

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Anni-Brit Rimmelg 164343IABB

**ESIMESED SAMMUD TAAVI
MAJANDUSTARKVARA UUELE
ANDMEBAASISÜSTEEMILE ÜLEVIIMISEL**

Bakalaureusetöö

Juhendaja: Erki Eessaar
PhD

Tallinn 2020

Autorideklaratsioon

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

Autor: Anni-Brit Rimmelg

25.05.2020

Annotatsioon

Pärandsüsteem on organisatsiooni toimimiseks vajalik tarkvara või tehnoloogia, mida võib käesoleval hetkel turul pakutavate alternatiividega võrreldes lugeda aegunuks ning mis takistab organisatsiooni infosüsteemis soovitud uuenduste ning muudatuste läbiviimist. Käesoleva töö eesmärgiks on teha esimesed ettevalmistavad sammud Taavi majandustarkvara pärandandmebaasi kaasajastamisel. Kaasajastamiseks on plaanis hetkel kasutusel olevalt DBF-failidel põhineva xBase-tüüpi andmebaasisüsteemilt üle minna SQL-andmebaasisüsteemile. Töö käigus viidi TAAVI Palk mooduli näitel läbi andmebaasi migratsiooni ettevalmistamiseks kolm sammu – olemasolevate probleemide teadvustamine, andmebaasi tükeldamine ning andmebaasi disaini parandamine ühe andmebaasi alamosa ehk registri näitel.

Töö tulemusteks on analüüs olemasoleva andmebaasi disaini ning selle platvormi põhilistest probleemidest, SQL-andmebaasi disaini parimate praktikate alusel ühe registri parandamisel saadud disaini mudel ning planeeritava migratsiooni läbiviimiseks koostatud esialgne etapiviisiline plaan. Ettevõtte seisukohast on töö tulemused sisendiks, mis aitavad otsustada, kas ning kuidas üleminek päriselus läbi viia.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 6 peatükki, 11 joonist, 6 tabelit.

Abstract

First Steps in Migrating Taavi Business Software to a New Database Management System

Legacy systems are software or technology employed by an organisation that are necessary to the functions of the organisation, but can be considered outdated compared to present-day alternatives available in the market and are standing in the way of making desired updates and changes to the information system of the organisation. The aim of this thesis is to make the first necessary steps needed in preparation for modernising the legacy database used by Taavi Business Software. The modernisation plan is to migrate from the currently used file-based xBase database management system to an SQL-based database management system. In this work, three preparatory stages for the database migration – analysing the current issues, decomposing the current database into subsystems known as registers in order to simplify its further development, and improving the design of the database in the example of a single register – were performed.

The first result of this work is an analysis of the main issues with the current platform and design of the database. The limits of the platform were analysed both from a theoretical standpoint and the business perspective of the organisation. The author used the SQL database design antipatterns (from Bill Karwin) and code smells (from Robert C. Martin) to identify a number of problems in the design of existing database. The second result is an improved design model of a register. It was created by following database design best practices such as normalization, applying the orthogonality principle, and eliminating occurrences of antipatterns and code smells. The third result is a suggested roadmap for proceeding with the migration.

The results of this thesis serve as an input for the developers of the system to decide on how it would be best to proceed with the migration process.

The thesis is in Estonian and contains 46 pages of text, 6 chapters, 11 figures, 6 tables.

Lühendite ja mõistete sõnastik

Enterprise Architect	Disaini ning modelleerimistarkvara tarkvara ning ärisüsteemide arenduse ning analüüsi jaoks.
JDBC	<i>Java Database Connectivity</i> , Java andmebaasipöördus. Java keeles kirjutatud tarkvaras töötav andmebaasipöörduse spetsifikatsioon.[1]
ODBC	<i>Open Database Connectivity</i> , avatud andmebaasipöördus. Microsofti poolt loodud standardiseeritud meetod, mis võimaldab mistahes programmis, andmebaasidraiveri vahendusel, kasutada mistahes andmebaasisüsteemis loodud andmebaasi. [1]
pärandsüsteem	Arvutisüsteem, mis on kasutusel olnud juba pikka aega.
pöördprojekteerimine	Olemasoleva tehise (näiteks tarkvara) disaini ja arhitektuuri tundmaõppimine ja esitamine mudelitena.
SQL	<i>Structured Query Language</i> on relatsioonilist andmemudelt realiseeriv andmebaasikeel, mis muuhulgas võimaldab hallata andmeid, andmebaasiobjekte, õiguseid ja andmete kasutamise tehinguid.
TAAVI majandustarkvara	Taavi Tarkvara OÜ poolt arendatud laia funktsionaalsusega tarkvara
tarkvara migreerimine	Olemasoleva tarkvara asendamist uue tarkvaraga, mis kasutab teistsugust arhitektuuri, töötab teisel platvormil või on loodud mõnes teises tarkvarakeeles/arenduskeskkonnas.
UML	<i>Unified Modeling Language</i> , unifitseeritud modelleerimiskeel. Standardiseeritud ja paljudes valdkondades kasutatav keel, mis on mõeldud visuaalsete mudelite koostamiseks.

Sisukord

1 Sissejuhatus	10
1.1 Probleem.....	10
1.2 Eesmärgid	10
1.3 Ülevaade töö struktuurist.....	11
2 Teoreetiline taust	12
2.1 Pärandüsteemid	12
2.2 DBF-failidel põhinevad andmebaasisüsteemid.....	14
2.3 Pöördprojekteerimine	16
2.4 Tarkvara migreerimine	17
2.5 Tehniline võlg.....	18
3 Olemasoleva olukorra kirjeldus.....	20
3.1 Platvormist tulenevad probleemid	21
3.2 Olemasoleva andmebaasi disainiprobleemid.....	23
3.2.1 Kehvad nimed.....	23
3.2.2 Surnud tabelid.....	24
3.2.3 Duplikaadid	24
3.2.4 Tabeli väljas on atribuudi mitu väärtust komadega eraldatuna tekststringis.	25
3.2.5 Välisvõtme kitsenduste puudumine.....	25
3.2.5 Mitmeveerulised atribuudid.....	26
3.2.6 Metaandmete puru	27
3.3 Järeldused	28
4 Olemasoleva olukorra parandamise protsess.....	29
4.1 Protsess tervikuna	29
4.2 Andmebaasi tükeldamine	32
4.3 Vii läbi andmebaasi disaini pilootprojekt.....	34
4.4 Sobivaima andmebaasisüsteemi valik Saaty meetodil	36
5 Parandatud andmebaasi disain.....	38
5.1 Tükeldus registriteks.....	38
5.2 Parandatud andmebaasi disaini mudel.....	39

5.3 Erinevate paranduste ja muudatuste kokkuvõte/analüüs	51
5.4 Esialgne tagasiside arendajatelt	52
5.5 Järgmised sammud	52
6 Kokkuvõte	55
Kasutatud allikad	57
Lisa 1 – TAAVI Palk demoversiooniga kaasatulevad failid	60
Lisa 2 – TAAVI Palk demoandmebaasi dokumentatsiooni näidised	61
Lisa 3 – TAAVI Palk demoandmebaasi tabelite jaotus registritesse.....	64

Jooniste loetelu

Joonis 1. Kuvatõmmis Google Trends graafikust, mis näitab DBF-failivorminguga seotud otsingute populaarsuse muutust ajavahemikus 2004-2020 [12].	15
Joonis 2. Osaline kuvatõmmis väljamaksete seotud tabeli VALJAMAK veergude kirjeldusest.....	26
Joonis 3. Osaline kuvatõmmis maksudeklaratsioonidega seotud tabeli TSDALUS veergude kirjeldusest.	27
Joonis 4. Tegevusdiagramm "Olemasoleva olukorra parandamise üldprotsess".....	30
Joonis 5. Tegevusdiagramm "Andmebaasi tükeldamine".....	34
Joonis 6. Tegevusdiagramm "Vii läbi andmebaasi disaini pilootprojekt".	35
Joonis 7. Tegevusdiagramm "Andmebaasisüsteemi valimine".....	36
Joonis 8. Püsitasude registri algversiooni andmebaasi diagramm.....	40
Joonis 9. Tabel pysitasu ja sellesse välisvõtme andvad tabelid.....	46
Joonis 10. Tabel pysitasu_lahendus.	47
Joonis 11. Tabelid pysitasu_analkood ja pysitasu_allikas.....	48

Tabelite loetelu

Tabel 1. Taavi majandustarkvara moodulite lühiülevaade.....	20
Tabel 2. TAAVI palk andmebaasi registrid.....	38
Tabel 3. PYSITASU.dbf tabeli esialgsete veergude nimed, kirjeldused, andmetüübid ning sisendid.....	41
Tabel 4. PYSITASU.dbf tabelis tehtud veerunimetuste muudatused.	44
Tabel 5. Parandatud disaini mudelis esitatud tabelite kirjeldused.....	49
Tabel 6. Parandatud mudeli tabeli veergude definitsioonid ning näidisväärtused.	50

1 Sissejuhatus

Levib ütlus, et tarkvaralisi lahendusi võib tänapäevase kiire tehnoloogilise arengu tõttu lugeda aegunuks juba alates sellest päevast, mil need kasutusele võetakse. Vallaste e-teatmik [1] defineerib pärandüsteemi kui pikka aega kasutusel olnud arvutisüsteemi. Ehkki puudub ametlik reegel, millisest vanusest alates on süsteemi sobilik pärandüsteemina iseloomustada, on erinevates allikates üheks olulisemaks korduvaks tunnuseks asjaolu, et kasutusel olev tehnoloogia muutub turul konkureerimiseks vajalike muudatuste ning uuenduste läbiviimisel takistuseks.

1.1 Probleem

Käesoleva töö lahendatavaks probleemiks on ettevalmistused Taavi Tarkvara majandustarkvara poolt vajatava andmebaasisüsteemi väljavahetamiseks e andmebaasi migreerimiseks, võttes aluseks selle kõige olulisema mooduli, TAAVI Palk [2]. Töös ei tehta seda üleminekut, kuid töö tulemused on oluliseks sisendiks arendajatele otsustamiseks, kas ja kuidas peaks selle protsessiga jätkama.

Hetkel kasutusel olevalt failipõhiselt xBase-tüüpi andmebaasisüsteemilt on soov üle minna SQL-põhisele relatsioonilisele andmebaasisüsteemile. Uuele süsteemile ülemineku näol on tegemist pärandüsteemi kaasajastamise ülesandega. Käesolevast tööst võib mõelda kui pilootprojektist, mis aitab hiljem otsustada, kas ning kuidas selline üleminek päriselt läbi viia.

1.2 Eesmärgid

Töö eesmärgid on järgmised.

1. Analüüsida täpsemalt ettevõtte seisukohast olemasoleva platvormi puudujääke, mille tõttu migratsiooni plaanitakse.
2. Analüüsida TAAVI Palk mooduli näitel olemasoleva andmebaasi kvaliteeti ja andmebaasi disaini tehnilise võla taset.

3. Parandada uuele SQL-põhisele platvormile ette nähtud andmebaasi disaini lähtuvalt erinevatest headest andmebaasi disaini praktikatest ühe andmebaasi alamosa ehk registri näitel.
4. Sõnastada migratsiooni realiseerimiseks vajalikud järgmised sammud.

Töös on kasutusel disaini-tegevusuuringu meetoodika [3]. See tähendab, et töö tulemuseks on tehnilised tehised (andmebaaside disaini mudelid), mida loomise käigus jooksvalt ja pidevalt tarkvara arendajatele ette näidatakse ning nende tagasiside alusel täiendatakse. Töö tulemusena tekib uut teadmist olemasoleva süsteemi ja seda liiki süsteemide migreerimise protsessi kohta.

1.3 Ülevaade töö struktuurist

Teises peatükis annab autor ülevaate temaga seotud olulistest mõistetest.

Kolmandas peatükis analüüsitakse olemasoleva andmebaasi puudujääke, mis tulenevad nii andmebaasi platvormist kui ka andmebaasi praegusest disainist.

Neljandas peatükis on autor toonud välja soovitusliku protsessi olemasoleva andmebaasi olukorra parandamiseks ning selle migratsiooniks.

Viienda peatüki sisuks on andmebaasi tükeldamise ning andmebaasi disaini parandamise tulemused. Disaini parandamiseks on valitud üks konkreetne register. Samuti sisaldab see peatükk soovitusi järgmisteks sammudeks. Lisaks vahendatakse ettevõtte arendajate tagasisidet parandatud mudeli rakendatavuse kohta tegeliku migratsiooni puhul.

Kuues peatükk võtab tehtud töö kokku.

2 Teoreetiline taust

Antud peatüki eesmärgiks on defineerida ning anda lühike ülevaade selle töö jaoks olulistest mõistetest.

2.1 Pärandsüsteemid

Pärandsüsteem (inglise keeles *legacy system*) on subjektiivne mõiste, mis tähistab organisatsioonis jätkuvalt kasutusel olevat ning väärtusliku teadmusvaraga seotud tarkvara või tehnoloogiat, mida võib võrreldes turul pakutavate alternatiividega lugeda aegunuks ning mis takistab organisatsiooni infosüsteemile vajalike muudatuste ning uuenduste läbiviimist. Sageli iseloomustavad pärandsüsteeme suur maht ning keerukus, läbiva organiseerituse ning dokumentatsiooni puudumine, piiratud võimekus liidestuda teiste toodetega, kogutud andmete ebatäpsus, teiste süsteemidega ühildamatud andmeformaadid ja süsteemi ülalhoiuks ning arendamiseks vajalike oskuste kandjate vähesus tööturul [4].

Pärandsüsteemide keerukust aitavad seletada Lehmani [5] poolt sõnastatud tarkvara evolutsiooni seadused (seaduspärasused), mille kohaselt tarkvara eluea jooksul suureneb selle keerukus, sest kasutajate rahulolu huvides tuleb kogu aeg lisada uut funktsionaalsust. Kui selle juures ei pöörata piisavalt tähelepanu tarkvara hallatavuse tagamisele, tehnilist võlga põhjustavate disainipuuduste kõrvaldamisele ning keerukuse kontrolli all hoidmisele, siis hakkab tarkvara kvaliteet langema.

Pärandsüsteemide suhtes on organisatsioonidel üldistatult kolm valikut – ülalhoid, täielik väljavahetus või kaasajastamine. Õige valiku tegemine nõuab sageli pikemat analüüsi. Piirid nende valikute vahel on hägused (mis hetkest muutuvad pisiparandused kapitaalremondiks ja remont terve süsteemi väljavahetamiseks).

Ülalhoiu all peetakse silmas süsteemile jooksvalt väikeste paranduste tegemist ilma selle struktuuri muutmata. Siia alla ei kuulu uut sorti arhitektuuri või kasutajaliidese

arendamine. Kõik infosüsteemid vajavad pidevat ülalhoidu, kuid pärandüsteemide tuleb selle valiku puhul arvestada ajapikku aina enam konkurentsivõime kaotamise ning ülalhoidmiseks sobivate oskustega töötajate leidmisega seotud lisakuludega [6].

Täielik väljavahetus tähendab pärandüsteemi asendamiseks uue süsteemi nullist (puhtalt lehelt) ehitamist. Tegu on väga palju vahendeid nõudva valikuga ning see on mõistlik valik ainult siis kui kaasajastamine ei ole enam sellega võrreldes kuluefektiivsem. Lisaks suurele arendus- ning testimistöömahule, tuleb pärandüsteemi väljavahetuse puhul eraldada lisavahendeid ka uue süsteemi jaoks vajalike oskuste õpetamisele [6]. Täielik väljavahetamine ("no legacy" põhimõte) võib olla tarkvara valdajate ja arendajate ametlik poliitika nagu kirjeldab näiteks Raidma [7], mille kohaselt lähtutakse Eestis X-tee ja riiklike infosüsteemide arendamisel põhimõttest, et iga 10–12 aasta tagant tuleb süsteem ümber teha (reorganiseerida) ja võtta selle käigus kasutusele ka uued tehnoloogiad. Niimoodi tagatakse uute tehnoloogiate kasutuselevõtt, süsteemi kaasaegsus ja turvalisus.

Kaasajastamine tähendab suuremaid muudatusi kui ülalhoiu puhul, kuid siiski olemasoleva süsteemi sisu säilitamist märkimisväärsel määral. Kaasajastamise puhul eristatakse valge kasti ning musta kasti lähenemist. Valge kasti kaasajastamine hõlmab endas pärandüsteemi sisemuse mõistmist. Juhul kui info pole vajalikul määral kättesaadav, tuleb läbida sisuliselt pöördprojekteerimise protsess. Kui arusaamine pärandüsteemi toimimisest on olemas, viiakse enamasti läbi süsteemi põhjalik sisene ümberstruktureerimine, eesmärgiga tõsta selle võimekust ning tulevast hooldatavust [6]. Käesolevat tööd saab lugeda valge kasti tüüpi kaasajastamise algusfaasi juhtumiuuringuks.

Musta kasti kaasajastamise puhul piirduakse enamasti süsteemi sisendite ja väljundite tundmaõppimisega. Sellisel juhul kasutatakse sageli lisakihi realiseerimist, mis võimaldab vana süsteemi ühendust kaasaegsemate liidestega. Tegemist on pealiskaudsema ning lihtsama lahendusega, sest sisuliselt jääb pärandüsteem siiski kasutusele [6].

Pärandüsteemide kaasajastamisega nii äriliste kui ka tehnoloogiliste parimate praktikate kohta põhjalikumaks uurimiseks sobib hästi Seacord et al. õpik "Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices" [6].

2.2 DBF-failidel põhinevad andmebaasisüsteemid

Antud töös vaatluse alla võetud infosüsteem (majandustarkvara TAAVI) kasutab oma praeguse lahendusena DBF-failidel põhinevat andmebaasisüsteemi, mida saab käsitleda klassikalise näitena pärandisüsteemist.

.dbf laiendusega struktureeritud andmeid säilitav failiformaat on olnud kasutusel 1983. aastast, mil see loodi 1980ndate ühe populaarseima mikroarvutitele mõeldud dBase-nimelise andmebaasisüsteemi tarbeks. dBase süsteemiga kaasas käiv samanimeline protseduuriline programmeerimiskeel sarnaneb BASIC keelele [8].

Töö kirjutamise ajaks ligi 40-aastase ajaloo vältel on dBase süsteem olnud aluseks paljude sarnaste, edasiarendatud ning laiema funktsionaalsusega süsteemide loomisel nagu FoxPro, Visual FoxPro, Visual Objects, Visual dBase ja Clipper. Kõik need süsteemid baseeruvad DBF-failivormingul ning kasutavad lisaks indeksfaile laiendustega nagu näiteks .ndx, .mdx, .cdx. Nende süsteemide koondnimetuseks sai xBase-tüüpi andmebaasisüsteemid [9].

xBase-tüüpi andmebaasisüsteemi puhul on tegu ISAM-tüüpi failipõhise andmebaasisüsteemiga, mida võib lugeda kaasaegsete SQL-andmebaasisüsteemide eelkäijaks. ISAM ehk *Indexed Sequential Access Method* tähistab failipõhise andmebaasi haldamise tehnikaid, mis kasutavad fikseeritud pikkusega kirjeid, indekseid ning failisüsteemi lukke [10]. Igale andmebaasi tabelile ja indeksile vastab eraldi fail. Näiteks TAAVI Palk demoversiooniga tuleb kaasa Lisa 1 esitatud kataloog.

