TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Spordiklubi treeningute allsüsteem

Andmebaasid II, IDU0230

|  |  |
| --- | --- |
| Üliõpilane: Heidi Korp,  Jelena Kiblik |  |
| Õpperühm: IAPB |  |
| Matrikli nr: 164418, 164027 |  |
| e-posti aadress: [hekorp@ttu.ee](mailto:hekorp@ttu.ee), [jekibl@ttu.ee](mailto:jekibl@ttu.ee) |  |

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

Tallinn

2018

**Sisukord**

[**Strateegiline analüüs**](#_tyjcwt) **[8](#_tyjcwt)**

[**Terviksüsteemi üldvaade**](#_1t3h5sf) **[8](#_1t3h5sf)**

[**Organisatsiooni eesmärgid**](#_2s8eyo1) **[8](#_2s8eyo1)**

[**Infosüsteemi eesmärgid**](#_lnxbz9) **[8](#_lnxbz9)**

[**Lausendid**](#_35nkun2) **[9](#_35nkun2)**

[**Põhiobjektid**](#_1ksv4uv) **[11](#_1ksv4uv)**

[**Põhiprotsessid ja neid käivitavad sündmused**](#_44sinio) **[11](#_44sinio)**

[**Tegutsejad**](#_3j2qqm3) **[14](#_3j2qqm3)**

[**Asukohad**](#_1y810tw) **[14](#_1y810tw)**

[**Terviksüsteemi tükeldus allsüsteemideks**](#_2xcytpi) **[14](#_2xcytpi)**

[**Treeningute funktsionaalse allsüsteemi eskiismudelid**](#_qsh70q) **[16](#_qsh70q)**

[**Eesmärgid**](#_1pxezwc) **[16](#_1pxezwc)**

[**Allsüsteemi kasutavad pädevusalad**](#_2p2csry) **[17](#_2p2csry)**

[**Allsüsteemi poolt vajatavad registrid**](#_147n2zr) **[17](#_147n2zr)**

[**Allsüsteemi ühe põhiprotsessi tegevusdiagramm**](#_3o7alnk) **[18](#_3o7alnk)**

[**Allsüsteemi kasutusjuhtude eskiismudel**](#_ihv636) **[18](#_ihv636)**

[**Mittefunktsionaalsed nõuded**](#_41mghml) **[21](#_41mghml)**

[**Treeningute registri eskiismudelid**](#_3fwokq0) **[24](#_3fwokq0)**

[**Eesmärgid**](#_4f1mdlm) **[24](#_4f1mdlm)**

[**Registrit kasutavad pädevusalad**](#_19c6y18) **[24](#_19c6y18)**

[**Registrit teenindavad funktsionaalsed allsüsteemid**](#_28h4qwu) **[24](#_28h4qwu)**

[**Infovajadused, mida register aitab rahuldada**](#_37m2jsg) **[24](#_37m2jsg)**

[**Seosed teiste registritega**](#_46r0co2) **[25](#_46r0co2)**

[**Ärireeglid**](#_3l18frh) **[25](#_3l18frh)**

[**Registri kontseptuaalne eskiismudel**](#_4k668n3) **[26](#_4k668n3)**

[**Detailanalüüs**](#_1egqt2p) **[27](#_1egqt2p)**

[**Treeningute funktsionaalse allsüsteemi detailanalüüs**](#_2dlolyb) **[27](#_2dlolyb)**

[**Kasutusjuhtude mudel**](#_3cqmetx) **[27](#_3cqmetx)**

[**Treeningute funktsionaalse allsüsteemi vajatavate registrite detailanalüüs**](#_4bvk7pj) **[36](#_4bvk7pj)**

[**Kontseptuaalne andmemudel**](#_1664s55) **[36](#_1664s55)**

[**Andmebaasioperatsioonide lepingud**](#_43ky6rz) **[45](#_43ky6rz)**

[**Registri põhiobjekti seisundidiagramm**](#_3hv69ve) **[49](#_3hv69ve)**

[**CRUD maatriks**](#_2w5ecyt) **[50](#_2w5ecyt)**

[**Füüsiline disain**](#_3vac5uf) **[51](#_3vac5uf)**

[**Treeningute funktsionaalse allsüsteemi vajatavate registrite füüsiline disain**](#_pkwqa1) **[51](#_pkwqa1)**

[**Realisatsioon PostgreSQLis**](#_279ka65) **[56](#_279ka65)**

[**Andmebaasi loomine**](#_36ei31r) **[56](#_36ei31r)**

[**Skeemid**](#_1ljsd9k) **[56](#_1ljsd9k)**

[**Domeenid**](#_2koq656) **[56](#_2koq656)**

[**Tabelid ja arvujada generaatorid**](#_3jtnz0s) **[56](#_3jtnz0s)**

[**Vaated**](#_4iylrwe) **[61](#_4iylrwe)**

[**Protseduursed keeled**](#_1d96cc0) **[63](#_1d96cc0)**

[**Trigeri funktsioonid ja trigerid**](#_2ce457m) **[63](#_2ce457m)**

[**Reeglid**](#_3bj1y38) **[64](#_3bj1y38)**

[**Funktsioonid**](#_2pta16n) **[64](#_2pta16n)**

[**Indeksid**](#_3oy7u29) **[66](#_3oy7u29)**

[**Välisvõtmete veergudele lisatavad indeksid**](#_j8sehv) **[66](#_j8sehv)**

[**Täiendavad sekundaarsed indeksid**](#_1idq7dh) **[67](#_1idq7dh)**

[**Funktsioonil põhinevad indeksid**](#_2hio093) **[67](#_2hio093)**

[**Klassifikaatorite väärtustamise SQL laused**](#_1vsw3ci) **[67](#_1vsw3ci)**

[**JSON formaadis lähteandmete laadimine**](#_2uxtw84) **[69](#_2uxtw84)**

[**Täiendavate testandmete lisamine**](#_3u2rp3q) **[70](#_3u2rp3q)**

[**Andmebaasi statistika kogumine**](#_odc9jc) **[72](#_odc9jc)**

[**Päringu täitmisplaani näide**](#_1nia2ey) **[72](#_1nia2ey)**

[**Rollid ja kasutajad**](#_11si5id) **[73](#_11si5id)**

[**Üleliigsete õiguste äravõtmine**](#_20xfydz) **[74](#_20xfydz)**

[**Õiguste jagamine**](#_302dr9l) **[74](#_302dr9l)**

[**Andmebaasiobjektide kustutamine**](#_3z7bk57) **[75](#_3z7bk57)**

[**Õiguste äravõtmine**](#_thw4kt) **[75](#_thw4kt)**

[**Domeenide kustutamine**](#_1smtxgf) **[76](#_1smtxgf)**

[**Tabelite ja arvujada generaatorite kustutamine**](#_2rrrqc1) **[76](#_2rrrqc1)**

[**Vaadete kustutamine**](#_16x20ju) **[76](#_16x20ju)**

[**DROP VIEW IF EXISTS treeningute\_kategooriad CASCADE;**](#_l7a3n9) **[77](#_l7a3n9)**

[**Indeksite kustutamine**](#_356xmb2) **[77](#_356xmb2)**

[**Funktsioonide ja trigerite kustutamine**](#_44bvf6o) **[77](#_44bvf6o)**

[**Reeglite kustutamine**](#_ymfzma) **[78](#_ymfzma)**

[**Kasutajate ja rollide kustutamine**](#_1xrdshw) **[78](#_1xrdshw)**

[**Laienduste kustutamine**](#_2wwbldi) **[78](#_2wwbldi)**

[**Kasutatud materjalid**](#_2b6jogx)79

**AUTORIDEKLARATSIOON**

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

[Üliõpialase nimi]

[Allkiri]

**Sissejuhatus (Andmebaasid II)**

Tabel 1 esitab ülevaate andmebaasi ja rakenduse realisatsioonist.

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

|  |  |
| --- | --- |
| Kasutatav andmebaasisüsteem ja versioon | PostgreSQL 11.1 |
| Rakenduse loomise vahendid (arenduskeskkonnad, keeled ja nende versioonid) | MS Access 2016  pgAdmin 3.2  PostgreSQL 11.1  Enterprise Architect 12.0.1208 |
| Andmebaasi server **(andmebaas PEAB olema apex.ttu.ee serveris)** | apex.ttu.ee |
| PostgreSQL andmebaasi nimi (kui kasutate PostgreSQLi) | t164027 |
| Oracle skeemi nimi (kui kasutate Oraclet) | - |
| Millise töökoha rakendus realiseerib? | juhataja töökoht |
| Rakenduse aadress (kui on veebirakendus) **(rakenduse panek apex.ttu.ee serverisse pole kohustuslik; aadress tuleb esitada ka pgApex ja Oracle APEX rakenduse puhul)** | - |
| Rakenduse nimi (kui rakendus on tehtud pgApex või Oracle APEX keskkonnas) | - |
| Kui tegemist on kahekihilise klient-server süsteemiga ja rakendus on kasutaja arvutis, siis mida peab kasutaja arvutis tegema, et seda rakendust kasutada | Et kasutaja saaks käivitada rakendust, peab ta oma arvutisse installeerima PostgreSQL ODBC draiveri ning seadistama andmeühenduse. Seda saab teha ODBC Data Sources kaudu, kui panna paika õige pordi number, serveri aadress ja ühenduse nimi. |
| Kasutajanimi ja parool rakendusse sisselogimiseks | kasutajanimi: [ward.richard@comvoy.co.uk](mailto:ward.richard@comvoy.co.uk)  parool: incididunt |
| 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. | Aktiivsed\_mitteaktiivsed\_treeningud |
| 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. | Aktiivsed\_mitteaktiivsed\_treeningud, Koik\_treeningud, Treeningu\_detailid, Treeningu\_kategooria\_omamine, Treeningute\_koondaruanne, Treeningute\_kategooriad |
| 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. | Andmebaasisüsteemiga ühenduse loomiseks  kasutatav kasutajanimi (t164027) on  kirjutatud mooduli Refreshing algusesse  (konstandi username väärtuseks). |
| 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. | Paroolid on salvestatud räsi kujul ning kasutatud on Blowfishi krüptograafilist algoritmi. Räsi arvutame järgneval viisil: public.crypt('parool', public.gen\_salt('bf', 11)), kus 'parool' on näiteks kellegi parool, 'bf' on sool ning 11 on korduste arv, mida kasutatakse räsiväärtuse arvutamisel. Funktsiooni crypt kasutamiseks peab installeerima pgcrypto mooduli. Seda saag teha järgneva käsuga: 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 juhataja, luuakse andmebaasis funktsioon f\_on\_juhataja(text, text). Funktsiooni väljakutsel on esimene argument e-meil ja teine argument parool. |
| Kas olete oma töö iseseisva töö hindamismudeli alusel enne esitamist ise üle kontrollinud? | Ei |
| Kui vastasite eelmises punktis „Jah“, siis milline oli punktisumma? | - |

# 

# 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 spordiklubi infosüsteemist.

### 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 võimalust spordiga tegelemiseks ning soodustada tervisliku eluviisi järgimist.
* Hoida püsikliente
* Olla konkurentsivõimeline

### Infosüsteemi eesmärgid

* Tagada ülevaade organisatsiooniga seotud isikute isikuandmetest
* Tagada ülevaade organisatsiooni töötajatest
* Tagada ülevaade organisatsiooni klientidest
* Tagada ülevaade organisatsiooni juhata tööülesannetest
* Tagada ülevaade organisatsiooni kommunikatsioonijuhi tööülesannetest
* Tagada ülevaade organisatsiooni audititest
* Tagada ülevaade organisatsiooni tüüpprobleemist
* Tagada ülevaade organisatsiooni sündmustest
* Tagada ülevaade organisatsiooni kampaaniast
* Tagada ülevaade organisatsiooniga seotud teistest organisatsioonidest, sh partneritest, mis osutavad analüüsitavale organisatsioonile teenuseid
* Võimaldada klassifikaatorite abil andmete liigitamist ja seostamist seostamiseks väljaspool analüüsitava organisatsiooni vastutusala oleva informatsiooniga
* Tagada ülevaade organisatsiooni sõlmitud lepingutest
* Tagada ülevaade tegelike ja raamatupidamises arvestatud varade täpse vastavuse kindlakstegemiseks läbiviidud inventuuridest
* Tagada ülevaade organisatsiooni valduses olevatest dokumentidest
* Tagada ülevaade arvetest, mida on organisatsioonile esitatud, või mida organisatsioon on ise esitanud, sh nende arvete tasumisest
* Koguda ja analüüsida klientide tagasisidet, et võimaldada neid edaspidi veelgi paremini kohelda
* Tagada ülevaade klientide tellimustest
* Tagada ülevaadet organisatsiooni eeskirjade rikkumisega seotud intsidentidest
* Võimaldada töötajatel panustada organisatsiooni arengusse, tehes ettepanekuid tarkvara ja töökorralduse parandamise kohta
* Tagada ülevaade organisatsioonis läbiviidavatest arendustöödest
* Tagada ülevaade organisatsiooni toimise aluseks olevatest eelarvetest ja nende täitmisest
* Tagada ülevaade treeningust, millega tehingute (transaktsioonide) tegemine on üks organisatsiooni põhieesmärk
* Võimaldada vaadata kõiki hetkel saadaolevaid treeninguid
* Võimaldada kliendile registreerida end treeningule
* Võimaldada kliendile teostada makseid treeningute eest
* Tagada ülevaade kõikidest süsteemis olevatest treeningutest
* Võimaldada registreerida klienti püsikliendiks
* Tagada ülevaade organisatsiooni treeninguplaanidest
* Võimaldada lisada treeninguplaani
* Võimaldada muuta treeninguplaani
* Võimaldada kustutada treeninguplaani
* Võimaldada muuta tunniplaani
* Võimaldada lisada tunniplaani
* Võimaldada kustutada tunniplaani
* Võimaldada muuta personaalset tunniplaani
* Võimaldada lisada personaalset tunniplaani
* Võimaldada kustutada personaalset tunniplaani
* Tagada ülevaade töögraafikutest
* Võimaldada lisada töögraafikut
* Võimaldada muuta töögraafikut
* Võimaldada kustutada töögraafikut
* Tagada ülevaade tehtud tööst
* Tagada ülevaade ettekirjutustest
* Tagada ülevaade reklaamidest
* Tagada ülevaade pääsukaartide hulgast
* Tagada ülevaade trahvidest
* Võimaldada lisada trahvi
* Võimaldada muuta trahvi olekut
* Võimaldada kustutada trahvi

