TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Suusaklubi radade arvestus

Andmebaasid II IDU0230

|  |  |
| --- | --- |
| Üliõpilane: | Rasmus Rüngenen, Martin Kalvik, Nikita Kovalenko |
| Õpperühm: | IAPB43, IAPB42, IAPB41 |
| Matrikli nr: | 164640IAPB, 164664IAPB, 172779IAPB |
| e-posti aadress: | [r.ryngenen@gmail.com](mailto:r.ryngenen@gmail.com)  [martin.kalvik@hotmail.com](mailto:martin.kalvik@hotmail.com)  [nikova@ttu.ee](mailto:nikova@ttu.ee) |
|  |  |

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

Tallinn

2018

**Sisukord**

[**1**](#_14ykbeg) **STRATEEGILINE ANALÜÜS 7**

[1.1](#_3oy7u29) Terviksüsteemi üldvaade 7

*[1.1.1](#_243i4a2)**Organisatsiooni eesmärgid 7*

*[1.1.2](#_j8sehv)**Infosüsteemi eesmärgid 7*

[*1.1.3*](#_338fx5o) *Lausendid 8*

[*1.1.4*](#_1idq7dh) *Põhiobjektid 8*

[*1.1.5*](#_42ddq1a) *Põhiprotsessid ja neid käivitavad sündmused 9*

[*1.1.6*](#_2hio093) *Tegutsejad 10*

[*1.1.7*](#_wnyagw) *Asukohad 11*

[*1.1.8*](#_3gnlt4p) *Terviksüsteemi tükeldus allsüsteemideks 11*

[1.2](#_2jxsxqh) Radade funktsionaalse allsüsteemi eskiismudelid 12

*[1.2.1](#_1vsw3ci)**Eesmärgid 12*

[*1.2.2*](#_4fsjm0b) *Allsüsteemi kasutavad pädevusalad 12*

[*1.2.3*](#_2uxtw84) *Allsüsteemi poolt vajatavad registrid 12*

[*1.2.4*](#_1a346fx) *Allsüsteemi ühe põhiprotsessi tegevusdiagramm 13*

[*1.2.5*](#_3u2rp3q) *Allsüsteemi kasutusjuhtude eskiismudel 14*

[*1.2.6*](#_2bn6wsx) *Mittefunktsionaalsed nõuded 16*

[1.3](#_3as4poj) Radade registri eskiismudelid 19

[*1.3.1*](#_2981zbj) *Eesmärgid 19*

[*1.3.2*](#_odc9jc) *Registrit kasutavad pädevusalad 19*

[*1.3.3*](#_38czs75) *Registrit teenindavad funktsionaalsed allsüsteemid 19*

[*1.3.4*](#_1nia2ey) *Infovajadused, mida register aitab rahuldada 19*

[*1.3.5*](#_47hxl2r) *Seosed teiste registritega 19*

[*1.3.6*](#_2mn7vak) *Ärireeglid 20*

[*1.3.7*](#_11si5id) *Registri kontseptuaalne eskiismudel 20*

[**2**](#_3ls5o66) **DETAILANALÜÜS 22**

[2.1](#_41mghml) Radade funktsionaalse allsüsteemi detailanalüüs 22

[*2.1.1*](#_20xfydz) *Kasutusjuhtude mudel 22*

[2.2](#_vx1227) Radade funktsionaalse allsüsteemi vajatavate registrite detailanalüüs 29

[*2.2.1*](#_4kx3h1s) *Kontseptuaalne andmemudel 29*

[*2.2.2*](#_302dr9l) *Andmebaasioperatsioonide lepingud 35*

[*2.2.3*](#_1f7o1he) *Registri põhiobjekti seisundidiagramm 39*

[2.3](#_3z7bk57) CRUD maatriks 40

[**3**](#_2eclud0) **FÜÜSILINE DISAIN 41**

[3.1](#_46r0co2) Radade funktsionaalse allsüsteemi vajatavate registrite füüsiline disain 41

[**4**](#_thw4kt) **REALISATSIOON POSTGRESQLIS 42**

[4.1](#_3dhjn8m) Andmebaasi loomine 42

[4.2](#_1smtxgf) Skeemid 42

[4.3](#_4cmhg48) Domeenid 42

[4.4](#_2rrrqc1) Tabelid ja arvujada generaatorid 42

[4.5](#_16x20ju) Vaated 42

[4.6](#_3qwpj7n) Protseduursed keeled 42

[4.7](#_261ztfg) Trigeri funktsioonid ja trigerid 42

[4.8](#_l7a3n9) Reeglid 42

[4.9](#_356xmb2) Funktsioonid 42

[4.10](#_1kc7wiv) Indeksid 43

[*4.10.1*](#_44bvf6o) *Välisvõtmete veergudele lisatavad indeksid 43*

[*4.10.2*](#_2jh5peh) *Täiendavad sekundaarsed indeksid 43*

[*4.10.3*](#_ymfzma) *Funktsioonil põhinevad indeksid 43*

[4.11](#_3im3ia3) Klassifikaatorite väärtustamise SQL laused 43

[4.12](#_1xrdshw) JSON formaadis lähteandmete laadimine 43

[4.13](#_25b2l0r) Täiendavate testandmete lisamine 43

[4.14](#_4hr1b5p) Andmebaasi statistika kogumine 43

[4.15](#_2wwbldi) Päringu täitmisplaani näide 43

[4.16](#_1c1lvlb) Rollid ja kasutajad 43

[4.17](#_3w19e94) Üleliigsete õiguste äravõtmine 43

[4.18](#_2b6jogx) Õiguste jagamine 44

[4.19](#_qbtyoq) Andmebaasiobjektide kustutamine 44

[*4.19.1*](#_3abhhcj) *Õiguste äravõtmine 44*

[*4.19.2*](#_1pgrrkc) *Domeenide kustutamine 44*

[*4.19.3*](#_49gfa85) *Tabelite ja arvujada generaatorite kustutamine 44*

[*4.19.4*](#_2olpkfy) *Vaadete kustutamine 44*

[*4.19.5*](#_13qzunr) *Indeksite kustutamine 44*

[*4.19.6*](#_3nqndbk) *Funktsioonide ja trigerite kustutamine 44*

[*4.19.7*](#_22vxnjd) *Reeglite kustutamine 44*

[*4.19.8*](#_i17xr6) *Kasutajate ja rollide kustutamine 44*

[*4.19.9*](#_39kk8xu) *Laienduste kustutamine 44*

[**5**](#_320vgez) **REALISATSIOON ORACLES 45**

[5.1](#_1h65qms) Andmebaasi kohandamine keele- ja kultuurikeskkonnale 45

[5.2](#_415t9al) Skeem 45

[5.3](#_2gb3jie) Tabelid ja arvujada generaatorid. 45

[5.4](#_vgdtq7) Sünonüümid 45

[5.5](#_3fg1ce0) Vaated 45

[5.6](#_1ulbmlt) Paketid 45

[5.7](#_4ekz59m) Trigerid 45

[5.8](#_2tq9fhf) Indeksid 46

[*5.8.1*](#_18vjpp8) *Välisvõtmete veergudele lisatavad indeksid 46*

[*5.8.2*](#_3sv78d1) *Täiendavad sekundaarsed indeksid 46*

[*5.8.3*](#_280hiku) *Funktsioonil põhinevad indeksid 46*

[5.9](#_n5rssn) Klassifikaatorite väärtustamise SQL laused 46

[5.10](#_375fbgg) XML formaadis lähteandmete laadimine 46

[5.11](#_4du1wux) Täiendavate testandmete lisamine 46

[5.12](#_1maplo9) Andmebaasi statistika kogumine 46

[5.13](#_46ad4c2) Päringu täitmisplaani näide 46

[5.14](#_2lfnejv) Rollid ja kasutajad 46

[5.15](#_10kxoro) Õiguste jagamine 46

[5.16](#_3kkl7fh) Andmebaasiobjektide kustutamise SQL laused 47

[*5.16.1*](#_1zpvhna) *Õiguste äravõtmine 47*

[*5.16.2*](#_4jpj0b3) *Tabelite ja arvujada generaatorite kustutamine 47*

[*5.16.3*](#_2yutaiw) *Sünonüümide kustutamine 47*

[*5.16.4*](#_1e03kqp) *Vaadete kustutamine 47*

[*5.16.5*](#_3xzr3ei) *Indeksite kustutamine 47*

[*5.16.6*](#_2d51dmb) *Pakettide kustutamine 47*

[*5.16.7*](#_sabnu4) *Trigerite kustutamine 47*

[*5.16.8*](#_3c9z6hx) *Kasutajate ja rollide kustutamine 47*

[**6**](#_2y3w247) **KASUTATUD MATERJALID 48**

**NB!** Kõik üliõpilased, kes **ei tee** õppeaines *Andmebaasid I* iseseisvat tööd kasutades MS Access andmebaasisüsteemi, peavad selle õppeaine iseseisva töö dokumendi lõpus esitama andmebaasi tabelite ja kitsenduste loomise SQL laused. Kes kasutavad MS Accessi, need taolisi lauseid esitama ei pea.

