

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Külli Kangro

**Veebirakendus päringuvigade
analüüsimiseks PostgreSQL-i logide abil
kursusel „Andmebaasid“**

Bakalaureusetöö (9 EAP)

Juhendajad: Piret Luik, PhD
Marina Lepp, PhD

Tartu 2024

Veebirakendus päringuvigade analüüsimiseks PostgreSQL-i logide abil kursusel „Andmebaasid“

Lühikokkuvõte:

Tartu Ülikooli kursusel „Andmebaasid“ õpetatakse muuhulgas SQL-päringute koostamist ning enamasti puudub osalejatel varasem kogemus andmebaaside haldamisest. SQL-i päringukeele esmasel õppimisel tehakse mitmesuguseid vigu, millest õppejõud soovivad saada kiiret ülevaadet, et vajadusel pakkuda õppijatele täiendavat tuge. Selle bakalaureusetöö raames luuakse veebirakendus, mis analüüsib õppijate päringuvigu kursusel kasutatava andmebaaside haldussüsteemi PostgreSQL-i logide abil. Rakendus suudab analüüsida korraga terve rühma logifaile ning teeb päringuvigadest kokkuvõtteid nii terve rühma kui ka iga üliõpilase kohta eraldi.

Võtmesõnad: PostgreSQL, logifailid, analüüs, veebirakendus

CERCS: P175 Informaatika, süsteemiteooria, S281 Arvuti õpiprogrammide kasutamise meetoodika ja pedagoogika

Web Application for Analysing SQL Query Errors in PostgreSQL Log Files for the Course „Databases“

Abstract:

The course „Databases“ at the University of Tartu teaches the formulation of SQL queries to students who usually have no prior experience in the field of databases. It is common for people to make mistakes when first learning SQL and the instructors require an overview of these to offer additional support to learners, if needed. The purpose of this thesis was to create a web application for analysing the query errors of students using the logs generated by the PostgreSQL database management system used in the course. The application can analyse logs for an entire group of students and provide summaries of query errors for both the entire group and students individually.

Keywords: PostgreSQL, log files, analysis, web application

CERCS: P175 Informatics, systems theory, S281 Computer-assisted education

Sisukord

Sissejuhatus	5
1. Päringuvead SQL-i õppimisel	7
2. PostgreSQL ja logid	9
2.1 PostgreSQL.....	9
2.2 PostgreSQL-i logid	9
3. Olemasolevad analüsaatorid.....	14
3.1 Logide analüsaatorid õppetöös	14
3.2 Olemasolevad PostgreSQL-i logide analüsaatorid ja nende puudused	14
4. Veebirakendus PostgreSQL logide analüüsimiseks	16
4.1 Analüsaatori funktsionaalsed nõuded	16
4.2 Analüsaatori mittefunktsionaalsed nõuded	17
4.3 Tehnoloogia ja arhitektuur	18
4.4 Päringuvigade kategooriad ja PostgreSQL-i veateated	19
4.5 Veebirakenduse kasutus ja funktsionaalsus.....	23
4.5.1 Logifaili üleslaadimine.....	23
Analüüsi tulemuste vaade	24
5. Tagasiside ja võimalikud edasiarendused	29
5.1 Kasutajate tagasiside	29
5.2 Võimalikud muudatused ja edasiarendused	31
Kokkuvõte	33

Lisad	37
I. PostgreSQL JSON-formaadis täielik logiobjekt	37
II. Rühma analüüsi kokkuvõtte vaade	38
III. Individuaalse analüüsi vaade	39
IV. Litsents	41

Sissejuhatus

Tartu Ülikooli kursusel „Andmebaasid“ [1] antakse ülevaade relatsiooniliste andmebaaside loomisest, kasutamisest ja teooria alustest. Sealhulgas õpetatakse kursusel päringukeele SQL (*Structured Query Language*) abil andmete lisamist, lugemist ja muutmist andmebaasis. SQL päringute õppimisel kasutatakse kursusel andmebaaside haldamissüsteemi PostgreSQL, mis muuhulgas võimaldab automaatselt logida kõiki kasutaja tehtud päringuid ja tekkinud veateateid [2].

SQL-i esmasel õppimisel tehakse päringute koostamisel mitmesuguseid vigu [3]. „Andmebaaside“ kursusel kasutatakse ümberpööratud klassiruumi metoodikat, kus uued teemad omandatakse iseseisvalt, rakendatakse kodutööde lahendamisel ning seejärel arutatakse tehtut praktikumides. Kodutööde tähtaeg on pühapäeva keskööl, mistõttu ei jõua õpperühmade juhendajad uue nädala alguses toimuvate praktikumide ajaks kodutöid analüüsida ja puudub selgem pilt, milliseid teemasid oleks enam tarvis arutada. Õppejõud vajavad kiiret ülevaadet, kui palju ja mis tüüpi vigu õppijad SQL-päringute koostamisel kõige rohkem teevad, et saada aimu, millistel teemadel või ka millistele üliõpilastele tuleks pakkuda täiendavat tuge ja juhendamist.

Hetkel puudub „Andmebaaside“ kursusel päringuvigade jooksvaks jälgimiseks sobiv tööriist. Olemasolevad rakendused PostgreSQL-i logide analüüsimiseks keskenduvad süsteemi jõudlust parandavale infole, päringute veateadete sektsioon on väike või puudulik ja selles kajastatakse vaid mõned levinumad veateated või ei eristata veatüüpe piisavalt detailsealt [4, 5, 6]. Lisaks puudub neil veebipõhine kasutajaliides, kuhu mugavalt õpperühma logifaile korraga analüüsimiseks sisestada – olemasolevad lahendused on mõeldud kasutamiseks käsurealt või arvutisse installeeritava rakendusena. Mõlemal juhul tuleb programmile sisendiks anda analüüsitava logide asukoht kettal.

Kõigest sellest lähtuvalt on käesoleva töö eesmärk luua veebirakendus, mis võimaldab analüüsida päringuvigu PostgreSQL-i logifailides. Töö esimene peatükk annab ülevaate SQL-i õppimisel sagedamini tehtavatest päringuvigadest. Teine peatükk tutvustab PostgreSQL-i süsteemi ning selles loodavate logifailide struktuuri. Kolmandas peatükis kirjeldatakse olemasolevate lahenduste puudusi ja funktsionaalseid ning mittefunktsionaalseid nõudeid loodavale analüsaatorile. Neljas peatükk põhjendab analüsaatori tehnoloogiaga valikuid, kirjeldab analüüsitavaid veateateid ning analüsaatori

funktsionaalsust ja kasutamist. Viies peatükk võtab kokku praktikumijuhendajate tagasiside ja arutleb võimalikke edasiarendusi.

1. Päringuvead SQL-i õppimisel

Selles töös loodav analüsaator uurib SQL-päringute koostamisel tehtud vigu. SQL on levinuim päringukeel relatsiooniliste andmebaaside jaoks, ülikooli tasemel on seda õpetatud aastakümneid [7, 8]. Deklaratiivse keelena erineb see paljudest programmeerijatele tuttavatest imperatiivsetest keeltest – päringut koostades keskendutakse eelkõige ülesande või soovitava tulemuse, mitte lahenduskäigu kirjeldamisele [9].

Päringuvigu SQL-i esmasel õppimisel vaadeldakse sageli kahes põhikategoorias: süntaksivead ja semantilised vead [3, 10]. Süntaksivigadeks loetakse kõik vead, mille korral päringut ei jooksutata ja andmebaasi haldussüsteem tagastab veateate. Semantilisteks loetakse vigu, mis on süntaksi mõttes korrektsed ning käivituvad seega edukalt ja tagastavad tulemuse (k.a tühja tulemuse), kuid tulemus ei ole ülesande püstitust arvestades õige [11]. Taolisi semantilisi vigu on uuritud veel omakorda kahes kategoorias: vigaseid päringuid eristatakse selle järgi, kas tulemuse õigsuse hindamiseks on tarvis teada ülesande püstitust või mitte. Sellisel käsitlusel on nimetatud esimest kategooriat semantilisteks vigadeks ja teist loogikavigadeks [8].

Vaatame näitena päringut joonisel 1. Tegu võib olla korrektse päringuga juhul, kui ülesandeks on leida tabelist read, kus nimi on Stella või Martin. Kui aga ülesande püstituses on küsitud teistsuguseid nimesid, on antud päring vale ja tegu oleks semantilise veaga.

```
SELECT *  
  
FROM inimesed  
  
WHERE nimi = 'Stella' OR nimi = 'Martin';
```

Joonis 1. Võimalik semantiline viga

Vaatame näitena päringut joonisel 2. See päring on süntaktiliselt õige, aga annab alati tühja tulemuse, sõltumata ülesande püstitusest. Sellist viga võib nimetada loogikaveaks.

```
SELECT *  
  
FROM inimesed  
  
WHERE nimi = 'Stella' AND nimi = 'Martin';
```

Joonis 2. Loogikaviga

Selles töös käsitletakse ainult süntaksivigu ehk vigu, mille korral annavad andmebaasi haldussüsteemid veateate. Vaatame näitena päringut joonisel 3. Kuna tunnus „nimi“ on tõenäoliselt sõne, ei saa seda arvuga võrrelda. Seetõttu ei tagastaks selline päring tühja või valet tulemust, vaid süsteem annaks veateate.

```
SELECT * FROM inimesed WHERE nimi > 5;
```

Joonis 3. Süntaksiviga

Vaatame näitena päringut joonisel 4. Sõltuvalt kontekstist võib see päring olla edukas või tekitada erinevaid veateateid. Näiteks annaks andmebaasi haldussüsteem veateate juhtudel, kui tabelit „Asulad“ ei eksisteeri, tunnuse „id“ andmetüüp ei ole arv või tabelil on kitsendus, et „id“ peab olema unikaalne, aga tabelis on juba asula võtmega 11.

```
INSERT INTO Asulad (id, nimi) VALUES (11, 'Rakvere');
```

Joonis 4. Võimalik süntaksiviga

Süntaksivigu tehakse õppimise käigus reeglina kõige rohkem ning ülesandeid jäetakse pooleli pigem süntaksi kui semantiliste vigade tõttu [7]. Ülesannete lõppvastustes aga esineb enamasti kõige rohkem loogikavigu, tõenäoliselt seetõttu, et süntaksivigadega kaasnevad veateated aitavad süntaksi parandada [10]. Käesolevas töös uuritakse vigu lähtuvalt andmebaasi haldussüsteemi veateadetest ja ei analüüsita tehtud päringute sisu, milleks kasutatakse kursusel automaatkontrolli, seega loogikavigu edaspidi ei käsitleta.