### Lausendid

* Töötaja on isik
* Klient on isik
* Juhataja on töötaja
* Kommunikatsioonijuht on töötaja
* Juhataja planeerib arendustööd
* Juhataja koostab eelarve
* Juhataja registreerib kliendi püsikliendiks
* Partner on organisatsioon või isik
* Spordihoone omanik on partner
* Pank on partner
* Meie organisatsioon pakub teenust
* Partneriga sõlmitakse leping
* Partner esitab arve
* Auditi koostab partnerorganisatsioon
* Intsident on varade või reeglite registreeritud rikkumine
* Tüüpprobleem on kliendi või töötaja edastatud probleem
* Sündmus on uudis, teade või ettepanek
* Kampaania on üritus
* Klassifikaatorite haldur registreerib klassifikaatori
* Meie organisatsioon esitab kliendile ja püsikliendile arve
* Arve on dokument
* Eelarve on dokument
* Klient annab tagasisidet
* Klientide tellimus sisaldab treeningpaketti
* Töötaja teeb infosüsteemi tööd puudutava ettepaneku
* Arendustöö aluseks on klientide tagasiside ja töötajate ettepanekud
* Treeningute haldur registreerib treeningu
* Treeningute haldur unustab treeningu
* Treeningute haldur muudab treeningu seisundit
* Treeningut iseloomustab null või rohkem treeningu kategooriat
* Treeningu kategooria on klassifikaator
* Uudistajale pakuvad huvi treeningute andmed
* Püsikliendi ja treeneri nõusolekul muudetakse maksulepingut
* Klient valib paketi
* Klient muudab paketti
* Klient annab paketi üle
* Klient tasub treeningu eest
* Süsteem kogub treeningutega seotud andmeid
* Treeningute haldur koostab andmete põhjal statistikat
* Treeningud toimuvad tunniplaani järgi
* Treeningud toimuvad treeninguplaani järgi
* Treeningute haldur koostab tunniplaani
* Treeningute haldur koostab treeninguteplaani
* Spordikeskus on ruum
* Treeningud toimuvad spordikeskuses
* Klient ja treener registreerivad personaalse treeningu paketi
* Treener koostab personaalse treeninguplaani
* Töögraafik on dokument
* Juhataja koostab töögraafiku
* Töötamine on tasustatud tegevus organisatsiooni huvides
* Ettekirjutus on dokument
* Ettekirjutus on loodud riiklikul või kohaliku omavalitsuse tasandil
* Reklaam on teave partnerorganisatsiooni tegevuse kohta
* Reklaami kuvamise eest teenib ettevõte kasumit
* Pääsukaart on objekt
* Pääsukaart reguleerib spordikeskuse ruumidesse sisenemist
* Spordikeskuse esindaja väljastab pääsukaardid treeningu haldurile
* Treeningu haldur edastab pääsukaardid treenerile
* Treener väljastab kaardi kliendile allkirja vastu
* Trahv on organisatsiooni reeglite rikkumise tõttu tasumisele kuuluv rahasumma
* Klient on kohustatud pääsukaardi kaotamisel tasuma spordikeskusele trahvi

### Põhiobjektid

* Isik
* Töötaja
* Juhataja
* Kommunikatsioonijuht
* Klient
* Organisatsioon
* Partner
* Audit
* Tüüpprobleem
* Sündmus
* Kampaania
* Klassifikaator
* Intsident
* Leping
* Dokument
* Arve
* Kliendi tagasiside
* Töötaja ettepanek
* Arendustöö
* Eelarve
* Treening
* Tunniplaan
* Treeninguplaan
* Kliendi tellimus
* Spordikeskus
* Personaalne treeninguplaan
* Personaalne tunniplaan
* Töögraafik
* Töötamine
* Ettekirjutus
* Reklaam
* Pääsukaart
* Trahv

### Põhiprotsessid ja neid käivitavad sündmused

Tabel 2 toob välja süsteemi mõned põhiprotsessid ning iga sellise protsessi kohta üks või rohkem sündmust, mis tingivad selle protsessi käivitumise.

**Tabel 2 Protsesside ja sündmuste vastavustabel.**

|  |  |
| --- | --- |
| **Põhiprotsess** | **Sündmused, mis selle põhiprotsessi käivitavad** |
| Isiku registreerimine | Organisatsiooni vaatevälja satub uus isik, kellega organisatsioon soovib astuda mingil viisil lepingulistesse suhetesse |
| Isiku surnuks märkimine | Organisatsioonini jõuab teave, et isik on surnud |
| Töötaja ametikoha muutmine | Töötaja liigub karjääriredelil |
| Töötaja tööle võtmine | Organisatsiooni tuleb tööle uus töötaja |
| Töötaja ajutiselt töölt vabastamine | Töötajat hakatakse kahtlustama organisatsiooni huve kahjustavas teos |
| Töötaja puhkusele siirdumine | Töötaja võtab välja kasutamata puhkuse |
| Töötaja töösuhte lõpetamine | Tööandja ja töötaja soovivad lõpetada töölepingu |
| Kliendi registreerimine | Organisatsiooni teenuseid soovib hakata kasutama uus eraklient |
| Organisatsiooni registreerimine | Organisatsiooni vaatevälja ilmub uus partnerorganisatsioon, näiteks pank või spordisaali rendile andja |
| Partneri musta nimekirja kandmine | Partner ei täida endale lepinguga võetud kohustusi |
| Klassifikaatori väärtuse lisamine | Tekib vajadus uue klassifikaatori väärtuse lisamiseks (nt tänu sellele, et täienes rahvusvaheline standard või tänu sellele, et organisatsiooni äriprotsesse otsustati muuta) |
| Klassifikaatori väärtuse muutmine | Selgus, et klassifikaatori väärtuse registreerimisel oli tehtud viga |
| Lepingu sõlmimine | Huvitatud osapool (isik või organisatsioon) soovib astuda organisatsiooniga vastastikku kasulikesse lepingulistesse suhetesse |
| Lepingu peatamine | Vähemalt üks lepingu osapooltest teatab, et ta pole ajutiselt võimeline lepingus toodud tingimusi täitma, kuid tal on huvi tulevikus lepingu täitmist jätkata |
| Lepingu ühepoolne katkestamine | Vähemalt üks lepingu osapooltest teatab, et ta pole püsivalt võimeline lepingus toodud tingimusi täitma |
| Lepingu pikendamine | Lepingu osapooled on oma lepingulise suhtega rahul ja soovivad selle pikendamist |
| Auditi koostamine | Organisatsioon soovib saada audiitori teenust |
| Dokumendi arvelevõtmine | Organisatsiooni valdusesse jõuab uus dokument (nt partner või klient saadab ametliku kirja) |
| Arve makstuks märkimine | Arve esitajale on arve täies mahus  (sh võimalikud viivised) makstud |
| Kliendi tagasiside registreerimine | Klient esitab kaebuse |
| Kliendi tellimuse registreerimine | Klient soovib osta treeningpaketti, et hakata regulaarselt treeningutel käima |
| Intsidendi registreerimine | Isik rikub organisatsiooni eeskirja vastu |
| Töötaja ettepaneku registreerimine | Töötaja näeb võimalust muuta organisatsiooni tööd efektiivsemaks, sh parandada infosüsteemi tarkvara |
| Sündmuse registreerimine | Ettevõtteni jõuab uus informatsioon, mida on vaja klientidele ja töötajatele edastada |
| Tüüpprobleemi registreerimine | Kliendid või töötajad on korduvalt maininud sama probleemi olemasolu |
| Arendustöö algatamine | Kliendi tagasiside, töötaja ettepanek või intsident annab põhjust täiendada infosüsteemi tarkvara ning võimalik, et lisaks ka töökorraldust |
| Arendustöö kinnitamine | Juhataja kiidab väljapakutud arendustöö heaks ja leiab selle täitmiseks vahendid |
| Eelarve kinnitamine | Saabub eelarve vastuvõtmise tähtaeg |
| Treeningu registreerimine | Organisatsiooni jõuab teave uue treeningu kohta |
| Treeningu unustamine | Selgus, et organisatsiooni jõudnud teave treeningu kohta on enneaegne ning sellisel kujul treeningut ei ole vaja registreerida |
| Treeningu aktiveerimine | On vaja muuta võimalikuks treeningu kasutamine tehingutes |
| Treeningu ajutiselt kasutusest eemaldamine (mitteaktiivseks muutmine) | Treeningu kasutamine tehingutes on vaja ajutiselt peatada, kuna seoses treeninguga on ilmnenud ajutise iseloomuga probleemid |
| Treeningu lõplikult kasutusest eemaldamine (lõpetamine) | Treeningu kasutamine tehingutes on vaja lõpetada, kuna seoses treeninguga on ilmnenud püsiva iseloomuga probleemid või kuna treening on oma aja lihtsalt ära elanud |

### Tegutsejad

* Juhataja (ka omanik)
* Treeningute haldur
* Klassifikaatorite haldur
* Klient
* Uudistaja
* Treener
* Partner
* Pank
* Spordikeskuse esindaja
* Kommunikatsioonijuht

### 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, kus on olemas arvuti.
* Treeningud toimuvad spordikeskuses treeneri juhendamisel.

### Terviksüsteemi tükeldus allsüsteemideks

Järgnevalt esitatakse infosüsteemi jaotus kolme erinevat liiki allsüsteemideks.

Organisatsiooni sisesed pädevusalad.

* Juhataja
* Treeningute haldur
* Klassifikaatorite haldur
* Treener
* Partnerlussuhete juht
* Kliendinõustaja
* Personalijuht
* Veebirakenduse haldur
* Raamatupidaja

Organisatsiooni välised pädevusalad.

* Klient
* Uudistaja
* Partner
* Pank
* Spordikeskus

Tabel 3 esitab sisulised funktsionaalsed allsüsteemid ja nende teenindatavad registrid (seotud organisatsiooni põhitegevusega).

**Tabel 3 Sisulised allsüsteemid.**

|  |  |
| --- | --- |
| **Funktsionaalne allsüsteem** | **Register, mida see funktsionaalne allsüsteem teenindab** |
| Treeningute funktsionaalne allsüsteem | Treeningute register |
| Tunniplaanide funktsionaalne allsüsteem | Tunniplaanide register |
| Treeninguplaanide funktsionaalne allsüsteem | Treeninguplaanide register |
| Kampaaniate funktsionaalne allsüsteem | Kampaaniate register |
| Klientide tagasiside funktsionaalne allsüsteem | Klientide tagasiside register |
| Töötajate ettepanekute funktsionaalne allsüsteem | Töötajate ettepanekute register |
| Arendustööde funktsionaalne allsüsteem | Arendustööde register |
| Auditi funktsionaalne allsüsteem | Auditi register |
| Tüüprobleemide funktsionaalne allsüsteem | Tüüprobleemide register |
| Sündmuste funktsionaalne allsüsteem | Sündmuste register |
| Töögraafikute funktsionaalne allsüsteem | Töögraafikute register |

Tabel 4 esitab administratiivsed funktsionaalsed allsüsteemid ja nende teenindatavad registrid (võivad olla kasutusel paljudes erinevate eesmärkide ja tegevusaladega organisatsioonides).

**Tabel 4 Administratiivsed allsüsteemid.**

|  |  |
| --- | --- |
| **Funktsionaalne allsüsteem** | **Register, mida see funktsionaalne allsüsteem teenindab** |
| Isikute funktsionaalne allsüsteem | Isikute register |
| Töötajate funktsionaalne allsüsteem | Töötajate register |
| Organisatsioonide funktsionaalne allsüsteem | Organisatsioonide register |
| Klientide funktsionaalne allsüsteem | Klientide register |
| Partnerite funktsionaalne allsüsteem | Partnerite register |
| Klassifikaatorite funktsionaalne allsüsteem | Klassifikaatorite register |
| Lepingute funktsionaalne allsüsteem | Lepingute register |
| Dokumentide funktsionaalne allsüsteem | Dokumentide register |
| Eelarvete funktsionaalne allsüsteem | Eelarvete register |
| Arvete funktsionaalne allsüsteem | Arvete register |
| Kliendite tagasiside funktsionaalne allsüsteem | Kliendite tagasiside register |
| Klientide tellimused | Klientide tellimuste register |
| Intsidentide funktsionaalne allsüsteem | Intsidentide register |
| Arendustööde funktsionaalne allsüsteem | Arendustööde register |
| Ettekirjutuste funktsionaalne allsüsteem | Ettekirjutuste register |
| Reklaamide funktsionaalne allsüsteem | Reklaamide  register |
| Pääsukaartide  funktsionaalne allsüsteem | Pääsukaartide  register |
| Trahvide funktsionaalne allsüsteem | Trahvide register |

## Treeningute funktsionaalse allsüsteemi eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

* Muuta võimalikuks treeningu kasutamine erinevates tehingutes (transaktsioonides), mille läbiviimist infosüsteem toetab
* Võimaldada treeningut elektrooniliselt registreerida
* Võimaldada määrata treeningu hetkeseisundit vastavalt elutsüklile
* Võimaldada muuta süsteemile teadaolevaid andmeid treeningu kohta
* Võimalik treeningu andmed kustutada ehk infosüsteemi mõttes unustada, kuid teha seda ainult siis, kui treening pole veel kordagi aktiivsesse kasutusse läinud ja seega pole sellega seotud veel ühtegi tehingut
* Võimaldada vastata fikseeritud päringutele treeningu kohta
* Anda ülevaade ja hallata treeningute detaile
* Anda ülevaade ja hallata kampaaniate registrit
* Anda ülevaade ja hallata klientide tagasiside registrit
* Anda ülevaade ja hallata klientide tellimuste registrit
* Võimaldada vaadata aktiivseid treeninguid
* Võimaldada tuvastada kasutajat
* Võimaldada vaadata treeningute koodaruannet
* Võimaldada registreerida treeningut
* Võimaldada lõpetada treeningut ehk viia see mitteaktiivsesse olekusse
* Võimaldada aktiveerida treeningut
* Võimaldada vaadata kõiki ootel või mitteaktiivseid teeninguid

### Allsüsteemi kasutavad pädevusalad

* Juhataja
* Treeningute haldur
* Klassifikaatorite haldur
* Veebirakenduse haldur
* Raamatupidaja
* Klient
* Treener
* Uudistaja

### Allsüsteemi poolt vajatavad registrid

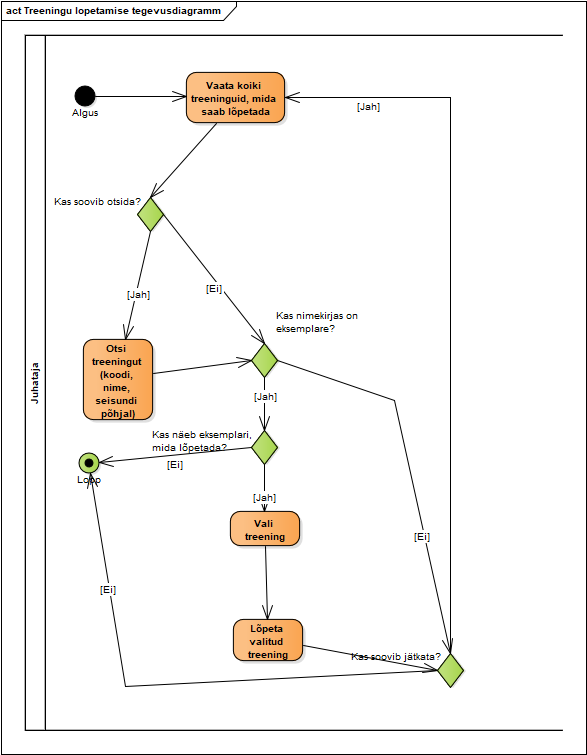
Allsüsteem teenindab treeningute registrit.

