

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Sten Ärm 179322IAIB

**LÄHTEKOODIREDAKTORIS NOTEPAD++
SQL PROGRAMMEERIMIST LIHTSUSTAVA
PISTIKPROGRAMMI TÄIUSTAMINE**

Bakalaureusetöö

Juhendaja: Erki Eessaar, PhD

Tallinn 2025

Autorideklaratsioon

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

Autor: Sten Ärm

15.05.2025

Annotatsioon

Käesoleva bakalaureusetöö eesmärk oli edasi arendada ja täiustada teksti- ja lähtekoodiredaktorile Notepad++ loodud pistikprogrammi NppDB. Varasemad tööd olid loonud unikaalse funktsionaalsusega pistikprogrammi SQL-koodi eelkontrolliks, kuid selle kasutusmugavus ja funktsionaalsuste hulk jäid alla kaasaegsetele SQL-haldusvahenditele. Seetõttu keskendus edasiarendus kahele eesmärgile: esiteks programmi kasutusmugavuse oluline parandamine ja teiseks uute, väärtust lisavate funktsionaalsuste implementeerimine.

Töö eesmärgi täitmiseks selgitati läbi põhjaliku analüüsi välja pistikprogrammile vajalikud ja mõistlikud funktsionaalsused. Analüüs hõlmas üheteistkümne erineva SQL-lausete käivitamise ja haldamise tööriista võrdlust ning küsitluse teel tagasiside kogumist seitsmelt andmebaasikeelte aktiivselt kasutajalt, et mõista, milliseid täiendusi nemad vajalikuks peavad. Samuti võeti arvesse varasemates lõputöödes välja toodud täiendusvajadusi ning uuriti tehisintellekti vestlusrobotite kaasamist SQL-vigade parandamisse.

Arendustöö viidi läbi iteratiivsel viisil. Uusi lahendusi testisid ja valideerisid nii töö autor iseseisvalt kui ka koos juhendajaga iganädalastel kohtumistel. Arendustsükli lõpus koguti täiendatud pistikprogrammi kohta tagasisidet ka samadelt kasutajatelt, kes osalesid analüüsis.

Käesoleva töö tulemusena valminud tarkvara on avatud lähtekoodiga. See avaldati MIT litsentsiga ja on avalikult kättesaadav GitHub'ist aadressilt: <https://github.com/stenarm/NppDB>

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 8 peatükki, 8 joonist, 19 tabelit.

Abstract

Enhancement of a Plugin that Simplifies SQL Programming in the Notepad++ Source Code Editor

The primary aim of this bachelor's thesis was to further develop and improve NppDB, a plugin created for the Notepad++ text and source code editor. The NppDB plugin, building upon prior thesis work of students of Tallinn University of Technology, offered a unique functionality for pre-checking and executing SQL code. However, it lagged behind contemporary SQL management tools such as DataGrip, DBeaver, DbSchema, and DbVisualizer in terms of user-friendliness and the breadth of features. This development initiative was therefore intended to achieve two main objectives: firstly, to significantly enhance the program's user-friendliness, and secondly, to introduce new functionalities that would provide users with greater value and a more comprehensive toolset.

To achieve these objectives, a systematic approach was adopted, beginning with a thorough analysis to identify necessary and reasonable functionalities for the plugin. This analytical phase was multifaceted. It involved a comparative study of eleven existing tools for executing and managing SQL queries, assessing their features related to SQL execution and validation, and their overall usability. Concurrently, crucial feedback was gathered from seven active users of database languages through a web-based questionnaire. This survey aimed to understand their expectations, identify pain points with the existing NppDB, and pinpoint functionalities they deemed essential for an improved experience. Furthermore, development needs and suggestions identified in the previous theses concerning NppDB were carefully considered and integrated into the planning. An additional area of investigation within this phase was the potential for integrating Artificial Intelligence (AI) by leveraging Large Language Models (LLMs) like ChatGPT, Gemini, and Claude to assist users in SQL error correction through effective prompt generation.

The development process itself was structured around an iterative software development methodology, adhering to agile principles. Following the pre-analysis, a release plan was formulated in collaboration with the thesis supervisor, who acted as the product owner. Tasks were prioritized, and complexity was estimated using the Fibonacci sequence. The

development cycle consisted of several iterations, typically lasting 6-8 days, each beginning and concluding with review meetings with the supervisor. This iterative approach allowed for flexibility, such as addressing unforeseen issues and changing iteration plans if deemed necessary. New functionalities were implemented, and existing ones were refined based on the established plan. A key focus was on improvements like enabling SQL statement checking without an active database connection and enhancing the user interface for managing database connections and viewing query results. The integration of AI-assisted SQL error diagnosis, by generating contextual prompts for LLMs, was a notable new feature.

Testing and validation were integral to each development iteration. The author conducted independent testing to ensure the correctness and completeness of new solutions. These solutions were further validated during weekly meetings with the supervisor, who provided critical feedback and identified potential edge cases. At the end of the overall development cycle, the enhanced NppDB plugin was presented to the same group of seven validators who had participated in the initial analysis phase. Their feedback on the usability and utility of the newly implemented features was collected through a subsequent questionnaire, providing valuable insights into the practical impact of the enhancements.

The software developed as a result of this work is an open-source Notepad++ plugin with improved usability and new features, including AI-assisted SQL error troubleshooting. It has been released under the MIT license and is publicly available on GitHub at: <https://github.com/stenarm/NppDB>, allowing for community access and further contributions. The thesis details the analytical, developmental, and validation processes undertaken to achieve these enhancements, providing a comprehensive account of the improvements made to the NppDB plugin.

The thesis is written in Estonian and contains 28 pages of text, 8 chapters, 8 figures, and 19 tables.

Lühendite ja mõistete sõnastik

.NET	Microsofti arendatav avatud lähtekoodiga programmeerimisraamistik
ANTLR4	Parseri generaator (<i>ANother Tool for Language Recognition</i>) [1]
C#	Objektorienteeritud programmeerimiskeel
MS Access	Microsofti loodud SQL-andmebaasisüsteem, mis on mõeldud peamiselt lauaarvutitele ja väiksematele võrgukeskkondadele [2]
NppDB	Notepad++ pistikprogramm, mis võimaldab ühenduda erinevate andmebaasidega, et redaktoris SQL lauseid käivitada ja kontrollida [3]
NuGet Package	Paketid .NET rakendustes, mis lihtsustavad oluliselt koodi ja funktsionaalsuse taaskasutamist ning sõltuvuste haldamist [4]
Pistikprogramm	Väiksem programm, mis lisab suuremale programmile uusi funktsionaalsuseid [5]
PostgreSQL	Avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem [6]
SQL	<i>Structured Query Language</i> , relatsioonilise andmemudeli alusel väljatöötatud andmebaasikeel [7]

Sisukord

1	Sissejuhatus	10
1.1	Taust ja probleem	10
1.2	Ülesandepüstitus	11
1.3	Töö struktuur	11
2	Metoodika	13
2.1	Töö protsess	13
2.1.1	Eelanalüüs	13
2.1.2	Väljalaske plaani koostamine	16
2.1.3	Tarkvaraarendus koos jooksva tagasiside küsimisega	16
2.2	Kasutatud tööriistad	17
3	Olemasolevate SQL lausete käivitamist ja kontrollimist võimaldavate programmide analüüs	18
3.1	Funktsionaalsused ja kasutusmugavused seoses SQL lausete käivitamisega	18
3.2	Funktsionaalsused ja kasutusmugavused seoses SQL lausete kontrollimisega	18
3.3	Valideerijate hinnang pistikprogrammile NppDB enne täienduste lisamist	20
4	Tehisintellekti kaasamine SQL vigade parandamisse	21
4.1	Tehisintellekti vestlusrobotid	21
4.2	Efektiivse viiba koostamine	22
4.3	Teooria rakendamine NppDB pistikprogrammis	22
4.4	Erinevate keelemudelite vastuste analüüs	23
5	Notepad++ pistikprogrammi täiendamine	26
5.1	Võimalikud täiendused	26
5.2	Väljalaske plaan	28
5.3	Tehtud täiendused	29
5.4	Tegemata jäänud täiendused	30
6	Tulemuste valideerimine läbi kasutajate tagasiside	32
6.1	Küsimustik	32
6.2	Küsitlemise protsess	32
6.3	Tulemused	32
6.4	Tagasiside alusel tehtud täiendused	34
7	Töö tegemise refleksioon	35
7.1	Tagasisivaade lõputöö edenemisele	35
7.2	Kuidas võiks tarkvara edasi arendada?	35

8 Kokkuvõte	37
Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	42
Lisa 2 - Valideerijate vastused analüüsi küsitlusele	43
Lisa 3 - Algne väljalaske plaan	48
Lisa 4 - Lõplik väljalaske plaan	49
Lisa 5 - Valideerijate vastused tagasiside küsitlusele	50
Lisa 6 - Kuvatõmmised rakendusest	54

Tabelite loetelu

1	SQL lausete käivitamisega seotud funktsionaalsuste analüüs.	18
2	SQL lausete kontrolliga seotud funktsionaalsuste analüüs.	19
3	Võimalikud täiendused.	26

1 Sissejuhatus

Käesolev bakalaureusetöö keskendub tekstiredaktoris kasutatava SQL-lauset analüüsi ja käivitamise võimalusega pistikprogrammi kasutusmugavuse analüüsile ning edasisele arendamisele.

1.1 Taust ja probleem

2023. aasta kevadsemestril kaitstud magistritöö "MS Accessi andmebaasides SQL programmeerimist lihtsustava pistikprogrammi loomine lähtekoodiredaktorile Notepad++" raames kavandas ja arendas autor Priit Post pistikprogrammi, mis võimaldas Notepad++ redaktoris käivitada SQL lauseid MS Access andmebaasis ning andis kasutajale tagasisidet käivitatavates lausetes esinevate probleemide kohta [8].

Antud programmi arendas edasi Andres Eelma 2023. aasta sügissemestril kaitstud bakalaureusetöö "Lähtekoodiredaktoris Notepad++ SQL programmeerimist lihtsustava pistikprogrammi edasiarendamine MS Accessi ja PostgreSQL jaoks" kirjutamise raames [9]. Autor lisas programmile juurde võimekuse ühenduda PostgreSQL andmebaasidega ning täiendas SQL lauset vigade otsimise funktsionaalsust.

Nende kahe töö tulemusena valminud ja täiendatud pistikprogrammil NppDB [10] on unikaalne funktsionaalsus ilma SQL lauset käivitamata SQL koodi eelkontrollida ning kasutajale otsest tagasisidet anda. Kasutusmugavuse ja arusaadavuse poolest on aga antud rakendus nõrgem kui teised SQL lauseid käivitada ja kontrollida võimaldavad rakendused, näiteks DataGrip [11], DBeaver [12], DbSchema [13] ja DbVisualizer [14].

Valdkonna konkurentsitihedust ja kiiret arengut illustreerib asjaolu, et isegi laialt levinud ja pika ajalooga tööriistad nagu Microsoft Access saavad pidevalt olulisi uuendusi. Näiteks teatas Microsoft 2024. aasta novembris uue funktsionaalsuse Monaco SQL Editor integreerimisest Accessi Microsoft 365 versiooni. See uus redaktor toob Accessi kasutajateni kaasaegsed funktsioonid nagu täiustatud süntaksi esiletõstmise, automaatse koodilõpetamise, reanummerdamise, kommentaaride eraldi talletamise, ligipääsu laiendatud käskude valikule läbi paremkliki menüü ning heleda/tumeda teema toe, parandades oluliselt SQL-koodiga töötamise kogemust [15][16].

Samas tuleb arvesse võtta ka asjaolu, et 2025. aasta märtsi seisuga on redaktoril veel mitmeid teadaolevaid probleeme, sealhulgas aeglane laadimine, kommentaaride lisamise piirangud vaid lause algusesse ning kasutajaliidese käitumisega seotud ebakõlad, mille parandamisega Microsoft aktiivselt tegeleb [16]. See näitab, et ka suurtootjate uued funktsioonid läbivad iteratiivset arendus- ja parendustsüklit, mis on sarnane käesoleva lõputöö

raames NppDB pistikprogrammiga tehtavale tööle.

1.2 Ülesandepüstitus

Käesoleva bakalaureusetöö eesmärk on NppDB pistikprogrammi veelgi täiustada. Eesmärgiks ei ole lisada uute andmebaasisüsteemide tuge, vaid oluliselt täiustada tuge olemasolevatele andmebaasisüsteemidele, muutes protsesse mugavamaks ja lihtsamini arusaadavamaks ning lisades uusi funktsionaalsuseid, mis kasutajakogemust paremaks muudavad.

Kõigepealt on vaja teha põhjalik analüüs, et leida, milliseid funktsionaalsuseid on pistikprogrammile vaja ja mõistlik lisada. Selleks tuleb võrrelda erinevaid SQL lauseid käivitada võimaldavaid programme eesmärgiga leida funktsionaalsused ning kasutusmugavused, mis hetkel Notepad++-põhises lahenduses puuduvad. Lisaks tuleb läbi viia küsitlus, et uurida andmebaasikeeltega tegelevate ekspertide käest, mida nende arvates antud programm juurde vajaks. On ka juba teada mõned funktsionaalsused, mida varasemas lõputöös vajalike täiendustena välja toodi.

Seejärel tuleb pistikprogrammi edasi arendada, mis saab toimuma iteratiivse tarkvaraarenduse põhimõtetel. Pärast analüüsi loob autor koostöös juhendajaga toote vajaduste nimekirja. Töö autor teeb programmis täiendusi vastavalt antud nimekirjale ning testib uusi lahendusi nii ise kui ka koos juhendajaga, kes seda programmi igapäevaselt kasutab. Lõpuks küsitakse uuesti tagasisidet analüüsi kaasatud ekspertidelt.

Pistikprogrammi täiendus peab olema avatud lähtekoodiga ning avaldatud koodihoidlas GitHub [17].

1.3 Töö struktuur

Bakalaureusetöö koosneb kaheksast peatükist.

Esimeses peatükis asub töö sissejuhatus, kus antakse ülevaade töö taustast ja probleemist ning esitatakse ülesandepüstitus.

Teises peatükis tuuakse välja töö protsess, milles kirjeldatakse lahti töö kirjutamisel läbitud etapid, ning esitatakse nimekiri kasutatud tööriistadest.

