja esitysvälineenä [Sahu *et al.* 2020; Tian 2023]. Myös systemaattisen kirjallisuuskatsauksen artikkeleissa mainittiin usein visualisointi graafitietokantojen hyötynä. Monissa sovelluksissa hyödynnettiin graafitietokannan omia visualisointiominaisuuksia [Bajaj *et al.* 2018; Padayachy *et al.* 2018]: joissain puolestaan kolmannen osapuolen työkaluja [Hu *et al.* 2019; Messina *et al.* 2018]. Visualisointi helpottaa graafien eksploratiivista tutkimista: visuaalisesta graafista on helppo tutkia solmuja ja niihin liittyviä suhteita [Mei *et al.* 2020] ja näin saattaa paljastua uutta tietoa [Chien & Hsieh 2020]. Visualisoinnin haasteita käsiteltiin Cermankin ja Stramkovan [2021] artikkelissa: suuri määrä solmuja tuotti epäselvän esityksen ja oli laskennallisesti raskasta. Myös aiemmassa kirjallisuudessa visualisointiin liitettiin ongelmia: käyttäjät eivät välttämättä pysty muodostamaan graafeja haluamallaan tavalla ja isojen graafien visualisointi voi olla hankalaa [Sahu *et al.* 2020]

Turvallisuuskysymykset nousivat esiin graafitietokantojen potentiaalisena ongelmana Yuanin ja muiden [2020] artikkelissa: heidän mukaansa monista graafitietokannoista puuttuvat eritasoiset käyttöoikeudet, tietokannan kerrokset sekä samanaikaisten käyttäjien hallintaominaisuus. Myös aiemmassa kirjallisuudessa on nostettu esiin turvallisuushuolia graafitietokantoihin liittyen: Bozan ja Muñozin [2017] internetistä löytämillä graafitietokantapalvelimilla oli usein puutteellisia konfiguraatioita, jotka altistivat kannat hyökkäyksille. Hurlburt [2015] puolestaan esitti turvallisuushuolena yksityisyydensuojan, mikäli graafitietokannassa olevaa sensitiivistä tietoa ja tietokannan entiteettien identiteettejä ei ole salattu asianmukaisesti.

Muita systemaattisen kirjallisuuskatsauksen artikkeleissa harvoin mainittuja graafitietokantojen haittoja olivat matalampi kypsyys relaatiotietokantoihin verrattuna [Pacaci *et al.* 2020; Yuan *et al.* 2020], levytilan käyttö [Miranda *et al.* 2021; Nguyen & Do 2017], epästabiilius [Guia *et al.* 2017; Padayachy *et al.* 2018] ja uusien algoritmien toteutuksen vaikeus graafitietokannassa [Pujante *et al.* 2021]. Aiemmassa kirjallisuudessa graafitietokantojen kypsyyden on todettu vaihtelevan suuresti eri graafitietokantojen välillä [Angles & Gutierrez 2018]. Epästabiilius on myös liitetty kypsyyteen: mitä enemmän järjestelmää on testattu, sitä vähemmän siinä on virheitä ja sitä kypsempi se on [Batra & Tyagi 2012].

5.3 Työn rajoitteet ja jatkotutkimusaiheet

Työhön liittyy niin systemaattista kirjallisuuskatsausta kuin laadullista tutkimusta koskevia rajoitteita. Systemaattisen kirjallisuuskatsauksen rajoitteet koskevat aineiston otantaa, seulontaa ja analyysia. Aineiston analyysissa korostuu rajoitteena subjektiivisuus, joka on myös samalla laadullisen tutkimuksen ominaispiirre.

Systemaattisen kirjallisuuskatsauksen otantaan liittyy mahdollista vinoumaa. Ensinnäkin hakutietokannaksi valittiin Scopus, jonka tiedettiin sisältävän laajasti vaikutusvaltaisia tietojenkäsittelytieteen julkaisuja (erityisesti ACM:n ja IEEE:n tietokannat). Tästä huolimatta on luultavaa, että joitakin relevantteja, Scopusin viitteisiin kuulumattomia artikkeleita on jäänyt aineiston ulkopuolelle.

Toisena otantaan liittyvänä rajoitteena on hakutermien käyttö. Hakulauseke ei välttämättä kata kaikkia sovelluksia. Lisäksi hakutulosten runsaan määrän vuoksi hakulauseke rajattiin koskemaan vain artikkelin otsikkoa. Jos mukaan olisi otettu tämän lisäksi osumat artikkelien avainsanoista ja tiivistelmistä, olisi artikkeleita tullut huomattavasti enemmän. Suurin osa näistä artikkeleista vaikutti kuitenkin testihakujen perusteella tutkimuksen tarkoitukseen sopimattomilta.

Aineiston seulontaan liittyy useampia rajoitteita. Ensinnäkin aineisto rajattiin kattamaan hakuhetkestä viisi edellistä vuotta (2017–2021), koska mukaan haluttiin vain uudehkoja tutkimuksia. Toinen merkittävä seulontaan liittyvä rajoite ovat artikkelien sivumäärät: alimmaksi hyväksyttäväksi pituudeksi asetettiin 6 sivua. Tämä rajasi pois 69 artikkelia, jotka olivat tyypillisesti 4–5 sivun pituisia konferenssiartikkeleita. Rajausta perusteltiin sillä, että mukaan haluttiin artikkeleita, joissa on ollut mahdollisuus tarkastella monipuolisemmin graafitietokantasovellusta: erityisesti hyötyjen ja haittojen tunnistaminen olisi ollut hankalampaa lyhyistä artikkeleista.

Kolmas seulontaan liittyvä rajoite on artikkelien saatavuus. Aineistoon sisällytettiin vain sellaiset artikkelit, jotka sisältyivät yliopiston tilaukseen tai joista oli saatavilla *open access* -versio. Ei saatavilla olevia artikkeleita oli kuitenkin kohtuullisen vähän, 16 kappaletta, joten tämä rajoite ei ole kovin merkittävä.

Neljäs seulontaan liittyvä rajoite on artikkelien sisällön arvioiminen. Sisällöllisistä kriteereistä kaksi tuotti eniten tulkinnanvaraisia tilanteita: artikkeleissa tuli olla jokin sovellusala määriteltynä ja niissä tuli esitellä jonkinlainen toteutettu sovellus. Sovellusalaa määrittäessä piti tulkita esimerkiksi, onko artikkelissa vain käytetty jonkin sovellusalan dataa esimerkkinä ilman muuta kytköstä sovellusalaan. Sovelluksen toteuttamista arvioidessa jouduttiin miettimään esimerkiksi voiko sovellukseksi laskea sen, että hyvin suppea aineisto on ladattu testimielessä graafitietokantaan.

Aineiston analyysissa käytettiin koodausta ja laadullista sisällönanalyysia: näihin liittyy tyypillisiä laadulliseen tutkimuksen rajoitteita. Laadullisen tutkimuksen yksi ominaispiirre on subjektiivisuus: tämä tarkoittaa niin tutkittavien kohteiden (artikkelien)

subjektiivisuutta kuin tutkijan subjektiivisuutta aineiston tulkinnassa [Juhila 2021b]. Tässä työssä tutkittavien kohteiden subjektiivisuus näkyy siinä, että artikkelien sisältö hyväksyttiin sellaisenaan aineistoksi: artikkeleissa olevat väittämät ja toteamukset edustavat kirjoittajien näkemyksiä graafitietokantojen ominaisuuksista. Tutkijan subjektiivisuus puolestaan näkyy siinä, että koodaus ja luokittelu on syntynyt tutkijan sen hetkisen tietämyksen perusteella. On epätodennäköistä, että toinen tutkija päätyisi täysin samanlaiseen luokitukseen: luultavasti kuitenkin samankaltaisia teemoja nousisi esiin.

Aineistoon liittyvänä rajoitteena voidaan mainita myös *julkaisuharha* (publication bias). Julkaisuharha tarkoittaa, että akateeminen tutkimus päätyy julkaistuksi todennäköisemmin, jos sen tulokset ovat positiivisia kuin jos ne olisivat negatiivisia [Kitchenham & Charters 2007, s. 15]. Julkaisuharha liitetään tyypillisesti määrälliseen tutkimukseen, mutta se koskee yhtä lailla myös laadullista tutkimusta [Petticrew *et al.* 2008]. Tässä tutkimuksessa viitteitä julkaisuharhasta saatiin siinä, että graafitietokantojen hyötyjä tunnistettiin huomattavasti enemmän kuin haittoja: monissa artikkeleissa ei mainittu haittoja ollenkaan. Tätä voi toki selittää myös se, että artikkelien ensisijaisena tavoitteena on yleensä ollut raportoida toteutettu sovellus eikä pohtia mahdollisia tietokantavalintaan liittyviä ongelmia.

Jatkotutkimusaiheet

Tämän työn tutkimuskysymykset ovat melko laajoja: jatkotutkimuksessa voitaisiin keskittyä tarkemmin johonkin tiettyyn graafitietokantojen osa-alueeseen. Tarkemmin rajatussa tutkimuksessa voisi päästä syvemmälle kyseisen aiheen käsittelyyn. Tutkimusotetta voisi myös halutessaan viedä määrällisempään suuntaan.

Yhtenä jatkotutkimusaiheena olisi tutkia laajemmin muita graafitietokantojen ominaisuuksia kuin suorituskykyä. Tietokantoja arvioidaan monesti suorituskyvyn kautta, vaikka olisi tarpeen tarkastella niitä myös muista näkökulmista [Lourenço *et al.* 2015; Tian 2023]. Tässä tutkimuksessa hyötyinä nousivat esiin esimerkiksi graafitietomallin intuitiivisuus ihmiskäyttäjälle ja graafien visualisointi hyödyllisenä työkaluna. Olisi kiinnostavaa tietää enemmän siitä, millaisissa tilanteissa näiden ominaisuuksien merkitys korostuu.

