

Tallinna Tehnikaülikool

Prillipoe infosüsteem "Optometrist"

Iseseisev töö

Tristan Krass

179302IADB33

Juhendaja: Priit Rospel

TALLINN 2019

Autori deklaratsioon

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

30.12.2019

(kuupäev)

(allkiri)

Sisukord

Sissejuhatus.....	4
1. Ülesande püstitus	5
2. Analüüs	6
3. Projekt.....	7
3.1 Olemi-suhte diagramm.....	7
3.2 Olemite semantika	7
3.3 Olemite omadused	9
3.4 Projekteeritud andmebaasi loomise skript	26
3.5 Andmebaasi loomine	27
3.6 Andmebaasi täitmine kirjetega	28
3.7 Andmebaasist päringute tegemine	31
Kokkuvõte.....	34
Kasutatud allikad	35
Lisa 1. Andmebaasi loomise skript.....	36
Lisa 2. Olemi suhte diagramm	42

Sissejuhatus

Nägemine on üks viiest meelest, kuid seda võib lugeda kõige olulisemaks, sest ligi 80% informatsioonist tuleb nägemismeelte kaudu (Eagle Vision). Aina enam veedab inimene oma aega ekraani taga. Sellest tulenevalt on aina rohkematel inimestel probleeme nägemisega. Antud infosüsteem küll ei lahenda inimeste nägemise probleemi, kuid annab prillipoodidele, optometristidele ja klientidele mugava keskkonna, mille kaudu on võimalik broneerida aegu erinevate optometristide juurde, teha veebipõhiseid silmateste ja osta kõik vajalik silmade tervise heaks.

Iseseisva töö raames on tehtud analüüs probleemile ja ära kirjeldatud, milline võiks antud infosüsteemi andmemudel välja näha. Lisaks on olemas olemi-suhte diagramm ja vajalik skript andmebaasi loomiseks.

1. Ülesande püstitus

Eesmärk on luua infosüsteem, mille abil on inimestel võimalik kontrollida oma silmanägemist ja broneerida aeg optometri juurde. Lisaks sellele on võimalik klientidel ühendada oma andmed digilooga, et ka patsientide perearstidel oleks võimalik patsiendi andmetele ligi pääseda.

2. Analüüs

Esialgse minimaalse töötava toote (MVP) juurde peavad kindlasti kuuluma silmade kontrolltestid, mida on võimalik igal kliendil teha kodus arvuti tagant lahkumata. Kontrolltestid peaksid olema niivõrd usaldusväärsed, et anda tagasisidet, kas patsiendil tuleks pöörduda optomeetri juurde või mitte. Reeglina soovitatakse optomeetri juures käia kord kahe aasta tagant, mis võib igapäeva tegemiste kõrvalt ununeda (Instrumentarium). Sellest tulenevalt teavitab süsteem kliente meili teel ja sõnumi teel, kui on aeg uuesti silmatesti teha.

Lisaks patsientidele, kes teevad silmateste, on süsteemi võimalik kasutada ka optometristidel, silmaarstidel ja administraatoritel. Optometristil peab olema võimalik näha kõikide klientide silmatestide tulemuste andmeid, kes on selleks loa andnud. Andmete põhjal on optometristil võimalik teha otsus, kas patsient peaks siiski optomeetri poole pöörduma või mitte. Täna on võimalik optometristiks õppida Tallinna Tervishoiu Kõrgkoolis, mille õppekaval on ettenähtud 48 EAP-d praktikat (Tallinna Tervishoiu Kõrgkool). Seega, kui osa patsiente on nõus jagama oma kontrolltestide tulemusi, siis on tudengitel võimalus õppida ja anda patsientidele tagasisidet, mis annab võimaluse tunnis õpitut kinnistada ja osa praktikamahust täita.

Selleks, et patsientide elu veel mugavamaks muuta on olemas võimalus tellida e-poe vahendusel kõik vajaminevad vahendid silmaterviseks. E-pood on eraldiseisev moodul silmapoe infosüsteemist ning ehitades antud lahendust mikroteenuste põhimõtetal, siis oleks tegu ühega mitmest mikroteenusest.

Viimaks peaks patsiendil olema võimalik broneerida aeg optomeetri juurde. Kusjuures vastuvõtt võib toimuda veebis või prillipoes. Broneeringusüsteem hõlbustab optomeetri tööd, sest optometristil on võimalik paremini planeerida oma päeva ning tekib võimalus ka osa tööst teha kodust, kui kõik vastuvõtud antud päevaks on planeeritud veebis.

3. Projekt

3.1 Olemi-suhte diagramm

Olemi suhte diagramm on leitav iseseisva töö lõpust peatükist Lisa 2 (vt Lisa 2).

3.2 Olemite semantika

Tabel 1. Olemite semantika.

Tabeli nimi	Semantika
Hind	Siin tabelis hoitakse kõiki teenuste ja toodetega seotud hindasid.
Teenus	Siin on kirjas pakutava teenuse või toote nimetus/nimi.
TeenuseLiik	Siin on kirjeldatud ära spetsiifiline teenus, mida pakutakse. Näiteks optometri sti vastuvõtt.
ArveRida	Siin on kirjas vastava teenuse või toote nimetus ja kogus.
Arve	Siin on kajastatud kõik arveread, arve päis ja arveridade summa. Käibemaks ja muu oluline, mis käib arve juurde.
Makse	On vajalik selleks, et pidada järge, millal keegi on millist teenust või toodet ostnud või tellinud.
MakseLiik	Siin on kirjeldatud, millise maksega on tegu (kaardimakse, sularaha).

