TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatikainstituut

Infosüsteemide õppetool

Spordiklubi infosüsteem - treeningute allsüsteem

Andmebaasid I – IDU0230

|  |  |
| --- | --- |
| Üliõpilane:   1. Sander Saarsen 2. Kaspar Metsa 3. Liidia Laada |  |
| Õpperühm:   1. IAPB53 2. IAPB13 3. IAPB33 |  |
| Matrikli nr:   1. 134683IAPB 2. 154768IAPB 3. 142463IAPB |  |
| e-mail:   1. Sander.saarsen@gmail.com 2. kasparme@gmail.com 3. liidia.laada@gmail.com |  |

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

Tallinn  
2015

**Sisukord**

[1 Strateegia etapp 8](#_Toc423474676)

[1.1 Terviksüsteemi üldvaade 8](#_Toc423474677)

[1.1.1 Organisatsiooni eesmärgid 8](#_Toc423474678)

[1.1.2 Infosüsteemi eesmärgid 8](#_Toc423474679)

[1.1.3 Lausendid 8](#_Toc423474680)

[1.1.4 Põhiobjektid 9](#_Toc423474681)

[1.1.5 Põhiprotsessid 9](#_Toc423474682)

[1.1.6 Põhilised sündmused 9](#_Toc423474683)

[1.1.7 Tegutsejad 10](#_Toc423474684)

[1.1.8 Asukohad 10](#_Toc423474685)

[1.1.9 Terviksüsteemi tükeldus allsüsteemideks 10](#_Toc423474686)

[1.1.9.1 Pädevusalad 10](#_Toc423474687)

[1.1.9.2 Funktsionaalsed allsüsteemid 10](#_Toc423474688)

[1.1.9.3 Registrid 11](#_Toc423474689)

[1.2 Treeningute funktsionaalse allsüsteemi eskiismudelid 11](#_Toc423474690)

[1.2.1 Eesmärgid 11](#_Toc423474691)

[1.2.2 Allsüsteemi kasutavad pädevusalad 11](#_Toc423474692)

[1.2.3 Allsüsteemi poolt kasutatavad registrid 11](#_Toc423474693)

[1.2.4 Allsüsteemi ühe põhiprotsessi tegevusdiagramm 12](#_Toc423474694)

[1.2.5 Allsüsteemi kasutusjuhtude eskiismudel 12](#_Toc423474695)

[1.2.6 Mittefunktsionaalsed nõuded 15](#_Toc423474696)

[1.3 Treeningute funktsionaalse allsüsteemi teenindatava treeningute registri eskiismudelid 17](#_Toc423474697)

[1.3.1 Eesmärgid 17](#_Toc423474698)

[1.3.2 Registrit kasutavad pädevusalad 17](#_Toc423474699)

[1.3.3 Registrit teenindavad funktsionaalsed allsüsteemid 17](#_Toc423474700)

[1.3.4 Infovajadused, mida register aitab rahuldada 17](#_Toc423474701)

[1.3.5 Seosed teiste registritega 18](#_Toc423474702)

[1.3.6 Ärireeglid 18](#_Toc423474703)

[1.3.7 Registri kontseptuaalne eskiismudel 18](#_Toc423474704)

[2 Detailanalüüsi etapp 19](#_Toc423474705)

[2.1 Treeningute funktsionaalse allsüsteemi detailanalüüs 19](#_Toc423474706)

[2.1.1 Kasutusjuhtude mudel 19](#_Toc423474707)

[2.2 Treeningute funktsionaalse allsüsteemi kasutatavate registrite detailanalüüs 24](#_Toc423474708)

[2.2.1 Kontseptuaalne andmemudel 24](#_Toc423474709)

[2.2.1.1 Olemi-suhte diagramm 24](#_Toc423474710)

[2.2.1.2 Olemitüüpide definitsioonid 25](#_Toc423474711)

[2.2.1.3 Atribuutide definitsioonid 26](#_Toc423474712)

[2.2.2 Andmebaasioperatsioonide lepingud 29](#_Toc423474713)

[2.2.3 Registri põhiobjekti seisundidiagramm 31](#_Toc423474714)

[2.3 CRUD maatriks 31](#_Toc423474715)

[2.4 Infosüsteemi rollide kirjeldused 32](#_Toc423474716)

[3 Füüsiline disain 33](#_Toc423474717)

[3.1 Treeningute funktsionaalse allsüsteemi kasutatavate registrite füüsiline disain 33](#_Toc423474718)

[3.1.1 Füüsiline andmebaasiskeem 33](#_Toc423474719)

[4 Realisatsioon PostgreSQLis 34](#_Toc423474720)

[4.1 Andmebaasi loomine 34](#_Toc423474721)

[4.2 Skeemid 34](#_Toc423474722)

[4.3 Domeenid 34](#_Toc423474723)

[4.4 Tabelid ja arvujada generaatorid 34](#_Toc423474724)

[4.5 Vaated 34](#_Toc423474725)

[4.6 Protseduursed keeled 34](#_Toc423474726)

[4.7 Trigeri funktsioonid ja trigerid 34](#_Toc423474727)

[4.8 Reeglid 34](#_Toc423474728)

[4.9 Funktsioonid 34](#_Toc423474729)

[4.10 Indeksid 34](#_Toc423474730)

[4.10.1 Välisvõtmetele lisatavad indeksid 35](#_Toc423474731)

[4.10.2 Täiendavad sekundaarsed indeksid 35](#_Toc423474732)

[4.10.3 Funktsioonil põhinevad indeksid 35](#_Toc423474733)

[4.11 Klassifikaatorite väärtustamise SQL laused 35](#_Toc423474734)

[4.12 Täiendavate testandmete lisamine 35](#_Toc423474735)

[4.13 Andmebaasi statistika kogumine 35](#_Toc423474736)

[4.14 Päringu täitmisplaani näide 35](#_Toc423474737)

[4.15 Rollid ja kasutajad 35](#_Toc423474738)

[4.16 Õiguste jagamine 35](#_Toc423474739)

[4.17 Andmebaasiobjektide kustutamine 35](#_Toc423474740)

[4.17.1 Domeenide kustutamine 35](#_Toc423474741)

[4.17.2 Tabelite kustutamine 35](#_Toc423474742)

[4.17.3 Arvujada generaatorite kustutamine 35](#_Toc423474743)

[4.17.4 Vaadete kustutamine 35](#_Toc423474744)

[4.17.5 Indeksite kustutamine 36](#_Toc423474745)

[4.17.6 Funktsioonide ja trigerite kustutamine 36](#_Toc423474746)

[4.17.7 Reeglite kustutamine 36](#_Toc423474747)

[4.17.8 Kasutajate ja rollide kustutamine 36](#_Toc423474748)

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.

1. Sander Saarsen

[Allkiri]

1. Kaspar Metsa

[Allkiri]

1. Liidia Laada

[Allkiri]

**Sissejuhatus (Andmebaasid II)**

|  |  |
| --- | --- |
| Kasutatav andmebaasisüsteem  ja versioon | PostgreSQL 9.4.4 |
| Rakenduse loomise vahendid  (arenduskeskkonnad, keeled ja  nende versioonid) | MS Access 2013 |
| Andmebaasi server | apex.ttu.ee |
| PostgreSQL andmebaasi nimi | t134683 |
| Millise töökoha rakendus  realiseerib? | Juhataja |
| Kui tegemist on kahekihilise  klient-server süsteemiga ja  rakendus on kasutaja arvutis, siis  mida peab kasutaja arvutis  tegema, et seda rakendust  kasutada | "Rakenduse toimimiseks peab kasutaja arvutis olema installeeritud PostgreSQL ODBC draiverid ja loodud andmeühenduse spetsifikatsioon (data source) nimega PostgreSQL\_yhendus. Andmeühenduse spetsifikatsioonis peavad server, andmebaas, kasutajanimi ja parool olema määramata." [Eessaar, 2015] |
| Kasutajanimi ja parool  rakendusse sisselogimiseks | Kasutajanimi: juhataja1  Parool: parool1 |
| Millistes vormides toimub pöördumine andmebaasiserveris  talletatud rutiinide poole? Faili puhul palun näidata ka kataloog. | Ekraanivormil „Sissereg“ pöördutakse funktsiooni f\_sisselogimine\_juhataja, et identifitseerida juhataja. |
| Millistes vormides toimub pöördumine andmebaasis loodud vaadete poole? Faili puhul palun näidata ka kataloog. | Vormil „Treeningu\_võimalikud\_läbiviijad“, pöördutakse andmete saamiseks vaate „treeningute\_labiviijad“ poole ning vormidel „Treeningute\_haldamine“ ja „Treeningute\_haldamine\_alamvorm“ vaate „koik\_treeningud“ poole |
| Kuhu (millisesse moodulisse,  faili) on rakenduses kirjutatud  andmebaasisüsteemiga  ühenduse loomiseks mõeldud  andmebaasi kasutaja nimi (see  peab olema minimaalsete  vajalike õigustega kasutaja)?  Faili puhul palun näidata ka  kataloog. | Kasutaja nimi asub Refreshing mooduli alguses. |
| 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 (tabelis Isik veergudes **kasutajanimi**ja **parool**). |
| Kuidas on andmebaasis  salvestatud parool kaitstud? | "Andmebaasis hoitakse parooli räsiväärtust. See väärtus leitakse andmebaasisüsteemi poolt kasutades sisendina avatekstina parooli ning süsteemi-genereeritud soola.  Parooli räsiväärtuse leidmiseks kasutatakse PostgreSQL funktsiooni crypt koos soola genereerimiseks mõeldud funktsiooniga gen\_salt. Räsiväärtuse leidmiseks kasutatakse Blowfish plokkšifril (krüptograafilisel algoritmil) põhinevat algoritmi, mille puhul võib parooli pikkus olla kuni 72 märki. Seda algoritmi eelistati, kuna Cybernetica AS (2013) hindab selle viie aasta jooksul turvaliseks primitiiviks (erinevalt algoritmidest DES ja MD5, mida crypt samuti toetab).  Veerus parool lubatud maksimaalne väärtuse suurus peab valitud algoritmi korral olema 60 märki (veerg on tüüpi VARCHAR(60)).  gen\_salt funktsioon võimaldab lisaks algoritmile määrata ka algoritmi kasutatavat korduste arvu (kui algoritm seda võimaldab). Mida suurem korduste arv, seda rohkem võtab aega räsiväärtuse arvutamine, aga ka selle murdmine. Blowfish algoritmi korral on vaikimisi korduste arv 6, kuid võimalik korduste arvu vahemik on 4 kuni 31. Määran korduste arvuks 11, mille puhul räsiväärtuse leidmisel ja kontrollimisel veel olulist töökiiruse langust ei täheldatud.  Funktsiooni crypt kasutamiseks on PostgreSQL andmebaasis CREATE EXTENSION lauset kasutades installeeritud lisamoodul pgcrypto. Installeerimise tulemusel loodavad skeemiobjektid paigutatakse skeemi public.  CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;" [Eessaar, 2015] |
| Kuidas toimub kasutaja  autentimine? | Kontrollitakse kasutaja sisestatud parooli, mida võrreldakse andmebaasis oleva, paroolist genereeritud, räsiväärtusega. Lisaks kontrollitakse, et kasutaja ametikohaks oleks juhataja ja et ta ei oleks töölt lahkunud(seisund „Lopetatud“). Seda kontrolli teeb andmebaas kasutades funktsiooni „f\_sisselogimine\_juhataja(text, text)“.  Funktsiooni väljakutsel on esimene argument kasutajanimi ja teine argument parool.  Kui need tingimused 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.  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. |
| Kas olete oma töö iseseisva töö  hindamismudeli alusel enne  esitamist ise üle kontrollinud? | Jah |

# Strateegia etapp

## Terviksüsteemi üldvaade

### Organisatsiooni eesmärgid

1. Teenida omanikele kasumit
2. Pakkuda parimat hinna ja kvaliteedi suhet.
3. Võimalus valida mitmete erinevate võimaluste vahel, kombineerides kliendile parimat sobivat paketi.
4. Pakkuda klientidele mitmekesiseid sportimisvõimalusi
5. Propageerida tervislikku ja paremat elustiili.
6. Pakkuda alati huvitavat ja innovatiivset treeningprogramme
7. Pakume võimalust kasutada personaaltreenerite abi

### Infosüsteemi eesmärgid

**Funktsionaalsed** eesmärgid:

1. *(****Põhiobjekt****:* ***treening*** *ja* ***treeningu******toimumine****)* Kasutaja saab internetis broneerida treeningut rühmatrenni valides sobiva aja ja sobiva treeningu(kuutasu eest)
   1. **Mittefunktsionaalne** eesmärk: 80% rühmatreeningute broneeringutest tehakse internetis või kohapeal olevatest arvutitest, mitte telefoni teel. See aitab hoida palgakulusid kokku.
2. Kasutaja saab internetis broneerida eratreeneriga jõusaali eratreeningut valides sobiva aja ja treeneri(väljaspool kuutasu)
   1. **Mittefunktsionaalne** eesmärk: 80% rühmatreeningute broneeringutest tehakse internetis või kohapeal olevatest arvutitest, mitte telefoni teel. See aitab hoida palgakulusid kokku.
3. Internetis näha statistikat kasutajate rühmatreeningutest: millised on populaarsed rühmatrennid, millised populaarsed rühmatreenerid.
   1. **Mittefunktsionaalne** eesmärk: Keskmine päringute tegemise aeg 2 sekundit.
4. Internetis näha statistikat kasutajate eratreeningutest: millised populaarsed ajad, millised populaarsed treenerid.
   1. **Mittefunktsionaalne** eesmärk: Keskmine päringute tegemise aeg 2 sekundit.
5. Võimaldada töötajatel uusi treeninguid elektrooniliselt registreerida
6. Võimaldada töötajatel elektrooniliselt treeninguid muuta
7. Saada ülevaade treeningutel käivatest klientidest
8. Võimaldada elektroonilist dokumendihaldust
9. Saada ülevaade spordiklubi ruumidest
10. Saada ülevaade spordiklubi varadest