Allsüsteem loeb.

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

### Allsüsteemi ühe põhiprotsessi tegevusdiagramm

Joonis 1 esitab treeningu lõpetamise protsessi kirjelduse tegevusdiagrammina.

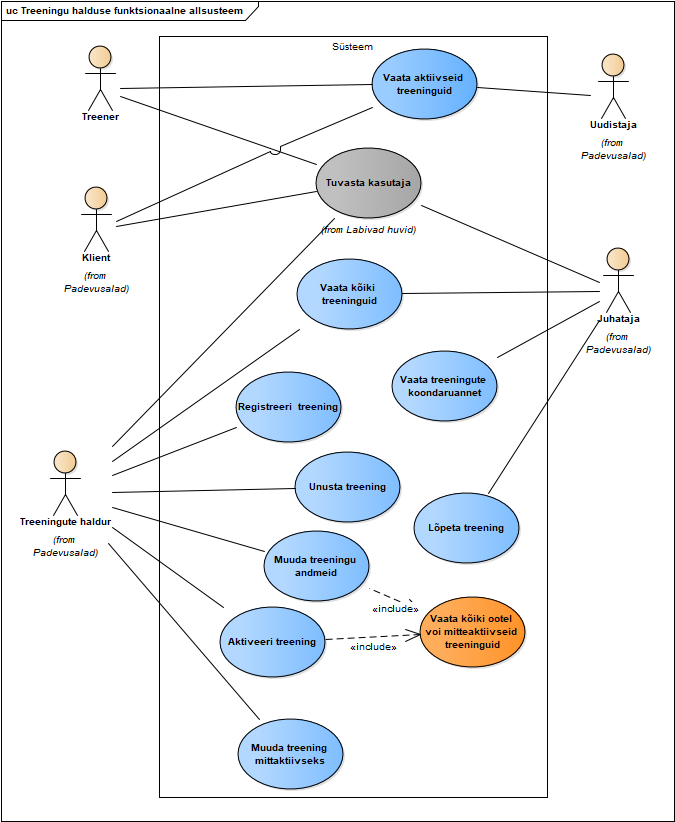


**Joonis 1 Treeningu lõpetamise tegevusdiagramm.**

### Allsüsteemi kasutusjuhtude eskiismudel

Joonis 2 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 tähistatud kasutusjuhud, mis esitavad läbivaid huvisid ning on seotud rohkem kui ühe funktsionaalse allsüsteemiga.



**Joonis 2 Treeningute funktsionaalse allsüsteemi kasutusjuhtude diagramm.**

**Kasutusjuht**: Tuvasta kasutaja

**Tegutsejad**: Treeningute haldur, Juhataja, Klient – (edaspidi Subjekt), Treener

**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 treening

**Tegutsejad**: Treeningute haldur

**Kirjeldus**: Treeningute haldur registreerib uue treeningu.

**Kasutusjuht**: Unusta treening

**Tegutsejad**: Treeningute haldur

**Kirjeldus**: Treeningute haldur vaatab ootel treeningute nimekirja, valib sealt treeningu ja kustutab selle andmebaasist. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Muuda treeningu andmeid

**Tegutsejad**: Treeningute haldur

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

**Kasutusjuht**: Aktiveeri treening

**Tegutsejad**: Treeningute haldur

**Kirjeldus**: Treeningute haldur vaatab ootel või mitteaktiivsete treeningute nimekirja, valib sealt treeningu ja muudab selle aktiivseks.

**Kasutusjuht**: Muuda treening mitteaktiivseks

**Tegutsejad**: Treeningute haldur

**Kirjeldus**: Treeningute haldur vaatab aktiivsete treeningute nimekirja, valib sealt treeningu ja muudab selle mitteaktiivseks. Subjekt saab nimekirja sorteerida ja filtreerida.

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

**Tegutsejad**: Treeningute haldur

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

**Kasutusjuht**: Vaata kõiki treeninguid

**Tegutsejad**: Treeningute haldur, Juhataja, treener – (edaspidi Subjekt)

**Kirjeldus**: Subjekt saab vaadata treeningute nimekirja. Subjekt saab nimekirja sorteerida ja filtreerida. Samuti saab ta iga treeningu korral vaadata selle kõiki detailseid andmeid, sh hetkeseisund, registreerimise aeg ja registreerinud töötaja.

**Kasutusjuht**: Lõpeta treening

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata treeningute koondaruannet

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata aktiivseid treeninguid

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

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

### Mittefunktsionaalsed nõuded

Tabel 5 esitab vaadeldava allsüsteemi mittefunktsionaalsed nõuded.

**Tabel 5 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). Treeninguga seotud transaktsioonilisi (tehingute) andmeid on rohkem (kümneid kuni sadu tuhandeid ridu), 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 Enterprise Architect. Alternatiivina võib kasutada tasuta pakutavat andmete modelleerimise vahendit DB MAIN  (<http://www.rever.eu/en/content/db-main-homepage>). Prototüübi koostamiseks kasutatakse töölaua andmebaasisüsteemi MS Access või LibreOffice Base, kuhu on integreeritud kasutajaliidese ehitamise vahendid. |
| keel | Süsteemi kasutajaliides ja dokumentatsioon peavad olema eesti keeles. Süsteem tuleks üles ehitada nii, et ei oleks väga raske lisada kasutajaliidesesse uusi keeli (inglise keel). |
| kasutajaliides | Töötavas süsteemis peab klientidele ja uudistajatele mõeldud kasutajaliides olema kindlasti veebipõhine. Töötajatele mõeldud rakendus võib olla kahekihiline, kus kasutaja arvutis on rakendus ning see suhtleb üle arvutivõrgu serveril paikneva andmebaasisüsteemiga. Soovi korral on võimalik selle jaoks MS Accessis või LibreOffice Base abil tehtud prototüüpi evolutsioneerida nii, et kasutatakse nendes loodud kasutajaliidest, kuid andmebaas on serveril.  Nõuded kasutajaliidese ülesehitusele.   * Ülesehituse põhimõtteid tuleb järjekindlalt järgida. * Rakenduses peab olema peavorm või pealehekülg, kust saab töökohaga seotud tegevuste juurde edasi liikuda. * Välisvõtme väärtuste registreerimiseks tuleb kasutada liitbokse või hüpikaknaid. * Kohustuslikud sisestusväljad tuleb tähistada (nt lisades lipikule \*). * Andmete lisamiseks ning andmete muutmiseks mõeldud väljad peavad erinevalt välja nägema (nt olema erineva taustavärviga). * Kuupäevad tuleb esitada formaadis DD.MM.YYYY * Kellaajad tuleb esitada formaadis HH24:MI * Ajatemplid tuleb esitada formaadis DD.MM.YYYY HH24:MI * Tegevused, mida süsteem saab ise teha (nt kindlaks tegema, kes andmed registreeris), peab tegema süsteem ilma kasutajalt tagasiside küsimisega tülitamata. * Kasutajaliideses ei tohi kuvada surrogaatvõtmete väärtuseid. * Kõikides olemite nimekirjades tuleb esitada selline hulk andmeid, et nende andmete alusel oleks võimalik olemeid üksteisest üheselt eristada ning et need andmed oleksid konkreetse kasutaja jaoks mõistetavad ja sisukad. * Andmete sisestamiseks ja vaatamiseks mõeldud väljade juures peab olema võimalikult arusaadavalt ja täielikult välja toodud nende andmete tähendus.   + Mõõtmistulemusi või rahasummasid esitavate atribuutide väärtuste juures tuleb esitada ühik – rahasummade puhul valuuta tähis ning mõõtmistulemuste korral mõõtühik.   + Hinna/maksumuse välja juures tuleb öelda, kas väärtus sisaldab ka maksudeks minevat osa. |
| 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*.   Treeningute funktsionaalne allsüsteem teenindab treeningute registrit, mille turvaklass on (<https://www.riigiteataja.ee/akt/13125331?leiaKehtiv>):  K2T1S2  **K2** – töökindlus – 99% (lubatud summaarne seisak nädalas ~ 2 tundi); lubatav nõutava reaktsiooniaja kasv tippkoormusel – minutid (1÷10);  **T1** – info allikas, selle muutmise ja hävitamise fakt peavad olema tuvastatavad; info õigsuse, täielikkuse ja ajakohasuse kontroll erijuhtudel ja vastavalt vajadusele;  **S2** – salajane info: info kasutamine on lubatud ainult teatud kindlatele kasutajate gruppidele, juurdepääs teabele on lubatav juurdepääsu taotleva isiku õigustatud huvi korral; |

## Treeningute registri eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

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

### Registrit kasutavad pädevusalad

* Juhataja
* Treeningute haldur
* Klassifikaatorite haldur
* Klient
* Treener

### Registrit teenindavad funktsionaalsed allsüsteemid

Treeningute registrit teenindab (loeb ja muudab) treeningute funktsionaalne allsüsteem.

### Infovajadused, mida register aitab rahuldada

* Ootel treeningute nimekiri, kus on vähemalt treeningu kood.
* Aktiivsete treeningute nimekiri, kus on vähemalt treeningu kood.
* Ootel või mitteaktiivsete treeningute nimekiri, kus on vähemalt treeningu kood ja seisundi nimetus. Võib olla ka treeningu nimetus.
* Aktiivsete või mitteaktiivsete treeningute nimekiri, kus on vähemalt treeningu kood ja seisundi nimetus. Võib olla ka treeningu nimetus.
* Kõikide treeningute nimekiri, kus on vähemalt treeningu kood ja seisundi nimetus. Võib olla ka treeningu nimetus.
* Treeningu detailandmed, kus seotud klassifikaatorite väärtuste koodide asemel on nimetused ning esitatakse info ka treeningu registreerinud töötaja kohta (eesnimi, perenimi, meiliaadress).
* Iga treeningu seisundi kohta kõigi selles seisundis olevate treeningute arv.

### Seosed teiste registritega

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

**Klassifikaatorite register** – Klassifikaatorite registriga on treening seotud olemitüübi treeningu\_seisundi\_liik kaudu. Selle abil registreeritakse treeningu hetkeseisund. Samuti on iga treeninguga seotud null või rohkem treeningu kategooriat, mis on samuti klassifikaator.

Selleks, et saaks registreerida andmeid treeningust sõltuvates registrites, näiteks tunni- ja treeninguplaanides, peavad olema registreeritud treeningu andmed ja seega peab olema realiseeritud treeningu register.

Selleks, et saaks koostada treeningpaketti, mida klient saaks tellida ning mille eest saab firma arvet nõuda, peavad olema registreeritud vastavad treeningud ning treeningute andmed.

### Ärireeglid

Jõustatavad treeningu registri põhjal

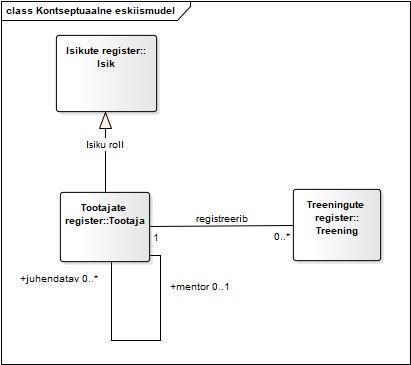
* Igal treeningul on unikaalne kood
* Iga treening on käesoleval ajahetkel täpselt ühes seisundis vastavalt oma elutsüklile.
* Iga treeninguga on seotud null või rohkem kategooriat
* Iga treeningu ja iga kategooria tüübi vahel saab olla maksimaalselt üks seos
* Iga treeningu puhul on vaja registreerida töötaja, kes treeningu andmed registreeris ning treeningu registreerimise aeg. Neid andmeid ei tohi tagantjärgi muuta
* Treeningute andmeid (sh treeningu kategooriasse kuulumine) (v.a seisund) saab muuta vaid siis, kui see on ootel või mitteaktiivses seisundis
* Treeningute andmete muutmisel ei saa muuta seda registreerinud töötajat ja registreerimise aega
* Treeningute andmeid saab andmebaasist kustutada vaid siis, kui see on ootel seisundis
* Treeninguid saab aktiveerida vaid siis, kui see on seotud vähemalt ühe treeningu kategooriaga
* Igal treeningul on nimi (lühikirjeldus), ei tohi olla null
* Iga aktiivse treeninguga peab olema seotud vähemalt üks treener
* Treeningul võib treener puududa vaid juhul, kui treening on mitteaktiivses või ootel seisundis

Jõustatavad teiste registrite põhjal, kuid vajalikud treeningute funktsionaalse allsüsteemi toimimiseks

* Iga isiku kasutajanimena kasutatakse tema unikaalset meiliaadressi
* 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.
* Iga isik, kes muudab või haldab treeningutega seotud andmeid, on süsteemi registreeritud organisatsiooni töötajana
* Igal isikul peab olema vähemalt üks kahest - eesnimi ja perenimi peavad olema registreeritud.
* Sünnikuupäev ei tohi olla suurem kui registreerimise aeg.

### Registri kontseptuaalne eskiismudel

Joonis 3 esitab esimese versiooni treeningute registri kontseptuaalse andmemudeli olemisuhte diagrammist.



**Joonis 3 Treeningute registri kontseptuaalne eskiismudel.**

# Detailanalüüs

Selles peatükis kirjeldatakse detailselt ja mittetehniliselt funktsionaalse allsüsteemi/registri paari, mille eskiismudelid esitati strateegilise analüüsi dokumendis. Registrite vaadet on laiendatud kõigi registritega, mida treeningute funktsionaalne allsüsteem vajab toimimiseks.

## Treeningute funktsionaalse allsüsteemi detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt treeningute funktsionaalse allsüsteemi toimimist.

### Kasutusjuhtude mudel

Treeningute 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**: Treeningute haldur, Juhataja, Treener, Klient – (edaspidi Subjekt).

**Osapooled ja nende huvid**:

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

*Subjekt võib üritada süsteemi siseneda kuni kolm korda.*

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

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

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

**Kasutusjuht:** Registreeri treening

**Primaarne tegutseja**: Treeningute haldur

**Osapooled ja nende huvid**:

* Treeningute haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate treeningute 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 treeningutest ning uue treeningu tekkimisel ei tohi selle registreerimisega viivitada.
* Klient, Uudistaja: Soovivad võimalikult täpset infot treeningute kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga treeninguid kasutava kliendi rollis.

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

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

**Järeltingimused**: Treening on registreeritud ja treening on seisundis „Ootel“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

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

3a Treeningute haldur soovib treeningu mõnest määratud kategooriast kohe eemaldada.

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

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

**Kasutusjuht:** Unusta treening

**Primaarne tegutseja**: Treeningute haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Treeningute haldur on autenditud ja autoriseeritud. Treening on registreeritud ja on seisundis „Ootel“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Treeningute haldur avaldab soovi treening unustada, st selle andmed süsteemist kustutada.
2. **Süsteem** kuvab ootel treeningute nimekirja, kus on kood ja reg. aeg.(**OP3.1**)
3. Treeningute haldur valib nimekirjast treeningu ja annab korralduse see unustada.
4. **Süsteem** salvestab andmed (**OP2**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi ootel treeningut, siis ei saa treeningute haldur jätkata.