Amet	Siin on ära kirjeldatud isiku roll prillipoes või kliinikus, kus programmi kasutatakse. Ameteid on alati võimalik juurde lisada.
Isik	Siin tabelis hoitakse kõikide inimeste andmeid olenemata sellest, kas nad on arstid või kliendid või keegi muu. Tabelis on kõik, kes meie platvormile registreerivad või ka isikud, kes pole veel ise registreerinud, kuid müügitiim on nad lisanud kui potentsiaalsed kliendid.
IsikAmetis	Kuna isikul saab olla mitu ametit ja ühel ametil saab töötada mitu isikut, siis on vaja vahetabelit, mis hoiustaks, mis isik on seotud konkreetse ametiga. Siin tabelis on kirjas isikuId ja ametiId.
Linn	On vajalik selleks, et pidada järge, millises linnas ükski prillipood on.
PrilliPood	Siin tabelis on ühe prillipoe andmed (alates, kuni ja nimetus). Atribuute on kindlasti tulevikus võimalik juurde lisada.
PrilliPoodLinnas	Tegu on vahetabeliga, mis on vajalik kirjeldamiseks seost linna ja prillipoe vahel.
Kasutaja	Kasutaja võib olla klient, arst, optometrist või keegi teine, kes kasutab antud programmi.
KasutajaGrupp	On grupp kasutajatest, kus on ära kirjeldatud antud gruppi kuuluvate kasutajate õigused. Näiteks süsteemadministratoorid, kellel on õigus teha enamik võimalikest operatsioonidest.
KasutajaGrupis	Kirjeldab ära, millisesse gruppi konkreetne kasutaja kuulub.
GrupiOigus	Siin tabelis on ära kirjeldatud erinevate gruppide õigused.
Tegevus	Siin on ära kirjeldatud, milline tegevus on lubatud.

TegevusGrupp	Tegevuste loetelu. Siin on mitu tegevust kokku lisatud. Näiteks õigus lugeda, muuta ja uuendada dokumente.
SilmaKontroll	See on kogu mudeli kõige tähtsam tabel. Siin tabelis hoitakse silmakontrolli tulemusi.
Aeg	Siin tabelis on kirjas, millal keegi vastuvõtule tuleb.
Ruum	Füüsiline ruum, kus on kirjeldatud ära ruumi omadused. Ruumi on võimalik broneerida.
IsikRuumis	Vahetabel ruumi ja isiku vahel. On vajalik selleks, et pidada arvet, kes millises ruumis parasjagu on.

3.3 Olemite omadused

Tabel 2. Olemi Isik atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
IsikId	INTEGER	NOT NULL	Tabeli Isik primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Isik ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.

Isikukood	VARCHAR	NULL	Inimese isikukood, mille abil on võimalik saada lisainformatsiooni inimese kohta ning hõlbustab siduda patsiendi andmeid digilooga.
Eesnimi	VARCHAR	NOT NULL	Isiku eesnimi.
Perenimi	VARCHAR	NOT NULL	Isiku perekonnanimi.
MuudNimed	VARCHAR	NULL	Isiku muud nimed.
Email	VARCHAR	NOT NULL	Isiku meiliaadress, mille kaudu on võimalik tulevikus talle teavitusi saata.
Aadress	VARCHAR	NULL	Isikuga kirja teel suhtlemiseks.
TelefoniNumber	VARCHAR	NULL	Isiku telefoninumber. Kui isik ei ilmu vastuvõtule, siis on võimalik ta kätte saada ka telefoni teel, et kinnitada, kas isik tuleb vastuvõtule või mitte.
Synnikuupäev	DATE	NOT NULL	Sünnikuupäev on vajalik, sest kõigil isikutel ei pruugi olla isikukoodi. Peale selle on see oluline määramaks, kas tegu on alaealise kontoga või täiskasvanu kontoga.
KontoNumber	VARCHAR	NULL	Selleks, et oleks võimalik teha isikule võimalikke tagasimakseid.
VasakuSilma-Tugevus	DECIMAL	NULL	Isiku vasaku silma tugevus.
ParemaSilma-Tugevus	DECIMAL	NULL	Isiku Parema silma tugevus.

Tabel 3. Olemi SilmaKontroll atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
SilmaKontrollId	INTEGER	NOT NULL	Tabeli SilmaKontroll primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist SilmaKontroll ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Kindla silmakontrolli (testi) nimetus. Meil võib olla erinevaid teste silma tugevuse kontrolliks.
KontrolliKuupaev	TIMESTAMP	NOT NULL	Kuupäev, millal patsient silmakontrolli sooritas. Lisatakse andmebaasi automaatselt teiste andmete sisestamise hetkel.
VasakSilm- Tulemus	VARCHAR	NOT NULL	Siin on kirjas kõik relevantne, mis on seotud vasaku silmaga testi sooritamise kohta, et hiljem oleks võimalik andmete põhjal järeldusi teha. On salvestatud küll sõnena andmebaasi, kuid tegelikult käsitleme rakenduse kihis seda JSONi kujul. Uurides erinevaid andmebaasijuhtimissüsteeme leidsin, et PostgreSQL andmebaas toetab ka JSONi andmetüüpi (PostgreSQL 9.4.25 Documentation, 1996, ptk 8.14). Seega kasutades Postgres andmebaasi on võimalik salvestada ka tulemust lihtsalt JSONi kujul.
ParemSilm- Tulemus	VARCHAR	NOT NULL	Siin on kirjas kõik relevantne, mis on seotud parema silmaga testi sooritamise kohta, et hiljem oleks võimalik andmete põhjal järeldusi teha. On salvestatud

			küll sõnena andmebaasi, kuid tegelikult käsitleme rakenduse kihis seda JSONi kujul. Uurides erinevaid andmebaasijuhtimissüsteeme leidsin, et PostgreSQL andmebaas toetab ka JSONi andmetüüpi (PostgreSQL 9.4.25 Documentation). Seega kasutades Postgres andmebaasi on võimalik salvestada ka tulemust lihtsalt JSONi kujul.
Kommentaar	VARCHAR	NULL	Vabas vormis kommentaar, mis võib ka tühjaks jääda. Kommentaari saab lisada, kui midagi läks kontrolli ajal valesti või esines muid kaebusi.
AegId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab ajale, millal silmakontroll tehti.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kes silmakontrolli sooritas.