Erinevalt kaasaegsetest SQL-andmebaasisüsteemidest (andmebaasisüsteemid, milles andmete ja andmebaasi haldamiseks on kasutusel SQL andmebaasikeel) ei võimalda ISAM-tüüpi süsteemid näiteks keskse serveri lahendust, suurt samaaegsete kasutajate arvu, veebiliidese realiseerimist, JDBC või ODBC andmeühendust ja keerukamaid andmeturvalisust tagavaid lahendusi. Üheks olulisemaks puudujäägiks on probleemid andmete terviklikkuse tagamisega. Tüüpilisel juhul asuvad andmed kliendi failiserveris ning lokaalvõrgu töökohad viivad läbi käsked samade andmete suhtes samaaegselt. Probleemid töökohapõhiste kontrollalgoritmide või võrguühendusega viivad kergesti andmete kahjustumiseni [10].

xBase-süsteemide kasutus levis väga laialdaselt 1980ndatel ning 1990ndatel aastatel erinevates suurustes organisatsioonide seas üle maailma. Eelmainitud nõrkuste tõttu on selliste süsteemide kasutajate hulgas huvi SQL-süsteemidele või ka NoSQL süsteemidele üle minna ainult kasvutrendis, samas kui huvi xBase-süsteemide eneste vastu aina langeb (vt joonis 1). Andmebaasisüsteemide populaarsuse indeksis [11] on näiteks dBase andmebaasisüsteem 2020. aasta maikuu seisuga 37. kohal, Taavi majandustarkvara poolt kasutatav SAP Advantage Database Server 85. kohal ja populaarsuse trend on mõlemal languse suunas.



Joonis 1. Kuvatõmmis Google Trends graafikust, mis näitab DBF-failivorminguga seotud otsingute populaarsuse muutust ajavahemikus 2004-2020 [12].

Turul tegutsevad mitmed tarkvara migratsioonile spetsialiseerunud konsultatsiooniettevõtted, mis pakuvad tugiteenuseid just seda tüüpi süsteemidelt SQL-andmebaasisüsteemidele ülemineku elluviimiseks – nii olemasoleva andmebaasi analüüsimisel ning ettevalmistamisel kui ka sellega seotud rakenduste teisendamisel [13].

Piiratud vahendite tingimustes on terviklik andmebaasisüsteemi migratsioon väljakutset pakkuv eesmärk, mistõttu leidub mitmeid teenusepakkujaid, mis pakuvad DBF-failidel põhinevate andmebaaside ning SQL-andmebaasisüsteemide vahel üleminekulahendusi.

Üks tuntum nendest on SAP Advantage Database Server, mis võimaldab realiseerida SQL-andmebaasisüsteemidele omast funktsionaalsust, hoides andmeid jätkuvalt ISAM-tüüpi andmebaasis [14]. Alternatiivselt pakub näiteks Poola ettevõtte OTC lahendust, et xBase-süsteemile rajatud rakendusi oleks võimalik SQL-andmebaasisüsteemile migreeritud andmebaasiga ilma rakenduse lähtekoodi muutmata edasi kasutada [15]. Seevastu Alaska Software pakutavat Xbase++ reklaamitakse kui Xbase-süsteemide kaasajastatud versiooni, mis lubab lahendusi kõigile eelmainitud ISAM-süsteemide nõrkustele, säästes arendajat oma tarkvara ümberkirjutamisest ning lubades andmebaasi kiiret ning piiranguteta migratsiooni PostgreSQL andmebaasisüsteemi [16].

2.3 Pöördprojekteerimine

Pöördprojekteerimist (inglise keeles *reverse engineering*) tehakse eesmärgiga pöördprojekteeritavat inimtekkelist tehnilist tehist paremini tundma õppida, et näiteks luua sellega sarnane või sellest parem tehis või et luua selle tehise kohta puuduolev dokumentatsioon. Pöördprojekteerimine on tarkvara kontekstis olemasoleva tarkvara disaini ja arhitektuuri tundmaõppimine ja esitamine mudelitena. Andmebaasi pöördprojekteerimise tulemuseks on selle struktuuri ja käitumise esitus mudelitena (näiteks UML klassidiagrammidena). Olemasolevad tarkvara- ja infosüsteemide modelleerimiseks mõeldud CASE vahendid pakuvad lähtekoodi ja andmebaasi pöördprojekteerimise võimalusi [17].

Andmebaaside pöördprojekteerimise korral võivad sisenditeks olla näiteks olemasoleva andmebaasi andmesõnastik (pöördprojekteerimise tarkvara loob andmebaasi mudeli, lugedes andmebaasi kirjeldust andmebaasi alamosaks olevast süsteemikataloogist), selle andmebaasi loomiseks mõeldud andmekirjelduskeele laused, selles säilitatavad näidisandmed või seda kasutava rakenduse lähtekood [17]. Andmebaaside pöördprojekteerimise viiside kohta saab detailsemalt lugeda näiteks Hainaut et al. kirjutatud vastavanimelisest alapeatükist Springer kirjastuse poolt avaldatud entsüklopeedias *Encyclopedia of Database Systems* [18].

Konkreetselt DBF-failidel põhineva pärandüsteemi andmebaasi pöördprojekteerimise kohta kirjutasid Yeh et al. [19] ulatusliku juhtumiuuringu, mis käsitles sisuliselt dokumenteerimata andmebaasi põhjal kogu süsteemi kohta olemi-suhte diagrammide loomist.

Käesoleva töö kontekstis on vaja kasutada pöördprojekteerimist, et saada aru olemasoleva andmebaasi struktuurist ja probleemidest. Peatükis 5 luuakse pöördprojekteerimist kasutades olemasoleva DBF-andmebaasi põhjal Enterprise Architect CASE vahendis andmebaasi disaini kirjeldav klassidiagramm, misjärel asutakse sellel diagrammil väljendatud disaini parandama.

2.4 Tarkvara migreerimine

Tarkvara migreerimine tähendab olemasoleva tarkvara asendamist uue tarkvaraga, mis kasutab teistsugust arhitektuuri, töötab teisel platvormil või on loodud mõnes teises tarkvarakeeles/arenduskeskkonnas [20]. Tarkvara migreerimise alamliigiks on andmebaasi migreerimine, mille sisuks andmebaasisüsteemi tarkvara asendamine.

Zhang [21] jagab andmebaasi migratsiooni tinglikult kolme suuremasse faasi – ettevalmistuse faas, andmesiirde faas ning testimise faas.

Ettevalmistuse faasis tuleb identifitseerida olemasolevas andmebaasis kasutusel olevad andmebaasiobjektid (näiteks tabelid, vaated ning protseduurid). Samuti tuleb selgeks teha andmebaasi kasutavad programmid ning milliseid andmeid nende poolt kasutatakse. Tuttavaks tuleb saada süsteemi kasutajatega ning kasutajaõigustega. Ettevalmistuse faasi käigus tuleb ühtlasi hinnata migratsiooniga kaasnevaid kulutusi ning riske ning leida selle läbiviimiseks sobilikud tööriistad [21].

Andmesiirde faas tähendab andmete ülekandmist süsteemis varem kasutusel olnud andmebaasidest ning välistest allikatest uude, loodud andmebaasi. Andmesiirdeks vajalikud programmid (sh skriptid) võib luua ise või siis kasutada mõnda valmislahendust nagu näiteks SpectralCore pakutav Full Convert [22].

Testimise faasi hulka kuulub siirdatud andmete kvaliteedi kontroll. Samuti veendutakse andmebaasi kasutavate programmide jaoks vajaminevate andmebaasiobjektide (näiteks vaadete, triggerite, protseduuride ning funktsioonide) korrektses toimimises. Kindlasti tuleb teostada ka ulatuslikud testimisprojektid erinevate lõpp-kasutajate seisukohast nii migreeritud andmebaasile kui seda kasutavatele rakendustele [21].

TAAVI tarkvara jaoks on minimaalselt vajalik andmebaasi migreerimine uuele andmebaasisüsteemile. Käesolev töö loob pinnast pärandandmebaasi migreerimiseks ning töö tegevused ja tulemused kuuluvad ettevalmistuse faasi. Ei saa välistada, et protsessi käigus ilmneb, et olemasolevat rakendustarkvara pole otstarbekas kaasajastada, vaid see tuleks puhtalt lehelt alustades uue arhitektuuriga ja uusi tehnoloogiaid kasutades realiseerida. Rakendustarkvara migreerimise üle otsustamine ei ole käesoleva töö ülesanne.

2.5 Tehniline võlg

Tehniline võlg (inglise keeles *technical debt*) on 1992. aastal programmeerija Ward Cunninghami poolt välja pakutud metafoorne mõiste, mis kõrvutab tarkvaraarendusprotsessi raha laenamisega. Lihtsad ja kiiresti tehtud, kuid korralikult läbitöötamata ning parimaid praktikaid mitte järgivad lahendused, on võrreldavad kõrgete intressidega. Sealjuures tähistab intress tulevikus süsteemi tööshoidmiseks tehtavat lisatööd. Mida rohkem tehakse tööd olemasoleva süsteemi disaini parandamisega, seda vähem intressi peab tulevikus maksma. Mida kauem süsteemis parandustega viivitatakse, seda suuremaks võlg kasvab. Mingist piirist võib võlg minna ületamatult suureks ning tekib tehnilise pankroti efekt, mis tähendab, et süsteemi muutmine on läinud nii keerukaks ning riskantseks, et mõistlikum on see maha kanda ning uuesti alustada. Teisalt võib võla võtmine olla kaalutletud valik, kui leitakse, et tulemuse esialgne kiirem valmimine on olulisem, kui selle hea hallatavuse ja edasiarendatavuse tagamine tulevikus. Tehnilise võla metafoori kasutatakse kõige sagedamini lähtekoodi kontekstis, aga seda saab kasutada ka näiteks andmebaasi disaini, tarkvara testimise, dokumentatsiooni või seotud platvormide ja seadmetega tehtud valede/ ebaoptimaalsete valikute kirjeldamiseks [23]. Tehnilise võla tunnuseks on halvad lõhnad võlaga tehises, mis osutavad mingil viisil problemaatilistele tehnilistele lahendustele.

Käosaar [24] kirjutab näiteks sellest, kuidas peaaegu kõigile võla tundemärkideks olevatele lähtekoodi halbadele lõhnadele leidub vaste ka süsteemianalüüsi mudelite puhul. Tehnilise võlaga süsteem võib käesoleval hetkel töötada nii nagu vaja (seal pole ühtegi tööd takistavat tarkvara viga ja süsteem teeb seda, mida sellel parajasti oodatakse), kuid süsteemi ülesehitus/esitus/tavad on sellised, mis muudavad sellest tulevikus arusaamise, selle õppimise, selle haldamise ja parandamise keerulisemaks.

Tehnilist võlga uurivast teadustööst saab süstemaatilise ülevaate näiteks Li et al. [25] poolt ajakirjas *Journal of Systems and Software* avaldatud artiklist, mis kaardistas ning analüüsis teemaga seotud teaduspublikatsioonide ajavahemikus 1992 kuni 2013.

Eelmainitud kaardistava uurimustöö tulemusi rakendas Paide [26] Eesti kontekstis, uurides kaheksa ettevõtte näitel tehnilise võla fenomeni Eesti info- ja kommunikatsioonitehnoloogia sektoris.

Tehnilisest võlast vabanemiseks tuleb võlaga tehist refaktoreerida. Andmebaasi refaktoreerimisest üldisemalt – see tähendab lisaks andmestruktuuride disaini kvaliteedi tõstmisele ka näiteks seal hoitavate andmete kvaliteedi, selle arhitektuurilise (kuidas andmebaas on seotud väliste programmidega) kvaliteedi, selles realiseeritud andmekasutuse loogika ja äriloogika (protseduurid, triggerid jm) kvaliteedi parandamist – on kirjutatud ulatuslikult Sadalage ning Ambleri õpikus “Refactoring Databases: Evolutionary Database Design” [27].

Antud töö kontekstis uuritakse tehnilise võla esinemist TAAVI tarkvara andmebaasis kõigepealt üldiselt, vaadeldes terve ühe mooduli andmebaasi (vt peatükk 3) ja siis uurides lähemalt selle mooduli andmebaasi ühte alamosa e registrit (peatükk 4). Tehnilise võla olemasolu mõjutab oluliselt andmebaasi migratsiooni, sest migreerimiseks tuleb kõigepealt olemasolevast andmebaasist aru saada, mida võlg raskendab ning migreerimine oleks hea võimalus üritada võlast niipalju kui võimalik vabaneda (mida rohkem võlga, seda suurem pingutus). Kui võlaks on näiteks lähteandmebaasiga seotud kontrollide puudumine või andmete dubleerimisest tulenevad vastuolud, siis raskendab see andmesiiret ja vajalikuks võib osutuda eraldi töövoog lähteandmebaasis andmete korrastamiseks ja parandamiseks, enne kui saab hakata üldse andmesiiret läbi viima.

3 Olemasoleva olukorra kirjeldus

TAAVI majandustarkvara koosneb töö kirjutamise ajal üheteistkümnest erinevast moodulist, mille hulgast kliendid saavad valida enda ettevõttele vajalikud. Kõik moodulid on loetletud tabelis 1, kus on iga mooduli kohta välja toodud selle eesmärk ning demoversiooni olemasolul sellega kaasneva andmebaasi tabelite arv.

Tabel 1. Taavi majandustarkvara moodulite lühiülevaade.

Mooduli nimi	Mooduli tabelite arv	Mooduli eesmärk
TAAVI Personal	175	Personaliarvestuse korraldus.
TAAVI Ladu	173	Hulgi- ning jaemüügi ettevõtete müügitegevuse organiseerimine.
TAAVI Palk	148	Töötasude arvestamine ning nende maksustamine.
TAAVI Finants	116	Finantsarvestuse korraldus.
TAAVI Kliendihaldus	105	Klientidega seotud protsesside haldus.
TAAVI Täitur	95	Kohtutäiturite töö lihtsustamine täitetoimikute haldamisel ja nendega seotud raamatupidamise korraldamisel.
TAAVI Laenulepingud	94	Laenu andmisega tegelevate firmade laenulepingute haldus ning nendega seotud raamatupidamise organiseerimine.
TAAVI Põhivara	42	Põhivarade arvestuse pidamine.
TAAVI Eelarve	37	Eelarve koostamiseks ning selle täitmise jälgimine.
TAAVI Tööjaarvestus	Demoversiooniga ei kaasne tabeleid	Tööajatabelite koostamine.
TAAVI Reisid	Demoversioon ei ole kättesaadav	Reisibüroode töö lihtsustamine reise haldamisel.

Kõigist moodulitest on suurima kasutajate hulgaga palgaarvestuseks mõeldud TAAVI Palk, mille demoversiooni andmebaasi edaspidi täpsemalt analüüsitakse.

Olemasoleva olukorra kirjeldamisel lähtuti ettevõtte juhiga tehtud intervjuude [28] tulemustest ja tarkvara ning andmebaasi iseseisvast uurimisest.

3.1 Platvormist tulenevad probleemid

TAAVI majandustarkvara kasutab andmebaasisüsteemina hetkel DBF-failidel põhinevat süsteemi. Ehkki spetsiifilised arendusvahendid ning keeled on ajapikku vahetunud, on tarkvara arendatud xBase perekonda kuuluvate süsteemide baasil alates selle arendamise algusest 1991. aastal.

DBF-failide halduseks kasutatakse SAP Advantage Database Server tarkvara, mis tuleb kliendi serverisse eraldi paigaldada. Enamike klientide puhul asuvad kõik tabelid kliendi enda serveris ning andmete haldamise eest vastutab klient. Kliendi erisoovil pakutakse andmemahukatele lahendustele ka programmi ning andmete majutusteenust koos kaugligipääsu lahendusega, kasutades kaugtöölaua ühendust.

Põhjuseid failipõhiselt andmebaasisüsteemilt SQL-andmebaasisüsteemile migreerumiseks on ettevõtte juhi sõnul mitmeid.

Üks peamistest uuele süsteemile ülemineku põhjustest on tööaja kokkuhoid, seda nii ettevõtte enda arendusmeeskonna kui ka kliendipoolsete andmebaasi administraatorite seisukohast. Kaasaegsetes klient-server tüüpi lahendusi võimaldavates SQL-andmebaasisüsteemides on arvukalt vaikimisi sisse ehitatud funktsionaalsusi, nagu näiteks erinevate krüpteeringute tugi või IP-aadressi põhine kaugligipääs, mille asendamiseks on hetkel vaja arendusmeeskonnal kirjutada erilahendusi või leida eraldi tarkvara, mida nende funktsionaalsuste täitmiseks on tarvis rakendusse integreerida. SQL-andmebaasisüsteemi puhul oleks oluliselt lihtsam ning kulutõhusam pakkuda klientidele ka pilvemajutusel põhinevat veebipõhist majandustarkvarateenust.

Kuna failipõhised andmebaasisüsteemid on tänapäeval laiemat kasutust kaotamas, kulub kliendi infosüsteemis nende efektiivseks kasutuselevõtuks ning haldamiseks enamasti lisa-aega. Halduse eest vastutavad töötajad tuleb viia kurssi nii kasutusel oleva haldustarkvara kui ka DBF-failisüsteemi spetsiifikaga. See toob kaasa lisanduvat ajakulu nii kliendi kui ka tarkvara arendava ettevõtte seisukohast, sest tuleb arvestada sagedase klienditoe vajadusega nii juurutamisel kui ka haldusküsimuste ning probleemide lahendamise juures. Võimalikult levinud ning standardse lahenduse soov on ka peamine põhjus, miks uue andmebaasisüsteemi valikul välistati automaatselt erinevad NoSQL süsteemid.

Samuti on oluline asjaolu hetkel kasutusel olevate arendusvahendite stagneerumine, kuna failipõhised andmebaasisüsteemid on muutumas aina vähem levinumaks. Üheks näiteks on aruannete koostamiseks, mis on üks TAAVI majandustarkvara tähtsamatest funktsionaalsustest. Selleks kasutusel olev R&R Report Writer aruandegeneraator [29] on üks vähestest DBF-formaati toetavast oma kategooria tarkvarast, kuid selle edasine arendus on peatunud. SQL-põhiseid süsteeme toetavate aruandegeneraatorite valik on märksa suurem ning neid arendatakse pidevalt edasi. Lisaks on probleemiks nendele arendusvahenditele kättesaadava Microsofti operatsioonisüsteemide toe puudumine ning nende võimekus toetada kaasaegse disainiga kasutajaliidese arendusvõimalusi.

