

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Arvutiteaduse instituut

Martti Kakk

**Automaatkontrollide arendus õppetöö
tagasisidestamiseks aines Andmebaasid**

Matemaatika- ja informaatikaõpetaja õppekava

Magistritöö (15 EAP)

Juhendajad: PhD Piret Luik

MSc Vambola Leping

Tartu 2023

Automaatkontrollide arendus õppetöö tagasisidestamiseks aines Andmebaasid

Magistritöö

Martti Kakk

Lühikokkuvõte

Magistritöö eesmärgiks oli luua ainele Andmebaasid LTAT.03.004 uued automaatkontrollid PostgreSQL keeles ja koguda automaatkontrollide kasutamiskogemuse kohta tagasisidet nii üliõpilastelt kui õppejõududelt. Eesmärk sai täidetud 2023. aasta kevadsemestril, kui töö autor arendas vastavad automaatkontrollid ja need võeti praktikumijuhendajate poolt kasutusele. Läbi viidud küsitlusi analüüsides selgus, et üliõpilaste arvates annaks automaatkontrolli tagasiside arusaadavust täiendada, praktikumijuhendajate arvates oli arusaadavus piisav. Tulevikus sooviksid nii üliõpilased kui praktikumijuhendajad kasutusel näha täisautomaatseid kontrole.

CERCS teaduseriala: S270 Pedagoogika ja didaktika

Märksõnad: andmebaas, automaatkontroll, automaatne tagasiside

Developing automated tests in Databases course for giving feedback to students

Master thesis

Martti Kakk

Abstract

The aim of this thesis was to create automated tests in Databases course in PostgreSQL and to gather feedback on user experience from both, the students and lecturers. The automated tests were written during the spring semester of 2023 and were put into use already during the development. From analyzing the conducted surveys, it was discovered that from the students' point of view the feedback given by the automated tests could be better, but the lecturers thought it was sufficient. In the future both, the students and the lecturers wish to be able to use fully automated tests.

CERCS research specialisation: S270 Pedagogy and didactics

Key Words: database, automated tests, automated feedback

Sisukord

Sissejuhatus	5
1. Teoreetiline ülevaade	7
1.1. Automaatkontrollimise ajalugu	7
1.2. Andmebaaside automaatkontrollimise ajalugu	9
1.3. Andmebaaside õpetamine ja tagasiside andmine	10
1.4. Automaatkontrollidega seotud probleemid	12
2. Automaatkontrollide koostamise tööprotsess	13
2.1. Praktikumi- ja kodutööde kirjeldused ja nende hindamine	13
2.2. Automaatkontrolli funktsionaalsed ja mittefunktsionaalsed nõuded	14
2.3. Automaatkontrollide koostamine	16
2.3.1. Automaatkontrolli häälestamine	16
2.3.2. Tabelite, funktsioonide ja protseduuride loomine	17
2.3.3. Ühtlustatud vaheandmete sisestamine andmebaasi	19
2.3.4. Kontrollprotseduuride töövoog	19
2.4. Praktikumijuhendaja automaatkontrolli kasutamise töövoog	20
2.5. Probleemid automaatkontrollide arendamisel	21
3. Tagasiside kogumine loodud automaatkontrollidele	23
3.1. Valim	23
3.2. Andmete kogumine	24
3.3. Andmete töötlemine	25
3.4. Üliõpilaste küsitluste tulemused	25
3.4.1. Üldine tagasisidetabeli arusaadavus	25
3.4.2. Enda lahendustele saadud tagasiside arusaadavus	27
3.4.3. Üliõpilaste ootused automaatkontrollidele	29
3.5. Praktikumijuhendajate küsitluste tulemused	30

5. Arutelu	33
Kokkuvõte	36
Tänuavaldused	37
Kasutatud kirjandus	38
Lisad	45
Lisa 1. Esimene küsitlus tudengitele	45
Lisa 2. Teine küsitlus tudengitele	50
Lisa 3. Küsitlus praktikumijuhendajatele	55
Lisa 4. Automaatkontrollide kasutusjuhend	60
Lisa 5. Pakkfaili kasutusjuhend	66
Lisa 6. Automaakontrolli failid	76
Litsents	77

Sissejuhatus

Teadmiste kontrollimine on kõigil õppetasemetel õppeprotsessi tähtis osa, sest võimaldab hinnata õppijate arengut ja saavutusi. Õpetajate ülekoormus on suureks probleemiks ülemaailmselt ning teadmiste kontrollimisel on selles töömahus suur osa (Hunt ja Karm, 2022; Murry, 1990). Vastavalt õppemeetoditele on võimalik tuua välja lahendusi, mis toetaksid nii õpetajaid kui ka õppijaid ning võimaldaksid vähendada töömahtu, mis kulub õpitud teadmiste hindamiseks (Fonte jt, 2013; Gupta, 2012).

Informaatika on juba mõnda aega ülikooli astujate seas populaarne eriala olnud ja iga aastaga huviliste hulk aina kasvab (Muuli jt, 2021). Mida suurem on huviliste arv, seda suurem on õppejõudude töökoormus ja selle vähendamiseks saab näiteks automatiseerida kodutööde kontrollimist (Muuli jt, 2021). Kodutööde käsitsi hindamisel on keeruline hoida ühtlast kvaliteeti, sest eri õppejõud annavad tagasisidet ja hindavad lahendusi erinevalt (Romli jt, 2015). Suurest mahust ja keerulisest ülesannete ülesehitusest tingituna ei pruugi olla võimalik käsitsi kõiki kodutööde osasid kontrollida (Auffarth jt, 2008), võiks õppejõudude töökoormuse vähendamiseks olla automaatkontrollid heaks praktikaks, mis võimaldaks ühtlustada kodutööde tagasisidet ja hindamist ning tagada põhjalikuma kontrolli.

Käesolevas töös on keskendutud kasvava populaarsusega Tartu Ülikooli ainele Andmebaasid LTAT.03.004, kus töö autor juhendas 2023. aasta kevadsemestril praktikume. Aines on 12 praktikumitööd ja 6 kodutööd (Andmebaasid - Kursused - Arvutiteaduse instituut, i.a.), mida praktikumijuhendajad peavad kontrollima ning nende kõigi käsitsi kontrollimine nõuab praktikumijuhendajatelt väga palju aega.

Varem on käesoleva töö autor bakalaureusetöö raames loonud samale ainele automaatkontrollid (Kakk, 2020), kuid aine ülesehitust muudeti 2023. aastal. Aine jaoks loodi täielikult uued õppematerjalid ja andmebaaside õpetamisel kasutatav andmebaasi juhtsüsteem vahetati SQL Anywhere'ist PostgreSQL'ile, mis tingis käesoleva töö autori bakalaureusetöös arendatud automaatkontrollide kohandamise ja edasise arendamise. Muutuste analüüsi käigus selgus, et mõistlik on luua uued automaatkontrollid.

Lõputöö eesmärgiks on luua Andmebaasid LTAT.03.004 ainele uued automaatkontrollid PostgreSQL keeles ja koguda automaatkontrollide kasutamiskogemuse kohta tagasisidet nii üliõpilastelt kui õppejõududelt. Lähtudes eesmärgist sõnastati järgnevad uurimisküsimused:

- 1) Kuidas koostada automaatkontrolli PostgreSQL'i peale koostatud kodutööde kontrollimiseks?
- 2) Millised on üliõpilaste hinnangud loodud automaatkontrollidele?
- 3) Millised on praktikumijuhendajate hinnangud loodud automaatkontrollidele?

Magistritöö on jaotatud neljaks sisupeatükiks. Esimeses peatükis selgitatakse, mis on automaatkontroll, antakse ülevaade automaatkontrollide ajaloost ja arengust nii Tartu Ülikoolis kui mujal maailmas ning kirjeldatakse nii õppejõudude kui üliõpilaste ootusi automaatkontrollile. Teises peatükis kirjeldatakse magistritöö käigus loodud automaatkontrolli arendusprotsessi ja käsitletakse lühidalt automaatkontrolli loomise käigus tekkinud probleeme. Kolmandas peatükis käsitletakse kvantitatiivset uuringut, mis viidi läbi automaatkontrollile tagasiside kogumiseks. Neljandas peatükis arutletakse töö tulemuste üle.

Töös kasutatavad põhimõisted

Pakkfail (ingl batch file) sisaldab pakina töödeldavaid üksusi (Hanson ja Tavast, i.a.). Iga üksus on käsu rida, mis käivitatakse järjest pakkfaili käivitamisel (Laurie, i.a.).

SQL ehk struktureeritud päringukeel (ingl structured query language) on poolstandardne päringukeel. Päringukeel on komplekt reegleid, mille alusel konstrueeritakse päringuid andmebaasist andmete otsimiseks. (Liikane ja Kesa, i.a.)

SQL skriptifail (ingl script file) on tekstifail, mille sees on SQL laused ja see on kasulik kui soovitakse käivitada samasid SQL lauseid mitmel korral (Application Express SQL Workshop and Utilities Guide, i.a.).

1. Teoreetiline ülevaade

Teoreetilises ülevaates tuuakse välja, kuidas on aja vältel üldine automaathindamine välja kujunenud ja kuidas on andmebaaside automaatkontrollid välja kujunenud. Seejärel kirjeldatakse andmebaaside õpetamisstiile ja tagasiside tüüpe. Viimasena tuuakse välja automaatkontrollidega seotud probleemid.

Roosalu (2022: 5) on öelnud, et „automaathindajad on tarkvaralised tööriistad, millele saab ette anda mingi sisendi ning seejärel väljundina tagasi saada mingi hinne või hinnang esitatud sisendi kohta“. Kontrollide automatiseerimine on vajalik selleks, et korduvalt kontrollida keerulisi ja mahukaid programmikoode (Fonte jt, 2013; Gupta, 2012; Kakk, 2020). Veel on automaatkontrollide eelisteks kiirus ja inimlike vigade puudumine (Sepp, 2018), aga kiirus ei ole automaatkontrolli peamine eesmärk, sest oluline on meeles pidada, et üliõpilastel on rohkem kasu sisukast tagasisidest, mitte vaid saadud hindest (Ala-Mutka, 2015). Vastavalt sisendi struktuurile ja hindamiskriteeriumitele saab väljund olla see, mis parajasti vajalik on (Roosalu, 2022). Üliõpilase esitatud lahendus võib automaatkontrollist saada täiesti õige tulemuse, aga see ei tähenda, et kontrollid on perfektsed ja neid ei saa edasi arendada või täiendada (Muuli jt, 2021).

1.1. Automaatkontrollimise ajalugu

Automaatkontrollimise tausta väljatoomine on oluline teadmaks, mida on varasemalt tehtud, ja seeläbi on võimalik nende tulemustest õppida. Ajaloo tundmine võimaldab varasemate automaatkontrollide vigadest õppida. Samuti on võimalik nende abil enda automaatkontrollide lahendust täiendada.

Kõige esimene teadaolev automaatkontrollimise süsteem loodi aastal 1960 ja seda kasutati üliõpilaste programmide peal, mis olid kirjutatud assemblerkeeles ning automaathindamise tulemuseks oli „*wrong answer*“ või „*programm complete*“ (Hollingsworth, 1960). Järgmised automaathindamise süsteemid loodi programmeerimiskeelele ALGOL (Forsythe ja Wirth, 1965; Naur, 1964). Üliõpilase töö kontrollimiseks anti sisendiks mitmeid andmestikke ja võrreldi üliõpilase vastuseid õigete vastustega ning anti tulemusele vastav tagasiside ja hinnang (Forsythe ja Wirth, 1965). Aastal 1969 arendati automaathindamist edasi ja leiti lahendus, kuidas saaks kontrollida kolmes erinevas programmeerimiskeeles olevaid programme: Minigol, Algol ja KDF 9 Assembly Code (Hext ja Winings, 1969).

Järgmine suur muutus automaatkontrollimise süsteemides toimus, kui hakati kasutama käsurida ja kasutajaliideseid (Douce jt, 2005). Üks selline tööriist loodi aastal 1989 õppejõududele, kus kontrolliti, kas programm töötab õigesti ja mõistlikult (Isaacson ja Scott, 1989). Nende automaatkontroll oli skriptipõhine, võttis skripti abiga kaustadest esitatud ülesanded, lasi kontrollist läbi ja kirjutas tulemused faili, mida õppejõud sai vaadata (Isaacson ja Scott, 1989). Samal aastal loodi aga teistmoodi töötav automaatkontroll, mida said üliõpilased ise kasutada, et kontrollida oma programmi korrektsust (Reek, 1989). Kontroll töötas põhimõttel, et üliõpilase programmi tulemust võrreldi oodatud tulemusega ja anti hinnang (Reek, 1989). Lisaks oli üliõpilastel piirang, kui mitu korda said nad oma kodutööd esitada, mis sundis üliõpilast oma vigu analüüsima, mitte lihtsalt katsetama (Reek, 1989).

Mida võimekamaks muutus tehnoloogia, seda rohkem hakkas ka tulema välja mitmeid erinevaid automaathindamise süsteeme nagu The Cassandra Project (von Matt, 1994), ja ASSYST (Jackson ja Usher, 1997). Veel tehti BOSS süsteem (Joy ja Luck, 1998) ja Ceilidh (Higgins jt, 2003) süsteem, mis põhinesid vastavalt C ja Java keeltele. Java ja C keeltele oli tehtud veel mitmeid automaatkontrolli süsteeme, millest mitmed arenesid edasi veebipõhisteks automaatkontrollideks (Douce jt, 2005). Edasi arenesid automaatkontrollid nagu BOSS ja Ceilidh (uue nimega CourseMaker) veebiliidestega kontrollideks, mis andsid õppejõududele mitmeid võimalusi juurde (Douce jt, 2005). Õppejõud said juhendada mitut üliõpilaste rühma, luua ülesandeid mugavalt juurde ja hinnata üliõpilaste töid kiiremini (Douce jt, 2005).

Tartu Ülikoolis alustati teadaolevalt esimeste automaatkontrollide kasutamist aastal 2014, mil arendati automaatkontrollid ainele „Algoritmid ja andmestruktuurid“ (Plangi, 2014). Käesoleval aastal kasutatakse automaatkontrolle väga mitmes programmeerimisaines, millest suurimad on „Programmeerimine“ ja „Programmeerimine II“, „Automaadid, keeled ja translaatorid“ ja „Objektorienteeritud programmeerimine“ (Automaadid, keeled ja translaatorid - Kursused - Arvutiteaduse instituut, i.a.; Programmeerimine - Kursused - Arvutiteaduse instituut, i.a.). Need ained kasutavad erinevaid veebiliidestele ehitatud automaatkontrolle nagu Moodle VPL (Virtual Programming Lab) keskkonda ja Lahendus (Programmeerimine - Kursused - Arvutiteaduse instituut, i.a.). Aastal 2022. sügissemestril kasutas Lahendus keskkonda 15 erinevat kooli ja loodi 13 uut programmeerimise kursust, millel osales kokku umbes 500 õpilast ja 15 õpetajat. Lahenduse keskkonna kasutajate hulgas

on üle 150 õpetaja ja 4000 õpilase ning loodud on üle 100 kursuse ja 150 erineva ülesande, millele õpilased on esitanud kokku umbes 270 000 lahendust.¹

1.2. Andmebaaside automaatkontrollimise ajalugu

Andmebaasid on keerukama ülesehitusega ja võivad hõlmata mitmetasandilist ja kompleksset süsteemi. Nende jaoks automaatkontrollide loomiseni jõuti alles sajandivahetusel. Aastal 1998 arendati Canterbury ülikoolis välja vahend SQL-Tutor, millega anti üliõpilaste päringuülesannete lahendustele tagasiside, aga ei pandud hindeid (Mitrovic, 1998). SQL-Tutor oli loodud andmebaasi haldusvahendile Ingres ja tööriista eesmärk oli anda naturaalses keeles tagasiside, et tudeng saaks aru, mis on tema päringuga valesti. (Mitrovic, 1998). Aastal 2004 arendati Sydney Tehnikaülikoolis välja tööriist AsseSQL (Prior ja Lister, 2004) ja samal aastal töötati välja ka SQLator tööriist Queenslandi Ülikoolis, mis töötas samal põhimõttel nagu AsseSQL (Dekeyser jt, 2007; Sadiq jt, 2004). AsseSQL ja SQLator kontrollisid üliõpilaste lahendusi heuristiliste meetodite abil, mille hulka kuulus kõigi üliõpilaste lahendatud päringulausete käivitamine juba valmis tehtud andmebaasi vastu ja kontrollimine, kas tulemus on õige või vale (Dekeyser jt, 2007; Prior ja Lister, 2004).

Aastal 2007 arendati välja automaatkontroll SQLify, mille eesmärk oli luua e-keskkond, mis kataks varasemate automaathindamiste süsteemide puudujääke ja täiendaks olemasolevaid (Dekeyser jt, 2007). SQLify annaks korralikku tagasiside ja näitaks skeemide abil, kuidas üliõpilase päringulahendus töötab, kaasaks üliõpilase töö hindamiseks teiste üliõpilaste abi ja ühtlustaks hindamist (Dekeyser jt, 2007). Aastal 2010 loodi e-keskkond Exploratorium, mis koosnes kolmest tööriistast, et õppida SQL päringutega seotud teemasid (Brusilovsky jt, 2010).

Nagu ka varasemate tööriistadega AsseSQL ja SQL-Tutor, võrreldi mitme uuema e-õppe keskkonna (näiteks SQL-ACME ja DB-Learn) puhul üliõpilase päringulahenduse tulemust õige tulemusega – kui üliõpilase tulemus oli õige, sai punkti ja kui oli juba osaliselt vale, sai vea (Nalintippayawong jt, 2017; Prados jt, 2006; Prior ja Lister, 2004; Sadiq jt, 2004). Järgmisena arendati aSQLg, mis lubas mitmesammulist hindamist, kus viimase sammuna sai ka õppejõud hinde panna (Kleiner jt, 2013).

Aastal 2015 arendati XDa-Ta süsteem, mis kasutas päringute kontrollimiseks mitut erinevat andmemudelit ning püüdis nende abil kinni tüüpilisemad vead, mis võisid päringute

¹Juhendajalt saadud andmed

kirjutamisel tekkida (Bhangdiya jt, 2015). XDa-Ta kontrollmudeleid arendati jõudsalt edasi, mis kindlustas süsteemi ja suutis tuvastada isegi poolõigeid päringuid ning neid vastavalt hinnata (Chandra jt, 2016). Aastal 2018. tuli välja e-keskkonnale loodud SQL tester, mis oli automaatkontroll, mille eesmärk oli luua harjutustestid, mida üliõpilane sai korduvalt proovida ning kohese tagasiside abil kiirelt ülesannet edasi lahendada (Kleerekoper ja Schofield, 2018).

Suurem osa ajast on arendatud automaatkontrolle SQL päringute kontrollimiseks ja vähemal määral on tehtud arendusi andmebaasi objektide kontrollimiseks. Üks sellistest arendustest tehti Uus-Meremaal Otago Ülikoolis 2018. aastal, milleks oli poolautomaatne süsteem, et õppejõud saaksid kontrollida üliõpilaste loodud SQL skeeme ühiktestide abil (Stanger, 2018). Automaatkontroll kontrollib, kas loodud skeem vastab nõuetele ja annab vastava tagasiside (Stanger, 2018). Automaatkontrolle oli võimalik kasutada nii õppejõududel hindamiseks kui ka üliõpilastel, kes said kontrollida, kas skeem vastab ülesandes seatud tingimustele (Stanger, 2018). Nende loodud automaatkontroll vaatab, et üliõpilaste loodud andmebaasi struktuur (näiteks tabelite ja veergude nimed) oleksid korrektsed, seejärel kontrollitakse terviklikkust mitme erineva õige ja vale sisenditega (Stanger, 2018).

Esimene automaatkontrollide arendus üliõpilaste kodutööde kontrollimiseks Tartu Ülikooli andmebaaside aines tehti aastal 2015 kasutades Javat ja SQL skripti (Siiber, 2015). Kahjuks jäi arendus poolikuks ja seda ei võetud laialdaselt kasutusele (Kakk, 2020). Aastal 2018 arendati ainele Andmebaasid „Edu“ andmebaasi automaatkontrolle ainult Java keeles (Sepp, 2018). Aastal 2019 hakati looma automaatkontrolle andmebaaside ainele „Sissejuhatus andmebaasidesse“ (Õunmaa, 2019). Kahjuks jäid ka Sepa ja Õunmaa automaatkontrollid poolikuks ning neid ei võetud kasutusele (Kakk, 2020). Aastal 2020 arendas käesoleva magistritöö autor automaatkontrollid ainele Andmebaasid, mis võeti kasutusele mitmes andmebaase õpetavas aines.