Tabel 4. Olemi IsikAmetis atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
IsikAmetisId	INTEGER	NOT NULL	Tabeli IsikAmetis primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist IsikAmetis ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Alates	DATE	NOT NULL	Kuupäev, millal isik alustas tööd konkreetsel ametil.
Kuni	DATE	NOT NULL	Kuupäev, millal isik lõpetas töö konkreetsel ametil.

AmetId	INTEGER	NOT NULL	On vajalik, sest tegu on vahetabeliga ameti ja isiku vahel. Kuna on vaja meeles pidada, milline isik millisel ametil töötab, siis tänu AmetIdle on võimalik meeles pidada, millisel töökohal antud isik töötab.
IsikId	INTEGER	NOT NULL	IsikId kaudu on võimalik seostada amet kindla isikuga, kes töötab konkreetsel ametil.
PrilliPoodId	INTEGER	NOT NULL	On vajalik selleks, et leida pood, kus antud isik töötab. Kuna prillipoes on mitu ametit, siis on võimalik seostada kõik konkreetsed tööd konkreetse prillipoega.

Tabel 5. Olemi Amet atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
AmetId	INTEGER	NOT NULL	Tabeli Amet primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Amet ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Ameti nimetus. Näiteks optometrist.

Tabel 6. Olemi Makse atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
MakseId	INTEGER	NOT NULL	Tabeli Makse primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Makse ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Kuupaev	DATETIME	NOT NULL	Kuupaev, millal mingi teenuse või toote eest maksti.
MakseLiikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab Makseliigi tabelile, kus on ära kirjeldatud, millist liiki maksega on tegu.
ArveId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab sellele arvele, mille eest klient maksis.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kellele arve väljastati.

Tabel 7. Olemi MakseLiik atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
MakseLiikId	INTEGER	NOT NULL	Tabeli MakseLiik primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist MakseLiik ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.

Maskseviis	CHAR	NOT NULL	On kahe- või ühetäheline lühend sellest, milline antud maksetüüp on. Kui tegu on kaardimaksega, siis võiks lühend olla näiteks CP ehk Card Payment.
------------	------	----------	---

Tabel 8. Olemi Arve atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
ArveId	INTEGER	NOT NULL	Tabeli Arve primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Arve ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
ArveValjastatud	TIMESTAMP	NOT NULL	Kuupäev, millal arve tehti ja saadeti isikule.
Viitenumber	VARCHAR	NULL	Viitenumber, mille abil on võimalik seostada isikut ja arvet. Võib olla ka tühi.
ArveTahtaeg	DATE	NOT NULL	Kuupäev, mis hetkeks peab arve olema tasutud.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kellele arve väljastati.
KasutajaId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab kasutajale, kellele arve väljastati.

Tabel 9. Olemi ArveRida atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
ArveRidaId	INTEGER	NOT NULL	Tabeli ArveRida primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist ArveRida ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Toote või teenuse nimetus konkreetset arvereaal.
Kommentaar	VARCHAR	NULL	Soovi korral saab lisada arvereaale ka kommentaari. Näiteks prillidega tuleb kaasa tasuta puhastusvedelik.
Kogus	INTEGER	NOT NULL	Näitab kui palju mingit toodet või teenust kokku osteti. Vaikimisi on väärtus 1.
ArveId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab arvele, millega antud arverida seotud on.
TeenusId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab sellele teenusele, mida müüdi.

Tabel 10. Olemi Teenus atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
TeenusId	INTEGER	NOT NULL	Tabeli Teenus primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Teenus ja liites

			sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Teenuse nimetus. Näiteks optomeetri visiit.
TeenuseLiikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab sellele teenuseliigile, mida müüdi.

Tabel 11. Olemi TeenuseLiik atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
TeenuseLiikId	INTEGER	NOT NULL	Tabeli TeenuseLiik primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist TeenuseLiik ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Teenuseliigi nimetus. Näiteks kontaktläätsete määramine.

Tabel 12. Olemi Hind atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
HindId	INTEGER	NOT NULL	Tabeli Hind primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Hind ja liites

			sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Alates	DATE	NOT NULL	Kuupäev, mis hetkest antud hind kehtima hakkas.
Kuni	DATE	NOT NULL	Kuupäev, mis hetkest antud hind kehtivuse lõpetas.
Hind	DECIMAL	NOT NULL	Hind, mis on kohustuslik lisada igale teenusele või tootele, mida müüakse.
TeenusId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab sellele teenusele, mida müüdi.

Tabel 13. Olemi PrilliPoodLinnas atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
PrilliPood- LinnasId	INTEGER	NOT NULL	Tabeli PrilliPoodLinnas primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist PrilliPoodLinnas ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Alates	DATE	NOT NULL	Kuupäev, millal prillipood antud linnas avati.
Kuni	DATE	NOT NULL	Kuupäev, millal prillipood antud linnas suleti.

LinnId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab linnale, kus antud prillipood asub.
PrilliPoodId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab prillipoele, mis asub antud linnas.

Tabel 14. Olemi PrilliPood atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
PrilliPoodId	INTEGER	NOT NULL	Tabeli PrilliPood primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist PrilliPood ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimi	VARCHAR	NOT NULL	Igal prillipoel on nimi.
Alates	DATE	NOT NULL	Kuupäev, millal prillipood avati.
Kuni	DATE	NOT NULL	Kuupäev, millal prillipood suleti.

Tabel 15. Olemi Linn atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
LinnId	INTEGER	NOT NULL	Tabeli Linn primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Linn ja liites sellele

			ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimi	VARCHAR	NOT NULL	Igal linnal peab olema nimi.

