TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

{Nutitelefonide e-pood}

{Andmebaasid II IDU0230}

|  |  |
| --- | --- |
| Üliõpilane: | Kairit Sims,  Tanel Tõemets,  Raido Õunapuu |
| Õpperühm: | IABB53 |
| Matrikli nr: | 154767,  155605,  154811 |
| E-mail: | kairitsims@gmail.com  tanel.toemets@gmail.com  raidoounapuu@gmail.com |

|  |  |
| --- | --- |
| Juhendaja: | Erki Eessaar |

Tallinn

2017

**Sisukord**

[1 Strateegiline analüüs 7](#_Toc501191041)

[1.1 Terviksüsteemi üldvaade 7](#_Toc501191042)

[1.1.1 Organisatsiooni eesmärgid 7](#_Toc501191043)

[1.1.2 Infosüsteemi eesmärgid 7](#_Toc501191044)

[1.1.3 Lausendid 8](#_Toc501191045)

[1.1.4 Põhiobjektid 9](#_Toc501191046)

[1.1.5 Põhiprotsessid 9](#_Toc501191047)

[1.1.6 Põhilised sündmused 10](#_Toc501191048)

[1.1.7 Tegutsejad 11](#_Toc501191049)

[1.1.8 Asukohad 12](#_Toc501191050)

[1.1.9 Terviksüsteemi tükeldus allsüsteemideks 12](#_Toc501191051)

[1.2 Kaupade funktsionaalse allsüsteemi eskiismudelid 14](#_Toc501191052)

[1.2.1 Eesmärgid 14](#_Toc501191053)

[1.2.2 Allsüsteemi kasutavad pädevusalad 14](#_Toc501191054)

[1.2.3 Allsüsteemi poolt vajatavad registrid 14](#_Toc501191055)

[1.2.4 Allsüsteemi ühe põhiprotsessi tegevusdiagramm 15](#_Toc501191056)

[1.2.5 Allsüsteemi kasutusjuhtude eskiismudel 16](#_Toc501191057)

[1.2.6 Mittefunktsionaalsed nõuded 18](#_Toc501191058)

[1.3 Kaupade registri eskiismudelid 21](#_Toc501191059)

[1.3.1 Eesmärgid 21](#_Toc501191060)

[1.3.2 Registrit kasutavad pädevusalad 21](#_Toc501191061)

[1.3.3 Registrit teenindavad funktsionaalsed allsüsteemid 21](#_Toc501191062)

[1.3.4 Infovajadused, mida register aitab rahuldada 21](#_Toc501191063)

[1.3.5 Seosed teiste registritega 22](#_Toc501191064)

[1.3.6 Ärireeglid 22](#_Toc501191065)

[1.3.7 Registri kontseptuaalne eskiismudel 23](#_Toc501191066)

[2 Detailanalüüs 24](#_Toc501191067)

[2.1 Kaupade funktsionaalse allsüsteemi detailanalüüs 24](#_Toc501191068)

[2.1.1 Kasutusjuhtude mudel 24](#_Toc501191069)

[2.2 Kaupade funktsionaalse allsüsteemi vajatavate registrite detailanalüüs 32](#_Toc501191070)

[2.2.1 Kontseptuaalne andmemudel 32](#_Toc501191071)

[2.2.2 Andmebaasioperatsioonide lepingud 41](#_Toc501191072)

[2.2.3 Registri põhiobjekti seisundidiagramm 47](#_Toc501191073)

[2.3 CRUD maatriks 48](#_Toc501191074)

[3 Füüsiline disain 49](#_Toc501191075)

[3.1 Kaupade funktsionaalse allsüsteemi vajatavate registrite füüsiline disain 49](#_Toc501191076)

[4 Realisatsioon PostgreSQLis 55](#_Toc501191077)

[4.1 Andmebaasi loomine 55](#_Toc501191078)

[4.2 Skeemid 55](#_Toc501191079)

[4.3 Domeenid 55](#_Toc501191080)

[4.4 Tabelid ja arvujada generaatorid 55](#_Toc501191081)

[4.5 Vaated 60](#_Toc501191082)

[4.6 Protseduursed keeled 61](#_Toc501191083)

[4.7 Trigeri funktsioonid ja trigerid 61](#_Toc501191084)

[4.8 Reeglid 63](#_Toc501191085)

[4.9 Funktsioonid 63](#_Toc501191086)

[4.10 Indeksid 64](#_Toc501191087)

[4.10.1 Välisvõtmete veergudele lisatavad indeksid 64](#_Toc501191088)

[4.10.2 Täiendavad sekundaarsed indeksid 65](#_Toc501191089)

[4.10.3 Funktsioonil põhinevad indeksid 65](#_Toc501191090)

[4.11 Klassifikaatorite väärtustamise SQL laused 65](#_Toc501191091)

[4.12 JSON formaadis lähteandmete laadimine 68](#_Toc501191092)

[4.13 Täiendavate testandmete lisamine 69](#_Toc501191093)

[4.14 Andmebaasi statistika kogumine 69](#_Toc501191094)

[4.15 Päringu täitmisplaani näide 70](#_Toc501191095)

[4.16 Rollid ja kasutajad 70](#_Toc501191096)

[4.17 Üleliigsete õiguste äravõtmine 70](#_Toc501191097)

[4.18 Õiguste jagamine 70](#_Toc501191098)

[4.19 Andmebaasiobjektide kustutamine 70](#_Toc501191099)

[4.19.1 Õiguste äravõtmine 70](#_Toc501191100)

[4.19.2 Domeenide kustutamine 70](#_Toc501191101)

[4.19.3 Tabelite ja arvujada generaatorite kustutamine 70](#_Toc501191102)

[4.19.4 Vaadete kustutamine 71](#_Toc501191103)

[4.19.5 Indeksite kustutamine 72](#_Toc501191104)

[4.19.6 Funktsioonide ja trigerite kustutamine 72](#_Toc501191105)

[4.19.7 Reeglite kustutamine 72](#_Toc501191106)

[4.19.8 Kasutajate ja rollide kustutamine 72](#_Toc501191107)

[4.19.9 Laienduste kustutamine 72](#_Toc501191108)

[4.20 Kasutatud materjalid 74](#_Toc501191109)

Autorideklaratsioon

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

[Üliõpialase nimi]

[Allkiri]

**Sissejuhatus (Andmebaasid II)**

Tabel 1 esitab ülevaate andmebaasi ja rakenduse realisatsioonist.

Tabel Sissejuhatus õppeaine "Andmebaasid II" töö tulemuse kohta.

|  |  |
| --- | --- |
| Kasutatav andmebaasisüsteem ja versioon | PostgreSQL 10.1 |
| Rakenduse loomise vahendid (arenduskeskkonnad, keeled ja nende versioonid) | MS Access 2016 |
| Andmebaasi server **(andmebaas PEAB olema apex.ttu.ee serveris)** | apex.ttu.ee |
| PostgreSQL andmebaasi nimi (kui kasutate PostgreSQLi) | t155605 |
| Oracle skeemi nimi (kui kasutate Oraclet) | - |
| Millise töökoha rakendus realiseerib? | juhataja |
| Rakenduse aadress (kui on veebirakendus) **(rakenduse panek apex.ttu.ee serverisse pole kohustuslik; aadress tuleb esitada ka pgApex ja Oracle APEX rakenduse puhul)** | - |
| Rakenduse nimi (kui rakendus on tehtud pgApex või Oracle APEX keskkonnas) | - |
| Kui tegemist on kahekihilise klient-server süsteemiga ja rakendus on kasutaja arvutis, siis mida peab kasutaja arvutis tegema, et seda rakendust kasutada | „Rakenduse kasutamiseks peab kasutaja arvutis olema PostgreSQL ODBC draiverid ja loodud PostgreSQL\_yhendus nimeline andmeühenduse spetsifikatsioon (data source). Andmeühenduse spetsifikatsioonis peavad server, andmebaas, kasutajanimi ja parool olema määramata.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem) |
| Kasutajanimi ja parool rakendusse sisselogimiseks | kasutajanimi : joy.hawkins@geekosis.name parool: sint |
| Kui rakenduse failil on eraldi parool (nt MS Accessi korral – ei ole kohustuslik), siis mis see on? | - |
| Millistes vormides (MS Accessi korral), lehekülgedel (pgApex või Oracle APEX korral), failides (mingis programmeerimiskeeles loodud rakenduse puhul) toimub pöördumine andmebaasiserveris talletatud rutiinide poole? Faili puhul palun näidata ka kataloog. | Ekraanivormil *Sissereg* nupul *Jätka* vajutamisel käivitatakse funktsioon *f\_on\_juhataja*, mis on mõeldud juhataja identifitseerimiseks (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem).  Ekraanivormil *Aktiivsed\_mitteaktiivsed\_kaubad* nupul *Lõpeta* vajutamisel käivitatakse funktsioon *f\_kauba\_lopetamine*, mis muudab kauba seisundi lõpetatuks. |
| Millistes vormides (MS Accessi korral), lehekülgedel (pgApex või Oracle APEX korral), failides (mingis programmeerimiskeeles loodud rakenduse puhul) toimub pöördumine andmebaasis loodud vaadete poole? Faili puhul palun näidata ka kataloog. | Ekraanivormil *Koik\_kaubad* pöördutakse vaate *koik\_kaubad* poole, mis kuvab kõikide kaupade nimekirja.  Ekraanivormil Kaupade\_detailid pöördutakse vaate *kaupade\_detailid* poole, mis kuvab konkreetse kauba detailsed andmed.  Ekraanivormil *Detailandmed* asub alamvorm Kaupade\_kategooriad, kust pöördutakse vaate *kaupade\_kategooriad* poole, mis kuvab kõik kategooriad, kuhu konkreetne kaup kuulub.  Ekraanivormil *Detailandmed* asub alamvorm Kaupade\_variandid, kust pöördutakse vaate *kaupade\_variandid* poole, mis kuvab kõik variandid, mis konkreetsel kaubal on.  Ekraanivormil *Kaupade\_koondaruanne* pöördutakse vaate *kaupade\_koondaruanne* poole.  Ekraanivormil *Aktiivsed\_mitteaktiivsed\_kaubad* pöördutakse vaate *aktiivsed\_mitteaktiivsed\_kaubad* poole, mis kuvab aktiivsete ja mitteaktiivsete kaupade nimekirja. |
| Kuhu (millisesse moodulisse, faili) on rakenduses kirjutatud andmebaasi-süsteemiga ühenduse loomiseks mõeldud andmebaasi kasutaja nimi (see peab olema minimaalsete vajalike õigustega kasutaja)? Faili puhul palun näidata ka kataloog. | Minimaalsete õigustega andmebaasi-süsteemiga ühenduse loomiseks mõeldud kasutajanimi(t155605\_juhataja) on kirjutatud moodulisse *Refreshing*, konstandi *username* väärtuseks (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem). |
| Kas rakendus suhtleb andmebaasisüsteemiga kui üks kasutaja või vastab igale lõppkasutajale eraldi andmebaasi kasutaja? | „Rakendus suhtleb andmebaasisüsteemiga kui üks kasutaja“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem). |
| Kas lõppkasutajate kasutajanimed ja paroolid on andmebaasis (kui jah, siis mis tabeli mis veergudes)? | Jah, kasutajanimeks on veerg *e\_meil*, mis asub tabelis *Isik*. Parooliks on veerg *parool,* mis asub samuti tabelis *Isik*. |
| Kuidas on andmebaasis salvestatud parool kaitstud? Kui kasutate räsiväärtuse leidmist, siis tuleb selgelt nimetada räsi arvutamise algoritm. Kui kasutate soola, siis tuleb see selgelt välja öelda ning kirjeldada selle genereerimist ja omadusi. | „Andmebaasis hoitakse parooli räsiväärtust. See väärtus leitakse andmebaasisüsteemi poolt kasutades sisendina avatekstina parooli ning süsteemi-genereeritud soola.  Parooli räsiväärtuse leidmiseks kasutatakse PostgreSQL funktsiooni crypt koos soola genereerimiseks mõeldud funktsiooniga gen\_salt. Räsiväärtuse leidmiseks kasutatakse Blowfish plokkšifril (krüptograafilisel algoritmil) põhinevat algoritmi, mille puhul võib parooli pikkus olla kuni 72 märki. Seda algoritmi eelistati, kuna Cybernetica AS (2013) hindab selle viie aasta jooksul turvaliseks primitiiviks (erinevalt algoritmidest DES ja MD5, mida crypt samuti toetab).  Veerus parool lubatud maksimaalne väärtuse suurus peab valitud algoritmi korral olema 60 märki (veerg on tüüpi VARCHAR(60)).  gen\_salt funktsioon võimaldab lisaks algoritmile määrata ka algoritmi kasutatavat korduste arvu (kui algoritm seda võimaldab). Mida suurem korduste arv, seda rohkem võtab aega räsiväärtuse arvutamine, aga ka selle murdmine. Blowfish algoritmi korral on vaikimisi korduste arv 6, kuid võimalik korduste arvu vahemik on 4 kuni 31. Määran korduste arvuks 11, mille puhul räsiväärtuse leidmisel ja kontrollimisel veel olulist töökiiruse langust ei täheldatud.  Funktsiooni crypt kasutamiseks on PostgreSQL andmebaasis CREATE EXTENSION lauset kasutades installeeritud lisamoodul pgcrypto. Installeerimise tulemusel loodavad skeemiobjektid paigutatakse skeemi public.  CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem) |
| Kuidas toimub kasutaja autentimine? | Kasutaja audentimiseks kasutatakse võrreldakse kasutaja sisestatud parooli põhjal genereeritud räsiväärtust andmebaasis salvestatud parooli räsiväärtusega.  Samuti kontrollitakse, et kasutaja töötaks juhatajana ja oleks seisundis, mis lubab tal süsteemi kasutada.  „See kontroll toimub serveris, et andmebaasis salvestatud räsiväärtus ei liiguks üle võrgu kliendi juurde.  Antud juhul viib kontrolli läbi andmebaasisüsteem. Kontrollimaks, kas kasutaja on sobivas seisundis juhataja, luuakse andmebaasis funktsioon f\_on\_juhataja(text, text). Funktsiooni väljakutsel on esimene argument emeil ja teine argument parool.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem) |
| Kas olete oma töö iseseisva töö hindamismudeli alusel enne esitamist ise üle kontrollinud? | Jah |
| Kui vastasite eelmises punktis „Jah“, siis milline oli punktisumma? | 59 (koos lisapunktidega) |

# Strateegiline analüüs

Selles peatükis vaadeldakse tervet infosüsteemi, leitakse selle allsüsteemid ning esitatakse ühele põhiobjektile vastava funktsionaalse allsüsteemi/registri paari eskiismudelid.

## Terviksüsteemi üldvaade

Järgnevalt esitatakse ülevaade nutitelefonide müügiga tegeleva e-poe toimimisest.

### Organisatsiooni eesmärgid

* Teenida omanikele kasumit
* Pakkuda head ja kiiret teenindust, mis jätaks klientidele hea mulje ning suurendaks võimalust, et nad saavad püsiklientideks ja soovitavad pakutavaid teenuseid ka oma tuttavatele
* Olla kõigile osapooltele usaldusväärne lepingupartner
* Pakkuda klientidele kõige uuemaid tootemudeleid
* Saavutada turuliidri positsioon

### Infosüsteemi eesmärgid

* Saada ülevaade organisatsiooniga seotud isikutest
* Saada ülevaade organisatsiooni töötajatest
* Saada ülevaade organisatsiooni klientidest
* Saada ülevaade organisatsiooni sõlmitud lepingutest
* Võimaldada klassifikaatorite abil andmete liigitamist ja seostamist seostamiseks väljaspool organisatsiooni vastutusala oleva informatsiooniga
* Saada ülevaade kaupadest, millega tehingute (transaktsioonide) tegemine on üks organisatsiooni põhieesmärk
* Võimaldada elektrooniliselt muuta erinevate kaupade andmeid
* Võimaldada klientidel tellida meie poolt pakutavaid kaupu
* Võimaldada klientidel jätta tagasisidet
* Võimaldada klientidel vormistada garantiijuhtum või kauba tagastus
* Võimaldada töötajatel määrata hinnareegleid ja teha sooduspakkumisi
* Saada ülevaade arvetest
* Saada ülevaade tarnetellimuste arvetest
* Saada ülevaade tellimustest
* Saada ülevaade tarnetellimustest
* Saada ülevaade garantiijuhtumistest ja kauba tagastustest
* Saada ülevaade hinnareeglitest ja sooduspakkumistest
* Saada ülevaade kliendi tagasisidest
* Saada ülevaade laoliikumistest
* Saada ülevaade lepingutest
* Saada ülevaade varadest
* Saada ülevaade partneritest
* Saada ülevaade töötaja töötamistest
* Võimaldada töötajal koostada töögraafikut
* Saada ülevaade töögraafikutest

### Lausendid

* Isikul on hetkeseisund
* Isiku seisundi liik on klassifikaator
* Töötaja on isik
* Töötajal on hetkeseisund
* Töötaja seisundi liik on klassifikaator
* Töötaja töötab ametis
* Amet on klassifikaator
* Klient on isik
* Kliendil on hetkeseisund
* Kliendi seisundi liik on klassifikaator
* Klient sõlmib lepingu
* Lepingul on hetkeseisund
* Lepingu seisundi liik on klassifikaator
* Töötaja registreerib kauba
* Kaupa iseloomustab null või rohkem kategooriat
* Kauba kategooria on klassifikaator
* Kaubal on hetkeseisund
* Kauba seisundi liik on klassifikaator
* Klient teostab klienditellimuse
* Töötaja täidab klienditellimuse
* Töötaja esitab partnerile tarnetellimuse
* Partner tarnib tellitud kauba
* Nutitelefonide e-pood on organisatsioon
* Partner on organisatsioon
* Töötaja töötab töögraafiku alusel
* Nutitelefonide e-pood omab vara
* Töötaja jälgib laoliikumisi
* Kaubal on bränd
* Bränd on klassifikaator
* Kaubal on variandid
* Variant on värv
* Värv on klassifikaator
* Kaubal on tüüp
* Kauba tüüp on nutitelefon
* Nutitelefonil on esimene kaamera
* Esimene kaamera on klassifikaator
* Nutitelefonil on tagumine kaamera
* Tagumine kaamera on klassifikaator
* Nutitelefonil on sisemälu
* Sisemälu on klassifikaator
* Nutitelefonil on diagonaal
* Diagonaal on klassifikaator
* Nutitelefonil on protsessor
* Protsessor on klassifikaator
* Nutitelefonil on ekraani resolutsioon
* Ekraani resolutsioon on klassifikaator
* Klient tellib kaupa
* Töötaja esitab arve
* Klient tasub arve
* Klient annab tagasisidet
* Klient soovib kaupa tagastada
* Klient teavitab garantiijuhtumist
* Töötaja määrab hinnareeglid
* Töötaja määrab sooduspakkumised

### Põhiobjektid

* Isik
* Töötaja
* Klient
* Klassifikaator
* Leping
* Kaup
* Klienditellimus
* Arve
* Partner
* Organisatsioon
* Sooduspakkumine
* Hinnareegel
* Töögraafik
* Töötamine
* Vara
* Kliendi tagasiside
* Laoliikumine
* Garantiijuhtum
* Tarnetellimus
* Tagastus

### Põhiprotsessid

* Isiku registreerimine
* Isiku surnuks märkimine
* Töötaja tööle võtmine
* Töötaja ametikoha muutmine
* Töötaja ajutiselt töölt vabastamine
* Töötaja puhkusele siirdumine
* Kliendi poolt tellimuse esitamine
* Klassifikaatori väärtuse lisamine
* Klassifikaatori väärtuse muutmine
* Lepingu sõlmimine
* Lepingu peatamine
* Lepingu ühepoolne katkestamine
* Lepingu pikendamine
* Kauba tellimine
* Kauba registreerimine
* Kauba unustamine
* Kauba aktiveerimine
* Kauba ajutiselt kasutusest eemaldamine (mitteaktiivseks muutmine)
* Kauba lõplikult kasutusest eemaldamine (lõpetamine)
* Klienditellimuse täitmine
* Klienditellimuse tagasilükkamine
* Arve väljastamine
* Arve maksmine
* Partneritega suhtlemine
* Organisatsiooni eesmärkide täitmine
* Hinnareegli määramine
* Sooduspakkumise määramine
* Töögraafiku koostamine
* Töötamine töögraafiku järgi
* Varade arvele võtmine
* Varade mahakandmine
* Kliendi tagasiside lugemine
* Laoliikumiste jälgimine
* Garantiijuhtumite menetlemine
* Tarnetellimuste esitamine partneritele
* Kauba tagastamine

