

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Vlada Plaskovitskaja 152929IABM

**MEDITSIINITÖÖTAJATE TÖÖ- JA
VASTUVÕTUGRAAFIKUTE
ANDMEKVALITEEDI MONITOORING
ANDMELAO LAHENDUSE ABIL**

Magistritöö

Juhendaja: Eduard Ševtšenko
PhD

Tallinn 2018

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: Vlada Plaskovitskaja

07.05.2018

Annotatsioon

Antud magistritöö eesmärgiks on analüüsida ja võrrelda andmelao disaini kontseptsioone ning luua valitud kontseptsiooni järgi andmeladu ja aruandluskeskkond infosüsteemide tarbeks, mis tuvastab lähteallikate andmekvaliteedi probleeme. Selle eesmärk on aidata kaasa nii andmekvaliteedi paranemisele kui ka ettevõtte tööprotsessi parendamisele. Töö autor osales projektis nii andmelao arendaja kui ka analüütiku rollis.

Püstitatud eesmärkide saavutamiseks luuakse kõigepealt analüüs kaardistamaks hetkeolukorda ettevõttes. Seejärel tehakse nõuete analüüs uue lahenduse loomiseks. Selleks viiakse läbi intervjuud juhtidega. Järgmisena viiakse läbi projekti riski analüüs, mille käigus identifitseeritakse riske, hinnatakse nende mõju ja tõenäosust ning pakutakse lahendusi. Teooria osas võrreldakse erinevaid andmelao koostamise lähenemisviise, valitakse antud ettevõtte jaoks sobivaima lähenemisviisi ja põhjendatakse, ning luuakse andmeladu ja aruanded, mis aitaksid parandada hetkeolukorda.

Töö põhitulemuseks on toimiv süsteem, mis on arendatud lähtudes ettevõtte vajadustest, selle tööprotsesside ja lähteallikate analüüsist ning parimast andmelao arendamise praktikast. Teistele ettevõtetele võib pakkuda huvi see, kuidas saada oma äritegevuse käigus tekkinud andmetest rohkem kasu, või kuidas monitoorida andmete kvaliteeti paremini ning töötada nendega efektiivsemalt. Käesolev töö annab ülevaate, kuidas see on ühe konkreetse näite puhul realiseeritud. Arendajatele võib huvi pakkuda töö realiseerimise osa, milles autor kirjeldab süsteemi arendust ja selleks võetud modelleerimise tehnikaid, metoodikat ja komponente.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 63 leheküljel, 6 peatükki, 30 joonist, 6 tabelit.

Abstract

Monitoring the data quality of medical workers' time sheet and consultation schedule using a data warehouse solution

The aim of the present Master's thesis was to analyse and compare different concepts of data warehouse design. On the basis of the chosen concept, a data warehouse and a reporting system for information systems were created with the purpose of detecting problems with data quality in sources. The goal of such procedure is to aid in improving the data quality as well as enhancing the work procedures of the company. The author of this thesis participated in the project as the developer of the data warehouse and as the analyst.

First step towards achieving the goals of the present thesis was to conduct an analysis for mapping the current situation in the company. This was followed by an analysis of requirements for creating a new solution. Requirements are determined through interviews carried out with managers. Next, a risk assessment for the project is completed, identifying risks and evaluating their impact and likelihood while also providing possible solutions. The theoretical section compares different approaches to creating a data warehouse and the most suitable approach for this specific situation is adopted and such choice justified. Following that, the data warehouse and reporting system are created to improve the current situation in the company.

The main result of this thesis is a functioning system that has been developed arising from the needs of one specific company, analysis of its work processes and data sources and the best practices of creating a data warehouse. Other companies might be interested in how to use the data collected over business activities more effectively and the present thesis provides an insight into one case study. Developers might be interested in the process of creating a data warehouse, over the course of which the author describes the development of the system and the applied modelling techniques.

The thesis is in Estonian and contains 63 pages of text, 6 chapters, 30 figures, 6 tables.

Lühendite ja mõistete sõnastik

AS-IS	Hetkeseis
BI	<i>Business Intelligence</i> Äriintelligentsus
BPMN	<i>Business Process Model and Notation</i> Notatsioon äriprotsesside modelleerimiseks
DW	<i>Data Warehouse</i> Andmeladu
ETL	<i>Extract, transform, load</i> Hõiva, töötle, laadi. Andmete pärimine lähteallikast, töötlemine ja teisendamine ning andmelattu laadimine.
HIS	<i>Hospital Information System</i> Haigla infosüsteem
SQL	<i>Structured Query language</i> Andmebaasi päringukeel
OLAP	<i>Online Analytical Processing</i> Reaalaja analüüsi süsteem [1]
OLTP	<i>Online Transaction Processing</i> Reaalaja tehingutöötluste süsteem [1]
SSIS	<i>SQL Server Integration Services</i> Microsofti komponent ETL protsesside koostamiseks
TO-BE	Tulevikuvaade

Sisukord

1 Sissejuhatus	11
1.1 Taust ja probleem	11
1.2 Eesmärk	12
1.3 Metoodika.....	13
1.4 Ülevaade tööst	14
2 Andmelao tutvustus	15
2.1 Andmelao tähendus	15
2.2 Andmelao ajalugu.....	16
2.3 Andmelao disaini kontseptsioonid.....	17
2.3.1 Bill Inmoni lähenemisviis.....	17
2.3.2 Ralph Kimballi lähenemisviis	18
2.3.3 Dan Linstedti lähenemisviis	21
2.3.4 Andmelao disaini kontseptsioonide võrdlus.....	24
2.4 ETL protsess	27
2.5 Metaandmed	28
2.6 Äriintelligentsus.....	29
3 Andmelao arenduse projekti riskid.....	31
4 Andmelao lahenduse loomine meditsiinasutusele.....	32
4.1 Andmelao eesmärk ja vajalikkus	32
4.2 Andmelao ja ärianalüütika lahendusele esitatavad nõuded	33
4.3 Andmelao arendamise projekti riskianalüüs.....	34
4.4 Tööaegade graafikute koostamise ja muutmise protsesside analüüs.....	36
4.4.1 Hetkeseis – AS-IS.....	36
4.4.2 Tulevikuvaade – TO-BE.....	38
4.4.3 Tööajagraafiku muutmise protsessi simuleerimine	41
4.5 Personali- ja haigla infosüsteemide sarnasused ning eripärad.....	42
4.6 Lähteallikate andmemudelite analüüs	43
4.7 Andmelao modelleerimise metoodika valik ja põhjendus.....	49
4.8 Andmelao arendusprotsess	49

4.9 Andmelao projekti juurutamine	51
4.10 ETL protsessi koostamine	52
4.11 Dimensionaalne andmemudel.....	56
4.12 Ärianalüütika vahendi juurutamine ja aruannete loomine.....	57
4.13 Lahenduse arhitektuur	60
4.14 Ülevaade tööriistadest.....	60
5 Tulemuste analüüs	62
5.1 Kasu organisatsioonile.....	63
5.2 Autori ettepanekud	65
5.3 Autori tulemus	66
6 Kokkuvõte	67
Kasutatud kirjandus	69
Lisa 1 – HIS andmebaasi mudeli dokumenteerimine	72
Lisa 2 – Personali infosüsteemi vaadete dokumenteerimine	77
Lisa 3 – Andmelao dimensioonitabelite dokumenteerimine	81
Lisa 4 – ETL protsessiga seotud SQL laused	85
Lisa 5 – ETL arenduse AS-IS ja TO-BE protsessi simulatsioon.....	90

Jooniste loetelu

Joonis 1. Üldine ülevaade andmelaost.....	16
Joonis 2. Andmelao arhitektuur Inmoni lähenemisviisi järgi [7].	18
Joonis 3. Andmelao arhitektuur Kimballi lähenemisviisi järgi [7].	19
Joonis 4. Tähtskeem ehk <i>star schema</i>	20
Joonis 5. Lumehelbe skeem ehk <i>snowflake schema</i>	20
Joonis 6. Galaktika skeem ehk <i>galaxy schema</i>	21
Joonis 7. Data Vault ettevõtte andmelao arhitektuur [7]	22
Joonis 8. <i>Example of Data Vault hubs (physical design)</i> [5, p. 101].	23
Joonis 9. <i>Data Vault Link examples (physical design)</i> [5].	23
Joonis 10. <i>Data Vault satellite examples (physical design)</i> [5].	24
Joonis 11. Andmeladu ja analüütilised keskkonnad [4, p. 18].	29
Joonis 12. AS-IS tööajagraafiku muutmise protsess.	39
Joonis 13. TO-BE tööajagraafiku muutmise protsess.	40
Joonis 14. AS-IS protsessi simulatsiooni tulemused.	41
Joonis 15. TO-BE protsessi simulatsiooni tulemused.	42
Joonis 16. Personali infosüsteemi tööajatabeli staatused.	44
Joonis 17. Skript personali infosüsteemi tööajatabelite käivitamiseks.....	46
Joonis 18. Personali infosüsteemi tööajatabelitega seotud vaated.	47
Joonis 19. HIS tööajatabelite andmemudel.	48
Joonis 20. Kimballi elutsükli arendusmetoodika.....	50
Joonis 21. Andmelao projekti ülesanded.	51
Joonis 22. ETL protsess SSIS platvormis.....	53
Joonis 23. Muutujate loomine ja omistamine.....	53
Joonis 24. Skripti käivitamine <i>Execute SQL Task</i> komponendis.	54
Joonis 25. Andmete töötlemine <i>Derived Column</i> komponendiga SSIS platvormis.....	54
Joonis 26. Töö- ja vastuvõtugraafikute andmemudel.....	57
Joonis 27. Personali- ja haigla infosüsteemi graafikute võrdlusaruanne.....	59
Joonis 28. Andmelao lahenduse arhitektuur.....	60
Joonis 29. AS-IS ETL arenduse protsess.	64

Joonis 30. TO-BE ETL arenduse protsess..... 64

Tabelite loetelu

Tabel 1. Lähenemisviiside ja andmelao modelleerimistehnikate põhimõtteline võrdlus [7].	25
Tabel 2. Metodoloogiline ja arhitektuuriline modelleerimise tehnikate võrdlus [7].....	25
Tabel 3. Andmete modelleerimise võrdlus erinevates lähenemisviisides [7].	26
Tabel 4. Elutsükli juhtimise võrdlus erinevates modelleerimise tehnikates [7]......	26
Tabel 5. Riskianalüüs.	35
Tabel 6. Riskimaatriks.	36

1 Sissejuhatus

Iga eduka ettevõtte tulevik ja edukus sõltub sellest, kuidas ta suudab kasutada olemasolevaid andmeid, mida võib olla tohututes kogustes. Seda kõike selleks, et parandada tootlikkust, vähendada kulusid ja suurendada kasumit. Samuti on oluline, kuidas organisatsioonid andmeid talletavad. Näiteks uuenevad operatiivandmebaasides andmed reaalajas, kuid nende esitlusviis pole ärikasutajale otsuste tegemiseks mugav ega arusaadav. Seevastu andmelattu laetakse andmeid perioodiliselt, näiteks kord ööpäevas, kus selle käigus andmeid filtreeritakse, kontrollitakse ja teisendatakse. Selle käigus luuakse ärianalüütikaks sobiv andmestik.

Käesoleva lõputöö teema on hetkel eriti aktuaalne, sest paljud ettevõtted loovad praegu andmeladusid käsitsi, kirjutades suuremahulisi SQL (*Structured Query language*) lauseid või protseduure. Antud lõputöö annab ülevaate erinevatest võimalustest, kuidas ehitada andmeladu ETL (*Extract, Transform, Load*) protsessi kasutades. Andmelao kasutamine aitab tõsta arendajate tootlikkust, andmekvaliteeti ja päringute efektiivsust.

1.1 Taust ja probleem

Magistritöös keskendutakse ühele meditsiinasutusele, kus on hetkel kasutusel andmeladu ja aruandluskeskkond, mida on arendatud ühe lähtesüsteemi, haigla infosüsteemi (HIS) põhjal. Süsteem on kliinilise info keskne. Asutuses on teisi infosüsteeme, üks neist on personali infosüsteem. Mõlemad infosüsteemid sisaldavad meditsiinitöötajate tööaegade graafikute mooduleid. Kuna hetkel liidestus kahe infosüsteemi vahel puudub, siis peetakse info vahetamiseks tugitöötajate ja klienditeeninduse vahel telefonikõnesid ja e-kirjavahetust. Klienditeenindus kogub info kokku ning teeb parandusi arstide vastuvõtuaegade graafikutes. Andmeid sünkroniseeritakse käsitsi. Info liikumine on katkendlik ning sõltub igas protsessi ahelas inimtegevusest, mille toimimise tagamiseks tuleb omakorda sünkroniseerida sünkroniseerimisprotsessis osalevate inimeste puhkusegraafikuid. Probleeme tekitab asjaolu, et graafiku koostaja peab pidevalt otsima muudatusi ning erinevusi kahe

programmi vahel, ning kulutab selleks väga palju hinnalist tööaega. Patsiendi jaoks on kõige olulisem probleem, kui arst on kliinilise infosüsteemi järgi tööl, aga personali infosüsteemi järgi hoopis puhkusel, koolitusel või haige. Selliste juhtumite puhul võib patsient saada patsiendiportaalist või registratuurist valesid andmeid.

