

Alexi Saarinen

**TIETOKONEKIELTÄ KIRJOITTAESSA
TEHDYT VIRHEET: SYSTEMAATTINEN
KIRJALLISUUSKARTOITUS**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2023

TIIVISTELMÄ

Saarinen, Aleksi

Tietokonekieltä kirjoittaessa tehdyt virheet: Systemaattinen kirjallisuuskartoitus

Jyväskylä: Jyväskylän yliopisto, 2023, 45 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja(t): Taipalus, Toni

Tietokonekielet ovat avainasemassa nykyajan jatkuvasti kasvavassa digitaalisessa maailmassa. Kaikki tietokoneilla ja älypuhelimilla käytettävät ohjelmistot, sovellukset ja palvelut ovat tavalla tai toisella linkittyneitä tietokonekieliin. Tietokonekielten kirjoittaminen on monimutkainen tehtävä, joka vaatii kirjoittajalta monimuotoista osaamista matemaattisella, loogisella ja usein liiketoiminnallisellakin tasolla. Kirjoittajalla on oltava kokemusta erilaisten kielten syntaktisista, semanttisista ja loogisista säännöistä. Tämän tutkielman tavoitteena on tarjota katsaus tutkittuun tietoon tietokonekielten kirjoittamisessa tapahtuvien virheiden saralla käyttäen systemaattisen kirjallisuuskartoituksen menetelmää. Kyseisestä aiheesta ei ole saatavilla aikaisempaa kartoitusta. Katsauksessa tarkastellaan saatavilla olevien tutkimusten määrää, millä foorumeilla tutkimuksia on julkaistu ja miten tutkimusten määrä on kehittynyt ajan saatossa. Vaikkakin tiedon määrä aihealueen parissa on lisääntynyt viimeisen kymmenen vuoden aikana, aihealueesta ei ole saatavilla tieteellistä tutkimusta kovinkaan laajasti. Tutkielma osoittaa, että aiheesta julkaistu tutkimus keskittyy suurimmilta osin tietokonekielten oppimiseen ja opettamiseen. Aiheesta tehtävä jatkotutkimus voi tarjota paljon uudenlaista tietoa eri taitojen opettamiseen, joita vaaditaan tietokonekielten kirjoittamiseen. Toisaalta laajempi tutkimus aiheen parissa voi edistää aloittelevien kirjoittajien työelämään siirtymistä ja tätä kautta auttaa myös yrityksiä rekrytoimaan valmiimpia osaajia.

Avainsanat: systemaattinen kirjallisuuskartoitus, tietokonekielet, kirjoitusvirheet, virhekoodit

ABSTRACT

Saarinen, Aleksi

User errors in computer languages: Systematic mapping study

Jyväskylä: University of Jyväskylä, 2023, 45 pp.

Information Systems, Master's Thesis

Supervisor(s): Taipalus, Toni

Computer languages are of paramount importance in today's ever-expanding digital world. All software, applications, and services used on computers and smartphones are in one way or another linked to computer languages. Writing computer languages is a complex task that requires the author to possess diverse expertise at mathematical, logical, and often business levels. The author must have experience with the syntactic, semantic, and logical rules of various languages. The aim of this thesis is to provide visibility into the researched knowledge regarding errors occurring in computer language writing, using the method of systematic literature mapping. There is no previous survey available on this topic. The review examines the amount of available information, where information can be found, and how the amount of information has evolved over time. Although the amount of information in the field has increased over the past decade, it is not widely available. The thesis shows that the available information mostly focuses on learning and teaching computer languages. Further research on the topic can provide a wealth of new information on teaching different skills required for writing computer languages. On the other hand, broader research in this area can facilitate the transition of novice writers into the workforce and, in turn, help companies recruit more skilled professionals.

Keywords: systematic literature mapping, computer languages, typing errors, error messages

KUVIOT

Kuvio 1: Käytetyimmät ohjelmointikielet ammattikehittäjien keskuudessa	8
Kuvio 2: SQL-kielen jaottelu	9
Kuvio 3: Systemaattisen kirjallisuuskartoituksen prosessi	13
Kuvio 4. Lumipallo -menetelmä	20
Kuvio 5: Artikkeleiden valintaprosessi	23
Kuvio 6: Hyväksytyjen artikkeleiden avainsanat	26
Kuvio 7: Hyväksytyjen artikkeleiden määrä vuosittain	27
Kuvio 8: Hyväksytyjen artikkeleiden määrä julkaisukanavittain	28

TAULUKOT

Taulukko 1: Esimerkki systemaattisen kirjallisuuskartoituksen luokittelusta ...	14
Taulukko 2: Artikkeleiden hyväksymiskriteerit	22
Taulukko 3: Hyväksytyjen artikkeleiden 10 yleisintä avainsanaa.	24
Taulukko 4: Hyväksytyjen artikkeleiden määrä tietokannoittain.	27

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO.....	6
2	TIETOKONEKIELET	7
2.1	Ohjelmointikielet	7
2.2	SQL.....	8
2.3	Merkintäkielet	9
2.4	Ohjelmointikielien oppiminen.....	9
2.5	Virheen ja suoritusvirheen määritelmä	10
2.6	Syntaktiset ja semanttiset virheet	11
3	SYSTEMAATTISET KIRJALLISUUSKATSAUKSET JA -KARTOITUKSET	
	12	
3.1	Systemaattinen kirjallisuuskatsaus	12
3.2	Systemaattinen kirjallisuuskartoitus.....	12
3.3	Systemaattisten menetelmien erot	15
4	TUTKIMUKSEN TOTEUTUS.....	16
4.1	Tutkimuskysymykset.....	16
4.2	Tietolähteet	17
4.3	Käytetyt hakutermit	17
4.4	Lumipallo -menetelmä.....	19
	4.4.1 Backward snowballing	20
	4.4.2 Forward snowballing.....	21
4.5	Valintakriteerit	21
4.6	Aineiston hallinta ja luokittelu	22
5	TULOKSET.....	24
6	POHDINTA	29
6.1	Tutkimuksen arviointi ja rajoitteet.....	30
6.2	Mahdollisuudet jatkotutkimukselle.....	30
7	JOHTOPÄÄTÖKSET	ERROR! BOOKMARK NOT DEFINED.
	LÄHTEET	33
	LIITE 1 ENSIMMÄINEN LIITE.....	35
	LIITE 2 TOINEN LIITE	44

1 JOHDANTO

Tietokonekielten kirjoittaminen tai ohjelmointi on monimutkainen tehtävä monestakin eri näkökulmasta. Ohjelmoijassa kirjoittajan tulee hallita sekä kirjoitetun kielen syntaktinen kielioppi, että myös kielen semanttiset sekä loogiset säännökset. Tietokonekieliä kirjoittaessa tehdyt inhimilliset virheet ovat yleisiä erityisesti aloittelijatasolla ja tämän ilmiön vaikutuksista tietoa on saatavilla vähäisesti (Bosse & Gerosa, 2017). Tutkimuksia aloittelijoiden ja noviisien kyvystä oppia tietokonekieli on olemassa verrattaen enemmän (Kadar ym., 2021). Tämän tutkimuksen tarkoituksena onkin esitellä, kuinka paljon tutkittua tietoa tietokonekielten kirjoittamiseen liittyvistä virheistä on saatavilla sekä missä ja milloin tämä tieto on julkaistu. Tässä tutkimuksessa käytetään systemaattisen kirjallisuuskartoituksen menetelmää, jonka tarkoituksena on saada mahdollisimman laaja kuva aihealueen tutkitun tiedon määrän nykytilanteesta. Valitusta aihealueesta ei ole olemassa aikaisemmin teetettyä kartoitusta.

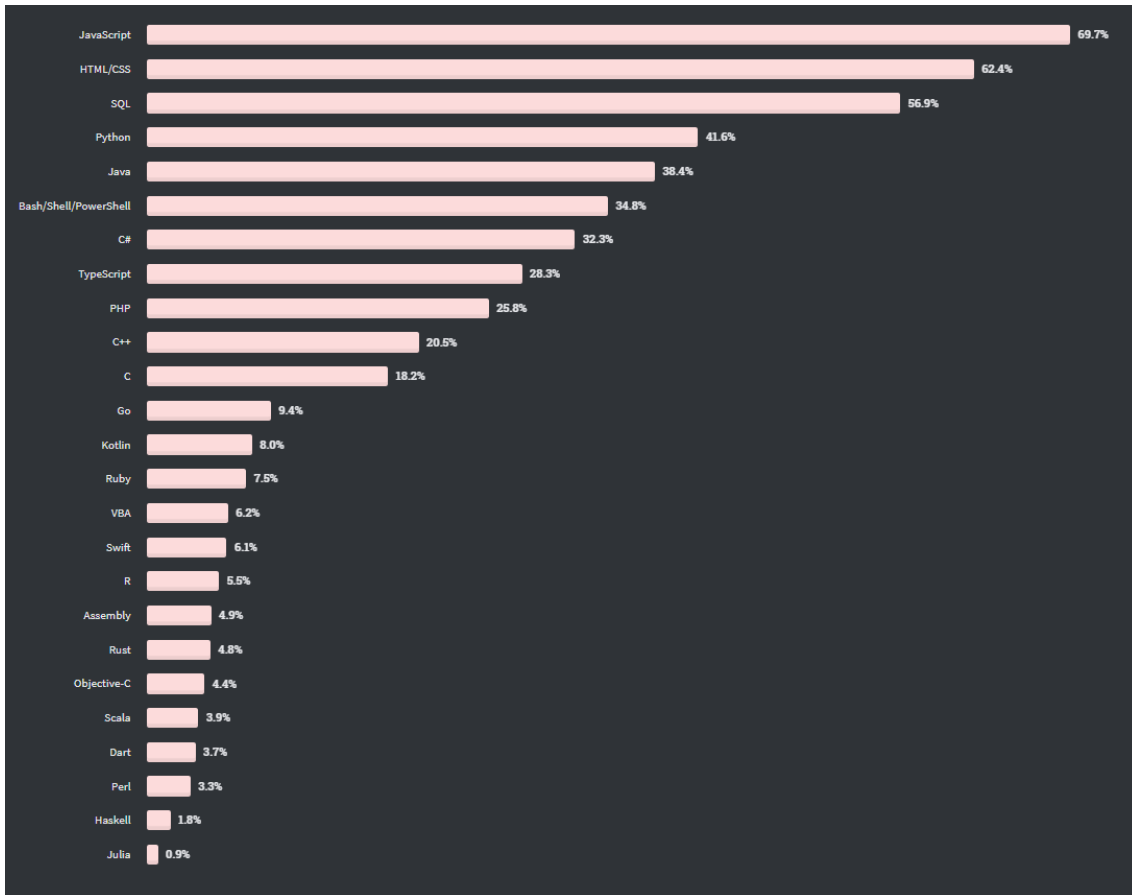
Tutkimuksen luvussa kaksi esitellään tutkimuksessa käytetty teoreettinen viitekehys, missä kerrotaan tarkemmin käsiteltävistä termeistä ja teemoista. Luvussa kolme pureudutaan syvemmin systemaattisten tutkimusmenetelmien pariin, missä käydään läpi systemaattisen kirjallisuuskatsauksen ja systemaattisen kirjallisuuskartoituksen menetelmät. Luvussa neljä esitetään tämän tutkimuksen toteutuksen kulku ja siinä käytetyt systemaattiset menetelmät. Luvussa viisi esitetään tutkimuksen tulokset ja analysointi. Tutkimuksessa käytettyyn aineistoon valikoitui kaikkiaan 43 artikkelia, jotka käsittelevät tavalla tai toisella tietokonekielten kirjoittamisessa tapahtuvia virheitä. Tuloksista voidaan havaita, että kyseisen aiheen tutkiminen on yleistynyt viimeisen vuosikymmenen aikana, vaikkakin edelleen merkittävä osa tutkitusta tiedosta kohdistuu aloittelijoiden tai noviisien kykyyn oppia tietokonekieliä, missä yhtenä osana on tietokonekieliä kirjoittaessa syntyvät virheet. Kuudennessa luvussa käsitellään tutkimuksen tuloksista syntynyttä pohdintaa ja mahdollisia jatkotutkimusaiheita. Viimeisessä luvussa esitetään tutkimuksen johtopäätökset.