### Põhilised sündmused

* Organisatsiooni vaatevälja satub uus isik, kellega organisatsioon soovib astuda mingil viisil lepingulistesse suhetesse
* Isik sureb
* Organisatsiooni tuleb tööle uus töötaja
* Töötaja liigub karjääriredelil
* Töötajat hakatakse kahtlustama organisatsiooni huve kahjustavas teos
* Töötaja võtab välja kasutamata puhkuse
* Klient valib e-poest välja talle sobiva toote
* Tekib vajadus uue klassifikaatori väärtuse lisamiseks (nt tänu sellele, et täienes rahvusvaheline standard või tänu sellele, et ettevõtte äriprotsesse otsustati muuta)
* Selgus, et klassifikaatori väärtuse registreerimisel oli tehtud viga
* Võetakse kasutusele uus klassifikaator
* Huvitatud osapool (isik või organisatsioon) soovib astuda organisatsiooniga vastastikku kasulikesse lepingulistesse suhetesse
* Vähemalt üks lepingu osapooltest teatab, et ta pole ajutiselt võimeline lepingus toodud tingimusi täitma
* Vähemalt üks lepingu osapooltest teatab, et ta pole püsivalt võimeline lepingus toodud tingimusi täitma
* Lepingu osapooled on oma lepingulise suhtega rahul ja soovivad selle pikendamist
* Organisatsiooni jõuab teave uue kauba kohta
* Selgus, et organisatsiooni jõudnud teave kauba kohta on enneaegne ning sellisel kujul kaupa ei ole vaja registreerida
* On vaja muuta võimalikuks kauba kasutamine tehingutes
* Kauba kasutamine tehingutes on vaja ajutiselt peatada, kuna seoses kaubaga on ilmnenud ajutise iseloomuga probleemid
* Kauba kasutamine tehingutes on vaja lõpetada, kuna seoses kaubaga on ilmnenud püsiva iseloomuga probleemid või kuna kaup on oma aja lihtsalt ära elanud
* Klient tellib kaupa
* Tellimuse põhjal koostatakse arve
* Kliendile edastatakse arve
* Klient jätab tellitud kauba eest esitatud arve tasumata
* Partnerite otsimine
* Organisatsiooni registreerimine äriregistris
* Organisatsiooni eesmärkide püstitamine
* Kauba sisseostuhinnale maksude ja juurdehindluse protsendi lisamine ning väljamüügihinna väljaselgitamine
* Kliendile soodsamate hindade pakkumise soov
* Töötajate töötamiste reguleerimisvajaduse ilmnemine
* Töögraafiku jõusse viimine
* Ettevõtte käsutusse saabub uus vara
* Teostatakse inventuur ettevõtte varadele
* Ettevõtte käsutuses olevat vara enam ei eksisteeri, s.t vara on hävitatud, hävinud või kadunud
* Klient annab talle osutatud teenuste kohta tagasisidet
* Uute toodete tarnetellimuse põhjal saabumine
* Klienditellimuste täitmine ehk kauba väljasaatmine
* Klient vormistab garantiijuhtumi
* Soov tellida uut kaupa
* Soov täiendada olemasoleva kauba varusid
* Klient ei ole tellitud kaubaga rahul

### Tegutsejad

* Juhataja (ka omanik)
* Kauba haldur
* Klassifikaatorite haldur
* Klient
* Uudistaja
* Kliendihaldur

### Asukohad

* Kliendid (on süsteemis registreeritud) ja uudistajad (veebikülalised; tuvastamata kasutajad) kasutavad veebirakendust, mille poole pöördumiseks on vaja arvutit, veebilehitsejat ja veebiühendust.
* Töötajad töötavad neile spetsiaalselt ettenähtud ruumides. Igale töötajale on ettenähtud oma arvuti.

### Terviksüsteemi tükeldus allsüsteemideks

Järgnevalt esitatakse infosüsteemi jaotus kolme erinevat liiki allsüsteemideks.

Organisatsiooni sisesed pädevusalad.

* Juhataja
* Kauba haldur
* Klassifikaatorite haldur
* Kliendihaldur

Organisatsiooni välised pädevusalad.

* Klient
* Uudistaja

Tabel 2 esitab sisulised funktsionaalsed allsüsteemid ja nende teenindatavad registrid (seotud organisatsiooni põhitegevusega).

Tabel Sisulised allsüsteemid.

|  |  |
| --- | --- |
| Funktsionaalne allsüsteem | Register, mida see funktsionaalne allsüsteem teenindab |
| Klientide funktsionaalne allsüsteem | Klientide register |
| Kaupade funktsionaalne allsüsteem | Kaupade register |
| Klienditellimuste funktsionaalne allsüsteem | Klienditellimuste register |
| Arvete funktsionaalne allsüsteem | Arvete register |
| Sooduspakkumiste funktsionaalne allsüsteem | Sooduspakkumiste register |
| Kliendi tagasiside funktsionaalne allsüsteem | Kliendi tagasiside register |
| Garantiijuhtumite funktsionaalne allsüsteem | Garantiijuhtumite register |
| Tagastuse funktsionaalne allsüsteem | Tagastuste register |
| Hinnareeglite funktsionaalne allsüsteem | Hinnareeglite register |

Tabel 3 esitab administratiivsed funktsionaalsed allsüsteemid ja nende teenidatavad registrid (võivad olla kasutusel paljudes erinevate eesmärkide ja tegevusaladega organisatsioonides).

Tabel Administratiivsed allsüsteemid.

|  |  |
| --- | --- |
| Funktsionaalne allsüsteem | Register, mida see funktsionaalne allsüsteem teenindab |
| Isikute funktsionaalne allsüsteem | Isikute register |
| Töötajate funktsionaalne allsüsteem | Töötajate register |
| Klassifikaatorite funktsionaalne allsüsteem | Klassifikaatorite register |
| Lepingute funktsionaalne allsüsteem | Lepingute register |
| Varade funktsionaalne allsüsteem | Varade register |
| Partnerite funktsionaalne allsüsteem | Partnerite register |
| Töögraafikute funktisonaalne allsüsteem | Töögraafikute register |
| Laoliikumiste funktsionaalne allsüsteem | Laoliikumiste register |
| Tarnetellimuste funktsionaalne allsüsteem | Tarnetellimuste register |
| Organisatsioonide funktsionaalne allsüsteem | Organisatsioonide register |
| Töötamise funktsionaalne allsüsteem | Töötamise register |

## Kaupade funktsionaalse allsüsteemi eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

* Muuta võimalikuks kaupade kasutamine erinevates tehingutes (transaktsioonides), mille läbiviimist infosüsteem toetab
* Võimaldada kaupa elektrooniliselt registreerida
* Võimaldada määrata kaupade hetkeseisundit vastavalt elutsüklile
* Võimaldada muuta süsteemile teadaolevaid andmeid kaupade kohta
* Võimalik kaupade andmed kustutada e infosüsteemi mõttes unustada, kuid teha seda ainult siis, kui kaup pole veel kordagi aktiivsesse kasutusse läinud
* Võimaldada vastata fikseeritud päringutele kaupade kohta

### Allsüsteemi kasutavad pädevusalad

* Juhataja
* Kauba haldur
* Uudistaja
* Klient
* Kliendihaldur

### Allsüsteemi poolt vajatavad registrid

Allsüsteem teenindab kaupade registrit.

Allsüsteem loeb.

* Isikute register
* Töötajate register
* Klassifikaatorite register
* Klientide register

### Allsüsteemi ühe põhiprotsessi tegevusdiagramm

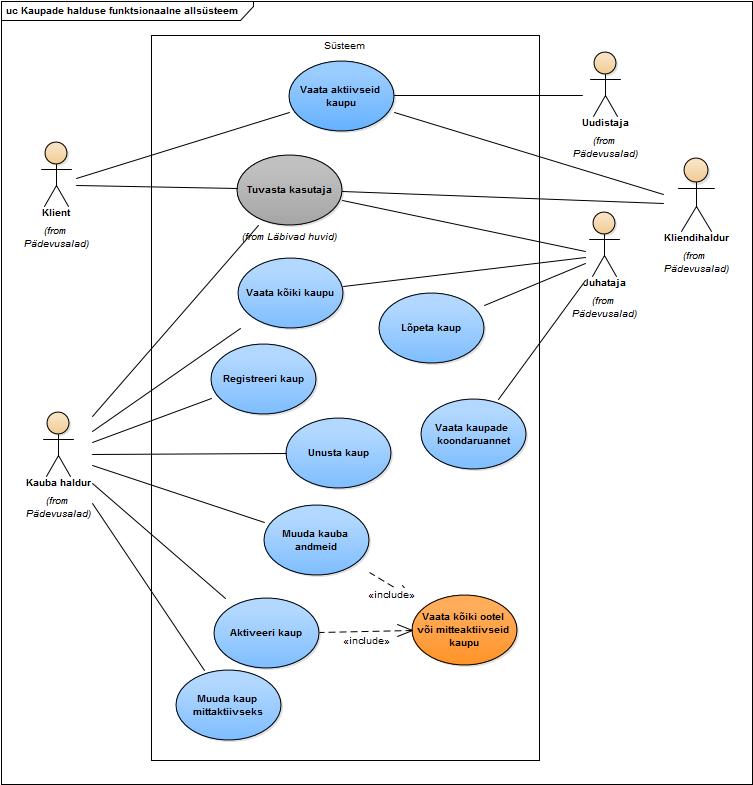
Joonis 1 esitab kauba lõpetamise protsessi kirjelduse tegevusdiagrammina. 

Joonis kaupade lõpetamise tegevusdiagramm.

### Allsüsteemi kasutusjuhtude eskiismudel

Joonis 2 esitatud kasutusjuhtude diagrammil on värvidel järgmine tähendus.

* Kollasega on tähistatud põhikasutusjuhud.
* **Oranžiga** on tähistatud abistavad kasutusjuhud (sisuliselt kasutusjuhu fragmendid), mis on kirja pandud selleks, et mitte kirjeldada mitmekordselt erinevates kasutusjuhtudes esinevat ühesugust funktsionaalsust.
* **Halliga** on tähistatudkasutusjuhud, mis esitavad läbivaid huvisid ning on seotud rohkem kui ühe funktsionaalse allsüsteemiga.



Joonis Kaupade funktsionaalse allsüsteemi kasutusjuhtude diagramm.

**Kasutusjuht**: Tuvasta kasutaja

**Tegutsejad**: Kauba haldur, Juhataja, Klient, Kliendihaldur – (edaspidi Subjekt)

**Kirjeldus**: Subjekt identifitseerib ennast. Selleks sisestab ta kasutajanime, parooli ja oma rolli süsteemis. Süsteem autendib subjekti, st kontrollib subjekti väidetavat identiteeti. Süsteemi sisenemiseks peab subjekt olema ka sobivas seisundis. Kui subjekt on autenditud (isik on tuvastatud ja identiteet kontrollitud), siis lubatakse subjekt süsteemi siseneda, vastasel juhul mitte. Lisaks autoriseeritakse subjekt, andes talle juurdepääsu infosüsteemi objektidele.

**Kasutusjuht**: Registreeri kaup

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur registreerib uue kauba.

**Kasutusjuht**: Unusta kaup

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur vaatab ootel kaupade nimekirja, valib sealt kauba ja kustutab selle andmebaasist. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Muuda kauba andmeid

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur vaatab ootel või mitteaktiivsete kaupade nimekirja, valib sealt kauba ja muudab selle andmeid. Ei ole võimalik muuta kauba registreerimise aega ja infot selle kohta, kes kauba registreeris. Samuti ei kuulu muudatuste hulka kauba seisundi muutmine (selleks on eraldi kasutusjuhud). Samas saab muuta kauba kategooriatesse kuuluvust.

**Kasutusjuht**: Aktiveeri kaup

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur vaatab ootel või mitteaktiivsete kaupade nimekirja, valib sealt kauba ja muudab selle aktiivseks.

**Kasutusjuht**: Muuda kaup mitteaktiivseks

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur vaatab aktiivsete kaupade nimekirja, valib sealt kauba ja muudab selle mitteaktiivseks. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Vaata kõiki ootel või mitteaktiivseid kaupu

**Tegutsejad**: Kauba haldur

**Kirjeldus**: Kauba haldur saab vaadata nimekirja ootel või mitteaktiivses seisundis olevatest kaupadest. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Vaata kõiki kaupu

**Tegutsejad**: Kauba haldur, Juhataja – (edaspidi Subjekt)

**Kirjeldus**: Subjekt saab vaadata kaupade nimekirja. Subjekt saab nimekirja sorteerida ja filtreerida. Samuti saab ta iga kauba korral vaadata selle detailseid andmeid.

**Kasutusjuht**: Lõpeta kaup

**Tegutsejad**: Juhataja

**Kirjeldus**: Juhataja vaatab aktiivsete või mitteaktiivsete kaupade nimekirja, valib sealt kauba ja lõpetab selle. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Vaata kaupade koondaruannet

**Tegutsejad**: Juhataja

**Kirjeldus**: Juhata näeb iga kauba seisundi kohta selle koodi, nimetust ja selles seisundis olevate kaupade arvu. Kui seisundiga pole seotud ühtegi kaupa, siis on see arv 0.

**Kasutusjuht**: Vaata aktiivseid kaupu

**Tegutsejad**: Kliendihaldur, Uudistaja, Klient – (edaspidi Subjekt)

**Kirjeldus**: Subjekt valib kategooria ja näeb kõigi sellesse kuuluvate aktiivses seisundis olevate kaupade kõiki andmeid, v.a registreerimise aeg ja registreerinud töötaja.

### Mittefunktsionaalsed nõuded

Tabel 4 esitab vaadeldava allsüsteemi mittefunktsionaalsed nõuded.

Tabel Allsüsteemi mittefunktsionaalsed nõuded.

| Tüüp | Nõude kirjeldus |
| --- | --- |
| andmebaasi-süsteem | Süsteem peab andmete hoidmiseks kasutama SQL‑andmebaasisüsteemi abil loodud andmebaasi. Tegemist on äritarkvaraga, mis kasutab tööks struktureeritud andmeid ning neid andmeid ei hakka olema väga palju (räägime maksimaalselt mõnest tuhandest reast). Kaupadega seotud transaktsioonilisi (tehingute) andmeid on rohkem (kümneid kuni sadu tuhandeid kirjeid), kuid ka nende haldamisega tulevad tänapäeva SQL süsteemid toime.  Seega puudub vajadus mõne NoSQL süsteemi kasutamise järele. Serverite operatsioonisüsteemiks peaks olema Linux, et vähendada süsteemi maksumust. Andmebaasisüsteemina on soovitav kasutada PostgreSQLi, kuna see on avatud lähtekoodiga, seda pakutakse tasuta, see jälgib küllaltki hästi SQL standardit, see pakub häid võimalusi andmebaasi programmeerijale ning sellele on suur kasutajate kogukond (st abi ja tuge pole keeruline leida). |
| arendusvahendid | Arendusvahendina tuleks kasutada organisatsioonile hangitud CASE tarkvara Rational Rose või Enterprise Architect. Alternatiivina võib kasutada tasuta pakutavat andmete modelleerimise vahendit DB MAIN  (<http://www.rever.eu/en/content/db-main-homepage>). Prototüübi koostamiseks kasutatakse töölaua andmebaasisüsteemi MS Access või LibreOffice Base, kuhu on integreeritud kasutajaliidese ehitamise vahendid. |
| keel | Süsteemi kasutajaliides ja dokumentatsioon peavad olema eesti keeles. Süsteem tuleks üles ehitada nii, et ei oleks väga raske lisada kasutajaliidesesse uusi keeli (inglise keel). |
| kasutajaliides | Töötavas süsteemis peab klientidele ja uudistajatele mõeldud kasutajaliides olema kindlasti veebipõhine. Töötajatele mõeldud rakendus võib olla kahekihiline, kus kasutaja arvutis on rakendus ning see suhtleb üle arvutivõrgu serveril paikneva andmebaasisüsteemiga. Soovi korral on võimalik selle jaoks MS Accessis või LibreOffice Base abil tehtud prototüüpi evolutsioneerida nii, et kasutatakse nendes loodud kasutajaliidest, kuid andmebaas on serveril.  Nõuded kasutajaliidese ülesehitusele.   * Ülesehituse põhimõtteid tuleb järjekindlalt järgida. * Rakenduses peab olema peavorm või pealehekülg, kust saab töökohaga seotud tegevuste juurde edasi liikuda. * Välisvõtme väärtuste registreerimiseks tuleb kasutada liitbokse või hüpikaknaid. * Kohustuslikud sisestusväljad tuleb tähistada (nt lisades lipikule \*). * Andmete lisamiseks ning andmete muutmiseks mõeldud väljad peavad erinevalt välja nägema (nt olema erineva taustavärviga). * Kuupäevad tuleb esitada formaadis DD.MM.YYYY * Kellaajad tuleb esitada formaadis HH24:MI * Ajatemplid tuleb esitada formaadis DD.MM.YYYY HH24:MI * Tegevused, mida süsteem saab ise teha (nt kindlaks tegema, kes andmed registreeris) peab tegema süsteem ilma kasutajalt tagasiside küsimisega tülitamata. * Kasutajaliideses ei tohi kuvada surrogaatvõtmete väärtuseid. * Kõikides olemite nimekirjades tuleb esitada selline hulk andmeid, et nende andmete alusel oleks võimalik olemeid üksteisest üheselt eristada ning et need andmed oleksid konkreetse kasutaja jaoks mõistetavad ja sisukad. * Mõõtmistulemusi või rahasummasid esitavate atribuutide väärtuste juures tuleb esitada ühik – rahasummade puhul valuuta tähis ning mõõtmistulemuste korral mõõtühik. |
| töökiirus | Päringu tegemisel ei tohi vastuse kuvamine võtta aega rohkem kui 5 sekundit. Andmete muudatuse salvestamine süsteemi poolt ei tohi võtta aega rohkem kui 5 sekundit. |
| töökindlus | Allsüsteemi tõrgeteta töö on hädavajalik organisatsiooni tõrgeteta töötamiseks. Tõrked tekitaksid suurt praktilist kahju ja ka moraalset kahju. Kuna allsüsteem haldab põhiandmeid, mis loovad konteksti transaktsioonlistele (tehingute) andmetele, siis põhjustaks allsüsteemi töö tõrge ka tõrkeid vastavate transaktsiooniliste andmete kogumisel ja töötlemisel.  Taasteaja siht (*recovery time objective*)("maksimaalne talutav süsteemi käideldamatuse kestus pärast intsidenti" (AKIT)): Juhul kui tekib veaolukord ja andmebaas või rakendus kahjustub, siis tuleb need taastada viimase tehtud varukoopia põhjal. Seda tuleb teha tunni jooksul peale rikke põhjuse kõrvaldamist ja serveri töökorda saamist.  Taasteseisu siht (*recovery point objective*)("intsidendijärgsele taastele seatud eesmärk ajahetkena, millele eelnevad andmed peavad olema täielikult taastatud (näiteks eelmine tund, eelmine tööpäev, eelmine nädal)"(AKIT)): Maksimaalselt võivad kaotsi minna viimase 24 tunni andmed, st et sellele eelnevad andmed peavad olema täielikult taastatud. |
| varukoopiad | Kuna hallatavad andmed on organisatsiooni jaoks väga olulised, siis tuleb vähemalt kord päevas teha andmetest varukoopia ja säilitada koopiaid mitmes erinevas asukohas. |
| turvalisus | Kui parooli hoitakse andmebaasis, siis ei tohi see olla avatekst, vaid peab olema parooli räsiväärtus, mis on leitud selle parooli jaoks genereeritud soola kasutades. Igal parooli jaoks tuleb genereerida uus sool. Räsiväärtuse leidmiseks ei tohi kasutada MD5 või SHA-1 räsifunktsioone, sest need on juba liiga ebaturvalised ja võimaldavad liiga lihtsalt algset parooli teada saada ning selle kaudu kasutaja identiteet varastada.  Kasutajanimed peavad olema tõstutundetud. Seega, näiteks:   * kui süsteemis on registreeritud kasutajanimi *Kasutaja1*, siis ei saa registreerida kasutajanime *kasutaja1,* * kui süsteemis on registreeritud kasutajanimi *Kasutaja1*, siis kasutaja tuvastamisel loetakse see samaväärseks sisestatud kasutajanimega *kasutaja1*.   Kaupade funktsionaalne allsüsteem teenindab kaupade registrit, mille turvaklass on (<https://www.riigiteataja.ee/akt/13125331?leiaKehtiv>):  K2T1S2  **K2** – töökindlus – 99% (lubatud summaarne seisak nädalas ~ 2 tundi); lubatav nõutava reaktsiooniaja kasv tippkoormusel – minutid (1÷10);  **T1** – info allikas, selle muutmise ja hävitamise fakt peavad olema tuvastatavad; info õigsuse, täielikkuse ja ajakohasuse kontroll erijuhtudel ja vastavalt vajadusele;  **S2** – salajane info: info kasutamine on lubatud ainult teatud kindlatele kasutajate gruppidele, juurdepääs teabele on lubatav juurdepääsu taotleva isiku õigustatud huvi korral; |

## Kaupade registri eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

Säilitada informatsiooni kaupade kohta sellises mahus, et oleks tagatud kaupade funktsionaalses allsüsteemis defineeritud eesmärkide täitmine.