1.3. Andmebaaside õpetamine ja tagasiside andmine

Tavaline õpetamisstiil, mis hõlmab endas klassiruumis istumist, tahvlit ja tunniülesandeid, ei ole kuhugi kadunud. Proovitud on kasutada erinevaid õpetamisviise, et üliõpilastel oleks huvitavam ja kaoks ära tunne nagu õpitaks ainet (Barata jt, 2013; Busch, 2014; Venter, 2020). Näiteks Mehhiko Ülikoolis viidi läbi katse, kus võrreldi kaht gruppi, kus üks grupp õppis tavapärasel viisil ja teisel grupil olid mängulised võistlused punktide peale. Läbi viidud katses toimus SQL päringute tagasisidestamine skripti abil, mis kontrollis, kas päring on olemas, kas

päringu süntaksinõuded on täidetud, kas päring annab tulemuse, kas päringu tulemus on õige ning kas päringu tulemusel olevad veerud on õiged ja õiges järjekorras. Katse tulemusena nähti, et mängulises grupis oli üliõpilaste motivatsioon palju kõrgem, kogeti rohkem emotsioone, näiteks tunti end rohkem keskendununa, lõbusalt, suurendati pädevustunnet ja kogeti väljakutseid. (Morales-Trujillo ja García-Mireles, 2021)

Tagasiside on terves õppeprotsessis väga oluline (Hunt ja Karm, 2022). Nii õppejõud kui üliõpilased on tajunud tagasisidet kui teavet tehtud töö eest, mis kajastab hinnangut ja annab soovitusi edaspidisteks muudatusteks (Hunt ja Karm, 2022). Ainetes, kus on palju osalejaid, on oluline, et tagasiside andmist saaks automatiseerida, sest see vähendab õppejõu töökoormust (Muuli jt, 2021). Üliõpilased on väljendanud arusaama, et tõhus tagasiside on tasakaalukas ehk sisaldab nii positiivseid aspekte kui ka kriitikat tehtud töö kohta, kuid antud tagasiside peab olema põhjendatud (Hunt ja Karm, 2022). Mõned üliõpilased on välja toonud, et tagasiside võiks olla ainult kriitika, sest positiivsed aspektid on teada ja ainult kriitika arendab edasi (Hunt ja Karm, 2022). Üliõpilaste arvates on tõhus tagasiside aus, põhjendatud (ka maksimumpunktide korral), pigem pikem ja põhjalikum kui lühike ja üldine ning rõhutati, et tõhus tagasiside on kohene, see tähendab, et saabub võimalikult kiirelt pärast töö esitamist (Dawson jt, 2019; Hunt ja Karm, 2022; Li ja De Luca, 2014).

Enamik automaatkontrolle kontrollivad üliõpilase lahenduse puhul ainult tulemuse õigsust (Kakk, 2020; Nalintipayawong jt, 2017; Prados jt, 2006; Prior ja Lister, 2004; Sadiq jt, 2004), aga ei hinnata probleemi lahendamise protsessi, mis on oluline (Muuli jt, 2021). Mõnesid automaatkontrolle luues on üritatud olukorda parandada kaasates keelemudeleid, mis ei hinda ainult tulemust, vaid ka SQL päringute sisu (Bhangdiya jt, 2015; Chandra jt, 2016). Kontrollimise teeb veel keerulisemaks olukord nendes ainetes, kus hilisemad kodutööd põhinevad varasematel lahendustel (Muuli jt, 2021).

SQLify hindamissüsteemides kasutati automaatkontrollide tagasisidele lisaks ka kaasüliõpilaste hinnanguid, mis andsid üliõpilastele laiemat ja arusaadavamat tagasisidet (Dekeyser jt, 2007). Automaatkontrollide tagasiside võiks olla võimalikult detailne, et tudengid saaksid lihtsamalt aru, kus viga tekkis, sest tihti ei saa tudengid isegi aru, miks automaatkontroll ebaõnnestus ja mis valesti on (Muuli jt, 2021). Muuli jt (2021) uurimuses jõuti järeldusele, et hindamine peaks olema nii automaatne kui ka isiklikum tagasiside õppejõudude poolt, sest automaatkontrollid annavad tagasiside enamikule iseseisvatele ülesannetele ja õppejõud saavad tagasisidestada suuremaid töid.

1.4. Automaatkontrollidega seotud probleemid

Automaatkontrollid ei ole täiuslikud ning kuigi õppejõud nõustuvad, et automaatkontroll säästab aega, siis mitmed õppejõud umbusaldavad automaatkontrolle ja teevad vahel ise pistelisi kontrolle (Muuli jt, 2021). Seepärast käsitletaksegi selles alapeatükis, millised on automaatkontrollidega seotud probleemid.

Automaatkontrollide ohukohaks võib olla, et tudengid võivad hakata keskenduma ainult sellele, et kontrollidest läbi saada ja teadmiste saamine võib jääda pinnapealseks (Muuli jt, 2021). Õppimine saab tähenduse, kui õpitakse uut informatsiooni olemasolevaga siduma (Lujan ja DiCarlo, 2006). Ka liigsete vihjete andmine võib õppimise kasutuks teha, sest tudeng parandab lihtsalt antud näite põhjal vea ning ei õpi midagi (Muuli jt, 2021). Selle probleemi lahenduseks on pakutud, et üliõpilastel oleks esitamispiirang, sest see sunniks pingutama, et teemat selgeks saada (Muuli jt, 2021). Mõned intervjueeritud õppejõud tahtsid, et üliõpilastel oleks ainult üks esitamiskatse, aga enamik soovisid ikkagi mitut katset kodutöö kontrollimiseks (Muuli jt, 2021). Teine lahendus oleks see, et automaatkontrollid on ehitatud selliselt, et üliõpilased peavad olema nutikad ja oma lahendusi enne esitamist põhjalikult katsetama, mitte toetuma ainult automaatkontrollide tagasisidele (Chen, 2004; Edwards, 2003).

Muuli jt (2021) tehtud töös toodi välja, et automaatkontrollide loomine ja korrashoidmine on ajakulukas ja keeruline ning vähesed Arvutiteaduste instituudis on selleks võimelised. Tihti on ainult arendajad võimelised automaatkontrolle korras hoidma (Keuning jt, 2016). Hetkel on süsteem habras ja halvasti dokumenteeritud, mitmete jaoks on oluline luua keskkond või lihtsustada uute automaatkontrollide loomist (Muuli jt, 2021). Seda on ka mujal täheldatud, aga kahjuks on kõik tööriistad instituudi sisesed ja puudub ühtne standard ja koostöö (Deeva jt, 2021; Muuli jt, 2021).

2. Automaatkontrollide koostamise tööprotsess

Võrreldes 2020. aastaga on aines Andmebaasid toimunud rohkelt muudatusi. Bakalaureusetöö automaatkontrollid koostati 2020. aastal andmebaasi mootorile SQL Anywhere, aga 2023. aastast mindi üle andmebaasi mootorile PostgreSQL. Seepärast koostati uued automaatkontrollid PostgreSQL keeles ([Lisa 6](#)). Aastal 2020. kasutati Andmebaaside aines haldusvahendina SAP SQL Centralit, aga 2023. aastal võeti kasutusele Dbeaveri haldusvahend.

PostgreSQL keelele üleminek ja aine ümberehitus tõid kaasa automaatkontrollide täieliku ümberkirjutamise ja tagasiside- ning hindamissüsteemi muutumise. Varasemaga võrreldes muutus ka see, et enam ei kontrollita nii palju tudengi andmebaasi esialgset loomisprotsessi, vaid pigem üliõpilase oskusi, kuidas andmebaasis muudatusi teha. Kuigi automaatkontroll hindab üliõpilase tööd punktidega, siis automaatkontrolli esmamõte on anda üliõpilasele tagasisidet selle kohta, millises seisus on üliõpilase andmebaas vastavalt käesolevatele nõutele.

Automaatkontrolli loomise protsess algas iga praktikumi- ja kodutöö läbilahendamisega. Seejärel tehti analüüs ja leiti igas ülesandes põhivead, mis võivad tekkida. Leitud vigade abil koostati tagasiside, mis on võimalikult täpne ja arusaadav üliõpilase jaoks.

2.1. Praktikumi- ja kodutööde kirjeldused ja nende hindamine

Andmebaaside aines oli 2023. aastal oli 12 praktikumi (12 praktikumitööd ja 6 iseseisvat kodutööd) (Andmebaasid - Kursused - Arvutiteaduse instituut, i.a.). Esimesed kaks praktikumi ja kaks kodutööd olid päringute teemal, mida käesoleva töö raames loodud automaatkontrollid ei kontrolli. Kolm praktikumi olid teoreetilistel teemadel, mis tähendab, et ka neile ei olnud vaja automaatkontrolle luua. Viimase praktikumi ülesanded on küll andmebaasiga seotud, aga sellise iseloomuga, et neid ei olnud vajalik ega mõistlik automaatselt kontrollida. Ülejäänud praktikumi- ja kodutööde kontrollimiseks kasutati automaatkontrolle. Seitsme praktikumi tööd ja kõik kodutööd tehakse andmebaasile Male, aga ühes praktikumis löid üliõpilased uue andmebaasi nimega Edu. Viimase praktikumi ülesanded on küll andmebaasiga seotud, aga sellise iseloomuga, et neid ei olnud vajalik ega mõistlik automaatselt kontrollida.

Praktikumide ja kodutööde punktisüsteem (tabel 1) oli kirjas aine ülesehituses. Praktikumide hindamisel ei olnud oluline, kui mitu viga on tudeng lahendamisel teinud, vaid oluline oli praktikumi töölehe ülesannete lahendamine, seega said kõik ülesande lahendajad alati ühe punkti, kuid neile anti tagasisidet sooritatud ülesannete, tekkinud vigade ja sooritamata

ülesannete kohta. Kodutööde puhul oli seevastu oluline hinnata üliõpilase eksimusi lahendamisel. Kodutööde eest võis maksimaalselt saada kaks punkti ja lisaks ka tagasiside lahendusele.

Tabel 1. Automaatselt kontrollitavate praktikumide ja kodutööde kirjeldused (Andmebaasid - Kursused - Arvutiteaduse instituut, i.a.)

Sisu	Hindamine
Veergude lisamine ja muutmine, kitsenduste lisamine ja kustutamine, andmete lisamine ja muutmine	Töölehe esitamine
Tabelite loomine, andmete lisamine ja välisvõtmete loomine	Töölehe esitamine
Tabeli loomine, andmete lisamine, veergude lisamine ja muutmine ning välisvõtme lisamine Lisaks on kaks päringu ülesannet	Ülesanded 1-5 annavad igaüks 0,2 punkti Ülesanded 6-7 annavad igaüks 0,5 punkti
Kolme vaate ja ühe materiaalse vaate loomine	Ülesanded 1-4 annavad igaüks 0,5 punkti
Vaadete ja materiaalsete vaadete loomine	Töölehe esitamine
Uue andmebaasi „Edu“ loomine, tabelite loomine, andmete sisestamine ja vaadete loomine	Töölehe esitamine
Kolme funktsiooni ja ühe protseduuri loomine	Ülesanded 1-4 annavad igaüks 0,5 punkti
Funktsioonide ja protseduuride loomine	Töölehe esitamine
Indeksite ja trigerite loomine	Ülesanded 1 ja 2 annavad igaüks 0,2 punkti ning ülesanded 3 ja 4 annavad igaüks 0,8 punkti
Trigerite loomine	Töölehe esitamine

2.2. Automaatkontrolli funktsionaalsed ja mittefunktsionaalsed nõuded

Iga süsteem luuakse eesmärgiga pakkuda kasutajale mõnda funktsiooni – süsteemi käitumist või tegevust, mille abil kasutaja saavutab oma soovitud tulemuse (Vorteil ja Laanpere, i.a.). Selleks, et kasutajate ootuseid süsteemi käitumise ja tegevuste kohta lihtsalt ja kiirelt kontrollida on olemas süsteemi funktsionaalsed nõuded (Vorteil ja Laanpere, i.a.). Suurem osa

funktsionaalseid nõudeid tulid varasematest automaatkontrollidest (Kakk, 2020), aga arenduse käigus praktikumijuhendajate soovil lisandus neile veel kaks (tabel 2).

Tabel 2. Automaatkontrollide funktsionaalsed nõuded

Varasematest automaatkontrollidest	Kujunesid välja arenduse käigus
1) Automaatkontrollimist saab häälestada	7) On võimalik määrata kontrollimise seadeid
2) Automaatkontrolli tagasiside peab määrama vea asukoha ja olemuse	8) On võimalik määrata eraldajat tulemusefailis
3) Automaatkontroll peab andma hinnangu	
4) Automaatkontroll annab teada kontrollitava üliõpilase nime	
5) Automaatkontroll väljastab tulemuse csv failina	
6) Automaatkontroll kontrollib andmebaasi oleku vastavust hetkenõuetega	

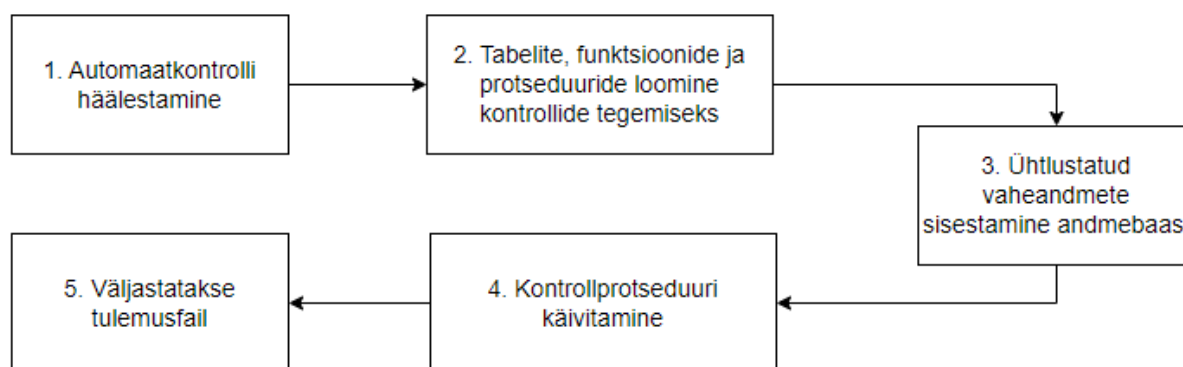
Mittefunktsionaalsed nõuded on süsteemile kehtestatud piirangud või nõuded, mis määravad tarkvara kvaliteediattribuudi (Mis on mittefunktsionaalsed nõuded, i.a.). Mittefunktsionaalsed nõuded käsitlevad selliseid probleeme nagu mastaapsus, hooldatavus, jõudlus, mobiilsus, turvalisus, töökindlus ja palju muud (Mis on mittefunktsionaalsed nõuded, i.a.). Mittefunktsionaalsed nõuded käsitlevad tarkvarasüsteemide olulisi kvaliteediprobleeme (Mis on mittefunktsionaalsed nõuded, i.a.). Käesolevas töös arendatud automaatkontrollides võeti kasutusele järgnevad mittefunktsionaalsed nõuded:

- 1) Automaatkontroll ei tööta kauem kui 10 sekundit
- 2) Automaatkontroll ei tohi lõpetada tööd veateatega vaid tagasisidega üliõpilasele
- 3) Kontrollitav andmebaas ei saa mõjutada kontrollimiseks vajalike tegevuste läbiviimist
- 4) Automaatkontrolli käivitamine on võimalikult kiire
- 5) Automaatkontrolli esialgne seadistus on lihtne

2.3. Automaatkontrollide koostamine

Automaatkontroll, mis luuakse andmebaasi kodutöö kontrollimiseks on oma sisult SQL skript, mis kirjutati PostgreSQL keeles. PostgreSQL'is on eripära, mis võimaldab kirjutada protseduure ja funktsioone protseduuriliselt. Kõik käesolevas töös loodud automaatkontrollide protseduurid või funktsioonid on protseduurilised ning need kõik on ühe protseduuri sees.

Automaatkontrollid koosnevad viiest suuremast osast: automaatkontrolli häälestamine, tabelite, funktsioonide ja kontrollprotseduuride loomine, ühtlustatud vaheandmete sisestamine andmebaasi, kontrollide käivitamine ning tulemusfaili väljastamine. Joonisel 2 on kujutatud automaatkontrollide üldist töövoogu. Alapeatükkides kirjeldatakse iga sammu lähemalt.



Joonis 1. Automaatkontrolli üldine töövoog

2.3.1. Automaatkontrolli häälestamine

Kõik automaatkontrollide protseduurid ja funktsioonid olid protseduuri nimega „kontroll“ sees, mille loomine oli oluline, et saaks häälestada automaatkontrolli (joonis 2 samm 1). Selline häälestamine võimaldas täita funktsionaalseid nõudeid 1, 7 ja 8 (tabel 2). Automaatkontrolli häälestamisel pandi paika, mis ülesandeid kontrollitakse, kuhu väljastati tulemusfail ja millisel kujul väljastati andmed. Ülesannete kontrollimise määramiseks kasutati tunnust „versioon“, mis oli erinev iga praktikumi- ja kodutöö kontrollimiseks. Teine tunnus oli „folder_path“, mille väärtuseks oli kasutaja arvutis failiteekond, kuhu salvestati tulemusfail. Tunnused „txt_lugemis_andmed_delimiter“ ja „csv_lugemis_andmed_delimiter“ määrasid tulemusfailis tulemuste eraldaja.

2.3.2. Tabelite, funktsioonide ja protseduuride loomine

Automaatkontrollide kasutamiseks olid olulised abitabelid. Tabeleid kasutati kontrollide tagasiside salvestamiseks ja kontrollides kasutatavate muutujate hoidmiseks. Tagasiside abitabeliks oli tabel nimega „Staatustabel“. Staatustabel näitas peale kontrollide lõppu, millises olukorras oli üliõpilase esitatud andmebaas vastavalt kontrollitava versiooni ja kõigi sellele eelnenud versioonide nõuetele. Tabelis 3 kirjeldatakse staatustabeli seitset veergu ja nende tähendusi. Veerud „Ylesanne“ ja „Kontrolli_nimi“ kirjeldasid vea asukohta ja vea olemust näitas veerg „Tagasiside“, mis vastasid 2. funktsionaalsele nõudele (Automaatkontrolli tagasiside peab määrama vea asukoha ja olemuse). Veerud „Punktid“ ja „Max_punktid“ vastasid 3. funktsionaalsele nõudele (Automaatkontroll peab andma hinnangu).

Tabel 3. Staatustabeli kirjeldus

Veeru nimi	Veeru selgitus
Ylesanne	Millise praktikumi- või kodutöös antud kontroll toimus. Näiteks „Praktikum 3“, „Kodutöö 4“.
Kontrolli_nimi	Kontrolli nimetus. Näiteks „Tabel "Isikud" Veergude arv“, „Vaate "v_partiidpisi" partii 10 valge punktid“.
Tagasiside	Veerg, kus oli kirjas, kontrollitulemus. Näiteks „on olemas“, „on valed, peab olema 1.0“
Olek	Tagasiside liigitamine: OK, VIGA ja Hindepunkt
Punktid	Punktid, mida saadi ühe kontrolli tulemusel.
Max_punktid	Maksimum punktid, mida oli võimalik saada ühe kontrolli õigel tulemusel.
Jr	Määras ära, mis järjekorras tulemused väljastati tulemusfaili.

Teiseks abitabeliks oli tabel „Muutujad“, mida kasutati kontrollides antavate punktide või järjekorra väärtuste hoidmiseks. Tabelis „Muutujad“ oli kolm veergu (tabel 4). Veerg „Taisarv“ oli kasutuses muutujatel, mis määrasid tulemuste järjekorra või mis oli maksimaalne õigete kontrollide arv vastavas versioonis. Veerg „Komaarv“ oli kasutuses muutujatel, mis määrasid punktide arvu, mida kontroll õigel tulemusel andis või valel tulemusel kaotas

Tabel 4. Muutujate tabeli kirjeldus

Veeru nimi	Veeru selgitus
Nimi	Muutuja nimetus, mille nimetus vihjas kasutusale, näiteks praktikum_3_jr, kodutoo_punktid.
Taisarv	Täidetud juhul, kui veeru väärtuseks pidi olema täisarv, näiteks järjekorra numbrid.
Komaarv	Täidetud juhul, kui väärtuseks pidi olema komaga arv, näiteks ülesannete punktid.

Kontrollprotseduuride abistamiseks oolid loodud abiprotseduurid ja abifunktsioonid. Abiprotseduure kasutati andmete sisestamiseks ja lõpp-punktide arvutamiseks. Abifunktsioone kasutati eelkontrollide tegemiseks.

Praktikumide automaatkontrollide arv oli igas praktikumis erinev ning olenes sellest, kui palju muudatusi oli vaja andmebaasi objektidega teha ja kui palju oli ülesandeid praktikumi töölehtedel. Kodutööde automaatkontrollide arv olenes analüüsi käigus leitud võimalike vigade või milliseid õppe-eesmärke sooviti antud ülesannetega saavutada. Järgnevas tabelis (tabel 5) on kõigi praktikumi- ja koduste tööde kontrollide arvud.