2 TIETOKONEKIELET

Yksinkertaisesti ilmaistuna tietokonekielet ovat notaatioita, joiden avulla tietokoneelle kerrotaan mitä tietokoneen kuuluu tehdä seuraavaksi. Määritelmän hienovaraiset piirteet voivat kuitenkin vaikuttaa siihen, miten merkitys voi muuttua. Määritelmän mukaan tietokoneelle ”kerrotaan”, mitä tehdä, joka viittaa siihen, että tietokoneen pitää myös ”kuunnella”. (Ko, 2016, s. 32). Vaikkakin tämänkaltaisessa viestimissuhteessa toinen osapuoli on eloton, on kyseessä kuitenkin kahden entiteetin välinen kommunikoinnin muoto. Tietokone myös yleensä vastaa vastapuolelle, tässä tapauksessa ihmiselle jotain riippuen siitä, onko tietokoneelle viestitty asia kirjoitettu oikein vai väärin. Tässä luvussa on kuvailtu tarkemmin erilaisia tietokonekieliä: ohjelmointikieliä yleisesti, tietokannanhallintajärjestelmien kyselykieltä (SQL), sekä usein webkehityksessä käytettyjä merkintäkieliä (markup languages).

2.1 Ohjelmointikieliset

Kuten mikä tahansa muukin luonnollinen kieli, ohjelmointikieli koostuu kielellisistä säännöistä eli syntaksista sekä kielellisestä semantiikasta, joiden avulla voidaan muodostaa kokonaisia toimivia lauseita, joilla on jokin tietty merkitys (Schmidt, 2003). Ohjelmointikieliset jaetaan yleisesti neljään kategoriaan, jotka ovat imperatiivinen, funktionaalinen, looginen ja oliopohjainen kieli (Sebeste ym. 1999, s. 44). Kategorioiden lisäksi ohjelmointikieliset ovat jakautuneet omiksi kieliksi, kuten Java ja C++, jotka kuuluvat oliopohjaisiin kieliin (Stamouli & Huggard, 2006). Tämä jaon voi mieltää samankaltaiseksi luonnollisten kielten kieliperheiden kanssa, joissa esimerkiksi germaaniin kieliin kuuluu mm. Englannin kieli, saksan kieli, ruotsin kieli jne. (Konig & Van der Auwera, 2013, s. 1–3). Yleisimmin käytetyt ohjelmointikieliset on kuvattu kuviossa 1.



Kuvio 1: Käytetyimmät ohjelmointikielät ammattikehittäjien keskuudessa (Stack Overflow 2020 Developer Survey, 2020)

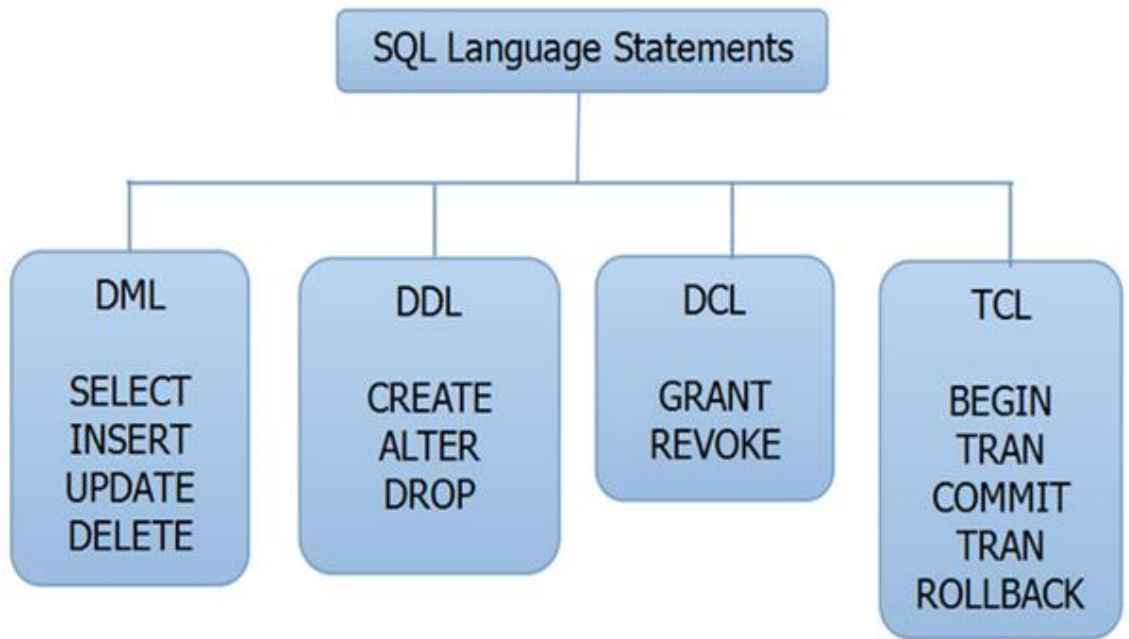
2.2 SQL

SQL (Structured Query Language) on tietokannanhallintajärjestelmiä varten luotu kyselykieli, jonka alkuperäinen versio kehitettiin IBM:llä 1970-luvun alussa ja sitä kutsuttiin alun perin nimellä Sequel ja joka perustuu Codd:n (1970) luomaan relaatiomalliin. Vuonna 1986 SQL-standardoitiin American National Standards -instituutiossa (ANSI) ja International Organization for Standardization:ssa (ISO). Tätä uutta standardia kutsuttiin SQL-86:ksi (Elmasri & Navathe 2017, s. 207–208).

SQL-kieli koostuu kahdesta pääalakielestä, jotka ovat:

1. DDL (Data Definition Language) tarjoaa erilaisia komentoja tietokantojen relaatiokaavojen määrittämiseen, relaatioiden poistamiseen tai muutosten tekemiseen relaatiokaavoihin.
2. DML (Data Manipulation Language) tarjoaa komentoja tietojen hakemiseen tietokannasta sekä tietueiden lisäämiseen, poistamiseen tai muokkaamiseen tietokannassa. (Connolly & Begg, 2005, s. 90-91)

SQL-kielestä löytyy vielä kaksi lisälakieltä, joita ei ole määritelty virallisessa SQL-standardissa, mutta jotka molemmat pitävät sisällään hyödyllisiä toiminnallisuuksia. Nämä ovat DCL (Data Control Language) ja TCL (Transaction Control Language) (Taipalus & Seppänen, 2020).



Kuvio 2: SQL-kielen jaottelu

2.3 Merkintäkielet

Merkintäkielet kuten yleisimpänä HTML on usein web-suunnittelussa käytetty merkintäkieli. Merkintäkieliä käytetään pääsäännöllisesti web-suunnitteluun, eli verkossa käytettävien sivustojen vuorovaikutussuunnitteluun (Interaction Design, IxD), käyttöliittymäsuunnitteluun (User Interface, UI) ja käyttäjäkokemussuunnitteluun (User Experience, UX) (Robbins, 2012, s. 5). Vaikkakin merkintäkielet tyyliltään muistuttavat ohjelmointikieliltä, eivät ne ole virallisesti ohjelmointikieliä, eikä niitä voi vertailla muiden ohjelmointikielien kanssa (Sebeste ym., 1999, s. 45). Merkintäkielillä on kuitenkin olemassa oma syntaksi ja niiden parissa työskentely voi muistuttaa paljonkin ohjelmointia ja tästä syystä merkintäkielet on sisällytetty myös yhdeksi tämän tutkimuksen temaksi.

2.4 Ohjelmointikielien oppiminen

Ohjelmoinnin oppiminen voi tapahtua monella tavalla, mutta yleisesti parhaiten tunnetut oppimismenetelmät jakautuvat kahteen kategoriaan, joita ovat

syvälinen ja pinnallinen oppiminen. Syvät oppijat keskittyvät saavuttamaan syvämmän kokonaisymmärryksen opittavasta asiasta, kun taas pinnalliset oppijat keskittyvät yksittäisten asioiden muistamiseen. On hankalaa määrittää kumpi näistä oppimistavoista ohjelmoinnin kannalta on tehokkaampaa ja optimaalinen strategia näyttääkin olevan molempien ääripäiden puolivälissä. Oppimistavasta huolimatta oppimista määrittää kuitenkin ensisijaisesti motivaatio opiskeltavaa asiaa kohtaan. Joillakin voi olla luonnostaan aito kiinnostus opiskeltavaa aihetta kohtaan (sisäinen motivaatio). Jotkut taas voivat nähdä jonkin taidon oppimisen edistävänä tekijänä vaikkapa tulevaa uraa ajatellen (ulkoinen motivaatio). Motivaation muoto näyttää olevankin vaikuttava tekijä myös ohjelmoinnin oppimiseen ja oppijat, jotka omaavat ainoastaan ulkoista motivaatiota, ovat useammin niitä, jotka kokevat ohjelmoinnin oppimisen hankalaksi. (Jenkins, 2002, s. 54).

Kokenut ohjelmoija taitaa monenlaisia eri taitoja. Näitä taitoja ovat esimerkiksi ongelmanratkaisutaito, jonka lisäksi vaaditaan jonkinlaista ymmärrystä eri matemaattisista tekijöistä, joita ohjelmakehitys vaatii. Ohjelmoijan tulee myös osata käsitellä tietokonetta sujuvasti. Ohjelmoinnissa ei toisin sanoen ole kyse vain yhdestä taidosta vaan sarjasta erilaisia taitoja, joita ohjelmoijan pitää kurinalaisesti harjoitella ennen kuin osaa tuottaa toimivia ohjelmia. Ohjelmoinnissa on kyse prosessista, jonka lopputuloksena syntyy toimiva ohjelma. Yksinkertaisuudessaan tämä tarkoittaa jonkin tosielämän ongelman määrittämistä, sen tulkitsemista ja muuttamista algoritmiksi, jonka perusteella lopullinen ohjelma vasta kirjoitetaan (Jenkins, 2002, s. 55).

2.5 Virheen ja suoritusvirheen määritelmä

Tässä kappaleessa kuvatut määritelmät virheestä ja suoritusvirheestä pohjautuvat molemmat luonnollisten kielten kieliopillisiin virheisiin ja suoritusvirheisiin. Tämä lähestymistapa on valikoitu ajatellen tietokonekieliä kirjoittaessa tehtyjen virheiden samankaltaisuutta luonnollisten kielten kanssa.

Virheen määritelmässä on hyvä ottaa huomioon virheen (error) ja virheellisen suorituksen (mistake) erot. Brownin (2000, s. 226) mukaan virheellinen suoritus tarkoittaa suoritusvirhettä, eli sitä, että tunnettua järjestelmää ei käytetä oikealla tavalla. Usein tämänkaltaiset virheet ovat enemmänkin lipsahduksia, jotka kokenut suorittaja pystyy havainnoimaan jo virhettä tehtäessä ja myös korjaamaan tehdyn virheen pikaisesti (tämän tutkimuksen kontekstissa tietokonekielten kielioppia tai syntaksia). Selkeäksi virheeksi taas voidaan määritellä huomattava poikkeama, josta paljastuu suorittajan puutteellinen taitotaso suoritettavaa toimintoa kohtaan. Ellisin (1997, s. 51) mukaan suoritusvirhe muodostuu, kun suorittaja ei osaa toteuttaa opittua osaamistaan esimerkiksi muistiin pohjautuvista syistä, kun taas virhe voi johtua perustuen aiempaan osaamiseen ja sitä väärin yhdistämällä, olettamalla tai väärän hypoteesin muodostamalla pohjautuen rajoittuneeseen osaamistasoon.

2.6 Syntaktiset ja semanttiset virheet

Syntaksivirhe tietojenkäsittelyn alalla tarkoittaa virhettä koodin tai ohjelmointikielen syntaksissa, joka on kirjoitettu ohjelmoijan toimesta. Syntaksivirheitä havaitaan yleisesti ohjelmistolla, jota kutsutaan kääntäjäksi (compiler). Kääntäjä havaitsee kirjoitetun virheen ja antaa viestin, jonka perusteella ohjelmoija voi havaita tekemänsä virheen (Techopedia, 2017). Toisin sanoen syntaksivirheet muodostuvat, kun valittua kieltä käytetään väärällä tavalla. Esimerkkinä syntaksivirheestä voidaan pitää vaikkapa puuttuvaa pilkkua jonkin tietyn komennon jälkeen, jolloin kääntäjä ei osaa tulkita kirjoitettua koodia oikein ja antaa tästä syystä syntaksivirheen. Syntaksivirheitä ei tule sekoittaa loogisiin virheisiin. Semanttiset virheet tai loogiset virheet ovat huomattavasti hankalampia havaita, koska ne eivät välttämättä tuota minkäänlaista näkyvää virhettä. Looginen virhe syntyy yleisesti ohjelmoijan järjestyksen seurauksena eikä ohjelmointikielen virheenä (Ataway, 2013, s. 221). Tästä esimerkkinä voidaan käyttää vaikkapa väärää keroainta, kun celsiusasteita yritetään muuttaa fahrenheitiksi. Kirjoitettu ohjelma saattaa toimia tässä tapauksessa syntaksinmukaisesti oikein, mutta palauttaa ajettaessa silti väärän tuloksen.