### Registrit kasutavad pädevusalad

* Juhataja (ka omanik)
* Kauba haldur
* Klient
* Uudistaja
* Kliendihaldur

### Registrit teenindavad funktsionaalsed allsüsteemid

Kaupade registrit teenindab (loeb ja muudab) kaupade funktsionaalne allsüsteem.

### Infovajadused, mida register aitab rahuldada

* Ootel kaupade nimekiri, kus on vähemalt kauba kood ja nimetus.
* Aktiivsete kaupade nimekiri, kus on vähemalt kauba kood ja nimetus.
* Ootel või mitteaktiivsete kaupade nimekiri, kus on vähemalt kauba kood, nimetus ja seisundi nimetus.
* Aktiivsete või mitteaktiivsete kaupade nimekiri, kus on vähemalt kauba kood, nimetus ja seisundi nimetus.
* Kõikide kaupade nimekiri, kus on vähemalt kauba kood, nimetus ja seisundi nimetus.
* Kaupade detailandmed, kus seotud klassifikaatorite väärtuste koodide asemel on nimetused ning esitatakse info ka kauba registreerinud töötaja kohta (eesnimi, perenimi, e-meili aadress).
* Iga kauba seisundi kohta kõigi selles seisundis olevate kaupade arv.

### Seosed teiste registritega

**Töötajate register** – Töötajate registriga on kaubad seotud olemitüübi **Töötaja** kaudu. Töötaja registreerib kauba andmed ning süsteemis säilitatakse info selle kohta, milline töötaja need andmed registreeris.

**Klassifikaatorite register** – Klassifikaatorite registriga on kaup seotud olemitüübi kauba\_seisundi\_liik kaudu. Selle abil registreeritakse kauba hetkeseisund. Samuti on iga kaup seotud null või rohkema kauba kategooriaga, mis on samuti klassifikaator. Kaup on seotud ka klassifikaatorite registriga olemitüübi bränd kaudu.

Selleks, et saaks registreerida andmeid lepingute, laoliikumiste, tarnetellimuste, klienditellimuste, arvete, sooduspakkumiste, garantiijuhtumite ja tagastuste registrites, peavad olema registreeritud kauba andmed ja seega peab olema realiseeritud kaupade register.

### Ärireeglid

Jõustatavad kaupade registri põhjal

* Igal kaubal on unikaalne kood
* Igal kaubal on unikaalne nimetus
* Iga kaup on käesoleval ajahetkel täpselt ühes seisundis vastavalt oma elutsüklile.
* Iga kaup on seotud null või rohkema kategooriaga
* Iga kauba ja iga kategooria vahel saab olla maksimaalselt üks seos
* Iga kauba puhul on vaja registreerida töötaja, kes kauba andmed registreeris ning kauba registreerimise aeg
* kauba andmeid (sh kauba kategooriasse kuulumine) (v.a seisund) saab muuta vaid siis, kui see on ootel või mitteaktiivses seisundis.
* Kauba andmete muutmisel ei saa muuta seda registreerinud töötajat ja registreerimise aega
* Kauba andmeid saab andmebaasist kustutada vaid siis, kui see on ootel seisundis
* Kauba saab aktiveerida vaid siis, kui see on seotud vähemalt ühe kauba kategooriaga

Jõustatavad teiste registrite põhjal, kuid vajalikud kauba funktsionaalse allsüsteemi toimimiseks

* Iga isiku kasutajanimena kasutatakse tema unikaalset e-meili aadressi
* Iga isiku unikaalseks identifikaatoriks on kombinatsioon isikukoodist ja selle väljastanud riigi koodist
* Iga kliendi korral tuleb lähtuvalt isikuandmete kaitse seadusest registreerida, kas ta on nõus või mitte teda käsitlevate andmete töötlemisega tarbijaharjumuste uurimiseks või otseturustuseks ja andmete üleandmisega kolmandatele isikutele, kes soovivad neid kasutada tarbijaharjumuste uurimiseks või otseturustuseks. Kliendil on enda andmete selline töötlemine õigus igal ajal keelata.

### Registri kontseptuaalne eskiismudel

Joonis 3 esitab esimese versiooni kaupade registri kontseptuaalse andmemudeli olemi‑suhte diagrammist.



Joonis Kaupade registri kontseptuaalne eskiismudel.

# Detailanalüüs

Selles peatükis kirjeldatakse detailselt ja mittetehniliselt funktsionaalse allsüsteemi/registri paari, mille eskiismudelid esitati strateegilise analüüsi dokumendis.

## Kaupade funktsionaalse allsüsteemi detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt kaupade funktsionaalse allsüsteemi toimimist.

### Kasutusjuhtude mudel

Kaupade funktsionaalse allsüsteemi kasutusjuhtude diagramm (vt joonis 2).

**Punasega** viidatakse andmebaasioperatsioonidele, mis seisnevad ainult andmete lugemises. **Sinisega** viidatakse andmebaasioperatsioonidele, mis tegelevad andmebaasis andmete muutmisega.

**Kasutusjuht**: Tuvasta kasutaja

**Primaarne tegutseja**: Kauba haldur, Juhataja, Klient, Kliendihaldur – (edaspidi Subjekt).

**Osapooled ja nende huvid**:

* Kauba haldur, Juhataja, Klient, Kliendihaldur: Soovivad siseneda süsteemi ja teha tegevusi neile antud volituste piires.

**Käivitav sündmus**: Subjekt soovib süsteemi siseneda.

**Eeltingimused**: Subjekt on süsteemis kasutajaks registreeritud ning ta on sobivas rollis ja seisundis.

**Järeltingimused**: On tehtud kindlaks, kas subjektil on õigus süsteemi siseneda või mitte. Subjekt on autenditud ja talle on antud võimalus kasutada süsteemi talle antud volituste piires (subjekt on autoriseeritud).

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib siseneda süsteemi.
2. **Süsteem** palub subjektil ennast identifitseerida.
3. Subjekt identifitseerib ennast (sisestades kasutajanime, parooli).
4. **Süsteem** kontrollib, kas esitatud volitustõendiga (antud juhul parooliga) subjekti andmed on süsteemis olemas või mitte ning milline on tema roll ja seisund süsteemis **(OP1.1)**.
5. **Süsteem** annab subjektile volituse süsteemi kasutada ja annab talle juurdepääsu infosüsteemi objektidele.

*Subjekt võib üritada süsteemi siseneda kuni kolm korda.*

**Laiendused (või alternatiivne sündmuste käik)**:

5a. Kui süsteem ei leia esitatud volitustõendiga subjekti või pole subjekt sobivas rollis ja seisundis, siis ei saa subjekt õigust süsteemi kasutada.

* **Süsteem** kuvab subjektile teate, et sisselogimine ebaõnnestus. Selleks, et süsteemi toimimist võimalikule ründajale mitte reeta, ei ütle süsteem täpset põhjust.

**Kasutusjuht:** Registreeri nutitelefon

**Primaarne tegutseja**: Kauba haldur

**Osapooled ja nende huvid**:

* Kauba haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate nutitelefonide andmed ja et need andmed oleksid võimalikult täpsed.
* Juhataja: Soovib, et organisatsiooni kasum ja klientide rahulolu oleks võimalikult suur ja selleks peab juhatajal olema ülevaade kõigist nutitelefonidest ning uue kauba tekkimisel ei tohi selle registreerimisega viivitada.
* Klient, Uudistaja: Soovivad võimalikult täpset infot nutitelefonide kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga nutitelefone kasutava kliendi rollis.

**Käivitav sündmus**: Organisatsiooni jõuab teave uue nutitelefoni kohta, millega kliendid saavad hakata tulevikus tehinguid tegema.

**Eeltingimused**: Kauba haldur on autenditud ja autoriseeritud.

**Järeltingimused**: nutitelefon on registreeritud ja kaup on seisundis „Ootel“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Kauba haldur avaldab soovi uus nutitelefon registreerida.
2. **Süsteem** avab vormi, kus saab uue nutitelefoni registreerida. Seal on muuhulgas võimalik määrata, millistesse kategooriatesse kaup kuulub, sest süsteem pakub kategooriate valiku (**OP2.1**). Samuti on võimalik lisada erinevaid variante, sest süsteem pakub variantide valiku (**OP2.3**).
3. Kauba haldur sisestab nutitelefoni andmed ja valib kategooriad, millesse kaup kuulub. Samuti sisestab kauba haldur kauba variandid. Kauba haldur ei saa registreerida kauba algseisundit, registreerimise aega ning viidet registreerimise läbiviinud töötajale – seda teeb süsteem automaatselt. Ta annab korralduse salvestada.
4. **Süsteem** salvestab nutitelefoni andmed (**OP1**), ükshaaval kõikide kategooriasse kuulumiste andmed (**OP7**) ning kõik kauba variandid(**OP9**).

*Kauba haldur võib samme 1-4 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

2.1a Kui ühtegi kauba kategooriat pole registreeritud, siis kategooriate valikut ei pakuta ning kauba kategooriasse kuulumist ei saa registreerida.

2.2a Kui ühtegi värvi pole registreeritud, siis värvide valikut ei pakuta ning kauba variante ei saa registreerida.

3.1a Kauba haldur soovib kaupa mõnest määratud kategooriast kohe eemaldada.

3.1b **Süsteem** kuvab nimekirja kategooriatest, kuhu kaup juba kuulub. Iga kategooria juures on ka selle kategooria tüübi nimetus. (**OP2.2**)

3.1c **Süsteem** salvestab kategooriast eemaldamise (**OP8**).

3.2a Kauba haldur soovib kaubalt mõne määratud variandi kohe eemaldada.

3.2b **Süsteem** kuvab nimekirja kauba variantidest, mis kaubal juba on. Iga variandi juures on ka selle variandi värvi nimetus. (**OP2.4**)

3.2c **Süsteem** salvestab variandi eemaldamise (**OP10**).

**Kasutusjuht:** Unusta kaup

**Primaarne tegutseja**: kauba haldur

**Osapooled ja nende huvid**:

* kauba haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate kaupade andmed ja et need andmed oleksid võimalikult täpsed. Kui on selge, et kaupu sellisel kujul ei teki, siis soovib selle andmed segaduste vältimiseks süsteemist eemaldada.
* Juhataja: Soovib, et organisatsiooni kasum ja klientide rahulolu oleks võimalikult suur ja selleks peab juhatajal olema ülevaade kõigist kaupadest ning uue kauba tekkimisel ei tohi selle registreerimisega viivitada. Samas ei soovi ta näha kaupu, millest asja ei saa.
* Klient, Uudistaja: Soovivad võimalikult täpset infot kauba kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga kaupu kasutava kliendi rollis.

**Käivitav sündmus**: Organisatsiooni jõuab teave, et kaup sellisel kujul ei realiseeru ning seda ei saa hakata klientidele tehinguteks pakkuma.

**Eeltingimused**: Kauba haldur on autenditud ja autoriseeritud. Kaup on registreeritud ja on seisundis „Ootel“.

**Järeltingimused**: Kauba andmed on süsteemist kustutatud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Kauba haldur avaldab soovi kaup unustada, st selle andmed süsteemist kustutada.
2. **Süsteem** kuvab ootel kaupade nimekirja, kus on kood, nimetus, hind, kirjeldus, pildi aadress ja registreerimise aeg. (**OP3.1**)
3. Kauba haldur valib nimekirjast kauba ja annab korralduse see unustada.
4. **Süsteem** salvestab andmed (**OP2**).

*Kauba haldur võib samme 1-4 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

3a. Kauba haldur saab nimekirja kõigi kuvatud väljade järgi sorteerida ja filtreerida.

3b. Kui nimekirjas ei ole ühtegi ootel kaupa, siis ei saa kauba haldur jätkata.

**Kasutusjuht:** Muuda nutitelefoni andmeid

**Primaarne tegutseja**: kauba haldur

**Osapooled ja nende huvid**:

* kauba haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate nutitelefonide andmed ja et need andmed oleksid võimalikult täpsed.
* Juhataja: Soovib, et organisatsiooni kasum ja klientide rahulolu oleks võimalikult suur ja selleks peab juhatajal olema täpne ülevaade kõigist nutitelefonidest.
* Klient, Uudistaja: Soovivad võimalikult täpset infot nutitelefonide kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga nutitelefone kasutava kliendi rollis.

**Käivitav sündmus**: Ilmneb, et nutitelefoni andmete registreerimisel on tehtud viga või

nutitelefoni atribuutide väärtuste ja seoste hulgas on toimunud muudatus (siia hulka ei kuulu seisundimuudatus, millega tegelemiseks on eraldi kasutusjuhud).

**Eeltingimused**: Kauba haldur on autenditud ja autoriseeritud. Kaup on registreeritud ja on seisundis „Ootel“ või „Mitteaktiivne“.

**Järeltingimused**: Nutitelefoni andmed on muudetud, kuid kauba seisund ning info kauba registreerija ning registreerimise aja kohta ei ole muutunud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Kauba haldur soovib muuta nutitelefoni andmeid.
2. *Käivitub kasutusjuht „Vaata kõiki ootel või mitteaktiivseid kaupu“*
3. Kauba haldur valib nimekirjast kauba ja annab korralduse vaadata selle detailseid andmeid.
4. **Süsteem** kuvab muutmiseks mõeldud väljades nutitelefoni põhiandmed (kauba\_kood, nimetus, hind, kirjeldus, pildi aadress) (**OP4.1**) ning sellega seotud kategooriate, kategooriate tüüpide nimetused (**OP2.2**), variantide ja värvuste nimetused(**OP2.4**). Seal on muuhulgas võimalik määrata, millistesse kategooriatesse kaup kuulub, sest süsteem pakub kategooriate valiku (**OP2.1**), samuti millised on kauba variandid(**OP2.3**).
5. Kauba haldur muudab nutitelefoni andmeid ja annab korralduse salvestada.
6. **Süsteem** salvestab andmed (**OP6**).

*Kauba haldur võib samme 1-6 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

5.1a Kauba haldur võib lisada kauba uude kategooriasse ja anda korralduse salvestada.

* **Süsteem** salvestab andmed (**OP7**).

5.1b Kauba haldur võib eemaldada kauba kategooriast ja anda korralduse salvestada.

* **Süsteem** salvestab andmed (**OP8**).

5.1c Kui ühtegi kauba kategooriat pole registreeritud, siis kategooriate valikut ei pakuta ning kauba kategooriasse kuulumist ei saa registreerida.

5.2a Kauba haldur võib lisada kaubale uue variandi ja anda korralduse salvestada.

* **Süsteem** salvestab andmed (**OP9**).

5.2bKauba haldur võib eemaldada kaubalt variandi ja anda korralduse salvestada.

* **Süsteem** salvestab andmed (**OP10**).

5.2c Kui ühtegi kauba varianti pole registreeritud, siis variantide valikut ei pakuta ning kauba varianti ei saa registreerida.

**Kasutusjuht:** Aktiveeri kaup

**Primaarne tegutseja**: Kauba haldur

**Osapooled ja nende huvid**:

* Kauba haldur, Juhataja: Soovib, et iga kauba kohta oleks teada tema koht üldises kauba elutsüklis, mis ühtlasi määrab tegevused, mida selle kaubaga saab teha.
* Kauba haldur: Soovib, et kaupa saaks kasutada uutes tehingutes.
* Klient, Uudistaja: Soovivad näha kõiki aktiivseid kaupu, et otsustada, kas siduda ennast selle organisatsiooniga kaupasid kasutava kliendi rollis.

**Käivitav sündmus**: Kaupade ooteperiood või kauba seoses tekkinud ajutised probleemid on lahenenud ning kauba põhjal saab uuesti tehinguid teha.

**Eeltingimused**: Kauba haldur on autenditud ja autoriseeritud. Kaup on registreeritud ja on seisundis „Ootel“ või „Mitteaktiivne“. Kaup on määratud vähemalt ühte kauba kategooriasse.

**Järeltingimused**: Kaup on seisundis „Aktiivne“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Kauba haldur soovib aktiveerida kaupu.
2. *Käivitub kasutusjuht „Vaata kõiki ootel või mitteaktiivseid kaupu“*
3. Kauba haldur valib nimekirjast kauba ja annab korralduse see aktiivseks muuta.
4. **Süsteem** salvestab andmed (**OP3**).

*Kauba haldur võib samme 1-4 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

3a. Kui nimekirjas ei ole ühtegi ootel või mitteaktiivset kaupa, siis ei saa kauba haldur jätkata.

4a. Kui kaup ei kuulu ühtegi kauba kategooriasse, siis aktiveerimine ebaõnnestub.

**Kasutusjuht**: Muuda kaup mitteaktiivseks

**Primaarne tegutseja**: Kauba haldur

**Osapooled ja nende huvid**:

* Kauba haldur, Juhataja: Soovib, et iga kauba kohta oleks teada tema koht üldises kauba elutsüklis, mis ühtlasi määrab tegevused, mida selle kaubaga saab teha.
* Kauba haldur: Soovib kauba andmeid muuta või tegeleda sellega tekkinud ajutiste probleemidega, olles samal ajal veendunud, et keegi ei saa sellega algatada uusi tehinguid.
* Klient, Uudistaja: Soovivad näha kõiki aktiivseid kaupu, et otsustada, kas siduda ennast selle organisatsiooniga kaupu kasutava kliendi rollis (kui huvi pakkuv kaup ei ole selles nimekirjas, siis see on talle samuti oluline informatsioon).

**Käivitav sündmus**: kauba kasutamine tehingutes on vaja ajutiselt peatada kuna seoses selle kaubaga on ilmnenud ajutise iseloomuga probleemid

**Eeltingimused**: Kauba haldur on autenditud ja autoriseeritud. Kaup on registreeritud ja on seisundis „Aktiivne“.

**Järeltingimused**: Kaup on seisundis „Mitteaktiivne“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Kauba haldur avaldab soovi kaup mitteaktiivseks muuta.
2. **Süsteem** kuvab aktiivsete kaupade nimekirja, kus on kood, nimetus, hind, kirjeldus, pildi aadress ja registreerimise aeg (**OP6.1**)
3. Kauba haldur valib nimekirjast kauba ja annab korralduse see mitteaktiivseks muuta.
4. **Süsteem** salvestab andmed (**OP4**).

*Kauba haldur võib samme 1-4 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

3a. Kauba haldur saab nimekirja kõigi kuvatud väljade järgi sorteerida ja filtreerida.

3b. Kui nimekirjas ei ole ühtegi aktiivset kaupa, siis ei saa kauba haldur jätkata.

**Kasutusjuht**: Vaata kõiki ootel või mitteaktiivseid kaupu

**Primaarne tegutseja**: kauba haldur.

**Osapooled ja nende huvid**:

* Kauba haldur: Soovib sisendit juhtimisotsuste tegemiseks.

**Käivitav sündmus**: Subjekt soovib muuta kauba andmeid, sh kauba seisundit.

**Eeltingimused**: Subjekt on autenditud ja autoriseeritud.

**Järeltingimused**: On leitud seisundis „Ootel“ või „Mitteaktiivne“ olevate kaupade nimekiri.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata ootel või mitteaktiivsete kaupade nimekirja
2. **Süsteem** kuvab ootel või mitteaktiivses seisundis kaupade nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, hind, kirjeldus, pildi aadress ja registreerimise aeg (**OP7.1**)

**Laiendused (või alternatiivne sündmuste käik)**:

2a. Kauba haldur saab nimekirja kõigi kuvatud väljade järgi sorteerida ja filtreerida.

**Kasutusjuht**: Vaata kõiki kaupu

**Primaarne tegutseja**: Kauba haldur, Juhataja – (edaspidi Subjekt).

**Osapooled ja nende huvid**:

* Juhataja, kauba haldur: Soovib sisendit juhtimisotsuste tegemiseks.

**Käivitav sündmus**: Subjekt tahab mingil põhjusel vaadata kaupade detailseid andmeid (sealhulgas juba lõpetatud kauba andmeid). Näiteks soovib subjekt näha, milliseid kaupu on orfanisatsioon kunagi pakkunud või milliseid see praegu pakub.

**Eeltingimused**: Subjekt on autenditud ja autoriseeritud.

**Järeltingimused**: On leitud kõikide kaupade detailsed andmed.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata kõikide kaupade andmeid.
2. **Süsteem** kuvab kõigi kaupade nimekirja, kus on kood, nimetus, bränd, hetkeseisundi nimetus, hind, kirjeldus (**OP8.1**)
3. Subjekt valib kauba, mida ta soovib detailsemalt vaadata.
4. **Süsteem** kuvab vaatamiseks mõeldud väljades kauba põhiandmed (kauba\_kood, nimetus, hind, kirjeldus, pildi aadress, registreerimise aeg, registreerinud töötaja eesnimi, perenimi ja e-meili aadress) (**OP8.2**) kaubaga seotud nutitelefoni andmed, kaubaga seotud kategooriate ja kategooriate tüüpide nimetused (**OP2.2**) ning kauba variandid ja värvuste nimetused (**OP2.4**).