Olemasoleva andmelao arendamine ja hooldamine on asutusele kulukas. ETL-i teenuse arendus on haiglas usaldatud arenduspartneri kätte, mistõttu vajavad kõik muudatused ETL-is kooskõlastust ka teiste partnerhaiglatega. Nii on igasuguste muudatuste tegemine kulukas nii rahalises kui ka ajalises mõttes. Asutus kaalus kahe infosüsteemi liidestamist, kuid ka see oleks kulukas ning keerukas. Pidevalt muutuv loogika eeldab paratamatult inimkontrolli.

1.2 Eesmärk

Magistritöö eesmärk on arendada kahe infosüsteemi tarbeks olemasolevate vahendite abil alternatiivne süsteem (andmeladu ja aruandluskeskkond), mis tuvastab lähteallikate andmekvaliteedi probleeme andmete vastuolude seiramise abil. Selle eesmärk on aidata kaasa nii andmekvaliteedi paranemisele kui ka tööprotsessi organiseerimisele, kuna erinevate lähtesüsteemide andmete kuvamine ühe vaatenägemise tagab parema info liikumise. Eesmärk on loodava lahenduse abil avastada ka varjatud ebakõlasid töögraafikute moodulite kasutamisel.

Lahendus aitab ära hoida suuremahulisi kulutusi, mis kaasneksid kahe erineva infosüsteemi vahele liidese loomisega. Samuti toob see teenuse ettevõtte haldusalasse ning aitab vähendada sõltuvust arenduspartneritest. Uue süsteemi loomisel tuleb lähtuda organisatsiooni vajadustest, analüüsist ja parimast andmelao arendamise praktikast. Lahendus peaks olema jätkusuutlik ning dokumenteeritud.

Magistritöö raames pannakse peamiselt rõhku graafikute aruannetele. Tulevikus saab andmelattu kaasata ka infosüsteeme, mis seotud raamatupidamine, kõnekeskuse, apteegisüsteemi ja siseveebiga.

Uus andmeladu ja aruandluskeskkond on planeeritud luua *Microsoft SQL Server 2016 Standard Edition* toote perekonda kuuluvate vahendite abil, kuna need litsentsid on asutusel juba olemas.

1.3 Metoodika

Töö põhihüpotees oli, et kahe infosüsteemi graafikute andmed pole vastavuses ja sisaldavad palju ebatäpsusi. Teine hüpotees oli, et pärast vastava andmelao ja ärianalüütika vahendite juurutamist info hakkab liikuma operatiivsemalt ja veatumalt kuni seni.

Vastavalt sellele oli metoodika kavandatud järgmiselt:

Kogutakse erinevatel viisidel teavet hetkeolukorra välja selgitamiseks. Uuritakse, milliseid probleeme esineb praegu graafikute kasutamisel, ning mida oodatakse uuest lahendusest. Olukorrast selge pildi saamiseks suheldakse asutuse töötajatega, kelle tööülesanded on seotud kliinilise töö graafikutega ning nende haldamisega.

Analüüsitakse kahe infosüsteemi andmekooseisu, kaardistatakse andmemudelid olemasolevate kohta ja disainitakse mudel loodava süsteemi jaoks.

Teadusartiklite abil antakse ülevaade andmelao disaini kontseptsioonidest ning võrreldakse nende sobivust erinevate ettevõtete jaoks, seejärel valitakse sobivaim.

Arendatakse andmeladu ja valitakse andmeanalüüsi vahend, mille lähtesüsteemiks saab andmeladu.

Luuakse aruannete komplekt, mis toob välja andmete vasturääkivused kahes infosüsteemis. Asutusele antakse soovitusi infosüsteemide parendamiseks. Andmelao ülesehitus dokumenteeritakse. Arendusprotsessis lähtutakse *Kimball Lifecycle* metoodikast.

Pärast vastava andmelao ja andmeanalüüsi vahendite juurutamist peab info liikuma senisest operatiivsemalt ja veatumalt. Selleks võrreldakse praegust olukorda ja tulevikuolukorda AS-IS ja TO-BE mudeli abil. Viiakse läbi protsessi simulatsioon BMPN tehnikaga *Bizagi* töövahendis ning tuuakse välja efektiivsemate protsesside kasutamisest saadav ajavõit.

1.4 Ülevaade tööst

Antud töö koosneb kuuest peatükist. Teises peatükis antakse ülevaade erinevatest olulisematest mõistetest (andmeladu, ETL protsess, äriintelligentsus) ning andmelao arhitektuurist. Võrreldakse erinevaid andmelao modelleerimise tehnikaid. Olulisemad infoallikad on Paulraj Ponniah raamat „Data warehousing fundamentals For IT Professionals. Second Edition“ (2010), Vincent Rainardi raamat „Building a Data Warehouse“ (2008), Ralph Kimball ja Margy Ross raamat „The Data Warehouse Toolkit, Third Edition. The Definitive Guide to Dimensional Modeling“ (2013) ja teadusartikkel „Comparative Study of Data Warehouses Modelling Approaches: Inmon, Kimball and Data Vault“ (2016).

Kolmandas peatükis tuuakse välja andmelao arenduse projekti riskid ja nende põhjused.

Neljandas peatükis antakse ülevaade hetkel toimuvatest ja protsessidest kasutusel olevatest infosüsteemidest; analüüsitakse olemasolevaid andmeallikaid ning valitakse sobivaim andmelao modelleerimise meetodika. Valik tehakse teises peatükis kirjeldatud disaini kontseptsioonide võrdluse põhjal. Vastavalt ärikasutajate nõuetele disainitakse andmemudel, luuakse andmeladu selleks vajaliku ETL protsessiga. Arendatakse aruannete komplekt aruandlusvahendis. Kasutatakse võrdlevat AS-IS ja TO-BE mudelit, mis võimaldab hinnata protsesside efektiivsust enne ja pärast pakutud lahenduse juurutamist.

Viiendas peatükis keskendutakse tehtud töö järelanalüüsile, analüüsitakse probleeme ja antakse soovitusi juhatusele nende ennetamiseks või lahendamiseks.

2 Andmelao tutvustus

Järgnevate alapeatükkide eesmärgiks on anda ülevaade andmelaost ning andmelao modelleerimise tehnikatest, tuginedes teaduskirjandusele ja teadusartiklitele. Välja tuuakse põhilised mõisted ja võrreldakse erinevaid tehnikaid. Viimastes alapeatükkides kirjeldatakse metaandmeid ja ETL protsessi.

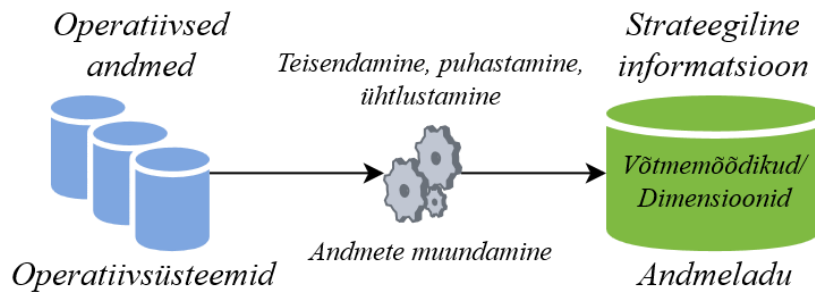
2.1 Andmelao tähendus

Iga ettevõtte edukus sõltub õigetest otsustest. Õigete otsuste tegemiseks tuleb arvestada asjakohaste andmetega. Nende andmete parimaks allikaks on hästi kavandatud andmeladu (inglise keeles *Data Warehouse* ehk *DW*).

Andmeladu on mitmest erinevast või ühest andmeallikast loogiliselt kokku integreeritud andmekogum, mis toetab otsuste tegemist ning võimaldab luua paindlikke aruandeid [2]. Harilikult kasutatakse ärianalüütikaks andmeid, mis ulatuvad tagasi paljudesse aastasesse. Selle jaoks on võimalik pärida andmeid andmelaost või tavapäringutena operatiivbaasist. Andmeladu hoitakse tootmiskeskonnast eraldi ning tavaliselt uuendatakse tervikportsudena, mitte alati kui allikasüsteemis toimub tehing [3, p. 1].

Alates 1980. aastate lõpust ja 1990ndateni hakkasid ettevõtted ehitama andmeladusid strateegilise informatsiooni hankimise eesmärgil. Andmeladu on sisuliselt muutunud ettevõtte teabe allikaks strateegilise otsuste tegemisel. Joonis 1 näitab üldist ülevaadet andmelaost kui ettevõtte strateegilise teabe allikast [4, p. 14].

Üldine protsess koosneb info kogumistest erinevatest andmeallikatest, andmete teisenduskihist, kus andmeid puhastatakse ja töödeldakse ning andmete laadimisest sobival kujul vastavalt andmemudelile andmelattu, mis on aluseks andmeanalüüside jaoks.



Joonis 1. Üldine ülevaade andmelaost.

Andmelao funktsionaalne määratlus on järgmine [4, p. 15]:

- Pakub integreeritud ja tervikvaadet ettevõttest.
- Annab ettevõttele võimaluse kergesti kätte saada praeguse ja ajaloolise informatsiooni strateegiliste otsuste tegemiseks.
- Võimaldab sooritada otsuste toetamiseks tehinguid operatiivsüsteemi takistamata.
- Tagab ettevõtte poolt kogutavate andmete terviklikkuse.
- Paindlik ja interaktiivne strateegilise teabe allikas.

Lisaks sellele, kui tootmisbaasis andmed pidevalt uuenevad ja muutuvad, siis andmelaos fikseeritakse andmete kindel seis kuni järgmise laadimiseni. See võimaldab aruanded teatud ajaperioodi jaoks muuta võrreldavaks.

2.2 Andmelao ajalugu

Enne andmeladude kasutusele võttu pärisid kasutajad vajalikku teavet operatiivsüsteemide baasidesse salvestatud toorandmetest. Operatiivandmebaasi päringute eeliseks on reaalajas teabe kättesaamine, kuid päringute abil töötlemata andmete muutmine analüütiliselt sobivale kujule aeglustab operatiivandmebaasi ning koos sellega ka operatiivinfosüsteemi tööd. Seda põhjustab andmete agregeerimine, mis nõuab suuremahuliste andmete lugemist, et esitada kokkuvõtte tehingutest (nt müük ühes kuus, tulu aastas). Ühe andmebaasi kasutamine samal ajal (nii operatiiv- kui ka analüütilisel eesmärgil) koormab andmebaasi. Selleks, et tagada kiiret juurdepääsu otsustusprotsessis nõutavale teabele, võtsid ettevõtted kasutusele äriotsuste tegemise tööriistad, mis ühendavad erinevaid laiendatavaid ja interaktiivseid IT-tehnoloogiaid ja vahendeid, mida kasutatakse andmete töötlemisel ja analüüsimisel [5, pp. 2-3].

Esimesena võttis kasutusele ja defineeris mõiste „andmeladu“ Bill Inmon 1970 aastatel. Andmelao kontseptsioon pärineb 1980.aastate lõpust, kui *IBM* teadlased Barry Devlin ja Paul Murphy arendasid eksperimentaalset „äriandmelao“. 1991. aastal Bill Inmon tegi andmelaod praktiliseks kui publitseeris raamatu „Building the Data Warehouse“ [6]. Seetõttu peetakse teda põhjendatult paljudes ringkondades andmelao isaks.