### Lausendid

1. Töötaja on isik
2. Töötaja osaleb spordiklubi töös
3. Klient on isik
4. Klient osaleb treeningutel
5. Kliendile esitatakse arve
6. Klient tasub arve
7. Treeningpindu kasutatakse treeningute jaoks
8. Treening on teenus
9. Treeninguid on kindel arv
10. Treeningud toimuvad töötajate poolt määratud aegadel
11. Varustust kasutatakse treeningutel
12. Spordiklubi omab varustust
13. Spordiklubi valduses on ruumid
14. Spordiklubi seob lepinguid partneritega
15. Partner pakub spordiklubile teenust
16. Partner esitab spordiklubile arve
17. Spordiklubi tasub arve
18. Treening toimub ruumis
19. Võetakse kasutusele uus klassifikaator

### Põhiobjektid

1. Osapool
2. Töötaja
3. Klient
4. Treeningpinnad/ruumid
5. Treening
6. Treeningu toimumine
7. Varustus
8. Partner
9. Leping
10. Tellimus partnerile
11. Arve
12. Klassifikaator

### Põhiprotsessid

1. Treeningule registreerimine
2. Treeningu registreeringu eemaldamine
3. Uute treeningute registreerimine
4. Treeningute aktiivseks muutmine
5. Treeningute mitteaktiivseks muutmine
6. Treeningute muutmine
7. Treeningute kustutamine
8. Treeningute statistika moodustamine
9. Töötaja töölevõtmine
10. Klassifikaatori registreerimine
11. Arve väljastamine
12. Arve tasumine

### Põhilised sündmused

1. Administraator registreerib uue treeningu, treeneri soovil ja juhtkonna heakskiidul
2. Administraator kustutab treeningu juhtkonna soovil, kuna ei ole piisavalt registreeritud kliente antud treeningule
3. Administraator kustutab treeningu juhtkonna soovil, kuna spordiklubil pole piisavalt vahendeid treeningu jätkamiseks.
4. Administraator kustutab treeningu juhtkonna soovi, seoses treeneri lahkumisega
5. Treening muudetakse mitteaktiivseks, kuna pole piisavalt registreeringuid antud treeningule
6. Treening muudetakse mitteaktiivseks, kuna spordiklubil pole piisavalt vahendeid treeningu jätkamiseks
7. Treening muudetakse mitteaktiivseks, seoses treeneri lahkumisega
8. Treening muudetakse mitteaktiivseks, kuna treener ei suuda treeningut läbi viia
9. Administraator muudab treeningu aktiivseks, juhtkonna ja treeneri heakskiidul
10. Administraator muudab treeningut juhtkonna soovil
11. Treener muudab enda treeningut
12. Klient registreerib treeningule
13. Treeningu toimumine
14. Kliendile koostatakse arve
15. Klient tasub arve
16. Võetakse kasutusele uus klassifikaatori väärtus

### Tegutsejad

Süsteemisisesed tegutsejad:

1. Omanikud – otsustamaks kasumlike laienduste kohta
2. Juhatuse liikmed – koondaruanded töö edenemisest
3. Müügijuht (Teadmaks populaarseid treeninguid ja treenereid)
4. Personalijuht (Populaarsed treenerid, eksklusiivsust kindlustavad lepingud jms)
5. Ostujuht(Treeningmasinate, treeningvarustus, muusikavarustus)
6. Treenerid et oma graafikuid efektiivselt kujundada ja õppida mis treeningud on populaarsed
7. Administraatorid et oskaks soovitada populaarsemaid treeninguid ja populaarseid treenerid

Süsteemivälised tegutsejad:

1. Internetis registreeritud klient näeb statistikat oma treeningute kohta
2. Külaline(registreerimata) näeb võimalikke treeningaegasid ja spordiklubis töötavaid treenereid

### Asukohad

* Kliendid kasutavad spordiklubi treeningruume ja riietusruume.
* Töötajad töötavad spordiklubis neile ettenähtud ruumides

### Terviksüsteemi tükeldus allsüsteemideks

#### Pädevusalad

Organisatsiooni sisesed pädevusalad.

1. Omaniku pädevusala
2. Juhatuse liikme pädevusala
3. Müügijuhi pädevusala
4. Personalijuhi pädevusala
5. Ostujuhi pädevusala
6. Treeneri pädevusala
7. Administraatori pädevusala

Organisatsiooni välised pädevusalad.

1. Kliendi pädevusala

#### Funktsionaalsed allsüsteemid

Sisulised allsüsteemid (seotud organisatsiooni põhitegevusega).

* Treeningute funktsionaalne allsüsteem
* Treeningute toimumise funktsionaalne allsüsteem

Administratiivsed allsüsteemid (võivad olla kasutusel paljudes erinevates organisatsioonides, mille tegevusalad ja eesmärgid on erinevad).

* Osapoolte funktsionaalne allsüsteem
* Töötajate funktsionaalne allsüsteem
* Klientide funktsionaalne allsüsteem
* Ruumide funktsionaalne allsüsteem
* Varustuse funktsionaalne allsüsteem
* Partnerite funktsionaalne allsüsteem
* Lepingute funktsionaalne allsüsteem
* Tellimuste funktsionaalne allsüsteem
* Arvete funktsionaalne allsüsteem
* Klassifikaatorite funktsionaalne allsüsteem

#### Registrid

Sisulised registrid (seotud organisatsiooni põhitegevusega).

* Treeningute register
* Treeningute toimumise register

Administratiivsed registrid (võivad olla kasutusel paljudes erinevates organisatsioonides, mille tegevusalad ja eesmärgid on erinevad).

* Osapoolte register
* Töötajate register
* Klientide register
* Ruumide register
* Varustuse register
* Partnerite register
* Lepingute register
* Tellimuste register
* Arvete Register
* Klassifikaatorite register

## Treeningute funktsionaalse allsüsteemi eskiismudelid

### Eesmärgid

1. Võimaldada klientidel saada ülevaate treeningutest
2. Võimaldada klentidel registreerida treeningutele
3. Võimaldada klientidel näha statistikat treeningutest
4. Võimaldada töötajatel(treener ja administraator) treeningut tühistada
5. Võimaldada töötajatel(treener ja administraator) treeningut muuta
6. Võimaldada töötajatel(juhataja, treener ja administraator) saada ülevaate treeningutest

### Allsüsteemi kasutavad pädevusalad

Organisatsiooni sisesed pädevusalad.

1. Omaniku pädevusala (kasutusjuhtide eskiismudelil juhataja)
2. Juhatuse liikme pädevusala (kasutusjuhtide eskiismudelil juhataja)
3. Müügijuhi pädevusala (kasutusjuhtide eskiismudelil juhataja)
4. Personalijuhi pädevusala (kasutusjuhtide eskiismudelil juhataja)
5. Ostujuhi pädevusala (kasutusjuhtide eskiismudelil juhataja)
6. Treeneri pädevusala
7. Administraatori pädevusala

Organisatsiooni välised pädevusalad.

1. Kliendi pädevusala

### Allsüsteemi poolt kasutatavad registrid

Sisulised registrid (seotud organisatsiooni põhitegevusega).

* Treeningute register

Administratiivsed registrid (võivad olla kasutusel paljudes erinevates organisatsioonides, mille tegevusalad ja eesmärgid on erinevad).

* Osapoolte register
* Töötajate register
* Klientide register
* Klassifikaatorite register

### Allsüsteemi ühe põhiprotsessi tegevusdiagramm



### Allsüsteemi kasutusjuhtude eskiismudel



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

Järgnevalt esitatakse kasutusjuhtude tekstikirjeldused kõrgtaseme formaadis.

**Kasutusjuht**: Kasutaja tuvastamine

**Tegutsejad:** Klient, juhataja, treener, administraator

**Kirjeldus:** Subjekt identifitseerib ennast. Selleks sisestab ta kasutajanime, parooli ja oma rolli süsteemis. Süsteem autendib subjekti, st. kontrollib subjekti väidetavat identiteeti. Kui subjekt on identifitseeritud, siis lubatakse subjekt süsteemi siseneda, vastasel juhul mitte.

*Märkus:* Kasutusjuhtu "Kasutaja tuvastamine" kasutatakse kõikides

funktsionaalsetes allsüsteemides.

**Kasutusjuht**: Treeningute registreerimine

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator lisab uue treeningu. Administraator seadistab maksimaalse inimeste arvu kes treeningule saavad registreerida.

**Kasutusjuht**: Treeningute kustutamine

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator kustutab treeningu täielikult süsteemist sest treeningut enam ei toimu. Kui klient on selleks treeninguks end juba registreerinud, siis saadab süsteem kliendile sellekohase teate.

**Kasutusjuht**: Treeningute aktiivseks muutmine

**Tegutsejad:** Administraator

**Kirjeldus:** Treeningu aktiivseks muutmine. Näiteks lühiajaliselt treeneri haigestumise korral või pikaajaliselt seoses väiksemate klientide arvuga suvel.

**Kasutusjuht**: Treeningute muutmine

**Tegutsejad:** Administraator, Treener

**Kirjeldus:** Treeningu andmete muutmine näiteks treeneri, ruumi või treeningu nimetuse muutmisel.

**Kasutusjuht**: Võimaliku läbiviija lisamine

**Tegutsejad:** Administraator

**Kirjeldus:** Treeningu läbiviija lisamine administraatori poolt.

**Kasutusjuht**: Võimaliku läbiviija muutmine

**Tegutsejad:** Administraator

**Kirjeldus:** Treeningu läbiviija muutmine administraatori poolt.

**Kasutusjuht**: Treeningute mitteaktiivseks muutmine

**Tegutsejad:** Administraator, Treener, Juhataja

**Kirjeldus:** Treeningu ajutiselt mitteaktiivseks muutmine. Näiteks lühiajaliselt treeneri haigestumise korral või pikaajaliselt seoses väiksemate klientide arvuga suvel.

**Kasutusjuht**: Treeningute vaatamine

**Tegutsejad:** Administraator, Treener, Juhataja

**Kirjeldus:** Administraator, treener ja juhataja vaatavad kõiki treeninguid. Saab vaadata treeningutele registreerinud kliente.

**Kasutusjuht**: Kliendi poolt aktiivsete treeningute vaatamine

**Tegutsejad:** Klient

**Kirjeldus:** Klient vaatab kõiki aktiivseid treeninguid kuhu ta registreeritud on.

### Mittefunktsionaalsed nõuded

| Tüüp | Nõude kirjeldus |
| --- | --- |
| andmebaasi-süsteem | Süsteem kasutab andmete hoidmiseks SQL-andmebaasisüsteemi abil loodud andmebaasi.  Serverite operatsioonisüsteemiks on Linux või Windows.  Andmebaasisüsteemiks kasutada tasuta PostgreSQLi või Microsofti Accessi. |
| arendusvahendid | Arendusvahendina kasutada CASE tarkvara Rational Rose. |
| keel | Süsteemi kasutajaliides ja dokementatsioon peavad olema eesti keeles, samuti tuleb süsteemi ehitades arvestada sellega, et uute keelte lisamist kasutajaliidesesse ei oleks liialt raske implementeerida. |
| kasutajaliides | Kasutajaliides on veebipõhine. Kujundus pole oluline. |
| töökiirus | Päringute 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 | Süsteem peab olema töökindel ja tõrgeteta.  Taasteaja siht (*recovery time objective*)("maksimaalne talutav süsteemi käideldamatuse kestus pärast intsidenti" (AKIT, 2015)): Süsteemi taastamiseks kuluv aeg peaks jääma alla kahe tunni.  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, 2015)): Maksimaalselt võivad kaotsi minna viimase 24 tunni andmed. |
| varukoopiad | Kord päevas tuleks teha andmebaasis olevatest andmetest varukoopia. |
| turvalisus | Määran vastuvõtuaegade haldamise süsteemi turvaklassi  põhinedes infosüsteemi turvameetmete süsteemil:  https://www.riigiteataja.ee/akt/13125331?leiaKehtiv  Vastuvõtuaegade haldamise allsüsteem teenindab vastuvõtuaegade registrit. Vastuvõtuaegade registri turvaklass  võiks olla:  K1T1S2  **K1** – töökindlus – 90% (lubatud summaarne seisak nädalas ~  ööpäev); lubatav nõutava reaktsiooniaja kasv tippkoormusel –  tunnid (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;  Kui parooli hoitakse andmebaasis, siis ei tohi see olla avatekst,  vaid peab olema parooli räsiväärtus, mis on leitud selle parooli  jaoks genereeritud soola kasutades. |

## Treeningute funktsionaalse allsüsteemi teenindatava treeningute registri eskiismudelid

### Eesmärgid

Säilitada informatsiooni treeningutest selliselt, et oleks tagatud treeningute allsüstemis defineeritud eesmärkide täitmine

### Registrit kasutavad pädevusalad

Registrist loevad andmeid kliendid ja töötajad. Registris muudavad andmeid töötajad.