**Kasutusjuht:** Muuda treeningu andmeid

**Primaarne tegutseja**: Treeningute haldur

**Osapooled ja nende huvid**:

* Treeningute haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate treeningute 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 treeningutest.
* Klient, Uudistaja: Soovivad võimalikult täpset infot treeningu kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga treeningut kasutava kliendi rollis.

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Treeningute haldur soovib muuta treeningu andmeid.
2. *Käivitub kasutusjuht „Vaata kõiki ootel või mitteaktiivseid treeninguid“*
3. Treeningute haldur valib nimekirjast treeningu ja annab korralduse vaadata selle detailseid andmeid.
4. **Süsteem** kuvab muutmiseks mõeldud väljades info kõigi treeningute ja selle võimalike alamtüüpide muudetavate omaduste ning seoste kohta. Muuta ei ole võimalik treeningu seisundit, registreerijat ning registreerimise aega. Muuhulgas kuvab süsteem muutmiseks treeningu põhiandmed (treeningu\_kood, nimetus, kestus, reg. aeg.) (**OP4.1**) ning sellega seotud kategooriate ja kategooriate tüüpide nimetused (**OP2.2**). Seal on muuhulgas võimalik määrata, millistesse kategooriatesse treening kuulub, sest süsteem pakub kategooriate valiku (**OP2.1**).
5. Treeningute haldur muudab andmeid ja annab korralduse salvestada.
6. **Süsteem** salvestab andmed (**OP6**).

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

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

5a. Treeningute haldur võib lisada treeningu uude kategooriasse ja anda korralduse salvestada.

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

5b Treeningute haldur võib eemaldada treeningu kategooriast ja anda korralduse salvestada.

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

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

**Kasutusjuht:** Aktiveeri treening

**Primaarne tegutseja**: Treeningute haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Treeningute haldur on autenditud ja autoriseeritud. Treening on registreeritud ja on seisundis „Ootel“ või „Mitteaktiivne“. Treening on määratud vähemalt ühte treeningu kategooriasse.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

*Treeningute 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 treeningut, siis ei saa treeningute haldur jätkata.

4a. Kui treening ei kuulu ühtegi treeningu kategooriasse, siis aktiveerimine ebaõnnestub.

**Kasutusjuht**: Muuda treening mitteaktiivseks

**Primaarne tegutseja**: Treeningute haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Treeningute haldur on autenditud ja autoriseeritud. Treening on registreeritud ja on seisundis „Aktiivne“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Treeningute haldur avaldab soovi treening mitteaktiivseks muuta.
2. **Süsteem** kuvab aktiivsete treeningute nimekirja, kus on kood, reg. aeg.(**OP6.1**)
3. Treeningute haldur valib nimekirjast treeningu ja annab korralduse see mitteaktiivseks muuta.
4. **Süsteem** salvestab andmed (**OP4**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi aktiivset treeningut, siis ei saa treeningute haldur jätkata.

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

**Primaarne tegutseja**: Treeningute haldur.

**Osapooled ja nende huvid**:

* Treeningute haldur: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata ootel või mitteaktiivsete treeningute nimekirja
2. **Süsteem** kuvab ootel või mitteaktiivses seisundis treeningute nimekirja, kus on kood, hetkeseisundi nimetus. (**OP7.1**)

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

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

**Kasutusjuht**: Vaata kõiki treeninguid

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

**Osapooled ja nende huvid**:

* Juhataja, treeningute haldur, treener: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata kõikide treeningute andmeid.
2. **Süsteem** kuvab kõigi treeningute nimekirja, kus on kood, hetkeseisundi nimetus. (**OP8.1**)
3. Subjekt valib treeningu, mida ta soovib detailsemalt vaadata.
4. **Süsteem** kuvab vaatamiseks mõeldud väljades andmed kõigi treeningute ja selle võimalike alamtüüpide omaduste ning seoste kohta. Muuhulgas kuvab süsteem treeningute põhiandmed (treeningu\_kood, registreerimise aeg, registreerinud töötaja eesnimi, perenimi ja meiliaadress) (**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 treeningut, siis ei saa subjekt jätkata.

**Kasutusjuht**: Lõpeta treening

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

**Käivitav sündmus**: Treeningu kasutamine tehingutes on vaja püsivalt lõpetada, kuna seoses treeninguga on ilmnenud püsiva iseloomuga probleemid või kuna treening on oma aja lihtsalt ära elanud

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Juhataja avaldab soovi treening lõpetada.
2. **Süsteem** kuvab aktiivsete või mitteaktiivsete treeningute nimekirja, kus on treeningu kood, treeningu nimetus, hetkeseisundi nimetus. (**OP9.1**)
3. Juhataja valib nimekirjast treeningu 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 treeningut, siis ei saa juhataja jätkata.

**Kasutusjuht**: Vaata treeningute koondaruannet

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

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

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

**Järeltingimused**: Treeningute koondaruanne on moodustatud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

**Kasutusjuht**: Vaata aktiivseid treeninguid

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

**Osapooled ja nende huvid**:

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib näha kõiki organisatsiooni pakutavaid aktiivseid treeninguid.
2. **Süsteem** kuvab nimekirja kategooriatest (**OP2.1**)
3. Subjekt valib konkreetse kategooria.
4. **Süsteem** kuvab sellesse kuuluvate aktiivsete treeningute nimekirja. Iga treeningu kohta esitatakse kood ja reg. aeg.(**OP11.2**).
5. Subjekt valib treeningu, mida ta soovib detailsemalt vaadata.
6. **Süsteem** kuvab vaatamiseks mõeldud väljades andmed kõigi treeningute ja selle võimalike alamtüüpide omaduste ning seoste kohta, v.a treeningu hetkeseisund, registreerimise aeg ja registreerija. Muuhulgas kuvab süsteem treeningu põhiandmed (treeningu\_kood, nimetus, kestus, reg. aeg) (**OP11.3**) ning sellega seotud kategooriate ja kategooriate tüüpide nimetused (**OP2.2**).

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

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

4b. Subjekt võib vaadatavate treeningute hulka nimekirjas näidatavate andmete järgi sorteerida ning filtreerida.

## Treeningute funktsionaalse allsüsteemi vajatavate registrite detailanalüüs

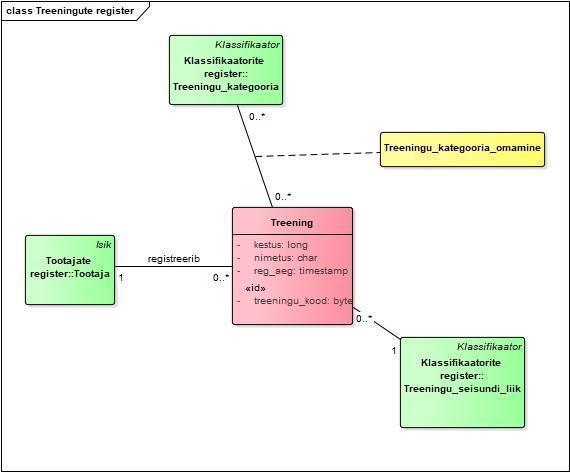
Järgnevalt kirjeldatakse detailselt ja mittetehniliselt treeningute funktsionaalse allsüsteemi vajatavate registrite struktuuri ja toimimist.

### Kontseptuaalne andmemudel

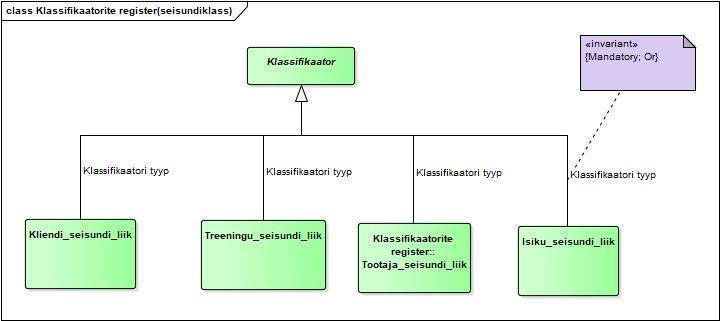
Järgnevalt esitatakse kontseptuaalne andmemudel, mis koosneb olemisuhte diagrammidest ja nendel olevate olemitüüpide ja atribuutide sõnalistest kirjeldustest.

Joonis 4esitatud olemisuhte diagrammidel on värvidel järgmine tähendus.

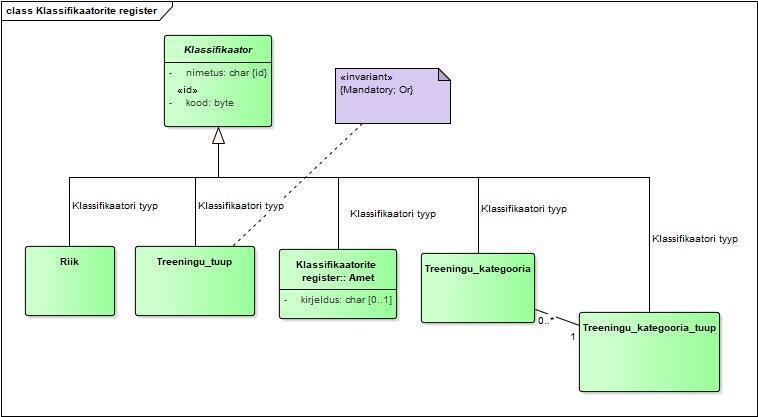
* **Punasega** on tähistatud *treeningute registri* põhiobjekt.
* Kollasega on tähistatud *treeningute registrisse* kuuluvad mitte-põhiobjektid.
* **Rohelisega** on tähistatud teistesse registritesse kuuluvad objektid, mida on antud juhul vaja treeningute funktsionaalse allsüsteemi toimimise tagamiseks.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

****

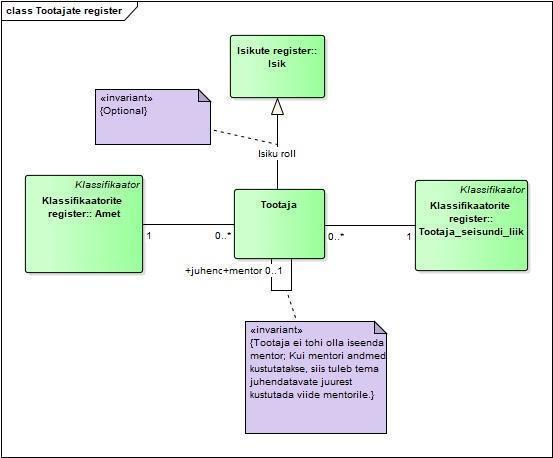
**Joonis 4 Treeningute registri olemi-suhte diagrammid**

****

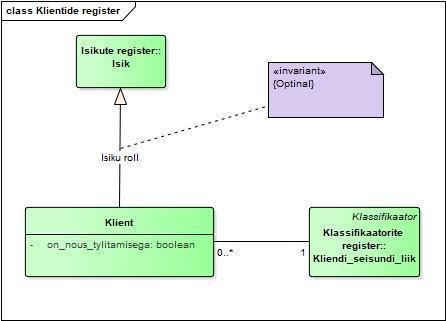
**Joonis 5 Klassifikaatorite registri(seisundiklass) olemi-suhte diagrammid.**

****

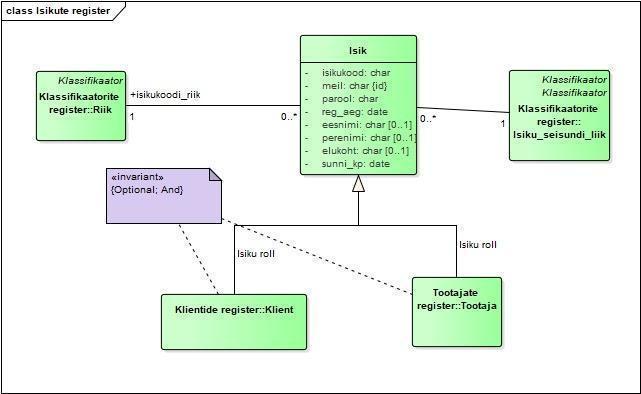
**Joonis 6 Klassifikaatorite registri olemi-suhte diagrammid.**

****

**Joonis 7 Töötajate registri olemi-suhte diagrammid.**

****

**Joonis 8 Klientide registri olemi-suhte diagrammid.**

****

**Joonis 9 Isikute registri olemi-suhte diagrammid.**

Tabel 6 esitab olemi-suhte diagrammidel esitatud olemitüüpide sõnalised kirjeldused.

**Tabel 6 Olemitüüpide sõnalised kirjeldused.**

|  |  |  |
| --- | --- | --- |
| **Olemitüübi nimi**  **(teised nimed)** | **Kuuluvus registrisse** | **Definitsioon** |
| Amet | Klassifikaatorite register | Amet on töölepingus sätestatud ametikohustuste üldnimetus. Ametid on klassifikaatorid. |
| 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. |
| Kliendi\_seisundi\_liik | Klassifikaatorite register | Seisundiklassifikaator, mis võimaldab fikseerida iga kliendi puhul tema hetkeseisundi vastavalt üldisele kliendi elutsüklile. |
| Klient | Klientide register | Eraisik, kes kasutab spordiklubi teenuseid. |
| 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) |
| 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. |
| Treeningu\_kategooria | Klassifikaatorite register | Võimaldab treeningute klassifitseerimist erinevatesse kategooriatesse ja selle alusel treeningute rühmitamist teatud põhjusel huvipakkuvateks hulkadeks. Tegemist on üksteist mittevälistavate kategooriatega, st üks ja sama treening võib kuuluda korraga mitmesse sama tüüpi kategooriasse. |
| Treeningu\_kategooria\_tüüp | Klassifikaatorite register | Võimaldab rühmitada treeningute klassifitseerimiseks kasutatavaid kategooriaid ühise nime alla. Need nimed kirjeldavad, mis liiki klassifikatsiooniga on tegemist. Iga treeningu ja iga treeningu tüübi vahel on maksimaalselt üks seos. |
| Treening | Treeningute register | Treening on treeningute halduri loodud ainekaart, mille eesmärk on treeningu tutvustamine eelkõige klientidele. |
| Treeningu\_kategooria\_  omamine | Treeningute register | Näitab treeningu kuulumist kategooriatesse. Iga treeningu ja iga treeningu kategooria vahel võib olla üks või mitu seost. |

Tabel 7 esitab atribuutide sõnalised kirjeldused.