3 SYSTEMAATTISET KIRJALLISUUSKATSAUKSET JA -KARTOITUKSET

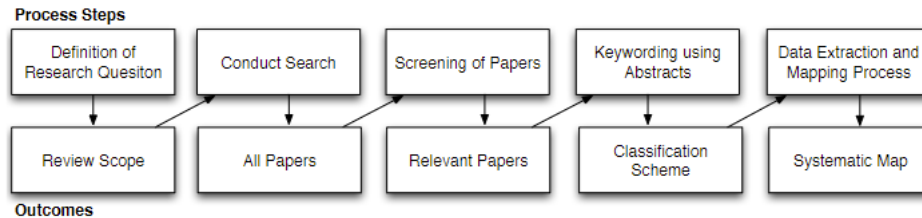
3.1 Systemaattinen kirjallisuuskatsaus

Systemaattinen kirjallisuuskatsaus on yksi toisen asteen tutkimusmenetelmistä, joka on viimeisten vuosien aikana kerännyt paljon huomiota ja on polveutunut lääketieteen alalta. Lyhyesti kuvattuna systemaattinen kirjallisuuskatsaus koostaa yhteen aikaisemmin tehdyt tutkimukset, koostaa ne yhdeksi ja kuvaa tutkimusmenetelmät sekä tutkimusten tulokset (Petersen, 2008, s. 1). Systemaattisen kirjallisuuskatsauksen höydyiksi on kuvattu hyvin määritellyn menetelmän puolueettomuutta, perinteiseen kirjallisuuskatsaukseen verrattaen laajemmat tilanteet ja kontekstit voivat mahdollistaa enemmän johtopäätöksiä ja tilastollisten meta-analyysien kautta voidaan havaita helpommin eristäytyneitä, vaikeammin tavoitettavia tutkielmia valitun aiheen parista. Heikkouksina taas voidaan nähdä huomattavaa vaivaa vaativa tutkimusmenettely.

3.2 Systemaattinen kirjallisuuskartoitus

Systemaattinen kirjallisuuskartoitus taas kuvaa valitun aiheen parista löytynyttä tutkimusta kategorisoimalla tutkimukset isommiksi teemoiksi tai kokonaisuuksiksi, joka antaa usein visuaalisen tiivistelmän tai kartan tutkittavan aiheen tämänhetkisestä tiedon määrästä. Tämän kaltainen kartoitus on yleensä vähemmän vaivalloinen kuin systemaattinen kirjallisuuskatsaus ja tarjoaa karkeamman yleiskuvan tutkittavasta aiheesta (Petersen, 2008, s. 2). Mikäli alustavan katsauksen aloituksessa käy ilmi, että tutkittavasta aiheesta tietoa on saatavilla vähäisesti tai tutkittava aihe on erityisen laaja, on hyvä käyttää menetelmänä systemaattista kirjallisuuskartoitusta systemaattisen kirjallisuuskatsauksen sijaan. (Kitchenham

& Charters, 2007, s 4-5). Menetelmä mahdollistaa tutkittavan aiheen tarkastelun korkealla tarkkuustasolla. Menetelmän avulla voidaan myös tunnistaa alueet, joilla tarvitaan lisää alkuperäistutkimusta. (Kitchenham & Charters, 2007, s. 5). Systemaattinen kirjallisuuskartoituksen toteuttamiseksi voi käyttää esimerkiksi Petersenin (2008) menetelmää, jota on käytetty myös tämän tutkielman pohjana. Prosessi on kuvattuna kuviossa 3 ja selostettuna kuvion alla.



Kuvio 3: Systemaattisen kirjallisuuskartoituksen prosessi (Petersen ym., 2008, s. 2)

Petersenin (2008) mukaan systemaattista kirjallisuuskartoitus prosessia voi kuvastaa seuraavalla tavalla:

1. Tutkimuskysymysten määrittely (Tutkimuksen laajuus)

Tutkimuksen pääasiallinen tavoite on tarjota tarkka näkemys siitä, kuinka paljon valitusta aiheesta on saatavilla tutkittua tietoa ja minkälaisilla menetelmillä tutkimukset on suoritettu. Toisena tavoitteena on saada selville, missä eri julkaisukanavissa nämä kartoitetut tutkimukset on julkaistu.

2. Alkuperäistutkimusaineiston haku

Alkuperäisen aineiston haku suoritetaan valittuihin tieteellisiin tietokantoihin, esimerkiksi toimialan perusteella. Hakuun käytetään erilaisia hakusanoja, jotka kuvastavat tutkittavaa aihetta, ja joissa tutkimuskysymykset on otettu huomioon.

3. Tutkimusaineiston seulonta

Tässä vaiheessa luodaan kriteerit, joidenka perusteella tutkimukset joko sisällytetään kartoitukseen tai poissuljetaan kartoituksesta.

4. Tutkimusten avainsanoitus sekä luokittelu

Avainsanojen avulla tutkimukset ovat vaivattomampaa kategorisoida sekä luokitella. Kategorisoinnin ja luokittelun avulla myös kartan systemaattinen muodostaminen helpottuu (Petersen ym., 2008, s. 3–4) Tämän lisäksi tutkimukseen perehtyminen on myös tutkimuksen ulkopuoliselle lukijalle helpompaa. Tutkimusten luokittelun voi toteuttaa esimerkiksi Wieringan ym. Menetelmällä, joka on esitelty taulukossa 1.

Taulukko 1: Esimerkki systemaattisen kirjallisuuskartoituksen luokittelusta (Wieringa ym. 2006)

Kategoria	Kuvaus
Validointitutkimus	Tutkittavat tekniikat ovat uusia ja ne eivät ole vielä käytössä käytännössä. Käytetyt tekniikat ovat esimerkiksi kokeiluja, jotka toteutetaan laboratoriossa.
Arviointitutkimus	Tutkittavat tekniikat ovat jo käytössä ja tekniikka on arvioitu. Toisin sanoen tekniikalla on näyttöä käytännön toimivuudesta (ratkaisun toteutus) ja toteutuksen seurauksesta syntyneet hyödyt ja haitat on arvioitu (toteutuksen arviointi). Tähän sisältyy myös ongelmakohtien havainnointi toimialakohtaisesti.
Ratkaisuehdotus	Suora ehdotus esitettyyn ongelmaan. Ratkaisu voi olla joko täysin uudenlainen tai merkittävä laajenus olemassa olevaan tekniikkaan nähden. Ratkaisun mahdollisia hyötyjä ja soveltuvuutta esitetään pienien esimerkkien kautta tai vahvan argumentoinnin avulla.
Filosofiset artikkelit	Nämä artikkelit hahmottelevat uuden tavan lähestyä olemassa olevia aiheita jäsentämällä alaa taksonomian tai käsitteellisen viitekehyksen muodossa.
Mielipideartikkelit	Nämä artikkelit ilmaisevat jonkun henkilökohtaisen mielipiteen, onko käytetty tekniikka hyvä tai huono, tai miten asia olisi tullut ratkaista. Artikkelit eivät nojaa mihinkään olemassa olevaan työhön ja tutkimusmenetelmiin.
Kokemukseen perustuvat artikkelit	Nämä artikkelit selittävät, miten jokin on tehty käytännössä. Nämä perustuvat kirjoittajan henkilökohtaisiin kokemuksiin.

5. Varsinaisen aineiston keruu (kartoitus)

Avainsanoituksen ja luokittelun jälkeen varsinaisen aineistonkeruu aloitetaan. Tässä vaiheessa voidaan vielä laajentaa tai supistaa valittuja luokitteluja sekä kategorioita. Kun aineisto on kerätty, luokiteltu ja kategorisoitu, voidaan tulosten analysointi aloittaa. Tulosten analysoinnin avulla voidaan havaita tutkittavan aihealueen laajuutta ja mahdollisuuksia tulevia tutkimuksia varten.

3.3 Systemaattisten menetelmien erot

Systemaattiset kirjallisuuskartoitukset on suunniteltu tarjoamaan laaja ylätason katsaus tutkittavaan aihealueeseen, jonka tavoitteena esittää, onko aihealueesta tutkittua tietoa ja missä määrin tietoa on saatavilla. Kartoitusmenetelmää voidaan käyttää systemaattisen kirjallisuuskatsauksen edeltävänä toimenpiteenä, jos ei olla varmoja tutkittavan aihealueen laajuudesta. Kartoituksia voidaan hyödyntää myös esimerkiksi väitöskirjatyöskentelyn tukena, jolloin tutkija voi hankkia itselleen korkean tason ymmärryksen jostakin tutkittavasta aiheesta ennen varsinaisen työn aloitusta. Kitchenhamin & Chartersin (2007) mukaan systemaattiset tutkimusmenetelmät eroavat toisistaan seuraavalla tavalla:

- Kartoitustutkimuksissa tutkimuskysymykset ovat yleensä laajempia.
- Kartoitustutkimusten hakutermit ovat vähemmän spesifejä kuin systemaattisessa kirjallisuuskatsauksessa, jolloin kartoitukset todennäköisemmin tuottavat laajemman lopputuloksen. Kartoituksen tavoite onkin luoda laaja kuva tutkittavasta aiheesta sen sijaan, että saataisiin tarkka näkemys jostain kapeammasta näkökulmasta.
- Tiedonkeruuprosessi kartoitustutkimuksissa on myös laajempi, jonka tuloksena saadaan yleensä paljon dataa. Kartoitustutkimusta voidaanakin kuvailla enemmän kategorisoivana ja luokittelevana toimenpiteenä, jota voidaan myöhemmin hyödyntää tarkemman kirjallisuuskatsauksen tukena.
- Kartoitusmenetelmän analysointivaiheessa on kyse kerätyn tiedon tiivistämisestä, jonka avulla voidaan vastata esitettyihin tutkimuskysymyksiin. Tämän vuoksi kartoitustutkimuksien analyysit eivät yleensä ole niin syväluotaavia kuin kirjallisuuskatsauksissa.
- Kirjallisuuskartoitusten tulosten levittäminen voi olla rajoitetumpaa kuin kirjallisuuskatsauksissa. Kartoituksien tavoite on vaikuttaa tulevien alkuperäistutkimusten suuntaan.

4 TUTKIMUKSEN TOTEUTUS

Tämä tutkimus on toteutettu systemaattisen kirjallisuuskartoituksen menetelmällä ja tutkimusta tehdessä on hyödynnetty Petersenin ym. (2008), Kitchenhamin & Chartersin (2007) sekä Taipaluksen (2023) kirjallisuuskartoitusmenetelmiä. Systemaattisen kirjallisuuskartoituksen päätavoite on tarjota yleiskatsaus jostakin tutkittavasta aiheesta, jonka avulla voidaan saada näkymä tutkittavan aiheen laajuudesta, laadusta sekä tehtävän tutkimuksen trendeistä. Toinen tavoite on tunnistaa eri alustat, joissa tutkittavasta aiheesta olevat tutkimukset on julkaistu. Nämä tavoitteet määrittelevät myös systemaattisen kirjallisuuskartoituksen tutkimuskysymyksiä (Petersen ym., 2008).

