TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Autorendi ettevõtte infosüsteemi autode funktsionaalne allsüsteem

Andmebaasid II, ITI0207

|  |  |
| --- | --- |
| Üliõpilane: | Richardas Keršis |
| Õpperühm: | IABM13 |
| Matrikli nr: | 192406IABM |
| e-posti aadress: | richardas.kersis@gmail.com |
|  |  |
| Üliõpilane: | Melvin Kriisa |
| Õpperühm: | TAI |
| Matrikli nr: | 195716 |
| e-posti aadress: | melkri@ttu.ee |
|  |  |

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

Tallinn

2019

**Sisukord**

[1 Strateegiline analüüs 8](#_Toc27073252)

[1.1 Terviksüsteemi üldvaade 8](#_Toc27073253)

[1.1.1 Organisatsiooni eesmärgid 8](#_Toc27073254)

[1.1.2 Infosüsteemi eesmärgid 8](#_Toc27073255)

[1.1.3 Lausendid 9](#_Toc27073256)

[1.1.4 Põhiobjektid 10](#_Toc27073257)

[1.1.5 Põhiprotsessid 11](#_Toc27073258)

[1.1.6 Põhilised sündmused 12](#_Toc27073259)

[1.1.7 Tegutsejad 13](#_Toc27073260)

[1.1.8 Asukohad 13](#_Toc27073261)

[1.1.9 Terviksüsteemi tükeldus allsüsteemideks 14](#_Toc27073262)

[1.2 Autode funktsionaalse allsüsteemi eskiismudelid 15](#_Toc27073263)

[1.2.1 Eesmärgid 15](#_Toc27073264)

[1.2.2 Allsüsteemi kasutavad pädevusalad 15](#_Toc27073265)

[1.2.3 Allsüsteemi poolt vajatavad registrid 15](#_Toc27073266)

[1.2.4 Allsüsteemi ühe põhiprotsessi tegevusdiagramm 17](#_Toc27073267)

[1.2.5 Allsüsteemi kasutusjuhtude eskiismudel 17](#_Toc27073268)

[1.2.6 Mittefunktsionaalsed nõuded 20](#_Toc27073269)

[1.3 Autode registri eskiismudelid 22](#_Toc27073270)

[1.3.1 Eesmärgid 23](#_Toc27073271)

[1.3.2 Registrit kasutavad pädevusalad 23](#_Toc27073272)

[1.3.3 Registrit teenindavad funktsionaalsed allsüsteemid 23](#_Toc27073273)

[1.3.4 Infovajadused, mida register aitab rahuldada 23](#_Toc27073274)

[1.3.5 Seosed teiste registritega 23](#_Toc27073275)

[1.3.6 Ärireeglid 24](#_Toc27073276)

[1.3.7 Registri kontseptuaalne eskiismudel 24](#_Toc27073277)

[2 Detailanalüüs 26](#_Toc27073278)

[2.1 Autode funktsionaalse allsüsteemi detailanalüüs 26](#_Toc27073279)

[2.1.1 Kasutusjuhtude mudel 26](#_Toc27073280)

[2.2 Autode funktsionaalse allsüsteemi vajatavate registrite detailanalüüs 34](#_Toc27073281)

[2.2.1 Kontseptuaalne andmemudel 34](#_Toc27073282)

[2.2.2 Andmebaasioperatsioonide lepingud 45](#_Toc27073283)

[2.2.3 Registri põhiobjekti seisundidiagramm 47](#_Toc27073284)

[2.3 CRUD maatriks 49](#_Toc27073285)

[3 Füüsiline disain 50](#_Toc27073286)

[3.1 Autode funktsionaalse allsüsteemi vajatavate registrite füüsiline disain 50](#_Toc27073287)

[4 Realisatsioon PostgreSQLis 55](#_Toc27073288)

[4.1 Andmebaasi loomine 55](#_Toc27073289)

[4.2 Skeemid 55](#_Toc27073290)

[4.3 Domeenid 55](#_Toc27073291)

[4.4 Tabelid ja arvujada generaatorid 55](#_Toc27073292)

[4.5 Vaated 61](#_Toc27073293)

[4.6 Protseduursed keeled 63](#_Toc27073294)

[4.7 Trigeri funktsioonid ja trigerid 63](#_Toc27073295)

[4.8 Reeglid 64](#_Toc27073296)

[4.9 Rutiinid 64](#_Toc27073297)

[4.10 Indeksid 65](#_Toc27073298)

[4.10.1 Välisvõtmete veergudele lisatavad indeksid 66](#_Toc27073299)

[4.10.2 Täiendavad sekundaarsed indeksid 66](#_Toc27073300)

[4.10.3 Funktsioonil põhinevad indeksid 66](#_Toc27073301)

[4.11 Klassifikaatorite väärtustamise SQL laused 66](#_Toc27073302)

[4.12 JSON formaadis lähteandmete laadimine 68](#_Toc27073303)

[4.13 Täiendavate testandmete lisamine 69](#_Toc27073304)

[4.14 Andmebaasi statistika kogumine 72](#_Toc27073305)

[4.15 Päringu täitmisplaani näide 72](#_Toc27073306)

[4.16 Rollid ja kasutajad 72](#_Toc27073307)

[4.17 Üleliigsete õiguste äravõtmine 72](#_Toc27073308)

[4.18 Õiguste jagamine 73](#_Toc27073309)

[4.19 Andmebaasiobjektide kustutamine 73](#_Toc27073310)

[4.19.1 Õiguste äravõtmine 73](#_Toc27073311)

[4.19.2 Domeenide kustutamine 74](#_Toc27073312)

[4.19.3 Tabelite ja arvujada generaatorite kustutamine 74](#_Toc27073313)

[4.19.4 Vaadete kustutamine 75](#_Toc27073314)

[4.19.5 Indeksite kustutamine 75](#_Toc27073315)

[4.19.6 Rutiinide kustutamine 76](#_Toc27073316)

[4.19.7 Trigerite kustutamine 76](#_Toc27073317)

[4.19.8 Reeglite kustutamine 76](#_Toc27073318)

[4.19.9 Kasutajate ja rollide kustutamine 76](#_Toc27073319)

[4.19.10 Laienduste kustutamine 76](#_Toc27073320)

[5 Kasutatud materjalid 77](#_Toc27073321)

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.

Richardas Keršis

Melvin Kriisa

**Sissejuhatus (Andmebaasid II)**

Tabel 1 esitab ülevaate andmebaasi ja rakenduse realisatsioonist.

Tabel 1 Sissejuhatus õppeaine "Andmebaasid II" töö tulemuse kohta.

|  |  |
| --- | --- |
| Kasutatav andmebaasisüsteem ja versioon | PostgreSQL 12 |
| Rakenduse loomise vahendid (arenduskeskkonnad, keeled ja nende versioonid) | pgApex |
| Andmebaasi server **(andmebaas PEAB olema apex.ttu.ee serveris)** | apex.ttu.ee |
| PostgreSQL andmebaasi nimi (kui kasutate PostgreSQLi) | t192406 |
| 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)** | <http://apex.ttu.ee/pgapex2/public/index.php/app/29/92> |
| Rakenduse nimi (kui rakendus on tehtud pgApex või Oracle APEX keskkonnas) | a192406\_Autorent |
| Kui tegemist on kahekihilise klient-server süsteemiga ja rakendus on kasutaja arvutis, siis mida peab kasutaja arvutis tegema, et seda rakendust kasutada |  |
| Kasutajanimi ja parool rakendusse sisselogimiseks | kasutajanimi: cole.nichols@ezent.biz  parool: dolore |
| 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 *Lõpeta autosid* pöördutakse nupule *Lõpeta* vajutamise järel kasutaja-defineeritud funktsiooni *f\_lopeta\_auto* poole. Funktsioon on mõeldud auto lõpetamiseks. |
| 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 *Kõik autod* esitatavad andmed leitakse vaatest a*utode\_detailid.*  Ekraanivormil *Autode koondaruanne* esitatavad andmed leitakse vaatest *autode\_koondaruanne.*  Ekraanivormil *Lõpeta autosid* esitatavad andmed leitakse vaatest *aktiivsed\_ja\_mitteaktiivsed\_autod.*  Ekraanivormil *Auto detailid* esitatavad andmed leitakse vaatest *autode\_detailid.* |
| 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. | Andmebaasisüssteemiga ühenduse loomiseks kasutatav kasutaja nimi (t192406\_Autorendi\_juhataja) |
| Kas rakendus suhtleb andmebaasisüsteemiga kui üks kasutaja või vastab igale lõppkasutajale eraldi andmebaasi kasutaja? | Rakendus suhtleb andmebaasiga kui üks kasutaja |
| Kas lõppkasutajate kasutajanimed ja paroolid on andmebaasis (kui jah, siis mis tabeli mis veergudes)? | Jah (tabeli Isik veerud e\_meil ja parool) |
| 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 en\_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; |
| Kuidas toimub kasutaja autentimine? | Kontrollimaks kasutaja identiteeti võrreldakse kasutaja etteantud parooli põhjal genereeritud räsiväärtust andmebaasis salvestatud parooli räsiväärtusega. Lisaks kontrollitakse, et kasutaja töötaks ametikohal, mille esindajana ta soovib sisse logida ning oleks seisundis, mis lubab süsteemi kasutada. Kui need väärtused on võrdsed ja nimetatud lisatingimused on täidetud, siis on kasutaja tuvastatud ja ta võib hakata süsteemi kasutama. 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 õppejõud, luuakse andmebaasis funktsioon f\_autendi\_juhataja(text, text). Funktsiooni väljakutsel on esimene argument kasutaja meiliaadress(e\_meil) ja teine argument parool. TSITEERIDA NÄITEPROJEKT |
| Kas olete oma töö iseseisva töö hindamismudeli alusel enne esitamist ise üle kontrollinud? | Jah |
| Kui vastasite eelmises punktis „Jah“, siis milline oli punktisumma? | 60 |

# 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 autorendi ettevõtte infosüsteemi 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 võimalikult laia valikut erinevat liiki sõidukeid
* Hoida ettevõtte autopark tehniliselt heas korras ja kaasaegne
* Pakkuda ettevõtte töötajatele positiivset ja tunnustust pakkuvat sisekliimat
* Pakkuda teenust kliendile just seal, kus ta seda kõige enam vajab

### 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 autodest, millega tehingute (transaktsioonide) tegemine on üks organisatsiooni põhieesmärk
* Saada ülevaade organisatsiooni käsutuses olevatest varadest
* Võimaldada organisatsioonil luua vara tarnetellimusi
* Saada ülevaade organisatsiooni töötajate töögraafikust
* Saada ülevaade vabadest autodest
* Saada ülevaade kasutusel olevatest autodest
* Saada ülevaade organisatsiooniga seotud lepingupartneritest
* Võimaldada hallata hinnakirja
* Saada ülevaade autode rikete ajaloo kohta
* Võimaldada autosid rutiinselt hooldusse saata
* Saada ülevaade autode kindlustuslepingutest
* Saada ülevaade autode tehnilise ülevaatusese seisundist
* Võimaldada klientidel autosid rentida
* Võimaldada rentimist tühistada
* Võimaldada jälgida autode tagastamisi
* Võimaldada pakkuda kliendile sobivat lisavarustust
* Saada ülevaade kliendi minevikust
* Võimaldada esitada kahjunõudeid
* Võimaldada esitada arveid
* Võimaldada kliendil jätta tagasiside oma rendikogemuse kohta
* Võimaldada klientidele pakkuda soodustusi
* Saada ülevaade organisatsioonile tehtud ettekirjutustest

### 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
* Juhataja on töötaja
* Autode haldur on töötaja
* Klient on isik
* Kliendil on hetkeseisund
* Uudistaja on süsteemi tuvastamata kasutaja
* Kliendi seisundi liik on klassifikaator
* Juhataja on organisatsiooni omanik
* Autode haldur haldab autosid
* Klassifikaatorite haldur haldab klassifikaatoreid
* Klienditeenindaja on töötaja
* Autojuht on töötaja
* Raamatupidaja on lepingupartner
* Klient sõlmib lepingu
* Lepingul on hetkeseisund
* Lepingu seisundi liik on klassifikaator
* Töötaja registreerib auto
* Autot iseloomustab null või rohkem kategooriat
* Auto kategooria on klassifikaator
* Autol on hetkeseisund
* Auto seisundi liik on klassifikaator
* Organisatsioonil on hetke seisund
* Organisatsiooni seisundi liik on klassifikaator
* Organisatsioon võib olla klient
* Töötajal on töögraafik
* Töögraafikul on hetkeseisund
* Töögraafiku hetkeseisund on klassifikaator
* Auto on hinnastatud hinnakirja alusel
* Hinnakirjal on hetkeseisund
* Hinnakirja seisundi liik on klassifikaator
* Lisavarustusel on hetkeseisund
* Lisavarustuse seisundi liik on klassifikaator
* Rentimisel on hetkeseisund
* Rentimise seisundi liik on klassifikaator
* Soodustusel on hetkeseisund
* Soodustuse seisundi liik on klassifikaator
* Kindlustusleping on iga auto kohta eraldi
* Kindlustuslepingul on hetkeseisund
* Kindlustuslepingu seisundi liik on klassifikaator
* Ülevaatus on iga auto kohta eraldi
* Ülevaatusel on hetkeseisund
* Ülevaatuse seisundi liik on klassifikaator
* Arvel on hetkeseisund
* Arve seisundi liik on klassifikaator
* Kahjunõudel on hetkeseisund
* Kahjunõude seisundi liik on klassifikaator
* Inventuuril on hetkeseisund
* Inventuuri seisundi liik on klassifikaator
* Rikkel on hetkeseisund
* Rikke seisundi liik on klassifikaator
* Hooldusel on hetkeseisund
* Hoolduse seisundi liik on klassifikaator
* Vara tarnetellimusel on hetkeseisund
* Vara tarnetellimuse seisundi liik on klassifikaator
* Lepingupartner on teine organisatsioon
* Lepingupartneril on hetkeseisund
* Lepingupartneri seisundi liik on klassifikaator
* Lepingupartner sõlmib lepingu
* Tagasisidel on hetkeseisund
* Tagasiside seisundi liik on klassifikaator
* Organisatsioonile tehakse ettekirjutus
* Ettekirjutusel on hetkeseisund
* Ettekirjutuse seisundi liik on klassifikaator
* Ettekirjutuse koostab andmekaitse inspektsioon