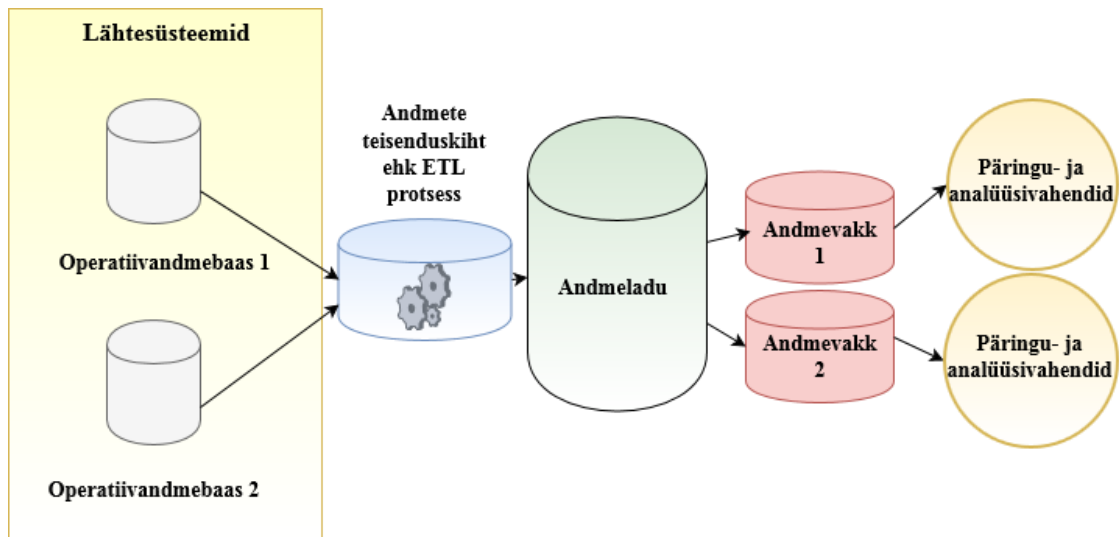
Alguses arendati ebatõhusaid süsteeme, mis olid väga aeglased ning andmeid laeti sinna harva, umbes kord nädalas. Mõne aja pärast tulid kasutusele efektiivsemad tehnoloogiad, mis aitasid andmelaadimise sagedust lähendada reaalaajalistele protsessidele.

2.3 Andmelao disaini kontseptsioonid

Andmelao struktureerimiseks on mitmeid lähenemisi. Kaks põhilist tuntud mudelit on Bill Inmoni „ülalt-alla“ („*top-down*“) ja Ralph Kimballi „alt-üles“ („*bottom-up*“). Kolmas lähenemisviis on Dan Linstedi pakutud *Data Vault*. Selles peatükis tutvustatakse neid kolme lähenemisviisi ja nende erinevusi; samuti nende eeliseid erinevate juhtumite puhul.

2.3.1 Bill Inmoni lähenemisviis

Bill Inmoni 90ndatel loodud andmelao struktuur (Joonis 2) hõlmab kogu ettevõtte andmeladu, selmet arvestada vaid üksikuid infoosi. Tema mudelis on andmebaasid jagatud neljale tasemele: operatiivne, andmeladu, osakonnapõhine andmevakk (inglise keeles *data mart*) ja üksikud individuaalsed tasemed. Viimased kolm on otseselt seotud andmelaoga, aga operatiivsel tasemel asuvad ettevõtte tehinguandmed, mida kasutatakse igapäevaselt. Need andmeid teisendatakse ja laetakse ETL (*Extract, Transform, Load*) abil andmelattu. Inmoni järgi on andmeladu ja andmevakk füüsiliselt eraldatud. Andmelaol on oma füüsiline olemus, mis on orienteeritud andmete ladustamisele, jälgitavusele ja skaleeritavusele uutele nõuetele vastamiseks. Andmevakkade struktuur vastab kasutajate nõuetele ning on orienteeritud tulemustele [7, p. 96].



Joonis 2. Andmelao arhitektuur Inmoni lähenemisviisi järgi [7].

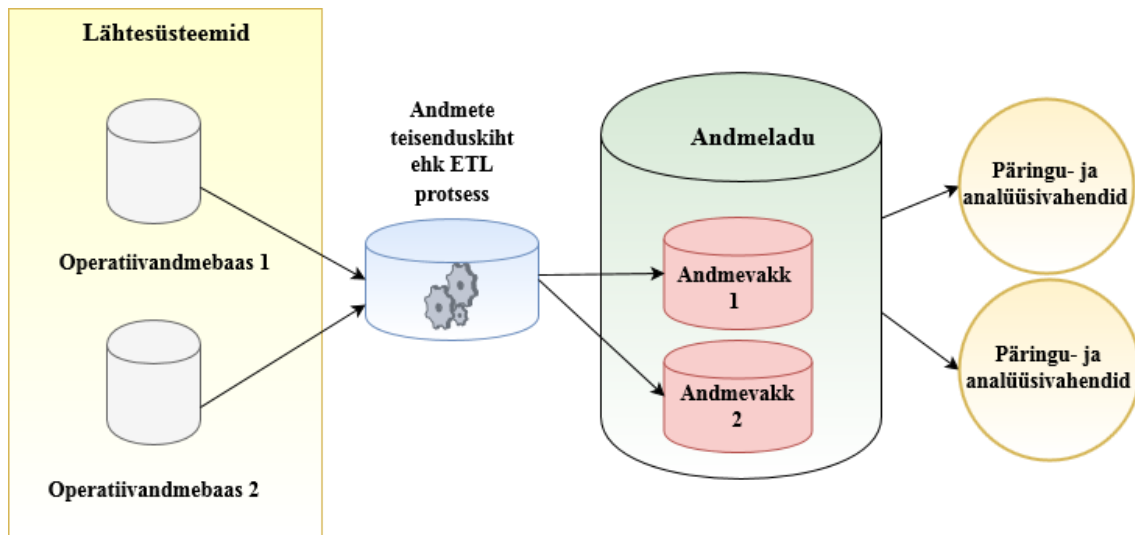
Selle lähenemisviisi puhul denormaliseeritakse andmed andmevakkades. Enne seda on andmed normaliseeritud ja neid hoitakse kõige madalamal detailsusastmel.

2.3.2 Ralph Kimballi lähenemisviis

Ralph Kimballi lähenemise järgi on andmeladu tehinguandmete koopia, mis on struktureeritud päringu ja analüüsi jaoks. Ka tema andmelao arhitektuur on pärit 90ndatest.

Kimballi andmelao arhitektuuri kontseptsioon põhineb dimensionaalsel modelleerimisel. Erinevalt Inmonist ei arva Kimball, et lõppkasutajaid tuleks isoleerida. Tema mudel on orienteeritud kasutaja vajadustele, ning lõppkasutajad on projekti kaasatud juba varastes etappides ja saavad andmelao struktureerimist mõjutada.

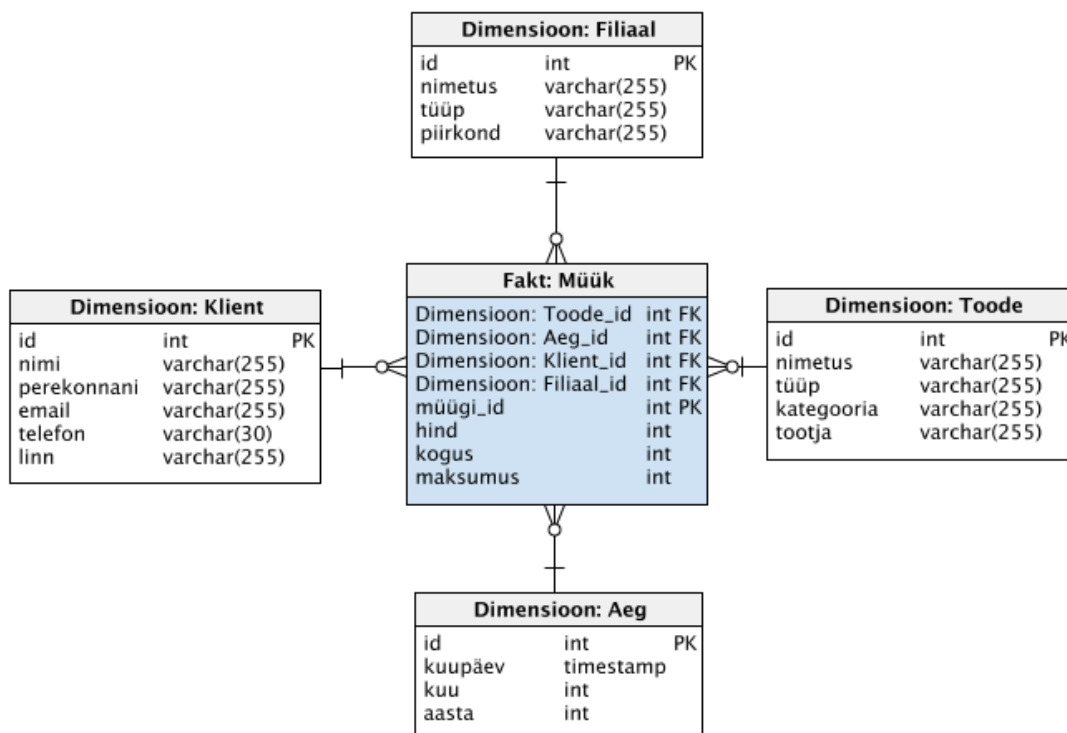
Kimballi järgi on andmeladu andmevakkade kogum (Joonis 3) [7, p. 96], [8].



Joonis 3. Andmelao arhitektuur Kimballi lähenemisviisi järgi [7].

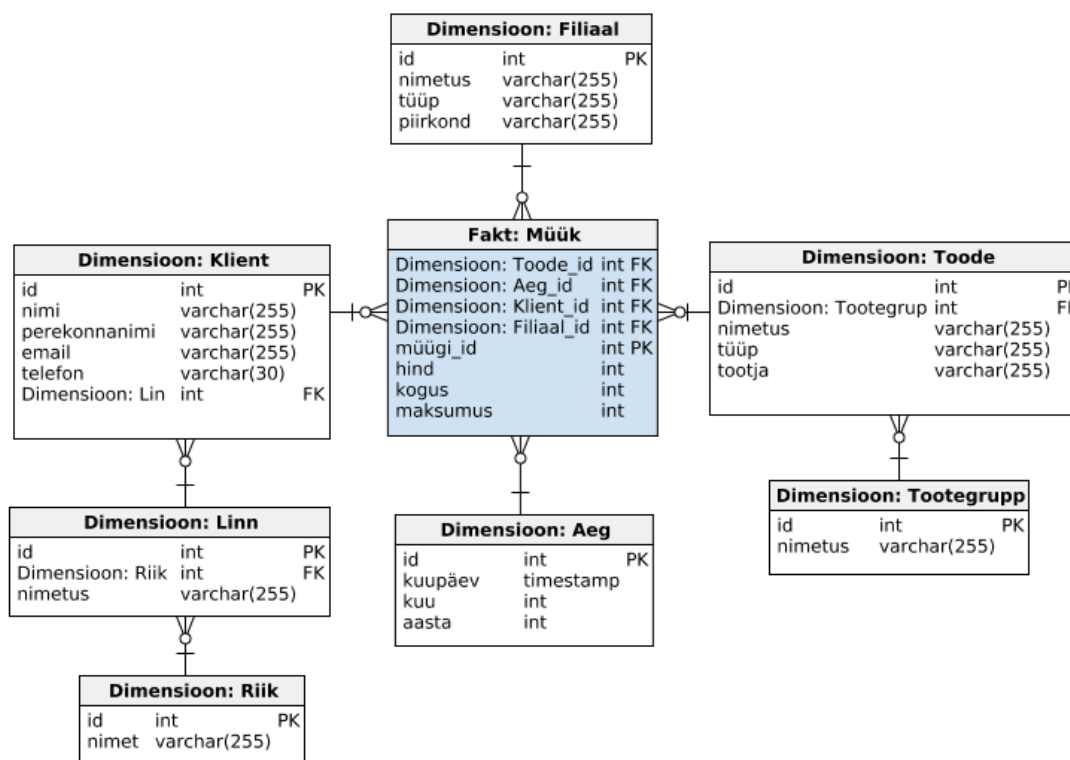
Tema lähenemisviisi järgi koosneb andmevakk dimensioonidest ja faktidest. Dimensioonides on andmed subjekti kohta, näiteks konkreetsete klientide või toodete kohta; faktid on andmed mõõdikute kohta kas summeritult või detailselt, näiteks müüdud toodete arv [9, p. 42]. Andmelao arendust võib alustada üksiku andmevaka loomisest ühe konkreetse osakonna või äriprotsessi jaoks [7].

Andmevakkades andmed on denormaliseeritud ning tabelid on seotud täht- (inglise keeles *star schema*), lumehelbe- (inglise keeles *snowflake schema*) või galaktika- skeemi (inglise keeles *galaxy schema*) alusel. Tähtskeemis (Joonis 4) dimensioonidel pole alamdimensioone (inglise keeles *a subdimension*) [3, p. 7].