4.1 Tutkimuskysymykset

Tutkimuskysymysten muodostaminen on systemaattisen tutkimuksen yksi tärkeimmistä vaiheista, koska tutkimuskysymysten määrittely ohjaa lopulta koko tutkimuksen metodologiaa (Kitchenham & Charters, 2007). Tutkimuskysymykset voidaan systemaattisen kirjallisuuskartoituksen osalta toteuttaa kahdella tavalla. Tutkimuskysymyksiä määriteltäessä voidaan joko valita kysymyksiä, jotka ovat tyypillisiä tämän kaltaisen tutkimuksen tekemiseen. Nämä kysymykset ovat valitun tutkimuksen kartoittaminen esimerkiksi julkaisuvuosien, julkaisupaikkojen, lähestymistavan tai teeman mukaan. Toinen vaihtoehto on luoda kysymykset tieteenalaan pohjautuen (Taipalus, 2023, s.3). Tämän tutkimuksen tutkimuskysymykset on muotoiltu ensimmäistä vaihtoehtoa mukaillen. Tämän tutkimuksen tavoitteena on kartoittaa, kuinka paljon tietoa tietokonekielten kirjoittamisessa tapahtuvista virheistä löytyy ja tämän tavoitteen ympärille muodostui myös tutkimuksen tutkimuskysymykset. Kysymysten luonnilla ei millään muotoa pyritty rajaamaan saatavia hakutuloksia, koska jo tutkimusta aloittaessa oli tieto siitä, että aiheen parista löytyy verrattaen vähän tutkittua tietoa. Tutkimuksen aikana kerätyllä aineistolla ja niistä koostetulla analyysillä vastataan alla luetuihin tutkimuskysymyksiin.

Tutkimuskysymykset:

1. Kuinka laajasti tieteellisessä kirjallisuudessa on tutkittu tietokonekielten kirjoittamisessa tapahtuvia virheitä?
2. Mitkä ovat tietokonekielten kirjoittamisessa tapahtuvia virheitä käsittelevien tutkimusten julkaisukanavat?
3. Mitkä ovat tietokonekielten kirjoittamisessa tapahtuvia virheitä käsittelevien tutkimusten julkaisutrendit?

4.2 Tietolähteet

Tutkimukseen käytetyt tietokannat valittiin tietotarpeeseen pohjautuen niin, että ensin suoritettiin karsintahaut useampaan tietokantaan, jotka lähtökohtaisesti olivat havaittu sisältävän paljon tietotekniikkaan liittyviä julkaisuja. Tämän jälkeen tehtiin valinta perustuen hakutermeillä löydettyjen artikkelien määrään. Karsintahaussa olivat mukana ACM Digital Library, CiteSeerX, Google Scholar, IEEE Explorer, ISI Web of Science ja Science Direct. Karsinnan perusteella näistä lähteistä poistettiin CiteSeerX ja ISI Web of Science puutteellisista tai toisteisista hakutuloksista johtuen.

4.3 Käytetyt hakutermit

Haut valituilla termeillä toteutettiin aikavälillä 21.2.2023 – 1.3.2023. Ensimmäisillä hakukerroilla, haku toteutettiin ainoastaan koskemaan SQL-kieleen liittyviä julkaisuja, tulokset jäivät kuitenkin niin vähäisiksi, ettei kunnollista kartoitusta niiden pohjalta voitu tehdä. Tästä syystä haku laajennettiin koskettamaan muitakin tietokonekieliä sekä web-kehityskieliä. Käytössä olevat hakutermit ovat hyvin spesifejä, jotta tulokseksi saatiin todellinen kuva tutkimuskysymyksiin vastaavien julkaisujen määrästä. Hakutermin laajennuksen jälkeenkin julkaisujen määrä jäi kohtuullisen maltilliseksi. Tässä vaiheessa voidaan todeta, että tutkittavasta aiheesta tietoa on saatavilla melko vähäisesti. Alun perin hakusanat suunniteltiin vastaamaan ainoastaan tietokannanhallintajärjestelmissä käytettävien kyselykielten pohjalta, mutta niin kuin tämän kappaleen alussa todettiin, näillä hakutuloksilla ei saatu tarpeeksi laajaa lopputulosta. Artikkeleiden hakemisessa käytettiin sekä kokonaisvaltaisempia hakulauseita ja eri avainsanoista yhdisteltyjä koostelausekkeita.

Lopullisessa haussa käytetyt hakulauseet:

- "Typing mistakes in writing SQL"
- "Typing errors in writing SQL"
- "Syntax errors in writing SQL"
- "Typing mistakes in writing database management system languages"

- "Typing errors in writing database management system languages"
- "Syntax errors in writing database management system languages"
- "Typing mistakes in writing computer languages"
- "Typing errors in writing computer languages"
- "Syntax errors in writing computer languages"
- "Typing mistakes in writing web development languages"
- "Typing errors in writing web development languages"
- "Syntax errors in writing web development languages"

Tarkemmista hakusanoista yhdisteltyjä koostelauseita:

- Typing mistakes + SQL
- Typing mistakes + database management system
- Typing mistakes+ computer language
- Typing mistakes + web development
- Typing + error + SQL
- Typing + error + database management system
- Typing + error + computer language
- Typing + error + web development
- Syntax + error + SQL
- Syntax + error + database management system
- Syntax + error + computer language
- Syntax + error + web development

Tietokannoittain (hakulauseet):

ACM Digital Library

1. Typing mistakes in writing SQL" OR "Typing mistakes in writing computer languages" OR "Typing mistakes in writing web development languages"
2. "Typing errors in writing SQL language" OR "Typing errors in writing computer languages" OR "Typing errors in writing web development languages"
3. "Syntax errors in writing SQL language" OR "Syntax errors in writing computer languages" OR "Syntax errors in writing web development languages"

Google Scholar

1. Typing mistakes in writing SQL language" OR "Typing mistakes in writing computer languages" OR "Typing mistakes in writing web development languages"
2. "Typing errors in writing SQL language" OR "Typing errors in writing computer languages" OR "Typing errors in writing web development languages"
3. "Syntax errors in writing SQL language" OR "Syntax errors in writing computer languages" OR "Syntax errors in writing web development languages"

IEEEExplorer

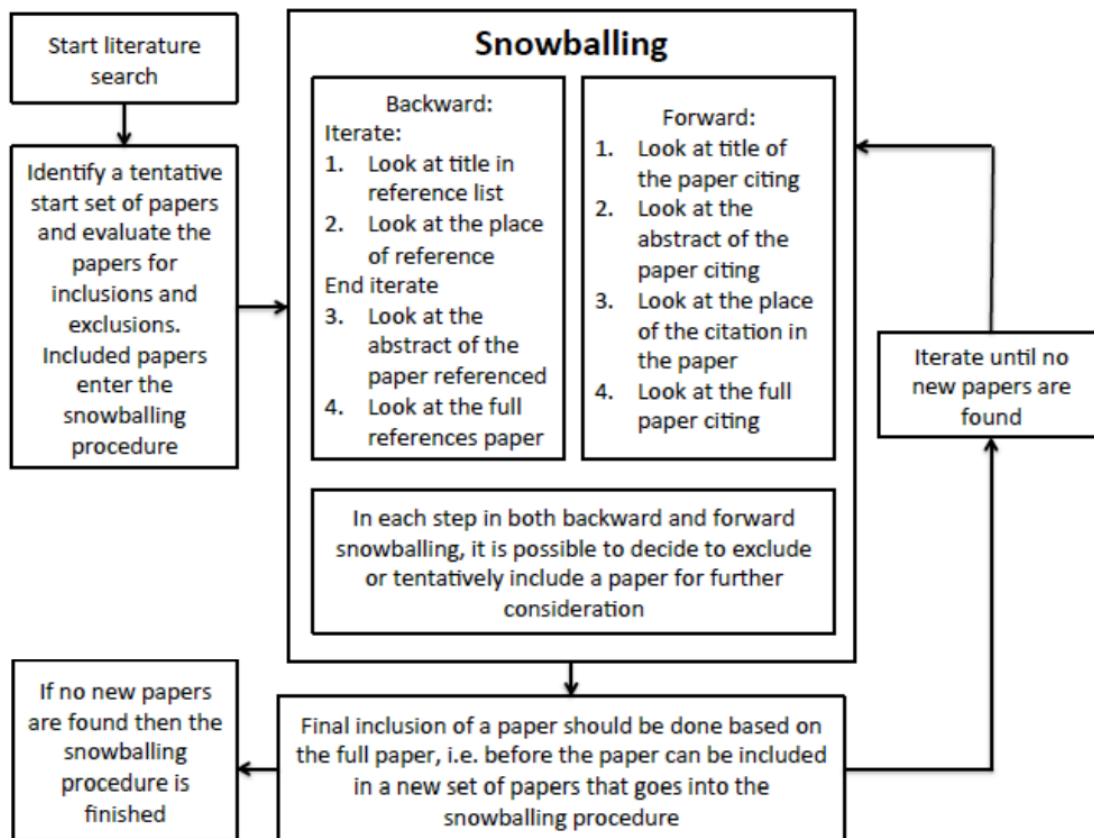
1. "Typing mistakes in writing SQL language" OR "Typing mistakes in writing computer languages" OR "Typing mistakes in writing web development languages"
2. "Typing errors in writing SQL language" OR "Typing errors in writing computer languages" OR "Typing errors in writing web development languages"
3. "Syntax errors in writing SQL language" OR "Syntax errors in writing computer languages" OR "Syntax errors in writing web development languages"

ScienceDirect

1. "Typing mistakes in writing SQL language" OR "Typing mistakes in writing computer languages" OR "Typing mistakes in writing web development languages"
2. "Typing errors in writing SQL language" OR "Typing errors in writing computer languages" OR "Typing errors in writing web development languages"
3. "Syntax errors in writing SQL language" OR "Syntax errors in writing computer languages" OR "Syntax errors in writing web development languages"

4.4 Lumipallo -menetelmä

Sana- ja lausehakujen tuloksena saatujen artikkelien määrä jäi suppeaksi, joten tämän tutkimuksen tekemisessä päätettiin käyttää lumipallo -menetelmää. Lumipallo -menetelmällä (Snowball procedure) tarkoitetaan menetelmää, jossa tutkimuksen aikana kerättyihin lähdetutkimuksiin suoritetaan kattava aineiston laajennus. Lumipallo -menetelmän onnistumiseksi vaaditaan kuitenkin jo ennalta kerätty lähdeaineisto (Wohlin, 2014). Wohlin (2014) kuvaa lumipallo -menetelmää tiivistettynä seuraavalla tavalla:



Kuvio 4. Lumipallo -menetelmä (Wohlin 2014)

1. Ensimmäinen vaihe on tunnistaa avainsanat ja koostaa hakulausekkeet sekä suorittaa varsinainen lähdehaku eri tietokannoista.
2. Toisena vaiheena perehdytään alustaviin hakutuloksiin ja arvioidaan, mitkä tutkimukset otetaan mukaan seuraavaan vaiheeseen.
3. Seuraavaksi suoritetaan varsinainen lumipallo -menetelmä, jossa toteutetaan sekä taaksepäin suuntaava menetelmä (backward snowballing) ja eteenpäin suuntaava menetelmä (forward snowballing). Molempien menetelmien tarkempi kuvaus löytyy tämän listauksen alta.
4. Taaksepäin suuntaavaa ja eteenpäin suuntaavaa vaihetta toistetaan, kunnes uusia löydöksiä valitun tutkimusaiheen piiristä ei enää synny.
5. Kun uusia hakutuloksia ei enää löydy, on menetelmä suoritettu.

Lumipallo menetelmä on yksi johtavista menetelmistä systemaattisen kirjallisuustutkimuksen toteuttamiseksi ohjelmistotuotannon alalla (Wohlin ym., 2022, s. 2)

4.4.1 Backward snowballing

Taaksepäin suuntaavan lumipallo -menetelmän tarkoitus on selvittää mahdollisimman paljon tietoa valittujen artikkeleiden lähdeaineistosta ja tarkastaa onko artikkeli relevantti kirjallisuuskartoituksen näkökulmasta. (Wohlin, 2014, s. 3).

Toisin sanoen tämä tarkoittaa ennalta valitun lähdeaineiston sisältävien tutkimusten viiteluetteloiden tarkastelua, joista valitaan mukaan relevantit teokset (Taipalus, 2023). Menetelmän tavoitteena on perehtyä viiteluettelosta löytyvien artikkelien otsikoihin, julkaisupaikkoihin, kirjoittajiin, tiivistelmiin ja tarvittaessa myös tutkimusten leipäteksteihin ja tehdä päätös sisällytetäänkö kukin tutkimus osaksi kirjallisuuskartoitusta. (Wohlin, 2014, s. 3)

4.4.2 Forward snowballing

Eteenpäin suuntaava lumipallo -menetelmä puolestaan ottaa kantaa missä uudemmissa tutkimuksissa on viitattu alkuperäisaineiston tutkimuksiin. Kun kaikki tutkimukset, joissa viittaus ilmenee, on listattu, suoritetaan tälle aineistolle samankaltainen seulonta, kuin taaksepäin suuntaavassa lumipallo -menetelmässä (Wohlin, 2014, s. 3).