Tabel 5. Kontrollide arv teemade kaupa

Sisu	Kontrollprotseduuride arv
Muudatused veergude, kitsenduste ja andmetega	15
Tabeli loomine	8
Muudatused tabelis	7
Vaadete loomine	23
Vaadete harjutamine	64
Edu andmebaasi loomine	31
Funktsioonide loomine	8
Funktsioonide ja protseduuride loomine	30
Indeksite ja trigerite loomine	10
Trigerite loomine	21

2.3.3. Ühtlustatud vaheandmete sisestamine andmebaasi

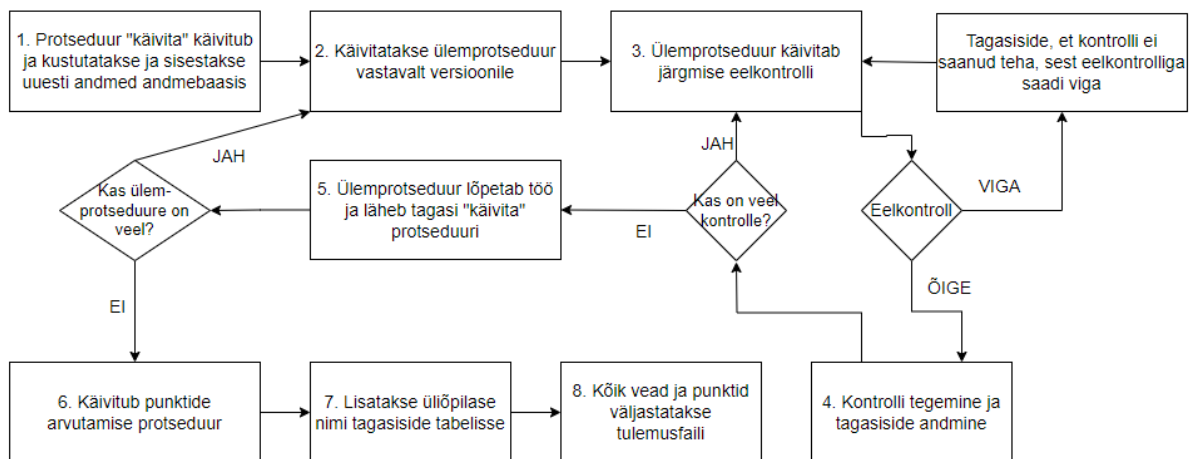
Kuni neljanda teemani ei tehtud andmete ühtlustamist (joonis 2 samm 3), sest nendes praktikumides õppisid üliõpilased, kuidas andmebaasi uusi andmeid lisada või neid kustutada ning oluline oli kontrollida, kas nad said sellega õigesti hakkama. Kolmandas teemas oli viimane andmete muutmise töö ja alates järgmisest oli oluline, et kõigi üliõpilaste andmete seis andmebaasis oleks kontrollimise hetkel ühesugune. Seepärast kustutati alates neljandast teemast üliõpilase andmebaasist kõik andmed ja sisestati uuesti ühtlustatud vaheandmed. Ideaaljuhul oleks üliõpilasel lahendust esitades andmebaasis täpselt samasugune andmete seis nagu oli sisestatavates vaheandmetes.

2.3.4. Kontrollprotseduuride töövoog

Kontrollprotseduuri töövoogu on kujutatud joonisel 3. Kontrollimise alustamiseks oli loodud protseduur „käivita“ (samm 1), mille sees käivitati vastavalt versioonile automaatkontrollid. Alati kontrolliti ka eelnevaid versioone ehk eelnevaid teemasid. Kuuendat teemat kontrolliti ainsana eraldi (ilma eelnevaid versioone kontrollimata), sest selle teema ülesanded olid andmebaasis „Edu“, mitte andmebaasis „Male“ nagu teiste teemade puhul.

Järgmisena kutsuti välja ülemprotseduur vastavalt versiooni numbrile (samm 2) ja alustati kontrollringiga, mis koosnes eelkontrollist (samm 3) ja kontrollimisest (samm 4). Kui kontrollid olid kõik tehtud, lõpetas ülemprotseduur töö (samm 5) ning töövoos mindi tagasi „käivita“ protseduuri. Kui oli veel ülemprotseduure, mida pidi käivitama, mindi tagasi samm 2 juurde ja alustati uue kontrollringiga.

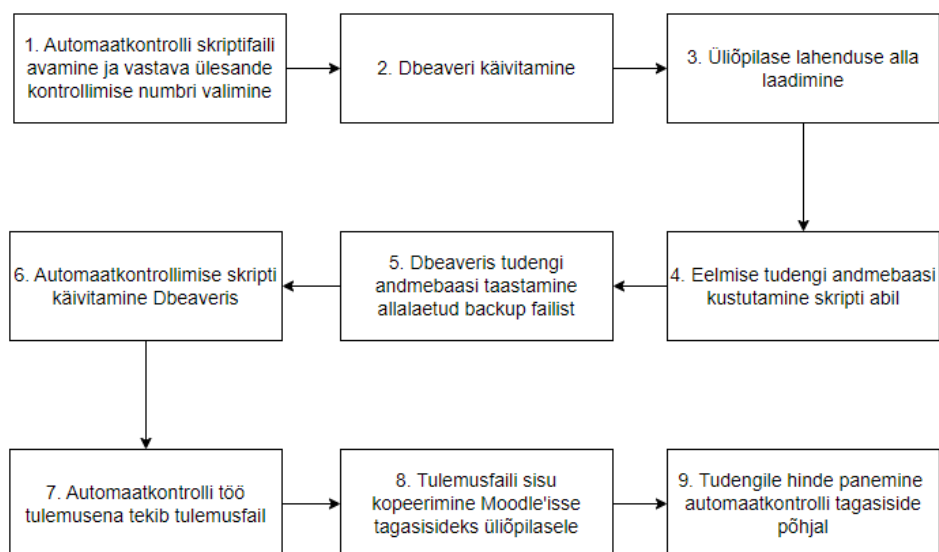
Viimaseks arvutati punktid ja väljastati tulemus (sammud 6 ja 7). Tulemusfaili väljastati ainult kontrolli käigus tekkinud vead, punktid ja üliõpilase nimi ning mõnede teemade puhul ka see, kui mitu kontrolli läbiti maksimaalsest kontrollide arvust õigesti.



Joonis 2. Kontrollprotseduuride töövoog

2.4. Praktikumijuhendaja automaatkontrolli kasutamise töövoog

Andmebaaside aines oli kasutusel Dbeaveri kasutajaliides, aga arenduse käigus loodi automaatkontrollide käivitamiseks ka pakkfail, seega oli praktikumijuhendajatel kaks automaatkontrolli käivitamisviisi. Joonisel 1 on näha Dbeaveriga kontrollimise (Lisa 4) töövoogu ja seda, millistest sammudest see koosneb. Kui praktikumijuhendaja hakkas praktikumi- või koduseid töid kontrollima, siis sammud 1 ja 2 olid alati ühekordsed tegevused. Juhendajad pidid panema paika versiooni (samm 1), mis määras ära ülesande mida kontrolliti. Sammud 4-7 olid seotud automaatkontrolli protsessiga ja tehti Dbeaveris.



Joonis 3. Praktikumijuhendaja töövoog Dbeaveri abil

Kui praktikumijuhendaja kasutas automaatkontrollide käivitamiseks pakkfaili ([Lisa 5](#)), siis polnud vaja Dbeaverit kasutada. Pakkfaili kasutamisel suurenes ülesannete kontrollimise kiirus ja vähenes üldine klikkide arv, mida põhjustasid joonisel 1 sammud 4-6, sest pakkfail käivitas käsureal nende sammude käsklused Dbeaverit avamata.

2.5. Probleemid automaatkontrollide arendamisel

Kuna automaatkontrollide arendamine toimus 2023. aasta kevadsemestril aine toimumisega samaaegselt, sai töö autor probleemide esinemisel praktikumijuhendajatelt alati kiire ja otsese tagasiside, mille põhjal sai teha automaatkontrollides parandusi ja täiendusi. Probleeme tekkis automaatkontrollide arendamisel palju ja kuigi enamikule leidsid lihtsad lahendused, tuli mõned automaatkontrollid põhjalikumalt ümber teha.

Üks suurem probleem, mis tekkis iga versiooni kontrollis, oli automaatkontrolli sisemine viga, kui üliõpilane oli esitanud andmebaasi, mille olukord polnud selline nagu peaks. Töö autoril tuli leida paindlikke lahendusi, et üliõpilase töö tagasisidestamine ei kannataks sellest, kui varasemalt on midagi valesti või puudulikult lahendatud. Selle probleemi lahendamiseks loodi eelkontrollid (joonis 4), mis enne lahenduse õigsuse kontrollimist kontrollivad andmebaasi olukorda (näiteks kas veergude nimed on õigesti muudetud või endiste nimedega).

```
606      if      (select check_column_exists('v_punktid','mangija'))=1 and
607              (select check_column_exists('v_punktid','punkt'))=1 and
608              (select check_column_exists('v_punktid','partii'))=1 then
609
610          if      (select check_column_datatype('v_punktid','mangija','int%')) = 1 then
611              if      (select check_column_datatype('v_punktid','punkt','numeric')) = 1 then
612                  if      (select check_column_datatype('v_punktid','partii','int%')) = 1 then
613
614                      -- vaate v_punktid partii 299 mangija 76 punktid: 0.5
615                      select count(*) into check_count from v_punktid where partii = 299 and mangija = 76;
616                      if      check_count = 1 then
```

Joonis 4. Eelkontrollide näidis

Esimeste praktikumitööde kontrollimisel leidsid praktikumijuhendajad, et Dbeaveris andmebaasi taastamine ei kustuta eelmise tudengi andmebaasi sisu ära, vaid jättis uue tudengi andmebaasi objekte, mis andsid automaatkontrolliga valepositiivseid tulemusi. Selle probleemi lahendamiseks loodi uus SQL skript lisa_tabelite_kustutamine, mille praktikumijuhendajad said peale iga üliõpilase töö kontrolli käivitada, et puhastada andmebaas eelmise tudengi tööst. Loodud skript lisati ka pakkfaili.

Kuna kontrolle oli igas ülesandes palju ja üliõpilane võis jätta varasemad vead parandamata, siis tekkis olukord, kus hilisemate tööde kontrollimisel ei olnud aru saada, kui palju oli üliõpilane lahendanud käesolevat ülesannet, sest kõik varasemad vead olid ka tagasisidetabelis näha. Praktikumijuhendajal puudus seega ülevaade, milline osa tagasisidest käis viimati kontrollitud lahenduse kohta. Lahenduseks loodi tagasisidetabelisse lisarida, mis andis teada, kui mitu kontrolli andsid õige tulemuse ja kui palju oli maksimaalne õigete tulemuste arv.

3. Tagasiside kogumine loodud automaatkontrollidele

Uurimisküsimustele „Mida peavad praktikumijuhendajad automaatkontrollide puhul oluliseks?“ ja „Millised on üliõpilaste hinnangud loodud automaatkontrollidele?“ vastuste saamiseks koguti informatsiooni küsitlusankeetide abil. Küsitlused viidi läbi Tartu Ülikoolis 2023. aasta kevadsemestril aines Andmebaasid. Aines osalenud üliõpilased vastasid kahele erinevale ankeedile semestri erinevates etappides ja praktikumijuhendajatele saadeti üks ankeet semestri keskel. Ankeetide koostamiseks kasutati Google Vormi.

3.1. Valim

Aines Andmebaasid osales 2023. aasta kevadsemestril umbes 340 üliõpilast. Üliõpilased võisid vastata mõlemale küsitlusankeedile, aga ka ainult esimesele või ainult teisele. Küsitluste läbiviimisel oli kolm alamvalimit. Esimene alamvalim moodustus üliõpilastest, kes vastasid esimesele tagasisideankeedile. Teine alamvalim koosnes üliõpilastest, kes vastasid teisele tagasiside küsitlusele. Kolmas alamvalim moodustus praktikumijuhendajatest.

Esimesele küsitlusele vastas 180 üliõpilast, kellest 98 olid mehed ja 82 naised (vastavalt 54,6% ja 45,6%). Vastanutest 86,7% ehk 156 üliõpilase jaoks oli aine Andmebaasid kohustuslikuks aineks, 7,2% ehk 13 üliõpilase jaoks oli aine üliõpilase valitud moodulis kohustuslik, 4,4% ehk 8 üliõpilase jaoks oli tegu vabaainega ja 1,7% ehk 3 üliõpilase jaoks valikainega..

Vastanutelt küsiti taustainfo kogumiseks ka nende varasemat kogemust SQL keelega. Varasem kokkupuude SQLiga puuduv või vähene – 140 üliõpilast ehk 77,7% vastanutest polnud sellest keelest kas midagi kuulnud või teadsid vaid, et see on olemas. Vaid 22 üliõpilast (12,2%) olid SQLiga eelnevalt veidi tutvunud ja 15 üliõpilast (8,3%) olid seda keelt varasemalt mõnel kursusel õppinud. Vastajate seas oli 3 üliõpilast (1,7%), kes olid SQLiga varem töötanud.

Teisele küsitlusele vastas 107 üliõpilast, kelle hulgas oli 54 naist, 50 meest ja 3 inimest, kes ei soovinud sugu avaldada (vastavalt 50,5%, 46,7% ja 2,8%). Vastanutest 81,3% ehk 87 üliõpilase jaoks oli aine Andmebaasid kohustuslikuks aineks, 7,5% ehk 8 üliõpilase jaoks oli aine üliõpilase valitud moodulis kohustuslik, 5,6% ehk 6 üliõpilase jaoks valikaineks ja 4,7% ehk 5 üliõpilase jaoks oli tegu vabaainega.

Kolmandale küsitlusele vastas 13 praktikumijuhendajat, kellest 9 olid mehed ja 4 naised. Esmakordseid üliõpilastest praktikumijuhendajaid oli 4, ülejäänud 9 on andmebaaside praktikumijuhendajad olnud vähemalt kaks aastat.

3.2. Andmete kogumine

Andmeid koguti kolme küsitlusega kokku kolmel erineval korral. Üliõpilastele saadeti esimene ankeet 13.03.23 ja vastamiseks anti nädal aega. Esimese küsitluse ajaks oleksid üliõpilased võinud olla automaatkontrolliga tagasisidet saanud kaks korda. Kõik üliõpilased seda aga polnud, sest praktikumijuhendajad polnud jõudnud veel nende lahendusi hinnata, või polnud üliõpilased küsitluste lõpus olnud lisakommentaariks kirjutatud vastustest järeldades aru saanud, et nende tagasiside on automaatkontrolli poolt.

Esimeses küsitluses ([Lisa 1](#)) oli 4 taustaküsimust ja 11 sisulist küsimust, millest kuus olid viiepalliskaalaga küsimused, kolm vabatekstilised, üks valikvastustega ja üks jah/ei küsimus. Ankeedi sisuline osa oli jaotatud kaheks. Esimeses osas näidati üliõpilastele automaatkontrolli tagasisidetabeli pilti ning uuriti, kuidas nad tagasisidetabelist aru saavad. Teises osas uuriti, kuidas saavad üliõpilased aru nende endi esitatud lahendustele saabunud tagasisidest. Teise osa küsimusi suunati vastama vaid need üliõpilased, kes küsitluses vastasid, et on automaatkontrolli tagasisidet saanud.

Teine küsitlus saadeti üliõpilastele 24.04.23 ja vastamiseks anti samuti nädal aega. Teise küsitluse ajaks olid üliõpilased saanud automaatkontrolliga tagasisidet neljale kodutööle ja üheksale praktikumitööle. Teises küsitlusankeedis ([Lisa 2](#)) oli 3 taustaküsimust ja 12 sisulist küsimust, millest viis olid viiepalliskaalaga küsimused, neli vabatekstilised, kaks valikvastustega ja üks jah/ei küsimus. Ka teise küsitluse sisuline osa oli jaotatud kaheks. Esimeses osas uuriti, kuidas saavad üliõpilased aru enda lahendustele saadud automaatkontrolli tagasisidest. Teises osas uuriti üliõpilaste ootusi automaatkontrolli edasiarenduse kohta. Esimesele osale vastasid vaid need üliõpilased, kes küsitluse algul vastasid, et on automaatkontrolli tagasisidet saanud.

Kolmas küsitlus saadeti praktikumijuhendajatele 13.04.23 ja vastamiseks anti kaks nädalat. Küsitluse ajaks on praktikumijuhendajad saanud kasutada automaatkontrolle kolme kodutöö ja üheksa praktikumitöö kontrollimiseks. Ankeedi ([Lisa 3](#)) abil uuriti, kuidas hindavad õppejõud automaatkontrolli tagasiside arusaadavust, punktisüsteemi arusaadavust, automaatkontrolli mugavust kasutamisel ja samuti küsiti nende arvamusi automaatkontrolli edasiarenduse osas. Küsitluses oli lisaks 4 taustaküsimusele 13 sisulist küsimust, mille hulgas oli 7 viiepalliskaala küsimust, 3 valikvastustega küsimust ja 3 vabatekstilist küsimust.

3.3. Andmete töötlemine

Küsitluste tulemused koguti Google Vormis ja salvestati CSV formaadis failidesse. Andmed töödeldi ja analüüsiti statistiliste meetodite abil kasutades programmi Microsoft Excel.

Esimese küsitluse vastused töödeldi ja analüüsiti kahes osas (jaotus tehti küsitluse sisuliste osade põhjal). Esimese küsitluse esimeses osas analüüsiti kõigi vastanud üliõpilaste vastuseid. Teise osa vastuste hulgast eemaldati nende üliõpilaste vastused, kelle praktikumijuhendaja ei kasutanud käesoleva autori loodud automaatkontrolle (tuvastati praktikumijuhendajatelt saadetud küsitluse vastuste ja praktikumirühmade nimekirjade abil) või kes polnud oma lahendustele üldse automaatkontrolli tagasisidet saanud. Teise osa analüüsi jäi seega 180st küsitlusele vastanud üliõpilasest alles 80.

Teise küsitluse vastused töödeldi ja analüüsiti samuti kahes osas (osadeks jaotati küsitluse sisuliste osade põhjal). Küsitluse esimeses osas eemaldati osa üliõpilaste vastused täpselt samal põhimõttel nagu esimese küsitluse teise osa analüüsist, seega jäi esimesse osasse 107st vastajast alles 75. Teise küsitluse teises osas analüüsiti kõigi vastanud üliõpilaste vastuseid. Kolmanda küsitluse vastuste andmetöötluse käigus jagati vastused kolme ossa vastavalt praktikumijuhendajate vastustele. Mitte ühegi vastaja vastuseid analüüsist ei eemaldatud.

3.4. Üliõpilaste küsitluste tulemused

3.4.1. Üldine tagasisidetabeli arusaadavus

Esimese küsitluse esimeses osas uuriti üliõpilaste üldist arusaamist automaatkontrolli tagasisidetabelist. Küsimustikule vastanud pidid vastama kolmele küsimusele joonisel 5 toodud tulemustabeli järgi.