**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.

Rasmus Rüngenen

[Allkiri]

Martin Kalvik

[Allkiri]

Nikita Kovalenko

[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 |
| Rakenduse loomise vahendid (arenduskeskkonnad, keeled ja nende versioonid) | MS Access 2016 |
| Andmebaasi server **(andmebaas PEAB olema apex.ttu.ee serveris)** | Apex.ttu.ee |
| PostgreSQL andmebaasi nimi (kui kasutate PostgreSQLi) | t164640 |
| Oracle skeemi nimi (kui kasutate Oraclet) |  |
| Millise töökoha rakendus realiseerib? | juhataja |
| Rakenduse aadress (kui on veebirakendus) **(rakenduse panek apex.ttu.ee serverisse pole kohustuslik; aadress tuleb esitada ka pgApex ja Oracle APEX rakenduse puhul)** | - |
| Rakenduse nimi (kui rakendus on tehtud pgApex või Oracle APEX keskkonnas) | - |
| Kui tegemist on kahekihilise klient-server süsteemiga ja rakendus on kasutaja arvutis, siis mida peab kasutaja arvutis tegema, et seda rakendust kasutada | Rakenduse 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 |
| Kasutajanimi ja parool rakendusse sisselogimiseks | kasutajanimi: kasutaja1  parool: testparool2  kasutajanimi: kasutaja4  parool: testparool4 |
| Kui rakenduse failil on eraldi parool (nt MS Accessi korral – ei ole kohustuslik), siis mis see on? | - |
| Millistes vormides (MS Accessi korral), lehekülgedel (pgApex või Oracle APEX korral), failides (mingis programmeerimiskeeles loodud rakenduse puhul) toimub pöördumine andmebaasiserveris talletatud rutiinide poole? Faili puhul palun näidata ka kataloog. | Ekraanivormil *Sissereg* pöördutakse nupule *Jätka* vajutamise järel kasutaja-defineeritud funkstiooni *f\_on\_oppejoud*  poole. Funktsioon on mõeldud õppejõu identifitseerimiseks.   Ekraanivormil *Loo\_rada* poordutakse nupule |
| Millistes vormides (MS Accessi korral), lehekülgedel (pgApex või Oracle APEX korral), failides (mingis programmeerimiskeeles loodud rakenduse puhul) toimub pöördumine andmebaasis loodud vaadete poole? Faili puhul palun näidata ka kataloog. | Ekraanivormil *Aktiivsed\_mitteaktiivsed\_rajad* esitatavad andmed leitakse vaatest *Aktiivsed\_mitteaktiivsed\_rajad.*  Ekraanivormil *Vaata\_koiki\_radu* esitatavad andmed leitakse vaatest *Vaata\_koiki\_radu.*  Ekraanivormil |
| 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 (*juhataja\_ylevaade)* on kirjutatud mooduli *Refreshing* alugsesse (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 (tabelis Isik veerud *email* ja *parool)* |
| Kuidas on andmebaasis salvestatud parool kaitstud? Kui kasutate räsiväärtuse leidmist, siis tuleb selgelt nimetada räsi arvutamise algoritm. Kui kasutate soola, siis tuleb see selgelt välja öelda ning kirjeldada selle genereerimist ja omadusi. | Andmebaasis hoitakse parooli räsiväärtust. See väärtus leitakse andmebaasisüsteemi poolt kasutades sisendina avatekstina parooli ning süsteemi-genereeritud soola.  Parooli räsiväärtuse leidmiseks kasutatakse  PostgreSQL funktsiooni crypt koos soola  genereerimiseks mõeldud funktsiooniga 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; |
| 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 email ja teine argument parool. |
| Kas olete oma töö iseseisva töö hindamismudeli alusel enne esitamist ise üle kontrollinud? |  |
| 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 radade infosüsteemist.

### Organisatsiooni eesmärgid

* Teenida omanikele kasumit
* Pakkuda head ja korralikul tasemel suusaradu, mis jätaks klientidele hea mulje ning suurendaks võimalust, et neist saavad korduskülastajad ja soovitavad suusaradasid ka oma tuttavatele
* Pakkuda inimestel võimalust edendada tervisesporti
* Võimaldada klientidele laenutada sportimiseks vajalikku varustust
* Pakkuda spordiüritustele võimaliku toimumiskohta
* Pakkuda huvilistele ülevaadet suusaradade korrasolust
* Hooldada organisatsiooni radu

### Infosüsteemi eesmärgid

* Tagada ülevaade organisatsiooniga seotud isikute isikuandmetest
* Tagada ülevaade organisatsiooni töötajatest
* Tagada ülevaade organisatsiooni klientidest
* 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 organisatsiooni käsutuses olevatest varadest
* Võimaldada organisatsioonil varade otsalõppemisel või riknemisel varusid täiendada, tehes tarnetellimusi tarnijatele (partneriteks olevad organisatsioonid)
* Tagada ülevaade varade laoseisu muutusest, mida väljendatakse laoliikumistena
* 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 ülevaadet organitsatsiooni varadega toimunud intsidentidest, sh rikked ja vargused
* 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 radadest, millega tehingute (transaktsioonide) tegemine on üks organisatsiooni põhieesmärk
* Tagada raja pileti müügi salvestamine infosüsteemis
* Tagada ülevaade toimuvatest üritustest
* Tagada ülevaade toimunud üritustest
* Tagada ülevaade radade erinevatest raskusastmetest

### Lausendid

* Töötaja on isik
* Klient on isik
* Partner on organisatsioon
* Partneriga sõlmitakse leping
* Varade haldur, kes allub juhatajale, valdab vara
* Vara otsalõppemisel teeb varade haldur partnerile vara tarnetellimuse
* Vara tarnetellimuse täitmisele järgneb vara laoliikumine
* Varade tegeliku seisu kindlakstegemiseks toimub varade halduri poolt läbiviidav inventuur
* Partner esitab arve
* Arve on ühtlasi dokument
* Arvete korrektsuse eest vastutab juhiabi
* Meie organisatsioon müüb kliendile raja pileti
* Raja pilet on ühtlasi dokument
* Klient annab tagasisidet
* Varaga toimub intsident
* Töötaja teeb infosüsteemi tööd puudutava ettepaneku
* Juhataja planeerib arendustöö
* Juhataja koostab eelarve
* Raja haldur registreerib radade seisu
* Rada iseloomustab null või rohkem suusaraja kategooriat
* Raja kategooria on klassifikaator
* Raja raskus on klassifikaator
* Klassifikaatorite haldur registreerib klassifikaatori
* Uudistajale pakuvad huvi radade andmed
* Radade haldur sisestab andmeid radade seiskorra kohta
* Klienditeenindaja teenindab kliente varustuse laenutusel
* Töötaja teeb infosüsteemi tööd puudutava ettepaneku
* Töötaja kannab infosüsteemi sisse toimuva võistluse andmed
* Võistlus on organisatsiooni radadel toimuv üritus
* Raja haldur teostab radade hooldamist

### Põhiobjektid

* Isik
* Töötaja
* Klient
* Organisatsioon
* Partner
* Klassifikaator
* Leping
* Vara
* Vara tarnetellimus
* Vara laoliikumine
* Inventuur
* Dokument
* Arve
* Kliendi tagasiside
* Intsident
* Töötaja ettepanek
* Arendustöö
* Eelarve
* Rada
* Võistlus
* Raja pilet
* Hooldamine

### 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 |
| Töötaja töösuhte lõpetamine  Kliendisuhte lõpetamine  Organisatsiooniga koostöö lõpetamine | Organisatsioonini jõuab teave, et soovitakse lõpetada kas koostöö või töösuhe |
| 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 |
| Kliendi registreerimine | Organisatsiooni teenuseid soovib hakata kasutama uus eraklient |
| Organisatsiooni registreerimine | Organisatsiooni vaatevälja ilmub uus partnerorganisatsioon, näiteks tarnija või remonditeenuste pakkuja |
| 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 või on vaja klassifikaatori väärtust uuendada |
| 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 |
| Vara arvelevõtmine | Organisatsioonile ostetakse või võetakse rendile uut vara |
| Vara tarnetellimuse tegemine | Mingit liiki vara saab otsa ning organisatsiooni toimimise tagamiseks peab seda juurde tellima |
| Vara tarnetellimuse täidetuks märkimine | Kogu tarnetellimusega tellitud vara jõuab kohale |
| Vara laoliikumise registreerimine | Saabub saadetis, millega täidetakse osaliselt või täielikult üks või mitu vara tarnetellimust |
| Inventuuri alustamine | Majandusaasta lõpp  Erakordne sündmus nagu õnnetus (nt tulekahju, veekahju) või rünne (nt vargus) |
| Inventuuri lõpetamine | Varade haldur esitab olemasolevate varade kohta ülevaate |
| 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 või oma arvamuse pakutava teenuse kohta |
| Intsidendi registreerimine | Vara tabab rike |
| Töötaja ettepaneku registreerimine | Töötaja näeb võimalust muuta organisatsiooni tööd efektiivsemaks, sh parandada infosüsteemi tarkvara |
| 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 |
| Raja registreerimine | Organisatsiooni jõuab teave uue raja kohta |
| Raja unustamine | Selgus, et organisatsiooni jõudnud teave radade kohta on enneaegne ning sellisel kujul rada ei ole vaja registreerida |
| Raja aktiveerimine | On vaja muuta võimalikuks raja kasutamine tehingutes |
| Raja ajutiselt kasutusest eemaldamine (mitteaktiivseks muutmine) | Raja kasutamine tehingutes on vaja ajutiselt peatada, kuna seoses rajaga on ilmnenud ajutise iseloomuga probleemid |
| Radade lõplikult kasutusest eemaldamine (lõpetamine) | Raja kasutamine tehingutes on vaja lõpetada, kuna seoses rajaga on ilmnenud püsiva iseloomuga probleemid või kuna rada on oma aja lihtsalt ära elanud |
| Võistluse registreerimine | Organisatsiooni jõuab teave uue võistluse kohta |
| Võistluse info muutmine | Selgub, et võistluse info ei vasta tegelikkusele ning on vaja teha muudatusi andmetes |
| Võistluse kustutamine | Võistlus on lõppenud või võistlus tühistatakse enneaegselt |
| Raja pileti müümine | Klient soovib osta raja pileti |
| Raja pileti registreerimine | Raja pileti ostu tagajärjel on vaja muuta raja pilet kehtivaks |
| Raja pileti eemaldamine | Raja piletil olev kasutusaeg on täis |

### Tegutsejad

* Juhataja (ka omanik)
* Juhiabi
* Varade haldur
* Personali juht
* Radade haldur
* Klassifikaatorite haldur
* Klienditeenindaja
* Klient
* Uudistaja
* Töötaja

### Asukohad

* Kliendid (on süsteemis registreeritud) ja uudistajad (veebikülalised; tuvastamata kasutajad) kasutavad veebirakendust, mille poole pöördumiseks on vaja arvutit, veebilehitsejat ja veebiühendust.
* Töötajad töötavad neile spetsiaalselt ettenähtud ruumides. Igale töötajale on ettenähtud oma arvuti.
* Ettenähtud ruumides, kus saab töötajalt raja pileti, on olemas iseteenindus masinad.
* Iseteenindus masina kasutamiseks peab klient olema registreeritud.
* Radade halduril on olemas tahvelarvuti, millega saab läbi veebirakenduse anda jooksvalt infot radade seisukorra kohta.

### Terviksüsteemi tükeldus allsüsteemideks

Järgnevalt esitatakse infosüsteemi jaotus kolme erinevat liiki allsüsteemideks.

Organisatsiooni sisesed pädevusalad.

* Juhataja
* Radade haldur
* Klassifikaatorite haldur
* Juhiabi
* Varade haldur
* Personali juht
* Töötaja
* Klienditeenindaja

Organisatsiooni välised pädevusalad.