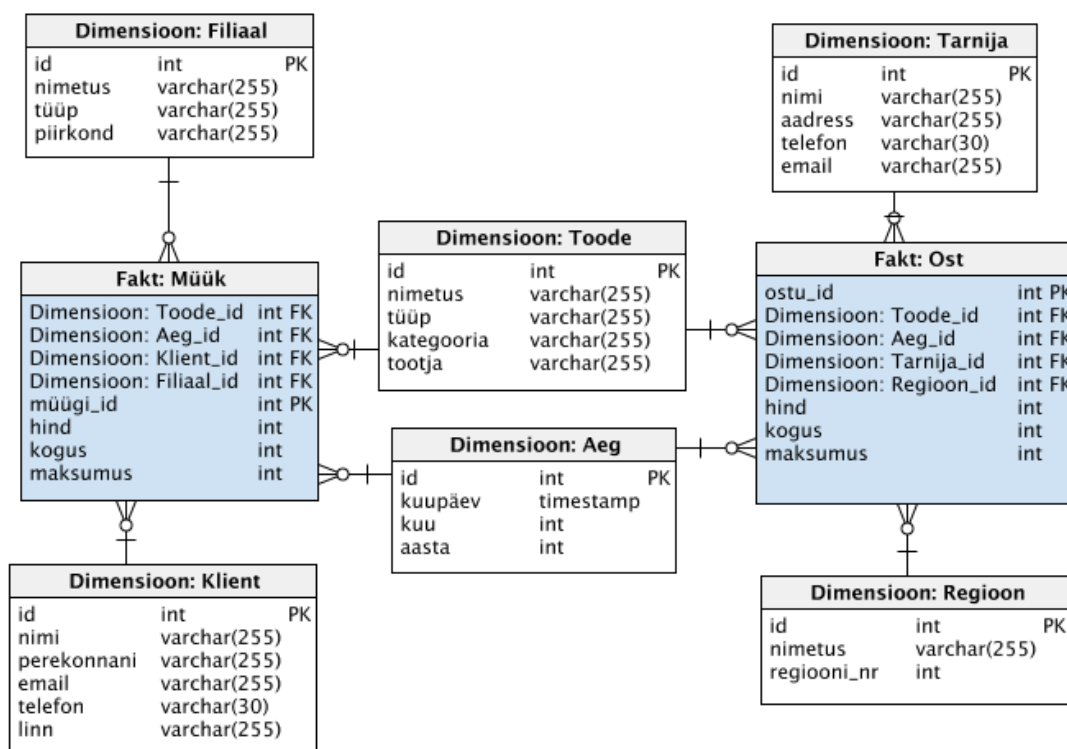
Joonis 4. Tähtskeem ehk *star schema*.

Lumehelbe skeemis (Joonis 5) dimensioonidel võivad olla alamdimensioonid. Selle eesmärgiks on vähendada andmete koondamist [3, p. 7].



Joonis 5. Lumehelbe skeem ehk *snowflake schema*.

Galaktika skeemis (Joonis 6) on kaks või rohkem faktitabelit, mis on seotud dimensioonide abil [3, p. 7].



Joonis 6. Galaktika skeem ehk *galaxy schema*.

Tähtskeemi eeliseks on see, et see on lihtsam kui lumehelbe ja galaktika skeem, mis võimaldab ETL protsessidele lihtsamini laadida andmeid dimensionaalsesse andmemudelisse. Lumehelbe skeemi eeliseks on see, et mõned analüütilised rakendused töötavad lumehelbe skeemiga paremini, võrreldes tähtskeemi või galaktika skeemiga. Teine lumehelbe skeemi eelis on, et see koondab vähem andmeid ja võtab vähem kettaruumi. Galaktika skeemi eeliseks on võime modelleerida äriprotsesse täpsemalt kasutades mitmeid faktitabeleid [3, p. 7].

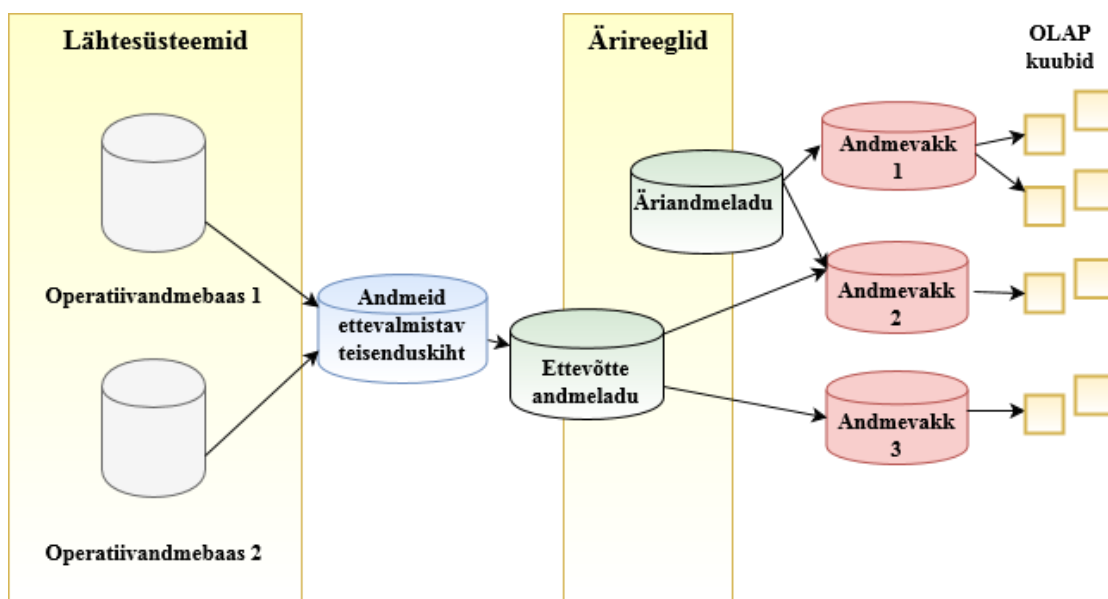
2.3.3 Dan Linstedti lähenemisviis

Dan Linstedt pakkus välja 2000ndate alguses kolmanda lähenemisviisi, *Data Vault Modeling* (Joonis 7). Tegemist on detailidele orienteeritud lähenemisega, mis võimaldab andmete ajaloo jälgimist ja unikaalselt ühendatud normaliseeritud tabelleid. Tabelid toetavad üht või mitut toimivat ärivaldkonda [7].

Linstedt leiab, et Inmoni kolmas normaalkuju (3NF) ega Kimballi dimensionaalne modelleerimine pole jätkusuutlikud, kui andmete maht suureneb. *Data Vault*'i

ülesehitus sõltub rohkem ärioloogikast kui andmestruktuuride muutustest. *Data Vault* võimaldab andmete paralleelset laadimist, kusjuures neid pole töödeldud ega filtreeritud; andmed ei muutu. Struktuurne ja kirjeldav informatsioon hoitakse andmemudelid eraldi, mistõttu on see paindlikum ja paremini kohanev äriprotsesside muudatustega [7].

Data Vault'i struktuur on kolmetasandiline, nii et toorandmeladu on lõppkasutajatest ja andmekaevandamise eri kihtidest eraldatud [7].

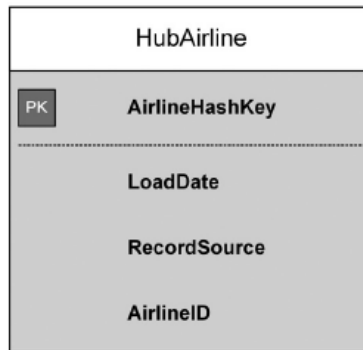


Joonis 7. Data Vault ettevõtte andmelao arhitektuur [7].

Data Vault kolm taset on andmeid ettevalmistav teisenduskiht, andmeladu ja andmevakad. Andmeid ettevalmistav teisenduskiht toetab andmete laadimist mitmest allikast; andmeladu, n-ö südamikukiht sisaldab eelmisest kihist pärit muudatusteta andmeid; andmevakad, n-ö esitluskiht, mis võimaldab OLAP (*Online Analytical Processing*) kuupide abil andmeid aruandlustes ja andmekaevanduses kasutada.

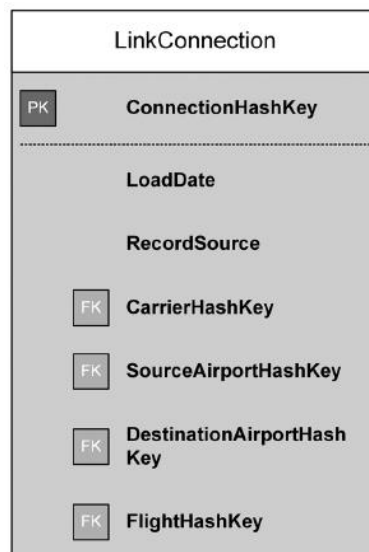
Data Vault mudelis on kolm põhilist olemit liiki:

Rummud (inglise keeles *hubs*) – äriprotsesside unikaalne ärivõtmete loend (Joonis 8). Eraldab ettevõtte ärivõtmed ülejäänud mudelist.



Joonis 8. *Example of Data Vault hubs (physical design)* [5, p. 101].

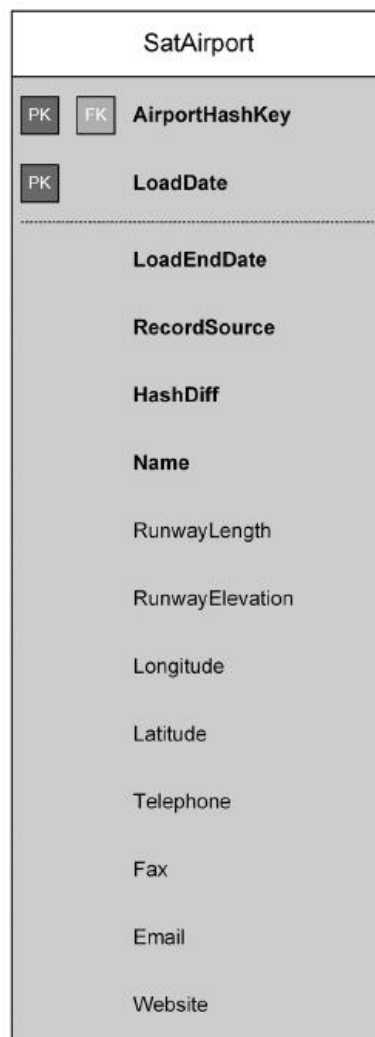
Kodarad (inglise keeles *links*) – ühendavad ärivõtmeid. Kodarad on modelleeritud rummude vahel. Need vastutavad tehingute modelleerimise, ühenduste, hierarhiate ja ärimõistete määratlemise eest. Joonisel 9 kodar LinkConnections ühendab kokku neli rummu: vedajat, lähtelennujaama, sihtlennujaama ja lennunumbrit. Lisaks kodar sisaldab vajalikke metaandmeid (andmeid andmete kohta) LoadDate ja RecordSource [5, p. 119].



Joonis 9. *Data Vault Link examples (physical design)* [5].

Sateliidid (inglise keeles *satellites*) – sisaldavad andmeid, mis kirjeldavad äriobjekti, seost või tehingut. Need annavad konteksti rummudele ja kodaratele. Kuna kontekst ettevõttes sageli muutub, siis satelliidi kirjeldavad andmed samuti muutuvad aja jooksul. Satelliit on ühendatud ainult ühe rummuga või kodaraga. Kirjeldavateks andmeteks on näiteks, inimese nimi ja perekonnanimi, vabade kohtade arv lennukis [5, p. 119]. Joonisel 10 SatAirpot sõltub HubAirportist läbi AirportSeq. Primaarne võti

koosneb AirportSeq ja LoadDate (kuupäev ja aeg, millal kirje oli andmelaos nähtud esimest korda) kombinatsioonist [5, pp. 126-127].



Joonis 10. Data Vault satellite examples (physical design) [5].

Lisaks, satelliidid kasutavad LoadEndDate ja RecordSource metaandmete atribuute. Selleks, et kiirendada satelliidi otsimist ridade muutuste tuvastamiseks, kasutab satelliit valikulist HashDiff atribuuti. Ülejäänud satelliit koosneb alliksüsteemist tulenevatest kirjeldavatest atribuutidest [5, pp. 126-127].

2.3.4 Andmelao disaini kontseptsioonide võrdlus

Nii Kimballi kui ka Inmoni mudelis on andmelaos kasutusel ETL protsess. Nii *Data Vault* kui ka Inmoni lähenemisviisi järgi on andmeladu kõige suurem hoidla ettevõttes. *Data Vault* ja Kimballi lähenemisviiside sarnasused on iteratiivne lahendus ning ettevalmistusala kasutamine andmete taastamiseks ja sünkroniseerimiseks [7, pp. 97-

98], [10, p. 15], [11]. Vaatamata lähenemisviiside sarnasustele on neil ka hulgaliselt erinevusi, mis kirjeldatakse alltoodud tabelites. Tabelid põhinevad allikatel [7], [10], [11], [12].

Järgmises tabelis (Tabel 1) tuuakse välja lähenemisviiside ning modelleerimistehnikate põhimõtteline võrdlus.

Tabel 1. Lähenemisviiside ja andmelao modelleerimistehnikate põhimõtteline võrdlus [7].

