

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Chris Rak

Klasside ja relatsiooniliste tabelite päriluse vastenduskeemide analüüs

Bakalaureusetöö

Juhendaja: Jaanus Pöial
PhD

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Chris Rak

16.05.2022

Annotatsioon

Valdav osa tänapäevaseid infosüsteeme kasutavad andmete talletamiseks ja käsitlemiseks andmebaase. Üheks probleemseks osutuda võivaks tegevuseks nende kasutamisel tarkvaraarenduses on päriluse vastendamine.

Käesoleva bakalaureusetöö eesmärk oli analüüsida objektorienteeritud programmeerimise klasside ja relatsiooniliste andmebaaside tabelite päriluse vastenduskeeme ja nende jõudlust, luua disainimustrite kataloog ning pakkuda tulevaste arenduste jaoks optimaalne lähenemine.

Eesmärkide saavutamiseks mõõdeti ja analüüsiti kolme vastenduskeemi jõudlust nelja erineva päringu abil. Iga vastenduskeemi jõudluse mõõtmiseks kasutati 16 erinevat pärilushierarhia ja andmemahu konfiguratsiooni.

Mõõtmistulemuste analüüsi ja analüütiliste hierarhiate meetodi kasutamise tulemusena osutus optimaalseks vastenduskeemiks klassitabeli pärilus, mille keskmine jõudlus erines alternatiividest kuni 124%.

Töö tulemusena loodi päriluse vastenduskeemide disainimustrite kataloog ning tehti soovitus optimaalseks valikuks edasiste arenduste tarvis.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 54 leheküljel, 6 peatükki, 29 joonist, 33 tabelit.

Abstract

The Analysis of Inheritance Mapping Schemes for Classes and Relational Tables

Most modern information systems use databases to store and handle their data. One of the operations that may become problematic for their use in software development is inheritance mapping.

This bachelor's thesis aimed to analyze inheritance mapping schemes for classes of object-oriented programming paradigm and tables of relational databases and make a suggestion for the most optimal design pattern for future developments.

The performances of three different mapping schemes were measured and analyzed using four database queries. For each mapping scheme, 16 different configurations of inheritance hierarchies and data sizes were used.

The findings from the measurement analysis and analytic hierarchy process results indicated class table inheritance to be the most optimal mapping scheme. Its average performance differed from the alternatives by up to 124%.

As a result of this thesis, a catalog of inheritance mapping schemes' design patterns was created and a recommendation was made for the most optimal design choice. The final selection was based on the result of the analytic hierarchy process model and its cross-sensitivity analysis.

The results of the thesis aid software architects in finding the best design approach in the stage of laying out the initial architecture for a project and in its later stage of adding new features. The decision model of the analytical hierarchies can be adjusted to conform to the required use cases and characteristics of different software.

The thesis is in Estonian and contains 54 pages of text, 6 chapters, 29 figures, 33 tables.

Lühendite ja mõistete sõnastik

AHP	<i>Analytic hierarchy process</i> , analüütiliste hierarhiate meetod.
ANOVA	<i>Analysis of variance</i> , dispersioonanalüüs, <i>One-Way ANOVA</i> – ühefaktoriline dispersioonanalüüs.
CLTI	<i>Class table inheritance</i> , klassitabeli päriluse vastenduskeem.
CM	<i>Consistency Measure</i> , kooskõla määr Saaty meetodi hinnangute järjepidevuse kontrollimiseks.
COTI	<i>Concrete table inheritance</i> , konkreetse tabeli päriluse vastenduskeem.
CSV	<i>Comma Separated Values</i> , komaeraldusega väärtuste fail.
ERD	<i>Entity Relational Diagram</i> , olemi-suhte diagramm.
HSD	<i>Honestly Significant Difference</i> , aus oluline erinevus, Tukey <i>post hoc</i> test rühmadevaheliste tegelike statistiliselt oluliste erinevuste selgitamiseks.
NBS	National Bureau of Standards, USA Riiklik Standardibüroo.
NoSQL	Mitterelatsiooniline alternatiiv klassikalistele relatsioonilistele andmebaasidele.
OOP	Objektorienteeritud programmeerimine.
ORM	<i>Object-relational mapping</i> , objekt-relatsiooniline vastendamine.
SITI	<i>Single table inheritance</i> , üksitabeli päriluse vastenduskeem.
SQL	<i>Structured Query Language</i> , struktuurpäringukeel relatsiooniliste andmebaaside haldamiseks.

Sisukord

1 Sissejuhatus	12
2 Teoreetiline taust	14
2.1 Andmete püsivus ja andmebaasid.....	14
2.1.1 Relatsiooniline andmemudel	15
2.1.2 Objektorienteeritud andmemudel	16
2.2 Objektide ja relatsiooniliste mudelite kattumatus	17
2.3 Pärilus	18
3 Erinevad vastenduskeemide disainilahendused	21
3.1 Üksiktabeli pärilus	21
3.2 Klassitabeli pärilus	24
3.3 Konkreetse tabeli pärilus	26
4 Eksperiment	30
4.1 Eksperimendi eesmärk.....	30
4.2 Eksperimendi kirjeldus	31
4.3 Andmemudelite projekteerimine	32
4.4 Testandmete genereerimine	34
4.5 Mõõdetavad päringud	37
4.5.1 Esimene päring	37
4.5.2 Teine päring.....	37
4.5.3 Kolmas päring	38
4.5.4 Neljas päring.....	39
4.6 Eksperimendi mõõtmistulemused.....	40
4.6.1 Esimese päringu tulemused	41
4.6.2 Teise päringu tulemused.....	42
4.6.3 Kolmanda päringu tulemused.....	43
4.6.4 Neljanda päringu tulemused	44
4.7 Eksperimendi mõõtmistulemuste analüüs	45
4.7.1 Eksperimendi mõõtmistulemuste analüüsi meetoodika	45
4.7.2 Päringute tulemuste eelanalüüs	46

4.7.3	Esimese päringu tulemuste analüüs	46
4.7.4	Teise päringu tulemuste analüüs	47
4.7.5	Kolmanda päringu tulemuste analüüs	48
4.7.6	Neljanda päringu tulemuste analüüs	49
4.7.7	Eksperimendi tulemuste kokkuvõte	49
5	Optimaalse mustri valimine analüütiliste hierarhiate meetodit kasutades.....	51
5.1	Otsustusmudeli koostamine	51
5.1.1	Põhikriteeriumite olulisus.....	54
5.1.2	Alamkriteeriumite olulisus	54
5.1.3	Alternatiivide võrdlus	55
5.2	Tulemuste analüüs ja järeldused.....	60
5.3	Tundlikkuse analüüs	61
5.3.1	Üksiku alamklassi andmete pärimise kestvuse tundlikkuse analüüs.....	61
5.3.2	Kõigi alamklasside metaandmete pärimise kestvuse tundlikkuse analüüs....	62
5.3.3	Muude hinnanguliste kriteeriumite tundlikkuse analüüs.....	63
6	Kokkuvõte	64
	Kasutatud kirjandus	66
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	68
	Lisa 2 – Andmebaasiskeemide salvestusruumid	69
	Lisa 3 – Eksperimendi dispersioonanalüüsi meetod valik	71
	Lisa 4 – Eksperimendi mõõtmistulemuste Saaty teisendused	75

Jooniste loetelu

Joonis 1. Relatsioonilise andmemudeli näidise olemi-suhte diagramm.....	15
Joonis 2. Tellimuse ja tellimuse rea klasside programmikood.....	16
Joonis 3. Üldistamine ühiste omaduste eraldamisel ja koondamisel.....	19
Joonis 4. Spetsialiseerimine spetsiifiliste omaduste eraldamisel.....	19
Joonis 5. Abstraktse ülemklassiga pärilushierarhia.....	20
Joonis 6. Andmebaasi disaini diagramm üksiktabeli päriluse vastendusskeemi muustrile.	23
Joonis 7. Näidisandmete lisamise päring üksiktabeli päriluse vastendusskeemi kasutades.....	23
Joonis 8. Ühe alamklassi andmete päring üksiktabeli päriluse vastendusskeemi kasutades.....	23
Joonis 9. Andmebaasi disaini diagramm klassitabeli päriluse vastendusskeemi muustrile.	25
Joonis 10. Ülemklassi näidisandmete lisamise päring klassitabeli päriluse vastendusskeemi kasutades.....	25
Joonis 11. Alamklasside näidisandmete lisamise päringud klassitabeli päriluse vastendusskeemi kasutades.....	25
Joonis 12. Ühe alamklassi andmete päring klassitabeli päriluse vastendusskeemi kasutades.....	26
Joonis 13. Uue alamklassi toe lisamise päring klassitabeli päriluse vastendusskeemi kasutades.....	26
Joonis 14. Andmebaasi disaini diagramm konkreetse tabeli päriluse vastendusskeemi muustrile.....	28
Joonis 15. Näidisandmete lisamise päringud konkreetse tabeli päriluse vastendusskeemi kasutades.....	28
Joonis 16. Ühe alamklassi andmete päring konkreetse tabeli päriluse vastendusskeemi kasutades.....	28
Joonis 17. Ülemklassi andmete päring konkreetse tabeli päriluse vastendusskeemi kasutades.....	29

Joonis 18. Uue alamklassi toe lisamise päring konkreetse klassi päriluse vastendusskeemi kasutades.....	29
Joonis 19. Eksperimendi klassimudeli näidis.....	32
Joonis 20. Eksperimendi andmebaasi diagramm kahe alamklassi ja üksiktabeli vastendusskeemiga.	33
Joonis 21. Eksperimendi andmebaasi diagramm kahe alamklassi ja klassitabeli vastendusskeemiga.	33
Joonis 22. Eksperimendi andmebaasi diagramm kahe alamklassi ja konkreetse tabeli vastendusskeemiga.	34
Joonis 23. Üksiktabeli ja konkreetse tabeli päriluse vastendusskeemide salvestusruumide erinevused.	36
Joonis 24. Vastendusskeemide jõudluserinevused.	50
Joonis 25. Päriluse vastendusskeemi valiku otsustusmudel.	53
Joonis 26. Analüütiliste hierarhiate meetodi rakendamise lõpptulemus.	60
Joonis 27. Üksiku alamklassi andmete pärimise kestvuse põhikriteeriumi tundlikkuse analüüs.	61
Joonis 28. Kõigi alamklasside metaandmete pärimise kestvuse põhikriteeriumi tundlikkuse analüüs.	62
Joonis 29. Muude hinnanguliste kriteeriumite põhikriteeriumi tundlikkuse analüüs.....	63

Tabelite loetelu

Tabel 1. Näidisandmemudeli tellimuse olemi ilmingud andmebaasitabelis.	15
Tabel 2. Näidisandmemudeli tellimuse rea olemi ilmingud andmebaasitabelis.	16
Tabel 3. Testandmete mahud.....	36
Tabel 4. Esimese päringu (Q1) andmebaasikäsud.....	37
Tabel 5. Teise päringu (Q2) andmebaasikäsud.	38
Tabel 6. Kolmanda päringu (Q3) andmebaasikäsud.	39
Tabel 7. Neljanda päringu (Q4) andmebaasikäsud.....	40
Tabel 8. Mõõtmistulemuste esitamise mall.	40
Tabel 9. Esimese päringu (Q1) mõõtmistulemused üksiktabeli päriluse (SITI) korral. .	41
Tabel 10. Esimese päringu (Q1) mõõtmistulemused klassitabeli päriluse (CLTI) korral.	41
Tabel 11. Esimese päringu (Q1) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.....	41
Tabel 12. Teise päringu (Q2) mõõtmistulemused üksiktabeli päriluse (SITI) korral. ...	42
Tabel 13. Teise päringu (Q2) mõõtmistulemused klassitabeli päriluse (CLTI) korral. .	42
Tabel 14. Teise päringu (Q2) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.....	42
Tabel 15. Kolmanda päringu (Q3) mõõtmistulemused üksiktabeli päriluse (SITI) korral.	43
Tabel 16. Kolmanda päringu (Q3) mõõtmistulemused klassitabeli päriluse (CLTI) korral.....	43
Tabel 17. Kolmanda päringu (Q3) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.....	43
Tabel 18. Neljanda päringu (Q4) mõõtmistulemused üksiktabeli päriluse (SITI) korral.	44
Tabel 19. Neljanda päringu (Q4) mõõtmistulemused klassitabeli päriluse (CLTI) korral.	44
Tabel 20. Neljanda päringu (Q4) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.....	44

Tabel 21. Esimese päringu vastendusskeemide statistiliselt olulised jõudluserinevused.	46
Tabel 22. Teise päringu vastendusskeemide statistiliselt olulised jõudluserinevused. ..	47
Tabel 23. Kolmanda päringu vastendusskeemide statistiliselt olulised jõudluserinevused.	48
Tabel 24. Neljanda päringu vastendusskeemide statistiliselt olulised jõudluserinevused.	49
Tabel 25. Otsustusmodeli põhi- ja alamkriteeriumid.	52
Tabel 26. Põhikriteeriumite võrdlus.	54
Tabel 27. Päringute kestvuse kriteeriumite alamkriteeriumite võrdlus.	55
Tabel 28. Hinnanguliste kriteeriumite alamkriteeriumite võrdlus.	55
Tabel 29. Üksiku alamklassi andmete päringukestvuste hinnangud.	57
Tabel 30. Ülemklassi andmete päringukestvuste hinnangud.....	58
Tabel 31. Lausete keskmised keerukused.	58
Tabel 32. Hinnanguliste kriteeriumite hinnangud.	59
Tabel 33. Analüütiliste hierarhiate meetodi rakendamise lõpptulemus.	60

1 Sissejuhatus

Valdav osa tänapäevaseid infosüsteeme kasutavad vähemalt mingil määral andmete talletamiseks ja käsitlemiseks andmebaase [1].

Andmete efektiivne talletamine ja käsitlemine on keeruline ülesanne, mida raskendab objektorienteeritud programmeerimise (OOP) printsiibil arendatud rakenduste ja relatsiooniliste andmebaaside andmestruktuuride fundamentaalselt erinev ülesehitus. Üheks tegevuseks, mis OOP-il põhinevate objektide andmebaasi talletamisel probleemseks võib osutuda, on päriluse vastendamine. Vastenduskeemide ehk pärilushierarhiate andmebaasis kujutamise viiside seast optimaalse valiku tegemine ei ole triviaalne, kuna arvestada tuleb nii alamklasside arvu, andmete hulga kui ka nende peamiste kasutusviisidega.

Bakalaureusetöö eesmärgiks on analüüsida ja võrrelda erinevaid päriluse vastenduskeeme, nende kasutusviise ja jõudlusomadusi erinevate päringutüüpide puhul, ning valida nende seast optimaalne.

Töö autor on arendajana puutunud kokku projektidega, milles on kasutatud pärilushierarhiate vastendamiseks erinevaid lähenemisi ilma põhjalikku analüüsi läbi viimata. Seoses uue projekti ja selle arhitektuuri paika seadmisega on oluline analüüsida, millised on erinevate lähenemiste positiivsed ja negatiivsed küljed ning millest lähtuda disainimustri valiku tegemisel.

Kuigi päriluse vastenduskeemide tööpõhimõtteid on kirjeldatud nii erialakirjanduses kui ka teadustöodes, siis ei ole neis välja toodud kasutamissoovituste kohta esitatud jõudluste ega muid arvulisi andmeid. Vähestes olemasolevates jõudlusvõrdlustes on mitmeid puudujääke, millega käesoleva lõputöö raames tegeldakse.

Töös kasutatavaks meetodikaks on disainiteadus, mille käigus kaardistatakse päriluse vastenduskeemid ning luuakse tehised disainimustrite kataloogi kujul. Jõudluse analüüs viiakse läbi erinevate pärilushierarhiate konfiguratsioonide ja andmemahtudega ning selle käigus mõõdetakse nendel põhinevate lahenduste jõudlust erinevate andmebaasipäringute

abil. Mõõtmistulemuste dispersioonanalüüsi tulemused on osaks analüütiliste hierarhiate meetodi abil läbi viidud optimaalse vastendusskeemi valiku protsessist.

Lõputöö tulemusena valmib disainimustrite põhjalik analüüs ja arvandmetel põhinev võrdlus, mille abil on võimalik teha edasiste arenduste tarvis erinevatele andmestruktuuridele ja -mahtudele vastav optimaalne valik. Tulemused on kasuks tarkvaraarhitektidele nii rakenduste loomise algfaasis kui ka olemasolevatele uue funktsionaalsuse lisamisel.

Töö esimeses osas antakse ülevaade OOP-i objektide ja relatsiooniliste andmete salvestamise ja käsitlemise erinevustest ning nende omavahelise ühildamise keerukusest. Olulisima osana käsitletakse pärilust kui üht OOP-i põhiprintsiipi.

Töö teine osa koosneb disainimustrite kataloogina esitatud erinevatest päriluse vastendusskeemide lahendustest.

Töö kolmas osa käsitleb läbiviidava eksperimendi eesmärki, kirjeldust, mõõtmistulemusi ja nende dispersioonanalüüsi.

Töö neljandas osas kasutatakse optimaalse disainimustri väljavalimiseks analüütiliste hierarhiate meetodit. Samuti viiakse läbi saadud tulemuste tundlikkuse analüüs.

2 Teoreetiline taust

Kaasaegsete rakenduste loomiseks on laialdaselt kasutusel kaks peamist tehnoloogilist suunda. Äri loogika jaoks kasutatakse suuresti objektorienteeritud programmeerimist ning andmete hoiustamiseks relatsioonilisi andmebaase. Pakkudes taaskasutatavuse, robustsuse ja hooldatavuse eeliseid, on terviklike süsteemide elluviimisel objektorienteeritud programmeerimine võtmetehnoloogiaks. Relatsioonilisi andmebaase kasutatakse sellistes süsteemides püsivate andmekogudena [2].

Käesolevas peatükis antakse lühiülevaade andmete püsivusest relatsiooniliste andmebaaside ja objektorienteeritud programmeerimiskeelte mõistes ning kirjeldatakse objektide ja relatsioonide mittekattuvuse probleemi. Olulisima osana käsitletakse pärilust kui üht kattumatuset ilmingut ja OOP-i põhiprintsiipi ning selgitatakse vajadust päriluse vastendusskeemide järel.