Kolmandas peatükis analüüsitakse erinevaid SQL lausete käivitamist ja kontrollimist võimaldavaid programme. Samuti tutvustatakse valideerijate hinnangut NppDB pistikprogrammile enne täienduste lisamist.

Neljandas peatükis antakse ülevaade tehisintellekti vestlusrobotitest ja nende kasulikusest tarkvaraarenduses. Samuti kirjeldatakse, kuidas NppDB programmis realiseeriti keelemudelile suunatud viipade (ingl *prompt*) koostamise funktsionaalsus.

Viiendas peatükis tuuakse välja pistikprogrammi väljalaske plaani koostamiseks tehtud sammud ning arenduse käigus ellu viidud ning tegemata jäänud täiendused.

Kuuendas peatükis valideeritakse täiendatud tarkvara, saades selle kohta tagasisidet pistikprogrammi valideerimiseks kaasatud ekspertidelt.

Seitsmendas peatükis tuuakse välja, kuidas lõputöö tegemine kulges ning mida saaks ja võiks antud seisus rakendusega edasi teha.

Lõpuks esitatakse bakalaureusetöö kokkuvõte.

2 Metoodika

Käesolevas peatükis kirjeldatakse lahti lõputöö tegemise protsess ning tuuakse välja selle tegemise käigus kasutatud vahendid.

2.1 Töö protsess

Antud lõputöö viidi läbi kolmes etapis.

2.1.1 Eelanalüüs

Tarkvaraarendus on nii aja, inimressursi kui ka raha poolest väga kulukas tegevus. Et tagada ressursside sihipärane kasutus ning arendada tarkvara, mis on püsiv ja vajab vähe lisamuudatusi, on mõistlik enne arendustööde alustamist teha põhjalik analüüs. Käesoleva lõputöö raames viis autor läbi kaheosalise eelanalüüsi.

1. Olemasolevaid SQL-i haldamise ja käivitamise tööriistasid võrdlev analüüs.
2. Valideerijalt tagasiside küsimine NppDB pistikprogrammi olemasoleva versiooni kohta.

Analüüsiosa eesmärk oli tuvastada NppDB pistikprogrammi tugevused ja nõrkused võrreldes teiste turul olevate lahendustega ning leida võimalikke funktsionaalsusi ja kasutusmugavuse aspekte, mida NppDB-sse lia. Võrdluseks valiti kümme erinevat SQL lauseid käivitada või kontrollida võimaldavat rakendust. Valikukriteeriumid olid järgmised:

- **Populaarsus ja turuosa:** Kaasati laialt levinud ja tunnustatud tööriistu (nt DataGrip [11], DBBeaver [12], DBVisualizer [14]), et mõista tööstusstandardeid ja kasutajate ootusi.
- **Funktsionaalne mitmekesisus:** Valikusse lisati erinevat tüüpi rakendusi – nii kasutaja arvutisse installeeritavaid graafilise kasutajaliidesega programme, veebipõhiseid lahendusi kui ka käsureapõhiseid tööriistu (nt SQLFluff [18]), et saada ülevaade erinevatest lähenemistest SQL-i haldamisele.
- **Spetsiifilised omadused:** Otsiti rakendusi, mis pakuvad NppDB-s puuduvaid või vähearenenud funktsioone, näiteks täiustatud SQL-i vormindamist, ühendusevaba lausete kontrolli või spetsiifilisi dialektipõhiseid abivahendeid.
- **Kättesaadavus ja otsingutulemused:** Analüüsitud tarkvarad leiti internetist, kasutades Google otsingumootorit, mis peegeldab tavakasutaja viisi sarnaste tööriistade leidmiseks.

Analüüsitud tarkvarad:

1. Notepad++ koos NppDB ja Free Notepad++ SQL Formatter pistikprogrammidega. See on käesolevas töös edasiarendatava tarkvara versioon enne täienduste tegemist (kasutaja arvutisse installeeritav)
2. SQLFluff 3.4.0 [18] (käsureapõhine)
3. DataGrip 2024.2.2 [11] (kasutaja arvutisse installeeritav)
4. CodersTool [19] (veebipõhine)
5. SQL Fiddle [20] (veebipõhine)
6. ExtendsClass [21] (veebipõhine)
7. FlySpeed SQL Query 4.10.3.1 [22] (kasutaja arvutisse installeeritav)
8. DBeaver 24.2.3 [12] (kasutaja arvutisse installeeritav)
9. DBSchema 9.6.3 [13] (kasutaja arvutisse installeeritav)
10. DBVisualizer 24.2.2 [14] (kasutaja arvutisse installeeritav)
11. Visual Studio Code Database Client 7.6.3 [23] (kasutaja arvutisse installeeritav)

Võrdluseks valitud funktsionaalsused ja kasutusmugavuse aspektid (täpsemalt kirjeldatud tabelites 1 ja 2) leiti järgmiselt:

- **Olemasolevad NppDB funktsioonid:** Hinnati NppDB-s juba olemasolevate võimaluste realiseerimist võrreldes teiste tööriistadega.
- **Juhendaja sisend:** Kaasati funktsionaalsusi, mille puudumisele või parendusvajadusele NppDB-s juhtis tähelepanu lõputöö juhendaja (nt lausete ühendusevaba kontroll).
- **Autori tähelepanekud:** Lisati funktsionaalsusi, mida töö autor pidas NppDB jaoks kasulikuks ja mis teistes sarnastes tööriistades esinesid (nt parendusettepanekute tegemine ja nende kopeerimise võimalus).
- **Võrdlusanalüüsi käigus ilmnenu:** Uurides teisi programme, tuvastati täiendavaid kasulikke omadusi, mida NppDB-s ei olnud (nt käivitatud lausete ajaloos vigade ja paranduste kuvamine).

Iga analüüsitud tarkvara puhul hindas autor subjektiivselt iga tuvastatud funktsionaalsuse olemasolu ja kasutusmugavust. Kui funktsionaalsus programmis puudus, jäeti vastav tabeli lahter tühjaks. Olemasolevatele funktsionaalsustele anti hinnang kolmepallisüsteemis (A, B, C), kus:

- **Hinnang A (väga mugav ja intuitiivne kasutada):** Funktsionaalsus on kergesti leitav, selle kasutamine on loogiline ja ei vaja eraldi juhendamist ega pikemat õppimist. Kasutajakogemus on sujuv ja efektiivne.
- **Hinnang B (normaalselt mugav ja arusaadav kasutada):** Funktsionaalsus on leitav ja kasutatav, kuid võib vajada lühikest tutvumist või abiinfo lugemist. Mõned aspektid võivad olla vähem intuitiivsed, kuid üldiselt on funktsioon toimiv ja eesmärgipärane.
- **Hinnang C (ebamugav ja keeruline kasutada):** Funktsionaalsust on raske leida, selle kasutamine on segane, ebaloogiline või nõuab põhjalikumat juhendamist. Kasutajakogemus võib olla frustreeriv või aeganõudev.

Selle hindamissüsteemi eesmärk oli saada struktureeritud ülevaade erinevate tööriistade tugevustest ja nõrkustest ning tuvastada parimaid praktikaid, mida NppDB arendamisel arvesse võtta.

Teise eelanalüüsi osana kaasati seitse valideerijat, kes andsid tagasisidet NppDB pistikprogrammi olemasolevale versioonile (enne käesoleva töö raames tehtud täiendusi). Valideerijad leidis töö autor enda tutvusringkonnast. Kõik valideerijad on omandanud IT-alase kõrghariduse ning puutuvad regulaarselt kokku või on varem kokku puutunud andmebaasidega ning SQL lausetega töötamisega. Esialgse tagasiside kogumiseks kasutati veebipõhist küsimustikku, mille eesmärk oli saada ülevaade valideerijate kogemustest NppDB pistikprogrammi toonase versiooniga. Küsimustik keskendus järgmistele aspektidele:

- **Intuitiivsus ja arusaadavus:** Paluti välja tuua funktsionaalsused, mida valideerijad pidasid kõige intuitiivsemaks ja kergemini mõistetavaks ning põhjendada oma valikut.
- **Segadust tekitavad aspektid:** Küsiti, millised programmi osad või funktsioonid tekitasid kõige rohkem segadust või olid raskesti kasutatavad, koos selgitustega, miks see nii oli.
- **Soovitud täiendused ja puudujäägid:** Valideerijatel paluti kirjeldada, milliseid funktsionaalsusi nad NppDB-le juurde lisaksid või millest nad olemasoleva pistikprogrammi kasutamisel puudust tundsid. See avatud küsimus andis väärtuslikku sisendit võimalike uute arendussuundade kohta.
- **Taustainfo:** Koguti ka infot valideerijate haridustaseme ja andmebaasisüsteemidega töötamise kogemuse kohta.

Valideerijate ülesanne oli testida pistikprogrammi ja seejärel täita küsimustik. See aitab tuvastada olemasoleva versiooni peamised probleemkohad ja kasutajate ootused, mis omakorda suunasid edasist arendustööd. Täpsemad vastused küsimustele nii valideerijate tausta kui ka eelanalüüsi käigus kogutud sisendi kohta on välja toodud Lisas 2.

2.1.2 Väljalaske plaani koostamine

Eelanalüüsi tulemusena koostas autor võimalike täienduste nimekirja (tabel 3), kuhu ta koondas:

1. funktsionaalsused, mida võrreldi eelanalüüsis erinevate rakenduste vahel, aga mida pistikprogrammis NppDB veel ei eksisteeri,
2. valideerijate soovitusel, mida autor kogus arendusele eelneva küsitluse käigus (Lisa 2),
3. autori ja juhendaja ideed, mida nad ühiselt hindasid vajalikeks, aga mis ei tulnud eelanalüüsi käigus välja.

Oli selge, et kõiki võimalikke täiendusi pole võimalik lõputöö mahtu arvestades ühe arendustsükliga realiseerida. Seetõttu oli vaja koostada väljalaske plaan. Väljalaske plaani koostamiseks kasutati agiilset väljalaske planeerimist [24]. Antud töö kontekstis on tootemanikuks juhendaja. Arendusmeeskonna asemel on üks arendaja, kelleks on töö autor.

Juhendaja määras täienduste esialgse prioriteedi. Autor tõi iga täienduse juures välja esialgse keerukuse hinnangu ning seda täiendust realiseerinud programmide arvu või seda täiendust soovitanud valideerijate arvu, kui need täiendused olid antud eelanalüüsi osadest tulenevad. Keerukuse hinnangutena kasutati Fibonacci jada arve. Lisaks määras autor iteratsioonide ajakava ning otsustas prioriteetide järjekorra ning keerukuse hinnangute põhjal, millises iteratsioonis millised funktsionaalsused implementeerida. Seejärel arutati tulemus üheskoos läbi ning lepiti kokku töösse minev väljalaske plaan. Esialgne väljalaske plaan on esitatud Lisas 3.

2.1.3 Tarkvaraarendus koos jooksva tagasiside küsimisega

Lõputöö arendus toimus agiilsel viisil, iteratsioonidena. Iteratsioonid olid 6-8 päeva pikad ning algasid ja lõppesid töö autori ja juhendaja vaheliste kohtumistega. Kohtumistel tegi töö autor juhendajale demo valminud funktsionaalsustest ja muudatustest, juhendaja esitas autorile küsimusi ja muudatusettepanekuid ning üheskoos lepiti kokku järgmise iteratsiooni täpne plaan.

Tarkvaraarenduse protsessi käigus tuli ette ka olukordi, kus tuli asju teha paika pandud väljalaske plaanist erinevalt. Esimene näide sellest oli kohe enne esimest iteratsiooni, kui töö autor avastas, et mõned NppDB pistikprogrammis kasutatavad *NuGet Package*'id olid aegunud ning ei võimaldanud enam koodi kompileerida. Seega esimene iteratsioon kuluski just selle probleemi lahendamiseks. Muud väiksemad erinevused esialgse väljalaske plaani ja tegelike iteratsioonide vahel tulenesid sellest, et mõne iteratsiooni plaan täideti algselt kavandatud kiiremini, mis võimaldas autoril valmis teha ka mõne hilisemaks planeeritud väiksema funktsionaalsuse. Tegelik väljalaske plaan on esitatud Lisas 4.

Lisatud funktsionaalsuste testimine toimus kahes etapis. Algse testimise viis arenduse käigus läbi töö autor ise, et tagada uute funktsionaalsuste korrektsus ja valmisolek. Testimise teine etapp toimus töö autori ja juhendaja kohtumistel, kus juhendaja küsis töö autori käest uute funktsionaalsuste kohta täpsustavaid küsimusi ning tõi välja võimalikke erandjuhtusid. Mitmel korral tõi testimise teine etapp välja problemaatilisi kohti, mida töö autor ise ei olnud arenduse käigus leidnud ning mida järgmise iteratsiooni käigus parandati.

Pärast iteratsioonide lõppu lasi töö autor antud lõputöö jaoks leitud seitsmel valideerijal uuendustega pistikprogrammi testida. Valideerijad esitasid enda arvamuse uute funktsionaalsuste kohta vastates töö autori koostatud veebipõhisele küsimustikule. Täpsemad vastused antud valideerimise käigus küsitud küsimuste kohta on välja toodud lisas 5. Valideerijate tagasiside alusel jõudis autor veel teha väiksemaid täiendusi.

2.2 Kasutatud tööriistad

Käesoleva lõputöö eesmärkide saavutamiseks kasutati varasemalt loodud pistikprogrammi NppDB [10] ning lähtekoodi redaktorit Notepad++ [25] versioon 8.4.7. SQL lausete töötlemiseks kasutati parseri generaatorit ANTLR4 [1].

Tarkvara arendati C# [26] keeles ja .NET [27] raamistikus, arendustöödeks kasutati tarkvara JetBrains Rider [28].

Veebiküsitluste läbiviimiseks kasutati Google Forms [29] keskkonda. Kohtumisteks juhendajaga kasutati MS Teamsi [30]. Valminud tarkvara avaldati GitHubis [17].

3 Olemasolevate SQL lausete käivitamist ja kontrollimist võimaldavate programmide analüüs

Selles peatükis esitatakse NppDB programmi täienduste eelanalüüsi tulemused.

Tabelis 1 ja 2 on veergudes esitatud arvud viited jaotises 2.1.1 esitatud programmidele.