Jatkotutkimusaiheena voidaan esittää myös graafitietokantojen tarkasteleminen kriittisemmin: tällä vastattaisiin osittain potentiaaliseen julkaisuharhaan. Merkittävin vastaan tullut kriittinen tutkimus oli Sahun ja muiden [2020] artikkeli, jossa tuli esiin, että erittäin suuret graafit ovat yleisiä sovellusalasta riippumatta ja niiden käsittelyyn graafitietokannoissa liittyy usein hankaluuksia. Lisäksi heidän löydöksenään oli, että graafitietokantoja käytetään usein relaatiotietokannoille sopivissa sovelluskohteissa, mille ei löydy tukea akateemisesta tutkimuksesta. Jatkotutkimuksessa olisi hyödyllistä selvittää laajemmin graafitietokantojen todellisia käyttökohteita sekä millaisia hyötyjä ja haasteita niiden käyttäjät kokevat. [Sahu *et al.* 2020]

6 Yhteenveto

Tämän tutkimuksen tavoitteena oli kartoittaa akateemisessa tutkimuksessa esiintyviä graafitietokantojen sovellusaloja sekä graafitietokantoihin liitettyjä hyötyjä ja haittoja. Lisäksi tavoitteena oli kartoittaa mitä graafitietokantoja sovelluksissa on käytetty.

Tutkimusmenetelmänä oli systemaattinen kirjallisuuskatsaus, jossa tunnistettiin 111 kriteerit täyttävää artikkelia vuosilta 2017–2021. Artikkeleja analysoitiin laadullisen sisällönanalyysin keinoin: koodaus ja luokittelu olivat sen keskeisiä työkaluja. Luokittelussa jokaiselle artikkelille määritettiin sovellusala ja käytetty graafitietokanta: lisäksi artikkeleista kartoitettiin niissä mainittuja graafitietokantojen hyötyjä ja haittoja. Tutkimustulosten kuvailussa tukena käytettiin aineiston kvantifiointia.

Graafitietokantojen sovellusaloja tunnistettiin 25: sovellusaloissa korostuivat sellaiset alat, joissa tieto on mallinnettavissa kompleksisina verkkoina. Suosituimpia aloja olivat bioinformatiikka, sosiaaliset verkostot, tietoverkot ja geografinen tieto. Näitä hieman vähemmän sovelluksia oli energia-alalla, suosittelujärjestelmissä, koneoppimisessa ja ohjelmistotuotannossa.

Graafitietokannoista ylivoimaisesti käytetyin oli Neo4j: se oli käytössä valtaosassa artikkelien sovelluksista. Neo4j:n lisäksi tunnistettiin yhdeksän käytettyä yksittäistä graafitietokantaa, joista neljä oli artikkelin kirjoittajien itse kehittämiä. Lisäksi kuudessa artikkelissa käytettiin useampaa kuin yhtä graafitietokantaa.

Graafitietokantojen käytölle tunnistettiin kymmenen hyötyä. Yleisimmin mainitut hyödyt olivat graafikyselyiden ja -algoritmien hyödyntäminen sekä graafitietokantojen soveltuvuus verkottuneelle datalle: nämä mainittiin lähes kaikissa artikkeleissa. Hieman näitä harvemmin mainittuja hyötyjä olivat graafitietokantojen selitysvoima erilaisissa analyyseissa sekä niiden suorituskyky. Noin puolessa artikkeleista mainittiin hyötyinä graafitietokantojen visualisointiominaisuudet, tietokantakaavion joustavuus ja graafitietomallin ymmärrettävyys. Harvimmin mainittuja hyötyjä olivat graafikyselykielten ymmärrettävyys, graafitietomallin soveltuvuus heterogeeniselle datalle sekä tuki rinnakkaisuudelle.

Graafitietokantojen eri haittoja tunnistettiin yhdeksän: niitä mainittiin kuitenkin huomattavasti hyötyjä harvemmin. Yleisimmin mainitut haitat olivat suorituskyky ja graafitietokantojen opettelu: molemmat oli mainittu usein myös hyötynä. Näitä selittäviä seikkoja ovat, että graafitietokantojen suorituskyvyssä on eroja eri sovellusten välillä, graafitietokantojen ja -kyselykielien koettu vaikeustaso taas riippuu tutkijoiden näkemyksistä. Muita harvemmin mainittuja haittoja olivat muun muassa graafitietokantojen soveltumattomuus tietynlaiselle datalle, epästabiilius, alempi kypsyysaste sekä puutteelliset turvallisuusominaisuudet.

Löydökset ovat pitkälti linjassa aiemman tutkimuksen kanssa hyötyjen ja haittojen osalta: isoimmat erot olivat sovellusaloissa. Aiemmassa kirjallisuudessa mainittiin usein

etenkin petosten tutkinta sovelluskohteena, mutta tässä tutkimuksessa se ei noussut erityisesti esiin. Toisaalta tämän tutkimuksen tuloksissa ohjelmistotuotanto ja esineiden internet esiintyivät joidenkin artikkelien sovellusaloina, kun taas aiemmassa kirjallisuudessa ne olivat saaneet vain vähän mainintoja.

Tutkimuksen merkittävimpinä rajoitteina ovat otoksen rajallisuus ja laadulliselle tutkimukselle tyypillinen aineiston analyysin subjektiivisuus. Jatkotutkimusideana voidaan esittää graafitietokantojen sovellusalojen sekä koettujen hyötyjen ja haittojen tutkiminen käytännön soveltajien näkökulmasta. Graafitietokantojen arviointi on usein keskittynyt aiemmassa tutkimuksessa suorituskykyyn: olisi hyödyllistä arvioida graafitietokantoja myös muista näkökulmista.

7 Viiteluettelo

7.1 Kirjallisuus

Amarasinghe, S., Chlipala, A., Devadas, S., Ernst, M., Goldman, M., Guttag, J., Jackson, D., Miller, R., Rinard, M. & Solar-Lezama, A. (2014). *Concurrency.* [Opetusmateriaali]. Massachusetts Institute of Technology. Viitattu 3.3.2023. https://web.mit.edu/6.005/www/fa14/classes/17-concurrency/

Amazon.com (27.10.2022). *AWS announces Amazon Neptune Serverless.* [Lehdistötiedote]. Viitattu 8.3.2023. https://press.aboutamazon.com/2022/10/aws-announces-amazon-neptune-serverless

Amazon Web Services (2023a). *Amazon Neptune.* [Verkkosivu]. Viitattu 3.3.2023. https://aws.amazon.com/neptune/

Amazon Web Services (2023b). *Neptune graph data model.* [Tekninen dokumentaatio]. Viitattu 9.3.2023. https://docs.aws.amazon.com/neptune/latest/userguide/feature-overview-data-model.html

Angles, R. (2018). The property graph database model. *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*.

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J. & Vrgoč, D. (2017). Foundations of modern query languages for graph databases. *ACM Computing Surveys,* 50(5): 68.

Angles, R. & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, 40(1): 1.

Angles, R. & Gutierrez, C. (2018). An introduction to graph data management. Teoksessa Fletcher, G., Hidders, J. & Larriba-Pey, J. L. (toim.), *Graph Data Management*, 1–32. Springer Cham.

ArangoDB (2023). *ArangoDB use cases.* [Tekninen dokumentaatio]. Viitattu 3.3.2023. https://www.arangodb.com/docs/stable/use-cases.html#arangodb-as-a-graph-database

Barceló, P., Libkin, L. & Reutter, J. (2011). Querying graph patterns. *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 199–210.

Batra, S., & Tyagi, C. (2012). Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering*, 2(2), 509–512.

Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G. & Hoefler, T. (2019). *Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries.* arXiv:1910.09017

Bhowmick, S. S., Choi, B. & Li, C. (2017). Graph querying meets HCI: State of the art and future directions. *Proceedings of the 2017 ACM International Conference on Management of Data*, 1731–1736.

Boza, M. H. & Muñoz, A. (2017). (In)security in graph databases – Analysis and data leaks. *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017),* 4, 303–310

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M. & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571–583.

Cabra, M. (12.5.2016). *How the ICIJ used Neo4j to unravel the Panama Papers.* [Blogikirjoitus]. Viitattu 5.3.2023. https://neo4j.com/blog/icij-neo4j-unravel-panama-papers/

Chamberlin, D. D. & Boyce, R. F. (1974). SEQUEL: A structured English query language. *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control,* 249–264.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377–387.

Codd, E. F. (1980). Data models in database management. *Proceedings of the 1980 Workshop on Data Abstraction, Databases and Conceptual Modeling*, 112–114.

Davidson, J. & Michaelson, G. (2018). Expressiveness, meanings and machines. *Computability,* 7(4), 367–394.

Davies, R. (2015). *Internet of Things: Opportunities and challenges*. [Raportti]. European Parliamentary Research Service (EPRS).

Davoudian, A., Chen, L. & Liu, M. (2017). A survey on NoSQL stores. *ACM Computing Surveys*, 51(2): 40.

Dgraph Labs (2022). *Dgraph database overview.* [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://dgraph.io/docs/dgraph-overview/

Dias, A. H., Correia, L. H. & Malheiros, N. (2021). A systematic literature review on virtual machine consolidation. *ACM Computing Surveys*, 54(8): 176.

Do, T.-T.-T., Mai-Hoang, T.-B., Nguyen, V.-Q. & Huynh, Q.-T. (2022). Query-based performance comparison of graph database and relational database. *Proceedings of the 11th International Symposium on Information and Communication Technology,* 375–381.

Ehrlinger, L. & Wöß, W. (2016). Towards a definition of knowledge graphs. *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16).*

Elmasri, R. & Navathe, S. B. (2004). *Fundamentals of Database Systems.* Pearson. Addison-Wesley.

Elsevier (2023). *Scopus source list: Scopus sources october 2022.* [Excel-taulukko]. Viitattu 16.2.2023. *https://www.elsevier.com/__data/assets/excel_doc/0015/91122/extlistJanuary2023.xlsx*

Fernandes, D. & Bernardino, J. (2018). Graph databases comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. *Proceedings of the 7th International Conference on Data Science, Technology and Applications*, 373–380.

Fink, A. (2014). *Conducting Research Literature Reviews: From the Internet to the Paper.* Sage Publications, Inc.

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P. & Taylor, A. (2018). Cypher: An evolving query language for property graphs. *Proceedings of the 2018 International Conference on Management of Data,* 1433–1445.

Gartner (2023). *Master Data Management (MDM).* [Verkkosivu]. Viitattu 5.2.2023. https://www.gartner.com/en/information-technology/glossary/master-data-management-mdm

Grossman, D. & Anderson, R. E. (2012). Introducing parallelism and concurrency in the data structures course. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 505–510.