* Klient
* Uudistaja

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

**Tabel 3 Sisulised allsüsteemid.**

|  |  |
| --- | --- |
| **Funktsionaalne allsüsteem** | **Register, mida see funktsionaalne allsüsteem teenindab** |
| Radade funktsionaalne allsüsteem | Radade register |
| Klientide funktsionaalne allsüsteem | Klientide register |

Tabel 4 esitab administratiivsed funktsionaalsed allsüsteemid ja nende teenidatavad 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 allsüstee | Organisatsioonide register |
| Partnerite funktsionaalne allsüsteem | Partnerite register |
| Klassifikaatorite funktsionaalne allsüsteem | Klassifikaatorite register |
| Lepingute funktsionaalne allsüsteem | Lepingute register |
| Varade funktsionaalne allsüsteem | Varade register |
| Vara tarnetellimuste funktsionaalne allsüsteem | Vara tarnetellimuste register |
| Vara laoliikumiste funktsionaalne allsüsteem | Vara laoliikumiste register |
| Inventuuride funktsionaalne allsüsteem | Inventuuride register |
| Dokumentide funktsionaalne allsüsteem | Dokumentide register |
| Arvete funktsionaalne allsüsteem | Arvete register |
| Klientide tagasiside funktsionaalne allsüsteem | Klientide tagasiside register |
| Intsidentide funktsionaalne allsüsteem | Intsidentide register |
| Töötajate ettepanekute funktsionaalne allsüsteem | Töötajate ettepanekute register |
| Arendustööde funktsionaalne allsüsteem | Arendustööde register |
| Eelarvete funktsionaalne allsüsteem | Eelarvete register |
| Raja piletite müügi funktsionaalne allsüsteem | Raja piletite register |
| Võistluste funktsionaalne allsüsteem | Võistluste register |

## Radade funktsionaalse allsüsteemi eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

* Muuta võimalikuks radade kasutamine erinevates tehingutes (transaktsioonides), mille läbiviimist infosüsteem toetab
* Võimaldada rada elektrooniliselt registreerida
* Võimaldada määrata raja hetkeseisundit vastavalt elutsüklile
* Võimaldada muuta süsteemile teadaolevaid andmeid raja kohta
* Võimalik raja andmed kustutada e infosüsteemi mõttes unustada, kuid teha seda ainult siis, kui rada pole veel kordagi aktiivsesse kasutusse läinud ja seega pole sellega seotud veel ühtegi tehingut
* Võimaldada uudistajatel ja klientidel pärida infot radade seisukorra kohta
* Võimaldada vastata fikseeritud päringutele radade kohta

### Allsüsteemi kasutavad pädevusalad

* Juhataja
* Radade haldur
* Uudistaja
* Klient
* Klienditeenindaja

### Allsüsteemi poolt vajatavad registrid

Allsüsteem teenindab radade registrit.

Allsüsteem loeb.

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

### Allsüsteemi ühe põhiprotsessi tegevusdiagramm

Joonis 1 esitab radade lõpetamise protsessi kirjelduse tegevusdiagrammina. 

**Joonis 1 Radade 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 Radade funktsionaalse allsüsteemi kasutusjuhtude diagramm.**

**Kasutusjuht**: Tuvasta kasutaja

**Tegutsejad**: Radade haldur, Töötaja, Juhiabi, Varade haldur, Personali haldur, Juhataja, Klient – (edaspidi Subjekt)

**Kirjeldus**: Subjekt identifitseerib ennast. Selleks sisestab ta kasutajanime ja parooli. 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 rada

**Tegutsejad**: Radade haldur

**Kirjeldus**: Radade haldur registreerib uue raja.

**Kasutusjuht**: Unusta rada

**Tegutsejad**: Radade haldur

**Kirjeldus**: Radade haldur vaatab ootel radade nimekirja, valib sealt raja ja kustutab selle andmebaasist. Subjekt saab nimekirja sorteerida ja filtreerida.

**Kasutusjuht**: Muuda raja andmeid

**Tegutsejad**: Radade haldur

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

**Kasutusjuht**: Aktiveeri rada

**Tegutsejad**: Radade haldur

**Kirjeldus**: Radade haldur vaatab ootel või mitteaktiivsete radade nimekirja, valib sealt raja ja muudab selle aktiivseks.

**Kasutusjuht**: Muuda rada mitteaktiivseks

**Tegutsejad**: Radade haldur

**Kirjeldus**: Radade haldur vaatab aktiivsete radade nimekirja, valib sealt raja ja muudab selle mitteaktiivseks. Subjekt saab nimekirja sorteerida ja filtreerida.

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

**Tegutsejad**: Radade haldur

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

**Kasutusjuht**: Vaata kõiki radu

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

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

**Kasutusjuht**: Lõpeta rada

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata radade koondaruannet

**Tegutsejad**: Juhataja

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

**Kasutusjuht**: Vaata aktiivseid radu

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

**Kirjeldus**: Subjekt valib kategooria ja näeb kõigi sellesse kuuluvate aktiivses seisundis olevate radade kõiki andmeid, v.a 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). Radadega 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 Rational Rose või Enterprise Architect. Alternatiivina võib kasutada tasuta pakutavat andmete modelleerimise vahendit DB MAIN  (<http://www.rever.eu/en/content/db-main-homepage>). Prototüübi koostamiseks kasutatakse töölaua andmebaasisüsteemi MS Access või LibreOffice Base, kuhu on integreeritud kasutajaliidese ehitamise vahendid. |
| 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. |
| 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). |
| 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. |
| 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*.   Radade funktsionaalne allsüsteem teenindab radade 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; |
| varukoopiad | Kuna hallatavad andmed on organisatsiooni jaoks väga olulised, siis tuleb vähemalt kord päevas teha andmetest varukoopia ja säilitada koopiaid mitmes erinevas asukohas. |

## Radade registri eskiismudelid

Järgnevalt esitatakse eskiismudelid, mida detailanalüüsi käigus täpsustatakse ja täiendatakse.

### Eesmärgid

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

### Registrit kasutavad

* Juhataja
* Radade haldur
* Klienditeenindaja
* Klient
* Uudistaja

### Registrit teenindavad funktsionaalsed allsüsteemid

Radade registrit teenindab (loeb ja muudab) radade funktsionaalne allsüsteem.

### Infovajadused, mida register aitab rahuldada

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

### Seosed teiste registritega

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

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

Selleks, et saaks registreerida andmeid raja pileti registris, peavad olema registreeritud radade andmed ja klientide andmed ja seega peab olema realiseeritud radade register ning klientide register vastavalt.

### Ärireeglid

Jõustatavad radade registri põhjal

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

Jõustatavad teiste registrite põhjal, kuid vajalikud raja funktsionaalse allsüsteemi toimimiseks

* Iga isiku kasutajanimena kasutatakse tema unikaalset e-meili aadressi
* Iga isiku unikaalseks identifikaatoriks on kombinatsioon isikukoodist ja selle väljastanud riigi koodist

### Registri kontseptuaalne eskiismudel

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



**Joonis 3 Radade 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 radade funktsionaalne allsüsteem vajab toimimiseks.

## Radade funktsionaalse allsüsteemi detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt radade funktsionaalse allsüsteemi toimimist.

### Kasutusjuhtude mudel

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

**Osapooled ja nende huvid**:

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

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

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

**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 rada

**Primaarne tegutseja**: Radade haldur

**Osapooled ja nende huvid**:

* Radade haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate radade 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 radadest ning uue raja tekkimisel ei tohi selle registreerimisega viivitada.
* Klient, Uudistaja: Soovivad võimalikult täpset infot radade kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga radu kasutava kliendi rollis.

**Käivitav sündmus**: Organisatsiooni jõuab teave uue raja kohta, millega kliendid saavad hakata tulevikus kasutama.

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

**Järeltingimused**: Rada on registreeritud ja rada on seisundis „Ootel“.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

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

3a Radade haldur soovib rada mõnest määratud kategooriast kohe eemaldada.

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

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

**Kasutusjuht:** Unusta rada

**Primaarne tegutseja**: Radade haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Radade haldur on autenditud ja autoriseeritud. Rada on registreeritud ja on seisundis „Ootel“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Radade haldur avaldab soovi rada unustada, st selle andmed süsteemist kustutada.
2. **Süsteem** kuvab ootel radade nimekirja, kus on kood ja nimetus.(**OP3.1**)
3. Radade haldur valib nimekirjast raja ja annab korralduse see unustada.
4. **Süsteem** salvestab andmed (**OP2**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi ootel rada, siis ei saa radade haldur jätkata.

**Kasutusjuht:** Muuda raja andmeid

**Primaarne tegutseja**: Radade haldur

**Osapooled ja nende huvid**:

* Radade haldur: Soovib, et süsteemis oleks kõikide organisatsioonile teadaolevate radade 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 radadest.
* Klient, Uudistaja: Soovivad võimalikult täpset infot radade kohta, mida organisatsioon pakub, et otsustada, kas siduda ennast selle organisatsiooniga radu kasutava kliendi rollis.

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

5a. Radade haldur võib lisada raja uude kategooriasse ja anda korralduse salvestada.

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

5b Radade haldur võib eemaldada raja kategooriast ja anda korralduse salvestada.

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

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

…

**Kasutusjuht:** Aktiveeri rada

**Primaarne tegutseja**: Radade haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Radade haldur on autenditud ja autoriseeritud. Rada on registreeritud ja on seisundis „Ootel“ või „Mitteaktiivne“. Rada on määratud vähemalt ühte radade kategooriasse.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

*Radade 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 rada, siis ei saa radade haldur jätkata.

4a. Kui rada ei kuulu ühtegi radade kategooriasse, siis aktiveerimine ebaõnnestub.

**Kasutusjuht**: Muuda Rada mitteaktiivseks

**Primaarne tegutseja**: Radade haldur

**Osapooled ja nende huvid**:

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

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

**Eeltingimused**: Radade haldur on autenditud ja autoriseeritud. Rada on registreeritud ja on seisundis „Aktiivne“.

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Radade haldur avaldab soovi rada mitteaktiivseks muuta.
2. **Süsteem** kuvab aktiivsete radade nimekirja, kus on kood, nimetus, raja\_raskus, raja\_pikkus (**OP6.1**)
3. Radade haldur valib nimekirjast raja ja annab korralduse see mitteaktiivseks muuta.
4. **Süsteem** salvestab andmed (**OP4**).

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

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

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

3b. Kui nimekirjas ei ole ühtegi aktiivset rada, siis ei saa radade haldur jätkata.

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

**Primaarne tegutseja**: Radade haldur.

**Osapooled ja nende huvid**:

* Radade haldur: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

**Kasutusjuht**: Vaata kõiki radu

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

**Osapooled ja nende huvid**:

* Juhataja, radade haldur: Soovib sisendit juhtimisotsuste tegemiseks.