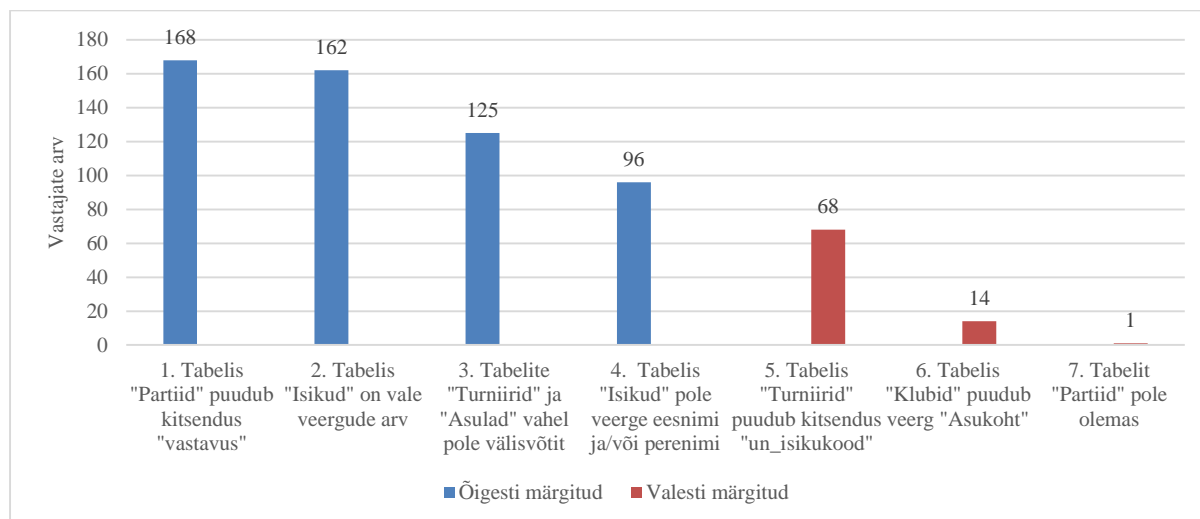
	ABC ylesanne	ABC kontrolli_nimi	ABC tagasiside	ABC olek	123 punktid	123 max_punktid
1	Praktikum 3	Tabel "Partiid" Kitsendus "vastavus"	ei ole olemas	VIGA	0	0
2	Praktikum 3	Tabel "Isikud" Veergude arv	on vale, peab olema 8	VIGA	0	0
3	Praktikum 3	Tabel "Isikud" Kitsendus "un_isikukood"	ei ole olemas	VIGA	0	0
4	Praktikum 3	Tabel "Isikud" Isikul "Irys"	puuduvad veerud eesnimi ja/või perenimi	VIGA	0	0
5	Praktikum 3	Tabel "Isikud" Veerg "Klubis"	ei ole olemas	VIGA	0	0
6	Praktikum 3	Tabel "Partiid" Kitsendus "ajakontroll"	ei ole olemas	VIGA	0	0
7	Praktikum 4	-	-	Hindepunkt	1	1
8	Kodutoo 3	Välisvõti "fk_turniir_2_asula"	ei ole olemas	VIGA	0	0.1
9	Kodutoo 3	-	-	Hindepunkt	0.4	0.5

Joonis 5. Automaatkontrolli tulemuste tabel, mida näidati üliõpilastele esimeses küsitluses

Üliõpilased pidid valikvastustest valima variandid, mis kajastavad tagasisidetabelis välja toodud vigu. Õiged valikvastusevariandid, mida oleks pidanud märkima, on:

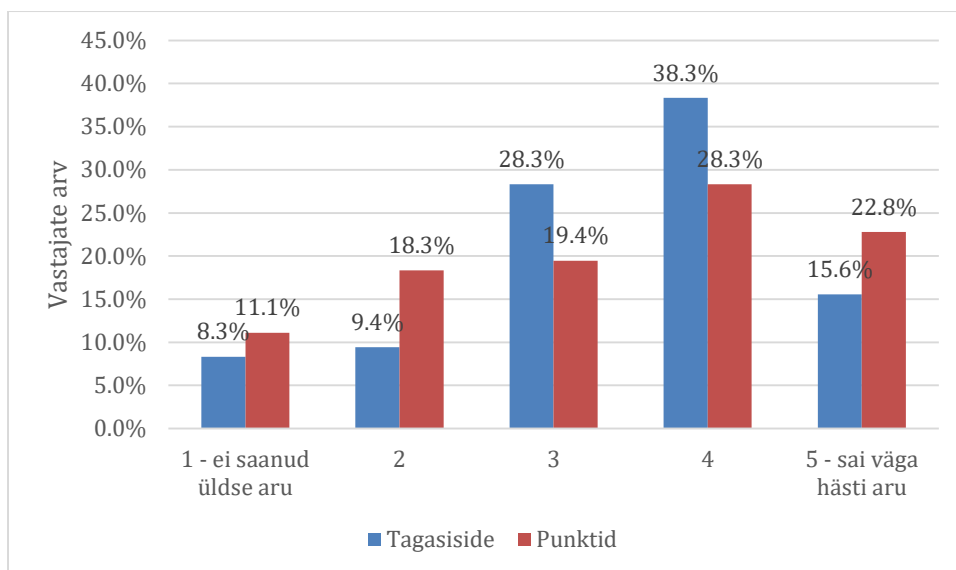
1. Tabelis "Partiid" puudub kitsendus "vastavus",
2. Tabelis "Isikud" on vale veergude arv,
3. Tabelis "Isikud" pole veerge eesnimi ja/või perenimi,
4. Tabelite "Turniirid" ja "Asulad" vahel pole välisvõtit.

Joonisel 6 on näha, kui palju valiti igat valikvastuse varianti. Kõige paremini olid tagasisidetabelis arusaadavad vead *Tabelis "Partiid" puudub kitsendus "vastavus"* ja *Tabelis "Isikud" on vale veergude arv*, neid valisid 180st üliõpilasest vastavalt 168 (93,3%) ja 162 (90,0%) vastanutest.



Joonis 6. Tagasisidetabeli põhjal vastatud valikvastuste jaotumine

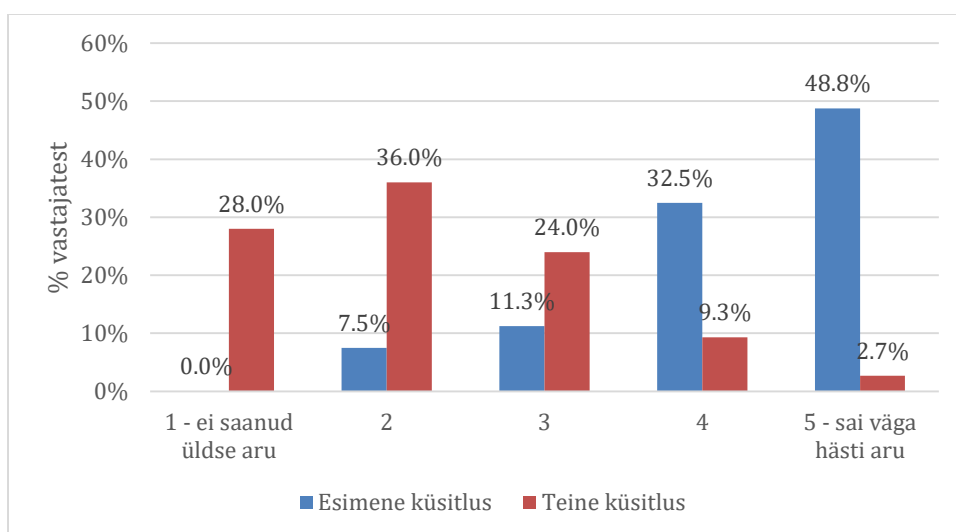
Tagasisidetabeli pildil oleva tagasiside arusaadavuse kohta saab väita, et üle poolte (97 üliõpilast ehk 53,9%) said tagasisidetabelist üle keskmise hästi aru (joonis 7). Pildil olevate saadud/kaotatud punktide arusaadavuse osas ka saadud/kaotatud punktidest said üle poolte üliõpilastest üle keskmise hästi aru (kokku 92 üliõpilast ehk 51,1%).



Joonis 7. Tagasiside ja saadud/kaotatud punktide arusaadavus tagasisidetabeli pildi põhjal

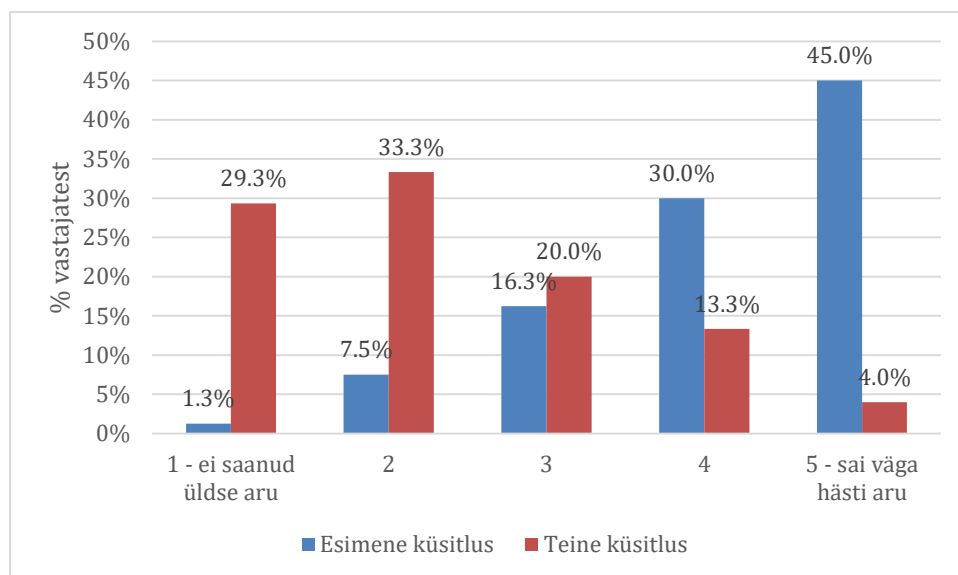
3.4.2. Enda lahendustele saadud tagasiside arusaadavus

Nii esimeses kui teises küsitluses uuriti üliõpilastelt, mil määral saavad nad aru nende endi esitatud lahendustele automaatkontrolli antud tagasisidest ja saadud/kaotatud punktidest. Joonisel 8 välja toodud tagasiside arusaadavuse võrdluses on näha, et teise küsitluse ajal andsid üliõpilased tagasiside arusaadavusele madalamaid hindeid kui esimese küsitluse ajal. Esimeses küsitluses ei valinud varianti „ei saa üldse aru“ mitte ükski üliõpilane, teises küsitluses aga 28,0% kõigist vastanutest.



Joonis 8. Automaatkontrolli tagasiside arusaadavus üliõpilaste seas

Joonisel 9 on näha üliõpilaste arusaamine saadud ja kaotatud punktidest ning esimese küsitluse ajal said üliõpilased saadud/kaotatud punktidest paremini aru kui teise küsitluse ajal.



Joonis 9. Automaatkontrolliga saadud/kaotatud punktide arusaadavus üliõpilaste seas

Kodutööd 1 ja 2 kontrolliti õppejõudude poolt käsitsi, praktikum 3 ja kodutöö 3 aga automaatselt. Üliõpilastelt küsiti, mil määral said nad automaatselt kontrollitud lahenduste tagasiside kiiremini kui käsitsi kontrollitud lahenduste puhul. Vastajatest 65,0% ehk 52 üliõpilast valisid viiepalliskaalal keskmise variandi ehk nad ei tundnud tagasiside saabumise kiiruses vahet. Automaatkontrolli tagasiside saabumise hindasid kiiremaks 28,7% ehk 23 üliõpilast ja aeglasemaks 6,3% ehk 5 üliõpilast.

Veel küsiti esimeses küsitluses üliõpilastelt, mil määral said nad automaatselt kontrollitud lahenduste tagasisidest ja saadud/kaotatud punktidest paremini aru kui käsitsi kontrollitud lahenduste puhul. Vastajatest 70,0% ehk 56 said automaatse kontrolli tagasisidest ja punktidest sama hästi aru kui käsitsi antud tagasiside ja punktide puhul. Automaatne tagasiside ja punktid olid arusaadavamad 16,3%-le ehk 13 üliõpilasele ja vähem arusaadavamad 13,7%-le ehk 11 üliõpilasele.

Teises küsitluses uuriti üliõpilastelt, kas nende esitatud lahendustele saadud tagasiside on küsitluste vahelise ajaga, ligikaudu ühe kuuga, arusaadavamaks muutunud. 46,7% ehk 35

üliõpilast valis viiepalliskaalal keskmise variandi ehk peaaegu pooltel üliõpilastel jääb tagasisidest arusaamine umbes samale tasemele. 22 üliõpilast ehk 29,5% valisid variante 4 ja 5 (5 – „muutus oluliselt arusaadavamaks“) ja 18 üliõpilast ehk 24,0% variante 1 ja 2 (1 – „ei muutunud üldse arusaadavamaks“).

Kahe küsitluse vahel tehti automaatkontrolli tagasisidesse muudatus, et tagasiside lõpus oleks rida, mis kajastab õigesti tehtud ülesannete ja maksimaalsete võimalike õigete vastuste arvu. Üliõpilastelt küsiti, mil määral tagasisidetabelisse lisatud reast (joonis 10) aru saadi. Vastajatest 38,7% ehk 29 üliõpilast valis viiepalliskaalal variandi 5 („sain väga hästi aru“) ja 37,3% ehk 28 üliõpilast valis variandi 4. Varianti 1 („ei saanud üldse aru“) valis 2,7% ehk 2 üliõpilast.

Praktikum 7

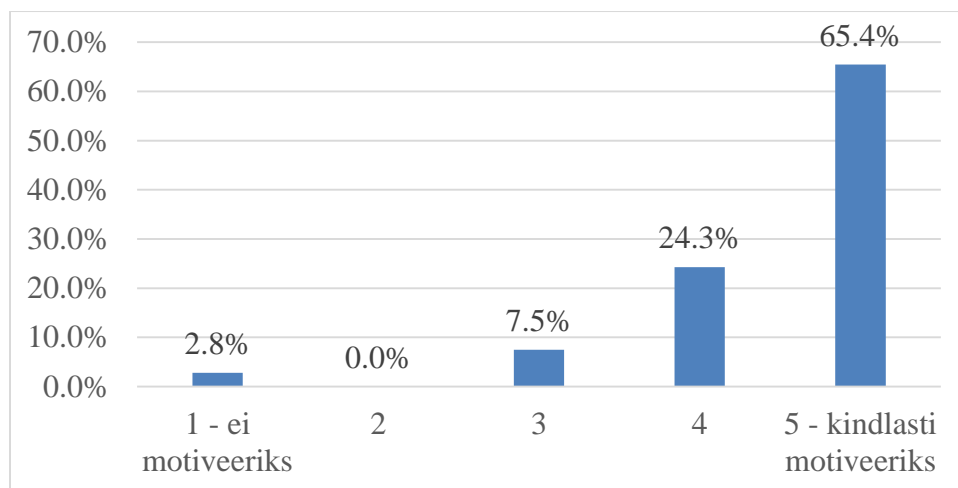
Õigesti tehtud: 57

Maksimum oiged: 64

Joonis 10. Tagasisidetabelisse lisatud rida

3.4.3. Üliõpilaste ootused automaatkontrollidele

Teise küsitluse lõpus küsiti üliõpilastelt nende arvamust ja ootusi automaatkontrolli edasiarenduse kohta. Esiteks küsiti, kas üliõpilased eelistaksid automaatkontrolli, mida nad saaksid enne lahenduse esitamist ise kasutada, et kontrollida lahenduse õigsust või automaatkontrolli Moodle'i keskkonnas, kus nad saaksid kohe teada lahenduse tulemuse ja punktid. 83 üliõpilast ehk 77,6% vastanuist eelistas automaatkontrolli, mida saaks enne lahenduse esitamist ise kasutada, 12 üliõpilast ehk 11,2% vastanuist eelistas automaatkontrolli Moodle'is keskkonnas ja ülejäänud 12-l (11,2%) üliõpilastest ei olnud vahet, missugune automaatkontroll olla võiks. Seejärel küsiti, et kui üliõpilastel oleks endil võimalik automaatkontrolli kasutada, kas see motiveeriks neid rohkem pingutama ja/või vigu parandama (joonis 11).



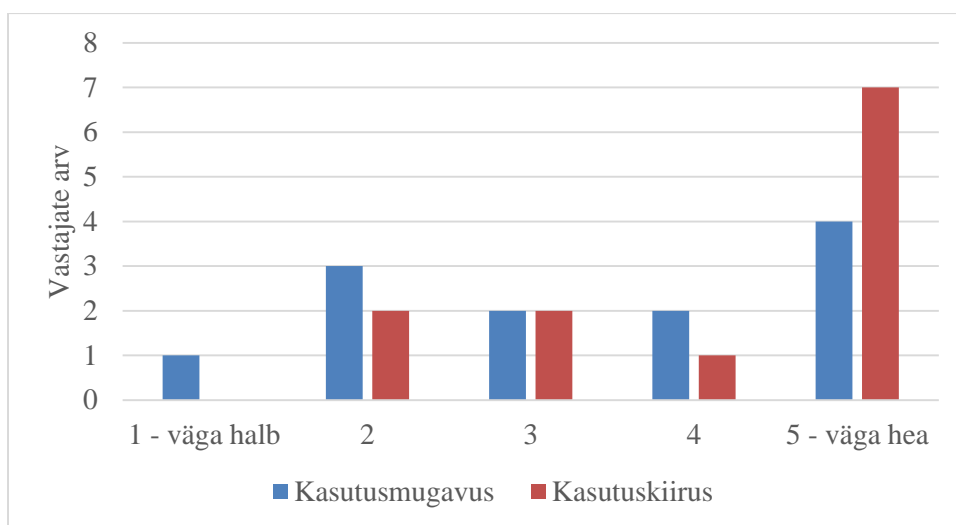
Joonis 11. Üliõpilaste motiveeritus vigade parandamiseks neile kättesaadavate automaatkontrollide olemasolu korral

Lõpuks küsiti üliõpilastelt veel, et missugust tagasisidet eelistavad nad kõige enam saada – ainult automaatkontrollilt, ainult õppejõult või mõlemat. Selgus, et 84 üliõpilast ehk 78,5% eelistaks saada nii automaatkontrolli kui õppejõu tagasisidet, 7 üliõpilast ehk 6,5% soovib ainult õppejõu tagasisidet, 4 üliõpilast ehk 3,7% soovib ainult automaatkontrolli tagasisidet ning 12 üliõpilasel ehk 11,2% üliõpilastest ei ole vahet, missugust tüüpi tagasisidet enda lahendustele saavad.

3.5. Praktikumijuhendajate küsitluste tulemused

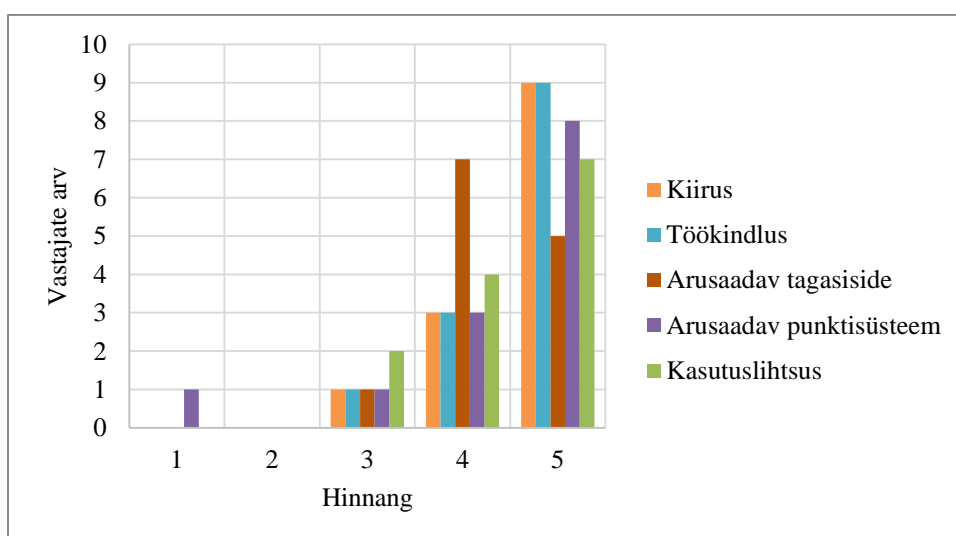
Praktikumijuhendajate küsitluses uuriti, kuidas praktikumijuhendajad üldiselt automaatkontrollide tagasisidest aru saanud on ja kui selgesti on nad mõistnud, kus üliõpilasel vead on tekkinud. Üldise arusaadavuse küsimuses valis 5 praktikumijuhendajat viiepalliskaalal hinde 5 („väga arusaadav“) ja 7 praktikumijuhendajat hinde 4. Vigadest arusaamise puhul valis 2 õppejõudu hinde 5 („sain hästi aru“), 8 praktikumijuhendajat hinde 4 ja 2 valisid hinde 3.

Seejärel uuriti, kuidas on praktikumijuhendajad rahul automaatkontrollide kasutusmugavuse ja -kiirusega (joonis 12). Kasutusmugavusega olid viiepalliskaalal „väga rahul“ variandi valinud 4 praktikumijuhendajat ja kõige madalama variandi („väga halb“) üks praktikumijuhendaja. Kasutuskiiiruse küsimuse puhul valisid viiepalliskaalal varianti 5 („väga hea“) 7 praktikumijuhendajat ja madalaimat varianti ei valinud ükski praktikumijuhendaja.



Joonis 12. Praktikumijuhendajate hinnang automaatkontrollide kasutusmugavusele ja -kiirusele

Automaatkontrollide mugavamaks kasutamiseks lõi töö autor pakkfaili ja küsitluse tulemusel selgus, et seda kasutas 13st vastanud praktikumijuhendajast 2. Veel küsiti praktikumijuhendajatelt pakkfaili kasutusjuhendi arusaadavuse kohta ning 13st vastanust 3 hindas juhendit hindegaga 5 („väga hea“), 4 hindegaga 4, 5 hindegaga 3 ja 1 hindegaga 1 („väga kehv“). Praktikumijuhendajatel paluti viiepalliskaalal hinnata, kui oluliseks peavad nad automaatkontrolli puhul erinevaid omadusi (joonis 13).