Gyssens, M., Paredaens, J. & Van Gucht, D. (1990). A graph-oriented object database model. *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems,* 417–424.

Harsu, M. 2012. *Ohjelmointikielet: Periaatteet, käsitteet, valintaperusteet.*

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., De Melo, G., Gutierrez, C., Kirrane, S., Gayo, J., Navigli, R., Neumaier, S., Ngomo A., Polleres, A., Rashid, S., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S. & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys, 54*(4): 71.

Hogenboom, F., Milea, V., Frasincar, F. & Kaymak, U. (2010). RDF-GL: a SPARQL-based graphical query language for RDF. Teoksessa Chbeir, R., Badr, Y., Abraham, A. & Hassanien A. (toim.), *Emergent Web Intelligence: Advanced Information Retrieval*, 87–116.

Holzschuher, F. & Peinl, R. (2016). Querying a graph database – language selection and performance considerations. *Journal of Computer and System Sciences*, 82(1), 45–68.

Horwitz, L. (2015). *360-degree customer view.* [Verkkosivu]. Viitattu 5.3.2023. https://www.techtarget.com/searchcustomerexperience/definition/360-degree-customer-view

Hurlburt, G. F. (2015). High tech, high sec.: Security concerns in graph databases. *IT Professional*, 17(1), 58–61.

Hurlburt, G. F., Thiruvathukal, G. K. & Lee, M. R. (2017). The graph database: jack of all trades or just not SQL? *IT Professional*, 19(6), 21–25.

JCC Consulting (2022). *Graph query language GQL.* [Verkkosivu]. Viitattu 8.3.2023. https://www.gqlstandards.org/

Jin, C., Bhowmick, S. S., Xiao, X., Cheng, J. & Choi, B. (2010). GBLENDER: towards blending visual query formulation and query processing in graph databases. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data,* 111–122.

Juhila, K. (2021a). Koodaaminen. Teoksessa Vuori, J. (toim.), *Laadullisen tutkimuksen verkkokäsikirja.* Yhteiskuntatieteellinen tietoarkisto. Viitattu 18.2.2023. https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/analyysitavan-valinta-ja-yleiset-analyysitavat/koodaaminen/

Juhila, K. (2021b). Laadullisen tutkimuksen ominaispiirteet. Teoksessa Vuori, J. (toim.), *Laadullisen tutkimuksen verkkokäsikirja.* Yhteiskuntatieteellinen tietoarkisto. Viitattu 25.3.2023. https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/mita-on-laadullinen-tutkimus/laadullisen-tutkimuksen-ominaispiirteet/

Kaliyar, K. R. (2015). Graph databases: A survey. *International Conference on Computing, Communication and Automation (ICCCA2015)*, 785–790.

Kallava, J. (2018). *Relaatiotietokannasta graafitietokantaan – Graafitietokannan edut tietojärjestelmän tietovarastona*. [Pro gradu -tutkielma, Tampereen yliopisto]. Trepo-julkaisuarkisto. *https://urn.fi/URN:NBN:fi:uta-201808102359*

Kaplan, A. & Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62(1), 15–25.

Khan, W., Ahmed, E. & Shahzad, W. (2017). Predictive performance comparison analysis of relational & NoSQL graph databases. *International Journal of Advanced Computer Science and Applications*, 8(5), 523–530.

Kitchenham, B. & Charters, S. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* [Tekninen raportti].

Kotiranta, P., Junkkari, M. & Nummenmaa, J. (2022). Performance of graph and relational databases in complex queries. *Applied Sciences*, 12(13): 6490.

Kunii, H. S. (1987). DBMS with graph data model for knowledge handling. *Proceedings of the 1987 Fall Joint Computer Conference on Exploring Technology: Today and Tomorrow*, 138–142.

Kuper, G. M. & Vardi, M. Y. (1984). A new approach to database logic. *Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems,* 86–96.

Larriba-Pey, J. L., Martínez-Bazán, N. & Domínguez-Sal, D. (2014). Introduction to graph databases. Teoksessa Koubarakis, M., Stamou, G., Horrocks, I., Kolaitis, P., Lausen, G. & Weikum, G. (toim.), *Reasoning Web. Reasoning on the Web in the Big Data Era*, 171–194.

Lay, C.-H. (2022). *Query-by-example graafitietokannassa.* [Pro gradu -tutkielma, Tampereen yliopisto]. Trepo-julkaisuarkisto. https://urn.fi/URN:NBN:fi:tuni-202205114759

Liu, Y., Qu, S. & Fan, B. (2021). Current status and application analysis of graph database technology. *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 735–744.

Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M. & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, 2(1): 18.

Luscombe, N. M., Greenbaum, D. & Gerstein, M. (2001). What is bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine*, 40(4), 346–358.

Mainguenaud, M. (1995). Modelling the network component of geographical information systems. *International Journal of Geographical Information Systems*, *9*(6), 575–593.

MariaDB Foundation (2023). *MariaDB in brief.* [Verkkosivu]. Viitattu 28.3.2023. https://mariadb.org/en/

McCarthy, J. (1999). *What is AI?* [Verkkosivu]. Viitattu 5.3.2023. http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html

Merriam-Webster (2023a). Life science. Teoksessa *Merriam-Webster.com Dictionary*. Viitattu 5.3.2023. https://www.merriam-webster.com/dictionary/life%20science

Merriam-Webster (2023b). Social network. Teoksessa *Merriam-Webster.com Dictionary*. Viitattu 5.3.2023. https://www.merriam-webster.com/dictionary/social%20network

Mhedhbi, A., Lissandrini, M., Kuiper, L., Waudby, J. & Szárnyas, G. (2021). LSQB: a large-scale subgraph query benchmark. *Proceedings of the 4th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA).*

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

Neo4j (21.10.2015). *Neo4j opens up its graph query language, Cypher, with support from leading companies*. [Lehdistötiedote]. Viitattu 8.3.2023. https://neo4j.com/press-releases/graph-query-language/

Neo4j (31.12.2019). *A decade of graphs: Neo4j's top 10 biggest moments of the 2010s.* [Blogikirjoitus]. Viitattu 8.3.2023. https://neo4j.com/blog/neo4j-top-10-biggest-moments-of-2010s/

Neo4j (2023a). *Clauses.* [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://neo4j.com/docs/cypher-manual/current/clauses/

Neo4j (2023b). *Graph algorithms.* [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://neo4j.com/docs/graph-data-science/current/algorithms/

Neo4j (2023c). *Graph database use cases & solutions: Where to use a graph database.* [Verkkosivu]. Viitattu: 3.3.2023. https://neo4j.com/use-cases/

Neo4j (2023d). *neosemantics (n10s): Neo4j RDF & Semantics toolkit.* [Verkkosivu]. Viitattu 8.3.2023. https://neo4j.com/labs/neosemantics/

Neo4j (2023e). *Subqueries in Cypher.* [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://neo4j.com/docs/getting-started/current/cypher-intro/subqueries/

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2), 167–256.

OpenLink Software (2019). *OpenLink Software: Virtuoso homepage.* [Verkkosivu]. Viitattu 3.3.2023. https://virtuoso.openlinksw.com/

Oracle (2021). *17 Use Cases for Graph Databases and Graph Analytics*. https://www.oracle.com/a/ocom/docs/graph-database-use-cases-ebook.pdf

Oracle (2023). *PGQL 1.5 Specification.* [Tekninen dokumentaatio]. Viitattu 14.3.2023. https://pgql-lang.org/spec/1.5/

Pabón, M. C., Millán, M., Roncancio, C. & Collazos, C. A. (2019). GraphTQL: A visual query system for graph databases. *Journal of Computer Languages*, 51(3), 97–111.

Patil, S., Vaswani, G. & Bhatia, A. (2014). Graph databases – An overview. *International Journal of Computer Science and Information Technologies,* 5(1), 657–660.

Petticrew, M., Egan, M., Thomson, H., Hamilton, V., Kunkler, R. & Roberts, H. (2008). Publication bias in qualitative research: What becomes of qualitative research presented at conferences? *Journal of Epidemiology and Community Health*, 62(6), 552–554.

Pokorný, J. (2015). Graph databases: their power and limitations. Teoksessa Saeed, K. & Homenda, W. (toim.), *Computer Information Systems and Industrial Management: 14th IFIP TC 8 International Conference, CISIM 2015*, 58–69. Springer International Publishing.

Rizaldy, A., Fahriah, S. & Hartono, N. (2021). Systematic literature review: Current products, topic, and implementation of graph database. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 7(1), 43–58.

Rodriguez, M. A. & Neubauer, P. (2010). *Constructions from dots and lines.* arXiv:1006.2361

Roussopoulos, N. & Mylopoulos, J. (1975). Using semantic networks for data base management. *Proceedings of the 1st International Conference on Very Large Data Bases,* 144–172.

Roy-Hubara, N. & Sturm, A. (2020). Design methods for the new database era: a systematic literature review. *Software and Systems Modeling*, 19(2), 297–312.

Saaranen-Kauppinen, A. & Puusniekka, A. (2006). Kvantifiointi. Teoksessa Saaranen-Kauppinen, A. & Puusniekka, A. (toim.), *KvaliMOTV – Menetelmäopetuksen tietovaranto.* Yhteiskuntatieteellinen tietoarkisto. Viitattu 19.2.2023. https://www.fsd.tuni.fi/menetelmaopetus/kvali/L7_3_3.html

Sahu, S., Mhedhbi, A., Salihoglu, S., Lin, J. & Özsu, M. T. (2020). The ubiquity of large graphs and surprising challenges of graph processing: extended survey. *The VLDB Journal*, 29, 595–618.

Salminen, A. (2011). *Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin.* Vaasan yliopiston julkaisuja, Opetusjulkaisuja 62, Julkisjohtaminen 4.

Seitamaa-Hakkarainen, P. (2014). *Kvalitatiivinen sisällönanalyysi.* Viitattu 18.2.2023. https://metodix.fi/2014/05/19/seitamaa-hakkarainen-kvalitatiivinen-sisallon-analyysi/

Shipman, D. W. (1981). The functional data model and the data languages DAPLEX. *ACM Transactions on Database Systems*, *6*(1), 140–173.

Silberschatz, A., Korth, H. F. & Sudarshan, S. (1996). Data models. *ACM Computing Surveys*, 28(1), 105–108.