Kliendipoolse failide majutuse juures tuleb välja tuua ka ühe DBF-süsteemidele omase, andmete tervikluse (*data integrity*) rikkumise, suurenenud riski. Praktikas ei ole seda väga palju esinenud, kuid risk on suurem kui klient-server tüüpi lahenduse juures.

Migratsiooni realiseerimine on praegusel ajahetkel ka märksa soodsam kui minevikus, eelkõige kättesaadavate vabavaraliste SQL-andmebaasisüsteemide kvaliteedi ning pakutavate funktsionaalsuste arvu tõusu tõttu. TAAVI majandustarkvara arenduse algusaegadel – 1990ndatel – olid peamised usaldusväärsed SQL-andmebaasisüsteemid kättesaadavad Eesti turu kontekstis vaid suhteliselt kõrge litsentsitasu eest.

Lisaks eelpool välja toodud DBF-failidel põhineva andmebaasisüsteemi olemuslikest piirangutest, esineb praeguses andmebaasis laialdaselt andmete liiasust, tabelite ning väljade nimetuste ebastandardsust ning enamikke teooriapeatükis väljatoodud DBF-süsteemide nõrkusi. Sellest kirjutatakse lähemalt järgmises alapeatükis.

3.2 Olemasoleva andmebaasi disainiprobleemid

Mooduli TAAVI Palk demoversiooniga kaasnevasse andmebaasi kuulub 132 unikaalset DBF-tabelit (tabeleid oli kokku 148, kuid nende hulgas oli koopiaid ning ka ajaloolistel põhjustel tekitatud tabeleid mida rakenduse kaudu tuletatud väljade sisu dokumenteerivas tabelis ei olnud – ilmselt siis rakenduse kood neid üldse ei maini). Selles jaotises kontrollitakse selle andmebaasi näitel kas ja milliseid SQL-andmebaasi loogilise disaini antimustreid e nurimustreid [30] ja koodi halbade lõhnade [31] analooge selle disaini endas kätkeb. Need ei ole otseselt tarkvara tööd takistavad vead, kuid muudavad andmebaasist arusaamise, selle õppimise, haldamise ning edasiarendamise keerulisemaks.

Karwini loogilise disaini nurimustritest *ei leitud* sellest andmebaasist järgmiseid.

- Võtme-väärtuse paaride mudeli (universaalse mudeli) kasutamine (peatükk *Entity-Attribute-Value*).
- Külgnevusnimistud (*adjacency list*) hierarhiliste andmete esitamiseks (peatükk *Naive Trees*).
- Geneerilised ning mitte-läbimõeldud primaarvõtmed (peatükk *ID required*).
- Ühe ja sama välisvõtme veeru kasutamine erinevate vanemtabelitega seoste loomiseks (*Polymorphic associations*).

Järgnevas alajaotistes kirjutatakse leitud probleemidest pikemalt. Olemasoleva andmebaasi disaini probleemide uurimiseks pöördprojekteeriti see CASE vahendi (Enterprise Architect) abil ja loodi selle kohta andmebaasi disaini mudel.

3.2.1 Kehvad nimed

Kõige esimese probleemina hakkavad silma nimede andmise probleemid, nii tabelite kui veergude puhul. Nimedel puudub läbiv stiil – kasutatakse nii inglise- kui eestikeelseid nimesid ja nii läbivate väiketähtede kui ka läbivate suurtähtedega nimesid. Kõigele sellele vastab Martini lõhnade nimistust Järjekindlusetus (*Inconsistency*) koodiga G11. Näiteks on andmebaasis tasuliikide registris samaaegselt ingliskeelse nimega tabel PAYTYPE ning eestikeelse nime lühendiga TLTAB.

Paljude nimede puhul kasutatakse minimalistlikke lühendeid või väga üldiseid nimetusi, mis muudavad tabeli või veeru sisust nime järgi arusaamise inimkasutajale väga keeruliseks (tarkvaral muidugi pole vahet). Näiteks üks tuludeklaratsiooni andmetega seostuv tabel kannab nime ITMTAB, üks põhilisi töötajate andmeid koondava tabeli nimeks on PYSI, üks oluline pangaülekande andmetega seostuva tabeli nimi oli lihtsalt OBJEKT, ühe tööaegade registrisse liigituva tabeli nimeks on TP. Sellele vastab disaini halb lõhn koodiga G20: Funktsioonide nimed peaksid ütlema, mida nad teevad (*Function Names Should Say What They Do*). Selle halva lõhnaga annab Martin kaasa soovitus – juhul kui selleks, et aru saada, mida mingi funktsioon (või antud juhul tabel või veerg) teeb (antud juhul, mis andmed selles on), tuleb hakata lähemalt uurima selle realisatsiooni või dokumentatsiooni, tuleb selle nimi paremini mõistetavaks muuta.

3.2.2 Surnud tabelid

Martini üldiste halbade lõhnade nimistu koodiga G9 Surnud koodi (*Dead Code*) mõistele vastavad andmebaaside kontekstis mittekasutusel olevad (surnud) andmebaasiobjektid – käesoleva näite puhul tabelid ja veerud. Seda disaini halba lõhna esines andmebaasis märkimisväärselt. Mitmel juhul oli mahajäetud tabelite puhul tegu sisuliselt duplikaatidega, mis olid tekitatud ajutise lahendusena näiteks spetsiifilise impordi läbiviimiseks ning jäetud kustutamata. Näiteks väljaarvestatud töötasude jaoks mõeldud põhitabeli TASUD kõrval eksisteeris andmebaasis veel neli identse sisuga tabelit RTASUD, SIMTASUD, STASUD ning TASUP, mida rakendus realselt ei kasuta.

3.2.3 Duplikaadid

Duplikaadid/dubleerimine (Martini nimistus koodiga G5) on ühed tuntumatest disaini halbade lõhnadest, mida rõhutavad oma teostes ka teised populaarsed tarkvaradisainide teemadel kirjutavad autorid nagu Dave Thomas ning Andy Hunt oma DRY-printsiibiga (akronüüm inglise keelsest väljendist *Don't Repeat Yourself*) ning Kent Beck “*Once, and only once.*” põhimõttega [31] ning ka Ainult üks kord printsiip (*Once-only principle*) kasutajatelt informatsiooni kogumisel [32].

Andmebaasis võib olla erisugust dubleerimist. Andmebaasis olev dubleerimine tähendab andmete liiasust, mis kasvatab andmemahte, nõuab andmebaasirakenduste programmeerijatelt lisatööd dubleerimisest tingitud probleemide (andmete muutmise

anomaaliate) vältimisel (tuleb tagada, et liiasus oleks kontrollitud, selle kontrollimiseks tuleb kirjutada ja testida programmikoodi) ja avab võimaluse, et andmebaasis on vaatamata kõigele registreeritud mittekooskõlalisi väiteid reaalse maailma kohta.

Dubleerimine võib tekkida sellest, kui tabelitele ei ole defineeritud kitsendusi, mis välistaks tabelisse korduvate ridade lisamise. Kasutusel olev süsteem ei paku võimalust andmebaasi tasemel kitsendusi defineerida. Primaarvõtme ja unikaalsuse kitsenduste puudumisel ei ole võimalik andmebaasipõhine mehhanism, mis hoiaks automaatselt ära tabelisse korduvate ridade (duplikaatide näide) tekke. Seetõttu on risk duplikaatide tekkeks suhteliselt suur. Korduvate ridade vältimine tuleb realiseerida väljaspool andmebaasi ennast, rakenduse funktsionaalsuse kaudu.

Dubleerimine võib tekkida ka sellest, kui andmebaasis olevad andmestruktuurid (antud juhul tabelid) ei vasta normaliseerimise reeglitele (esineb andmete liiasus ühe andmestruktuuri piires) või ortogonaalse andmebaasi disaini printsiibile (esineb andmete liiasus üle erinevate andmestruktuuride) [33].

3.2.4 Tabeli väljas on atribuudi mitu väärtust komadega eraldatuna tekststringis

Seda nurimustrit kirjeldab Karwin käsiraamatu teises peatükis pealkirjaga *Jaywalking*. Probleem seisneb selles, et kui tabeli ühte lahtrisse (välja) sisestatakse sisuliselt rohkem kui üks atribuudi väärtus, esitades need ühe tekstilise väärtuse moodustava eraldajatega eraldatud nimekirjana. Selliste andmetega veerud muudavad päringud keerukamaks, tekitavad potentsiaalseid probleeme väärtuste valideerimisel, loendamisel ning ka pikkuse määramisel. Vaadeldud andmebaasis esines seda probleemi vaid mõne üksiku veeru korral, kuid mis andmete dubleerimise tõttu esinesid siiski mitmetes tabelites. Näiteks raamatupidamislikuks arvestuseks vajalike analüütikakoodide hoiustamiseks mõeldud veerus olid kaheksas tabelis lubatud lisaks üksikule koodile ka komaga eraldatud list koodidest.

3.2.5 Välisvõtme kitsenduste puudumine

Seda nurimustrit kirjeldab Karwin käsiraamatu viiendas peatükis pealkirjaga *Keyless Entry*. Probleem seisneb selles, et tabelid on omavahel teatud veergude kaudu seotud, kuid välisvõtme kitsendusi pole defineeritud. Sellisel juhul peab terviklikkuse tagama andmete haldaja (rakendus, andmebaasi kallale otse pääsev administraator) andmete

lisamisel, uuendamisel ja kustutamisel. Olemasolevas andmebaasis esines seda nurimustrit kõikjal, sest DBF-failivorming ei võimalda mistahes kitsenduste defineerimist.

3.2.5 Mitmeveerulised atribuudid

See antimuster on kirjeldatud Karwini poolt kaheksandas peatükis pealkirjaga *Multi-column attributes*. Probleem seisneb selles, et atribuudi jaoks, millel võib olla korraga mitmeid väärtuseid, tekitatakse tabelisse lisaveerud, selle asemel, et neid põhitabelist eraldi vahetabelisse üle viia. Selle tagajärjel võib muresid tekkida nii liigsete tühjade väljadega kui ka potentsiaalselt siiski veergude puudu jäämisega, kui algselt tekitatud kogus veerge pole jätkuvalt piisav.

Vaadeldavas andmebaasis esines seda nurimustrit mõõdukalt, sageli märkuste või kommentaaride säilitamiseks (nt atribuudid nimega MEMO1, MEMO2, MEMO3), aga ka näiteks väljamaksete tabelis tasukoodide markeerimiseks, vt joonis 2.

VALJAMAK	30	TASU01	Tasukood 01
VALJAMAK	31	TASU02	Tasukood 02
VALJAMAK	32	TASU03	Tasukood 03
VALJAMAK	33	TASU04	Tasukood 04
VALJAMAK	34	TASU05	Tasukood 05
VALJAMAK	35	TASU06	Tasukood 06
VALJAMAK	36	TASU07	Tasukood 07
VALJAMAK	37	TASU08	Tasukood 08
VALJAMAK	38	TASU09	Tasukood 09
VALJAMAK	39	TASU10	Tasukood 10
VALJAMAK	40	TASU11	Tasukood 11
VALJAMAK	41	TASU12	Tasukood 12
VALJAMAK	42	TASU13	Tasukood 13
VALJAMAK	43	TASU14	Tasukood 14
VALJAMAK	44	TASU15	TSD-sse mittekuuluv tasu
VALJAMAK	45	TASU16	Tasukood 15
VALJAMAK	46	TASU17	Tasukood 16
VALJAMAK	47	TASU18	Tasukood 17
VALJAMAK	48	TASU19	Tasukood 18
VALJAMAK	49	TASU20	Tasukood 19
VALJAMAK	50	TASU21	Tasukood 20
VALJAMAK	51	TASU22	Tasukood 21
VALJAMAK	52	TASU23	Tasukood 22

Joonis 2. Osaline kuvatõmmis väljamaksete seotud tabeli VALJAMAK veergude kirjeldusest.

3.2.6 Metaandmete puru

Seda nurimustrit kirjeldab Karwin käsiraamatu üheksandas peatükis pealkirjaga *Metadata Tribbles*. Probleem seisneb selles, kui suure ning inforohke tabeli korral, mille puhul on teada info täiendavat lisandumist või muutumist, lisatakse ridade arvu minimeerimise eesmärgiga tabelisse suur hulk veerge või tekitatakse samast tabelist kloon, sageli näiteks erinevate ajaperioodide kohta. Mingist konkreetsest andmeväärtusest, mis peaks asuma tabeli lahtris, on tekitatud kas tabeli või veeru nimi. Vaadeldavas andmebaasis esines seda nurimustrit palju, eriti maksudeklaratsioonidega seotud tabelites. Näiteks tabel TSDALUS (joonis 3) koosneb 554-st veerust, mis esindavad deklaratsioonivormi iga võimalikku rida, sealjuures on teada, et vormi struktuuri uuendatakse riigi poolt regulaarselt.

TSDALUS	160	L6511	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 1
TSDALUS	161	L6512	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 2
TSDALUS	162	L6513	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 3
TSDALUS	163	L6514	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 4
TSDALUS	164	L6515	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 5
TSDALUS	165	L6516	Lisa 6 Rida 5 abitabel 4 rida 1 veerg 6
TSDALUS	166	L6521	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 1
TSDALUS	167	L6522	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 2
TSDALUS	168	L6523	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 3
TSDALUS	169	L6524	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 4
TSDALUS	170	L6525	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 5
TSDALUS	171	L6526	Lisa 6 Rida 5 abitabel 4 rida 2 veerg 6
TSDALUS	172	L6531	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 1
TSDALUS	173	L6532	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 2
TSDALUS	174	L6533	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 3
TSDALUS	175	L6534	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 4
TSDALUS	176	L6535	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 5
TSDALUS	177	L6536	Lisa 6 Rida 5 abitabel 4 rida 3 veerg 6
TSDALUS	178	L651	Lisa 6 Rida 5.1
TSDALUS	179	L652	Lisa 6 Rida 5.2
TSDALUS	180	L653	Lisa 6 Rida 5.3
TSDALUS	181	L654	Lisa 6 Rida 5.4
TSDALUS	182	L66	Lisa 6 Rida 6
TSDALUS	183	L661	Lisa 6 Rida 6.1
TSDALUS	184	L662	Lisa 6 Rida 6.2

Joonis 3. Osaline kuvatõmmis maksudeklaratsioonidega seotud tabeli TSDALUS veergude kirjeldusest.

3.3 Järeldused

Uuritud andmebaasis esineb probleeme küllaltki palju ja see ei ole üllatus, sest ka varasemad uurimistööd on näidanud probleemide esinemist olemasolevate süsteemide sh pärandisüsteemide andmebaasides.

Aibast [34] analüüsis muuhulgas kolme vabavalalise tarkvara SQL-andmebaase eeskätt kitsenduste nimetamise seisukohalt, kuid vaatles ka teiste andmebaasiobjektide nimesid. Ta leidis nimede andmis järjekindlusetust ja nimede hulgas arusaamatuid lühendeid. “LedgerSMB andmebaasis kasutati andmebaasiobjektide nimedes kõige rohkem lühendeid, nii pikemates kitsenduste nimedes nime osadena kui ka veeru ja tabeli nimedes. Kuue tabeli ja 12 veeru nimed ei selgitanud objekti sisu ning nende puhul ei olnud nimedest võimalik aru saada, mis andmeid seal täpselt hoitakse.” [34]

Jaotises 3.2.1 kirjeldatud nimetamise praktikatel oli otsene mõju ka käesolevale tööle, sest peatükis 5 kirjeldatud ühe registri disaini parandamine oli raskendatud tänu sellele, et nimede põhjal oli andmete tähendusest keeruline aru saada ning ka tabelite disaini kohta käiv dokumentatsioon ei pakkunud piisavalt lisainfot (heade nimede puudus torkab eriti silma, kui ka süsteemi toetav dokumentatsioon on ebapiisav).

Lapõnin [35] analüüsis kahte avatud lähtekoodiga tarkvara andmebaasi SQL disaini nurimustrite esinemise seisukohalt ja jõudis järeldusele, et “Mõlema andmebaasi disain kannatab liigse denormaliseerimise, paralleelsete veergude, välisvõtmete ebapiisava deklareerimise ja klassifikaatorite tabelite puudumise tõttu”. Ühes uuritud andmebaasis (XTuplePostbooks programmi oma) oli primaarvõti deklareeritud 97%-l tabelitel ja teises (OpenEMR programmi oma) 79%-l. Ka varasem Blaha [36] uuring viitab sarnastele probleemidele SQL-andmebaasides. Blaha pöördprojekteeris 35 andmebaasi, uuris nende disaini ja avastas näiteks, et 25% arendajatest ei jõusta primaarvõtmeid ning 90% andmebaasides ei olnud deklareeritud välisvõtmeid ning 10%-s, kus seda oli tehtud, ei olnud seda tehtud täielikult. Weber [37] soovib kasutada välisvõtmete puudumist SQL-andmebaasides tehnilise võla mõõdikuna ja teeb tähelepaneku, et elektrooniliste meditsiinkirjete süsteemis OSCAR on välisvõtme deklaratsioone suhteliselt palju puudu, kuigi kasutatud andmebaasisüsteem neid toetab. Käesolevas töös kasutatud andmebaasisüsteemis ei saa välisvõtmeid deklareerida.