Tabel 16. Olemi Kasutaja atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
KasutajaId	INTEGER	NOT NULL	Tabeli Kasutaja primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Kasutaja ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
KasutajaNimi	VARCHAR	NOT NULL	Iga kasutaja saab valida endale kasutajanime.
Alates	DATE	NOT NULL	Kuupäev, millal konto loodi.
Kuni	DATE	NOT NULL	Kuupäev, millal konto kustutati ära.
YlemKasutaja-Id	INTEGER	NULL	Välisvõti (<i>Foreign key</i>), mis viitab samale kasutajate tabelile. Osa kasutajatel võib olla mitu kasutajat. Näiteks lapsevanemad kontrollivad laste kontosid.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kellele antud kasutaja kuulub.

Tabel 17. Olemi Ruum atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
RuumId	INTEGER	NOT NULL	Tabeli Ruum primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Ruum ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Igal ruumil on kindel nimetus. Näiteks optometri sti kabinet.
Alates	DATE	NOT NULL	Kuupäev, millal antud ruumi hakati kasutama.
Kuni	DATE	NOT NULL	Kuupäev, millal antud ruumi enam ei kasutata.
PrilliPoodId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab prillipoele, kus antud ruum asub.

Tabel 18. Olemi IsikRuumis atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
IsikRuumisId	INTEGER	NOT NULL	Tabeli IsikRuumis primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist IsikRuumis ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Alates	DATE	NOT NULL	Visiidi alguse aeg.

Kuni	DATE	NOT NULL	Visiidi lõpp.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kes antud ruumis viibis.
RuumId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab ruumile, kus isik või isikud visiidi ajal viibisid.

Tabel 19. Olemi Aeg atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
AegId	INTEGER	NOT NULL	Tabeli Aeg primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Aeg ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi
Alates	DATE	NOT NULL	Aeg, millal alustati silmakontrolli või millal on planeeritud silmakontrolli aeg.
Kuni	DATE	NOT NULL	Aeg, millal silmakontroll lõppes.
IsikId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab isikule, kellele antud aeg on broneeritud.
RuumId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab ruumile, kuhu antud aeg on broneeritud.

Tabel 20. Olemi KasutajaGrupis atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
Kasutaja-GrupisID	INTEGER	NOT NULL	Tabeli KasutajaGrupis primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist KasutajaGrupis ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Alates	DATE	NOT NULL	Kuupäev, mis tähistab kasutaja kuulumist kindlasse gruppi.
Kuni	DATE	NOT NULL	Kuupäev, mis tähistab kasutaja kuulumise lõppemist kindlasse gruppi.
KasutajaId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab kasutajale, kes antud kasutajagrupperi kuulub.
Kasutajagrupp-Id	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab kasutajagrupile, kuhu antud kasutaja kuulub.

Tabel 21. Olemi Tegevus atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
TegevusId	INTEGER	NOT NULL	Tabeli Tegevus primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist Tegevus ja liites

			sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi
Nimetus	VARCHAR	NOT NULL	Igal tegevusel peab olema nimetus, mis kirjeldaks ära tegevuse.
TegevusKood	VARCHAR	NOT NULL	Tegevuse unikaalne kood, mille abil on võimalik samanimelisi tegevusi omavahel eristada.
Alates	DATE	NOT NULL	Kuupäev, mis tähistab kindla tegevuse jõustumise kuupäeva.
Kuni	DATE	NOT NULL	Kuupäev, mis tähistab kindla tegevuse lõppu.
TegevuseGrupp- Id	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab tegevusgrupile, kuhu antud tegevus kuulub.

Tabel 22. Olemi TegevuseGrupp atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
TegevuseGrupp- Id	INTEGER	NOT NULL	Tabeli TegevuseGrupp primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist TegevuseGrupp ja liites sellel ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Igal tegevusgrupil on nimetus, mis peaks olema iseloomulik tegevusgrupile.

Alates	DATE	NOT NULL	Kuupäev, millal tegevusgrupp loodi.
Kuni	DATE	NOT NULL	Kuupäev, millal tegevusgrupp kustutati.
YlemTegevuse-GruppId	INTEGER	NULL	Välisvõti (<i>Foreign key</i>), mis viitab samale tabelile, (TegevuseGrupp) kui tegevusgrupil on ülem tegevusgruppe.

Tabel 23. Olemi GrupiOigus atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/ NOT NULL	Semantika
GrupiOigusId	INTEGER	NOT NULL	Tabeli GrupiOigus primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist GrupiOigus ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Igal grupiõigusel on iseloomustav nimetus.
Alates	DATE	NOT NULL	Kuupäev, millal grupiõigus kehtima hakkas.
Kuni	DATE	NOT NULL	Kuupäev, millal grupiõigus kehtivuse lõpetas.
TegevusId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab kindlale tegevusele, mis grupiõigustesse kuulub.

KasutajaGruppId	INTEGER	NOT NULL	Välisvõti (<i>Foreign key</i>), mis viitab kindlale kasutajagrupile, mis grupiõigustesse kuulub.
-----------------	---------	-------------	--

Tabel 24. Olemi KasutajaGrupp atribuutide semantika.

Veeru Nimi	Andmetüüp	NULL/NOT NULL	Semantika
Kasutaja- GruppId	INTEGER	NOT NULL	Tabeli KasutajaGrupp primaarvõti (<i>Primary key</i>), mis omistatakse uue kirje lisamisel võttes senise maksimaalse ID väärtuse tabelist KasutajaGrupp ja liites sellele ühe. See on peidetud võti, mida ei näidata kasutajale kunagi.
Nimetus	VARCHAR	NOT NULL	Informatiivne nimetus antud kasutajagrupile.
Alates	DATE	NOT NULL	Kuupäev, millal antud kasutajagrupp loodi.
Kuni	DATE	NOT NULL	Kuupäev, millal antud kasutajagrupp kustutati.
Kirjeldus	VARCHAR	NULL	Kommentaar või kirjeldus, mis annab lisainformatsiooni kasutajagrupi kohta.

3.4 Projekteeritud andmebaasi loomise skript

Projekteeritud andmebaasi loomise skript on leitav iseseisva töö lõpust peatükist Lisa 1 (vt Lisa 1).