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

1. Subjekt soovib vaadata kõikide radade andmeid.
2. **Süsteem** kuvab kõigi radade nimekirja, kus on Raja\_kood, hetkeseisundi nimetus, nimetus (**OP8.1**)
3. Subjekt valib raja, mida ta soovib detailsemalt vaadata.
4. **Süsteem** kuvab vaatamiseks mõeldud väljades andmed kõigi radade ja selle võimalike alamtüüpide omaduste ning seoste kohta. Muuhulgas kuvab süsteem radade põhiandmed (Raja\_kood, nimetus, raja\_pikkus, reg\_aeg, registreerinud töötaja eesnimi, perenimi ja e-meili aadress)
5. (**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 rada, siis ei saa subjekt jätkata.

**Kasutusjuht**: Lõpeta rada

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

**Käivitav sündmus**: Raja kasutamine tehingutes on vaja püsivalt lõpetada, kuna seoses rajaga on ilmnenud püsiva iseloomuga probleemid või kuna rada on oma aja lihtsalt ära elanud

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

**Järeltingimused**: Raja seisund on muutunud „Lõpetatud“, kuid raja andmed on süsteemis endiselt alles. Radade andmeid ei tohi süsteemist füüsiliselt kustutada, sest sellega seoses tuleks kustutada info kõikide tehingute kohta, millega rajad on seotud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

**Kasutusjuht**: Vaata radade koondaruannet

**Primaarne tegutseja**: Juhataja

**Osapooled ja nende huvid**:

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

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

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

**Järeltingimused**: Radade koondaruanne on moodustatud.

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

**Kasutusjuht**: Vaata aktiivseid radu

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

**Osapooled ja nende huvid**:

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

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

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

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

**Stsenaarium (tüüpiline sündmuste järjestus)**:

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

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

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

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

## Radade funktsionaalse allsüsteemi vajatavate registrite detailanalüüs

Järgnevalt kirjeldatakse detailselt ja mittetehniliselt radade 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 *radade registri* põhiobjekt.
* Kollasega on tähistatud *radade registrisse* kuuluvad mitte-põhiobjektid.
* **Rohelisega** on tähistatud teistesse registritesse kuuluvad objektid, mida on antud juhul vaja radade funktsionaalse allsüsteemi toimimise tagamiseks. 

**Joonis 4 Laiendatud radade registri olemi-suhte diagramm.**



**Joonis 5 Klassifikaatorite registri olemi-suhte diagramm.**



**Joonis 6 Isikute registri olemi-suhte diagramm.**



**Joonis 7 Töötaja registri olemi-suhte diagramm.**



**Joonis 8 Klientide registri oli-suhte diagramm.**

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. |
| 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) |
| Rada | Radade register | Maastikule loodud alad, kus saavad kliendid suusatada. |
| Raja\_kategooria | Klassifikaatorite register | Võimaldab raja klassifitseerimist erinevatesse kategooriatesse ja selle alusel raja rühmitamist teatud põhjusel huvipakkuvateks hulkadeks. Tegemist on üksteist mittevälistavate kategooriatega, st üks ja sama rada võib kuuluda korraga mitmesse sama tüüpi kategooriasse. |
| Raja\_kategooria\_  omamine | Radade register | Näitab raja kuulumist kategooriatesse. Iga raja ja iga raja kategooria vahel võib olla maksimaalselt üks seos. |
| Raja\_kategooria\_tüüp | Klassifikaatorite register | Võimaldab rühimatada rada klassifitseerimiseks kasutatavaid kategooriaid ühise nime alla. Need nimed kirjeldavad, mis liiki klassifikatsiooniga on tegemist. |
| Raja\_raskus | Klassifikaatorite register | Raja raskus on klassifikaator, mis määrab ära, kui keeruline on rada läbida. |
| 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. |

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 | 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 | e\_meil (e\_mail, meil, meiliaadress, e-posti aadress) | Aadress, millele saab üle võrgu (ühest arvutist või tööjaamast teise) saata isikule mõeldud kirjalikke sõnumeid. Kasutatakse kasutaja tuvastamisel kasutajanimena.  **{Registreerimine on kohustuslik. Isiku 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.**  **e\_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 | eesnimi | "Lapsele pärast sündi (registreerimisel) pandav nimi, osa isikunimest. Eesnimi asetseb harilikult perekonnanime ees, harva järel (nt Ungari pruugis)." (ESTERM)  **{Vähemalt üks kahest – eesnimi või perenimi peab olema registreeritud. Eesnimi ei tohi olla tühi string või ainult tühikutest koosnev string.}** | Mart |
| Isik | perenimi (perekonna- nimi) | "Nimi, mis on isikul ühine teiste tema perekonna liikmetega" (ESTERM)  **{Vähemalt üks kahest – eesnimi või perenimi peab olema registreeritud. Perenimi ei tohi olla tühi string või ainult tühikutest koosnev string.}** | Mets |
| Isik | sünni\_kp | Isiku sünni kuupäev sünnikoha kohaliku aja järgi.  **{Registreerimine on kohustuslik. Sünni kuupäeva võimalikud väärtused on vahemikus 01. jaanuar 1900 ja 31. detsember 2100 (otspunktid kaasa arvatud). Sünni kuupäev ei tohi olla suurem isiku registreerimise ajast}** | 12.08.1993 |
| Isik | reg\_aeg | Isiku registreerimise aeg kuupäeva ja kellaaja täpsusega. Selle võib süsteem ise automaatselt määrata.  **{Registreerimine on kohustuslik. Väärtus peab olema vahemikus 01. jaanuar 2010 00:00:00 ja 31. detsember 2100 kell 23:59:59 (otspunktid kaasa arvatud)}** | 12.08.2014 17:01:05 |
| Isik | elukoht | Isiku alalise elukoha aadress.  "Koha-aadress on territooriumi haldusjaotuse hierarhiast ja ametlikest kohanimedest lähtuv aadressobjekti tekstilis-numbriline kirje või tunnus. Ühele objektile võib määrata mitu koha-aadressi. Ühele objektile määratud koha-aadressid on paralleelaadressid." ("Aadressandmete süsteemi kehtestamine")  Näide: Tallinn, 34124, Ehitajate tee 62-12. Harjumaa, Viimsi vald, Kaku küla, Laane talu.  **{Elukoht ei tohi olla tühi string, ainult tühikutest koosnev string või ainult numbritest koosnev string.}** | Tallinn, Pikk tn. 12 |
| 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 raja\_kategooria nimetus, mis peab olema unikaalne kombinatsioonis raja\_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.}** | AktiivneR |
| Klient | on\_nous\_tylitamisega | Kas klient on nõus oma tarbijaharjumuste uurimisega või otseturundusega (TRUE) või mitte (FALSE)  **{Registreerimine on kohustuslik. Vaikimisi väärtus FALSE}** | FALSE |
| Rada | Raja\_kood | Raja arvuline kood, mis sisestatakse inimkasutaja poolt, mitte ei genereerita süsteemi poolt.  **{Raja unikaalne identifikaator. Registreerimine on kohustuslik}** | 222 |
| Rada | reg\_aeg | Raja 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 |
| Rada | nimetus | Raja nimetuse alusel on võimalik kliendil rada tuvastada.  **{Registreerimine on kohustuslik.**  **Nimetus ei tohi olla tühi string või**  **ainult tühikutest koosnev string.}** | Astaku rada |
| Rada | pikkus | Raja pikkus on määratud meetrites.    **{Registreerimine on kohustuslik. Väärtus peab olema nullist suurem.}** | 2000 |

### Andmebaasioperatsioonide lepingud

**OP1 Registreeri rada(p\_ Raja\_kood, p\_ nimetus, p\_ pikkus, raja raskuse identifikaator, töötaja identifikaator)**

**Eeltingimused**:

* Raja\_seisundi\_liik eksemplar osl (millel on nimetus="Ootel") on

registreeritud

* Töötaja eksemplar t (millel on töötaja identifikaator) on registreeritud
* Raja raskuse eksemplar r (millel on raja raskuse identifikaator) on

registreeritud

**Järeltingimused**:

--Loo eksemplare

* Raja eksemplar o on registreeritud

--Väärtusta atribuute

* o.Raja\_kood:= p\_Raja\_kood
* o.reg\_aeg:= hetke kuupäev + kellaaeg
* o.nimetus:=p\_nimetus
* o.pikkus:=p\_nimetus

--Loo seoseid

* o ja osl seos on registreeritud
* o ja t seos on registreeritud
* o ja r seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri Rada

**OP2 Unusta rada(p\_raja\_kood)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja\_kood) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl (millel on nimetus="Ootel")

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

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

**Kasutus kasutusjuhtude poolt**: Unusta rada

**OP3 Aktiveeri rada (p\_raja\_ kood)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja \_kood) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl\_vana (nimetus="Ootel") või (nimetus="Mitteaktiivne")
* Raja\_seisundi\_liik eksemplar osl\_uus (millel on nimetus="Aktiivne") on registreeritud
* Leidub vähemalt üks Raja\_kategooria\_omamine eksemplar oko, mis on seotud o

**Järeltingimused**:

--Kustuta seoseid

* o ja osl\_vana seos on kustutatud

--Loo seoseid

* o ja osl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Aktiveeri rada

**OP4 Muuda rada mitteaktiivseks (p\_raja\_kood)**

**Eeltingimused**:

* Raja eksemplar o(millel on Raja\_kood=p\_raja\_kood) on aktiivne
* Raja\_seisundi\_liik eksemplar osl\_uus (millel on nimetus="Mitteaktiivne") on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl\_vana (nimetus="Aktiivne")
* Leidub vähemalt üks raja\_kategooria\_omamine eksemplar oko, mis on seotud o

**Järeltingimused**:

--Kustuta seoseid

* o ja osl\_vana seos on kustutatud

--Loo seoseid

* o ja osl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda rada mitteaktiivseks

**OP5 Lõpeta rada(p\_raja\_kood)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja\_kood) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl\_vana ((millel on nimetus="Aktiivne") või (millel on nimetus="Mitteaktiivne"))
* Raja\_seisundi\_liik eksemplar osl\_uus (millel on nimetus="Lõpetatud") on registreeritud

**Järeltingimused**:

--Kustuta seoseid

* o ja osl\_vana seos on kustutatud

--Loo seoseid

* o ja osl\_uus seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Lõpeta rada

**OP6 Muuda raja andmeid (p\_raja\_kood\_vana, p\_raja\_kood\_uus, p\_nimetus, p\_pikkus, raja raskuse identifikaator)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja\_kood\_vana) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))
* Raja raskuse eksemplar r (millel on raja raskuse identifikaator) on

registreeritud

**Järeltingimused**:

--Väärtusta atribuute

* o.Raja\_kood:= p\_raja\_kood\_uus
* o.pikkus:=p\_pikkus
* o.nimetus:= p\_nimetus