2. PostgreSQL ja logid

Andmebaaside haldamiseks SQL-i kaudu leidub mitmeid süsteeme. See peatükk tutvustab PostgreSQL-i süsteemi, mille abil õpetatakse Tartu Ülikooli kursusel „Andmebaasid“ SQL-päringute koostamist. Lisaks kirjeldatakse PostgreSQL-i süsteemi poolt loodava logi formaati ja analüüsiks vajalikke või kasulikke seadistusi.

2.1 PostgreSQL

PostgreSQL on avatud lähtekoodiga süsteem, mis on loodud peamiselt relatsiooniliste andmebaaside loomiseks ja haldamiseks SQL-päringutega. PostgreSQL on tuntud oma töökindluse, jõudluse ja mitmekülgse funktsionaalsuse poolest, see on tasuta, avatud lähtekoodiga ning töötab kõigil levinud operatsioonisüsteemidel [12]. Sellel on palju laiendusi, näiteks on võimalik luua uusi andmetüüpe ja funktsioone. Lisaks on PostgreSQL osutunud praktikas kergesti laiendatavaks andmete hulga või samaaegsete kasutajate koguse suure kasvu korral [12]. Tartu Ülikooli kursusel „Andmebaasid“ on päringute loomiseks ja PostgreSQL-is andmebaasidega suhtluseks kasutusel rakendus DBeaver [13].

2.2 PostgreSQL-i logid

PostgreSQL süsteem võimaldab erinevaid viise oma serveri tegevuse logimiseks [2]:

- standardveavoog,
- CSV-formaat,
- JSON-formaat,
- *syslog* (UNIX-i laadsed süsteemid),
- *event log* (Windows).

Neist mugavaim programmiga lugemiseks ja logiobjektideks teisendamiseks on JSON-formaat [2]. JSON-formaadis väljastatud PostgreSQL-i logifailis on iga logitud sündmuse kohta loodud üks JSON-objekt (Lisa 1). Iga logiobjekt paikneb logifailis ühel eraldi real. Selles töös vaatleme eelkõige väljasid *timestamp*, *error_severity*, *state_code*, *message*, *statement*, *hint*, *context* ja *internal_query*, kusjuures korrektsel päringul puuduvad alati väljad *statement*, *hint*, *context* ja *internal_query* (vt joonised 5 ja 6).

```
{
  "timestamp": "2024-02-23 16:18:54.983 EET",
  "error_severity": "ERROR",
  "state_code": "42883",
  "message": "function age(date, integer) does not exist",
  "hint": "No function matches the given name and argument types. You might
  need to add explicit type casts.",
  "statement": "select eesnimi, perenimi, sugu, age(synniaeg, 1) from
  isikud"
}
```

Joonis 5. Mõned vaadeldavad väljad vigase päringuga logiobjektile

```
{
  "timestamp": "2024-02-23 15:57:34.526 EET",
  "error_severity": "LOG",
  "message": "execute <unnamed>: select eesnimi, perenimi from isikud"
}
```

Joonis 6. Vaadeldavad väljad korrektse päringuga logiobjektile

Vaadeldavate väljade tähendused on järgmised [2]:

- 1) *timestamp* – ajahetk, mil logitud sündmus toimus;
- 2) *error_severity* – veatase, mis annab märku, kas ja kui tõsise veaga on tegu.
Päringutega seotult esinevad kaks veataset:
 - „LOG“ viitab tavapärasele andmebaasi tegevusele, nagu edukad päringud,
 - „ERROR“ viitab veale, mis põhjustas käsu peatamise;
- 3) *state_code* – veakood;
- 4) *message* – sündmuse kirjeldus. Vigase päringu korral on kirjelduses veateade (vt joonis 5), korrektse päringu korral sisaldub päring kirjelduses (vt joonis 6);
- 5) *statement* – vigane päring. Väli esineb ainult vigase päringu korral;
- 6) *hint* – vihje vea parandamiseks juhul, kui PostgreSQL-i süsteem oskab lahendust pakkuda. Väli võib esineda ainult vigase päringu korral;
- 7) *context* – vea kontekst, näiteks asukoht funktsioonis, mille kutsumisest viga tuleneb. Väli võib esineda ainult vigase päringu korral;

- 8) *internal_query* – päring, mis veateate põhjustas, juhul kui see erineb kasutaja tehtud päringust, näiteks funktsioonis sisalduv vigane päring. Väli võib esineda ainult vigase päringu korral.

Juhul kui korrektne päring käivitab vigase funktsiooni, kajastub logis kõigepealt kasutaja tehtud päring korrektsena. Järgneb veateatega logikanne, kus välja *statement* väärtuseks on sama päring ja lisaks on eraldi väljal *internal_query* funktsioonist leitud vigane päring, mis veateate põhjustas ning väljal *context* kirjas funktsiooni nimi ja selles leiduva vigase päringu asukoht (vt joonis 7).

```
{
  "timestamp": "2024-04-29 12:58:46.675 EEST",
  "error_severity": "LOG",
  "message": "execute <unnamed>: DELETE FROM klubid WHERE nimi='Klubi Kustutamiseks'"
}
{
  "timestamp": "2024-04-29 12:58:46.675 EEST",
  "error_severity": "ERROR",
  "state_code": "42883",
  "message": "operator does not exist: character varying = integer",
  "hint": "No operator matches the given name and argument types. You might need to add explicit type casts.",
  "internal_query": "select id from asulad where nimi = old.asula",
  "context": "PL/pgSQL function tg_kustuta_klubi() line 4 at SQL statement",
  "statement": "DELETE FROM klubid WHERE nimi='Klubi Kustutamiseks'"
}
```

Joonis 7. Vaadeldavad väljad funktsioonist tuleneva veateate korral

Vaikimisi ehk ilma kasutajapoolse seadistusega kirjutatakse logid standardveavoo formaadis (vt joonis 8), kus puuduvad paljud tehnilised väljad, mis on olemas JSON-formaadis logidel (vt lisa 1). Standardveavoo kujul logikannetel on siiski olemas analüüsiks kasutatavad väljad, välja arvatud *state_code*. Vaikimisi seadete korral aga ei logita korrektseid päringuid – kõigist tehtud päringutest kajastuvad logifailis vaid vigased päringud koos vastavate veateadete, vihjete, konteksti ja sisemiste päringutega, mis kirjutatakse eraldi ridadele.

```

2023-08-08 14:30:45.460 EEST [25444] ERROR:  syntax error at or near "AS" at character 65
2023-08-08 14:30:45.460 EEST [25444] STATEMENT:  SELECT lavanimi, extract (year from
age(current_date, synniaeg) AS hetkevanus
FROM lauljad
WHERE extract (year from synniaeg) > 1980 AND
LENGTH(lavanimi) <=4
ORDER BY 2;

2024-05-02 15:37:26.107 EEST [17132] ERROR:  relation "p" does not exist at character 13
2024-05-02 15:37:26.107 EEST [17132] QUERY:   DELETE FROM p WHERE id=NEW.id
2024-05-02 15:37:26.107 EEST [17132] CONTEXT:  PL/pgSQL function f_ajakontroll2() line 8 at
SQL statement
2024-05-02 15:37:26.107 EEST [17132] STATEMENT:  INSERT INTO partiid (turniir, algushetk,
valge, must, valge_tulemus, must_tulemus) VALUES (41, '2005-01-12 08:05:00', 73, 92, 1, 1)

```

Joonis 8. Standardveavoona logifaili sissekanded

Logimise ja muude PostgreSQL-i seadete muutmiseks on PostgreSQL-i programmifailide seas tekstifail „postgresql.conf“. Seadetefailis on sektsioonid eraldatud pealkirjadega, logimisseadeid sisaldav sektsioon on pealkirjaga „REPORTING AND LOGGING“. Suurem osa seadeid on välja kommenteeritud „#“-märgiga rea alguses, nende väärtuseks on informatiivsel eesmärgil pandud vaikeväärtused (vt joonis 9). Selliste seadete muutmiseks tuleb lisaks välja väärtuse muutmisele eemaldada rea algusest kommentaarisümbol.

```

#-----
# REPORTING AND LOGGING
#-----

# - Where to Log -

#log_destination = 'stderr'      # Valid values are combinations of
                                # stderr, csvlog, jsonlog, syslog, and
                                # eventlog, depending on platform.
                                # csvlog and jsonlog require
                                # logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on          # Enable capturing of stderr, jsonlog,
                                # and csvlog into log files. Required
                                # to be on for csvlogs and jsonlogs.
                                # (change requires restart)

```

Joonis 9. Lõik PostgreSQL-i seadetefailist

Lisaks on mõnede väljade juures selgitavad kommentaarid, näiteks logide seadistamiseks JSON-vormingusse tuleb logimisseadete sektsioonis välja *log_destination* väärtuseks

sisestada *'jsonlog'*. Kui logis soovitakse näha infot mitte ainult vigade, vaid kõikide tehtud päringute kohta, siis tuleks välja *log_statement* väärtuseks panna *'all'*. Vaikimisi kasutatakse ühte logifaili 24 tundi, mille järel luuakse ja hakatakse täitma uut logifaili. Kursuse „Andmebaasid“ tarbeks pannakse välja *log_rotation_age* väärtuseks *'150d'* ehk 150 päeva, et üks logifail oleks kasutusel kursuse lõpuni.

3. Olemasolevad analüsaatorid

Selles peatükis tutvustatakse paari logide analüsaatorit, mis loodi teiste Tartu Ülikooli kursuste toetamiseks ning kirjeldatakse olemasolevate PostgreSQL-i logide analüsaatorite puuduseid lähtuvalt kursuse „Andmebaasid“ vajadustest ja kasutuse eripäradest.