### Põhiobjektid

* Isik
* Organisatsioon
* Töötaja
* Klient
* Klassifikaator
* Töögraafik
* Leping
* Auto
* Hinnakiri
* Lisavarustus
* Rentimine
* Soodustus
* Kindlustusleping
* Ülevaatus
* Arve
* Kahjunõue
* Inventuur
* Rike
* Hooldus
* Vara tarnetellimus
* Lepingupartner
* Tagasiside
* Ettekirjutus

### Põhiprotsessid

* Isiku registreerimine
* Isiku surnuks märkimine
* Töötaja tööle võtmine
* Töötaja ametikoha muutmine
* Töötaja ajutiselt töölt vabastamine
* Töötaja puhkusele siirdumine
* Klassifikaatori väärtuse lisamine
* Klassifikaatori väärtuse muutmine
* Lepingu sõlmimine
* Lepingu peatamine
* Lepingu ühepoolne katkestamine
* Lepingu pikendamine
* Auto registreerimine
* Auto unustamine
* Auto aktiveerimine
* Auto ajutiselt kasutusest eemaldamine (mitteaktiivseks muutmine)
* Auto lõplikult kasutusest eemaldamine (lõpetamine)
* Töögraafiku määramine töötajale
* Hinnakirjas muudatuste tegemine
* Lisavarustuse pakkumine
* Sõiduki rentimine kliendile
* Soodustuse pakkumine kliendile
* Auto kindlustuslepingu olemasolu kontrollimine
* Auto ülevaatuse olemasolu kontrollimine
* Arve esitamine rentimise eest
* Arve makseseisundi kontrollimine
* Kahjunõude esitamine
* Inventuuri tegemine
* Rikke talletamine auto ajalukku
* Auto hooldusele saatmine
* Auto tehnilise ülevaatuse seisundi uuendamine
* Auto tehnilisse ülevaatusse saatmine
* Uue lepingupartneriga lepingu sõlmimine
* Tagasiside saamine
* Tagasiside põhjal otsuste langetamine
* Ettekirjutuse saamine

### 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
* 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
* 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 auto kohta
* Selgus, et organisatsiooni jõudnud teave auto kohta on enneaegne ning sellisel kujul autot ei ole vaja registreerida
* On vaja muuta võimalikuks auto kasutamine tehingutes
* Auto kasutamine tehingutes on vaja ajutiselt peatada, kuna seoses autoga on ilmnenud ajutise iseloomuga probleemid
* Auto kasutamine tehingutes on vaja lõpetada, kuna seoses autoga on ilmnenud püsiva iseloomuga probleemid või kuna auto on oma aja lihtsalt ära elanud
* Töötajatele on tarvis luua järgneva kuu töögraafik
* Turul on konkurentsitingimused muutunud ja vaja hinnakirja kaasajastada
* Klient soovib autoga kaasa saada lisavarustust
* Auto antakse üle kliendi käsutusse
* Kliendile pakutakse auto rentimist soodustingimuste alusel
* Veendumaks, et sõidukitel on kehtiv liikluskindlustus tehakse regulaarset kontrolli kindlustuse seisundi üle
* Veendumaks, et sõidukitel on kehtiv ülevaatus tehakse regulaarset kontrolli ülevaatuse seisundi üle
* Rendilepingu sõlmimise järel kliendiga esitatakse kliendile arve müüdavate teenuste eest
* Enne kliendile auto üle andmist on vaja veenduda, et klient on rentimise eest arve tasunud
* Klient on rikkunud ettevõttele kuuluvat vara ning peab hüvitama tekitatud kahjud
* Organisatsiooni varade üle arve pidamiseks on tarvis regulaarselt inventuuri teostada
* Autoga juhtus õnnetus ning rike on vaja talletada auto ajalukku
* Hoidmaks ettevõtte pakutavate teenuste kvaliteeti kõrgel tuleb rutiinselt autosid hooldada
* Auto on läbinud tehnilise ülevaatuse ning tohib taas teedel sõita
* Auto tehniline ülevaatus on aegumas ning sõidukiga tehingute jätkamiseks tuleb autol sooritada tehniline ülevaatus
* Ettevõtte on tellinud uue sõiduki, millega tehinguid teha, ning andmed sõiduki ostu kohta tuleb talletada süsteemi
* Ettevõte on alustanud koostööd uue organisatsiooniga ning andmed partnerluse kohta tuleb talletada süsteemi
* Rentnik annab kasutatud teenuste kohta tagasisidet ning seda tuleb süsteemis hoida edasiselt paremate otsuste langetamiseks
* Süsteemi on talletatud tagasiside ning selle põhjal tehakse otsus tulevikus parema teenuse osutamiseks
* Andmekaitse inspektsioon teeb ettevõttele ettekirjutuse ning sellele on vaja reageerida

### Tegutsejad

* Juhataja (ka omanik)
* Autode haldur
* Klassifikaatorite haldur
* Klient
* Uudistaja
* Autojuht
* Klienditeenindaja
* Raamatupidaja

### 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
* Autode haldur
* Klassifikaatorite haldur
* Autojuht
* Klienditeenindaja

Organisatsiooni välised pädevusalad.

* Klient
* Uudistaja
* Raamatupidaja

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

Tabel 2 Sisulised allsüsteemid.

|  |  |
| --- | --- |
| Funktsionaalne allsüsteem | Register, mida see funktsionaalne allsüsteem teenindab |
| Klientide funktsionaalne allsüsteem | Klientide register |
| Autode funktsionaalne allsüsteem | Autode register |
| Hinnakirja funktsionaalne allsüsteem | Hinnakirja register |
| Lisavarustuse funktsionaalne allsüsteem | Lisavarustuse register |
| Rentimise funktsionaalne allsüsteem | Rentimiste register |
| Soodustuste funktsionaalne allsüsteem | Soodustuste register |
| Kahjunõuete funktsionaalne allsüsteem | Kahjunõuete register |
| Rikete funktsionaalne allsüsteem | Rikete register |
| Tagasiside funktsionaalne allsüsteem | Tagasiside register |

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

Tabel 3 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 |
| Organisatsioonide funktsionaalne allsüsteem | Organisatsioonide register |
| Töögraafiku funktsionaalne allsüsteem | Töögraafiku register |
| Kindlustuste funktsionaalne allsüsteem | Kindlustuste register |
| Ülevaatuste funktsionaalne allsüsteem | Ülevaatuste register |
| Inventuuri funktsionaalne allsüsteem | Inventuuri register |
| Hoolduse funktsionaalne allsüsteem | Hoolduste register |
| Vara tarnetellimuste funktsionaalne allsüsteem | Vara tarnetellimuste register |
| Lepingupartnerite funktsionaalne allsüsteem | Lepingupartnerite register |
| Arvete funktsionaalne allsüsteem | Arvete register |
| Ettekirjutuste funktsionaalne allsüsteem | Ettekirjutuste register |

## Autode funktsionaalse allsüsteemi eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

* Muuta võimalikuks auto kasutamine erinevates tehingutes (transaktsioonides), mille läbiviimist infosüsteem toetab
* Võimaldada auto elektrooniliselt registreerida
* Võimaldada määrata auto hetkeseisundit vastavalt elutsüklile
* Võimaldada muuta süsteemile teadaolevaid andmeid auto kohta
* Võimalik auto andmed kustutada e infosüsteemi mõttes unustada, kuid teha seda ainult siis, kui auto pole veel kordagi aktiivsesse kasutusse läinud
* Võimaldada vastata fikseeritud päringutele auto kohta
* Võimaldada vaadata määratud elutsüklis olevate autode andmeid
* Võimaldada auto kategoriseerimist klassifikaatorite abiga
* Võimaldada saada ülevaade kogu autopargi kohta

### Allsüsteemi kasutavad pädevusalad

* Juhataja
* Autode haldur
* Uudistaja
* Klient
* Klassifikaatorite haldur
* Autojuht
* Klienditeenindaja

### Allsüsteemi poolt vajatavad registrid

Allsüsteem teenindab autode registrit.

Allsüsteem loeb.

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

### Allsüsteemi ühe põhiprotsessi tegevusdiagramm

Joonis 1 esitab auto lõpetamise protsessi kirjelduse tegevusdiagrammina.



Joonis 1 Auto lõpetamise tegevusdiagramm.

### Allsüsteemi kasutusjuhtude eskiismudel

esitatud kasutusjuhtude diagrammil on värvidel järgmine tähendus.

* Sinisega 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 2 Auto funktsionaalse allsüsteemi kasutusjuhtude diagramm.

**Kasutusjuht**: Tuvasta kasutaja

**Tegutsejad**: Autode haldur, Klienditeenindaja, Juhataja, Klient – (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 auto

**Tegutsejad**: Autode haldur

**Kirjeldus**: Autode haldur registreerib uue auto.

**Kasutusjuht**: Unusta auto

**Tegutsejad**: Autode haldur

**Kirjeldus**: Autode haldur vaatab ootel autode nimekirja, valib sealt auto ja kustutab selle andmebaasist. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Muuda auto andmeid

**Tegutsejad**: Autode haldur

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

**Kasutusjuht**: Aktiveeri auto

**Tegutsejad**: Autode haldur

**Kirjeldus**: Autode haldur vaatab ootel või mitteaktiivsete autode nimekirja, valib sealt auto ja muudab selle aktiivseks.

**Kasutusjuht**: Muuda auto mitteaktiivseks

**Tegutsejad**: Autode haldur

**Kirjeldus**: Autode haldur vaatab aktiivsete autode nimekirja, valib sealt auto ja muudab selle mitteaktiivseks. Subjekt saab nimekirja sorteerida ja filtreerida.

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

**Tegutsejad**: Autode haldur

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

**Kasutusjuht**: Vaata kõiki autosid

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

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

**Kasutusjuht**: Lõpeta auto

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata autode koondaruannet

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata aktiivseid autosid

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

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

### Mittefunktsionaalsed nõuded

Tabel 4 esitab vaadeldava allsüsteemi mittefunktsionaalsed nõuded.

Tabel 4 Allsüsteemi mittefunktsionaalsed nõuded.

| Tüüp | Nõude kirjeldus |
| --- | --- |
| andmebaasi-süsteem | Süsteem peab andmete hoidmiseks kasutama SQLandmebaasisüsteemi abil loodud andmebaasi. Tegemist on äritarkvaraga, mis kasutab tööks struktureeritud andmeid ning neid andmeid ei hakka olema väga palju (räägime maksimaalselt mõnest tuhandest reast). Autodega 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  (<https://www.rever.eu/en/db-main/>). Prototüübi koostamiseks kasutatakse töölaua andmebaasisüsteemi MS Access või LibreOffice Base, kuhu on integreeritud kasutajaliidese ehitamise vahendid.  Töötavas süsteemis peab klientidele ja uudistajatele mõeldud rakendus 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. |
| 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 | 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:SS * Ajatemplid tuleb esitada formaadis DD.MM.YYYY HH24:MI:SS * 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*.   Autode funktsionaalne allsüsteem teenindab auto registrit, mille turvaklass on (<https://www.riigiteataja.ee/akt/13125331?leiaKehtiv>):  K3T1S1  **K3** – töökindlus – 99,9% (lubatud summaarne seisak nädalas ~ 10 minutit); lubatav nõutava reaktsiooniaja kasv tippkoormusel – sekundid (1÷10);  **T1** – info allikas, selle muutmise ja hävitamise fakt peavad olema tuvastatavad; info õigsuse, täielikkuse ja ajakohasuse kontroll erijuhtudel ja vastavalt vajadusele;  **S1** – info asutusesiseseks kasutamiseks: juurdepääs teabele on lubatav juurdepääsu taotleva isiku õigustatud huvi korral; |

## Autode registri eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

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

### Registrit kasutavad pädevusalad

* Juhataja
* Autode haldur
* Klient
* Uudistaja
* Klienditeenindaja
* Klassifikaatorite haldur

### Registrit teenindavad funktsionaalsed allsüsteemid

Autode registrit teenindab (loeb ja muudab) autode funktsionaalne allsüsteem.

### Infovajadused, mida register aitab rahuldada

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

### Seosed teiste registritega

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

**Klassifikaatorite register** – Klassifikaatorite registriga on auto seotud olemitüübi auto\_seisundi\_liik kaudu. Selle abil registreeritakse auto hetkeseisund. Samuti on iga auto seotud null või rohkema auto kategooriaga, mis on samuti klassifikaator.

Selleks, et saaks registreerida andmeid hinnakirja, rentimise, kahjunõuete, kindlustuse, ülevaatuste, hoolduste, rikete registrites, peavad olema registreeritud auto andmed ja seega peab olema realiseeritud autode register.