3.1 Funktsionaalsused ja kasutusmugavused seoses SQL lausete käivitamisega

SQL lausete käivitamise analüüsist tabelis 1 ilmnes, et installeeritavad rakendused ei võimalda tavaliselt lauseid kontrollida ilma aktiivse andmebaasiühendusega. Seda võimaldavad vaid veebilehitseja ja käsurea-põhised programmid. Kasutaja arvutisse installeeritavad rakendused pakuvad kasutajale palju rohkem SQL lausete kirjutamisega seonduvat abi.

Tabel 1: SQL lausete käivitamisega seotud funktsionaalsuste analüüs.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Graafiline kasutajaliides	B		B	B	B	B	B	B	B	B	B
Töötab ühendusevabalt		B		B	B	B					
Nõuab andmebaasiühendust	B		A				C	A	B	A	A
SQL võtmesõnade pakkumine	A		A		A		B	A			A
Tabelite nimede pakkumine			C				A	B			A
Veergude nimede pakkumine			C				A	B			A
Käivitatava osa valimine	B		B					B	B	A	A
Käivitatud lausete ajaloo vaatamine	B		A				B	B	A	B	B
Ajaloos kuvatakse ka lausete vead							B	B			
Ajaloos kuvatakse pakutud parandus								B			

3.2 Funktsionaalsused ja kasutusmugavused seoses SQL lausete kontrollimisega

SQL lausete kontrollimise analüüsist tabelis 2 ilmnes, et arvutisse installeeritavad rakendused annavad kasutajale rohkem tagasisidet kui veebilehitseja ja käsurea-põhised programmid. Enamik neist rakendustest pakub leitud vigadele parandusi ning kolmes neist saab parandust ka kopeerida. Hetkel ei paku ükski analüüsitud rakendus uut ja parandatud lauset ning pakutud parandus asub kas lause käivitamise tulemuse asemel või eraldi

hüpikaknas.

Tabel 2: SQL lausete kontrolliga seotud funktsionaalsuste analüüs.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Stiilivigade automaat-kontroll		B									
Automaatse stiiliparanduse võimalus	B			B		B	A		A		B
Lause sisu kontrollimise võimalus (andmebaasiühendus on loodud)	B		A				C	A	A	A	A
Lause sisu kontrollimise võimalus (andmebaasiühendus ei ole loodud)		C									
Vea kirjeldus kirjutatakse kontrollitud lauseid sisaldavasse faili											
Pakub leitud veale paranduse			A				C	A	A	A	
Parandus asub kuvatud vea juures			B				B	B	B	B	
Parandust saab kopeerida			B				B	B			
Parandus kirjutatakse kontrollitud lauseid sisaldavasse faili											
Pakub parandatud lause											
Parandatud lause asub kuvatud vea juures											
Parandatud lauset saab kopeerida											
Parandatud lause kirjutatakse kontrollitud lauseid sisaldavasse faili											

3.3 Valideerijate hinnang pistikprogrammile NppDB enne täienduste lisamist

Nagu mainitud alajaotises 2.1.1, koguti eelanalüüsi teise osana tagasisidet seitsmelt valideerijalt NppDB pistikprogrammi toonase versiooni kohta. Nende hinnangul oli pistikprogrammi kõige intuitiivsemaks ja arusaadavamaks osaks neljal juhul seitsmest ühenduste haldur (*Connection manager*) ning selle kaudu andmebaasi lisamine ja ühendamine (Lisa 2, tabel 4). Peamise segadust tekitava asjaoluna tõid neli valideerijat välja SQL-lause käivitamise F9 klahvivajutusega, kuna see ei töötanud alati järjepidevalt ning polnud intuitiivne (Lisa 2, tabel 5).

Avatud küsimusele võimalike täienduste kohta andsid valideerijad mitmesuguseid ja kasulikke ideid, millest mõned kattusid ka teiste programmide analüüsil leitud funktsionaalsustega. Näiteks toodi korduvalt välja vajadus parema SQL-süntaksi esiletõstmise järele, mis aktiveeruks kohe uue faili avamisel või andmebaasiühenduse loomisel, ilma et peaks faili eraldi sql-laiendiga salvestama. Samuti sooviti täiustatud automaatlõpetamise funktsionaalsust (*autocomplete*), mis pakuks soovitusi vastavalt ühendatud andmebaasi struktuurile. Mitmed valideerijad mainisid ka vajadust mugavama lausete käivitamise järele, näiteks võimalust käivitada lause sõltumata kursori asukohast real või käepärasema nupu lisamist F9 asemel. Lisaks pakuti välja ideid nagu tehingute kinnitamise ja tühistamise (*Commit* ja *Rollback*) funktsionaalsuse lisamine, päringu tulemuste parem haldamine (nt kõikide tulemuste vahelehtede korraga sulgemine) ning selgemad tööriistavihjedad ja legendid kasutajaliidese elementide mõistmiseks (Lisa 2, tabel 6). Kõik need ettepanekud lisati autori poolt võimalike täienduste nimekirja (tabel 3) ning need andsid olulist sisendit väljalaske plaani koostamiseks.

4 Tehisintellekti kaasamine SQL vigade parandamisse

Kuna pistikprogrammi üks kasulik funktsionaalsus oleks SQL vigade paranduste välja-pakkumine ja suured keelemudelid on viimastel aastatel hoogsalt arenenud, siis tekkis mõte, et tarkvara võiks paranduse asemel genereerida viiba, millega kasutaja saaks oma eelistatud keelemudelilt küsida vigase lause parandust ja tagasiside selle lause kohta.

Selles peatükis analüüsitakse erinevaid teadusartikliteid ja juhendeid, saamaks ideid NppDB pistikprogrammis tehisintellekti vestlusrobotile viiba e prompti genereerimiseks.

4.1 Tehisintellekti vestlusrobotid

Tehisintellekti vestlusrobotid e vestlusbotid, tuntud ka kui juturobotid, on tehisintellektil põhinevad süsteemid, mis on loodud inimestega loomulikus keeles suhtlemiseks. Paljud kaasaegsed vestlusrobotid, näiteks OpenAI loodud ChatGPT [31], Google'i arendatud Gemini (varasemalt Bard) [32] ning Anthropicu Claude [33], baseeruvad suurtel keelemudelitel (LLM – *Large Language Models*). Need mudelid on treenitud tohutul hulgal tekstilistel andmetel, mis võimaldab neil mõista konteksti, genereerida inimestele mõistetavat teksti, vastata küsimustele, tõlkida keeli ning isegi luua loomingulist sisu. Kuigi üldotstarbelised vestlusrobotid on laialt levinud, arendatakse neid ka spetsiifilistele valdkondadele, näiteks klienditeenindusele või haridusele, kohandatud lahendusi.

Tehisintellektil põhinevad tööriistad on programmeerimismaastikul muutumas üha populaarsemaks, pakkudes arendajatele tuge mitmesuguste ülesannete täitmisel. Näiteks GitHub Copilot, mis on treenitud miljardite koodiridade põhjal, on näidanud märkimisväärt mõju arendajate tööviljakusele, aidates neil ülesandeid täita oluliselt kiiremini, eriti rutiinsete ja korduvate koodilõikude genereerimisel [34]. Samas on laialt kasutatavad keelemudelid nagu ChatGPT osutunud programmeerimisel väga mitmekülgsedeks abilisteks, suutes genereerida koodi, aidata vigade otsimisel ja parandamisel, selgitades keerukaid koodilõike ning isegi abistades algoritmide disainimisel [35]. Kuigi need vahendid pakuvad märkimisväärt võimalust programmeerimise kiirendamiseks ja lihtsustamiseks, on oluline meeles pidada, et nende genereeritud väljundid vajavad koodi õigsuse, turvalisuse ja optimaalsuse tagamiseks kriitilist hindamist ja inimeste poolset kontrolli [35]. See tõdemus on asjakohane ka SQL vigade parandamisel, kus tehisintellekti pakutud lahendused võivad vajada kontekstipõhist kohandamist ja valideerimist.

4.2 Efektiivse viiba koostamine

Efektiivse viiba koostamine on väga tähtis selleks, et saada suurtelt keelemudelitelt (LLM) võimalikult täpseid ja asjakohaseid vastuseid. Parimate praktikate kohaselt peaks viip olema võimalikult selge ja spetsiifiline, sisaldades piisavalt konteksti, täpset ülesande kirjeldust ning soovitud väljundi formaati [36][37]. Näiteks SQL vigade parandamisel tähendaks see vigase SQL lause, ilmnenud veateate ja kasutatava andmebaasisüsteemi (PostgreSQL või MS Access) täpsustamist. Samuti on kasulik anda modelile persooni roll, näiteks paluda tal käituda kui kogenud SQL-arendaja [37], ning vajadusel pakkuda mõningaid näiteid (*few-shot prompting*), illustreerimaks oodatavat tulemust [36].

Keerukamate probleemide lahendamisel, nagu süntaksi- või loogikavigade leidmine ja parandamine SQL-koodis, võib oluliselt parandada tulemust, see kui modelil palutakse oma mõttekäik samm-sammult lahti kirjutada. Seda toetab *Chain-of-Thought* (CoT) promptimismeetod, kus modelile esitatakse näiteid, mis sisaldavad lahenduskäiguni viinud arutluskäiku, julgustades modelit sarnast protsessi järgima [38]. Isegi lihtne lisandus viibale nagu "*Let's think step by step*" võib esile kutsuda modelis parema arutlusvõime ilma spetsiifiliste näideteta (*zero-shot reasoning*) [39]. NppDB pistikprogrammi kontekstis tähendab see, et genereeritav viip tehisintellekti vestlusrobotile peaks sisaldama mitte ainult veateadet ja koodi, vaid ka juhiseid, mis soodustavad analüütilist lähenemist vea parandamisele, paludes modelil selgitada vea põhjust ja pakutud paranduse loogikat.

4.3 Teooria rakendamine NppDB pistikprogrammis

Töö autor rakendas eelnevates alapeatükkides käsitletud teooriaid NppDB pistikprogrammi funktsionaalsuse loomisel, mis genereerib kasutajale viiba SQL-lausetega vigade analüüsimiseks ja parandamiseks tehisintellekti vestlusroboti abil. Genereeritud viip (vt Lisa 6, joonis 4, tähis 8) moodustati järgides parimaid praktikaid, et vestlusroboti poolt antava vastuse täpsus ja kasulikkus oleks võimalikult suur. Eesmärk oli luua võimalikult universaalne viip, mis sobiks kasutamiseks erinevate laialt levinud suurte keelemudelitega. Kasutajale antakse võimalus genereeritud viip kopeerida ja kasutada seda enda eelistatud vestlusrobotis.

Töö autor lähtus mitmest efektiivse viiba koostamise strateegiast. Esiteks, viip algab vestlusrobotile selge rolli määramisega ("*You are an expert SQL developer and troubleshooter.*"), mis aitab suunata mudeli käitumist ja teadmiste rakendamist vastavalt Google AI soovitudele [37]. Seejärel esitatakse probleem selgelt ja struktureeritult, pakkudes kogu vajalikku konteksti [36][37]. See hõlmab:

1. Kasutatavat SQL dialekti,

2. Kogu SQL lauset, milles viga esineb,
3. NppDB enda poolt antud tagasisidet või veateadet,
4. Vea täpset asukohta koodis, sealhulgas rea- ja veerunumbrit,
5. Vastavat koodilõiku, et anda visuaalset konteksti vea ümbruse kohta.

Pärast konteksti andmist esitatakse vestlusrobotile konkreetsed, mitmeosalised juhised soovitud väljundi kohta [36]. Viibale on lisatud ka otsene palve kasutada selget, nummerdatud samm-sammulist mõttekäiku ("*Please, using a clear, numbered step-by-step thought process:*"). Need juhised paluvad mudelil:

1. Selgitada veateate tähendust antud lause ja dialekti kontekstis.
2. Tuvastada vea kõige tõenäolisemad põhjused.
3. Pakkuda konkreetset, parandatud SQL koodilõiku probleemi lahendamiseks.
4. Vajadusel soovitada SQL-i parimaid praktikaid sarnaste probleemide vältimiseks tulevikus.

Otsene juhised kasutada samm-sammulist mõttekäiku lisati *zero-shot Chain-of-Thought* (CoT) (ilma näideteta mõttekäigu-põhine lähenemine) viipamise põhimõtteid järgides [39]. See, koos üksikasjalise ja mitmeosalise ülesandepüstitusega, julgustab mudelit probleemi analüütiliselt lahkama ja esitama struktureeritud, läbimõeldud ning põhjendatud vastuse [38]. Selle lähenemisega püüdis töö autor maksimaalselt ära kasutada kaasaegsete suurte keelemudelite võimekust abistada koodi mõistmisel, vigade leidmisel ja parandamisel [34][35], pakkudes NppDB kasutajale täiendavat tuge SQL lausetes leiduvate vigade parandamisel.

4.4 Erinevate keelemudelite vastuste analüüs

Et hinnata genereeritud viiba efektiivsust ja erinevate suurte keelemudelite (LLM) võimekust SQL-vigade analüüsimisel ja parandamisel, esitati jaotises 4.3 kirjeldatud põhimõtetel koostatud viip (Lisa 6, joonis 8) viiele erinevale laialt levinud vestlusrobotile: Google Gemini [32], xAI Grok [40], OpenAI ChatGPT [31], Anthropic Claude [33] ja DeepSeek Coder [41]. Kõigile mudelitele anti identne viip (Lisa 6, joonis 8), milles sisalduv vigane SQL lause kasutas PostgreSQL dialekti. Viip sisaldas NppDB poolt genereeritud veateadet, vea asukohta ning koodilõiku. Mudelitel paluti samm-sammult selgitada veateadet, tuvastada vea põhjus, pakkuda parandatud koodilõiku ning soovitada parimaid praktikaid sarnaste vigade vältimiseks.