	Inmon	Kimball	Data Vault
Olulisemad kasutajad	IT-professionaalid	Lõppkasutajad	Lõppkasutajad
Eesmärgid	Täielik tehniline lahendus, mis põhineb tõestatud meetoditel ja tehnoloogiatel.	Lahendus, mis muudab andmete pärimise lõppkasutajatele lihtsamaks.	Kindel ja terviklik lahendus, mis põhineb tõestatud meetoditel.
Integreerimine erinevatest andmeallikatest	Andmete töötlemine ETL protsessis.	Andmete töötlemine ETL protsessis.	Satelliitide ja ärivõtmete eraldamine muudab protsessi lihtsamaks.
ETL protsessi keerukus	Teisendamise reeglid on lihtsad, kui andmemudel on sarnane andmeallika mudeliga.	Teisendamised OLTP (<i>Online Transaction Processing</i>) ja dimensionaalse mudeli vahel on keerulised.	Rummude, kodarate ja satelliitide laadimine on lihtsad.

Metodoloogiline ja arhitektuuriline võrdlus: Tabel 2 võrdleb kolme andmelao arenduse lähenemisviisi struktuuri, keerukust, kulukust ja andmelao arendamise aega [7].

Tabel 2. Metodoloogiline ja arhitektuuriline modelleerimise tehnikate võrdlus [7].

	Inmon	Kimball	Data Vault
Andmelao struktuur	Atomaarsel tasemel toidab andmeladu osakondade andmevakkasid.	Andmevakkade kogum moodustab andmelao.	Data Vault toetab andmevakkasid.
Arendusmeetod	Spiraalne arendamine.	Neljaastmeline protsess.	Agiilne arendusmeetodika.
Keerukus	Üsna keeruline.	Lihtne.	Lihtne.

Kulukus	Esialgssed kulud on suuremad, järgnevad arengukulud madalamad.	Algkulud on väiksemad, ent iga järgnev samm maksab rohkem.	Kogu projekti maksumus on väiksem kui teiste lähenemisviiside oma.
Juurutusaeg	Pikk	Lühike	Lühike

Andmete modelleerimine: Tabelis 3 võrreldakse andmete modelleerimist. Vaadeldakse andmetele orienteeritud tööriistu ja lõppkasutaja modelleerimisprotsessi kaasamist [7].

Tabel 3. Andmete modelleerimise võrdlus erinevates lähenemisviisides [7].

	Inmon	Kimball	Data Vault
Andmetele orienteeritus	Andmetele orienteeritud.	Protsessidele orienteeritud.	Protsessidele ja andmetele orienteeritud.
Tööriistad	Klassikalised modelleerimise tööriistad.	Dimensionaalne modelleerimine.	Rummude, kodarate ja satelliitide modelleerimine.
Lõppkasutaja kaasamine	Vähene.	Suur.	Suur.

Elutsükli juhtimine: Tabelis 4 on esitatud täiendavate kriteeriumide erinevused.

Tabel 4. Elutsükli juhtimise võrdlus erinevates modelleerimise tehnikates [7].

	Inmon	Kimball	Data Vault
Uute analüüsinõuete pandlikkus	Mudel muutub, kui andmelaos pole vajalikke andmeid.	Uued nõuded mõjutavad andmemudelit.	Mudelis pole vaja teha muudatusi. Tuleb kohandada laadimisloogikat andmevakkadesse.
Mudeli muutmise keerukus	Ajaloolised andmed tuleb mõnel juhul migreerida.	Mõnedel juhtudel tuleb mõned tabelid ümberstruktureerida.	Ainus muudatus on asjakohaste satelliitide lisamine.
Audit ja jälgitavus	Ajalooline teave salvestatakse iga uue kirje lisamisel.	Kasutab kontseptsiooni „aeglaselt muutuv dimensioon“ ajalooliste muutuste salvestamiseks.	Ajalooline teave salvestatakse lisades uusi kodaraid ja satelliite.

Päringujõudlus	Päringud on aeglased kolmanda normaalkuju andmete struktuuri tõttu.	Mudel on väga tõhus denormaliseeritud dimensioonide ja faktide pärimiseks.	Kõrge standardimine teeb päringud väga aeglaseks; analüüsimiseks ja aruandluseks tuleb kasutada dimensionaalseid andmevakkasid.
----------------	---	--	---

Esitatud võrdluste põhjal võib öelda, et igal meetodil on oma eelised ja puudused, ning ükski neist ei vasta täielikult kõikidele nõuetele.

Inmoni lähenemisviis sobib ettevõttele eelkõige juhul, kui analüüsi nõuded pole selgelt määratletud, või kui andmevakkasid on plaanis hakata kasutama erinevates BI süsteemides. Samuti peaksid selle kasutamiseks olema andmeallikate struktuurid suhteliselt stabiilsed.

Kimballi lähenemisviisi eelis on kõrge päringujõudlus; dimensionaalne mudel on kasutajatele arusaadav. Seda lähenemisviisi on soovitatav kasutada andmevakkade jaoks, kui nõuded neile on täpselt määratletud.

Data Vault on andmelao arendamiseks sobilik, kui kasutada tuleb mitut andmeallikat, mille struktuur regulaarselt muutub (agiilsetes projektides). *Data Vault* on parim valik juhul, kui ettevõttele on kõige olulisemad skaleeritavus, paindlikkus ja tootlikkus [7].

2.4 ETL protsess

ETL on lühend ingliskeelsetest sõnadest *Extract*, *Transform*, *Load*, mis tähendab hõiva, töötlemist, laadi [13]. See on andmete teisenduskiht, mis hõivab andmeid lähteallikast, töötleb ning seejärel laeb neid andmelattu.

ETL protsess täidab 3 ülesannet:

- *Extract* ehk hõiva – andmete kättesaamine erinevat liiki operatiivsüsteemidest. Tihti peale kasutatakse relatsioonilise andmebaase või isegi tavalisi faile (.txt, .xlsx, .docx) ja veebiteenuseid.

- *Transform* ehk töötle – andmete korrastamise samm. Sellel sammul teostatakse näiteks teisendamist, puhastamist, duplikaatide eemaldamist, andmete ühtlustamist ning uute väärtuste lisamist.
- *Load* ehk laadi – andmete laadimine andmelattu. Andmed peavad vastama andmelao reeglitele ja piirangutele, näiteks primaarvõtmed peavad olema unikaalsed. Selles etapis defineeritakse sihtpunkti ning määratakse kui tihti peaks laadimine toimuma.

ETL on hea tööriist suuremahuliste andmete liigutamiseks ja ülekandmiseks. See sobib olukorras, kui on tegemist keerukate reeglite ja teisendustega. ETL-i abil saab integreerida mitmeid andmekogumeid erinevat tüüpi allikatest.

2.5 Metaandmed

Üldlevinud definitsiooni järgi metaandmed on „andmed andmete kohta“. Ralph Kimball sõnastab oma raamatus „*The Data Warehouse Toolkit. Third Edition*“ metaandmed kui andmelao entsüklopeediat [8, p. 173], mis kirjeldab andmelao struktuuri, toiminguid ja sisu.

Metaandmeid on võimalik klassifitseerida kolmel viisil [14, pp. 115-117]:

- Tehnilised metaandmed - kirjeldavad andmestruktuure, tabeleid, indekseid, veerge, andmetüüpe, andmebaase, dimensioone ja mõõdikuid. ETL protsessides tehnilised metaandmed sisaldavad detaile andmete lähte- ja sihtallikatest ning teisendusest.
- Ärilised metaandmed - kirjeldavad andmelao kasutajasõbralikumal viisil ning on suunatud lõppkasutajatele. Need näitavad, mis andmed on andmelaos, kust pärinevad, mida tähendavad ja milline on seos teiste andmelao andmetega. Väljadel kuvatavad nimetused ja sisu kirjeldused on äriliste metaandmete põhinäide.
- Protsessi metaandmed - kirjeldavad erinevate operatsioonide tulemusi. ETL protsessis need salvestavad andmeid ülesannete täitmise kohta, näiteks käivituse algus- ja lõpuaeg ning kestus. Sarnased metaandmed genereeritakse kui kasutaja teeb andmelaos päringuid. Protsessi metaandmed kasutatakse andmelao

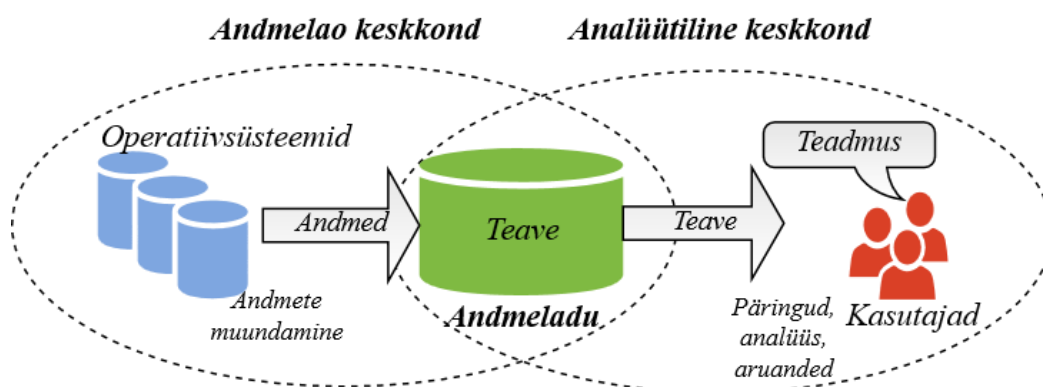
populaarsuse jälgimiseks ja turvalisuse eesmärgil. Need andmed aitavad leida vigu ETL protsessis või päringus.

Eelpool viidatud allikas väidetakse, et enamus arendajatest suhtuvad dokumentatsiooni vastumeelselt ning seepärast jäävad metaandmed tihtipeale projekti plaanist välja. Lõpuks siiski tunnistatakse, et see on oluline osa mis aitab kasutajatel mõista sisulist poolt.

2.6 Äriintelligentsus

Äriintelligentsus on rakenduste ja tehnoloogiate kogum [4, p. 18], mis aitab mõista äri olukordi tehes ettevõtte ja kolmandate osapoolte andmetega mitmesuguseid analüüse. See aitab kaasa strateegiliste, taktikaliste ja operatiivsete äriotsuste tegemisele ning vajalike meetmete võtmisele äri efektiivsuse tõstmiseks [3, p. 12]. Esiteks viitab see mõiste süsteemidele ja tehnoloogiatele, mis koguvad, puhastavad, konsolideerivad ja säilitavad ettevõtete andmed. Teiseks äriintelligentsus viitab tööriistadele, tehnikatele ja andmeanalüüsi rakendustele.

Andmeladu ja ärianalüütika lahendused täiendavad üksteist ja peavad töötama kooskõlas. Joonisel 11 on näidatud kaks keskkonda, andmeladu, mis teisendab andmed teabeks ja analüütika lahendus, mis loob tarbijale teabest teadmust [4, p. 18].



Joonis 11. Andmeladu ja analüütilised keskkonnad [4, p. 18].

Ärianalüütika lahenduseid võib jagada kolme kategooriasse [3, p. 13]:

- Aruandlus, näiteks tulemuslikkuse võtmenäitajad, nagu globaalsed müügiandmed äriüksuste ja teeninduskoodide kaupa.
- OLAP, nagu agregeerimine, üksikasjade läbivaatamine (inglise keeles *drill down*).
- Andmekaevandamine, nagu kirjeldav analüüs ehk klassifitseerimine, klasterdamine, mustrite analüüs, anomaaliade analüüs, visualiseerimine ja prognoosiv analüüs.

Andmelao aruanded kasutatakse äriteave esitamiseks, kuid neid samuti kasutatakse andmete haldamise eesmärgil, andmete kvaliteedi monitoorimiseks, andmelao rakenduste kasutamise ja ETL tegevuste jälgimiseks [3, p. 13].

Käesoleva töö raames arendatakse aruandeid andmete kvaliteedi monitoorimise eesmärgil.

3 Andmelao arenduse projekti riskid

Käesolevas peatükis antakse ülevaade andmelao arenduse projektiga kaasnevatest potentsiaalsetest riskidest ja nende põhjustest. Iga projektiga on seotud riske, kuid andmelao projektides riskid on tavaliselt suuremad kui teistes projektides.

Andmelao arenduse projektide ebaõnnestumise põhjused võivad olla järgmised [15]:

- Puudub toetav isik, kellel on ettevõttes mõjuvõim, et projektile prioriteete saada.
- Puudub ülevaade projekti mõjust ettevõttele.
- Projekti ulatus langeb kokku teiste arendustega.
- Ressursside puudumine.
- Projekt nõuab tööprotsesside muutmist.
- Tähtaja ületamine.
- Uus ja tundmatu meetoodika.
- Lähteandmete kättesaadavuse probleem.
- Lähteallika andmemudeli keerukus.
- Andmete arusaamise probleemid.

Varajane riskide avastamine ja planeerimine võivad oluliselt vähendada kahjulikku mõju projektile. Projekti alguses on tarvis teha riskianalüüs, mis aitab keskenduda olulisematele riskidele.