2.1 Andmete püsivus ja andmebaasid

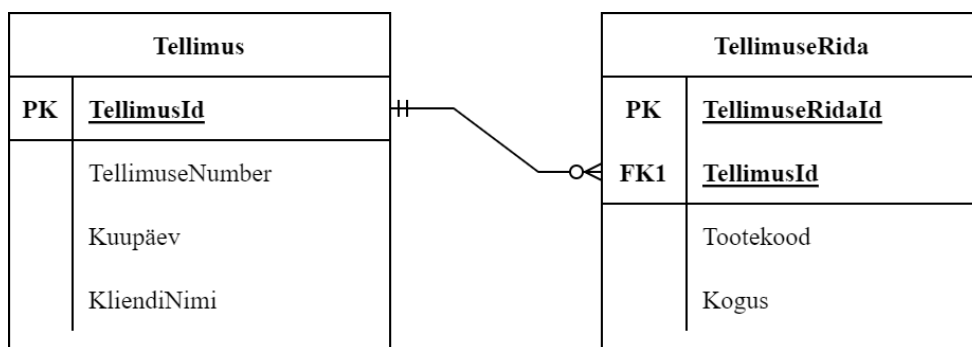
Andmete püsivuseks (ingl *persistence*) nimetatakse mõistet, mis võimaldab andmetel püsima jääda ka pärast neid loonud protsessi lõppemist. Enamik juhtudel talletatakse andmed mõnele füüsilisele infot säilitavale andmekandjale. Tarkvara tasemel on andmete püsivust võimalik saavutada mitmel viisil, näiteks võib andmed salvestada failina või organiseerituna läbi muude tarkvarakihtide [3].

Üheks andmete püsivaks säilitamise viisiks on andmebaaside kasutamine. Andmebaasiks nimetatakse organiseeritud andmete kogu, mis on korraldatud viisil, mis lubab hilisemal ajal andmetele efektiivset ja täpset ligipääsu [4]. Tüüpiliselt koosnevad need skeemidest, tabelitest, protseduuridest ning vaadetest. Tänapäevased andmebaasid võib peamiselt jagada kaheks: relatsioonilised ja mitterelatsioonilised (viimaseid nimetatakse tihti peale ka NoSQL andmebaasideks). Valdava osa tänapäevaste relatsiooniliste andmebaasidega suhtlemiseks kasutatakse struktuurpärangukeelt SQL (*Structured Query Language*), mis relatsioonilist mudelit küll täielikult ei järgi, kuid siiski sellel põhineb [5]. Kui SQL andmebaasid talletavad andmeid rangelt määratletud piirangutega tabelites, siis NoSQL

andmebaasid kasutavad tabelite ja nendevaheliste suhete asemel olenevalt konkreetsest lähenemisest lihtsamaid andmestruktuure. Mitterelatsiooniliste andmebaaside näidete alla kuuluvad võti-väärtuspaaride kogudel, dokumendibaasidel ja graafidel põhinevad andmebaasid nagu näiteks Redis, Azure Cosmos DB, MongoDB ja ArangoDB. Käesolevas lõputöös kasutatakse näidete ja jõudluse mõõtmiste tarbeks Microsoft SQL Serverit, mille puhul on tegemist relatsioonilise andmebaasisüsteemiga.

2.1.1 Relatsiooniline andmemudel

Relatsioonilisel mudelil põhinevas SQL andmebaasi tabelis on enamasti vähemalt kaks veergu, millest üks märgib tabeli kirjete unikaalset domeen-võtit ja ülejäänud kirje teisi atribuute. Relatsioonilise andmemudeli näidise olemi-suhte diagrammi (ERD – *Entity Relationship Diagram*) esitab joonis 1.



Joonis 1. Relatsioonilise andmemudeli näidise olemi-suhte diagramm.

Olemi-suhte diagrammile vastava andmemudeli tellimuse andmebaasitabelit koos näidisandmetega esitab tabel 1.

Tabel 1. Näidisandmemudeli tellimuse olemi ilmingud andmebaasitabelis.

TellimusId	TellimuseNumber	Kuupäev	KliendiNimi
17	#12345	2022-01-01	Mart Tamm
18	#12347	2022-01-01	Mari Kask
19	#12349	2022-02-01	Mikk Kuusk
20	#13251	2022-02-01	Mart Tamm

Olemi-suhte diagrammile vastava andmemudeli tellimuse rea andmebaasitabelit koos näidisandmetega esitab tabel 2.

Tabel 2. Näidisandmemudeli tellimuse rea olemi ilmingud andmebaasitabelis.

TellimuseRidaId	TellimusId	Tootekood	Kogus
168	17	46005	3
169	17	39967	2
170	18	68776	4
171	19	37338	1

Iga olemi ilmingud esitatakse andmebaasitabeli ühe reana ning seosed erinevate tabelite vahel teostatakse andmeväljade sama semantikaga andmeväljade väärtuste kaudu [6].

2.1.2 Objektorienteeritud andmemudel

Objektorienteeritud programmeerimiskeeled, nagu näiteks C# ja Java, kasutavad andmete hoidmiseks klassidel põhinevaid objekte. Kui lihtsamatest võti-väärtuspaaridest koosnevate objektide andmebaasi salvestamine on suhteliselt triviaalne, siis keerulisemate domeenimudelite korral, kus klassid on omavahel seotud ja need üksteisele viitavad, tuleb objektidevahelised seosed relatsioonideks vastendada. Põhjus seisneb relatsiooniliste andmebaaside eripäras, mille kohaselt on andmebaasis võimalik talletada vaid skalaarseid väärtusi [7]. Joonis 2 esitab tellimuse ja tellimuse rea klasside programmikoodi C# keeles, millest nähtub, et OOP-printsiiibil kirjutatud klasside struktuur ei vasta üks-ühele andmebaasi tabelite omale. Objektide ja relatsiooniliste mudelite kattumatust käsitleb põhjalikumalt järgmine alampeatükk.

```
public class Tellimus
{
    public string TellimuseNumber { get; set; }
    public DateTime Kuupäev { get; set; }
    public string KliendiNimi { get; set; }
    public ICollection<TellimuseRida> TellimuseRead { get; set; }
}

public class TellimuseRida
{
    public Tellimus Tellimus { get; set; }
    public string TooteKood { get; set; }
    public int Kogus { get; set; }
}
```

Joonis 2. Tellimuse ja tellimuse rea klasside programmikood.

Oma olemuselt on objektorienteeritud programmeerimiskeelte objektid mööduvad (ingl *transient*), kuna need luuakse programmi töö käigus ning kaotatakse selle lõppedes. Kuigi selline tööviis võib sobida väiksemamahuliste rakenduste tarvis, siis samadele andmete korduvat ligipääsu vajavate suuremate ja pikema tööajaga rakenduste puhul tuleb kasutada lähenemist, mis toetab andmete püsivust.

2.2 Objektide ja relatsiooniliste mudelite kattumatus

Objektide ja relatsiooniliste mudelite kattumatus, mida nimetatakse laiemalt objektorienteeritud ja relatsioonilise maailmapildi kokkusobimatuses, tuleneb faktist, et kui relatsiooniteooria põhineb vektorite ehk järjestatud väärtuste jadade vahelistel suhetel, mida päritakse (ingl *queried*), siis objektide paradigma põhineb objektidevahelistel suhetel, mida läbitakse (ingl *traversed*) [8].

Objektorienteeritud keelte andmemudelite erinevused relatsiooniliste andmebaaside omadest võivad rakenduse arendusfaasis probleeme tekitada. Mõned näited sellistest erinevustest on uju- ja püsikomaarvude erinev esitamine ning näiteks SQL-is vajadus piiritleda sõnede maksimaalne pikkus – C#-is see puudub. Fundamentaalsed probleemidena tuuakse kattumatusel puhul välja granulaarsus, pärilus, identiteet, ühenduvus ja andmete vahel navigeerimine [9]. Käesoleva bakalaureusetöö fookuses on pärilus ning selle kasutamisel tekkivate kitsaskohtade ületamine vastendamist kasutades.

Oma olemuselt on rakenduse domeenimudeli ja andmebaasiskeemi vaheline vastendamine mitteisomorfne. Domeenimudeli poolt vaadatuna jäetakse andmebaasi vastendamata teatavad aspektid objektimudelist, nagu näiteks klasside ja klassimuutujate funktsionaalsus. Sarnaselt toimitakse ka vastupidise vastenduse korral, kus relatsioonilise poole pealt ei esitata domeenimudelis andmebaasi kõigi skeemide kõiki tabeleid, veergusid ega ka salvestatud protseduure [10].

Enamikul juhtudel, kuid eriti neil, kus andmemudel on kõrgelt normaliseeritud, ei esine skeemi esitamisel domeenimudelina suurt keerukust. Olemi-suhte mudeli põhjal vastendatakse iga olem kindla domeeniklassiga ning iga olemitevaheline suhe ühe klassi välja või viitetüübiga ning teise klassi välja või mitmeväärtuselise tüübiga. Samas ei ole igasuguse domeenimudeli relatsioonilisse andmebaasi talletamine sedavõrd lihtne.

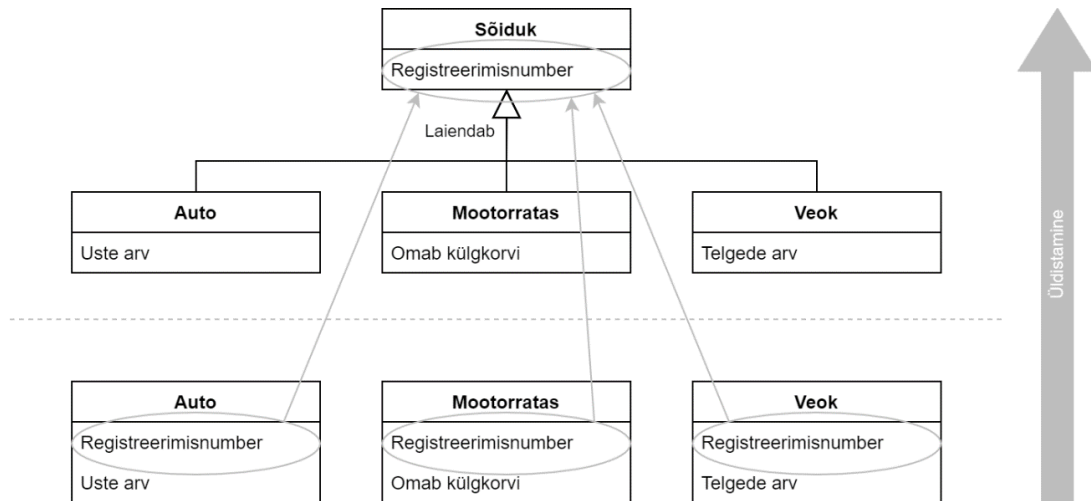
Abstraktseid klasse, pärilust või liideseid kasutavate domeenimudelite puhul tuleb leida tasakaalupunkt domeeni keerukuse ja mõistliku jõudluse saavutamise vahel [2].

Et programmis kasutatavaid objekte oleks võimalik andmebaasi salvestada, võib kasutada vahekihti, mis aitab lahendada programmi klasside ja andmebaasiobjektide kokkusobimatus. Vahekihi abil vastendatakse automaatselt SQL päringud ja käsklused ja/või konvertitakse automaatselt programmiobjekte lihtsamateks väärtusteks, mida andmebaasitabelites on võimalik talletada. Tarkvarateeke, mis eraldavad programmi mälus hoitavad objektid andmebaasist ning mille eesmärk on liigutada andmeid programmi ja andmebaasi vahel neid üksteisest eraldatuna hoides, nimetatakse ORM-ideks ehk objekt-relatsioonivastendajateks (ingl *Object-Relational Mappers*) [11]. ORM-id loovad relatsioonilise andmebaasi kohale kõrgetasemelise abstraktsiooni, mille abil saab andmebaasiga suhelda, kasutades SQL-i asemel rakenduse enda programmeerimiskeelt ja selle vahendeid. ORM-ide tööpõhimõtet on kirjeldatud mitmetes teadustöodes [10], [12], mistõttu käesolevas lõputöös sellele täiendavalt ei keskenduta.

2.3 Pärilus

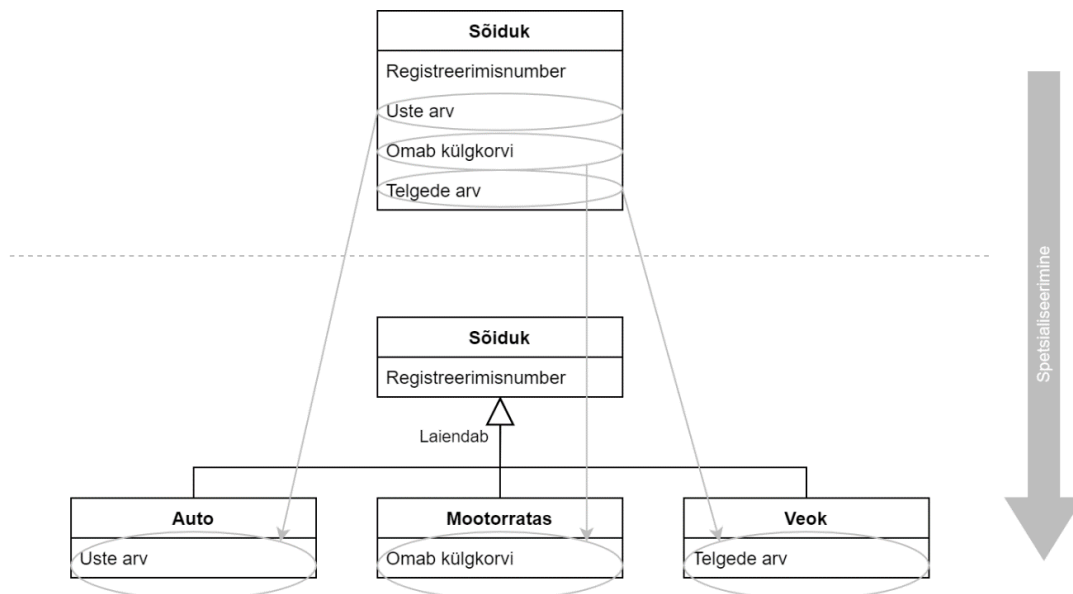
Pärilust nimetatakse üheks objektorienteeritud programmeerimise paradigma põhikontseptiks [13]. Päriluse puhul on tegemist mehhanismiga, mille kohaselt luuakse uued klassid olemasolevate põhjal, pärides nende omadused ja käitumise ning saavutades sellega andmete ja programmikoodi dubleerimise vähenemise. Selline lähenemine võimaldab säästa tarkvaraarendusele kuluvat aega ning lihtsustab süsteemi efektiivset kasutuselevõttu ja hooldust [14].

Päriluse peamisteks võtmesõnadeks on üldistamine ja spetsialiseerimine. Üldistamiseks on kahe või enama klassi ühiste omaduste eraldamine ja koondamine uude klassi (Joonis 3). Sellist ühist klassi nimetatakse baasklassiks ehk ülemklassiks ning sellest pärinevaid klasse tuletatud klassideks ehk alamklassideks [15].



Joonis 3. Üldistamine ühiste omaduste eraldamisel ja koondamisel.

Vastupidiselt üldistamisele tähendab spetsialiseerimine olemasolevast klassist selle eriomaduste väljatoomist. Spetsialiseerimise korral eraldatakse klassist ainult mõnede tüüpide jaoks olulised omadused ning luuakse nende jaoks omaette alamklassid (Joonis 4). Alamklass pärib kõik ülemklassi omadused ning spetsialiseerib seda lisades vaid enda jaoks olulised omadused [15].



Joonis 4. Spetsialiseerimine spetsiifiliste omaduste eraldamisel.

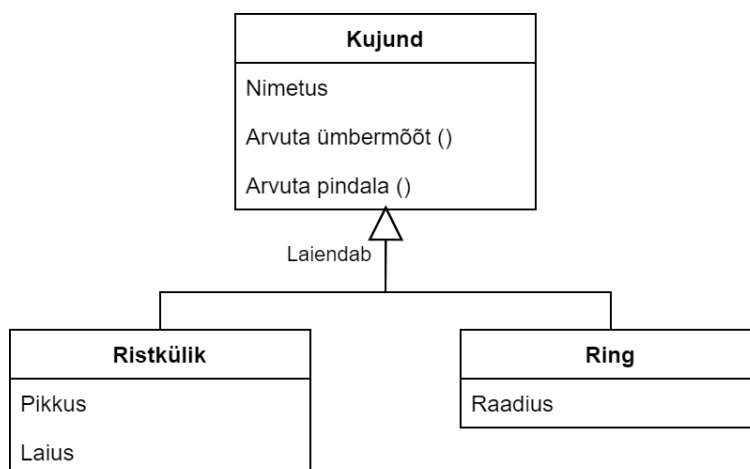
Käesoleva töö raames on oluline eristada otseseid ja kaudseid ülemklasse. Kui otseseks ülemklassiks on klass, millest alamklass vahetult pärineb, siis kaudseks ülemklassiks võib olla mis tahes otsesest ülemklassist pärilushierarhias kõrgemal asuv ülemklass. See

erinevus on oluline järgmises peatükis esitatud päriluse vastenduskeemide defineerimisel.

Pärlust käsitledes on oluline eristada suhteid see-on (ingl *is-a*) ja sel-on-olemas (ingl *has-a*). See-on pärlussuhte korral saab alamklassi objekti kasutada kui ülemklassi objekti, nt mootorratas on sõiduk. Sel-on-olemas kompositsioonisuhetega esitatakse aga omadussuhet, kus objektile on olemas viide teisele objektile, nt autol on olemas piduripedaal [14]. Kui enamikes objektorienteeritud programmeerimiskeeltes, sh C#-is ja Javas on võimalik esitada mõlemat liiki suhteid, siis relatsioonilistes andmebaasides on võimalik andmete omavaheliseks sidumiseks kasutada vaid sel-on-olemas kompositsioonisuhet. Sellest piirangust tulenevalt on OOP-i pärluse esitamiseks relatsioonilistes andmebaasides vajalik kasutada vastendamist [11].

Mõned programmeerimiskeeled, nagu C++ ja Python, toetavad mitmest pärimist, mille korral alamklass pärineb korraga mitmest ülemklassist. Teised keeled, nagu C# ja Java, sellist lähenemist ei toeta, vaid kasutusel on vaid ühene pärimine, et vältida mitmese pärimisega kaasnevat probleeme, nagu näiteks teemanti probleemi [16].

Abstraktsed klassid on klasside alamliik, mida kunagi ei initsialiseerita ehk millest objekte ei looda. Selliseid klasse kasutatakse pärlushierarhias vaid ülemklassidena eesmärgiga deklareerida pärinevatele alamklassidele ühine baasstruktuur. Konkreetsete objektide loomiseks on abstraktsed klassid liiga üldised, kuna need kirjeldavad vaid alamklasside ühisosa (Joonis 5). Abstraktsete klasside vastandiks on konkreetsed klassid, mis on objektide initsialiseerimiseks piisavalt spetsiifilised [14].



Joonis 5. Abstraktse ülemklassiga pärlushierarhia.

3 Erinevad vastendusskeemide disainilahendused

Tulenevalt päriluse probleemi standardse lahenduse puudumisest relatsioonilistes andmebaasides tuleb päriluse esitamiseks kasutada vastendamist. Kuigi PostgreSQL andmebaasis on pärilus toetatud, siis selle puhul ei ole tegemist pelgalt relatsioonilise vaid objekt-relatsioonilise andmebaasiga, mis toetab lisaks mõnesid OOP-ile eripäraseid võimalusi [17]. Selles kasutatav eripärane lähenemine jääb käesoleva lõputöö skoobist väljapoole.