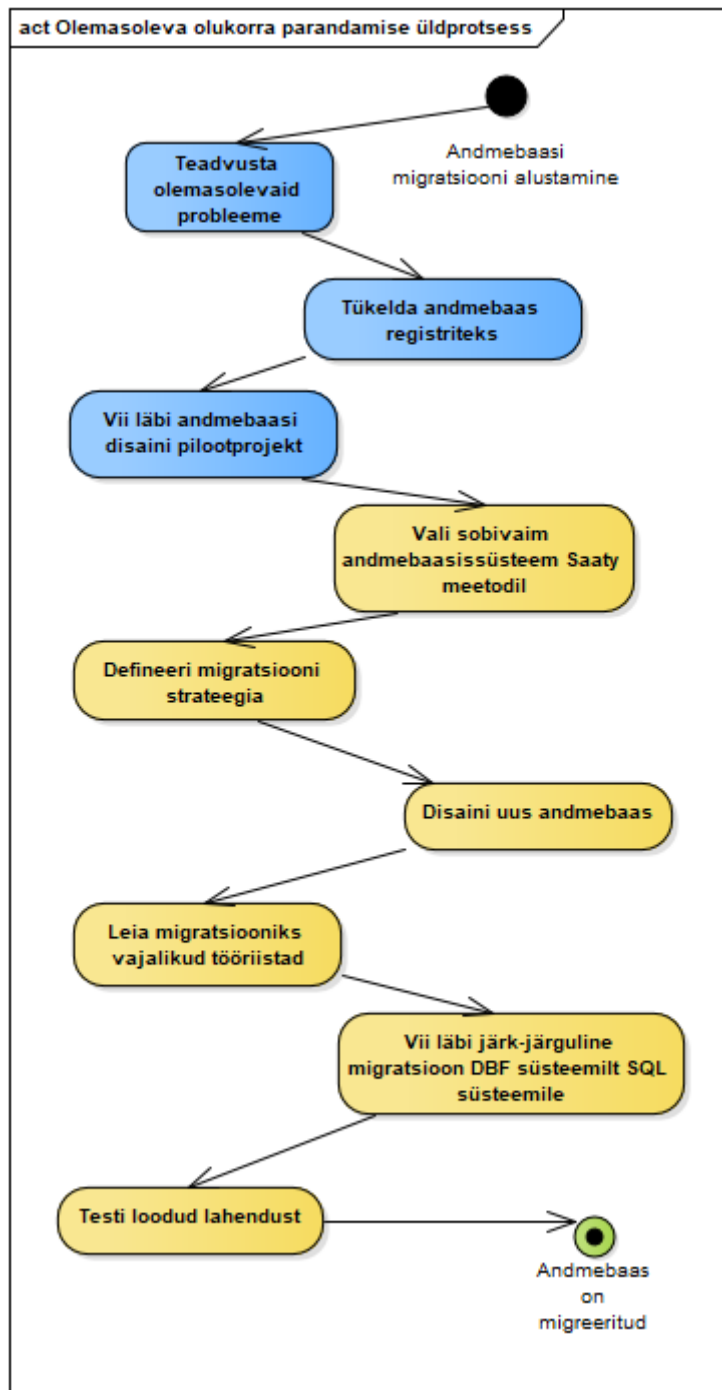
4 Olemasoleva olukorra parandamise protsess

Kasutusel olevate andmebaaside uutele andmebaasisüsteemidele migreerimiseks on välja töötatud erinevaid meetodikaid, olenevalt algsüsteemi ning sihtsüsteemi olemusest, andmebaasi struktuurist ning protsessi eesmärkidest. Hea ülevaate erinevate tänapäevaste andmebaaside migratsiooni meetodikate kohta saab näiteks Elamparithi et al. [38] ülevaateartiklist.

Käesolevas peatükis kirjeldatakse autori poolt kavandatud ülemineku protsessi, mida töö autor soovib TAAVI tarkvara arendajatel järgida.

4.1 Protsess tervikuna

Tegevusdiagrammil “Olemasoleva olukorra parandamise üldprotsess” (joonis 4) on välja toodud käesoleva töö mõistes eduka andmebaasi migratsiooni tinglikud etapid. Protsess ei ole planeeritud rangelt klassikalise koskmudelina (nagu diagramm mulje jätab). Vajadusel on lubatud eelnevatesse etappidesse tagasipöördumine.



Joonis 4. Tegevusdiagramm "Olemasoleva olukorra parandamise üldprotsess".

Käesoleva töö skooopi kuuluvad tegevusdiagrammil välja toodud esimesed kolm etappi (diagrammil tähistatud sinisega). Olemasoleva olukorra probleemide teadvustamise tulemus on leitav peatükis 3. Olemasoleva süsteemi andmebaasi түкeldamine on protsessina detailsemalt kirjeldatud alapeatükis 4.2 ning selle rakendamise tulemused on kirjeldatud alapeatükis 5.1. Olemasoleva andmebaasi disaini parandamise pilootprojekti samme kirjeldatakse peatükis 4.3 ja tulemus asub alapeatükis 5.2.

Järgnevalt tutvustatakse lühidalt edasisi etappe.

Etapis “Vali sobivaim andmebaasisüsteem Saaty meetodil” valitakse eelvaliku läbinud SQL-andmebaasisüsteemide hulgast kõige sobivam kasutades analüütiliste hierarhiate meetodit e Saaty meetodit. Selle protsessi kondikava esitatakse jaotises 4.4. Esialgset alternatiivide hulka ning võimalikke otsustuskriteeriumeid nimetatakse jaotises 5.5.

Etapis “Defineeri migratsiooni strateegia” tuleb välja töötada migratsiooni üldstrateegia, kus tuleks muuhulgas:

- defineerida migratsiooni erinevad faasid,
- planeerida kuidas olemasolevad klientide andmed turvaliselt ning ning terviklikult konverteerida,
- valida meetodid migratsiooni järgseks andmete ning funktsionaalsuste valideerimiseks,
- planeerida täpsemalt olemasoleva pärandsisüsteemi kasutamisest võõrutamine jpm [39].

Etapi “Disaini uus andmebaas” peamiseks eesmärgiks on välja töötada iga andmebaasi registri jaoks sobiv disainilahendus uue andmebaasisüsteemi jaoks. Sellel etapil võetakse arvesse pilootprojekti ning sellele järgnenud arutelude tulemused ja andmebaasisüsteemi valik. Andmebaasisüsteemi valik määrab *näiteks* selle, milliseid andmetüüpe on võimalik tabelite veergude puhul kasutada, milliseid kitsendusi saab andmebaasi tasemel jõustada ning millist tüüpi indekseid on võimalik ja soovitatav tabelite veergudele luua. Pilootprojekti tulemuste hindamise alusel tuleb otsustada, millist tüüpi parandusi tasub uues andmebaasis teha ja millised tingimused liiga suuri muutuseid andmebaasi kasutavas tarkvaras. Samuti on tarvis teha selgeks valitud sihtandmebaasisüsteemi võimaldatavad andmetüübid ning leida millised nendest vastavad kõige paremini olemasolevas andmebaasis kasutusel olevatele tüüpidele. Siinkohal on mõistlik ka analüüsida, millised pakutavatest andmetüüpidest sobivad andmebaasis hoitavatele andmetele kõige paremini, st ei ole ilmtingimata tarvis kinni jääda vanas süsteemis tehtud valikutesse. Tarkvara poolt läbiviidavaid andmebaasioperatsioone (andmete otsingud ja muutmised) analüüsides tuleb määrata uue andmebaasi esialgne sobilik indekseid komplekt. Etapp loetakse lõpenuks kui iga registri kohta on loodud uue andmebaasisüsteemi jaoks sobiv disaini

model kus on ka tehtud arendajate jaoks aktsepteeritav hulk tehnilise võla vähendusi. Jaotises 5.4 tutvustatakse arendajate tagasisidet käesolevas töös läbiviidud pilootprojektile, mis on selle etapi üheks sisendiks.

Etapis “Leia migratsiooniks vajalikud tööriistad” tuleb vastavalt valitud andmebaasisüsteemile ning eelnevalt defineeritud migratsiooni strateegiale valida sobilikud migratsiooni tööriistad või arendada vajalikud skriptid ise.

Järgnevalt toimub migratsiooni järk-järguline läbiviimine. Järk-järguline tähendab, et migratsiooni tehakse TAAVI moodulite ja selle kasutatavate registrite kaupa. Selle etapi käigus tuleb luua nii disaini mudeli alusel uus andmebaas, muuta rakendusi, et need kasutaksid uue disainiga andmebaasi kui kanda andmed üle uutesse andmestruktuuridesse (andmesiire). Selle etapi lõpuks töötab kogu rakendus SQL-andmebaasisüsteemiga.

Etapis “Testi loodud lahendust” tuleb eelkõige veenduda, et nii andmebaas kui sellega seotud rakendused töötavad nii nagu tarvis ja et andmete kvaliteet on säilinud.

Andmebaasi migratsiooni strateegia väljatöötamise kohta saab väga praktilise ning põhjaliku ülevaate näiteks Preston Zhang'i käsiraamatust “Practical Guide to Large Database Migration” [21].

4.2 Andmebaasi tükeldamine

Andmebaasi tükeldamise etapi põhiliseks eesmärgiks oli olemasoleva andmebaasi tabelite jaotus loogilise terviku moodustavatesse alamosadesse e registritesse. Sellise jaotuse eesmärk on hallata süsteemi keerukust ning võimaldada järk-järgulisi tegevusi süsteemi paremaks muutmisel ja migreerimisel. Selle asemel, et käsitleda andmebaasi ühe suure amorfse tervikuna, võimaldab tükeldus valida näiteks korraga parandamiseks väiksema andmebaasi alamosa. Seda lähenemist rakendatakse ka käesolevas töös, kus TAAVI Palk mooduli andmebaas jaotatakse alamosadeks ja seejärel tegeletakse pilootprojekti käigus ühe alamosa parandamisega. Registrite defineerimise esimeseks sammuks on süsteemi põhiolemitüüpide leidmine, tuginedes olemasoleva rakenduse kasutajaliidesele ning rakenduse demoversiooniga kaasnevale kasutusjuhendile [40].

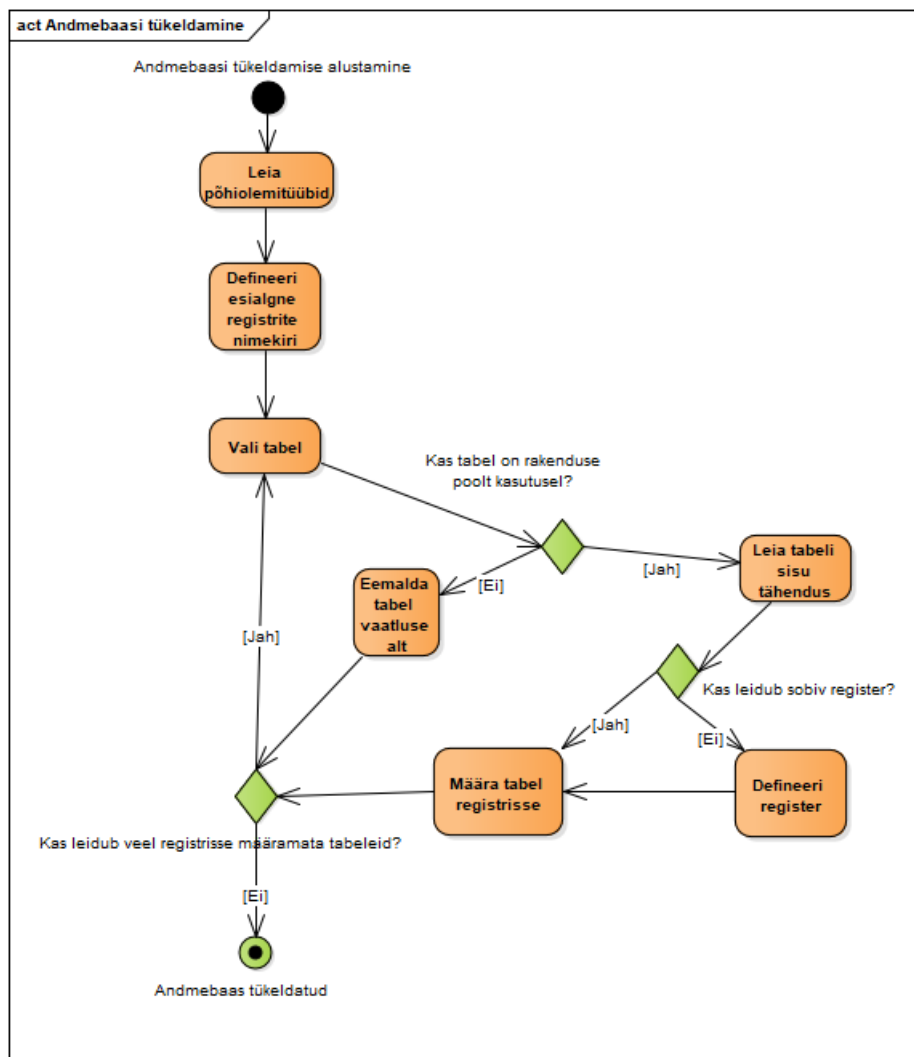
Leitu põhjal koostatakse esialgne registrite nimekiri – igale põhiolemitüübile vastab oma register.

Esialgsesse nimekirja lisatakse ka klassifikaatorite register. Klassifikaatorite registrisse paigutatakse tabelid, mille sisu kirjeldab ja liigitab andmebaasi teistes tabelites olevaid andmeid [41].

Register on andmebaasi loogiline alamosa, mis võimaldab hoida mingile põhiolemitüübile vastavaid andmeid [42]. Põhiolemitüüp on selliste füüsiliste või abstraktsete asjade (olemite) üldistus, mille kohta peab süsteem andmeid koguma kusjuures vastavat tüüpi olemitel võivad olla süsteemi seisukohast huvipakkuvad elutsükliid, mis määravad, kuidas neid andmeid peab süsteemis käsitlema SQL-andmebaasi/xBase andmebaasi korral on iga registri realiseerimiseks kasutusel üks või rohkem tabelit.

Fowler [43] liigitab tarkvara strateegiliseks ja tarbetarkvaraks. Strateegiline tarkvara on ettevõttele strateegiliselt oluline, aitab sellel konkurentidest eristuda ja selle kaudu on põhieesmäärke saavutada. Tarbetarkvara pakub tugiteenuseid. Piir nende vahel on sageli hägune ja oma elu jooksul võib tarkvara liikuda ühest liigist teise. Seda klassifikatsiooni saab kasutada et liigitada ka allsüsteeme, sh registreid. Erinevates sisulistes registrites olevad andmed võivad olla seotud andmetega ühes ja samas tarberegistris.

Pärast esialgse registrite nimekirja defineerimist saab alustada tabelite üksahaaval registritesse jaotamisega. Enne registri määramist tuleb teha selgeks, kas tabel on rakenduse poolt veel kasutusel. Juhul kui vastus on ei, siis eemaldatakse tabel vaatluse alt. Seejärel tuleb aru saada tabeli sisust ning kontrollida, kas sellele eksisteerib olemasolevas registrite nimekirjas sobilik register. Kui mitte, siis tuleb luua uus, vaadeldava tabeli sisule vastav register. Andmebaas on tükeldatud kui kõik andmebaasi poolt kasutusel olevad tabelid on määratud registritesse. Protsessi illustreerib tegevusdiagramm “Andmebaasi tükeldamine” (joonis 5).



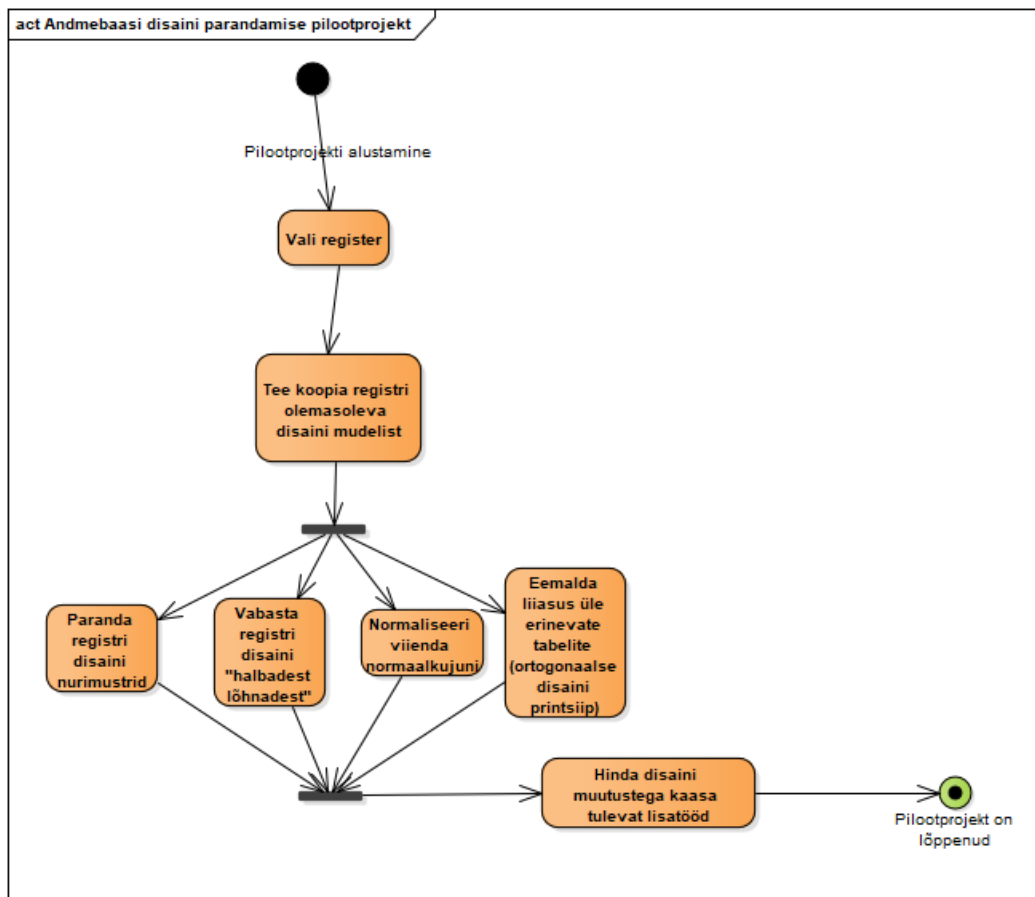
Joonis 5. Tegevusdiagramm “Andmebaasi tükeldamine”.

Tükeldamisprotsessi planeerimisel võeti aluseks Eessaare [42] väljapakutud praktikad andmekesksete süsteemide arendamiseks.

Erinevate infosüsteemide andmepõhiseks tükeldamiseks ja modelleerimiseks välja töötatud meetodite kohta saab veel lugeda näiteks Reijers et al. [44] ülevaateartiklist, mis aetas põhirõhu just mudelite kasutatavuse kvaliteedi aspektile.

4.3 Vii läbi andmebaasi disaini pilootprojekt

Etapi eesmärgiks on välja töötada ühe andmebaasi registri jaoks soovituslik ideaalmudel tuginedes andmebaasi disaini parimatele praktikatele. Selle alusel saavad arendajad otsustada, kas ja kui suures ulatuses soovivad nad migreerimise käigus andmebaasi disaini parandada.



Joonis 6. Tegevusdiagramm “Vii läbi andmebaasi disaini pilootprojekt”.

Töökäik algab valitud registri algse disaini mudeli koopia tegemisest (vt Joonis 6). Sellele järgnevad paralleelsed disaini parandavad protsessid.

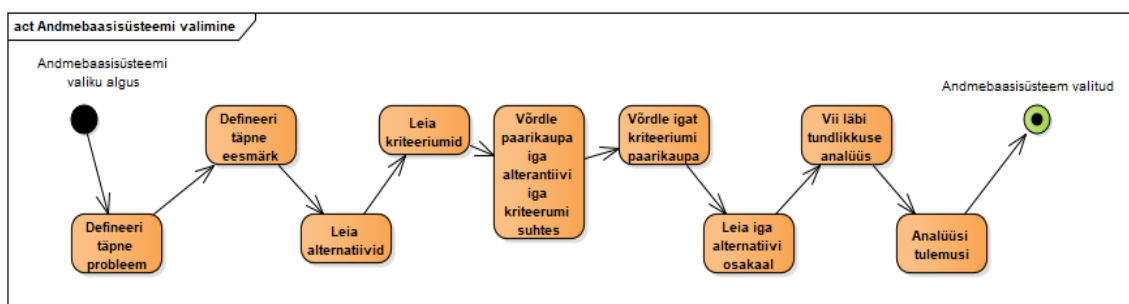
1. Registri disaini nurimustrite parandamine [30].
2. Registri disaini “halbade löhnadest” [31] vabastamine.
3. Registrisse kuuluvate tabelite normaliseerimine viienda normaalkujuni. Normaliseerimine on andmebaasi korrastamine, mille eesmärgiks on vältida andmestruktuuri piires andmete kordumist ja sellest tingitud andmete muutmise anomaaliaid [45].
4. Registri ning sellega seotud registrite erinevate tabelite vahel olevate liiasuste eemaldamine järgides ortogonaalse disaini printsiipi [33].