### Ärireeglid

Jõustatavad autode registri põhjal

* Igal autol on unikaalne kood
* Igal autol on unikaalne nimetus
* Iga auto on käesoleval ajahetkel täpselt ühes seisundis vastavalt oma elutsüklile.
* Iga auto on seotud null või rohkema kategooriaga
* Iga auto ja iga kategooria vahel saab olla maksimaalselt üks seos
* Iga auto puhul on vaja registreerida töötaja, kes auto andmed registreeris ning auto registreerimise aeg
* Auto andmeid (sh auto kategooriasse kuulumine) (v.a seisund) saab muuta vaid siis, kui see on ootel või mitteaktiivses seisundis.
* Auto andmete muutmisel ei saa muuta seda registreerinud töötajat ja registreerimise aega
* Auto andmeid saab andmebaasist kustutada vaid siis, kui see on ootel seisundis
* Auto saab aktiveerida vaid siis, kui see on seotud vähemalt ühe auto kategooriaga
* Igal autol on määratud kütuse liik
* Organisatsioon töötab kindla riigi territooriumil, organisatsiooni autode registreerimisnumbrid on unikaalsed ja ei kandu edasi teistele sõidukitele.

Jõustatavad teiste registrite põhjal, kuid vajalikud autode 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 autode registri kontseptuaalse andmemudeli olemi‑suhte diagrammist.



Joonis 3 Autode registri kontseptuaalne eskiismudel.

# Detailanalüüs

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

## Autode funktsionaalse allsüsteemi detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt autode funktsionaalse allsüsteemi toimimist.

### Kasutusjuhtude mudel

Autode 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**: Autode haldur, Juhataja, Klienditeenindaja, Klient – (edaspidi Subjekt).

**Osapooled ja nende huvid**:

* Autode haldur, Juhataja, Klienditeenindaja, Klient: 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. Süsteemi sisenemiseks ja selle kasutamiseks ei tohi isik olla seisundis "surnud" ning tema roll (klient või töötaja) ei tohi olla seisundites, mis tähistavad organisatsiooniga sõlmitud suhte peatamist või lõppemist.

**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 auto

**Primaarne tegutseja**: Autode haldur

**Osapooled ja nende huvid**:

* Autode haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate autode 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 autodest ning uue auto tekkimisel ei tohi selle registreerimisega viivitada.
* Klient, Uudistaja: Soovivad võimalikult täpset infot auto kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga autot kasutava kliendi rollis.

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

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

**Järeltingimused**: Auto on registreeritud ja auto on seisundis „Ootel“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

2a. Kui ühtegi auto kategooriat pole registreeritud, siis kategooriate valikut ei pakuta ning auto kategooriasse kuulumist ei saa registreerida.

3a Autode haldur soovib auto mõnest määratud kategooriast kohe eemaldada.

3b **Süsteem** kuvab nimekirja kategooriatest, kuhu auto juba kuulub. Iga kategooria juures on ka selle kategooria tüübi nimetus. (**OP2.2**)

3c **Süsteem** salvestab kategooriast eemaldamise (**OP8**).

**Kasutusjuht:** Unusta auto

**Primaarne tegutseja**: Autode haldur

**Osapooled ja nende huvid**:

* Autode haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate autode andmed ja et need andmed oleksid võimalikult täpsed. Kui on selge, et autot 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 autodest ning uue auto tekkimisel ei tohi selle registreerimisega viivitada. Samas ei soovi ta näha autosid, millest asja ei saa.
* Klient, Uudistaja: Soovivad võimalikult täpset infot autode kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga autot kasutava kliendi rollis.

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

**Eeltingimused**: Autode haldur on autenditud ja autoriseeritud. Auto on registreeritud ja on seisundis „Ootel“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Autode haldur avaldab soovi auto unustada, st selle andmed süsteemist kustutada.
2. **Süsteem** kuvab ootel autode nimekirja, kus on auto\_kood, nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood (**OP3.1**)
3. Autode haldur valib nimekirjast auto ja annab korralduse see unustada.
4. **Süsteem** salvestab andmed (**OP2**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi ootel autot, siis ei saa autode haldur jätkata.

**Kasutusjuht:** Muuda auto andmeid

**Primaarne tegutseja**: Autode haldur

**Osapooled ja nende huvid**:

* Autode haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate autode 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 autodest.
* Klient, Uudistaja: Soovivad võimalikult täpset infot autode kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga autot kasutava kliendi rollis.

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Autode haldur soovib muuta auto andmeid.
2. *Käivitub kasutusjuht „Vaata kõiki ootel või mitteaktiivseid autosid“*
3. Autode haldur valib nimekirjast auto ja annab korralduse vaadata selle detailseid andmeid.
4. **Süsteem** kuvab muutmiseks mõeldud väljades auto põhiandmed (auto\_kood, nimetus, mark, mudel, valjalaske\_aasta, mootori\_maht, auto\_kütuse\_liik, istekohtade\_arv, reg\_number, vin\_kood) (**OP4.1**) ning sellega seotud kategooriate ja kategooriate tüüpide nimetused (**OP2.2**). Seal on muuhulgas võimalik määrata, millistesse kategooriatesse auto kuulub, sest süsteem pakub kategooriate valiku (**OP2.1**).
5. Autode haldur muudab auto andmeid ja annab korralduse salvestada.
6. **Süsteem** salvestab andmed (**OP6**).

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

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

5a. Autode haldur võib lisada auto uude kategooriasse ja anda korralduse salvestada.

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

5b Autode haldur võib eemaldada auto kategooriast ja anda korralduse salvestada.

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

5c Kui ühtegi autode kategooriat pole registreeritud, siis kategooriate valikut ei pakuta ning auto kategooriasse kuulumist ei saa registreerida.

**Kasutusjuht:** Aktiveeri Auto

**Primaarne tegutseja**: Autode haldur

**Osapooled ja nende huvid**:

* Autode haldur, Juhataja: Soovib, et iga auto kohta oleks teada tema koht üldises auto elutsüklis, mis ühtlasi määrab tegevused, mida selle autoga saab teha.
* Autode haldur: Soovib, et autot saaks kasutada uutes tehingutes.
* Klient, Uudistaja: Soovivad näha kõiki aktiivseid autosid, et otsustada, kas siduda ennast selle organisatsiooniga autot kasutava kliendi rollis.

**Käivitav sündmus**: Auto ooteperiood või autoga seoses tekkinud ajutised probleemid on lahenenud ning auto põhjal saab uuesti tehinguid teha.

**Eeltingimused**: Autode haldur on autenditud ja autoriseeritud. Auto on registreeritud ja on seisundis „Ootel“ või „Mitteaktiivne“. Auto on määratud vähemalt ühte auto kategooriasse.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

*Auto 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 autot, siis ei saa autode haldur jätkata.

4a. Kui auto ei kuulu ühtegi autodekategooriasse, siis aktiveerimine ebaõnnestub.

**Kasutusjuht**: Muuda auto mitteaktiivseks

**Primaarne tegutseja**: Autode haldur

**Osapooled ja nende huvid**:

* Autode haldur, Juhataja: Soovib, et iga auto kohta oleks teada tema koht üldises auto elutsüklis, mis ühtlasi määrab tegevused, mida selle auoga saab teha.
* Autode haldur: Soovib auto 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 autosid, et otsustada, kas siduda ennast selle organisatsiooniga autot kasutava kliendi rollis (kui huvi pakkuv auto ei ole selles nimekirjas, siis see on talle samuti oluline informatsioon).

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

**Eeltingimused**: Autode haldur on autenditud ja autoriseeritud. Auto on registreeritud ja on seisundis „Aktiivne“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Autode haldur avaldab soovi auto mitteaktiivseks muuta.
2. **Süsteem** kuvab aktiivsete autode nimekirja, kus on auto\_kood, nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood (**OP6.1**)
3. Autode haldur valib nimekirjast auto ja annab korralduse see mitteaktiivseks muuta.
4. **Süsteem** salvestab andmed (**OP4**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi aktiivset autot, siis ei saa autode haldur jätkata.

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

**Primaarne tegutseja**: Autode haldur.

**Osapooled ja nende huvid**:

* Autode haldur: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata ootel või mitteaktiivsete autode nimekirja
2. **Süsteem** kuvab ootel või mitteaktiivses seisundis autode nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood (**OP7.1**)

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

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

**Kasutusjuht**: Vaata kõiki autosid

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

**Osapooled ja nende huvid**:

* Juhataja, Autode haldur: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata kõikide autode andmeid.
2. **Süsteem** kuvab kõigi autode nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood (**OP8.1**)
3. Subjekt valib auto, mida ta soovib detailsemalt vaadata.
4. **Süsteem** kuvab vaatamiseks mõeldud väljades auto põhiandmed andmed (auto\_kood, nimetus, mark, mudel, valjalaske\_aasta, mootori\_maht, auto\_kütuse\_liik, istekohtade\_arv, reg\_number, vin\_kood, registreerimise aeg, registreerinud töötaja eesnimi, perenimi ja e-meili aadress) (**OP8.2**) ning sellega seotud kategooriate ja kategooriate tüüpide nimetused (**OP2.2**).

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

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

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

**Kasutusjuht**: Lõpeta auto

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

**Käivitav sündmus**: Auto kasutamine tehingutes on vaja püsivalt lõpetada, kuna seoses autoga on ilmnenud püsiva iseloomuga probleemid või kuna auto on oma aja lihtsalt ära elanud

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Juhataja avaldab soovi auto lõpetada.
2. **Süsteem** kuvab aktiivsete või mitteaktiivsete autode nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood (**OP9.1**)
3. Juhataja valib nimekirjast auto 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 autot, siis ei saa juhataja jätkata.

**Kasutusjuht**: Vaata auto koondaruannet

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

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

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

**Järeltingimused**: Auto koondaruanne on moodustatud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

**Kasutusjuht**: Vaata aktiivseid autosid

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

**Osapooled ja nende huvid**:

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib näha kõiki organisatsiooni pakutavaid aktiivseid autosid.
2. **Süsteem** kuvab nimekirja kategooriatest (**OP2.1**)
3. Subjekt valib konkreetse kategooria.
4. **Süsteem** kuvab sellesse kuuluvate aktiivsete autode nimekirja. Iga auto kohta esitatakse kood, nimetus, kategooriate ja nende tüüpide nimetused, valjalaske\_aasta, mark, mudel, istekohtade arv, mootori maht, kütuse liik (**OP11.2**)

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

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

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

## Autode funktsionaalse allsüsteemi vajatavate registrite detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt autode 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.

esitatud olemi‑suhte diagrammidel on värvidel järgmine tähendus.

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



Joonis 4 Auto registri olemi-suhte diagramm



Joonis 5 Isikute registri olemi-suhte diagramm.



Joonis 6 Klientide registri olemi-suhte diagramm.



Joonis 7 Töötajate registri olemi-suhte diagramm.



Joonis 8 Laiendatud auto registri olemi-suhte diagrammid.

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

Tabel 5 Olemitüüpide sõnalised kirjeldused.

| **Olemitüübi nimi**  **(teised nimed)** | **Kuuluvus registrisse** | **Definitsioon** |
| --- | --- | --- |
| Amet | Klassifikaatorite register | Amet on töölepingus sätestatud ametikohustuste üldnimetus. Ametid on klassifikaatorid. Näited on juhataja ja klienditeenindaja. |
| Auto | Autode register | Auto on sõiduvahend, mis on organisatsioonis varana arvel ning millega sooritatakse tehinguid organisatsiooni klientidega. „Auto on vähemalt kolmerattaline ja kaheteljeline mootorsõiduk reisijate või veoste vedamiseks rööpmeta teedel või maastikul.“ (Vikipeedia) Auto all mõeldakse sõiduautot. „Sõiduautodeks loetakse tavaliselt reisijate vedamiseks mõeldud väiksemaid, 2–9[viide?]-kohalisi autosid.“ (Vikipeedia) |
| Auto\_kategooria | Klassifikaatorite register | Võimaldab auto klassifitseerimist erinevatesse kategooriatesse ja selle alusel autode rühmitamist teatud põhjusel huvipakkuvateks hulkadeks. Tegemist on üksteist mittevälistavate kategooriatega, st üks ja sama auto võib kuuluda korraga mitmesse sama tüüpi kategooriasse. Näited on pereauto ja väikeauto. |
| Auto\_kategooria\_  omamine | Autode register | Näitab auto kuulumist kategooriatesse. Iga auto ja iga auto kategooria vahel võib olla maksimaalselt üks seos. |
| Auto\_kategooria\_tüüp | Klassifikaatorite register | Võimaldab rühmitada auto klassifitseerimiseks kasutatavaid kategooriaid ühise nime alla. Need nimed kirjeldavad, mis liiki klassifikatsiooniga on tegemist. Näide on sihtgrupp. |
| Auto\_kütuse\_liik | Klassifikaatorite register | Võimaldab auto klassifitseerimist kütuseliigi alusel. Näited on bensiin ja diisel. |
| Auto\_mark | Klassifikaatorite register | Võimaldab auto klassifitseerimist auto tootnud organisatsiooni alusel. Näited on Opel ja Tesla. |
| Isik | Isikute register | Mistahes organisatsiooniga seotud füüsiline isik (eraisik). Isik võib olla seotud organisatsiooniga näiteks kui klient või kui töötaja. |
| Isiku\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga isiku puhul tema hetkeseisundi vastavalt üldisele isikute elutsüklile. Näited on elus ja surnud. |
| 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) |
| Kliendi\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga kliendi puhul tema hetkeseisundi vastavalt üldisele kliendi elutsüklile. Näited on aktiivne ja mustas nimekirjas. |
| Klient | Klientide register | Organisatsioonis end kliendina registreerinud isik, kellel on võimalik osta organisatsiooni poolt pakutavaid tooteid või teenuseid. |
| 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) |
| 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. |
| Töötaja\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga töötaja puhul tema hetkeseisundi vastavalt üldisele töötajate elutsüklile. Näited on tööl ja puhkusel. |

Tabel 6 esitab atribuutide sõnalised kirjeldused.