Käesolevas peatükis esitatakse kolm päriluse vastendamiseks kasutatavat disainimustrit: üksiktabeli pärilus, klassitabeli pärilus ja konkreetse tabeli pärilus. Ühise struktuuri tarvis võetakse aluseks Sarah Marion Miku magistritöö [5]. Selles käesoleva töö jaoks mõnevõrra kohandanud mustrite esitamise struktuuri kohaselt kuuluvad iga disainimustri juurde järgnevad atribuudid:

- disainimustri eestikeelsed nimetused;
- disainimustri ingliskeelsed nimetused algallikast kui ka muud levinud nimetused;
- kirjeldus disainimustri tööpõhimõttest ja selle eripäradest;
- kirjalikele allikatele tuginevad eelised disainimustri valiku korral;
- kirjalikele allikatele tuginevad puudused disainimustri valiku korral;

Lisaks esitatakse kahe alamklassiga pärilushierarhia struktuuri koodinäide ning sellele vastav SQL andmebaasi olemi-suhte diagramm.

3.1 Üksiktabeli pärilus

Üksiktabeli päriluse eestikeelsed nimetused on ühe tabeli pärilus ja tabel hierarhia kohta. Inglisekeelsed nimetused on *Single Table Inheritance* ja *Table per Hierarchy* [11], [18].

Üksiktabeli päriluse vastendusskeemi tööpõhimõtte seisneb klasside pärilushierarhia esitamises ühe tabelina [2]. Ühe alamklassi andmete sisestamisel jäetakse tühjaks kõik andmebaasitabeli veerud, mis selle alamklassi puhul olulised ei ole [18]. Tabelis kasutatakse klasside eristamiseks spetsiaalset tunnusvälja, mis klassiandmete

salvestamisel talletatakse ning lugemisel aitab otsustada, millisele alamklassile andmebaasitabeli rida vastab [19].











Eelised üksiktabeli päriluse vastendusskeemi kasutamisel on järgmised:

- Kogu pärilushierarhia esitatakse andmebaasis ühe tabelina ning kogu info on kättesaadav ühest kohast [10].
- Klassiväljade pärilushierarhias liigutamine ei vaja andmebaasistruktuuris muudatuste tegemist [11].
- Andmete pärimisel ei ole tarvis kasutada ühendpäringuid [11].

Puudused üksiktabeli päriluse vastendusskeemi kasutamisel on järgmised:

- Vajab spetsiaalset tunnusvälja alamklasside eristamiseks [19].
- Andmeväljade olulisus sõltub vaadeldavast kontekstist [11].
- Andmemudel kaldub kõrvale andmemudeli kolmandast normaalkujust ning põhjustab seoste riknemise ja mälu raiskamise anomaaliaid [18].
- Vaid üksikute alamklasside poolt kasutatavad veerud põhjustavad suure hulga tühjade väljade tekkimise andmebaasitabelisse ning andmeruumi raiskamist [11].
- Suure hulga jõudlust vähendavate indeksite ja sageli toimivate lukustamiste puhul võib üksik tabel osutada liialt suureks [11].
- Tabeli lubatud maksimaalne veergude arv on alamklasside vahel ühiselt jagatud, mistõttu võib väga suure hulga alamklasside puhul jõuda kätte andmebaasispetsiifiline praktiline piir.
- Väljade jaoks on kasutada ainult üks nimeruum, mistõttu tuleb olla kindel, et sama nime ei kasutata erinevate alamklasside väljade jaoks. Selle puhul on kasulikud ühendnimed, kus klassi nime kasutatakse välja nimes ees- või järelliitena [11].

Üksiktabeli päriluse vastendusskeemi kasutatavat andmebaasi diagrammi esitab joonis 6.

Meedia	
 Id	int
 Pealkiri	nvarchar(200)
 Avaldaja	nvarchar(200)
 AvaldamiseHetk	datetime2
 OnTasuline	bit
 MTunnus	smallint
 ASonadeArv	int
 AOnKommenteeritav	bit
 VKestvus	int
 VOnSubtiitritega	bit

Joonis 6. Andmebaasi disaini diagramm üksiktabeli päriluse vastenduskeemi muustrile.

Uue kirje lisamiseks tuleb INSERT lauses defineerida tunnusväli (näiteks MTunnus), millega oleks võimalik artiklite ja videote kirjeid eristada. Näidisandmete sisestamisel on esitatud päring nelja näidiskirjega, millest kaks vastavad alamklassi Artikkel andmetele ning kaks alamklassi Video omadele (Joonis 7).

```
INSERT INTO Meedia (Pealkiri, Avaldaja, AvaldamiseHetk,
                   OnTasuline, MTunnus, ASonadeArv,
                   AOnKommenteeritav, VKestvus, VOnSubtiitritega)
VALUES ('Artikkel 1', 'Avaldaja 1', '2022-01-01', 1, 1, 100, 1, null, null),
      ('Artikkel 2', 'Avaldaja 2', '2022-01-02', 0, 1, 50, 1, null, null),
      ('Video 1', 'Avaldaja 3', '2022-01-02', 1, 2, null, null, 30000, 0),
      ('Video 2', 'Avaldaja 3', '2022-01-03', 0, 2, null, null, 60000, 1)
```

Joonis 7. Näidisandmete lisamise päring üksiktabeli päriluse vastenduskeemi kasutades.

Üksiku alamklassi andmete pärimisel (Joonis 8) tuleks sarnaselt andmete sisestamisele defineerida päringu parameetrina alamklasse eristav tunnus.

```
SELECT Id,
       Pealkiri,
       Avaldaja,
       AvaldamiseHetk,
       OnTasuline,
       ASonadeArv,
       AOnKommenteeritav
FROM Meedia
WHERE MTunnus = 1
```

Joonis 8. Ühe alamklassi andmete päring üksiktabeli päriluse vastenduskeemi kasutades.

3.2 Klassitabeli pärilus

Klassitabeli päriluse eestikeelseks nimetuseks on tabel tüübi kohta. Inglise keeles nimetatakse seda kui *Class Table Inheritance* või *Table per Type* [11], [20].

Klassitabeli päriluse vastendusskeemi tööpõhimõte seisneb klasside pärilushierarhia esitamises ühe tabeliga iga alamklassi kohta [20]. Ülemklassile vastab tabel, mis sisaldab kõigi alamklasside jaoks ühiseid väljasid ning igale alamklassile vastab omaette tabel.

Domeenimudeli klasside väljad vastenduvad otse vastavate tabelite väljadega ning alamklasside primaarvõtmed toimivad ka ülemklassidele vastavate tabelite välisvõtmetena [19].

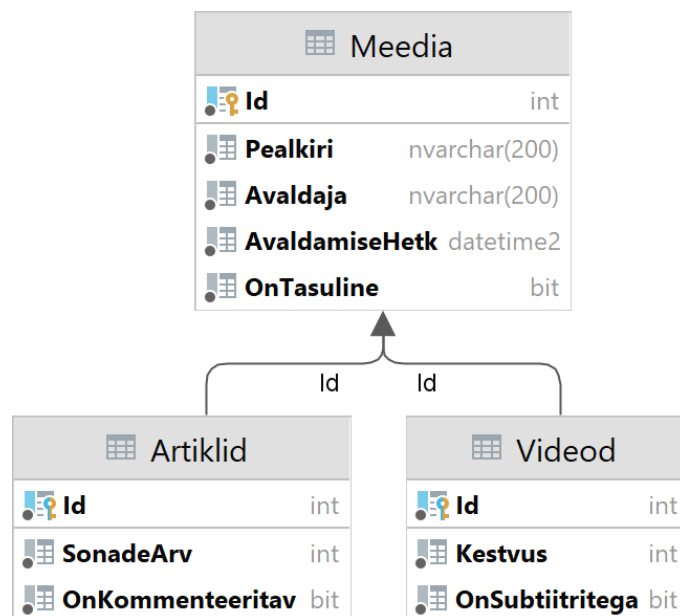
Eelised klassitabeli päriluse vastendusskeemi kasutamisel on järgmised:

- Andmebaasi tabelid on lihtsamini arusaadavad ega kuluta tarbetult andmeruumi, kuna iga tabeli veerg on selle tabeli vaates oluline [19].
- Andmebaasi ja domeenimudeli vaheline suhe on üheselt mõistetav [11].
- Ei vaja alamklasside eristamiseks eraldi tunnusvälja.

Puudused klassitabeli päriluse vastendusskeemi kasutamisel on järgmised [11]:

- Objekti laadimine hõlmab mitme tabeliga tegelemist ehk ühendpäringute või mitme päringu tegemist ja mõnedel juhtudel nende tulemuste mälus sidumist.
- Igasugune väljade hierarhias tõstmine või langetamine põhjustab andmebaasis muudatusi.
- Ülemklassi tabelid võivad jõudluse osas osutada pudelikaelaks.
- Kõrge normaliseerituse tõttu võivad kindlaotstarbelised päringud olla raskesti arusaadavad.

Klassitabeli päriluse vastendusskeemi kasutatavat andmebaasi diagrammi esitab joonis 9.



Joonis 9. Andmebaasi disaini diagramm klassitabeli päriluse vastenduskeemi muustrile.

Uue kirje lisamine vajab üldjuhul kahte eraldi päringut. Esmalt kirje ülemklassi lisamise päringut esitab joonis 10.

```

INSERT INTO Meedia (Pealkiri, Avaldaja, AvaldamiseHetk, OnTasuline)
VALUES ('Artikkel 1', 'Avaldaja 1', '2022-01-01', 1),
      ('Artikkel 2', 'Avaldaja 2', '2022-01-02', 0),
      ('Video 1', 'Avaldaja 3', '2022-01-02', 1),
      ('Video 2', 'Avaldaja 3', '2022-01-03', 0)
  
```

Joonis 10. Ülemklassi nädisandmete lisamise päring klassitabeli päriluse vastenduskeemi kasutades.

Seejärel kirje alamklassidele vastavatesse tabelitesse lisamise päringuid esitab joonis 11.

```

INSERT INTO Artiklid (Id, SonadeArv, OnKommenteeritav)
VALUES (1, 100, 1),
      (2, 50, 1)

INSERT INTO Videod (Id, Kestvus, OnSubtiitritega)
VALUES (3, 30000, 0),
      (4, 60000, 1)
  
```

Joonis 11. Alamklasside nädisandmete lisamise päringud klassitabeli päriluse vastenduskeemi kasutades.

Üksiku alamklassi kõigi andmete pärimiseks tuleb selle ülemklassi andmete laadimiseks kasutada ühendpäringut ülemklassile vastava tabeliga (Joonis 12).

```

SELECT m.Id,
       m.Pealkiri,
       m.Avaldaja,
       m.AvaldamiseHetk,
       m.OnTasuline,
       a.SonadeArv,
       a.OnKommenteeritav
FROM Artiklid AS a
     LEFT JOIN Meedia AS m ON a.Id = m.Id

```

Joonis 12. Ühe alamklassi andmete päring klassitabeli päriluse vastendusskeemi kasutades.

Uue alamklassi kasutuselevõtu puhul tuleb klassitabeli päriluse puhul lisada ka uus andmebaasitabel, mis sisaldaks ainult selle alamklassi spetsiifilisi väljasid (Joonis 13).

```

CREATE TABLE Fotod
(
    Id          int          not null
              constraint Fotod_pk
              primary key
              constraint Fotod_Meedia_Id_fk
              references Meedia,
    Fotograaf  nvarchar(200) not null,
    Korgus     int          not null,
    Laius      int          not null
)

```

Joonis 13. Uue alamklassi toe lisamise päring klassitabeli päriluse vastendusskeemi kasutades.

3.3 Konkreetse tabeli pärilus

Konkreetse tabeli päriluse eestikeelsed nimetused on tabel konkreetse klassi kohta ja lehetabeli pärilus. Inglisekeelseteks nimetusteks on *Concrete Table Inheritance*, *Table per Concrete Class* ja *Leaf Table Inheritance* [11], [21].

Konkreetse tabeli päriluse vastendusskeemi tööpõhimõte seisneb klasside pärilushierarhia esitamises ühe tabeliga iga konkreetse klassi kohta. Lisaks konkreetse alamklassi väljadele sisaldab tabel teiste alamklassidega ühise ülemklassi väljasid [19].

Kuna alamklassidele vastavad andmebaasitabelid ei ole omavahel seotud, siis võib esineda probleem, kus erinevate tabelite primaarvõtmed kattuvad. Üheks lahenduseks on iga tabeli identiteediveeru jaoks erinevate algväärtuse määramine selliselt, et suurte

algväärtuste vahemike tõttu ja andmete hulka arvesse võttes tabelitevahelisi identiteediveeru väärtuste duplikaate ei tekiks [11]. Teiseks variandiks on andmebaasipoolse identiteedi funktsionaalsuse väljalülitamine ning primaarvõtmete genereerimise programmikoodile delegeerimine [21].

Eelised konkreetse tabeli päriluse vastenduskeemi kasutamisel on järgmised:

- Tabeliveerud sisaldavad vastava klassi kogu informatsiooni [19].
- Alamklassi andmete salvestamine valele alamklassile vastavasse tabelisse ei ole võimalik [19].
- Iga tabeli andmeid päritakse vaid sellele vastava klassi väljade andmeid pärides.
- Ei vaja alamklasside eristamiseks eraldi tunnusvälja.
- Ainult alamklassi tarvis andmeid pärides ei tehta ühendpäringuid [11].

Puudused konkreetse tabeli päriluse vastenduskeemi kasutamisel on järgmised:

- Primaarvõtmete haldamine võib osutuda keeruliseks [11].
- Igasugune väljade hierarhias tõstmine või langetamine põhjustab andmebaasis muudatusi, ent siiski mitte nii olulisel määral kui klassitabeli päriluse puhul [11].
- Ülemklassi välja muutmisel tuleb muudatus teha ka kõigis tabelites, milledes see väli sisaldub, kuna ülemklassi väljad on duplitseeritud üle tabelite [21].
- Metaandmete puudumise tõttu ei ole võimalik ainult tabelit vaadates alamklassi veerge ülemklassi omadest eristada [19].
- Andmebaasisuhete jõustamine abstraktsete klasside puhul ei ole võimalik [11].
- Kogu pärilushierarhia ülemklassi kohta päringute tegemine vajab kõikide tabelite kontrollimist, mis põhjustab mitmekordset andmebaasi poole pöördumist või vajadust keeruliste ühendpäringute järele [11].

Klasside konkreetse tabeli päriluse vastenduskeemi kasutades loodud andmebaasi diagrammi esitab joonis 14.

Artiklid	
Id	int
Pealkiri	nvarchar(200)
Avaldaja	nvarchar(200)
AvaldamiseHetk	datetime2
OnTasuline	bit
SonadeArv	int
OnKommenteeritav	bit

Videod	
Id	int
Pealkiri	nvarchar(200)
Avaldaja	nvarchar(200)
AvaldamiseHetk	datetime2
OnTasuline	bit
Kestvus	int
OnSubtiitritega	bit

Joonis 14. Andmebaasi disaini diagramm konkreetse tabeli päriluse vastendusskeemi muustrile.

Kirjete lisamise päringud sisaldavad lisaks alamklassi andmetele ka ülemklassi andmeid (Joonis 15).

```

INSERT INTO Artiklid (Pealkiri, Avaldaja, AvaldamiseHetk,
                    OnTasuline, SonadeArv, OnKommenteeritav)
VALUES ('Artikkel 1', 'Avaldaja 1', '2022-01-01', 1, 100, 1),
      ('Artikkel 2', 'Avaldaja 2', '2022-01-02', 0, 50, 1)

INSERT INTO Videod (Pealkiri, Avaldaja, AvaldamiseHetk,
                   OnTasuline, Kestvus, OnSubtiitritega)
VALUES ('Video 1', 'Avaldaja 3', '2022-01-02', 1, 3000, 0),
      ('Video 2', 'Avaldaja 3', '2022-01-03', 0, 6000, 1)

```

Joonis 15. Näidisandmete lisamise päringud konkreetse tabeli päriluse vastendusskeemi kasutades.

Kui üksikutele alamklassidele vastavatest tabelitest andmete pärimine toimib kõige tavalisema SELECT lausega (Joonis 16), siis kõigi alamklasside jaoks ühise ülemklassi andmete pärimiseks tuleb kasutada UNION ALL klauslit iga pärilushierarhiasse kuuluva alamklassi kohta.

```

SELECT Id, Pealkiri, Avaldaja,
       AvaldamiseHetk, OnTasuline, SonadeArv,
       OnKommenteeritav
FROM Artiklid

```

Joonis 16. Ühe alamklassi andmete päring konkreetse tabeli päriluse vastendusskeemi kasutades.

Ülemklassi andmete pärimise päringut esitab joonis 17.

```

SELECT Id, Pealkiri, Avaldaja,
       AvaldamiseHetk, OnTasuline
FROM Artiklid
UNION ALL
SELECT Id, Pealkiri, Avaldaja,
       AvaldamiseHetk, OnTasuline

FROM Videod

```

Joonis 17. Ülemklassi andmete päring konkreetse tabeli päriluse vastendusskeemi kasutades.

Konkreetse tabeli päriluse vastendusskeemi kasutamisel tuleb pärilushierarhiasse uue alamklassi lisamisel luua ka uus andmebaasitabel, mis sisaldaks lisaks lisatava alamklassi spetsiifilistele väljadele ka ülemklassi väljasid (Joonis 18). Et vältida identiteediprobleeme, seatakse primaarvõtme algväärtus vaikumisi väärtusest suurem.

```

CREATE TABLE Fotod
(
    Id                int identity (2000000001 ,1)
                    constraint Fotod_pk
                    primary key,
    Pealkiri          nvarchar(200) not null,
    Avaldaja          nvarchar(200) not null,
    AvaldamiseHetk    datetime2     not null,
    OnTasuline        bit            not null,
    Korgus            int            not null,
    Laius             int            not null
)

```

Joonis 18. Uue alamklassi toe lisamise päring konkreetse klassi päriluse vastendusskeemi kasutades.

4 Eksperiment

Käesolev peatükk käsitleb lõputöö käigus läbiviidava eksperimendi eesmärke, kirjeldust, kasutatavat metoodikat, mõõtmisi ja nende tulemuste dispersioonanalüüsi.

4.1 Eksperimendi eesmärk

Kuigi päriluse vastendusskeemide tööpõhimõtet on kajastatud erialakirjanduses ja mitmetes teadustöodes, siis konkreetsete jõudlustestide kättesaadavus on äärmiselt piiratud. Üldiseid soovitusi skeemide olukorrapõhisteks kasutamisteks esitavad mitmed allikad [11], [19], kuid ei kaasa neisse jõudlustestide ega muid arvulisi andmeid.