### Registrit teenindavad funktsionaalsed allsüsteemid

Treeningute registrit teenindab treeningute funktsionaalne allsüsteem.

Treeningute toimimise registrit teenindab treeningute toimumise funktsionaalne allsüsteem ja treeningute funktsionaalne allsüsteem. Kui treeningu andmine lõpetatakse jäädavalt siis selle täitmise korral pöördutakse ka treeningute toimimise registri poole, et muuta selle treeningu toimumise aktiivset seisundit.

### Infovajadused, mida register aitab rahuldada

1. Iga treeneri rühmatreeningute nimekiri
2. Iga treeneri eratreeningute nimekiri
3. Vabade treeningute nimekiri
4. Klientide poolt registreeritud treeningute nimekiri

### Seosed teiste registritega

Sisulised registrid (seotud organisatsiooni põhitegevusega).

* Treeningute register

Administratiivsed registrid (võivad olla kasutusel paljudes erinevates organisatsioonides, mille tegevusalad ja eesmärgid on erinevad).

* Osapoolte register
* Töötajate register
* Klientide register
* Klassifikaatorite register

### Ärireeglid

1. Ainult juhatuse otsusega saab lõpetada treening jäädavalt.
2. Kliendid ja töötajad peavad omama Eesti Vabariigi isikukoodi.
3. Andmeid ei saa muuta treeningutel, mille seisund on "Lõplikult peatatud".
4. Ühtegi juba registreeritud treeningut ei tohi andmebaasist ära kustutada.
5. Uusi treeninguid saab registreerida ainult aktiivses seisundis tootaja.
6. Ainult administraator saab registreerida uusi treeninguid.

### Registri kontseptuaalne eskiismudel



# Detailanalüüsi etapp

## Treeningute funktsionaalse allsüsteemi detailanalüüs

### Kasutusjuhtude mudel

Vastuvõtuaegade funktsionaalse allsüsteemi kasutusjuhtude diagramm (vt joonis 2).

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

**Kasutusjuht**: Kasutaja tuvastamine

**Primaarne tegutseja:** Klient, juhataja, treener, administraator(edaspidi subjekt)

**Osapooled ja nende huvid:**

* Klient, juhataja, treener või administraator: Soovivad siseneda süsteemi ja teha tegevusi neile antud volituste piires.

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

**Eeltingimused:** Subjekt on süsteemis kasutajaks registreeritud.

**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.

**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 **(OP1.1)**.

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

5a. Kui süsteem ei leia esitatud volitustõendiga subjekti, siis ei saa subjekt õigust süsteemi kasutada.

**Kasutusjuht**: Treeningute registreerimine

**Primaarne tegutseja:** Administraator, juhataja

Osapooled ja nende huvid:

* Klient: On huvitatud, et võimalikult varakult saaks teada uute treeningute toimumisest.
* Administraator/juhataja: On huvitatud, et klient saaks võimalikult vara teada treeningute lisandumisest.

**Käivitatav sündmus:** Juhataja soovib uut treeningut registreerida.

**Eeltingimused:** Administraator/juhataja on identifitseeritud.

**Järeltingimused:** Treening on registreeritud.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1. Treener, administraator või juhataja avaldab soovi lisada uus treening.
2. Administraator või juhataja sisestab treeningu andmed (nimetus, spordiala, kirjeldus).
3. **Süsteem** salvestab treeningu andmed ja treeningut registreeriva töötaja isikukoodi ja nime. **(OP5.1)**

*Administraator/ juhataja võivad korrata samme 2-3 nii mitu korda kui soovivad.*

**Kasutusjuht**: Treeningute kustutamine

**Primaarne tegutseja:** Administraator

**Osapooled ja nende huvid:**

* **Klient:** Soovib saada võimalikult kiiresti informatsiooni ära lõpetatud treeningutest.
* **Administraator:** Soovib kustutada treeningud mida spordiklubil ei ole võimalik anda.

**Käivitatav sündmus:** Administraator soovib kustutada treeningu kuna spordiklubil ei ole võimalik seda mingil põhjusel anda.

**Eeltingimused:** Administraator on identifitseeritud. Treening on registreeritud.

**Järeltingimused:** Treening on kustutatud.

Stsenaarium (tüüpiline sündmuste järjestus):

1. Administraator avaldab soovi kustutada treening
2. Administraator valib aktiivse või mitteaktiivse treeningu mille soovib kustutada
3. **Süsteem** kuvab treeningu andmed **(OP3.1)**. Samuti kuvab **süsteem** treeningule registreerinud klientide nimed, e-meili aadressid. **(OP3.3)**
4. Administaator annab korralduse tühistada konkreetne treening
5. **Süsteem** küsib, kas administraator soovib kindlasti treeningut tühistada.
6. Administraator kinnitab oma soovi kustutada treening.
7. **Süsteem** saadab kliendile, kes oli registreerunud kustutatud treeningule selle kohta e-meili ning tühistab treeningu. **(OP5.4)**

**Kasutusjuht**: Treeningute aktiivseks muutmine

**Tegutsejad:** Administraator, Treener, Juhataja

**Osapooled ja nende huvid:**

* Administraator, treener, juhataja: Soovib, et klientidel oleks võimalikult täpne informatsioon treeningute kohta.
* Klient:Soovib saada korrektset informatsiooni treeningute kohta.

**Käivitav sündmus:** Administraator, treener või juhataja soovib muuta treeningu ajutiselt aktiivseks.

**Eeltingimused:** Administraator,juhataja või treener on identifitseeritud. Treeningu andmed on registreeritud.

**Järeltingimused:** Treening on muudetud aktiivseks.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1. Treener, juhataja või administraator avaldab soovi muuta treeningu aktiivseks
2. Treener või administraator on valinud aktiivse treeningu, mille soovib muuta aktiivseks.
3. **Süsteem** kuvab treeningu andmed **(OP3.1)** ning treeningule registreerinud klientide andmed ja e-meili aadressid. **(OP3.3)** Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. **(OP2.2).**
4. Treener või administraator valib vajadusel treeningu mille soovib muuta aktiivseks.
5. **Süsteem** salvestab treeningu andmed **(OP5.3).**

**Kasutusjuht**: Treeningute muutmine

**Primaarne tegutseja:** Administraator või Treener

**Osapooled ja nende huvid:**

* Administraator, treener: Soovib, et klientidel oleks võimalikult täpne informatsioon treeningute kohta.
* Klient:Soovib saada korrektset informatsiooni treeningute kohta, et leida endale sobiv treening.

**Käivitav sündmus:** Administraator või treener soovib muuta treeningu nime, kirjeldust või soovib muuta ruumi, kus toimub treening.

**Eeltingimused:** Administraator või treener on identifitseeritud. Treeningu andmed on registreeritud.

**Järeltingimused:** Treeningu nimi või kirjeldus on muudetud või treeninguga seotud ruum.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1.Treener või administraator avaldab soovi muuta treeningu andmeid

2.Treener või administraator on valinud aktiivse treeningu, mille andmeid ta soovib muuta.

**3.Süsteem** kuvab treeningu andmed **(OP3.1).** ning treeningule registreerinud klientide andmed ja e-meili aadressid. **(OP3.3)** Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. **(OP2.2).**

4.Treener või administraator muudab vajadusel valitud treeningu andmeid.

**5.Süsteem** salvestab treeningu andmed **(OP5.5).**

**Kasutusjuht**: Võimaliku läbiviija lisamine

**Primaarne tegutseja:** Administraator

**Osapooled ja nende huvid:**

* Administraator, treener, juhataja: Soovivad, et õige treeningu läbiviija määratakse treeningule kohe kui vastav informatsioon on teada.
* Klient:Soovib saada kiiresti korrektset informatsiooni treeningu läbiviija kohta, et valida endale sobiv treening.

**Käivitav sündmus:** Treener või juhataja annab teada administraatorile, et on vaja lisada treeningule läbiviija.Administraator soovib lisada läbiviija nime, kes treeningut annab.

**Eeltingimused:** Administraator ja võimalik läbiviija on identifitseeritud. Treeningu andmed on registreeritud.

**Järeltingimused:** Treeningu võimalik läbiviija on lisatud. Treening on seotud Võimalik\_läbiviija.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1. Administraator avaldab soovi lisada treeningule läbiviija,
2. Administraator on valinud aktiivse treeningu, mille andmeid ta soovib muuta.
3. **Süsteem** kuvab treeningu andmed **(OP3.1).** ning treeningule registreerinud klientide andmed ja e-meili aadressid. **(OP3.3)** Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. **(OP2.2).**
4. Treener või administraator valib vajadusel treeningule treeneri.
5. **Süsteem** salvestab treeningu andmed **(OP5.6).**

**Kasutusjuht**: Võimaliku läbiviija muutmine

**Primaarne tegutseja:** Administraator

**Osapooled ja nende huvid:**

* Administraator, treener, juhataja: Soovivad, et õige treeningu läbiviija määratakse treeningule kohe kui vastav informatsioon on teada.
* Klient:Soovib saada kiiresti korrektset informatsiooni treeningu läbiviija kohta, et valida endale sobiv treening.

**Käivitav sündmus:** Treener või juhataja annab teada administraatorile, et on vaja muuta treeningule läbiviija.Administraator soovib muuta läbiviija nime, kes treeningut annab.

**Eeltingimused:** Administraator ja võimalik läbiviija on identifitseeritud. Treeningu andmed on registreeritud.

**Järeltingimused:** Treeningu võimalik läbiviija on muudetud. Treening on seotud Võimalik\_läbiviija.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1. Administraator avaldab soovi muuta treeningule läbiviijat.
2. Administraator on valinud aktiivse treeningu, mille andmeid ta soovib muuta.
3. **Süsteem** kuvab treeningu andmed **(OP3.1).** ning treeningule registreerinud klientide andmed ja e-meili aadressid. **(OP3.3)** Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. **(OP2.2).**
4. Administraator muudab vajadusel treeningule võimalikku läbiviijat.
5. **Süsteem** salvestab võimaliku läbiviija muudatused **(OP5.7).**

**Kasutusjuht**: Treeningute mitteaktiivseks muutmine

**Tegutsejad:** Administraator, Treener, Juhataja

**Osapooled ja nende huvid:**

* Administraator, treener, juhataja: Soovib, et klientidel oleks võimalikult täpne informatsioon treeningute kohta.
* Klient:Soovib saada korrektset informatsiooni treeningute kohta.

**Käivitav sündmus:** Administraator, treener või juhataja soovib muuta treeningu ajutiselt mitteaktiivseks.

**Eeltingimused:** Administraator,juhataja või treener on identifitseeritud. Treeningu andmed on registreeritud.

**Järeltingimused:** Treening on muudetud miiteaktiivseks.

**Stsenaarium** (tüüpiline sündmuste järjestus):

1. Treener, juhataja või administraator avaldab soovi muuta treeningu mitteaktiivseks
2. Treener või administraator on valinud aktiivse treeningu, mille soovib muuta mitteaktiivseks.
3. Süsteem kuvab treeningu andmed (OP3.1) ning treeningule registreerinud klientide andmed ja e-meili aadressid. (OP3.3) Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. (OP2.2).
4. Treener või administraator valib vajadusel treeningu mille soovib muuta mitteaktiivseks.
5. Süsteem salvestab treeningu andmed (OP5.2).

**Kasutusjuht**: Treeningute vaatamine

**Primaarne tegutseja:** Administraator, Treener või Juhataja

**Osapooled ja nende huvid:**

* Administraator, Treener või Juhataja: Soovib täpset ulevaadet enda välja kuulutatud aktiivsetest treeningutest ja mitteaktiivsetest treeningutest, et treeningud oleksid hästi korraldatud.
* Klient: Soovib, et spordiklubi oleks teadlik enda poolt välja kuulutatud treeningutest ning oleks treeningu alguseks valmis treeningut andma.

**Käivitav sündmus:** Administraator, treener või juhataja soovib teada, milliseid treeninguid on spordiklubi välja kuulutanud.

**Eeltingimused:** Administraator, treener või juhataja on identifitseeritud. Kõik aktiivsed ja mitteaktiivsed treeningud on registreeritud.

**Järeltingimused:** On leitud kõik aktiivsed ja mitteaktiivsed treeningud.

**Stsenaarium(tüüpiline sündmuste järjestus):**

1. Administraator, treener või juhataja avaldab soovi vaadata aktiivseid ja mitteaktiivsed treeninguid.
2. Administraator, treener või juhataja valib konkreetse treeningu.
3. **Süsteem** kuvab treeningu andmed **(OP3.1)** ning treeningule registreerinud klientide andmed ja e-meili aadressid. **(OP3.3)** Süsteem kuvab ruumide nimekirja ja treenerite nimekirja. **(OP2.2).**

**Kasutusjuht**: Aktiivsete treeningute vaatamine

**Primaarne tegutseja:** Klient

**Osapooled ja nende huvid:**

* Klient: Soovib teada, millistele treeningutele ta on ennast registreerinud, et jõuda õigeks ajaks kohale.
* Spordiklubi: Soovib et kliendid teaksid, millistele vastuvõtuaegadele nad on registreerinud ning tuleksid kohale õigeks ajaks.