Tabel 6 Atribuutide sõnalised kirjeldused.

| **Olemitüübi nimi** | **Atribuudi nimi (teised nimed)** | **Atribuudi definitsioon** | **Näiteväärtus** |
| --- | --- | --- | --- |
| Amet | kirjeldus | Ametist tulenevate õiguste ja kohustuste vabatekstiline kirjeldus.  **{Kirjeldus ei tohi olla tühi string või ainult tühikutest koosnev string. Kasutage andmetüüpi, mis võimaldab suurimat võimalikku stringi pikkust.}** | Juhib organisatsiooni igapäevast tööd ning langetab strateegilisi otsuseid |
| Auto | auto\_kood | Auto arvuline kood, mis sisestatakse inimkasutaja poolt, mitte ei genereerita süsteemi poolt.  **{Auto unikaalne identifikaator. Registreerimine on kohustuslik}** | 222 |
| Auto | nimetus  (hüüdnimi) | Auto hüüdnimi, millega sellele äratuntavalt viidata.  **{Auto unikaalne identifikaator. Võib olla kuni 50 märki. Registreerimine on kohustuslik. Nimetus ei tohi olla tühi string ega ainult tühikutest koosnev string.}** | Nimetus |
| Auto | reg\_aeg | Auto registreerimise aeg kuupäeva ja kellaaja täpsusega. Selle võib süsteem ise automaatselt määrata.  **{Registreerimine on kohustuslik. Väärtus peab olema vahemikus 01. jaanuar 2010 00:00:00 ja 31. detsember 2100 kell 23:59:59 (otspunktid kaasa arvatud).}** | 22.03.2015 12:33:12 |
| Auto | valjalaske\_ aasta | Auto tootmisaasta.  **{Registreerimine on kohustuslik.**  **Väärtus peab olema vahemikus 2000 – 2100 (otspunktid kaasa arvatud).}** | 2009 |
| Auto | istekohtade\_ arv | Auto istekohtade arv.  **{Registreerimine on kohustuslik.**  **Väärtus peab olema vahemikus 2 kuni 11 (otspunktid kaasa arvatud).}** | 4 |
| Auto | mudel | Auto mudel. Mudel on autotootja poolt kasutatav nimi, et kirjeldada ja reklaamida ühesuguste omadustega autosid.  **{Registreerimine on kohustuslik. Mudel ei tohi olla tühi string ega tühikutest koosnev string.}** | A4 |
| Auto | mootori\_maht | Auto mootori maht liitrites.  **{Registreerimine on kohustuslik. Ei tohi olla negatiivne arv. Kümnendmurd täpsusega üks koht peale koma.}** | 2.5 |
| Auto | reg\_number (registreerimismärk, registrinumber) | Eesti autoregistri poolt väljastatud registreerimisnumber. Erinevatel autodel võib erinevatel aegadel olla sama number.  **{Registreerimine on kohustuslik.**  **Ei tohi leiduda kahte aktiivses seisundis autot, millel on sama registrinumber, st aktiivsete autode registrinumbrid peavad olema unikaalsed.**  **Registrinumber ei tohi olla tühi string ega tühikutest koosnev string. Registrinumbris on lubatud suurtähed ja numbrid. Märkide hulk peab olema 2-9 (otspunktid kaasa arvatud). Registrinumber peab järgima ühte järgmistest mustritest.**   1. **Üks või rohkem suurtähte, millele järgneb üks või rohkem numbrit.** 2. **Kaks või kolm numbrit, millele järgneb kolm suurtähte.}** | QWE321 |
| Auto | vin\_kood | Tehase poolt väljastatud unikaalne auto kood.  **{Auto unikaalne identifikaator. Registreerimine on kohustuslik. Vin kood peab olema 11-17 märki (otspunktid kaasa arvatud) ning tohib sisaldada ainult numbreid ja suurtähti.}** | WAUFFAFM3CA000000 |
| Isik | isikukood | Riigi poolt väljastatud isiku identifikaator, mis on unikaalne selle väljastanud riigi piires.  **{Registreerimine on kohustuslik. Koos riigi identifikaatoriga on isiku unikaalne identifikaator.**  **Isikukoodis on lubatud tähed, numbrid, tühikud, sidekriipsud, plussmärgid, võrdusmärgid ja kaldkriipsud.**  **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 | sünni\_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, e-posti aadress) | 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 täpselt ühte "@" märki. Võib olla kuni 254 märki pikk (Veljovic, 2017).}** | 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 00:00:00 ja 31. detsember 2100 kell 23:59:59 (otspunktid kaasa arvatud).}** | 12.08.2014 17:01:43 |
| 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.**  **Kui kood on tekstiline väärtus, siis ei tohi see olla tühi string või ainult tühikutest koosnev string.}** | 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. Erandiks on Auto\_kategooria nimetus, mis peab olema unikaalne kombinatsioonis Auto\_kategooria\_tüübiga, st erinevat tüüpi kategooriates võib olla sama nimetusega kategooriaid.**  **Registreerimine on kohustuslik. Nimetus ei tohi olla tühi string või**  **ainult tühikutest koosnev string.}** | Aktiivne |
| Klient | on\_nous\_tylitamisega | Kirjeldab ära kas klient on nõus saama organisatsiooni poolt saadetavaid reklaamkirju  **{Registreerimine on kohustuslik. Vaikeväärtus on ’False’.}** | True |

### Andmebaasioperatsioonide lepingud

**OP1 Registreeri auto(p\_auto\_kood, p\_nimetus, p\_valjalaske\_aasta, p\_istekohtade\_arv, p\_mudel, p\_mootori\_maht, p\_reg\_number, töötaja identifikaator, kütuse liik identifikaator, mark identifikaator)**

**Eeltingimused**:

* Auto\_seisundi\_liik eksemplar asl (millel on nimetus="Ootel") on registreeritud
* Auto\_kütuse\_liik eksemplar akl (millel on kütuse liik identifikaator) on registreeritud
* Auto\_mark eksemplar am (millel on mark identifikaator) on registreeritud
* Töötaja eksemplar t (millel on töötaja identifikaator) on registreeritud

**Järeltingimused**:

--Loo eksemplare

* Auto eksemplar a on registreeritud

--Väärtusta atribuute

* a.auto\_kood:= p\_auto\_kood
* a.nimetus:= p\_nimetus
* a.reg\_aeg:= hetke kuupäev + kellaaeg
* a. valjalaske\_aasta:= p\_ valjalaske\_aasta
* a.istekohtade\_arv:= p\_istekohtade\_arv
* a.mudel:= p\_mudel
* a.mootori\_maht:= p\_mootori\_maht
* a.reg\_number:= p\_reg\_number
* a.vin\_kood:= p\_vin\_kood

--Loo seoseid

* a ja asl seos on registreeritud
* a ja t seos on registreeritud
* a ja akl seos on registreeritud
* a ja am seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri auto

**OP2 Unusta auto(p\_auto\_kood)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. a on seotud auto\_seisundi\_liik eksemplariga asl (millel on nimetus="Ootel")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

1. a ja kõik selle seosed on andmebaasist kustutatud

**Kasutus kasutusjuhtude poolt**: Unusta auto

**OP3 Aktiveeri Auto (p\_auto\_kood)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. a on seotud auto\_seisundi\_liik eksemplariga asl\_vana (millel on nimetus="Ootel") või (nimetus="Mitteaktiivne")
3. auto\_seisundi\_liik eksemplar asl\_uus (millel on nimetus="Aktiivne") on registreeritud
4. Leidub vähemalt üks auto\_kategooria\_omamine eksemplar ako, mis on seotud a-ga

**Järeltingimused**:

--Kustuta seoseid

1. a ja asl\_vana seos on kustutatud

--Loo seoseid

1. a ja asl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Aktiveeri auto

**OP4 Muuda auto mitteaktiivseks autoks(p\_auto\_kood)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. a on seotud auto\_seisundi\_liik eksemplariga asl\_vana (millel on nimetus="Aktiivne")
3. auto\_seisundi\_liik eksemplar asl\_uus (millel on nimetus="Mitteaktiivne") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

1. a ja asl\_vana seos on kustutatud

--Loo seoseid

1. a ja asl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda auto mitteaktiivseks

**OP5 Lõpeta auto(p\_auto\_kood)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. a on seotud auto\_seisundi\_liik eksemplariga asl\_vana (millel on nimetus="Aktiivne") või (nimetus="Mitteaktiivne")
3. auto\_seisundi\_liik eksemplar asl\_uus (millel on nimetus="Lõpetatud") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

1. a ja asl\_vana seos on kustutatud

--Loo seoseid

1. a ja asl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Lõpeta auto

**OP6 Muuda auto andmeid (p\_auto\_kood\_vana, p\_auto\_kood\_uus, p\_nimetus, p\_valjalaske\_aasta, p\_istekohtade\_arv, p\_mudel, p\_mootori\_maht, p\_reg\_number, kütuse liik identifikaator, mark identifikaator)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood\_vana) on registreeritud
2. a on seotud auto\_seisundi\_liik eksemplariga asl (millel on nimetus="Ootel") või (nimetus="Mitteaktiivne")
3. Auto\_kütuse\_liik eksemplar akl (millel on kütuse liik identifikaator) on registreeritud
4. Auto\_mark eksemplar am (millel on mark identifikaator) on registreeritud

**Järeltingimused**:

--Väärtusta atribuute

1. a.auto\_kood:= p\_auto\_kood\_uus
2. a.nimetus:= p\_nimetus
3. a. valjalaske\_aasta:= p\_ valjalaske\_aasta
4. a.istekohtade\_arv:= p\_istekohtade\_arv
5. a.mudel:= p\_mudel
6. a.mootori\_maht:= p\_mootori\_maht
7. a.reg\_number:= p\_reg\_number
8. a.vin\_kood:= p\_vin\_kood

--Kustuta seoseid

1. a olemasolev seos kütuse liigiga on kustutatud.
2. a olemasolev seos auto margiga on kustutatud

--Loo seoseid

1. a ja akl seos on registreeritud.
2. a ja am seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda auto andmeid

**OP7 Lisa auto kategooriasse (p\_auto\_kood, auto kategooria identifikaator)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. auto\_kategooria eksemplar ak (millel on auto kategooria identifikaator) on registreeritud
3. a on seotud auto\_seisundi\_liik eksemplariga asl (millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne")

**Järeltingimused**:

--Loo eksemplare

1. auto\_kategooria\_omamine eksemplar ako on registreeritud

--Loo seoseid

1. a ja ako seos on registreeritud
2. ak ja ako seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri Auto, Muuda auto andmeid

**OP8 Eemalda auto kategooriast (p\_auto\_kood, auto kategooria identifikaator)**

**Eeltingimused**:

1. Auto eksemplar a (millel on auto\_kood=p\_auto\_kood) on registreeritud
2. Auto\_kategooria eksemplar ak (millel on auto kategooria identifikaator) on registreeritud
3. a on seotud auto\_seisundi\_liik eksemplariga asl (millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

1. auto\_kategooria\_omamine eksemplar ako, mis on seotud a-ga ja mis on seotud ak-ga, on koos oma seostega kustutatud

**Kasutus kasutusjuhtude poolt**: Registreeri auto, Muuda auto andmeid

### Registri põhiobjekti seisundidiagramm

esitab seisundidiagrammi, mis kirjeldab registri põhiobjekti auto kõikvõimalikke elutsükleid*.*



Joonis 9 Auto 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 7 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 |  |  | R |  |  |  |  |  | R |
| Isiku\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Töötaja\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Auto\_kategooria |  | R |  | R |  |  |  | R |  |  | R | R |
| Auto\_kategooria\_tüüp |  | R |  | R |  |  |  | R |  |  | R |  |
| Auto\_seisundi\_liik |  | R | R | R | R | R | R | R | R | R | R | R |
| Isik | R |  |  |  |  |  |  | R |  |  |  | R |
| Töötaja | R | R |  |  |  | R |  | R |  |  |  | R |
| Auto |  | C | RD | RU | RU | RU | R | R | RU | R | R | CRUD |
| Auto\_kategooria\_omamine |  | CRD | D | CRD |  |  |  |  |  |  | R | CRD |
| Auto\_kütuse\_liik |  | R |  | R |  |  |  | R |  |  | R | R |
| Auto\_mark |  | R |  | R |  |  |  | R | R |  | R | R |
| Klient | R |  |  |  |  |  |  |  |  |  |  | R |
| Kliendi\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |

1 – Tuvasta kasutaja

2 – Registreeri auto

3 – Unusta auto

4 – Muuda auto andmeid

5 – Aktiveeri auto

6 – Muuda auto mitteaktiivseks

7 – Vaata kõiki ootel või mitteaktiivseid autosid

8 – Vaata kõiki autosid

9 – Lõpeta auto

10 – Vaata auto koondaruannet

11 – Vaata aktiivseid autosid

# Füüsiline disain

Selles peatükis esitatakse mudel, mis kirjeldab autode funktsionaalse allsüsteemi toimimiseks vajalike registrite tehnilist lahendust MS Access andmebaasisüsteemis.