**Tabel 7 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 |
| 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 | 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 ja /. Isikukood ei tohi olla tühi string või ainult tühikutest koosnev string}** | 39204010231 |
| Isik | meil (meil, 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 tõstutundetu 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.**  **meil peab sisaldama täpselt ühte "@" märki. Võib olla kuni 254 märki pikk.}** | kalamees@hot.ee |
| Isik | parool | Isiku identsust tõendav teadmuslik (miski, mida isik teab) volitustõend. Andmebaasis salvestatakse parooli ja soola põhjal leitud räsiväärtus.  **{Registreerimine on kohustuslik.}** | $2a$11$FsKdoFDJePwuYtyg2hBxz.e8AwSODaO/nFGGacEm05vIgOBNG9dHC |
| Isik | 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 | 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).}päeva kuupäev kellaajaga (otspunktid kaasa arvatud)}** | 12.08.2014 17:01:05 |
| 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 |
| 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 treeningu\_kategooria nimetus, mis peab olema unikaalne kombinatsioonis treeningu\_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, kas klient on või ei ole nõus saama treeningettevõtte uudiseid ja pakkumisi meilile **{Vaikimisi väärtus on false}** | true |
| Treening | kestus | Arvuline väärtus, mis kirjeldab ühe treeningu kestust (minutites).  **{Registreerimine kohustuslik. Väärtused võivad olla vahemikus 30 kuni 90.}** | 60 |
| Treening | nimetus | Treeningu nimetus sisaldab tavaliselt spordiala, millega treeningu jooksul tegeletakse.  **{On unikaalne. Registreerimine on kohustuslik. Nimetus ei tohi olla tühi string või koosneda ainult tühikutest.}** | 400 m jooks |
| Treening | reg\_aeg | X 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 12:33:04 |
| Treening | treeningu\_kood | Treeningu arvuline kood, mis sisestatakse inimkasutaja poolt, mitte ei genereerita süsteemi poolt.  **{Treeningu unikaalne identifikaator. Registreerimine on kohustuslik}** | 222 |

### Andmebaasioperatsioonide lepingud

**OP1 Registreeri Treening(p\_treeningu\_kood, p\_nimetus, p\_kestus, töötaja identifikaator, treeningu tüübi identifikaator)**

**Eeltingimused**:

* Treeningu\_seisundi\_liik eksemplar tsl (millel on nimetus="Ootel") on registreeritud
* Treeningu\_tüüp eksemplar pt (millel on treeningu tüübi identifikaator) on registreeritud
* Töötaja eksemplar tt (millel on töötaja identifikaator) on registreeritud

**Järeltingimused**:

--Loo eksemplare

* Treeningu eksemplar t on registreeritud

--Väärtusta atribuute

* t.treeningu\_kood:= p\_treeningu\_kood
* t.reg\_aeg:= hetkekuupäev + kellaaeg
* t.nimetus:= p\_nimetus
* t.kestus:= p\_kestus

--Loo seoseid

* t ja tsl seos on registreeritud
* t ja pt seos on registreeritud
* t ja tt seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri treening

**OP2 Unusta treening(p\_treening\_kood)**

**Eeltingimused**:

* Treening eksemplar T (millel on treening\_kood=p\_treening\_kood) on registreeritud
* t on seotud Treeningu\_seisundi\_liik eksemplariga tsl (millel on nimetus="Ootel")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

* t, kõik sellega otseselt või kaudselt seotud alamtüüpi olemid ja kõigi nende olemite seosed on andmebaasist kustutatud
* kõik t-ga seotud Treeningu\_kategooria\_omamine eksemplarid ja nende eksemplaride seosed on andmebaasist kustutatud

**Kasutus kasutusjuhtude poolt**: Unusta treening

**OP3 Aktiveeri treening (p\_treeningu\_kood)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_treeningu\_kood) on registreeritud
* t on seotud treningu\_seisundi\_liik eksemplariga tsl\_vana (nimetus="Ootel") või (nimetus="Mitteaktiivne")
* Treeningu\_seisundi\_liik eksemplar tsl\_uus (millel on nimetus="Aktiivne") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

* t ja tsl\_vana seos on kustutatud

--Loo seoseid

* t ja tsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Aktiveeri treening

**OP4 Muuda treening mitteaktiivseks treeninguks(p\_treeningu\_kood)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_treeningu\_kood) on registreeritud
* t on seotud treeningu\_seisundi\_liik eksemplariga tsl\_vana (millel on nimetus="Aktiivne")
* Treeningu\_seisundi\_liik eksemplar tsl\_uus (millel on nimetus="Mitteaktiivne") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

* t ja tsl\_vana seos on kustutatud

--Loo seoseid

* t ja tsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda treening mitteaktiivseks

**OP5 Lõpeta treening(p\_treeningu\_kood)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_ treeningu\_kood) on registreeritud
* t on seotud treeningu\_seisundi\_liik eksemplariga tsl\_vana ((millel on nimetus="Aktiivne") või (millel on nimetus="Mitteaktiivne"))
* Treeningu\_seisundi\_liik eksemplar tsl\_uus (millel on nimetus="Lõpetatud") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

* t ja tsl\_vana seos on kustutatud

--Loo seoseid

* t ja tsl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Lõpeta treening

**OP6 Muuda treeningu andmeid (p\_treeningu\_kood, p\_nimetus, p\_kestus, treeningu tüübi identifikaator)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_ treeningu\_kood\_vana) on registreeritud
* t on seotud treeningu\_seisundi\_liik eksemplariga tsl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))
* Treeningu\_tüüp eksemplar pt (millel on treeningu tüübi identifikaator) on registreeritud

**Järeltingimused**:

--Väärtusta atribuute

* t.nimetus:= p\_treeningu\_nimetus\_uus
* t.treeningu\_kood:= p\_treeningu\_kood\_uus 3
* t.kestus:= p\_kestus

--Kustuta seoseid

* t olemasolev seos treeningu tüübiga on kustutatud

--Loo seoseid

* t ja pt seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda treeningu andmeid

**OP7 Lisa treeningu kategooriasse (p\_treeningu\_kood, treeningu kategooria identifikaator)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_treeningu\_kood) on registreeritud
* Treeningu\_kategooria eksemplar tk (millel on treeningu kategooria identifikaator) on registreeritud
* t on seotud treeningu\_seisundi\_liik eksemplariga tsl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))

**Järeltingimused**:

--Loo eksemplare

* treeningu\_kategooria\_omamine eksemplar tko on registreeritud

--Loo seoseid

* t ja tko seos on registreeritud
* tk ja tko seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri treening, Muuda treeningu andmeid

**OP8 Eemalda treening kategooriast (p\_treeningu\_kood, treeningu kategooria identifikaator)**

**Eeltingimused**:

* Treeningu eksemplar t (millel on treeningu\_kood=p\_treeningu\_kood) on registreeritud
* Treeningu\_kategooria eksemplar tk (millel on treeningu kategooria identifikaator) on registreeritud
* t on seotud treeningu\_seisundi\_liik eksemplariga tsl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))

**Järeltingimused**:

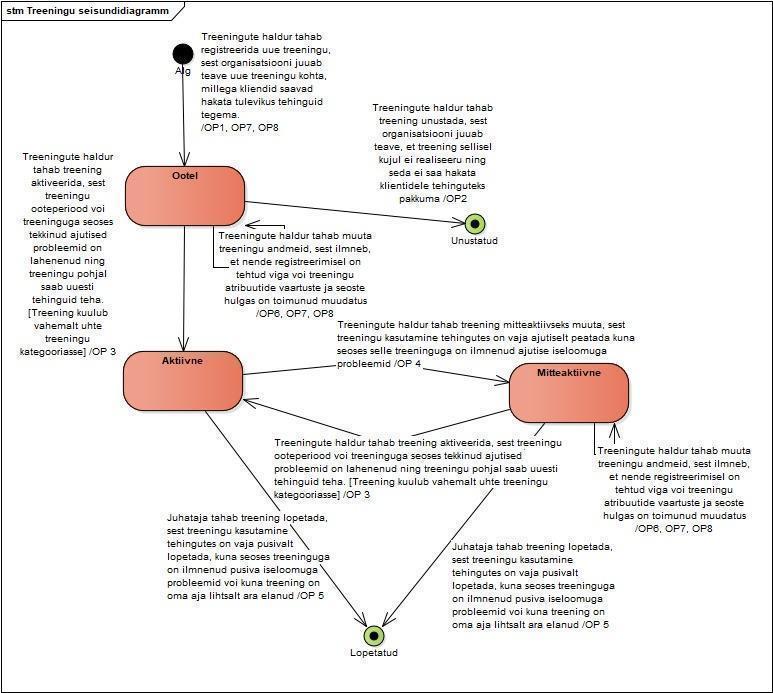
--Kustuta eksemplare ja seoseid

* treeningu\_kategooria\_omamine eksemplar tko, mis on seotud t-ga ja mis on seotud tk-ga, on koos oma seostega kustutatud

**Kasutus kasutusjuhtude poolt**: Registreeri treening, Muuda treeningu andmeid

### Registri põhiobjekti seisundidiagramm

Joonis 10 esitab seisundidiagrammi, mis kirjeldab registri põhiobjekti treeningu kõikvõimalikke elutsükleid*.*



**Joonis 10 Treeningu seisundidiagramm**

## CRUD maatriks

Tabel 8 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 8 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 |  |  |  |  |  |  |  |  |  |  |  |  |
| Amet | R | R |  |  |  |  |  |  |  |  |  | R |
| Isiku\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Töötaja\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Treeningu\_kategooria |  | R |  | R |  |  |  | R |  |  | R | R |
| Treeningu\_kategooria\_tüüp |  | R |  | R |  |  |  | R |  |  | R | R |
| treeningu\_seisundi\_liik |  | R | R | R | R | R | R | R | R | R | R | R |
| Isik | R |  |  |  |  |  |  | R |  |  |  | R |
| Töötaja | R | R |  |  |  |  |  | R |  |  |  | R |
| Treeningu\_kategooria\_omamine |  | CRD | D | CRD |  |  |  |  | R |  | R | CRD |
| Treening |  | C | RD | RU | RU | RU | R | R | RU | R | R | CRU |

1 – Tuvasta kasutaja

2 – Registreeri treening

3 – Unusta treening

4 – Muuda treeningu andmeid

5 – Aktiveeri treening

6 – Muuda treening mitteaktiivseks

7 – Vaata kõiki ootel või mitteaktiivseid treeninguid

8 – Vaata kõiki treeninguid

9 – Lõpeta treening

10 – Vaata treeningute koondaruannet

11 – Vaata aktiivseid treeninguid

# Füüsiline disain

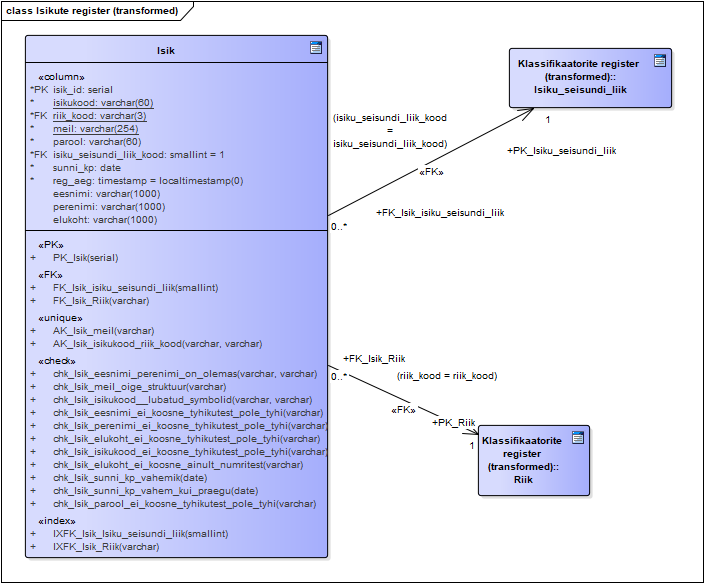
Selles peatükis esitatakse mudel, mis kirjeldab treeningu funktsionaalse allsüsteemi toimimiseks vajalike registrite tehnilist lahendust treeningute andmebaasisüsteemis.

## Treeningute funktsionaalse allsüsteemi vajatavate registrite füüsiline disain

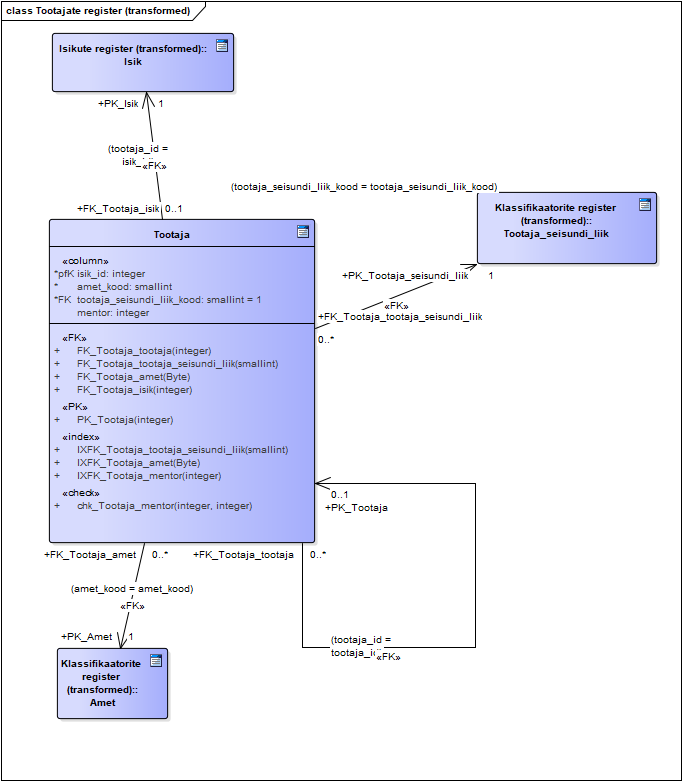
Joonis 11-Joonis 16 esitavad vastuvõtuaegade funktsionaalset allsüsteemi vajatavate registrite füüsilise disaini andmebaasisüsteemi PostgreSQL jaoks. MS Accessi korral tuleb arvestada järgnevate erinevustega. Kuigi CHECK kitsendusi saab luua ka MS Accessi andmebaasis, siis prototüübis realiseeritakse need veeru või tabeliga seotud valideerimisreegli abil.

Tabel 9 Andmetüüpide erinevusi PostgreSQLis ja MS Accessis.