4 Andmelao lahenduse loomine meditsiinasutusele

Käesolevas peatükis tutvustatakse praktilist osa, mis seisneb meditsiinasutuse jaoks uue andmelao ja ärianalüütika lahenduse loomises. Projekti alguses viiakse läbi riskianalüüs ja selgitatakse välja nõuded. Antakse ülevaade protsessidest ja tööriistadest, tutvustatakse olemasolevaid andmeallikaid, selgitatakse andmelao vajalikkust ning tuginedes teooriale arendatakse välja ETL protsess, andmeladu ja aruandlusvahendite abil aruanded.

4.1 Andmelao eesmärk ja vajalikkus

Meditsiinasutusel on hetkel kasutusel andmeladu ja aruandluskeskkond, mida on arendatud ühe lähtesüsteemi, haigla infosüsteemi HIS põhjal. Süsteem on kliinilise info keskne. Asutusel on olemas ka personali infosüsteem. Mõlemad süsteemid sisaldavad meditsiinitöötajate tööaegade graafikute andmeid. Kuna hetkel liidestus kahe infosüsteemi vahel puudub, siis peetakse info vahetamiseks tugitöötajate ja klienditeeninduse vahel telefonikõnesid ja e-kirjavahetust. Andmeid sünkroniseeritakse käsitsi. Info liikumine on katkendlik ning sõltub igas protsessi ahelas inimtegevusest, mistõttu andmetes leidub palju vastuolusid.

Käesolevas töös arendatakse kahe infosüsteemide tarbeks välja alternatiivne andmeladu ja aruandluskeskkond olemasolevate vahendite abil. Alternatiivse andmelao ja ärianalüütika lahenduse eesmärk on tuvastada lähteallikate andmekvaliteedi probleeme andmete vastuolude seire abil, aitamaks kaasa nii andmekvaliteedi paranemisele kui ka tööprotsessi organiseerimisele, kuna erinevate lähtesüsteemide andmete kuvamine ühe vaadena tagab info parema liikumise ja motiveerib töötajaid sisestama õiged andmed õigeaegselt. Teisene eesmärk on loodava lahenduse abil avastada varjatud ebakõlasid töögraafikute moodulite kasutamisel.

Patsiendi jaoks on kõige olulisem probleem, kui meditsiinilise infosüsteemi ja personali infosüsteemi töögraafikud sattuvad vastuollu selliselt, et personali infosüsteemi järgi on

arst puhkusel, koolitusel või haige, ent kliinilise infosüsteemi järgi on arst töö. Patsiendid arvestavad selle infoga, mida pakub patsiendiportaal või registratuur.

Loodav lahendus aitab ära hoida suuri kulutusi liidese loomisele kahe erineva infosüsteemi vahel ning tuua teenuse ettevõtte haldusalasse, loomata seejuures uusi sõltuvusi arenduspartneritest. Olemasoleva kliinilise süsteemi andmelao arendamine ning hooldamine on asutusele kallis. ETL-i teenuse arendus haiglas on usaldatud arenduspartneri kätte. Kõik muudatused ETL-is vajavad kooskõlastust ka teiste partnerhaiglatega, mistõttu see on kulukas nii rahalises kui ka ajalisel mõttes. Kaaluti kahe infosüsteemi liidestamist, kuid see läheks kalliks ning on ka keerukas. Pidevalt muutuv loogika eeldab paratamatult inimkontrolli.

4.2 Andmelao ja ärianalüütika lahendusele esitatavad nõuded

Et andmelao/BI süsteem oleks edukas, peab äripool selle omaks võtma. Parimate toodete ja platvormidega loodud elegantselt lahendusest pole kasu, kui äriühing seda kasutama ei hakka. Erinevalt operatiivsüsteemidest, mille puhul kasutajal pole muud valikut, on BI süsteemi kasutamine vabatahtlik. Äripool kasutab meeeldi süsteemi teabeallikana, kui see on lihtne ja kiire.

Andmelao arendaja peab hästi teadma äripoole vajadusi ning suutma neid tehniliselt realiseerida. Antud projekti nõuete määratlemiseks tegid andmelao arendaja ja andmelaost huvitatud isikud koostööd ja kirjeldasid *funktsionaalseid nõudeid* järgmiselt:

- leida vasturääkivusi kahe infosüsteemi ambulatoorsete tööajagraafikute andmetes;
- leida, mis tüüpi vead esinevad ambulatoorsetes tööajagraafikutes;
- leida, mis kuupäevadel ja kellaaegadel vead esinevad;
- leida ambulatoorsete tööajagraafikute vead töötajate lõikes.

Järgnevalt esitatakse süsteemile *mittefunktsionaalsed nõuded* lähtuvalt tarkvara kvaliteedinäitajatest, mis on esitatud rahvusvahelises ISO/IEC 25000 standardiseerias [16], [17]. Samuti lähtutakse Ralph Kimballi raamatus „*The Data Warehouse Toolkit. Third Edition*“ andmelaole kirjeldavatest nõuetest [8]:

- *Funktsionaalne sobivus.* Süsteem peab täitma funktsionaalsetes nõuetes välja toodud ärikasutajate eesmärged. Andmed peavad olema usaldusväärsed ehk hoolikalt kogutud mitmesugustest allikatest, korrastatud ja kvaliteetsed. Süsteem peab andmeid päringu vastusena laadima vaid siis, kui need sobivad kasutamiseks.
- *Soorituse tõhusus.* Päringud peavad saama vastuseid maksimaalselt 1 minutiga.
- *Töökindlus.* Andmelao/ BI süsteemi andmete laadimine peab toimuma öösiti kell 1:00, et mitte koormata süsteeme päevasel ajal. Andmelao/BI süsteemid peavad andma teavet õigeaegselt (igal hommikul värsked andmed).
- *Kasutatavus.* Andmed peavad olema ärikasutajate jaoks selged: kui kaks tulemuslikkuse mõõdet on sama nimega, siis peab neil olema ka sama tähendus. BI süsteem peab olema eestikeelne.
- *Turvalisus.* Andmelao/BI süsteem peab olema turvaline ja tõhusalt kontrollima juurdepääsu konfidentsiaalsele teabele. Süsteemi ei saa siseneda paroolita, ning tööajagraafikute vaatamisõigused on graafikute koostajatel, juhatajatel ja vanemklienditeenindajatel. Päringute kirjutamise õigused on andmeanalüütikutel.
- *Hooldatavus.* Andmelao/BI peab kohanema muutustega. Kasutajate vajadused, ärinõuded, andmed ja tehnoloogia võivad muutuda. Kui äripool küsib uusi küsimusi või andmelaole lisatakse uusi andmeid, ei tohi olemasolevaid andmeid ega rakendusi muuta.

Andmelao ülesehitus sõltub äripoole nõuetest. Sageli luuakse andmelaod kliendi nõudeid ja vajadusi täielikult mõistmata, ning seetõttu projektid ebaõnnestuvad. Niisiis tuleb loodav süsteem välja arendada lähtudes organisatsiooni vajadustest, analüüsist ja parimast andmelao arendamise praktikast. Lahendus peaks olema jätkusuutlik ning dokumenteeritud.

4.3 Andmelao arendamise projekti riskianalüüs

Selleks, et andmelao projekt oleks edukas on vajalik enne projekti tegevustega alustamist kaardistada riske. Antud projektis oli tehtud kvalitatiivne riskianalüüs [18]. Analüüsi käigus identifitseeriti riske, hinnati nende mõju ja tõenäosust ning pakuti lahendusi (Tabel 5).

Tabel 5. Riskianalüüs.

ID	Riski kirjeldus	Tõenäosus	Riski mõju	Rakendatavad abinõud
R1	Puudub toetav isik, kellel on ettevõttes mõjuvõim, et projektile prioriteete saada.	Keskmine	Keskmine	Leida isik, kes on projektist huvitatud ning hoida suhet projekti vältel.
R2	Puudub ülevaade projekti mõjust ettevõttele.	Keskmine	Keskmine	Huvitatud isikutele tuleb anda ülevaade projekti väärtustest.
R3	Projekti ulatus langeb kokku teiste arendustega.	Keskmine	Väike	Projekti planeerimisel selgitada välja, milliseid arendusi kavandatakse.
R4	Ressursside puudumine.	Suur	Suur	Jagada ülesandeid personali vahel.
R5	Projekt nõuab tööprotsesside muutmist.	Suur	Keskmine	Vestelda töötajatega ja selgitada neile projekti kasulikkust.
R6	Tähtaja ületamine.	Suur	Keskmine	Lisada projektiplaani olulisematele töödele reservajad.
R7	Uus ja tundmatu meetodika.	Suur	Suur	Konsulteerida välisspetsialistidega.
R8	Lähteandmete kättesaadavuse probleem.	Keskmine	Suur	Planeerimisel selgitada välja, kas on võimalik luua ühendused andmebaasidega.
R9	Lähteallika andmemudeli keerukus.	Suur	Suur	Tellida tarkvara arendajatelt vaated ja vaadete kirjeldused.
R10	Andmete arusaadavuse probleemid.	Suur	Suur	Konsulteerida töötajatega, kes kasutavad lähteallika süsteemi.

Seejärel koostati tõenäosuse ja mõju maatriks (Tabel 6). Tabeli veergudes on toodud mõju suurused ja ridades tõenäosused. See meetod aitab välja selgitada, millised riskid on kõige olulisemad.

Tabel 6. Riskimaatriks.

Riski realiseerumise tõenäosus	Riski mõju		
	Väike	Keskmine	Suur
Suur		R5, R6	R4, R7, R9, R10
Keskmine	R3	R1, R2	R8
Väike			

Kõige olulisemateks riskideks osutusid ressursside puudumine, uus ja tundmatu tehnoloogia, lähteallika andmemudeli keerukus ja andmete arusaadavuse probleemid. Riskide vältimiseks töö autor suhtles asutuse töötajatega, kelle tööülesanded on seotud tööajagraafikutega ning nende haldamisega ja tehnilistes küsimustes konsulteeris välisspetsialistidega.

4.4 Tööaegade graafikute koostamise ja muutmise protsesside analüüs

Pärast vastava andmelao ja andmeanalüüsi vahendite juurutamist, peab info liikuma operatiivsemalt ja veatumalt kui seni. Selleks kirjeldatakse tööaegade graafikute koostamise ja muutmise protsessi, võrreldakse praegust olukorda ning tuleviku olukorda AS-IS ja TO-BE mudeli abil, mis võimaldavad hinnata pakutud lahenduse efektiivsust. Viiakse läbi protsessi simuleerimise *Bizagi* töövahendis ning tuuakse välja ajalise võidu protsesside vahel.

4.4.1 Hetkeseis – AS-IS

Töögraafik on vanemklienditeenindaja poolt haigla infosüsteemi sisestatud iga meditsiinitöötaja ambulatoorsete vastuvõtude ning uuringute/protseduuride ajakava päevade ja kellaegade lõikes. Vastuvõtte/uuringuid/protseduure teostava keskuse/ osakonna juhataja ja õendusjuht vastutavad graafikute õigeaegse koostamise ning sellega seonduva informatsiooni õigsuse eest.

Meditsiinitöötajad kooskõlastavad oma tööajad osakondade/kliinikute juhatajatega ning edastavad seejärel graafikute koostajatele (näiteks, kliinikute assistentidele). Graafikuid avavad HIS-is kliinikutest/osakondadest saadud informatsiooni alusel vanemklienditeenindajad, kes vastutavad tööaegade õigeaegse ja korrektse sisestamise eest haigla infosüsteemi.

Vastuvõttude ja uuringute/protseduuride graafikud ning neis kavandatavad muudatused peavad olema edastatud klienditeenindusosakonnale vähemalt 4 kalendrikuud ette. Kinnitatud puhkusegraafikud ja puhkuste ning koolituste ajad ja muud vastuvõtte mõjutavad muudatused edastab graafiku koostaja (kliiniku assistent) e-posti teel Exceli tabeli kujul, kus muudatused on sisse viidud värvilistena. Muudatusi saab sisse viia päevast, mil vastuvõtu või uuringu/protseduuri aegu ei ole veel haigla infosüsteemis avatud. Avatud ja broneeritud vastuvõtu tühistamine peab olema põhjendatud. Broneeritud vastuvõtu tühistamise või muutmise vältimiseks tuleb keskuse/osakonna juhatajal/õendusjuhil leida asendustöötaja. Vajaduse korral tuleb klienditeenindusosakonnale teatada uued vastuvõtu- ja protseduuri/uuringu ajad, millal broneeritud patsiendid vastu võetakse.

Peale muudatuse/tühistamise sisestamist haigla infosüsteemi saadab vanemklienditeenindaja e-posti teel vastuse muudatuse/tühistamise algatanud töötajale.