## Autode funktsionaalse allsüsteemi vajatavate registrite füüsiline disain



Joonis 10 Autode registri füüsilise disaini andmebaasi diagramm



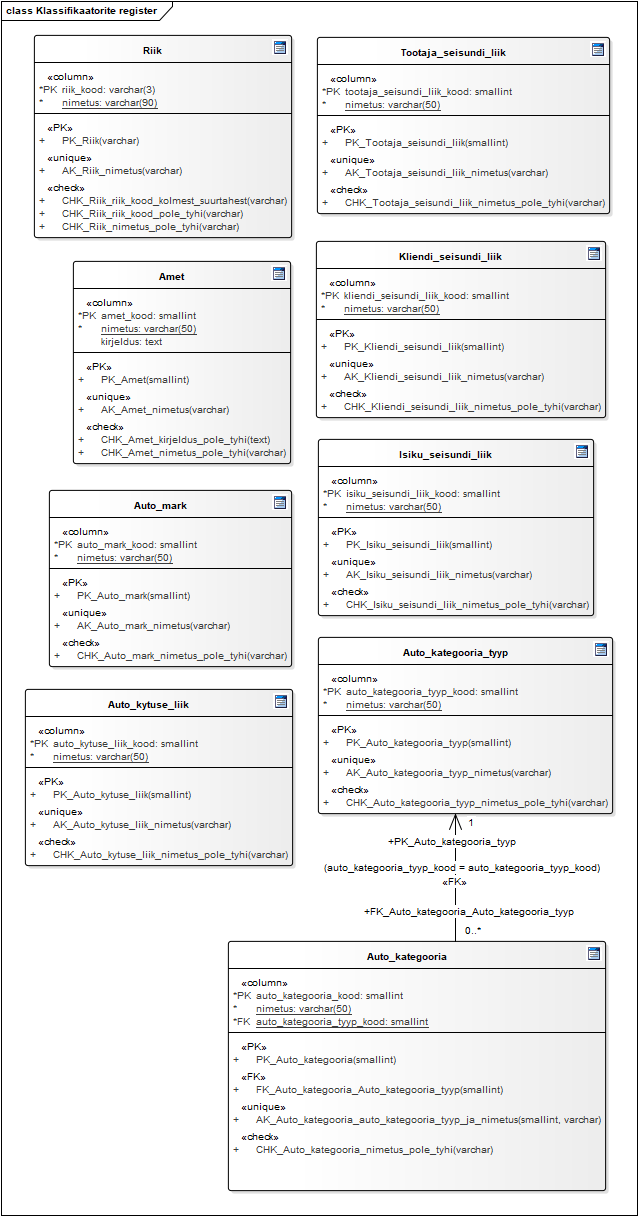
Joonis 11 Klientide registri füüsilise disaini andmebaasi diagramm



Joonis 12 Töötajate registri füüsilise disaini andmebaasi diagramm.



Joonis 13 Isikute registri füüsilise disaini andmebaasi diagramm.



Joonis 14 Klassifikaatorite registri füüsilise disaini andmebaasi diagramm.

# Realisatsioon PostgreSQLis

Selles peatükis esitatakse andmebaasi PostgreSQLis **(12)** realiseerimiseks mõeldud laused.

## Andmebaasi loomine

Andmebaasis vastuvotuajad kasutatav märkide klassifikatsioon ja märkide võrdlusreeglistik vastavad eesti keele reeglitele. Järgnev käsk tuleb käivitada shelli promptis.

createdb -l et\_EE.utf8 -T template0 t192406

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.

SET client\_encoding=LATIN9;

Järgnevalt esitatakse andmebaasiobjektide loomise SQL laused.

## Skeemid

Loon skeemiobjektid skeemis nimega public, mis loodi andmebaasi loomisel automaatselt.

## Domeenid

CREATE DOMAIN **d\_nimetus** VARCHAR(50) NOT NULL

CONSTRAINT chk\_nimetus\_pole\_tyhi CHECK (VALUE!~'^[[:space:]]\*$');

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

CONSTRAINT chk\_reg\_aeg\_on\_vahemikus

CHECK ((VALUE >= '2010-01-01') AND (VALUE < '2101-01-01'));

## Tabelid ja arvujada generaatorid

Arvujada generaatorid luuakse andmebaasisüsteemi poolt automaatselt tänu

CREATE TABLE lausetes SERIAL notatsiooni kasutamisele ning seotakse

automaatselt veergudega, mille puhul seda kasutatakse.

CREATE TABLE **Auto\_kategooria\_tyyp**(

auto\_kategooria\_tyyp\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

CONSTRAINT PK\_Auto\_kategooria\_tyyp PRIMARY KEY (auto\_kategooria\_tyyp\_kood),

CONSTRAINT AK\_Auto\_kategooria\_tyyp\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Auto\_kategooria\_tyyp\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Auto\_kategooria**(

auto\_kategooria\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

auto\_kategooria\_tyyp\_kood smallint NOT NULL,

CONSTRAINT PK\_Auto\_kategooria PRIMARY KEY (auto\_kategooria\_kood),

CONSTRAINT AK\_Auto\_kategooria\_auto\_kategooria\_tyyp\_ja\_nimetus UNIQUE (auto\_kategooria\_tyyp\_kood,nimetus),

CONSTRAINT CHK\_Auto\_kategooria\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$'),

CONSTRAINT FK\_Auto\_kategooria\_Auto\_kategooria\_tyyp FOREIGN KEY (auto\_kategooria\_tyyp\_kood) REFERENCES Auto\_kategooria\_tyyp (auto\_kategooria\_tyyp\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE **Auto\_mark**(

auto\_mark\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

CONSTRAINT PK\_Auto\_mark PRIMARY KEY (auto\_mark\_kood),

CONSTRAINT AK\_Auto\_mark\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Auto\_mark\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Auto\_kytuse\_liik**(

auto\_kytuse\_liik\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

CONSTRAINT PK\_Auto\_kytuse\_liik PRIMARY KEY (auto\_kytuse\_liik\_kood),

CONSTRAINT AK\_Auto\_kytuse\_liik\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Auto\_kytuse\_liik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Auto\_seisundi\_liik**(

auto\_seisundi\_liik\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

CONSTRAINT PK\_Auto\_seisundi\_liik PRIMARY KEY (auto\_seisundi\_liik\_kood),

CONSTRAINT AK\_Auto\_seisundi\_liik\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Auto\_seisundi\_liik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Isiku\_seisundi\_liik**(

isiku\_seisundi\_liik\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

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

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

CONSTRAINT CHK\_Isiku\_seisundi\_liik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Kliendi\_seisundi\_liik**(

kliendi\_seisundi\_liik\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

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

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

CONSTRAINT CHK\_Kliendi\_seisundi\_liik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Tootaja\_seisundi\_liik**(

tootaja\_seisundi\_liik\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

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

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

CONSTRAINT CHK\_Tootaja\_seisundi\_liik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Riik**(

riik\_kood varchar(3) NOT NULL,

nimetus varchar(90) NOT NULL,

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

CONSTRAINT AK\_Riik\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Riik\_riik\_kood\_kolmest\_suurtahest CHECK (riik\_kood ~ '^[A-Z]{3}$'),

CONSTRAINT CHK\_Riik\_riik\_kood\_pole\_tyhi CHECK (riik\_kood!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Riik\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

);

CREATE TABLE **Amet**(

amet\_kood smallint NOT NULL,

nimetus varchar(50) NOT NULL,

kirjeldus text,

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

CONSTRAINT AK\_Amet\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Amet\_kirjeldus\_pole\_tyhi CHECK (kirjeldus!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Amet\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$')

)WITH (FILLFACTOR=90);

CREATE TABLE **Isik**(

isik\_id serial NOT NULL,

isikukood varchar(50) NOT NULL,

riik\_kood varchar(3) NOT NULL,

e\_meil varchar(254) NOT NULL,

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

synni\_kp date NOT NULL,

parool varchar(60) NOT NULL,

reg\_aeg timestamp NOT NULL DEFAULT LOCALTIMESTAMP(0),

eesnimi varchar(1500),

perenimi varchar(1000),

elukoht varchar(120),

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

CONSTRAINT AK\_Isik\_isikukood\_ja\_riik\_kood UNIQUE (isikukood,riik\_kood),

CONSTRAINT CHK\_Isik\_isikukood\_pole\_tyhi CHECK (isikukood!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Isik\_isikukood\_ainult\_lubatud\_symbolid CHECK (isikukood ~ '^([[:alpha:]]|[[:digit:]]|[[:space:]]|-|\+|=|\\|\/)\*$'),

CONSTRAINT CHK\_Isik\_eesnimi\_voi\_perenimi\_olemas CHECK ((eesnimi IS NOT NULL) OR (perenimi IS NOT NULL)),

CONSTRAINT CHK\_Isik\_eesnimi\_pole\_tyhi CHECK (eesnimi!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Isik\_perenimi\_pole\_tyhi CHECK (perenimi!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Isik\_synni\_kp\_on\_vahemikus CHECK ((synni\_kp >= '1900-01-01') AND (synni\_kp < '2101-01-01')),

CONSTRAINT CHK\_Isik\_synni\_kp\_pole\_suurem\_reg\_ajast CHECK (synni\_kp <= reg\_aeg),

CONSTRAINT CHK\_Isik\_elukoht\_pole\_tyhi CHECK (elukoht!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Isik\_elukoht\_pole\_ainult\_numbritest CHECK (elukoht!~'^[[:digit:]]\*$'),

CONSTRAINT CHK\_Isik\_e\_meil\_sisaldab\_tapselt\_uhe\_at\_marki CHECK (e\_meil ~ '^[^@]\*@[^@]\*$'),

CONSTRAINT CHK\_Isik\_reg\_aeg\_on\_vahemikus CHECK ((reg\_aeg >= '2010-01-01') AND (reg\_aeg < '2101-01-01')),

CONSTRAINT FK\_Isik\_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\_Riik FOREIGN KEY (riik\_kood) REFERENCES Riik (riik\_kood) ON DELETE No Action ON UPDATE Cascade

)WITH (FILLFACTOR=90);

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\_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\_Isik FOREIGN KEY (isik\_id) REFERENCES Isik (isik\_id) ON DELETE Cascade ON UPDATE No Action

)WITH (FILLFACTOR=90);

CREATE TABLE **Tootaja**(

isik\_id integer NOT NULL,

amet\_kood smallint NOT NULL,

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

mentor integer,

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

CONSTRAINT CHK\_Tootaja\_pole\_enda\_mentor CHECK (mentor <> isik\_id),

CONSTRAINT FK\_Tootaja\_Tootaja FOREIGN KEY (mentor) REFERENCES Tootaja (isik\_id) ON DELETE Set Null ON UPDATE No Action,

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

CONSTRAINT FK\_Tootaja\_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\_Isik FOREIGN KEY (isik\_id) REFERENCES Isik (isik\_id) ON DELETE Cascade ON UPDATE No Action

)WITH (FILLFACTOR=90);

CREATE TABLE **Auto**(

auto\_kood integer NOT NULL,

nimetus varchar(50) NOT NULL,

mudel varchar(100) NOT NULL,

valjalaske\_aasta smallint NOT NULL,

reg\_number varchar(9) NOT NULL,

istekohtade\_arv smallint NOT NULL,

mootori\_maht decimal(3,1) NOT NULL,

vin\_kood varchar(17) NOT NULL,

reg\_aeg timestamp NOT NULL DEFAULT LOCALTIMESTAMP(0),

registreerija\_id integer NOT NULL,

auto\_kytuse\_liik\_kood smallint NOT NULL,

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

auto\_mark\_kood smallint NOT NULL,

CONSTRAINT PK\_Auto PRIMARY KEY (auto\_kood),

CONSTRAINT AK\_Auto\_vin\_kood UNIQUE (vin\_kood),

CONSTRAINT AK\_Auto\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Auto\_nimetus\_pole\_tyhi CHECK (nimetus!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Auto\_reg\_aeg\_on\_vahemikus CHECK ((reg\_aeg >= '2010-01-01') AND (reg\_aeg < '2101-01-01'));,

CONSTRAINT CHK\_Auto\_valjalaske\_aasta\_on\_vahemikus CHECK (valjalaske\_aasta >= 2000 AND

valjalaske\_aasta <= 2100),

CONSTRAINT CHK\_Auto\_istekohtade\_arv\_on\_vahemikus CHECK (istekohtade\_arv >= 2 AND

istekohtade\_arv <= 11),

CONSTRAINT CHK\_Auto\_mudel\_pole\_tyhi CHECK (mudel!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Auto\_mootori\_maht\_on\_positiivne CHECK (mootori\_maht >= 0),

CONSTRAINT CHK\_Auto\_reg\_number\_pole\_tyhi CHECK (reg\_number!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Auto\_reg\_number\_min\_pikkus CHECK (LENGTH(reg\_number) >= 2),

CONSTRAINT CHK\_Auto\_reg\_number\_ainult\_suurtahed\_ja\_numbrid CHECK (reg\_number ~ '^([[:upper:]]|[[:digit:]])\*$'),

CONSTRAINT CHK\_Auto\_reg\_number\_muster CHECK (reg\_number ~ '^([[:digit:]]{2}|[[:digit:]]{3})[[:upper:]]{3}$' OR reg\_number ~ '^[[:upper:]]+[[:digit:]]+$'),

CONSTRAINT CHK\_Auto\_vin\_kood\_min\_pikkus CHECK (LENGTH(vin\_kood) >= 11),

CONSTRAINT CHK\_Auto\_vin\_kood\_ainult\_suurtahed\_ja\_numbrid CHECK (vin\_kood ~ '^([[:upper:]]|[[:digit:]])\*$'),

CONSTRAINT FK\_Auto\_Auto\_mark FOREIGN KEY (auto\_mark\_kood) REFERENCES Auto\_mark (auto\_mark\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Auto\_Auto\_kytuse\_liik FOREIGN KEY (auto\_kytuse\_liik\_kood) REFERENCES Auto\_kytuse\_liik (auto\_kytuse\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Auto\_Auto\_seisundi\_liik FOREIGN KEY (auto\_seisundi\_liik\_kood) REFERENCES Auto\_seisundi\_liik (auto\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Auto\_Tootaja FOREIGN KEY (registreerija\_id) REFERENCES Tootaja (isik\_id) ON DELETE No Action ON UPDATE No Action

)WITH (FILLFACTOR=90);

CREATE TABLE **Auto\_kategooria\_omamine**(

auto\_kood integer NOT NULL,

auto\_kategooria\_kood smallint NOT NULL,

CONSTRAINT PK\_Auto\_kategooria\_omamine PRIMARY KEY (auto\_kood,auto\_kategooria\_kood),