Kõige põhjalikum leitud viide arvandmeid esitanud jõudlustestile on Microsoft Entity Framework Core'i dokumentatsioon [22]. Selles esitatakse üksiktabeli ja klassitabeli pärilusvastendusskeemide võrdluse tulemusi Entity Framework Core objekt-relatsioonivastendaja kasutamisel, mille puhul on tegemist Microsofti loodud relatsioonide kaardistamise raamistikuga. Sellel jõudlustestil esineb käesoleva töö tarvis järeltulemuste tegemiseks mitmeid puudujääke:

- Võrdlusesse ei ole kaasatud konkreetse tabeli päriluse vastendusskeemi, mille kasutamist Entity Framework Core ei toeta.
- Kasutatavad päringud on genereeritud automaatselt ORM-i poolt üheainsa pärilushierarhia tarbeks, mistõttu võivad tulemused teistsuguste pärilushierarhiate korral erineda.
- Mõõtmistulemustes ei ole kajastatud vaid pärilushierarhiate endi jõudlust vaid ka Entity Framework Core'i vastendamisprotsessi jõudlust.

Eksperimendi eesmärgiks on võtta arvesse selle testi puudujääke ning viia läbi kolme vastendusskeemi oluliselt põhjalikumad jõudlustestid ja tulemuste analüüs.

4.2 Eksperimendi kirjeldus

Eksperimendi käigus viiakse läbi jõudlustestid kolme erineva andmebaasi disainilahendusega, kasutades klasside ja tabelite vastendamiseks eelmises peatükis kaetud päriluse vastendusskeeme (üksiktabeli-, klassi- ja konkreetse tabeli vastendusskeemid). Pärilushierarhiatena kasutatakse nelja sarnast klassimudelit, millest esimesel on kokku kaks, teisel neli, kolmandal kaheksa ning neljandal kuusteist alamklassi.

Eksperimendi planeerimisel, läbiviimisel ja tulemuste analüüsimisel lähtutakse USA Riikliku Standardibüroo (NBS – *National Bureau of Standards*) teadusväljaandes käsitletud andmebaasi võrdlustestide metoodikast [23], mis jaguneb kolmeks peamiseks osaks:

- Jõudlustesti disain – jõudlustesti läbiviiva keskkonna ülesseadmine, mis hõlmab süsteemi konfiguratsiooni, testandmete ja -koormuste ning uurimisaluste muutujate valikut.
- Jõudlustesti läbiviimine – jõudluse mõõtmiste läbiviimine ja andmete kogumine.
- Jõudlustesti analüüs – jõudlustesti mõõtmistulemuste analüüsimine ja jõudluse omavaheline võrdlemine mitme testitava süsteemi korral.

Andmebaasisüsteemina on eksperimendi korraldamiseks kasutusel Microsoft SQL Server 2019, mille valiku põhjuseks on autori varasem ulatuslik kogemus sellega töötamisel ning igapäevane kasutamine tööülesannete lahendamisel. Mõõtmiste läbiviimiseks kasutatava arvuti tehnilised andmed on 32 GB muutmälu, 6 tuuma ja 12 lõimega protsessor ning Windows 11 operatsioonisüsteem.

Vastendusskeemide jõudlust võrreldakse SELECT päringute abil, pärides andmeid nii üksikutele alamklassidele kui ka ülemklassidele vastavatest tabelitest. Samuti võrreldakse eraldi väiksema ja suurema mahuga andmekogusid. Andmemahud ulatuvad alates 800 000 kuni 6 400 000 eksemplarini pärilushierarhia kohta ning nende suuruste valikul on arvestatud, et päringute kestvused erinevate konfiguratsioonide lõikes üksteisest piisavalt erineksid.

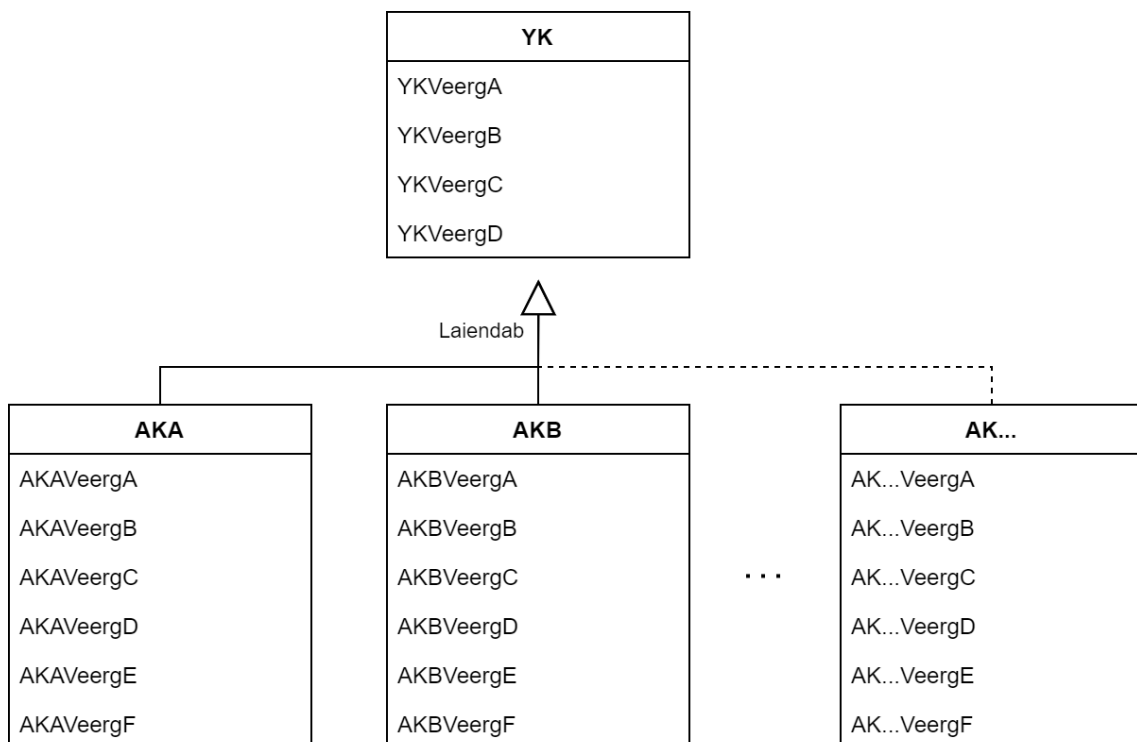
Kõik mõõtmised sooritatakse viiel korral ning iga päringu kestvus arvutatakse mõõtmistulemuste geomeetrilise keskmisena, mis tulenevalt täitmistevahelistest

sõltuvustest sobib aritmeetilise keskmisega võrreldes paremini. Täitmised on omavahel sõltuvad, kuna andmebaasisüsteem kasutab päringute täitmisel täitmisplaane ja osa andmete vahemällu laadimist, mistõttu sama päringu kestvus korduvatel täitmistel väheneb.

Mõõtmistulemuste põhjal viiakse läbi dispersioonanalüüs päriluse vastenduskeemide omavaheliste oluliste jõudluserinevuste määramiseks ning tehakse nende alusel järeldused.

4.3 Andmemudelite projekteerimine

Eksperimendi tarvis luuakse neli sarnast klassimudelit (Joonis 19), mille põhjal luuakse iga pärilushierarhia jaoks erinevate alamklasside arvudega andmebaasiskeemid. Iga alamklassi väljade arv ja nende tüübid on identsed, et vältida nende erinevustest tingitud mõju avaldumist mõõtmistulemustele. Pärilushierarhiate alamklassid nimetatakse skeemi „AK...“ järgi, milles „...“ tähistab ladina tähestiku tähti A-Z vastavalt alamklasside arvule pärilushierarhias.



Joonis 19. Eksperimendi klassimudeli näidis.

Klassiväljade nimed järgivad skeemi „... Veerg_“, milles „...“ esitab alamklassi nime ning „_“ esitab välja nime lühikuju. Näiteks AKCVeergF tähistab vähemalt kolme alamklassiga pärilushierarhia kolmanda klassi kuuendat välja.

Üksiktabeli päriluse korral lisatakse enama arvu alamklassidega pärilushierarhiate puhul andmebaasitabelisse K tabeliveerud lisanduvate alamklasside klassiväljade jaoks. Klassitabeli ja konkreetse tabeli päriluse korral lisanduvad andmebaasiskeemi alamklasside arvu suurenemisel nendele klassidele vastavad andmebaasitabelid.

Kahe alamklassiga üksiktabeli pärilushierarhia andmebaasimudelit esitab joonis 20.

K	
Id	int
YKVeergA	nvarchar(200)
YKVeergB	nvarchar(200)
YKVeergC	nvarchar(200)
YKVeergD	datetime2
KTunnus	smallint
AKAVeergA	nvarchar(200)
AKAVeergB	nvarchar(200)
AKAVeergC	nvarchar(200)
AKAVeergD	nvarchar(1000)
AKAVeergE	nvarchar(400)
AKAVeergF	nvarchar(200)
AKBVeergA	nvarchar(200)
AKBVeergB	nvarchar(200)
AKBVeergC	nvarchar(200)
AKBVeergD	nvarchar(1000)
AKBVeergE	nvarchar(400)
AKBVeergF	nvarchar(200)

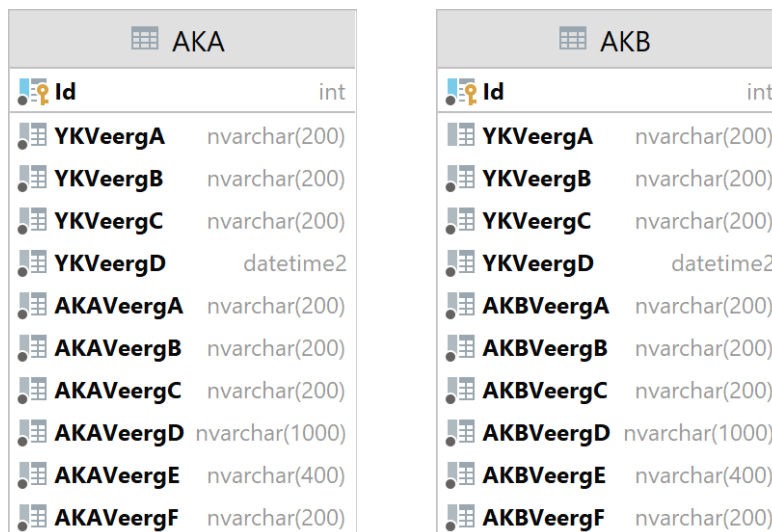
Joonis 20. Eksperimendi andmebaasi diagramm kahe alamklassi ja üksiktabeli vastenduskeemiga.

Kahe alamklassiga klassitabeli pärilushierarhia andmebaasimudelit esitab joonis 21.

YK		AKA		AKB	
Id	int	Id	int	Id	int
YKVeergA	nvarchar(200)	AKAVeergA	nvarchar(200)	AKBVeergA	nvarchar(200)
YKVeergB	nvarchar(200)	AKAVeergB	nvarchar(200)	AKBVeergB	nvarchar(200)
YKVeergC	nvarchar(200)	AKAVeergC	nvarchar(200)	AKBVeergC	nvarchar(200)
YKVeergD	datetime2	AKAVeergD	nvarchar(1000)	AKBVeergD	nvarchar(1000)
		AKAVeergE	nvarchar(400)	AKBVeergE	nvarchar(400)
		AKAVeergF	nvarchar(200)	AKBVeergF	nvarchar(200)

Joonis 21. Eksperimendi andmebaasi diagramm kahe alamklassi ja klassitabeli vastenduskeemiga.

Kahe alamklassiga konkreetse tabeli pärilushierarhia andmebaasimudelit esitab joonis 22.



Joonis 22. Eksperimendi andmebaasi diagramm kahe alamklassi ja konkreetse tabeli vastenduskeemiga.

Andmebaasiskeeme luuakse kokku 48 (iga pärilushierarhia kohta 16) ning need nimetatakse vastavalt skeemile „scnn_rcmmmmk_hier“, milles „nn“ esitab andmemudeli alamklasside arvu, „mmmm“ esitab pärilushierarhia andmemahtu alamklasside eksemplaride arvuna tuhandetes ning „hier“ esitab kasutatavat hierarhiat (SITI – *single table inheritance* – üksiktabeli pärilus, CLTI – *class table inheritance* – klassitabeli pärilus ja COTI – *concrete table inheritance* – konkreetse tabeli pärilus). Näiteks klassitabeli pärilust kasutava 16 alamklassi ja 800 000-eksemplarise andmemahuga andmebaasiskeemi nimeks on sc16_rc0800k_clti.

Tulenevalt suurest andmebaasiskeemide arvust ja erinevate pärilushierarhiate konfiguratsioonidest luuakse töö autori arendatud Pythoni programmi abil SQL skriptid, mis sisaldavad andmebaasiskeemide, -tabelite ja -indeksite loomise lauseid.

4.4 Testandmete genereerimine

Andmebaasi täitmiseks vajalike testandmete genereerimiseks kasutakse Mockaroo-nimelist veebiteenust, millega on võimalik luua erinevate semantikatega testandmete kogusid ning neid erinevates vormingutes eksportida. Kuna veebirakenduse tasuta versioon võimaldab eksportida korraga vaid tuhanderealisi andmekogusid, siis

kasutatakse testandmete mahu suurendamiseks töö autori loodud Pythoni programmi. See võimaldab Mockaroost eksporditud andmete põhjal genereerida ja ette valmistada juhuslikud, kuid semantiliselt siiski korrektsed andmekogud iga testitava andmebaasiskeemi jaoks. Testandmete genereerimisel ja vormindamisel võtab programm arvesse kasutatavat pärilushierariat, alamklasside arvu ning alamklassidele vastavat andmehulka. CSV (ingl *Comma Separated Values* – komaeraldusega väärtused) vormingus faili genereeritud andmed imporditakse andmebaasi kasutades SQL-i BULK INSERT lauseid.

Semantiliselt esitavad testandmed veebilehtedel avaldatud kuulutuste pärilushierariaid, milles ülemklass esitab kuulutuse metaandmeid ning alamklassid esitavad kuulutuse liikide, näiteks töö- või müügikuulutuste, spetsiifilisemaid andmeid. Kõigis jõudlustestides esitavad kuulutuse alamklassid oma struktuurilt ja semantikalt töökuulutust, et vältida võimalikku andmetüüpide ja -mahtude erinevustest tulenevat mõju päringute kestvustele. Genereeritud testandmete struktuur ja semantika on järgnev:

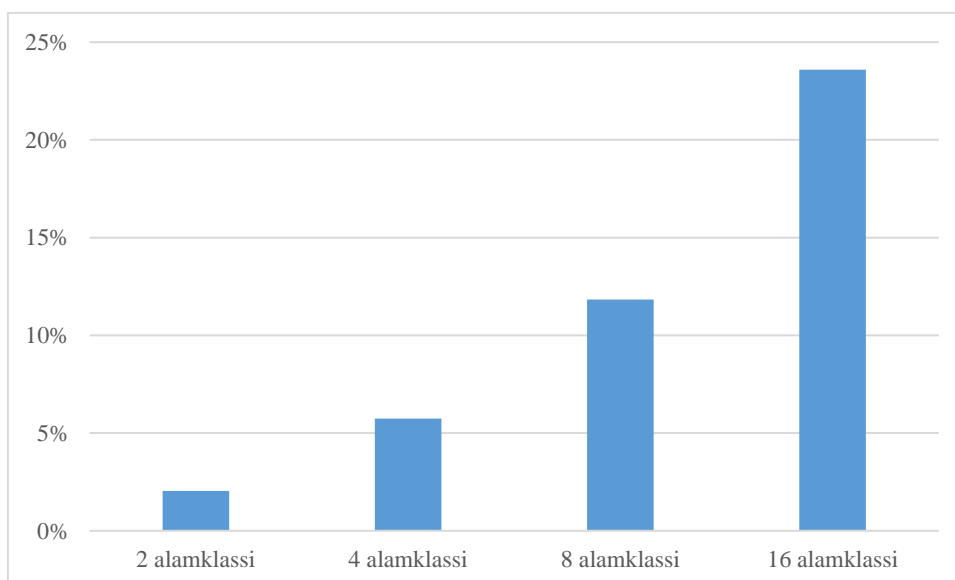
- YKVeergA: kuni 200 tähemärgiga kuulutuse koostaja eesnimi;
- YKVeergB: kuni 200 tähemärgiga kuulutuse koostaja perenimi;
- YKVeergC: kuni 200 tähemärgiga kuulutuse avaldamise veebilehe aadress;
- YKVeergD: kuulutuse avaldamise kuupäev vahemikus 01.01.2015 – 31.12.2021;
- AK...VeergA: kuni 200 tähemärgiga töökuulutuse ameti nimetus;
- AK...VeergB: kuni 200 tähemärgiga töökuulutuse ettevõtte nimi;
- AK...VeergC: kuni 200 tähemärgiga töökuulutuse töökoha aadress;
- AK...VeergD: kuni 1000 tähemärgiga töökuulutuse tekst lausetena;
- AK...VeergE: kuni 400 tähemärgiga töökuulutuse kontaktisiku ees- ja perenimi;
- AK...VeergF: kuni 200 tähemärgiga töökuulutuse kontaktisiku e-postiaadress;

Pärilushierarhiate klassieksemplaride arvu ühe alamklassi kohta esitab tabel 3 kogu pärilushierarhia andmemahu ja alamklasside arvu kombinatsioonidena.

Tabel 3. Testandmete mahud

		Pärilushierarhia klassieksemplaride arv			
		800 000	1 600 000	3 200 000	6 400 000
Alamklasside arv	2	400 000	800 000	1 600 000	3 200 000
	4	200 000	400 000	800 000	1 600 000
	8	100 000	200 000	400 000	800 000
	16	50 000	100 000	200 000	400 000

Salvestusruume kõigi vastenduskeemide ja konfiguratsioonide korral esitavad lisa 2 tabelid 34–36. Kõigi konfiguratsioonide korral on väikseima salvestusruumiga konkreetse tabeli pärilus ja suurimaga üksiktabeli pärilus. Üksik- ja klassitabeli päriluse vastenduskeemidel erinevad need kõigi konfiguratsioonide puhul 2%. Üksiktabeli ja konkreetse tabeli salvestusruumide erinevused ulatuvad kahest protsendist väikseima alamklasside arvu korral kuni 24 protsendini suurima alamklasside arvu korral (Joonis 23).



Joonis 23. Üksiktabeli ja konkreetse tabeli päriluse vastenduskeemide salvestusruumide erinevused.

4.5 Mõõdetavad päringud

Eksperimendi käigus käivitatakse iga andmebaasiskeemi kohta neli erinevat lugemispäringut, mille abil mõõdetakse vastendusskeemide valiku mõju päringute kestvustele. Iga päriluse vastendusskeemi päringud esitatakse nelja alamklassi ja 800 000-ekseplarise andmemahuga (sc04_rc0800k_hier) andmebaasiskeemi näitel.