Joonis 13. Praktikumijuhendajate hinnang automaatkontrolli omadustele

Praktikumijuhendajate küsitluses käsitleti ka automaatkontrollide edasiarendamist. Küsiti, kui automaatsed peaksid automaatkontrollid nende arvates olema ja 9 praktikumijuhendajat arvas, et täielikult automaatsed (praktikumijuhendaja osa tööde kontrollimisel puuduks ja üliõpilased saaksid oma lahendusi ise automaatkontrolliga kontrollida) ning 4 praktikumijuhendajat arvas, et poolautomaatsed (praktikumijuhendaja käivitab kontrolli, näeb üliõpilase andmebaasi ja saab lisada täiendavat e isikupärasemat tagasisidet).

Uuriti ka, et kui automaatkontrollid oleksid täielikult automaatsed ehk üliõpilane laeks oma lahenduse ise üles ja saaks kohe ka tulemuse, mitu võimalust peaks üliõpilastel töö esitamiseks olema. Vastanutest 6 arvasid, et lõpmatu arv kordi, 3 arvasid, et 3-4 võimalust, 3 arvasid, et kaks korda ja üks praktikumijuhendaja arvas, et peaks olema 1 esitusvõimalus.

5. Arutelu

Nagu töös varasemalt on välja toodud, siis töö autori esimene soov oleks olnud varem arendatud automaatkontrolle ümber kujundada. Ainekursuse muutmine tähendas aga, et luua tuli täiesti uued kontrollid. Kuna automaatkontrollide arendamine toimus paralleelselt semestriga, siis lõi see hea võimaluse loodut testida ja pidevalt paremaks muuta. Kõik loodud automaatkontrollid olid edukad ja võeti kursuse raames kasutusele. Kuigi automaatkontrollid võeti koheselt kasutusele, siis suured muutused õppeaines tähendasid kiiret tempot ja suurt koormust.

Esimene uurimisküsimus oli „Kuidas koostada automaatkontrolle PostgreSQL'i peale koostatud kodutööde kontrollimiseks?“. Eelnevat tööd analüüsides selgus, et vaja on luua täiesti uued automaatkontrollid. Algolukord püüti lihtsuse huvides tekitada selline, et säiliksivad samad funktsionaalsed nõuded nagu 2020. aastal loodud kontrollides. Koostati automaatkontrollid, mis sarnaselt varem andmebaasides kasutusel olnud automaatkontrollidega (Mitrovic, 1998) andsid tudengile tagasiside, mis on tema andmebaasis valesti. Nagu ka varasemate tööriistadega AsseSQL ja SQL-Tutor, võrreldi käesoleva töö käigus arendatud automaatkontrollide puhul üliõpilase lahendust õige tulemusega – kui üliõpilase tulemus oli õige, sai punkti ja kui oli juba osaliselt vale, sai vea (Nalintipayawong jt, 2017; Prados jt, 2006; Prior ja Lister, 2004; Sadiq jt, 2004).

Automaatkontrollide loomisel tuli välja rohkelt kitsaskohti, mis aeglustasid tunduvalt kontrollide arendust või mille varasemal ilmnmisel oleks saanud mõnd probleemi vältida. Kuna automaatkontrollide arendus toimus samaaegselt ainesse ülesannete loomisega, tuli arenduse jooksul ette väga palju ootamatuid muudatusi ja vigade tekkimiseks oli suur oht. Kogu Dbeaveri keskkond ja aines kasutatav PostgreSQL keel olid kõigile uued, mis tekitas paljudele rohkelt küsimusi ja segadust. Automaatkontrolle tuli arendada nädalakaupa ja tihti tekkis olukord, kus sama nädala jooksul tuli arendada järgmise nädala ülesannetele kontrolle ja parandada käesoleva nädala kontrollides ilmnenuv veakohti.

Vaatamata kõigile tagasilöökidele ja murekohtadele said automaatkontrollid siiski valmis ja ükski automaatselt kontrollida planeeritud lahendus ei jäänud automaatselt kontrollimata. Ainus miinuskoht praeguste kontrollide puhul on, et neid suudab hallata ainult käesoleva töö autor. Sarnane probleem toodi välja ka Keuning jt (2016) uurimuses, kus öeldi, et tihti on automaatkontrollide korrashoidmiseks võimalised ainult nende arendajad.

Teine uurimisküsimus oli „Millised on üliõpilaste hinnangud loodud automaatkontrollidele?“. Magistritöö käigus läbi viidud küsitluste tulemustest selgus, et üliõpilaste üldine arusaamine automaatkontrolli tagasisidetabelist on pigem hea. Huvipakkuvamad tulemused on nende küsimuste osas, kus taheti üliõpilastelt teada, kuidas on nad aru saanud otseselt nende endi lahendustele antud tagasisidest ja punktidest. Esimeses küsitluses, mis viidi läbi umbes semestri esimese kolmandiku lõpus, selgus, et üliõpilased said enda lahendustele saadud tagasisidest ja punktidest palju paremini aru kui teises küsitluses, mis viidi läbi umbes semestri teise kolmandiku lõpus.

Arusaadavuse vähenemise üheks põhjuseks võib olla asjaolu, et semestri algul lahendasid üliõpilased lihtsamaid ülesandeid, milles tehtud vigadele saadud tagasiside oli ka lühem, konkreetsem ja selgemini mõistetav. Kui semestri jooksul ülesanded aina keerukamaks muutusid, muutus pikemaks ja keerulisemaks ka tagasiside. Kuna iga lahenduse puhul kontrollis automaatkontroll ka kõiki eelnevaid vastuseid ning osa üliõpilasi eelnevates ülesannetes tehtud vigu ei parandanud, võis tagasiside hulk kasvada üpris pikaks, mis võis tingida ka tagasiside kehvema arusaadavuse. Üheks abistavaks meetmeks võiksid tagasisides olla suunavad vihjed, mille abil saaksid üliõpilased oma vigu parandada. Nagu ka Keuning jt (2016) ning Keuning jt (2018) on väitnud, siis üliõpilane ei pruugi vigu parandada ainult automaatkontrollipõhise tagasiside põhjal, vaid vajab tagasisides ka suuniseid, kuidas vigu parandada. Liiga täpsed juhised võivad negatiivsete tagajärgedega olla üliõpilastele, kes on suure saavutussooviga (Deeva jt, 2021). Samuti on öeldud, et heas tagasisides tuuakse põhjendatult välja nii positiivseid ja negatiivseid aspekte tehtud töö kohta (Hunt ja Karm, 2022).

Kolmas uurimisküsimus oli „Millised on praktikumijuhendajate hinnangud loodud automaatkontrollidele?“. Praktikumijuhendajad hindasid semestri keskel toimunud küsitluses automaatkontrollide tagasiside pigem hea või väga hea hindeg. Automaatkontrolli kasutusmugavus ja kasutuskiiirus hinnati viiepalliskaalal kergelt positiivsema hinde suunas. Erinevate automaatkontrolli omaduste hindeid analüüsid selgus, et praktikumijuhendajad hindavad kõige kõrgemalt automaatkontrolli kasutuskiirust ja töökindlust. Veidi ebaolulisemaks hinnati arusaadavat tagasisidet ja punktisüsteemi. Punktisüsteemi puhul hindas üks praktikumijuhendaja olulisust lausa madalaima hindeg, mida ühegi teise omaduse puhul öelda ei saa.

Enamik praktikumijuhendajatest arvas, et tuleks kasutusele võtta täielikult automaatsed kontrollid, mis muudaks õppejõu osa tööde kontrollimisel minimaalseks. Üliõpilaste poolt selgus, et üle kolmveerandi vastanutest sooviks enda lahendustele nii õppejõu kui automaatkontrolli antud tagasisidet. Sarnased tulemused selgusid ka Muuli jt (2021) uurimuses, kus leiti, et hindamine peaks endas hõlmama nii automaatset kui personaalset tagasisidet.

Edasiseks arendamiseks oleks variantideks luua eraldi keskkonda automaatkontroll, mida saaksid üliõpilased enne lahenduse esitamist selle õigsuse kontrollimiseks ise kasutada või siduda automaatkontroll õppekeskkonnaga Moodle, kuhu lahenduse laadides saaksid üliõpilased kohe teada tulemuse ja punktide arvu. Kuna üheks põhjuseks tagasiside saamisel on võimalus enda eksimustest õppida ja neid parandada, uuriti üliõpilastelt, mil määral oleksid nad rohkem motiveeritud pingutama ja/või vigasid parandama, kui neil oleks neile endile kättesaadav automaatkontroll ning tugev enamus hindas, et oleksid väga motiveeritud.

Kokkuvõte

Magistritöö eesmärk oli luua ainele Andmebaasid uued automaatkontrollid PostgreSQL keeles ja koguda automaatkontrollide kasutamiskogemuse kohta tagasisidet nii üliõpilastelt kui õppejõududelt. Eesmärk sai täidetud 2023. aasta kevadsemestril, kui töö autor arendas vastavad automaatkontrollid ja need võeti praktikumijuhendajate poolt kasutusele.

Tagasiside kasutamiskogemuse kohta koguti küsitluste abil üliõpilastelt ja praktikumijuhendajatelt. Üliõpilaste küsitluste tulemusel tuli välja, et semestri esimeses pooles oli automaatkontrollide tagasisidest arusaamine palju arusaadavam kui semestri teises pooles, mille põhjusteks võis olla hilisemate automaatkontrollide keerukus või üliõpilase liigne vigade hulk, sest lahenduste esitamiskordade vahel poldud parandusi tehtud. Samamoodi langes üliõpilaste arusaam punktisüsteemist. Praktikumijuhendajate arusaam automaatkontrollide tagasisidest ja punktisüsteemist oli väga hea. Praktikumijuhendajatelt küsiti lisaks, mida nad automaatkontrollide omaduste hulgas oluliseks hindavad ja kõige olulisemateks hinnati kasutuskindlust ja kasutuskiirust.

Üliõpilased eelistavad küsitlustulemuste analüüsi põhjal tagasisidet nii automaatkontrollidelt kui ka personaalselt praktikumijuhendajalt. Nii üliõpilased kui praktikumijuhendajad soovivad, et aines oleksid kasutusel täisautomaatsed kontrollid, mida üliõpilased saaksid ise kasutada. Täielikult automaatsete ja üliõpilastele kättesaadavad automaatkontrollid ongi käesoleva magistritöö edasiarenduse võimalus. Samuti saaks tulevikus katta käesoleva töö käigus katmata jäetud SQL päringute kontrollid.

Tänuavaldused

Sooviksin tänada enda juhendajaid, Piret Luike ja Vambola Lepingut, kes toetasid õppeprotsessis ja pühendasid aega nõustamisele ning tagasisidestamisele. Samuti sooviksin tänada aine Andmebaasid praktikumijuhendajaid, kes automaatkontrolle kasutasid ja tagasiside andsid. Tänan oma perekonda ja sõpru, eriti Kreetet ja Eliset, emotsionaalse toe ja sisuka tagasiside eest.

Kasutatud kirjandus

- Ala-Mutka*, K. M. (2015). A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2), 83–102.
- Andmebaasid—Kursused—Arvutiteaduse instituut. (s.a.). Salvestatud 7. mai 2023, <https://courses.cs.ut.ee/2023/Andmebaasid/spring>
- Application Express SQL Workshop and Utilities Guide. (s.a.). Salvestatud 12. mai 2023, https://docs.oracle.com/cd/E14373_01/user.32/e13370/sql_rep.htm#AEUTL191
- Auffarth, B., López-Sánchez, M., Campos i Miralles, J., & Puig, A. (2008). *System for Automated Assistance in Correction of Programming Exercises (SAC)*. 104–113. <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-72947>
- Automaadid, keeled ja translaatorid—Kursused—Arvutiteaduse instituut. (s.a.). Salvestatud 31. märts 2023, <https://courses.cs.ut.ee/2023/AKT/spring>
- Barata, G., Gama, S., Jorge, J., & Gonçalves, D. (2013). Improving participation and learning with gamification. *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, 10–17. <https://doi.org/10.1145/2583008.2583010>
- Bhangdiya, A., Chandra, B., Kar, B., Radhakrishnan, B., Reddy, K. V. M., Shah, S., & Sudarshan, S. (2015). The XDa-TA system for automated grading of SQL query assignments. *2015 IEEE 31st International Conference on Data Engineering*, 1468–1471. <https://doi.org/10.1109/ICDE.2015.7113403>
- Brusilovsky, P., Sosnovsky, S., Yudelso, M. V., Lee, D. H., Zadorozhny, V., & Zhou, X. (2010). Learning SQL Programming with Interactive Tools: From Integration to Personalization. *ACM Transactions on Computing Education*, 9(4), 19:1-19:15. <https://doi.org/10.1145/1656255.1656257>

- Busch, C. (2014). *ECGBL2014-8th European Conference on Games Based Learning: ECGBL2014*. Academic Conferences and Publishing International.
- Chandra, B., Joseph, M., Radhakrishnan, B., Acharya, S., & Sudarshan, S. (2016). Partial marking for automated grading of SQL queries. *Proceedings of the VLDB Endowment*, 9(13), 1541–1544. <https://doi.org/10.14778/3007263.3007304>
- Chen, P. M. (2004). An automated feedback system for computer organization projects. *IEEE Transactions on Education*, 47(2), 232–240. <https://doi.org/10.1109/TE.2004.825220>
- Dawson, P., Henderson, M., Mahoney, P., Phillips, M., Ryan, T., Boud, D., & Molloy, E. (2019). What makes for effective feedback: Staff and student perspectives. *Assessment & Evaluation in Higher Education*, 44(1), 25–36. <https://doi.org/10.1080/02602938.2018.1467877>
- Deeva, G., Bogdanova, D., Serral, E., Snoeck, M., & De Weerd, J. (2021). A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education*, 162, 104094. <https://doi.org/10.1016/j.compedu.2020.104094>
- Dekeyser, S., Raadt, M., & Lee, T. (2007). *Computer Assisted Assessment of SQL Query Skills*. 63, 53–62.
- Douce, C., Livingstone, D., & Orwell, J. (2005). Automatic test-based assessment of programming: A review. *Journal on Educational Resources in Computing*, 5(3), 4-es. <https://doi.org/10.1145/1163405.1163409>
- Edwards, S. H. (2003). Improving student performance by evaluating how well students test their own programs. *Journal on Educational Resources in Computing*, 3(3), 1-es. <https://doi.org/10.1145/1029994.1029995>

- Fonte, D., Cruz, D. da, Gañarski, A. L., & Henriques, P. R. (2013). A Flexible Dynamic System for Automatic Grading of Programming Exercises. J. P. Leal, R. Rocha, & A. Simões (Toim), *2nd Symposium on Languages, Applications and Technologies* (Kd 29, lk 129–144). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. <https://doi.org/10.4230/OASIcs.SLATE.2013.129>
- Forsythe, G. E., & Wirth, N. (1965). Automatic grading programs. *Communications of the ACM*, 8(5), 275–278. <https://doi.org/10.1145/364914.364937>
- Gupta, S. (2012). *Automatic Assessment of Programming assignment*. 315–323. <https://doi.org/10.5121/csit.2012.2129>
- Hext, J. B., & Winings, J. W. (1969). An automatic grading scheme for simple programming exercises. *Communications of the ACM*, 12(5), 272–275. <https://doi.org/10.1145/362946.362981>
- Higgins, C., Hegazy, T., Symeonidis, P., & Tsintsifas, A. (2003). The CourseMarker CBA System: Improvements over Ceilidh. *Education and Information Technologies*, 8(3), 287–304. <https://doi.org/10.1023/A:1026364126982>
- Hollingsworth, J. (1960). Automatic graders for programming classes. *Communications of the ACM*, 3(10), 528–529. <https://doi.org/10.1145/367415.367422>
- Hunt, P., & Karm, M. (2022). “Feedback is like a tail wind”: Pre-service teachers’ and teacher educators’ perceptions of feedback. *Eesti Haridusteaduste Ajakiri. Estonian Journal of Education*, 10(1), Article 1. <https://doi.org/10.12697/eha.2022.10.1.07>
- Isaacson, P. C., & Scott, T. A. (1989). Automating the execution of student programs. *ACM SIGCSE Bulletin*, 21(2), 15–22. <https://doi.org/10.1145/65738.65741>

- Jackson, D., & Usher, M. (1997). Grading student programs using ASSYST. *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*, 335–339. <https://doi.org/10.1145/268084.268210>
- Joy, M., & Luck, M. (1998). Effective electronic marking for on-line assessment. *ACM SIGCSE Bulletin*, 30(3), 134–138. <https://doi.org/10.1145/290320.283096>
- Kakk, M. (2020). *Automaattestid andmebaaside ainele* [Bakalaureusetöö]. Tartu Ülikool, Informaatika õppekava.
- Keuning, H., Jeurig, J., & Heeren, B. (2016). Towards a Systematic Review of Automated Feedback Generation for Programming Exercises. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 41–46. <https://doi.org/10.1145/2899415.2899422>
- Kleerekoper, A., & Schofield, A. (2018). SQL tester: An online SQL assessment tool and its impact. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 87–92. <https://doi.org/10.1145/3197091.3197124>
- Kleiner, C., Tebbe, C., & Heine, F. (2013). Automated grading and tutoring of SQL statements to improve student learning. *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, 161–168. <https://doi.org/10.1145/2526968.2526986>
- L. Liikane, M. Kesa. *Arvutisõnastik*. (s.a.). Salvestatud 12. mai 2023, <https://www.keeleeveeb.ee/dict/speciality/computer/dict.cgi?word=SQL&lang=en>
- Laurie, V. (s.a.). *The Command Line in Windows: Batch file basics*. Salvestatud 12. mai 2023, <https://commandwindows.com/batch.htm>

- Li, J., & De Luca, R. (2014). Review of assessment feedback. *Studies in Higher Education*, 39(2), 378–393. <https://doi.org/10.1080/03075079.2012.709494>
- Lujan, H. L., & DiCarlo, S. E. (2006). Too much teaching, not enough learning: What is the solution? *Advances in Physiology Education*, 30(1), 17–22. <https://doi.org/10.1152/advan.00061.2005>
- Mis on mittefunktsionaalsed nõuded: Näited, määratlus, täielik juhend – visure lahendused. (s.a.). *Visure Solutions*. Salvestatud 11. mai 2023, <http://visuresolutions.com/et/blogi/mittefunktsionaalsed-n%C3%B5uded/>
- Mitrovic, A. (1998). Learning SQL with a computerized tutor. *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, 307–311. <https://doi.org/10.1145/273133.274318>
- Morales-Trujillo, M. E., & García-Mireles, G. A. (2021). Gamification and SQL: An Empirical Study on Student Performance in a Database Course. *ACM Transactions on Computing Education*, 21(1), 3:1-3:29. <https://doi.org/10.1145/3427597>
- Murry, J. P. (1990). Better Testing for Better Learning. *College Teaching*, 38(4), 148–152. <https://doi.org/10.1080/87567555.1990.10532431>
- Muuli, E., Lepp, M., Palm, R., & Luik, P. (2021). Automation of assessment and feedback in IT teaching from the teaching staff perspective. *2021 IEEE Frontiers in Education Conference (FIE)*, 1–9. <https://doi.org/10.1109/FIE49875.2021.9637290>
- Nalintippayawong, S., Atchariyachanvanich, K., & Julavanich, T. (2017). DBLearn: Adaptive e-learning for practical database course — An integrated architecture approach. *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 109–114. <https://doi.org/10.1109/SNPD.2017.8022708>

- Naur, P. (1964). Automatic grading of students' ALGOL programming. *BIT Numerical Mathematics*, 4(3), 177–188. <https://doi.org/10.1007/BF01956028>
- Plangi, S. (2014). *Automaatne programmeerimisülesannete kontrollija Tartu Ülikooli kursuse "Algoritmid ja andmestruktuurid" jaoks* [Thesis]. <https://dspace.ut.ee/handle/10062/55967>
- Prados, F., Boada, I., Soler, J., & Poch, J. (2006). A Web-based tool for teaching and learning SQL. *Congrés: 7th International Conference on Information Technology Based Higher Education and Training: ITHET 2006*. <https://acmex.udg.edu/articles/ithet2006.pdf>
- Prior, J. C., & Lister, R. (2004). The backwash effect on SQL skills grading. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, 32–36. <https://doi.org/10.1145/1007996.1008008>
- Programmeerimine—Kursused—Arvutiteaduse instituut*. (s.a.). Salvestatud 31. märts 2023, <https://courses.cs.ut.ee/2022/programmeerimine/fall>
- Reek, K. A. (1989). The TRY system -or- how to avoid testing student programs. *Proceedings of the twentieth SIGCSE technical symposium on Computer science education*, 112–116. <https://doi.org/10.1145/65293.71198>
- Romli, R., Sulaiman, S., & Zamli, K. Z. (2015). Improving Automated Programming Assessments: User Experience Evaluation Using FaSt-generator. *Procedia Computer Science*, 72, 186–193. <https://doi.org/10.1016/j.procs.2015.12.120>
- Roosalu, S. (2022). *Automaatsetidel põhinev lahendus Androidi kodutööde automaatseks hindamiseks õppetöös* [Bakalaureusetöö]. Tartu Ülikool, Informaatika õppekava.
- Sadiq, S., Orłowska, M., Sadiq, W., & Lin, J. (2004). SQLator: An online SQL learning workbench. *Proceedings of the 9th annual SIGCSE conference on Innovation and*

- technology in computer science education*, 223–227.
<https://doi.org/10.1145/1007996.1008055>
- Sepp, R. (2018). *Automaattide loomine andmebaasile Edu* [Thesis].
<https://dspace.ut.ee/handle/10062/66003>
- Siiber, M. (2015). *Aine „Andmebaasid“ automaattestimise vahend*. Tartu Ülikool, Informaatika õppekava.
- Stanger, N. (2018). *Semi-automated assessment of SQL schemas via database unit testing*. Asia-Pacific Society for Computers in Education (APSCE).
<https://ourarchive.otago.ac.nz/handle/10523/9694>
- V. Hanson, A. Tavast. *Arvutikasutaja sõnastik*. (s.a.). Salvestatud 12. mai 2023,
<https://www.keeleeveeb.ee/dict/speciality/aks/dict.cgi?word=batch&lang=en>
- Venter, M. (2020). Gamification in STEM programming courses: State of the art. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 859–866.
<https://doi.org/10.1109/EDUCON45650.2020.9125395>
- von Matt, urs. (1994). Kassandra: The automatic grading system. *ACM SIGCUE Outlook*, 22(1), 26–40. <https://doi.org/10.1145/182107.182101>
- Vorteil, V., & Laanpere, J. (s.a.). *Tarkvara arendusnõuded*. Salvestatud 11. mai 2023,
<https://web.htk.tlu.ee/digitalu/testimine/chapter/tarkvara-arendusnouded/>
- Õunmaa, M. (2019). *Automaattide loomine sessioonõppe ainele „Sissejuhatus andmebaasidesse“* [Thesis]. <https://dspace.ut.ee/handle/10062/66242>

Lisad

Lisa 1. Esimene küsitlus tudengitele

Sugu *

- ☐ Mees
- ☐ Naine

Ees- ja perenimi *

Your answer

Aine "Andmebaasid" on minu õppekavas *

- ☐ kohustuslik aine
- ☐ minu valitud moodulis kohustuslik
- ☐ valikaine
- ☐ vabaaine

Enne kursust kui heaks hindasite oma SQL keele oskusi *

- ☐ Polnud sellest keelest midagi kuulnudki
- ☐ Teadsin, et selline keel on olemas
- ☐ Olin veidi selle keelega tutvunud
- ☐ Olin seda keelt varasemalt mõnel kursusel õppinud
- ☐ Töötan/töötasin igapäevaselt selles keeles programmeerimisega

Vasta järgmisele küsimustele 1-3 alloleva pildi järgi.