Kõik viis keelemudelit suutsid korrektselt tuvastada probleemi olemuse – alampäringu tagastatud veergude arvu mittevastavuse peapäringu NOT IN tingimuses nimetatud veergude arvule. Samuti pakkusid kõik mudelid korrektse paranduse, mis seisnes alampäringust liigse veeru (linn) eemaldamises. See näitab, et genereeritud viip oli piisavalt selge ja informatiivne, et suunata erinevaid mudeleid õige lahenduseni. Vastuste detailsuses ja esitusviisis esines siiski mõningaid erinevusi:

- **Google Gemini 2.5** andis väga põhjaliku ja struktureeritud vastuse. See selgitas detailselt vea olemust, tõi välja peamise põhjuse ning pakkus lisaks NOT IN parandusele ka alternatiivset lahendust NOT EXISTS abil, selgitades selle eeliseid (nt parem NULL-ide käsitlemine ja potentsiaalselt parem jõudlus). Gemini rõhutas ka parimaid praktikaid, nagu alampäringute iseseisev testimine ja tabelite aliaste kasutamine, esitades info kompaktselt.
- **xAI Grok 3** pakkus samuti väga detailset ja hästi struktureeritud vastust. Sarnaselt Geminile selgitas Grok põhjalikult vea tagamaid ja pakkus NOT EXISTS alternatiivi koos selgitustega. Groki vastus oli eriti ulatuslik parimate praktikate osas, tuues välja seitse konkreetset soovitusi, sealhulgas skeemi eelduste valideerimise ja päringu jõudluse profileerimise.
- **OpenAI ChatGPT-4o** vastus oli samuti korrektne ja selge. See keskendus peamiselt NOT IN operaatori korrektsele kasutamisele ja pakkus samuti NOT EXISTS alternatiivi. Parimate praktikate osas oli ChatGPT vastus sarnane Geminile ja Grokile, kuid veidi lühem.
- **Anthropic Claude Sonnet 4** lähenes probleemile samuti samm-sammult, pakkudes mitmeid võimalikke paranduslahendusi (Solution A, B, C), sealhulgas ka varianti, kus võrreldakse mitut veergu (eeldusel, et see oli kasutaja soov). See näitab mudeli võimet kaaluda erinevaid stsenaariume. Parimate praktikate loetelu oli põhjalik ja sarnanes teistele mudelitele.
- **DeepSeek Coder V3** andis kõige lühema, kuid siiski korrektse vastuse. See tuvastas vea ja pakkus kolme parandusvõimalust, sealhulgas NOT EXISTS ja LEFT JOIN kasutamise võimalusi. Parimate praktikate osa oli kokkuvõtlikum kui teistel mudelitel.

Kokkuvõttes võib öelda, et kõik testitud keelemudelit suutsid antud viiba põhjal pakuda kasutajale väärtuslikku abi SQL-vea lahendamisel. Gemini ja Grok eristusid oma vastuste põhjalikkuse ja alternatiivsete lahenduste detailse selgitamise poolest. Claude'i võime pakuda mitut stsenaariumipõhist lahendust oli samuti tähelepanuväärne. See analüüs kinnitab, et NppDB poolt genereeritud viip on piisavalt kasulik erinevate keelemudelite kasutajatele, ning et kaasaegsed keelemudelit on võimekad abivahendid SQL-

programmeerimisel tekkivate probleemide lahendamisel. Valik, millist mudelit kasutaja eelistab, võib sõltuda vastuse detailsusastmest ja esitusviisist.

5 Notepad++ pistikprogrammi täiendamine

Selles peatükis esitatakse NppDB programmi edasiarendamisega seotud tulemused.

5.1 Võimalikud täiendused

Eelanalüüsi põhjal koostas autor võimalike täienduste tabeli 3, kus on iga täienduse juurde märgitud, kui mitmes analüüsitud programmis antud täiendus eksisteeris või mitu valideerijat antud täienduse välja pakkus.

Tabel 3: Võimalikud täiendused.

Täiendus	Programmide arv	Valideerijate arv
Pakub leitud veale paranduse	5	
Parandus asub kuvatud vea juures	5	
Lausete kontroll töötab ühendusevabalt	4	
Tabelite nimede pakkumine	4	
Veergude nimede pakkumine	4	
Parandust saab kopeerida	3	
Ajaloos kuvatakse ka lausete vead	2	
Ajaloos kuvatakse pakutud parandus	1	
Stiilivigade automaatkontroll	1	
Vea kirjeldus kirjutatakse kontrollitud lauseid sisaldavasse faili	0	
Parandus kirjutatakse kontrollitud lauseid sisaldavasse faili	0	
Pakub parandatud lause	0	
Parandatud lause asub kuvatud vea juures	0	
Parandatud lauset saab kopeerida	0	
Parandatud lause kirjutatakse kontrollitud lauseid sisaldavasse faili	0	
SQL süntaksi esiletõstmine alates uue faili avamisest		2
Kiirklahvide spikri kuvamine kasutajale		1
Vahelehe nimeks on andmebaasi nimi, millega see vaheleht seotud on		1
Süsteemikataloogis on iga laiendatava kausta kõrval '+' märk		1

Tabel 3: Võimalikud täiendused.

Täiendus	Programmide arv	Valideerijate arv
Päringud tabelitele ja veergudele, mille nimes on täpitähed, töötavad		1
Lisada juurde erinevaid eelkoostatud päringuid		1
Päringu tulemustes saab veergude sorteerimist eemaldada		1
Lause käivitamine käepärasema nupuga kui 'F9'		1
Aktiivne tulemuste vaheleht on paremini märgistatud		1
Esialgse testbaasi genereerimise võimekus		1
Kui andmebaasil ei ole parooli, siis parooli ei küsita		1
Lause käivitamine töötab ka siis, kui kursor on lause lõpus		1
<i>Commit</i> ja <i>Rollback</i> funktsionaalsuste olemasolu		1
Legend tabeli struktuurist ja selle ikoonidest aru saamiseks		1
Ühenduse andmete vaatamise võimalus, kui ühendus on avatud		
Ühenduse andmete muutmise võimalus ilma andmebaasi külge ühendumata		
Paremad tööriistavihjed süsteemikataloogis		
Kõiki tulemuste vahelehti saab sulgeda ühe nupuvajutusega		
Aknas <i>Messages</i> kuvatavat teksti saab kustutada ühe nupuvajutusega		
Tulemuste akna paremas nurgas on kirjas ühendatud andmebaasisüsteem		
Erinevate SQL lausete mitte käivitumise parandamine		
Nupuvajutusega tehisintellekti vestlusrobotile algandmete genereerimiseks päringulause moodustamine vastavalt valitud andmebaasi tabelitele		
Kuvatud vea juures on viip, mis genereerib tehisintellekti vestlusrobotile päringulause antud veale potentiaalse paranduse küsimiseks		
Uued lausekontrollid		
Erinevad veaparandused		

5.2 Väljalaske plaan

Töö juhendaja kui tooteomaniku koostatud funktsionaalsuste nimekiri prioriteetsuse järjekorras (kõige olulisemad on eespool):

1. Lausete kontroll töötab ühendusevabalt.
2. Päringud tabelitele ja veergudele, mille nimes on täpitähed, töötavad.
3. Kõiki tulemuste vahelehti saab sulgeda ühe nupuvajutusega.
4. Aknas *Messages* kuvatavat teksti saab kustutada ühe nupuvajutusega.
5. Tulemuste akna paremas nurgas on kirjas ühendatud andmebaasisüsteem ja selle versioon.
6. Aktiivne tulemuste vaheleht on paremini märgistatud.
7. SQL süntaksi esiletõstmine alates ühenduse loomisest andmebaasiga.
8. Vahelehe nimeks on andmebaasi nimi, millega see vaheleht seotud on, ning päringu tegemise aeg.
9. Tulemuste vahelehe laius muutub vastavalt nime pikkusele.
10. Päringu reavahetused on olemas ka tulemuste vahelehe tööriistavihjes.
11. PostgreSQL ühenduse andmete vaatamise võimalus, kui ühendus on avatud.
12. PostgreSQL ühenduse andmete muutmise võimalus ilma andmebaasi külge ühendumata.
13. Paremad tööriistavihjed ühenduse halduri nuppude juures.
14. Ühenduses on iga laiendatava kausta kõrval '+' märk.
15. Valides *Disconnect* ei ole enam näha ühenduse all olevat hierarhiat.
16. Muuta ühenduse hüpikmenüü valikud paremini arusaadavaks.
17. Andmebaasi tabeli hüpikmenüüst saab valida järgmised tegevused: *All rows*; *Random 100 rows*; *Drop table (RESTRICT)*; *Drop table (CASCADE)*;
18. Kui MS Accessi andmebaasil ei ole parooli, siis parooli ei küsita; kui on, siis küsitakse.
19. Kiirklahvide spikri kuvamine kasutajale.
20. Tööriistavihjed andmebaasi tabeli ikoonidest aru saamiseks.

21. Suurele keelemudelile leitud vea parandust küsiva viiba genereerimine.

Prioriteetide nimekirja põhjal moodustati algne väljalaske plaan (Lisa 3, tabel 10), kus on iga täienduse juures välja toodud selle hinnatud keerukus, planeeritud iteratsioon, kus täiendus implementeeritakse, programmide arv, kus antud täiendus olemas oli, ning valideerijate arv, kes antud täiendust lisada soovitasid.

5.3 Tehtud täiendused

Pistikprogrammi NppDB täiustati vastavalt väljalaske plaanile (Lisa 3, tabel 10) ning ellu viidi enamik planeeritud muudatustest. Iteratiivse arendusprotsessi käigus tehti aga muudatusi väljalaske plaanis paika pandud ülesannete järjekorras. Sellest tulenevalt koostati lõplik väljalaske plaan (Lisa 4, tabel 11), mis näitab iga täienduse keerukust, implementeerimise planeeritud ja tegelikku iteratsiooni, selle arendusele kulunud tundide arvu ning seda, kas täiendus sai valmis. Kokku kulus käesoleva töö raames tarkvara arendusele 287 tundi.

Kõigepealt tehti ära planeerimata ja ootamatu, kuid pistikprogrammi stabiilsuse ja edasise arendatavuse seisukohast kriitiline töö – aegunud .NET teekide uuendamine (Lisa 4, täiendus 0). Enne selle sammu läbimist ei olnud töö autoril võimalik rakendust kompileerida ning pistikprogrammiks vajalikke faile genereerida. Lisaks sellele oli üks olulisemaid tehtud täiendusi SQL lausete analüüsi võimaldamine ilma aktiivse andmebaasiühenduseta. Tänu sellele saavad kasutajad valideerida lauseid ka ilma andmebaasi lisamiseta või selle külge ühendumiseta, valides avanevast hüplikaknast (Lisa 6, joonis 7, tähis 12), millise andmebaasisüsteemi kohta analüüsitav lause käib.

Kasutajaliidese mugavuse parandamiseks tehti mitmeid täiendusi tulemuste ja ühenduste haldamise vaadetes. Tulemuste akna haldamise lihtsustamiseks lisati järgmised täiendused:

- Võimalus sulgeda kõik avatud tulemuste vahelehed korraga ühe nupuvajutusega (Lisa 6, joonis 1, tähis 1).
- Nupp teadeteakna (*Messages*) sisu tühjendamiseks (Lisa 6, joonis 1, tähis 2).
- Selgemalt eristuv aktiivne tulemuste vaheleht (Lisa 6, joonis 2, tähis 3).
- Tulemuste vahelehe nimeks määratakse nüüd ühendatud andmebaasi nimi ja lause käivitamise aeg, kusjuures vahelehe laius kohandub automaatselt vastavalt nime pikkusele (Lisa 6, joonis 2, tähis 3).
- Tulemuste akna paremasse ülanurka kuvatakse info aktiivse andmebaasiühenduse (andmebaasisüsteem, versioon ja failinimi) kohta (Lisa 6, joonis 2, tähis 4).

Andmebaasiühenduste halduris (*Database Connect Manager*) tehti järgmised parandused:

- Ühenduse puuvaates kuvatakse iga laiendatava kausta kõrval '+' nupp, mis parandab navigeerimise intuiitsust (Lisa 6, joonis 3, tähis 7).
- Täiustati tööriistavihjeid nii ühenduse halduri nuppude kui ka tabelite veergude juures, et anda kasutajale rohkem kontekstipõhist informatsiooni (Lisa 6, joonis 3, tähis 6, mis näitab detailset infot veeru kohta).
- Lisati kiirklahvide spikri kuvamise nupp, mis aitab kasutajatel kiiremini program-
miga tuttavaks saada ja efektiivsemalt töötada (Lisa 6, joonis 3, tähis 5).
- Täiendati andmebaasisüsteemide kontekstimenüü valikute sõnastusi, et kasutajal oleks lihtsam valikute tähendusest aru saada (Lisa 6, joonis 6, tähis 11).
- Lisati võimalus PostgreSQL andmebaasiühenduse andmeid vaadata ja muuta ühen-
duse ajal ning neid muuta ka siis, kui ühendus pole aktiivne, läbi kontekstimenüü
valiku "*Edit database connection details*" (Lisa 6, joonis 6, tähis 10).
- Pärast andmebaasist lahtiühendamist (*Disconnect*) peidetakse ühenduse all olnud
hierarhia, muutes vaate selgemaks.
- MS Accessi andmebaaside puhul parandati parooliküsimise loogikat: parooli küsi-
takse nüüd ainult siis, kui see on andmebaasile seatud.

Lisati ka mitu uut funktsionaalsust, mis laiendavad pistikprogrammi võimekust:

- Üks tähtsamaid uuendusi on võimalus genereerida SQL lausest leitud vea korral
automaatselt viip tehisintellekti vestlusrobotile. See viip sisaldab kogu vajalikku
konteksti (veateade, SQL lause, dialekt) vea analüüsimiseks ja parandusettepane-
kute saamiseks (Lisa 6, joonis 4, tähis 8). Lisaks on kasutajal võimalik viiba malli
ise muuta ja salvestada otse Notepad++ keskkonnas. Selle jaoks lisati andmebaasi-
ühenduste haldurisse eraldi nupp (Lisa 6, joonis 3, tähis 8).
- Täiendati andmebaasitabelite kontekstimenüüs olevaid kiirkäskusid sagedamini ka-
sutatavate operatsioonide jaoks, näiteks lisati juhusliku 100 rea kuvamine (*Select
random 100 rows*) ning täpsustati tabeli kustutamine kahel erineval viisil (*Drop
table (RESTRICT/CASCADE)*) (Lisa 6, joonis 5, tähis 9).

5.4 Tegemata jäänud täiendused

Kolm täiendust jäid erinevatel põhjustel tegemata. Täienduse "Päringud tabelitele ja veer-
gudele, mille nimes on täpitähed, töötavadärendamisel ei leidnud töö autor koodist antud

probleemi teket üles ning ka tooteomanik ei suutnud enam viga korrata.