4.5.1 Esimene päring

Esimeseks päringuks (tähistusega Q1) on üksiku alamklassi andmete pärimine vastavalt sama alamklassi ühe klassivälja kohta käivale tingimusele. Päritavaks alamklassiks on AKA ning seatavaks tingimuseks alamklassi klassivälja AKAVeergA väärtuse sarnasus sõnega „Manager“, kasutades SQL klauslit LIKE.

Semantiliselt esitab päring kõigi selliste töökuulutuste pärimist, mille puhul sisaldub ametinimetuses alamsõne „Manager“. Andmebaasikäske esitab tabel 4.

Tabel 4. Esimese päringu (Q1) andmebaasikäsud.

Üksiktabeli pärilus	Klassitabeli pärilus	Konkreetse tabeli pärilus
<pre>SELECT id, YKVeergA, YKVeergB, YKVeergC, YKVeergD, AKAVeergA, AKAVeergB, AKAVeergC, AKAVeergD, AKAVeergE, AKAVeergF FROM sc04_rc0800k_siti.K AS k WHERE k.KTunnus = 1 AND k.AKAVeergA LIKE '%'Manager%'</pre>	<pre>SELECT yk.id, yk.YKVeergA, yk.YKVeergB, yk.YKVeergC, yk.YKVeergD, aka.AKAVeergA, aka.AKAVeergB, aka.AKAVeergC, aka.AKAVeergD, aka.AKAVeergE, aka.AKAVeergF FROM sc04_rc0800k_clti.AKA AS aka LEFT JOIN sc04_rc0100k_clti.YK AS yk ON aka.Id = yk.Id WHERE aka.AKAVeergA LIKE '%'Manager%'</pre>	<pre>SELECT id, YKVeergA, YKVeergB, YKVeergC, YKVeergD, AKAVeergA, AKAVeergB, AKAVeergC, AKAVeergD, AKAVeergE, AKAVeergF FROM sc04_rc0800k_coti.AKA AS aka WHERE aka.AKAVeergA LIKE '%'Manager%'</pre>

4.5.2 Teine päring

Teiseks päringuks (tähistusega Q2) on üksiku alamklassi andmete pärimine vastavalt ülemklassi ühe klassivälja kohta käivale tingimusele. Päritavaks alamklassiks on AKB

ning seatavaks tingimuseks on ülemklassi klassivälja YKVeergC väärtuse vastavus sõnega „https://nasa.gov“.

Semantiliselt esitab päring kõigi selliste töökuulutuste pärimist, mille puhul on kuulutuse avaldamiskohaks veebileht aadressiga „https://nasa.gov“. Andmebaasikäske esitab tabel 5.

Tabel 5. Teise päringu (Q2) andmebaasikäsud.

Üksiktabeli pärilus	Klassitabeli pärilus	Konkreetse tabeli pärilus
<pre>SELECT Id, YKVeergA, YKVeergB, YKVeergC, YKVeergD, AKBVeergA, AKBVeergB, AKBVeergC, AKBVeergD, AKBVeergE, AKBVeergF FROM sc04_rc0800k_siti.K AS k WHERE k.KTunnus = 2 AND k.YKVeergC = 'https://nasa.gov'</pre>	<pre>SELECT yk.Id, yk.YKVeergA, yk.YKVeergB, yk.YKVeergC, yk.YKVeergD, akb.AKBVeergA, akb.AKBVeergB, akb.AKBVeergC, akb.AKBVeergD, akb.AKBVeergE, akb.AKBVeergF FROM sc04_rc0800k_clti.AKB AS akb LEFT JOIN sc04_rc0800k_clti.YK AS yk ON akb.Id = yk.Id WHERE yk.YKVeergC = 'https://nasa.gov'</pre>	<pre>SELECT Id, YKVeergA, YKVeergB, YKVeergC, YKVeergD, AKBVeergA, AKBVeergB, AKBVeergC, AKBVeergD, AKBVeergE, AKBVeergF FROM sc04_rc0800k_coti.AKB AS akb WHERE akb.YKVeergC = 'https://nasa.gov'</pre>

4.5.3 Kolmas päring

Kolmandaks päringuks (tähistusega Q3) on ülemklassi ühe andmevälja 30 enim esineva väärtuse ja nende esinemise arvu kahanevas järjekorras pärimine. Järjestamise aluseks võetavaks ülemklassi klassiväljaks on YKVeergC.

Semantiliselt esitab päring 30 enim esinevat kuulutuste avaldamiskohta ja nende esinemise sagedused. Andmebaasikäske esitab tabel 6.

Tabel 6. Kolmanda päringu (Q3) andmebaasikäsud.

Üksiktabeli pärilus	Klassitabeli pärilus	Konkreetsel tabeli pärilus
<pre>SELECT TOP 30 YKVeergC, COUNT(*) AS Sagedus FROM sc04_rc0800k_siti.K GROUP BY YKVeergC ORDER BY COUNT(*) DESC</pre>	<pre>SELECT TOP 30 YKVeergC, COUNT(*) AS Sagedus FROM sc04_rc0800k_clti.YK GROUP BY YKVeergC ORDER BY COUNT(*) DESC</pre>	<pre>SELECT TOP 30 yk.YKVeergC, COUNT(*) AS Sagedus FROM (SELECT YKVeergC FROM sc04_rc0800k_coti.AKA UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKB UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKC UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKD) AS yk GROUP BY yk.YKVeergC ORDER BY COUNT(*) DESC</pre>

4.5.4 Neljas päring

Neljandaks päringuks (tähistusega Q4) on ülemklassi ühe andmevälja kõigi unikaalsete väärtuste pärimine. Päritavaks unikaalseks ülemklassi klassiväljaks on YKVeergC.

Semantiliselt esitab päring kõik unikaalsed kuulutuste avaldamiskohad. Andmebaasikäske esitab tabel 7.

Tabel 7. Neljanda päringu (Q4) andmebaasikäsud.

Üksiktabeli pärilus	Klassitabeli pärilus	Konkreetse tabeli pärilus
SELECT DISTINCT YKVeergC FROM sc04_rc0800k_siti.K	SELECT DISTINCT YKVeergC FROM sc04_rc0800k_clti.YK	SELECT DISTINCT yk.YKVeergC FROM (SELECT YKVeergC FROM sc04_rc0800k_coti.AKA UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKB UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKC UNION ALL SELECT YKVeergC FROM sc04_rc0800k_coti.AKD) AS yk

4.6 Eksperimendi mõõtmistulemused

Eksperimendi mõõtmistulemused koondatakse vastavalt mallile tabelitesse ning esitatakse viie järjestikuse mõõtmise geomeetrilise keskmisena (Tabel 8). Päringute kestvused esitatakse millisekundites.

Tabel 8. Mõõtmistulemuste esitamise mall.

Qn		Pärilushierarhia klassieksemplaride arv tuhandetes			
		800	1 600	3 200	6 400
Alamklasside arv	2				
	4				
	8				
	16				

Tulemuste esitamisel kasutatakse vastendusskeemideülest värviskaalade gradient tingimusvormingut, mille puhul esitavad rohelised värvitoonid lühemaid, punased pikemaid ning kollased ja oranžid vahepealseid päringukestvusi.

4.6.1 Esimese päringu tulemused

Tabel 9 esitab esimese päringu üksiktabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 9. Esimese päringu (Q1) mõõtmistulemused üksiktabeli päriluse (SITI) korral.

Q1		Pärilushierarhia klassieksemplaride arv tuhandetes			
		SITI	800	1 600	3 200
Alamklasside arv	2	2299,4	3986,3	9319,3	18609,4
	4	1135,6	2260,6	4420,0	9191,7
	8	160,4	968,8	2009,8	4023,6
	16	144,2	606,0	1143,6	2204,8

Tabel 10 esitab esimese päringu klassitabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 10. Esimese päringu (Q1) mõõtmistulemused klassitabeli päriluse (CLTI) korral.

Q1		Pärilushierarhia klassieksemplaride arv tuhandetes			
		CLTI	800	1 600	3 200
Alamklasside arv	2	953,6	1883,3	3797,5	7577,6
	4	507,3	977,0	2017,9	3742,6
	8	249,7	470,2	1000,9	1942,8
	16	115,0	260,8	518,7	1017,8

Tabel 11 esitab esimese päringu konkreetse tabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 11. Esimese päringu (Q1) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.

Q1		Pärilushierarhia klassieksemplaride arv tuhandetes			
		COTI	800	1 600	3 200
Alamklasside arv	2	937,4	1806,9	3537,5	7032,7
	4	451,4	877,7	1802,7	3558,4
	8	234,9	461,2	904,8	1785,9
	16	117,5	229,7	446,0	916,6

4.6.2 Teise päringu tulemused

Tabel 12 esitab teise päringu üksiktabeli päriluse vastendusskeemi mõõtmistulemused.

Tabel 12. Teise päringu (Q2) mõõtmistulemused üksiktabeli päriluse (SITI) korral.

Q2		Pärilushierarhia klassieksemplaride arv tuhandetes			
		SITI	800	1 600	3 200
Alamklasside arv	2	107,8	101,0	207,7	431,9
	4	106,2	97,5	188,0	381,0
	8	107,3	96,9	177,3	381,8
	16	118,1	92,6	181,8	358,1

Tabel 13 esitab teise päringu klassitabeli päriluse vastendusskeemi mõõtmistulemused.

Tabel 13. Teise päringu (Q2) mõõtmistulemused klassitabeli päriluse (CLTI) korral.

Q2		Pärilushierarhia klassieksemplaride arv tuhandetes			
		CLTI	800	1 600	3 200
Alamklasside arv	2	37,8	76,4	153,9	319,7
	4	21,2	43,4	90,4	194,0
	8	20,5	33,4	62,2	136,9
	16	19,9	29,2	57,0	115,3

Tabel 14 esitab teise päringu konkreetse tabeli päriluse vastendusskeemi mõõtmistulemused.

Tabel 14. Teise päringu (Q2) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.

Q2		Pärilushierarhia klassieksemplaride arv tuhandetes			
		COTI	800	1 600	3 200
Alamklasside arv	2	54,5	109,6	113,2	253,2
	4	28,0	59,2	107,1	108,5
	8	14,3	27,4	53,8	108,7
	16	12,1	14,5	31,5	54,7

4.6.3 Kolmanda päringu tulemused

Tabel 15 esitab kolmanda päringu üksiktabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 15. Kolmanda päringu (Q3) mõõtmistulemused üksiktabeli päriluse (SITI) korral.

Q3		Pärilushierarhia klassieksemplaride arv tuhandetes			
		SITI	800	1 600	3 200
Alamklasside arv	2	117,1	103,6	180,9	341,6
	4	112,5	103,0	186,7	349,7
	8	121,1	111,7	196,9	377,2
	16	132,7	123,5	216,7	402,9

Tabel 16 esitab kolmanda päringu klassitabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 16. Kolmanda päringu (Q3) mõõtmistulemused klassitabeli päriluse (CLTI) korral.

Q3		Pärilushierarhia klassieksemplaride arv tuhandetes			
		CLTI	800	1 600	3 200
Alamklasside arv	2	32,9	49,3	76,5	129,3
	4	32,6	47,9	75,3	130,6
	8	30,9	48,9	76,0	129,5
	16	33,0	48,1	78,5	131,9

Tabel 17 esitab kolmanda päringu konkreetse tabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 17. Kolmanda päringu (Q3) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.

Q3		Pärilushierarhia klassieksemplaride arv tuhandetes			
		COTI	800	1 600	3 200
Alamklasside arv	2	108,7	195,5	170,9	304,0
	4	109,3	200,1	377,5	311,5
	8	115,1	203,0	386,2	725,9
	16	118,9	216,8	407,5	743,8

4.6.4 Neljanda päringu tulemused

Tabel 18 esitab neljanda päringu üksiktabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 18. Neljanda päringu (Q4) mõõtmistulemused üksiktabeli päriluse (SITI) korral.

Q4		Pärilushierarhia klassieksemplaride arv tuhandetes			
		SITI	800	1 600	3 200
Alamklasside arv	2	100,3	93,5	170,4	325,3
	4	103,3	93,6	174,4	333,5
	8	109,2	97,5	185,0	347,4
	16	123,3	109,8	205,2	387,4

Tabel 19 esitab neljanda päringu klassitabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 19. Neljanda päringu (Q4) mõõtmistulemused klassitabeli päriluse (CLTI) korral.

Q4		Pärilushierarhia klassieksemplaride arv tuhandetes			
		CLTI	800	1 600	3 200
Alamklasside arv	2	19,9	35,4	64,9	120,8
	4	19,3	35,2	62,0	126,5
	8	20,7	35,3	63,9	120,5
	16	20,4	34,3	63,0	122,9

Tabel 20 esitab neljanda päringu konkreetse tabeli päriluse vastenduskeemi mõõtmistulemused.

Tabel 20. Neljanda päringu (Q4) mõõtmistulemused konkreetse tabeli päriluse (COTI) korral.

Q4		Pärilushierarhia klassieksemplaride arv tuhandetes			
		COTI	800	1 600	3 200
Alamklasside arv	2	98,5	185,7	153,2	297,4
	4	97,6	184,2	363,6	291,3
	8	117,0	190,5	371,0	715,5
	16	115,0	226,8	372,9	725,0

4.7 Eksperimendi mõõtmistulemuste analüüs

Käesolevas peatükis käsitletakse eksperimendi mõõtmistulemuste analüüsi meetodikat, viiakse mõõtmistulemuste põhjal läbi analüüs ning tehakse sellest järeldused.

4.7.1 Eksperimendi mõõtmistulemuste analüüsi meetodika

Kõigi pärilushierarhiate päringute tulemused jaotatakse nelja gruppi:

- Väike alamklasside arv, väike andmemaht (VV) – vastab 2 – 4 alamklassi ja 800 000 – 1 600 000 eksemplariga konfiguratsiooni tulemustele.
- Väike alamklasside arv, suur andmemaht (VS) – vastab 2 – 4 alamklassi ja 3 200 000 – 6 400 000 eksemplariga konfiguratsiooni tulemustele.
- Suur alamklasside arv, väike andmemaht (SV) – vastab 8 – 16 alamklassi ja 800 000 – 1 600 000 eksemplariga konfiguratsiooni tulemustele.
- Suur alamklasside arv, suur andmemaht (SS) – vastab 8 – 16 alamklassi ja 3 200 000 – 6 400 000 eksemplariga konfiguratsiooni tulemustele.

Analüüsi eesmärk on kontrollida vastenduskeemide vaheliste erinevuste statistilist olulisust, kasutades selleks dispersioonanalüüsi. Analüüsi tulemusena on võimalik näidata, kas vastenduskeemi valik omab mõju päringu kestvusele. Erinevuste ilmnemisel kasutatakse paariviisilist *post hoc* võrdlustesti, et teha kindlaks, täpselt milliste skeemide vahel erinevus ilmneb ning esitatakse ka nende protsentuaalsed suurused.

Dispersioonanalüüsi meetod valitakse iga päringu jaoks kahe alternatiivi seast, milleks on ANOVA (*Analysis of variance*) ühefaktoriline dispersioonanalüüs ja Kruskal–Wallise test. ANOVA puhul on tegemist parameetrilise testiga, mille eeliseks mitteparameetrilise Kruskal–Wallise ees on selle suurem statistiline võimsus. Samas seab ANOVA kasutamine analüüsitavatele andmetele mõned eeldused, nagu näiteks tulemuste homoskedastiivsus ehk püsihajuvus ja tulemuste jaotuse vastavus normaaljaotusele [24].

Paariviisiliseks võrdluseks tehakse valik Tukey HSD (*Honestly Significant Difference* – aus oluline erinevus) ja Dunni võrdlustestide seast. Esimene neist on *post hoc* testiks ANOVA-le ja teine Kruskal–Wallise testile.

Dispersioonanalüüsi meetodite valimiseks viiakse mõõtmistulemustega läbi eelanalüüs, mille käigus määratakse Levene'i testide abil nende püsihajuvus ning Shapiro–Wilki testi

abil kontrollitakse, kas mõõtmistulemused vastavad normaaljaotusele. Lisaks määratakse Pearsoni korrelatsioonikordaja abil alamklasside eksemplaride arvu ja päringute kestvuse vahelised sõltuvused. Nii eelanalüüsi püsihajuvuse ja normaaljaotusele vastavuse kui ka dispersioonanalüüsi käigus statistilise olulisuse määramiseks kasutatakse olulisuse nivood $\alpha = 0,05$. Nii ANOVA kui ka Kruskal–Wallise testi puhul on erinevus statistiliselt oluline siis, kui olulisuse tõenäosus p on väiksem olulisuse nivoo α .

4.7.2 Päringute tulemuste eelanalüüs

Levene'i testi puhul ei ole mõõtmistulemused püsihajuvad ega Shapiro-Wilki testi puhul ei vasta need normaaljaotusele, kui olulisuse tõenäosus p on väiksem olulisuse nivoo α . Päringute mõõtmistulemuste Levene'i ja Shapiro-Wilki testide põhjal osutub kõigi päringute dispersioonanalüüsi meetodi valikuks Kruskal–Wallise test (Lisa 3, Tabelid 37, 39, 41 ja 43).

Kõigi mõõtmiste puhul esineb alamklasside eksemplaride arvu ja päringute kestvuse vahel tugev positiivne sõltuvus (Lisa 3, Tabelid 38, 40, 42 ja 44). Sõltuvus loetakse tugevaks, kui Pearsoni korrelatsioonikordaja jääb 0,5 ja 1,0 vahele [25].

4.7.3 Esimese päringu tulemuste analüüs

Esimese päringu mõõtmistulemustest selgub, et statistiliselt oluliselt erinevad vastendusskeemid kõigil juhtudel välja arvatud suure alamklasside arvu ja väikese andmemahu korral (VV: $p = 0,000$; VS: $p = 0,000$; SV: $p = 0,265$; SS: $p = 0,000$).

Dunni võrdlustesti tulemustest järeldub, et kõigi vastendusskeemide seas, mille korral esineb statistiliselt oluline erinevus, ilmneb see SITI (üksiktabeli pärilushierarhia) ja CLTI (klassitabeli pärilushierarhia) ning SITI ja COTI (konkreetses tabeli pärilushierarhia) vahel (Tabel 21). Protsentuaalset erinevust tähistab tabelis d_r .

Tabel 21. Esimese päringu vastendusskeemide statistiliselt olulised jõudluserinevused.