Saadud tulemust, ideaalmudelit, analüüsitakse koos tarkvara arendajatega ning hinnatakse andmebaasi muutustega kaasnevat lähtekoodi muutmiseks vajalikku lisatööd,

muutuste hädavajalikkust ning kasulikkust. Saadud info on sisendiks uue andmebaasi disainimise etapile.

4.4 Sobivaima andmebaasisüsteemi valik Saaty meetodil

Andmebaasisüsteemi valimise protsessi aluseks on analüütiliste hierarhiate meetod ehk Saaty meetod, mille eesmärgiks on otsuste tegemiseks süstemaatilise lähenemise pakkumine. Protsessi üldist kulgu illustreerib tegevusdiagramm “Andmebaasisüsteemi valimine” (joonis 7). Ka see diagramm kujutab ideaalvarianti ning tegelikult on võimalik eelmistesse etappidesse pöördumine, et näiteks tundlikkuse analüüsi tulemusena eemaldada analüüsist kriteeriume või muuta kriteeriumitele antud hinnanguid.



Joonis 7. Tegevusdiagramm “Andmebaasisüsteemi valimine”.

Saaty meetod aitab vähendada otsuste tegemisel subjektiivsust ning pakub arvutuskäigu, kuidas jõuda valikute e alternatiivide suhtelist headust esitavate arvuliste väärtusteni. Saaty meetodi põhiliseks tegevuseks on otsustaja poolt paarikaupa võrdluste tegemine. Sellisel viisil võrreldakse omavahel nii kriteeriume, et selgitada välja nende suhteline olulisus kui ka alternatiive kriteeriumite suhtes, et lõpuks jõuda alternatiivide suhtelise headuse leidmiseni. “Meetodi kasutamise võib jagada järgmisteks etappideks:

- Probleemi defineerimine - öeldakse, et õige probleemipüstitus on pool lahendust, vale probleemipüstitus toob vaid probleeme juurde
- Eesmärgi defineerimine
- Modelleerimine ehk süsteemi analüüs (tükeldamine ehk dekomponeerimine) ja süntees (hierarhia koostamine)
- Mõjurite (kriteeriumite) leidmine
- Valikute (alternatiivide) leidmine
- Valikute paarikaupa võrdlemine iga mõjuri suhtes

- Mõjurite paarikaupa võrdlemine
- Valikute osakaalude leidmine
- Tulemuste analüüs, sealhulgas kindlasti ka hinnangute kooskõla ja tulemuste tundlikkuse analüüs.” [46]

Saaty meetodi rakendamisest Eesti väikese ning keskmise suurusega ettevõtetele sobiva andmebaasisüsteemi valiku tegemisel on kirjutanud Dmitrijev [47]. Töös võeti vaatluse alla kolm populaarset vabavaralist andmebaasisüsteemi – MySQL, PostgreSQL ning FirebirdSQL. Seda allikat saab ka kasutada valimaks kriteeriumeid, mille alusel lõpuks valida sobivaim SQL-andmebaasisüsteem TAAVI majandustarkvara jaoks.

Eesti keeles Saaty meetodiga lähemalt tutvumiseks sobib väga hästi Võhandu [48] õpik.

5 Parandatud andmebaasi disain

Selles peatükis uuritakse olemasolevat andmebaasi detailsemalt, mooduli TAAVI Palk andmebaasi näitel. Eesmärgiks ei ole mitte terve mooduli andmebaasi ümberdisainimine, vaid olemasoleva andmebaasi olukorra detailsem uurimine andmebaasi väikese alamosa näitel. Eesmärgiks on saada paremini aru, kui palju ja millist tuleks teha tööd süsteemi uuele tehnilisele lahendusele üleviimisel. Süsteemi keerukuse haldamiseks pakutakse välja selle suure ja paljude ülesannetega mooduli jaotus alamosadeks e allsüsteemideks. Andmebaasi disaini parandus tehakse läbi ühe registri kohta.

5.1 Tükeldus registriteks

Töö autor jaotas TAAVI Palk mooduli andmebaasi registriteks ja esitab need registrid tabelis 2. Nimed on sorteeritud tähestiku järjekorras.

Tabel 2. TAAVI palk andmebaasi registrid.

Strateegilised registrid	Tugiregistrid (tarberegistrid, administratiivsed registrid)
1. Aruannete register	1. Allüksuste register
2. Arvestuste register	2. Ametikohtade register
3. Lisatasude register	3. Analüütikakoodide register
4. Palgalehtede register	4. Asukohtade register
5. Püsitasude register	5. Brigaadide register
6. Sotsiaalmaksu register	6. Deposiitide register
7. Tasude register	7. Erisoodustuste register
8. Tuludeklaratsiooni register	8. Filiaalide register
9. Tulumaksu register	9. Finantsarvestuse register
10. Tööaegade register	10. Haiguslehtede register
11. Tööde register	11. Honoraride register
12. Töötajate register	12. Isikute register
13. Väljamaksete register	13. Kasutajaliideste register
14. Ülekannete register	14. Kasutajate register
	15. Klassifikaatorite register
	16. Klientide register
	17. Komanderingute register

Strateegilised registrid	Tugiregistrid (tarberegistrid, administratiivsed registrid)
	18. Lepingute register 19. Pankade register 20. Programmilogi register 21. Puhkuste register 22. Riigieelarve allikate register 23. Seadistuste register 24. Sõidukite register 25. Tasuliikide register 26. Teenuste register 27. Toodete register 28. Võlgade register

Iga individuaalse TAAVI Palk demoversiooni andmebaasi tabeli kuuluvust registrisse väljendab lisast 3 leitav tabel “TAAVI Palk demoandmebaasi tabelite jaotus registritesse”. Kui välja jätta täpsed duplikaadid, kuulus demoversiooni andmebaasi 131 tabelit ning 131 neile vastavat indeksfaili.

5.2 Parandatud andmebaasi disaini mudel

Täpsema vaatluse alla valitakse püsitasude registri. Püsitasude register kogub andmeid iga töötaja jaoks ettenähtud püsitasude arvutamiseks valitud perioodil. Siin määratakse mitmed algandmed, mis on vajalikud palgaarvestuseks.

Püsitasude registrisse kuulub algses andmebaasis kaks tabelit, PYSITASU.dbf ning PTCOPY.dbf, mis on esimese täpne koopia ning mis on jäänud demopaketti ajutise lahendusena kunagise spetsiifilise probleemi jaoks. Seega tuleb sisuliselt vaadata selle registri puhul ainult ühte tabelit.

Esialgne PYSITASU.dbf tabel koosneb 24-st veerust ning saab vaikumisi osa andmetest töötajate andmeid kirjeldavast tabelist PYSI.dbf ja tasuliikide tabelit kirjeldavast tabelist TLTAB.dbf (joonis 8).



Joonis 8. Püsitasude registri algversiooni andmebaasi diagramm.

Järgnevalt esitatakse antud registri disaini parandamiseks läbi viidud sammud.

1. Tabeli veergude sisu ning päritolu mõistmine, mille koondtulemus on toodud tabelis 3. Antud sammu juures oli suureks abiks nii andmebaasi struktuuri dokumentatsioon kui ka arendajate antud lisanduvad selgitused. Väljavõtte dokumentatsioonist, kus kirjeldatakse tabelleid PYSITASU.dbf ning sellega kaht otseselt seotud tabelit PYSI.dbf ning TLTAB.dbf leiab lisast 2. C tähistab andmetüüpi Character, N tähistab andmetüüpi Numeric ja D tähistab andmetüüpi Date [49].

Tabel 3. PYSITASU.dbf tabeli esialgsete veergude nimed, kirjeldused, andmetüübid ning sisendid.

Veeru nimi	Andmetüüp	Veeru sisu seletus	Veeru sisendid
PKVTABN	C	Töötaja identifikaator TAAVI Personal andmebaasis. Kasutusel juhul kui klient kasutab mõlemat moodulit.	Vaikimisi PYSI.dbf-st/otsesisestus.
TABN	C	Töötaja identifikaator TAAVI Palk andmebaasis.	Otsesisestus (käitub sisuliselt välisvõtmena PYSI.dbf suhtes).
TL	C	Tasuliigi identifikaator, mis määrab püsitasu arvutusalgoritmi.	Vaikimisi PYSI.dbf-st/otsesisestus. Vastavalt tasuliigile TLTAB.dbf-st arvutusalgoritm.
SUMMA	N	Püsitasu arvutussumma. Võimalik sisend püsitasu arvutamiseks.	Vajadusel otsesisestus.
PROTSENT	N	Püsitasu arvutusprotsent. Võimalik sisend püsitasu arvutamiseks.	Vajadusel otsesisestus.
AEG	N	Püsitasu arvutamiseks tehtud töö aeg, tundides või päevades. Võimalik sisend püsitasu arvutamiseks.	Vajadusel otsesisestus.
KSIF	C	Näitab, millist kulukontot antud püsitasu tasumiseks kasutatakse.	Vaikimisi PYSI.dbf-st/otsesisestus.

Veeru nimi	Andmetüüp	Veeru sisu seletus	Veeru sisendid
ALGUS	D	Püsitasu kehtivuse alguskuupäev.	Otsesisestus.
LOPP	D	Püsitasu kehtivuse lõppkuupäev.	Otsesisestus.
SISESTUS	D	Püsitasu sisestamise kuupäev.	Genereeritakse (fikseeritakse LAHENDUS.dbf-s).
MAKSTUD	D	-	Ei kasutata.
ALLYKSUS	N	Palgaarvestuses on oluline määrata millisest allüksusest palk tuleb. Töötaja võib ajutiselt töötada ka võõras allüksuses.	Vaikimisi PYSI.dbf-st/otsesisestus.
PALGALEHT	N	Näitab, millisel palgalehel antud püsitasu kasutatakse.	Vaikimisi PYSI.dbf-st/otsesisestus.
NIMI	C	Töötaja nimi.	Genereeritakse PYSI.dbf-st, indekseerimise abiveerg.
XALLYKSUS	N	Igale töötajale on määratud tema vaikimisi allüksus, mille alt palka arvestatakse.	Genereeritakse PYSI.dbf-st, indekseerimise abiveerg.
ANALKOOD	C	Finantsarvestuseks vajalike koodide määramine.	Vaikimisi PYSI.dbf-st/otsesisestus.
PROJEKT	C	Finantsarvestuseks vajalike koodide määramine (projekti kood on analüütikakoodide alamtüüp).	Vaikimisi PYSI.dbf-st/otsesisestus.
ALLIKAS	C	Kasutatakse riigitöötajatele riigieelarve allika määramisel, millest antud tasu tuleb.	Vaikimisi PYSI.dbf-st/otsesisestus.
MARKUSED	C	Märkused.	Vajadusel otsesisestus.
KOORMKOEUF	N	Koormuse korrektsiooni koefitsient.	Vajadusel otsesisestus.
CRC	C	Abiveerg, milles oleva väärtuse abil saab kontrollida, kas andmeid	Genereeritakse.

Veeru nimi	Andmetüüp	Veeru sisu seletus	Veeru sisendid
		on peale selle väärtuse arvutamist muudetud.	
VIIMARVEST	D	Kasutajamugavuse mõttes näidatakse siin, milline on viimane lahendus, kus püsitasu on arvestatud.	Genereeritakse.
LEPRECID	N	TAAVI Personali moodulis oleva töölepingu identifikaator, mille alt seda püsitasu arvestada.	Otsesisestus/TAAVI personali mooduliga sidumiseks.
EIARVESTA	D	Selles väljas saab kasutaja sisestada konkreetse lahenduse identifikaatori (milleks on kuupäev), kus ei taheta seda püsitasu arvestada.	Vajadusel otsesisestus.

2. Primaarvõtme (*pysitasu_id*) lisamine.
3. Ortogonaalse disaini printsiibi rakendamine. Jätame alles ainult veerud, mis käivad rangelt püsitasu olemitüübi kohta. Kõik püsitasuga seotud, kuid seda mitte otseselt kirjeldavad veerud asendame välisvõtme või vahetabelitega, mis seovad *pysitasu* tabeli mõnele teisele olemitüübile vastava tabeliga.
4. Tabeli puhastamine mitte-kasutusel olevatest “surnud” veergudest (MAKSTUD).
5. Tabeli puhastamine tehnilistest abiveergudeks nagu indekseerimist mugavamaks muutvad XALLYKSUS ja NIMI.
6. Tabeli puhastamine veergudest, millele vastavad väärtused saab päringuga dünaamiliselt leida (VIIMARVEST).
7. Liiasuse vähendamine. Veerg PROJEKT on analüütikakoodi alamliik, mille kohta on analüütikakoode defineerivas tabelis ANALYYS.dbf eraldi tõeväärtustüüpi veerg – kas on projektikood või mitte. Sisuliselt on tegu sama olemitüübi jaoks kahe veeru kasutamisega, mis tõttu kaotame veeru PROJEKT ära.
8. Veerunimedede arusaadavamaks muutmine. Kõik tehtud muudatused on välja toodud tabelis 4.

Tabel 4. PYSITASU.dbf tabelis tehtud veerunimetuste muudatused.

Vana nimi	Uus nimi
pkvtabn	tootaja_pkvtabnr
tabn	tootaja_tabnr
tl	tasuliik
ksif	kulukonto
sisestus	sisestus_kp
algus	algus_kp
lopp	lopp_kp
leprecid	leping_id
summa	pysitasu_summa
protsent	pysitasu_protsent
aeg	pysitasu_aeg
crc	crc_hash

9. Normaliseerimise reeglitele vastavuse kontrollimine ja vajadusel nende reeglite rakendamine.

9.1. Kontrollime vastavust esimesele normaalkujule. Selleks peavad tabelis esitatavad väärtused olema neid kasutava süsteemi mõttes atomaarsed ja tabelis peab olema vähemalt üks kandidaatvõti. Analüütikakoodide jaoks oli algses tabelis veerg nimega ANALKOOD, kuhu oli võimalik sisestada komadega eraldatud listi analüütikakoodidest, mistõttu polnud atomaarsuse nõue täidetud, kuid selle jaoks tekitati 3. sammu käigus vajalik vahetabel. Samuti deklareeriti tabelile primaarvõti.

- 9.2. Kontrollime vastavust teise normaalkuju reeglitele. Iga mitte-primaarvõtme veerg peab olema täielikult funktsionaalselt sõltuv primaarvõtmest, st sõltub kogu primaarvõtmest, mitte mõnest selle osast. Püsitasude registri uue versiooni tabelid rahuldavad seda reeglit.
- 9.3. Kontrollime vastavust kolmanda normaalkuju reeglitele. Tabelite veergude vahel puuduvad ülekanduvad sõltuvused. Püsitasude registri uue versiooni tabelid rahuldavad seda reeglit.
- 9.4. Kontrollime vastavust kõrgematele normaalkujudele, mille tuvastamise kohta vahendab Eessaar [33] praktilisi reegleid. Mõistele “relvar” (*relational variable*, relatsiooniline muutuja) vastab antud kontekstis mõiste “tabel”.

“R1: Kui relvar on kolmandal normaalkujul (või Boyce/Coddi normaalkujul) ja iga selle kandidaatvõti on lihtne, siis on see relvar ka viiendal normaalkujul.

R2: Kui relvar on Boyce/Coddi normaalkujul ja mõni selle kandidaatvõti on lihtne, siis on see relvar ka neljandal normaalkujul (aga ei pruugi olla viiendal normaalkujul).

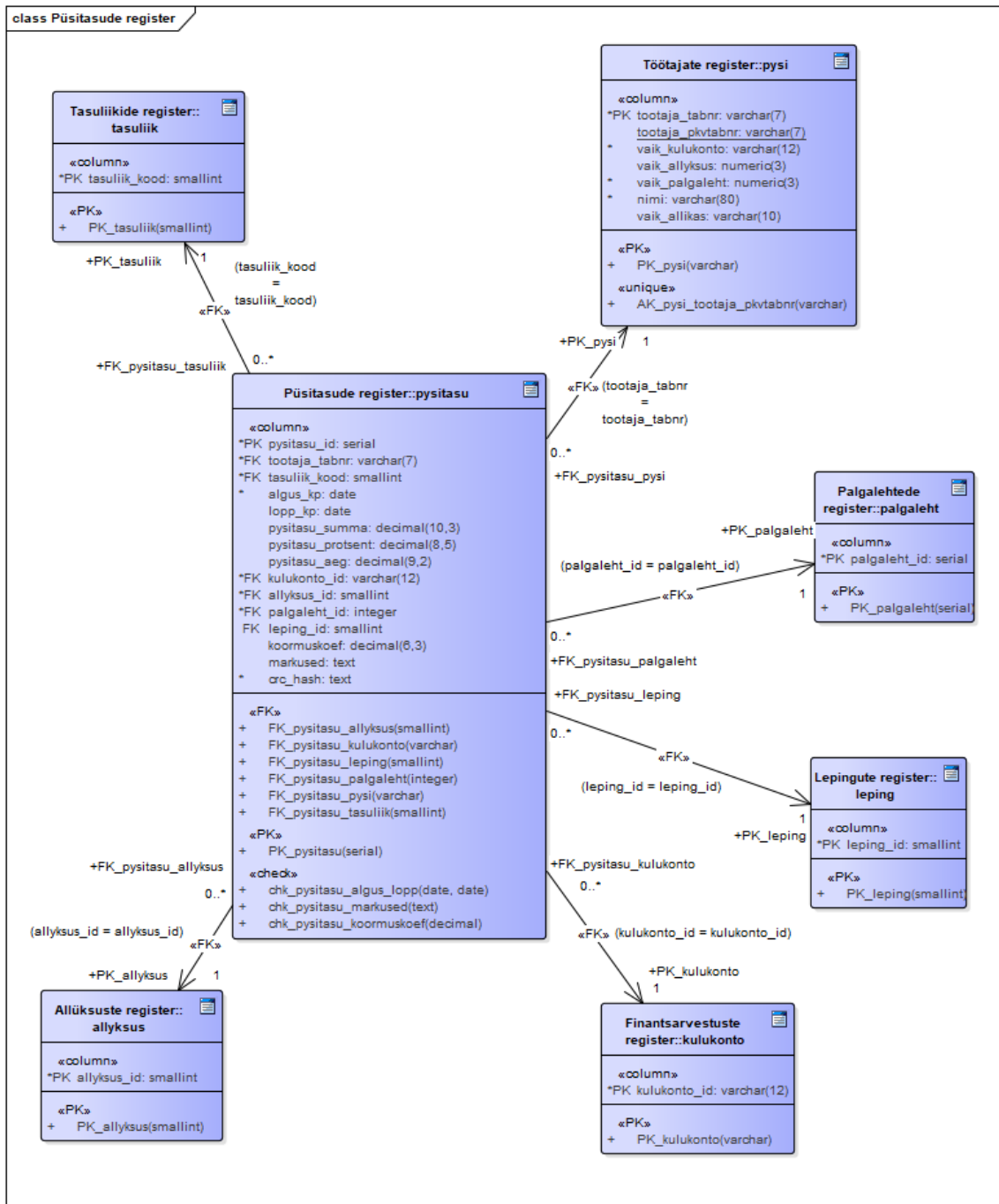
R3: Kui relvar on Boyce/Coddi normaalkujul ja sel on vähemalt üks atribuut, mis ei kuulu ühtegi kandidaatvõtmesse, siis see relvar on ka viiendal normaalkujul. Enamasti on Boyce/Coddi normaalkujul olevad relvarid ka viiendal normaalkujul.” [33]

Esimese reegli kohaselt võib tabeli *pysitasu* lugeda viiendale normaalkujule vastavaks. Ka ülejäänud püsitasude registri uue versiooni tabelid (*pysitasu_anakood*, *pysitasu_allikas*, *pysitasu_lahendus*) on viiendal normaalkujul, sest neid tabeleid ei saa enam jagada väiksemateks osadeks, mille kokkuühendamisel saab taastada algse tabeli.