**Laiendused (või alternatiivne sündmuste käik)**:

3a. Subjekt saab nimekirja kõigi kuvatud väljade järgi sorteerida ja filtreerida.

3b. Kui nimekirjas ei ole ühtegi kaupa, siis ei saa subjekt jätkata.

**Kasutusjuht**: Lõpeta kaup

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

* Kauba haldur, Juhataja: Soovib, et iga kauba kohta oleks teada tema koht üldises kauba elutsüklis, mis ühtlasi määrab tegevused, mida selle kaubaga saab teha.
* Juhataja: Soovib anda kõigile huvitatud osapooltele teada, et kaubaga enam tehinguid ei tehta (kuid kõik käimasolevad tehingud tuleb vastavalt kehtivale korrale lõpetada). Samas soovib ta kauba andmete süsteemis säilimist, et ei läheks kaotsi info kauba ja sellega seotud tehingute kohta.
* Klient, Uudistaja: Soovivad näha kõiki aktiivseid kaupu, et otsustada, kas siduda ennast selle organisatsiooniga kaupasid kasutava kliendi rollis (kui huvi pakkuv kaup ei ole selles nimekirjas, siis see on talle samuti oluline informatsioon).

**Käivitav sündmus**: Kauba kasutamine tehingutes on vaja püsivalt lõpetada, kuna seoses kaubaga on ilmnenud püsiva iseloomuga probleemid või kuna kaup on oma aja lihtsalt ära elanud

**Eeltingimused**: Juhataja on autenditud ja autoriseeritud. Kaup on registreeritud ja on seisundis „Aktiivne“ või „Mitteaktiivne“.

**Järeltingimused**: Kauba seisund on muutunud „Lõpetatud“, kuid kauba andmed on süsteemis endiselt alles. Kauba andmeid ei tohi süsteemist füüsiliselt kustutada, sest sellega seoses tuleks kustutada info kõigi tehingute kohta, millega kaup on seotud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Juhataja avaldab soovi kaup lõpetada.
2. **Süsteem** kuvab aktiivsete või mitteaktiivsete kaupade nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, hind, kirjeldus (**OP9.1**)
3. Juhataja valib nimekirjast kauba ja annab korralduse see lõpetada.
4. **Süsteem** salvestab andmed (**OP5**).

*Juhataja võib samme 1-4 läbida nii mitu korda kui soovib.*

**Laiendused (või alternatiivne sündmuste käik)**:

3a. Juhataja saab nimekirja kõigi kuvatud väljade järgi sorteerida ja filtreerida.

3b. Kui nimekirjas ei ole ühtegi aktiivset või mitteaktiivset kaupa, siis ei saa juhataja jätkata.

**Kasutusjuht**: Vaata kauba koondaruannet

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

* Juhataja: Soovib sisendit juhtimisotsuste tegemiseks.
* Kauba haldur: Soovib, et juhataja teeks häid otsuseid ja äri kestaks.

**Käivitav sündmus**: Juhataja soovib juhtimisotsuste tegemiseks seada, kui palju on iga kauba elutsükli seisundi kohta kaupu, mis on parajasti selles seisundis.

**Eeltingimused**: Juhataja on autenditud ja autoriseeritud. Kauba seisundi liigid on registreeritud.

**Järeltingimused**: Kauba koondaruanne on moodustatud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Juhataja soovib vaadata kauba koondaruannet
2. **Süsteem** kuvab iga kauba elutsükli seisundi kohta selle seisundi koodi, nimetuse (suurtähtedega) ja hetkel selles seisundis olevate kaupade arvu. Kui selles seisundis pole hetkel ühtegi kaupa, siis on arv 0. Seisundid on sorteeritud kaupade arvu järgi kahanevalt. Kui mitmel seisundil on samasugune kaupade arv, siis need on sorteeritud suurtähtedega nime järgi tähestiku järjekorras. (**OP10.1**)

**Laiendused (või alternatiivne sündmuste käik)**:

2a. Kui ükski kauba seisundi liik pole registreeritud, siis ei saa olla ka registreeritud mitte ühtegi kaupa ja sellisel juhul tagastab päring null rida.

**Kasutusjuht**: Vaata aktiivseid kaupu

**Primaarne tegutseja**: Uudistaja, Klient – (edaspidi Subjekt).

**Osapooled ja nende huvid**:

* Kauba haldur, Juhataja: Tahavad, et võimalikel huvilistel oleks täpne ülevaade organisatsiooni pakutavast ja et see kallutaks neid organisatsiooni kliendiks hakkama
* Uudistaja, Klient: Soovivad näha organisatsiooni pakutavate kaupade nimekirja, et langetada tarbimisotsuseid.

**Käivitav sündmus**: Subjekt tunneb huvi organisatsiooni poolt hetkel pakutavate kaupade kohta, et otsustada, kas ennast tulevikus organisatsiooniga tihedamalt siduda.

**Eeltingimused**: Klient on autenditud ja autoriseeritud, uudistaja ei ole autenditud ja autoriseeritud.

**Järeltingimused**: Aktiivsete kaupade nimekiri on leitud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib näha kõiki organisatsiooni pakutavaid aktiivseid kaupu.
2. **Süsteem** kuvab nimekirja kategooriatest (**OP2.1**)
3. Subjekt valib konkreetse kategooria.
4. **Süsteem** kuvab sellesse kuuluvate aktiivsete kaupade nimekirja. Iga kauba kohta esitatakse kood, nimetus, kategooriate ja nende tüüpide nimetused, hind, kirjeldus (**OP11.2**)

**Laiendused (või alternatiivne sündmuste käik)**:

4a. Kui pole ühtegi aktiivset kaupa, siis on nimekiri tühi.

4b. Subjekt võib vaadatavate kaupade hulka koodi ja nimetuse järgi sorteerida ning filtreerida.

## Kaupade funktsionaalse allsüsteemi vajatavate registrite detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt kaupade funktsionaalse allsüsteemi vajatavate registrite struktuuri ja toimimist.

### Kontseptuaalne andmemudel

Järgnevalt esitatakse kontseptuaalne andmemudel, mis koosneb olemi‑suhte diagrammidest ja nendel olevate olemitüüpide ja atribuutide sõnalistest kirjeldustest.

Joonis 4esitatud olemi‑suhte diagrammidel on värvidel järgmine tähendus.

* **Punasega** on tähistatud *kaupade registri* põhiobjekt.
* Kollasega on tähistatud *kauba registrisse* kuuluvad mitte-põhiobjektid.
* **Rohelisega** on tähistatud teistesse registritesse kuuluvad objektid, mida on antud juhul vaja kaupade funktsionaalse allsüsteemi toimimise tagamiseks.



**Joonis 4 Kaupade registri olemi-suhte diagramm**



**Joonis 5 Isikute registri olemi-suhte diagramm**



**Joonis 6 Klassifikaatorite registri olemi-suhte diagramm**



**Joonis 7 Töötajate registri olemi-suhte diagramm**



**Joonis 8 Klientide registri olemi-suhte diagramm**

Tabel 5 esitab olemi-suhte diagrammidel esitatud olemitüüpide sõnalised kirjeldused.

Tabel Olemitüüpide sõnalised kirjeldused.

| **Olemitüübi nimi**  **(teised nimed)** | **Kuuluvus registrisse** | **Definitsioon** |
| --- | --- | --- |
| Isik | Isikute register | Mistahes organisatsiooniga seotud füüsiline isik (eraisik). Isik võib olla seotud organisatsiooniga näiteks kui klient või kui töötaja. |
| Töötaja | Töötajate register | Organisatsioonis (kui tööandja juures) töölepingu alusel töötav ja selle organisatsiooni juhtimisele ning kontrollile alluv isik, kes saab oma töö eest töölepingus kokkulepitud tasu. |
| Klassifikaator | Klassifikaatorite register | Klassifikaatorid on "mistahes andmed, mida kasutatakse andmebaasis teiste andmete liigitamiseks või andmebaasis olevate andmete seostamiseks väljaspool organisatsiooni vastutusala oleva informatsiooniga." (Chisholm, 2000) |
| Isiku\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga isiku puhul tema hetkeseisundi vastavalt üldisele isikute elutsüklile. |
| Töötaja\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga töötaja puhul tema hetkeseisundi vastavalt üldisele töötajate elutsüklile. |
| Amet | Klassifikaatorite register | Amet on töölepingus sätestatud ametikohustuste üldnimetus. Ametid on klassifikaatorid. |
| Riik | Klassifikaatorite register | "Riik on kindla territooriumiga sõltumatu (suveräänne) üksus (juriidiline lähenemine).“ (Vikipeedia) Riikidena käsitletakse riike ja territooriumeid, mis on kirjeldatud Eesti Statistika lehel olevas riikide ja territooriumite klassifikaatori dokumendis, mis on omakorda eestindatud versioon rahvusvahelisest standardist "International Standard Codes for the Representation of the Names of Countries (ISO 3166) |
| Kauba\_kategooria | Klassifikaatorite register | Võimaldab kauba klassifitseerimist erinevatesse kategooriatesse ja selle alusel kauba rühmitamist teatud põhjusel huvipakkuvateks hulkadeks. Tegemist on üksteist mittevälistavate kategooriatega, st üks ja sama kaup võib kuuluda korraga mitmesse sama tüüpi kategooriasse. |
| Kauba\_kategooria\_tyyp | Klassifikaatorite register | Võimaldab rühimatada kauba klassifitseerimiseks kasutatavaid kategooriaid ühise nime alla. Need nimed kirjeldavad, mis liiki klassifikatsiooniga on tegemist. |
| Kaup | Kaupade register | Nutitelefonid on nutitelefonide e-poe jaoks ostu-müügi objektid ehk kaubad. Kauba all mõeldakse kaubaartikkleid. Nutitelefonide e-pood ostab nutitelefone partneritelt ning müüb neid peale maksude ja juurdehindluse ostuhinnale juurdearvestamist klientidele. |
| Kauba\_kategooria\_  omamine | Kaupade register | Näitab kauba kuulumist kategooriatesse. Iga kauba ja iga kauba kategooria vahel võib olla maksimaalselt üks seos. |
| Nutitelefon | Kaupade register | Seade, milles on ühendatud pihuarvuti ja mobiiltelefoni funktsioonid. |
| Brand | Klassifikaatorite register | Toote omaduste kombinatsioon, mis annab tootele identiteedi ja eristab seda teiste toodete hulgast. |
| Kauba\_seisundi\_liik | Klassifikaatorite register | Kauba hetkeolukorra iseloomustus. Näiteks: aktiivne, mitteaktiivne. |
| Ekraan\_resolutsioon | Klassifikaatorite register | Ekraani eraldusvõime, mis näitab ekraani teravust. Antud suurus on esitatud pikslites kujul laius korda kõrgus. Näiteks: 750 x 1334 |
| Protsessor | Klassifikaatorite register | Nutitelefoni keskne osa, mis mõistab käsklusi ja annab neid. Esitatud on protsessori tüüp lähtuvalt tuumade arvu. Näiteks: Quad-Core |
| Diagonaal | Klassifikaatorite register | Nutitelefoni ekraani suurus, mis ühendab kahte mitte ühel küljel asuvat tippu ning antud suurus on esitatud tollides. Näiteks: 5,2 |
| Sisemalu | Klassifikaatorite register | Nutitelefoni mäluseade, kuhu salvestatakse andmeid. Selle suurus on esitatud gigabaitides. |
| Kaamera | Klassifikaatorite register | Nutitelefoni tagaküljel või esiküljel asuva ülesvõtteaparaadi efektiivsuse väljendus megapikslites. Näiteks: 12 |
| Kauba\_variant | Kaupade register | Kauba varianti kasutatakse sama kauba erinevate versioonide eristamiseks. Näiteks: eri värvi versioon samast kaubast. |
| Varv | Klassifikaatorite register | Silmaga tajutav kauba omadus, mille määrab sellelt kehalt hajunud või seda keha läbinud valgus. Väärtus on esitatud sõnaliselt. |
| Klient | Klientide register | Nutitelefonide e-poe poolt pakutavaid müügiteenuseid kasutav eraisik. |
| Kliendi\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga kliendi puhul tema hetkeseisundi vastavalt üldisele klientide elutsüklile. |

Tabel 6 esitab atribuutide sõnalised kirjeldused.

Tabel Atribuutide sõnalised kirjeldused.

| **Olemitüübi nimi** | **Atribuudi nimi (teised nimed)** | **Atribuudi definitsioon** | **Näiteväärtus** |
| --- | --- | --- | --- |
| Klassifikaator | kood | Klassifikaatori väärtust esitav kood, mida saab kasutada selle väärtuse lühidalt esitamiseks. Kood võib olla tekstiline või numbriline väärtus. Kood peaks olema võimalikult hästi meeldejääv. See tähendab, et kui kasutaja näeb koodi, siis seostub see tema jaoks võimalikult lihtsalt koodiga iseloomustatava klassifikaatori väärtusega.  **{Klassifikaatori unikaalne identifikaator, mis on unikaalne klassifikaatori tüübi piires. Registreerimine on kohustuslik.**  **Riikide koodid koosnevad vastavalt ISO 3166 standardile täpselt kolmest suurtähest.**  **Diagonaali kood peab olema suurem kui 0.**  **Kaamera kood peab olema suurem kui 0.**  **Sisemälu kood peab olema suurem kui 0.}** | EST |
| Klassifikaator | nimetus | Klassifikaatori väärtuse ametlik nimetus. Riikide nimetused leitakse Eesti Statistika kodulehelt alajaotusest Riikide ja territooriumide klassifikaator 2013v1.  **{Klassifikaatori unikaalne identifikaator, mis on unikaalne klassifikaatori tüübi piires. Registreerimine on kohustuslik. Nimetus ei tohi olla tühi string või**  **ainult tühikutest koosnev string.}** | Aktiivne |
| Amet | kirjeldus | Ametist tulenevate õiguste ja kohustuste vabatekstiline kirjeldus.  **{Kirjeldus ei tohi olla tühi string või ainult tühikutest koosnev string.}** | Juhib organisatsiooni igapäevast tööd ning langetab strateegilisi otsuseid |
| Kaup | kaup\_kood | Kauba arvuline kood, mis sisestatakse inimkasutaja poolt, mitte ei genereerita süsteemi poolt.  **{Kauba unikaalne identifikaator. Registreerimine on kohustuslik}** | 222 |
| Kaup | nimetus | Kauba tekstiline nimetus.  **{Kauba unikaalne identifikaator. Registreerimine on kohustuslik. Nimetus ei tohi olla tühi string ega ainult tühikutest koosnev string.}** | Nimetus |
| Kaup | reg\_aeg | Kauba registreerimise aeg kuupäeva ja kellaaja täpsusega. Selle võib süsteem ise automaatselt määrata.  **{Registreerimine on kohustuslik. Väärtus peab olema vahemikus 01. jaanuar 2010 00:00 ja 31. detsember 2100 kell 23:59 (otspunktid kaasa arvatud)}** | 22.03.2015 12:33 |
| Kaup | hind | Kauba väärtuse rahaline väljendus.  Hind on esitatud eurodes ning koosneb sisseostuhinna, maksude ja juurdehindluse summast. Tegu on väljamüügihinnaga.  **{Registreerimine on kohustuslik.**  **Väärtus peab olema suurem kui 0.}** | 548.5 |
| Kaup | kirjeldus | Kauba olulistemate tunnuste kirjalik ülevaatlik esitlus.  **{Registreerimine on kohustuslik. Kirjeldus ei tohi olla tühi string ega ainult tühikutest koosnev string.}** | Suurepärane kaamera ning kõige võimsam olemas olev protsessor. |
| Kaup | pildi\_aadress | Pildi füüsiline aadress serveris.  **{Väärtus peab sisaldama punkti ja faililaiendit („.jpg“ „.gif“ )}** | src/images/iphone7.jpg |
| Isik | isikukood | Riigi poolt väljastatud isiku identifikaator, mis on unikaalne selle väljastanud riigi piires.  **{Registreerimine on kohustuslik. Koos riigi identifikaatoriga on isiku unikaalne identifikaator.**  **Isikukood ei tohi olla tühi string või ainult tühikutest koosnev string}** | 39204010231 |
| Isik | eesnimi | "Lapsele pärast sündi (registreerimisel) pandav nimi, osa isikunimest. Eesnimi asetseb harilikult perekonnanime ees, harva järel (nt Ungari pruugis)." (ESTERM)  **{Vähemalt üks kahest – eesnimi või perenimi peab olema registreeritud. Eesnimi ei tohi olla tühi string või ainult tühikutest koosnev string.}** | Mart |
| Isik | perenimi (perekonna- nimi) | "Nimi, mis on isikul ühine teiste tema perekonna liikmetega" (ESTERM)  **{Vähemalt üks kahest – eesnimi või perenimi peab olema registreeritud. Perenimi ei tohi olla tühi string või ainult tühikutest koosnev string.}** | Mets |
| Isik | synni\_kp | Isiku sünni kuupäev sünnikoha kohaliku aja järgi.  **{Registreerimine on kohustuslik. Sünni kuupäeva võimalikud väärtused on vahemikus 01. jaanuar 1900 ja 31. detsember 2100 (otspunktid kaasa arvatud). Sünni kuupäev ei tohi olla suurem isiku registreerimise ajast}** | 12.08.1993 |
| Isik | elukoht | Isiku alalise elukoha aadress.  "Koha-aadress on territooriumi haldusjaotuse hierarhiast ja ametlikest kohanimedest lähtuv aadressobjekti tekstilis-numbriline kirje või tunnus. Ühele objektile võib määrata mitu koha-aadressi. Ühele objektile määratud koha-aadressid on paralleelaadressid." ("Aadressandmete süsteemi kehtestamine")  Näide: Tallinn, 34124, Ehitajate tee 62-12. Harjumaa, Viimsi vald, Kaku küla, Laane talu.  **{Elukoht ei tohi olla tühi string, ainult tühikutest koosnev string või ainult numbritest koosnev string.}** | Tallinn, Pikk tn. 12 |
| Isik | e\_meil (e\_mail, meil, meiliaadress) | Aadress, millele saab üle võrgu (ühest arvutist või tööjaamast teise) saata isikule mõeldud kirjalikke sõnumeid. Kasutatakse kasutaja tuvastamisel kasutajanimena.  **{Registreerimine on kohustuslik. Isiku tõstutundetud unikaalne identifikaator. Teiste sõnadega, kui süsteemis on näiteks meiliaadress** [**Mati@mets.ee**](mailto:Mati@mets.ee)**, siis meiliaadressi** [**mati@mets.ee**](mailto:mati@mets.ee) **lisada ei saa.**  **e\_meil peab sisaldama "@" märki. Võib olla kuni 254 märki pikk.}** | kalamees@hot.ee |
| Isik | parool | Isiku identsust tõendav teadmuslik (miski, mida isik teab) volitustõend. Andmebaasis salvestatakse parooli ja soola põhjal leitud räsiväärtus.  **{Registreerimine on kohustuslik.}** | $2a$11$FsKdoFDJePwuYtyg2hBxz.e8AwSODaO/nFGGacEm05vIgOBNG9dHC |
| Isik | reg\_aeg | Isiku registreerimise aeg kuupäeva ja kellaaja täpsusega. Selle võib süsteem ise automaatselt määrata.  **{Registreerimine on kohustuslik. Väärtus peab olema vahemikus 01. jaanuar 2010 00:00 ja 31. detsember 2100 kell 23:59 (otspunktid kaasa arvatud)}** | 12.08.2014 17:01 |
| Nutitelefon | on\_sormejäljelugeja | Määrab, kas nutitelefonil on seade, mis loeb sõrmejälgi.  **{Väärtus peab olema boolean tüüpi vaikimisi false}** | true |
| Nutitelefon | on\_veekindel | Määrab, kas nutitelefon on vastupidav niisketes tingimustes või vee all.  **{Väärtus peab olema boolean tüüpi vaikimisi false}** | false |
| Klient | on\_nous\_tylitamisega | Määrab, kas klient on nõus, et nutitelefonide e-pood talle reklaami edastab.  **{Väärtus peab olema boolean tüüpi vaikimisi false}** | false |

### Andmebaasioperatsioonide lepingud

**OP1 Registreeri Nutitelefon(p\_kauba\_kood, p\_nimetus, töötaja identifikaator, brändi identifikaator, nutitelefon, kaamera identifikaator (eesmine kaamera), kaamera identifikaator (tagumine kaamera), sisemälu identifikaator, diagonaali identifikaator, protsessori identifikaator, ekraani resolutsiooni identifikaator, p\_hind, p\_kirjeldus, p\_reg\_aeg, p\_pildi\_aadress, p\_on\_veekindel, p\_on\_sõrmejäljelugeja)**