**Käivitav sündmus:** Klient soovib teada, millistele treeningutele ta on registreerinud.

**Eeltingimused:** Klient on identifitseeritud. Kliendiga seotud treeningule tulek ning sellega seotud treeninguaeg on registreeritud.

**Järeltingimused:** On leitud kõik kliendiga seotud treeningule tulekud.

**Stsenaarium (tüüpiline sündmuste järjestus):**

1.Klient avaldab soovi vaada enda aktiivseid treeninguid.

**Kirjeldus:** Klient vaatab kõiki aktiivseid treeninguid kuhu ta registreeritud on.

## Treeningute funktsionaalse allsüsteemi kasutatavate registrite detailanalüüs

### Kontseptuaalne andmemudel

Kontseptuaalses andmemudelis peab olema kirjeldatud vähemalt seitse olemitüüpi.

#### Olemi-suhte diagramm

Punasega on tähistatud registri (antud juhul vastuvõtuaegade registri) põhiobjekt. Kollasega on tähistatud registrisse (antud juhul vastuvõtuaegade registrisse) kuuluvad mitte-põhiobjektid.

Rohelisega on tähistatud teistesse registritesse kuuluvad objektid, mida on antud juhul vaja vastuvõtuaegade funktsionaalse allsüsteemi toimimise tagamiseks.





#### 

#### Olemitüüpide definitsioonid

| **Olemitüübi nimi**  **(aliased)** | **Kuuluvus registrisse** | **Definitsioon** |
| --- | --- | --- |
| Amet | Klassifikaatorite register | Spordiklubi töötaja töölepingus sätestatud Ametikohustuse üldnimetus. Nt. Administraator, treener. |
| Treeningu\_seisundi\_liik | Klassifikaatorite register | Näitab, kas antud treening on aktiivne või mitteaktiivne. |
| Kliendi\_seisundi\_liik | Klassifikaatorite register | Kliendi olek süsteemis, kas ta on registreeritud treeningule või mitte. |
| Spordiala | Klassifikaatorite register | Ala, millega tegeletakse antud treengingul. |
| Töötaja\_seisundi\_liik | Klassifikaatorite register | Näitab, kas töötaja on spordiklubis saadaval(nt. Ei ole haige või koolitusel) |
| 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) |
| Treening | Treeningute register | Sporditreeningu liik, kus treener kohtub klientidega, et läbi viia vastavad harjutused ja ülesanded. Treening on nagu õppeaine koolis. |
| Võimalik\_läbiviija | Treeningute register | Isik, kes viib treeningut läbi. |
| Klient | Klientide register | Spordiklubi külastav ja treeningutel osalev isik. |
| Töötaja | Töötajate register | Spordiklubis töötav isik. |
| Isik | Osapoolte register | Mistahes spordiklubiga seotud inimene. |

#### Atribuutide definitsioonid

| **Olemitüübi nimi** | **Atribuudi nimi** | **Atribuudi definitsioon** | **Näiteväärtus** |
| --- | --- | --- | --- |
| Isik | Eesnimi | "Lapsele pärast sündi (registreerimisel)  pandav nimi, osa isikunimest. Eesnimi  asetseb harilikult perekonnanime ees,  harva järel (nt Ungari pruugis)."  (http://mt.legaltext.ee/esterm/concept.a  sp?conceptID=196&term=eesnimi)  {Registreerimine on kohustuslik.  Eesnimi ei tohi olla tühi string või  ainult tühikutest koosnev string.  Eesnimi võib sisaldada ainult tähti, tühikuid või kriipse.  Eesnimi algab suure tähega.  Ülejäänud tähed on väikesed.  Kui inimesel on mitu eesnime, siis  need on eraldatud ühe tühiku või  kriipsuga. Ka iga järgnev eesnimi algab  suure tähega.} | Maali |
| Isik | Perenimi | "Nimi, mis on isikul ühine teiste tema  perekonna liikmetega"  (http://mt.legaltext.ee/esterm/concept.a  sp?conceptID=997&term=perekonnani  mi)  {Registreerimine on kohustuslik.  Perenimi ei tohi olla tühi string või  ainult tühikutest koosnev string.  Perenimi võib sisaldada ainult tähti,  tühikuid või kriipse. Perenimi algab  suure tähega. Ülejäänud tähed on  väikesed. Kui inimesel on mitu  perenime, siis need on eraldatud ühe tühiku või kriipsuga. Ka iga järgnev perenimi algab suure tähega.} | Tamm |
| Isik | Elukoht | Isiku alalise elukoha aadress.  "Koha-aadress on territooriumi  haldusjaotuse hierarhiast ja ametlikest  kohanimedest lähtuv aadressobjekti  tekstilis-numbriline kirje või tunnus.  Ühele objektile võib määrata mitu  koha-aadressi. Ühele objektile  määratud koha-aadressid on  paralleelaadressid." ("Aadressandmete  süsteemi kehtestamine")  Näide: Tallinn, 34124, Ehitajate tee 62-  12. Harjumaa, Viimsi vald, Kaku küla,  Laane talu.  {Elukoht ei ole kohustuslik. Elukoht ei tohi olla tühi string, ainult  tühikutest koosnev string või ainult  numbritest koosnev string.} | Tartu, Herne tn, 15 |
| Isik | Kasutajanimi | Isikule antud unikaalne nimi, mida  kasutatakse tema autentimiseks.  {Isiku unikaalne identifikaator.  Registreerimine on kohustuslik.  Kasutajanimi ei tohi olla lühem kui  neli märki. Kasutajanimi ei tohi  sisaldada tühikuid.} | Spordimees12 |
| Isik | Parool | Isiku identsust tõendav teadmuslik  (miski, mida isik teab) volitustõend.  {Registreerimine on kohustuslik} | Parool123 |
| Isik | Isikukood | Eesti Vabariigi isikukood. "Isikukood on  isiku soo ja sünniaja alusel  moodustatud isiku üheselt  kindlaksmääramist võimaldav arv."  ("Rahvastikuregistri seadus”) "Oluline  on teada, et isikukoodid ei kuulu  delikaatsete isikuandmete alla.  Isikukoodid on tavalised isikuandmed  ja nende kasutamisele ei ole seatud  rohkem piiranguid kui näiteks inimese nime või sünniaja kasutamisele." (  "Andmekaitse inspektsiooni juhis.  Isikukoodi kasutamine")  {Isiku unikaalne identifikaator.  Registreerimine on kohustuslik.  Isikukood koosneb 11-st  numbrimärgist. Esimene märk on  numbrimärk vahemikus 3 kuni 6. See  on soo tähis. Neljas märk on number 0  või 1 See on sünnikuu esimene  number. Kuues märk on number  vahemikus 0 kuni 3. See on sünni  päeva esimene number. Seitsmes mark on number vahemikus 0-9. See on sünnipäeva teine number. Märgid 8 ja 9 tähistavad maakonna numbrit.} | 49207252705 |
| Isik | Email | Aadress, millele saab üle võrgu (ühest  arvutist või tööjaamast teise) saata  isikule mõeldud kirjalikke sõnumeid.  {Isiku unikaalne identifikaator.  Registreerimine on kohustuslik. e\_mail  peab sisaldama täpselt ühte "@" märki.} | kasutaja@gmail.com |
| Isik | Registreerimise aeg | Kuupäev ja kellaaeg, millal isik registreeriti.  {Registreerimine kohustuslik. Peab olema unikaalne.} | 10.10.2010 17:15 |
| Treening | Nimetus | Treeningu nimi, millega antakse edasi treeningu sisu.  {Registreerimine kohustuslik, ei tohi olla tühi string. Peab olema unikaalne} | Fitness |
| Treening | Kirjeldus | Vabas vormis kirjeldus, milline antud treening on.  {Ei tohi koosned ainult tühikutest, ei ole kohustuslik} | Bodypump on jõutreeningu programm, kus keskendutakse jõu- ja vastupidavusharjutustele, mida sooritatakse kangidega millel asuvad raskused. |
| Võimalik\_läbiviija | Alguse\_aeg | Antud treeningu antud isiku poolt läbiviimise perioodi esimene kuupäev.  {Registreerimine on kohustuslik } | 01.01.2015 |
| Võimalik\_läbiviija | Lõpu-aeg | Antud treeningu antud isiku poolt läbiviimise perioodi viimane kuupäev.  {Registreerimine on kohustuslik. Lõpu aeg ei tohi olla väiksem kui alguse aeg} | 14.12.2015 |
| Klassifikaator | Kood | Klassifikaatori väärtust esitav kood,  mida saab kasutada selle väärtuse  lühidalt esitamiseks. Kood võib olla  tekstiline või numbriline väärtus.  {Klassifikaatori unikaalne identifikaator.  Registreerimine on kohustuslik.} | 1 |
| Klassifikaator | Nimetus | Klassifikaatori väärtuse ametlik  nimetus.  {Klassifikaatori unikaalne identifikaator.  Registreerimine on kohustuslik.  Nimetus ei tohi olla tühi string või  ainult tühikutest koosnev string.} | aktiivne |
| Klassifikaator | Kirjeldus | Klassifikaatori kirjeldus  {Kirjeldus ei ole kohustuslik. Ei tohi koosned ainult tühikutest } | See kirjeldus kirjeldab andmeid. |

### Andmebaasioperatsioonide lepingud

Tuleb kirja panna andmete lisamise/muutmise/kustutamise operatsioonid, millele viidatakse laiendatud formaadis kasutusjuhtudes.

**OP5.1** **Treeningu registreermine** (**viide töötajale, viide spordialale, nimetus, kirjeldus)**

**Eeltingimused:**

- Töötaja ja Spordiala on registreeritud.

**Järeltingimused:**

**--Olemite registreerimine**

- On loodud Treening.

**--Atribuutide väärtustamine**

- Treening.nimetus = **nimetus.**

- Treening.kirjeldus = **kirjeldus.**

**--Seoste registreerimine**

- Treening on seotud Spordialaga mille alla treening liigitatakse.

- Treening on seotud Treeningu\_seisundi\_liigiga, mille nimetus = „AKTII“.

- Treening on seotud töötajaga, kes treeningu registreeris

**Kasutus kasutusjuhtude poolt:** Treeningu registreerimine.

**OP5.2 Treeningu mitteaktiivseks muutmine (treeningu\_id)**

**Eeltingimused:**

- Treening ja Treeningu\_seisundi\_liik on registreeritud.

- Treening on seotud Treeningu\_seisundi\_liigiga, mille nimetus = „AKTII“.

**Järeltingimused:**

**--Seoste registreerimine**

- Treening on seotud Treeningu\_seisundi\_liigiga, mille nimetus = „AJUTI“.

**Kasutus kasutusjuhtude poolt:** Treeningu mitteaktiivseks muutmine

**OP5.3 Treeningu aktiivseks muutmine (treeningu\_id)**

**Eeltingimused:**

- Treening ja Treeningu\_seisundi\_liik on registreeritud.

- Treening on seotud Treeningu\_seisundi\_liigiga, mille nimetus = „AJUTI“.

**Järeltingimused:**

**--Seoste registreerimine**

- Treening on seotud Treeningu\_seisundi\_liigiga, mille nimetus = „AKTII“.

Kasutus kasutusjuhtude poolt: Treeningu aktiivseks muutmine

**OP5.5 Treeningu andmete muutmine (treeningu\_id, nimetus, kirjeldus, viide spordialale)**

**Eeltingimused:**

- Treening ja Treeningu\_seisundi\_liik on registreeritud.

**Järeltingimused:**

**--Atribuutide väärtustamine**

- Treening.nimetus = **nimetus.**

- Treening.kirjeldus = **kirjeldus.**

- Treening on seotud Spordialaga mille alla treening liigitatakse.

**Kasutus kasutusjuhtude poolt:** Treeningu andmete muutmine.

**OP5.6 Võimaliku läbiviija lisamine (viide treeningule, viide töötajale, alguse\_aeg, lopu\_aeg)**

**Eeltingimused:**

- Treening ja töötaja on registreeritud.

**Järeltingimused:**

**--Seoste registreerimine**

- On loodud Võimalik\_läbiviija

Võimalik\_läbiviija.alguse\_aeg = alguse\_aeg

Võimalik\_läbiviija.lopu\_aeg = lopu\_aeg

Võimalik läbiviija on seotud Treeninguga

Võimalik läbiviija on seotud Töötajaga

**Kasutus kasutusjuhtude poolt:** Võimaliku läbiviija lisamine.

**OP5.7 Võimaliku läbiviija muutmine (viide treeningu võimalikule läbiviimisele, viide töötajale, alguse\_aeg, lõpuaega)**

**Eeltingimused:**

- Treening ja töötaja on registreeritud. Viide treeningu ja võimalik\_läbiviija vahel on registreeritud.