3.5 Andmebaasi loomine

Kasutan andmebaasijuhtsüsteemina PostgreSQL andmebaasi. Selleks, et mitte alla tõmmata vajaminevaid draivereid ja PostgreSQLi ennast, kasutasin dockerit. See on väga lihtne - vaja läheb dockerit (tuleb alla laadida dockeri kodulehelt), klienti ning natuke oskusi käsureal (vt Joonis 1).

```
$ docker run -d -p 5432:5432 -e POSTGRES_USER=tristan  
POSTGRES_PASSWORD>Hello5orld --name raspa postgres
```

Joonis 1. Dockeri konteineri jooksutamine käsurealt.

Nüüd jookseb konteineri sees PostgreSQL andmebaas, kuhu saan ühendada pordi 5432 kaudu.

Seejärel sisenen jooksvasse konteinerisse, kus saan alustada käskude täitmist (vt Joonis 2).

```
$ docker exec -it postgres1 sh
```

Joonis 2. Dockeri konteinerisse sisenemise käsklus käsurealt.

Seejärel loon andmebaasi (vt Joonis 3).

```
# createdb -U postgres optometrist
```

Joonis 3. Andmebaasi loomine dockeri konteinerisse käsurealt.

Andmebaasi nimega optometrist juurde tuleb nüüd 23 tabelit. Töö tegemiseks kasutan DataGrip programmi.

3.6 Andmebaasi täitmine kirjetega

Valin tabelid Amet, IsikAmetis ja Isik, et täita need andmetega. Andmebaasi täites selgub, et tabelil IsikAmetis on välisvõti PrilliPoodId ehk iga isik, kes töötab konkreetsel ametil on seotud ka kindla prillipoega. Seetõttu täidan ka neljanda tabeli - Prillipood andmetega.

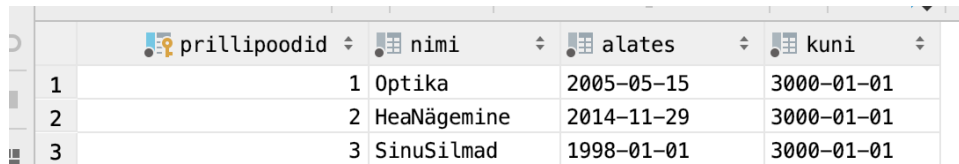
Sisestan prillipoed tabelisse Prillipood (vt Joonis 4).

```
INSERT INTO PrilliPood (Nimi, Alates, Kuni)
VALUES ('Optika', '2005-05-15', '3000-01-01'),
       ('HeaNägemine', '2014-11-29', '3000-01-01'),
       ('SinuSilmad', '1998-01-01', '3000-01-01');
```

Joonis 4. Prillipoodide lisamine tabelisse Prillipood.

Kontrollin, kas protseduur oli edukas (vt Joonis 5).

```
SELECT * FROM PrilliPood;
```



	prillipoodid	nimi	alates	kuni
1	1	Optika	2005-05-15	3000-01-01
2	2	HeaNägemine	2014-11-29	3000-01-01
3	3	SinuSilmad	1998-01-01	3000-01-01

Joonis 5. Prillipoodide kontrollimine tabelist PrilliPood.

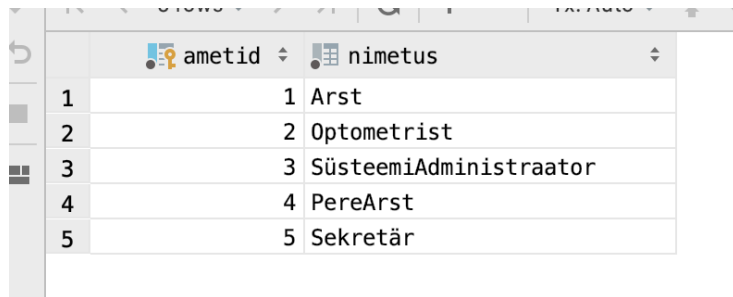
Sisestan ametid tabelisse Amet (vt Joonis 6).

```
INSERT INTO Amet (Nimetus)
VALUES ('Arst'),
       ('Optometrist'),
       ('SüsteemiAdministraator'),
       ('PereArst'),
       ('Sekretär');
```

Joonis 6. Ametite lisamine tabelisse Amet.

Kontrollin, kas protseduur oli edukas (vt Joonis 7).

```
SELECT * FROM Amet;
```



The screenshot shows a database query result in a table format. The table has two columns: 'ametid' and 'nimetus'. The data is as follows:

	ametid	nimetus
1	1	Arst
2	2	Optometrist
3	3	SüsteemiAdministraator
4	4	PereArst
5	5	Sekretär

Joonis 7. Ametite kontrollimine tabelist Amet.

Sisestan isikud tabelisse Isik (vt Joonis 8).

```

INSERT INTO Isik (Isikukood, Eesnimi, Perenimi, Email, SynniKuupaev, Sugu,
VasakuPrilliTugevus,
                ParemaPrilliTugevus, Address)
VALUES ('39806302730', 'Tristan', 'Krass', 'tristan@frontava.ee', '1998-06-30', 'M', -
2.25, -1.75, 'Sambliku 34'),
        ('49710125624', 'Katriin', 'Tahm', 'kairiintahm@gmail.com', '1998-06-30', 'N',
-5, 2, 'Akadeemia Tee 7'),
        ('34503035120', 'Morgen', 'Tjorven', 'morgentjorver@outlook.com', '1945-03-03',
'M', 0, 0, 'Koidu 27'),
        ('49208202132', 'Kaireen', 'Rebane', 'kaireenrebane@hotmail.com', '1992-08-20',
'N', 2.50, 3.00, 'Meeliku 24'),
        ('39705124624', 'Mark', 'Ventsla', 'markvenstsla@gmail.com', '1997-05-12', 'M',
-3.25, -3.0, 'Tuule tee 15');
INSERT INTO Isik(Eesnimi, Perenimi, Email, SynniKuupaev, Sugu)
VALUES ('Gustaf', 'Ögon', 'gustafogon@gmail.com', '1954-08-18', 'M'),
        ('Laura', 'Sild', 'sildlaura@hotmail.com', '2001-01-01', 'N');

INSERT INTO Isik(Isikukood, Eesnimi, Perenimi, Email, SynniKuupaev, Sugu)
VALUES ('46712240571', 'Marta', 'Gustnetsova', 'drgustnetsova@gmail.com', '1967-12-
24', 'N'),
        ('44409120034', 'Sandra', 'Ilus', 'SandraIlus@optika.ee', '1944-09-12', 'N'),
        ('48511210276', 'Maarja', 'Laus', 'Maarja@sinusilmad.ee', '1985-11-21', 'N');

```

Joonis 8. Isikute lisamine tabelisse Isik.