Täienduse "SQL süntaksi esiletõstmine alates andmebaasiühenduse loomisest"realiseerimisel ei leidnud autor viisi, kuidas NppDB pistikprogrammist saata Notepad++ seadetele vastavat teadet süntaksi esiletõstmiseks vajaliku keele info muutmiseks.

Täienduse "Päringu reavahetused on olemas ka tulemuste vahelehe tööriistavihjes"arendamisel ei leidnud töö autor koodis kohta, mis eemaldas lause tekstist olemasolevad reavahetused.

Kõigi kolme täienduse puhul võttis autor ühel hetkel vastu otsuse aja säästmise huvides antud täiendus tegemata jätta ning järgmise ülesande juurde liikuda.

6 Tulemuste valideerimine läbi kasutajate tagasiside

Käesolevas peatükis kirjeldatakse NppDB pistikprogrammi täiendatud versiooni valideerimise protsessi ja tulemusi.

6.1 Küsimustik

Pistikprogrammi täiustuste hindamiseks koguti kasutajatelt tagasisidet veebipõhise küsitluse abil Google Forms keskkonnas. Küsimustik oli suunatud samadele valideerijatele, kes osalesid ka eelanalüüsi etapis esialgse NppDB versiooni hindamisel. Küsimustiku eesmärk oli hinnata täiendatud pistikprogrammi uute funktsioonide leitavust ja kasutatavust, programmi üldist funktsionaalsust pärast tehtud muudatusi ning tuvastada võimalikke probleeme või edasisi parandusettepanekuid. Küsimustik sisaldas nii skaalal põhinevaid hindamisküsimusi kui ka avatud küsimusi, mis võimaldasid valideerijatel anda põhjalikumat sõnalist tagasisidet (vt Lisa 5). Kokku laekus tagasiside seitsmelt valideerijalt.

6.2 Küsitlemise protsess

Valideerijatele anti ligipääs NppDB pistikprogrammi täiendatud versioonile ning paluti neil seda testida, pöörates erilist tähelepanu uutele ja muudetud funktsionaalsustele. Pärast pistikprogrammi katsetamist paluti valideerijatel täita tagasisideküsimustik. Selline lähenemine tagas, et antud hinnangud põhinesid praktilisel kasutuskogemusel ning olid seeläbi võimalikult objektiivsed ja asjakohased. Küsitlus viidi läbi pärast kõigi planeeritud arendusiteratsioonide lõppu.

6.3 Tulemused

Valideerijate tagasiside täiendatud NppDB pistikprogrammile oli üldiselt hea. Detailsemad vastused on esitatud Lisas 5. Uute funktsioonide leidmist hindasid valideerijad keskmiselt hindegas 4,0 viiepallisüsteemis (kus 1 tähendas väga keerulist ja 5 väga lihtsat) (Lisa 5, tabel 12). Uute funktsioonide kasutamise lihtsus sai keskmiseks hindeks 4,3 (Lisa 5, tabel 13). Need tulemused näitavad, et uued funktsioonid on kasutajate jaoks üldiselt hästi leitavaid ja nende kasutamine on valdavalt arusaadav, kuigi üksikutel juhtudel võis leidmisega raskusi esineda. Pistikprogrammi funktsionaalsust pärast tehtud täiendusi hindasid kõik seitse vastanud valideerijat varasemast paremaks. Keskmine hinnang oli 4,4 viiepallisüsteemis (kus 1 tähendas oluliselt halvemat ja 5 oluliselt paremat) (Lisa 5, tabel 14), mis viitab märgatavale paranemisele võrreldes eelmise versiooniga.

Kõige kasulikumate uute funktsioonide ja täiendustena töid valideerijad välja (Lisa 5, tabel 15):

- Üldine kasutajasõbralikkuse paranemine, näiteks laiendatavate kaustade juures „+“ märgi kuvamine süsteemikataloogis.
- Andmebaasi tabeli hüpikmenüüst erinevate eelmääratletud operatsioonide (nt „Kõik read“, „Juhuslikud 100 rida“, „Drop table“) valimise võimalus.
- Kiirklahvide spikri kuvamine.
- Tehisintellekti vestlusrobotile suunatud viiba genereerimise võimalus SQL vigade analüüsimiseks.
- SQL analüsaatori funktsionaalsus.
- Kõigi tulemuste vahelehtede korraga sulgemise nupp.

Mõningate funktsioonide osas tehti ka parendusettepanekuid või juhiti tähelepanu väiksematele probleemidele (Lisa 5, tabelid 16 ja 17):

- Üks valideeriija soovitas, et tabeli kustutamise (Drop table) nupule võiks lisada kinnitusküsimuse.
- Tehisintellekti viiba genereerimise funktsiooni kohta märgiti, et selle kasutusvõimalus (ainult vea ilmnemisel) võiks olla kiirklahvide spikris selgemalt kirjeldatud, kuna praegune sõnastus võib tekitada eksiarvamuse, et seda saab kasutada igal ajal.
- Üks valideeriija pidas MS Alert heli akna „Show plugin shortcuts“ kuvamisel segadust tekitavaks.
- Juhiti tähelepanu olukorrale, kus pärast süntaksivea analüüsi ja analüüsitulemuste kustutamist jääb teade „Analysis found errors“ päringu tulemuste kasti alles, mis võib kasutajamugavust pärssida.

Valideerijate valmisolek NppDB laienduse uuendatud versiooni soovitada oli kõrge. Küsimusele, kui tõenäoliselt nad soovitsid pistikprogrammi kolleegile või sõbrale (skaalal 0 – kindlasti ei soovitaks, 10 – kindlasti soovitaksin), andsid valideerijad hinnanguteks 5 (üks vastaja), 8 (neli vastajat) ja 9 (kaks vastajat) (Lisa 5, tabel 18), keskmise hinnanguga 7,9.

Selle tulemuse täiendavaks analüüsiks saab kasutada *Net Promoter Score* (NPS) metoodikat [42]. Antud valideerimise puhul oli seitsmest vastajast kaks soovitajat (hinne 9), neli passiivset (hinne 8) ning üks kriitik (hinne 5). NPS arvutatakse soovitajate protsendi

ja kriitikute protsendi vahena, seega on NPS antud juhul ligikaudu 14%. Skaalal -100% kuni 100% peegeldab see üldiselt positiivset meelestatust, mis vastab keskmise soovitamise hindele 7,9. Siinkohal tuleb aga arvestada, et NPS-i tulemused on väikese valimi (antud juhul 7 valideerijat) puhul piiratud statistilise tähendusega ning seda ei soovitata kasutada ainsa mõõdikuna kasutajakogemuse hindamiseks.

6.4 Tagasiside alusel tehtud täiendused

Valideerijatelt saadud tagasiside oli väärtuslik ning andis olulist sisendit pistikprogrammis tehtud täienduste parendamiseks. Kuigi käesoleva bakalaureusetöö ajaline piirang ja maht ei võimaldanud kõiki tagasisides esitatud ettepanekuid vahetult pärast valideerimisküsitlust realiseerida, viiakse need ellu pärast lõputöö esitamist. Peamised muudatused, mida töö autor planeerib pistikprogrammis veel teha, tuginedes valideerijate tagasisidele, on järgmised:

- Kasutajaliidese täpsustused, näiteks kinnitusdialoogi lisamine tabeli kustutamise funktsioonile.
- Tehisintellekti viiba genereerimise funktsiooni kohta antava info selguse parandamine, et kasutajad mõistaksid paremini selle kontekstipõhist rakendatavust.
- Väiksemate kasutajamugavuse probleemide lahendamine, nagu MS Alert heli kasutamine kiirklahvide spikri kuvamisel ning „Analysis found errors“ teate püsijäämise küsimus pärast analüüsitulemuste kustutamist.

Nende täienduste tegemine aitab NppDB pistikprogrammi veelgi kasutajasõbralikumaks ja funktsionaalsemaks muuta.

7 Töö tegemise reflektatsioon

Selles peatükis heidab autor pilgu tagasi tehtud tööle.

7.1 Tagasivaade lõputöö edenemisele

Käesoleva lõputöö tegemisel oli eelanalüüs uute funktsionaalsuste valimisel üllatavalt sisukas ja kasulik. Enne analüüsi läbi viimist arvas töö autor, et valdav enamik uutest funktsionaalsustest tuleb ikkagi ise välja mõelda, aga tänu põhjalikule ja ajamahukale analüüsile tuli välja üllatavalt palju võimalikke täiendusi, mis antud pistikprogrammi paremaks muudaksid.

Antud töö suurimateks puudusteks on kindlasti tarkvara arendusega liiga hilja alustamine ning uute funktsionaalsuste kohta saadud tagasiside vähesus. Kuigi töö autoril oli võimalus tarkvara arendusega alustada juba semestri alguses, jättis ta selle alles märtsi keskpaika ning avastas alles siis, et programmi teegid vajasid uuendamist. Seega lükkus kogu planeeritud arendusprotsess veel pool kuud edasi. Sellest tulenevalt jäi ka tagasiside kogumiseks vähe aega ning tagasisidest saadud sisend tuleb realiseerida alles pärast lõputöö esitamist.

Ühe huvitava aspektina arendustsüklite juures saab kindlasti välja tuua iteratiivse tarkvaraarenduse kasulikkuse antud lõputöö kirjutamisel. Tänu iganädalastele kohtumistele sai juhendaja kui tooteomanik näha lisatavaid funktsionaalsuseid ka nende implementeerimise faasis ning välja tuua erinevaid muudatusettepanekuid, mis muutsid lõpptulemuse kasutatavust mugavamaks ja paremaks. Hea näitena saab välja tuua tulemuste aknas andmebaasisüsteemi kuvamise, mille juurde ei olnud algselt planeeritud lisada selle versiooni, ning millele just iteratsioonide vahelisel kohtumisel tähelepanu pöörditi ning muudatusena plaani lisati.

Lõputöö kirjutamise käigus jõudis ka autor ise arusaamani, et NppDB pistikprogramm on tegelikult väga kasulik tarkvara igale SQL koodi kirjutajale ning seda on koos Notepad++ tekstiredaktoriga väga lihtne käivitada ja kasutada. Koos implementeeritud kasutusmugavust parandavate funktsioonidega saavad sellest rakendusest kasu leida veelgi rohkemad arendajad nii ülikoolis kui ka väljaspool.

7.2 Kuidas võiks tarkvara edasi arendada?

Kuigi pistikprogrammi on antud lõputöö raames kasutajasõbralikumaks muudetud, saab seda kindlasti edasi arendada, et kasutajatel oleks sellest veel rohkem kasu. Kõige oluli-

sem oleks rakendust täiendada veelgi rohkemate lausevigade kontrollimise võimalustega ning lisada tuge uutele andmebaasisüsteemidele, millega pistikprogramm saaks ühenduda.

Lisaks saab mitmes suunas edasi arendada keelemudelitega seotud funktsionaalsuseid. Näiteks saaks genereerida viipasid keelemudelilt järgmiste küsimuste küsimiseks.

1. Genereeri valitud tabelitesse testandmete lisamise laused.
2. Paku soovitusi, kuidas olemasolevat (korrektset) lauset efektiivsemaks muuta.
3. Loo kommentaare või selgitusi valitud lausele.
4. Loo alternatiivseid lauseid, mis sama ülesannet lahendavad.
5. Tõlgi etteantud inimkeelne küsimus valitud tabeli põhjal täidetavaks päringuks.

Väga kasulik oleks ka NppDB registreerimine ametliku Notepad++ pistikprogrammina.

8 Kokkuvõte

Käesoleva bakalaureusetöö peamine eesmärk oli lähtekoodiredaktorile Notepad++ loodud SQL programmeerimist lihtsustava pistikprogrammi NppDB edasiarendamine ja täiustamine. Töö keskendus kahele peamisele sihile: programmi kasutusmugavuse oluline parandamine ning uute, väärtust andvate funktsionaalsuste lisamine.

Eesmärkide saavutamiseks viidi esmalt läbi põhjalik eelanalüüs, et teha kindlaks vajalikud ja mõistlikud täiendused. See protsess hõlmas olemasolevate SQL lausete käivitamise ja haldamise tööriistade funktsionaalsuseid ja kasutusmugavusi võrdleva analüüsi tegemist. Lisaks koguti veebipõhise küsitluse kaudu väärtuslikku sisendit SQL keele aktiivsetelt kasutajatelt, selgitamaks välja nende ootused ja vajadused pistikprogrammi edasise arenguga seoses. Arvesse võeti ka varasemates NppDB-ga seotud lõputöodes esile toodud täiendustepanekuid. Üheks oluliseks uurimisvaldkonnaks kujunes tehisintellekti vestlusrobotide kaasamine SQL vigade parandamisse, mille tarbeks analüüsiti efektiivsete viipade koostamise teooriaid ja parimaid praktikaid.

Pistikprogrammi edasiarendus toimus iteratiivse tarkvaraarenduse metoodika alusel. Pärast eelanalüüsi ja juhendajaga konsulteerimist koostati väljalaske plaan, mis määras uute funktsionaalsuste suhtelise olulisuse üksteisega võrreldes. Arendustsükkel koosnes mitmest iteratsioonist, mille käigus lisati uusi funktsionaalsuseid ja parandati olemasolevaid. Iga iteratsiooni tulemusi testiti ja valideeriti nii autori enda poolt kui ka koostöös juhendajaga. Arendustöö tähtsaimateks tulemusteks olid tehisintellekti vestlusrobotile suunatud viipade genereerimise funktsionaalsuse lisamine, SQL lause analüüsi võimaldamine ilma andmebaasiühenduseta ning üldise kasutusmugavuse mitmekülgne parandamine. Arendustsükli lõpus koguti täiendatud pistikprogrammi kohta tagasisidet ka eelanalüüsis osalenud valideerijatelt. Nende hinnangul olid tehtud täiendused väärtuslikud ning muutsid pistikprogrammi oluliselt kasutajasõbralikumaks.

Bakalaureusetöö tulemusena valminud tarkvara on kasutuseks valmis, kuid edasi arendatav nii kasutusmugavuse, uute funktsionaalsuste kui ka uute võimalike andmebaasisüsteemide toe poolest. Käesoleva töö tulemus avaldati MIT litsentsiga ning on avalikult kättesaadav GitHubi keskkonnas aadressil: <https://github.com/stenarm/NppDB>.