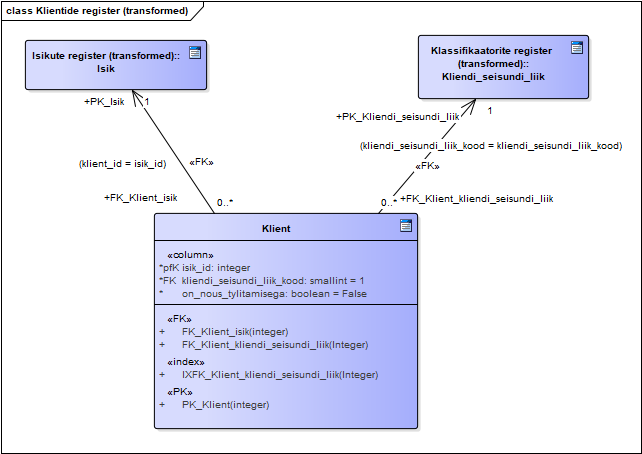
|  |  |
| --- | --- |
| Andmetüüp ja väljapikkus PostgreSQLis | Andmetüüp ja väljapikkus MS Accessis |
| SERIAL | AUTOINCREMENT |
| VARCHAR(1000) TEXT | VARCHAR(255) (255 on maksimaalne võimalik väljapikkus VARCHAR korral) |
| BYTEA | ATTACHMENT (seda ei saa määrata SQL lausetes, vaid tuleb määrata tabeli graafilises disaineris) |



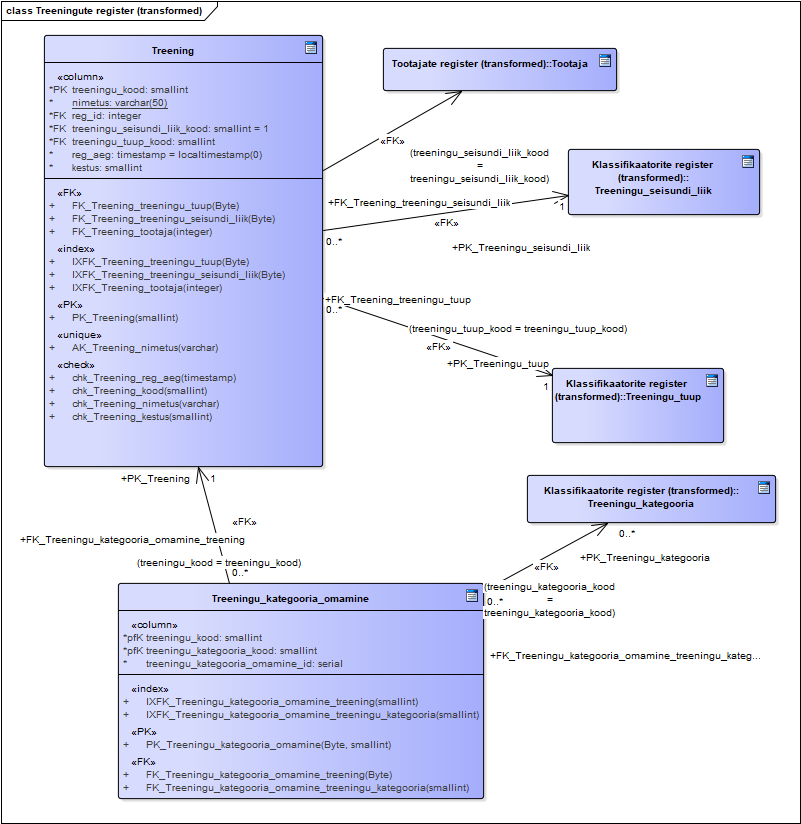
Joonis 11 Isikute registri füüsilise disaini andmebaasi diagramm



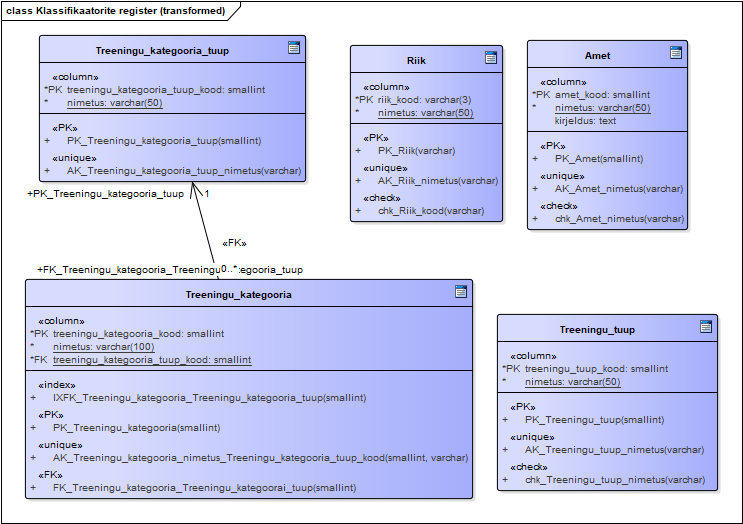
Joonis 12 Töötajate registri füüsilise disaini andmebaasi diagramm



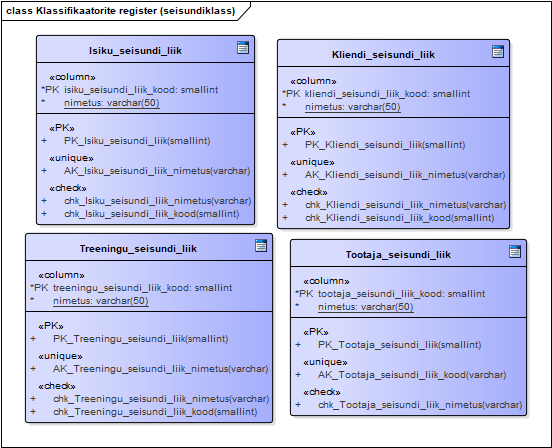
Joonis 13 Klientide registri füüsilise disaini andmebaasi diagramm



Joonis 14 Treeningute registri füüsilise disaini andmebaasi diagramm



Joonis 15 Klassifikaatorite registri füüsilise disaini andmebaasi diagramm



Joonis 16 Klassifikaatorite registri(seisundiklass) füüsilise disaini andmebaasi diagramm

# Realisatsioon PostgreSQLis

Selles peatükis esitatakse andmebaasi PostgreSQLis **(11.1)** realiseerimiseks mõeldud laused.

## Andmebaasi loomine

Serveris tuleb konsooli kirjutada psql , et jõuda PostgreSQL keskkonda. Selles tuleb kirjutada andmebaasi loomiseks:[3]

createdb -l et\_EE.utf8 -T template0 t164027

## Skeemid

Skeemid loodi andmebaasi loomisel automaatselt.

## Domeenid

CREATE DOMAIN **d\_reg\_aeg** TIMESTAMP without time zone

NOT NULL DEFAULT localtimestamp(0)

CONSTRAINT CHK\_reg\_aeg\_vahemikus CHECK

(VALUE BETWEEN '2010-01-01' AND '2100-12-31 23:59:59');

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

CONSTRAINT CHK\_nimetus\_pole\_tyhi CHECK (VALUE !~ '^[[:space:]]+$')

CONSTRAINT CHK\_nimetus\_pole\_tyhi\_string CHECK(VALUE!='');

## Tabelid ja arvujada generaatorid

Kuna kasutan tabelite loomisel SERIAL notatsiooni, siis süsteem loob arvujada generaatorid automaatselt ja seob need ka tabelitega.[3]

CREATE TABLE **Treeningu\_kategooria\_tuup**(

treeningu\_kategooria\_tuup\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Treeningu\_kategooria\_tuup PRIMARY KEY (treeningu\_kategooria\_tuup\_kood),

CONSTRAINT AK\_Treeningu\_kategooria\_tuup\_nimetus UNIQUE (nimetus)

);

CREATE TABLE **Treeningu\_kategooria**(

treeningu\_kategooria\_kood smallint NOT NULL,

nimetus d\_nimetus,

treeningu\_kategooria\_tuup\_kood smallint NOT NULL,

CONSTRAINT PK\_Treeningu\_kategooria PRIMARY KEY (treeningu\_kategooria\_kood),

CONSTRAINT AK\_Treeningu\_kategooria\_nimetus\_Treeningu\_kategooria\_tuup\_kood UNIQUE

(treeningu\_kategooria\_tuup\_kood,nimetus),

CONSTRAINT FK\_Treeningu\_kategooria\_Treeningu\_kategooria\_tuup FOREIGN KEY (treeningu\_kategooria\_tuup\_kood) REFERENCES Treeningu\_kategooria\_tuup

(treeningu\_kategooria\_tuup\_kood) ON DELETE No Action ON UPDATE Cascade

);

CREATE TABLE **Treeningu\_seisundi\_liik**(

treeningu\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Treeningu\_seisundi\_liik PRIMARY KEY (treeningu\_seisundi\_liik\_kood),

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

);

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

tootaja\_seisundi\_liik\_kood smallint NOT NULL, nimetus d\_nimetus,

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

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

);

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

kliendi\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

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

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

);

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

isiku\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

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

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

);

CREATE TABLE **Riik**(

riik\_kood varchar(3) NOT NULL, nimetus d\_nimetus,

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

CONSTRAINT AK\_Riik\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Riik\_riik\_kood\_koosneb\_kolmest\_suurest\_tahest

CHECK (riik\_kood~'^[A-Z]{3}$')

);

CREATE TABLE **Amet**(

amet\_kood smallint NOT NULL,

nimetus d\_nimetus,

kirjeldus text,

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

CONSTRAINT AK\_Amet\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Amet\_kirjeldus\_ei\_koosne\_tyhikutest\_pole\_tyhi

CHECK (kirjeldus !~ '^[[:space:]]\*$')

) WITH (FILLFACTOR = 90);

CREATE TABLE **Isik**(

isik\_id serial NOT NULL,

isikukood varchar(60) NOT NULL,

riik\_kood varchar(3) NOT NULL,

meil varchar(254) NOT NULL,

parool varchar(60) NOT NULL,

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

sunni\_kp date NOT NULL,

reg\_aeg d\_reg\_aeg,

eesnimi varchar(1000) NULL,

perenimi varchar(1000) NULL,

elukoht varchar(1000) NULL,

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

CONSTRAINT AK\_Isik\_meil UNIQUE (meil),

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

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

CONSTRAINT CHK\_Isik\_meil\_oige\_struktuur CHECK (meil::text ~ '^.\*@.\*$'::text AND meil::text !~ '^.\*@.\*@.\*$'::text),

CONSTRAINT CHK\_Isik\_parool\_ei\_koosne\_tyhikutest\_pole\_tyhi CHECK (parool!~'^[[:space:]]\*$'),

CONSTRAINT CHK\_Isik\_sunni\_kp\_vahemik CHECK (sunni\_kp >='1900-01-01' AND sunni\_kp <= '2100-12-31'),

CONSTRAINT CHK\_Isik\_sunni\_kp\_vahem\_kui\_praegu CHECK (sunni\_kp <= reg\_aeg),

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

CONSTRAINT CHK\_Isik\_isikukood\_lubatud\_symbolid CHECK (isikukood::text ~\* '^[a-z0-9 /-]+$'::text),

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

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

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

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

CONSTRAINT FK\_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\_Isik FOREIGN KEY (isik\_id) REFERENCES Isik (isik\_id)

ON DELETE Cascade ON UPDATE No Action,

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

) 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\_mentor CHECK (isik\_id <> mentor),

CONSTRAINT FK\_Tootaja\_Amet FOREIGN KEY (amet\_kood) REFERENCES Amet (amet\_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,

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

) WITH (FILLFACTOR = 90);

CREATE TABLE **Treeningu\_tuup**(

treeningu\_tuup\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Treeningu\_tuup PRIMARY KEY (treeningu\_tuup\_kood),

CONSTRAINT AK\_Treeningu\_tuup\_nimetus UNIQUE (nimetus)

);

CREATE TABLE **Treening**(

treeningu\_kood smallint NOT NULL,

nimetus d\_nimetus,

reg\_id integer NOT NULL,

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

treeningu\_tuup\_kood smallint NOT NULL,

reg\_aeg d\_reg\_aeg,

kestus smallint NOT NULL,

CONSTRAINT PK\_Treening PRIMARY KEY (treeningu\_kood),

CONSTRAINT AK\_Treening\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Treening\_kestus\_oige\_vahemik CHECK (kestus >=30 AND kestus <=90),

CONSTRAINT FK\_Treening\_Tootaja FOREIGN KEY (reg\_id) REFERENCES Tootaja (isik\_id)

ON DELETE No Action ON UPDATE No Action,

CONSTRAINT FK\_Treening\_Treeningu\_seisundi\_liik FOREIGN KEY (treeningu\_seisundi\_liik\_kood)

REFERENCES Treeningu\_seisundi\_liik (treeningu\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Treening\_Treeningu\_tuup FOREIGN KEY (treeningu\_tuup\_kood) REFERENCES Treeningu\_tuup

(treeningu\_tuup\_kood) ON DELETE No Action ON UPDATE Cascade

) WITH (FILLFACTOR = 90);

CREATE TABLE **Treeningu\_kategooria\_omamine**(

treeningu\_kood smallint NOT NULL,

treeningu\_kategooria\_kood smallint NOT NULL,

CONSTRAINT PK\_Treeningu\_kategooria\_omamine PRIMARY KEY

(treeningu\_kood, treeningu\_kategooria\_kood),

CONSTRAINT AK\_Treeningu\_kategooria\_omamine\_Treeningu\_katogooria\_Treening

UNIQUE (treeningu\_kood, treeningu\_kategooria\_kood),

CONSTRAINT FK\_Treeningu\_kategooria\_omamine\_Treening FOREIGN KEY (treeningu\_kood)

REFERENCES Treening (treeningu\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Treeningu\_kategooria\_omamine\_Treeningu\_kategooria FOREIGN KEY

(treeningu\_kategooria\_kood) REFERENCES Treeningu\_kategooria (treeningu\_kategooria\_kood)

ON DELETE No Action ON UPDATE Cascade

);

## Vaated

Kõik suuremad päringud tuleb realiseerida vaadetena.