3.1 Logide analüsaatorid õppetöös

Tartu Ülikoolis on varasemalt loodud mitmeid logifailide analüsaatoreid programmeerimise algkursustel kasutatava programmeerimiskeskonna Thonny jaoks, mille abil õpetatakse programmeerimiskeelt Python. Johan Erik Pukk koostas bakalaureusetöö raames Thonny logifailidele tähepaaride analüsaatori, mis saab olla aluseks tulevasele tippimisstiilide analüüsile, mille abil on võimalik ennustada eksamitulemusi ja tuvastada plagiaati [14]. Rene Küti bakalaureusetöö raames loodud Thonny logide analüüsi veebirakendus kuvab iga analüüsitud logifaili kohta algusaja, lõppaja, programmi käivitamiste arvu, vigade arvu jm ning võimaldab kogu kasutussessiooni taasesitamist [15]. On leitud, et Thonny logide analüüsil põhinev tagasiside kursuse vältel aitas algajatel programmeerijatel saavutada parema eksamitulemuse [16].

3.2 Olemasolevad PostgreSQL-i logide analüsaatorid ja nende puudused

Üks levinumaid ja kiiremaid PostgreSQL-i logide analüsaatoreid on pgBadger [4]. See on tasuta iseseisev programm, mis näitab sisestatud logifailide põhjal andmebaasi kasutusstatistikat. Analüsaator tuvastab automaatselt logi formaadi (standardveavoog, CSV-formaat, JSON-formaat, *syslog*) ning suudab edukalt töödelda ka väga suurt või kokkupakitud faili. Analüüsi tulemus on kohandatav – ebavajalikud komponendid saab eemaldada, jättes alles ainult vigade ja sündmuste kokkuvõtte. Vigade ja sündmuste kokkuvõttes tuuakse välja veatasemete esinemissagedused sektordiagrammina, ebanormaalsete logisündmuste (veatasemed *WARNING*, *ERROR*, *FATAL*, *PANIC*) jaotus ajas ning esinenud veateated koos koguse ja näidispäringuga [17]. Programm tuleb installeerida arvutisse ja on mõeldud kasutamiseks käsurealt, logikokkuvõtted koostatakse HTML-failidena.

Analüsaator „pganalyze“ Log Insights [5] eristab lisaks logisündmustele ka erinevaid päringuvigade tüüpe veakoodi (vt ptk 2.2) abil [18]. See on tasuline graafilise

kasutajaliidesega programm, mis näitab statistikat andmebaasi töö ja kasutuse kohta, näiteks automaatsed protsessid ja häired serveri töös, ühendused andmebaasiga, kitsenduste rikkumised, aeglasel ja tühistatud päringud ning kasutaja vead [5]. Rakenduse kasutamiseks tuleb see alla laadida ja anda käsureal ligipääs PostgreSQL-i logifailide kaustale. Log Insights on eelkõige mõeldud pidevalt lisanduvate andmebaasi tegevuse logide jälgimiseks.

Olemasolevad PostgreSQL-i logide analüsaatorid on mõeldud eelkõige tarkvararakendustega seotud andmebaaside töö ja kasutuse jälgimiseks ning optimeerimiseks [4, 5, 6]. Seetõttu on analüüsi tulemusel rõhk jõudlusel, kasutussagedusel ja süsteemis esinevatel probleemidel – esile tuuakse kõige aeglasemad, mahukamad ja enim kasutatud päringud ning tõrked andmebaasi töös ja ühenduses. Tõrgete osas tuuakse enamasti välja küll sagedasemad veateated, mis ilmnevad muuhulgas seoses vigaste päringutega, kuid sel juhul piirdub analüüs vea ja selle esinemissageduse kuvamisega [4, 6]. Kuna tarkvararakendustes kasutatakse tavaliselt korrektseid, varasemalt testitud päringuid, siis ei pruugi päringuvigade analüüsiks erilist vajadust olla. Analüsaator „pganalyze“ Log Insights eristab muuhulgas päringuvigade tüüpe, näiteks defineerimata väärtuste kasutamine päringutes, kitsenduste rikkumised ja üldised süntaksivead, kuid tulemused visualiseeritakse vaid eraldi ajaskaaladel ja koos suure hulga ebavajaliku infoga serveri töö kohta [5]. Olemasolevad analüsaatorid ei ole kasutatavad veebirakendusena, need tuleb arvutisse alla laadida.

Tartu Ülikooli kursusel „Andmebaasid“ kasutatakse päringute õpetamiseks erinevaid väiksemaid andmebaase, mis koosnevad testandmetest. Rõhk on päringute sisu õigsusel, päringute jõudlus on teisejärguline ning väheste andmetega ei pruugi jõudlusprobleemid selgelt välja joonistuda võrreldes massandmetega. Väikesed erinevused päringute täitmise kiirusel võivad tuleneda arvutite eripäradest. Seega on suurem osa olemasolevate PostgreSQL-i logide analüsaatorite pakutavast funktsionaalsusest üleliigne ning vajalik osa ehk päringuvigade analüüs on puudulik või liiga pealiskaudne.

4. Veebirakendus PostgreSQL logide analüüsimiseks

Siin peatükis esitatakse loodava veebirakenduse funktsionaalsed ja mittefunktsionaalsed nõuded ning kirjeldatakse analüsaatori tehnoloogilist arhitektuuri, funktsionaalsust ja kasutust. PostgreSQL logide analüsaatori veebirakendus asub aadressil <https://postgresql-log-analyser.onrender.com/>.

4.1 Analüsaatori funktsionaalsed nõuded

Funktsionaalsed nõuded on ettenähtud tegevused, mida rakendus peab olema võimeline kasutaja heaks sooritama [19]. Analüsaatori funktsionaalsus otsustati enamjaolt enne arenduse algust koostöös juhendajaga ning täiendati jooksvalt tagasiside põhjal rakenduse kasutuse käigus. PostgreSQL-i logide analüsaatori veebirakenduse funktsionaalsed nõuded on:

1. Analüsaator peab suutma analüüsida üksikut logifaili, mis on JSON- või standardveavoo formaadis (vastavalt „*json*“ või „*log*“ lõpuga failinimed).
2. Analüsaator peab suutma analüüsida kursuse „Andmebaasid“ Moodle'i keskkonnast saadavat kodutööde ZIP-faili.
3. Analüsaator peab sisendina vastu võtma analüüsi ajavahemiku
4. Analüsaator peab analüüsima vaid valitud ajavahemikku langevaid päringuid
5. Analüsaator peab ZIP-failis igast kaustast analüüsima kõiki JSON-formaadis või standardveavoo kujul logifaile (vt nõue 1), ignoreerima võib „*zip*“ lõpuga faile.
6. Analüsaator peab ZIP-faili analüüsi tulemuses esitama kokkuvõtted kogu rühma kohta ja iga üliõpilase kohta eraldi.
7. Analüsaator peab iga üliõpilase analüüsi kokkuvõttes näitama järgnevat infot:
 - a. analüüsitud failide nimed,
 - b. tehtud päringute koguarv,
 - c. korrektsete päringute arv,
 - d. korrektsete päringute osakaal päringute koguarvust,
 - e. vigaste päringute arv,
 - f. vigaste päringute osakaal päringute koguarvust,
 - g. ühekordsete vigade arv,
 - h. korrektsete ja vigaste päringute visuaalne ajaline järjestus,
 - i. esimese analüüsitud päringu aeg,

- j. viimase analüüsitud päringu aeg,
 - k. veatüüpide jaotus arvudena: süntaksivead, defineerimata väärtuse vead, kitsenduse rikkumised, trükivead, agregeeritud funktsiooni vead, topelt defineerimine, muud vead,
 - l. korduvad vead: veateade, korduste arv ja vastav päring,
 - m. logide sisu järgneva infoga iga logikande kohta: päringu aeg, päring, päringu korrektsus, vigase päringu korral veatüüp ja veateade.
8. Analüsaator peab rühma kokkuvõttes näitama järgnevat infot:
- a. tehtud päringute koguarv,
 - b. korrektsete päringute arv,
 - c. korrektsete päringute osakaal päringute koguarvust,
 - d. vigaste päringute arv,
 - e. vigaste päringute osakaal päringute koguarvust,
 - f. ühekordsete vigade arv,
 - g. korrektsete ja vigaste päringute osakaalud visuaalselt
 - h. veatüüpide jaotus arvudena: süntaksivead, defineerimata väärtuse vead, kitsenduse rikkumised, trükivead, agregeeritud funktsiooni vead, topelt defineerimine, muud vead,
 - i. korduvad vead: veateade, korduste arv.

Eelduste kohaselt koosneb analüüsitud ZIP-fail üliõpilaste nimedega kaustadest, milles asub muuhulgas iga vastava üliõpilase logifail või logifailid. Kui rakendusse sisestatakse muu struktuuriga ZIP-fail, ei pruugi väljund olla õige.

4.2 Analüsaatori mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded süsteemi omadused ja näitajad, mis kirjeldavad, kui hästi süsteem oma tööd teeb [20]. Veebirakenduse mittefunktsionaalsed nõuded otsustas suuresti töö autor positiivse kasutajakogemuse tagamiseks. PostgreSQL-i logide analüsaatori veebirakenduse funktsionaalsed nõuded on:

1. Analüsaator peab lubama kuni 20 samaaegset kasutajat ilma muutusteta jõudluses.
2. Analüsaator peab ühe rühma analüüsi tulemuse kuvama 5 sekundi jooksul.
3. Analüsaatori kasutajaliides peab olema intuitiivne.

4. Analüsaatori põhifunktsionaalsus peab asuma minimaalse navigatsiooni kaugusel pealehest.
5. Analüsaator peab kogu rühma kokkuvõtte või individuaalse analüüsi tulemuse kuvama ühes vaates.

Nõue lubada samaaegselt kuni 20 kasutajat ilma jõudlusprobleemideta tuleneb kursuse juhendajate arvust, et kõik õppejõud ja ka analüsaatori arendaja saaksid vajadusel rakendust korraga kasutada.

4.3 Tehnoloogia ja arhitektuur

Valminud PostgreSQL logide analüsaator on veebirakendus, mis koosneb eraldi serverist ja kasutajaliidestest¹. Eraldi serveri loomine ei olnud analüsaatori praeguse funktsionaalsuse jaoks vajalik, kuid arhitektuuri valikul on arvestatud juhendajate soove tulevase funktsionaalsuse jaoks, mis lisab serveripoolsele keerukust.