**Eeltingimused**:

* kauba\_seisundi\_liik eksemplar xsl (nimetus="Ootel") on registreeritud
* Töötaja eksemplar t (töötaja identifikaator) on registreeritud
* Brändi eksemplar b (brändi identifikaator) on registreeritud
* Kaamera eksemplar k1 (kaamera identifikaator (eesmine kaamera)) on registreeritud
* Kaamera eksemplar k2 (kaamera identifikaator (tagumine kaamera)) on registreeritud
* Sisemälu eksemplar s (sisemälu identifikaator) on registreeritud
* Diagonaali eksemplar d (diagonaali identifikaator) on registreeritud
* Protsessori eksemplar p (protsessori identifikaator) on registreeritud
* Ekraani eksemplar e (ekraani identifikaator) on registreeritud

**Järeltingimused**:

--Loo eksemplare

* Kauba eksemplar x on registreeritud
* Nutitelefoni eksemplar y on registreeritud

--Väärtusta atribuute

* x.Kauba\_kood:= p\_kauba\_kood
* x.nimetus:= p\_nimetus
* x.reg\_aeg:= hetke kuupäev + kellaaeg
* x.hind:= p\_hind
* x.kirjeldus:= p\_kirjeldus
* x.pildi\_aadress:=p\_pildi\_aadress
* y.on\_veekindel:=p\_on\_veekindel
* y.on\_sõrmejäljelugeja:=p\_on\_sõrmejäljelugeja

--Loo seoseid

* x ja xsl seos on registreeritud
* x ja t seos on registreeritud
* x ja b seos on registreeritud
* x ja y seos on registreeritud
* y ja k1 seos on registreeritud
* y ja k2 seos on registreeritud
* y ja s seos on registreeritud
* y ja d seos on registreeritud
* y ja p seos on registreeritud
* y ja e seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri nutitelefon

**OP2 Unusta kaup(p\_kauba\_kood)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

1. x ja kõik selle seosed on andmebaasist kustutatud

**Kasutus kasutusjuhtude poolt**: Unusta kaup

**OP3 Aktiveeri kaup (p\_kauba\_kood)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl\_vana (nimetus="Ootel") või (nimetus="Mitteaktiivne")
3. Kauba\_seisundi\_liik eksemplar xsl\_uus (nimetus="Aktiivne") on registreeritud
4. Leidub vähemalt üks kauba\_kategooria\_omamine eksemplar xko, mis on seotud x

**Järeltingimused**:

--Kustuta seoseid

1. x ja xsl\_vana seos on kustutatud

--Loo seoseid

1. x ja xsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Aktiveeri kaup

**OP4 Muuda kaup mitteaktiivseks (p\_kauba\_kood)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl\_vana (nimetus=“Aktiivne“)
3. Kauba\_seisundi\_liik eksemplar xsl\_uus (nimetus=“Mitteaktiivne“) on registreeritud

**Järeltingimused**:

--Kustuta seoseid

1. x ja xsl\_vana seos on kustutatud

--Loo seoseid

1. x ja xsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda kaup mitteaktiivseks

**OP5 Lõpeta kaup(p\_kauba\_kood)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl\_vana (nimetus="Aktiivne") või (nimetus="Mitteaktiivne")
3. kauba\_seisundi\_liik eksemplar xsl\_uus (nimetus="Lõpetatud") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

1. x ja xsl\_vana seos on kustutatud

--Loo seoseid

1. x ja xsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Lõpeta kaup

**OP6 Muuda nutitelefoni andmeid (p\_kauba\_kood\_vana, p\_kauba\_kood\_uus, p\_nimetus, brändi identifikaator, nutitelefon, kaamera identifikaator (eesmine kaamera), kaamera identifikaator (tagumine kaamera), sisemälu identifikaator, diagonaali identifikaator, protsessori identifikaator, ekraani resolutsiooni identifikaator, p\_hind, p\_kirjeldus, p\_pildi\_aadress, p\_on\_veekindel, p\_on\_sõrmejäljelugeja)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood\_vana) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel") või (nimetus="Mitteaktiivne")
3. x on seotud brändi eksemplariga xb
4. x on seotud nutitelefoni eksemplariga xn
5. Nutitelefoni eksemplar y on registreeritud
6. y on seotud kaamera (eesmine kaamera) eksemplariga yk1
7. y on seotud kaamera (tagumine kaamera) eksemplariga yk2
8. y on seotud sisemälu eksemplariga ys
9. y on seotud diagonaali eksemplariga yd
10. y on seotud protsessori eksemplariga yp
11. y on seotud ekraani resolutsiooni eksemplariga ye
12. Brändi eksemplar b (brändi identifikaator) on registreeritud
13. Kaamera eksemplar k1 (kaamera identifikaator (eesmine kaamera)) on registreeritud
14. Kaamera eksemplar k2 (kaamera identifikaator (tagumine kaamera)) on registreeritud
15. Sisemälu eksemplar s (sisemälu identifikaator) on registreeritud
16. Diagonaali eksemplar d (diagonaali identifikaator) on registreeritud
17. Protsessori eksemplar p (protsessori identifikaator) on registreeritud
18. Ekraani eksemplar e (ekraani identifikaator) on registreeritud

**Järeltingimused**:

--Väärtusta atribuute

1. x.kauba\_kood:= p\_kauba\_kood\_uus
2. x.nimetus:= p\_nimetus
3. x.hind:= p\_hind
4. x.kirjeldus:= p\_kirjeldus
5. x.pildi\_aadress:=p\_pildi\_aadress
6. y.on\_veekindel:=p\_on\_veekindel
7. y.on\_sõrmejäljelugeja:=p\_on\_sõrmejäljelugeja

--Kustuta seoseid

1. x ja xb seos on kustutatud
2. x ja xn seos on kustutatud
3. y ja yk1 seos on kustutatud
4. y ja yk2 seos on kustutatud
5. y ja ys seos on kustutatud
6. y ja yd seos on kustutatud
7. y ja yp seos on kustutatud
8. y ja ye seos on kustutatud

--Loo seoseid

* x ja b seos on registreeritud
* y ja k1 seos on registreeritud
* y ja k2 seos on registreeritud
* y ja s seos on registreeritud
* y ja d seos on registreeritud
* y ja p seos on registreeritud
* y ja e seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda nutitelefoni andmeid

**OP7 Lisa kaup kategooriasse (p\_kauba\_kood, kauba kategooria identifikaator)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. kauba\_kategooria eksemplar xk (kauba kategooria identifikaator) on registreeritud
3. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel") või (nimetus="Mitteaktiivne")

**Järeltingimused**:

--Loo eksemplare

1. kauba\_kategooria\_omamine eksemplar xko on registreeritud

--Loo seoseid

1. x ja xko seos on registreeritud
2. xk ja xko seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri nutitelefon, Muuda nutitelefoni andmeid

**OP8 Eemalda kaup kategooriast (p\_kauba\_kood, kauba kategooria identifikaator)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. kauba\_kategooria eksemplar xk (kauba kategooria identifikaator) on registreeritud
3. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel") või (nimetus="Mitteaktiivne")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

1. kauba\_kategooria\_omamine eksemplar xko, mis on seotud x-ga ja mis on seotud xk-ga, on koos oma seostega kustutatud

**Kasutus kasutusjuhtude poolt**: Registreeri nutitelefon, Muuda nutitelefoni andmeid

**OP9 Lisa kauba variant(p\_kauba\_kood, kauba värvuse identifikaator)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel") või (nimetus="Mitteaktiivne")
3. värvuse eksemplar v (kauba variandi värvuse identifikaator) on registreeritud

**Järeltingimused**:

--Loo eksemplare

1. kauba\_variant eksemplar xv on registreeritud

--Loo seoseid

1. x ja xv seos on registreeritud
2. xv ja v seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri nutitelefon, Muuda nutitelefoni andmeid

**OP10 Eemalda kauba variant (p\_kauba\_kood, kauba värvuse identifikaator)**

**Eeltingimused**:

1. Kauba eksemplar x (kauba\_kood=p\_kauba\_kood) on registreeritud
2. värvuse eksemplar v (kauba variandi värvuse identifikaator) on registreeritud
3. x on seotud kauba\_seisundi\_liik eksemplariga xsl (nimetus="Ootel") või (nimetus="Mitteaktiivne")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

1. kauba\_variandi eksemplar xv, mis on seotud x-ga ja mis on seotud v-ga, on koos oma seostega kustutatud

**Kasutus kasutusjuhtude poolt**: Registreeri nutitelefon, Muuda nutitelefoni andmeid

### Registri põhiobjekti seisundidiagramm

Joonis 5 esitab seisundidiagrammi, mis kirjeldab registri põhiobjekti kaup kõikvõimalikke elutsükleid*.*



Joonis 9 kauba seisundidiagramm

## CRUD maatriks

Tabel 7 olev CRUD maatriks esitatakse *olemitüüpide* ja *kasutusjuhtude* täpsusega. Maatriksi veergudele vastavad kasutusjuhud ja ridadele olemitüübid.

Oranžil taustal on esitatud olemitüübid, mis kuuluvad vaadeldava allsüsteemi teenindatavasse registrisse.

Tabel CRUD maatriks.

| Kasutusjuhud  Olemitüübid | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Kokku |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Klassifikaator | R | R | R | R | R | R | R | R | R | R | R | R |
| Riik |  |  |  |  |  |  |  |  |  |  |  | R |
| Amet | R | R |  |  |  |  |  |  |  |  |  | R |
| Isiku\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Töötaja\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Kauba\_kategooria |  | R | R | R |  |  |  | R |  |  | R | R |
| Kauba\_kategooria\_tüüp |  | R | R | R |  |  |  | R |  |  | R | R |
| Kauba\_seisundi\_liik |  | R | R | R | R | R | R | R | R | R | R | R |
| Isik | R |  |  |  |  |  |  | R |  |  |  | R |
| Töötaja | R | R |  |  |  |  |  | R |  |  |  | R |
| Kauba\_kategooria\_omamine |  | CRD | CD | CRD |  |  |  |  |  |  | R | CRD |
| Kaup |  | C | RD | RU | RU | RU | R | R | RU | R | R | CRU |
| Kauba\_variant |  | CR | D | CRUD | R | R | R | R | R |  | R | CRUD |
| Nutitelefon |  | CR | RD | RU | R | R | R | R | R |  | R | CRU |
| Klient | R |  |  |  |  |  |  |  |  |  |  | R |
| Kliendi\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Bränd |  | R |  | R |  |  | R | R |  |  | R | R |
| Diagonaal |  | R |  | R |  |  | R | R |  |  | R | R |
| Kaamera |  | R |  | R |  |  | R | R |  |  | R | R |
| Ekraan\_resolutsioon |  | R |  | R |  |  | R | R |  |  | R | R |
| Protsessor |  | R |  | R |  |  | R | R |  |  | R | R |
| Sisemälu |  | R |  | R |  |  | R | R |  |  | R | R |
| Värv |  | R |  | R |  |  | R | R |  |  | R | R |

1 – Tuvasta kasutaja

2 – Registreeri nutitelefon

3 – Unusta kaup

4 – Muuda nutitelefoni andmeid

5 – Aktiveeri kaup

6 – Muuda kaup mitteaktiivseks

7 – Vaata kõiki ootel või mitteaktiivseid kaupasid

8 – Vaata kõiki kaupasid

9 – Lõpeta kaup

10 – Vaata kauba koondaruannet

11 – Vaata aktiivseid kaupasid

# Füüsiline disain

Selles peatükis esitatakse mudel, mis kirjeldab kaupade funktsionaalse allsüsteemi toimimiseks vajalike registrite tehnilist lahendust nutitelefonide müügiga tegeleva e-poe andmebaasisüsteemis.

## Kaupade funktsionaalse allsüsteemi vajatavate registrite füüsiline disain



**Joonis 10 Kaupade registri füüsilise disaini andmebaasi diagramm.**



**Joonis 11 Kaupade registris oleva nutitelefoni füüsilise disaini andmebaasi diagramm.**



**Joonis 12 Klassifikaatorite registri füüsilise disaini andmebaasi diagramm.**



**Joonis 13 Klassifikaatorite registris olevate seisundite füüsilise disaini andmebaasi diagramm.**



**Joonis 14 Isikute registri füüsilise disaini andmebaasi diagramm.**



**Joonis 15 Klientide registri füüsilise disaini andmebaasi diagramm.**



**Joonis 16 Töötajate registri füüsilise disaini andmebaasi diagramm.**

# Realisatsioon PostgreSQLis

Järgnevas peatükis on esitatud andmebaasi andmebaasisüsteemis PostgreSQL (10.1) realiseerimiseks mõeldud laused.

## Andmebaasi loomine

Andmebaasis t155605 vastab märkide klassifikatsioon ja võrdlusreeglistik eesti keele reeglitele. Alljärgnevalt on toodud käsk, mis tuleb käivitada shelli promtis. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

Createdb -l et\_EE.utf8 -T template0 t155605

„Kui andmebaasiga töötamiseks kasutatakse psqli läbi SSH Secure Shell Clienti, siis järgnev käsk tuleb käivitada iga sessiooni algul. Muidu annavad õ, ä, ö, ü, Õ, Ä, Ö, Ü tähed käskudes veateateid ning pole päringu tulemustes loetavad.„ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

SET client\_encoding=LATIN9;

Järgnevalt on esitatud andmebaasiobjektide loomiseks mõeldud SQL laused.

## Skeemid

Antu projektis luuakse skeemiobjektid skeemis nimega public. Antud skeem on loodud automaatselt andmebaasi loomisel. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

## Domeenid

CREATE DOMAIN d\_nimetus AS VARCHAR(100) NOT NULL

CONSTRAINT chk\_Nimetus\_ei\_koosne\_tyhikutest CHECK (VALUE!~'^[[:space:]]+$')

CONSTRAINT chk\_Nimetus\_pole\_tyhi\_string CHECK (VALUE!='');

CREATE DOMAIN d\_kirjeldus AS TEXT

CONSTRAINT chk\_Kirjeldus\_ei\_koosne\_tyhikutest CHECK (VALUE!~'^[[:space:]]+$')

CONSTRAINT chk\_Kirjeldus\_pole\_tyhi\_string CHECK (VALUE!='');

CREATE DOMAIN d\_reg\_aeg AS timestamp NOT NULL DEFAULT LOCALTIMESTAMP(0)

CONSTRAINT chk\_Reg\_aeg\_on\_vahemikus\_01\_01\_2010\_ja\_31\_12\_2100 CHECK (VALUE BETWEEN '2010-01-01' AND '2100-12-31 23:59:59');

## Tabelid ja arvujada generaatorid

Arvujada generaatorid luuakse andmebaasisüsteemi poolt automaatselt. Selleks tuleb kasutada tabeli loomise lauses SERIAL notatsiooni, mille tulemusena seotakse arvujada generaator vastava veeruga mille puhul SERIAL notatsiooni kasutati. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

CREATE TABLE Kauba\_kategooria\_tyyp(

kauba\_kategooria\_tyyp\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Kauba\_kategooria\_tyyp PRIMARY KEY(kauba\_kategooria\_tyyp\_kood),

CONSTRAINT AK\_Kauba\_kategooria\_tyyp\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Kaup\_kategooria(

kauba\_kategooria\_kood smallint NOT NULL,

nimetus d\_nimetus,

kauba\_kategooria\_tyyp\_kood smallint NOT NULL,

CONSTRAINT PK\_Kaup\_kategooria PRIMARY KEY (kauba\_kategooria\_kood),

CONSTRAINT AK\_Kauba\_kategooria\_nimetus UNIQUE (nimetus),

CONSTRAINT FK\_Kaup\_kategooria\_on\_seotud\_Kauba\_kategooria\_tyyp FOREIGN KEY (kauba\_kategooria\_tyyp\_kood) REFERENCES Kauba\_kategooria\_tyyp (kauba\_kategooria\_tyyp\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE Isiku\_seisundi\_liik(

isiku\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Isiku\_seisundi\_liik PRIMARY KEY (isiku\_seisundi\_liik\_kood),

CONSTRAINT AK\_Isiku\_seisundi\_liik\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Kauba\_seisundi\_liik(

kauba\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Kauba\_seisundi\_liik PRIMARY KEY (kauba\_seisundi\_liik\_kood),

CONSTRAINT AK\_Kauba\_seisundi\_liik\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Kliendi\_seisundi\_liik(

kliendi\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Kliendi\_seisundi\_liik PRIMARY KEY (kliendi\_seisundi\_liik\_kood),

CONSTRAINT AK\_Kliendi\_seisundi\_liik\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Tootaja\_seisundi\_liik(

tootaja\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Tootaja\_seisundi\_liik PRIMARY KEY (tootaja\_seisundi\_liik\_kood),

CONSTRAINT AK\_Tootaja\_seisundi\_liik\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Brand(

brand\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Brand PRIMARY KEY (brand\_kood),

CONSTRAINT AK\_Brand\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Diagonaal(

diagonaal\_kood decimal(2,1) NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Diagonaal PRIMARY KEY (diagonaal\_kood),

CONSTRAINT AK\_Diagonaal\_nimetus UNIQUE (nimetus),

CONSTRAINT chk\_Diagonaal\_kood\_peab\_olema\_suurem\_nullist CHECK (diagonaal\_kood>0)

);

CREATE TABLE Ekraan\_resolutsioon(

ekraan\_resolutsioon\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Ekraan\_resolutsioon PRIMARY KEY (ekraan\_resolutsioon\_kood),

CONSTRAINT AK\_Ekraan\_resolutsioon\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Kaamera(

kaamera\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Kaamera PRIMARY KEY (kaamera\_kood),

CONSTRAINT AK\_Kaamera\_nimetus UNIQUE (nimetus),

CONSTRAINT chk\_Kaamera\_kood\_peab\_olema\_suurem\_nullist CHECK (kaamera\_kood>0)

);

CREATE TABLE Protsessor(

protsessor\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Protsessor PRIMARY KEY (protsessor\_kood),

CONSTRAINT AK\_Protsessor\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Sisemalu(

sisemalu\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Sisemalu PRIMARY KEY (sisemalu\_kood),

CONSTRAINT AK\_Sisemalu\_nimetus UNIQUE (nimetus),

CONSTRAINT chk\_Sisemalu\_kood\_peab\_olema\_suurem\_nullist CHECK (sisemalu\_kood>0)

);

CREATE TABLE Varv(

varv\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Varv PRIMARY KEY (varv\_kood),

CONSTRAINT AK\_Varv\_nimetus UNIQUE (nimetus)

);

CREATE TABLE Amet(

amet\_kood smallint NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

CONSTRAINT PK\_Amet PRIMARY KEY (amet\_kood),

CONSTRAINT AK\_Amet\_nimetus UNIQUE (nimetus)

) WITH (fillfactor=90);

CREATE TABLE Riik(

riik\_kood char(3) NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Riik PRIMARY KEY (riik\_kood),

CONSTRAINT AK\_Riik\_nimetus UNIQUE (nimetus),

CONSTRAINT chk\_Riik\_kood\_peab\_olema\_kolm\_tahemarki\_pikk CHECK (riik\_kood~'^[A-Z]{3}$')

);

CREATE TABLE Isik(

isik\_id serial NOT NULL,

isikukood varchar(50) NOT NULL,

isikukood\_riik varchar(3) NOT NULL,

e\_meil varchar(254) NOT NULL,

synni\_kp date NOT NULL,

parool varchar(60) NOT NULL,

reg\_aeg d\_reg\_aeg,

isiku\_seisundi\_liik\_kood smallint NOT NULL DEFAULT 1,

eesnimi varchar(1000) ,

perenimi varchar(1000) ,

elukoht varchar(1000) ,

CONSTRAINT PK\_Isik PRIMARY KEY (isik\_id),

CONSTRAINT FK\_Isik\_on\_seotud\_Isiku\_seisundi\_liik FOREIGN KEY (isiku\_seisundi\_liik\_kood) REFERENCES Isiku\_seisundi\_liik (isiku\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Isik\_on\_seotud\_Riik FOREIGN KEY (isikukood\_riik) REFERENCES Riik (riik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT AK\_Isik\_e\_meil UNIQUE (e\_meil),

CONSTRAINT AK\_Isik\_isikukood\_Isikukood\_riik UNIQUE (isikukood,isikukood\_riik),

CONSTRAINT chk\_Isik\_eesnimi\_ei\_koosne\_tyhikutest CHECK (eesnimi!~'^[[:space:]]+$'),

CONSTRAINT chk\_Isik\_eesnimi\_pole\_tyhi\_string CHECK (eesnimi!=''),

CONSTRAINT chk\_Isik\_eesnimi\_voi\_perenimi\_peab\_olema\_registreeritud CHECK ((eesnimi IS NOT NULL) OR (perenimi IS NOT NULL)),

CONSTRAINT chk\_Isik\_perenimi\_ei\_koosne\_tyhikutest CHECK (perenimi!~'^[[:space:]]+$'),