**Järeltingimused:**

**--Seoste registreerimine**

Võimalik\_läbiviija.alguse\_aeg = alguse\_aeg

Võimalik\_läbiviija.lõpu\_aeg = lõpu\_aeg

Võimalik läbiviija on seotud uue Töötajaga

**Kasutus kasutusjuhtude poolt:** Võimaliku läbiviija muutmine.

### Registri põhiobjekti seisundidiagramm

Registri põhiobjekt on Treening.

****

**Joonis 5. Treeningu seisundidiagramm.**

## CRUD maatriks

| Kasutusjuhud  Olemitüübid | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Kokku |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Isik | R |  |  |  |  |  | R | R |  |  | R |
| Töötaja | R |  |  |  |  | R | R | R |  |  | R |
| Klient | R |  |  |  |  |  | R |  |  |  | R |
| Klassifikaator | R | R |  |  | R | R | R | R | R | R | R |
| Treeningu\_seisundi\_liik | R |  |  |  | R | R | R | R | R | R | R |
| Ametikoht | R |  |  |  |  |  |  |  |  |  | R |
| Riik | R |  |  |  |  |  |  |  |  |  | R |
| Kliendi\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  | R |
| Töötaja\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  | R |
| Spordiala | R | R |  |  |  |  | R | R |  |  | R |
| Treening |  | CR | RD | RU | RU | RU | R | R | R | R | CRUD |
| Võimalik\_läbiviija |  | R | D |  |  |  |  | R | C | RU | CRUD |

1. Kasutaja tuvastamine
2. Treeningute registreerimine
3. Treeningute kustutamine
4. Treeningute muutmine
5. Treeningute aktiivseks muutmine
6. Treeningute mitteaktiivseks muutmine
7. Treeningute vaatamine
8. Aktiivsete treeningute vaatamine
9. Võimaliku läbiviija lisamine
10. Võimaliku läbiviija muutmine

## Infosüsteemi rollide kirjeldused

| **Rolli nimi** | **Kirjeldus** |
| --- | --- |
| Klient | Klient on spordiklubi külastav isik. Kliendil on õigus vaadata spordiklubis toimuvaid treeninguid ning nendele registreeruda. Lisaks on õigus kliendil vaadata treeninguid kuhu tema on registreerunud ning neid ka tühistada. Kliendil pole õigust luua, muuta, kustutada treeninguid ning pole õigust vaadata teiste klientide registreerimisi ja neid kustutada. |
| Administraator | Administraator on spordiklubi töötaja. Adminsitraatoril on õigus registreerida uusi treeninguid, neid vaadata, muuta, kustutada, aktiivseks- ja mitteaktiivseks muuta. Administraatoril pole õigusi registreerida kliente treeningutele ning klientide registreeringuid kustutada. |
| Treener | Treener on spordiklubi töötaja, kes viib läbi treeninguid. Treeneril on õigus vaadata treeninguid ning muuta ja mitteaktiivseks muuta enda treeninguid. Treeneril pole õigusi registreerida, kustutada ja aktiivseks muuta treeninguid. Lisaks ei saa treener muuta ja mitteaktiivseks teha teiste treenerite treeninguid. Veel pole treeneril õigusi registreerida kliente treeningutele ning klientide registreeringuid kustutada ega vaadata. |
| Juhataja | Juhataja on spordiklubi töötaja. Juhatajal on õigus kõiki treeninguid vaadata ning mitteaktiivseks muuta. Juhatajal pole õigusi teha uusi treeninguid, neid muuta, kustutada või aktiivseks muuta. Lisaks pole juhatajal õigusi registreerida kliente treeningutele ning klientide registreeringuid kustutada ega vaadata. |
| Andmebaasi administraator | Spordiklubi andmebaasi administraator on töötaja, kelle ülesandeks on tagada andmebaasi töö, viies läbi andmebaasi hooldust, jälgimist ja häälestamist. Tal on õigus vaadata kõiki andmeid, kuid tal pole õigust andmeid lisada, muuta ega kustutada. |

# 

# Füüsiline disain

## 3.1 Treeningute funktsionaalse allsüsteemi kasutatavate registrite füüsiline disain

### Füüsiline andmebaasiskeem







Klassifikaatorite registri füüsilise disaini andmebaasi diagramm (UML notatsioonis).



Klientide registri füüsilise disaini andmebaasi diagramm (UML notatsioonis).



Osapoolte registri füüsilise disaini andmebaasi diagramm (UML notatsioonis).



Töötajate registri füüsilise disaini andmebaasi diagramm (UML notatsioonis).



Treeningute registri füüsilise disaini andmebaasi diagramm (UML notatsioonis).

# Realisatsioon PostgreSQLis

„Andmebaasi loomiseks kasutatakse andmebaasisüsteemi **PostgreSQL (9.4.4)**.“ [Eessaar, 2015]

## Andmebaasi loomine

Andmebaasis *t134683* kasutatav märkide klassifikatsioon ja märkide

võrdlusreeglistik vastavad eesti keele reeglitele.

createdb -l et\_EE.utf8 -T template0

„Järgnevalt esitatakse SQL laused selle andmebaasiobjektide loomiseks.“ [Eessaar, 2015]

## Skeemid

„Loon skeemiobjektid skeemis nimega *public*, mis loodi andmebaasi loomisel automaatselt.“[Eessaar, 2015]

## Domeenid

CREATE DOMAIN d\_nimetus AS VARCHAR(70) NOT NULL

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

CONSTRAINT chk\_nimetus\_ei\_ole\_tuhi\_string CHECK(VALUE!='');

CREATE DOMAIN d\_kirjeldus AS VARCHAR(255)

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

CONSTRAINT chk\_kirjeldus\_ei\_ole\_tuhi\_string CHECK(VALUE!='');

## Tabelid ja arvujada generaatorid

CREATE TABLE Treeningu\_seisundi\_liik (

treeningu\_seisundi\_liik\_kood CHAR(5) NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

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

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

)WITH (fillfactor=80);

CREATE TABLE Spordiala (

spordiala\_kood CHAR(5) NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

CONSTRAINT AK\_Spordiala\_nimetus UNIQUE (nimetus),

CONSTRAINT PK\_Spordiala\_kood PRIMARY KEY (spordiala\_kood)

)WITH (fillfactor=80);

CREATE TABLE Tootaja\_seisundi\_liik (

tootaja\_seisundi\_liik\_kood CHAR(5) NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

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

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

)WITH (fillfactor=80);

CREATE TABLE Klient\_seisundi\_liik (

kliendi\_seisundi\_liik\_kood CHAR(5) NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

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

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

)WITH (fillfactor=80);

CREATE TABLE Amet (

amet\_kood CHAR(5) NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

CONSTRAINT AK\_Amet\_nimetus UNIQUE (nimetus),

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

)WITH (fillfactor=80);

CREATE TABLE Tootaja (

tootaja\_id SMALLSERIAL NOT NULL,

isikukood CHAR ( 11 ) NOT NULL,

amet\_kood CHAR(5) NOT NULL,

tootaja\_seisundi\_liik\_kood CHAR(5) NOT NULL,

CONSTRAINT AK\_Tootaja\_isikukood UNIQUE (isikukood),

CONSTRAINT PK\_Tootaja\_kood PRIMARY KEY (tootaja\_id));

CREATE TABLE Treening (

treening\_id SMALLSERIAL NOT NULL,

spordiala\_kood CHAR(5) NOT NULL,

treeningu\_seisundi\_liik\_kood CHAR(5) NOT NULL,

tootaja\_id SMALLINT NOT NULL,

nimetus d\_nimetus,

kirjeldus d\_kirjeldus,

CONSTRAINT AK\_Treening\_nimetus UNIQUE (nimetus),

CONSTRAINT PK\_Treening\_id PRIMARY KEY (treening\_id)

)WITH (fillfactor=70);

CREATE TABLE Klient (

klient\_id SERIAL NOT NULL,

kliendi\_seisundi\_liik\_kood CHAR(5) NOT NULL,

isikukood CHAR ( 11 ) NOT NULL,

CONSTRAINT PK\_Klient\_kood PRIMARY KEY (klient\_id),

CONSTRAINT AK\_Klient\_isikukood UNIQUE (isikukood));

CREATE TABLE Voimalik\_labiviija (

voimalik\_labiviija\_id SERIAL NOT NULL,

treening\_id SMALLINT NOT NULL,

tootaja\_id SMALLINT NOT NULL,

alguse\_aeg DATE NOT NULL,

lopu\_aeg DATE NOT NULL,

CONSTRAINT AK\_Voimalik\_labiviija\_treening\_id\_tootaja\_id\_alguse\_aeg UNIQUE (alguse\_aeg, tootaja\_id, treening\_id),

CONSTRAINT PK\_Voimalik\_labiviija\_id PRIMARY KEY (voimalik\_labiviija\_id),

CONSTRAINT chk\_Voimalik\_labiviija\_alguse\_aeg\_vahemikus\_2000\_2100 CHECK (alguse\_aeg BETWEEN '2000-01-01' AND '2100-01-01'),

CONSTRAINT chk\_Voimalik\_labiviija\_lopu\_aeg\_vahemikus\_2000\_2100 CHECK (lopu\_aeg BETWEEN '2000-01-01' AND '2100-01-01'),

CONSTRAINT chk\_Voimalik\_labiviija\_lopu\_aeg\_suurem\_kui\_alguse\_aeg CHECK (lopu\_aeg>=alguse\_aeg));

CREATE TABLE Isik (

isikukood CHAR ( 11 ) NOT NULL,

kasutajanimi VARCHAR ( 10 ) NOT NULL,

email VARCHAR ( 70 ) NOT NULL,

eesnimi VARCHAR ( 60 ) NOT NULL,

perenimi VARCHAR ( 60 ) NOT NULL,

parool VARCHAR ( 60 ) NOT NULL,

registreerimise\_aeg TIMESTAMP DEFAULT LOCALTIMESTAMP(0) NOT NULL,

elukoht VARCHAR ( 255 ),

CONSTRAINT PK\_Isik\_isikukood PRIMARY KEY (isikukood),

CONSTRAINT AK\_Isik\_kasutajanimi UNIQUE (kasutajanimi),

CONSTRAINT AK\_Isik\_email UNIQUE (email),

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

CONSTRAINT chk\_Isik\_kasutajanimi\_ei\_tohi\_sisaldada\_tuhikuid CHECK (kasutajanimi!~'^.\*[[:space:]].\*$'),

CONSTRAINT chk\_Isik\_eesnimi\_ainult\_esimene\_taht\_igal\_eesnimel\_suur CHECK (eesnimi~'^([[:upper:]]{1}[[:lower:]]+)(([[:space:]]|-)[[:upper:]]{1}[[:lower:]]+)\*$'),

CONSTRAINT chk\_Isik\_registreerimise\_aeg\_vahemikus\_2000\_2100 CHECK (registreerimise\_aeg BETWEEN '2000-01-01' AND '2100-01-01'),

CONSTRAINT chk\_Isik\_isikukood\_koik\_reeglid CHECK (isikukood~'^([3-6]{1}[[:digit:]]{2}[0-1]{1}[[:digit:]]{1}[0-3]{1}[[:digit:]]{5})$'),

CONSTRAINT chk\_Isik\_kasutajanimi\_ei\_tohi\_olla\_luhem\_kui\_4\_marki CHECK (char\_length(kasutajanimi)>3),

CONSTRAINT chk\_Isik\_perenimi\_ainult\_esimene\_taht\_igal\_perenimel\_suur CHECK (perenimi~'^([[:upper:]]{1}[[:lower:]]+)(([[:space:]]|-)[[:upper:]]{1}[[:lower:]]+)\*$'),

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

CONSTRAINT chk\_Isik\_email\_peab\_sisaldama\_at\_marki CHECK (email LIKE '%@%'),

CONSTRAINT chk\_Isik\_elukoht\_ei\_ole\_tyhi\_string CHECK (elukoht <> ''))WITH (fillfactor=70);

ALTER TABLE Voimalik\_labiviija ADD CONSTRAINT FK\_Tootaja\_tootaja\_id FOREIGN KEY (tootaja\_id) REFERENCES Tootaja (tootaja\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE Voimalik\_labiviija ADD CONSTRAINT FK\_Treening\_treening\_id FOREIGN KEY (treening\_id) REFERENCES Treening (treening\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE Treening ADD CONSTRAINT FK\_Treeningu\_seisundi\_liik\_treeningu\_seisundi\_liik\_kood FOREIGN KEY (treeningu\_seisundi\_liik\_kood) REFERENCES Treeningu\_seisundi\_liik (treeningu\_seisundi\_liik\_kood) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE Treening ADD CONSTRAINT FK\_Spordiala\_spordiala\_kood FOREIGN KEY (spordiala\_kood) REFERENCES Spordiala (spordiala\_kood) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE Treening ADD CONSTRAINT FK\_Tootaja\_tootaja\_id FOREIGN KEY (tootaja\_id) REFERENCES Tootaja (tootaja\_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE Klient ADD CONSTRAINT FK\_Klient\_seisundi\_liik\_kliendi\_seisundi\_liik\_kood FOREIGN KEY (kliendi\_seisundi\_liik\_kood) REFERENCES Klient\_seisundi\_liik (kliendi\_seisundi\_liik\_kood) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE Klient ADD CONSTRAINT FK\_Isik\_isikukood FOREIGN KEY (isikukood) REFERENCES Isik (isikukood) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE Tootaja ADD CONSTRAINT FK\_Tootaja\_seisundi\_liik\_tootaja\_seisundi\_liik\_kood FOREIGN KEY (tootaja\_seisundi\_liik\_kood) REFERENCES Tootaja\_seisundi\_liik (tootaja\_seisundi\_liik\_kood) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE Tootaja ADD CONSTRAINT FK\_Amet\_amet\_kood FOREIGN KEY (amet\_kood) REFERENCES Amet (amet\_kood) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE Tootaja ADD CONSTRAINT FK\_Isik\_isikukood FOREIGN KEY (isikukood) REFERENCES Isik (isikukood) ON DELETE CASCADE ON UPDATE CASCADE;