Kasutatud kirjandus

- [1] T. Parr. *ANTLR*. [Online]. URL: <https://www.antlr.org/> (Loetud: 16.02.2025).
- [2] Microsoft. *Microsoft Access*. [Online]. URL: <https://www.microsoft.com/en-us/microsoft-365/access> (Loetud: 23.02.2025).
- [3] S. Jung. *NppDB*. [Online]. URL: <https://github.com/gutkyu/NppDB> (Loetud: 27.01.2025).
- [4] Microsoft. *An Introduction to NuGet*. [Online]. URL: <https://learn.microsoft.com/en-us/nuget/what-is-nuget> (Loetud: 13.03.2025).
- [5] E. Entsüklopeedia. *Mittetulundusühing Entsüklopeedia. pistikprogramm*. [Online]. URL: <http://entsyklopeedia.ee/artikkel/pistikprogramm> (Loetud: 09.03.2025).
- [6] The PostgreSQL Global Development Group. *PostgreSQL*. [Online]. URL: <https://www.postgresql.org/about/> (Loetud: 23.02.2025).
- [7] W3Schools. *SQL Tutorial*. [Online]. URL: <https://www.w3schools.com/sql/> (Loetud: 23.02.2025).
- [8] P. Post. „MS Accessi andmebaasides SQL programmeerimist lihtsustava pistikprogrammi loomine lähtekoodiredaktorile Notepad++“. [Online]. Magistritöö. Tallinn, Eesti: Infotehnoloogia teaduskond, TalTech, 2023. URL: <https://digikogu.taltech.ee/et/item/0ac16423-5434-450d-a961-26fa087f91d4> (Loetud: 27.01.2025).
- [9] A. Eelma. „Lähtekoodiredaktoris Notepad++ SQL programmeerimist lihtsustava pistikprogrammi edasiarendamine MS Accessi ja PostgreSQL jaoks“. [Online]. Bakalaureusetöö. Tallinn, Eesti: Infotehnoloogia teaduskond, TalTech, 2023. URL: <https://digikogu.taltech.ee/et/Item/08388be4-7094-4e6b-aefc-3c44a23eaa30> (Loetud: 27.01.2025).
- [10] A. Eelma. *NppDB*. [Online]. URL: <https://github.com/aneelm/NppDB> (Loetud: 27.01.2025).
- [11] JetBrains. *DataGrip: The Cross-Platform IDE for Databases & SQL by JetBrains*. [Online]. URL: <https://www.jetbrains.com/datagrip/> (Loetud: 04.02.2025).

- [12] S. Rider. *DBeaver*. [Online]. URL: <https://dbeaver.io/> (Loetud: 04.02.2025).
- [13] Wise Coders GmbH. *DbSchema*. [Online]. URL: <https://dbschema.com/> (Loetud: 04.02.2025).
- [14] DbVis Software. *DbVisualizer*. [Online]. URL: <https://www.dbvis.com/> (Loetud: 04.02.2025).
- [15] Microsoft Access Blog. „Try the new modern SQL editor in Access!“ (2024). [Online]. URL: <https://techcommunity.microsoft.com/blog/accessblog/try-the-new-modern-sql-editor-in-access/4289430> (Loetud: 11.05.2025).
- [16] C. Riddington. „Monaco SQL Editor“ (2024). [Online]. URL: <https://www.accessforever.org/post/monaco-sql-editor> (Loetud: 11.05.2025).
- [17] GitHub. *GitHub*. [Online]. URL: <https://github.com/> (Loetud: 27.01.2025).
- [18] SQLFluff. *SQLFluff - The SQL Linter for humans*. [Online]. URL: <https://sqlfluff.com/> (Loetud: 04.02.2025).
- [19] Coders Tool. *Coder's Tool, Online Tools and Utilities*. [Online]. URL: <https://www.coderstool.com/> (Loetud: 04.02.2025).
- [20] ZZZ Projects. *SQL Fiddle*. [Online]. URL: <https://sqlfiddle.com/> (Loetud: 04.02.2025).
- [21] ExtendsClass. *ExtendsClass, Free Tools for Web Designers and Webmasters*. [Online]. URL: <https://extendsclass.com/> (Loetud: 04.02.2025).
- [22] ActiveDBSoft. *FlySpeed SQL Query*. [Online]. URL: <https://www.activedbsoft.com/overview-querytool.html> (Loetud: 04.02.2025).
- [23] Microsoft. *Database Client*. [Online]. URL: <https://marketplace.visualstudio.com/items?itemName=cweijan.vscode-database-client2> (Loetud: 04.02.2025).
- [24] T. Norman. „Agile Release Planning 101“ (2012). [Online]. URL: <https://tommynorman.blogspot.com/2012/09/agile-release-planning-101.html> (Loetud: 17.02.2025).
- [25] D. Ho. *Notepad++*. [Online]. URL: <https://notepad-plus-plus.org/> (Loetud: 27.01.2025).

- [26] Microsoft. *C#*. [Online]. URL: <https://dotnet.microsoft.com/en-us/languages/csharp> (Loetud: 03.02.2025).
- [27] Microsoft. *.NET*. [Online]. URL: <https://dotnet.microsoft.com/en-us/> (Loetud: 03.02.2025).
- [28] JetBrains. *Rider*. [Online]. URL: <https://www.jetbrains.com/rider/> (Loetud: 03.02.2025).
- [29] Google. *Forms*. [Online]. URL: <https://docs.google.com/forms/> (Loetud: 06.02.2025).
- [30] Microsoft. *Microsoft Teams*. [Online]. URL: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software> (Loetud: 20.01.2025).
- [31] OpenAI. *ChatGPT*. [Online]. URL: <https://openai.com/chatgpt> (Loetud: 02.03.2025).
- [32] Google DeepMind. *Gemini*. [Online]. URL: <https://deepmind.google/technologies/gemini/> (Loetud: 02.03.2025).
- [33] Anthropic. *Claude*. [Online]. URL: <https://www.anthropic.com/claude> (Loetud: 02.03.2025).
- [34] S. Peng, E. Kalliamvakou, P. Cihon ja M. Demirer. „The Impact of AI on Developer Productivity: Evidence from GitHub Copilot“. *arXiv preprint arXiv:2302.06590* (2023). [Online]. arXiv: 2302.06590. URL: <https://arxiv.org/abs/2302.06590> (Loetud: 04.03.2025).
- [35] H. Tian *et al.* „Is ChatGPT the Ultimate Programming Assistant – How far is it?“ *arXiv preprint arXiv:2304.11938* (2023). [Online]. arXiv: 2304.11938. URL: <https://arxiv.org/abs/2304.11938> (Loetud: 04.03.2025).
- [36] OpenAI. *Best practices for prompt engineering with OpenAI API*. [Online]. URL: <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api> (Loetud: 04.03.2025).
- [37] Google. *Prompt design strategies*. [Online]. URL: <https://ai.google.dev/gemini-api/docs/prompting-strategies> (Loetud: 04.03.2025).

- [38] J. Wei *et al.* „Chain-of-Thought Prompting Elicits Reasoning in Large Language Models“. *arXiv preprint arXiv:2201.11903* (2022). [Online]. arXiv: 2201.11903. URL: <https://arxiv.org/abs/2201.11903> (Loetud: 05.03.2025).
- [39] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo ja Y. Iwasawa. „Large Language Models are Zero-Shot Reasoners“. *arXiv preprint arXiv:2205.11916* (2022). [Online]. arXiv: 2205.11916. URL: <https://arxiv.org/abs/2205.11916> (Loetud: 05.03.2025).
- [40] X.AI LLC. *Grok*. [Online]. URL: <https://x.ai/> (Loetud: 02.03.2025).
- [41] Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co. *Deepseek*. [Online]. URL: <https://deepseek.chat/> (Loetud: 02.03.2025).
- [42] T. Fessenden. „Net Promoter Score: What a Customer-Relations Metric Can Tell You About Your User Experience“ (2024). [Online]. URL: <https://www.nngroup.com/articles/nps-ux/> (Loetud: 05.03.2025).

Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Sten Ärm

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Lähtekoodiredaktoris Notepad++ SQL programmeerimist lihtsustava pistikprogrammi täiustamine”, mille juhendaja on Erki Eessaar
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2025

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

Lisa 2 - Valideerijate vastused analüüsi küsitlusele

Tabel 4: Milline NppDB funktsionaalsus on sinu arvates kõige intuitiivsem või arusaadavam? Miks?

Vastaja	Vastus
1	Connection manager, sest see on väga sarnane kõikide teiste populaarsete tööriistade connection manager osale.
2	See kuidas tulemusi kuvatakse
3	Võimalus teha paremkliik tabelile ja valida top 100 kirjet. See on kõige lihtsam ja kiirem viis, kuidas tabelis olevat infot kiirelt lugeda ja kätte saada.
4	Andmebaasi lisamine kuna roheline rist ja teised sümbolid on üldlevinud andmebaaside tarkvarades
5	Süsteemikataloogis tabeli kirjelduste vaatamine.
6	Andmebaasi lisamise/ühendamise/lahtiühendamise nupud on üsna üheselt mõistetavad. Nuppudel olev sümboolika ja värvid toetavad seda.
7	Uue andmebaasi loomine. Tundus selge

Tabel 5: Milline NppDB funktsionaalsus tekitab sinus kui kasutajas kõige rohkem segadust? Miks?

Vastaja	Vastus
1	Selle installimine. Suvaliste failide alla laadimine ja .dll failide unblockimine jms. Asi pole isegi segaduse tekitamises, aga oma tööarvutis ega isiklikus arvutis ei teeks iial midagi sellist turvalisuse kaalutlustel. Pidin testimiseks oma vana läpaka välja otsima.
2	MS Accessi andmebaasiga ühendudes küsiti mult parooli, kuigi andmebaasil parooli peal pole. Käsklusi saab käivitada ainult käsku highlightides. Kui mu kursor oli teksti juures (kohe pärast käsu lõppu, pärast semikoolonit), siis F9 ei töötanud
3	Kui päring on ühel real ja pole highlighted, siis kursor peab olema rea alguses, et F9 abil see käivituks. See on segane, kuna näiteks MS SQL Serveris ei ole kursori asukoht oluline - kui päring on ühel real ja päring käivitada sõltumata kursori asukohast, siis tagastatakse vastav tulemus.

Tabel 5: Milline NppDB funktsionaalsus tekitab sinus kui kasutajas kõige rohkem segadust? Miks?

Vastaja	Vastus
4	Päringute jooksumiseks vajalik F9 vajutamine. Olen harjunud, et SQL käsu ees on olemas linnuke käivitamiseks. Lisaks commit või rollback nuppude olemasolu. Kohe ei olnud arusaadav, kas tabeli täiendamine toimus või mitte ning kas rollback on üldse võimalik. Notepad++ on küll olemas nupp Undo, aga sellel on teine otstarve.
5	Tabeli struktuuri kirjeldus. Ei saa aru mis ikoonid tähendavad. Näidis Hotell andmebaasile viidates jääb mulje, et Reserveerimine tabelil on mitu primary keyd.
6	Panen kolm asja: * See, kuidas päringute vastused tulevad nõ eraldi vahelehtedena eraldi aknaossa, iga päringu kohta ajaline märge. Palju päringuid tehes läheb see timestamp'ide pilt üsna kirjuks, eriti kui tekst vahelehele ära ei mahu. * Miks F9 ei toimi, kui kursor on rea lõpus? Kasutajana oleks hulga mugavam, kui saaks lauset käivitada kohe peale selle kirjutamist, kui hakata lauset selekteerima või kursorit rea algusesse viima. * Aktiivset vahelehte antud vahelehtede tulba alumisena ja vahetades vahelehed muudavad järjekorda, mis võib tekitada segadust kui vahelehtedel on näha ainult "ResultX < timestamp > ". Hakka siis pärast otsima, kus su "Result 1" kadus seal järjekorras.
7	Kui lood andmebaasi, siis vahepeal on raske arusaada, kas

Tabel 6: Milliseid funktsionaalsuseid võiks sinu arvates NppDB-le juurde lisada? Millest tunned antud pistikprogrammi kasutamise juures puudust?

Vastaja	Vastus
1	Andmebaasi nime peal Right-click – > 'Open' tekstiaken võiks olla kohe võimaline syntax highlightingut tegema. Praegu peab selleks faili .sql laiendiga salvestama, et syntax highlightingut näha.

Tabel 6: Milliseid funktsionaalsuseid võiks sinu arvates NppDB-le juurde lisada? Millest tunned antud pistikprogrammi kasutamise juures puudust?

Vastaja	Vastus
2	Ma ei tea kellele see programm suunatud on, seega on keeruline öelda. Võibolla visuaale, selleks et Principle of Least Astonishment rakendada. Ma üllatusin selle peale, mis avanes, kui vajutasin öpenändmebaasi peale. Ma ei olnud kindel, kas ma tõesti pean tavalisse tekstifaili oma käske kirjutama. Ma oleks oodanud mingisugust akent, kus on vähemalt "run" nupp juures, et ma teaks, et pean siia kirjutama oma käsud ja siis saan vajutada nupule, et käske käivitada. Märkasin et notepad++ default autocomplete töötab. Võibolla saaks asendada default autocomplete'i mingi parema autocomplete'ga, mis oskab anda soovitusi vastavalt andmebaasile, millega sa ühendatud oled.
3	Kursori asukohast sõltumata võimalus käivitada päring F9 abil. Lisaks parandada viga tabelite puhul, kus ise päringut kirjutades (nt select top 10 * from Külaline puhul) saabub veateade: NppDB.dll just crashed in runPluginCommand(<i>size_t</i> : 23). Kui vastav käsklus anda andmebaasi menüüst käsuga Select top 100, siis SQL päring toimib.
4	Commit ja Rollback võimalus. Nupp käsu jooksumiseks. Tulevikus, kui on rohkem kui üks andmebaas laetud, võiks vahelehe nimi olla andmebaasi nimi, millega antud vaheleht on seotud. Hetkel olid Notepad tavapärased lehe nimetused ("New 1", "New 2").
5	Süsteemikataloogil võiks iga expanditava kausta, tabeli jms ees olla '+', praegu on see üksnes iga taseme esimesel ning teistele tekib pärast topeltklikki. Tabeli kitsenduste propertyd. Primary key on aru saadav, kuid teised mitte. Näidis Hotell andmebaasile viidates, siis pole arusaadav, mis kas Hotell tabeli veergudel on indeksid, unikaalsed indeksid vms. Rohkem sageli kasutatavaid eelkoostatud päringuid. Praegu on tabelitel päringud esimese 100 kirje pärimiseks ning tabeli droppimiseks, kuid mugav oleks näiteks veeru lisamise päring. Veerul võiks olla selle kustutamise päring jne. Sama tabeli loomiseks nt. Saaks käivitada üherealist päringut (failis aint 1 rida), kui pole rea lõpus tühikut/faili lõpus tühja rida. Autocomplete oleks amazing. Query highlighting: keywords, functions etc. Query formatter. Ei näe resulti kui päringuid on liiga palju (scroll võimalus puudub). Resultis saab veerge sorteerida, kuid sorteerimisit ei saa maha võtta. Päringu käivitamise aeg. Paremini formaaditud veateated. Kui proovin alustada transaktsiooni nt (pole tuge), siis toetatud statementid on eraldatud punktidega, mitte komadega jms.