Q1	SITI – CLTI		SITI – COTI	
	d_r	p	d_r	p
VV	77%	0,000	82%	0,000
VS	83%	0,000	89%	0,000
SS	71%	0,000	79%	0,000

Konfiguratsioonideülelel on halvima jõudlusega üksiktabeli pärilus, erinedes keskmiselt klassitabeli ja konkreetse tabeli pärilustest vastavalt 77% ja 83%. Üksiktabeli päriluse halvimat tulemust selgitab vajadus filtreerida oluliselt suuremat hulka andmebaasitabelite ridu, kuna otsitavas tabelis on lisaks huvipakkuva alamklassi andmetele ka kõigi teiste alamklasside andmed. Parima jõudlusega on konkreetse tabeli pärilus, mispuhul on sellise tulemuse põhjuseks võimalus leida kõik huvipakkuva alamklassi andmed ühest, vaid selle alamklassi andmeid sisaldavast tabelist.

4.7.4 Teise päringu tulemuste analüüs

Teise päringu mõõtmistulemustest selgub, et statistiliselt oluliselt erinevad vastenduskeemid kõigi alamklasside arvu ja andmemahu konfiguratsioonide korral (VV: $p = 0,000$; VS: $p = 0,015$; SV: $p = 0,000$; SS: $p = 0,000$).

Dunni võrdlustesti tulemustest järeldub, et pea kõigi alamklasside arvu ja andmemahu konfiguratsioonide seas ilmneb statistiliselt oluline erinevus SITI ja CLTI ning SITI ja COTI vahel (Tabel 22). Vaid väikese alamklasside arvu ja suure andmemahuga konfiguratsiooni korral ilmneb see ka SITI ja COTI vahel.

Tabel 22. Teise päringu vastenduskeemide statistiliselt olulised jõudluserinevused.

Q2	SITI – CLTI		SITI – COTI	
	d_r	p	d_r	p
VV	79%	0,000	49%	0,049
VS	-		70%	0,013
SV	120%	0,000	144%	0,000
SS	99%	0,007	126%	0,000

Nii nagu esimese päringu puhul, on ka teise päringu tulemustest konfiguratsioonideülelel halvima jõudlusega üksiktabeli pärilus, erinedes keskmiselt klassitabeli ja konkreetse tabeli pärilustest vastavalt 99% ja 97%. Nimetatud tulemus on samuti selgitatav vajadusega töötada päringu kestel läbi oluliselt suuremat hulka andmeid. Parima tulemusega on sarnaselt esimesele päringule konkreetse tabeli pärilus. Üksiktabeli päriluse oluliselt halvemaid tulemusi suuremate andmemahude juures selgitab vajadus filtreerida ülemklassi klassiväljale vastavat andmebaasitabeli veergu. Selles veerus on lisaks huvipakkuva alamklassi metaandmetele ka kõigi teiste alamklasside metaandmed.

Klassitabeli päriluse puhul vähendab jõudlust vajadus teha ühendpäringuid ülemklassile vastavasse andmebaasitabelisse. Konkreetse tabeli päriluse korral tuleb päringuid teha ühteainsasse vaid huvipakkuva alamklassi kõiki andmeid sisaldavasse tabelisse, mis tingib ka paremad tulemused – seda eriti suuremate alamklasside arvu ja andmemahude korral.

4.7.5 Kolmanda päringu tulemuste analüüs

Kolmanda päringu mõõtmistulemustest selgub, et statistiliselt oluliselt erinevad vastenduskeemid kõigi alamklasside arvu ja andmemahu konfiguratsioonide korral (VV: $p = 0,000$; VS: $p = 0,000$; SV: $p = 0,000$; SS: $p = 0,000$).

Dunni võrdlustesti tulemustest järeldub, et kõigi alamklasside arvu ja andmemahu konfiguratsioonide seas ilmneb statistiliselt oluline erinevus SITI ja CLTI ning CLTI ja COTI vahel (Tabel 23). Suure alamklasside arvu ja suure andmemahuga konfiguratsiooni puhul ilmneb see lisaks SITI ja COTI vahel.

Tabel 23. Kolmanda päringu vastenduskeemide statistiliselt olulised jõudluserinevused.

Q3	SITI – CLTI		SITI – COTI		CLTI – COTI	
	d_r	p	d_r	p	d_r	p
VV	91%	0,000	-		117%	0,000
VS	88%	0,000	-		95%	0,000
SV	101%	0,000	-		121%	0,000
SS	97%	0,002	62%	0,022	138%	0,000

Konfiguratsioonideülevalt on halvima jõudlusega konkreetse tabeli pärilus, erinedes keskmiselt üksiktabeli ja klassitabeli pärilustest vastavalt 62% ja 118%. Halvim on selle jõudlus just suure alamklasside arvu ja suure andmemahuga konfiguratsioonidega, mis juhul erineb see klassitabeli päriluse omast 138%. Sellist tulemust selgitab vajadus kasutada päringus iga pärilushierarhia alamklassi jaoks UNION klauslit. Kõigi konfiguratsioonide lõikes on parima tulemusega klassitabeli pärilus.

4.7.6 Neljanda päringu tulemuste analüüs

Neljanda päringu mõõtmistulemustest selgub, et statistiliselt oluliselt erinesid vastendusskeemid kõigi alamklasside arvu ja andmemahu konfiguratsioonide korral (VV: $p = 0,000$; VS: $p = 0,000$; SV: $p = 0,000$; SS: $p = 0,000$).

Dunni võrdlustesti tulemustest järeldub, et kõigi alamklasside arvu ja andmemahu konfiguratsioonide seas ilmneb statistiliselt oluline erinevus SITI ja CLTI ning CLTI ja COTI vahel (Tabel 24). Suure alamklasside arvu ja suure andmemahuga konfiguratsiooni puhul ilmneb see lisaks SITI ja COTI vahel.

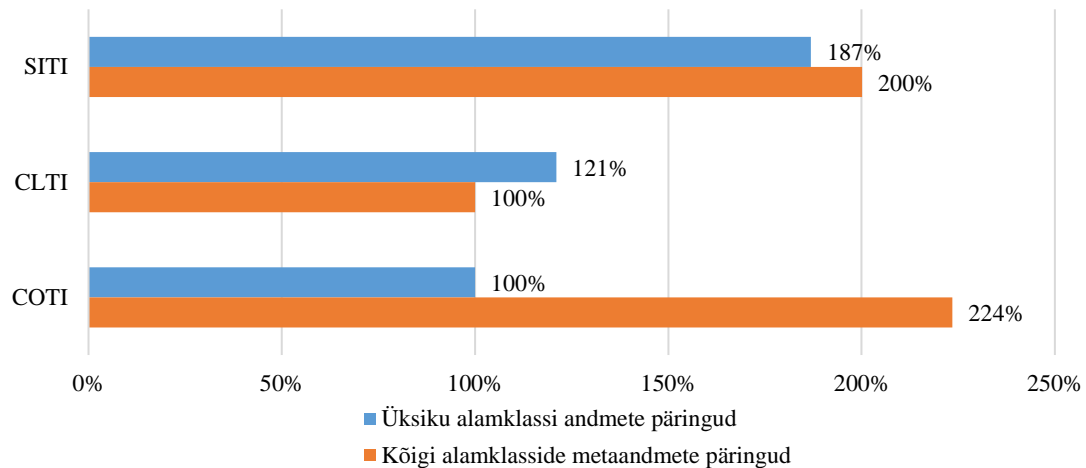
Tabel 24. Neljanda päringu vastendusskeemide statistiliselt olulised jõudluserinevused.

Q4	SITI – CLTI		SITI – COTI		CLTI – COTI	
	d_r	p	d_r	p	d_r	p
VV	112%	0,000	-		135%	0,000
VS	91%	0,000	-		99%	0,000
SV	120%	0,000	-		142%	0,000
SS	101%	0,002	64%	0,023	142%	0,000

Nagu kolmanda päringu puhul, on ka neljanda päringu tulemuste seast konfiguratsioonideülevalt halvima jõudlusega konkreetse tabeli pärilus, erinedes keskmiselt üksiktabeli ja klassitabeli pärilustest vastavalt 64% ja 129%. Samuti on selle jõudlus halvim just suure alamklasside arvu ja suure andmemahu korral, mis juhul erineb see klassitabeli päriluse omast 144%. Kõigi konfiguratsioonide lõikes on parima tulemusega klassitabeli pärilus.

4.7.7 Eksperimendi tulemuste kokkuvõte

Eksperimendi käigus läbi viidud analüüsides tulemustest selgub, et kolme päriluse vastendusskeemi seast on enamikul juhtudel parima jõudlusega klassitabeli pärilus. Vastendusskeemide jõudluserinevusi esitab joonis 24.



Joonis 24. Vastendusskeemide jõudluserinevused.

Üksiku alamklassi andmete lugemispäringute puhul osutub suurima jõudlusega vastendusskeemiks konkreetse tabeli pärilus, mille erinevus üksiktabeli pärilusega on 87%. Kuigi statistiliselt olulist erinevust klassitabeli ja konkreetse tabeli päriluste vahel ei ilmne, siis mõõtmisandmetest tuleneb siiski nendevaheline 21% erinevus.

Kõigi alamklasside metaandmete lugemispäringute puhul on parima jõudlusega vastendusskeemiks klassitabeli pärilus, mille erinevus võrreldes üksiktabeli pärilusega on 100% ning halvima jõudlusega vastendusskeemiks osutunud konkreetse tabeli pärilusega 124%. Viimase jõudlus väheneb eriti alamklasside arvu ja andmemahtude suurenedes.

5 Optimaalse mustri valimine analüütiliste hierarhiate meetodit kasutades

Käesolevas peatükis tehakse kolme päriluse vastendusskeemi seast edasiste arenduste tarvis optimaalne valik, kasutades selleks analüütiliste hierarhiate meetodit.

Analüütiliste hierarhiate meetod (AHP – *analytic hierarchy process*) ehk Saaty meetod on matemaatiku Thomas L. Saaty loodud raamistik, mis põhineb alternatiivide paarikaupa võrdlemisel ning võimaldab subjektiivsetele hinnangute alusel jõuda objektiivsete tulemusteni [26].

Meetodi eeliseks on võimalus anda objektidele absoluutsete hinnangute asemel suhtelisi hinnanguid ning tugineda nende andmisel näiteks kogemusele, intuitsioonile või õpitule. Meetodi rakendamise käigus luuakse otsustusmudel, mis sisaldab kõiki otsuse tegemise aluseks olevaid kriteeriume ning võimalikke alternatiive.

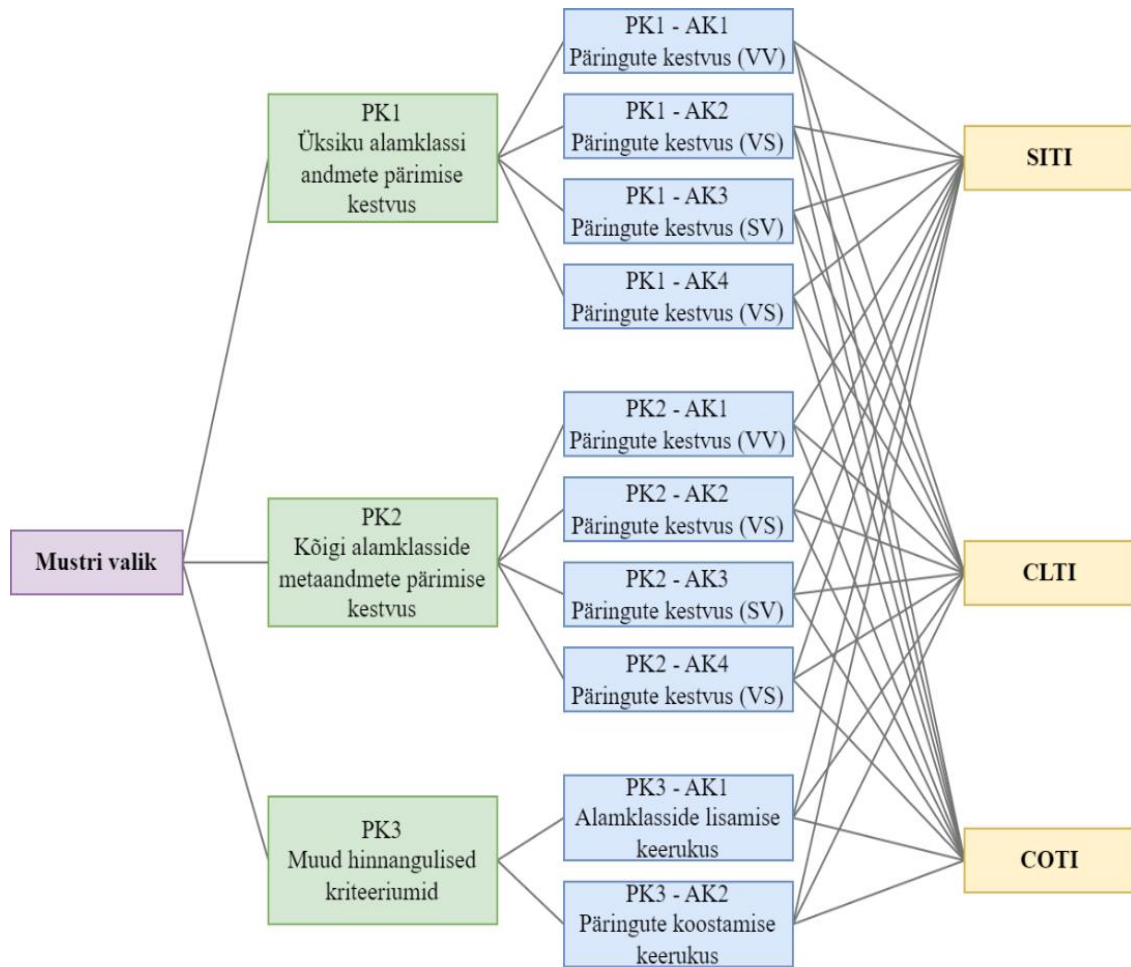
5.1 Otsustusmudeli koostamine

Otsustusmudeli koostamisel defineeritakse eesmärk ning kriteeriumid, mille põhjal erinevaid alternatiive ehk käsitletud vastendusskeeme võrdlema asutakse. Käesoleva töö puhul on eesmärgiks optimaalse päriluse vastendusskeemi mustri valimine edasiste arenduste tarvis ning selleks on töö autor defineerinud kolm olulisimat põhikriteeriumit ja kümme alamkriteeriumit. Kriteeriumite valik põhineb autori varasematel kogemustel pärilushierarhiatega töötamisel. Eelmises peatükis käsitletud salvestusruumi kriteeriumiks ei valita, kuna see erineb vastendusskeemide lõikes keskmiselt vaid ligi 10%. Alternatiivideks on töö raames analüüsitud ja eksperimendi käigus kasutatud päriluse vastendusskeemide disainimustrid. Analüütiliste hierarhiate meetodi kriteeriumite otsustusmudelit esitab tabel 25.

Tabel 25. Otsustusmodeli põhi- ja alamkriteeriumid.

Põhikriteerium	Alamkriteerium
Üksiku alamklassi andmete pärimise kestvus	Päringute kestvus väikese alamklasside arvu ja väikese andmemahu juures (VV)
	Päringute kestvus väikese alamklasside arvu ja suure andmemahu juures (VS)
	Päringute kestvus suure alamklasside arvu ja väikese andmemahu juures (SV)
	Päringute kestvus suure alamklasside arvu ja suure andmemahu juures (SS)
Kõigi alamklasside metaandmete pärimise kestvus	Päringute kestvus väikese alamklasside arvu ja väikese andmemahu juures (VV)
	Päringute kestvus väikese alamklasside arvu ja suure andmemahu juures (VS)
	Päringute kestvus suure alamklasside arvu ja väikese andmemahu juures (SV)
	Päringute kestvus suure alamklasside arvu ja suure andmemahu juures (SS)
Muud hinnangulised kriteeriumid	Alamklasside lisamise keerukus
	Päringute koostamise keerukus

Kolme põhikriteeriumi ja kümne alamkriteeriumiga otsustusmodeli struktuuri esitab joonis 25. Saaty meetodi rakendamiseks kasutakse Web-HIPRE tarkvara.



Joonis 25. Päriluse vastendusskeemi valiku otsustusmudel.

Kriteeriumite ja alternatiivide omavaheliseks võrdlemiseks kasutatakse Saaty fundamentaalskaalat, mis omab väärtusi ühest üheksani. Paaritud arvud tähistavad kasvavas järjekorras samaväärset, mõõdukat, olulist, väga tugevat ja ekstreemset paremust või tähtsust ning paarisarvud nende vahelisi väärtuseid. Mõõtmistulemuste põhjal antavate hinnangute tarvis kasutatakse ka täpsemaid murdarve.

Antavate hinnangute järjepidevust esitab Web-HIPRE tarkvara arvutatav kooskõla määr (ingl *consistency measure*) CM, mis kooskõlaliste hinnangute puhul võiks, kuid ei pea jääma alla 0,1 [27].

5.1.1 Põhikriteeriumite olulisus

Olulisim tingimus vastendusskeemi valikul on autori hinnangul päringute kestvus üksikute alamklasside andmete pärimisel, kuna seda liiki päringuid sooritatakse suure tõenäosusega kõige rohkem. Teisel kohal on ülemklassi andmetega seotud päringute kestvus ning viimasel kohal muud kriteeriumid, kuna nendega seotud tegevusi tehakse oluliselt harvemini. Tabel 26 esitab põhikriteeriumite omavahelise võrdluse tulemused. Võrdluse kooskõla määr on 0,079.

Tabel 26. Põhikriteeriumite võrdlus.

	Üksiku alamklassi andmete pärimise kestvus	Kõigi alamklasside metaandmete pärimise kestvus	Muud hinnangulised kriteeriumid	Kaal
Üksiku alamklassi andmete pärimise kestvus	1,00	3,00	4,00	0,625
Kõigi alamklasside metaandmete pärimise kestvus	0,33	1,00	2,00	0,238
Muud hinnangulised kriteeriumid	0,25	0,50	1,00	0,136

5.1.2 Alamkriteeriumite olulisus

Üksiku alamklassi andmete pärimise kestvuse ja kõigi alamklasside metaandmete pärimise kestvuse põhikriteeriumite alamkriteeriume käsitletakse siinkohal koos.

Kõige olulisem alamkriteerium on väikese alamklasside arvu ja väikese andmemahuga konfiguratsiooni (VV) päringute kestvus, kuna autori hinnangul on sellise konfiguratsiooni esinemine kõige tõenäolisem. Olulisuselt teisel kohal on päringute kestvus väikese alamklasside arvu ja suure andmemahu juures, kuna autori hinnangul on tõenäolisem, et süsteemis esineb pigem andmemahu kui alamklasside arvu suurenemine (Tabel 27). Võrdluse kooskõla määr on 0,087.

Tabel 27. Päringute kestvuse kriteeriumite alamkriteeriumite võrdlus.