Silberschatz, A., Stonebraker, M. & Ullman, J. (1991). Database systems: achievements and opportunities. *Communications of the ACM*, 34(10), 110–120.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339.

Solid IT (2022). *DB-Engines Ranking of Graph DBMS.* [Verkkosivu]. Viitattu 3.6.2022. https://db-engines.com/en/ranking/graph+dbms

Solid IT (2023). *DB-Engines Ranking - popularity ranking of database management systems*. [Verkkosivu]. Viitattu 7.3.2023. https://db-engines.com/en/ranking

Stardog Union (2023). *RDF graph data model*. [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://docs.stardog.com/tutorials/rdf-graph-data-model

Taylor, R. W. & Frank, R. L. (1976). CODASYL data-base management systems. *ACM Computing Surveys*, 8(1), 67–103.

The Apache Software Foundation (2022). *Apache TinkerPop: Gremlin.* [Verkkosivu]. Viitattu 8.3.2023. https://tinkerpop.apache.org/gremlin.html

Tian, Y. (2023). The world of graph databases from an industry perspective. *ACM SIGMOD Record*, *51*(4), 60–67.

TigerGraph (2023a). *Graph database in banking & financial services.* [Verkkosivu]. Viitattu 5.3.2023. https://www.tigergraph.com/solutions/financial-services/

TigerGraph (2023b). *Graph database use cases.* [Verkkosivu]. Viitattu 3.3.2023. https://www.tigergraph.com/solutions/

TigerGraph (2023c). *GSQL language reference.* [Tekninen dokumentaatio]. Viitattu 8.3.2023. https://docs.tigergraph.com/gsql-ref/current/intro/

TigerGraph (2023d). *Native graph database engine.* [Verkkosivu]. Viitattu 8.3.2023. https://www.tigergraph.com/tigergraph-db/

TigerGraph (1.3.2023e). *TigerGraph reports exceptional customer growth and product leadership as more market-leading companies tap the power of graph.* [Lehdistötiedote]. Viitattu 8.3.2023. https://www.tigergraph.com/press-article/tigergraph-reports-exceptional-customer-growth-and-product-leadership-as-more-market-leading-companies-tap-the-power-of-graph/

Timón-Reina, S., Rincón, M. & Martínez-Tomás, R. (2021). An overview of graph databases and their applications in the biomedical domain. *Database: The Journal of Biological Databases and Curation*, 2021: baab026.

Tompa, F. W. (1989). A data model for flexible hypertext database systems. *ACM Transactions on Information Systems*, 7(1), 85–100.

Tsichritzis, D. C. & Lochovsky, F. H. (1976). Hierarchical data-base management: A survey. *ACM Computing Surveys*, 8(1), 105–123.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. & Wilkins, D. (2010). A comparison of a graph database and a relational database: a data provenance perspective. *Proceedings of the 48th Annual Southeast Regional Conference*.

Voronkov, A., Iwaya, L. H., Martucci, L. A. & Lindskog, S. (2017). Systematic literature review on usability of firewall configuration. *ACM Computing Surveys,* 50(6): 87.

Vuori, J. (2021). Laadullinen sisällönanalyysi. Teoksessa Vuori, J. (toim.), *Laadullisen tutkimuksen verkkokäsikirja.* Yhteiskuntatieteellinen tietoarkisto. Viitattu 18.2.2023. https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/analyysitavan-valinta-ja-yleiset-analyysitavat/laadullinen-sisallonanalyysi/

W3C (21.3.2013). *SPARQL 1.1 Overview.* [W3C:n suositus]. Viitattu 24.2.2023. https://www.w3.org/TR/sparql11-overview/

W3C (25.2.2014). *RDF 1.1 Concepts and Abstract Syntax.* [W3C:n suositus]. Viitattu 23.2.2023. https://www.w3.org/TR/rdf11-concepts/

W3C (16.12.2019). *The RDF Concepts Vocabulary (RDF).* [RDF-sanasto]. Viitattu 14.3.2023. http://www.w3.org/1999/02/22-rdf-syntax-ns#

Watters, C. & Shepherd, M. A. (1990). A transient hypergraph-based model for data access. *ACM Transactions on Information Systems*, 8(2), 77–102.

Wood, P. T. (2012). Query languages for graph databases. *ACM Sigmod Record*, 41(1), 50–60.

Yasuda, Y. D., Martins, L. E. G. & Cappabianco, F. A. (2020). Autonomous visual navigation for mobile robots: A systematic literature review. *ACM Computing Surveys*, 53(1): 13.

Zhang, K. (2010). Introduction to the special issue on graph visualization. *Journal of Visual Languages and Computing*, *21*(4).

Zhou, L., Chen, R., Xia, Y. & Teodorescu, R. (2018). C-graph: A highly efficient concurrent graph reachability query framework. *Proceedings of the 47th International Conference on Parallel Processing*.

7.2 Systemaattisen kirjallisuuskatsauksen aineisto

Allen, D., Hodler, A., Hunger, M., Knobloch, M., Lyon, W., Needham, M. & Voigt, H. (2019). Understanding trolls with efficient analytics of large graphs in Neo4j. Teoksessa Grust, T., Naumann, F., Böhm, A., Lehner, W., Härder, T., Rahm, E., Heuer, A., Klettke, M. & Meyer, H. (toim.), *Datenbanksysteme für Business, Technologie und Web (BTW 2019), Lecture Notes in Informatics (LNI), Gesellschaft für Informatik,* 377–396.

Almeida, R., da Silva, W., Castro, K., Walter, M. E., Araujo, A., Holanda, M. & Lifschitz, S. (2017). AProvBio: An architecture for data provenance in bioinformatics workflows using graph database. *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2139–2144.*

Ariadi, A., Shi, T., Ma, H., da Silva, A. S. & Hartmann, S. (2021). A graph database supported GA-based approach to social network analysis. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*.

Arora, R., Maurya, A. M. & Sharma, Y. (2021). Application of Java relationship graphs (JRG) to plagiarism detection in Java projects: A Neo4j graph database approach. *2021 The 4th International Conference on Software Engineering and Information Management (ICSIM 2021)*, 46–51.

Au, H., & Lee, K. (2017). Graph database technology and k-means clustering for digital forensics. *Proceedings of the 16th European Conference on Cyber Warfare and Security,* 24–33.

Aung, T. T. & Nyunt, T. T. S. (2020). Community detection in scientific co-authorship networks using Neo4j. *2020 IEEE Conference on Computer Applications (ICCA)*.

Bajaj, V., Panda, R. B., Dabas, C. & Kaur, P. (2018). Graph database for recipe recommendations. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* 276–281.

Balaur, I., Saqi, M., Barat, A., Lysenko, A., Mazein, A., Rawlings, C. J., Ruskin, H. J. & Auffray, C. (2017). EpiGeNet: A graph database of interdependencies between genetic and epigenetic events in colorectal cancer. *Journal of Computational Biology*, 24(10), 969–980.

Barakat, O. L., Koll, D. & Fu, X. (2017). Gavel: Software-defined network control with graph databases. *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN),* 279–286.

Bedmar, I. S., Martínez, P. & Martín, A. C. (2017). Search and graph database technologies for biomedical semantic indexing: experimental analysis. *JMIR Medical Informatics*, 5(4): e48.

Bellini, P. & Nesi, P. (2018). Performance assessment of RDF graph databases for smart city services. *Journal of Visual Languages and Computing*, 45, 24–38.

Bollen, E., Hendrix, R., Kuijpers, B. & Vaisman, A. (2021). Towards the Internet of Water: Using graph databases for hydrological analysis on the Flemish river system. *Transactions in GIS*, 25(6), 2907–2938.

Boselli, R., Cesarini, M., Mercorio, F., Mezzanzanica, M. & Vaccarino, A. (2017). A pipeline for multimedia Twitter analysis through graph databases: preliminary results. *Proceedings of the 6th International Conference on Data Science, Technology and Applications (DATA 2017)*, 343–349.

Brandizi, M., Singh, A., Rawlings, C. & Hassani-Pak, K. (2018). Towards FAIRer biological knowledge networks using a hybrid linked data and graph database approach. *Journal of Integrative Bioinformatics*, 15(3).

Bukhari, S. A. C., Pawar, S., Mandell, J., Kleinstein, S. H. & Cheung, K.-H. (2021). LinkedImm: a linked data graph database for integrating immunological data. *BMC Bioinformatics*, 22(9): 105.

Carnaz, G., Nogueira, V. B. & Antunes, M. (2021). A graph database representation of portuguese criminal-related documents. *Informatics* 8(2): 37.

Celesti, A., Celesti, F., Galletta, A., Fazio, M. & Villari, M. (2020). Improving machine learning algorithm processing time in tele-rehabilization through a NoSQL graph database approach: a preliminary study. *2020 IEEE Symposium on Computers and Communications (ISCC).*

Cermak, M. & Sramkova, D. (2021). GRANEF: Utilization of a graph database for network forensics. *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, 785–790.

Chen, H., Vasardani, M., Winter, S. & Tomko, M. (2018). A graph database model for knowledge extracted from place descriptions. *ISPRS International Journal of Geo-Information,* 7(6): 221.

Chien, W.-T. & Hsieh, S.-H. (2020). A web-based approach to dynamically assessing space conflicts by integrating BIM and graph database. *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC 2020),* 307–312.

Constantinov, C., Mocanu, M., Poteraş, C. & Popa, B. (2018). Using a graph database for evaluating and enhancing a social reputation engine. *2018 19th International Carpathian Control Conference (ICCC)*, 518–523.

Costa, R. L., Gadelha, L., Ribeiro-Alves, M. & Porto, F. (2017). GeNNet: an integrated platform for unifying scientific workflows and graph databases for transcriptome data analysis. *PeerJ*, 5: e3509.

D'Agostino, D., Liò, P., Aldinucci, M. & Merelli, I. (2021). Advantages of using graph databases to explore chromatin conformation capture experiments. *BMC Bioinformatics*, 22(2): 43.

Dharmawan, I. N. P. W. & Sarno, R. (2017). Book recommendation using Neo4j graph database in BibTeX book metadata. *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 47–52.