Sellega loetakse registri disaini parandamine lõppenuks.

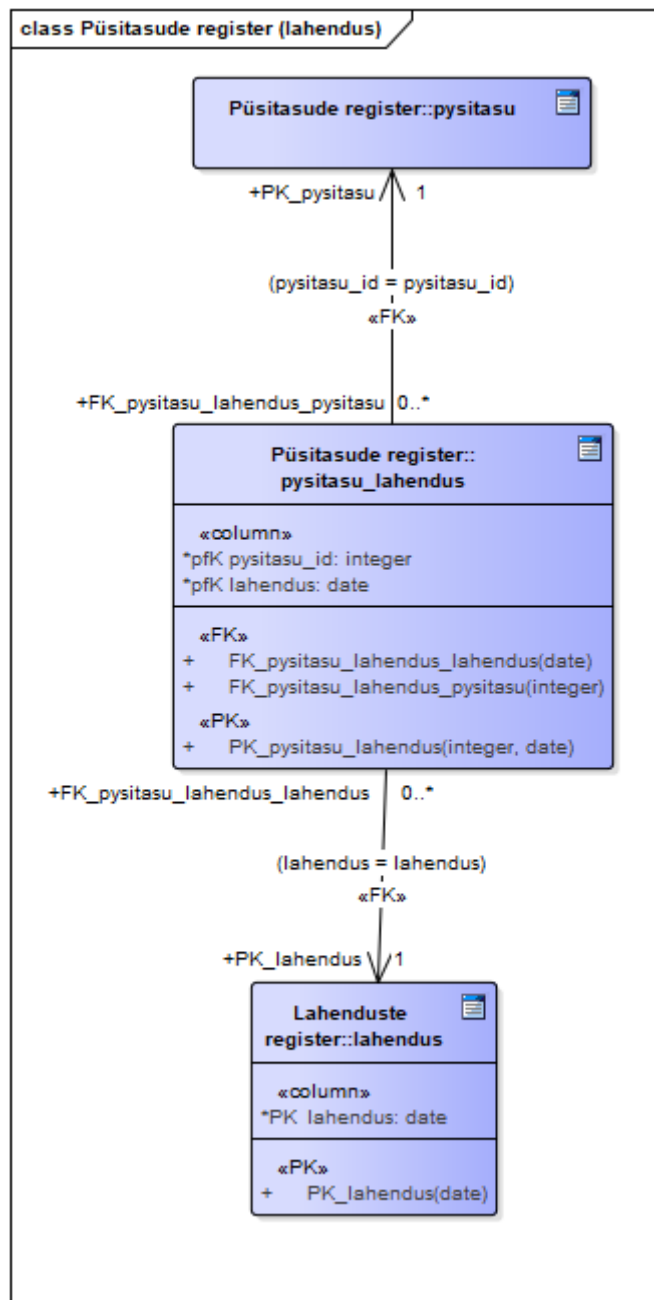
Disaini parandamise protsessi tulemuseks on järgnev mudel, mille diagrammid on loetavuse eesmärgil esitatud osade kaupa. Andmebaasisüsteemi valikut ei ole veel tehtud. Mudelis kasutatakse PostgreSQL andmetüüpe, mis on üks võimalik alternatiiv, millele TAAVI süsteemis üle minna. Mudelis ei esitata infot arendajate poolt loodavate indeksite kohta (primaarvõtmete ja unikaalsuse kitsenduste toetuseks loob süsteem indeksid

automaatselt, kuid välisvõtme veerge automaatselt ei indekseerita). Joonisel 9 on püsitasude registri keskne tabel ja sellesse välisvõtmed andvad tabelid teistest registritest.

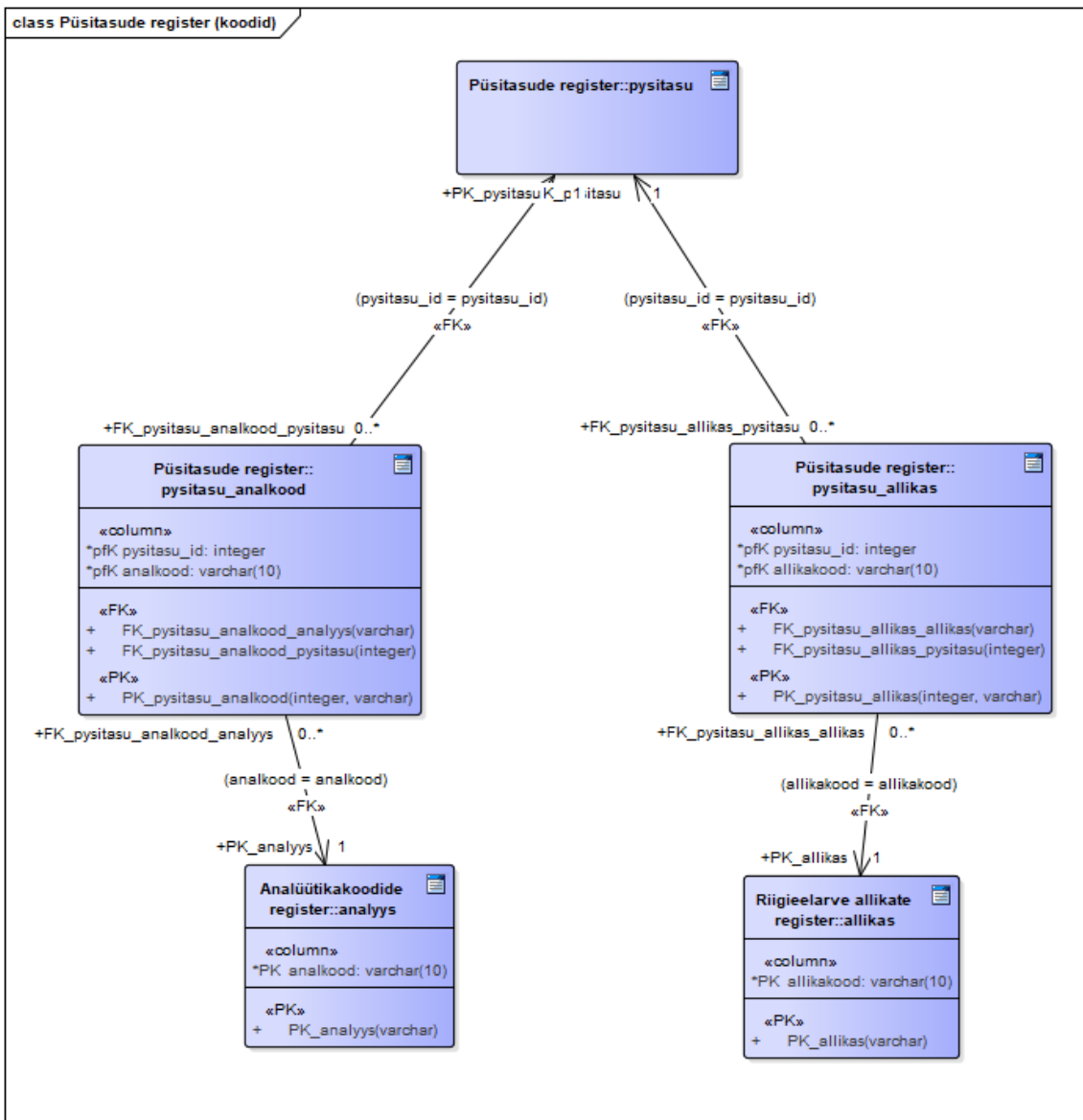


Joonis 9. Tabel pysitasu ja sellesse välisvõtme andvad tabelid.

Joonisel 10 on näidatud *pysitasu* tabelist välisvõtme saav tabel *pysitasu_lahendus* ja joonisel 11 *pysitasu* tabelist välisvõtme saavad tabelid *pysitasu_analood* ja *pysitasu_allikas*.



Joonis 10. Tabel pysitasu_lahendus.



Joonis 11. Tabelid pysitasu_analkood ja pysitasu_allikas.

Parandatud mudelis on püsitasude registri tabelid seotud üheksa eraldi olemitüüpi esindava tabeliga. Sellest kolme olemitüübi puhul oli püsitasude suhtes tegemist mitumitmele seosega, mistõttu tekitati ka vajalikud vahetabelid. Seega on parandatud mudelis 13 tabelit. Parandatud mudelis kuulub püsitasude registrisse neli tabelit. Tabel 5 kirjeldab ülevaatlilikult iga mudelis välja toodud tabeli sisu.

Tabel 5. Parandatud disaini mudelis esitatud tabelite kirjeldused.

Tabeli nimi	Registri nimi	Tabeli sisu
allikas	Riigieelarve allikate register	Riigieelarve allikad (vajalikud riigitöötajate palgaarvestusel).
allyksus	Allüksuste register	Allüksused koos tunnustega.
analüüs	Analüütikakoodide register	Raamatupidamiseks vajalikud analüütikakoodid.
kulukonto	Finantsarvestuste register	Finantsarvestuse kontod.
lahendus	Lahenduste register	TAAVI Palk programmiga koostatud arvestused (lahendused).
leping	Lepingute register	Töölepingud.
palgaleht	Palgalehtede register	Palgalehed.
pysi	Töötajate register	Töötajate koondandmed.
pysitasu	Püsitasude register	Algandmed iga töötaja jaoks ettenähtud püsitasude arvutamiseks määratletud perioodil.
pysitasu_allikas	Püsitasude register	Vahetabel püsitasude ning riigieelarve allikakoodide vahel, kuna iga püsitasu on seotud null või rohkema allikaga ning iga allikas on seotud null või rohkema püsitasuga.
pysitasu_analüüs	Püsitasude register	Vahetabel püsitasude ning analüütikakoodide vahel, kuna iga püsitasu on seotud null või rohkema analüütikakoodiga ning iga analüütikakood on seotud null või rohkema püsitasuga.
pysitasu_lahendus	Püsitasude register	Vahetabel püsitasude ning lahenduste vahel kuna iga püsitasu on seotud null või rohkema lahendusega ning iga lahendus on seotud null või rohkema püsitasuga.

Tabeli nimi	Registri nimi	Tabeli sisu
tasuliik	Tasuliikide register	Tasuliigid koos erinevate tunnustega, sh nende arvutusalgoritmid.

Tabelis 6 on toodud parandatud mudeli tabeli *pysitasu* veergude definitsioonid koos näidisväärtustega.

Tabel 6. Parandatud mudeli tabeli veergude definitsioonid ning näidisväärtused.

Tabeli nimi	Veeru nimi	Veeru definitsioon	Näidisväärtus
pysitasu	pysitasu_id	Surrogaatvõtme veerg püsitasu jaoks.	42
pysitasu	tootaja_tabnr	Välisvõtme veerg. Töötaja identifikaator, kellele püsitasu makstakse	A123456
pysitasu	tasuliik_kood	Välisvõtme veerg. Tasuliigi identifikaator, mis määrab püsitasu arvutusalgoritmi.	K9
pysitasu	pysitasu_summa	Püsitasu arvutussumma. Võimalik sisend püsitasu arvutamiseks.	500,00
pysitasu	pysitasu_protsent	Püsitasu arvutusprotsent. Võimalik sisend püsitasu arvutamiseks.	13,00000
pysitasu	pysitasu_aeg	Püsitasu arvutamiseks tehtud töö aeg, tundides või päevades. Võimalik sisend püsitasu arvutamiseks.	40,00
pysitasu	algus_kp	Püsitasu kehtivuse alguskuupäev.	2020-05-25
pysitasu	lopp_kp	Püsitasu kehtivuse lõpukuupäev. Lõpu kuupäev ei tohi olla alguse kuupäevast väiksem.	2020-12-25
pysitasu	kulukonto_id	Välisvõtme veerg. Näitab, millist kulukontot antud püsitasu tasumiseks kasutatakse.	KONTO1234567
pysitasu	allyksus_id	Välisvõtme veerg. Näitab, millise allüksuse alt antud püsitasu arvestatakse.	111

Tabeli nimi	Veeru nimi	Veeru definitsioon	Näidisväärtus
pysitasu	palgaleht_id	Välisvõtme veerg. Näitab, millisel palgalehel antud püsitasu kasutatakse.	222
pysitasu	leping_id	Välisvõtme veerg. Näitab, millisele töölepingule antud püsitasu vastab.	A123456789
pysitasu	koormuskoef	Võimalik sisend püsitasu arvutamiseks. Ei tohi olla negatiivne arv.	0,055
pysitasu	markused	Kasutajapoolsed vabas vormis märkused antud püsitasu kohta. Ei tohi olla tühi string või ainult tühimärkidest koosnev string.	„Parandatud 01.04.2020“
pysitasu	crc_hash	Abiveerg, milles oleva väärtuse abil saab kontrollida, kas andmeid on peale selle väärtuse arvutamist muudetud.	eaaf4698dbb4fd4e7 65ca14bcb45f927

5.3 Erinevate paranduste ja muudatuste kokkuvõte/analüüs

Püsitasude register on uuritavas andmebaasis leiduvate hulgaliste probleemide jaoks hea esindaja. Nagu kogu andmebaasis, esinesid registris kolmandas peatükis väljatoodud disaini probleemidest kehvad nimetused, surnud tabelid (ja veerud) ja mitteatomaarsed väärtused tabeli väljades, puudusid xBase platvormi piiratuse tõttu välisvõtme kitsendused, võimalus vältida andmebaasi tasemel korduvaid ridu ja võimalus jõustada andmebaasi tasemel täiendavaid deklaratiivseid lisakitsendusi (mille realiseerimiseks saab SQL-andmebaasides luua CHECK kitsendusi).

Registrit tehti parandamise käigus suurel hulgal muudatusi. Kõigepealt jäeti kõrvale inglisekeelse nimetusega duplikaattabel PTCOPY.dbf. Tänu ebavajalikest veergudest vabanemise ning normaliseerimisele kuulub parandatud tabelisse algse 24 veeru asemel kõigest 15.

Disaini parandamise käigus lisati *pysitasu* tabelile kokku kuus välisvõtme kitsendust ning lisaks veel kolm mitu-mitmele seost realiseerivat vahetabelit. Duplikaatide vältimiseks

määrati tabelile ka ühest veerust koosneb primaarvõti. Töö autorit jäi tabeli juures jätkuvalt häirima ainult ühe kandidaatvõtme olemasolu, kuid hetkel ei suudetud peale surrogaatvõtme lisamise muud unikaalsuse kitsendust leida. Selleks on vajalik täiendav süsteemianalüüs. Hetkel pakutud lahenduse korral võidakse sama püsitasu registreerida (kogemata või tarkvara vea tõttu) korduvalt, süsteem genereerib nendele ridadele erineva võtmeväärtuse, vormiliselt on ridade unikaalsus tagatud, kuid sisuliselt mitte.

Samuti leiti ja jõustati CHECK kitsenduste abil täiendavad piirangud.

- Lõpu kuupäev peab olema suurem võrdne alguse kuupäevast.
- Märkused ei tohi olla tühi string ja ainult tühimärkidest koosnev string.
- Püsitasu arvutamiseks kasutatav koormuskoeffitsient ei tohi olla negatiivne.

5.4 Esialgne tagasiside arendajatelt

Arendajate tagasiside näidiseks valitud registri parandatud mudelile oli mitmetine.

SQL-tüüpi andmebaasile üleminekuks hädavajalikud sammud nagu primaarvõtme lisamine ning atomaarsuse tagamine võeti küsimusteta vastu. Samuti reageeriti positiivselt mitte-kasutusel olevate veergude ära kaotamisele. Andmebaasi üldise puhastamise vajalikkus käis kogu käesoleva töö jaoks tehtud intervjuude käigus andmebaasi sisu kirjeldamisel korduvalt temaks.

Viienda normaalkujuni normaliseerimise ning ortogonaalse printsiibi range rakendamise suhtes oli seisukoht esialgu pisut kahtlev. Kuue välisvõtme veeru ning kolme mitu-mitmele seost esindava vahetabeli realiseerimine tähendab ulatuslikku rakenduse lähtekoodi muutmist ning vajatakse kindlust, et selleks vajaminev töö on lisapingutust ning ajakulu väärt.

5.5 Järgmised sammud

Arvestades arendajate tagasisidet tuleb enne registrite disaini läbiva parandamise juurde asumist hinnata disaini muutuste reaalset kasu nii tarkvarale endale kui selle arendusmeeskonnale. Adekvaatse hinnangu andmiseks tuleks leida ettevõtte vajadustega arvestav meetodika. Autori arvates tasub jällegi kaaluda analüütiliste hierarhiate meetodi

e Saaty meetodi kasutamist nii nagu alapeatükis 4.4 kirjeldatud viisil tuleb Saaty meetodit kasutades valida sobiv andmebaasisüsteem.

Üheks võimalikuks meetodiks on Eessaare [50] vahendatud Albarak ja Bahsooni [51] meetod, mis hõlmab tehnilise võla mõistet, et määrata normaliseerimise prioriteedid ja leida tabelid, mida tasuks normaliseerida. Kasutades majandusteooriast pärit portfelli teooriat võetakse eesmärgiks moodustada kombinatsioon nendest tabelitest, mille puhul normaliseerimise kasu andmete kvaliteedile ning andmebaasi töökiirusele oleks võimalikult suur ja riskid võimalikult väikesed.