Tämän työn osalta jotkin menetelmässä mainitut tarkemmat rajaukset, kuten leipätekstiin tarkempi perehtyminen jätettiin tekemättä, koska tarkemmin rajaamalla koostettava aineisto olisi jäänyt liian suppeaksi. Rajaukset tehtiin otsikon, julkaisualustan ja tiivistelmän perusteella, mutta artikkelin sisältöä ei sen tarkemmin tarkasteltu.

4.5 Valintakriteerit

Valintakriteerejä käytetään poissulkemaan sellaisia tutkimuksia, jotka eivät ole kartoituksen kannalta olennaisia (Petersen ym. 2008, s. 3). Tässä tutkimuksessa käytettävä aineisto on itsessään jo melko kapea, joten valintakriteerit suunniteltiin niin, että aineiston määrä pysyi kohtuullisena. Tutkimuksen aikaisemmissa vaiheissa saatujen hakutulosten karsintaa tehtiin esimerkiksi tietokantoja valittaessa, jolloin muodostui selkeä rajaus siitä, miltä toimialalta julkaisuja listaukseen valitaan. Tarkemmat hyväksymiskriteerit on esitetty taulukossa 2.

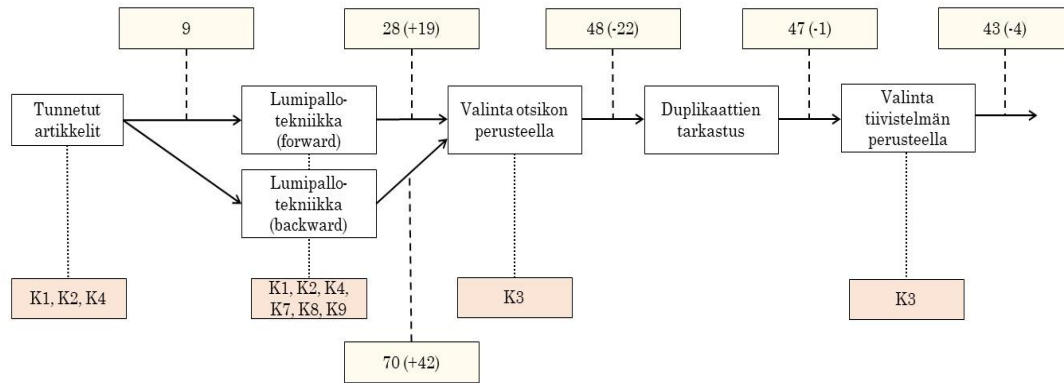
Taulukko 2: Artikkeleiden hyväksymiskriteerit

Kriteerit	Tunniste
Hyväksymiskriteerit	
Artikkeli on akateeminen	K1
Artikkeli on englanninkielinen	K2
Artikkelin otsikosta tai tiivistelmästä ilmenee, käsitteleekö artikkeli valittua aihealuetta	K3
Artikkeli on saatavilla digitaalisessa muodossa	K4
Hylkäämiskriteerit	
Artikkeli on kirjoitettu muulla kuin englannin kielellä	K5
Artikkelit, jotka eivät ole saatavilla digitaalisessa muodossa	K6
Kirjat	K7
Osittaiset artikkelit	K8
Mielipidekirjoitukset	K9

4.6 Aineiston hallinta ja luokittelu

Aineisto kerättiin yksittäiseen Excel-työkirjaan, joka jaettiin eri tauluihin. Taulut jaoteltiin perustuen aineiston suodattamisen perusteella niin, että alkuperäinen aineisto listattiin Master- tauluun, joka sisälsi kaikki havainnot tutkittavan aiheen piiristä lumipallo -menetelmän toteuttamisen jälkeen. Jokaisesta artikkelista kerättiin tieto tutkimuksen nimestä, lähdetietokannasta, tekijöistä, julkaisuvuodesta, tutkimusmenetelmästä sekä tutkimuksen avainsanoista. Tämän jälkeen aineistolle suoritettiin duplikaattien tarkastus, jonka jälkeen ensimmäinen karsinta tehtiin artikkeleiden otsikoiden perusteella. Toinen karsintakierros tehtiin artikkeleiden tiivistelmän perusteella. Tässä vaiheessa artikkeleiden määrä oli supistunut niin maltilliseksi, että jatkokarsinnan toteuttamista ei koettu tarpeelliseksi. Jokainen karsintakierros tallennettiin omaan Excel-tiluun. Lumipallo- menetelmän jälkeen jäljelle jääneet artikkelit karsittiin valintakriteerien perusteella, jonka jälkeen tutkittava lähdeaineisto todettiin valmiiksi.

Aineiston luokittelu tehtiin perustuen julkaisuvuoteen, julkaisukanavaan, lähdetietokantaan ja avainsanoihin. Avainsanoitus toteutettiin perehtymällä hyväksytyjen tutkimusten otsikoihin sekä tiivistelmiin ja määrittelemällä näiden tietojen pohjalta 3 avainsanaa, jotka kuvastavat tutkimuksen aihetta ja teemaa. Lopulliseen avainsanalistaukseen valittiin 3 avainsanaa jokaista tutkimusta kohden. Avainsanojen määräksi valikoitui 3, koska avainsanalistaus olisi muuten kasvanut tarpeettoman pitkäksi, joka olisi hankaloittanut aineiston luokittelua avainsanojen perusteella. Tämänkaltaiseen luokittelumenetelmään päädyttiin siitä syystä, että tutkimusaineisto on vähäinen ja että tarkemman luokittelun seurauksena tulosten esittely muuttuisi epäselväksi.



Kuvio 5: Artikkeleiden valintaprosessi (ks. Valintakriteerit taulukosta 2)

5 TULOKSET

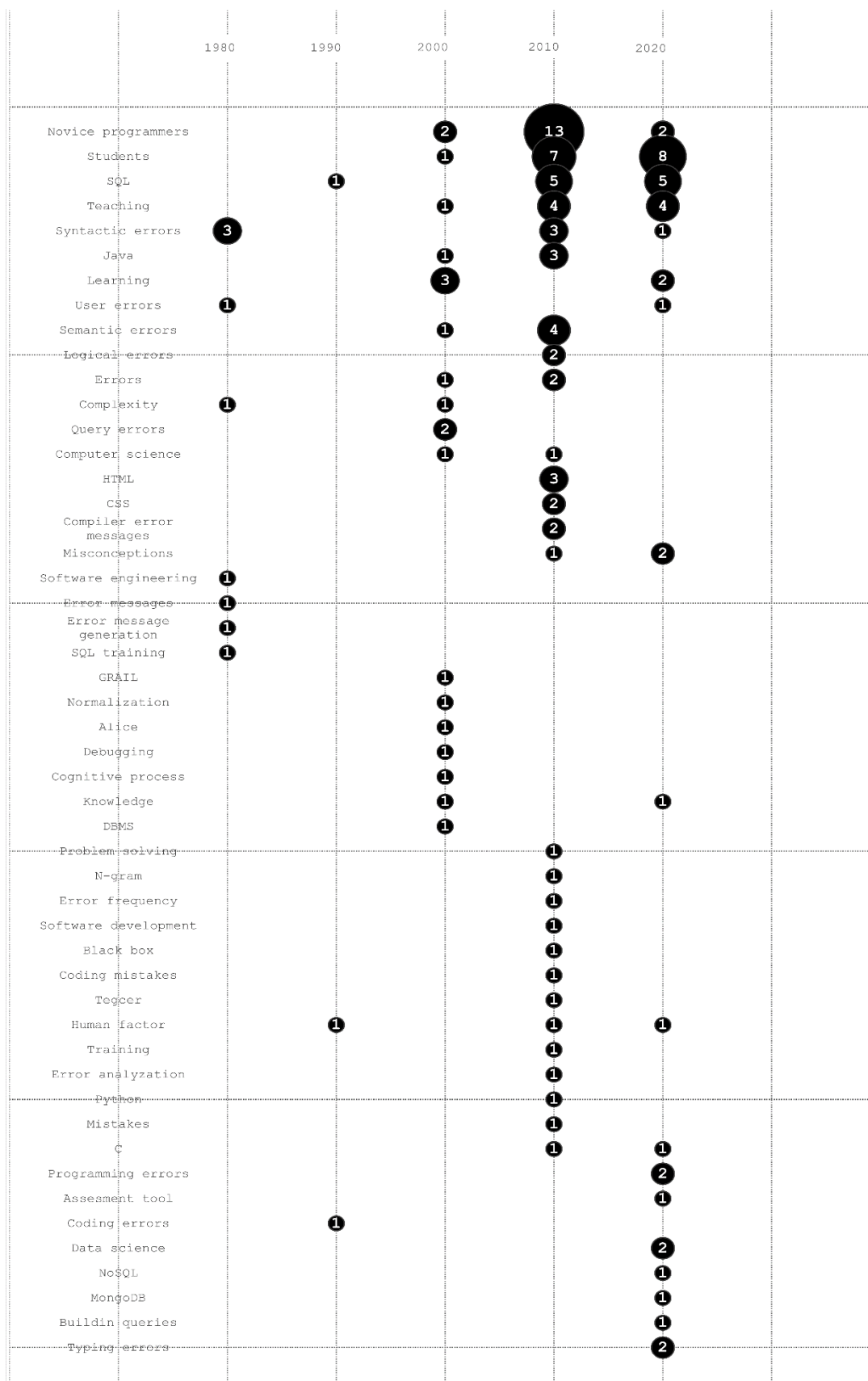
Lähdeaineisto tallennettiin ensin suodattamattomana Master -tauluun, johon ensimmäisenä listattiin lähtöaineisto. Tämän jälkeen lähtöaineistolle suoritettiin lumipallo -menetelmä (eteenpäin sekä taaksepäin suuntaava), jonka tulokset listattiin Master- taulun jatkeeksi. Lumipallo -menetelmän tuloksena listattiin yhteensä 70 artikkelia, jotka koostettiin kahden eri iteraatiokerran tuloksena. Kolmatta iteraatiota ei nähty tarpeelliseksi, koska tulokset eivät muuttuneet ensimmäisen ja toisen iteraation välillä. Lumipallo - menetelmän jälkeen aineistolle tehtiin karsinta otsikon perustella, jonka seurauksena aineisto kaventui 48 artikkeliin. Tämän vaiheen jälkeen jäljelle jäänyt aineisto käytiin vielä kertaalleen läpi ja aineistolle tehtiin duplikaattien tarkastus valitsemalla Excelin asetuksista ehdollinen muotoilu, jonka avulla duplikaattiarvot karsittiin pois. Tarkastuksen seurauksena jäljelle jäi 47 tutkimusta. Seuraava karsintakierros tehtiin perehtymällä tarkemmin artikkelien tiivistelmiin. Tämän vaiheen tuloksena havaittiin vielä 1 duplikaatti, joka ei esiintynyt ehdollisen muotoilun avulla. Tämän lisäksi havaittiin 3 artikkelia, jotka eivät käsitelleet tutkittavaa aihetta riittävällä tarkkuudella. Karsitut 3 artikkelia käsitelivät tietokonekielten syntaksivirheitä teknisemmästä näkökulmasta ja miten ongelmat ilmenevät tietokoneiden toimivuuden kannalta sen sijaan, että kyseessä olisi inhimillisestä syystä johtuva virhe. Lopulliseksi aineistoksi valikoitui 43 artikkelia, jotka ovat listattuna liitteessä 1. Kuvaus prosessin etenemisestä on esitetty kuviossa 5. Tässä kappaleessa esitellään lopullisesta artikkelilistauksesta koostetut tutkimustulokset.

Taulukko 3: Hyväksytyjen artikkeleiden avainsanojen 10 yleisintä avainsanaa. Aineistohaku toteutettiin aikavälillä 21.2.2023 - 1.3.2023.