Server, kus toimub logifailide lahtipakkimine, lugemine ja analüüs, on kirjutatud programmeerimiskeeles Java, kasutades Spring Boot raamistikku. Programmeerimiskeele valiku üheks peamiseks põhjuseks on selle laialdane levik ja kompetents Tartu Ülikoolis – soovi korral on tulevastel kasutajatel lihtne leida rakendusele edasiarendaja. Teine põhjus on Java programmeerimiskeele valikuks on Spring Boot raamistiku kasutamise võimalus. Spring Boot on laialdaselt levinud raamistik Java keelele, mis lihtsustab tarkvararakenduste loomist [21]. See sisaldab suurel hulgal eelkonfigureeritud komponente, nagu sisseehitatud andmebaasid ja serverid, ning suudab neid suuresti ise peidetult hallata, vähendades seeläbi süsteemi loomiseks kirjutatava koodi kogust.

Analüsaatori kasutajaliides on kirjutatud Vue.js raamistikus. Vue.js on avatud lähtekoodiga JavaScripti raamistik, mis võimaldab luua komponentidest koosnevaid kasutajaliideseid [22]. Mainitud komponendid sisaldavad HTML-i, CSS-i ja JavaScripti ning on enamasti kirjutatud kasutades spetsiaalset süntaksi, mis sarnaneb HTML-ile, kuid võimaldab kuvatavate elementidega siduda muutuvaid andmeid, luua tingimuslikke struktuure jpm [23]. Raamistiku valimisel peeti oluliseks selle laialdast levikut, et leiduks võimalikult palju kergesti ligipääsetavaid ressursse, mis arendust toetavad. Seetõttu oli valikus kolm

¹ Programmi lähtekood on saadaval aadressil <https://github.com/kangrok/postgresql-log-analyser>

populaarseimat JavaScripti raamistikku React, Angular ja Vue.js, millest otsustati Vue.js kasuks, kuna selle loomisel on eriti keskendunud kiirele õpitavusele [24]. Kasutajaliidese loomisel kasutati laialdaselt teeki Vuetify, mis sisaldab valmis disainitud Vue.js komponente, nagu tabelid, menüüd, nupud jpm [25]. Vuetify abil saab kiiresti luua hea kasutajakogemusega veebilehti, kuna valmis komponendid kasutavad ühtset stiili ja nende disainimisel on pööratud suurt tähelepanu kasutajasõbralikkusele. Diagrammide loomiseks kasutatakse teeki Chart.js, mis on avatud lähtekoodiga ja toetatud ka Vue-s [26].

4.4 Päringuvigade kategooriad ja PostgreSQL-i veateated

PostgreSQL-i logikannete veatüübi määramiseks soovitatakse üldjuhul kasutada veakoodi (*state_code*), mitte veateate teksti (*message*), sest viimane võib sõltuda PostgreSQL-i versioonist [27]. Kuna analüsaatori sisendiks võivad olla ka standardveavookujul logifailid, kus veakoodid ei kajastu, kasutatakse päringuvigade liigitamiseks siiski enamasti veateateid.

Analüüsitavad veateated on näiteks järgnevad:

- 1) „column "isikud.ranking" must appear in the GROUP BY clause or be used in an aggregate function“;
- 2) „syntax error at or near "from"“;
- 3) „column "arv" does not exist“;
- 4) „syntax error at or near ", "“;
- 5) „aggregate functions are not allowed in WHERE“.

Analüüsi käigus jaotatakse vead üheksasse kategooriasse: süntaksivead, defineerimata väärtuse kasutamine, kitsenduse rikkumine, trükivead, agregeeritud funktsiooni vead, topelt defineerimine, andmetüüpide mittevastavus, ise loodud veateated, muud vead. Vigade kategooriad valiti välja eelneva aasta logifailide analüüsi alusel ning kohendati jooksvalt kursuse käigus õppejõududelt saadud sisendi põhjal.

Trükivigadeks loetakse vaid sellised päringulogides leiduvad vead, kus PostgreSQL-i server on tuvastanud, et mingit päringus leiduvat väärtust (näiteks tabeli või tunnuse nime) süsteemis ei eksisteeri, kuid leidub sarnase nimega väärtus. Sellisel juhul on logikandel olemas vihje väli (*hint*), kus pakutakse olemasolevat väärtust vigase asemele (joonis 10).

```
"message": "column 'lõpuhetk' does not exist",  
"hint": "Perhaps you meant to reference the column 'partiid.lopphetk'",  
"statement": "SELECT eesnimi, perenimi FROM isikud, partiid WHERE  
right(lõpuhetk, 5) - right(algushetk, 5) < 11"
```

Joonis 10. Trükiveaga päring, veateade ja vihje logis

Andmetüüpide mittevastavuseks loetakse esiteks vead, kus PostgreSQL-i server on tuvastanud, et on kasutatud valet andmetüüpi. Sellisel juhul on JSON-formaadis logikandel olemas veakood „42804“ (joonis 11) ehk PostgreSQL-i süsteemis on tegu *datatype_mismatch*-tüüpi veaga [27]. Lisaks loetakse andmetüüpide mittevastavuseks vead, kus on päringus kasutatud mingit operaatorit või muud funktsiooni valede andmetüüpidega ning veateate kohaselt sellist funktsiooni ei eksisteeri. Sellisel juhul on logikandes vihje, mis soovib proovida andmetüüpe teisendada (joonis 12).

```
"state_code": "42804",  
"message": "argument of WHERE must be type boolean, not type numeric",  
"statement": "SELECT nimi FROM klubid WHERE (SELECT avg(valge_tulemus)  
FROM isikud, partiid WHERE partiid.valge = isikud.id AND isikud.klubi =  
klubid.id) "
```

Joonis 11. Andmetüüpide mittevastavusega päring, veakood ja -teade

```
"message": "operator does not exist: timestamp without time zone =  
integer",  
"hint": "No operator matches the given name and argument types. You might  
need to add explicit type casts.",  
"statement": "SELECT eesnimi, perenimi FROM isikud, partiid WHERE  
left(eesnimi, 1) = 'A' AND partiid.algushetk = 2005-04-03"
```

Joonis 12. Andmetüüpide mittevastavusega päring, veateade ja vihje

Defineerimata väärtuse kasutamiseks loetakse sellised vead, mille veateates on öeldud, et mingit päringus leiduvat väärtust ei eksisteeri ja mis pole eelnevate kirjelduste järgi tuvastatav trükiveana ega andmetüüpide mittevastavusena (joonis 13).

```
"message": "relation 'tabeli_nimi' does not exist",  
"statement": "SELECT * FROM tabeli_nimi"
```

Joonis 13. Defineerimata väärtusega päring ja veateade

Agregeeritud funktsiooni vigadeks loetakse sellised vead, mille veateade sisaldab sõnu „aggregate“, „group“ või „having“, mis viitavad raskustele agregatsiooni mõistmisel (joonis 14).

```
"message": "aggregate functions are not allowed in GROUP BY",  
"statement": "select round(ranking, -2) as 'rängingu klass',  
count(ranking) as arv from isikud group by count(ranking)"
```

Joonis 14. Agregeeritud funktsiooni veaga päring ja veateade

Süntaksivigadeks loetakse vead, mille veateade sisaldab sõna „syntax“ ja mis pole eelnevate kirjelduste järgi tuvastatav agregeeritud funktsiooni veana (joonis 15). Enamasti on veateade kujul „syntax error at or near...“, millele järgneb sõna või sümbol päringust, kust alates on päringu süntaks vigane. See veatüüp võib sageli hõlmata ka trükivigu või muid veatüüpe, mida PostgreSQL server päringust ei tuvastanud.

```
"message": "syntax error at or near '(',  
"statement": "select synniaeg extract(month from synniaeg) as synnikuu"
```

Joonis 15. Süntaksiveaga päring ja veateade

Kitsenduse rikkumiseks loetakse vead, mille korral on veateates öeldud, milline väärtus või tegevus millist kitsendust rikub (joonis 16).

```
"message": "delete from asulad where nimi = 'Tartu'",  
"statement": "update or delete on table 'asulad' violates foreign key  
constraint 'fk_klubi_2_asula' on table 'klubid'"
```

Joonis 16. Kitsendust rikkuv päring ja veateade

Topelt defineerimiseks loetakse vead, kus on defineeritava väärtuse (tabeli, kitsenduse, funktsiooni vms) nimi sama, mis juba olemasoleval sama tüüpi väärtusel. Sellisel juhul veateates kirjas, et selline väärtus juba eksisteerib (joonis 17).

```
"message":"relation 'nimi_unique' already exists",  
"statement":"ALTER TABLE ONLY public.isikud ADD CONSTRAINT nimi_unique  
UNIQUE (eesnimi, perenimi)"
```

Joonis 17. Topelt defineeriv päring ja veateade

Ise loodud veateated on veakoodiga „P0001“ ehk tüüpi *raise_exception* (joonis 18). Sellised veateated on kasutaja ise lisanud „RAISE“ käsu abil ja sellised vigased päringud on enamasti tehtud testimise eesmärgil, mitte oskamatusel.

```
"state_code":"P0001"  
"message":"Mängija mängib juba samal ajal!",  
"statement":"INSERT INTO partiid (turniir, algushetk, valge, must,  
valge_tulemus, must_tulemus) VALUES (41, '2005-01-12 08:05:00.000', 73,  
92, 1, 1)"
```

Joonis 18. Ise loodud veateade ja selle põhjustanud päring

Kõik ülejäänud vead loetakse kategooriasse “**muud vead**”. Näited muudest vigadest on joonisel 19.

```
{...  
"message":"INSERT has more expressions than target columns",  
"statement":"INSERT INTO riigid(nimi) VALUES ('Estonia', 'Tartu')", ...}  
{...  
"message":"missing FROM-clause entry for table \"old\"",  
"statement":"SELECT * FROM asulad a, klubid k, turniirid t  
WHERE OLD.asula = a.id AND (k.asula = a.id OR t.asula = a.id)", ...}  
{...  
"message":"cannot drop function f_klubi_olemasolu() because other  
objects depend on it",  
"statement":"DROP FUNCTION public.f_klubi_olemasolu()", ...}
```

Joonis 19. Muud vead ja veateated

Vigade arvestamisel eristatakse nn ühekordseid ja korduvaid vigu. Ühe õppija korduvateks vigadeks loetakse sellised vead, mille kohta esineb logifailis mitu logiobjekti ühesuguse veateate ja päringuga ning mis on juba ühekordse veana arvestatud. Näiteks kui kasutaja jooksub ühte ja sama vigast päringut viis korda, loetakse neist esimene ühekordseks veaks ja ülejäänud neli korduvateks vigadeks. Kogu analüüsitava õpperühma peale korduvate lugemisel arvestatakse ainult ühesuguseid veateateid, mitte nendega seotud päringuid, sest on vähem tõenäoline, et erinevad kasutajad tekitavad ühesuguseid vigu täpselt ühesuguste päringutega. Õpperühma analüüsi tulemuses kajastuvad seega üldisemad probleemid, mis rühma õppuritel töö käigus tekkisid.