Selle analüüsi käigus ei tohi unustada andmebaasisüsteemile üleminekuga kaasneb ka uue andmemudeli kasutuselevõtt, mistõttu andmete hoidmise ja töötlemise põhimõtted muutuvad ning vanas platvormis kasutusel olnud lahendused võivad uuele süsteemile otse ületõstes olla mitteoptimaalsed. Näiteks, kui DBF-failidel põhinevas süsteemis vastas igale tabelile eraldi fail (vt Lisa 1), siis võis töökiiruse huvides tabelite denormaliseerimine ja andmete dubleerimine erinevatesse tabelitesse olla põhjendatud lahendus. SQL-andmebaasisüsteemides, kus siseskeem (mille osaks on failid) ja kontseptuaalne skeem (mille osaks on tabelid) on veidi paremini (kuigi mitte ideaalselt) eraldatud ei pruugi see võit olla enam nii suur ja selle lahenduse probleemid kaaluvad eelised üle. Denormaliseerimisest ühe SQL-andmebaasi näitel kirjutab näiteks Peek [52]. Eksperimentidele tuginedes jõuab ta järeldusele, et „kuigi andmebaasi denormaliseerimise praktika kasutamises leidub positiivseid külgi, on andmebaasi kõrge tasemeni normaliseerimine operatiivandmete andmebaasides siiski otstarbekam tegevus kui denormaliseerimine.“ [52]

Praeguseks läbiviidud intervjuudel baseerudes tulevad andmebaasisüsteemi valikul hindamismudelisse omavahel võrdlusse MariaDB, PostgreSQL, MySQL ning Microsoft SQL Server. Kaalutakse ka agnostilist lahendust (tarkvara võib kasutada erinevaid SQL-andmebaasisüsteeme), et ettevõtte klientidele, kes soovivad oma andmebaasi siiski ise majutada, jääks vabadus valida endale kõige sobilikum andmebaasimootor. Peamiste otsustuskriteeriumitena on nimetatud võimalikult madal hind, tuntus Eestis andmebaasidega töötavate professionaalide seas, installeerimise ning haldamise lihtsus ning Clipperi keeles kirjutatud rakendusega ühendamiseks vajalike lisamoodulite olemasolu.

Pärast hinnangute leidmist ja andmebaasisüsteemi valimist tuleb igal juhul jätkata registrite uue disaini koostamisega, piirdudes minimaalselt vähemalt primaarvõtmete ning hädavajalike välisvõtmete määramisega ja jõustamisega, sest andmebaasi skeem ei ole antud hetkel SQL-andmebaasisüsteemide aluseks oleva andmemudeli põhimõtetega kooskõlas. Samuti jätkub kindlasti andmebaasi puhastustöö.

Üldjoontes on plaan jätkata vastavalt alapeatükis 4.1 esitatud üldistatud tegevusdiagrammile.

6 Kokkuvõte

Käesolev töö tegeleb Taavi majandustarkvara kaasajastamise ülesandega. Täpsemalt on plaan migreerida DBF-failidel põhinev pärandandmebaas mõnda SQL-andmebaasisüsteemi. Selle protsessi käigus on võimalikud ja vajalikud ka suuremad või väiksemad muudatused andmebaasi disainis.

Töö esimeseks eesmärgiks oli analüüsida olemasoleva andmebaasi platvormist tulenevaid probleeme ettevõtte seisukohast. Töö käigus sõnastati üldistatult kolm kõige olulisemat põhjust, mis motiveerivad protsessi algatamist. Nendeks on tööaja kokkuhoid nii ettevõtte arendajate kui ka kliendipoolsete andmebaasi administraatorite seisukohast, võimalus pakkuda klientidele kulutõhusamat ning rohkemate võimalustega pilveteenust ja kasutusel olevate arendusvahendite stagneerumine.

Töö teiseks eesmärgiks oli analüüsida TAAVI Palk mooduli näitel andmebaasi disaini kvaliteeti. Andmebaasi kvaliteeti mõjutasid tugevalt DBF-failidel põhineva andmebaasisüsteemi olemuslikud piirangud – eelkõige kitsenduste andmebaasi tasemel defineerimise võimaluse puudumine. Primaarvõtme kitsenduste puudumisel ei saa automaatselt ära hoida korduvate ridade teket, välisvõtme kitsenduste puudumine tähendab, et viidete terviklikkuse eest peab vastutama näiteks rakendus või andmebaasi kallale otse pääsev administraator. Andmebaasis esines ka laialdaselt andmete liiasust, hulgaliselt “surnud” tabeleid ja veerge ning veel mitmeid andmebaaside nurimustreid.

Töö kolmandaks eesmärgiks oli parandada ühe andmebaasi alamosa näitel andmebaasi disaini lähtuvalt erinevatest andmebaasi disaini parimatest praktikatest, milleks on nurimustritest ning disaini “halbade löhnadest” vabanemine, normaliseerimine ning ortogonaalse printsiibi rakendamine. Valitud register koosnes algselt kahest tabelist. Registri disaini parandamine algas dubleeriva tabeli kõrvalejätmisega. Allesjäänud tabel koosnes 24 veerust, millest jäi parandamisprotsessi lõpuks alles 15. Tabelile lisati surrogaatvõti, kuus välisvõtme kitsendust ning registrisse lisati kolm mitu-mitmele seost realiseerivat vahetabelit. Lisaks muudeti tabeli veerunimesid kasutajasõbralikumaks.

Töö neljandaks eesmärgiks oli paika panna migratsiooni läbiviimiseks vajalikud järgmised soovituslikud sammud. Migratsiooniprotsessi süstemaatiliseks läbiviimiseks kirjeldati kokku üheksa soovituslikku üldetappi, millele vastavalt on plaanis ka jätkata.

Pärast arendajatelt parandatud disainiga näidisregistri mudeli kohta esmase tagasiside saamist, ilmnes ka kõige vajalikum järgmine samm – hinnata disainimuutuste reaalsel kasu nii tarkvarale endale kui selle arendusmeeskonnale.

Antud töö praktiliseks väärtuseks on pakkuda Taavi majandustarkvara arendajatele abistavat informatsiooni planeeritava andmebaasisüsteemi väljavahetamist puudutavate otsuste tegemisel. Töö võib olla huvipakkuv ka teiste pärandüsteemide kaasajastajatele ning neile, kes otsivad praktilisi näiteid turul olevate tarkvarasüsteemide olukorra kohta.

Autor tänab töö juhendajat, Erki Eessaart, igakülgse toetuse ning heade nõuannete eest.

Kasutatud allikad

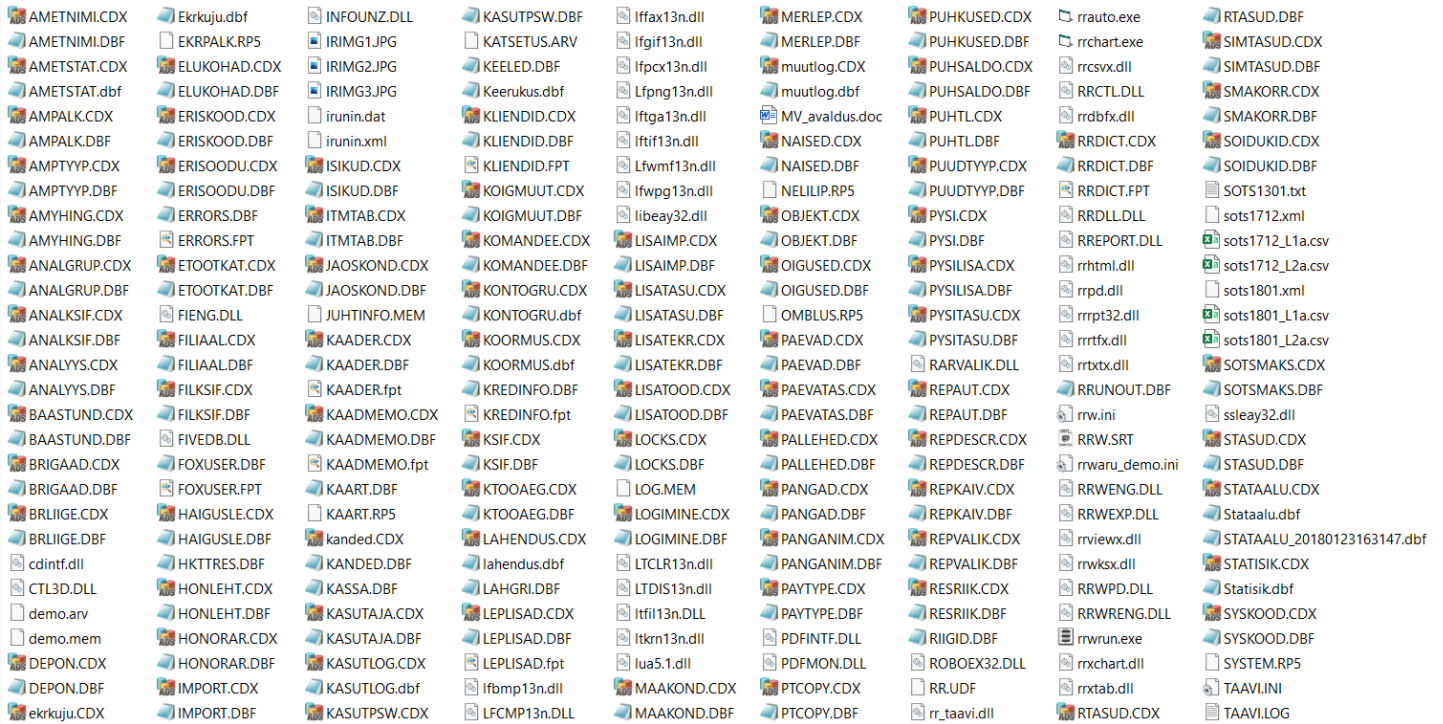
- [1] Vallaste e-teatmik [WWW] <http://www.vallaste.ee/> (22.05.2020)
- [2] Taavi Palk mooduli kirjeldus. [WWW] <https://taavi.ee/et/tooted/taavi-palk/tutvustus> (22.05.2020)
- [3] Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R. Action design research. — *MIS Quarterly*, 2011 35, 37-56.
- [4] Vijaya, A., Venkataraman, N. Modernizing legacy systems: A re-engineering approach — *International Journal of Web Portals*, 2018, 10(2), 50–60. [Online] IGI Global (22.05.2020)
- [5] Lehman, M.M. Laws of Software Evolution Revisited. — *Proceedings of 5th European Workshop, EWSPT '96 Nancy, France, October 9-11, 1996*. LNCS Vol. 1149, Berlin : Springer, 108–124.
- [6] Seacord, R.C., Lewis G.A, Plakosh, D. *Modernizing Legacy Systems*: Addison-Wesley Professional, 2003 [Online] O'Reilly Media (20.05.2020)
- [7] Raidmaa, K. Eesti e-riigi kesksete komponentide moderniseerimise uuring : bakalaureusetöö. Tallinn, Tallinna Tehnikaülikool, 2016. [WWW] <https://digikogu.taltech.ee/et/Item/31480182-3f3a-40ba-b6d9-b39e6321ad77> (23.05.2020)
- [8] Dbase. [WWW] <https://www.techopedia.com/definition/1192/dbase> (25.05.2020)
- [9] Bachman, E. Xbase File Format Description. [WWW] <https://www.clicketyclick.dk/databases/xbase/format/index.html> (25.05.2020)
- [10] Bogue, R. L. Explore the differences between ISAM and relational databases. [WWW] <https://www.techrepublic.com/article/explore-the-differences-between-isam-and-relational-databases/> (25.05.2020)
- [11] DB-Engines Ranking. [WWW] <https://db-engines.com/en/ranking> (25.05.2020)
- [12] Google Trends Explore .dbf File format. [WWW] <https://trends.google.com/trends/explore?date=all&q=%2Fg%2F11b669chvn> (25.05.2020)
- [13] Visual FoxPro to .NET. [WWW] <https://www.macrosoftinc.com/migration/visual-foxpro-to-net/> (25.05.2020)
- [14] SAP Advantage Database Server. [WWW] <https://www.sap.com/products/advantage-database-server.html> (25.05.2020)
- [15] OTC - tools for database applications. [WWW] <http://www.otc.pl/index.asp> (25.05.2020)
- [16] Alaska Software Products – Xbase++. [WWW] <https://www.alaska-software.com/products/overview.xcp> (25.05.2020)
- [17] Hernandez, M.J. *Database Design for Mere Mortals*. Addison-Wesley Professional, 2013. [Online] O'Reilly Media (22.05.2020)
- [18] Hainaut, J.L., Henrard, J., Englebert, V., Roland, D., Hick, J.M. *Database Reverse Engineering*. — *Encyclopedia of Database Systems*, Boston: Springer, 2009.

- [19] Yeh, D., Li, Y., Chu, W. Extracting entity-relationship diagram from a table-based legacy database. — *Journal of Systems and Software*, 2008, 81(5), 764–771. [Online] IEEE (22.05.2020)
- [20] What is Software Migration? [WWW] <https://www.canary.co.nz/services/software-migration> (25.05.2020)
- [21] Zhang, P. *Practical Guide to Large Database Migration*. CRC Press: Taylor & Francis Group, 2019.
- [22] Full Convert Tutorial. DBF to SQL Server migration. [WWW] <https://www.spectralcore.com/fullconvert/howto/dbf-to-sql> (25.05.2020)
- [23] Technical Debt. [WWW] <https://www.techopedia.com/definition/27913/technical-debt> (25.05.2020)
- [24] Käosaar, E. Lähtekoodi halbade lõhnade üldistamine süsteemianalüüsi mudelitele : bakalaureusetöö. Tallinna Tehnikaülikool, Tallinn, 2017. [WWW] <https://digikogu.taltech.ee/et/Item/d4736ffe-bd48-4112-81f6-e17a7a9cde98> (23.05.2020)
- [25] Li, Z., Avgeriou, P., Liang, P. A systematic mapping study on technical debt and its management. — *Journal of Systems and Software*, 2015, 101, 193–220. [Online] ScienceDirect (23.05.2020)
- [26] Paide, R.: Juhtumiuuring tehnilisest võlast Eesti IKT sektori ettevõtetes : magistrیتöö. Tallinna Tehnikaülikool, Tallinn, 2017. [WWW] <https://digikogu.taltech.ee/et/Item/73751810-09cd-4b15-888f-cad4f17e619f> (21.05.2020)
- [27] Ambler S., Sadalage P.J. *Refactoring Databases*. Addison-Wesley Professional, 2006 [Online] O'Reilly Media (22.05.2020)
- [28] Intervjuud ettevõtte arendusmeeskonna juhiga ajavahemikus veebruar-mai 2020
- [29] Liveware Publishing. Home of R&R ReportWorks. [WWW] <http://www.livewarepub.com/default.htm> (25.05.2020)
- [30] Karwin, B. *SQL Antipatterns. Avoiding The Pitfalls of Database Programming*. Pragmatic Bookshelf, 2014 [Online] O'Reilly Media (22.05.2020)
- [31] Martin, R.C. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2009.
- [32] Once-only principle [WWW] https://en.wikipedia.org/wiki/Once-only_principle (25.05.2020)
- [33] Eessaar, E. Andmebaasi loogilise disaini tulemuse parandamine ja headuse kontrollimine. Tallinna Tehnikaülikooli õppeaine “Andmebaasid I” teema 9 konspekt, 2020.
- [34] Aibast, K. Baastabelite kitsenduste nimede ühtlustamiseks mõeldud PostgreSQL laienduse loomine : bakalaureusetöö. Tallinna Tehnikaülikool, Tallinn, 2018. [WWW] <https://digikogu.taltech.ee/et/Item/05157bb4-763e-4a9d-bc50-2025692b8e1d> (25.05.2020)
- [35] Lapõnin, A. Andmebaasi loogilise ja füüsilise disaini antimustrite esinemine mõnedes vaba tarkvara poolt kasutatavates SQL-andmebaasides : bakalaureusetöö. Tallinna Tehnikaülikool, Tallinn, 2015. [WWW] <https://digikogu.taltech.ee/et/Item/5876763e-b6a7-42cc-a5e1-68123916c884> (22.05.2020)
- [36] Blaha, M. A Retrospective on Industrial Database Reverse Engineering Projects - Part 2. — *Eighth Working Conference On Reverse Engineering 2-5th October 2001, Stuttgart, Germany*, 147-153 [Online] IEEE (23.05.2020)

- [37] Weber, J.H., Cleve, A., Meurice, L., Ruiz, F.J. Managing technical debt in database schemas of critical software. — *Sixth International Workshop on Managing Technical Debt*, 2014, 43–46 [Online] IEEE (23.05.2020)
- [38] Elamparithi, M., Anuratha, V. A Review on Database Migration Strategies, Techniques and Tools. — *World Journal of Computer Application and Technology*, 2015, 3(3): 41-48. [WWW] https://www.researchgate.net/publication/299534668_A_Review_on_Database_Migration_Strategies_Techniques_and_Tools (23.05.2020)
- [39] Iyer, V., Venkataraman, B. Tips for database migration from obsolete databases to RDBMS. 2010. [WWW] <https://www.computerweekly.com/tip/Tips-for-database-migration-from-obsolete-databases-to-RDBMS> (25.05.2020)
- [40] TAAVI Palk demoversiooni rakendusesisene kasutusjuhend.
- [41] Klassifikaatorid. [WWW] <https://www.stat.ee/klassifikaatorid> (25.05.2020)
- [42] Eessaar, E. A Set of Practices for the Development of Data- Centric Information Systems. — *22nd International Conference on Information Systems Development, September 2013, Seville, Spain*. [WWW] https://www.researchgate.net/publication/265013970_A_Set_of_Practices_for_the_Development_of_Data-Centric_Information_Systems (25.05.2020)
- [43] Fowler, M. UtilityVsStrategicDichotomy. [WWW] 2010. <https://martinfowler.com/bliki/UtilityVsStrategicDichotomy.html> (25.05.2020)
- [44] Reijers, H.A., Vanderfeesten, I., Plomp, M.G.A. et al. Evaluating data-centric process approaches: Does the human factor factor in?. — *Softw Syst Model*, 2017, 16, 649–662, [Online] Springer.
- [45] Andmete modelleerimine: normaliseerimine [WWW] <https://www.metshein.com/unit/andmete-modelleerimine-normaliseerimine/> (25.05.2020)
- [46] Võhandu, L: Subjektiiivsetest hinnangutest objektiivsete tulemusteni. Loengukonspekt. Tallinna Tehnikaülikool. Informaatikainstituut.
- [47] Dmitrijev, O. Selection Criteria of the Database Management System for Estonian Small and Medium Sized Enterprises: magistritöö. Tallinna Tehnikaülikool, Tallinn, 2015. [WWW] <https://pdfs.semanticscholar.org/e0ff/293a8c47e2b024fe9b8c66f0d13cc86cd17a.pdf> (23.05.2020)
- [48] Võhandu, L: Subjektiiivsetest hinnangutest objektiivsete tulemusteni. Tallinn : Tallinna Tehnikaülikooli trükikoda, 1998.
- [49] DBF Field Types and Specifications [WWW] http://devzone.advantagedatabase.com/dz/webhelp/advantage9.0/server1/dbf_field_types_and_specifications.htm (25.05.2020)
- [50] Eessaar, E. Süsteemiarendus ja andmebaasi disaini koht selles. Tallinna Tehnikaülikooli õppeaine “Andmebaasid II” teema 1 konspekt, 2020.
- [51] Albarak, M., Bahsoon, R. Prioritizing technical debt in database normalization using portfolio theory and data quality metrics. — *In Proceedings of the 2018 International Conference on Technical Debt*, 2018, ACM. 31-40.
- [52] Peek, S. Denormaliseerimise praktika uurimine ühe SQL-andmebaasi näitel : bakalaureusetöö. Tallinna Tehnikaülikool, Tallinn, 2016. [WWW] <https://digikogu.taltech.ee/et/Item/d2dca054-b3fa-438a-ac84-70b6b3a0d7a3> (25.05.2020)

Lisa 1 – TAAVI Palk demoversiooniga kaasatulevad failid

Kuvatõmmis illustreerimaks TAAVI Palk demoversiooniga kaasa tulevat failide paketti.