	ABC ylesanne	ABC kontroll_nimi	ABC tagasiside	ABC olek	123 punktid	123 max_punktid
1	Praktikum 3	Tabel "Partiid" Kitsendus "vastavus"	ei ole olemas	VIGA	0	0
2	Praktikum 3	Tabel "Isikud" Veergude arv	on vale, peab olema 8	VIGA	0	0
3	Praktikum 3	Tabel "Isikud" Kitsendus "un_isikukood"	ei ole olemas	VIGA	0	0
4	Praktikum 3	Tabel "Isikud" Isikul "Irys"	puuduvad veerud eesnimi ja/või perenimi	VIGA	0	0
5	Praktikum 3	Tabel "Isikud" Veerg "Klubis"	ei ole olemas	VIGA	0	0
6	Praktikum 3	Tabel "Partiid" Kitsendus "ajakontroll"	ei ole olemas	VIGA	0	0
7	Praktikum 4	-	-	Hindepunkt	1	1
8	Kodutöö 3	Välisvõti "Rk_turniir_2_asula"	ei ole olemas	VIGA	0	0.1
9	Kodutöö 3	-	-	Hindepunkt	0.4	0.5

Missugused järgmistest vigadest esinevad tagasisidetabelis (märgi kõik sobivad)?

★

- ☐ Tabelis "Partiid" puudub kitsendus "vastavus"
- ☐ Tabelis "Isikud" on vale veergude arv
- ☐ Tabelis "Turniirid" puudub kitsendus "un_isikukood"
- ☐ Tabelis "Isikud" pole veerge eesnimi ja/või perenimi
- ☐ Tabelis "Klubid" puudub veerg "Asukoht"
- ☐ Tabelite "Turniirid" ja "Asulad" vahel pole välisvõtit
- ☐ Tabelit "Partiid" pole olemas

Kui selgesti saad tabeli põhjal aru, kus viga/vead on tekkinud? *

	1	2	3	4	5	
Ei saa üldse aru	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Saan väga hästi aru

Kui selgesti saad tabeli põhjal aru saadud/kaotatud punktidest? *

	1	2	3	4	5	
Ei saa üldse aru	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Saan väga hästi aru

Kas oled saanud vähemalt ühel korral automaattestiga hinnatud tagasiside? *

- ☐ Jah
- ☐ Ei

Automaatkontroll

Kui selgesti oled aru saanud Sinule antud praktikumi või kodutöö tagasisidest, mis on automaatkontrolliga kontrollitud? *

1 2 3 4 5

Ei saa üldse aru ☐ ☐ ☐ ☐ ☐ Saan väga hästi aru

Mis on Sinu jaoks jäänud tagasisidest puudu või tekitanud segadust?

Your answer

Kui selgesti saad aru Sulle antud tagasisides saadud/kaotatud punktidest automaatkontrolliga kontrollitud töödes? *

1 2 3 4 5

Ei saa üldse aru ☐ ☐ ☐ ☐ ☐ Saan väga hästi aru

Mis on Sulle tekitanud segadust punktide saamises?

Your answer

Kodutööd 1 ja 2 kontrolliti käsitsi, praktikum 3 ja koduöö 3 automaatselt.

Mil määral tunned, et said praktikum 3 ja kodutöö 3 tagasiside kiiremini kui kodutöödele 1 ja 2?

	1	2	3	4	5	
Sain palju aeglasemalt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sain palju kiiremini

Kodutööd 1 ja 2 kontrolliti käsitsi, praktikum 3 ja koduöö 3 automaatselt.

Mil määral oli praktikum 3 ja kodutöö 3 tagasiside ja punktid selgemini arusaadavad kui kodutöödel 1 ja 2?

	1	2	3	4	5	
Palju vähem arusaadavad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Palju selgemini arusaadavad

Kas soovid midagi automaatkontrollide kohta lisada?

Your answer

Lisa 2. Teine küsitlus tudengitele

Sugu *

- ☐ Mees
- ☐ Naine
- ☐ Ei soovi vastata

Ees- ja perenimi *

Your answer

Aine "Andmebaasid" on minu õppekavas *

- ☐ kohustuslik aine
- ☐ minu valitud moodulis kohustuslik
- ☐ valikaine
- ☐ vabaaine

Kas oled saanud vähemalt ühel korral automaattestiga hinnatud tagasiside? *

- ☐ Jah
- ☐ Ei

Automaatkontroll

Kas tagasiside on vahepeal arusaadavamaks muutunud võrreldes eelmise küsitluse ajal olnud tagasisidega? *

1 2 3 4 5

Pole üldse muutunud ☐ ☐ ☐ ☐ ☐ Oluliselt arusaadavamaks

Kuna tagasisidesse lisati uusi asju, siis palun vasta järgmisele küsimusele alloleva pildi abil.

Praktikum 7

Oigesti tehtud: 57

Maksimum oiged: 64

Kui arusaadav on pildil olev tagasiside? *

1 2 3 4 5

Väga arusaamatu ☐ ☐ ☐ ☐ ☐ Väga arusaadav

Kui selgesti oled aru saanud Sinule antud praktikumi või kodutöö tagasisidest, mis on automaatkontrolliga kontrollitud? *

1 2 3 4 5

Pole üldse aru saanud

☐☐☐☐☐

Olen väga hästi aru saanud

Mis on Sinu jaoks jäänud tagasisidest puudu või tekitanud segadust?

Your answer

Kui selgesti saad aru Sulle antud tagasisides saadud/kaotatud punktidest automaatkontrolliga kontrollitud töödes?

1 2 3 4 5

Ei saa üldse aru

☐☐☐☐☐

Saan väga hästi aru

Mis on Sulle tekitanud segadust punktide saamises?

Your answer

Automaatkontroll tudengitele

Millist automaatkontrolli lahendust sooviksid?

- ☐ Automaatkontroll, mida saaksid enne lahenduse esitamist ise kasutada, et kontrollida oma lahenduse õigsust
- ☐ Automaatkontrolli Moodle'i keskkonnas, kus saaksid kohe lahenduse tulemuse ja punktide arvu
- ☐ Pole vahet

Kui Sul ise oleks võimalik kasutada automaatkontrolli, kas see motiveeriks Sind rohkem pingutama ja/või vigu parandama?

	1	2	3	4	5	
Ei motiveeriks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Kindlasti motiveeriks

Kui automaatkontrolliga saaks oma lahendust kontrollida õppija, mitu korda võiks * ta saada ülesannet esitada?

Your answer

Missugust tagasisidet eelistad enda kodutööde lahendustele saada?

- ☐ Ainult õppejõu tagasiside
- ☐ Mõlemat, nii õppejõult kui automaatkontrolli oma
- ☐ Ainult automaatkontrolli tagasiside
- ☐ Pole vahet

Kas soovid midagi automaatkontrollide kohta lisada?

Your answer

Lisa 3. Küsitlus praktikumijuhendajatele

Nimi *

Your answer

Kelle automaatkontrolle oled kasutanud kodutööde kontrollimiseks? *

☐ Martti

☐ Karel

☐ Karl

☐ Other: _____

Kui kasutasid teisi automaatkontrolle peale Martti, siis miks?

Your answer

Millistes kodutöödes oled kasutanud Martti koostatud automaatkontrolli? *

☐ Praktikum 3

☐ Praktikum 4

☐ Kodutöö 3

☐ Kodutöö 4

☐ Praktikum 7

☐ Kodutöö 5

☐ Praktikum 9 ehk Edu

☐ Mitte üheski

Kui arusaadav on olnud automaatkontrolli tagasiside Sinu jaoks?

	1	2	3	4	5	
Väga arusaamatu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga arusaadav

Kui selgesti oled aru saanud, kus vead on tekkinud?

	1	2	3	4	5	
Pole üldse aru saanud	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Olen täielikult aru saanud

Kuidas hindad automaatkontrollide kasutusmugavust?

	1	2	3	4	5	
Väga ebamugav	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga mugav

Kuidas hindad automaattestide kasutuskiirust?

	1	2	3	4	5	
Väga aeglane	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga kiire

Kas oled kasutanud batchi, et kontrollida kodutöid? *

- ☐ Jah
- ☐ Ei

Kui töökindel on batch? *

	1	2	3	4	5	
Ei tööta kunagi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Töötab alati korrektselt

Miks Sa ei ole kasutanud batchi?

Your answer

Kui arusaadav on batchi kasutamise juhend? *

	1	2	3	4	5	
Väga keeruline	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga lihtne

Mis on kõige keerulisem?

Your answer

Palun hinda, kui olulised on Sinu jaoks järgnevad automaatkontrolli omadused. *

	1	2	3	4	5
Kiirus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Töökindlus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Arusaadav tagasiside	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Arusaadav punktisüsteem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kasutuslihtsus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kui automaatsed peaksid automaatkontrollid Sinu arvates olema? *

- ☐ Poolautomaatne kontroll. Sa ise käivitad kontrollid, näed tudengi andmebaasi ja saad lisada tudengile täpsustavat tagasisidet. Võtab rohkem aega, aga tagasiside on vajadusel isikupärasem.
- ☐ Täielikult automaatne kontroll. Õppejõul aega ei kulu, tudeng kontrollib ise automaatkontrolli abil. Tagasiside on ainult automaatne ja õppejõu tagasiside enamjaolt puudub.

Kui oleksid täiesti automaatsed kontrollid kas Moodle'is või mujal, kui mitu võimalust peaks tudengil olema kodutöö esitamiseks? *

- ☐ 1
- ☐ 2
- ☐ 3-4
- ☐ 5-6
- ☐ Lõpmatu

Muid ettepanekuid ja ideid automaatkontrollide kohta

Your answer

Lisa 4. Automaatkontrollide kasutusjuhend

1. Eeltöö

- 1.1. Loo C kaust nimega “Temp”
- 1.2. Lae alla zip fail Moodle’ist
- 1.3. Paki lahti enda valitud kohta
- 1.4. Kopeeri muutujad.csv ja ülejäänud tekstifailid kausta “Temp”

2. Ava ope_test.sql tekstiredaktoris ja kontrolli alguses on muutuja „version“ kuhu kirjuta vastava kontrolli number.

2.1. Versioonid on seletatud skriptis

```
create or replace procedure kontroll() as $kontroll$
declare version int := 6;
/*
Siin ma arad, mis ylesandeid kontrollitakse. Koik eelnevad kontrollivad ka eelmisi.
0 - praktikum 9 ehk EDU

2 - praktikum 3
3 - praktikum 4 ja kodutoo 3
4 - kodutoo 4
5 - praktikum 7
6 - kodutoo 5
7 - praktikum 10
8 - kodutoo 6
9 - praktikum 11
*/
```

3. Lisaks õppejõududel, kes soovivad C:\TEMP kausta asemel kasutada muud kausta, siis selle muutmise võimalus on ka olemas

```
/*
Jarmisega saad valida, kust kaustas on sul txt failid ja kuhu tekib tulemus.csv fail.
OLULINE - vaata, et peale kausta nime poleks kaldkriipsu \
default on - 'C:\TEMP'
*/
folder_path varchar(255) := 'C:\TEMP';
```

3.1 Lisaks saab muuta failide sisselugemis delimiteri ja väljastamise delimiter.

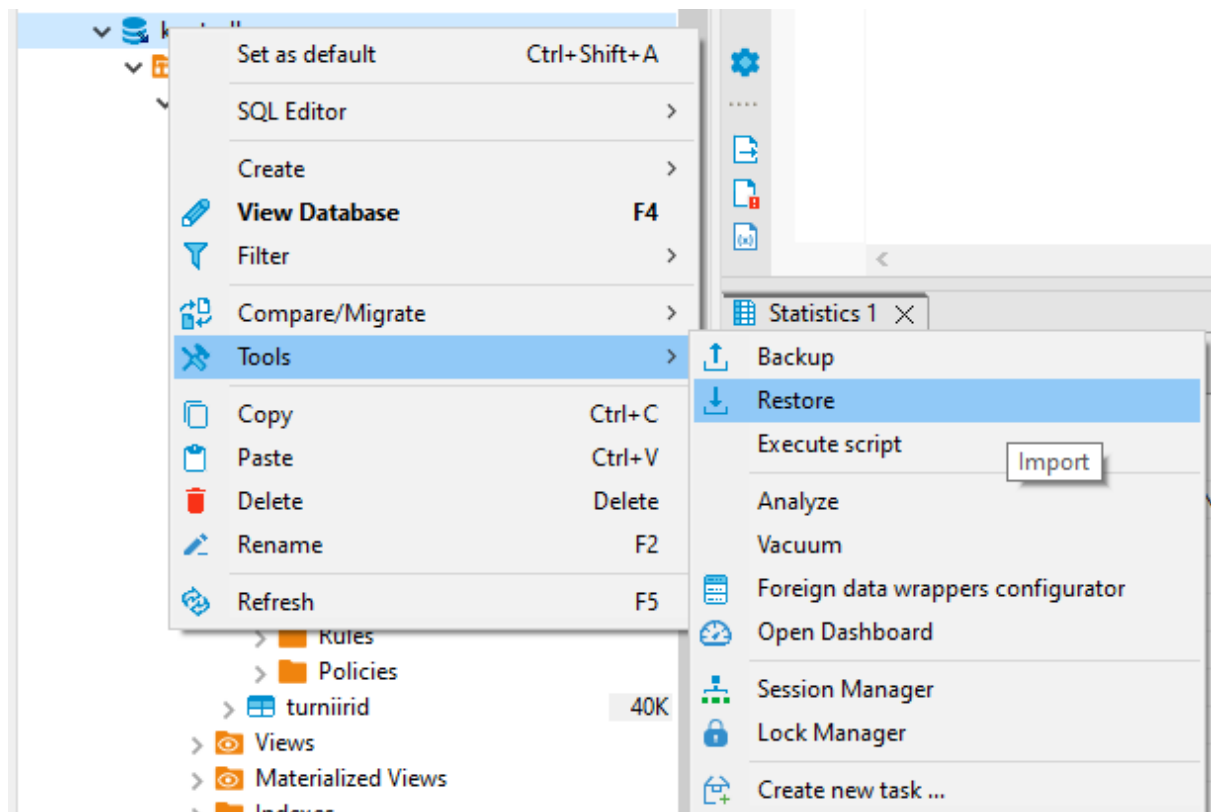
```
/* Jargmisega saad valida, mis delimiteriga tuleb tul
tulemus_andmed_delimiter varchar(10) := ',';
--tulemus_andmed_delimiter varchar(10) := E'\t';

/* Jargmised kaks ma aravad, mis DELIMITER on sissetul
txt_lugemis_andmed_delimiter varchar(10) := '\t';
csv_lugemis_andmed_delimiter varchar(10) := ',';
```