Avainsana	Määrä	%
Novice programmers	17	12,78 %
Students	16	12,03 %
SQL	11	8,27 %
Teaching	10	7,52 %
Syntactic errors	8	6,02 %
Java	5	3,76 %

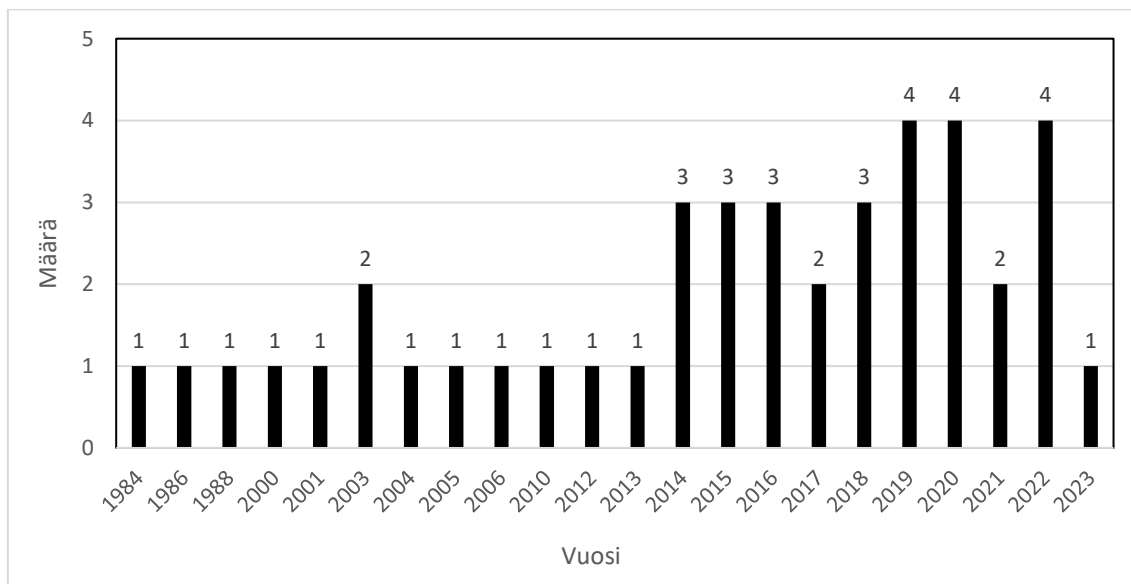
Learning	4	3,01 %
Semantic errors	4	3,01 %
Logical errors	4	3,01 %
Errors	3	2,26 %

Taulukossa 3 on kuvattu kaikkien hyväksytyjen artikkeleiden käytetympiä avainsanoja. Tästä voidaan havaita, että yleisimpiä teemoja ovat kokemattomien ohjelmoijien sekä opetukseen liittyvät teemat.



Kuvio 6: Hyväksytyjen artikkeleiden avainsanojen määrä vuosikymmenittäin. Aineistohaku toteutettiin aikavälillä 21.2.2023 – 1.3.2023.

Kuviossa 6 on esitetty hyväksytyjen artikkeleiden avainsanaston jakautumista vuosikymmenittäin. Kuviosta voidaan havaita, että suurin osa avainsanoista esiintyy artikkeleissa vain kerran. Avainsanat, jotka tavalla tai toisella liittyvät aloittelijoihin, opiskelijoihin tai opettamiseen toistuivat artikkeleiden avainsanastoissa useimmin. Teknologioihin kohdistuvista avainsanoista yleisimpiä olivat SQL ja Java.



Kuvio 7: Hyväksytyjen artikkeleiden määrä vuosittain. Aineistohaku toteutettiin aikavälillä 21.2.2023 – 1.3.2023.

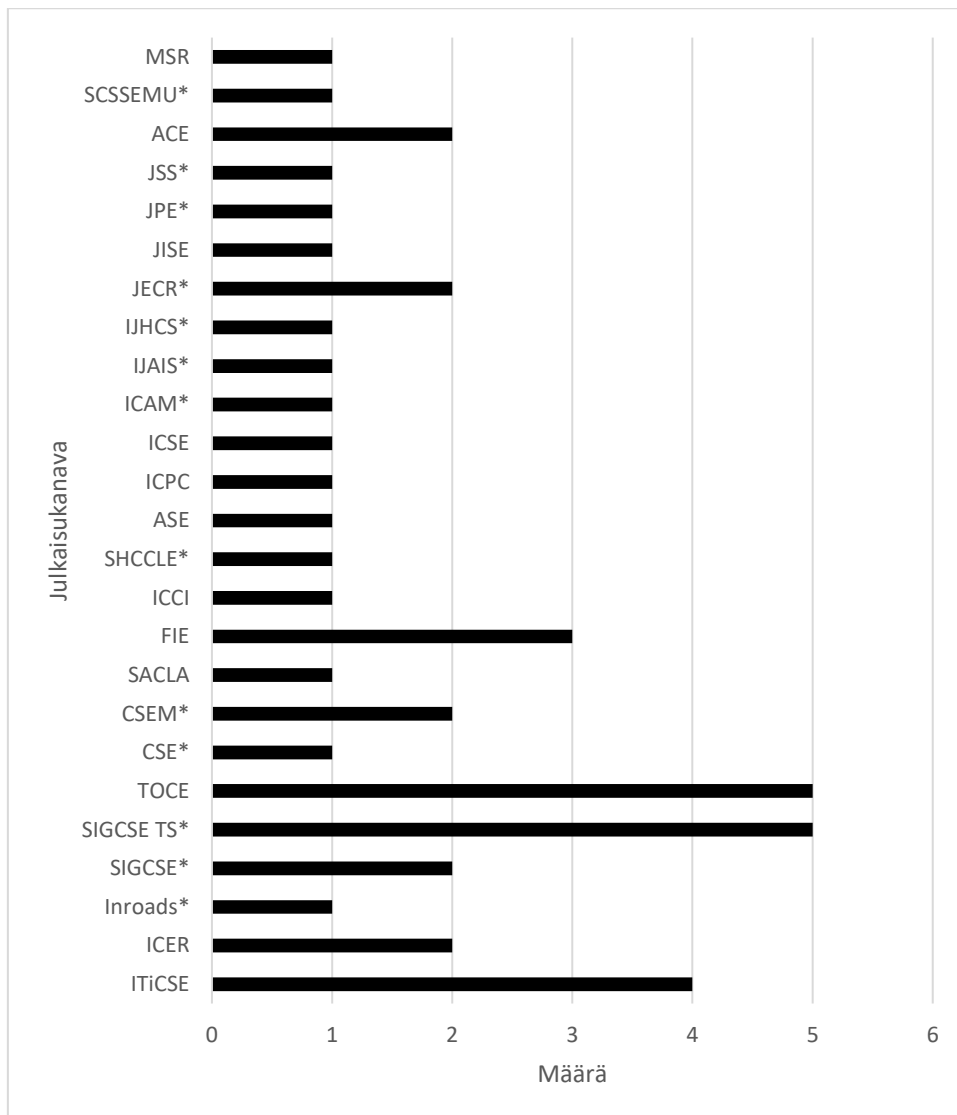
Kuviossa 7 on jaoteltu hyväksytyjen artikkeleiden määrä vuosittain. Kuviosta käy ilmi, että aiheesta on tehty tutkimusta enemmän viimeisen vuosikymmenen aikana verraten edeltäviin vuosikymmeniin. Vuoden 2010 jälkeen tutkimuksia on toteutettu lähes vuosittain, kun taas tätä edeltävänä aikana tutkimuksia on toteutettu epäsäännöllisesti.

Taulukko 4: Hyväksytyjen artikkeleiden määrä tietokannoittain. Aineistohaku toteutettiin aikavälillä 21.2.2023 – 1.3.2023.

Tietokanta	Artikkeleiden määrä
ACM Digital Library	26
Google Scholar	7
IEEE Explorer	7
ScienceDirect	3

Taulukossa 4 on esitelty hyväksytyjen artikkeleiden määrä tietokannoittain. Kaikki alun perin valituista tietokannoista eivät tuottaneet yhtäkään tulosta, joten nämä havainnot on kokonaan jätetty pois taulukosta. Suurin osa tähän tutkimukseen hyväksytyt artikkelit löytyivät Association of Computer Machinery -yhteisön digitaalisesti kirjastosta. ACM on kansainvälinen

tietotekniikan oppinut yhteisö, joka on julkaissut tietotekniikkaa koskevaa tutkittua tietoa vuodesta 1947 lähtien (ACM, 2023).



Kuvio 8: Hyväksytyjen artikkeleiden määrä julkaisukanavittain. Aineistohaku toteutettiin aikavälillä 21.2.2023 – 1.3.2023.

Kuviossa 8 on kuvailtu hyväksytyjen artikkeleiden määriä julkaisukanavoittain. Merkittävä osa eri julkaisukanavista olivat erinäisiä konferensseja. Julkaisukanavien lyhenteet on kuvailtu liitteessä 2.

6 POHDINTA

Jo tutkimusta aloittaessa kävi ilmi, että tutkittavasta aiheesta tietoa on saatavilla verrattaen vähäisesti. Tietokonekielistä itsessään löytyy paljon erilaista tietoa, mutta suurin osa tutkitusta tiedosta perustuu ennemmin teknologioiden suorituskykyyn kuin tietokonekielten kirjoittamiseen. Nykypäivänä erilaiset tietokonekieliä tulkitsevat tulkit ja konsolit ovat iso osa kehittäjien arkipäivää, mutta välillä tukeutuminen ainoastaan niiden tuottamaan virhearvioon tuottaa paljon viivettä sen sijaan, että pureudutaan ongelman juurisyyyhyn, joka yleisesti on tietokonekieltä kirjoittavan tahon aiheuttama. Tietokonekieliä kirjoittaessa on yleistä, että kirjoittajat tekevät virheitä ja joissain tapauksissa kirjoitusvirheestä syntyvä semanttinen tai looginen virhe voi olla jopa hankalampi havaita kuin syntaktisesta syystä johtuva virhe. Tämänkaltaisissa ongelmatapauksissa myöskään tekniset työkalut, kuten edellä mainitut tulkit tai konsolit, eivät pysty useinkaan tarjoamaan käsillä olevaan ongelmaan ratkaisua. Tekoälyn nousevana aikakautena on tärkeä muistaa, että syntaktinen koodin tai tekstin tuottaminen ei ole ongelmakohta vaan nimenomaan semanttinen ja looginen. Mitä pidemmälle tekoälyt kehittyvät, sitä vähemmän ihmiselle jää tarvetta miettiä syntaksiin liittyviä ongelmakohtia. Oikeamuotoinen kirjoitusasu ja koodin loogisuus sen sijaan nousevat enemmän esille. On tärkeää, että olemassa oleva kirjoitustaito -ja tieto säilyy ihmisillä sen sijaan, että koneet tuottavat jatkossa suurimman osan ihmisen toteuttamasta koodista. Tämä pätee mihin tahansa asiaan, jossa teknologia helpottaa jonkin toiminnon suorittamista. Jo nyt on havaittavissa tietynkaltaisen tiedon vähenemistä ihmisten keskuudessa, joista yhtenä hyvänä esimerkkinä on tietokoneiden konkreettinen toimivuus. Suurin osa ihmisistä toimii tänä päivänä ainoastaan tietokoneiden käyttäjinä, jolloin kokemus käyttäjärjestelmistä ja vaikkapa ohjelmoinnista säilyy, mutta tieto syvemmästä toiminnasta vähenee jatkuvasti. Tällöin osaavien ihmisten määrä tietokoneiden rakentamisesta ja fyysisestä toiminnasta voi pahimmassa tapauksessa vähentyä niin merkittävästi, ettei ihminen pysty enää rakentamaan toimivaa tietokonetta ilman tekoälyn apua. Tämän tapahtuessa ihminen on jälleen lukinnut itsensä olemaan riippuvainen jostakin omasta luomuksestaan.