CONSTRAINT chk\_Isik\_perenimi\_pole\_tyhi\_string CHECK (perenimi!=''),

CONSTRAINT chk\_Isik\_synni\_kp\_on\_vahemikus\_01\_01\_1900\_ja\_31\_12\_2100 CHECK (synni\_kp BETWEEN '1900-01-01' AND '2100-12-31'),

CONSTRAINT chk\_Isik\_synni\_kp\_ei\_tohi\_olla\_suurem\_kui\_reg\_aeg CHECK (synni\_kp<=reg\_aeg),

CONSTRAINT chk\_Isik\_elukoht\_ei\_koosne\_tyhikutest CHECK (elukoht!~'^[[:space:]]+$'),

CONSTRAINT chk\_Isik\_elukoht\_ei\_koosne\_numbritest CHECK (elukoht!~'^[[:digit:]]+$'),

CONSTRAINT chk\_Isik\_elukoht\_ei\_ole\_tyhi\_string CHECK (elukoht!=''),

CONSTRAINT chk\_Isik\_e\_meil\_peab\_sisaldama\_at\_marki CHECK (e\_meil LIKE '%@%')

);

CREATE TABLE Klient(

isik\_id integer NOT NULL,

kliendi\_seisundi\_liik\_kood smallint NOT NULL DEFAULT 1,

on\_nous\_tylitamisega boolean NOT NULL DEFAULT false,

CONSTRAINT PK\_Klient PRIMARY KEY (isik\_id),

CONSTRAINT FK\_Klient\_on\_seotud\_Kliendi\_seisundi\_liik FOREIGN KEY (kliendi\_seisundi\_liik\_kood) REFERENCES Kliendi\_seisundi\_liik (kliendi\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Klient\_on\_seotud\_Isik FOREIGN KEY (isik\_id) REFERENCES Isik (isik\_id) ON DELETE Cascade

);

CREATE TABLE Tootaja(

isik\_id integer NOT NULL,

amet\_kood smallint NOT NULL,

tootaja\_seisundi\_liik\_kood smallint NOT NULL DEFAULT 1,

CONSTRAINT PK\_Tootaja PRIMARY KEY (isik\_id),

CONSTRAINT FK\_Tootaja\_on\_seotud\_Tootaja\_seisundi\_liik FOREIGN KEY (tootaja\_seisundi\_liik\_kood) REFERENCES Tootaja\_seisundi\_liik (tootaja\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Tootaja\_on\_seotud\_Isik FOREIGN KEY (isik\_id) REFERENCES Isik (isik\_id) ON DELETE Cascade,

CONSTRAINT FK\_Tootaja\_on\_seotud\_Amet FOREIGN KEY (amet\_kood) REFERENCES Amet (amet\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE Kaup(

kaup\_kood integer NOT NULL,

nimetus varchar(255) NOT NULL,

reg\_aeg d\_reg\_aeg,

hind decimal(19,4) NOT NULL,

kirjeldus d\_kirjeldus,

registreerija\_id integer NOT NULL,

kauba\_seisundi\_liik\_kood smallint NOT NULL DEFAULT 1,

brand\_kood integer NOT NULL,

pildi\_aadress varchar(500) ,

CONSTRAINT PK\_Kaup PRIMARY KEY (kaup\_kood),

CONSTRAINT FK\_Kaup\_on\_seotud\_Brand FOREIGN KEY (brand\_kood) REFERENCES Brand (brand\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Kaup\_on\_seotud\_Tootaja FOREIGN KEY (registreerija\_id) REFERENCES Tootaja (isik\_id) ON DELETE No Action,

CONSTRAINT FK\_Kaup\_on\_seotud\_Kauba\_seisundi\_liik FOREIGN KEY (kauba\_seisundi\_liik\_kood) REFERENCES Kauba\_seisundi\_liik (kauba\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT AK\_Kaup\_nimetus UNIQUE (nimetus),

CONSTRAINT chk\_Kaup\_nimetus\_ei\_koosne\_tyhikutest CHECK (nimetus!~'^[[:space:]]+$'),

CONSTRAINT chk\_Kaup\_nimetus\_pole\_tyhi\_string CHECK (nimetus!=''),

CONSTRAINT chk\_Kaup\_hind\_peab\_olema\_suurem\_nullist CHECK (hind>0),

CONSTRAINT chk\_Kaup\_pildi\_aadress\_sisaldab\_punkti\_ja\_faililaiendit CHECK (pildi\_aadress~'([^\s]+(\.(jpg|gif))$)')

) WITH (fillfactor=90);

CREATE TABLE Kaubale\_kategooria\_omamine(

kaup\_kood integer NOT NULL,

kauba\_kategooria\_kood smallint NOT NULL,

CONSTRAINT PK\_Kaubale\_kategooria\_omamine PRIMARY KEY (kaup\_kood,kauba\_kategooria\_kood),

CONSTRAINT FK\_Kaubale\_kategooria\_omamine\_on\_seotud\_Kaup FOREIGN KEY (kaup\_kood) REFERENCES Kaup (kaup\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Kaubale\_kategooria\_omamine\_on\_seotud\_Kaup\_kategooria FOREIGN KEY (kauba\_kategooria\_kood) REFERENCES Kaup\_kategooria (kauba\_kategooria\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE Kaup\_variant(

kaup\_kood integer NOT NULL,

kauba\_varv\_kood smallint NOT NULL,

CONSTRAINT PK\_Kaup\_variant PRIMARY KEY (kaup\_kood,kauba\_varv\_kood),

CONSTRAINT FK\_Kauba\_variant\_on\_seotud\_Kaup FOREIGN KEY (kaup\_kood) REFERENCES Kaup (kaup\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Kauba\_variant\_on\_seotud\_Varv FOREIGN KEY (kauba\_varv\_kood) REFERENCES Varv (varv\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE Nutitelefon(

kaup\_kood integer NOT NULL,

on\_veekindel boolean NOT NULL DEFAULT false,

on\_sormejalelugeja boolean NOT NULL DEFAULT false,

eesmine\_kaamera\_kood smallint NOT NULL,

tagumine\_kaamera\_kood smallint NOT NULL,

sisemalu\_kood smallint NOT NULL,

diagonaal\_kood decimal(2,1) NOT NULL,

protsessor\_kood smallint NOT NULL,

ekraan\_resolutsioon\_kood smallint NOT NULL,

CONSTRAINT PK\_Nutitelefon PRIMARY KEY (kaup\_kood),

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_Kaup FOREIGN KEY (kaup\_kood) REFERENCES Kaup (kaup\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_eesmine\_Kaamera FOREIGN KEY (eesmine\_kaamera\_kood) REFERENCES Kaamera (kaamera\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_tagumine\_Kaamera FOREIGN KEY (tagumine\_kaamera\_kood) REFERENCES Kaamera (kaamera\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_Sisemalu FOREIGN KEY (sisemalu\_kood) REFERENCES Sisemalu (sisemalu\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_Diagonaal FOREIGN KEY (diagonaal\_kood) REFERENCES Diagonaal (diagonaal\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_Protsessor FOREIGN KEY (protsessor\_kood) REFERENCES Protsessor (protsessor\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Nutitelefon\_on\_seotud\_Ekraan\_resolutsioon FOREIGN KEY (ekraan\_resolutsioon\_kood) REFERENCES Ekraan\_resolutsioon (ekraan\_resolutsioon\_kood) ON DELETE No Action ON UPDATE Cascade

);

## Vaated

CREATE OR REPLACE VIEW Koik\_kaubad WITH (security\_barrier) AS (

SELECT Kaup.kaup\_kood AS kaup\_kood, Kaup.nimetus AS nimetus, Brand.nimetus AS brand, Kauba\_seisundi\_liik.nimetus AS seisund, Kaup.hind AS hind, Kaup.kirjeldus AS kirjeldus

FROM Kauba\_seisundi\_liik INNER JOIN (Brand INNER JOIN Kaup ON Brand.brand\_kood = Kaup.brand\_kood) ON Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood = Kaup.kauba\_seisundi\_liik\_kood);

COMMENT ON VIEW Koik\_kaubad IS 'Vaade kuvab kõigi kaupade nimekirja, kus on kood, nimetus, bränd, hetkeseisundi nimetus, hind, kirjeldus. Vaadet on vaja, kui juhataja või kauba haldur soovib vaadata kõikide kaupade andmeid.';

CREATE OR REPLACE VIEW Kaupade\_detailid WITH (security\_barrier) AS (

SELECT Kaup.kaup\_kood, Kaup.nimetus AS nimetus, Kaup.hind, Kaup.kirjeldus, Kaup.pildi\_aadress, Kaup.reg\_aeg, Kaup.registreerija\_id, Isik.eesnimi AS registreerija\_eesnimi, Isik.perenimi AS registreerija\_perenimi, Isik.e\_meil AS registreerija\_emeil, Kauba\_seisundi\_liik.nimetus AS seisund, Brand.nimetus AS brand, Nutitelefon.on\_veekindel, Nutitelefon.on\_sormejaljelugeja, Nutitelefon.eesmine\_kaamera\_kood AS eesmine\_kaamera, Nutitelefon.tagumine\_kaamera\_kood AS tagumine\_kaamera, Nutitelefon.sisemalu\_kood AS sisemalu, Nutitelefon.diagonaal\_kood AS diagonaal, Protsessor.nimetus AS protsessor, Ekraan\_resolutsioon.nimetus AS resolutsioon

FROM Protsessor INNER JOIN ((Isik INNER JOIN (Kauba\_seisundi\_liik INNER JOIN (Brand INNER JOIN Kaup ON Brand.brand\_kood = Kaup.kaup\_kood) ON Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood = Kaup.kauba\_seisundi\_liik\_kood) ON Isik.isik\_id = Kaup.registreerija\_id) LEFT JOIN (Ekraan\_resolutsioon INNER JOIN Nutitelefon ON Ekraan\_resolutsioon.ekraan\_resolutsioon\_kood = Nutitelefon.ekraan\_resolutsioon\_kood) ON Kaup.kaup\_kood = Nutitelefon.kaup\_kood) ON Protsessor.protsessor\_kood = Nutitelefon.protsessor\_kood);

COMMENT ON VIEW Kaupade\_detailid IS 'Vaade kuvab kauba põhiandmed (kauba\_kood, nimetus, hind, kirjeldus, pildi aadress, registreerimise aeg, registreerinud töötaja eesnimi, perenimi ja e-meili aadress) ning kaubaga seotud nutitelefoni andmed’. Vaadet on vaja, kui juhataja või kauba haldur soovib kaupade koondaruandes vaadata mõnda kaupa detailsemalt. ';

CREATE OR REPLACE VIEW Kaupade\_variandid WITH (security\_barrier) AS (

SELECT Kaup\_variant.kaup\_kood, Kaup\_variant.kauba\_varv\_kood, Varv.nimetus AS varv

FROM Kaup\_variant, Varv

WHERE (Kaup\_variant.kauba\_varv\_kood = Varv.varv\_kood));

COMMENT ON VIEW Kaupade\_variandid IS 'Vaade kuvab kõikide olemasolevate kaupade koodid, värvi koodid ja värvi nimetused. Vaadet on vaja, et kaupade detailandmeid vaadates oleks võimalik kuvada ka kauba variandid.';

CREATE OR REPLACE VIEW Kaupade\_kategooriad WITH (security\_barrier) AS (

SELECT Kaubale\_kategooria\_omamine.kaup\_kood, Kaup\_kategooria.nimetus AS kategooria, Kauba\_kategooria\_tyyp.nimetus AS kategooria\_tyyp

FROM Kaubale\_kategooria\_omamine, Kaup\_kategooria, Kauba\_kategooria\_tyyp

WHERE Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood = Kaup\_kategooria.kauba\_kategooria\_kood AND Kaup\_kategooria.kauba\_kategooria\_tyyp\_kood = Kauba\_kategooria\_tyyp.kauba\_kategooria\_tyyp\_kood);

COMMENT ON VIEW Kaupade\_kategooriad IS 'Vaade kuvab kõikide olemasolevate kaupade koodid, nendele vastavate kategooriate nimetused ja kauba kategooriate tüübid. Vaadet on vaja, et kaupade detailandmeid vaadates oleks võimalik kuvada ka kauba kategooriad.';

CREATE OR REPLACE VIEW Aktiivsed\_mitteaktiivsed\_kaubad WITH (security\_barrier) AS ( SELECT Kaup.kaup\_kood AS kaup\_kood, Kaup.nimetus AS nimetus, Kauba\_seisundi\_liik.nimetus AS seisund, Kaup.hind AS hind, Kaup.kirjeldus AS kirjeldus

FROM Kauba\_seisundi\_liik INNER JOIN Kaup ON Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood = Kaup.kauba\_seisundi\_liik\_kood

WHERE Kaup.kauba\_seisundi\_liik\_kood In (2,3));

COMMENT ON VIEW Aktiivsed\_mitteaktiivsed\_kaubad IS 'Vaade kuvab aktiivsete või mitteaktiivsete kaupade nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, hind ja kirjeldus. Vaadet läheb vaja kaupade lõpetamisel, et juhataja teaks, milliseid kaupu on võimalik lõpetada.';

CREATE OR REPLACE VIEW Kaupade\_koondaruanne WITH (security\_barrier) AS (

SELECT Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood AS Kauba\_seisundi\_liik\_kood, UPPER(Kauba\_seisundi\_liik.nimetus) AS seisund, Count(Kaup.kaup\_kood) AS kaupade\_arv

FROM Kauba\_seisundi\_liik LEFT JOIN Kaup ON Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood = Kaup.kauba\_seisundi\_liik\_kood

GROUP BY Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood, Kauba\_seisundi\_liik.nimetus

ORDER BY Count(Kaup.kaup\_kood) DESC , UPPER(Kauba\_seisundi\_liik.nimetus));

COMMENT ON VIEW Kaupade\_koondaruanne IS 'Vaade kuvab iga kauba elutsükli seisundi

kohta selle seisundi koodi, nimetuse (suurtähtedega) ja hetkel selles

seisundis olevate kaupade arvu. Vaadet on vaja, kui juhataja soovib vaadata kaupade koondaruannet.';

CREATE OR REPLACE VIEW Kaupade\_kategooriad\_ja\_kategooria\_tyybid WITH (security\_barrier) AS (

SELECT Kaup.kaup\_kood, string\_agg(Kauba\_kategooria\_tyyp.nimetus || ': ' || Kaup\_kategooria.nimetus, ', ') AS kategooriad\_ja\_kategooria\_tyybid

FROM Kaup LEFT JOIN (Kauba\_kategooria\_tyyp INNER JOIN (Kaup\_kategooria INNER JOIN Kaubale\_kategooria\_omamine ON Kaup\_kategooria.kauba\_kategooria\_kood = Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood) ON Kauba\_kategooria\_tyyp.kauba\_kategooria\_tyyp\_kood = Kaup\_kategooria.kauba\_kategooria\_tyyp\_kood) ON Kaup.kaup\_kood = Kaubale\_kategooria\_omamine.kaup\_kood

GROUP BY Kaup.kaup\_kood

ORDER BY Kaup.kaup\_kood);

COMMENT ON VIEW Kaupade\_kategooriad\_ja\_kategooria\_tyybid IS 'Vaade kuvab iga kauba kohta tema kategooriad koos kategooria tüüpidega. Kuvatakse ka tulemused, millel pole ühtegi kategooriat ja kategooria tüüpi';

CREATE OR REPLACE VIEW Kaubad\_ja\_registreerijad\_json WITH (security\_barrier) AS ( SELECT json\_build\_object(

'kaubad', json\_agg(

json\_build\_object(

'kaup\_kood', Kaup.kaup\_kood,

'kaup\_nimetus', Kaup.nimetus,

'kaup\_hind', Kaup.hind,

'registreerija', json\_build\_object(

'registreerija\_id', Kaup.registreerija\_id,

'registreerija\_amet', Amet.nimetus

)

)

)) AS kaubad\_ja\_registreerijad

FROM Kaup, Tootaja, Amet

WHERE Kaup.registreerija\_id = Tootaja.isik\_id AND Tootaja.amet\_kood = Amet.amet\_kood);

COMMENT ON VIEW Kaubad\_ja\_registreerijad\_json IS 'Vaade kuvab kõikide kaupade kauba koodi, kauba nimetuse ja kauba hinna ning kauba registreerija kohta registreerija id ja registreerija ameti. Info esitatakse Json formaadis.';

Järgnevalt on toodud vaate Kaubad\_ja\_registreerijad\_json põhjal tehtud päring, mille tulemuseks on tabel, kus on üks rida iga kauba eksemplari kohta ja üks veerg.

SELECT json\_array\_elements(kaupade\_ja\_registreerijate\_info->'kaubad')->>'kaup\_kood' AS kaup\_kood FROM kaupade\_ja\_registreerijate\_info\_json;

## Protseduursed keeled

Antud projektis on funktsioonid loodud PL/pgSQL keeles. Andmebaasisüsteemis PostgreSQL 10.1 installeeritakse PL/pgSQL keel kõikidesse andmebaasidesse vaikimisi. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

## Trigeri funktsioonid ja trigerid

CREATE OR REPLACE FUNCTION f\_kauba\_lopetamine ()

RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION ’Ei saa lõpetada kaupa, mis ei ole olekus aktiivne või mitteaktiivne!’;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_kauba\_lopetamine() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei saa lõpetada kaupa, mis ei ole olekus aktiivne või mitteaktiivne.';

CREATE TRIGGER trig\_kauba\_lopetamine BEFORE UPDATE OF kauba\_seisundi\_liik\_kood ON Kaup FOR EACH ROW WHEN ((OLD.kauba\_seisundi\_liik\_kood NOT IN (2, 3, 4)) AND (NEW.kauba\_seisundi\_liik\_kood=4)) EXECUTE PROCEDURE f\_kauba\_lopetamine();

CREATE OR REPLACE FUNCTION f\_kauba\_aktiveerimine ()

RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION ’Ei saa aktiveerida kaupa, mis ei ole olekus ootel või mitteaktiivne!’;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_kauba\_aktiveerimine() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei saa aktiveerida kaupa, mis ei ole olekus ootel või mitteaktiivne.';

CREATE TRIGGER trig\_kauba\_aktiveerimine BEFORE UPDATE OF kauba\_seisundi\_liik\_kood ON Kaup FOR EACH ROW WHEN ((OLD.kauba\_seisundi\_liik\_kood NOT IN (1, 2, 3)) AND (NEW.kauba\_seisundi\_liik\_kood=2)) EXECUTE PROCEDURE f\_kauba\_aktiveerimine();

CREATE OR REPLACE FUNCTION f\_kauba\_kategooriasse\_kuulumine ()

RETURNS trigger AS $$

BEGIN

IF (NEW.kaup\_kood NOT IN (SELECT kaup\_kood FROM Kaubale\_kategooria\_omamine FOR UPDATE)) THEN

RAISE EXCEPTION 'Ei saa aktiveerida kaupa, mis ei kuulu ühtegi kategooriasse';

ELSE

RETURN NEW;

END IF;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_kategooriasse\_kuulumine() IS 'See trigeri funktsioon aitab jõustada ärireegli: Aktiivses olekus kaup peab kuuluma vähemalt ühte kategooriasse.';

CREATE TRIGGER trig\_kauba\_kategooriasse\_kuulumine BEFORE UPDATE OF kauba\_seisundi\_liik\_kood ON Kaup FOR EACH ROW WHEN (NEW.kauba\_seisundi\_liik\_kood=2) EXECUTE PROCEDURE f\_kauba\_kategooriasse\_kuulumine();

CREATE OR REPLACE FUNCTION f\_kauba\_mitteaktiveerimine ()

RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION ’Ei saa mitteaktiveerida kaupa, mis ei ole olekus aktiivne!’;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_kauba\_mitteaktiveerimine() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei saa mitteaktiveerida kaupa, mis ei ole olekus aktiivne.';

CREATE TRIGGER trig\_kauba\_mitteaktiveerimine BEFORE UPDATE OF kauba\_seisundi\_liik\_kood ON Kaup FOR EACH ROW WHEN ((OLD.kauba\_seisundi\_liik\_kood NOT IN (2, 3)) AND (NEW.kauba\_seisundi\_liik\_kood=3)) EXECUTE PROCEDURE f\_kauba\_mitteaktiveerimine();

CREATE OR REPLACE FUNCTION f\_kauba\_kategooriasse\_lisamine ()

RETURNS trigger AS $$

DECLARE

m\_kauba\_seisundi\_liik\_kood SMALLINT;

BEGIN

SELECT kauba\_seisundi\_liik\_kood INTO m\_kauba\_seisundi\_liik\_kood FROM Kaup WHERE kaup\_kood=NEW.kaup\_kood FOR UPDATE;

IF (m\_kauba\_seisundi\_liik\_kood NOT IN (1, 3)) THEN

RAISE EXCEPTION 'Ei saa lisada kategooriasse kaupa, mis ei ole olekus ootel või mitteaktiivne! ';

ELSE

RETURN NEW;

END IF;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_kategooriasse\_lisamine() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei saa kategooriasse lisada kaupa, mis ei ole olekus ootel või mitteaktiivne.';

CREATE TRIGGER trig\_kauba\_kategooriasse\_lisamine BEFORE INSERT ON Kaubale\_kategooria\_omamine FOR EACH ROW EXECUTE PROCEDURE f\_kauba\_kategooriasse\_lisamine();

## Reeglid

Reegleid antud projektis ei looda.