Di Bratto, M., Di Maro, M., Origlia, A. & Cutugno, F. (2021). Dialogue analysis with graph databases: characterising domain items usage for movie recommendations. *Proceedings of the Eighth Italian Conference on Computational Linguistics CLiC-it 2021.*

Di Maro, M., Valentino, M., Riccio, A. & Origlia, A. (2017). Graph databases for designing high-performance speech recognition grammars. *12th International Conference on Computational Semantics (IWCS 2017)*.

Diez, F. P., Vasu, A. C., Touceda, D. S. & Cámara, J. M. S. (2017). Modeling XACML security policies using graph databases. *IT Professional*, 19(6), 52–57.

D'Onofrio, S., Wehrle, M., Portmann, E. & Myrach, T. (2017). Striving for semantic convergence with fuzzy cognitive maps and graph databases. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).*

Drakopoulos, G., Gourgaris, P. & Kanavos, A. (2020). Graph communities in Neo4j: Four algorithms at work. *Evolving Systems*, 11, 397–407.

Drakopoulos, G., Kanavos, A., Mylonas, P. & Sioutas, S. (2017). Defining and evaluating Twitter influence metrics: a higher-order approach in Neo4j. *Social Network Analysis and Mining*, 7: 52.

El Helou, S., Kobayashi, S., Yamamoto, G., Kume, N., Kondoh, E., Hiragi, S., Okamoto, K., Tamuri, H. & Kuroda, T. (2019). Graph databases for openEHR clinical repositories. *International Journal of Computational Science and Engineering*, 20(3), 281–298.

Elamin, R. & Osman, R. (2018). Implementing traceability repositories as graph databases for software quality improvement. *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 269–276.

Elmoselhy, A. M. & Ramadan, E. (2020). Utilizing graph database for inferring domain-disease associations. *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*.

Fabregat, A., Korninger, F., Viteri, G., Sidiropoulos, K., Marin-Garcia, P., Ping, P., Wu, G., Stein, L., D'Eustachio, P. & Hermjakob, H. (2018). Reactome graph database: Efficient access to complex pathway data. *PLoS Computational Biology*, 14(1): e1005968.

Fan, G., Zhu, M., Li, J., Wang, C. & Zhao, L. (2020). A graph database-based approach utilizing FAHP and directed bipartite graph for service composition. *Service Oriented Computing and Applications*, 14, 269–281.

Faralli, S., Velardi, P. & Yusifli, F. (2020). Multiple knowledge GraphDB (MKGDB). *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2325–2331.

Fourie, D., Claassens, S., Pillai, S., Mata, R. & Leonard, J. (2017). SLAMinDB: Centralized graph databases for mobile robotics. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 6331–6337.

Geepalla, E. & Asharif, S. (2020). Analysis of physical access control system for understanding users behavior and anomaly detection using Neo4j. *Proceedings of the 6th International Conference on Engineering & MIS 2020*.

Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M. & Seveso, A. (2021). Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing Journal*, 101:107049.

Gómez, L. I., Kuijpers, B. & Vaisman, A. A. (2019). Analytical queries on semantic trajectories using graph databases. *Transactions in GIS*, 23(5), 1078–1101.

Gong, F., Ma, Y., Gong, W., Li, X., Li, C. & Yuan, X. (2018). Neo4j graph database realizes efficient storage performance of oilfield ontology. *PLoS ONE*, 13(11): e0207595.

Gorawski, M. & Grochla, K. (2020). Performance tests of smart city IoT data repositories for universal linear infrastructure data and graph databases. *SN Computer Science*, 1: 31.

Guia, J., Soares, V. G. & Bernardino, J. (2017). Graph databases: Neo4j analysis. *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017)*, 1, 351-356.

Hall, R. J., Murray, C. W. & Verdonk, M. L. (2017). The fragment network: a chemistry recommendation engine built using a graph database. *Journal of Medicinal Chemistry*, 60(14), 6440–6450.

Hofer, D., Jäger, M., Mohamed, A. K. Y. S. & Küng, J. (2021). A study on time models in graph databases for security log analysis. *International Journal of Web Information Systems*, 17(5), 427–448.

Hor, A. E. H. & Sohn, G. (2021). Design and evaluation of a BIM-GIS integrated information model using RDF graph database. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 8, 175–182.

Hor, A. E. H., Sohn, G., Claudio, P., Jadidi, M. & Afnan, A. (2018). A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 89–96.

Hu, G.-M., Secario, M. K. & Chen, C.-M. (2019). SeQuery: an interactive graph database for visualizing the GPCR superfamily. *Database: The Journal of Biological Databases and Curation*, 2019: baz073.

Huang, S.-H., Yen, H.-P., Liu, Y.-H., Tseng, K.-H., Kung, J.-F., Lin, C.-C., Li, Y.-T., Chen, Y.-C. & Wang, C.-Y. (2021). Cluster tool performance analysis using graph database. *2021 IEEE 34th International System-on-Chip Conference (SOCC)*, 230–235.

Ismail, A., Nahar, A. & Scherer, R. (2017). Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard. *24th EG-ICE International Workshop on Intelligent Computing in Engineering 2017*.

Jamkhedkar, P., Johnson, T., Kanza, Y., Shaikh, A., Shankaranarayanan, N. K. & Shkapenyuk, V. (2018). A graph database for a virtualized network infrastructure. *Proceedings of the 2018 International Conference on Management of Data,* 1393–1405.

Kanavos, A., Drakopoulos, G. & Tsakalidis, A. (2017). Graph community discovery algorithms in Neo4j with a regularization-based evaluation metric. *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, 403–410.

Karunarathna, A., Senarath, D., Madhushanki, S., Weerakkody, C., Dayarathna, M., Jayasena, S. & Suzumura, T. (2020). Scalable graph convolutional network based link prediction on a distributed graph database server. *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, 107–115.

Kashef, M., Liu, Y., Montgomery, K. & Candell, R. (2021). Wireless cyber-physical system performance evaluation through a graph database approach. *Journal of Computing and Information Science in Engineering*, 21(2).

Konno, T., Huang, R., Ban, T. & Huang, C. (2017). Goods recommendation based on retail knowledge in a Neo4j graph database combined with an inference mechanism implemented in Jess. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*.

Kovács, T., Simon, G. & Mezei, G. (2019). Benchmarking graph database backends – What works well with Wikidata? *Acta Cybernetica*, 24(1), 43–60.

Küçükkeçeci, C. & Yazıcı, A. (2018). Big data model simulation on a graph database for surveillance in wireless multimedia sensor networks. *Big Data Research*, 11, 33–43.

Le, K. K., Whiteside, M. D., Hopkins, J. E., Gannon, V. P. & Laing, C. R. (2018). Spfy: an integrated graph database for real-time prediction of bacterial phenotypes and downstream comparative analyses. *Database: The Journal of Biological Databases and Curation*, 2018: bay086.

Le May, S., Carter, B. A., Gehly, S., Flegel, S. & Jah, M. (2020). Representing and querying space object registration data using graph databases. *Acta Astronautica*, 173, 392–403.

Lehotay-Kéry, P. & Kiss, A. (2020). Process, analyze and visualize telecommunication network configuration data in graph database. *Vietnam Journal of Computer Science*, 7(1), 65–76.

Lorincz, J., Huljić, V. & Begušić, D. (2020). Transforming product catalogue relational into graph database: a performance comparison. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO),* 523–528.

Lukács, D., Pongrácz, G. & Tejfel, M. (2020). Are graph databases fast enough for static P4 code analysis? *Proceedings of the 11th International Conference on Applied Informatics*, 213–223.

Lutu P. E. N. (2021) Methods for speeding up recommender system computations using a graph database. *Proceedings of the World Congress on Engineering 2021.*

Madani, K., Russo, C. & Rinaldi, A. M. (2019). Merging large ontologies using BigData GraphDB. *2019 IEEE International Conference on Big Data (Big Data)*, 2383–2392.

Mao, Z., Yao, H., Zou, Q., Zhang, W. & Dong, Y. (2021). Digital contact tracing based on a graph database algorithm for emergency management during the COVID-19 epidemic: Case study. *JMIR mHealth and uHealth*, 9(1): e26836.

Matter, H., Buning, C., Stefanescu, D. D., Ruf, S. & Hessler, G. (2020). Using graph databases to investigate trends in structure–activity relationship networks. *Journal of Chemical Information and Modeling*, 60(12), 6120–6134.

Mattioli, D. & Gubitoso, M. D. (2018). Application of graph database in the storage of heterogeneous omics data for the treatment in bioinformatics. *Proceedings of the 2018 10th International Conference on Bioinformatics and Biomedical Technology*, 51–56.

Maxwell, G., Kronmueller, M., Chang, D.-J. & Desoky, A. (2021). Measuring the "impact" of people, films, and TV using the IMDb graph database. *SoutheastCon 2021*.

Mei, S., Huang, X., Xie, C. & Mora, A. (2020). GREG – studying transcriptional regulation using integrative graph databases. *Database: The Journal of Biological Databases and Curation*, 2020: baz162.

Messina, A., Fiannaca, A., La Paglia, L., La Rosa, M. & Urso, A. (2018). BioGraph: a web application and a graph database for querying and analyzing bioinformatics resources. *BMC Systems Biology*, 12(5), 75–89.

Mezzanzanica, M., Mercorio, F., Cesarini, M., Moscato, V. & Picariello, A. (2018). GraphDBLP: a system for analysing networks of computer scientists through graph databases: GraphDBLP. *Multimedia Tools and Applications*, 77, 18657–18688.

Miranda, A., Arboleya, P., Suárez, L. & Carou, J. M. (2021). A common information model integration in a graph database for LV terminal distribution networks with PLC-based smart meters. *2021 IEEE Madrid PowerTech*.

Mondal, S., Basu, A. & Mukherjee, N. (2020). Building a trust-based doctor recommendation system on top of multilayer graph database. *Journal of Biomedical Informatics*, 110: 103549.

Muramudalige, S. R., Hung, B. W., Jayasumana, A. P., Ray, I. & Klausen, J. (2021). Enhancing investigative pattern detection via inexact matching and graph databases. *IEEE Transactions on Services Computing*, 15(5), 2780–2794.

Nguyen, S. H., Yao, Z. & Kolbe, T. H. (2017). Spatio-semantic comparison of large 3D city models in CityGML using a graph database. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science*, 99–106.

Nguyen, T. & Do, P. (2017). Managing and visualizing citation network using graph database and LDA model. *Proceedings of the 8th International Symposium on Information and Communication Technology*, 100–105.