6.1 Tutkimuksen arviointi ja rajoitteet

Haasteellisin osio tutkimuksen toteuttamisessa oli aineiston keruu sen puutteellisuuden vuoksi. Tutkimuksessa kokeiltiin erilaisia lähestymiskulmia lähtöaineiston hankkimiseksi, mutta loppuen lopuksi tutkimuksessa käytetyt hakutermit- ja lausekkeet tuottivat suurimman määrän tuloksia, jotka joltain osin vastasivat tutkittavan aiheen tematiikkaan. Odotukset tutkitun tiedon määrästä olivat tutkimusta aloittaessa paljon suuremmat verrattuna todelliseen tiedon määrään. Kaikesta huolimatta lopullinen tutkimusaineisto jäi niukaksi ja tästä johtuvasta syystä aiheen kokonaiskuva jää osittain vaillinaiseksi. Tutkittava aihe on itsessään melko kapea, joka vaikutti aineiston laajuuteen. Jälkikäteen ajateltuna, systemaattinen kirjallisuuskatsaus olisi voinut olla tämän tutkimuksen puitteissa parempi lähestymistapa. Lähdeaineistoa etsiessä, kävi ilmi, ettei aihe myöskään tutkimuskontekstissa ole erityisen suosittu. Tästä voi päätellä, ettei aihe ole erityistä kiinnostusta herättävä. Tämä tutkielma toteutettiin yksilötyönä, joka nostaa todennäköisyyksiä tutkimusmenetelmällisille virheille, kuten aineiston valintaan sekä aineiston kategorisointiin. Näiden tosiasioiden valossa, voidaan todeta, että mikäli jatkotutkimusta aiheen parissa suoritetaan, on parempi toteuttaa tutkimus esimerkiksi parityönä, jonka avulla tutkimus on suuremmalla varmuudella validi. Tutkimusta pystyy myös laajentamaan ottamaan kantaa yleisiä kielii kirjoittaessa tapahtuvia virheitä, jotka vaikuttavat esimerkiksi tekstin ymmärrettävyyteen. Aiheen supistaminen koskettamaan ainoastaan tietokonekieliä voi toimia niin rajoittavana tekijänä, että mitään maailmaa mullistavaa löydöstä tuskin löydetään.

6.2 Mahdollisuudet jatkotutkimukselle

Valtaosa tietokonekielten kirjoittamisesta kertovat tutkimukset olivat joko kohdistettu noviiseihin tai opiskelijoihin, joka sinänsä viittaa siihen, että oletetaan virheellisen kirjoittamisen tapahtuvan pääasiallisesti aloittelijatasoilla. Tämä varmasti pitää paikkansa johtuen siitä, että suurin osa varsinkin ilmiselvistä kirjoitusvirheistä tapahtuu todennäköisemmin harjaannuttamattoman tekijän käsissä kuin ammattilaisen. Tutkimusaiheen skaalaa voisi kuitenkin laajentaa koskettamaan myös kokeneempien kehittäjien saralla. Tietokonekielten kirjoittaminen on monessa suhteessa monimutkaista ja myös kokeneet kirjoittajat tekevät aika ajoin virheitä, joiden tutkiminen voisi tarjota hedelmällistä apua niin opetuskontekstissa kuin yrityksillekin. Tämän kaltainen tutkimus voisi edistää opiskelijoiden ja noviisitasoisten kehittäjien oppimista ja täten helpottaa siirtymistä työelämään. Monissa tietojenkäsittelyn alan yrityksessä nähdään ongelmaksi junior tasoisten työntekijöiden kouluttaminen sekä senior tason työntekijöiden tiedonsiirto tuleville sukupolville. Loppuen lopuksi minkä tahansa tietokonekielen kirjoittaminen vaatii valtavan määrän harjoittelua ja suurin osa, varsinkin alussa

tapahtuvasta oppimisesta liittyy jollain tavalla oikeinkirjoitukseen. Looginen ja semanttinen puoli kehittyy usein vasta kokemuksen kautta.

7 YHTEENVETO

Tämän kirjallisuuskartoituksen tarkoitus oli tutkia, kuinka paljon tutkittua tietoa on saatavilla tietokonekielten kirjoittamisessa tapahtuvista virheistä, mitkä ovat näiden tutkimusten julkaisukanavat ja mitkä ovat aihealuetta käsittelevien tutkimusten julkaisutrendit. Tutkimuksen motiivina toimi tutkijan oma henkilökohtainen kiinnostus tietokonekielten kirjoittamista kohtaan ja miten paljon kirjoitustaito merkitsee erilaisten ohjelmien ja tietoteknisten kokonaisuuksien toimivuuden kannalta. Aiheen tutkimisen mielenkiinto on korostunut nousevan tekoälyn aikakauden myötä ja on kiinnostavaa seurata, kuinka paljon tekoälyn tuottama apu vaikuttaa tietojenkäsittelyn alaan, mikäli tulevaisuudessa suurimman osan käytettävästä koodista ihmisen sijaan kirjoittaakin tekoäly. Tutkimusmetodina sovellettiin systemaattista kirjallisuuskartoitusta, joka tarjosi selkeän prosessin tutkimuksen toteuttamiseksi sekä tiedon keräämiseksi ja oli täten sopiva valinta tämän tutkielman puitteissa. Systemaattisen kirjallisuuskartoituksen tarkoitus on koostaa valitun aihealueen tutkitun tiedon laajuutta ja tutkielma osoittaakin, että valitusta aihealueesta tietoa on saatavilla vähäisesti, ja että tutkimuksien fokus on pääsäännöllisesti opetuksen ja oppimisen kontekstissa. Kaikki 43 tutkimukseen valittua artikkelia käsittelevät aihetta tietotekniikan näkökulmasta, jonka takia voidaan todeta, että tämänhetkinen tutkimus sijoittuu puhtaasti tietotekniikan alalle. Löydettyjen tutkimusten julkaisukanavat olivat myös pääosin tietotekniikkaan keskittyvien kanavien alaisuudessa. Tutkimuksen tulokset viittaavat kasvavaan trendiin ja aiheesta on viime vuosikymmenen aikana syntynyt enenevässä määrin uutta tietoa. Tutkimusta tehdessä kävi myös ilmi, että tietokonekielten kirjoittamisen inhimillinen aspekti ei ole aiheena erityisesti suosiossa vaan suurin osa tutkimuksista, jotka aihetta käsittelevät ovat keskittyneet tietokoneiden tai ohjelmistojen konkreettiseen toimintaan. Siitä huolimatta aihealueen parista löytyy paljon uusia näkökulmia tutkimukselle, mitä voidaan mahdollisesti hyödyntää tietokonekielten opetuksessa. Mikäli aihealueen tutkimista jatketaan myös tulevaisuudessa, yhtenä keinona laajemmille tuloksille voisi olla luonnollisten kielten teeman yhdistäminen osaksi jatkotutkimusta.

LÄHTEET

- ACM. (19.9.2023). ACM History. <https://www.acm.org/about-acm/acm-history>
- Attaway, D. C. (2013). *Matlab: a practical introduction to programming and problem solving*. Butterworth-Heinemann.
- Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*, 41(6), 1-6. <https://doi.org/10.1145/3011286.3011301>
- Brown, H. D. (2000). *Principles of Language Learning and Teaching*. New York, Longman, Inc.
- Connolly, T. M., & Begg, C. E. (2005). *Database systems: a practical approach to design, implementation, and management*. Pearson Education.
- Corder, S.P. (1973). *Introducing Applied Linguistics*, Penguin Books. DOI: 10.4236/jss.2018.612029
- Jenkins, T. (2002, August). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, No. 2002, pp. 53-58).
- Kadar, R., Wahab, N. A., Othman, J., Shamsuddin, M., & Mahlan, S. B. (2021). A study of difficulties in teaching and learning programming: a systematic literature review. *International Journal of Academic Research in Progressive Education and Development*, 10(3), 591-605. <http://dx.doi.org/10.6007/IJARPED/v10-i3/11100>
- Kitchenham, Barbara & Charters, Stuart. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2. doi: 10.1109/ESEM.2009.5314235
- Ko, A. J. (2016, November). What is a programming language, really?. In *Proceedings of the 7th international workshop on evaluation and usability of programming languages and tools* (pp. 32-33). <https://doi.org/10.1145/3001878.3001880>
- Konig, E., & Van der Auwera, J. (2013). *The germanic languages*. Routledge.
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. *International Conference on Evaluation & Assessment in Software Engineering*. DOI: 10.14236/ewic/EASE2008.8
- Robbins, J. N. (2012). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. " O'Reilly Media, Inc."
- Schmidt, D. A. (2003). Programming language semantics. *Encyclopedia of Computer Science*. John Wiley and Sons Ltd., GBR, 1463–1466.

- Sebesta, R. W., Mukherjee, S., & Bhattacharjee, A. K. (1999). *Concepts of programming languages* (Vol. 7). Reading, Massachusetts: Addison-Wesley.
- Stack Overflow (5.2.2020). Stack Overflow 2020 Developer Survey. <https://insights.stackoverflow.com/survey/2020>
- Stamouli, I., & Huggard, M. (2006, September). Object oriented programming and program correctness: the students' perspective. In *Proceedings of the second international workshop on Computing education research* (pp. 109-118). <https://doi.org/10.1145/1151588.1151605>
- Taipalus, T. (2023). Systematic Mapping Study in Information Systems Research. *Journal of the Midwest Association for Information Systems (JMWAIS)*, 2023(1), 2. DOI: 10.17705/3jmw.000079
- Taipalus, T., & Seppänen, V. (2020). SQL education: A systematic mapping study and future research agenda. *ACM Transactions on Computing Education (TOCE)*, 20(3), 1-33. <https://doi.org/10.1145/3398377>
- Techopedia. (10.2.2017). *Syntax Error*. <https://www.techopedia.com/definition/13391/syntax-error>
- Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements engineering*, 11, 102-107. <https://doi.org/10.1007/s00766-005-0021-6>
- Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1-10).
- Wohlin, C., Kalinowski, M., Felizardo, K. R., & Mendes, E. (2022). Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. *Information and Software Technology*, 147, 106908.

LIITE 1 VALITUT TUTKIMUKSET

Tutkimuksen nimi	Julkaisu-vuosi	Lähdetieto-kanta	Tekijät
Software Errors and Complexity: An Empirical Investigation	1984	Google Scholar	Basili, V. R., & Perricone, B. T. (1984). Software errors and complexity: an empirical investigation. <i>Communications of the ACM</i> , 27(1), 42-52.
Automatic generation of useful syntax error messages	1986	ACM Digital Library	E Kantorowitz and H Laor. 1986. Automatic generation of useful syntax error messages. <i>Softw. Pract. Exper.</i> 16, 7 (July. 1986), 627-640. https://doi.org/10.1002/spe.4380160703
Logical errors in database SQL retrieval queries	1988	Google Scholar	Buitendijk, R.B. Logical errors in database SQL retrieval queries. <i>Computer Science in Economics and Management</i> 1, 79-96 (1988). https://doi.org/10.1007/BF00427157
User errors in database query composition	1995	ScienceDirect	John B. Smelcer, User errors in database query composition, <i>International Journal of Human-Computer Studies</i> , Volume 42, Issue 4, 1995, Pages 353-381, ISSN 1071-5819, https://doi.org/10.1006/ijhc.1995.1017 .
The effect of programming language on error rates of novice programmers	2000	Google Scholar	McIver, L. (2000, April). The effect of programming language on error rates of novice programmers. In <i>PPIG</i> (p. 15).
The effects of normalization on end-user query errors: An experimental evaluation	2001	ScienceDirect	A.F Borthick, P.L Bowen, S.T Liew, F.H Rohde, The effects of normalization on end-user query errors: An experimental evaluation, <i>International Journal of Accounting Information Systems</i> , Volume 2, Issue 4, 2001, Pages 195-221, ISSN 1467-0895, https://doi.org/10.1016/S1467-0895(01)00023-9 .