## Funktsioonid

CREATE OR REPLACE FUNCTION f\_kauba\_lopetamine (p\_kaup\_kood Kaup.kaup\_kood%TYPE) RETURNS Kaup.kaup\_kood%TYPE AS $$

UPDATE Kaup SET kauba\_seisundi\_liik\_kood = 4 WHERE Kaup.kaup\_kood = p\_kaup\_kood

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_lopetamine(p\_kaup\_kood Kaup.kaup\_kood%TYPE) IS 'Funktsioon muudab tabelis Kaup funktsiooni sisendina antud kauba koodi põhjal seisundi lõpetatuks. Funktsiooni kasutatakse OP5 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_kauba\_mitteaktiveerimine (p\_kaup\_kood Kaup.kaup\_kood%TYPE) RETURNS Kaup.kaup\_kood%TYPE AS $$

UPDATE Kaup SET kauba\_seisundi\_liik\_kood = 3 WHERE Kaup.kaup\_kood = p\_kaup\_kood

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_mitteaktiveerimine(p\_kaup\_kood Kaup.kaup\_kood%TYPE) IS 'Funktsioon muudab tabelis Kaup funktsiooni sisendina antud kauba koodi põhjal seisundi mitteaktiivseks. Funktsiooni kasutatakse OP4 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_kauba\_kategooriasse\_lisamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE)

RETURNS Kaubale\_kategooria\_omamine.kaup\_kood%TYPE AS $$

INSERT INTO Kaubale\_kategooria\_omamine (kaup\_kood, kauba\_kategooria\_kood) VALUES (p\_kaup\_kood, p\_kauba\_kategooria\_kood)

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_kategooriasse\_lisamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE) IS 'Funktsioon lisab Kaubale\_kategooria\_omamine tabelisse kauba ja kaubale vastava kategooria, mis vastavad sisendina antud parameetritele kaup\_kood ja kauba\_kategooria\_kood. Funktsiooni kasutatakse OP7 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_kauba\_kategooriast\_eemaldamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE)

RETURNS Kaubale\_kategooria\_omamine.kaup\_kood%TYPE AS $$

DELETE FROM Kaubale\_kategooria\_omamine

WHERE Kaubale\_kategooria\_omamine.kaup\_kood = p\_kaup\_kood AND Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood = p\_kauba\_kategooria\_kood

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_kategooriast\_eemaldamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE) IS 'Funktsioon eemaldab Kaubale\_kategooria\_omamine tabelist kauba kategooriasse kuuluvuse, mis vastab sisendina antud parameetritele kaup\_kood ja kauba\_kategooria\_kood. Funktsiooni kasutatakse OP8 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_kauba\_variandi\_lisamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE)

RETURNS Kaup\_variant.kaup\_kood%TYPE AS $$

INSERT INTO Kaup\_variant (kaup\_kood, kauba\_varv\_kood) VALUES (p\_kaup\_kood, p\_kauba\_varv\_kood)

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_variandi\_lisamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE) IS 'Funktsioon lisab Kaup\_variant tabelisse kauba variandi, mis vastab sisendina antud parameetritele kaup\_kood ja kauba\_varv\_kood. Funktsiooni kasutatakse OP9 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_kauba\_variandi\_eemaldamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE)

RETURNS Kaup\_variant.kaup\_kood%TYPE AS $$

DELETE FROM Kaup\_variant

WHERE Kaup\_variant.kaup\_kood = p\_kaup\_kood AND Kaup\_variant.kauba\_varv\_kood = p\_kauba\_varv\_kood

RETURNING p\_kaup\_kood;

$$ LANGUAGE sql SECURITY DEFINER STRICT

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_kauba\_variandi\_eemaldamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE) IS 'Funktsioon eemaldab Kaup\_variant tabelis kauba variandi, mis vastab sisendina antud parameetritele kaup\_kood ja kauba\_varv\_kood. Funktsiooni kasutatakse OP10 läbiviimiseks.';

CREATE OR REPLACE FUNCTION f\_on\_juhataja(p\_e\_meil text, p\_parool text)

RETURNS boolean AS $$ DECLARE rslt boolean;

BEGIN

SELECT INTO rslt (Isik.parool = public.crypt(p\_parool, Isik.parool))

FROM Isik, Tootaja

WHERE Isik.isik\_id = Tootaja.isik\_id AND Upper(Isik.e\_meil)=Upper(p\_e\_meil) AND Tootaja.amet\_kood = 1 AND Tootaja.tootaja\_seisundi\_liik\_kood IN (1, 2, 3, 4);

RETURN coalesce(rslt, FALSE);

END;

$$ LANGUAGE plpgsql SECURITY DEFINER STABLE

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_on\_juhataja(p\_e\_meil text, p\_parool text) IS 'Selle funktsiooni abil autenditakse juhatajat. Parameetri p\_e\_meil oodatav väärtus on tõstutundetu e\_meil ja p\_parool oodatav väärtus on tõstutundlik avatekstiline parool. Juhatajal on õigus süsteemi siseneda, vaid siis kui tema seisundiks on tööl, katseajal, puhkusel või haiguslehel.';

## Indeksid

Järgnevalt on toodud indeksite loomise laused.

### Välisvõtmete veergudele lisatavad indeksid

CREATE INDEX IXFK\_Kauba\_kategooria\_tyyp ON Kaup\_kategooria (kauba\_kategooria\_tyyp\_kood ASC);

CREATE INDEX IXFK\_Isiku\_seisundi\_liik ON Isik (isiku\_seisundi\_liik\_kood ASC);

CREATE INDEX IXFK\_Riik ON Isik (isikukood\_riik ASC);

CREATE INDEX IXFK\_Kliendi\_seisundi\_liik ON Klient (kliendi\_seisundi\_liik\_kood ASC);

CREATE INDEX IXFK\_Amet ON Tootaja (amet\_kood ASC);

CREATE INDEX IXFK\_Tootaja\_seisundi\_liik ON Tootaja (tootaja\_seisundi\_liik\_kood ASC);

CREATE INDEX IXFK\_Brand ON Kaup (brand\_kood ASC);

CREATE INDEX IXFK\_Kauba\_seisundi\_liik ON Kaup (kauba\_seisundi\_liik\_kood ASC);

CREATE INDEX IXFK\_Tootaja ON Kaup (registreerija\_id ASC);

CREATE INDEX IXFK\_Kaup\_kategooria ON Kaubale\_kategooria\_omamine (kauba\_kategooria\_kood ASC);

CREATE INDEX IXFK\_Varv ON Kaup\_variant(kauba\_varv\_kood ASC);

CREATE INDEX IXFK\_Diagonaal ON Nutitelefon (diagonaal\_kood ASC);

CREATE INDEX IXFK\_Eesmine\_kaamera ON Nutitelefon (eesmine\_kaamera\_kood ASC);

CREATE INDEX IXFK\_Ekraan\_resolutsioon ON Nutitelefon (ekraan\_resolutsioon\_kood ASC);

CREATE INDEX IXFK\_Protsessor ON Nutitelefon (protsessor\_kood ASC);

CREATE INDEX IXFK\_Sisemalu ON Nutitelefon (sisemalu\_kood ASC);

CREATE INDEX IXFK\_Tagumine\_kaamera ON Nutitelefon (tagumine\_kaamera\_kood ASC);

### Täiendavad sekundaarsed indeksid

Täiendavaid sekundaarseid indekseid käesolevas projektis ei looda.

### Funktsioonil põhinevad indeksid

„Tagamaks kasutajanimede unikaalsuse sõltumata sellest, kas neis kasutatakse suur- või väiketähti ning kiirendamaks kasutajanimede tõstutundetut otsingut, asendan tabelites Isik veerule e\_meil loodud unikaalsuse kitsenduse unikaalse funktsioonil Upper põhineva indeksiga.„ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

ALTER TABLE Isik DROP CONSTRAINT AK\_Isik\_e\_meil;

CREATE UNIQUE INDEX AK\_Isik\_e\_meil ON Isik (Upper(e\_meil));

## Klassifikaatorite väärtustamise SQL laused

INSERT INTO Amet(amet\_kood, nimetus, kirjeldus) VALUES (1, ’Juhataja’, ’Juhataja juhib antud ettevõtte tegevust. Juhatajal ülesannete hulka kuulub kõigi kaupade vaatamine, vajadusel kaupade lõpetamine ning kaupade koondaruande vaatamine.’);

INSERT INTO Amet(amet\_kood, nimetus, kirjeldus) VALUES (2, ’Kauba haldur’, ’Kauba halduri ülesanneteks on kaupade haldamine ehk kaupade registreerimine, kaupade unustamine, kauba andmete muutmine, kaupade aktiveerimine ja kaupade mitteaktiivseks muutmine andmebaasis. Lisaks vaatab kauba haldur vajadusel kõiki aktiivseid või mitteaktiivseid kaupu.’);

INSERT INTO Amet(amet\_kood, nimetus, kirjeldus) VALUES (3, ’Kliendihaldur’, ’Kliendihalduri ülesandeks on klientidega suhtlemine, klientide nõustamine ja abistamine ning andmebaasist aktiivsete kaupade vaatamine, et teha kliendile asjakohaseid pakkumisi.’);

INSERT INTO Brand(brand\_kood, nimetus) VALUES (1, ’Apple’);

INSERT INTO Brand(brand\_kood, nimetus) VALUES (2, ’Samsung’);

INSERT INTO Brand(brand\_kood, nimetus) VALUES (3, ’Oneplus’);

INSERT INTO Brand(brand\_kood, nimetus) VALUES (4, ’Nokia’);

INSERT INTO Brand(brand\_kood, nimetus) VALUES (5, ’Huawei’);

INSERT INTO Diagonaal(diagonaal\_kood, nimetus) VALUES (4.7, ’4,7 tolli (eriti väike)’);

INSERT INTO Diagonaal(diagonaal\_kood, nimetus) VALUES (5.2, ’5,2 tolli (keskmise suurusega)’);

INSERT INTO Diagonaal(diagonaal\_kood, nimetus) VALUES (5.5, ’5,5 tolli (keskmisest pisut suurem)’);

INSERT INTO Diagonaal(diagonaal\_kood, nimetus) VALUES (5.7, ’5,7 tolli (suur)’);

INSERT INTO Diagonaal(diagonaal\_kood, nimetus) VALUES (5.8, ’5,8 tolli (suur)’);

INSERT INTO Ekraan\_resolutsioon(ekraan\_resolutsioon\_kood, nimetus) VALUES (1, ’750 x 1334’);

INSERT INTO Ekraan\_resolutsioon(ekraan\_resolutsioon\_kood, nimetus) VALUES (2, ’480 x 800’);

INSERT INTO Ekraan\_resolutsioon(ekraan\_resolutsioon\_kood, nimetus) VALUES (3, ’2436 x 1125’);

INSERT INTO Ekraan\_resolutsioon(ekraan\_resolutsioon\_kood, nimetus) VALUES (4, ’1080 x 1920’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (2, ’2 Mpx (väga väike)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (3, ’3 Mpx (väga väike)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (4, ’4 Mpx (väga väike)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (5, ’5 Mpx (väike)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (6, ’6 Mpx (keskmine)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (7, ’7 Mpx (keskmine)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (8, ’8 Mpx (keskmine)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (12, ’12 Mpx (suur)’);

INSERT INTO Kaamera(kaamera\_kood, nimetus) VALUES (16, ’16 Mpx (väga suur)’);

INSERT INTO Protsessor(protsessor\_kood, nimetus) VALUES (1, ’Quad-Core’);

INSERT INTO Protsessor(protsessor\_kood, nimetus) VALUES (2, ’Octa-Core’);

INSERT INTO Protsessor(protsessor\_kood, nimetus) VALUES (3, ’Dual-Core’);

INSERT INTO Protsessor(protsessor\_kood, nimetus) VALUES (4, ’Hexa-Core’);

INSERT INTO Sisemalu(sisemalu\_kood, nimetus) VALUES (16, ’16 gigabaiti (väike mälumaht’));

INSERT INTO Sisemalu(sisemalu\_kood, nimetus) VALUES (32, ’32 gigabaiti (pisut alla keskmise mälumaht));

INSERT INTO Sisemalu(sisemalu\_kood, nimetus) VALUES (64, ’64 gigabaiti (keskmine mälumaht));

INSERT INTO Sisemalu(sisemalu\_kood, nimetus) VALUES (128, ’128 gigabaiti (suur mälumaht));

INSERT INTO Varv(varv\_kood, nimetus) VALUES (1, ’Must’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (2, ’Valge’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (3, ’Hall’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (4, ’Kosmosehall’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (5, ’Hõbedane’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (6, ’Teemantmust’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (7, ’Punane’);

INSERT INTO Varv(varv\_kood, nimetus) VALUES (8, ’Kuldne’);

INSERT INTO Kauba\_kategooria\_tyyp(kauba\_kategooria\_tyyp\_kood, nimetus) VALUES (1, ’Välimus’);

INSERT INTO Kauba\_kategooria\_tyyp(kauba\_kategooria\_tyyp\_kood, nimetus) VALUES (2, ’Sihtgrupp’);

INSERT INTO Kauba\_kategooria\_tyyp(kauba\_kategooria\_tyyp\_kood, nimetus) VALUES (3, ’Kasutusotstarve’);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (1, ’Moekad telefonid’, 1);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (2, ’Naistetelefonid’, 2);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (3, ’Meestetelefonid’, 2);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (4, ’Lastetelefonid’, 2);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (5, ’Äriklassitelefonid’, 3);

INSERT INTO Kaup\_kategooria(kauba\_kategooria\_kood, nimetus, kauba\_kategooria\_tyyp\_kood) VALUES (6, ’Sportlikud telefonid’, 3);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (1, ’Katseajal’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (2, ’Puhkusel’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (3, ’Haiguslehel’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (4, ’Tööl’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (5, ’Töösuhe peatatud’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (6, ’Töösuhe lõpetatud’);

INSERT INTO Tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (7, ’Vallandatud’);

INSERT INTO Kliendi\_seisundi\_liik(kliendi\_seisundi\_liik\_kood, nimetus) VALUES (1, ’Aktiivne’);

INSERT INTO Kliendi\_seisundi\_liik(kliendi\_seisundi\_liik\_kood, nimetus) VALUES (2, ’Mitteaktiivne’);

INSERT INTO Kauba\_seisundi\_liik(kauba\_seisundi\_liik\_kood, nimetus) VALUES (1, ’Ootel’);

INSERT INTO Kauba\_seisundi\_liik(kauba\_seisundi\_liik\_kood, nimetus) VALUES (2, ’Aktiivne’);

INSERT INTO Kauba\_seisundi\_liik(kauba\_seisundi\_liik\_kood, nimetus) VALUES (3, ’Mitteaktiivne’);

INSERT INTO Kauba\_seisundi\_liik(kauba\_seisundi\_liik\_kood, nimetus) VALUES (4, ’Lõpetatud’);

INSERT INTO Kauba\_seisundi\_liik(kauba\_seisundi\_liik\_kood, nimetus) VALUES (5, ’Unustatud’);

INSERT INTO Isiku\_seisundi\_liik(isiku\_seisundi\_liik\_kood, nimetus) VALUES (1, ’Elus’);

INSERT INTO Isiku\_seisundi\_liik(isiku\_seisundi\_liik\_kood, nimetus) VALUES (2, ’Surnud’);

## JSON formaadis lähteandmete laadimine

„Kuna järgnevates lausetes pöördutakse paketis *postgres\_fdw* olevate funktsioonide poole, siis tuleb eelnevalt see pakett andmebaasi lisada.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

CREATE EXTENSION IF NOT EXISTS postgres\_fdw;

CREATE SERVER minu\_testandmete\_server\_apex FOREIGN DATA WRAPPER postgres\_fdw OPTIONS (host 'apex.ttu.ee', dbname 'testandmed', port '5432');

CREATE USER MAPPING FOR t155605 SERVER minu\_testandmete\_server\_apex OPTIONS (user 't155605', password 'Tanel123');

CREATE FOREIGN TABLE Riik\_jsonb ( riik JSONB ) SERVER minu\_testandmete\_server\_apex;

SELECT \* FROM Riik\_jsonb;

INSERT INTO Riik (riik\_kood, nimetus)

SELECT riik->>'Alpha-3 code' AS riik\_kood,

riik->>'English short name lower case' AS nimetus

FROM Riik\_jsonb;

SELECT \* FROM Riik;

„Kuna järgnevates lausetes pöördutakse paketis *pgcrypto* olevate funktsioonide poole, siis tuleb eelnevalt see pakett andmebaasi lisada.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;

SELECT \* FROM Isik\_jsonb;

INSERT INTO Isik(riik\_kood, isikukood, eesnimi, perenimi, e\_mail, synni\_kp, isiku\_seisundi\_liik\_kood, parool, elukoht)

SELECT riik\_kood, isikukood, eesnimi, perenimi, e\_mail, synni\_kp::date, isiku\_seisundi\_liik\_kood::smallint, parool, elukoht

FROM (SELECT isik->>'riik' AS riik\_kood,

jsonb\_array\_elements(isik->'isikud')->>'isikukood' AS isikukood, jsonb\_array\_elements(isik->'isikud')->>'eesnimi' AS eesnimi, jsonb\_array\_elements(isik->'isikud')->>'perekonnanimi' AS perenimi, jsonb\_array\_elements(isik->'isikud')->>'email' AS e\_mail, jsonb\_array\_elements(isik->'isikud')->>'synni\_aeg' AS synni\_kp, jsonb\_array\_elements(isik->'isikud')->>'seisund' AS isiku\_seisundi\_liik\_kood, jsonb\_array\_elements(isik->'isikud')->>'parool' AS parool, jsonb\_array\_elements(isik->'isikud')->>'aadress' AS elukoht

FROM isik\_jsonb) AS lahteandmed

WHERE isiku\_seisundi\_liik\_kood::smallint=1;

SELECT \* FROM Isik;

## 

## Täiendavate testandmete lisamine

INSERT INTO Kaup(kaup\_kood, nimetus, reg\_aeg, hind, kirjeldus, registreerija\_id, kauba\_seisundi\_liik\_kood, brand\_kood, pildi\_aadress) VALUES (1, ’Iphone 7’, 2017-11-07 14:08:49, 699.99, ’ Suurepärase kaameraga ning kiire protsessoriga telefon!’, 7, 1, 1, ’img/iphone7.jpg’);

INSERT INTO Kaup(kaup\_kood, nimetus, reg\_aeg, hind, kirjeldus, registreerija\_id, kauba\_seisundi\_liik\_kood, brand\_kood, pildi\_aadress) VALUES (2, ’Iphone X’, 2017-11-07 14:10:24, 1149.99,’ Telefonil on üks parimaid kaameraid ja ekraane mis hetkel saadaval on!’, 8, 2, 1, ’img/iphoneX.jpg’);

INSERT INTO Kaup(kaup\_kood, nimetus, reg\_aeg, hind, kirjeldus, registreerija\_id, kauba\_seisundi\_liik\_kood, brand\_kood, pildi\_aadress) VALUES (3, ’Oneplus 5’, 2017-12-18 16:45:43, 499.90, ’Väga võimeka protsessoriga nutitelefon!’, 7, 3, 3, ’img/oneplus5.jpg’);

INSERT INTO Kaup\_variant(kaup\_kood, kauba\_varv\_kood) VALUES (1, 6);

INSERT INTO Kaup\_variant(kaup\_kood, kauba\_varv\_kood) VALUES (2, 4);

INSERT INTO Kaup\_variant(kaup\_kood, kauba\_varv\_kood) VALUES (2, 5);

INSERT INTO Kaup\_variant(kaup\_kood, kauba\_varv\_kood) VALUES (3, 4);

INSERT INTO Kaup\_variant(kaup\_kood, kauba\_varv\_kood) VALUES (3, 8);

INSERT INTO Nutitelefon(kaup\_kood, on\_veekindel, on\_sormejaljelugeja, eesmine\_kaamera\_kood, tagumine\_kaamera\_kood, sisemalu\_kood, diagonaal\_kood, protsessor\_kood, erkaani\_resolutsioon\_kood) VALUES (1, true, true, 7, 12, 64, 4.7, 1, 1);

INSERT INTO Nutitelefon(kaup\_kood, on\_veekindel, on\_sormejaljelugeja, eesmine\_kaamera\_kood, tagumine\_kaamera\_kood, sisemalu\_kood, diagonaal\_kood, protsessor\_kood, erkaani\_resolutsioon\_kood) VALUES (2, true, false, 7, 12, 64, 5.8, 4, 3);

INSERT INTO Nutitelefon(kaup\_kood, on\_veekindel, on\_sormejaljelugeja, eesmine\_kaamera\_kood, tagumine\_kaamera\_kood, sisemalu\_kood, diagonaal\_kood, protsessor\_kood, erkaani\_resolutsioon\_kood) VALUES (3, false, true, 16, 16, 128, 5.5, 2, 4);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (1, 1);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (1, 2);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (1, 3);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (2, 1);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (2, 5);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (3, 2);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (3, 3);

INSERT INTO Kaubale\_kategooria\_omamine(kaup\_kood, kauba\_kategooria\_kood) VALUES (3, 6);

INSERT INTO Klient(isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (4, 1, false);

INSERT INTO Klient(isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (5, 1, false);

INSERT INTO Klient(isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (6, 1, false);

INSERT INTO Tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (7, 1, 4);

INSERT INTO Tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (8, 2, 4);

INSERT INTO Tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (9, 3, 4);

INSERT INTO Tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (10, 3, 2);

## Andmebaasi statistika kogumine

„Serveris on seadistatud prügikoristuse ning statistika kogumise automaatne käivitumine (autovacuum protsess).„ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

„Järgnev lause on mõeldud prügikoristuse ja statistika värskendamise erakorraliseks käivitamiseks.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

VACUUM ANALYZE;

„PostgreSQLis jäävad kustutatud ridade või muutmise tulemusena üleliigseks muutunud ridade andmed andmebaasi sisemisele tasemele alles. Nende lõplikuks kustutamiseks sisemiselt tasemelt tuleb käivitada prügikoristus – VACUUM. ANALYZE määrang lauses tagab, et lisaks prügikoristusele värskendatakse ka andmebaasi statistikat.“ (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

## Päringu täitmisplaani näide

Antud töös analüüsitakse päringut vaate aktiivsed\_mitteaktiivsed\_kaubad põhjal.