Kontrollin, kas operatsioon oli edukas (vt Joonis 9).

```
SELECT * FROM Isik;
```

isikid	isikukood	eesnimi	perenimi	muudnimed	email	kontonumber	synnikuupaev	sugu	vasakuprillitugevus	paremprillitugevus	address
1	39806302730	Tristan	Krass		tristan@frontava.ee		1998-06-30	M	-2.25	-1.75	Sambliku 34
2	49710125624	Katriin	Tahm		kairiintahm@gmail.com		1998-06-30	N	-5.00	2.00	Akadeemia Tee 7
3	34503035120	Morgen	Tjorven		morgentjorver@outlook.com		1945-03-03	M	0.00	0.00	Koidu 27
4	49208202132	Kaireen	Rebane		kaireenrebane@hotmail.com		1992-08-20	N	2.50	3.00	Meeliku 24
5	39705124624	Mark	Ventsla		markvenstsla@gmail.com		1997-05-12	M	-3.25	-3.00	Tuule tee 15
6		Gustaf	Ögon		gustafogon@gmail.com		1954-08-18	M			
7		Laura	Sild		sildlaura@hotmail.com		2001-01-01	N			
8	46712240571	Marta	Gustnetsova		drgustnetsova@gmail.com		1967-12-24	N			
9	44409120034	Sandra	Ilus		SandraIlus@optika.ee		1944-09-12	N			
10	48511210276	Maarja	Laus		Maarja@sinusilmad.ee		1985-11-21	N			

Joonis 9. Isikute kontrollimine tabelist Isik.

Viimaks sisestan andmed ka tabelisse IsikAmetis (vt Joonis 10).

```
INSERT INTO IsikAmetis(Alates, Kuni, AmetId, PrilliPoodId, IsikId)
VALUES ('2005-05-15', '3000-01-01', 1, 1, 8),
       ('2018-02-25', '3000-01-01', 2, 1, 9),
       ('2010-11-01', '3000-01-01', 5, 3, 10),
       ('2000-07-10', '2010-04-13', 2, 3, 6);
```

Joonis 10. Isikute ja andmete lisamine tabelisse IsikAmetis.

Kontrollin, kas operatsioon oli edukas (vt Joonis 11).

```
SELECT * FROM IsikAmetis;
```

	isikametisid	alates	kuni	ametid	prillipoodid	isikid
1	1	2005-05-15	3000-01-01	1	1	8
2	2	2018-02-25	3000-01-01	2	1	9
3	3	2010-11-01	3000-01-01	5	3	10
4	4	2000-07-10	2010-04-13	2	3	6

Joonis 11. Isikute ja andmete kontrollimine tabelist IsikAmetis.

3.7 Andmebaasist päringute tegemine

Teen päringu üle ühe tabeli (vt Joonis 12). Eesmärk on leida kõik isikud, kes on meessoost ja sündinud pärast 1990ndat aastat. Soovin isikute kohta teada nende e-maili ning ees- ja perekonnanime.

```
SELECT Eesnimi, Perenimi, Email FROM Isik WHERE SynniKuupaev >= '1990-01-01' AND Sugu = 'M';
```

Joonis 12. Meessoost isikute leidmine, kes on sündinud pärast 1990ndat aastat tabelist Isik.

Kontrollin saadud tulemust (vt Joonis 13).

	eesnimi	perenimi	email
1	Tristan	Krass	tristan@frontava.ee
2	Mark	Ventsla	markvenstsla@gmail.com

Joonis 13. Meessoost isikute tulemused tabelist Isik.

Teen päringu üle kahe tabeli (vt Joonis 14). Teen päringu üle IsikAmetis ja Amet tabeli.

Eesmärk on leida kõik optometristid ja sorteerida alguskuupäeva alusel kasvavas järjekorras ehk kes on kõige varem tööle asunud.

```
SELECT ia.Alates, a.Nimetus
FROM isikametis ia
INNER JOIN Amet A on ia.AmetId = A.AmetId
WHERE A.Nimetus = 'Optometrist'
ORDER BY Alates ASC;
```

Joonis 14. Kõige varem tööle asunud optometristide leidmine.

Kontrollin saadud tulemust (vt Joonis 15).

	alates	nimetus
1	2000-07-10	Optometrist
2	2018-02-25	Optometrist

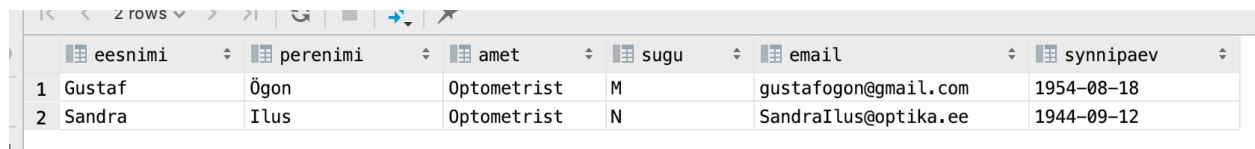
Joonis 15. Kõige varem tööle asunud optometristide tulemus.