Development and evaluation of a model of programming errors	2003	IEEEExplorer	Ko, A. J., & Myers, B. A. (2003, October). Development and evaluation of a model of programming errors. In IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 (pp. 7-14). IEEE.
Identifying and correcting Java programming errors for introductory computer science students	2003	ACM Digital Library	Maria Hristova, Ananya Misra, Megan Rutter, and Rebecca Mercuri. 2003. Identifying and correcting Java programming errors for introductory computer science students. SIGCSE Bull. 35, 1 (January 2003), 153-156. https://doi.org/10.1145/792548.611956
Cognitive process during program debugging	2004	IEEEExplorer	Shaochun Xu and V. Rajlich, "Cognitive process during program debugging," Proceedings of the Third IEEE International Conference on Cognitive Informatics, 2004., Victoria, BC, Canada, 2004, pp. 176-182, doi: 10.1109/COGINF.2004.1327473.
A study of the difficulties of novice programmers	2005	ACM Digital Library	Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. 2005. A study of the difficulties of novice programmers. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05). Association for Computing Machinery, New York, NY, USA, 14-18. https://doi.org/10.1145/1067445.1067453
Semantic errors in SQL queries: A quite complete list	2006	ScienceDirect	Stefan Brass, Christian Goldberg, Semantic errors in SQL queries: A quite complete list, Journal of Systems and Software, Volume 79, Issue 5, 2006, Pages 630-644, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2005.06.028 .
A one year empirical study of student programming bugs	2010	IEEEExplorer	R. C. Bryce, A. Cooley, A. Hansen and N. Hayrapetyan, "A one year empirical study of student programming bugs," 2010 IEEE Frontiers in Education Conference

			(FIE), Arlington, VA, USA, 2010, pp. FIG-1-F1G-7, doi: 10.1109/FIE.2010.5673143.
All syntax errors are not equal	2012	ACM Digital Library	Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2012. All syntax errors are not equal. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12). Association for Computing Machinery, New York, NY, USA, 75-80. https://doi.org/10.1145/2325296.2325318
Towards a taxonomy of errors in HTML and CSS	2013	ACM Digital Library	Park, T. H., Saxena, A., Jagannath, S., Wiedenbeck, S., & Forte, A. (2013, August). Towards a taxonomy of errors in HTML and CSS. In Proceedings of the ninth annual international ACM conference on International computing education research (pp. 75-82).
Meaningful categorisation of novice programmer errors	2014	IEEE Explorer	D. McCall and M. Kölling, "Meaningful categorisation of novice programmer errors," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, Spain, 2014, pp. 1-8, doi: 10.1109/FIE.2014.7044420.
Syntax errors just aren't natural: improving error reporting with language models	2014	ACM Digital Library	Joshua Charles Campbell, Abram Hindle, and José Nelson Amaral. 2014. Syntax errors just aren't natural: improving error reporting with language models. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014). Association for Computing Machinery, New York, NY, USA, 252-261. https://doi.org/10.1145/2597073.2597102
Investigating novice programming mistakes: educator beliefs vs. student data	2014	ACM Digital Library	Neil C.C. Brown and Amjad Altadmri. 2014. Investigating novice programming mistakes: educator beliefs vs. student data. In Proceedings of the tenth annual conference on International computing education research (ICER '14).

			Association for Computing Machinery, New York, NY, USA, 43–50. https://doi.org/10.1145/2632320.2632343
An analysis of HTML and CSS syntax errors in a web development course	2015	ACM Digital Library	Park, T. H., Dorn, B., & Forte, A. (2015). An analysis of HTML and CSS syntax errors in a web development course. <i>ACM Transactions on Computing Education (TOCE)</i> , 15(1), 1-21.
A Quantitative Study of the Relative Difficulty for Novices of Writing Seven Different Types of SQL Queries	2015	ACM Digital Library	Alireza Ahadi, Julia Prior, Vahid Behbood, and Raymond Lister. 2015. A Quantitative Study of the Relative Difficulty for Novices of Writing Seven Different Types of SQL Queries. In <i>Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '15)</i> . Association for Computing Machinery, New York, NY, USA, 201–206. https://doi.org/10.1145/2729094.2742620
37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data	2015	ACM Digital Library	Amjad Altadmri and Neil C.C. Brown. 2015. 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data. In <i>Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)</i> . Association for Computing Machinery, New York, NY, USA, 522–527. https://doi.org/10.1145/2676723.2677258
Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success	2016	ACM Digital Library	Alireza Ahadi, Vahid Behbood, Arto Vihavainen, Julia Prior, and Raymond Lister. 2016. Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In <i>Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)</i> . Association for Computing Machinery,

			New York, NY, USA, 401–406. https://doi.org/10.1145/2839509.2844640
Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries	2016	ACM Digital Library	Alireza Ahadi, Julia Prior, Vahid Behbood, and Raymond Lister. 2016. Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16). Association for Computing Machinery, New York, NY, USA, 272–277. https://doi.org/10.1145/2899415.2899464
Effective compiler error message enhancement for novice programming students	2016	ACM Digital Library	Brett A. Becker, Graham Glanville, Ricardo Iwashima, Claire McDonnell, Kyle Goslin & Catherine Mooney (2016) Effective compiler error message enhancement for novice programming students, Computer Science Education, 26:2-3, 148-175, DOI: 10.1080/08993408.2016.1225464
Do Enhanced Compiler Error Messages Help Students?: Results Inconclusive.	2017	ACM Digital Library	Raymond S. Pettit, John Homer, and Roger Gee. 2017. Do Enhanced Compiler Error Messages Help Students? Results Inconclusive. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). Association for Computing Machinery, New York, NY, USA, 465–470. https://doi.org/10.1145/3017680.3017768
Novice Java Programming Mistakes: Large-Scale Data vs. Educator Beliefs	2017	ACM Digital Library	Neil C. C. Brown and Amjad Altadmri. 2017. Novice Java Programming Mistakes: Large-Scale Data vs. Educator Beliefs. ACM Trans. Comput. Educ. 17, 2, Article 7 (June 2017), 21 pages. https://doi.org/10.1145/2994154

Errors and complications in SQL query formulation	2018	ACM Digital Library	Toni Taipalus, Mikko Siponen, and Tero Vartiainen. 2018. Errors and Complications in SQL Query Formulation. <i>ACM Trans. Comput. Educ.</i> 18, 3, Article 15 (September 2018), 29 pages. https://doi.org/10.1145/3231712
Learning programming, syntax errors and institution-specific factors	2018	ACM Digital Library	Alireza Ahadi, Raymond Lister, Shahil Lal, and Arto Hellas. 2018. Learning programming, syntax errors and institution-specific factors. In <i>Proceedings of the 20th Australasian Computing Education Conference (ACE '18)</i> . Association for Computing Machinery, New York, NY, USA, 90–96. https://doi.org/10.1145/3160489.3160490
Common logic errors made by novice programmers	2018	ACM Digital Library	Andrew Ettles, Andrew Luxton-Reilly, and Paul Denny. 2018. Common logic errors made by novice programmers. In <i>Proceedings of the 20th Australasian Computing Education Conference (ACE '18)</i> . Association for Computing Machinery, New York, NY, USA, 83–89. https://doi.org/10.1145/3160489.3160493
Targeted Example Generation for Compilation Errors	2019	IEEE Explorer	U. Z. Ahmed, R. Sindhgatta, N. Srivastava and A. Karkare, "Targeted Example Generation for Compilation Errors," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 2019, pp. 327-338, doi: 10.1109/ASE.2019.00039.
Query Structure and Data Model Mapping Errors in Information Retrieval Tasks	2019	Google Scholar	Casterella, G. I., & Vijayasathy, L. (2019). Query structure and data model mapping errors in information retrieval tasks. <i>Journal of Information Systems Education</i> , 30(3), 178.
A New Look at Novice Programmer Errors	2019	ACM Digital Library	Davin McCall and Michael Kölling. 2019. A New Look at Novice Programmer Errors. <i>ACM Trans. Comput. Educ.</i> 19, 4, Article 38

			(December 2019), 30 pages. https://doi.org/10.1145/3335814
The Error Landscape: Characterizing the Mistakes of Novice Programmers	2019	ACM Digital Library	Rebecca Smith and Scott Rixner. 2019. The Error Landscape: Characterizing the Mistakes of Novice Programmers. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 538–544. https://doi.org/10.1145/3287324.3287394
Explaining Causes Behind SQL Query Formulation Errors	2020	IEEE Explorer	T. Taipalus, "Explaining Causes Behind SQL Query Formulation Errors," 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274114 .
Sometimes It's Just Sloppiness - Studying Students' Programming Errors and Misconceptions	2020	ACM Digital Library	Ella Albrecht and Jens Grabowski. 2020. Sometimes It's Just Sloppiness - Studying Students' Programming Errors and Misconceptions. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 340–345. https://doi.org/10.1145/3328778.3366862
An Investigation of High School Students' Errors in Introductory Programming: A Data-Driven Approach	2020	Google Scholar	Qian, Y., & Lehman, J. (2020). An Investigation of High School Students' Errors in Introductory Programming: A Data-Driven Approach. <i>Journal of Educational Computing Research</i> , 58(5), 919–945. https://doi.org/10.1177/0735633119887508
Learning analytics of CS0 students programming errors: the case of data science minor	2020	ACM Digital Library	Olga Yarygina. 2020. Learning analytics of CS0 students programming errors: the case of data science minor. In Proceedings of the 23rd International Conference on Academic Mindtrek (AcademicMindtrek '20). Association for Computing Machinery, New York,

			NY, USA, 149-152. https://doi.org/10.1145/3377290.3377319
Insights from student solutions to MongoDB homework problems	2021	ACM Digital Library	Ridha Alkhabaz, Seth Poulsen, Mei Chen, and Abdussalam Alawini. 2021. Insights from Student Solutions to MongoDB Homework Problems. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '21). Association for Computing Machinery, New York, NY, USA, 276-282. https://doi.org/10.1145/3430665.3456308
SQLRepair: Identifying and Repairing Mistakes in Student-Authored SQL Queries	2021	IEEE Explorer	Presler-Marshall, K., Heckman, S., & Stolee, K. (2021, May). SQLRepair: Identifying and repairing mistakes in student-authored SQL queries. In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET) (pp. 199-210). IEEE.
Expert Perspectives on Student Errors in SQL	2022	ACM Digital Library	Miedema, D., Fletcher, G., & Aivaloglou, E. (2022). Expert Perspectives on Student Errors in SQL. ACM Transactions on Computing Education (TOCE).
So many brackets! An analysis of how SQL learners (mis)manage complexity during query formulation	2022	ACM Digital Library	Miedema, D., Fletcher, G., & Aivaloglou, E. (2022, May). So many brackets! An analysis of how SQL learners (mis) manage complexity during query formulation. In Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension (pp. 122-132).
Common Code Writing Errors Made by Novice Programmers: Implications for the Teaching of	2022	Google Scholar	Mase, M. B., & Nel, L. (2022, January). Common Code Writing Errors Made by Novice Programmers: Implications for the Teaching of Introductory Programming. In ICT Education: 50th Annual Conference of the Southern

Introductory Programming			African Computer Lecturers' Association, SACLA 2021, Johannesburg, South Africa, July 16, 2021, Revised Selected Papers (pp. 102-117). Cham: Springer International Publishing.
Identifying SQL Misconceptions of Novices: Findings from a Think-Aloud Study	2022	ACM Digital Library	Miedema, D., Aivaloglou, E., & Fletcher, G. (2022). Identifying SQL misconceptions of novices: Findings from a think-aloud study. <i>ACM Inroads</i> , 13(1), 52-65.
Types of Errors in Block Programming: Driven by Learner, Learning Environment	2023	Google Scholar	Ben-Yaacov, A., & Hershkovitz, A. (2023). Types of Errors in Block Programming: Driven by Learner, Learning Environment. <i>Journal of Educational Computing Research</i> , 61(1), 178-207. https://doi.org/10.1177/07356331221102312

LIITE 2 VALITTUJEN TUTKIMUSTEN JULKAISUKANAVAT

Julkaisukanava	Lyhenteet
ACM conference on Innovation and technology in computer science education	ITiCSE
ACM conference on International computing education research	ICER
ACM Inroads	Inroads*
ACM Sigcse Bulletin	SIGCSE*
ACM Technical Symposium on Computer Science Education	SIGCSE TS*
ACM Transactions on Computing Education	TOCE
Computer Science Education	CSE*
Computer Science in Economics and Management	CSEM*
Conference of the Southern African Computer Lecturers' Association	SACLA
IEEE Frontiers in Education Conference	FIE
IEEE International Conference on Cognitive Informatics	ICCI
IEEE Symposium on Human Centric Computing Languages and Environments	SHCCLE*
IEEE/ ACM International Conference on Automated Software Engineering	ASE
IEEE/ ACM International Conference on Program Comprehension	ICPC
IEEE/ ACM International Conference on Software Engineering: Software Engineering Education and Training	ICSE
International Conference on Academic Mindtrek	ICAM*
International Journal of Accounting Information Systems	IJAIS*
International Journal of Human-Computer Studies	IJHCS*
Journal of Educational Computing Research	JECR*
Journal of Information Systems Education	JISE
Journal of software: Practice and Experience	JPE*
Journal of Systems and Software	JSS*
Australasian Computing Education Conference	ACE
School of Computer Science and Software Engineering Monash University	SCSSEMU*

Working Conference on Mining Software Re- positories	MSR
---	-----

*-merkityt julkaisukanavat edustavat kanavia, joilla ei ole virallista lyhennystä.