--Kustuta seoseid

* o olemasolev seos raja raskusega on kostutatud

--Loo seoseid

* o ja r seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Muuda raja andmeid

**OP7 Lisa rada kategooriasse (p\_raja\_kood, Raja kategooria identifikaator)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja\_kood) on registreeritud
* Raja\_kategooria eksemplar ok (millel on Raja\_kategooria identifikaator) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))

**Järeltingimused**:

--Loo eksemplare

* Raja\_kategooria\_omamine eksemplar oko on registreeritud

--Loo seoseid

* o ja oko seos on registreeritud
* ok ja oko seos on registreeritud

**Kasutus kasutusjuhtude poolt**: Registreeri rada, Muuda raja andmeid

**OP8 Eemalda rada kategooriast (p\_raja\_kood, Raja kategooria identifikaator)**

**Eeltingimused**:

* Raja eksemplar o (millel on Raja\_kood=p\_raja\_kood) on registreeritud
* Raja\_kategooria eksemplar ok (millel on Raja\_kategooria identifikaator) on registreeritud
* o on seotud Raja\_seisundi\_liik eksemplariga osl ((millel on nimetus="Ootel") või (millel on nimetus="Mitteaktiivne"))

**Järeltingimused**:

--Kustuta eksemplare ja seoseid

* Raja\_kategooria\_omamine eksemplar oko, mis on seotud o-ga ja mis on seotud ok-ga, on koos oma seostega kustutatud

**Kasutus kasutusjuhtude poolt**: Registreeri rada, Muuda raja andmeid

### Registri põhiobjekti seisundidiagramm

Joonis 8 esitab seisundidiagrammi, mis kirjeldab registri põhiobjekti raja kõikvõimalikke elutsükleid*.*



**Joonis 8 Raja 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 |
| Riik | R |  |  |  |  |  |  |  |  |  |  | R |
| Amet | R | R |  |  |  |  |  |  |  |  |  | R |
| Isiku\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Töötaja\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Raja\_kategooria |  | R | R | R |  | R |  | R |  |  | R | R |
| Raja\_kategooria\_tüüp |  | R |  | R |  |  |  | R |  |  | R | R |
| Raja\_seisundi\_liik |  | R | R | R | R | R | R | R | R | R | R | R |
| Isik | R |  |  |  |  |  |  | R |  |  |  | R |
| Töötaja | R | R |  |  |  |  |  | R |  |  |  | R |
| Raja\_kategooria\_omamine |  | CR | D | CRD | R | R |  | R |  |  | R | CRD |
| Rada |  | C | RD | RU | RU | RU | R | R | RU | R | R | CRUD |
| Klient | R |  |  |  |  |  |  |  |  |  |  | R |
| Kliendi\_seisundi\_liik | R |  |  |  |  |  |  |  |  |  |  | R |
| Raja\_raskus |  | R |  | R | R | R |  | R | R |  |  | R |

1 – Tuvasta kasutaja

2 – Registreeri rada

3 – Unusta rada

4 – Muuda raja andmeid

5 – Aktiveeri rada

6 – Muuda rada mitteaktiivseks

7 – Vaata kõiki ootel või mitteaktiivseid radu

8 – Vaata kõiki radu

9 – Lõpeta rada

10 – Vaata radade koondaruannet

11 – Vaata aktiivseid radu

# Füüsiline disain

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

## Radade funktsionaalse allsüsteemi vajatavate registrite füüsiline disain



**Joonis 9 Isikute registri füüsilise disaini andmebaas**



**Joonis 10 Klassifikaatorite registri seisundidiagrammide füüsilise disaini andmebaas**



**Joonis 11 Klassifikaatorite registri füüsilise disaini andmebaas**



**Joonis 12 Klientide registri füüsilise disaini andmebaas**



**Joonis 13 radade registri füüsilise disaini andmebaas**



**Joonis 14 töötajate registri füüsilise disaini andmebaas**

# Realisatsioon PostgreSQLis

Selles peatükis esitatakse andmebaasi PostgreSQLis **(11)** realiseerimiseks mõeldud laused.

## Andmebaasi loomine

Andmebaasis t164640 kasutatav märkide klassifikatsioon ja märkide võrdlusreeglistik vastavad eesti keele reeglitele. Järgnev käsk tuleb käivitada shelli promptis

createdb -l et\_EE.utf8 -T template0 t164640

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

SET client\_encoding=LATIN9;

Järgnevalt esitatakse andmebaasiobjektide loomise SQL laused.

## Skeemid

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

## Domeenid

CREATE DOMAIN d\_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 VARCHAR(20) NOT NULL

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

CONSTRAINT chk\_nimetus\_ei\_ole\_tyhi\_string CHECK (VALUE <> '');

## Tabelid ja arvujada generaatorid

Arvujada generaatorid luuakse andmebaasisüsteemi poolt automaatselt tänu CREATE TABLE lausetes SERIAL notatsiooni kasutamisele ning seotakse automaatselt veergudega, mille puhul seda kasutatakse.

CREATE TABLE Tootaja\_seisundi\_liik(

tootaja\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Tootaja\_seisundi\_liik PRIMARY KEY

(tootaja\_seisundi\_liik\_kood),

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

) ;

CREATE TABLE Raja\_seisundi\_liik

(

raja\_seisundi\_liik\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Raja\_seisundi\_liik PRIMARY KEY (raja\_seisundi\_liik\_kood),

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

);

CREATE TABLE Raja\_raskus

(

raja\_raskus\_kood integer NOT NULL,

nimetus varchar(10) NOT NULL,

CONSTRAINT PK\_Raja\_raskus PRIMARY KEY (raja\_raskus\_kood),

CONSTRAINT AK\_Raja\_raskus\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Raja\_raskus\_nimetus\_ei\_koose\_tyhikutest

CHECK ((nimetus!~ '^[[:space:]]+$')),

CONSTRAINT CHK\_Raja\_raskus\_nimetus\_ei\_ole\_tyhi\_string

CHECK ((nimetus <> ''))

);

CREATE TABLE Raja\_kategooria\_tyyp

(

raja\_kategooria\_tyyp\_kood smallint NOT NULL,

nimetus d\_nimetus,

CONSTRAINT PK\_Raja\_kategooria\_tyyp PRIMARY KEY

(raja\_kategooria\_tyyp\_kood),

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

);

CREATE TABLE Raja\_kategooria

(

raja\_kategooria\_kood smallint NOT NULL,

nimetus d\_nimetus,

raja\_kategooria\_tyyp\_kood smallint NOT NULL,

CONSTRAINT PK\_Raja\_kategooria PRIMARY KEY (raja\_kategooria\_kood),

CONSTRAINT AK\_raja\_kategooria\_nimetus\_raja\_kategooria\_tyyp\_kood

UNIQUE (nimetus,raja\_kategooria\_tyyp\_kood),

CONSTRAINT FK\_Raja\_kategooria\_Raja\_kategooria\_tyyp

FOREIGN KEY(raja\_kategooria\_tyyp\_kood) REFERENCES Raja\_kategooria\_tyyp

(raja\_kategooria\_tyyp\_kood) ON UPDATE CASCADE

);

CREATE TABLE Riik

(

riik\_kood char(3) NOT NULL,

nimetus varchar(60) NOT NULL,

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

CONSTRAINT AK\_Riik\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Riik\_nimetus\_ei\_koosne\_tyhikutest

CHECK ((nimetus !~ '^[[:space:]]+$')),

CONSTRAINT CHK\_Riik\_nimetus\_ei\_ole\_tyhi\_string CHECK ((nimetus <> '')),

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

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

);

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 Amet

(

amet\_kood varchar(5) NOT NULL,

nimetus varchar(20) NOT NULL,

kirjeldus text NULL,

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

CONSTRAINT AK\_Amet\_nimetus UNIQUE (nimetus),

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

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

CONSTRAINT CHK\_Amet\_kirjeldus\_ei\_ole\_tyhi\_string CHECK ((kirjeldus <> ''))

) WITH (fillfactor=80);

CREATE TABLE Isik

(

isik\_id serial NOT NULL,

isikukood varchar(50) NOT NULL,

isikukoodi\_riik char(3) NOT NULL,

e\_meil varchar(254) NOT NULL,

parool varchar(60) NOT NULL,

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

synni\_kp date NOT NULL,

reg\_aeg timestamp without time zone NOT NULL DEFAULT localtimestamp(0),

eesnimi varchar(750) NULL,

perenimi varchar(810) NULL,

elukoht varchar(1000) NULL,

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

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

CONSTRAINT AK\_Isik\_Isikukood\_riik\_kood UNIQUE (isikukood,isikukoodi\_riik),

CONSTRAINT CHK\_Isik\_isikukood\_oige\_struktuur

CHECK (isikukood ~ '^([:digit:]\*[:alnum:]\*[:space:]\*[-]\*[/]\*)+')),

CONSTRAINT CHK\_Isik\_email\_oige\_struktuur

CHECK ((e\_meil ~ '^[a-z0-9.\_%-]+@[a-z0-9.-]+[.][a-z]+$')),

CONSTRAINT CHK\_Isik\_eesnimi\_perenimi\_on\_olemas

CHECK ((eesnimi IS NOT NULL) OR (perenimi IS NOT NULL)),

CONSTRAINT CHK\_Isik\_synni\_kp\_ajavahemik

CHECK (synni\_kp BETWEEN '1900-01-01' AND '2100-12-31'),

CONSTRAINT CHK\_Isik\_reg\_aeg\_ajavahemik

CHECK (reg\_aeg BETWEEN '2010-01-01' AND '2100-12-31 23:59:59'),

CONSTRAINT CHK\_Isik\_elukoht\_ei\_koosne\_tyhikutest

CHECK ((elukoht !~ '^[[:space:]]+$')),

CONSTRAINT CHK\_Isik\_eesnimi\_ei\_koosne\_tyhikutest

CHECK (eesnimi !~ '^[[:space:]]+$'),

CONSTRAINT CHK\_Isik\_perenimi\_ei\_koosne\_tyhikutest

CHECK ((perenimi !~ '^[[:space:]]+$')),

CONSTRAINT CHK\_Isik\_isikukood\_ei\_koosne\_tyhikutest

CHECK ((isikukood ~! '^[[:space:]]+$')),

CONSTRAINT CHK\_Isik\_eesnimi\_ei\_ole\_tyhi\_string CHECK (eesnimi <> ''),

CONSTRAINT CHK\_Isik\_perenimi\_ei\_ole\_tyhi\_string CHECK (perenimi <> ''),

CONSTRAINT CHK\_Isik\_isikukood\_ei\_ole\_tyhi CHECK ((isikukood <> '')),

CONSTRAINT CHK\_Isik\_elukoht\_ei\_ole\_tyhi CHECK ((elukoht <> '')),

CONSTRAINT CHK\_Isik\_elukoht\_ei\_koosne\_numbritest

CHECK ((elukoht !~ '^[[:digit:]]+$')),

CONSTRAINT FK\_Isik\_Isiku\_seisundi\_liik FOREIGN KEY

(isiku\_seisundi\_liik\_kood) REFERENCES Isiku\_seisundi\_liik

(isiku\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE Cascade,

CONSTRAINT FK\_Isik\_Riik FOREIGN KEY (isikukoodi\_riik) REFERENCES Riik

(riik\_kood) ON UPDATE Cascade

) WITH (fillfactor=80);