Teen päringu üle kolme tabeli (vt Joonis 16). Soovin teada saada kõikide optometristide eesnime, perekonnanime, soo, sünnikuupäeva, ameti ja e-maili. Selleks on mul vaja teha päring üle IsikAmetis, Isik ja Amet tabelite. Lisaks sellele soovin tulemuse saada järjestatult nooremast vanemani.

```
SELECT I.Eesnimi AS Eesnimi,  
       I.Perenimi AS pereNimi,  
       a.Nimetus AS amet,  
       i.Sugu AS Sugu,  
       I.Email AS email,  
       i.SynniKuupaev AS Synnipaev  
FROM isikametis ia  
     INNER JOIN Amet A on ia.AmetId = A.AmetId  
     INNER JOIN Isik I on ia.IsikId = I.IsikId  
WHERE A.Nimetus = 'Optometrist'  
ORDER BY I.SynniKuupaev DESC;
```

Joonis 16. Optometristide ees-, perekonnanime, soo, sünnipäeva, ameti ja e-maili leidmine vanuse järgi kasvavalt.

Kontrollin saadud tulemust (Joonis 17).



	eesnimi	perenimi	amet	sugu	email	synnipaev
1	Gustaf	Õgon	Optometrist	M	gustafogon@gmail.com	1954-08-18
2	Sandra	Ilus	Optometrist	N	SandraIlus@optika.ee	1944-09-12

Joonis 17. Optometristide ees-, perekonnanime, soo, sünnipäeva, ameti ja e-maili tulemus vanuse järgi kasvavalt.

Kokkuvõte

Infosüsteemi optimeerimiseks esialgne olemissuhte diagramm sai nii mõnegi täienduse töö tegemise jooksul. Peamiselt muutusid olemite atribuudid, millest osa tuli ära kustuda ja osa juurde lisada. Projekti raames sain koostada andmemudeli, mille põhjal tegin ka andmebaasi skripti. Infosüsteemi loomise käigus sain esmakordselt kasutada ka PostgreSQL andmebaasisüsteemi. Nagu ITs ikka, siis ei saa kunagi kindel olla, et antud andmemudel 100% töötab, seega rakenduskihi arendamise vältel võib selguda, et midagi on jäänud kahe silma vahele, kuid leian, et antud mudel on piisav esialgse minimaalse toote arendamiseks. Esimesed sammud rakenduskihi arenduseks on ka võetud. Minu Github kontolt (GitHub) on võimalik tutvuda ka esialgse serveripoolse arenduse lahendusega.

Kasutatud allikad

Eagle Vision. Silmakontroll. [WWW]

https://www.eaglevision.ee/teenus/silmakontroll/?utm_source=google&utm_medium=cp&utm_campaign=Search_Silmakontroll_EE&gclid=CjwKCAiAuqHwBRAQEiwAD-zr3XvI5z24-PofdzmoiQBNK9JhNKBCh31X2znEY-8fjUzsTcr-GdDSJBoCTRcQAvD_BwE (29.12.2019)

Github. [WWW] <https://github.com/tristankrass/Optometrlist> (27.12.2019)

Instrumentarium. [WWW] <https://www.instru.ee/et/hea-teada> (01.02.2020)

PostgreSQL 9.4.25 Documentation. (1996). [WWW]

<https://www.postgresql.org/docs/9.4/datatype-json.html> (27.12.2019)

Tallinna Tervishoiu Kõrgkool. Optometrist. [WWW] <https://www.ttk.ee/et/optometrist> (01.02.2020)

Lisa 1. Andmebaasi loomise skript

```
CREATE TABLE Isik
(
    IsikId          SERIAL PRIMARY KEY,
    Isikukood      VARCHAR(20),
    Eesnimi        VARCHAR(50) NOT NULL,
    Perenimi       VARCHAR(50) NOT NULL,
    MuudNimed     VARCHAR(100),
    Email          VARCHAR(65) NOT NULL,
    KontoNumber   VARCHAR(25),
    SynniKuupaev  DATE          NOT NULL,
    Sugu          CHAR(1)      NOT NULL,
    VasakuPrilliTugevus DECIMAL(10, 2),
    ParemaPrilliTugevus DECIMAL(10, 2),
    Aadress       VARCHAR(100)
);
```

```
CREATE TABLE Amet
(
    AmetId SERIAL PRIMARY KEY,
    Nimetus VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE PrilliPood
(
    PrilliPoodId SERIAL PRIMARY KEY,
    Nimi         VARCHAR(50) NOT NULL,
    Alates      DATE          NOT NULL,
    Kuni        DATE          NOT NULL
);
```

```

CREATE TABLE Kasutaja
(
    KasutajaId SERIAL PRIMARY KEY,
    KasutajaNimi VARCHAR(50) NOT NULL,
    Alates TIMESTAMP NOT NULL,
    Kuni DATE NOT NULL,
    IsikId INTEGER REFERENCES Isik (IsikId) NOT NULL,
    YlemKasutajaId INTEGER REFERENCES Kasutaja (KasutajaId)
);

```

```

CREATE TABLE Arve
(
    ArveId SERIAL PRIMARY KEY,
    ArveValjastatatud TIMESTAMP NOT NULL,
    ArveTahtaeg DATE NOT NULL,
    ViiteNumber VARCHAR(65),
    IsikId INTEGER References Isik (IsikId) NOT NULL,
    KasutajaId INTEGER References Kasutaja (KasutajaId) NOT NULL
);

```

```

CREATE TABLE MakseLiik
(
    MakseLiikId SERIAL PRIMARY KEY,
    MakseViis CHAR(2) NOT NULL
);

```

```

CREATE TABLE Makse
(
    MaksmineId SERIAL PRIMARY KEY,
    Kuupaev DATE NOT NULL,
    ArveId INTEGER REFERENCES Arve (ArveId) NOT NULL,
    MakseLiikId INTEGER REFERENCES MakseLiik (MakseLiikId) NOT NULL,
    IsikId INTEGER REFERENCES Isik (IsikId) NOT NULL
);