CONSTRAINT FK\_Auto\_kategooria\_omamine\_Auto FOREIGN KEY (auto\_kood) REFERENCES Auto (auto\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Auto\_kategooria\_omamine\_Auto\_kategooria FOREIGN KEY (auto\_kategooria\_kood) REFERENCES Auto\_kategooria (auto\_kategooria\_kood) ON DELETE No Action ON UPDATE Cascade

);

## Vaated

CREATE OR REPLACE VIEW **Aktiivsed\_ja\_mitteaktiivsed\_autod** WITH (security\_barrier) AS

SELECT Auto.auto\_kood, Auto.nimetus AS auto\_nimetus, Auto\_seisundi\_liik.nimetus AS hetke\_seisund, Auto\_mark.nimetus AS mark, Auto.mudel, Auto.valjalaske\_aasta, Auto.reg\_number, Auto.vin\_kood

FROM Auto

INNER JOIN Auto\_mark ON Auto\_mark.auto\_mark\_kood = Auto.auto\_mark\_kood

INNER JOIN Auto\_seisundi\_liik ON Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood = Auto.auto\_seisundi\_liik\_kood

WHERE ((Auto.auto\_seisundi\_liik\_kood) IN (2,3))

ORDER BY hetke\_seisund, Auto.auto\_kood;

COMMENT ON VIEW **Aktiivsed\_ja\_mitteaktiivsed\_autod** IS 'Vaade, mis kuvab aktiivsete või mitteaktiivsete autode nimekirja, kus on kood, nimetus, hetkeseisundi nimetus, mark, mudel, valjalaske\_aasta, reg\_number, vin\_kood. Vaade on mõeldud kasutamiseks juhatajale, kes soovib auto kasutamist lõpetada. Vaade realiseerib operatsiooni OP9.1.';

CREATE OR REPLACE VIEW **Autode\_kategooriate\_omamine** WITH (security\_barrier) AS

SELECT Auto\_kategooria\_omamine.auto\_kood, CONCAT(Auto\_kategooria.nimetus, ' (', Auto\_kategooria\_tyyp.nimetus, ')') AS kategooria

FROM Auto\_kategooria\_omamine

INNER JOIN Auto\_kategooria ON Auto\_kategooria\_omamine.auto\_kategooria\_kood = Auto\_kategooria.auto\_kategooria\_kood

INNER JOIN Auto\_kategooria\_tyyp ON Auto\_kategooria.auto\_kategooria\_tyyp\_kood = Auto\_kategooria\_tyyp.auto\_kategooria\_tyyp\_kood

ORDER BY kategooria;

COMMENT ON VIEW **Autode\_kategooriate\_omamine** IS 'Vaade, mis kuvab autode kategooriate ja kategooriate tüüpide nimetused (auto\_kood, kategooria\_nimetus(kategooria\_tyyp\_nimetus)). Vaade on mõeldud kasutamiseks juhatajale või autode haldurile, kes tahab mingil põhjusel vaadata autode detailseid andmeid. Vaade realiseerib operatsiooni OP2.2.';

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

SELECT Auto.auto\_kood, Auto.nimetus AS auto\_nimetus, Auto\_mark.nimetus AS mark, Auto.mudel, Auto.valjalaske\_aasta, Auto.mootori\_maht, Auto\_kytuse\_liik.nimetus AS kytuse\_liik, Auto.istekohtade\_arv, Auto.reg\_number, Auto.vin\_kood, Auto.reg\_aeg, CONCAT\_WS(' ' ,Isik.eesnimi, Isik.perenimi, Isik.e\_meil) AS registreerija, Auto\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Auto

INNER JOIN Auto\_mark ON Auto\_mark.auto\_mark\_kood = Auto.auto\_mark\_kood

INNER JOIN Auto\_kytuse\_liik ON Auto\_kytuse\_liik.auto\_kytuse\_liik\_kood = Auto.auto\_kytuse\_liik\_kood

INNER JOIN Auto\_seisundi\_liik ON Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood = Auto.auto\_seisundi\_liik\_kood

INNER JOIN Isik ON Isik.isik\_id = Auto.registreerija\_id

ORDER BY Auto.auto\_kood;

COMMENT ON VIEW **Autode\_detailid** IS 'Vaade, mis kuvab vaatamiseks mõeldud väljades auto põhiandmed (auto\_kood, nimetus, mark, mudel, valjalaske\_aasta, mootori\_maht, auto\_kütuse\_liik, istekohtade\_arv, reg\_number, vin\_kood, registreerimise aeg, registreerinud töötaja eesnimi, perenimi ja e-meili aadress, hetke\_seisund). Vaade on mõeldud kasutamiseks juhatajale või autode haldurile, kes tahab mingil põhjusel vaadata autode detailseid andmeid. Kasutatakse ka kõikide autode nimekirja kuvamiseks. Vaade realiseerib operatsioone OP8.1 ja OP8.2.';

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

SELECT Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood, UPPER(Auto\_seisundi\_liik.nimetus) AS auto\_seisundi\_liik\_nimetus, Count(Auto.auto\_kood) AS autode\_arv\_seisundis

FROM Auto\_seisundi\_liik

LEFT JOIN Auto ON Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood = Auto.auto\_seisundi\_liik\_kood

GROUP BY Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood, auto\_seisundi\_liik\_nimetus

ORDER BY autode\_arv\_seisundis DESC , auto\_seisundi\_liik\_nimetus;

COMMENT ON VIEW **Autode\_koondaruanne** IS 'Vaade, mis kuvab iga auto elutsükli seisundi kohta selle seisundi koodi, nimetuse (suurtähtedega) ja hetkel selles seisundis olevate autode arvu. Vaade on mõeldud kasutamiseks juhatajale, kes soovib sisendit juhtimisotsuste tegemiseks. Vaade realiseerib operatsiooni OP10.1.';

## Protseduursed keeled

Käesolevas projektis kasutatakse funktsioonide loomiseks PL/pgSQL keelt. PostgreSQL 10 andmebaasisüsteemis installeeritakse PL/pgSQL keel vaikimisi kõikidesse andmebaasidesse.

## Trigeri funktsioonid ja trigerid

CREATE OR REPLACE FUNCTION **TGF\_auto\_i()** RETURNS TRIGGER AS $$

BEGIN

RAISE EXCEPTION 'Uus auto saab lisada ainult algseisundiga "Ootel"';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION **TGF\_auto\_i()** IS 'Trigeri funktsioon, mis kontrollib auto lisamisel selle algseisundi korrektsust';

CREATE TRIGGER **TG\_auto\_i** BEFORE INSERT ON Auto

FOR EACH ROW WHEN (NEW.auto\_seisundi\_liik\_kood != 1) EXECUTE PROCEDURE TGF\_auto\_i();

CREATE OR REPLACE FUNCTION **TGF\_auto\_u()** RETURNS TRIGGER AS $$

BEGIN

RAISE EXCEPTION

'Seisundimuudatus pole lubatud.

Lubatud seisundimuudatused on:

Ootel => Aktiivne

Aktiivne => Mitteaktiivne

Aktiivne => Lõpetatud

Mitteaktiivne => Aktiivne

Mitteaktiivne => Lõpetatud';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION **TGF\_auto\_u()** IS 'Trigeri funktsioon, mis kontrollib autode seisundimuudatuste korrektsust';

CREATE TRIGGER **TG\_auto\_u** BEFORE UPDATE OF auto\_seisundi\_liik\_kood ON Auto

FOR EACH ROW

WHEN (NOT(

(OLD.auto\_seisundi\_liik\_kood = NEW.auto\_seisundi\_liik\_kood) OR

(OLD.auto\_seisundi\_liik\_kood = 1 AND NEW.auto\_seisundi\_liik\_kood = 2) OR

(OLD.auto\_seisundi\_liik\_kood = 2 AND NEW.auto\_seisundi\_liik\_kood = 3) OR

(OLD.auto\_seisundi\_liik\_kood = 2 AND NEW.auto\_seisundi\_liik\_kood = 4) OR

(OLD.auto\_seisundi\_liik\_kood = 3 AND NEW.auto\_seisundi\_liik\_kood = 2) OR

(OLD.auto\_seisundi\_liik\_kood = 3 AND NEW.auto\_seisundi\_liik\_kood = 4)))

EXECUTE PROCEDURE TGF\_auto\_u();

## Reeglid

Reegleid käesolevas projektis ei looda.

## Rutiinid

CREATE OR REPLACE FUNCTION **f\_aktiveeri\_auto**

**(p\_auto\_kood Auto.auto\_kood%TYPE)**

RETURNS BOOLEAN AS $$

WITH muudatus AS

(UPDATE Auto SET auto\_seisundi\_liik\_kood=2

WHERE

auto\_kood=p\_auto\_kood AND

(auto\_seisundi\_liik\_kood=1 OR auto\_seisundi\_liik\_kood=3) AND

EXISTS(

SELECT 1 FROM Auto\_kategooria\_omamine

WHERE p\_auto\_kood = auto\_kood FOR UPDATE) RETURNING auto\_kood)

SELECT Count(\*)>0 AS tulemus FROM muudatus;

$$ LANGUAGE SQL SECURITY DEFINER

SET search\_path=public, pg\_temp;

COMMENT ON FUNCTION **f\_aktiveeri\_auto(p\_auto\_kood Auto.auto\_kood%TYPE)** IS 'Selle funktsiooni abil aktiveeritakse autot, mis:

\* on registreeritud,

\* on seisundis "Ootel" või "Mitteaktiivne",

\* on määratud vähemalt ühte auto kategooriasse.

Parameetri p\_auto\_kood oodatav väärtus on aktiveeritava auto kood. Funktsioon realiseerib operatsiooni OP3.';

CREATE OR REPLACE FUNCTION **f\_muuda\_auto\_mitteaktiivseks**

**(p\_auto\_kood Auto.auto\_kood%TYPE)**

RETURNS BOOLEAN AS $$

WITH muudatus AS

(UPDATE Auto SET auto\_seisundi\_liik\_kood=3

WHERE auto\_kood=p\_auto\_kood AND auto\_seisundi\_liik\_kood=2 RETURNING auto\_kood)

SELECT Count(\*)>0 AS tulemus FROM muudatus;

$$ LANGUAGE SQL SECURITY DEFINER

SET search\_path=public, pg\_temp;

COMMENT ON FUNCTION **f\_muuda\_auto\_mitteaktiivseks(p\_auto\_kood Auto.auto\_kood%TYPE)** IS 'Selle funktsiooni abil deaktiveeritakse autot, mis on seisundis "Aktiivne". Parameetri p\_auto\_kood oodatav väärtus on aktiveeritava auto kood. Funktsioon realiseerib operatsiooni OP4.';

CREATE OR REPLACE FUNCTION **f\_lopeta\_auto**

**(p\_auto\_kood Auto.auto\_kood%TYPE)**

RETURNS Auto.auto\_kood%TYPE AS $$

UPDATE Auto SET auto\_seisundi\_liik\_kood=4 WHERE

auto\_kood=p\_auto\_kood AND auto\_seisundi\_liik\_kood IN (2,3)

RETURNING auto\_kood;

$$ LANGUAGE SQL SECURITY DEFINER

SET search\_path=public, pg\_temp;

COMMENT ON FUNCTION **f\_lopeta\_auto(p\_auto\_kood Auto.auto\_kood%TYPE)** IS 'Selle funktsiooni abil lõpetatakse autot, mis on seisundis "Aktiivne" või "Mitteaktiivne". Parameetri p\_auto\_kood oodatav väärtus on aktiveeritava auto kood. Funktsioon realiseerib operatsiooni OP5.';

## Indeksid

Järgnevalt esitatakse indeksite loomise laused.

### Välisvõtmete veergudele lisatavad indeksid

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

CREATE INDEX **IXFK\_Isik\_Riik** ON Isik (riik\_kood ASC);

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

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

CREATE INDEX **IXFK\_Tootaja\_Tootaja** ON Tootaja (mentor ASC);

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

CREATE INDEX **IXFK\_Auto\_Auto\_kytuse\_liik** ON Auto (auto\_kytuse\_liik\_kood ASC);

CREATE INDEX **IXFK\_Auto\_Auto\_mark** ON Auto (auto\_mark\_kood ASC);

CREATE INDEX **IXFK\_Auto\_Auto\_seisundi\_liik** ON Auto (auto\_seisundi\_liik\_kood ASC);

CREATE INDEX **IXFK\_Auto\_Tootaja** ON Auto (registreerija\_id ASC);

CREATE INDEX **IXFK\_Auto\_kategooria\_omamine\_Auto\_kategooria** ON Auto\_kategooria\_omamine (auto\_kategooria\_kood ASC);

### Täiendavad sekundaarsed indeksid

Täiendavad sekundaarsed indekseid käesolevas projektis ei looda

### Funktsioonil põhinevad indeksid

Tagamaks kasutajate meiliaadressite unikaalsuse sõltumata sellest, kas neis kasutatakse suur- või väiketähti ning kiirendamaks kasutajanimede tõstutundetut otsingut, loon tabeli Isik veerule e\_meil unikaalne funktsioonil Lower põhinev indeks.

CREATE UNIQUE INDEX **AK\_Isik\_e\_meil\_tostutundetud** ON Isik (Lower(e\_meil));

### Osalised indeksid

Tagamaks aktiivsete autode registrinumbrite unikaalsuse, loon tabeli Auto veerule reg\_number unikaalne indeks, mis toimib ainult aktiivsete autode puhul.