Tabel 6: Milliseid funktsionaalsuseid võiks sinu arvates NppDB-le juurde lisada? Millest tunned antud pistikprogrammi kasutamise juures puudust?

Vastaja	Vastus
6	* Mugavam päringute käivitamine - eriti just eelpooltoodud kursori osas. Mugav oleks ka, kui saaks mõne käepärasema klahviga käivitada, kui F9 (suure klaviatuuriga jääb see nupp tavaliselt peale kirjutamist käeulatusest kaugemale), aga eks see ole võibolla pigem harjumise asi. * Lausete käivitamisest tingitud vahelehti võiks saada kuidagi sulgeda - väikse ekraaniga läheb asi kiirelt käest ära. * Mitmete päringute tegemisel, aktiivne vaheleht võiks olla paremini märgistatud - mis tulemust ma nüüd siis hetkel vaatan? Praegu on kõik ühtlane hall.
7	Võiks olla mingi esialgse testbaasi genereerimise võimekus

Tabel 7: Mis on sinu kõrgeim omandatud haridustase?

Vastaja	Vastus
1	Bakalaureusekraad
2	Bakalaureusekraad
3	Magistrikraad
4	Bakalaureusekraad
5	Bakalaureusekraad
6	Magistrikraad
7	Bakalaureusekraad

Tabel 8: Mis eriala kraadi viimati omandasid?

Vastaja	Vastus
1	Informaatika
2	Informaatika
3	Äriinfotehnoloogia
4	Informaatika
5	Informaatika
6	Informaatika
7	Informaatika

Tabel 9: Kui palju puutud hetkel kokku või oled kokku puutunud SQL andmebaasidega ja päringutega töötamisega?

Vastaja	Vastus
1	Iganädalaselt
2	Backend veebiarendajana olen üpris palju kokku puutunud
3	Igapäevaselt
4	Olen läbinud ülikoolis andmebaaside aineid.
5	Igapäevaselt
6	Igapäevaselt
7	umbes 4 aastat

Lisa 3 - Algne väljalaske plaan

Tabel 10: Väljalaske plaan.

Täiendus	Keerukus	Planeeritud iteratsioon	Programmide arv	Valideerijate arv
1.	8	3	4	
2.	3	1		
3.	1	1		
4.	1	1		
5.	2	1		
6.	1	1		1
7.	3	2		2
8.	1	2		1
9.	1	2		
10.	1	2		
11.	2	2		
12.	2	2		
13.	1	4		
14.	1	4		1
15.	1	4		
16.	1	4		
17.	3	4		
18.	1	4		1
19.	1	4		1
20.	2	4		1
21.	8	5		

Lisa 4 - Lõplik väljalaske plaan

Tabel 11: Lõplik väljalaske plaan.

Täiendus	Keerukus	Planeeritud iteratsioon	Tehtud iteratsioon	Tundide arv	Tehtud
0.	8		1	43	Jah
1.	8	3	5	38	Jah
2.	3	1	2	21	Ei
3.	1	1	2	5	Jah
4.	1	1	2	6	Jah
5.	2	1	3	11	Jah
6.	1	1	2	3	Jah
7.	3	2	4	17	Ei
8.	1	2	2	3	Jah
9.	1	2	2	6	Jah
10.	1	2	3	23	Ei
11.	2	2	3	5	Jah
12.	2	2	3	4	Jah
13.	1	4	3	2	Jah
14.	1	4	4	13	Jah
15.	1	4	4	3	Jah
16.	1	4	3	1	Jah
17.	3	4	5	10	Jah
18.	1	4	2	13	Jah
19.	1	4	4	5	Jah
20.	2	4	4	9	Jah
21.	8	5	6	46	Jah

Lisa 5 - Valideerijate vastused tagasiside küsitlusele

Tabel 12: Kui lihtne on sinu hinnangul uute funktsioonide leidmine (skaalal 1 väga keeruline - 5 väga lihtne)?

Vastaja	Hinnang
1	4
2	5
3	4
4	4
5	5
6	4
7	2

Tabel 13: Kui lihtne on sinu hinnangul uute funktsioonide kasutamine (skaalal 1 väga keeruline - 5 väga lihtne)?

Vastaja	Hinnang
1	5
2	4
3	4
4	4
5	5
6	5
7	3

Tabel 14: Kuidas hindad NppDB laienduse funktsionaalsust pärast täiendusi (skaalal 1 oluliselt halvem - 5 oluliselt parem)?

Vastaja	Hinnang
1	5
2	4
3	4
4	4
5	4
6	5
7	5

--	--

Tabel 15: Milline konkreetne uus funktsioon või täiendus oli sinu arvates kõige kasulikum? Palun põhjenda lühidalt.

Vastaja	Vastus
1	Eelmine versioon polnud üldse kasutajasõbralik. Kasvõi see '+' märk iga laiendatava kausta kõrval on varasema versiooniga võrreldes täiesti nagu öö ja päev. Kui enne ma ei saanud üldse iseseisvalt aru, kuidas seda kasutama peab, siis nüüd on see juba kasutatav.
2	Kõige kasulikumad on: * Andmebaasi tabeli hüpikmenüüst saab valida järgmised tegevused: All rows; Random 100 rows; Drop table (RESTRICT); Drop table (CASCADE); - sest need on tavalised päringud mida tihti testimisel vaja. Tüütu on neid käsitsi välja kirjutada. * Kiirklahvide spikri kuvamine kasutajale - Suureks abiks kasutatavuse osas. Ei teagi kuidas oleks muidu välja mõelnud, kuidas näiteks AI viipa genereerida. * Suurele keelemudelile leitud vea parandust küsiva viiba genereerimine - Väga kaasaegne feature/võimekus
3	Visuaalselt on parem hoomata päringu tulemust.
4	SQL Analüsaator ja AI prompti genereerimine
5	Close all result tabs - palju päringuid tehes on tüütu nende vahelehti ükshaaval kinni panna.
6	Andmebaasi tabeli hüpikmenüüst erinevate tegevuste valimise võimalus. Pakutud tegevused on väga tüüpilised ja kõige rohkem kasutatavad andmete seisu kontrollimiseks valitud tabelis. See oluliselt hoiab tööaega kokku ning võimaldab vältida korduvat SQL koodi kirjutamist. Lisaks, kuna funktsionaalsus juba eksisteerib, see tähendab, et seda võib veel edaspidi laiendada soovitud tegevustega.
7	Andmebaasi tabeli hüpikmenüü, kuna see tundus kõike efektiivsema funktsioonina

Tabel 16: Kas mõni uus funktsioon või täiendus tundus sinu arvates ebavajalik või halvasti teostatud? Kui jah, siis milline ja miks?

Vastaja	Vastus
1	Mulle tundub et drop table nupp võiks hoiatada või küsida kinnitust enne tabeli kustutamist, aga see on väike asi

Tabel 16: Kas mõni uus funktsioon või täiendus tundus sinu arvates ebavajalik või halvasti teostatud? Kui jah, siis milline ja miks?

Vastaja	Vastus
2	Ainuke koht mis tekitas veidi segadust oli AI viiba genereerimise info kiirklahvide aknas. "Generate AI prompt" tekitas tunde, et võin seda feature'it kasutada ükskõik millal või kus. Tegelikuses saab seda kasutada ainult siis kui analüüs on tuvastanud päringus vea.
3	
4	
5	Otseselt mitte, kuid natuke häirib, et "Show plugin shortcuts" vajutades avatakse infoaken koos MS alert heliga. Loomu poolest ei ole sel tegevusel aga midagi nõ valet, seega see alert heli on natuke segadust tekitav.
6	Võibolla "Suurele keelemudelile leitud vea parandust küsiva viiba genereerimine" ei ole antud etapis väga vajalik funktsionaalsus, kuna SQL vead on suhteliselt lihtsad ning vea prompti saab ka iseseisvalt kirjutada. Tänapäeval on aktuaalsemad juba sisseehitatud AI assistendid, kuid kujutan ette, et Notepad++ kontekstis see pigem ei ole võimalik. Seega järgmistes etapides saab keskenduda pigem SQL lausete ja andmebaasi haldamisega seotud funktsionaalsuse laiendamisele. Aga muidu on ikkagi väga huvitav featuur.
7	

Tabel 17: Kas kogesid mingeid probleeme või raskusi uute funktsioonide kasutamisel? Kui jah, siis milliseid?

Vastaja	Vastus
1	
2	Jah, AI viiba genereerimise info kiirklahvide aknas. "Generate AI prompt" tekitas tunde, et võin seda kasutada ka näiteks probleemideta päringu juures (võiks näiteks genereerida viiba päringu sisu seletamiseks).
3	Algselt esines probleem plugina installeerimisel, aga kui Notepad++ uuesti installeerida ja plugin ka, siis kõik toimis ilusti.
4	

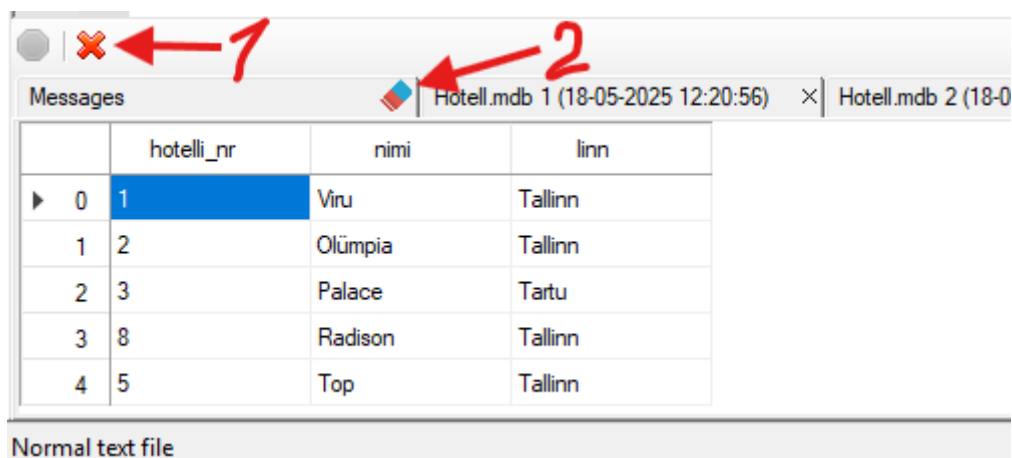
Tabel 17: Kas kogesid mingeid probleeme või raskusi uute funktsioonide kasutamisel? Kui jah, siis milliseid?

Vastaja	Vastus
5	Ehk ei ole otseselt seotud uute funktsioonidega, aga kui teha päringus süntaksi viga ning analüüsida lauset (Shift+F9), siis vea ilmnemisel kuvatakse vea põhjus ning päringu tulemuste kastis "Analysis found errors". Kui nüüd kustutada analüüsi tulemused (Ctrl+Shift+F9), siis vea põhjus kustub, kuid "Analysis found errors" jääb, mistõttu eelnevate päringute tulemuste juurde tagasi saamiseks tuleb spetsiaalselt käivitada korrektne päring. See pärsib natuke kasutajamugavust.
6	Raskusi ega probleeme ei kogenud, katsetamine läks sujuval, kõik vajalikud selgitused ja juhendid olid ka ette antud.
7	

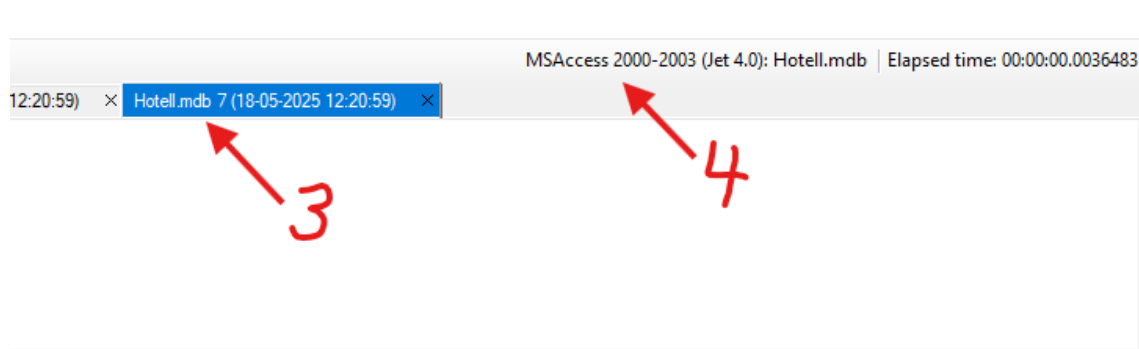
Tabel 18: Kui tõenäoliselt soovitaksid NppDB laienduse uuendatud versiooni oma kolleegile või sõbrale (skaalal 1 kindlasti ei soovitaks - 10 kindlasti soovitaksin)?