CREATE TABLE Tootaja

(

isik\_id integer NOT NULL,

amet\_kood varchar(5) NOT NULL,

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

mentor integer NULL,

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

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 No Action ON UPDATE Cascade,

CONSTRAINT FK\_Tootaja\_Tootaja\_seisundi\_liik

FOREIGN KEY (tootaja\_seisundi\_liik\_kood) REFERENCES Tootaja\_seisundi\_liik

(tootaja\_seisundi\_liik\_kood) ON UPDATE Cascade,

CONSTRAINT FK\_Tootaja\_mentor FOREIGN KEY (mentor) REFERENCES

Tootaja (isik\_id) ON DELETE Set Null ON UPDATE No Action

) WITH (fillfactor=80);

CREATE TABLE Klient

(

isik\_id integer NOT NULL,

on\_nous\_tylitamisega boolean NOT NULL DEFAULT false,

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

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

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

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

) WITH (fillfactor=80);

CREATE TABLE Rada

(

raja\_kood smallint NOT NULL,

nimetus d\_nimetus,

pikkus smallint NOT NULL,

reg\_aeg d\_reg\_aeg,

registreerija\_id smallint NOT NULL,

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

raja\_raskus\_kood smallint NOT NULL,

CONSTRAINT PK\_Rada PRIMARY KEY (Raja\_kood),

CONSTRAINT AK\_Rada\_nimetus UNIQUE (nimetus),

CONSTRAINT CHK\_Rada\_pikkus\_nulli\_suurem CHECK (pikkus > 0),

CONSTRAINT FK\_Rada\_Raja\_raskus FOREIGN KEY (raja\_raskus\_kood)

REFERENCES Raja\_raskus (raja\_raskus\_kood) ON DELETE No Action ON

UPDATE No Action,

CONSTRAINT FK\_Rada\_Raja\_seisundi\_liik FOREIGN KEY

(raja\_seisundi\_liik\_kood) REFERENCES Raja\_seisundi\_liik

(raja\_seisundi\_liik\_kood) ON DELETE No Action ON UPDATE No Action,

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

(isik\_id) ON DELETE No Action ON UPDATE No Action

) WITH (fillfactor=80);

CREATE TABLE Raja\_kategooria\_omamine

(

Raja\_kood smallint NOT NULL,

Raja\_kategooria\_kood smallint NOT NULL,

CONSTRAINT PK\_Raja\_kategooria\_omamine PRIMARY KEY

(Raja\_kood,Raja\_kategooria\_kood),

CONSTRAINT FK\_Raja\_kategooria\_omamine\_Rada FOREIGN KEY (Raja\_kood)

REFERENCES Rada (Raja\_kood) ON DELETE Cascade ON UPDATE Cascade,

CONSTRAINT FK\_Raja\_kategooria\_omamine\_Raja\_kategooria FOREIGN KEY

(Raja\_kategooria\_kood) REFERENCES Raja\_kategooria (raja\_kategooria\_kood)

ON UPDATE Cascade

);

## Vaated

CREATE OR REPLACE VIEW aktiivsed\_mitteaktiivsed\_rajad WITH (security\_barrier)

AS SELECT Rada.raja\_kood, Rada.nimetus AS raja\_nimetus, Rada.pikkus, Raja\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Raja\_seisundi\_liik

INNER JOIN Rada ON Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = Rada.raja\_seisundi\_liik\_kood

WHERE Rada.raja\_seisundi\_liik\_kood In (2,3);

COMMENT ON VIEW aktiivsed\_mitteaktiivsed\_rajad IS 'Näitab aktiivsed ja mitteaktiivseid radasi (kood, nimetus, pikkus ja seisundi liik)';

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

AS SELECT Raja\_kategooria\_omamine.raja\_kood, Raja\_kategooria.nimetus || ' (' || Raja\_kategooria\_tyyp.nimetus || ')' AS kategooria

FROM Raja\_kategooria\_tyyp

INNER JOIN (Raja\_kategooria INNER JOIN Raja\_kategooria\_omamine ON Raja\_kategooria.raja\_kategooria\_kood = Raja\_kategooria\_omamine.raja\_kategooria\_kood) ON Raja\_kategooria\_tyyp.raja\_kategooria\_tyyp\_kood = Raja\_kategooria.raja\_kategooria\_tyyp\_kood;

COMMENT ON VIEW radade\_kategooria\_omamine IS 'Näitab rajade kategooriad (kood ja rada kategooria)';

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

AS SELECT Rada.raja\_kood, Rada.nimetus AS koik\_rajad\_nimetus, Rada.pikkus, Raja\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Raja\_seisundi\_liik

INNER JOIN Rada ON Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = Rada.raja\_seisundi\_liik\_kood;

COMMENT ON VIEW koik\_rajad IS 'Näitab kõike radasi (kood, nimetus, pikkus ja raja hetke seisundi';

CREATE OR REPLACE VIEW radade\_koondaruanne WITH (security\_barrier)

AS SELECT Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood, upper(Raja\_seisundi\_liik.nimetus) AS seisundi\_nimetus, Count(Rada.raja\_kood) AS arv

FROM Raja\_seisundi\_liik

LEFT JOIN Rada ON Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = Rada.raja\_seisundi\_liik\_kood

GROUP BY Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood, upper(Raja\_seisundi\_liik.nimetus)

ORDER BY Count(Rada.raja\_kood) DESC , upper(Raja\_seisundi\_liik.nimetus);

COMMENT ON VIEW radade\_koondaruanne IS 'Näitab radade koondaruanne nende seisundi liigi põhjal (seisundi kood, seisundi nimetus ja arv, mis näitab, kui palju on niisugusi radasi';

CREATE OR REPLACE VIEW radade\_detailid WITH (security\_barrier)

AS SELECT Rada.raja\_kood, Rada.nimetus AS radade\_detailide\_nimetus, Rada.pikkus, Raja\_seisundi\_liik.nimetus AS hetke\_seisund, Isik.e\_meil AS registreerija\_email, (Trim(COALESCE(Isik.eesnimi, '') || ' ' || COALESCE(Isik.perenimi, ''))) AS registreerija\_nimi, Rada.reg\_aeg

FROM Raja\_seisundi\_liik

INNER JOIN (Isik INNER JOIN Rada ON Isik.isik\_id=Rada.registreerija\_id) ON Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = Rada.raja\_seisundi\_liik\_kood;

COMMENT ON VIEW radade\_detailid IS 'Näitab kõike radasi detailid (kood, nimetus, pikkus, hetke seisund, registreerija email, registreerija täis nimi, registratsiooni aeg)';

## Protseduursed keeled

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

## Trigeri funktsioonid ja trigerid

CREATE OR REPLACE FUNCTION f\_lisa\_rada\_viga() RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION 'Ei saa lisada rada, mille seisundi liik ei ole ootel.';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_lisa\_rada\_viga() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei

saa lisada rada, mille seisundi liik ei ole ootel';

CREATE TRIGGER trig\_lisa\_rada\_viga BEFORE INSERT

ON Rada FOR EACH ROW WHEN (NEW.raja\_seisundi\_liik\_kood <> 1) EXECUTE

FUNCTION f\_lisa\_rada\_viga();

CREATE TRIGGER trig\_unusta\_rada\_viga BEFORE DELETE

ON Rada FOR EACH ROW WHEN (OLD.raja\_seisundi\_liik\_kood <> 1) EXECUTE

FUNCTION f\_unusta\_rada\_viga();

CREATE OR REPLACE FUNCTION f\_unusta\_rada\_viga() RETURNS trigger AS $$

BEGIN

RAISE EXCEPTION 'Ei saa unustada rada, mille seisundi liik ei ole ootel.';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

COMMENT ON FUNCTION f\_unusta\_rada\_viga() IS 'See trigeri funktsioon aitab jõustada ärireegli: Ei

saa unustada rada, mille seisundi liik ei ole ootel';

## Reeglid

Reegleid käesolevas projektis ei looda.

## Funktsioonid

Tuleb luua vähemalt kolm funktsiooni, mis pole seotud trigeritega ja

mille poole rakendusest pöördutakse.