4.5 Veebirakenduse kasutus ja funktsionaalsus

Veebirakenduses on kaks peamist vaadet: logifailide üleslaadimine ja analüüsi tulemus, kus saab vaadata nii individuaalseid kui ka terve õpperühma kohta käivat analüüsi.

4.5.1 Logifaili üleslaadimine

Rakenduse esileht on mõeldud analüüsi sisendi andmiseks ehk analüüsitava faili üleslaadimiseks ja analüüsi ajavahemiku valimiseks (joonis 20).

The screenshot shows a web application interface with the following elements:

- Laadi üles logifailide kaust**: A section header for uploading log files.
- Vali fail...**: A button with a folder icon to select a file.
- ANALÜÜSI AJAVAHEMIK**: A section header for selecting the analysis time range.
- Radio buttons**: Three options for the time range:
 - ☐ Kõik logid
 - ☐ Viimane nädal
 - ☒ Muu ajavahemik
- Algu:** A date and time picker showing **05/13/2024 03:19 AM**.
- Lõpp:** A date and time picker showing **May 2024** for the month and a calendar grid for the days. The date **13** is selected. To the right of the calendar is a time picker showing **03**, **19**, and **AM**.
- ANALÜÜSI**: A blue button to start the analysis.

Joonis 20. Analüüsi sisendi valimine esilehel

Üles võib laadida üksiku logifaili (kas JSON-formaat või seadistamata standardveavoo formaat) või ZIP-faili, milles on iga kasutaja kohta eraldi kaust, kus asuvad logifailid. Analüüsi ajavahemikuks saab valida kas „kõik logid“ ehk kõik failides leiduvad logid sõltumata ajast, „viimane nädal“ ehk viimase 7 päeva jooksul tehtud logid või „muu ajavahemik“, kus saab ise valida analüüsi algus- ja lõppkuupäeva ja kellaja, kuhu vahele analüüsitavad päringud peavad jääma. Peale logifaili üleslaadimist ja analüüsi ajavahemiku valimist võib vajutada nuppu „Analüüsi“. Seejärel kuvatakse kasutajale analüüsi tulemus.

Analüüsi tulemuste vaade

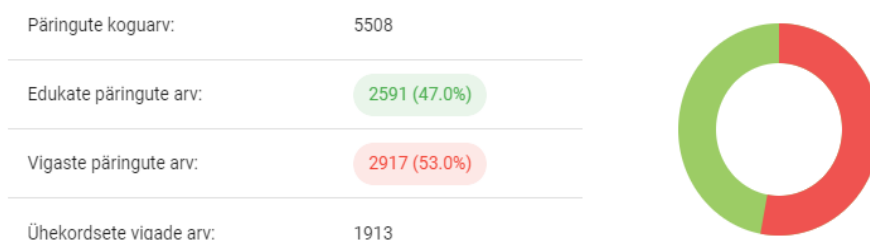
Kui üles laeti ZIP-fail, siis kuvatakse analüüsi tulemustes esimesena rühma kokkuvõte. Sel juhul kuvatakse vaate küljel menüü, mille abil saab liikuda individuaalsete analüüside vaadetesse valides vastava üliõpilase nime või minna tagasi logifaili üleslaadimise vaatesse, et alustada uut analüüsi muu failiga (joonis 21).



Joonis 21. Küljemenüü eri analüüside vahel navigeerimiseks

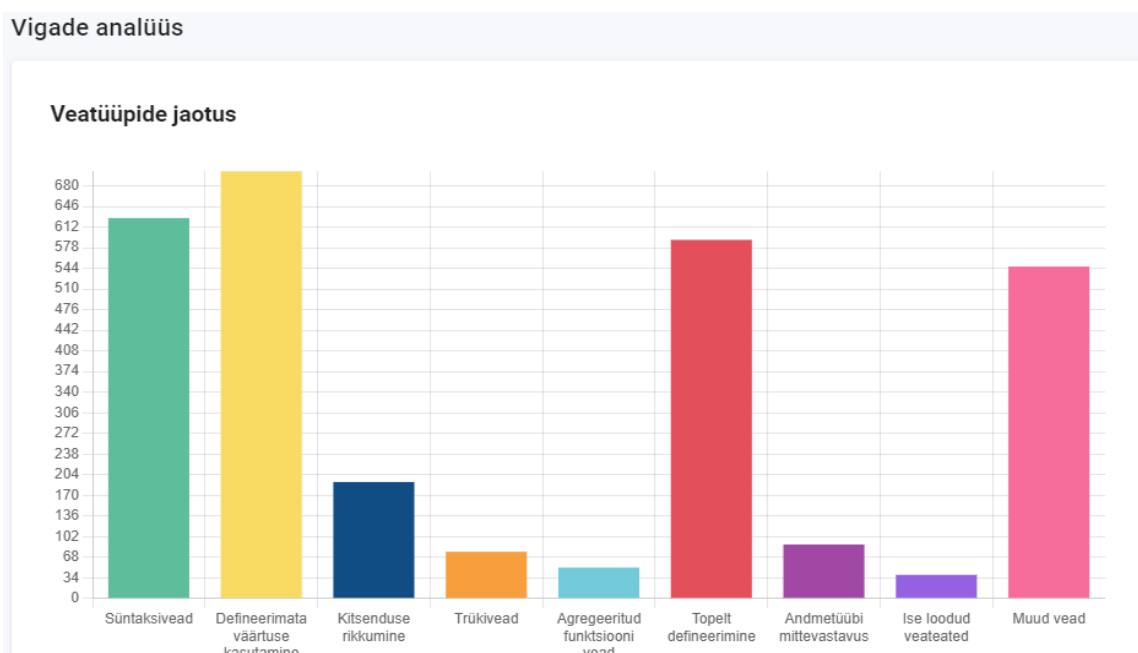
Rühma kokkuvõtte esimeses sektsioonis on näha rühmas tehtud päringute koguarv, korrektsete päringute arv ja protsent koguarvust, vigaste päringute arv ja protsent

koguarvust ning ühekordsete vigade arv (joonis 22). Lisaks on korrektsete ja vigaste päringute osakaalud visualiseeritud diagrammi abil.



Joonis 22. Rühma kokkuvõtte esimene seksioon

Teine seksioon pealkirjaga „Vigade analüüs“ illustreerib eri tüüpi vigade koguseid tulpdiaagrammi abil (joonis 23) eesmärgiga anda kiire ülevaade, millist tüüpi vigu õpperühmas kõige rohkem tegi. Diagramm näitab kõikide vigade jaotumist – muudesse kategooriatesse mittedsobivad vead on lisatud tulpa „Muud vead“. Lisaks kuvatakse vigade analüüsi seksioonis tabelina kõik korduvad vead ehk veateated, mis esinesid logides kokku rohkem kui ühel korral (joonis 24). Selle eesmärk on näidata konkreetseid vigu, mida õpperühmas ülesannete lahendamisel kõige rohkem sooritati. Korduvad vead on sorteeritud veateate koguste järgi kahanevalt, et enim esinenud vead oleksid esimesena näha. Tabeli all ääres saab valida, kui palju andmeid leheküljel näidatakse ning noolte abil lehekülgi sirvida.



Joonis 23. Veatüüpide jaotus vigade analüüsi seksioonis

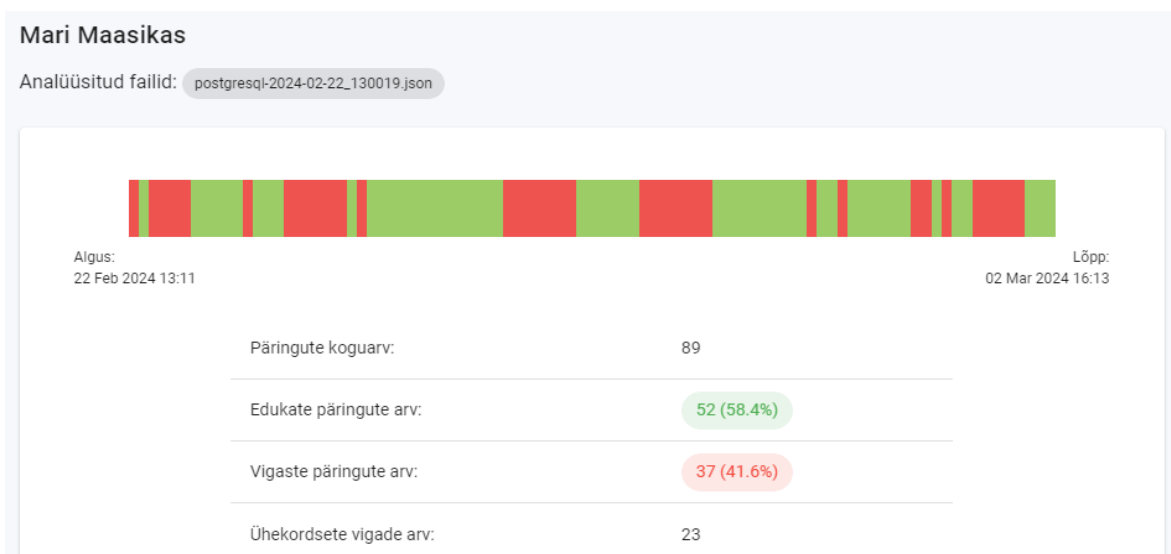
Korduvad vead

Veateade	Korduste arv
syntax error at or near "where"	19
column "isikud.perenimi" must appear in the GROUP BY clause or be used in an aggregate function	17
syntax error at or near "HAVING"	16
aggregate functions are not allowed in WHERE	14
syntax error at end of input	11

Items per page: 5 1-5 of 64 |< < > >|

Joonis 24. Korduvad vead vigade analüüsi sektsioonis rühma kokkuvõttes

Individuaalseid analüüse saab näha valides menüüst vastava üliõpilase nime. Individuaalse analüüsi vaates näidatakse esimeses sektsioonis sarnaselt rühma kokkuvõttega päringute koguarvu, korrektsete ja vigaste päringute arvu ja osakaalu ning ühekordsete vigade arvu. Lisaks näidatakse analüüsitud logifailide nimesid, päringute alguse ja lõpuaega ehk esimese ja viimase päringu tegemise aega ning õppejõudude soovil visualiseeritakse ka korrektsete ja vigaste päringute järjestust koos päringute algus- ja lõppajaga (joonis 25). Individuaalse vaate vigade analüüsi sektsioon on analoogne rühma kokkuvõttes kuvatava vigade analüüsi sektsiooniga, kuid korduvate vigade tabelis on lisaks võimalus iga rea kohta avada ka veateatele vastav päring (joonis 26).