CREATE VIEW **aktiivsed\_mitteaktiivsed\_treeningud** WITH (security\_barrier)

AS SELECT Treening.treeningu\_kood, Treening.nimetus AS treeningu\_nimetus, Treening.kestus, Treeningu\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Treeningu\_seisundi\_liik INNER JOIN Treening ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood

WHERE Treening.treeningu\_seisundi\_liik\_kood In (2,3);

COMMENT ON VIEW **aktiivsed\_mitteaktiivsed\_treeningud** IS 'Naitab aktiivsed ja mitteaktiivseid treeninguid (kood, nimetus, kestus ja seisundi liik)';

CREATE OR REPLACE VIEW **treeningute\_kategooria\_omamine** WITH (security\_barrier)

AS SELECT Treeningu\_kategooria\_omamine.treeningu\_kood, Treeningu\_kategooria.nimetus || ' (' || Treeningu\_kategooria\_tuup.nimetus || ')' AS kategooria

FROM Treeningu\_kategooria\_tuup INNER JOIN (Treeningu\_kategooria INNER JOIN Treeningu\_kategooria\_omamine ON Treeningu\_kategooria.treeningu\_kategooria\_kood = Treeningu\_kategooria\_omamine.treeningu\_kategooria\_kood) ON Treeningu\_kategooria\_tuup.treeningu\_kategooria\_tuup\_kood = Treeningu\_kategooria.treeningu\_kategooria\_tuup\_kood;

COMMENT ON VIEW **treeningute\_kategooria\_omamine** IS 'Naitab treeningude kategooriad (kood ja treeningu kategooria)';

CREATE OR REPLACE VIEW **koik\_treeningud** WITH (security\_barrier)

AS SELECT Treening.treeningu\_kood, Treening.nimetus AS treeningu\_nimetus, Treening.kestus, Treeningu\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Treeningu\_seisundi\_liik INNER JOIN Treening ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood;

COMMENT ON VIEW **koik\_treeningud** IS 'Naitab koike treeninguid (kood, nimetus, kestus ja treeningu hetke seisundi';

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

AS SELECT Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood, upper(Treeningu\_seisundi\_liik.nimetus) AS seisundi\_nimetus, Count(Treening.treeningu\_kood) AS arv

FROM Treeningu\_seisundi\_liik LEFT JOIN Treening ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood

GROUP BY Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood, upper(Treeningu\_seisundi\_liik.nimetus)

ORDER BY Count(Treening.treeningu\_kood) DESC, upper(Treeningu\_seisundi\_liik.nimetus);

COMMENT ON VIEW **treeningute\_koondaruanne** IS 'Naitab treeningute koondaruanne nende seisundi liigi pohjal (seisundi kood, seisundi nimetus ja arv, mis naitab, kui palju on niisugusi treeningusi';

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

AS SELECT Treening.treeningu\_kood, Treening.nimetus AS treeningu\_nimetus, Treening.kestus, Treeningu\_seisundi\_liik.nimetus AS hetke\_seisund, Isik.meil AS registreerija\_email, (COALESCE(eesnimi, perenimi) AS registreerija\_nimi, Treening.reg\_aeg

FROM Treeningu\_seisundi\_liik INNER JOIN (Isik INNER JOIN Treening ON Isik.isik\_id=Treening.reg\_id) ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood;

COMMENT ON VIEW **treeningute\_detailid** IS 'Naitab koike treeningusi detailid (kood, nimetus, kestus, hetke seisund, registreerija email, registreerija tais nimi, registratsiooni aeg)';

CREATE OR REPLACE VIEW **treeningute\_kategooriad** WITH (security\_barrier)

AS SELECT Treening.treeningu\_kood, string\_agg((((Treeningu\_kategooria\_tuup.nimetus || ': ') || LOWER(Treeningu\_kategooria.nimetus))), '; ') AS Treeningu\_kategooriad

FROM (((Treening LEFT JOIN Treeningu\_kategooria\_omamine ON ((Treening.treeningu\_kood = Treeningu\_kategooria\_omamine.treeningu\_kood)))

LEFT JOIN Treeningu\_kategooria ON ((Treeningu\_kategooria\_omamine.Treeningu\_kategooria\_kood = Treeningu\_kategooria.Treeningu\_kategooria\_kood)))

LEFT JOIN Treeningu\_kategooria\_tuup ON ((Treeningu\_kategooria\_tuup.Treeningu\_kategooria\_tuup\_kood = Treeningu\_kategooria.Treeningu\_kategooria\_tuup\_kood)))

GROUP BY Treening.treeningu\_kood

ORDER BY Treening.treeningu\_kood ASC;

COMMENT ON VIEW **treeningute\_kategooriad** IS 'See vaade näitab, millistesse kategooriatesse mingi treening kuulub.';

## Protseduursed keeled

Funktsioonides kasutasin PL/pgSQL keelt. See tuleb plpgsql laienduse installeerimisel. plpgsql laiendus on automaatselt juba installeeritud.

## Trigeri funktsioonid ja trigerid

Peab olema kasutatud vähemalt kahte trigerit või reeglit, mis teevad midagi muud kui tabeli ja arvujada generaatori sidumine. Samuti ei lähe arvesse trigerid/reeglid, mis lahendavad ülesandeid, mida saaks lahendada deklaratiivsel viisil.

CREATE OR REPLACE FUNCTION **f\_kustuta\_treening()** RETURNS trigger

LANGUAGE 'plpgsql'

COST 100

VOLATILE NOT LEAKPROOF

AS $BODY$

BEGIN

RAISE EXCEPTION 'Ainult ootel olevaid treeningu on võimalik kustutada!';

RETURN OLD;

END;

$BODY$;

COMMENT ON FUNCTION **f\_kustuta\_treening()** IS 'See trigger määrab ära selle, et kustutada saab ainult ootel olevaid treeningu.';

CREATE OR REPLACE FUNCTION **f\_muuda\_treeningu\_seisundi\_liik()**

RETURNS trigger

LANGUAGE 'plpgsql'

COST 100

VOLATILE NOT LEAKPROOF

AS $BODY$

BEGIN

RAISE EXCEPTION 'Nii pole võimalik muuta treeningu seisundit!';

RETURN NULL;

END;

$BODY$;

COMMENT ON FUNCTION **f\_muuda\_treeningu\_seisundi\_liik()** IS 'See trigger kontrollib, et toimub korrektne treeningu seisundi liigi muudatus. Kui old.treeningu\_seisundi\_liik\_kood=new.treeningu\_seisundi\_liik\_kood, siis treeningu seisundi kood jääb samaks.';

CREATE TRIGGER **trig\_treening\_kustuta\_treening** BEFORE DELETE ON Treening

FOR EACH ROW WHEN ((old.treeningu\_seisundi\_liik\_kood IS DISTINCT FROM 1)) EXECUTE PROCEDURE f\_kustuta\_treening();

CREATE TRIGGER **trig\_treening\_muuda\_treeningu\_seisundi\_liik** BEFORE UPDATE OF treeningu\_seisundi\_liik\_kood ON Treening

FOR EACH ROW WHEN ((NOT ((old.treeningu\_seisundi\_liik\_kood = new.treeningu\_seisundi\_liik\_kood) OR ((old.treeningu\_seisundi\_liik\_kood = 1) AND (new.treeningu\_seisundi\_liik\_kood = 3)) OR ((old.treeningu\_seisundi\_liik\_kood = 3) AND (new.treeningu\_seisundi\_liik\_kood = ANY (ARRAY[2, 4]))) OR ((old.treeningu\_seisundi\_liik\_kood = 2) AND (new.treeningu\_seisundi\_liik\_kood = ANY (ARRAY[3, 4])))))) EXECUTE PROCEDURE f\_muuda\_treeningu\_seisundi\_liik();

## Reeglid

Kasutasime trigereid mitte reegleid.

## Funktsioonid

Tuleb luua vähemalt kolm funktsiooni, mis pole seotud trigeritega ja

mille poole rakendusest pöördutakse.

CREATE OR REPLACE FUNCTION **f\_lisa\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE, p\_tuup\_kood Treening.treeningu\_tuup\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_registreerija\_id Treening.reg\_id%TYPE, p\_kestus Treening.kestus%TYPE)**

RETURNS Treening.treeningu\_kood%TYPE AS $$

INSERT INTO Treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, kestus)

VALUES (p\_treeningu\_kood, p\_tuup\_kood, p\_nimetus, p\_registreerija\_id, p\_kestus) ON CONFLICT DO NOTHING RETURNING treeningu\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION **f\_lisa\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE, p\_tuup\_kood Treening.treeningu\_tuup\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_registreerija\_id Treening.reg\_id%TYPE, p\_kestus Treening.kestus%TYPE)**

IS 'Selle funktsiooni abil registreeritakse uus treening. See funktsioon realiseerib andmebaasioperatsiooni OP1. Parameetri p\_treeningu\_kood oodatav vaartus on treeningu identifikaator, p\_tuup\_kood oodatav vaartus on treeningu tuubi identifikaator, p\_nimetus oodatav vaartus on treeningu nimetus, p\_registreerija\_id oodatav vaartus on treeningu registreerinud tootaja identifikaator, p\_treeningu\_seisundi\_liik oodatav vaartus on treeningu liigi identifikaator, p\_kestus oodatav vaartus on treeningu kestus minutites';

CREATE OR REPLACE FUNCTION **f\_unusta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE)**

RETURNS VOID AS $$

DELETE FROM Treening WHERE treeningu\_kood = p\_treeningu\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION **f\_unusta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE)** IS 'Selle funktsiooni abil unustatake (kustutakse) treening. See funktsioon realiseerib andmebaasioperatsiooni OP2. Parameetri p\_treeningu\_kood oodatav vaartus on treeningu identifikaator.';

CREATE OR REPLACE FUNCTION **f\_aktiveeri\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE)**

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_seisundi\_liik\_kood = 3 WHERE treeningu\_kood = p\_treeningu\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION **f\_aktiveeri\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE)** IS 'Selle funktsiooni abil aktiveeritakse treening. See funktsioon realiseerib andmebaasioperatsiooni OP3. Parameetri p\_treeningu\_kood oodatav vaartus on treeningu identifikaator.';

CREATE OR REPLACE FUNCTION **f\_lopeta\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE)**

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_seisundi\_liik\_kood = 4 WHERE treeningu\_kood = p\_treeningu\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION **f\_lopeta\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE)** IS 'Selle funktsiooni abil lopetatakse treening. See funktsioon realiseerib andmebaasioperatsiooni OP5. Parameetri p\_treeningu\_kood oodatav vaartus on treeningu identifikaator.';

CREATE OR REPLACE FUNCTION **f\_muuda\_treening(**

**p\_treeningu\_kood\_vana Treening.treeningu\_kood%TYPE,**

**p\_treeningu\_kood\_uus Treening.treeningu\_kood%TYPE,**

**p\_nimetus Treening.nimetus%TYPE,**

**p\_kestus Treening.kestus%TYPE)**

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_kood = p\_treeningu\_kood\_uus, nimetus = p\_nimetus, kestus = p\_kestus WHERE treeningu\_kood = p\_treeningu\_kood\_vana;

$$ LANGUAGE SQL SECURITY DEFINER

SET search\_path=public, pg\_temp;

COMMENT ON FUNCTION **f\_muuda\_treening (p\_treeningu\_kood\_vana**

**Treening.treeningu\_kood%TYPE, p\_treeningu\_kood\_uus Treening.treeningu\_kood%TYPE, p\_nimetus Treening.nimetus%type, p\_kestus Treening.kestus%TYPE)** IS'OP6 Protsess, mis muudab treeningu andmeid. Protseduuri oodatavadsisendid on treeningu vana identifikaator (parameeter p\_treeningu\_kood\_vana),treeningu uus identifikaator (parameeter p\_treeningu\_kood\_uus),nimetus (parameeter p\_nimetus) jakestus (parameeter p\_kestus). Protseduur muudab etteantud treeningu.';

CREATE OR REPLACE FUNCTION **f\_on\_juhataja(p\_kasutajanimi text, p\_parool text)**

RETURNS boolean AS $$ DECLARE rslt boolean;

BEGIN

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

from Isik

INNER JOIN Tootaja ON Isik.isik\_id = Tootaja.isik\_id

WHERE Upper(meil) = Upper(p\_kasutajanimi)

AND Tootaja.amet\_kood = 1

AND isiku\_seisundi\_liik\_kood = 1

AND Tootaja.tootaja\_seisundi\_liik\_kood IN (1, 2, 3, 4);

RETURN coalesce(rslt, FALSE);

END;

$$

LANGUAGE plpgsql

SECURITY DEFINER

STABLE

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION **f\_on\_juhataja(p\_kasutajanimi text, p\_parool text)**

IS 'Selle funktsiooni abil autenditakse juhatajat. Parameetri p\_kasutajanimi oodatav väärtus on tõstutundetu kasutajanimi ehk meil ja parool oodatav väärtus on tõstutundlik avatekstiline parool. Juhatahal on õigus süsteemi siseneda, vaid siis kui tema seisundiks on tööl, haiguslehel, puhkusel või katseajal.';

## Indeksid

Järgnevalt esitatakse indeksite loomise laused.

### Välisvõtmete veergudele lisatavad indeksid

Kõik välisvõtmete veerud tuleb indekseerida. Seda tuleb teha viisil, mis ei põhjusta üksteist dubleerivate indeksite loomist põhjusel, et andmebaasisüsteem loon mõningad indeksid automaatselt.

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\_seisundi\_liik ON Tootaja (tootaja\_seisundi\_liik\_kood ASC);

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

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

CREATE INDEX IXFK\_Treening\_Tootaja ON Treening (reg\_id ASC);

CREATE INDEX IXFK\_Treening\_tuup ON Treening (treeningu\_tuup\_kood ASC);

CREATE INDEX IXFK\_Treening\_seisundi\_liik ON Treening (treeningu\_seisundi\_liik\_kood ASC);

CREATE INDEX IXFK\_Treeningu\_kategooria\_omamine\_Treeningu\_kategooria\_kood ON Treeningu\_kategooria\_omamine (treeningu\_kategooria\_kood ASC);

CREATE INDEX IXFK\_Isik\_perenimi ON Isik (perenimi);

CREATE INDEX IXFK\_Isik\_eesnimi ON Isik (eesnimi);

### Täiendavad sekundaarsed indeksid

Puuduvad.

### Funktsioonil põhinevad indeksid

Tagamaks kasutajanimede unikaalsuse sõltumata sellest, kas neis kasutatakse suur- või väiketähti ning kiirendamaks kasutajanimede tõstutundetut otsingut, asendan tabelites Isik, Tootaja ja Klient veerule meil loodud unikaalsuse kitsenduse unikaalse funktsioonil Upper põhineva indeksiga.

ALTER TABLE Isik DROP CONSTRAINT AK\_Isik\_meil;

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

## 

## Klassifikaatorite väärtustamise SQL laused

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES (1, 'Juhataja', 'Vastutav isik töökoha eest');

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES (2, 'Treeningute haldur', 'Vastutav isik treeningute eest');

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES (3, 'Treener', 'Treener on sportlasi kutseliselt ettevalmistav ja õpetav inimene');

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, 'Katseajal');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (2, 'Tool');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (3, 'Puhkusel');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (4, 'Haiguslehel');

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

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus) VALUES (6, 'Töösuhe lõpetatud omal soovil');

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

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

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus) VALUES (2, 'Mitteaktiivne');

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus) VALUES (3, 'Aktiivne');

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus) VALUES (4, 'Lopetatud');

## JSON formaadis lähteandmete laadimine

Välistabelite kasutamiseks (et sealt JSON formaadis lähteandmeid kätte saada) läks vaja postgres\_fdw laiendust. Seega selle ka installeerisin.