## Vaated

CREATE OR REPLACE VIEW koik\_kliendid\_seisundiga WITH (security\_barrier) AS SELECT

Klient.isikukood AS klient\_isikukood,

Isik.eesnimi AS klient\_eesnimi,

Isik.perenimi AS klient\_perenimi,

Klient\_seisundi\_liik.nimetus AS klient\_seisund

FROM

(Klient INNER JOIN Klient\_seisundi\_liik ON Klient.kliendi\_seisundi\_liik\_kood = Klient\_seisundi\_liik.kliendi\_seisundi\_liik\_kood) INNER JOIN Isik ON Klient.isikukood = Isik.isikukood

ORDER BY Isik.perenimi, Isik.eesnimi;

COMMENT ON VIEW koik\_kliendid\_seisundiga IS 'Vaade naitab koigi susteemis registreeritud klientide isikukoode, ees-, perekonnanimesid ja hetkeseisundeid.';

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

AS SELECT

Treening.nimetus AS treening\_nimetus,

Treening.kirjeldus AS treening\_kirjeldus,

Isik.eesnimi || ' ' || Isik.perenimi AS registreerija\_nimi,

Isik.isikukood AS registreerija\_isikukood,

Treeningu\_seisundi\_liik.nimetus AS treening\_seisund

FROM Treeningu\_seisundi\_liik INNER JOIN ((Isik INNER JOIN Tootaja ON Isik.isikukood = Tootaja.isikukood) INNER JOIN Treening ON Tootaja.tootaja\_id = Treening.tootaja\_id) ON Treeningu\_seisundi\_liik.treeningu\_seisundi\_liik\_kood = Treening.treeningu\_seisundi\_liik\_kood

ORDER BY Treening.nimetus;

COMMENT ON VIEW koik\_treeningud IS 'Vaade annab ulevaate koigist spordiklubis pakutavatest treeningutest

ning naitab iga treeningu kohta selle nimetust, kirjeldust, treeningut registreerunud tootaja nime ja isikukoodi ning treeningu hetkeseisundi.';

CREATE OR REPLACE VIEW treeningute\_labiviijad WITH (security\_barrier)

AS SELECT

treening.nimetus AS treening\_nimetus,

voimalik\_labiviija.alguse\_aeg AS perioodi\_algus,

voimalik\_labiviija.lopu\_aeg AS perioodi\_lopp,

isik.eesnimi || ' ' || isik.perenimi AS labiviija\_nimi,

tootaja.isikukood AS labiviija\_isikukood

FROM

public.voimalik\_labiviija,

public.treening,

public.tootaja,

public.isik

WHERE

voimalik\_labiviija.tootaja\_id = tootaja.tootaja\_id AND

treening.treening\_id = voimalik\_labiviija.treening\_id AND

tootaja.isikukood = isik.isikukood

ORDER BY treening.nimetus, voimalik\_labiviija.alguse\_aeg;

COMMENT ON VIEW treeningute\_labiviijad IS 'Vaade naitab, kes tegeleb konkreetse treeningu labiviimisega konkreetsel ajaperioodil.';

CREATE OR REPLACE VIEW koik\_tootajad\_seisundiga WITH (security\_barrier)

AS SELECT

tootaja.isikukood AS tootaja\_isikukood,

isik.eesnimi AS tootaja\_eesnimi,

isik.perenimi AS tootaja\_perenimi,

isik.email AS tootaja\_epost,

amet.nimetus AS tootaja\_amet,

tootaja\_seisundi\_liik.nimetus AS tootaja\_seisund

FROM

public.tootaja,

public.isik,

public.tootaja\_seisundi\_liik,

public.amet

WHERE

isik.isikukood = tootaja.isikukood AND

tootaja.amet\_kood = amet.amet\_kood AND

tootaja.tootaja\_seisundi\_liik\_kood = tootaja\_seisundi\_liik.tootaja\_seisundi\_liik\_kood

ORDER BY isik.perenimi, isik.eesnimi;

COMMENT ON VIEW koik\_tootajad\_seisundiga IS 'Vaade naitab koigi susteemis registreeritud tootajate isikukoode, ees- ja perekonnanimesid, e-posti aadresseid, ameteid ja hetkeseisundeid.';

## Protseduursed keeled

“Käesolevas projektis kasutatakse funktsioonide loomiseks PL/pgSQL keelt. PostgreSQL 9.4.4 andmebaasisüsteemis installeeritakse PL/pgSQL keel vaikimisi kõikidesse andmebaasidesse.“ [Eessaar, 2015]

## Trigeri funktsioonid ja trigerid

CREATE OR REPLACE FUNCTION f\_trigger\_treeningu\_kustutamine() RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION 'Uhtegi juba registreeritud treeningut ei tohi andmebaasist ara kustutada.';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_trigger\_treeningu\_kustutamine() IS 'See trigeri funktsioon aitab joustada arireegli: uhtegi juba registreeritud treeningut ei tohi andmebaasist ara kustutada.';

CREATE OR REPLACE FUNCTION f\_trigger\_treeningu\_muutmine\_treeningu\_seisund() RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION 'Andmeid ei saa muuta treeningutel, mille seisund on "Loplikult peatatud"!';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_trigger\_treeningu\_muutmine\_treeningu\_seisund() IS 'See trigeri funktsioon aitab joustada arireegli: Andmeid ei saa muuta treeningutel, mille seisund on "Loplikult peatatud"';

CREATE OR REPLACE FUNCTION f\_trigger\_uus\_treening\_tootaja\_seisund() RETURNS trigger AS $$

DECLARE

m\_tootaja\_ID SMALLINT;

BEGIN

SELECT tootaja\_id INTO m\_tootaja\_ID FROM Tootaja WHERE tootaja\_seisundi\_liik\_kood <> 'AKTII' AND tootaja\_id=new.tootaja\_id FOR UPDATE;

IF (new.tootaja\_id = m\_tootaja\_ID) THEN

RAISE EXCEPTION 'Uusi treeninguid saab registreerida ainult aktiivses seisundis tootaja!';

ELSE

RETURN NEW;

END IF;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_trigger\_uus\_treening\_tootaja\_seisund() IS 'See trigeri funktsioon aitab joustada arireegli: uusi treeninguid saab registreerida ainult aktiivses seisundis tootaja!';

CREATE OR REPLACE FUNCTION f\_trigger\_uus\_treening\_amet() RETURNS trigger AS $$

DECLARE

m\_tootaja\_ID SMALLINT;

BEGIN

SELECT tootaja\_id INTO m\_tootaja\_ID FROM Tootaja WHERE amet\_kood <> 'ADMIN' AND tootaja\_id=new.tootaja\_id FOR UPDATE;

IF (new.tootaja\_id = m\_tootaja\_ID) THEN

RAISE EXCEPTION 'Ainult administraator saab registreerida uusi treeninguid!';

ELSE

RETURN NEW;

END IF;

END;

$$ LANGUAGE plpgsql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_trigger\_uus\_treening\_amet() IS 'See trigeri funktsioon aitab joustada arireegli: ainult administraator saab registreerida uusi treeninguid.';

CREATE TRIGGER trigger\_treeningu\_kustutamine BEFORE DELETE OR TRUNCATE ON

Treening FOR EACH STATEMENT EXECUTE PROCEDURE f\_trigger\_treeningu\_kustutamine();

CREATE TRIGGER trigger\_treeningu\_muutmine\_treeningu\_seisund BEFORE UPDATE ON

Treening FOR EACH ROW WHEN (OLD.treeningu\_seisundi\_liik\_kood = 'LOPLI') EXECUTE PROCEDURE

f\_trigger\_treeningu\_muutmine\_treeningu\_seisund();

CREATE TRIGGER trigger\_uus\_treening\_tootaja\_seisund BEFORE INSERT ON

Treening FOR EACH ROW EXECUTE PROCEDURE

f\_trigger\_uus\_treening\_tootaja\_seisund();

CREATE TRIGGER trigger\_uus\_treening\_amet BEFORE INSERT ON

Treening FOR EACH ROW EXECUTE PROCEDURE

f\_trigger\_uus\_treening\_amet();

## Reeglid

„Reegleid käesolevas projektis ei looda.“ [Eessaar, 2015]

## Funktsioonid

CREATE OR REPLACE FUNCTION f\_treeningu\_registreerimine

(Tootaja.tootaja\_id%TYPE, Spordiala.spordiala\_kood%TYPE, Treening.nimetus%TYPE, Treening.kirjeldus%TYPE)

RETURNS Treening.treening\_id%TYPE AS

$$ INSERT INTO Treening( tootaja\_id, spordiala\_kood, nimetus, kirjeldus)

VALUES ($1, $2, $3, $4)

RETURNING treening\_id;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_treeningu\_registreerimine

(Tootaja.tootaja\_id%TYPE, Spordiala.spordiala\_kood%TYPE, Treening.nimetus%TYPE, Treening.kirjeldus%TYPE)

IS 'Selle funktsiooni abil lisatakse uus treening. Funktsioon realiseerib andmebaasioperatsiooni OP5.1.

Funktsiooni esimene argument votab vastu registreeriva tootaja id, teine

argument spordiala koodi, kolmas argument treeningu nimetuse ja viimane neljas argument treeningu kirjelduse.';

CREATE OR REPLACE FUNCTION f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE)

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_seisundi\_liik\_kood = 'AJUTI'

WHERE nimetus = $1 AND treeningu\_seisundi\_liik\_kood = 'AKTII';

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE)

IS 'Selle funktsiooni abil muudetakse treening ajutiselt peatatuks. Funktsioon realiseerib andmebaasioperatsiooni OP5.2.

Funktsiooni argument votab vastu treeningu id.';

CREATE OR REPLACE FUNCTION f\_treeningu\_aktiivseks\_muutmine(Treening.treening\_id%TYPE)

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_seisundi\_liik\_kood = 'AKTII'

WHERE treening\_id = $1 AND treeningu\_seisundi\_liik\_kood = 'AJUTI';

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_treeningu\_aktiivseks\_muutmine(Treening.treening\_id%TYPE)

IS 'Selle funktsiooni abil muudetakse treening aktiivseks. Funktsioon realiseerib andmebaasioperatsiooni OP5.3.

Funktsiooni argument votab vastu treeningu id.';

CREATE OR REPLACE FUNCTION f\_treeningu\_kustutamine (Treening.treening\_id%TYPE)

RETURNS VOID AS $$

UPDATE Treening SET treeningu\_seisundi\_liik\_kood='LOPLI'

WHERE treening\_id = $1 AND treeningu\_seisundi\_liik\_kood<>'LOPLI';

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_treeningu\_kustutamine (Treening.treening\_id%TYPE) IS 'Selle

funktsiooni abil tuhistatakse treening. Funktsioon realiseerib andmebaasioperatsioon

OP5.4. Funktsiooni valjakutsel on argumendiks tuhistatava treeningu identifikaator.';

CREATE OR REPLACE FUNCTION f\_uue\_labiviija\_lisamine (Treening.treening\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE)

RETURNS VOID AS $$

INSERT INTO Voimalik\_labiviija(treening\_id, tootaja\_id, alguse\_aeg, lopu\_aeg)

VALUES ($1, $2, $3, $4);

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_uue\_labiviija\_lisamine (Treening.treening\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE) IS

'Selle funktsiooni abil lisatakse uus labiviija. See funktsioon realiseerib

andmebaasioperatsiooni OP5.6. Funktsiooni valjakutses on esimene argument treeningu id, mida uus labiviija hakkab andma ,

teine argument - tootaja, kes treeningut andma hakab, kolmas - antud tootaja poolt antud treeningu labiviimise perioodi esimene kuupaev ning neljas - antud tootaja poolt antud treeningu labiviimise perioodi viimane kuupaev.';

CREATE OR REPLACE FUNCTION f\_labiviija\_muutmine (Voimalik\_labiviija.voimalik\_labiviija\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE)

RETURNS VOID AS $$

UPDATE Voimalik\_labiviija SET tootaja\_id = $2, alguse\_aeg = $3, lopu\_aeg = $4

WHERE voimalik\_labiviija\_id = $1;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_labiviija\_muutmine (Voimalik\_labiviija.voimalik\_labiviija\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE) IS 'Selle

funktsiooni abil muudetakse treeningu labiviijat konkreetsel perioodil. Funktsioon realiseerib andmebaasioperatsiooni

OP5.7. Esimene argument: tabeli Voimalik\_labiviija rea identifikaator tabelis, kus asuvad muudetavad andmed, teine argument - tootaja id, kes on uueks treeningu labiviijaks, kolmas argument - treeningu labiviimise perioodi esimene kuupaev, neljas argument - treeningu labiviimise perioodi viimane kuupaev.';

CREATE OR REPLACE FUNCTION f\_sisselogimine\_juhataja(text, text)

RETURNS boolean AS $$

DECLARE

rslt boolean;

BEGIN

select INTO rslt (parool = public.crypt($2, parool)) from isik inner join tootaja ON isik.isikukood = tootaja.isikukood where tootaja.amet\_kood='JUHAT' and NOT tootaja\_seisundi\_liik\_kood='LOPET';

RETURN coalesce(rslt, FALSE);

END;

$$ LANGUAGE plpgsql SECURITY DEFINER STABLE

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION

f\_sisselogimine\_juhataja(text, text)

IS 'Selle funktsiooni abil autenditakse juhatajat. Funktsiooni valjakutsel on esimene argument kasutajanimi ja teine argument parool.