Nuijten, R. C. & Van Gorp, P. (2021). SciModeler: a metamodel and graph database for consolidating scientific knowledge by linking empirical data with theoretical constructs. *Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2021)*, 314–321.

Omasa, A. & Inoue, U. (2019). Extracting related concepts from Wikipedia by using a graph database system. *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 268–273.

Origlia, A., Rossi, S., Martino, S. D., Cutugno, F. & Chiacchio, M. L. (2021). Multiple-source data collection and processing into a graph database supporting cultural heritage applications. *ACM Journal on Computing and Cultural Heritage*, 14(4): 55.

Pacaci, A., Zhou, A., Lin, J. & Özsu, M. T. (2017). Do we need specialized graph databases? Benchmarking real-time social networking applications. *Proceedings of the Fifth International Workshop on Graph Data-management Experiences and Systems*.

Padayachy, T., Scholtz, B. & Wesson, J. (2018). An information extraction model using a graph database to recommend the most applied case. *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE),* 89–94.

Pan, Z. & Jing, Z. (2018). Modeling methods of big data for power grid based on graph database. *2018 International Conference on Power System Technology (POWERCON)*, 4340–4348.

Perçuku, A., Minkovska, D. & Stoyanova, L. (2017). Modeling and processing big data of power transmission grid substation using Neo4j. *Procedia Computer Science*, 113, 9–16.

Prusti, D., Das, D. & Rath, S. K. (2021). Credit card fraud detection technique by applying graph database model. *Arabian Journal for Science and Engineering*, 46, 8849–8868.

Pujante, L., Campos, M., Juarez, J. M., Cánovas-Segura, B. & Nicolás, A. M. (2021). Multi-resistant bacterial infection surveillance using a graph database with spatio-temporal information. *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021)*, 5, 741–746.

Quaeghebeur, E., Sanchez Perez-Moreno, S. & Zaaijer, M. B. (2020). WESgraph: a graph database for the wind farm domain. *Wind Energy Science*, 5, 259–284.

Rani, A., Goyal, N. & Gadia, S. K. (2021). Provenance framework for Twitter data using zero-information loss graph database. *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD),* 74–82.

Ravikumar, G. & Khaparde, S. A. (2017). A common information model oriented graph database framework for power systems. *IEEE Transactions on Power Systems*, 32(4), 2560–2569.

Sadyrin, D., Dergachev, A., Loginov, I., Korenkov, I. & Ilina, A. (2019). Application of graph databases for static code analysis of web-applications. *Proceedings of the 11th Majorov International Conference on Software Engineering and Computer Systems (MICSECS 2019).*

Satish, C. J. & Mahendran, A. (2018). Automated bug assignment in software maintenance using graph databases. *International Journal of Intelligent Systems and Applications*, 12(2), 27–36.

Schindler, T. (2017). Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats. Teoksessa Eibl, M. & Gaedke, M. (toim.), *INFORMATIK 2017. Gesellschaft für Informatik, Bonn*, 2371–2378.

Sharma, C., Sinha, R. & Leitao, P. (2019). IASelect: Finding best-fit agent practices in industrial CPS using graph databases. *Proceedings of the 17th International Conference on Industrial Informatics (INDIN2019)*, 1558–1563.

Shi, Y.-X., Zhang, B.-K., Wang, Y.-X., Luo, H.-Q. & Li, X. (2021). Constructing crop portraits based on graph databases is essential to agricultural data mining. *Information*, 12: 227.

Shoshi, A., Hofestädt, R., Zolotareva, O., Friedrichs, M., Maier, A., Ivanisenko, V. A., Dosenko, V. E. & Bragina, E. Y. (2018). GenCoNet – a graph database for the analysis of comorbidities by gene networks. *Journal of Integrative Bioinformatics*, 15(4): 20180049.

Simpson, C. M. & Gnad, F. (2020). Applying graph database technology for analyzing perturbed co-expression networks in cancer. *Database: The Journal of Biological Databases and Curation*, 2020: baaa110.

Smidt, H., Thornton, M. & Ghorbani, R. (2018). Smart application development for IoT asset management using graph database modeling and high-availability web services. *Proceedings of the Annual Hawaii International Conference on System Sciences*.

Swainston, N., Batista-Navarro, R., Carbonell, P., Dobson, P. D., Dunstan, M., Jervis, A. J., Vinaixa, M., Williams, A. R., Ananiadou, S., Faulon, J.-L., Mendes, P., Kell, D. B., Scrutton, N. S. & Breitling, R. (2017). biochem4j: Integrated and extensible biochemical knowledge through graph databases. *PLoS ONE*, 12(7): e0179130.

Tao, X., Liu, Y., Zhao, F., Yang, C. & Wang, Y. (2018). Graph database-based network security situation awareness data storage method. *EURASIP Journal on Wireless Communications and Networking*, 2018: 294.

Thapa, I. & Ali, H. (2021). A multiomics graph database system for biological data integration and cancer informatics. *Journal of Computational Biology*, 28(2), 209–219.

Thirupathi, L. & Padmanabhuni, V. N. R. (2021). Multi-level protection (Mlp) policy implementation using graph database. *International Journal of Advanced Computer Science and Applications*, 12(3).

Tripathi, G., Sharma, B. & Rajvanshi, S. (2017). A combination of internet of things (IoT) and graph database for future battlefield systems. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 1252–1257.

Tsitseklis, K., Krommyda, M., Karyotis, V., Kantere, V. & Papavassiliou, S. (2021). Scalable community detection for complex data graphs via hyperbolic network embedding and graph databases. *IEEE Transactions on Network Science and Engineering*, 8(2), 1269–1282.

Vágner, A. (2021). Route planning on GTFS using Neo4j. *Annales Mathematicae et Informaticae*, 54, 163–179.

Virk, A. & Rani, R. (2018). Efficient approach for social recommendations using graphs on Neo4j. *Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018),* 133–138.

Wang, D., Guo, Q., Song, Y., Gao, K. & Zhou, A. (2017). Cyber-physical security assessment and simulation based on graph database. *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*.

Wiseso, L. G., Imrona, M. & Alamsyah, A. (2020). Performance analysis of Neo4j, MongoDB, and PostgreSQL on 2019 national election big data management database. *2020 6th International Conference on Science in Information Technology (ICSITech)*, 91–96.

Yang, B., Dong, M., Wang, C., Liu, B., Wang, Z. & Zhang, B. (2021). IFC-based 4D construction management information model of prefabricated buildings and its application in graph database. *Applied Sciences*, 11:7270.

Yerashenia, N. & Bolotov, A. (2019). Computational modelling for bankruptcy prediction: Semantic data analysis integrating graph database and financial ontology. *2019 IEEE 21st Conference on Business Informatics (CBI),* 84–93.

Yuan, B., Pan, Z., Shi, F. & Li, Z. (2020). An attack path generation methods based on graph database. *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 1905–1910.

Yuan, C., Lu, Y., Liu, K., Liu, G., Dai, R. & Wang, Z. (2018). Exploration of bi-level PageRank algorithm for power flow analysis using graph database. *2018 IEEE International Congress on Big Data (BigData Congress)*, 143–149.

Zhu, Z., Zhou, X. & Shao, K. (2019). A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem. *Computers & Industrial Engineering*, 130, 671–686.

Liite 1: Systemaattisen kirjallisuuskatsauksen tulostaulukko

Tekijät	Sovellusala*	Graafitietokanta**	Sov. verkottuneelle datalle (+)	Sov. heterogeeniselle datalle (+)	Suorituskyky (+)	Rinnakkaisuus (+)	Visualisointi (+)	Selitysvoima (+)	Tietomallin ymmärrettävyys (+)	Tietokantakaavion joustavuus (+)	Kyselykielen ymmärrettävyys (+)	Graafikyselyt ja -algoritmit (+)	Opettelu (–)	Soveltum. tietynlaiselle datalle (–)	Suorituskyky (–)	Muut (–)***
Allen *et al.* 2019	SV	N4	x		x	x	x	x	x		x	x				
Almeida *et al.* 2017	BI	N4	x					x	x			x				
Ariadi *et al.* 2021	SV	N4	x		x			x				x			x	
Arora *et al.* 2021	OT	N4	x				x	x	x			x				
Au & Lee 2017	TV	N4	x		x			x				x				
Aung & Nyunt 2020	SV	N4	x				x	x			x	x				
Bajaj *et al.* 2018	SJ	N4	x		x		x	x	x			x				
Balaur *et al.* 2017	BI	N4	x	x	x		x	x				x				
Barakat *et al.* 2017	TV	N4	x		x	x			x		x	x				
Bedmar *et al.* 2017	AT	BG	x		x							x				
Bellini & Nesi 2018	GT	U		x	x			x				x				
Bollen *et al.* 2021	GT	N4	x		x	x		x	x		x	x			x	
Boselli *et al.* 2017	SV	N4	x				x	x				x				
Brandizi *et al.* 2018	BI	U	x					x		x	x	x			x	
Bukhari *et al.* 2021	BI	N4	x	x	x		x	x	x	x		x	x			
Carnaz *et al.* 2021	TU	N4		x			x	x		x		x				
Celesti *et al.* 2020	KO	N4	x		x	x						x				
Cermak & Sramkova 2021	TV	DG	x				x	x	x			x	x	x		SV
Chen *et al.* 2018	GT	N4	x		x		x	x	x			x				
Chien & Hsieh 2020	RK	N4	x		x			x				x				
Constantinov *et al.* 2018	SV	N4	x		x		x	x		x		x				
Costa *et al.* 2017	BI	N4	x	x			x	x		x		x				
D'Agostino *et al.* 2021	BI	N4	x		x		x	x	x			x				
Dharmawan & Sarno 2017	SJ	N4	x						x	x		x			x	
Di Bratto *et al.* 2021	KO	N4	x					x		x		x				
Di Maro *et al.* 2017	KO	N4	x		x			x	x		x	x	x			
Diez *et al.* 2017	TV	N4	x		x							x				
D'Onofrio *et al.* 2017	GT	OR	x		x				x	x		x		x		