	Päringute kestvus VV konf. juures	Päringute kestvus VS konf. juures	Päringute kestvus SV konf. juures	Päringute kestvus SS konf. juures	Kaal
Päringute kestvus VV konf. juures	1,00	2,00	6,00	4,00	0,504
Päringute kestvus VS konf. juures	0,50	1,00	4,00	3,00	0,301
Päringute kestvus SV konf. juures	0,17	0,25	1,00	0,5	0,073
Päringute kestvus SS konf. juures	0,25	0,33	2,00	1,00	0,123

Muude hinnanguliste alamkriteeriumite võrdlust esitab tabel 28. Nende seast mõnevõrra olulisem on päringute koostamise keerukus, kuna seda tegevust tuleb tõenäoliselt teha tihedamini kui uute alamklasside lisamist. Võrdluse kooskõla määr on 0,000.

Tabel 28. Hinnanguliste kriteeriumite alamkriteeriumite võrdlus.

	Alamklasside lisamise keerukus	Päringute koostamise keerukus	Kaal
Alamklasside lisamise keerukus	1,00	0,50	0,333
Päringute koostamise keerukus	2,00	1,00	0,667

5.1.3 Alternatiivide võrdlus

Alternatiivide võrdluses võrreldakse kolme pärilushierarhia vastendusskeeme kõigi alamkriteeriumite suhtes. Päringute kestvusi puudutavate hinnangute kaalude aluseks on eelmises peatükis läbiviidud põhikriteeriumiga seotud päringute jõudlustestide tulemused, mille põhjal on arvatud suhtelised headused.

Üksiku alamklassi andmete pärimise kestvuste suhteliste headuste leidmiseks viiakse esimese (Q1) ja teise (Q2) päringu tulemused samale skaalale ning arvutatakse nende keskmine (Lisa 4, Tabel 45).

Alternatiivide hinnangute leidmiseks arvutatakse esmalt iga alternatiivi päringukestvuste teisendatud keskmiste suhe. Kuna päringukestvuste suhted ei vasta otse nende tegelikele paremuste hinnangutele, siis teisendatakse need, kasutades valemit (1).

$$5 \ln x + 1 \tag{1}$$

Kui näiteks kolmekordse päringukestvuse erinevus suhtega 3,00 vastab teisendamata kujul mõõduka paremuse hinnangule, siis valemit kasutades teisendub see väärtuseks 6,49, mis vastab ligilähedaselt väga tugeva paremuse hinnangule. Mõnede päringukestvuste suhete korral ületab valemi tulemusel saadud hinnangu väärtus 9,00 piiri. Sellistel juhtudel kasutatakse hinnangu väärtusena ülempiiri 9,00, mille mõju hinnangute kooskõlamäärale on marginaalne.

Üksiku alamklassi andmete päringukestvuste hinnanguid esitab tabel 29.

Tabel 29. Üksiku alamklassi andmete päringukestvuste hinnangud.

VV		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,16	0,20	0,081
	CLTI	6,24	1,00	0,43	0,350
	COTI	4,89	2,35	1,00	0,570
	CM: 0,185				
VS		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,18	0,17	0,081
	CLTI	5,43	1,00	0,73	0,408
	COTI	5,79	1,36	1,00	0,511
	CM: 0,041				
SV		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,11	0,10	0,052
	CLTI	8,75	1,00	0,55	0,379
	COTI	9,00	1,81	1,00	0,568
	CM: 0,081				
SS		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,13	0,12	0,058
	CLTI	7,49	1,00	0,54	0,368
	COTI	8,34	1,85	1,00	0,574
	CM: 0,075				

Ülemklassi andmetega seotud päringute kestvuste suhteliste headuste leidmiseks viiakse kolmanda (Q3) ja neljanda (Q4) päringu tulemused samale skaalale ning arvutatakse nende keskmine (Lisa 4, Tabel 46).

Alternatiivide hinnangute leidmiseks arvutatakse iga alternatiivi päringute teisendatud keskmiste suhe ning teisendatakse need, kasutades valemit (1). Ülemklassi andmete päringukestvuste hinnanguid esitab tabel 30.

Tabel 30. Ülemklassi andmete päringukestvuste hinnangud.

VV		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,15	2,72	0,154
	CLTI	6,47	1,00	8,19	0,773
	COTI	0,37	0,12	1,00	0,073
CM: 0,114					
VS		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,18	1,42	0,143
	CLTI	5,71	1,00	6,13	0,746
	COTI	0,70	0,16	1,00	0,111
CM: 0,046					
SV		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,14	2,64	0,128
	CLTI	7,07	1,00	8,71	0,766
	COTI	0,38	0,11	1,00	0,106
CM: 0,079					
SS		SITI	CLTI	COTI	Kaal
	SITI	1,00	0,16	4,20	0,174
	CLTI	6,30	1,00	9,00	0,766
	COTI	0,24	0,11	1,00	0,059
CM: 0,155					

Lausete keerukust hinnatakse koodiridade arvu meetodil [28] ning selleks arvestatakse kokku väikese ja suure alamklasside arvuga konfiguratsioonide andmebaasiskeemi loomise, andmete lisamise ja pärimise füüsiliste koodiridade keskmine arv. Lausete keerukust koodiridade arvudena esitab tabel 31.

Tabel 31. Lausete keskmised keerukused.

Vastenduskeem	Andmebaasiskeemi loomine	Andmete lisamine	Andmete pärimine	Kokku
SITI	78	14	32	124
CLTI	194	17	34	245
COTI	200	13	74	287

Päringute koostamise keerukuse alamkriteeriumi hinnangute leidmiseks arvutatakse iga alternatiivi koodiridade keskmiste suhe ning teisendatakse see, kasutades samuti valemit (1). Eelistatuimaks peetakse üksiktabeli päriluse vastendusskeemi, mis ei nõua ühendpäringuid teiste tabelitega ning on väikseima koodiridade arvuga. Vähim eelistatumaks peetakse konkreetse tabeli päriluse vastendusskeemi, mille kasutamisel tuleb ülemklassi andmete pärimisel teha keerukamaid UNION klausliga päringuid. Alamklasside lisamise keerukuse hindamisel peetakse eelistatuimaks üksiktabeli päriluse vastendusskeemi, kuna selle kasutamisel ei ole tarvis muuta andmebaasi struktuuri. Konkreetse tabeli päriluse vastendusskeemi puhul on tegemist vähim eelistatud alternatiiviga, kuna tabelite loomisel tuleb teha lisatööd identiteediveeru väärtuste unikaalsuse jõustamiseks. Tabel 32 esitab muude hinnanguliste kriteeriumite suhtelisi hinnanguid.

Tabel 32. Hinnanguliste kriteeriumite hinnangud.

Alamklasside lisamise keerukus		SITI	CLTI	COTI	Kaal
	SITI	1,00	5,00	6,00	0,726
	CLTI	0,20	1,00	2,00	0,172
	COTI	0,17	0,50	1,00	0,102
	CM: 0,085				
Päringute koostamise keerukus		SITI	CLTI	COTI	Kaal
	SITI	1,00	4,40	5,20	0,700
	CLTI	0,23	1,00	1,79	0,183
	COTI	0,19	0,56	1,00	0,117
	CM: 0,073				

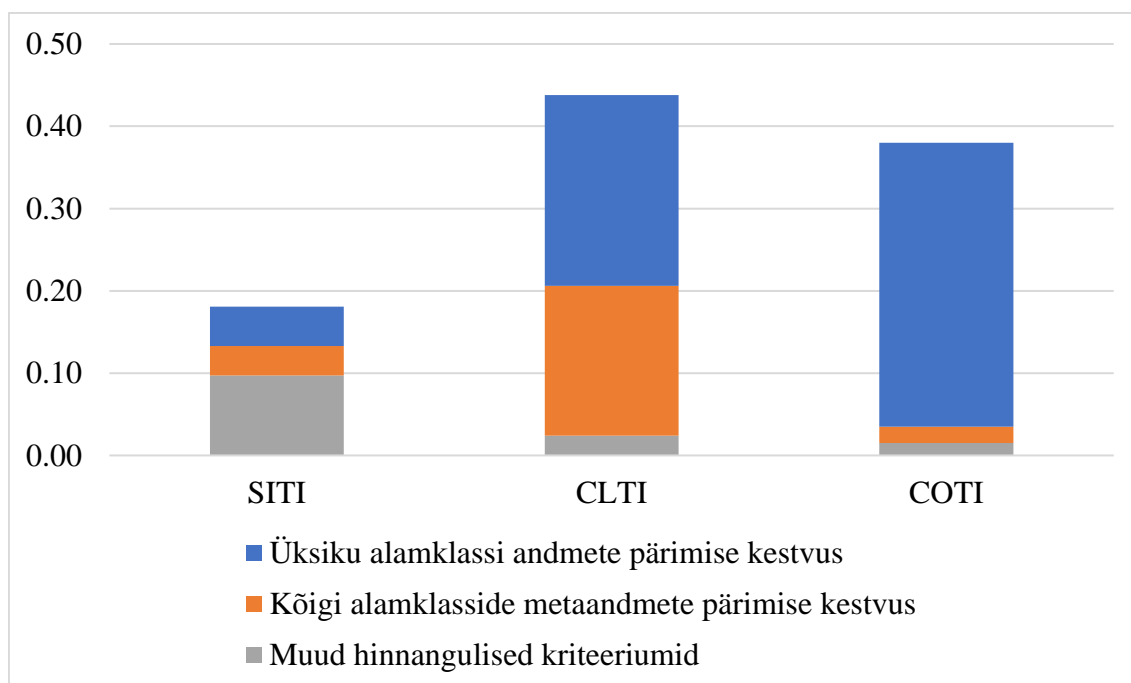
5.2 Tulemuste analüüs ja järeldused

Alternatiivide suhtelisi headuseid optimaalse vastenduskeemi valikuks esitab tabel 33.

Tabel 33. Analüütiliste hierarhiate meetodi rakendamise lõpptulemus.

Põhikriteerium	SITI	CLTI	COTI
Üksiku alamklassi andmete pärimise kestvus	0,048	0,232	0,345
Kõigi alamklasside metaandmete pärimise kestvus	0,036	0,182	0,020
Muud hinnangulised kriteeriumid	0,097	0,024	0,015
Kokku	0,180	0,439	0,381

Otsustusmudeli tulemuste põhjal osutub optimaalseks päriluse vastenduskeemiks klassitabeli pärilus. Üksiku alamklassi andmete pärimise kestvuse põhikriteeriumi osas on konkreetse tabeli päriluse vastenduskeemi jõudlus klassitabeli omast küll parem, kuid viimase tõstab üldkokkuvõttes esimesele kohale oluliselt parem jõudlus kõigi alamklassi metaandmete pärimisel. Üksiktabeli päriluse vastenduskeem osutub parimaks vaid muude hinnanguliste kriteeriumite põhikriteeriumi ulatuses. Graafiliselt esitab tulemusi joonis 26.



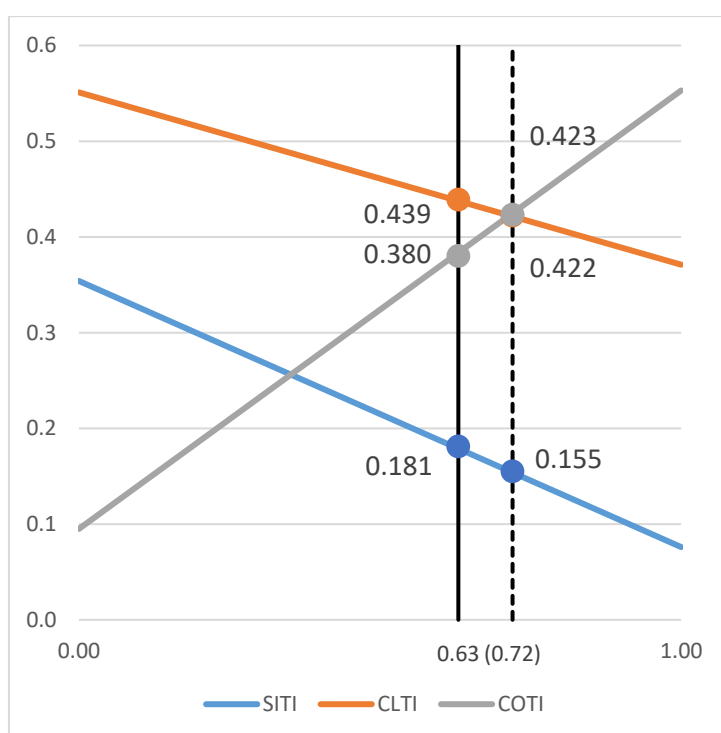
Joonis 26. Analüütiliste hierarhiate meetodi rakendamise lõpptulemus.

5.3 Tundlikkuse analüüs

Tundlikkuse analüüs viiakse läbi kõigi põhikriteeriumite suhtes ning see näitab, kas otsustusmudelil kriteeriumite osakaalude muutmisel muutub lõpptulemuses ka alternatiivide järjestus.

5.3.1 Üksiku alamklassi andmete pärimise kestvuse tundlikkuse analüüs

Esimese põhikriteeriumi tundlikkust analüüsidest järeldub, et lõpptulemuse muutumiseks tuleks kriteeriumi osakaalu tõsta tasemelt 0,63 tasemele 0,72. Sellisel juhul osutuks optimaalseks vastendusskeemiks konkreetse tabeli pärilus (Joonis 27).

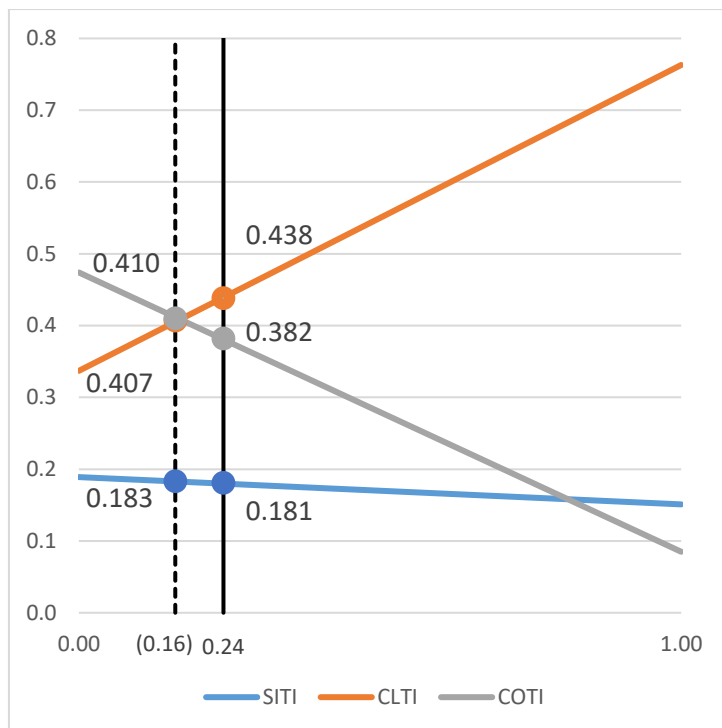


Joonis 27. Üksiku alamklassi andmete pärimise kestvuse põhikriteeriumi tundlikkuse analüüs.

Kui enne kriteeriumite osakaalude muutmist on kriteeriumi kaalu suhe teistesse kaaludesse $0,63 / (1 - 0,63) = 1,70$, siis pärast muutmist oleks see $0,72 / (1 - 0,72) = 2,57$. Sellest tulenevalt peaks põhikriteeriumi olulisust teiste põhikriteeriumite suhtes suurendama keskmiselt $2,57 / 1,70 = 1,51$ korda. Osatähtsuse selline muutmine oleks põhjendatud juhul, kui enamik süsteemis tehtavad päringud oleksid seotud üksikute alamklasside andmetega ning ülemklassi metaandmete päringuid tehtaks oluliselt vähem.

5.3.2 Kõigi alamklasside metaandmete pärimise kestvuse tundlikkuse analüüs

Teise põhikriteeriumi tundlikkust analüüsid järeldeb, et kriteeriumi osakaalu tuleks lõpptulemuse muutmiseks langetada tasemelt 0,24 tasemele 0,16, mis juhul osutuks optimaalseks vastenduskeemiks konkreetse tabeli pärilus (Joonis 28).

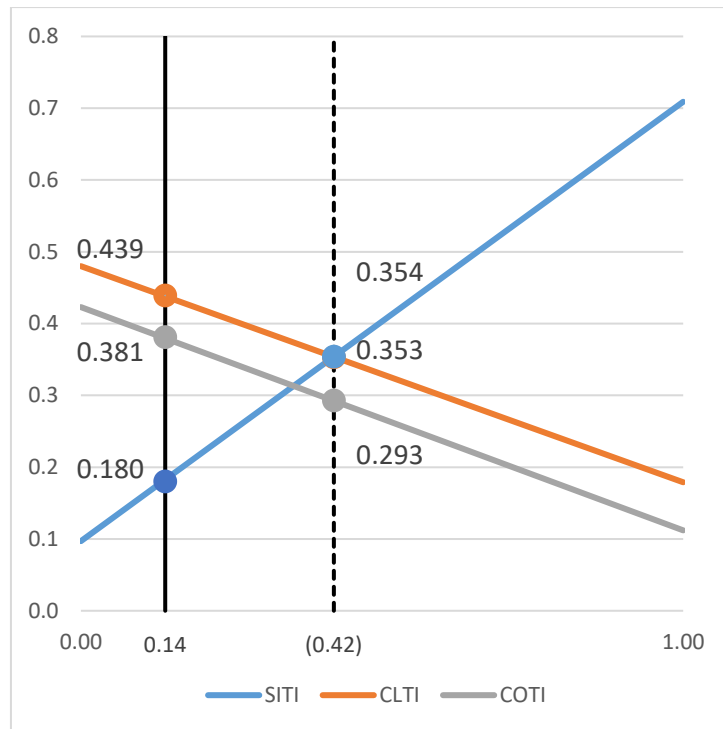


Joonis 28. Kõigi alamklasside metaandmete pärimise kestvuse põhikriteeriumi tundlikkuse analüüs.

Enne kriteeriumite osakaalude muutmist on nimetatud kriteeriumi kaalu suhe teistesse kaaludesse $0,24 / (1 - 0,24) = 0,32$ ning pärast muudatust $0,16 / (1 - 0,16) = 0,19$. Sellest tulenevalt peaks põhikriteeriumi olulisust teiste põhikriteeriumite suhtes vähendada keskmiselt $1 / (0,19 / 0,32) = 1,66$ korda. Muudatus oleks põhjendatud samadel juhtudel, mil üksiku alamklassi andmete pärimise kestvuse põhikriteeriumi osakaalu suurendamine.

5.3.3 Muude hinnanguliste kriteeriumite tundlikkuse analüüs

Muude hinnanguliste kriteeriumite põhikriteeriumi tundlikkuse analüüsist järeldub, et lõpptulemuse muutmiseks tuleks kriteeriumi osakaalu tõsta tasemelt 0,14 tasemele 0,42. Sellisel juhul osutuks optimaalseks vastenduskeemiks üksiktabeli pärilus (Joonis 29).