Vastaja	Hinnang (tõenäosus)
1	5
2	9
3	8
4	8
5	8
6	9
7	8

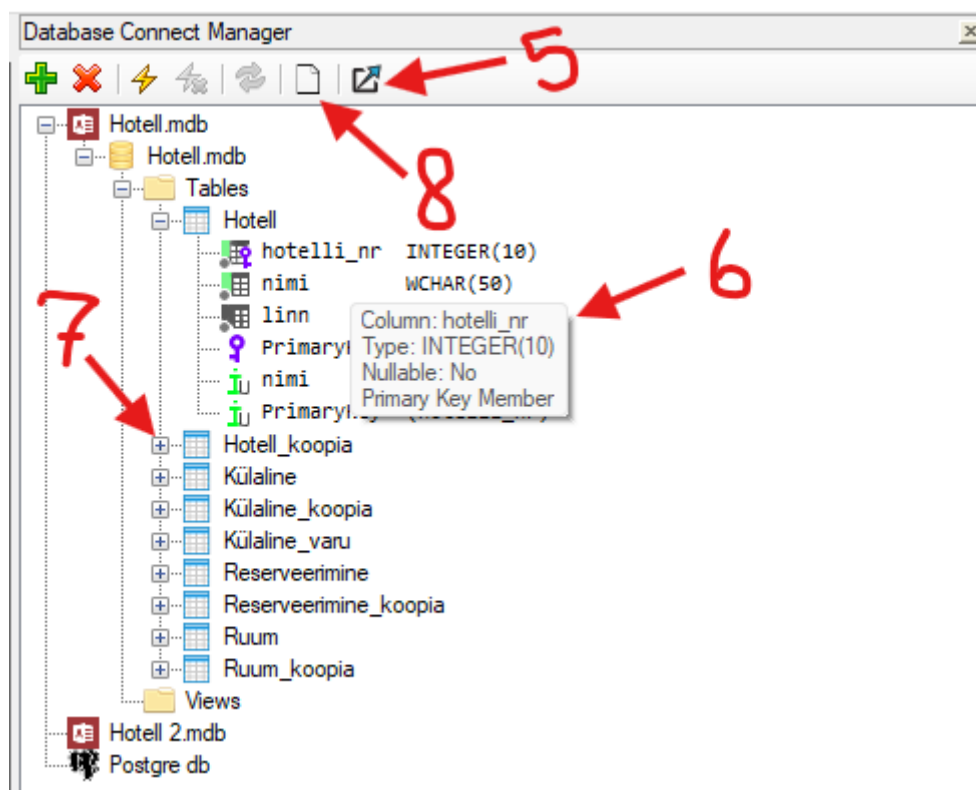
Lisa 6 - Kuvatõmmised täiendatud rakendusest



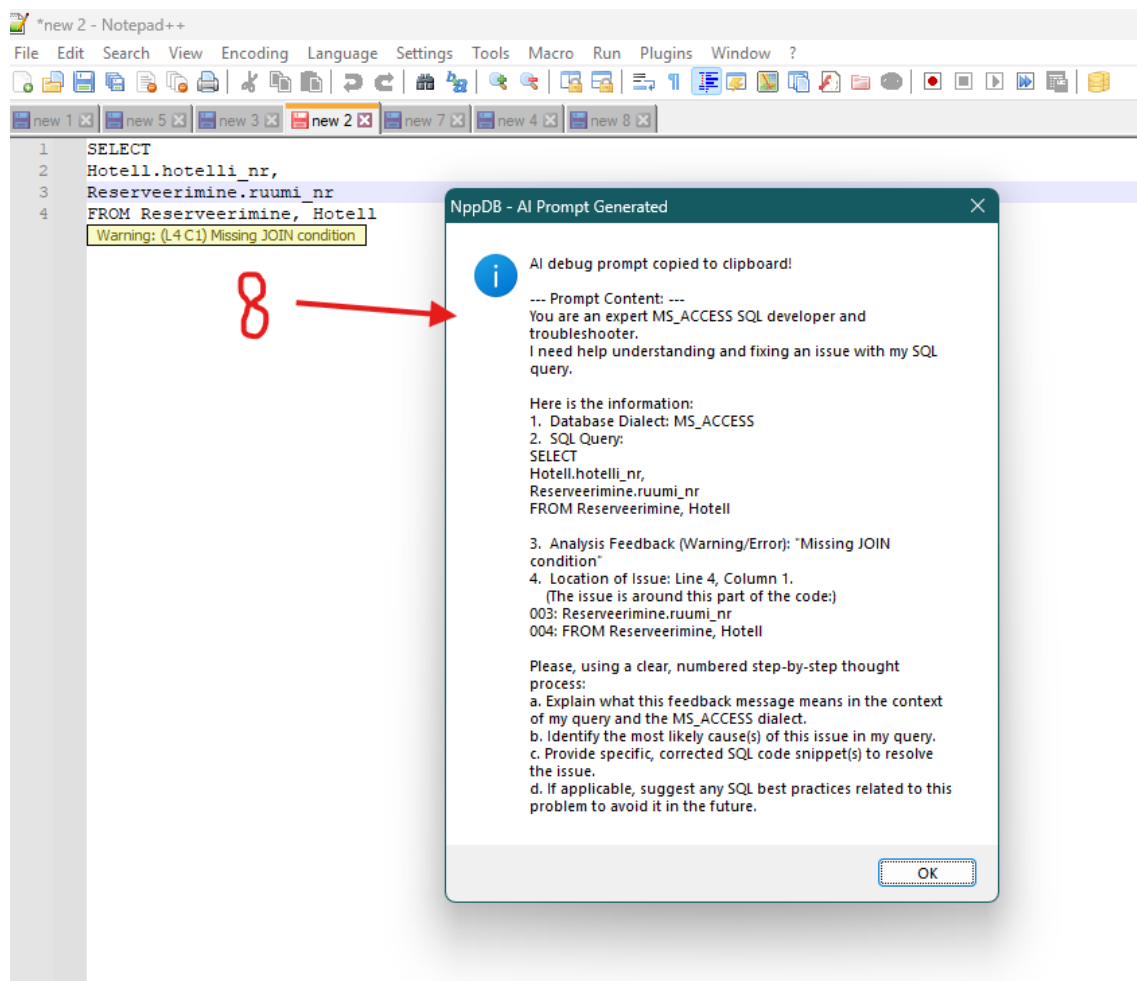
Joonis 1: Tulemuste akna täiendused 1-2.



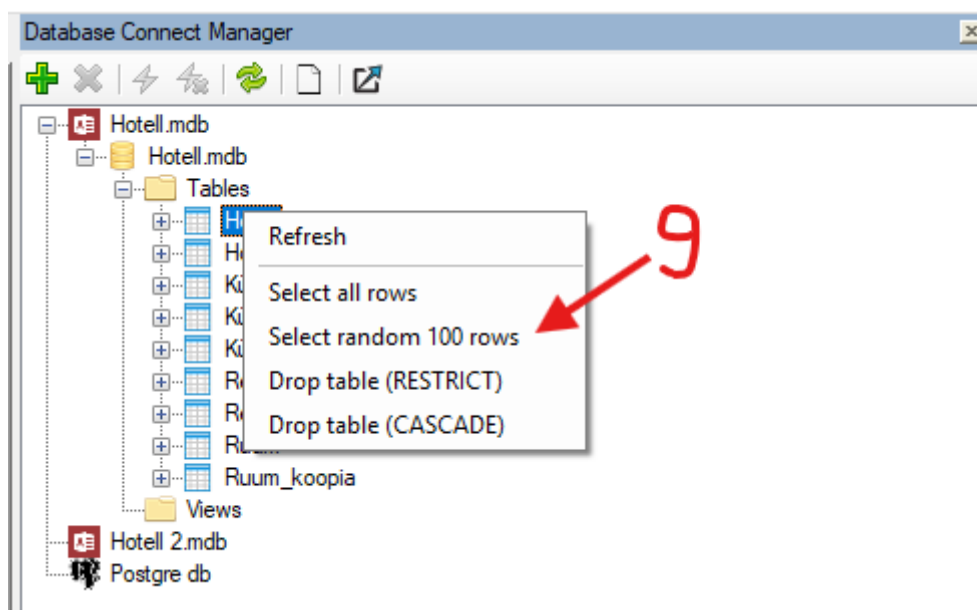
Joonis 2: Tulemuste akna täiendused 3-4.



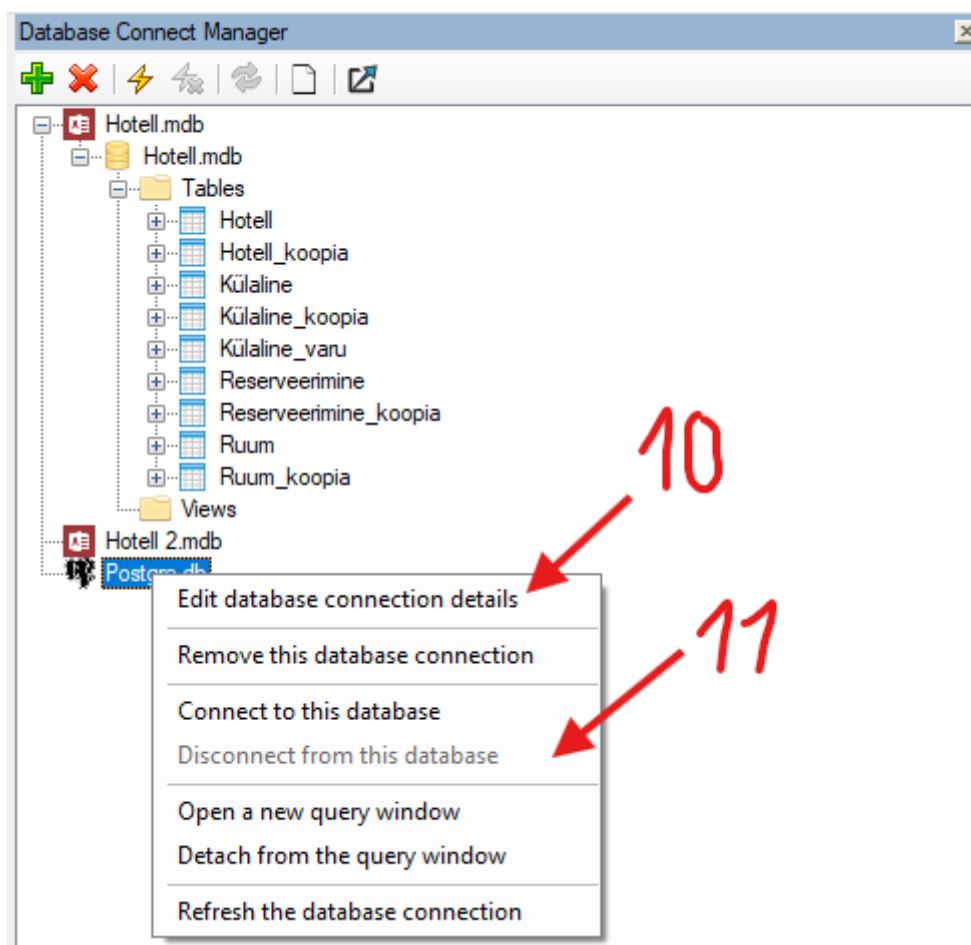
Joonis 3: Ühenduse akna täiendused 5-7.



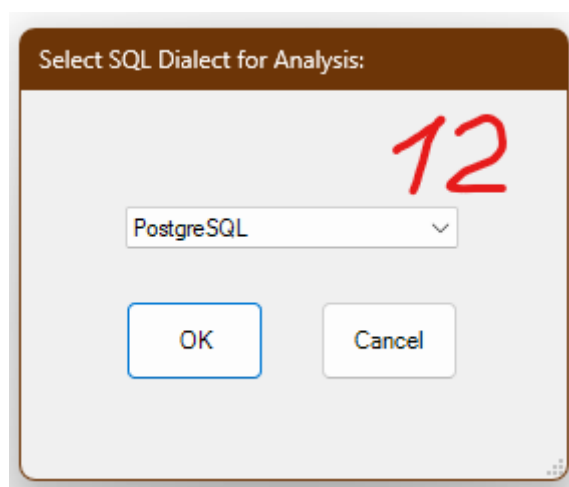
Joonis 4: Tulemuste akna täiendused 8.



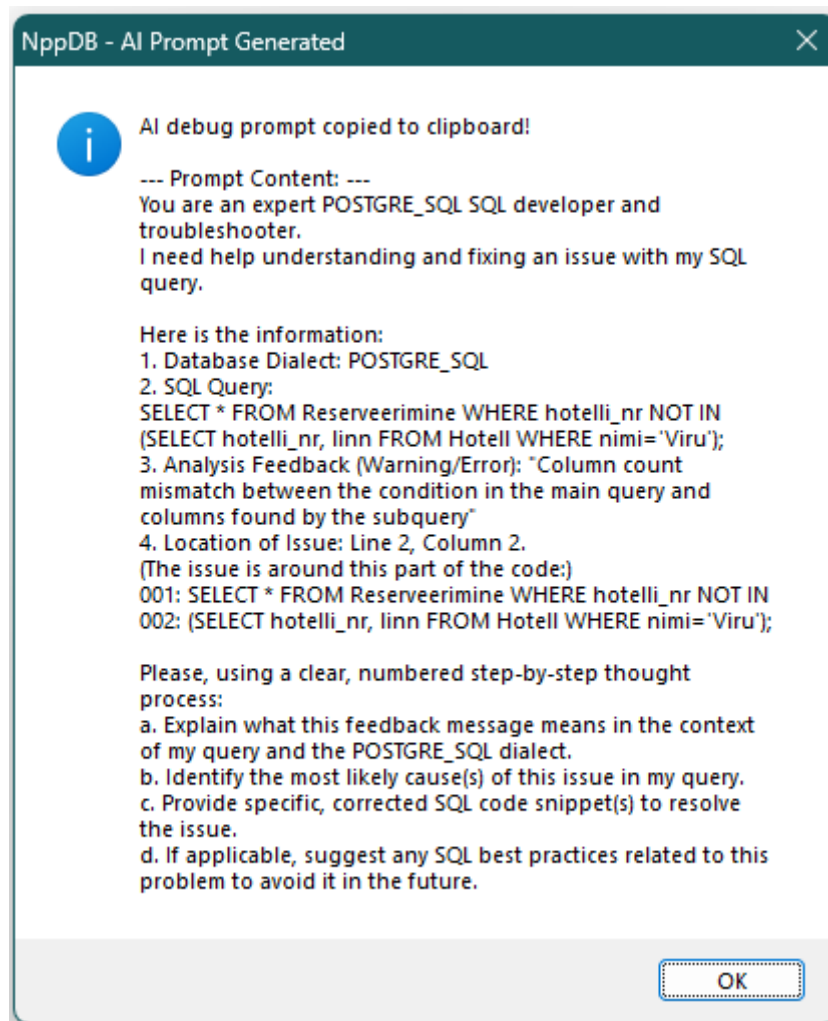
Joonis 5: Ühenduse akna täiendused 9.



Joonis 6: Ühenduse akna täiendused 10-11.



Joonis 7: Ühendusevaba analüüsi täiendus 12.



Joonis 8: Viip keelemudelile