Tekijät	Sovellusala*	Graafitietokanta**	Sov. verkottuneelle datalle (+)	Sov. heterogeeniselle datalle (+)	Suorituskyky (+)	Rinnakkaisuus (+)	Visualisointi (+)	Selitysvoima (+)	Tietomallin ymmärrettävyys (+)	Tietokantakaavion joustavuus (+)	Kyselykielen ymmärrettävyys (+)	Graafikyselyt ja -algoritmit (+)	Opettelu (–)	Soveltum. tietynlaiselle datalle (–)	Suorituskyky (–)	Muut (–)***
Drakopoulos *et al.* 2017	SV	N4	x					x				x				
Drakopoulos *et al.* 2020	SV	N4	x					x				x				
El Helou *et al.* 2019	TH	N4	x		x		x		x	x	x	x			x	
Elamin & Osman 2018	OT	N4	x		x		x	x	x		x	x				
Elmoselhy & Ramadan 2020	BI	N4	x	x				x				x				
Fabregat *et al.* 2018	BI	N4	x		x	x		x	x	x	x	x				
Fan *et al.* 2020	PT	N4	x		x			x		x		x				
Faralli *et al.* 2020	TG	N4	x	x	x			x		x	x	x				
Fourie *et al.* 2017	RO	N4	x		x	x				x		x				
Geepalla & Asharif 2020	TU	N4	x		x		x	x			x	x				
Giabelli *et al.* 2021	SJ	S2	x		x			x		x		x				
Gómez *et al.* 2019	GT	N4	x		x			x	x		x	x				
Gong *et al.* 2018	TG	N4	x		x					x		x				
Gorawski & Grochla 2020	GT	N4	x		x					x	x	x			x	
Guia *et al.* 2017	SV	N4	x		x				x		x	x	x	x		ES
Hall *et al.* 2017	BK	N4	x		x		x	x	x			x				
Hofer *et al.* 2021	TV	N4		x					x			x			x	
Hor & Sohn 2021	GT	N4	x		x		x	x	x	x		x				
Hor *et al.* 2018	GT	N4	x	x	x		x	x	x	x	x	x	x			
Hu *et al.* 2019	BI	SQ	x		x		x	x				x				
Huang *et al.* 2021	TO	N4	x		x			x				x				
Ismail *et al.* 2017	RK	N4	x				x	x				x	x			
Jamkhedkar *et al.* 2018	TV	NP	x		x			x	x	x	x	x				
Kanavos *et al.* 2017	SV	N4	x					x				x				
Karunarathna *et al.* 2020	KO	JG	x		x	x		x				x				
Kashef *et al.* 2021	EI	N4	x	x	x		x	x	x	x	x	x				

Tekijät	Sovellusala*	Graafitietokanta**	Sov. verkottuneelle datalle (+)	Sov. heterogeeniselle datalle (+)	Suorituskyky (+)	Rinnakkaisuus (+)	Visualisointi (+)	Selitysvoima (+)	Tietomallin ymmärrettävyys (+)	Tietokantakaavion joustavuus (+)	Kyselykielen ymmärrettävyys (+)	Graafikyselyt ja -algoritmit (+)	Opettelu (–)	Soveltum. tietynlaiselle datalle (–)	Suorituskyky (–)	Muut (–)***
Konno *et al.* 2017	SJ	N4	x		x		x	x	x			x				
Kovács *et al.* 2019	TG	U	x						x			x			x	
Küçükkeçeci & Yazıcı 2018	EI	U	x		x			x				x				
Le *et al.* 2018	BI	BG	x		x			x		x						
Le May *et al.* 2020	TT	N4	x	x				x		x		x				
Lehotay-Kéry & Kiss 2020	TV	N4	x		x		x		x			x				
Lorincz *et al.* 2020	PT	AR	x		x		x	x	x		x	x				
Lukács *et al.* 2020	OT	U	x			x						x				
Lutu 2021	SJ	N4	x		x							x			x	
Madani *et al.* 2019	TG	N4	x	x	x		x	x	x		x	x				
Mao *et al.* 2021	EP	N4	x	x			x	x				x				
Matter *et al.* 2020	BK	N4	x	x	x		x	x	x	x		x				
Mattioli & Gubitoso 2018	BI	N4	x	x	x			x		x	x	x			x	
Maxwell *et al.* 2021	KU	N4	x				x	x	x			x				
Mei *et al.* 2020	BI	N4	x	x			x	x	x			x				
Messina *et al.* 2018	BI	U		x			x	x		x	x	x				
Mezzanzanica *et al.* 2018	SV	N4	x		x			x	x	x	x	x			x	
Miranda *et al.* 2021	EN	TG	x		x	x			x	x		x				LT
Mondal *et al.* 2020	SJ	N4	x	x	x			x		x		x				
Muramudalige *et al.* 2021	SV	N4	x	x	x			x	x			x				
Nguyen & Do 2017	SV	N4	x		x		x	x			x	x				LT
Nguyen *et al.* 2017	GT	N4	x					x				x				
Nuijten & Van Gorp 2021	AT	N4	x				x	x		x		x				
Omasa & Inoue 2019	KO	N4	x		x			x			x	x				
Origlia *et al.* 2021	KU	N4		x	x			x		x		x				
Pacaci *et al.* 2017	SV	N4	x		x				x			x			x	KY
Padayachy *et al.* 2018	LS	N4	x		x		x	x		x		x				ES

Tekijät	Sovellusala*	Graafitietokanta**	Sov. verkottuneelle datalle (+)	Sov. heterogeeniselle datalle (+)	Suorituskyky (+)	Rinnakkaisuus (+)	Visualisointi (+)	Selitysvoima (+)	Tietomallin ymmärrettävyys (+)	Tietokantakaavion joustavuus (+)	Kyselykielen ymmärrettävyys (+)	Graafikyselyt ja -algoritmit (+)	Opettelu (–)	Soveltum. tietynlaiselle datalle (–)	Suorituskyky (–)	Muut (–)***
Pan & Jing 2018	EN	N4	x		x					x		x				
Perçuku *et al.* 2017	EN	N4	x	x	x				x	x	x	x				
Prusti *et al.* 2021	TU	N4	x					x		x	x	x				
Pujante *et al.* 2021	EP	N4	x				x	x		x	x	x	x	x		AT
Quaeghebeur *et al.* 2020	EN	N4	x			x	x	x	x			x	x			
Rani *et al.* 2021	SV	N4	x		x		x	x		x		x				
Ravikumar & Khaparde 2017	EN	N4	x		x	x	x			x		x				
Sadyrin *et al.* 2019	OT	N4	x					x		x		x				
Satish & Mahendran 2018	OT	N4	x		x			x				x				
Schindler 2017	TV	N4	x		x			x								
Sharma *et al.* 2019	PT	N4	x				x	x	x	x		x				
Shi *et al.* 2021	MT	N4	x				x	x	x		x	x				
Shoshi *et al.* 2018	BI	N4	x	x				x				x				
Simpson & Gnad 2020	BI	N4	x		x			x	x			x				
Smidt *et al.* 2018	EI	N4	x		x				x			x				
Swainston *et al.* 2017	BK	N4	x				x	x	x	x	x	x	x			
Tao *et al.* 2018	TV	N4	x	x	x		x	x			x	x				
Thapa & Ali 2021	BI	N4		x	x			x		x		x				
Thirupathi & Padmanabhuni 2021	TV	N4	x					x			x	x				
Tripathi *et al.* 2017	EI	N4	x	x	x		x	x	x			x				
Tsitseklis *et al.* 2021	SV	N4	x		x		x	x	x	x		x				
Vágner 2021	GT	N4	x								x	x		x		
Virk & Rani 2018	SJ	N4	x		x			x				x				
Wang *et al.* 2017	EN	OR	x		x			x	x	x						
Wiseso *et al.* 2020	PO	N4	x							x	x	x			x	
Yang *et al.* 2021	RK	N4	x				x	x	x			x				
Yerashenia & Bolotov 2019	RH	N4	x	x	x		x	x	x	x	x	x				
Yuan *et al.* 2018	EN	TG	x		x	x		x	x			x				

Tekijät	Sovellusala*	Graafitietokanta**	Sov. verkottuneelle datalle (+)	Sov. heterogeeniselle datalle (+)	Suorituskyky (+)	Rinnakkaisuus (+)	Visualisointi (+)	Selitysvoima (+)	Tietomallin ymmärrettävyys (+)	Tietokantakaavion joustavuus (+)	Kyselykielen ymmärrettävyys (+)	Graafikyselyt ja -algoritmit (+)	Opettelu (–)	Soveltum. tietynlaiselle datalle (–)	Suorituskyky (–)	Muut (–)***
Yuan *et al.* 2020	TV	N4	x		x		x	x		x	x	x				KY, TO
Zhu *et al.* 2019	TO	N4	x		x			x	x	x		x				

Taulukko 7.1. Systemaattisen kirjallisuuskatsauksen tulokset taulukoituna.

Taulukossa käytetään seuraavia merkintöjä:

(+) Graafitietokannan hyöty.

(–) Graafitietokannan haitta.

(*) Sovellusaloista on käytetty seuraavia lyhenteitä: AT = akateeminen tutkimus, BI = bioinformatiikka, BK = biokemia, EI = esineiden internet, EN = energia, EP = epidemiologia, GT = geografinen tieto, KO = koneoppiminen, KU = kulttuuri, LS = lainsäädäntö, MT = maatalous, OT = ohjelmistotuotanto, PO = politiikka, PT = palveluntuotanto, RH = rahoitus, RK = rakentaminen, RO = robotiikka, SJ = suosittelujärjestelmät, SV = sosiaaliset verkostot, TG = tietämysgraafit, TH = terveydenhuolto, TO = tuotannonohjaus, TT = tähtitiede, TU = turvallisuus ja TV = tietoverkot.

(**) Graafitietokannoista on käytetty seuraavia lyhenteitä: AR = ArangoDB, BG = Blazegraph, DG = Dgraph, JG = JasmineGraph, N4 = Neo4j, NP = Nepal, OR = OrientDB, S2 = S2JGraph, SQ = SeQuery, TG = TigerGraph ja U = Useita.

(***) Muut-sarakkeeseen on yhdistetty useita eri haittoja, joista on käytetty lyhenteitä: AT = algoritmien toteutus, ES = epästabiilius, KY = kypsyys, LK = levytilan käyttö, SV = suhteiden visualisointi ja TO = turvaominaisuudet.