Joonis 25. Individuaalse analüüsi esimene sektsioon

Korduvad vead

Veateade	Korduste arv	
syntax error at or near "as"	6	▼
column "isikud.ranking" must appear in the GROUP BY clause or be used in an aggregate function	6	▲
<pre>select round(ranking, -2) as klass, count(ranking) as arv from isikud group by klass order by ranking asc</pre>		
syntax error at or near "where"	6	▼
syntax error at or near ">"	3	▼

Joonis 26. Korduvad vead koos päringutega individuaalses analüüsis

Individuaalse analüüsi vaatesse kuulub ka kolmas sektsioon pealkirjaga „Kõik päringud“, milleks on päringulogide vaade. Vajadus sellise vaate järgi tulenes õppejõudude soovist teatud juhtudel näha konkreetsemalt, mida ja millal üliõpilane tegi, näiteks plagiaadijuhtumite tuvastamiseks arvestuspäringute korral, mis tehakse 90 minuti jooksul. Logifailist info haaramine on raskendatud, eriti kui need on JSON-formaadis, sest kogu päringuga seonduv info on paigutatud ühele reale koos paljude parameetritega, mis on õppejõududele ebaolulised. Analüsaatori päringulogis kuvatakse üliõpilase kogu päringulogi või viimased 200 logikannet, kui analüüsi ajavahemikku jääv logi on sellest pikem. Päringulogis kuvatakse iga logikande kohta päringu tegemise aeg, veatüüp (korrektse päringu korral „Korrektne“) ja päring (joonis 27). Päringud on vaikimisi lühendatud ühele reale, kogu sisu kuvamiseks saab ridu pikendada kas ükshaaval kasutades rea lõpus olevat nuppu või kõik read korraga kasutades tabeli kohal olevat nuppu „Pikenda päringud“, samuti saab kõik pikendatud read uuesti lühendada kasutades kõrvalasuvat nuppu „Lühenda päringud“. Vigaste päringute korral on pikendatud ridades näha ka veateade.

PIKENDA PÄRINGUD		LÜHENDA PÄRINGUD
Aeg	Vea tüüp	Päring
23/02/2024, 20:07:17	Süntaks	select round(ranking, -2) as klass, count(eesnimi) as arv from ...
23/02/2024, 20:07:18	Süntaks	select round(ranking, -2) as klass, count(eesnimi) as arv from isikud group by klass having >=5 order by ranking asc syntax error at or near ">="
23/02/2024, 20:07:25	Korrektne	select round(ranking, -2) as klass, count(eesnimi) as arv from ...

Joonis 27. Päringulogide tabel individuaalses analüüsis

Funktsiooni kasutusest tulenevate vigaste logikannete ülesehitus erineb veidi tavalisest logikandest (vt ptk 2.2) ehk veateatele ei vasta mitte päring, mille tõttu funktsiooni sisu käivitati, vaid funktsiooni sees leiduv vigane päring, mis kajastatakse väljal *internal_query*. Seetõttu kuvatakse päringulogi vaates päringu veerus logikandes leiduv sisemine päring, mis tegelikult veateate põhjustas. Lisaks kuvatakse sel juhul veatüübi veerus ka märksõna „Funktsioon“, mis viitab vea tegelikule päritolule ning pikendatud real vea kontekst (logikande väljalt *context*), mis sisaldab vigase funktsiooni nime (joonis 28).

22/04/2024, 16:20:06	Funktsioon Trükitviga	<pre> SELECT perenimi ', ' eesnimi AS mangija_nimi, CURRENT_DATE AS kuupaev FROM isikud JOIN klubid ON isikud.klubis = klubid.id WHERE klubid.nimi = klubi_nimi ORDER BY (SELECT SUM(punkte) FROM v_punktid WHERE v_punktid.mangija = isikud.id) DESC LIMIT 3 </pre> <p>column "punkte" does not exist Context: PL/pgSQL function f_klubiparimad(text) line 3 at RETURN QUERY</p>
22/04/2024, 16:22:10	Korrektne	drop function f_klubiparimad
22/04/2024, 16:22:33	Korrektne	CREATE OR REPLACE FUNCTION f_klubiparimad(klubi_nimi TEXT) RETU...

Joonis 28. Funktsiooniveaga logikanne päringulogis

Üksiku faili analüüsi korral kuvatakse vaid individuaalse analüüsi vaade ilma küljemenüüta. Rühma kokkuvõtte täisvaade on näha lisas 2 ja individuaalse analüüsi täisvaade lisas 3.

5. Tagasiside ja võimalikud edasiarendused

Analüsaatori veebirakendus tehti õppejõududele kättesaadavaks kursuse „Andmebaasid“ esimestel toimumisnädalatel. Analüsaatori funktsionaalsuste demonstreerimine ja seotud arutelu toimus kursuse juhendajatele mõeldud infokanalis ja ka juhendajate iganädalastel koosolekutel. Algses versioonis puudusid päringulogide sirvimise võimalus ning korrektsete ja vigaste päringute järjestust ei visualiseeritud, need komponendid lisati kursuse toimumise ajal, kui õppejõud väljendasid selleks vajadust.

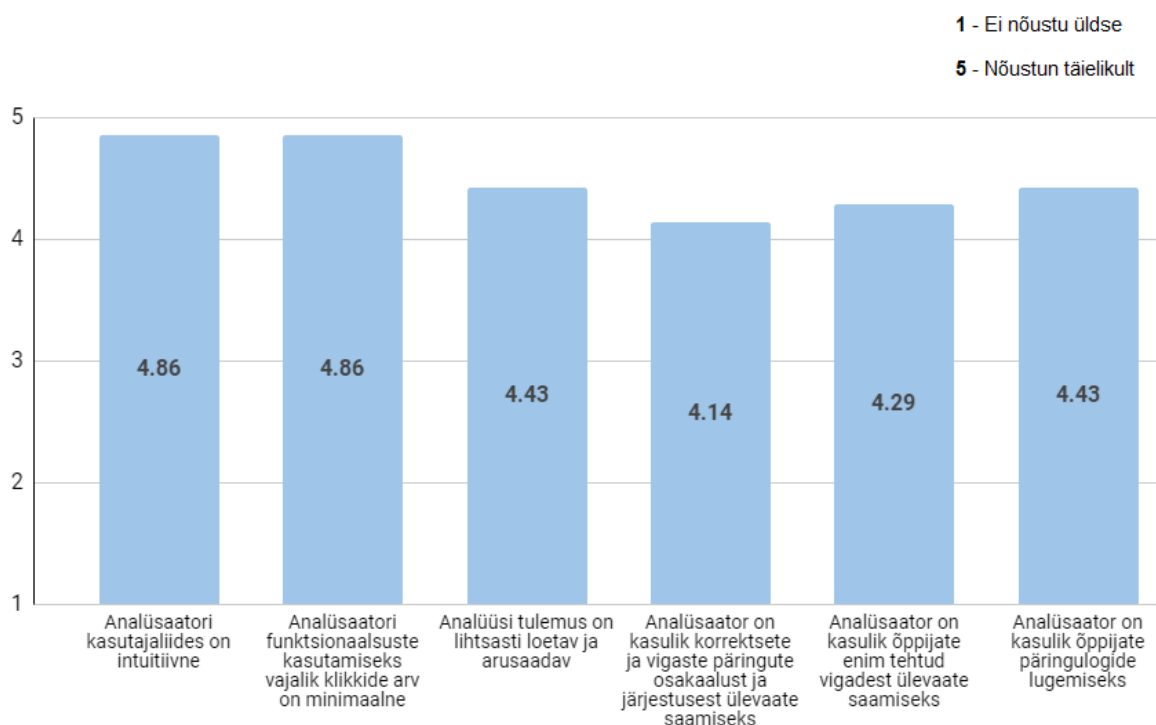
5.1 Kasutajate tagasiside

Kursuse 14 õppejõust seitse täitsid tagasisideküsimustiku, et välja selgitada, kas ja kuivõrd valminud veebirakendus täitis sellele seatud nõudeid ja algselt arutatud eesmärgid ning milliseid muudatusi ja edasiarendusi rakenduse kasutajad soovivad. Küsimustikus hinnati analüsaatori kohta esitatud väiteid Likerti 5-pallisel skaalal, kus „1“ vastab seisukohale „Ei nõustu üldse“ ja „5“ vastab seisukohale „Nõustun täielikult“. Lisaks küsiti soove muudatuste ja edasiarenduste kohta ning muid kommentaare ja ettepanekuid. Likerti skaalaga esitatud väited olid järgmised:

1. Analüsaatori kasutajaliides on intuitiivne.
2. Funktsionaalsuste kasutamiseks vajalik klikkide arv on minimaalne.
3. Analüüsi tulemus on lihtsasti loetav ja arusaadav.
4. Analüsaator on kasulik korrektsete ja vigaste päringute osakaalust ja järjestusest ülevaate saamiseks.
5. Analüsaator on kasulik õppijate enim tehtud vigadest ülevaate saamiseks.
6. Analüsaator on kasulik õppijate päringulogide lugemiseks.