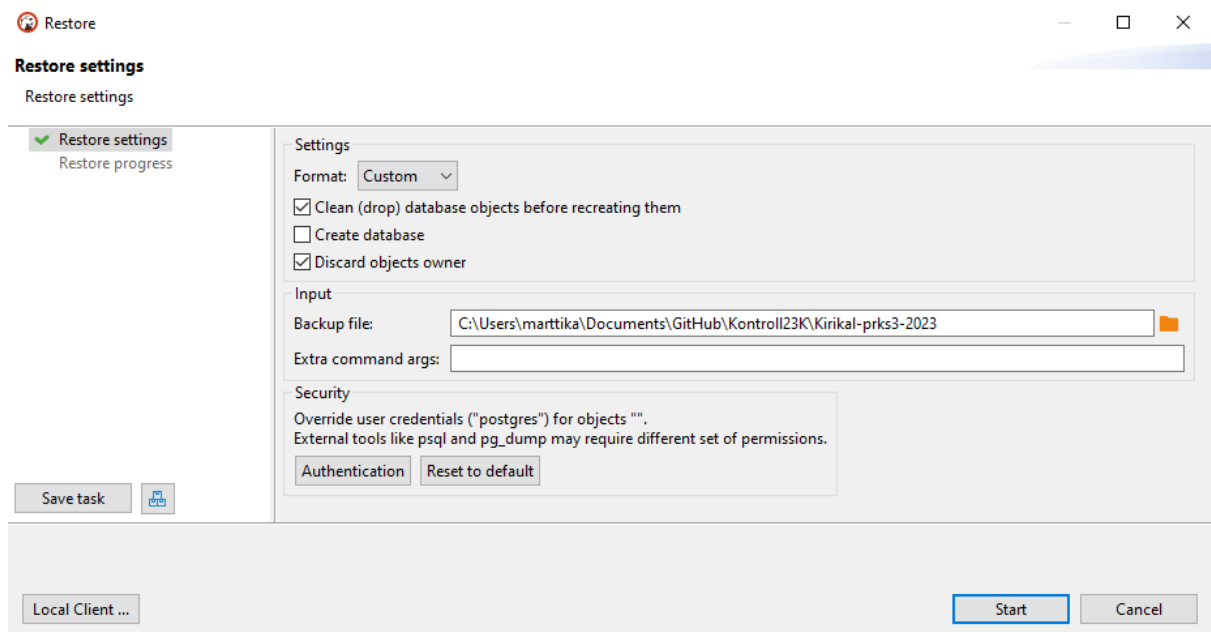
4. Käivita Dbeaver

5. Lae alla tudengi andmebaasi backup

5.1. Restore, järgmises punktis on restore sätted



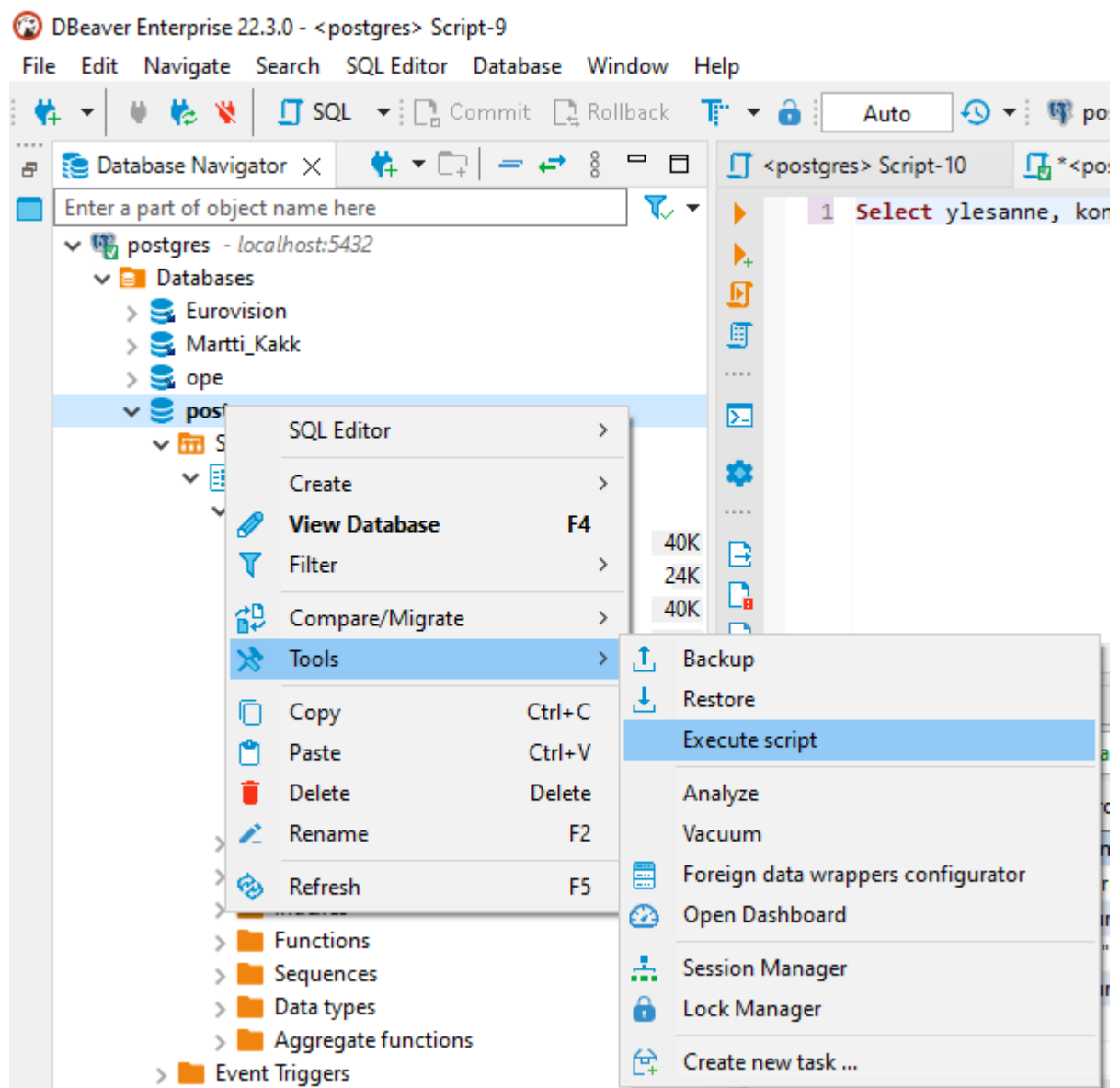
5.2. Restore sätted ja vailida tudengi backup fail



6. Kontroll skripti käivitamine

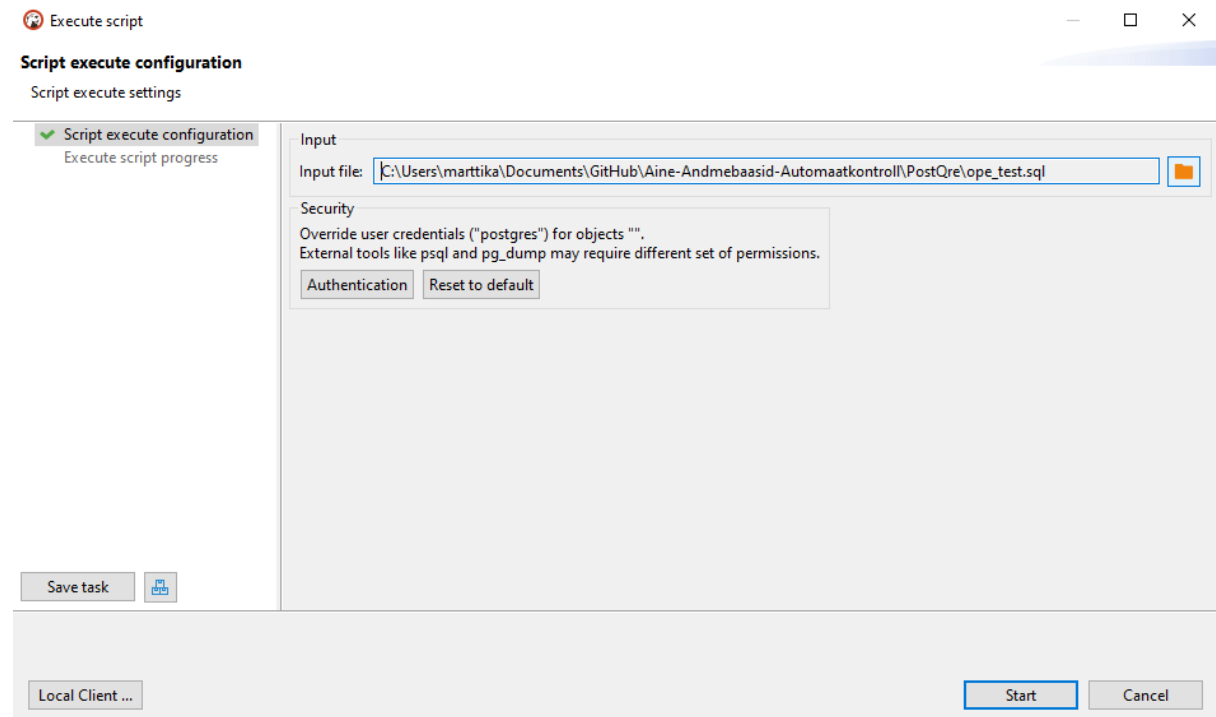
6.1. Vajuta parema hiireklõpsuga andmebaasil

6.2. Vali Tools -> Execute script



7. Avanenud aknas vali skriptifaili asukoht

7.1. Vajuta Start ja script hakkab tööle



8. Tulemuse vaatamine

8.1. Sulge skripti aken

8.2. Tudengile tagasiside andmiseks tekib TEMP kausta fail tulemuse.csv, avades saab tabelina kopeerida

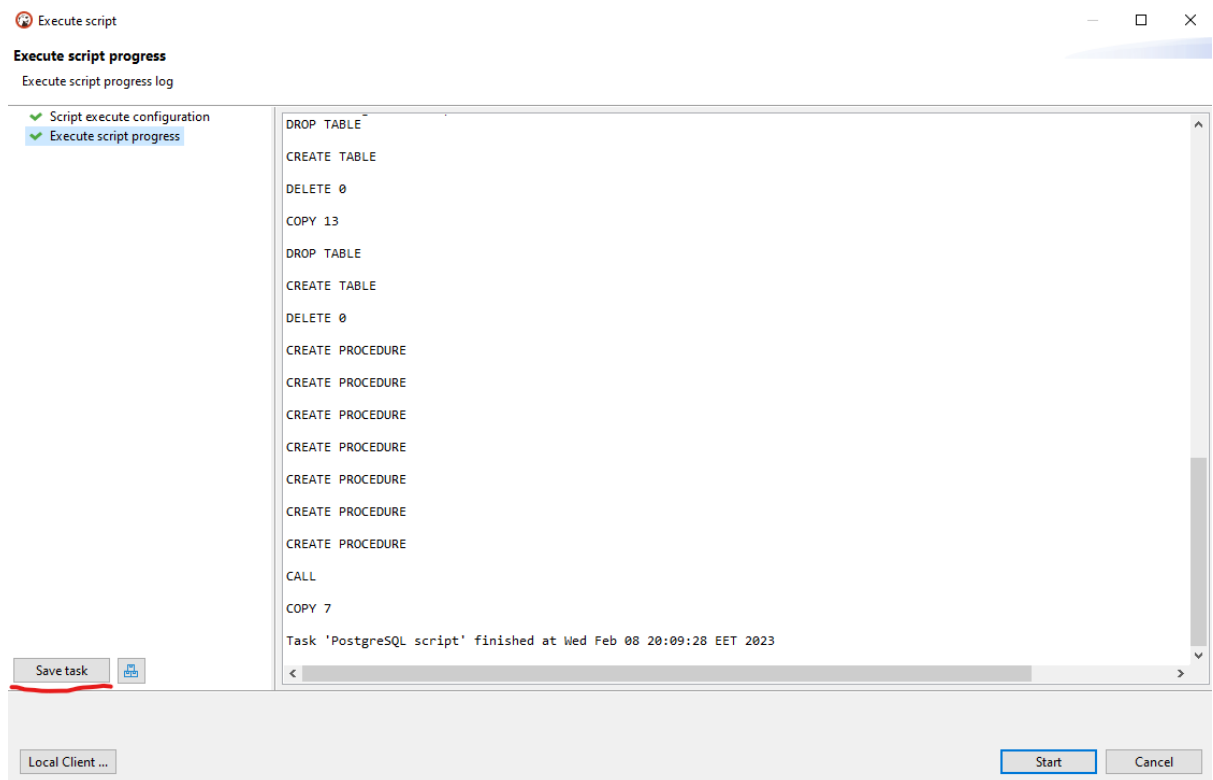
8.3. Kui soovid Dbeaverist tulemust näha, siis käivita Dbeaveri aknas järgnev päring:

```
Select ylesanne, kontrolli_nimi, tagasiside, olek, punktid, max_punktid From  
staatus where olek in ('VIGA','Hindepunktid') or ylesanne = 'Tudeng' order by  
jr asc;
```

Järgmisel lehel Taski tegemise õpetus.

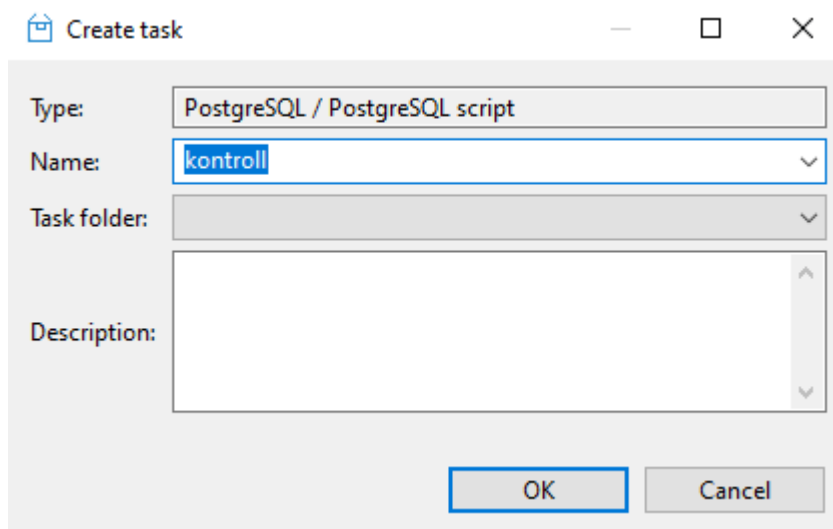
9. Kui oled skripti käivitamise aknas

9.1. All vasakul on Save task



5.2 Pane taskile nimeks kontroll

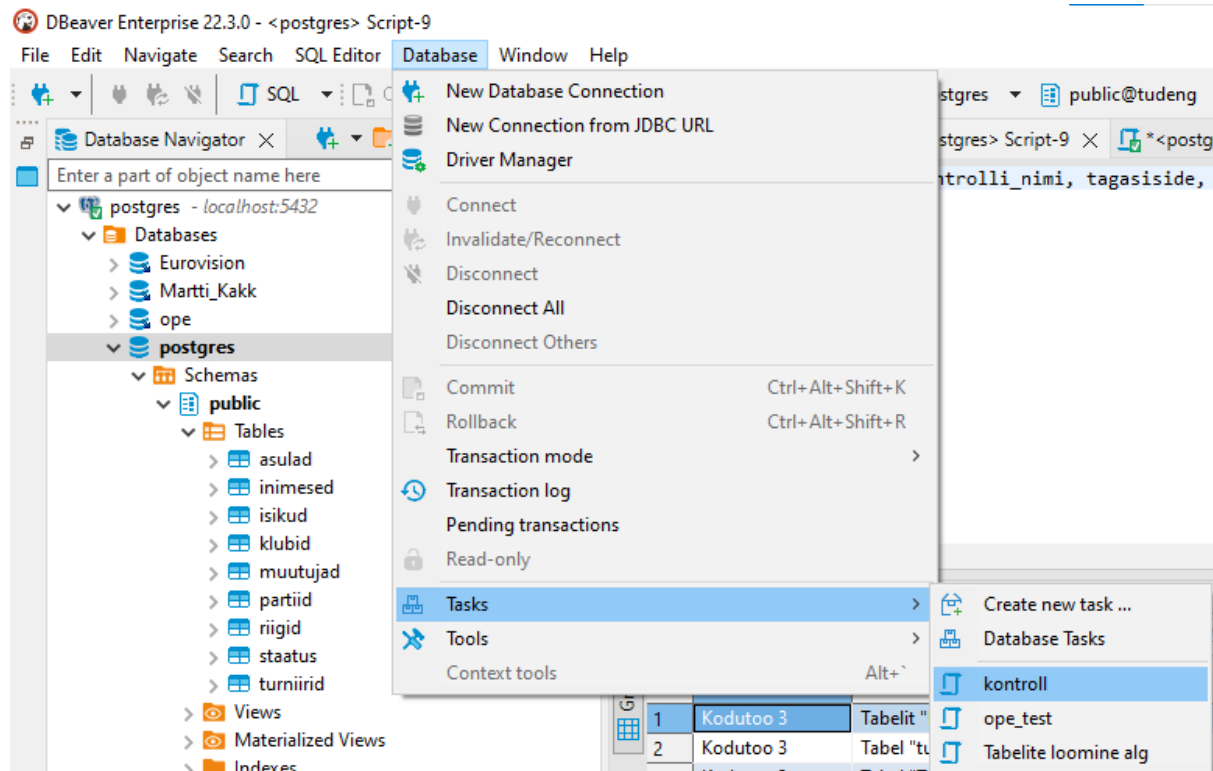
5.3 OK



10. Taski käivitamine

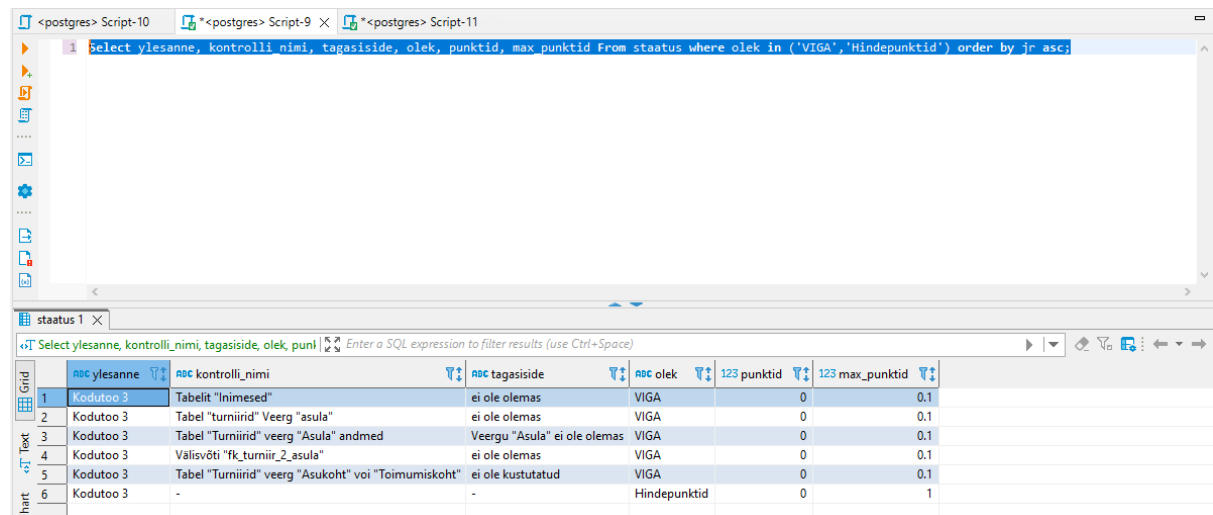
10.1. Tegumirealt Database

10.2. Tasks -> kontroll



6.3 Tulemuse nägemiseks käivita sama käsklus, mis on punktis 4.2

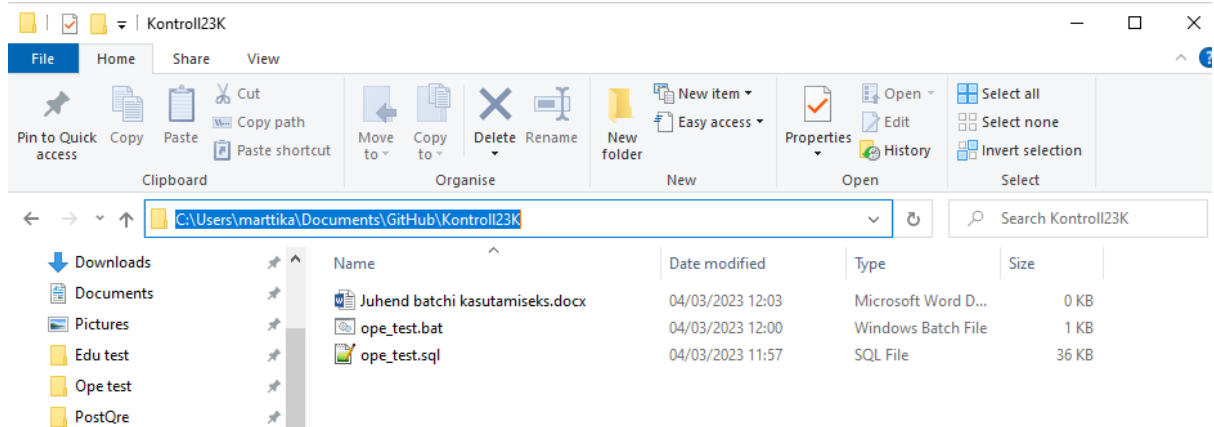
Select ylesanne, kontrolli_nimi, tagasiside, olek, punktid, max_punktid **From** staatus **where** olek **in** ('VIGA','Hindepunktid') **or** ylesanne = 'Tudeng' **order by** jr **asc**;



Lisa 5. Pakkfaili kasutusjuhend

Käesoleva pakkfaili kasutamisejuhend töötab ainult *Windowsi* operatsioonisüsteemis.