CREATE UNIQUE INDEX **AK\_Auto\_reg\_number\_aktiivne** ON Auto (reg\_number) WHERE auto\_seisundi\_liik\_kood = 2;

## Klassifikaatorite väärtustamise SQL laused

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES (1, 'Juhataja', NULL);

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES (2, 'Klienditeenindaja', NULL);

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES (3, 'Süsteemihaldur', NULL);

INSERT INTO auto\_kategooria\_tyyp (auto\_kategooria\_tyyp\_kood, nimetus) VALUES (1, 'Ruumikus');

INSERT INTO auto\_kategooria\_tyyp (auto\_kategooria\_tyyp\_kood, nimetus) VALUES (2, 'Sihtgrupp');

INSERT INTO auto\_kategooria (auto\_kategooria\_kood, nimetus, auto\_kategooria\_tyyp\_kood) VALUES (1, 'Pereauto', 1);

INSERT INTO auto\_kategooria (auto\_kategooria\_kood, nimetus, auto\_kategooria\_tyyp\_kood) VALUES (2, 'Väikeauto', 1);

INSERT INTO auto\_kategooria (auto\_kategooria\_kood, nimetus, auto\_kategooria\_tyyp\_kood) VALUES (3, 'Luksusauto', 2);

INSERT INTO auto\_kategooria (auto\_kategooria\_kood, nimetus, auto\_kategooria\_tyyp\_kood) VALUES (4, 'Minibuss', 1);

INSERT INTO auto\_kategooria (auto\_kategooria\_kood, nimetus, auto\_kategooria\_tyyp\_kood) VALUES (5, 'Kaubik', 1);

INSERT INTO auto\_kytuse\_liik (auto\_kytuse\_liik\_kood, nimetus) VALUES (1, 'Bensiin');

INSERT INTO auto\_kytuse\_liik (auto\_kytuse\_liik\_kood, nimetus) VALUES (2, 'Diisel');

INSERT INTO auto\_kytuse\_liik (auto\_kytuse\_liik\_kood, nimetus) VALUES (3, 'Gaas');

INSERT INTO auto\_kytuse\_liik (auto\_kytuse\_liik\_kood, nimetus) VALUES (4, 'Pistikhübriid');

INSERT INTO auto\_kytuse\_liik (auto\_kytuse\_liik\_kood, nimetus) VALUES (5, 'Elektriauto');

INSERT INTO auto\_mark (auto\_mark\_kood, nimetus) VALUES (1, 'Volkswagen');

INSERT INTO auto\_mark (auto\_mark\_kood, nimetus) VALUES (2, 'Opel');

INSERT INTO auto\_mark (auto\_mark\_kood, nimetus) VALUES (3, 'Nissan');

INSERT INTO auto\_mark (auto\_mark\_kood, nimetus) VALUES (4, 'Tesla');

INSERT INTO auto\_seisundi\_liik (auto\_seisundi\_liik\_kood, nimetus) VALUES (1, 'Ootel');

INSERT INTO auto\_seisundi\_liik (auto\_seisundi\_liik\_kood, nimetus) VALUES (2, 'Aktiivne');

INSERT INTO auto\_seisundi\_liik (auto\_seisundi\_liik\_kood, nimetus) VALUES (3, 'Mitteaktiivne');

INSERT INTO auto\_seisundi\_liik (auto\_seisundi\_liik\_kood, nimetus) VALUES (4, 'Lõpetatud');

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');

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, 'Mustas nimekirjas');

INSERT INTO tootaja\_seisundi\_liik (tootaja\_seisundi\_liik\_kood, nimetus) VALUES (1, 'Tööl');

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, 'Töösuhe peatatud');

INSERT INTO tootaja\_seisundi\_liik (tootaja\_seisundi\_liik\_kood, nimetus) VALUES (5, 'Vallandatud');

INSERT INTO tootaja\_seisundi\_liik (tootaja\_seisundi\_liik\_kood, nimetus) VALUES (6, 'Katseajal');

## JSON formaadis lähteandmete laadimine

Kuna järgnevates lausetes pöördutakse laienduse postgres\_fdw funktsioonide poole, siis tuleb eelnevalt see laiendus andmebaasi lisada.

CREATE EXTENSION IF NOT EXISTS postgres\_fdw WITH SCHEMA public;

CREATE SERVER minu\_testandmete\_server\_apex FOREIGN DATA WRAPPER

postgres\_fdw OPTIONS (host 'apex.ttu.ee', dbname 'testandmed',

port '5432');

--Lauseid käivitab kasutaja t192406

CREATE USER MAPPING FOR t192406 SERVER

minu\_testandmete\_server\_apex OPTIONS (user 't192406', password

'Erkionparim');

CREATE FOREIGN TABLE Isik\_jsonb (isik JSONB)

SERVER minu\_testandmete\_server\_apex;

CREATE FOREIGN TABLE Riik\_jsonb (riik JSONB)

SERVER minu\_testandmete\_server\_apex;

JSON formaadis lähteandmed lisatakse tabelite Isik ja Riik jaoks

INSERT INTO Isik(riik\_kood, isikukood, eesnimi, perenimi,

e\_meil, synni\_kp, isiku\_seisundi\_liik\_kood, parool, elukoht)

SELECT riik\_kood, isikukood, eesnimi, perenimi, e\_meil,

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\_meil,

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;

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;

Kuna järgnevates lausetes pöördutakse laienduse pgcrypto funktsioonide poole, siis tuleb eelnevalt see laiendus andmebaasi lisada.

CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;

Kasutame crypt funktsiooni, et kasutajate paroolid krüpteerida.

UPDATE Isik

SET parool = public.crypt(parool, public.gen\_salt('bf', 11));

## Täiendavate testandmete lisamine

INSERT INTO public.klient (isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (8, 1, true);

INSERT INTO public.klient (isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (9, 2, true);

INSERT INTO public.klient (isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (10, 1, false);

INSERT INTO public.klient (isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (11, 1, false);

INSERT INTO public.tootaja (isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (6, 1, 1, NULL);

INSERT INTO public.tootaja (isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (12, 3, 4, NULL);

INSERT INTO public.tootaja (isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (7, 2, 1, 6);

INSERT INTO public.tootaja (isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (11, 2, 2, 7);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (14, 'Phaeton1', 'Phaeton', 2011, '121YYY', 5, 3.0, 'WVWZZZ3CZEE075353', '2019-12-05 22:25:11', 7, 2, 1, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (15, 'Passat1', 'Passat', 2014, '955TTR', 5, 2.0, 'WVWZZZ3CZEE075372', '2019-12-05 22:27:25', 7, 2, 4, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (16, 'Passat2', 'Passat', 2014, '777HHG', 5, 1.6, 'WVWZZZ3CZEE075354', '2019-12-05 22:30:08', 7, 2, 4, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (17, 'Passat3', 'Passat', 2014, '879TRY', 5, 1.4, 'WVWZZZ3CZEE116550', '2019-12-05 22:35:45', 7, 1, 1, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (18, 'Touareg1', 'Touareg', 2010, '669UHJ', 5, 3.0, 'WVGZZZ7PZCD026979', '2019-12-05 22:38:51', 12, 2, 2, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (100, 'Astra1', 'Astra', 2012, '666ABV', 5, 1.6, 'W0L0AHL69CG047752', '2019-12-05 22:40:13', 11, 1, 1, 2);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (200, 'Tesla1', 'Model S', 2015, 'TES001', 5, 0.0, '5YJSA3H16EFP29293', '2019-12-05 22:42:51', 7, 5, 1, 3);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (11, 'Golf1', 'Golf', 2013, '123ABC', 5, 1.4, 'WVWZZZAUZGP120820', '2019-12-05 22:42:51', 7, 1, 1, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (300, 'Leaf1', 'Leaf', 2014, '849GJR', 5, 0.0, 'SJNFAAZE0U6014007', '2019-12-05 22:54:07', 11, 5, 1, 4);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (12, 'Golf2', 'Golf', 2016, '223BBC', 5, 1.4, 'WVWZZZAUZGP120828', '2019-12-05 22:21:51', 7, 1, 2, 1);

INSERT INTO public.auto (auto\_kood, nimetus, mudel, valjalaske\_aasta, reg\_number, istekohtade\_arv, mootori\_maht, vin\_kood, reg\_aeg, registreerija\_id, auto\_kytuse\_liik\_kood, auto\_seisundi\_liik\_kood, auto\_mark\_kood) VALUES (13, 'Golf3', 'Golf', 2013, '332XXA', 5, 1.6, 'WVWZZZ1KZDM649841', '2019-12-05 22:23:44', 7, 2, 3, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (12, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (13, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (15, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (17, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (100, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (11, 1);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (300, 2);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (200, 3);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (14, 3);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (18, 6);

INSERT INTO public.auto\_kategooria\_omamine (auto\_kood, auto\_kategooria\_kood) VALUES (16, 2);

## Andmebaasi statistika kogumine

Serveris on seadistatud prügikoristuse ning statistika kogumise automaatne käivitumine (autovacuum protsess).

Järgnev lause on mõeldud prügikoristuse ja statistika värskendamise erakorraliseks käivitamiseks.

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.

## Päringu täitmisplaani näide

Analüüsin päringut, mis on tehtud vaate *Aktiivsed\_ja\_mitteaktiivsed\_autod* põhjal. Joonis 15 esitab selle analüüsi tulemuse graafilisel kujul.

Vaade *Aktiivsed\_ja\_mitteaktiivsed\_autod* loomise kood on järgmine:

CREATE OR REPLACE VIEW **Aktiivsed\_ja\_mitteaktiivsed\_autod** WITH (security\_barrier) AS

SELECT Auto.auto\_kood, Auto.nimetus AS auto\_nimetus, Auto\_seisundi\_liik.nimetus AS hetke\_seisund, Auto\_mark.nimetus AS mark, Auto.mudel, Auto.valjalaske\_aasta, Auto.reg\_number, Auto.vin\_kood

FROM Auto

INNER JOIN Auto\_mark ON Auto\_mark.auto\_mark\_kood = Auto.auto\_mark\_kood

INNER JOIN Auto\_seisundi\_liik ON Auto\_seisundi\_liik.auto\_seisundi\_liik\_kood = Auto.auto\_seisundi\_liik\_kood

WHERE ((Auto.auto\_seisundi\_liik\_kood) IN (2,3))

ORDER BY hetke\_seisund, Auto.auto\_kood;

Enne täitmisplaani koostamist värskendan ANALYZE lause abil andmebaasi statistikat, et andmebaasisüsteemil oleks täitmisplaani koostamiseks võimalikult täpne sisend-informatsioon. Täitmisplaani tuleb lugeda alt üles ja seest välja (paremalt vasakule). Allpool/seespool esitatud operatsiooni tulemus on sisendiks ülalpool/väljaspool esitatud operatsioonile.

t192406=# ANALYZE;

ANALYZE

t192406=# EXPLAIN SELECT \* FROM Aktiivsed\_ja\_mitteaktiivsed\_autod;

QUERY PLAN

------------------------------------------------------------------

1 Sort (cost=3.50..3.51 rows=7 width=61)

2 Sort Key: auto\_seisundi\_liik.nimetus, auto.auto\_kood

3 -> Hash Join (cost=2.18..3.40 rows=7 width=61)

4 Hash Cond: (auto.auto\_seisundi\_liik\_kood = auto\_seisundi\_liik.auto\_seisundi\_liik\_kood)

5 -> Hash Join (cost=1.09..2.27 rows=7 width=53)

6 Hash Cond: (auto.auto\_mark\_kood = auto\_mark.auto\_mark\_kood)

7 -> Seq Scan on auto (cost=0.00..1.14 rows=7 width=48)

8 Filter: (auto\_seisundi\_liik\_kood = ANY ('{2,3}'::integer[]))

9 -> Hash (cost=1.04..1.04 rows=4 width=9)

10 -> Seq Scan on auto\_mark (cost=0.00..1.04 rows=4 width=9)

11 -> Hash (cost=1.04..1.04 rows=4 width=12)

12 -> Seq Scan on auto\_seisundi\_liik (cost=0.00..1.04 rows=4 width=12)

(12 rows)

Kuna EXPLAIN lauses ei kasutatud ANALYZE määrangut, siis uuritavat lauset tegelikult ei täidetud ning seega pole näha terve lause ning selle täitmiseks vajalike üksikute operatsioonide tegelikke täitmisaegu ning samuti plaani koostamiseks kulunud aega.

9-10 - Tabeli *Auto\_mark* täielikul läbiskaneerimisel leitakse selle kõik read. Need lähevad tabelite *Auto* ja *Auto\_mark* ühendamise sisendiks.

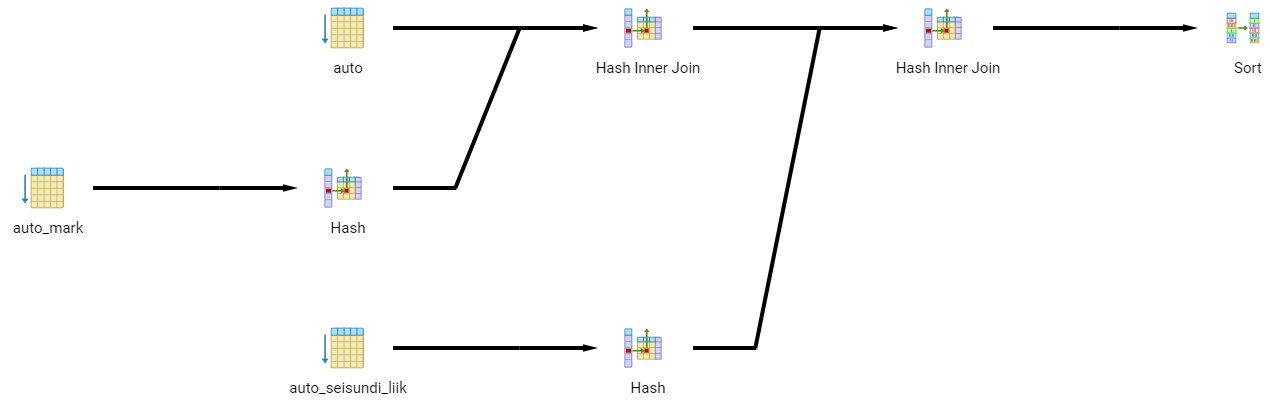
7-8 - Tabeli *Auto* täielikul läbiskaneerimisel leitakse vaid read, mis rahuldavad tingimust (auto\_seisundi\_liik\_kood = ANY ('{2,3}'::integer[])) – vaadatakse, et auto seisund oleks kas “Aktiivne” või “Mitteaktiivne”. Tabelite *Auto* ja *Auto\_mark* ühendamise operatsiooni sisendiks lähevad vaid leitud read tabelist *Auto*.