Väited 1 ja 2 esitati eesmärgiga hinnata analüsaatori vastavust mittefunktsionaalsetele nõuetele 3 ja 4. Väite 3 abil saab hinnata, kas analüüsi tulemuste esitamise viisi peaks muutma või parandama. Väited 4-6 on mõeldud hindamaks analüüsi tulemuse kõigi kolme osa vastavust eesmärkidele, selguse mõttes lisati iga väite juurde kuvatõmmis vastavast analüüsi tulemuse sektsioonist. Väide 4 vastab esimesele sektsioonile analüüsi tulemuses, milleks on korrektsete ja vigaste päringute osakaalude ja järjestuse kujutamine, väide 5 vastab teisele sektsioonile, milleks on vigade analüüs, ning väide 6 vastab kolmandale

sektsoonile, mis on loodud päringulogide sirvimiseks. Kõigist antud hinnangutest võetud keskmised on kujutatud tulpdiagrammi abil joonisel 29.



Joonis 29. Veebirakenduse tagasiside küsimustiku väidete keskmised hinnangud

Jooniselt 29 on näha, et üldiselt nõustuti kõikide esitatud väidetega. Mõned vastanud kasutasid võimalust jätta vabas vormis kommentaare, kus toodi esile, et mõnel juhul ei osatud analüüsi tulemusest järeldusi teha. Näiteks on enamasti üliõpilaste tehtud trükivead ja ka mõned süntaksivead sellised, mida vahel teeb ka õppejõud, mistõttu ei pruugi täiendav tugi vajalik olla. Üldiselt kirjutati, et analüsaator täidab edukalt oma eesmärgi kiire ülevaate andmiseks tehtud vigadest ja seeläbi õpperühma suurematest murekohtadest, samuti on selline ülevaade vähendanud vajadust kodutöid kontrollides eraldi avada üliõpilaste andmebaase, et seal toimuvat mõista. Samas sooviti veel mitmeid edasiarendusi ja detailsemat analüüsi.

5.2 Võimalikud muudatused ja edasiarendused

Analüsaatorit kasutanud õppejõudude tagasiside ja juhendajate soovide põhjal on koostatud järgnev nimekiri võimalikest muutustest ja edasiarendustest veebirakendusele:

1. Plagiaadikontroll.

Näiteks on õppejõud logide lugemisel avastanud, et mõned keerulisemad päringud on tehtud väga väikeste vahedega ja kahtlustatakse, et lahendus on kopeeritud. Plagiaadikontrolli lisades aga tuleks arvestada, et akadeemilistel petturitel tekib motivatsioon oma logifaile muuta või esitada kellegi teise oma. Tõenäoliselt tuleks sellisel juhul veebirakendusele lisada andmebaas, kus hoitakse eri hetkedel analüüsitud rühmade logisid, et neid kursuseüleselt võrrelda.

2. Teatud tüüpi veateadete mitte vigadeks lugemine.

Näiteks trükivigu võib näha inimliku veana, mida teevad sageli ka oskajad ja õppejõud, mistõttu võib neid mitte pidada osaks õppeprotsessist. Lisaks saadakse mõned veateated testimise tulemusena, näiteks vead kategoorias „Ise loodud veateated“ on logides tõenäoliselt meelega saavutatud, et testida „RAISE“ käsuga defineeritud veateadete tekkimist.

3. Integratsioon lahenduste automaatkontrolliga.

Soovitakse näha lahendusprotsessi ja vigade analüüsi lahendatud ülesannete kaupa, kuid ainult PostgreSQL-i logides ülesannete info ei kajastu. Lisaks eelistatakse, et failide üleslaadimisel oleks töö minimaalne, kuid praegu esitavad üliõpilased oma kodutöö automaatkontrollis ning logid Moodle-is, kust juhendajad peavad rühma logifailid alla laadima ning analüsaatorisse panema. Pakuti välja idee, et tudengid saaksid logifaili üles laadida hoopis automaatkontrolli ja sealtkaudu võiks need automaatselt analüsaatorisse jõuda.

4. Sorteerimine ja filtreerimine.

Tagasiside küsimustikus pakuti välja idee luua filtreerimiseks salvestatavad reeglid, mida saaks rakendada suure grupi tudengite peal, et näha trende. Andmeid soovitakse ka sorteerida ja näiteks grupeerida vigu kategoorias „Muud vead“.

5. Analüsaatori lisamine UT domeeni.

Hetkel töötab veebirakendus tasuta domeenil „onrender.com“, millel on suured jõudlusepiirangud. Seetõttu võib ühe ZIP-faili analüüsimine võtta aega mitu minutit, kuigi rakenduse koodi arvutis lokaalselt jooksutades toimub analüüs paari sekundiga.

Kokkuvõte

Selle bakalaureusetöö eesmärk oli luua Tartu Ülikooli kursusele „Andmebaasid“ veebirakendus, mis analüüsib veateateid PostgreSQL-i logifailides. Valminud veebirakenduses on võimalik üles laadida nii üksik logifail kui ka kursuse Moodle'i keskkonnast saadav ZIP-fail, mis sisaldab terve õpperühma või kursuse üliõpilaste logifaile. Analüüsi tulemuses kuvatakse rühma korral kokkuvõttev vaade, mis näitab kõigi üliõpilaste tehtud päringute jaotumist korrektseteks ja vigasteks, vigade jaotumist eri kategooriatesse ning korduvalt esinenud veateateid koos esinemissagedusega. Lisaks on võimalik vaadata analüüsi iga üliõpilase kohta eraldi, mis näitavad lisaks eelnevale korrektsete ja vigaste päringute järjestust, esimese ja viimase päringu aega, korduvate veateadete juures ka vastavat päringut ning kogu logi lehekülgede kaupa loetaval kujul. Analüüsida võib kõik üles laetud failis leiduvad logid või saab valida sobiva ajavahemiku, näiteks viimane nädal või kontrolltöö aeg.

Töös tutvustati ka varasemaid uuringuid SQL-i õppimisel tehtavate vigade kohta ning anti ülevaade PostgreSQL-i päringulogidest ja olemasolevatest võimalustest neid analüüsida. Lõpuks võeti kokku analüsaatorit kasutanud õppejõudude tagasiside, mille põhjal täidab rakendus suuresti oma eesmärgi anda kiire ülevaade üliõpilaste tehtud vigadest ülesannete lahendamisel. Töös on kirjeldatud ka juhendajate soove edasiarendusteks, nagu automaatne plagiaadituvastus ja integratsioon automaatkontrolli keskkonnaga.

Analüsaatori rakenduse piiranguteks on analüüsi põhinemine ainult veateadetele, mistõttu ei tuvastata päringutes loogika- ja semantilisi vigu, ning vähene tagasiside, mis peegeldab ainult poolte kursuse õppejõudude arvamust. Analüsaator on loodud võtma sisendiks PostgreSQL-i logifaile ja ei sobi seega muude andmebaasi haldussüsteemide logide analüüsimiseks. Valminud PostgreSQL-i logide analüüsi veebirakendus loodi kursuse „Andmebaasid“ tarbeks, kuid sobib kasutamiseks ka teisel Tartu Ülikooli kursusel „Sissejuhatus andmebaasidesse“ ja gümnaasiumile mõeldud kursusel „Tarkvaraarendus“, kus õpetatakse SQL-päringute koostamist samuti PostgreSQL-i süsteemi abil [28].

Viidatud kirjandus

- [1] Tartu Ülikooli Õppeinfosüsteem. Andmebaasid (6 EAP).
<https://ois2.ut.ee/#/courses/LTAT.03.004/details> (21.04.2024)
- [2] PostgreSQL Documentation: 16: 20.8. Error Reporting and Logging.
<https://www.postgresql.org/docs/16/runtime-config-logging.html> (21.04.2024)
- [3] Miedema, D., Aivaloglou, E., Fletcher, G. Identifying SQL Misconceptions of Novices: Findings from a Think-Aloud Study. *ACM Inroads*, Vol. 13, No. 1, pp. 52–65, 2022. <https://doi.org/10.1145/3514214>
- [4] Darold, G. pgBadger - PostgreSQL log analyzer. <https://pgbadger.darold.net> (21.04.2024)
- [5] pganalyze Log Insights. <https://pganalyze.com/docs/log-insights> (21.04.2024)
- [6] Avrot, L. Lætitia AVROT / pglog - Gitlab. https://gitlab.com/l_avrot/pglog (21.04.2024).
- [7] Ahadi, A., Behbood, V., Vihavainen, A., Prior, J., Lister, R. Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In *SIGCSE '16: Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pp. 401-406, 2016.
<https://doi.org/10.1145/2839509.2844640>
- [8] Taipalus, T., Siponen, M., Vartiainen, T. Errors and Complications in SQL Query Formulation. *ACM Transactions on Computing Education*, Vol. 18, No. 3, pp. 4, 2018. <https://doi.org/10.1145/3231712>
- [9] Lloyd, J. Practical Advantages of Declarative Programming. In *GULP-PRODE'94: Joint Conference on Declarative Programming*, pp. 18-30, 1994
- [10] Taipalus, T., Perälä, P. What to Expect and What to Focus on in SQL Query Teaching. In *SIGCSE '19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pp. 198-203, 2019.
<https://doi.org/10.1145/3287324.3287359>
- [11] Brass, S., Goldberg, C. Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software*, Vol. 79, No. 5, pp. 1-2, 2006.
<https://doi.org/10.1016/j.jss.2005.06.028>

- [12] PostgreSQL: About. <https://www.postgresql.org/about/> (21.04.2024)
- [13] DBeaver Community. <https://dbeaver.io/> (21.04.2024)
- [14] Pukk, J. E. N-graafide analüsaatori loomine. TÜ arvutiteaduste instituudi bakalaureusetöö. 2021. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71857 (29.04.2024)
- [15] Kütt, R. Thonny logide analüüsi veebirakendus. TÜ arvutiteaduste instituudi bakalaureusetöö. 2021. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71885 (29.04.2024)
- [16] Meier, H., Lepp, M. Effectiveness of Feedback Based on Log File Analysis in Introductory Programming Courses. *Journal of Educational Computing Research*, Vol. 61, No. 3, pp. 696–719, 2023. <https://doi.org/10.1177/07356331221132651>
- [17] Darold, G. pgBadger - PostgreSQL log analyzer - Events. <https://pgbadger.darold.net/examples/errors.html#most-frequent-errors-events> (29.04.2024).
- [18] Log Insights: Application / User errors - pganalyze. <https://pganalyze.com/docs/log-insights/app-errors> (29.04.2024)
- [19] IEEE Standard Glossary of Software Engineering Terminology. 1990. https://www.informatik.htw-dresden.de/~hauptman/SEI/IEEE_Standard_Glossary_of_Software_Engineering_Terminology%20.pdf (30.04.2024)
- [20] Nonfunctional Requirements - Scaled Agile Framework. <https://scaledagileframework.com/nonfunctional-requirements/> (29.04.2024)
- [21] What is Java Spring Boot? <https://www.ibm.com/topics/java-spring-boot> (25.03.2024)
- [22] Introduction | Vue.js. <https://vuejs.org/guide/introduction.html> (25.03.2024)
- [23] Template Syntax | Vue.js. <https://vuejs.org/guide/essentials/template-syntax.html> (25.03.2024)
- [24] Breedis, R. JavaScripti kasutajaliidese raamistike võrdlus. TÜ arvutiteaduste instituudi bakalaureusetöö. 2021. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=72077 (14.05.2024)
- [25] Vuetify - A Vue Component Framework. <https://vuetifyjs.com/en/> (29.04.2024)