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 AS i INNER JOIN tootaja AS t ON i.isik\_id = t.isik\_id WHERE

Upper(e\_meil)=Upper(p\_kasutajanimi) AND amet\_kood = 'juh' AND tootaja\_seisundi\_liik\_kood IN (2, 3);

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 ja p\_parool oodatav väärtus on

tõstutundlik avatekstiline parool. Juhatajal on õigus

süsteemi siseneda, vaid siis kui tema seisundiks on tööl

või haiguslehel.';

CREATE OR REPLACE FUNCTION f\_lisa\_rada(p\_rada\_kood Rada.raja\_kood%TYPE, p\_nimetus Rada.nimetus%TYPE, p\_pikkus Rada.pikkus%TYPE, p\_registreerija\_email Isik.e\_meil%TYPE, p\_raskuse\_nimetus Raja\_raskus.nimetus%TYPE)

RETURNS Rada.raja\_kood%TYPE AS $$

INSERT INTO rada(raja\_kood, nimetus, pikkus, registreerija\_id, raja\_raskus\_kood)

SELECT p\_rada\_kood, p\_nimetus, p\_pikkus, isik.isik\_id, raja\_raskus.raja\_raskus\_kood

FROM isik, raja\_raskus WHERE LOWER(e\_meil) = LOWER(p\_registreerija\_email) AND LOWER(nimetus) = LOWER(p\_raskuse\_nimetus)

ON CONFLICT DO NOTHING

RETURNING raja\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_lisa\_rada(p\_rada\_kood Rada.raja\_kood%TYPE, p\_nimetus Rada.nimetus%TYPE, p\_pikkus Rada.pikkus%TYPE, p\_registreerija\_email Isik.e\_meil%TYPE, p\_raskuse\_nimetus Raja\_raskus.nimetus%TYPE)

IS 'Selle funktsiooni abil registreeritaksa uus rada. See funktsioon realiseerib andmebaasioperatsiooni OP1. Parameetri p\_rada\_kood oodatav väärtus on raja identifikaator, p\_nimetus oodatav väärtus on raja nimetus, p\_pikkus oodatav väärtus on raja pikkus meetrides, p\_registreerija\_email oodatav väärtus on isiku e-mail, p\_raskuse\_nimetus oodatav väärtus on raja raskuse nimetus.';

CREATE OR REPLACE FUNCTION f\_unusta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

RETURNS VOID AS $$

DELETE FROM Rada WHERE raja\_kood = p\_rada\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_unusta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

IS 'Selle funktsiooni abil unustatake (kustutakse) rada. See funktsioon realiseerib andmebaasioperatsiooni OP2. Parameetri p\_rada\_kood oodatav väärtus on raja identifikaator.';

CREATE OR REPLACE FUNCTION f\_aktiveeri\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

RETURNS Rada.raja\_kood%TYPE AS $$

UPDATE Rada SET raja\_seisundi\_liik\_kood = 2 WHERE raja\_kood = p\_rada\_kood AND raja\_seisundi\_liik\_kood = 1

RETURNING raja\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_aktiveeri\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

IS 'Selle funktsiooni abil unustatake (kustutakse) rada. See funktsioon realiseerib andmebaasioperatsiooni OP2. Parameetri p\_rada\_kood oodatav väärtus on raja identifikaator.';

CREATE OR REPLACE FUNCTION f\_lopeta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

RETURNS Rada.raja\_kood%TYPE AS $$

UPDATE Rada SET raja\_seisundi\_liik\_kood = 4 WHERE raja\_kood = p\_rada\_kood AND raja\_seisundi\_liik\_kood IN (2,3)

RETURNING raja\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_lopeta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE)

IS 'Selle funktsiooni abil lõpetatakse rada. See funktsioon realiseerib andmebaasioperatsiooni OP4. Parameetri p\_rada\_kood oodatav väärtus on raja identifikaator.';

CREATE OR REPLACE FUNCTION f\_muuda\_rada(p\_vana\_raja\_kood Rada.raja\_kood%TYPE, p\_uus\_raja\_kood Rada.raja\_kood%TYPE, p\_rada\_nimetus Rada.nimetus%TYPE, p\_rada\_pikkus Rada.pikkus%TYPE)

RETURNS Rada.raja\_kood%TYPE AS $$

UPDATE Rada SET raja\_kood = p\_uus\_raja\_kood, nimetus = p\_rada\_nimetus, pikkus = p\_rada\_pikkus WHERE raja\_kood = p\_vana\_raja\_kood AND raja\_seisundi\_liik\_kood IN (1,3)

RETURNING raja\_kood;

$$ LANGUAGE sql SECURITY DEFINER

SET search\_path = public, pg\_temp;

COMMENT ON FUNCTION f\_muuda\_rada(p\_vana\_raja\_kood Rada.raja\_kood%TYPE, p\_uus\_raja\_kood Rada.raja\_kood%TYPE, p\_rada\_nimetus Rada.nimetus%TYPE, p\_rada\_pikkus Rada.pikkus%TYPE)

IS 'Selle funktsiooni abil muudetakse raja andmed. See funktsioon realiseerib andmebaasioperatsiooni OP6. Parameetri p\_vana\_raja\_kood oodatav väärtus on muutmise raja identifikaator, p\_uus\_raja\_kood oodatav väärtus on uus raja identifikaator, p\_rada\_nimetus oodatav väärtus on uus raja nimetus, p\_rada\_pikkus oodatav väärtus on uus raja pikkus.';

## 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\_Raja\_kategooria\_Raja\_kategooria\_tyyp ON Raja\_kategooria (raja\_kategooria\_tyyp\_kood ASC);

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

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

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

CREATE INDEX IXFK\_Tootaja\_Isik ON Tootaja (isik\_id ASC);

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

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

CREATE INDEX IXFK\_Rada\_Raja\_raskus ON Rada (raja\_raskus\_kood ASC);

CREATE INDEX IXFK\_Rada\_Raja\_seisundi\_liik ON Rada (raja\_seisundi\_liik\_kood ASC);

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

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

CREATE INDEX IXFK\_Raja\_kategooria\_omamine\_Rada ON Raja\_kategooria\_omamine (Raja\_kood ASC);

CREATE INDEX IXFK\_Raja\_kategooria\_omamine\_Raja\_kategooria ON Raja\_kategooria\_omamine (Raja\_kategooria\_kood ASC);

### Täiendavad sekundaarsed indeksid

ALTER TABLE isik DROP CONSTRAINT ak\_isik\_e\_meil;

CREATE UNIQUE INDEX ak\_isik\_e\_meil ON isik (Upper(e\_meil));

### Funktsioonil põhinevad indeksid

## Klassifikaatorite väärtustamise SQL laused

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('juh','juhataja','Antud töötaja peamisteks tegevusteks on suusabaasi radade ülevaatamine');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('r\_hal','Radade haldur','Peamisteks tegevusteks on radade hooldamine');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('juhab','Juhiabi','Aitab juhti igapäeva tegemistes');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('varha','Varade haldur','Vastutab firma materjaalsete väärtuste korrasolu üle');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('klaha','Klassifikaatorite haldur','Kontrollib, et kõik klassifikaatorid oleksid korrektsed');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('klete','Klienditeenidaja','Suhtleb klientidega');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('tootj','Töötaja','Teeb tööd');

INSERT INTO amet (amet\_kood, nimetus, kirjeldus) VALUES ('perju','Personalijuht','Vastutab kõikide töötajate heaolu eest ');

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 raja\_seisundi\_liik (raja\_seisundi\_liik\_kood, nimetus) VALUES (1,'Ootel');

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

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

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

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,'Tööl');

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');

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

INSERT INTO raja\_kategooria\_tyyp (raja\_kategooria\_tyyp\_kood, nimetus) VALUES (1, 'Raja profiil');

INSERT INTO raja\_kategooria\_tyyp (raja\_kategooria\_tyyp\_kood, nimetus) VALUES (2, 'Sihtrühm');

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (1,'Tasane', 1);

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (2,'Keskmine', 1);

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (3,'Mägine', 1);

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (4,'Noored', 2);

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (5,'Täiskasvanud', 2);

INSERT INTO raja\_kategooria (raja\_kategooria\_kood, nimetus,raja\_kategooria\_tyyp\_kood) VALUES (6,'Professionaalid', 2);

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (1,'Lihtne');

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (2,'Mõõdukas');

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (3,'Keskmine');

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (4,'Raskema poolne');

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (5,'Raske');

INSERT INTO raja\_raskus (raja\_raskus\_kood, nimetus) VALUES (6,'Professionaalidele');

## JSON formaadis lähteandmete laadimine

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

CREATE EXTENSION IF NOT EXISTS postgres\_fdw;

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

CREATE USER MAPPING FOR t164640 SERVER

minu\_testandmete\_server\_apex OPTIONS (user 't164640', password

'ro6raloko');

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;

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

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

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

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

synni\_kp::date, isiku\_seisundi\_liik\_kood::smallint, parool,

elukoht

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

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

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

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

jsonb\_array\_elements(isik->'isikud')->>'email' AS e\_mail,

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

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

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

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

FROM isik\_jsonb) AS lahteandmed

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

## Täiendavate testandmete lisamine

Kõigis tabelites peavad olema testandmed (vähemalt üks rida).

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (16,1,true);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (18,1,true);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (19,1,true);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (20,1,true);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (21,1,false);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (22,2,true);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (23,1, false);

INSERT INTO klient (isik\_id,kliendi\_seisundi\_liik\_kood, on\_nous\_tylitamisega) VALUES (24,2,true);

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

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (2,3);

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (2,6);

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (3,1);

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (3,4);

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (4,1);

INSERT INTO raja\_kategooria\_omamine (raja\_kood, raja\_kategooria\_kood) VALUES (4,4);

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

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

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (8, 'varha', 2, null);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (12, 'perju', 2, 1);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (9, 'tootj', 2, 12);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (10, 'tootj', 2, 12);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (11, 'klete', 2, 12);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (13, 'klaha', 2, 1);

INSERT INTO tootaja(isik\_id, amet\_kood, tootaja\_seisundi\_liik\_kood, mentor) VALUES (14, 'juhab', 3, 1);

INSERT INTO rada (raja\_kood, nimetus, pikkus, registreerija\_id, raja\_seisundi\_liik\_kood, raja\_raskus\_kood) VALUES (1,'Astaku rada', 2000, 1,1,2);

INSERT INTO rada (raja\_kood, nimetus, pikkus, registreerija\_id, raja\_seisundi\_liik\_kood, raja\_raskus\_kood) VALUES (2,Põllu rada', 1200,1,2,1);

INSERT INTO rada (raja\_kood, nimetus, pikkus, registreerija\_id, raja\_seisundi\_liik\_kood, raja\_raskus\_kood) VALUES (3,Lisa rada', 3333, 1,1,5);

INSERT INTO rada (raja\_kood, nimetus, pikkus, registreerija\_id, raja\_seisundi\_liik\_kood, raja\_raskus\_kood) VALUES (4,Nõlva rada', 2220, 1,2,2);

## Andmebaasi statistika kogumine

## Päringu täitmisplaani näide

Analüüsin päringut, mis on tehtud vaade Aktiivsed\_mitteaktiivsed\_rajad põhjal. Joonis 17 esitab selle analüüsi tulemuse graafilisel kujul.