```

```

CREATE TABLE IsikAmetis
(
    IsikAmetisId SERIAL PRIMARY KEY,
    Alates        DATE                                NOT NULL,
    Kuni          DATE                                NOT NULL,
    AmetId        INTEGER REFERENCES Amet (AmetId)    NOT NULL,
    PrilliPoodId INTEGER REFERENCES PrilliPood (PrilliPoodId) NOT NULL,
    IsikId        INTEGER REFERENCES Isik (IsikId)     NOT NULL
);

CREATE TABLE TeenuseLiik
(
    TeenuseLiikId SERIAL PRIMARY KEY,
    Nimetus        VARCHAR(50) NOT NULL
);

CREATE TABLE Teenus
(
    TeenusId        SERIAL PRIMARY KEY,
    Nimetus        VARCHAR(50)                                NOT NULL,
    TeenuseLiikId INTEGER REFERENCES TeenuseLiik (TeenuseLiikId) NOT NULL
);

CREATE TABLE ArveRida
(
    ArveRidaId SERIAL PRIMARY KEY,
    Nimetus        VARCHAR(50)                                NOT NULL,
    Kommentaar     VARCHAR(255),
    Kogus          integer                                    NOT NULL DEFAULT 1,
    ArveId         INTEGER REFERENCES Arve (ArveId)          NOT NULL,
    TeenusId       INTEGER REFERENCES Teenus (TeenusId)      NOT NULL
);

CREATE TABLE Hind
(
    HindId        SERIAL PRIMARY KEY,
    Alates        DATE                                NOT NULL,
    Kuni          DATE                                NOT NULL,
    TeenusId      INTEGER REFERENCES Teenus (TeenusId)      NOT NULL
);

```

CREATE TABLE Linn

```
(
  LinnId SERIAL PRIMARY KEY,
  Nimi VARCHAR(50) NOT NULL
);
```

CREATE TABLE Ruum

```
(
  RuumId SERIAL PRIMARY KEY,
  Nimetus VARCHAR(50) NOT NULL,
  Alates DATE NOT NULL,
  Kuni DATE NOT NULL,
  PrilliPoodId INTEGER REFERENCES PrilliPood (PrilliPoodId) NOT NULL
);
```

CREATE TABLE PrilliPoodLinnas

```
(
  PrilliPoodLinnasId SERIAL PRIMARY KEY,
  Alates DATE NOT NULL,
  Kuni DATE NOT NULL,
  LinnId INTEGER REFERENCES Linn (LinnId) NOT NULL,
  PrilliPoodId INTEGER REFERENCES PrilliPood (PrilliPoodId) NOT NULL
);
```

CREATE TABLE IsikRuumis

```
(
  IsikRuumisId SERIAL PRIMARY KEY,
  Alates DATE NOT NULL,
  Kuni DATE NOT NULL,
  IsikId INTEGER REFERENCES Isik (IsikId) NOT NULL,
  RuumId INTEGER REFERENCES Ruum (RuumId) NOT NULL
);
```

CREATE TABLE Aeg

```
(
  AegId SERIAL PRIMARY KEY,
  Alates DATE NOT NULL,
  Kuni DATE NOT NULL,
  IsikId INTEGER REFERENCES Isik (IsikId) NOT NULL,
  IsikRuumisId INTEGER REFERENCES IsikRuumis (IsikRuumisId) NOT NULL
);
```

```

CREATE TABLE SilmaKontroll
(
    SilmaKontrollId SERIAL PRIMARY KEY,
    Nimetus          VARCHAR(50)          NOT NULL,
    VasakuSilmaTulemus JSONB              NOT NULL,
    ParemaSilmaTulemus JSONB              NOT NULL,
    Kommentaar       VARCHAR(255),
    AegId            INTEGER REFERENCES Aeg (AegId) NOT NULL,
    IsikId           INTEGER REFERENCES Isik (IsikId) NOT NULL
);

```

```

CREATE TABLE KasutajaGrupp
(
    KasutajaGruppId SERIAL PRIMARY KEY,
    Nimetus          VARCHAR(50) NOT NULL,
    Alates          DATE          NOT NULL,
    Kuni            DATE          NOT NULL,
    Kirjeldus       VARCHAR(255)
);

```

```

CREATE TABLE KasutajaGrupis
(
    KasutajaGrupisId SERIAL PRIMARY KEY,
    Alates            DATE          NOT NULL,
    Kuni             DATE          NOT NULL,
    KasutajaId       INTEGER REFERENCES Kasutaja (KasutajaId) NOT NULL,
    KasutajaGruppId INTEGER REFERENCES KasutajaGrupp (KasutajaGruppId) NOT NULL
);

```

```

CREATE TABLE TegevuseGrupp
(
    TegevuseGruppId SERIAL PRIMARY KEY,
    Nimetus          VARCHAR(50) NOT NULL,
    Alates          DATE          NOT NULL,
    Kuni            DATE          NOT NULL,
    YlemTegevuseGruppId INTEGER REFERENCES TegevuseGrupp (TegevuseGruppId)
);

```



```

CREATE TABLE Tegevus
(
    TegevusId        SERIAL PRIMARY KEY,
    Nimetus          VARCHAR(50)          NOT NULL,
    Alates           DATE                 NOT NULL,
    Kuni             DATE                 NOT NULL,
    TegevusKood      VARCHAR(25)         NOT NULL,
    TegevuseGruppId INTEGER REFERENCES TegevuseGrupp (TegevuseGruppId) NOT NULL
);

```

```

CREATE TABLE Grupi0igus
(
    Grupi0igusId    SERIAL PRIMARY KEY,
    Nimetus          VARCHAR(25)          NOT NULL,
    Alates           DATE                 NOT NULL,
    Kuni             DATE                 NOT NULL,
    TegevusId        INTEGER REFERENCES Tegevus (TegevusId)          NOT NULL,
    KasutajaGruppId INTEGER REFERENCES KasutajaGrupp (KasutajaGruppId) NOT NULL
);

```