Joonis 29. Muude hinnanguliste kriteeriumite põhikriteeriumi tundlikkuse analüüs.

Enne kriteeriumite osakaalude muutmist on nimetatud kriteeriumi kaalu suhe teistesse kaaludesse $0,14 / (1 - 0,14) = 0,16$ ning pärast muudatust $0,42 / (1 - 0,42) = 0,72$. Sellest tulenevalt peaks põhikriteeriumi olulisust teiste põhikriteeriumite suhtes suurendama keskmiselt $0,72 / 0,16 = 4,49$ korda. Tegemist on väga suure muudatusega, mis oleks põhjendatud vaid juhul, kui alamklasse lisatakse süsteemi väga tihti või on tingimata oluline päringute koostamise vähene keerukus.

6 Kokkuvõte

Valdav osa tänapäevaseid infosüsteeme kasutavad andmete talletamiseks ja käsitlemiseks andmebaase.

Käesolevas bakalaureusetöös käsitleti päriluse vastendamiseks nimetatavat probleemi ja selle lahendamiseks ette nähtud päriluse vastendusskeemide disainimustreid.

Lõputöö eesmärgiks oli vastendusskeemide info süstematiseerimine ja nende analüüsimine ning disainimustrite kataloogi loomine koos optimaalse vastendusskeemi valikuga analüütiliste hierarhiate meetodil.

Töö tulemusena need eesmärgid saavutati ning valmis kolme vastendusskeemi põhjalik ja arvandmetele tugineva analüüsiga disainimustrite kataloog. Kataloogis on esitatud mustrite tööpõhimõtted, näited ning nende eelised ja puudused. Töö käigus viidi eksperimendi raames läbi jõudlusvõrdlused ning selle tulemusi arvestati optimaalse valiku tegemisel.

Ekspirimendi dispersioonanalüüsi tulemusena osutus üldkokkuvõttes parima jõudlusega vastendusskeemiks klassitabeli pärilus. Üksiku alamklassi andmete lugemisega seotud päringute puhul osutus parima jõudlusega vastendusskeemiks konkreetse tabeli pärilus, mille jõudlus erines üksiktabeli ja klassitabeli päriluste omadest vastavalt 87% ja 21%. Kõigi alamklasside metaandmete lugemisega seotud päringute puhul oli parim klassitabeli päriluse vastendusskeemi jõudlus, erinedes üksiktabeli ja konkreetse tabeli päriluse vastendusskeemide omadest vastavalt 100% ja 124%.

Ka jõudlusvõrdlustel tugineva analüütiliste hierarhiate meetodi tulemuste põhjal on optimaalseks päriluse vastendusskeemiks enamik juhtudel klassitabeli pärilus. Alternatiivide kasutamine osutus põhjendatuks mõningate eritingimuste esinemisel.

Lõputöö tulemused on abiks tarkvaraarhitektidele parima disainilahenduse leidmisel ning töö raames loodud analüütiliste hierarhiate otsustusmudelit on võimalik kohandada

vastavalt erinevate tarkvarade poolt kasutatavate andmete eripäradele ja kasutusjuhtudele.

Käesolevas töös käsitleti vaid kahetasandilisi pärilushierarhiaid, mistõttu on üheks võimalikuks töö edasiarenduseks enama kui kahetasandiliste pärilushierarhiate vastendusskeemide analüüsimine ja rakendamine. Samuti on võimalik viia läbi põhjalikum analüüs päringute koostamise keerukuse osas ning kaasata jõudlusvõrdlustesse lisaks lugemispäringutele ka muid operatsioone.

Kasutatud kirjandus

- [1] Freeform Dynamics, „Modernizing database technology stacks,“ Freeform Dynamics, Christchurch, 2020.
- [2] R. Craig, „Bridging the Object-Relational Divide,“ *ACM Queue*, pp. 18-28, 1. mai 2008.
- [3] S. Balzer, *Contracted Persistent Object Programming*, Zürich: Swiss Federal Institute of Technology Zürich, 2005.
- [4] H. Garcia-Molina, J. Ullman ja J. Widom, *Database Systems: The Complete Book*, New Jersey: Prentice Hall, 2008.
- [5] S. M. Mikk, „Graafide esitamine SQL-andmebaasides,“ Tallinna Tehnikaülikool, Tallinn, 2017.
- [6] P. Raspel, „Andmebaasisüsteemide alused: Loengumaterjalid,“ 15. detsember 2011. [Võrgumaterjal]. Available: <https://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/>. [Kasutatud 7. novembril 2021].
- [7] R. K. Soni, A. Ganeshan ja R. RV, *Spring: Developing Java Applications for the Enterprise*, Birmingham: Packt Publishing, 2017.
- [8] S. W. Ambler, *Building Object Applications that Work*, Cambridge: Cambridge University Press, 1998.
- [9] S. Bhushan, „Object/Relational Impedance Mismatch,“ 21. august 2021. [Võrgumaterjal]. Available: <https://medium.com/booleanbhushan/object-relational-impedance-mismatch-28fc2440dee4>. [Kasutatud 22. veebruaril 2022].
- [10] C. J. Ireland, „Object-Relational Impedance Mismatch: A Framework Based Approach,“ ProQuest, Milton Keynes, 2011.
- [11] M. Fowler, *Patterns of Enterprise Application Architecture*, Boston: Addison-Wesley Professional, 2002.
- [12] M. Machisa, „Object Relational Mapping for Enterprise Application Architecture,“ ePublications at Regis University, Denver, 2005.
- [13] T. Janssen, „OOP Concept: Inheritance,“ 14. detsember 2017. [Võrgumaterjal]. Available: <https://stackify.com/oop-concept-inheritance/>. [Kasutatud 20. veebruaril 2022].
- [14] P. J. Deitel ja H. M. Deitel, *Java How to Program, Early Objects, 10th Edition*, Harlow: Pearson, 2015.
- [15] J. Pöial, „Objektid, klassid, pärimine,“ [Võrgumaterjal]. Available: <https://enos.itcollege.ee/~jpoial/java/i200loeng3.html>. [Kasutatud 20. veebruaril 2022].
- [16] O. Tuna, „Multiple Inheritance in C++ and the Diamond Problem,“ 21. oktoober 2017. [Võrgumaterjal]. Available: <https://medium.com/free-code-camp/multiple-inheritance-in-c-and-the-diamond-problem-7c12a9ddbdec>. [Kasutatud 20. veebruaril 2022].

- [17] N. Arner, „PostgreSQL as an ORDBMS: What does that mean?“, 2021. [Võrgumaterjal]. Available: <https://arctype.com/blog/postgres-ordbms-explainer/>. [Kasutatud 21. veebruaril 2022].
- [18] M. Manavi, „Inheritance with EF Code First: Part 1 - Table per Hierarchy (TPH)“, 24. detsember 2010. [Võrgumaterjal]. Available: <https://weblogs.asp.net/manavi/inheritance-mapping-strategies-with-entity-framework-code-first-ctp5-part-1-table-per-hierarchy-tph>. [Kasutatud 16. veebruaril 2022].
- [19] B. Karwin, *SQL Antipatterns: Avoiding the Pitfalls of Database Programming*, Dallas: The Pragmatic Bookshelf, 2010.
- [20] M. Manavi, „Inheritance with EF Code First: Part 2 – Table per Type (TPT)“, 28. detsember 2010. [Võrgumaterjal]. Available: <https://weblogs.asp.net/manavi/inheritance-mapping-strategies-with-entity-framework-code-first-ctp5-part-2-table-per-type-tp>. [Kasutatud 16. veebruaril 2022].
- [21] M. Manavi, „Inheritance with EF Code First: Part 3 – Table per Concrete Type (TPC)“, 3. jaanuar 2011. [Võrgumaterjal]. Available: <https://weblogs.asp.net/manavi/inheritance-mapping-strategies-with-entity-framework-code-first-ctp5-part-3-table-per-concrete-type-tpc-and-choosing-strategy-guidelines>. [Kasutatud 16. veebruaril 2022].
- [22] Microsoft, „Modeling for Performance: Inheritance Mapping“, 11. märts 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/ef/core/performance/modeling-for-performance#inheritance-mapping>. [Kasutatud 25. veebruaril 2022].
- [23] D. R. Benigni, „A guide to performance evaluation of database systems“, *NBS Special Publications*, p. 54, detsember 1984.
- [24] K. Niglas, „Dispersioonanalüüs“, november 2013. [Võrgumaterjal]. Available: <http://www.cs.tlu.ee/~katrin/wp/wp-content/uploads/2013/11/dispersioon.pdf>. [Kasutatud 31. märtsil 2022].
- [25] Laerd Statistics, „Pearson Product-Moment Correlation“, Laerd Statistics, [Võrgumaterjal]. Available: <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>. [Kasutatud 13. märtsil 2022].
- [26] L. Võhandu, *Subjektiiivsetest hinnangutest objektiivsete tulemusteni*, Tallinn: Tallinna Tehnikaülikooli Kirjastus, 1998.
- [27] J. Mustajoki ja R. P. Hämmäläinen, „Web-HIPRE - Global decision support by value tree and AHP analysis“, *INFOR: Information Systems and Operational Research*, kd. 38, nr 3, pp. 208-220, 1999.
- [28] B. W. Boehm, S. Deeds-Rubin, V. Nguyen ja T. Tan, „A SLOC Counting Standard“, University of Southern California, Los Angeles, 2007.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Chris Rak

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Klasside ja relatsiooniliste tabelite päriluse vastendusskeemide analüüs“, mille juhendaja on Jaanus Pöial
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Andmebaasiskeemide salvestusruumid

Andmebaasiskeemide salvestusruumi koos testandmetega esitavad megabaitides üksiktabeli päriluse vastenduskeemi kohta tabel 34.

Tabel 34. Üksiktabeli päriluse vastenduskeemi andmebaasiskeemide salvestusruumid.

		Pärilushierarhia klassieksemplaride arv tuhandetes			
		800	1 600	3 200	6 400
Alamklasside arv	SITI				
	2	623.5	1246.6	2492.3	4985.2
	4	644.9	1289.1	2578.2	5230.2
	8	688.8	1376.8	2753.5	5506.1
	16	776.8	1553.7	3107.0	6213.6

Klassitabeli päriluse vastenduskeemi kohta esitab andmebaasiskeemide salvestusruumi koos testandmetega tabel 35.

Tabel 35. Klassitabeli päriluse vastenduskeemi andmebaasiskeemide salvestusruumid.

		Pärilushierarhia klassieksemplaride arv tuhandetes			
		800	1 600	3 200	6 400
Alamklasside arv	CLTI				
	2	624.7	1248.5	2495.6	4990.1
	4	625.2	1248.8	2496.4	4990.8
	8	626.6	1250.1	2497.1	4992.4
	16	629.1	1253.0	2499.5	4993.9

Andmebaasiskeemide salvestusruumi koos testandmetega esitavad megabaitides üksiktabeli päriluse vastendusskeemi kohta tabel 36.

Tabel 36. Konkreetse tabeli päriluse vastendusskeemi andmebaasiskeemide salvestusruumid.

		Pärilushierarhia klassieksemplaride arv tuhandetes			
		800	1 600	3 200	6 400
Alamklasside arv	2	611.0	1221.3	2441.7	4883.2
	4	611.4	1221.9	2442.6	4883.4
	8	613.2	1222.8	2443.4	4885.2
	16	616.1	1226.1	2445.7	4886.5

Lisa 3 – Eksperimendi dispersioonanalüüsi meetodi valik

Esimese päringu (Q1) dispersioonanalüüsi meetodi valikud esitab tabel 37.

Tabel 37. Dispersioonanalüüsi meetodi valik esimese päringu (Q1) tulemuste analüüsiks.

Q1	Üldkogumite homoskedastiivsus						Vastavus normaaljaotusele		Analüüsi meetodi valik
	SITI		CLTI		COTI		p	Otsus	
	p	Otsus	p	Otsus	p	Otsus			
VV	0,002	Ei	0,008	Ei	0,004	Ei	0,019	Ei	Kruskal–Wallis
VS	0,001	Ei	0,003	Ei	0,003	Ei	0,000	Ei	Kruskal–Wallis
SV	0,004	Ei	0,076	Jah	0,008	Ei	0,000	Ei	Kruskal–Wallis
SS	0,002	Ei	0,021	Ei	0,003	Ei	0,002	Ei	Kruskal–Wallis

Esimese päringu (Q1) korrelatsioonanalüüsi tulemusi esitab tabel 38.

Tabel 38. Esimese päringu (Q1) korrelatsioonanalüüsi tulemused.

Q1	SITI		CLTI		COTI	
	r	Sõltuvus	r	Sõltuvus	r	Sõltuvus
SC02	0,999	Tugev	1,000	Tugev	1,000	Tugev
SC04	1,000	Tugev	0,999	Tugev	1,000	Tugev
SC08	0,997	Tugev	0,960	Tugev	1,000	Tugev
SC16	0,996	Tugev	1,000	Tugev	1,000	Tugev

Teise päringu (Q2) dispersioonanalüüsi meetodi valikud esitab tabel 39.

Tabel 39. Dispersioonanalüüsi meetodi valik teise päringu (Q2) tulemuste analüüsiks.

Q2	Üldkogumite homoskedastiivsus						Vastavus normaaljaotusele		Analüüsi meetodi valik
	SITI		CLTI		COTI		<i>p</i>	Otsus	
	<i>p</i>	Otsus	<i>p</i>	Otsus	<i>p</i>	Otsus			
VV	0,000	Ei	0,060	Jah	0,000	Ei	0,000	Ei	Kruskal–Wallis
VS	0,000	Ei	0,150	Jah	0,000	Ei	0,000	Ei	Kruskal–Wallis
SV	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis
SS	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis

Teise päringu (Q2) korrelatsioonanalüüsi tulemusi esitab tabel 40.

Tabel 40. Teise päringu (Q2) korrelatsioonanalüüsi tulemused.

Q2	SITI		CLTI		COTI	
	<i>r</i>	Sõltuvus	<i>r</i>	Sõltuvus	<i>r</i>	Sõltuvus
SC02	0,986	Tugev	1,000	Tugev	0,969	Tugev
SC04	0,984	Tugev	0,999	Tugev	0,844	Tugev
SC08	0,979	Tugev	1,000	Tugev	1,000	Tugev
SC16	0,972	Tugev	1,000	Tugev	0,995	Tugev

Kolmanda päringu (Q3) dispersioonanalüüsi meetodi valikud esitab tabel 41.

Tabel 41. Dispersioonanalüüsi meetodi valik kolmanda päringu (Q3) tulemuste analüüsiks.

Q3	Üldkogumite homoskedastiivsus						Vastavus normaaljaotusele		Analüüsi meetodi valik
	SITI		CLTI		COTI		p	Otsus	
	p	Otsus	p	Otsus	p	Otsus			
VV	0,000	Ei	0,001	Ei	0,000	Ei	0,001	Ei	Kruskal–Wallis
VS	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis
SV	0,000	Ei	0,001	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis
SS	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis

Kolmanda päringu (Q3) korrelatsioonanalüüsi tulemusi esitab tabel 42.

Tabel 42. Kolmanda päringu (Q3) korrelatsioonanalüüsi tulemused.

Q3	SITI		CLTI		COTI	
	r	Sõltuvus	r	Sõltuvus	r	Sõltuvus
SC02	0,980	Tugev	0,999	Tugev	0,917	Tugev
SC04	0,984	Tugev	1,000	Tugev	0,696	Tugev
SC08	0,984	Tugev	0,999	Tugev	1,000	Tugev
SC16	0,985	Tugev	0,999	Tugev	0,999	Tugev

Neljanda päringu (Q4) dispersioonanalüüsi meetodi valikud esitab tabel 43.

Tabel 43. Dispersioonanalüüsi meetodi valik neljanda päringu (Q4) tulemuste analüüsiks.

Q4	Üldkogumite homoskedastiivsus						Vastavus normaaljaotusele		Analüüsi meetodi valik
	SITI		CLTI		COTI		p	Otsus	
	p	Otsus	p	Otsus	p	Otsus			
VV	0,000	Ei	0,000	Ei	0,000	Ei	0,001	Ei	Kruskal–Wallis
VS	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis
SV	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis
SS	0,000	Ei	0,000	Ei	0,000	Ei	0,000	Ei	Kruskal–Wallis

Neljanda päringu (Q4) korrelatsioonanalüüsi tulemusi esitab tabel 44.

Tabel 44. Neljanda päringu (Q4) korrelatsioonanalüüsi tulemused.

Q4	SITI		CLTI		COTI	
	r	Sõltuvus	r	Sõltuvus	r	Sõltuvus
SC02	0,985	Tugev	1,000	Tugev	0,906	Tugev
SC04	0,984	Tugev	0,999	Tugev	0,679	Tugev
SC08	0,984	Tugev	1,000	Tugev	1,000	Tugev
SC16	0,983	Tugev	1,000	Tugev	0,999	Tugev

Lisa 4 – Eksperimendi mõõtmistulemuste Saaty teisendused

Üksiku alamklassi päringute teisendatud keskmise kestvuse arvutusi esitab tabel 45.

Tabel 45. Üksiku alamklassi andmete päringute teisendatud mõõtmistulemuste keskmised.

Alternatiiv	Konf.	Q1	Q2	Q2 teisen.	Keskmine
SITI	VV	2420,47	103,14	3545,12	2982,80
	VS	10385,10	302,15	10385,10	10385,10
	SV	469,87	103,75	3565,96	2017,91
	SS	2345,45	274,76	9443,90	5894,68
CLTI	VV	1080,31	44,70	1010,42	1045,37
	VS	4283,90	189,50	4283,90	4283,90
	SV	273,94	25,76	582,39	428,17
	SS	1120,07	92,87	2099,50	1609,78
COTI	VV	1018,35	62,83	1719,54	1368,94
	VS	3982,83	145,52	3982,83	3982,83
	SV	260,84	17,08	467,56	364,20
	SS	1013,32	62,19	1702,09	1357,70

Ülemklassi päringute teisendatud keskmise kestvuse arvutusi esitab tabel 46.

Tabel 46. Ülemklassi andmete päringute teisendatud mõõtmistulemuste keskmised.

Alternatiiv	Konf.	Q3	Q4	Q4 teisen.	Keskmine
SITI	VV	109,04	97,68	103,66	106,35
	VS	122,25	250,88	266,23	265,48
	SV	264,72	109,96	116,69	119,47
	SS	298,43	281,22	298,43	298,43
CLTI	VV	40,67	27,47	30,53	35,60
	VS	102,95	93,55	103,97	103,46
	SV	40,21	27,67	30,75	35,48
	SS	103,97	92,57	102,89	103,43
COTI	VV	153,41	141,50	146,62	150,02
	VS	291,00	276,37	286,37	288,68
	SV	163,45	162,32	168,19	165,82
	SS	565,87	546,11	565,87	565,87