Juhatajal on oigus susteemi siseneda, vaid siis kui tema seisundiks pole "LOPET"';

## Indeksid

### Välisvõtmetele lisatavad indeksid

CREATE INDEX idx\_tootaja\_tootaja\_seisundi\_liik\_kood ON Tootaja (tootaja\_seisundi\_liik\_kood );

CREATE INDEX idx\_tootaja\_amet\_kood ON Tootaja (amet\_kood );

CREATE INDEX idx\_treening\_tootaja\_id ON Treening (tootaja\_id );

CREATE INDEX idx\_treening\_treeningu\_seisundi\_liik\_kood ON Treening (treeningu\_seisundi\_liik\_kood);

CREATE INDEX idx\_treening\_spordiala\_kood ON Treening (spordiala\_kood );

CREATE INDEX idx\_klient\_kliendi\_seisundi\_liik\_kood ON Klient (kliendi\_seisundi\_liik\_kood );

CREATE INDEX idx\_voimalik\_labiviija\_tootaja\_id ON Voimalik\_labiviija (tootaja\_id);

CREATE INDEX idx\_voimalik\_labiviija\_treening\_id ON Voimalik\_labiviija (treening\_id);

### Täiendavad sekundaarsed indeksid

CREATE INDEX idx\_isik\_perenimi ON Isik (perenimi );

CREATE INDEX idx\_isik\_eesnimi ON Isik (eesnimi );

### Funktsioonil põhinevad indeksid

„Funktsioonil põhinevaid indekseid käesolevas projektis ei looda.“ [Eessaar, 2015]

## Klassifikaatorite väärtustamise SQL laused

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES ('JUHAT', 'Juhataja', 'See amet korraldab kogu spordiklubi tood. Samuti otsib juhataja uusi kliente, uusi treenereid ja korraldab reklaami.');

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES ('TREEN', 'Treener', 'See amet viib treeninguid labi ja postitab veebisaitidele uudiseid treeningutega seoses.');

INSERT INTO amet(amet\_kood, nimetus, kirjeldus) VALUES ('ADMIN', 'Administraator', 'See amet vastab telefonile, aitab klientidel registreerida treeningutele, kontrollib et spordiklubisse sisenevad ainult kliendid ja muub spordijooke.');

INSERT INTO klient\_seisundi\_liik(kliendi\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('AKTII', 'Aktiivne', 'Seda liiki klient kaib regulaarselt trennis ja maksab kuumaksu.');

INSERT INTO klient\_seisundi\_liik(kliendi\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('PUHKU', 'Puhkusel', 'Seda liiki klient on treeningpuhkusel, selle jaoks on loa kinnitanud juhataja.');

INSERT INTO klient\_seisundi\_liik(kliendi\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('LOPET', 'Lopetatud', 'Seda liiki klient on varasemalt olnud spordiklubi klient ja enam ei ole.');

INSERT INTO spordiala(spordiala\_kood, nimetus, kirjeldus) VALUES ('AJOOG', 'Jooga algajatele', 'Jooga trenn algajatele.');

INSERT INTO spordiala(spordiala\_kood, nimetus, kirjeldus) VALUES ('EJOOG', 'Jooga edasijoudnutele', 'Jooga trenn edasijoudnutele.');

INSERT INTO spordiala(spordiala\_kood, nimetus, kirjeldus) VALUES ('BODYP', 'Bodypump', 'Bodypump trenn mis jargib rahvusvaheliselt regulaarselt muudetavat kava.');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('AKTII', 'Aktiivselt tool kaiv tootaja', 'Tootaja kes kaib regulaarselt tool.');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('PUHKU', 'Puhkusel olev tootaja', 'Tootaja kes on puhkusel.');

INSERT INTO tootaja\_seisundi\_liik(tootaja\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('LOPET', 'Lopetatud', 'See tootaja on varasemalt olnud spordiklubi tootaja ja enam ei ole.');

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('AKTII', 'Aktiivne', 'Treening toimub regulaarselt.');

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('LOPLI', 'Loplikult peatatud', 'Treening on juhatuse soovil loplikult peatatud.');

INSERT INTO treeningu\_seisundi\_liik(treeningu\_seisundi\_liik\_kood, nimetus, kirjeldus) VALUES ('AJUTI', 'Ajutiselt peatatud', 'Treening on juhatuse soovil ajutisel peatatud.');

## Täiendavate testandmete lisamine

INSERT INTO isik(isikukood, kasutajanimi, email, eesnimi, perenimi, parool, registreerimise\_aeg, elukoht) VALUES ('31212121212', 'Sass123', 'sander@sander.ee', 'Sander', 'Saarsen', 'Parool123', '2015-10-28 11:22:07', 'Akadeemia tee 99-99, 61616, Tallinn');

INSERT INTO isik(isikukood, kasutajanimi, email, eesnimi, perenimi, parool, registreerimise\_aeg, elukoht) VALUES ('49012122711', 'Maasikas8', 'mari.maasikas@mail.ee', 'Mari', 'Maasikas', 'Maasu156', '2015-10-28 11:31:23', 'Tamme 15, 61617, Viimsi');

INSERT INTO isik(isikukood, kasutajanimi, email, eesnimi, perenimi, parool, registreerimise\_aeg, elukoht) VALUES ('38512252222', 'bonbon', 'jtamm@gmail.com', 'Juhan', 'Tamm', 'kovapahkel', '2015-10-29 14:47:09', 'Pihlaka tn 3-4,Tallinn');

INSERT INTO isik(isikukood, kasutajanimi, email, eesnimi, perenimi, parool, registreerimise\_aeg, elukoht) VALUES ('60206021111', 'torm007', 'mailiis@mail.ee', 'Anna-Mai-Liis', 'Kask-Toomingas', 'vagaturvaline123', '2015-10-29 14:51:20', 'Kibuvitsa tn 13-14, Tallinn');

INSERT INTO isik(isikukood, kasutajanimi, email, eesnimi, perenimi, parool, registreerimise\_aeg, elukoht) VALUES ('46512122711', 'juhataja1', 'boss@treening.ee', 'Uku', 'Labidas', 'parool1', '2015-11-04 10:32:36', 'Mustamae tee 36-63, 61265, Tallinn');

INSERT INTO klient(klient\_id, kliendi\_seisundi\_liik\_kood, isikukood) VALUES (1, 'AKTII', '49012122711');

INSERT INTO klient(klient\_id, kliendi\_seisundi\_liik\_kood, isikukood) VALUES (2, 'LOPET', '60206021111');

INSERT INTO klient(klient\_id, kliendi\_seisundi\_liik\_kood, isikukood) VALUES (4, 'PUHKU', '38512252222');

INSERT INTO tootaja(tootaja\_id, isikukood,amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (1, '31212121212', 'ADMIN', 'AKTII');

INSERT INTO tootaja(tootaja\_id, isikukood,amet\_kood,

tootaja\_seisundi\_liik\_kood) VALUES (2, '38512252222', 'TREEN', 'PUHKU');

INSERT INTO tootaja(tootaja\_id, isikukood,amet\_kood, tootaja\_seisundi\_liik\_kood) VALUES (3, '46512122711', 'JUHAT', 'AKTII');

INSERT INTO treening(treening\_id, spordiala\_kood, treeningu\_seisundi\_liik\_kood, tootaja\_id, nimetus,kirjeldus) VALUES (1, 'AJOOG', 'AKTII', 1, 'Jooga algajatele', 'Tulge trenni');

INSERT INTO treening(treening\_id, spordiala\_kood, treeningu\_seisundi\_liik\_kood, tootaja\_id, nimetus, kirjeldus) VALUES (2, 'BODYP', 'AJUTI', 1, 'Extremely effective bodypump', 'erakordselt kovade inimeste jaoks');

INSERT INTO treening(treening\_id, spordiala\_kood, treeningu\_seisundi\_liik\_kood, tootaja\_id, nimetus, kirjeldus) VALUES (5, 'EJOOG', 'AKTII', 1, 'Jooga edasijoudnutele', 'veenita oma vasinud lihaseid');

INSERT INTO voimalik\_labiviija(voimalik\_labiviija\_id, treening\_id, tootaja\_id, alguse\_aeg, lopu\_aeg) VALUES (1, 1, 1, '2016-01-01', '2016-12-31');

INSERT INTO voimalik\_labiviija (voimalik\_labiviija\_id, treening\_id, tootaja\_id, alguse\_aeg, lopu\_aeg) VALUES (2, 2, 2, '2015-06-01', '2015-12-31');

INSERT INTO voimalik\_labiviija (voimalik\_labiviija\_id, treening\_id, tootaja\_id, alguse\_aeg, lopu\_aeg) VALUES (4, 5, 2, '2015-08-15', '2016-02-15');

## Andmebaasi statistika kogumine

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 lause. ANALYZE määrang lauses tagab, et lisaks prügikoristusele värskendatakse ka andmebaasi statistikat. Tabeliga saab siduda ka korralduse käivitada prügikoristus ning statistika kogumine automaatselt (autovacuum), kui tabelisse lisatud, muudetud või kustutatud ridade arv ületab ettemääratud lävendi. Käesolevas projektis seda võimalust ei kasutata. Käesolevas jaotises toodud lauset tuleks käivitada vähemalt kord päevas, soovitavalt siis, kui andmebaasi kasutamise koormus on madal (näiteks öösel)." [Eessaar, 2015]

## Päringu täitmisplaani näide

Analüüsin päringut, mis on tehtud vaate koik\_kliendid\_seisundiga põhjal.

CREATE OR REPLACE VIEW koik\_kliendid\_seisundiga WITH (security\_barrier) AS SELECT

Klient.isikukood AS klient\_isikukood,

Isik.eesnimi AS klient\_eesnimi,

Isik.perenimi AS klient\_perenimi,

Klient\_seisundi\_liik.nimetus AS klient\_seisund

FROM

(Klient INNER JOIN Klient\_seisundi\_liik ON Klient.kliendi\_seisundi\_liik\_kood = Klient\_seisundi\_liik.kliendi\_seisundi\_liik\_kood) INNER JOIN Isik ON Klient.isikukood = Isik.isikukood

ORDER BY Isik.perenimi, Isik.eesnimi

ANALYZE;

EXPLAIN SELECT \* FROM koik\_kliendid\_seisundiga;

**QUERY PLAN**

**------------------------------------------------------------------**

1. Sort (cost=3.39..3.40 rows=4 width=35)

2. Sort Key: isik.perenimi, isik.eesnimi

3. -> Hash Join (cost=2.16..3.35 rows=4 width=35)

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

5. -> Hash Join (cost=1.09..2.23 rows=4 width=32)

6. Hash Cond: (isik.isikukood = klient.isikukood)

7. -> Seq Scan on isik (cost=0.00..1.07 rows=7 width=26)

8. -> Hash (cost=1.04..1.04 rows=4 width=18)

9. -> Seq Scan on klient (cost=0.00..1.04 rows=4 width=18)

10. -> Hash (cost=1.03..1.03 rows=3 width=15)

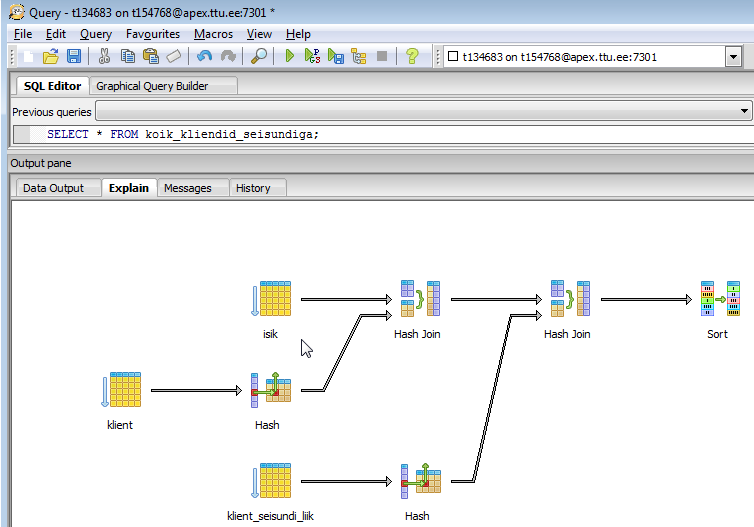
11. -> Seq Scan on klient\_seisundi\_liik (cost=0.00..1.03 rows=3 width=15

"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“ [Eessaar, 2015]

5-9 – Tabel Klient ühendatakse tabeliga Isik kasutades hash join algoritmi. Tabel Klient, kui oodatavalt väiksem tabel, valiti sisemiseks tabeliks. Tabelist Klient loetud veeru isikukood väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema tabelist Isik loetud väärtustega. Selle ühendamise kontekstis on Isik väline tabel. Nii veeru isikukood väärtused tabelist Klient kui ka veeru isikukood väärtused tabelist Isik leitakse tabelite täielikku läbiskaneerimist kasutades.