CREATE OR REPLACE VIEW aktiivsed\_mitteaktiivsed\_rajad WITH (security\_barrier)

AS SELECT Rada.raja\_kood, Rada.nimetus AS raja\_nimetus, Rada.pikkus, Raja\_seisundi\_liik.nimetus AS hetke\_seisund

FROM Raja\_seisundi\_liik

INNER JOIN Rada ON Raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = Rada.raja\_seisundi\_liik\_kood

WHERE Rada.raja\_seisundi\_liik\_kood In (2,3);

t164640=# analyze;

ANALYZE

t164640=# explain select \* from aktiivsed\_mitteaktiivsed\_rajad;

QUERY PLAN

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

1 Hash Join (cost=1.07..2.16 rows=2 width=25)

2 Hash Cond: (raja\_seisundi\_liik.raja\_seisundi\_liik\_kood = rada.raja\_seisundi\_liik\_kood)

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

4 -> Hash (cost=1.05..1.05 rows=2 width=17)

5 -> Seq Scan on rada (cost=0.00..1.05 rows=2 width=17)

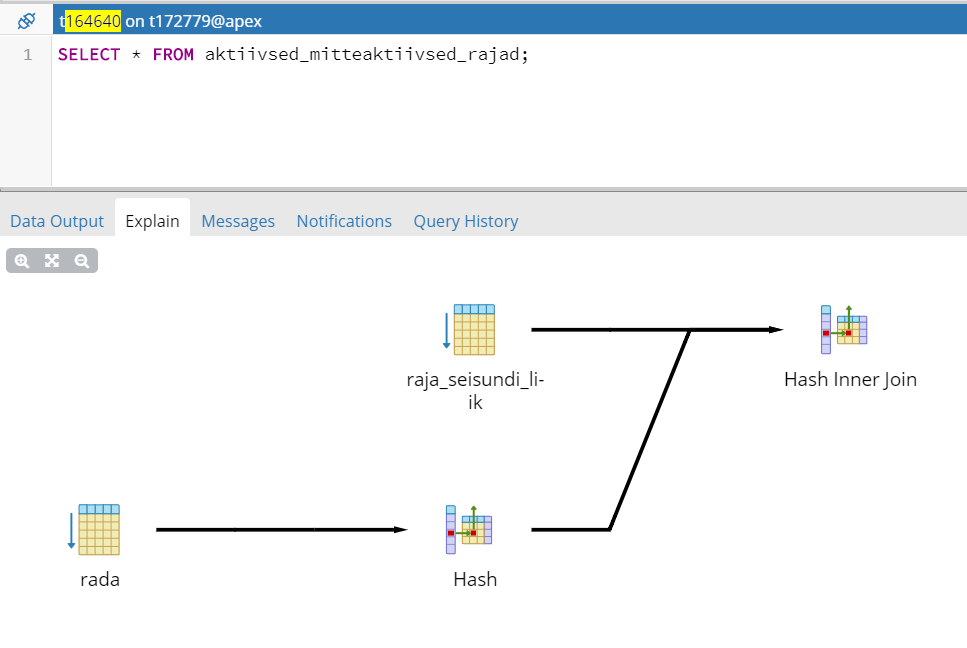
6 Filter: (raja\_seisundi\_liik\_kood = ANY ('{2,3}'::integer[]))

(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.

5-6 – Tabeli *Rada* läbiskaneerimisel leitakse vaid read, mis rahuldavad tingimust raja\_seisundi\_liik\_kood = ANY ('{2,3}'::integer[]) – võrreldatakse raja seisudi liike. Tabelite *Rada* ja *Raja\_seisundi\_liik* ühendamise operatsiooni sisendiks lähevad vaid leitud read tabelist *Rada*.

1-4 – Tabelid *Raja\_seisundi\_liik* ja *Ruum* ühendatakse kasutades hash join algoritmi. Tabelist *Raja\_seisundi\_liik* loetud *raja\_seisundi\_liik\_kood* väärtuste alusel leitakse räsiväärtused ja moodustatakse mällu ajutine räsitabel. Räsitabelis olevaid väärtuseid hakatakse võrdlema tabelist *Rada* loetud *raja\_seisundi\_liik\_kood* väärtuste põhjal leitud räsiväärtustega. Selle ühendamise kontekstis on *Raja\_seisundi\_liik* väline tabel. Nii *raja\_seisundi\_liik\_kood* väärtused tabelist *Raja\_seisundi\_liik* kui ka *raja\_seisundi\_liik\_kood* väärtused tabelist *Rada* leitakse tabelite täielikku läbiskaneerimist kasutades.



Joonis 17. Täitmisplaani vaadatuna visuaalselt PgAdmin’is.

## Rollid ja kasutajad

CREATE USER t164640\_suusaklubi\_juhataja WITH ENCRYPTED PASSWORD 'rajad1';

## Üleliigsete õiguste äravõtmine

REVOKE ALL ON DATABASE t164640 FROM t164640\_suusaklubi\_juhataja;

REVOKE ALL ON SCHEMA public FROM t164640\_suusaklubi\_juhataja;

## Õiguste jagamine

GRANT CONNECT ON DATABASE t164640 TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_lisa\_rada(p\_rada\_kood Rada.raja\_kood%TYPE, p\_nimetus Rada.nimetus%TYPE, p\_pikkus Rada.pikkus%TYPE, p\_registreerija\_email Isik.e\_meil%TYPE, p\_raskuse\_nimetus Raja\_raskus.nimetus%TYPE) TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_lisa\_rada\_viga() TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_unusta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE) TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_unusta\_rada\_viga() TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_aktiveeri\_rada(p\_rada\_kood Rada.raja\_kood%TYPE) TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_lopeta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE) TO t164640\_suusaklubi\_juhataja;

GRANT EXECUTE ON FUNCTION f\_muuda\_rada(p\_vana\_raja\_kood Rada.raja\_kood%TYPE, p\_uus\_raja\_kood Rada.raja\_kood%TYPE, p\_rada\_nimetus Rada.nimetus%TYPE, p\_rada\_pikkus Rada.pikkus%TYPE) TO t164640\_suusaklubi\_juhataja;

GRANT SELECT ON aktiivsed\_mitteaktiivsed\_rajad TO t164640\_suusaklubi\_juhataja;

GRANT SELECT ON koik\_rajad TO t164640\_suusaklubi\_juhataja;

GRANT SELECT ON rada\_kategooria\_omamine TO t164640\_suusaklubi\_juhataja;

GRANT SELECT ON radade\_detailid TO t164640\_suusaklubi\_juhataja;

GRANT SELECT ON radade\_koondaruanne TO t164640\_suusaklubi\_juhataja;

GRANT USAGE ON SCHEMA public TO t164640\_suusaklubi\_juhataja;

## Andmebaasiobjektide kustutamine

Järgnevalt esitatakse andmebaasiobjektide kustutamise laused.

### Õiguste äravõtmine

REVOKE ALL ON DATABASE t164640 FROM t164640\_suusaklubi\_juhataja;

REVOKE ALL ON SCHEMA public FROM t164640\_suusaklubi\_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 Raja\_kategooria\_omamine CASCADE;

DROP TABLE IF EXISTS Rada CASCADE;

DROP TABLE IF EXISTS Klient CASCADE;

DROP TABLE IF EXISTS Tootaja CASCADE;

DROP TABLE IF EXISTS Isik CASCADE;

DROP TABLE IF EXISTS Amet CASCADE;

DROP TABLE IF EXISTS Isiku\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Kliendi\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Riik CASCADE;

DROP TABLE IF EXISTS Raja\_kategooria CASCADE;

DROP TABLE IF EXISTS Raja\_kategooria\_tyyp CASCADE;

DROP TABLE IF EXISTS Raja\_raskus CASCADE;

DROP TABLE IF EXISTS Raja\_seisundi\_liik CASCADE;

DROP TABLE IF EXISTS Tootaja\_seisundi\_liik CASCADE;

### Vaadete kustutamine

DROP VIEW IF EXISTS aktiivsed\_mitteaktiivsed\_rajad;

DROP VIEW IF EXISTS koik\_rajad;

DROP VIEW IF EXISTS rada\_kategooria\_omamine;

DROP VIEW IF EXISTS radade\_detailid;

DROP VIEW IF EXISTS radade\_koondaruanne;

### Indeksite kustutamine

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

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

DROP INDEX IF EXISTS IXFK\_Isik\_Riik;

DROP INDEX IF EXISTS IXFK\_Tootaja\_Amet;

DROP INDEX IF EXISTS IXFK\_Tootaja\_Isik;

DROP INDEX IF EXISTS IXFK\_Tootaja\_mentor;

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

DROP INDEX IF EXISTS IXFK\_Rada\_Raja\_raskus;

DROP INDEX IF EXISTS IXFK\_Rada\_Raja\_seisundi\_liik;

DROP INDEX IF EXISTS IXFK\_Rada\_Tootaja;

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

DROP INDEX IF EXISTS IXFK\_Raja\_kategooria\_omamine\_Rada;

DROP INDEX IF EXISTS IXFK\_Raja\_kategooria\_omamine\_Raja\_kategooria;

### Funktsioonide ja trigerite kustutamine

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

DROP FUNCTION IF EXISTS f\_lisa\_rada(p\_rada\_kood Rada.raja\_kood%TYPE, p\_nimetus Rada.nimetus%TYPE, p\_pikkus Rada.pikkus%TYPE, p\_registreerija\_email Isik.e\_meil%TYPE, p\_raskuse\_nimetus Raja\_raskus.nimetus%TYPE);

DROP FUNCTION IF EXISTS f\_unusta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE);

DROP FUNCTION IF EXISTS f\_aktiveeri\_rada(p\_rada\_kood Rada.raja\_kood%TYPE);

DROP FUNCTION IF EXISTS f\_lopeta\_rada(p\_rada\_kood Rada.raja\_kood%TYPE);

DROP FUNCTION IF EXISTS f\_muuda\_rada(p\_vana\_raja\_kood Rada.raja\_kood%TYPE, p\_uus\_raja\_kood Rada.raja\_kood%TYPE, p\_rada\_nimetus Rada.nimetus%TYPE, p\_rada\_pikkus Rada.pikkus%TYPE);

### Reeglite kustutamine

Reegleid käesolevas projektis ei ole.

### Kasutajate ja rollide kustutamine

DROP USER IF EXISTS t164640\_suusaklubi\_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. Andmebaaside projekti tegemise mall. [WWW] <http://maurus.ttu.ee> (10.02.2018)
4. Country Codes - ISO 3166 [WWW]   
   <https://www.iso.org/iso-3166-country-codes.html> (10.02.2018)
5. Chisholm, M. (2000). *Managing Reference Data in Enterprise Databases:* Binding *Corporate Data to the Wider World.* Morgan Kaufmann.
6. Eesti Statistika. Riikide ja territooriumide klassifikaator 2013v1. [WWW] <http://metaweb.stat.ee/view_xml_multi_code.htm?id=3477719&siteLanguage=ee> (10.02.2018)
7. ESTERM [WWW] <http://termin.eki.ee/esterm/> (10.02.2018)
8. Isikuandmete kaitse seadus. [WWW] <https://www.riigiteataja.ee/akt/130122010011?leiaKehtiv> (10.02.2018)
9. 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)
10. Vikipeedia. Riik. [WWW] <https://et.wikipedia.org/wiki/Riik> (10.02.2018)