Vastuvõttude ja uuringute/protseduuride graafikutes tuleb vanemklienditeenindaja poolt tühistatud ajad tähistada järgmiselt: haigus, koolitus, valve, puhkus, koosolek, remont, lähetus, muu. Kui graafiku muutmine/tühistamine on keskuse/osakonna juhataja/õendusjuhi poolt kooskõlastatud, siis on meditsiinitöötaja kohustatud kontrollima, et muudatus kajastub tema graafikus.

Personali infosüsteemi sisestatakse graafikud kas kohe pärast klienditeenindusse Exceli andmefaili saatmist või jääb sinna vahele ajanihe, mis võib ulatuda kuni 3-4 kuuni.

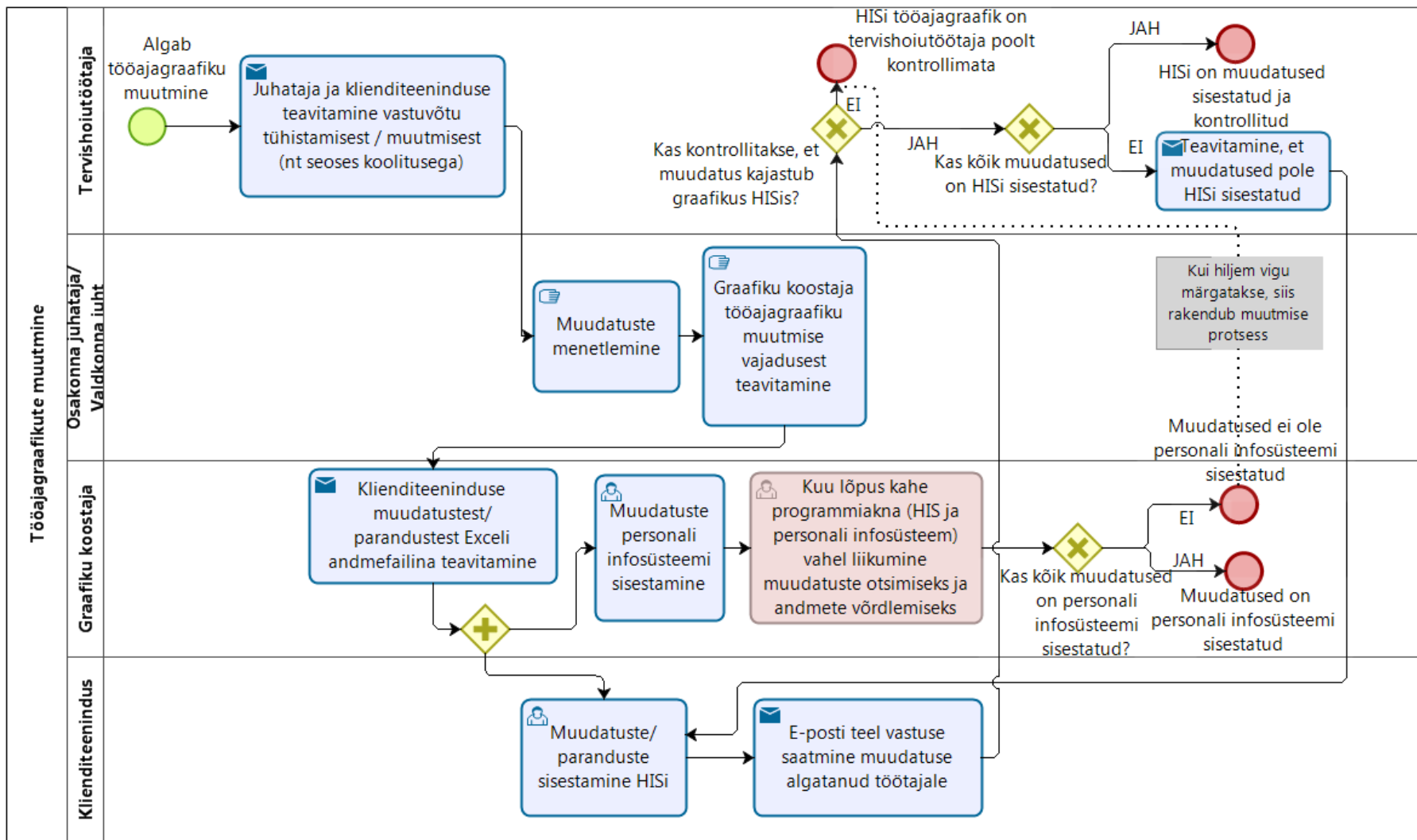
Vastuolud kahes infosüsteemis tekivad siis, kui töötaja ei saada oma muudatuse kohta teavitust või ei kontrolli, kas kõik muudatused on haigla infosüsteemis ka kajastatud. Probleem võib samuti tekkida ka siis, kui graafiku koostaja sisestab/avastab muudatusi liiga hilja, või kui graafikus tehtud muudatused kinnitatakse kontrollimata, kas antud ajavahemikul võib muudatusi teha või mitte.

Joonisel 12 esitatakse ülevaade praeguse graafikute muutmise tööprotsessist, kasutades *Bizagi Modeler* [19] töövahendit. Olemasoleva protsessi puuduseks on see, et graafikute muutmisel ei muudeta kohe ka personali infosüsteemi sisestatud aegu. Ka võidakse graafiku koostajat teavitada muudatusest hiljem või ei teavitata üldse. Vastuolud võivad tekkida ka vastupidi, st muudatused sisestatakse küll personali infosüsteemi, aga klienditeenindust neist ei teavitata. Kuu lõpus graafiku koostaja otsib muudatusi/ kahe programmi vahel erinevusi ning kulutab selleks väga palju kallist tööaega. Joonisel 12 on punasega märgistatud tegevus, mis on plaanis uue lahenduse abil parandada.

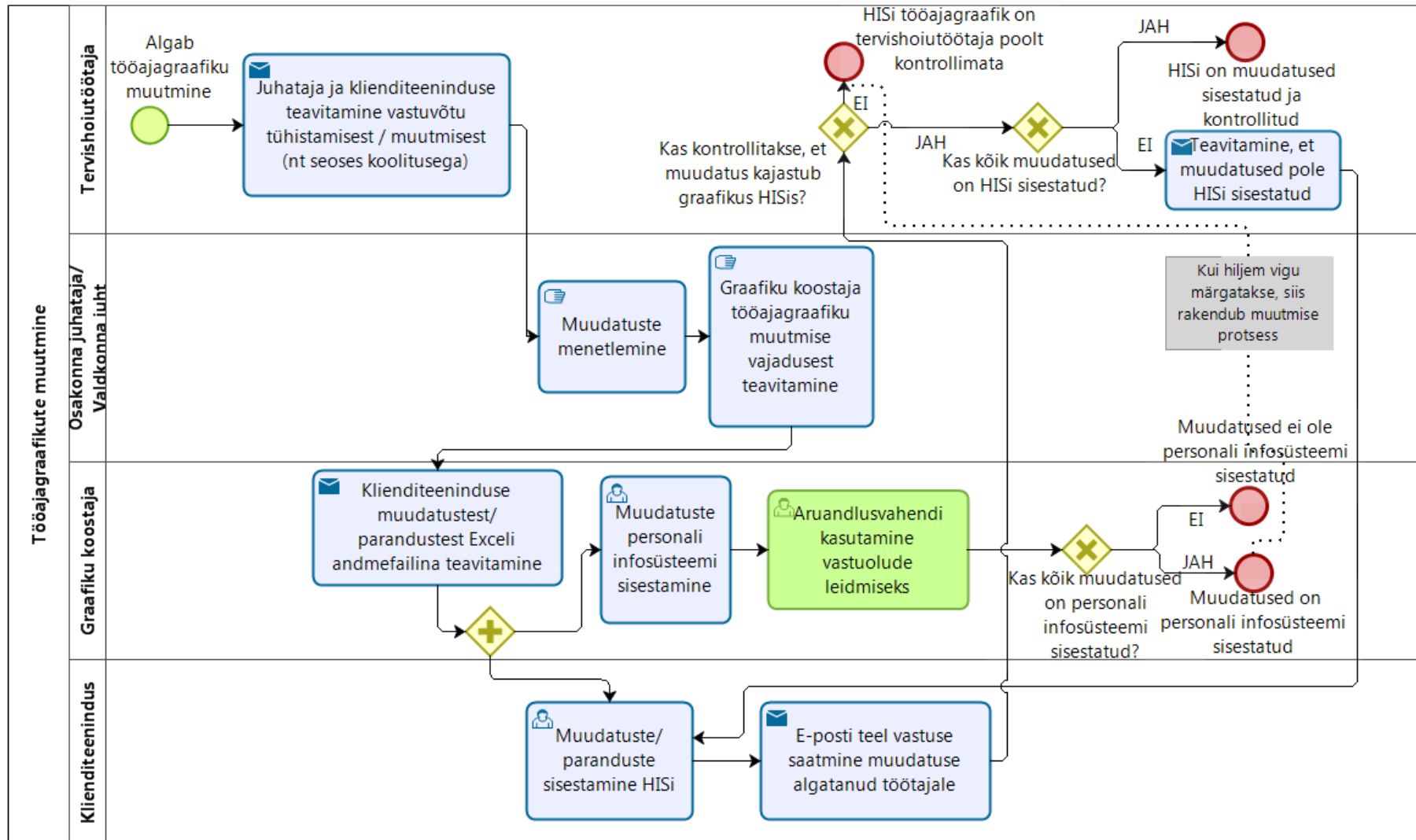
4.4.2 Tulevikuvaade – TO-BE

Eelmises alapeatükis toodi välja praegu kasutatava tööprotsessi puudused. Käesolevas alapeatükis antakse ülevaade sellest, kuhu soovitakse jõuda pärast andmelao ja ärianalüütika vahendi juurutamist. Joonisel 13 esitatakse ülevaade graafikute muutmise tööprotsessidest tulevikus. Rohelisega tähistatud tegevus aitaks parandada hetkeolukorda.

Lahendus on andmelao ja aruannete komplekti loomine, et tuvastada kahe infosüsteemi andmekvaliteedi probleeme andmete vastuolude seire abil. Tänu uuele lahendusele ei pea graafiku koostaja enam otsima muudatusi ega kulutama selleks kallist tööaega, kuna erinevate lähtesüsteemide andmete kuvamine ühe vaadena tagab info parema liikumise ja aitab vabaneda kahe programmi (personali infosüsteem ja haigla infosüsteem) vahel liikumisest selleks, et võrrelda andmeid. Aruanne näitab, kus esineb vastuolu ja aitab kiiremini vältida olukordi, kus personali infosüsteemi järgi on arst puhkusel, koolitusel või haige, aga kliinilise infosüsteemi järgi on arst tööl, ning patsiendid arvestavad selle infoga, mida pakub patsiendiportaal või registratuur. Ka aitab aruanne tagada, et kuu lõpul andmed kahes infosüsteemis kooskõlas oleksid. Aruanne motiveerib töötajaid teavitama ning kontrollima või sisestama andmeid nii personali infosüsteemi kui ka haigla infosüsteemi.



Joonis 12. AS-IS tööajagraafiku muutmise protsess.



Joonis 13. TO-BE tööajagraafiku muutmise protsess.

4.4.3 Tööajagraafiku muutmise protsessi simuleerimine

Pärast protsessi kaardistamist viidi läbi graafiku muutmise protsessi simulatsioon *Bizagi* töövahendis.

Töögraafikute käsitlemise suur miinus on ajamahukus. Ühe juhtumi käsitlemine alates muutmisest teavitamisest kuni muudatuste sisestamiseni ja sisendmaterjalide võrdlemiseni võttis keskmiselt 35 minutit (Joonis 14). TO-BE simulatsioonis (Joonis 15) vähenes selleks kulunud aeg 23 minutini. See tähendab, et suure hulga graafikute parandamise korral ulatub ajaline võit kümnete ja sadade töötundideni.

Name	Type	Instances completed	Instances started	Min. time (m)	Max. time (m)	Avg. time (m)	Total time (m)
Tööajagraafikute muutmine	Process	1000	1000	35	46	35.022	42564
Muudatuste menetlemine	Task	1000	1000	5	5	5	5000
Kas kõik muudatused on HISi sisestatud?	Gateway	934	934				
E-posti teel vastuse saatmine muudatuse algatanud töötajale	Task	1047	1047	2	2	2	2094
HISi on muudatused sisestatud ja kontrollitud	End event	887					
Muudatused on personali infosüsteemi sisestatud	End event	937					
Juhataja ja klienditeeninduse teavitamine vastuvõtu tühistamisest / muutmisest (nt seoses koolitusega)	Task	1000	1000	2	2	2	2000
Muudatuste/ paranduste sisestamine HISi	Task	1047	1047	5	5	5	5235
Kas kõik muudatused on personali infosüsteemi sisestatud?	Gateway	1000	1000				
Klienditeeninduse muudatustest/ parandustest Exceli andmefailina teavitamine	Task	1000	1000	5	5	5	5000
Muudatused ei ole personali infosüsteemi sisestatud	End event	63					
Algab tööajagraafiku muutmine	Start event	1000					
Teavitamine, et muudatused pole HISi sisestatud	Task	47	47	5	5	5	235
Kas kontrollitakse, et muudatus kajastub graafikus HISis?	Gateway	1047	1047				
Muudatuste personali infosüsteemi sisestamine	Task	1000	1000	5	5	5	5000
Kuu lõpus kahe programmiakna (HIS ja personali infosüsteem) vahel liikumine muudatuste otsimiseks ja andmete võrdlemiseks	Task	1000	1000	15	15	15	15000
Graafiku koostaja tööajagraafiku muutmise vajadusest teavitamine	Task	1000	1000	3	3	3	3000
HISi tööajagraafik on tervishoiutöötaja poolt kontrollimata	End event	113					
ParallelGateway	Gateway	1000	1000				

Joonis 14. AS-IS protsessi simulatsiooni tulemused.