Lisa 2 – TAAVI Palk demoandmebaasi dokumentatsiooni näidised

Kuvatõmmised tabelite PYSITASU.dbf, PYSI.dbf ning TLTAB.dbf väljade kirjeldustest.

Andmetabel	Jrkr	Andmeväli	Välja kirjeldus	Tüüp	Pikk	D
PYSITASU	1	PKVTABN	Tabelinumber personaliarvestuses	C	7	0
PYSITASU	2	TABN	Tabelinumber palgaarvestuses	C	7	0
PYSITASU	3	TL	Tasuliik	C	3	0
PYSITASU	4	SUMMA	Tasu summa	N	10	3
PYSITASU	5	PROTSENT	Arvutusprotsent	N	9	5
PYSITASU	6	AEG	Arvutuslik aeg	N	8	2
PYSITASU	7	KSIF	Kulukonto	C	12	0
PYSITASU	8	ALGUS	Arvestuse algus	D	8	0
PYSITASU	9	LOPP	Arvestuse lõpp	D	8	0
PYSITASU	10	SISESTUS	Sisestatud lahenduses	D	8	0
PYSITASU	11	MAKSTUD		D	8	0
PYSITASU	12	ALLYKSUS	Allüksuse kood	N	3	0
PYSITASU	13	PALGALEHT	Palgalehe kood	N	3	0
PYSITASU	14	NIMI	Töötaja nimi	C	80	0
PYSITASU	15	XALLYKSUS	Töötaja põhiallüksus	N	3	0
PYSITASU	16	ANALKOOD	Analüütikakood(id)	C	40	0
PYSITASU	17	PROJEKT	Projekti kood	C	10	0
PYSITASU	18	ALLIKAS	Allika kood	C	10	0
PYSITASU	19	MARKUSED	Märkused	C	60	0
PYSITASU	20	KOORMKOE	Koormuse korrektsioon	N	6	3
PYSITASU	21	CRC	MD5 HASH	C	32	0
PYSITASU	22	VIIMARVEST	Viimati arvestatud arvestuses	D	8	0
PYSITASU	23	LEPRECID	Personalilepingunumbri ID	N	10	0
PYSITASU	24	EIARVESTA	Seda püsitasu ei arvestata lahenduses	D	8	0

Andmetabel	Jrkr	Andmeväli	Välja kirjeldus	Tüüp	Pikk	D
PYSI	1	NIMI	Nimi	C	80	0
PYSI	2	EESNIMI	Eesnimi	C	30	0
PYSI	3	PERENIMI	Perekonnanimi	C	50	0
PYSI	4	PKVTABN	Kaadri tabelinumber	C	7	0
PYSI	5	TABN	Palgaarvestuse tabelinumber	C	7	0
PYSI	6	MKAT	Maksustamise kategooria	C	1	0
PYSI	7	KAT	*Ei kasutada: Töötaja ametirühm- VÕETAKSE AMETIKOHA KAARDILT	C	4	0
PYSI	8	KSIF	Töötaja palgakulukonto vaikumisi	C	12	0
PYSI	9	PST	Palgasüsteemi tunnus- EI KASUTATA	C	1	0
PYSI	10	AY	Ametiühingu tunnus	C	1	0
PYSI	11	ALLYKSUS	Allüksuse kood	N	3	0
PYSI	12	SUGU	Töötaja sugu	C	1	0
PYSI	13	TYKITOO	Kas on tükitööline	L	1	0
PYSI	14	SUMMA	Vaikumisi genereeritav põhipalk	N	10	2
PYSI	15	AVPROTS	Vaikumisi kasutatav avansiprotsent	N	3	0
PYSI	16	PALGALEHT	Vaikumisi palgaleht	N	3	0
PYSI	17	ELUKOHT	Elukoha omavalitsuse kood	N	5	0
PYSI	18	AADDRESS	Kodune aadress	C	70	0
PYSI	19	ISIKUKOOD	Isikukood	C	12	0
PYSI	20	SYNNIPÄEV	Sünnipäev	D	8	0
PYSI	21	ARVENR	Peamine arvelduskonto number	C	35	0
PYSI	22	PANK	Peamine pangakontor	C	10	0
PYSI	23	KUSTUTATUD	Kas töötaja eemaldatud palgaarvestusest	L	1	0
PYSI	24	KLIENT	Kliendikood raamatupidamises	N	5	0
PYSI	25	VALJAS	Töötaja hetkel arvestusest väljas-tunnus	L	1	0
PYSI	26	AMETIKOHT	Töötaja ametikoha kood	C	10	0
PYSI	27	MEILE_TOOL	Tööletuleku aeg	D	8	0
PYSI	28	LAHKUNUD	Töölt lahkumise aeg	D	8	0
PYSI	29	BAASTUNNID	Standardne päeva baastundide arv	N	8	2
PYSI	30	ANALKOOD	Vaikumisi analüütikakood	C	40	0
PYSI	31	SOTSMIIN	Sotsiaalmaks igal juhul miinimumpalgalt	L	1	0
PYSI	32	KOORMUS	Koormus	N	6	3
PYSI	33	ALLIKAS	Põhiallikas	C	10	0
PYSI	34	SOTSAMTA	Sotsiaalmaks alamäärata	L	1	0
PYSI	35	TARIIF	Vaikumisi tunnitariif	N	10	2
PYSI	36	EMAIL	E-maili aadress	C	50	0
PYSI	37	EITMKORR	Jäetakse tulumaks aasta lõpus ümber arvutamata	L	1	0
PYSI	38	RESIDKOOD	Residentriigi kood	C	3	0
PYSI	39	OECDLEPART	OECD mudellepingu artikli number	C	3	0
PYSI	40	MAKSELIIK	Mitteresidendi väljamakse liik	C	20	0
PYSI	41	TM2KUUP	Välismaise residentsustõendi väljastamise kuupäev	D	8	0
PYSI	42	FILIAAL	Asutuse kood	C	10	0
PYSI	43	TELEFON	Telefon	C	50	0
PYSI	44	AMETIYHING	Ametiühingu kood	C	10	0
PYSI	45	KOGPENSION	Liitunud kogumispensioni II sambaga.	D	8	0
PYSI	46	KOGPALGUS	Kogumispensioni kinnipidamise algus.	D	8	0
PYSI	47	KOGP2010	Esitanud avalduse kogumispensioni kinnipidamiste jätkamiseks 2010. aastal	L	1	0
PYSI	48	KOGPLOPP	Kogumispensioni kinnipidamise lõpp.	D	8	0
PYSI	49	VANPENSION	Vanaduspension alates	D	8	0
PYSI	50	MVALATES	Maksuvaba arvestame alates	D	8	0
PYSI	51	PROJEKT	Projekti kood	C	10	0
PYSI	52	KEEL	Suhtlemiskeel	C	10	0
PYSI	53	AYALATES	Ametiühingu alates	D	8	0
PYSI	54	AYKUNI	Ametiühingu kuni	D	8	0
PYSI	55	PRESPROTS	Preemiareseviprotsent	N	6	4
PYSI	56	SOTSMALAT	Sotsiaalmaks miinimumpalgalt alates kuust	D	8	0
PYSI	57	SOTSMKUNI	Sotsiaalmaks miinimumpalgalt kuni kuuni	D	8	0
PYSI	58	BOONUS	Saab boonustasu	L	1	0
PYSI	59	MARKUSED	Märkused	C	120	0
PYSI	60	KOGP2013	Esitanud avalduse suurendatud kogumispensioni kinnipidamiseks aastatel 2013-2014	L	1	0
PYSI	61	VANAARVENR	Vana arvelduskonto number	C	35	0
PYSI	62	A1RIIK	Tõendi A1/E101 väljastanud riik	C	2	0

Andmetabel	Jrkr	Andmeväli	Välja kirjeldus	Tüüp	Pikk	D
TLTAB	1	TL	Tasuliigi kood	C	3	0
TLTAB	2	NIMI	Tasuliigi nimetus	C	50	0
TLTAB	3	NIMIENG	Tasuliigi nimetus teises töökeeles	C	50	0
TLTAB	4	GRUPP	Tasuliigi grupp	C	30	0
TLTAB	5	ALLYKSUS	Allüksuse kood	N	3	0
TLTAB	6	ALG	Tasuliigi algoritm	C	2	0
TLTAB	7	T1	Kuulub puhkuskeskmisse	L	1	0
TLTAB	8	T2	Kuulub keskmise palga arvutamise alusesse	L	1	0
TLTAB	9	T3	Kuulub haiguskeskmisse	L	1	0
TLTAB	10	T4	Maksustatakse tulumaksuga	L	1	0
TLTAB	11	T5	Maksustatakse ametiühingumaksuga	L	1	0
TLTAB	12	T6	Kuulub muu kinnipidamise alusesse	L	1	0
TLTAB	13	T7	Kuulub alimendide arvestuse alusesse	L	1	0
TLTAB	14	T8	Kuulub täitelehtede alusesse	L	1	0
TLTAB	15	T9	Kuulub aastapreemia alusesse	L	1	0
TLTAB	16	MUUT	Kuulub muutuvatesse tasudesse	L	1	0
TLTAB	17	PYSI	Kuulub püsitasudesse	L	1	0
TLTAB	18	SIST	Sisestusreeglite tunnus	C	10	0
TLTAB	19	TUN	*Tasuliigi lisatunnus	C	2	0
TLTAB	20	KSIF	Vaikimisi konto	C	12	0
TLTAB	21	TYKITOO	Tükitöö tasuliik	L	1	0
TLTAB	22	TARIIF	Tükitöö tariif vaikumisi	N	10	4
TLTAB	23	T10	Kuulub põhitasusse	L	1	0
TLTAB	24	SOTSMAKS	**Ei kasutata	L	1	0
TLTAB	25	NULLIDA	Nullitakse kuuvahetusel	L	1	0
TLTAB	26	TUNNITASU	Arvestatakse tunnitasauna	L	1	0
TLTAB	27	PROTSENT	Vaikimisi protsent	N	8	4
TLTAB	28	STATARU	Aeg läheb statistikaaruandesse	L	1	0
TLTAB	29	KLIENT	Vaikimisi klient	N	5	0
TLTAB	30	NULLISUMMA	Summa nullitakse kuuvahetusel	L	1	0
TLTAB	31	NULLIPROTS	Protsent nullitakse kuuvahetusel	L	1	0
TLTAB	32	ANALKOOD	Vaikimisi analüütikakood	C	40	0
TLTAB	33	TMTL	Tulumaksu tasuliik	C	3	0
TLTAB	34	NULLAEG	Nullaeg tasul täidetakse täiskuu päevadega.	L	1	0
TLTAB	35	KESKMPER	Keskmissesse jagunemise periood kuudes	N	5	0
TLTAB	36	KESKALGUS	Keskmissesse jagunemise suhteline alguskuu	N	5	0
TLTAB	37	MITTETOO	Tasu mittetöötatud aja eest	L	1	0
TLTAB	38	STATVALJA	Tasu jääb statistikaaruandest välja	L	1	0
TLTAB	39	TYHIANAL	Ei kasutata töötaja kaardilt vaikumisi analüütikat	L	1	0
TLTAB	40	T11	Kuulub ametipalka	L	1	0
TLTAB	41	HAIGMAHA	Automaatselt ajast lahutame maha haigus- ja puhkuspäevad.	L	1	0
TLTAB	42	T12	Kuulub töötuskindlustuse alusesse	L	1	0
TLTAB	43	VMKOOD	Väljamaksukood TSD lisa 1 järgi kuni 2014 detsember	C	2	0
TLTAB	44	VMKOOD2015	Väljamaksukood TSD lisa 1 järgi alates 2015. aastast residendile.	C	2	0
TLTAB	45	VMKMRE2015	Väljamaksukood TSD lisa 2 järgi alates 2015. aastast mitteresidendile ILMA	C	3	0
TLTAB	46	VMKMRA1VAL	Väljamaksukood TSD lisa 2 järgi mitteresidendile Eestis tehtud töö ,välisriigi	C	3	0
TLTAB	47	VMKMRA1EST	Väljamaksukood TSD lisa 2 järgi VÄLIASPOOL Eestis tehtud töö ,EESTI A1/E1	C	3	0
TLTAB	48	STATRIDA	Statistikaaruande rida	C	2	0
TLTAB	49	T13	Kuulub kogumispensionimake alusesse	L	1	0
TLTAB	50	T14	Kuulub Soome tulumaksualusesse	L	1	0
TLTAB	51	MINUTITES	Aeg antud minutites	L	1	0
TLTAB	52	PS_HR_CD	Rootslaste konteering	C	10	0
TLTAB	53	PS_GRP_CD	Peoplesoft grupeering ID	C	10	0
TLTAB	54	KULUARULK1	Aruande "Tööandja kulutused" rida 1.lehel	C	2	0
TLTAB	55	KULUARULK2	Aruande "Tööandja kulutused" rida 2.lehel	C	2	0
TLTAB	56	TOOTSTRVE	Töötasustruktuuri aruande veeru tunnus	C	2	0
TLTAB	57	PASSIIVNE	Passiivne tasuliik	L	1	0
TLTAB	58	TKL2017AP	Statistikaameti tööjõukulu liikide klassifikaator	C	10	0
TLTAB	59	FONTES	Fontese tööperede aruande klassifikaator.	C	10	0

Lisa 3 – TAAVI Palk demoandmebaasi tabelite jaotus registritesse

Jrk	Registri nimi	Tabelid (tabeli järgi sulgudes on selle veergude arv)
1	Allüksuste register	ALLYANAL (11), ALLYKAIV (3), ALLYKSUS (29), IMPORT (2)
2	Ametikohtade register	AMETKOHT (29), AMETKSIF (4), AMPALK (6)
3	Analüütikakoodide register	ANALGRUP (6), ANALKSIF (3), ANALYYS (17)
4	Aruannete register	REPAUT (7), REPDESCR (3), REPKAIV (7), REPVALIK (5), RRDICT (12), STATAALU (39)
5	Asukohtade register	TOOKKOOD (4), VALLAD17 (9)
6	Brigaadide register	BRLIIGE (11)
7	Deposiitide register	DEPON (8)
8	Erisoodustuste register	ERISKOOD (6), ERISOODU (5)
9	Filiaalide register	FILIAAL (11), FILKSIF (4)
10	Finantsarvestuse register	KANDED (42), KSIF (28)
11	Haiguslehtede register	HAIGUSLE (38)
12	Honoraride register	HONLEHT (10), HONORAR (13)
13	Isikute register	ISIKUD (16)
14	Kasutajaliideste register	EKRKUJU (8), TAHVEL (12), WAATED (9)
15	Kasutajate register	KASUTAJA (12), KASUTPSW (10), OIGUSED (6), TEENOIG (3),
16	Klassifikaatorite register	AMETNIMI (2), AMPTYYP (2), AMETSTAT (4), AMYHING (4), BRIGAAD (2), ELUKOHAD (3), ETOOTKAT (4), KEERUKUS (3), KONTOGRU (2), MAAKOND (2), PAEVAD (2), PUUDTYYP (2), RESRIIK

Jrk	Registri nimi	Tabelid (tabeli järgi sulgudes on selle veergude arv)
		(3), RIIGID (3), TOOPERE (2), TOOTKAT (4), TTAMKOOD (2), JAOSKOND (5)
17	Klientide register	KLIENDID (102)
18	Komandeeringute register	KOMANDEE (38)
19	Lahenduste register	aa2demo (6), LAHENDUS (9)
20	Lepingute register	LEPLISAD (89)
21	Lisatasude register	LISAIMP (29), LISATASU (29), LISATEKR (29)
22	Palgalehtede register	PALLEHED (3)
23	Pankade register	PANGAD (10), PANGANIM (7)
24	Programmilogi register	ERRORS (2), KASUTLOG (7), KOIGMUUT (5), LOCKS (5), LOGIMINE (11), MUUTLOG (8)
25	Puhkuste register	PUHKUSED (27), PUHSALDO (54), PUHTL (3)
26	Püsitasude register	PTCOPY (25), PYSITASU (24)
27	Riigieelarve allikate register	ALLIKAS (3), ALLKSIF (4)
28	Seadistuste register	SYSKOOD (4), taaviini (4)
29	Sotsiaalmaksu register	SMAKORR (6), SOTSMAKS (17)
30	Sõidukite register	SOIDUKID (10)
31	Tasude register	PAEVATAS (27), RTASUD (28), SIMTASUD (28), STASUD (28), TASUD (28), TASUP (28), TOOTAB (14)
32	Tasuliikide register	PAYTYPE (15), TLTAB (59)
33	Teenuste register	TEENUS (3)
34	Toodete register	TOOTED (5), TOOTRIDA (7)
35	Tuludeklaratsioonide register	TSDA2015 (188), TSDALUS (554), TSDINF1 (13), TSDISIK (71), TSDLIS2F (13), TSDLISA3 (12)
36	Tulumaksu register	TMSUMMAD (17), TMTAB (16), TMVTAB (5), ITMTAB (12)
37	Tööaegade register	BAASTUND (4), KTOOAEG (188), TOOAEKR (10), TP (9), KOORMUS (7)

Jrk	Registri nimi	Tabelid (tabeli järgi sulgudes on selle veergude arv)
38	Tööde register	LISATOOD (19), TOOD (12), TOORIDA (14)
39	Töötajate register	KAADER (232), KAADMEMO (4), NAISED (20), PYSI (62), PYSILISA (63), STATISIK (4), TOOREG (24), TOOREGM (25)
40	Väljamaksete register	VALJAMAK (58), VMALL (22), VMSOTS (71), VMTASMA (24), VMTASUD (47)
41	Võlgade register	VOLAD (8)
42	Ülekannete register	OBJEKT (18), YLEKANNE (5)