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');

CREATE USER MAPPING FOR t164027 SERVER minu\_testandmete\_server\_apex OPTIONS (user 't164027', password '');

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

INSERT INTO Riik (riik\_kood, nimetus)

SELECT riik->>'Alpha-3 code' AS riik\_kood, riik->>'English short name lower case' AS nimetus

FROM Riik\_jsonb;

SELECT \* FROM Riik;

CREATE FOREIGN TABLE Isik\_jsonb (isik JSONB )SERVER minu\_testandmete\_server\_apex;

CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;

INSERT INTO Isik(riik\_kood, isikukood, eesnimi, perenimi, meil, sunni\_kp, isiku\_seisundi\_liik\_kood, parool, elukoht)

SELECT riik\_kood, isikukood, eesnimi, perenimi, meil, sunni\_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 meil,

jsonb\_array\_elements(isik->'isikud')->>'synni\_aeg' AS sunni\_kp,

jsonb\_array\_elements(isik->'isikud')->>'seisund' AS isiku\_seisundi\_liik\_kood,

jsonb\_array\_elements(isik->'isikud')->>'parool' AS parool,

jsonb\_array\_elements(isik->'isikud')->>'aadress' AS elukoht

FROM isik\_jsonb) AS lahteandmed

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

SELECT \* FROM Isik;

## Täiendavate testandmete lisamine

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

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

INSERT INTO klient(isik\_id, kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (7, 1, false);

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

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

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (1, 1, 2, NULL);

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

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

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (4, 3, 1, 2);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (5, 3, 2, NULL);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (6, 3, 1, 4);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (7, 3, 1, 5);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (8, 1, 2, NULL);

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (1, 'Jooksmine');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (2, 'Joutreening');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (3, 'Kaughüpe');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (4, 'Kettaheide');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (5, 'Kõrgushüpe');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (6, 'Maraton');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (7, 'Odavise');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (8, 'Sprint');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (9, 'Tõkkejooks');

INSERT INTO treeningu\_tuup(treeningu\_tuup\_kood, nimetus) VALUES (10, 'Vastupidavus');

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (1, 1, 'Jooksmine', 1, 2, '2018-04-10', 90);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (2, 2, 'Vastupidavus', 1, 2, '2018-04-10', 60);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (3, 2, 'Joutreening', 1, 1, '2018-04-10', 60);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (4, 2, 'Kaughüpe', 1, 2, '2018-04-10', 60);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (5, 2, 'Maraton', 1, 1, '2018-04-10', 60);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (6, 2, 'Sprint', 1, 1, '2017-03-13', 60);

INSERT INTO treening(treeningu\_kood, treeningu\_tuup\_kood, nimetus, reg\_id, treeningu\_seisundi\_liik\_kood, reg\_aeg, kestus) VALUES (7, 2, 'Tõkkejooks', 8, 2, '2019-01-09', 90);

INSERT INTO treeningu\_kategooria\_tuup(treeningu\_kategooria\_tuup\_kood, nimetus) VALUES (1, 'Raskusaste');

INSERT INTO treeningu\_kategooria\_tuup(treeningu\_kategooria\_tuup\_kood, nimetus) VALUES (2, 'Hooaeg');

INSERT INTO treeningu\_kategooria\_tuup(treeningu\_kategooria\_tuup\_kood, nimetus) VALUES (3, 'Treeningvahendid');

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (1, 'Algajatele', 1);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (2, 'Algajatele ja edasijõudnutele', 1);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (3, 'Edasijoudnutele', 1);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (4, 'Suvine spordiala', 2);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (5, 'Talvine spordiala', 2);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (6, 'Iga hooajane spordiala', 2);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (7, 'Tõkked', 3);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (8, 'Kettad', 3);

INSERT INTO treeningu\_kategooria(treeningu\_kategooria\_kood, nimetus, treeningu\_kategooria\_tuup\_kood) VALUES (9, 'Odad', 3);

INSERT INTO treeningu\_kategooria\_omamine(treeningu\_kood, treeningu\_kategooria\_kood) VALUES (1, 1);

INSERT INTO treeningu\_kategooria\_omamine(treeningu\_kood, treeningu\_kategooria\_kood) VALUES (1, 2);

INSERT INTO treeningu\_kategooria\_omamine(treeningu\_kood, treeningu\_kategooria\_kood) VALUES (2, 1);

INSERT INTO treeningu\_kategooria\_omamine(treeningu\_kood, treeningu\_kategooria\_kood) VALUES (2, 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.[3]

## Päringu täitmisplaani näide

Analüüsin päringut, mis on tehtud vaate *koik\_treeningud* põhjal.

CREATE OR REPLACE VIEW **koik\_treeningud** WITH (security\_barrier)

AS SELECT Treening.treeningu\_kood, Treening.nimetus AS treeningu\_nimetus, Treening.kestus, Treeningu\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Treeningu\_seisundi\_liik INNER JOIN Treening ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood;

t164027=# ANALYZE;

ANALYZE

t164027=# EXPLAIN SELECT \* FROM aktiivsed\_mitteaktiivsed\_treeningud;

Nested Loop (cost=0.00..17.30 rows=1 width=129)

Join Filter: (treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = treening.treeningu\_seisundi\_liik\_kood)

-> Seq Scan on treening (cost=0.00..16.12 rows=5 width=124)

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

-> Materialize (cost=0.00..1.03 rows=2 width=9)

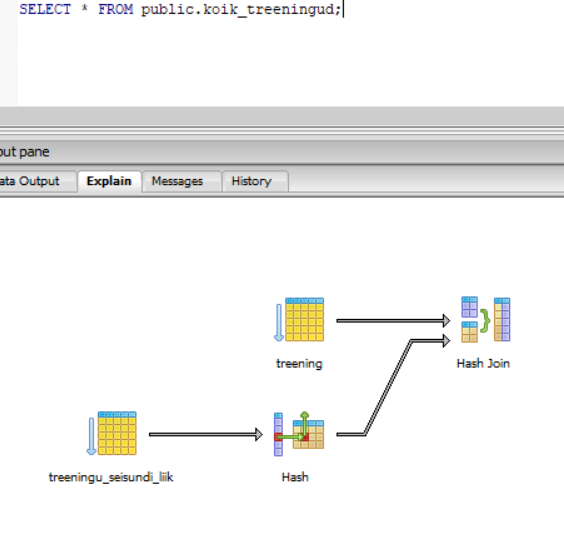
-> Seq Scan on treeningu\_seisundi\_liik (cost=0.00..1.02 rows=2 width=9)

(6 rows)

Enne täitmisplaani koostamist värskendasin ANALYZE lause abil andmebaasi statistikat, et andmebaasisüsteemil oleks täitmisplaani koostamiseks võimalikult täpne sisendinformatsioon. Täitmisplaani tuleb lugeda alt üles ja seest välja (paremalt vasakule). Allpool/seespool esitatud operatsiooni tulemus on sisendiks ülalpool/väljaspool esitatud operatsioonile. Täitmisplaani koostamise ajal olid tabelid peaaegu tühjad ja see võib mõjutada plaani valikut (näiteks antud juhul eelistatakse tabeli täielikku läbiskaneerimist indeksi kasutamisele). 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.[3]

1. – Tabeli Treening läbiskaneerimisel leitakse vaid read, mis rahuldavad tingimust treeningu\_seisundi\_liik\_kood = ANY ('{2,3}'::integer[]). Tabelite Treening ja Treeningu\_seisundi\_liik ühendamise operatsiooni sisendiks lähevad vaid leitud read tabelist Treening.

1-4 – Tabelid Treening ja Treeningu\_seisundi\_liik ühendatakse kasutades Nested Loop algoritmi. Tabelist Treening loetud Treeningu\_seisundi\_liik väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema tabelist Treeningu\_seisundi\_liik loetud Treeningu\_seisundi\_liik väärtuste põhjal leitud räsiväärtustega. Selle ühendamise kontekstis on Treeningu\_seisundi\_liik väline tabel.



Joonis 17 Täitmisplaan vaadatuna visuaalselt PgAdmin’is

## Rollid ja kasutajad

Käesolevas töös rolle ei looda. Luuakse rakendusele vastav kasutaja.

CREATE USER **t164027\_juhataja** WITH PASSWORD 'parool';

## Üleliigsete õiguste äravõtmine

REVOKE CONNECT, TEMP ON DATABASE **t164027** FROM PUBLIC;

REVOKE CREATE, USAGE ON SCHEMA **public** FROM PUBLIC;

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

REVOKE EXECUTE ON FUNCTION

**f\_lisa\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE, p\_tuup\_kood Treening.treeningu\_tuup\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_registreerija\_id Treening.reg\_id%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_unusta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_aktiveeri\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_lopeta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_muuda\_treening(p\_treeningu\_kood\_vana Treening.treeningu\_kood%TYPE, p\_treeningu\_kood\_uus Treening.treeningu\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_on\_juhataja(p\_kasutajanimi text, p\_parool text),**

**f\_kustuta\_treening(),**

**f\_muuda\_treeningu\_seisundi\_liik()**

FROM PUBLIC;

REVOKE EXECUTE ON ALL FUNCTIONS IN SCHEMA public FROM PUBLIC;

## Õiguste jagamine

GRANT CONNECT ON DATABASE **t164027** TO t164027\_juhataja;

GRANT USAGE ON SCHEMA **public** TO t164027\_juhataja;

GRANT SELECT, INSERT, UPDATE ON TABLE **Treening** TO t164027\_juhataja;

GRANT SELECT ON

**aktiivsed\_mitteaktiivsed\_treeningud,**

**treeningute\_kategooria\_omamine,**

**koik\_treeningud,**

**treeningute\_koondaruanne,**

**treeningute\_detailid,**

**treeningute\_kategooriad**

TO t164027\_juhataja;

GRANT EXECUTE ON FUNCTION

**f\_lisa\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE, p\_tuup\_kood Treening.treeningu\_tuup\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_registreerija\_id Treening.reg\_id%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_unusta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_aktiveeri\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_lopeta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_muuda\_treening(p\_treeningu\_kood\_vana Treening.treeningu\_kood%TYPE, p\_treeningu\_kood\_uus Treening.treeningu\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_on\_juhataja(p\_kasutajanimi text, p\_parool text)**

TO t164027\_juhataja;

ALTER DEFAULT PRIVILEGES REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC;

## Andmebaasiobjektide kustutamine

Järgnevalt esitatakse andmebaasiobjektide kustutamise laused.[3]

### Õiguste äravõtmine

REVOKE CONNECT ON DATABASE **t164027** FROM t164027\_juhataja;

REVOKE USAGE ON SCHEMA **public** FROM t164027\_juhataja;

REVOKE SELECT, INSERT, UPDATE ON TABLE **Treening** FROM t164027\_juhataja;

REVOKE SELECT ON

**aktiivsed\_mitteaktiivsed\_treeningud,**

**treeningute\_kategooria\_omamine,**

**koik\_treeningud,**

**treeningute\_koondaruanne,**

**treeningute\_detailid,**

**treeningute\_kategooriad**

FROM t164027\_juhataja;

REVOKE EXECUTE ON FUNCTION

**f\_lisa\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE, p\_tuup\_kood Treening.treeningu\_tuup\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_registreerija\_id Treening.reg\_id%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_unusta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_aktiveeri\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_lopeta\_treening(p\_treeningu\_kood Treening.treeningu\_kood%TYPE),**

**f\_muuda\_treening(p\_treeningu\_kood\_vana Treening.treeningu\_kood%TYPE, p\_treeningu\_kood\_uus Treening.treeningu\_kood%TYPE, p\_nimetus Treening.nimetus%TYPE, p\_kestus Treening.kestus%TYPE),**

**f\_on\_juhataja(p\_kasutajanimi text, p\_parool text)**

FROM t164027\_juhataja;

### Domeenide kustutamine

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

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

### Tabelite ja arvujada generaatorite kustutamine

DROP TABLE IF EXISTS **Treeningu\_kategooria\_tuup** CASCADE;

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

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

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

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

DROP TABLE IF EXISTS **Isiku\_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 **Treening** CASCADE;

DROP TABLE IF EXISTS **Treeningu\_tuup** CASCADE;

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

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

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

DROP USER MAPPING FOR t164027 SERVER minu\_testandmete\_server\_apex;

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

### Vaadete kustutamine

DROP VIEW IF EXISTS **aktiivsed\_mitteaktiivsed\_treeningud** CASCADE;

DROP VIEW IF EXISTS **treeningute\_kategooria\_omamine** CASCADE;

DROP VIEW IF EXISTS **koik\_treeningud** CASCADE;

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

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

### 

### DROP VIEW IF EXISTS treeningute\_kategooriad 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\_seisundi\_liik**;

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

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

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

DROP INDEX IF EXISTS **IXFK\_Treening\_tuup**;

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

DROP INDEX IF EXISTS **IXFK\_Treeningu\_kategooria\_omamine\_treening**;

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

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

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

### Funktsioonide ja trigerite kustutamine

DROP FUNCTION IF EXISTS **f\_lisa\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE, p\_tuup\_kood treening.treeningu\_tuup\_kood%TYPE, p\_nimetus treening.nimetus%TYPE, p\_registreerija\_id treening.reg\_id%TYPE, p\_kestus treening.kestus%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_aktiveeri\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_unusta\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_lopeta\_treening(p\_treeningu\_kood treening.treeningu\_kood%TYPE)** CASCADE;

DROP FUNCTION IF EXISTS **f\_on\_juhataja(p\_kasutajanimi text, p\_parool text)**;

DROP FUNCTION IF EXISTS **f\_kustuta\_treening()** CASCADE;

DROP FUNCTION IF EXISTS **f\_muuda\_treeningu\_seisundi\_liik()** CASCADE;

DROP TRIGGER IF EXISTS **trig\_treening\_kustuta\_treening** ON Treening CASCADE;

DROP TRIGGER IF EXISTS **trig\_treening\_muuda\_treeningu\_seisundi\_**liik ON Treening CASCADE;

### Reeglite kustutamine

Kuna projektis reegleid ei kasutata, siis ei esitata ka reeglite kustutamise lauseid.[3]

### Kasutajate ja rollide kustutamine

Kuna projektis rolle ei kasutata, siis ei esitata ka rollide kustutamise lauseid. [3]

DROP USER IF EXISTS **t164027\_juhataja**;

### Laienduste kustutamine

DROP EXTENSION IF EXISTS **pgcrypto** CASCADE;

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

# Kasutatud materjalid

1. AKIT. Andmekaitse ja infoturbe seletussõnastik. [WWW] <http://akit.cyber.ee/> (10.02.2018)
2. Andmebaasid I õppematerjalid. [WWW] <http://maurus.ttu.ee> (10.02.2018)
3. Andmebaasid II õppematerjalid. [WWW] <http://maurus.ttu.ee> (31.08.2018)
4. Andmebaaside projekti tegemise mall. [WWW] <http://maurus.ttu.ee> (10.02.2018)
5. Country Codes - ISO 3166 [WWW]   
   <https://www.iso.org/iso-3166-country-codes.html> (10.02.2018)
6. Chisholm, M. (2000). *Managing Reference Data in Enterprise Databases:* Binding *Corporate Data to the Wider World.* Morgan Kaufmann.
7. Eesti Statistika. Riikide ja territooriumide klassifikaator 2013v1. [WWW] <http://metaweb.stat.ee/view_xml_multi_code.htm?id=3477719&siteLanguage=ee> (10.02.2018)
8. ESTERM [WWW] <http://termin.eki.ee/esterm/> (10.02.2018)
9. Isikuandmete kaitse seadus. [WWW] <https://www.riigiteataja.ee/akt/130122010011?leiaKehtiv> (10.02.2018)
10. Infosüsteemide turvameetmete süsteem. Vabariigi Valitsuse 20.12 2007. a määrus nr 252. Elektrooniline Riigi Teataja.  
    [WWW] <https://www.riigiteataja.ee/akt/13125331?leiaKehtiv> (10.02.2018)
11. Vikipeedia. Riik. [WWW] <https://et.wikipedia.org/wiki/Riik> (10.02.2018)