Analüüsi tulemus on toodud joonisel 17.

CREATE OR REPLACE VIEW Aktiivsed\_mitteaktiivsed\_kaubad WITH (security\_barrier) AS ( SELECT Kaup.kaup\_kood AS kaup\_kood, Kaup.nimetus AS nimetus, Kauba\_seisundi\_liik.nimetus AS seisund, Kaup.hind AS hind, Kaup.kirjeldus AS kirjeldus

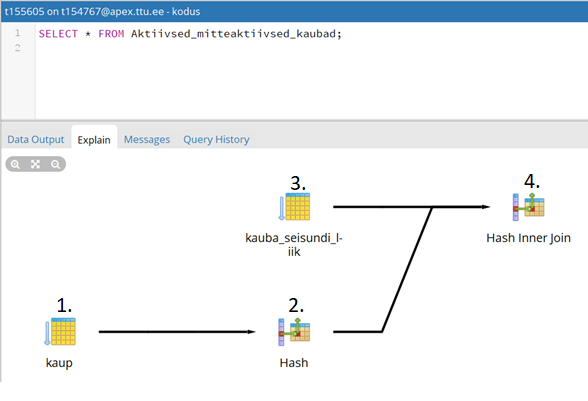
FROM Kauba\_seisundi\_liik INNER JOIN Kaup ON Kauba\_seisundi\_liik.kauba\_seisundi\_liik\_kood = Kaup.kauba\_seisundi\_liik\_kood

WHERE Kaup.kauba\_seisundi\_liik\_kood IN (2,3));

Enne täitmisplaani koostamist värskendasin andmebaasi statistikat, kasutades selleks ANALYZE lauset. Selle tegevuse eesmärgiks oli tagada, et andmebaasisüsteemil oleks täitmisplaani koostamiseks võimalikult täpne sisendinformatsioon. Täitmisplaani tuleb lugeda vastavalt joonisel 17 toodud numbritele alustades väikseimast numbrist. Täitmisplaani koostades oli tabelites vaid väga väike hulk testandmeid. Seega võib tabeleid pidada peaaegu tühjadeks, mis omakorda võis mõjutada plaani valikut. Näiteks eelistas antud täitmisplaanis andmebaasisüsteem tabelite läbiskaneerimist indeksite kasutamisele. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

1-2 – Tabeli *Kaup* läbiskaneerimisel leitakse vaid need read, kus *kauba\_seisundi\_liik\_kood* on 2 või 3 ehk kontrollitakse milline on väärtus veerus *kauba\_seisundi\_liik\_kood*. Tabelite *Kaup* ja *Kauba\_seisundi\_*liik ühendamise sisendiks lähevad vaid antud päringu tulemusel leitud read tabelist *Kaup.*

3-4 – Tabelid *Kaup* ja *Kauba\_seisundi\_liik* ühendatakse kasutades hash join algoritmi. Tabelist *Kaup* loetud *kauba\_seisundi\_liik\_kood* väärtuste abil leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema tabelist *Kauba\_seisundi\_liik* loetud *kauba\_seisundi\_liik\_kood* väärtuste põhjal leitud räsiväärtustega. Selle ühendamise kontekstis on *Kauba\_seisundi\_liik* väline tabel. Nii *kauba\_seisundi\_liik\_kood* väärtused tabelist *Kauba\_seisundi\_liik* kui ka *Kauba\_seisundi\_liik\_kood* väärtused tabelist *Kaup* leitakse tabelite täielikku läbiskaneerimist kasutades. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)



**Joonis 17 Visuaalne täitmisplaan PgAdmin’is.**

## Rollid ja kasutajad

Antud töös rolle loodud ei ole. Rakendusele on loodud allpool toodud kasutaja.

CREATE USER t155605\_juhataja WITH PASSWORD 'GHL5R8YSS';

## Üleliigsete õiguste äravõtmine

Avalikkusele vaikimisi antud õiguste äravõtmine.

REVOKE CONNECT, TEMP ON DATABASE t155605 FROM PUBLIC;

REVOKE CREATE, USAGE ON SCHEMA public FROM PUBLIC;

REVOKE USAGE ON LANGUAGE plpgsql FROM PUBLIC;

REVOKE EXECUTE ON FUNCTION

f\_kauba\_lopetamine (p\_kaup\_kood Kaup.kaup\_kood%TYPE),

f\_kauba\_mitteaktiveerimine (p\_kaup\_kood Kaup.kaup\_kood%TYPE),

f\_kauba\_kategooriasse\_lisamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE),

f\_kauba\_kategooriast\_eemaldamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE),

f\_kauba\_variandi\_lisamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE),

f\_kauba\_variandi\_eemaldamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE),

f\_on\_juhataja(p\_e\_meil text, p\_parool text),

f\_kauba\_lopetamine(),

f\_kauba\_aktiveerimine(),

f\_kauba\_mitteaktiveerimine(),

f\_kauba\_kategooriasse\_lisamine()

FROM PUBLIC;

Õigused eemaldatakse ka kõigi laiendustesse kuuluvate funktsioonide suhtes, mis antud projektis on skeemis public (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem).

REVOKE EXECUTE ON ALL FUNCTIONS IN SCHEMA public FROM PUBLIC;

## Õiguste jagamine

Järgnevate lausetega antakse õigused kasutajale t155605\_juhataja.

GRANT CONNECT ON DATABASE t155605 TO t155605\_juhataja;

GRANT USAGE ON SCHEMA public TO t155605\_juhataja;

GRANT SELECT ON

aktiivsed\_mitteaktiivsed\_kaubad,

kaupade\_detailid,

kaupade\_kategooriad,

kaupade\_koondaruanne,

kaupade\_variandid,

koik\_kaubad

TO t155605\_juhataja;

GRANT EXECUTE ON FUNCTION

f\_kauba\_lopetamine (p\_kaup\_kood Kaup.kaup\_kood%TYPE),

f\_on\_juhataja(p\_e\_meil text, p\_parool text)

TO t155605\_juhataja;

Järgnev lause ei mõjuta olemasolevaid funktsioone, kuid on mõeldud vaikimisi õiguste muutsmiseks tulevikus loodavate funktsioonide jaoks (Andmebaasid II õppematerjalid).

ALTER DEFAULT PRIVILEGES REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC;

## Andmebaasiobjektide kustutamine

Antud peatükis on toodud andmebaasiobjektide kustutamise laused.

### Õiguste äravõtmine

Kasutaja kustutamisel tuleb ära võtta ka kasutajale antud õigused.

REVOKE CONNECT ON DATABASE t155605 FROM t155605\_juhataja;

REVOKE SELECT ON

aktiivsed\_mitteaktiivsed\_kaubad,

kaupade\_detailid,

kaupade\_kategooriad,

kaupade\_koondaruanne,

kaupade\_variandid,

koik\_kaubad

FROM t155605\_juhataja;

REVOKE EXECUTE ON FUNCTION

f\_kauba\_lopetamine (p\_kaup\_kood Kaup.kaup\_kood%TYPE),

f\_on\_juhataja(p\_e\_meil text, p\_parool text)

FROM t155605\_juhataja;

### Domeenide kustutamine

DROP DOMAIN IF EXISTS d\_nimetus CASCADE;

DROP DOMAIN IF EXISTS d\_kirjeldus CASCADE;

DROP DOMAIN IF EXISTS d\_reg\_aeg CASCADE;

### Tabelite ja arvujada generaatorite kustutamine

ALTER TABLE IF EXISTS Kaup\_kategooria DROP CONSTRAINT IF EXISTS FK\_Kaup\_kategooria\_on\_seotud\_Kauba\_kategooria\_tyyp;

ALTER TABLE IF EXISTS Isik DROP CONSTRAINT IF EXISTS FK\_Isik\_on\_seotud\_Isiku\_seisundi\_liik;

ALTER TABLE IF EXISTS Isik DROP CONSTRAINT IF EXISTS FK\_Isik\_on\_seotud\_Riik;

ALTER TABLE IF EXISTS Klient DROP CONSTRAINT IF EXISTS FK\_Klient\_on\_seotud\_Kliendi\_seisundi\_liik;

ALTER TABLE IF EXISTS Klient DROP CONSTRAINT IF EXISTS FK\_Klient\_on\_seotud\_Isik;

ALTER TABLE IF EXISTS Tootaja DROP CONSTRAINT IF EXISTS FK\_Tootaja\_on\_seotud\_Tootaja\_seisundi\_liik;

ALTER TABLE IF EXISTS Tootaja DROP CONSTRAINT IF EXISTS FK\_Tootaja\_on\_seotud\_Isik;

ALTER TABLE IF EXISTS Tootaja DROP CONSTRAINT IF EXISTS FK\_Tootaja\_on\_seotud\_Amet;

ALTER TABLE IF EXISTS Kaup DROP CONSTRAINT IF EXISTS FK\_Kaup\_on\_seotud\_Brand;

ALTER TABLE IF EXISTS Kaup DROP CONSTRAINT IF EXISTS FK\_Kaup\_on\_seotud\_Tootaja;

ALTER TABLE IF EXISTS Kaup DROP CONSTRAINT IF EXISTS FK\_Kaup\_on\_seotud\_Kauba\_seisundi\_liik;

ALTER TABLE IF EXISTS Kaubale\_kategooria\_omamine DROP CONSTRAINT IF EXISTS FK\_Kaubale\_kategooria\_omamine\_on\_seotud\_Kaup;

ALTER TABLE IF EXISTS Kaubale\_kategooria\_omamine DROP CONSTRAINT IF EXISTS FK\_Kaubale\_kategooria\_omamine\_on\_seotud\_Kaup\_kategooria;

ALTER TABLE IF EXISTS Kaup\_variant DROP CONSTRAINT IF EXISTS FK\_Kauba\_variant\_on\_seotud\_Kaup;

ALTER TABLE IF EXISTS Kaup\_variant DROP CONSTRAINT IF EXISTS FK\_Kauba\_variant\_on\_seotud\_Varv;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_Kaup;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_eesmine\_Kaamera;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_tagumine\_Kaamera;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_Sisemalu;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_Diagonaal;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_Protsessor;

ALTER TABLE IF EXISTS Nutitelefon DROP CONSTRAINT IF EXISTS FK\_Nutitelefon\_on\_seotud\_Ekraan\_resolutsioon;

DROP TABLE IF EXISTS Kauba\_kategooria\_tyyp CASCADE;

DROP TABLE IF EXISTS Kaup\_kategooria CASCADE;

DROP TABLE IF EXISTS Isiku\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Kauba\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Kliendi\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Tootaja\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Brand CASCADE;

DROP TABLE IF EXISTS Diagonaal CASCADE;

DROP TABLE IF EXISTS Ekraan\_resolutsioon CASCADE;

DROP TABLE IF EXISTS Kaamera CASCADE;

DROP TABLE IF EXISTS Protsessor CASCADE;

DROP TABLE IF EXISTS Sisemalu CASCADE;

DROP TABLE IF EXISTS Varv CASCADE;

DROP TABLE IF EXISTS Amet CASCADE;

DROP TABLE IF EXISTS Riik CASCADE;

DROP TABLE IF EXISTS Isik CASCADE;

DROP TABLE IF EXISTS Klient CASCADE;

DROP TABLE IF EXISTS Tootaja CASCADE;

DROP TABLE IF EXISTS Kaup CASCADE;

DROP TABLE IF EXISTS Kaubale\_kategooria\_omamine CASCADE;

DROP TABLE IF EXISTS Kaup\_variant CASCADE;

DROP TABLE IF EXISTS Nutitelefon CASCADE;

DROP FOREIGN TABLE IF EXISTS Isik\_jsonb CASCADE;

DROP FOREIGN TABLE IF EXISTS Riik\_jsonb CASCADE;

DROP USER MAPPING FOR t155605 SERVER minu\_testandmete\_server\_apex;

DROP SERVER IF EXISTS minu\_testandmete\_server\_apex CASCADE;

Andmebaasisüsteem kustutas tabelite kustutamisel automaatselt ka antud tabelitega seotud arvujada generaatorid, sest kõnealused arvujada generaatorid olid loodud kasutades SERIAL tüüpi. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

### Vaadete kustutamine

DROP VIEW IF EXISTS Koik\_kaubad CASCADE;

DROP VIEW IF EXISTS Kaupade\_detailid CASCADE;

DROP VIEW IF EXISTS Kaupade\_variandid CASCADE;

DROP VIEW IF EXISTS Kaupade\_kategooriad CASCADE;

DROP VIEW IF EXISTS Aktiivsed\_mitteaktiivsed\_kaubad CASCADE;

DROP VIEW IF EXISTS Kaupade\_koondaruanne CASCADE;

DROP VIEW IF EXISTS Kaupade\_kategooriad\_ja\_kategooria\_tyybid CASCADE;

DROP VIEW IF EXISTS Kaubad\_ja\_registreerijad\_json CASCADE;

### Indeksite kustutamine

DROP INDEX IF EXISTS IXFK\_Kauba\_kategooria\_tyyp;

DROP INDEX IF EXISTS IXFK\_Isiku\_seisundi\_liik;

DROP INDEX IF EXISTS IXFK\_Riik;

DROP INDEX IF EXISTS IXFK\_Kliendi\_seisundi\_liik;

DROP INDEX IF EXISTS IXFK\_Amet;

DROP INDEX IF EXISTS IXFK\_Tootaja\_seisundi\_liik;

DROP INDEX IF EXISTS IXFK\_Brand;

DROP INDEX IF EXISTS IXFK\_Kauba\_seisundi\_liik;

DROP INDEX IF EXISTS IXFK\_Tootaja;

DROP INDEX IF EXISTS IXFK\_Kaup\_kategooria;

DROP INDEX IF EXISTS IXFK\_Varv;

DROP INDEX IF EXISTS IXFK\_Diagonaal;

DROP INDEX IF EXISTS IXFK\_Eesmine\_kaamera;

DROP INDEX IF EXISTS IXFK\_Erkaan\_resolutsioon;

DROP INDEX IF EXISTS IXFK\_Protsessor;

DROP INDEX IF EXISTS IXFK\_Sisemalu;

DROP INDEX IF EXISTS IXFK\_Tagumine\_kaamera;

DROP INDEX IF EXISTS AK\_Isik\_e\_meil;

### Funktsioonide ja trigerite kustutamine

DROP FUNCTION IF EXISTS f\_kauba\_lopetamine (p\_kaup\_kood Kaup.kaup\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_mitteaktiveerimine (p\_kaup\_kood Kaup.kaup\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_kategooriasse\_lisamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_kategooriast\_eemaldamine (p\_kaup\_kood Kaubale\_kategooria\_omamine.kaup\_kood%TYPE, p\_kauba\_kategooria\_kood Kaubale\_kategooria\_omamine.kauba\_kategooria\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_variandi\_lisamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_variandi\_eemaldamine (p\_kaup\_kood Kaup\_variant.kaup\_kood%TYPE, p\_kauba\_varv\_kood Kaup\_variant.kauba\_varv\_kood%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_on\_juhataja(p\_e\_meil text, p\_parool text) CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_lopetamine() CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_aktiveerimine() CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_mitteaktiveerimine() CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_kategooriasse\_lisamine() CASCADE;

DROP FUNCTION IF EXISTS f\_kauba\_kategooriasse\_kuulumine() CASCADE;

DROP TRIGGER IF EXISTS trig\_kauba\_lopetamine ON Kaup CASCADE;

DROP TRIGGER IF EXISTS trig\_kauba\_aktiveerimine ON Kaup CASCADE;

DROP TRIGGER IF EXISTS trig\_kauba\_mitteaktiveerimine ON Kaup CASCADE;

DROP TRIGGER IF EXISTS trig\_kauba\_kategooriasse\_lisamine ON Kaubale\_kategooria\_omamine CASCADE;

DROP TRIGGER IF EXISTS trig\_kauba\_kategooriasse\_kuulumine ON Kaup CASCADE;

### Reeglite kustutamine

Antud projektis reegleid ei kasutatud. Seega pole esitatud ka reeglite kustutamise lauseid. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

### Kasutajate ja rollide kustutamine

Antud projektis rolle ei kasutatud. Seega pole esitatud ka rollide kustutamise lauseid. (Ülikooli infosüsteemi vastuvõtuaegade allsüsteem)

DROP USER IF EXISTS t155605\_juhataja;

### Laienduste kustutamine

DROP EXTENSION IF EXISTS pgcrypto CASCADE;

DROP EXTENSION IF EXISTS postgres\_fdw CASCADE;

## Kasutatud materjalid

1. AKIT. Andmekaitse ja infoturbe seletussõnastik. [WWW] <http://akit.cyber.ee/> (29.01.2017)
2. Andmebaasid I õppematerjalid. [WWW] <http://maurus.ttu.ee>/aine\_index.php?aine=359 (16.12.2017)
3. Andmebaasid II õppematerjalid. [WWW] <http://maurus.ttu.ee>/aine\_index.php?aine=366 (16.12.2017)
4. Andmebaaside lisamaterjalid . [WWW] <http://maurus.ttu.ee/aine_index.php?aine=346> (16.12.2017)
5. Andmebaaside projekti tegemise mall. [WWW] <http://maurus.ttu.ee> (29.01.2017)
6. Country Codes - ISO 3166 [WWW] <http://www.iso.org/iso/home/standards/country_codes.htm> (29.01.2017)
7. Chisholm, M. (2000). *Managing Reference Data in Enterprise Databases:* Binding *Corporate Data to the Wider World.* Morgan Kaufmann.
8. Eesti keele seletav sõnaraamat. [WWW] <https://www.eki.ee/dict/ekss/> (30.03.2017)
9. Eesti Statistika. Riikide ja territooriumide klassifikaator 2013v1. [WWW] <http://metaweb.stat.ee/view_xml_multi_code.htm?id=3477719&siteLanguage=ee> (29.01.2017)
10. ESTERM [WWW] <http://termin.eki.ee/esterm/> (29.01.2017)
11. Isikuandmete kaitse seadus. [WWW] <https://www.riigiteataja.ee/akt/130122010011?leiaKehtiv> (29.01.2017)
12. Infosüsteemide turvameetmete süsteem. Vabariigi Valitsuse 20.12 2007. a määrus nr 252. Elektrooniline Riigi Teataja.  
    [WWW] <https://www.riigiteataja.ee/akt/13125331?leiaKehtiv> (29.01.2017)
13. Vikipeedia. Riik. [WWW] <https://et.wikipedia.org/wiki/Riik> (29.01.2017)
14. Vikipeedia. Kaup. [WWW] <https://et.wikipedia.org/wiki/Kaup> (26.04.2017)
15. Vikipeedia. Bränd. [WWW] <https://et.wikipedia.org/wiki/Br%C3%A4nd> (26.04.2017)
16. Ülikooli infosüsteemi vastuvõtuaegade allsüsteem. [WWW] <http://maurus.ttu.ee/ained/IDU0230_2017/doc/2/Naidisprojekt_IDU0230_vastuvotuajad_ver5_15.pdf> (16.12.2017)