5-6 - Tabelid *Auto* ja *Auto\_mark* ühendatakse kasutades hash join algoritmi. Tabelist *Auto\_mark* loetud *auto\_mark\_kood* väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema tabelist *Auto* loetud *auto\_mark\_kood* väärtuste põhjal leitud räsiväärtustega. Selle ühendamise kontekstis on *Auto* väline tabel.

11-12 - Tabeli *Auto\_seisundi\_liik* täielikul läbiskaneerimisel leitakse selle kõik read. Need lähevad tabelite *Auto* ja *Auto\_seisundi\_liik* ühendamise sisendiks.

3-4 - Tabelite *Auto* ja *Auto\_mark* ühendamise tulemus ning *Auto\_seisundi\_liik* ühendatakse kasutades hash join algoritmi. Tabelist *Auto\_seisundi\_liik* loetud a*uto\_seisundi\_liik\_kood* väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema *Auto* ja *Auto\_mark* ühendamise tulemusena tekkinud tabelist loetud a*uto\_seisundi\_liik\_kood* väärtuste põhjal leitud räsiväärtustega. Selle ühendamise kontekstis on *Auto* ja *Auto\_mark* ühendamise tulemus väline tabel.

1-2 - Eelmises punktis nimetatud ühendamisoperatsiooni tulemus sorteeritakse kahe väärtuse alusel: auto seisundi liigi nimetus ja auto kood. Prioritiseeritakse nimetuse väärtust.



Joonis 15 Vaate *Aktiivsed\_ja\_mitteaktiivsed\_autod* põhjal tehtud päringu täitmisplaan vaadatuna visuaalselt pgAdmin programmmis.

## Rollid ja kasutajad

Käesolevas töös rolle ei looda. Luuakse rakendusele vastav kasutaja.

CREATE USER **t192406\_Autorendi\_juhataja** WITH PASSWORD 'liimatainen';

## Üleliigsete õiguste äravõtmine

Laiale avalikkusele (PUBLIC) vaikimisi antud õiguste äravõtmine.

REVOKE ALL ON DATABASE **t192406** FROM PUBLIC;

REVOKE ALL ON SCHEMA **public** FROM PUBLIC;

REVOKE USAGE ON LANGUAGE **plpgsql** FROM PUBLIC;

REVOKE USAGE ON DOMAIN **d\_nimetus** FROM PUBLIC;

REVOKE USAGE ON DOMAIN **d\_reg\_aeg** FROM PUBLIC;

REVOKE EXECUTE ON FUNCTION

**f\_aktiveeri\_auto(p\_auto\_kood Auto.auto\_kood%TYPE),**

**f\_autendi\_juhataja(p\_e\_meil text, p\_parool text),**

**f\_lopeta\_auto(p\_auto\_kood Auto.auto\_kood%TYPE),**

**f\_muuda\_auto\_mitteaktiivseks(p\_auto\_kood Auto.auto\_kood%TYPE),**

**tgf\_auto\_i(),**

**tgf\_auto\_u()**

FROM PUBLIC;

Võtan õigused ka kõigi laiendustesse kuuluvate funktsioonide suhtes. Antud projektis on need skeemis public.

REVOKE EXECUTE ON ALL FUNCTIONS IN SCHEMA **public** FROM PUBLIC;

## Õiguste jagamine

GRANT CONNECT ON DATABASE **t192406** TO t192406\_Autorendi\_juhataja;

GRANT USAGE ON SCHEMA **public** TO t192406\_Autorendi\_juhataja;

GRANT EXECUTE ON FUNCTION

**f\_lopeta\_auto(p\_auto\_kood Auto.auto\_kood%TYPE),**

**f\_autendi\_juhataja(p\_e\_meil text, p\_parool text)**

TO t192406\_Autorendi\_juhataja;

GRANT SELECT ON

**Aktiivsed\_ja\_mitteaktiivsed\_autod,**

**Autode\_kategooriate\_omamine,**

**Autode\_detailid,**

**Autode\_koondaruanne,**

**Koik\_autod**

TO t192406\_Autorendi\_juhataja;

Vaikimisi õiguste muutmine tulevikus loodavate funktsioonide jaoks. See ei mõjuta olemasolevaid funktsioone.

ALTER DEFAULT PRIVILEGES REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC;

## Andmebaasiobjektide kustutamine

Järgnevalt esitatakse andmebaasiobjektide kustutamise laused.

### Õiguste äravõtmine

Enne kasutaja kustutamist tuleb ära võtta sellele antud õigused.

REVOKE CONNECT ON DATABASE **t192406** FROM t192406\_Autorendi\_juhataja;

REVOKE USAGE ON SCHEMA **public** FROM t192406\_Autorendi\_juhataja;

REVOKE EXECUTE ON FUNCTION

**f\_lopeta\_auto(p\_auto\_kood Auto.auto\_kood%TYPE),**

**f\_autendi\_juhataja(p\_e\_meil text, p\_parool text)**

FROM t192406\_Autorendi\_juhataja;

REVOKE SELECT ON

**Aktiivsed\_ja\_mitteaktiivsed\_autod,**

**Autode\_kategooriate\_omamine,**

**Autode\_detailid,**

**Autode\_koondaruanne,**

**Koik\_autod**

FROM t192406\_Autorendi\_juhataja;

### Domeenide kustutamine

DROP DOMAIN IF EXISTS **d\_nimetus** CASCADE;

DROP DOMAIN IF EXISTS **d\_reg\_aeg** CASCADE;

### Tabelite ja arvujada generaatorite kustutamine

ALTER TABLE IF EXISTS **Auto\_kategooria** DROP CONSTRAINT IF EXISTS FK\_Auto\_kategooria\_Auto\_kategooria\_tyyp;

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

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

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

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

ALTER TABLE IF EXISTS **Tootaja** DROP CONSTRAINT IF EXISTS FK\_Tootaja\_Tootaja;

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

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

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

ALTER TABLE IF EXISTS **Auto** DROP CONSTRAINT IF EXISTS FK\_Auto\_Auto\_mark;

ALTER TABLE IF EXISTS **Auto** DROP CONSTRAINT IF EXISTS FK\_Auto\_Auto\_kytuse\_liik;

ALTER TABLE IF EXISTS **Auto** DROP CONSTRAINT IF EXISTS FK\_Auto\_Auto\_seisundi\_liik;

ALTER TABLE IF EXISTS **Auto** DROP CONSTRAINT IF EXISTS FK\_Auto\_Tootaja;

ALTER TABLE IF EXISTS **Auto\_kategooria\_omamine** DROP CONSTRAINT IF EXISTS FK\_Auto\_kategooria\_omamine\_Auto;

ALTER TABLE IF EXISTS **Auto\_kategooria\_omamine** DROP CONSTRAINT IF EXISTS FK\_Auto\_kategooria\_omamine\_Auto\_kategooria;

DROP TABLE IF EXISTS **Auto\_kategooria\_tyyp** CASCADE;

DROP TABLE IF EXISTS **Auto\_kategooria** CASCADE;

DROP TABLE IF EXISTS **Auto\_mark** CASCADE;

DROP TABLE IF EXISTS **Auto\_kytuse\_liik** CASCADE;

DROP TABLE IF EXISTS **Auto\_seisundi\_liik** CASCADE;

DROP TABLE IF EXISTS **Isiku\_seisundi\_liik** CASCADE;

DROP TABLE IF EXISTS **Kliendi\_seisundi\_liik** CASCADE;

DROP TABLE IF EXISTS **Tootaja\_seisundi\_liik** CASCADE;

DROP TABLE IF EXISTS **Riik** CASCADE;

DROP TABLE IF EXISTS **Amet** CASCADE;

DROP TABLE IF EXISTS **Isik** CASCADE;

DROP TABLE IF EXISTS **Klient** CASCADE;

DROP TABLE IF EXISTS **Tootaja** CASCADE;

DROP TABLE IF EXISTS **Auto** CASCADE;

DROP TABLE IF EXISTS **Auto\_kategooria\_omamine** CASCADE;

DROP FOREIGN TABLE IF EXISTS **Riik\_jsonb** CASCADE;

DROP FOREIGN TABLE IF EXISTS **Isik\_jsonb** CASCADE;

DROP USER MAPPING FOR t192406 SERVER minu\_testandmete\_server\_apex;

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

Kuna arvujada generaatorid loodi SERIAL notatsiooni kasutamise abil, siis kustutas andmebaasisüsteem tabelite kustutamisel automaatselt nendega seotud arvujada generaatorid.

### Vaadete kustutamine

DROP VIEW IF EXISTS **Aktiivsed\_ja\_mitteaktiivsed\_autod** CASCADE;

DROP VIEW IF EXISTS **Autode\_kategooriate\_omamine** CASCADE;

DROP VIEW IF EXISTS **Autode\_detailid** CASCADE;

DROP VIEW IF EXISTS **Autode\_koondaruanne** CASCADE;

DROP VIEW IF EXISTS **Koik\_autod** CASCADE;

### Indeksite kustutamine

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

DROP INDEX IF EXISTS **IXFK\_Isik\_Riik**;

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

DROP INDEX IF EXISTS **IXFK\_Tootaja\_Amet**;

DROP INDEX IF EXISTS **IXFK\_Tootaja\_Tootaja**;

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

DROP INDEX IF EXISTS **IXFK\_Auto\_Auto\_kytuse\_liik**;

DROP INDEX IF EXISTS **IXFK\_Auto\_Auto\_mark**;

DROP INDEX IF EXISTS **IXFK\_Auto\_Auto\_seisundi\_liik**;

DROP INDEX IF EXISTS **IXFK\_Auto\_Tootaja**;

DROP INDEX IF EXISTS **IXFK\_Auto\_kategooria\_omamine\_Auto\_kategooria**;

### Rutiinide kustutamine

DROP FUNCTION IF EXISTS **f\_aktiveeri\_auto(p\_auto\_kood Auto.auto\_kood%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_muuda\_auto\_mitteaktiivseks(p\_auto\_kood Auto.auto\_kood%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_lopeta\_auto(p\_auto\_kood Auto.auto\_kood%TYPE)** CASCADE;

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

DROP FUNCTION IF EXISTS **TGF\_auto\_i()** CASCADE;

DROP FUNCTION IF EXISTS **TGF\_auto\_u()** CASCADE;

### Trigerite kustutamine

DROP TRIGGER IF EXISTS **TG\_auto\_i** ON Auto CASCADE;

DROP TRIGGER IF EXISTS **TG\_auto\_u** ON Auto CASCADE;

### Reeglite kustutamine

Kuna projektis reegleid ei kasutata, siis ei esitata ka reeglite kustutamise lauseid.

### Kasutajate ja rollide kustutamine

DROP USER IF EXISTS **t192406\_Autorendi\_juhataja**;

Kuna projektis rolle ei kasutata, siis ei esitata ka rollide kustutamise lauseid.

### Laienduste kustutamine

DROP EXTENSION IF EXISTS **pgcrypto** CASCADE;

DROP EXTENSION IF EXISTS **postgres\_fdw** CASCADE;

# Kasutatud materjalid

1. 1autorent [WWW] <http://www.1autorent.ee/esileht> (10.03.2017)
2. AKIT. Andmekaitse ja infoturbe seletussõnastik. [WWW] <http://akit.cyber.ee/> (29.01.2017)
3. Andmebaasid I õppematerjalid. [WWW] <https://maurus.ttu.ee/> (02.09.2019)
4. Andmebaaside projekti tegemise mall. [WWW] <https://maurus.ttu.ee/> (02.09.2019)
5. AutoCheck. What is a vehicle identification number (VIN)? [WWW] <https://www.autocheck.com/vehiclehistory/autocheck/en/vinbasics> (29.08.2018)
6. Country Codes - ISO 3166 [WWW]   
   <https://www.iso.org/iso-3166-country-codes.html> (02.09.2019)
7. Chisholm, M. (2000). *Managing Reference Data in Enterprise Databases:* Binding *Corporate Data to the Wider World.* Morgan Kaufmann.
8. Eesti Statistika. Riikide ja territooriumide klassifikaator 2013v1. [WWW] <http://metaweb.stat.ee/view_xml_multi_code.htm?id=3477719&siteLanguage=ee> (29.01.2017)
9. ESTERM [WWW] <http://termin.eki.ee/esterm/> (29.01.2017)
10. Europcar [WWW] <https://www.europcar.ee/> (02.09.2019)
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. Registreerimismärgid. [WWW] <https://www.mnt.ee/et/soiduk/registreerimismargid#tab-1> (06.09.2018)
14. Schema. Car. [WWW] <https://schema.org/Car> (13.05.2017)
15. Veljovic, I., 2017. What is the maximum length of a valid email address? MoonMail, 16.02.2018 [WWW] <https://blog.moonmail.io/what-is-the-maximum-length-of-a-valid-email-address-f712c6c4bc93> (10.02.2018)
16. Vikipeedia. Auto. [WWW] <https://et.wikipedia.org/wiki/Auto> (25.05.2017)
17. Vikipeedia. Riik. [WWW] <https://et.wikipedia.org/wiki/Riik> (29.01.2017)
18. Wikipedia. Car model. [WWW] <https://en.wikipedia.org/wiki/Car_model> (29.08.2018)