Liite 2: Systemaattisen kirjallisuuskatsauksen artikkelien aiheet

Sovellusala	**Artikkelien aiheet**
Akateeminen tutkimus	Tieteellisen tiedon metamalli (Nuijten & Van Gorp 2021)
	Tutkimusten semanttinen indeksointi (Bedmar *et al.* 2017)
Bioinformatiikka	Biologisen datan tietokanta (Brandizi *et al.* 2018; Bukhari *et al.* 2021; Fabregat *et al.* 2018; Mattioli & Gubitoso 2018; Messina *et al.* 2018)
	Bioinformatiikan analyysi (D'Agostino *et al.* 2021; Elmoselhy & Ramadan 2020; Le *et al.* 2018; Mei *et al.* 2020; Shoshi *et al.* 2018)
	Biologisen datan visualisointi (Hu *et al.* 2019)
	Syöpätutkimus (Balaur *et al.* 2017; Simpson & Gnad 2020; Thapa & Ali 2021)
	Bioinformatiikan työnkulkujen hallinta (Almeida *et al.* 2017; Costa *et al.* 2017)
Biokemia	Biokemiallinen analyysi (Matter *et al.* 2020)
	Biokemiallisen datan tietokanta (Swainston *et al.* 2017)
	Biokemian suosittelujärjestelmä (Hall *et al.* 2017)
Energia	Sähköverkkodatan tietokanta (Miranda *et al.* 2021; Pan & Jing 2018; Perçuku *et al.* 2017; Ravikumar & Khaparde 2017)
	Sähköverkkojen analysointi (Wang *et al.* 2017; Yuan *et al.* 2018)
	Tuulivoima (Quaeghebeur *et al.* 2020)
Epidemiologia	Kontaktien jäljitys (Mao *et al.* 2021; Pujante *et al.* 2021)
Esineiden internet (IoT)	IoT-laittaiden tuottaman datan analyysi (Küçükkeçeci & Yazıcı 2018)
	IoT-laitteiden hallinta (Smidt *et al.* 2018)
	IoT-laitteet taistelukentän hallintajärjestelmissä (Tripathi *et al.* 2017)
	Langattomien kyberfyysisten järjestelmien suorituskyvyn arviointi (Kashef *et al.* 2021)
Geografinen tieto	Rakennustietomallin ja geografisen datan integrointi (Hor *et al.* 2018; Hor & Sohn 2021)
	Kaupunkien mallintaminen (Nguyen *et al.* 2017)
	Julkisen liikenteen reitinsuunnittelu (Vágner 2021)
	Infrastruktuuri- ja IoT-mittaridatan yhdistäminen (Gorawski & Grochla 2020)
	Jokijärjestelmän tietokanta (Bollen *et al.* 2021)
	Paikkatietojen kuvaukset (Chen *et al.* 2018)
	Liikkuvien objektien analyysi (Gómez *et al.* 2019)
	Sumeat kognitiiviset kartat kognitiivisissa kaupungeissa (D'Onofrio *et al.* 2017)
	Älykaupunkipalvelut (Bellini & Nesi 2018)
Koneoppiminen	Koneoppimisalgoritmi etäkuntouksen palveluissa (Celesti *et al.* 2020)
	Neuroverkot (Karunarathna *et al.* 2020)
	Luonnollisen kielen prosessointi (Di Bratto *et al.* 2021; Omasa & Inoue 2019)
	Puheentunnistus (Di Maro *et al.* 2017)
Kulttuuri	Elokuva-alan vaikuttavuuden analyysi (Maxwell *et al.* 2021)
	Kulttuurisen perinnön tietokanta (Origlia *et al.* 2021)

Sovellusala	**Artikkelien aiheet**
Lainsäädäntö	Yleisimmin sovelletun tapauksen suosittelu (Padayachy *et al.* 2018)
Maatalous	Viljelyskasvien tietokanta (Shi *et al.* 2021)
Ohjelmistotuotanto	Automaattinen bugikorjausten osoitus (Satish & Mahendran 2018)
	Koodin jäljitettävyyden säilöminen (Elamin & Osman 2018)
	Plagiarismin havainnointi koodissa (Arora *et al.* 2021)
	Staattinen koodianalyysi (Lukács *et al.* 2020; Sadyrin *et al.* 2019)
Palveluntuotanto	Palvelun koostumuksen määritys (Fan *et al.* 2020)
	Palvelutoimittajien suosittelu (Sharma *et al.* 2019)
	Tuotekatalogit (Lorincz *et al.* 2020)
Politiikka	Vaalidatan analyysi (Wiseso *et al.* 2020)
Rahoitus	Konkurssien ennustaminen (Yerashenia & Bolotov 2019)
Rakentaminen	Rakennuksen tietomalli (Ismail *et al.* 2017)
	Tilakonfliktien ehkäisy rakennustyömaalla rakennuksen tietomallin ja graafitietokannan integroinnin avulla (Chien & Hsieh 2020)
	Rakentamisen tietomalli elementtitaloille (Yang *et al.* 2021)
Robotiikka	Mobiilirobotiikan tietokanta (Fourie *et al.* 2017)
Sosiaaliset verkostot	Reaaliaikaisen viestinnän sovellus (Pacaci *et al.* 2017)
	Sitaattiverkot (Mezzanzanica *et al.* 2018; Nguyen & Do 2017)
	Sosiaalisen maineen analysointi (Constantinov *et al.* 2018)
	Sosiaalisen verkoston tietokanta (Guia *et al.* 2017; Rani *et al.* 2021)
	Trollien analysointi (Allen *et al.* 2019)
	Tutkiva jäljittäminen (Muramudalige *et al.* 2021)
	Twitter-sisältöjen analysointi (Boselli *et al.* 2017; Drakopoulos *et al.* 2017)
	Yhteisöjen tunnistaminen (Ariadi *et al.* 2021; Aung & Nyunt 2020; Drakopoulos *et al.* 2020; Kanavos *et al.* 2017; Tsitseklis *et al.* 2021)
Suosittelujärjestelmät	Kirjojen suosittelujärjestelmä (Dharmawan & Sarno 2017)
	Suosittelujärjestelmän laskennan nopeutus (Lutu 2021)
	Lääkärien suosittelujärjestelmä (Mondal *et al.* 2020)
	Ruokareseptien suosittelujärjestelmä (Bajaj *et al.* 2018)
	Sosiaaliset suositukset (Virk & Rani 2018)
	Fyysisten tuotteiden suosittelujärjestelmä (Konno *et al.* 2017)
	Työnhaun suosittelujärjestelmä (Giabelli *et al.* 2021)
Terveydenhuolto	Sähköiset potilastiedot (El Helou *et al.* 2019)
Tietoverkot	Kyberrikosten tutkinta (Au & Lee 2017; Cermak & Sramkova 2021)
	Verkkoliikenteen analyysi (Hofer *et al.* 2021; Schindler 2017; Tao *et al.* 2018)
	Verkon pääsynhallinta (Diez *et al.* 2017; Thirupathi & Padmanabhuni 2021)
	Kyberhyökkäysteiden simulointi (Yuan *et al.* 2020)
	Tietoverkon hallinta (Barakat *et al.* 2017; Lehotay-Kéry & Kiss 2020; Jamkhedkar *et al.* 2018)
Tietämysgraafit	Ontologiat (Gong *et al.* 2018; Madani *et al.* 2019)
	Tietämysgraafien yhdistäminen (Faralli *et al.* 2020)
	Wikidatan tietokanta (Kovács *et al.* 2019)

Sovellusala	**Artikkelien aiheet**
Tuotannonohjaus	Tuotantolaitteiden suorituskyvyn analyysi (Huang *et al.* 2021)
	Tuotantotehtävien aikataulutus (Zhu *et al.* 2019)
Turvallisuus	Fyysisen pääsyn kontrollointijärjestelmä (Geepalla & Asharif 2020)
	Petosten havaitseminen (Prusti *et al.* 2021)
	Rikollisiin liittyvien dokumenttien tietokanta (Carnaz *et al.* 2021)
Tähtitiede	Avaruusobjektien rekisteröintidatan tietokanta (Le May *et al.* 2020)

Taulukko 7.2. Systemaattisen kirjallisuuskatsauksen artikkelien aiheet sovellusaloittain.

Liite 3: Vertailu systemaattisen kirjallisuuskatsauksen ja aiemman kirjallisuuden sovellusaloista

Sovellusala	SKK	AK	Huomioitavaa
Biotieteet	19*	10	Sisältää bioinformatiikan ja biokemian
Sosiaaliset verkostot	17*	11	Sisältää epidemiologian
Tietoverkot	11	10*	Sisältää kyberturvallisuuden
Geografinen tieto	10	8*	Sisältää kuljetukset ja liikenteen
Suosittelujärjestelmät	7	8	
Energia	7	4	
Tekoäly ja koneoppiminen	5*	5	Luokka laajennettu kattamaan tekoälyn
Ohjelmistotuotanto	5	0	
Tietämysgraafit	4	4	
Esineiden internet	4	1	
Markkinointi	3*	5	Sisältää palveluntuotannon
Rakentaminen	3	0	
Tuotannonohjaus	2	3	
Akateeminen tutkimus	2	2	
Kulttuuri	2	1	
Petosten tutkinta	1*	9	Jaettu Turvallisuus-sovellusalasta
Julkishallinto	1*	6	Jaettu Turvallisuus-sovellusalasta
Rahoitus	1	6	
Terveydenhuolto	1	1	
Laki	1	0	
Maatalous	1	0	
Politiikka	1	0	
Robotiikka	1	0	
Turvallisuus	1*	0	Jaettu Turvallisuus-sovellusalasta
Tähtitiede	1	0	
Datanhallinta	0	8	
Tietojärjestelmät	0	5	
Riskinhallinta	0	3	
Vähittäiskauppa	0	3	
Web ja hyperteksti	0	3	
Datatiede	0	2	
Hakukoneet	0	2	
Journalismi	0	2	
Telekommunikaatio	0	2	
Elintarvikkeet	0	1	
Henkilöstöhallinto	0	1	
Koulutus	0	1	

Taulukko 7.3. Systemaattisen kirjallisuuskatsauksen (SKK) artikkeleissa ja aiemman kirjallisuuden (AK) listauksissa esiintyneet graafitietokantojen sovellusalat. Tähdellä (*) merkittyjä luokkia on muunnettu Huomioitavaa-sarakkeessa kerrotulla tavalla.