1. Loo kaust, kuhu paned skriptifaili `ope_test.sql` ja pakkfaili `ope_test.bat`



2. Ava `ope_test.bat` teksti redaktoriga ja muuda endale vastavalt seadeid.

- a. Rida 1 `src=...` siia läheb loodud kausta failiteekond

```
1 SET src=C:\Users\
```

- b. Rida 11 asenda “andmebaasi_nimi” enda andmebaasi nimega, mida kasutad üliõpilase töö kontrollimiseks

```
11 psql -U postgres -d andmebaasi_nimi -f lisa_tabelite_kustutamine.sql
```

- c. Rida 12 asenda “andmebaasi_nimi” enda andmebaasi nimega, mida kasutad üliõpilase töö kontrollimiseks

```
12 pg_restore -U postgres -d andmebaasi_nimi -c male.backup
```

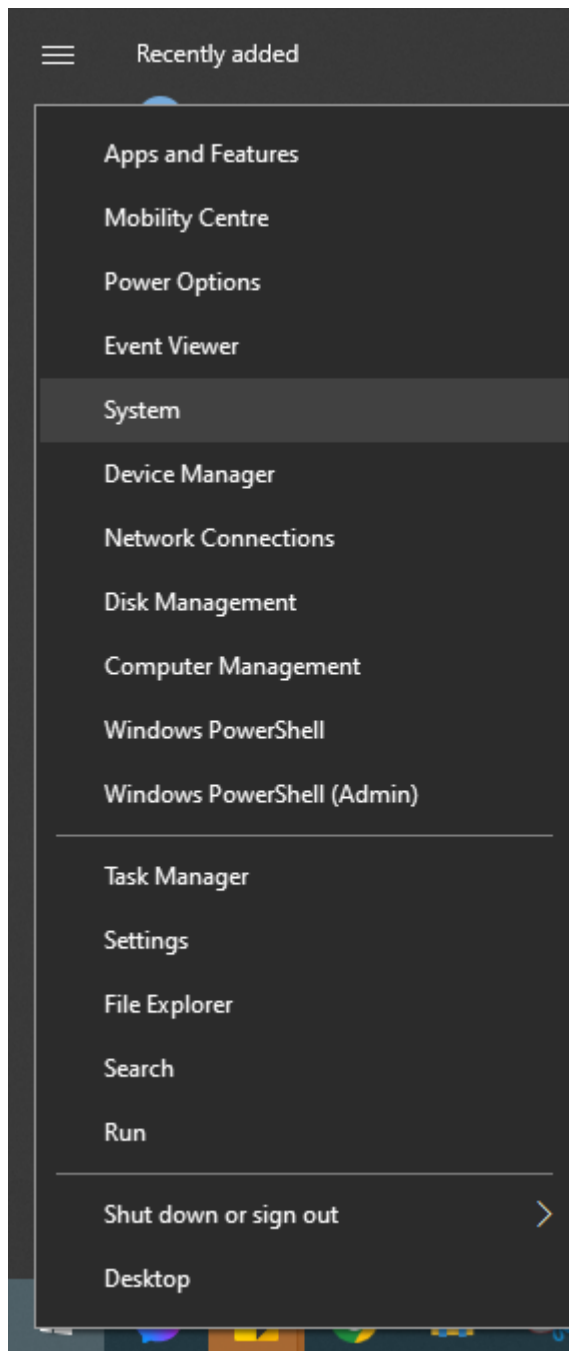
- d. Rida 13 asenda “andmebaasi_nimi” enda andmebaasi nimega, mida kasutad üliõpilase töö kontrollimiseks

```
13 psql -U postgres -d andmebaasi_nimi -f ope_test.sql
```

3. Nüüd sa pead lisama `psql`-li endale `PATHi` peale, mis nõuab Admini õiguseid

- a. Jälgi lingi juhist <https://www.java.com/en/download/help/path.html> või siin juhistes

- b. Vajuta Start peale parema hiirenupuga ja vali System



c. Vali paremalt poolt Advanced system settings

This page has a few new settings

Some settings from Control Panel have moved here, and you can copy your PC info so it's easier to share.

Related settings

[BitLocker settings](#)

[Device Manager](#)

[Remote desktop](#)

[System protection](#)

[Advanced system settings](#)


[Rename this PC \(advanced\)](#)

Help from the web

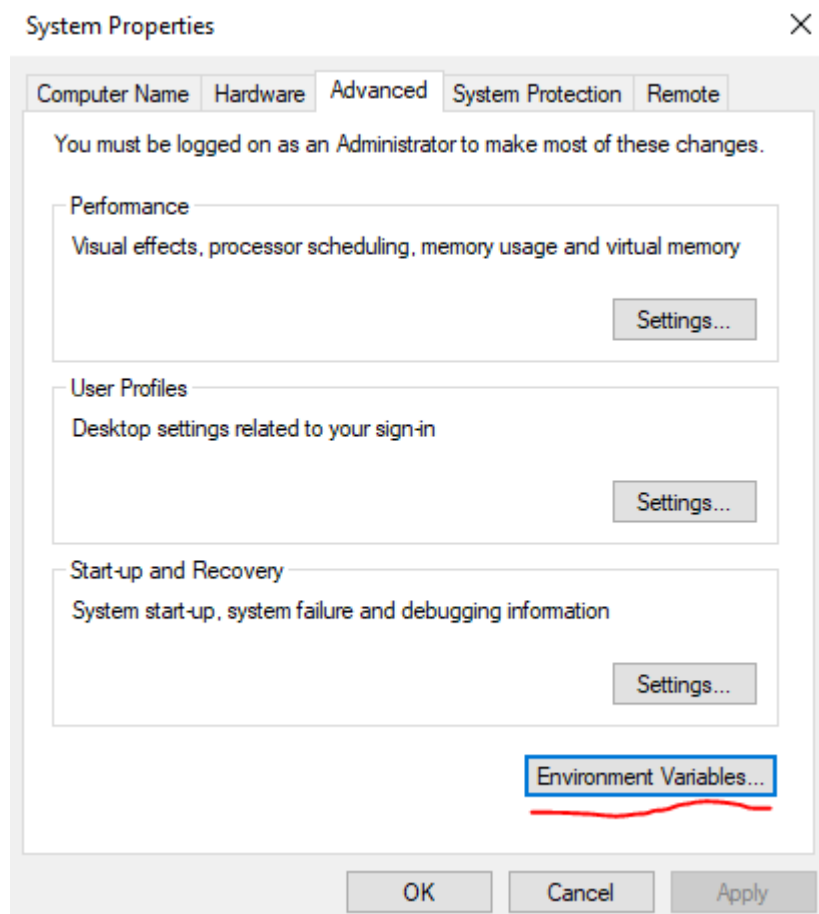
[Finding out how many cores my processor has](#)

[Checking multiple Languages support](#)

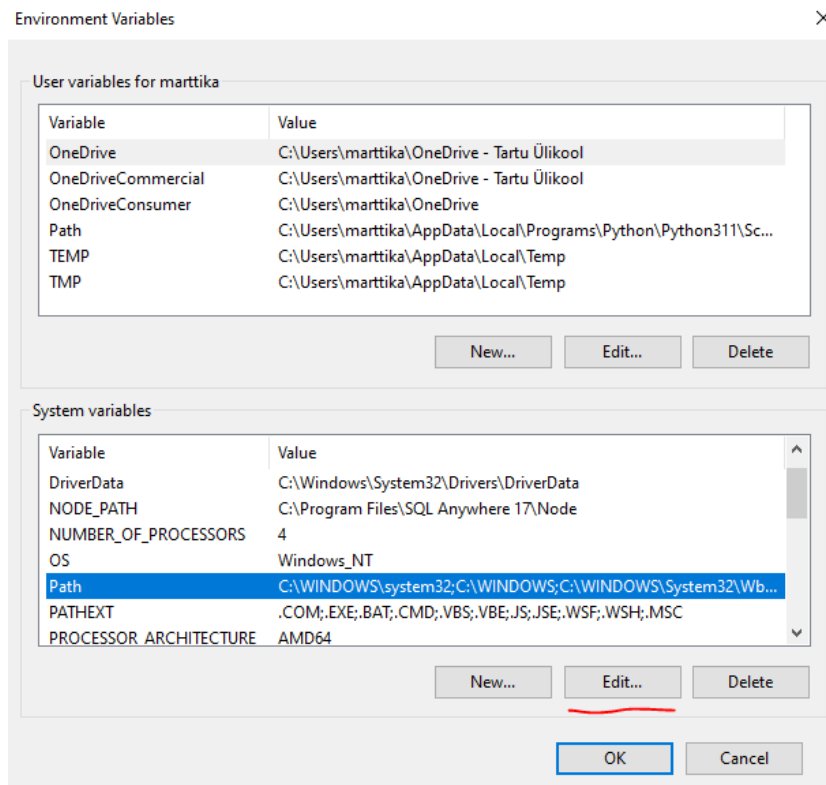
 [Get help](#)

 [Give feedback](#)

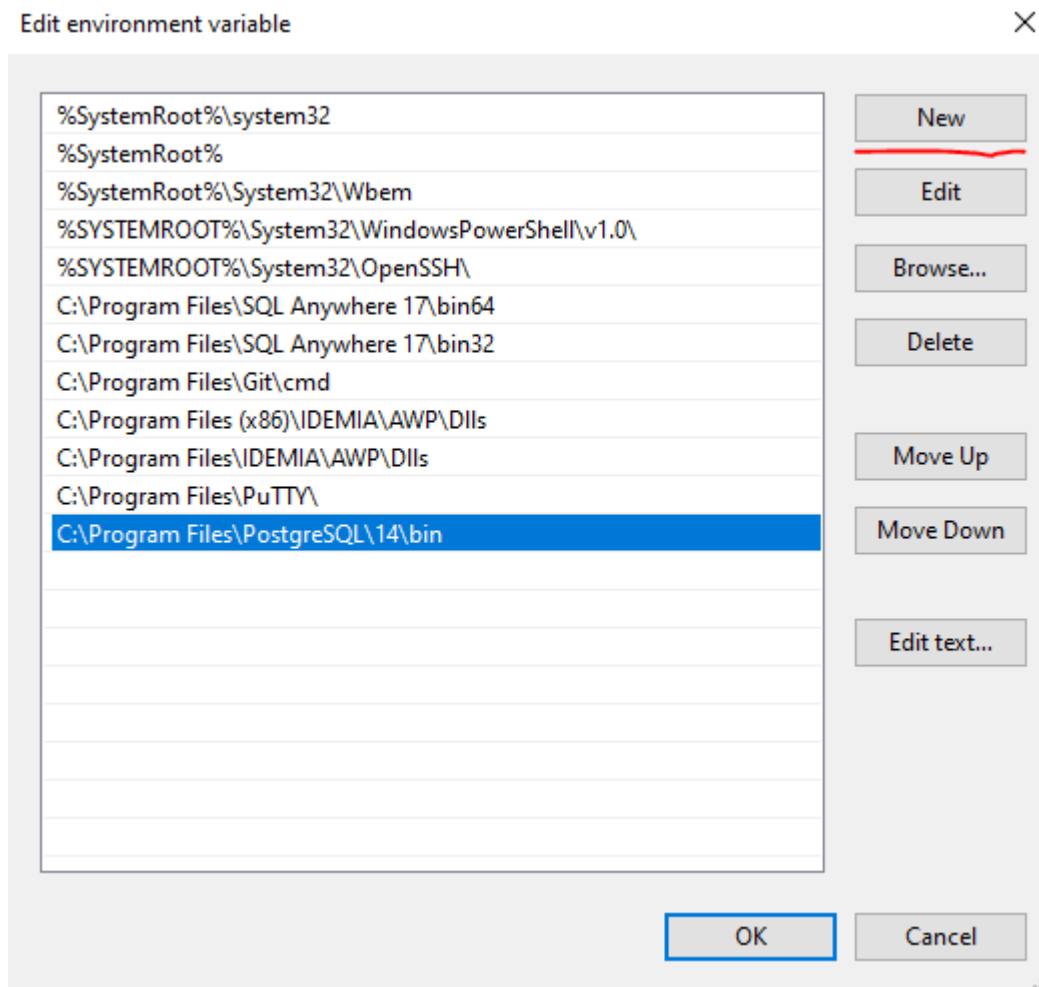
d. Ava Environment Variables



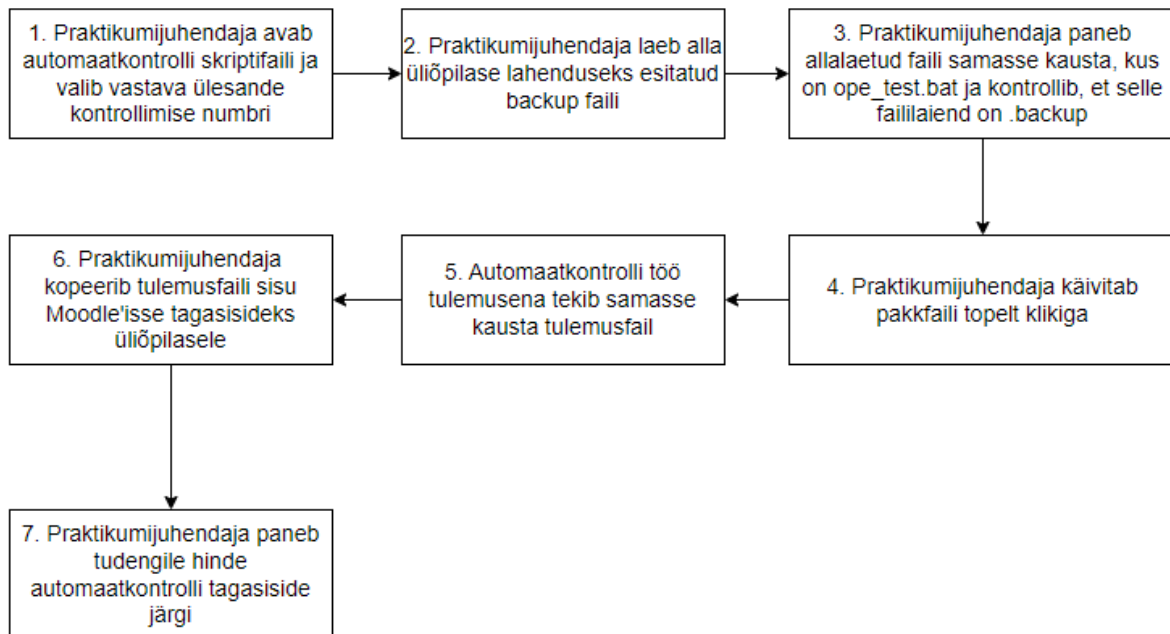
- e. Alumisest kastist System Variables otsi üles “Path” ja vajuta Edit



- f. Vajuta New ja tekib uus rida
- g. Kopeeri uuele reale Sinu PostgreSQL versiooni bin kaust
- h. Peaks olema sarnane siniseks märgitud reaga. (Sul võib olla 14 asemel 15)

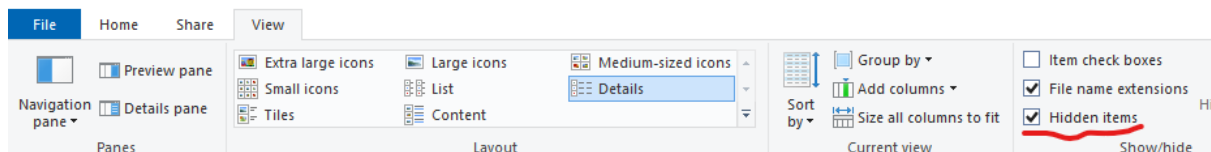


4. Nüüd mine kausta, kuhu panid ope_test.sql ja ope_test.bat failid.
5. Edaspidi tuleb praktikumijuhendaja töövoog
 - a. Pakkfaili käivitades küsitakse kasutajalt 3 korda parooli andmebaasiga ühendamiseks

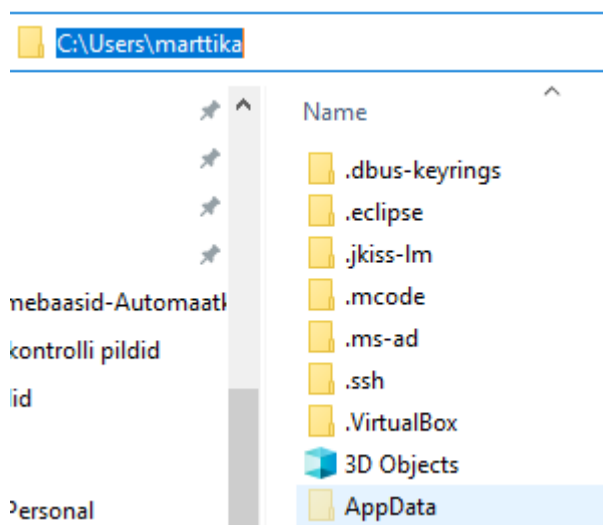


Kui ei soovi igakord parooli sisestada, siis järgi järgmist juhendit.

1. Mine kausta C:\Users\kasutaja_nimi, kus asenda “kasutaja_nimi” enda kasutajaga
2. Tee nähtavaks peidetud failid, ülevalt realt “View” ja vaata, et sul oleks “Hidden Items” kastike märgitud

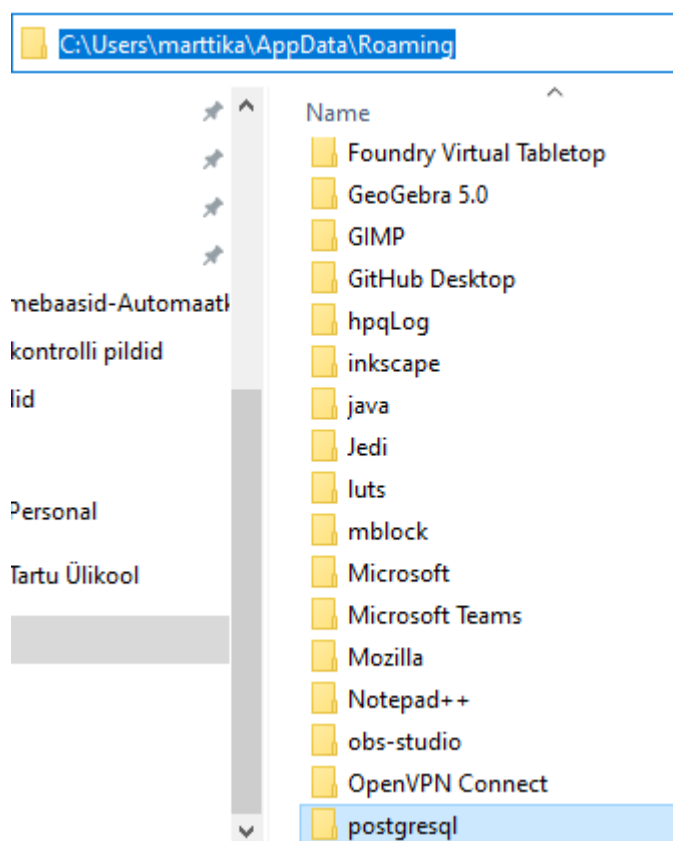


3. Ava kaust AppData



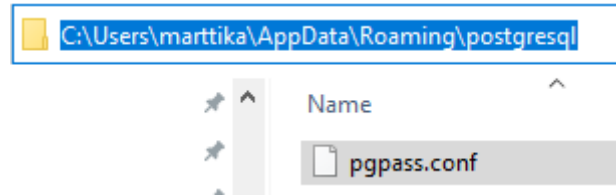
4. Ava Roaming

5. Ava postgresql või selle puudumisel loo selline kaust



6. Pane postgresql kausta fail pgpass.conf ja ava see tekstiredaktoris

Pgpass.conf kasutatakse selleks, et saaks käsurealt andmebaasi sisselogida ja ei peaks igakord sisestama vastava andmebaasi salasõna.



7. Muuda pgpass.conf faili vastavalt enda seadistustele, kus:

Host:port:database_name:username:password

Mul on: localhost:5432:kontroll:postgres:sql

Kontroll – vaheta enda andmebaasi nime vastu

Sql – vaheta enda andmebaasi salasõna vastu

Minul on :

```
1 localhost:5432:kontroll:postgres:sql
```

Seletus, mida pakkfail teeb:

```
1 SET src=C:\sinu faili teekond
2 FOR /F "eol=| delims=" %%I IN ('DIR %src%\*.backup /A-D /B /O-D /TW 2^>nul') DO (
3     SET "NewestFile=%%I"
4     GOTO FoundFile
5 )
6 ECHO No *.backup file found!
7 pause
8 GOTO :EOF
9
10 :FoundFile
11 ren %NewestFile% male.backup
12 psql -U postgres -d andmebaasi_nimi -f lisa_tabelite_kustutamine.sql
13 pg_restore -U postgres -d andmebaasi_nimi -c male.backup
14 psql -U postgres -d andmebaasi_nimi -f ope_test.sql
15 copy c:\TEMP\tulemus.csv . /Y
16 pause
```

- 1) SET src=... paneb paika, kus kaustas on sul pakkfaili ope_test.bat ja kuhu pannakse üliõpilase backup fail
- 2) FOR /F "eol=.... leiab backup faili
- 3) SET "NewestFile=%%I" - paneb muutujale NewestFile väärtuseks leitud backup nime
- 4) GOTO FoundFile - käivitab FoundFile funktsiooni
- 5) ECHO juhul kui ei leia backup faili, siis lõpetab töö
- 6) FoundFile funktsioon
- 7) ren %NewestFile% male.backup – Muudab leitud failinime male.backup nimeks
- 8) psql -U postgres -d andmebaasi_nimi -f lisa_tabelite_kustutamine.sql
Ühendab andmebaasiga ja käivitab skriptifaili lisa_tabelite_kustutamine.sql
- 9) pg_restore -U postgres -d andmebaasi_nimi -c male.backup
Ühendab andmebaasiga ja taastab andmebaasi üliõpilase backupiga
- 10) psql -U postgres -d andmebaasi_nimi -f ope_test.sql
Ühendab andmebaasiga ja käivitab automaatkontrolli skriptifaili ope_test.sql
- 11) copy c:\TEMP\tulemus.csv . /Y – Kopeerib tulemuse samasse kausta, kus on ope_test.bat
- 12) pause – Kui pakkfaili töö jõuab lõppu, siis aken ei lähe kohe kinni.

Lisa 6. Automaakontrolli failid

<https://github.com/nuubis/Aine-Andmebaasid-Automaatkontroll/tree/master/PostQre>

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Martti Kakk

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

Automaatkontrollide arendus ja tagasiside aines Andmebaasid

mille juhendajad on Piret Luik ja Vambola Leping

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.

Martti Kakk

16.05.2023