3-11 - Klient ja Isik ühendamise tulemusena leitud tabel ühendatakse tabeliga Klient\_seisundi\_liik kasutades hash join algoritmi. Tabel Klient\_seisundi\_liik, kui oodatavalt väiksem tabel, valiti sisemiseks tabeliks. Tabelist Klient\_seisundi\_liik loetud klient\_seisundi\_liik\_kood väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema ühendamise tulemusena leitud tabelist loetud väärtustega. Selle ühendamise kontekstis on ühendamise tulemusena leitud tabel väline tabel. klient\_seisundi\_liik\_kood väärtused tabelist Klient\_seisundi\_liik leitakse tabelite täielikku läbiskaneerimist kasutades.

1-2 – Punktis 3 nimetatud ühendamisoperatsiooni tulemus sorteeritakse alguses veeru perenimi ja seejärel veeru eesnimi alusel kasvavas järjekorras.



**Joonis 7. Täitmisplaan vaadatuna visuaalselt pgAdmin’is.**

## Rollid ja kasutajad

CREATE USER juhataja\_t142463 WITH PASSWORD 'parool1';

## Õiguste jagamine

„Laiale avalikkusele (PUBLIC) vaikimisi antud õiguste äravõtmine.“ [Eessaar,2015]

REVOKE CONNECT, TEMP ON DATABASE t134683 FROM PUBLIC;

REVOKE CREATE, USAGE ON SCHEMA public FROM PUBLIC;

REVOKE USAGE ON LANGUAGE plpgsql FROM PUBLIC;

REVOKE EXECUTE ON FUNCTION

f\_treeningu\_registreerimine

(Tootaja.tootaja\_id%TYPE, Spordiala.spordiala\_kood%TYPE, Treening.nimetus%TYPE, Treening.kirjeldus%TYPE), f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE), f\_treeningu\_aktiivseks\_muutmine(Treening.treening\_id%TYPE), f\_treeningu\_kustutamine (Treening.treening\_id%TYPE),

f\_uue\_labiviija\_lisamine (Treening.treening\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE),

f\_labiviija\_muutmine (Voimalik\_labiviija.voimalik\_labiviija\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE),

f\_sisselogimine\_juhataja(text, text),

f\_trigger\_treeningu\_kustutamine(),

f\_trigger\_treeningu\_muutmine\_treeningu\_seisund(),

f\_trigger\_uus\_treening\_tootaja\_seisund(),

f\_trigger\_uus\_treening\_amet()

FROM PUBLIC;

Kasutajale juhataja\_t142463 õiguste andmine.

GRANT CONNECT ON DATABASE t134683 TO juhataja\_t142463;

GRANT USAGE ON SCHEMA public TO juhataja\_t142463;

GRANT SELECT, UPDATE ON TABLE treening TO juhataja\_t142463;

GRANT SELECT ON TABLE isik TO juhataja\_t142463;

GRANT SELECT ON TABLE treeningu\_seisundi\_liik TO juhataja\_t142463;

GRANT SELECT ON TABLE tootaja TO juhataja\_t142463;

GRANT SELECT ON TABLE voimalik\_labiviija TO juhataja\_t142463;

GRANT SELECT ON

koik\_treeningud,

treeningute\_labiviijad

TO juhataja\_t142463;

GRANT EXECUTE ON FUNCTION f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE),

f\_sisselogimine\_juhataja(text, text),

f\_trigger\_treeningu\_muutmine\_treeningu\_seisund()

TO juhataja\_t142463;

„Vaikimisi õiguste muutmine tulevikus loodavate funktsioonide jaoks. See ei mõjuta olemasolevaid funktsioone.“[Eessaar,2015]

ALTER DEFAULT PRIVILEGES REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC;

## Andmebaasiobjektide kustutamine

### Domeenide kustutamine

DROP DOMAIN IF EXISTS d\_nimetus CASCADE;

DROP DOMAIN IF EXISTS d\_kirjeldus CASCADE;

### Tabelite kustutamine

ALTER TABLE IF EXISTS Tootaja DROP CONSTRAINT IF EXISTS FK\_Tootaja\_amet\_kood;

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

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

ALTER TABLE IF EXISTS Voimalik\_labiviija DROP CONSTRAINT IF EXISTS FK\_Tootaja\_tootaja\_id;

ALTER TABLE IF EXISTS Voimalik\_labiviija DROP CONSTRAINT IF EXISTS FK\_Voimalik\_labiviija\_treening\_id;

ALTER TABLE IF EXISTS Treening DROP CONSTRAINT IF EXISTS FK\_Treening\_treeningu\_seisundi\_liik\_kood;

ALTER TABLE IF EXISTS Treening DROP CONSTRAINT IF EXISTS FK\_Treening\_spordiala\_kood;

ALTER TABLE IF EXISTS Treening DROP CONSTRAINT IF EXISTS FK\_Treening\_tootaja\_id;

ALTER TABLE IF EXISTS Klient DROP CONSTRAINT IF EXISTS FK\_Klient\_kliendi\_seisundi\_liik\_kood;

ALTER TABLE IF EXISTS Klient DROP CONSTRAINT IF EXISTS FK\_Klient\_isikukood;

DROP TABLE IF EXISTS Tootaja CASCADE;

DROP TABLE IF EXISTS Voimalik\_labiviija CASCADE;

DROP TABLE IF EXISTS Treening CASCADE;

DROP TABLE IF EXISTS Amet CASCADE;

DROP TABLE IF EXISTS Tootaja\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Isik CASCADE;

DROP TABLE IF EXISTS Treeningu\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Spordiala CASCADE;

DROP TABLE IF EXISTS Klient\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Klient CASCADE;

### Arvujada generaatorite kustutamine

„Kuna arvujada generaatorid loodi SERIAL ja SMALLSERIAL tüüpide kasutamise abil, siis kustutas andmebaasisüsteem tabelite kustutamisel automaatselt nendega seotud arvujada generaatorid.“[Eessaar, 2015]

### Vaadete kustutamine

DROP VIEW IF EXISTS koik\_tootajad\_seisundiga CASCADE;

DROP VIEW IF EXISTS treeningute\_labiviijad CASCADE;

DROP VIEW IF EXISTS koik\_kliendid\_seisundiga CASCADE;

DROP VIEW IF EXISTS koik\_treeningud CASCADE;

### Indeksite kustutamine

DROP INDEX IF EXISTS idx\_tootaja\_tootaja\_seisundi\_liik\_kood;

DROP INDEX IF EXISTS idx\_tootaja\_amet\_kood;

DROP INDEX IF EXISTS idx\_treening\_tootaja\_id;

DROP INDEX IF EXISTS idx\_treening\_treeningu\_seisundi\_liik\_kood;

DROP INDEX IF EXISTS idx\_treening\_spordiala\_kood;

DROP INDEX IF EXISTS idx\_klient\_kliendi\_seisundi\_liik\_kood;

DROP INDEX IF EXISTS idx\_voimalik\_labiviija\_tootaja\_id;

DROP INDEX IF EXISTS idx\_voimalik\_labiviija\_treening\_id;

DROP INDEX IF EXISTS idx\_isik\_perenimi;

DROP INDEX IF EXISTS idx\_isik\_eesnimi;

### Funktsioonide ja trigerite kustutamine

DROP FUNCTION IF EXISTS f\_treeningu\_registreerimine

(Tootaja.tootaja\_id%TYPE, Spordiala.spordiala\_kood%TYPE, Treening.nimetus%TYPE, Treening.kirjeldus%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_treeningu\_aktiivseks\_muutmine(Treening.treening\_id%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_treeningu\_kustutamine (Treening.treening\_id%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_uue\_labiviija\_lisamine (Treening.treening\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_labiviija\_muutmine (Voimalik\_labiviija.voimalik\_labiviija\_id%TYPE,

Tootaja.tootaja\_id%TYPE, Voimalik\_labiviija.alguse\_aeg%TYPE, Voimalik\_labiviija.lopu\_aeg%TYPE) CASCADE;

DROP FUNCTION IF EXISTS f\_sisselogimine\_juhataja(text,text) CASCADE;

DROP FUNCTION IF EXISTS f\_trigger\_treeningu\_kustutamine() CASCADE;

DROP FUNCTION IF EXISTS f\_trigger\_treeningu\_muutmine\_treeningu\_seisund() CASCADE;

DROP FUNCTION IF EXISTS f\_trigger\_uus\_treening\_tootaja\_seisund() CASCADE;

DROP FUNCTION IF EXISTS f\_trigger\_uus\_treening\_amet() CASCADE;

DROP TRIGGER IF EXISTS trigger\_treeningu\_kustutamine ON Treening CASCADE;

DROP TRIGGER IF EXISTS trigger\_treeningu\_muutmine\_treeningu\_seisund ON Treening CASCADE;

DROP TRIGGER IF EXISTS trigger\_uus\_treening\_tootaja\_seisund ON Treening CASCADE;

DROP TRIGGER IF EXISTS trigger\_uus\_treening\_amet ON Treening CASCADE;

### Reeglite kustutamine

„Kuna projektis reegleid ei kasutata, siis ei esitata ka reeglite kustutamise lauseid.“ [Eessaar, 2015]

### Kasutajate ja rollide kustutamine

„Kuna projektis rolle ei kasutata, siis ei esitata ka rollide kustutamise lauseid. Enne kasutaja kustutamist tuleb ära võtta sellele antud õigused.“ [Eessaar, 2015]

REVOKE CONNECT ON DATABASE t134683 FROM juhataja\_t142463;

REVOKE USAGE ON SCHEMA public FROM juhataja\_t142463;

REVOKE SELECT, UPDATE ON TABLE treening FROM juhataja\_t142463;

REVOKE SELECT ON TABLE isik FROM juhataja\_t142463;

REVOKE SELECT ON TABLE treeningu\_seisundi\_liik FROM juhataja\_t142463;

REVOKE SELECT ON TABLE tootaja FROM juhataja\_t142463;

REVOKE SELECT ON TABLE voimalik\_labiviija FROM juhataja\_t142463;

REVOKE SELECT ON

koik\_treeningud,

treeningute\_labiviijad

FROM juhataja\_t142463;

REVOKE EXECUTE ON FUNCTION f\_treeningu\_mitteaktiivseks\_muutmine(Treening.nimetus%TYPE),

f\_sisselogimine\_juhataja(text, text),

f\_trigger\_treeningu\_muutmine\_treeningu\_seisund()

FROM juhataja\_t142463;

DROP USER IF EXISTS juhataja\_t142463;

# Kasutatud materjalid

1. AKIT. Andmekaitse ja infoturbe seletussõnastik. [WWW] <http://akit.cyber.ee/> (01.02.2015)
2. Andmebaasid II õppematerjalid. [WWW] <http://maurus.ttu.ee> (19.12.2015)
3. Erki Eessaare näiteprojekt aines Andmebaasid II [WWW] http://maurus.ttu.ee/ained/IDU0230\_2015/doc/2/Naidisprojekt\_IDU0230\_vastuvotuajad\_ver4\_5.pdf (11.12.2015)
4. Chisholm, M. (2000). *Managing Reference Data in Enterprise Databases:* Binding *Corporate Data to the Wider World.* Morgan Kaufmann.
5. Eesti Statistika. Riikide ja territooriumide klassifikaator 2013v1. [WWW] <http://metaweb.stat.ee/view_xml_multi_code.htm?id=3477719&siteLanguage=ee> (30.06.2015)
6. ESTERM [WWW] <http://mt.legaltext.ee/esterm/> (08.02.2015)
7. Infosüsteemide turvameetmete süsteem. Vabariigi Valitsuse 20.12 2007. a määrus nr 252. Elektrooniline Riigi Teataja.  
   [WWW] <https://www.riigiteataja.ee/akt/13125331?leiaKehtiv> (08.02.2015)
8. Vikipeedia. Riik. [WWW] <https://et.wikipedia.org/wiki/Riik> (30.06.2015)
9. Andmekaitse Inspektsioon (2013). Isikukoodi kasutamise juhend. [WWW] http://www.aki.ee/sites/www.aki.ee/files/elfinder/article\_files/Isikukoodi%20k asutamise%20juhend.pdf (28.08.2015)
10. Isikukoodide moodustamise, väljajagamise ja andmise kord. Regionaalministri määrus. Vastu võetud 07.01.2005 nr 4. Elektrooniline Riigi Teataja. [WWW] https://www.riigiteataja.ee/akt/838530?leiaKehtiv (06.06.2015)