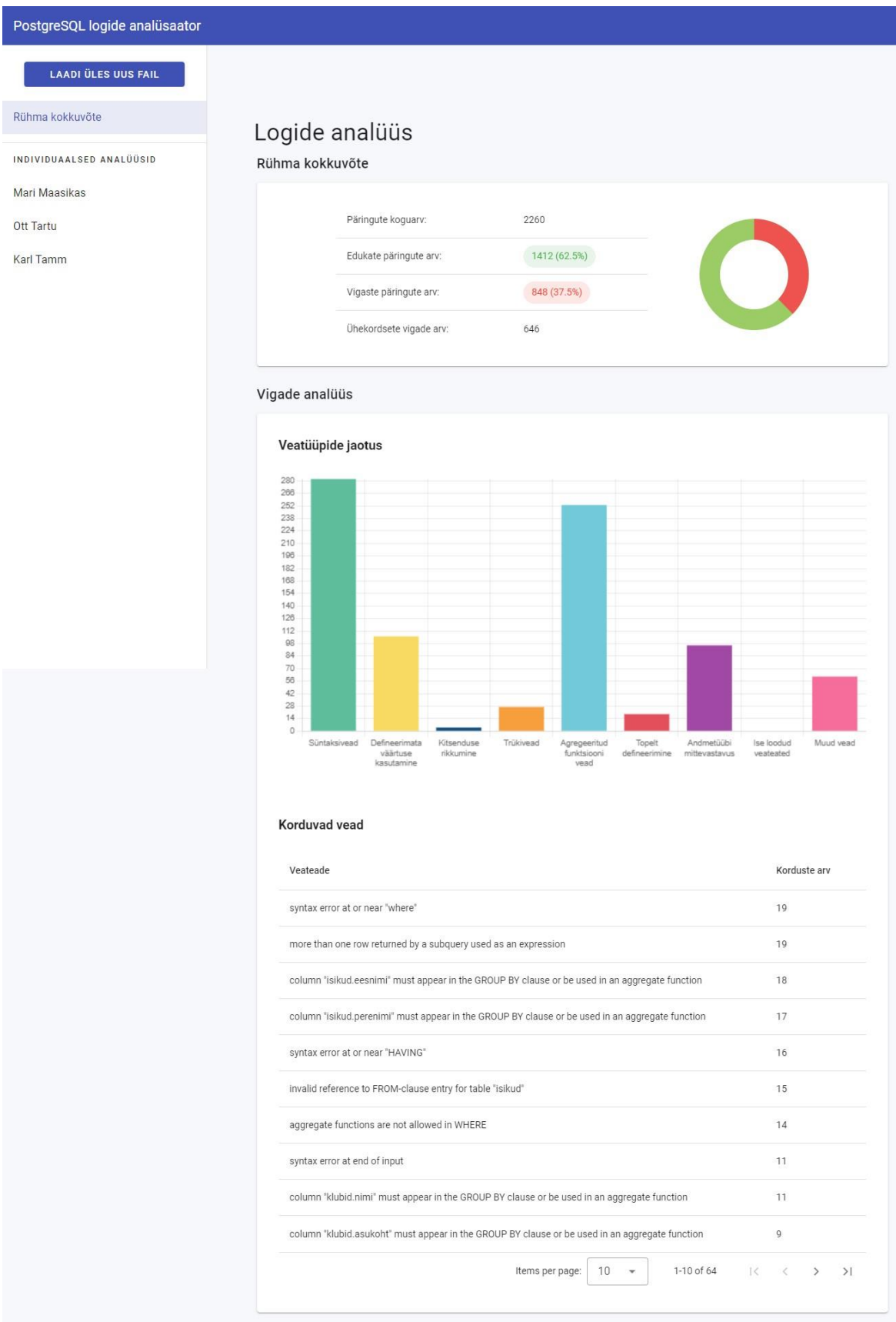
- [26] Juszczak, J. vue-chartjs. <https://vue-chartjs.org/> (29.04.2024)
- [27] PostgreSQL Documentation: 16: Appendix A. PostgreSQL Error Codes.
<https://www.postgresql.org/docs/current/errcodes-appendix.html> (05.05.2024)
- [28] Sissejuhatus – Lisamoodulid. <https://web.htk.tlu.ee/digitalu/lisamoodulid/front-matter/introduction/> (2024.05.15)

Lisad

I. PostgreSQL JSON-formaadis täielik logiobjekt

```
{
  "timestamp": "2024-02-23 16:18:54.983 EET",
  "user": "postgres",
  "dbname": "male",
  "pid": 12268,
  "remote_host": "127.0.0.1",
  "remote_port": 50272,
  "session_id": "65d8a3a7.2fec",
  "line_num": 38,
  "ps": "",
  "session_start": "2024-02-23 15:54:47 EET",
  "vxid": "5/69", "txid": 0,
  "error_severity": "ERROR",
  "state_code": "42883",
  "message": "function age(date, integer) does not exist",
  "hint": "No function matches the given name and argument types. You might need to
add explicit type casts.",
  "statement": "select eesnimi, perenimi, sugu, age(synniaeg, 1) from isikud",
  "cursor_position": 33,
  "application_name": "DBeaver 23.3.5 - SQLEditor <Script.sql>",
  "backend_type": "client backend",
  "query_id": 0
}
```

II. Rühma analüüsi kokkuvõtte vaade



III. Individuaalse analüüsi vaade

PostgreSQL logide analüsaator

LAADI ÜLES UUS FAIL

Rühma kokkuvõte

INDIVIDUAALSED ANALÜÜSID

Tudeng Üks

Tudeng Kaks

Tudeng Kolm

Tudeng Neli

Logide analüüs

Tudeng Kolm

Analüüsitud failid: postgresql-2024-04-15_220830.json

Algus:
29 Apr 2024 12:24

Lõpp:
29 Apr 2024 13:27

Päringute koguarv:

100

Edukate päringute arv:

70 (70.0%)

Vigaste päringute arv:

30 (30.0%)

Ühekordsete vigade arv:

22

Vigade analüüs

Veatüüpide jaotus

Süntaksivead

Defineerimata väärtuse kasutamine

Kitsenduse rikkumine

Trüki vead

Agregeeritud funktsiooni vead

Topelt defineerimine

Andmetüübi mittevastavus

Ise loodud vaealead

Muud vead

Korduvad vead

Veateade

Korduste arv

syntax error at or near "language"

3

operator does not exist: character varying = integer

3

column "musta_tulemus" of relation "partiid" does not exist

1

unterminated quoted string at or near "Mängija mängib juba samal ajal"; END IF; RETURN NEW; END; "

1

Items per page:

10

1-4 of 4

Kõik päringud

PIKENDA PÄRINGUD

LÜHENDA PÄRINGUD

Aeg

Vea tüüp

Päring

39

Korduvad vead

Veteade	Korduste arv	
syntax error at or near "language"	3	▼
operator does not exist: character varying = integer	3	▼
column "musta_tulemus" of relation "partiid" does not exist	1	▼
unterminated quoted string at or near "Mängija mängib juba samal ajal"; END IF; RETURN NEW; END;"	1	▼

Items per page: 10 1-4 of 4 |< < > >|

Kõik päringud

PIKENDA PÄRINGUD		LÜHENDA PÄRINGUD
Aeg	Vea tüüp	Päring
29/04/2024, 12:54:34	Korrektne	select * from turniirid t
29/04/2024, 12:54:46	Korrektne	select * from klubid k
29/04/2024, 12:55:38	Süntaks	create or replace function tg_kustuta_klubi() return trigger la... ▼
29/04/2024, 12:55:58	Süntaks	create or replace function tg_kustuta_klubi() return trigger la... ▼
29/04/2024, 12:55:58	Süntaks	create or replace function tg_kustuta_klubi() return trigger la... ▼
29/04/2024, 12:56:58	Süntaks	create or replace function tg_kustuta_klubi() return trigger la... ▼
29/04/2024, 12:57:14	Korrektne	create or replace function tg_kustuta_klubi() returns trigger l... ▼
29/04/2024, 12:57:56	Korrektne	create trigger tg_kustuta_klubi after delete on klubid for each... ▼
29/04/2024, 12:58:24	Korrektne	INSERT INTO Asulad VALUES (11, 'Rakvere')
29/04/2024, 12:58:30	Korrektne	INSERT INTO Klubid (nimi, asula) VALUES ('Klubi Kustutamiseks',... ▼
29/04/2024, 12:58:46	Korrektne	DELETE FROM klubid WHERE nimi='Klubi Kustutamiseks'
29/04/2024, 12:58:46	Funktsioon Andmetüübid	select id from asulad where nimi = old.asula ▼
29/04/2024, 12:59:10	Korrektne	INSERT INTO Asulad VALUES (11, 'Rakvere')
29/04/2024, 12:59:10	Kitsendus	INSERT INTO Asulad VALUES (11, 'Rakvere') ▼
29/04/2024, 12:59:22	Korrektne	DELETE FROM klubid WHERE nimi='Klubi Kustutamiseks'

Päringute arv leheküljel: 15 46-60 of 100 |< < > >|

IV. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Külli Kangro**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Veebirakendus päringuvigade analüüsimiseks PostgreSQL-i logide abil kursusel „Andmebaasid“,

mille juhendajad on Piret Luik ja Marina Lepp,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Külli Kangro

15.05.2024