Samuti vähenes lähteallikate vastuolude arv, kuna aruanded aitasid neid leida. AS-IS simulatsioonis tehtud muudatused kliinilises infosüsteemis polnud personali infosüsteemi sisestatud 63 korral, TO-BE protsessis vaid 14 korral.

Name	Type	Instances completed	Instances started	Min. time (m)	Max. time (m)	Avg. time (m)	Total time (m)
Tööajagraafikute muutmine	Process	1000	1000	23	46	23.652	30708
Muudatused on personali infosüsteemi sisestatud	End event	986					
Teavitamine, et muudatused pole HISi sisestatud	Task	59	59	5	5	5	295
Juhataja ja klienditeeninduse teavitamine vastuvõtu tühistamisest / muutmisest (nt seoses koolitusega)	Task	1000	1000	2	2	2	2000
ParallelGateway	Gateway	1000	1000				
E-posti teel vastuse saatmine muudatuse algatanud töötajale	Task	1059	1059	2	2	2	2118
Kas kontrollitakse, et muudatus kajastub graafikus HISis?	Gateway	1059	1059				
Muudatuste personali infosüsteemi sisestamine	Task	1000	1000	5	5	5	5000
Klienditeeninduse muudatustest/ parandustest Exceli andmefailina teavitamine	Task	1000	1000	5	5	5	5000
Muudatuste/ paranduste sisestamine HISi	Task	1059	1059	5	5	5	5295
Algab tööajagraafiku muutmine	Start event	1000					
HISi tööajagraafik on tervishoiutöötaja poolt kontrollimata	End event	106					
Graafiku koostaja tööajagraafiku muutmise vajadusest teavitamine	Task	1000	1000	3	3	3	3000
HISi on muudatused sisestatud ja kontrollitud	End event	894					
Muudatuste menetlemine	Task	1000	1000	5	5	5	5000
Kas kõik muudatused on HISi sisestatud?	Gateway	953	953				
Kas kõik muudatused on personali infosüsteemi sisestatud?	Gateway	1000	1000				
Muudatused ei ole personali infosüsteemi sisestatud	End event	14					
Aruandlusvahendi kasutamine vastuolude leidmiseks	Task	1000	1000	3	3	3	3000

Joonis 15. TO-BE protsessi simulatsiooni tulemused.

Tulemusena saab välja tuua aruannete kasutamise ja kaasneva ajavõidu ja lähteallikate vastuolude arvu vähenemine.

4.5 Personali- ja haigla infosüsteemide sarnasused ning eripärad

Arendatava andmelao lähteallikad on personali infosüsteem ja HIS. Personali infosüsteem on tarkvara, mis hõlmab personali ja palgaarvestust, tööajatabelit, koolituskeskust ja e-koolituskeskust. HIS on kliiniline infosüsteem, mis samuti sisaldab töötajate vastuvõtuaegade graafikuid ja tööajatabeleid.

Personali- ja haigla infosüsteemide tööajakavade ülesehitus ja eesmärk on erinevad. Personali infosüsteemi graafiku moodulit kasutatakse puhkuste, koolituste ja haiguste pidamiseks. Selle alusel tehakse järgmise kuu alguses kuulõpu palgaarvestus.

Haigla infosüsteemide graafikud on loodud patsientide jaoks. Neli kuud ette avatakse arstide ambulatoorsete vastuvõttude ajaaknad, kuhu võib patsientidele aegu broneerida. Tervishoiutöötajate koolituste, puhkuste, haigestumise või muudel põhjustel muudetavad graafikuajad on blokeeritud.

Personali infosüsteemis on tööajakavasid võimalik planeerida ja jälgida, kombineerides ametikohtade erinevusi. Üks inimene võib töötada mitmes struktuuriüksuses ja erinevatel ametikohtadel. Sel juhul on tal infosüsteemis mitu tööajatabelit, kus ühes tööajatabelis on märgitud puhkus, teises tööajad. Kui inimene töötab nii ambulatoorses kui ka statsionaarses osakonnas, siis avatakse HIS-is graafik ainult ambulatoorse töö jaoks. Seetõttu saab võrrelda ainult ambulatoorset tööd kajastavaid graafikuid. Personali infosüsteemis ei märgita eraldi, kas tööajatabel on seotud ambulatoorse või statsionaarse osakonnaga. Kahe infosüsteemi osakondade nimetused on erinevad. Võrreldavad on keskused, kuid üks keskus võib hõlmata nii ambulatoorset kui ka statsionaarset osakonda.

Käesolevas töös analüüsitava kahe infosüsteemi ühenduslüliks on töötajate isikukoodid ning võrdlemise aluseks on võetud HIS-i ambulatoorsed tööajad. Andmelao laadimise käigus kontrollitakse päringute abil, kas antud ajavahemikud on personali infosüsteemis blokeeritud või mitte. Kui graafikuajad on blokeeritud mõlemas infosüsteemis, siis aruanne näitab, et vastuolu pole. Andmete vasturääkivuse korral annab aruanne teada, mis tüüpi veaga on tegemist.

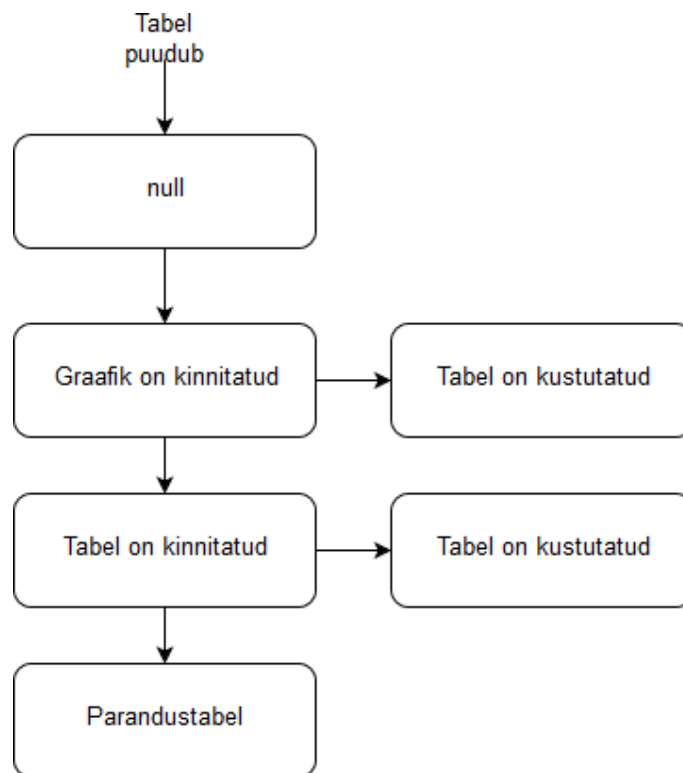
4.6 Lähteallikate andmemudelite analüüs

Personali infosüsteemi andmed paiknevad keerukas ja mahukas *Oracle* relatsioonilises andmebaasis ning neid haldab arenduspartner. Seetõttu olid personali infosüsteemi arendajate poolt loodud vaated (inglise keeles *SQL view*) tööajatabeli ja personalimooduli andmetega, kus andmed uuenevad kord päevas. Kokku oli loodud neli vaadet:

- TOOTAJA_ANDMED – töötajate andmed karjäärile lõikes. Üks töötaja võib samal ajal töötada mitmes osakonnas ja täita erinevaid tööülesandeid. Töötajal

võib töötamise ajal muutuda ametikoht, ametikoha nimetus, struktuuriüksus või lepingu vorm. Veerg KARJAARI_ID on tabeli unikaalne primaarvõti.

- TOOAJATABEL – tööajatabel põhiandmetega kuude lõikes. Tööajatabelil on erinevad staatused, mis on esitatud Joonisel 16 ja mida kirjeldatakse allpool. AJATABEL_ID on tabeli unikaalne primaarvõti.



Joonis 16. Personali infosüsteemi tööajatabeli staatused.

Tabel puudub (kirje puudub tööajatabelis) – tabel ei ilmu vaatesse, kui tööajatabeli koostaja ei ole seda kordagi avanud.

Null – tabel on staatuses „null“, kui tööajatabel on avatud ja sellega töötatakse, kuni toimub tööajagraafiku ja töötajatabeli kinnitamine, st graafik on veel kinnitamata. Puhkused on sees automaatselt niipea, kui töötaja need sisse kannab. Esimesel korral värskendatakse tööajatabelit alati automaatselt, kui tööajatabeli pidaja selle avab, ehk siis küsitakse esimest sissekannet koolituste ja puhkuste kohta. Teisel korral seda enam ei tehta, siis teeb seda tabeli pidaja ise manuaalse päringuga.

Graafik on kinnitatud – üks soovituslik vahestaatus, mida ei pea alati kasutama. Seda soovitatakse kasutada, kui tööajatabel on valmis, ja kuu keskel (vähemalt 5 päeva enne

järgneva kuu algust) antakse töötajatele järgmise kuu tööajagraafik (personali infosüsteemist tehakse väljatrukid, esitatakse struktuuriüksuse juhile allkirjastamiseks ja tutvustatakse töötajatele hiljemalt 5 päeva enne kuu algust). Seadus nõuab, et töötaja teaks umbes arvestada, millal ta on tööl ja millal vaba. Siin on ka soovitus vajutada nupule „Kinnita graafik“. Tööajatabelist luuakse 1.0 versioon, mis jääb *read-only* versioonina kättesaadavaks juhiks, kui tahetakse hiljem võrrelda, mis oli alguses plaanis ja mis nüüd. Graafik on aluseks tööajatablelile personali infosüsteemis. Graafikud on: tööajagraafik, valvegraafik, väljakutsete valvegraafik.

Tabel on kinnitatud – pärast graafiku kinnitamist teeb struktuuriüksuse juht või graafiku koostaja tööajatabelisse, valvegraafikusse või väljakutsete valvegraafikusse jooksvalt või kord kuu lõpul vajalikud muudatused (haigused, töölt puudumised, valvekordade vahetamised, koolitused, lähetused). Kuu lõpul kontrollib graafiku koostaja tööajatabeli, valvegraafiku ja väljakutsete valvegraafiku vastavust töötaja tegelikult töötatud ajale, lisab väljakutsete valvegraafiku olemasolul graafikusse väljakutsetunnid ning arvestuskuule järgneva kuu esimesel kuupäeval (kui struktuuriüksusel on väljakutsete valvegraafik) või arvestuskuu viimasel kuupäeval (kui struktuuriüksusel ei ole väljakutsete valvegraafikut) kinnitab tööajatabeli ehk luuakse 2.0 versioon. Selle alusel makstakse palka. Palgapäevaks peavad kõik tööajatabelid olema kinnitatud. Graafiku kinnitus ja tööajatabeli kinnitus võivad tekkida ka samal päeval. See sõltub tabeli pidajast, kuid tabelit kinnitada ilma graafiku kinnitusest ei saa. Juhul kui tööajatabelis on vaja teha muudatusi perioodil peale kinnitamist kuni palgapäevani töötaja haiguse, lähetuse või muu põhjuse tõttu, teavitab graafiku koostaja e-maili teel raamatupidamisosakonda ning raamatupidaja võtab kinnitused tabelist maha.

Tabel on kustutatud – graafiku ja tööajatabeli kinnitust saab tabeli pidaja maha võtta/kustutada, kuni sellel ei ole kõrgema tasemel kinnitust.

Parandustabel (3.0 versioon tööajatabelist) – tabel, mis luuakse erandkorras pärast tabeli kinnitamist (kui 2.0 versioon on tehtud ehk töötajatabel on kinnitatud) ja pärast palgapäeva, juhul kui tuleb teha muudatusi töötaja haiguse, lähetuse, koolituse või mõne muu põhjuse tõttu. See tabel luuakse lisaks kinnitatud tööajatablelile ning sisaldab ainult neid töötajaid puudutavaid parandusi, kelle puhul oli kinnitatud tabelis vaja muudatus teha. Parandustabel muudab vastava kuu palgaarvestust.

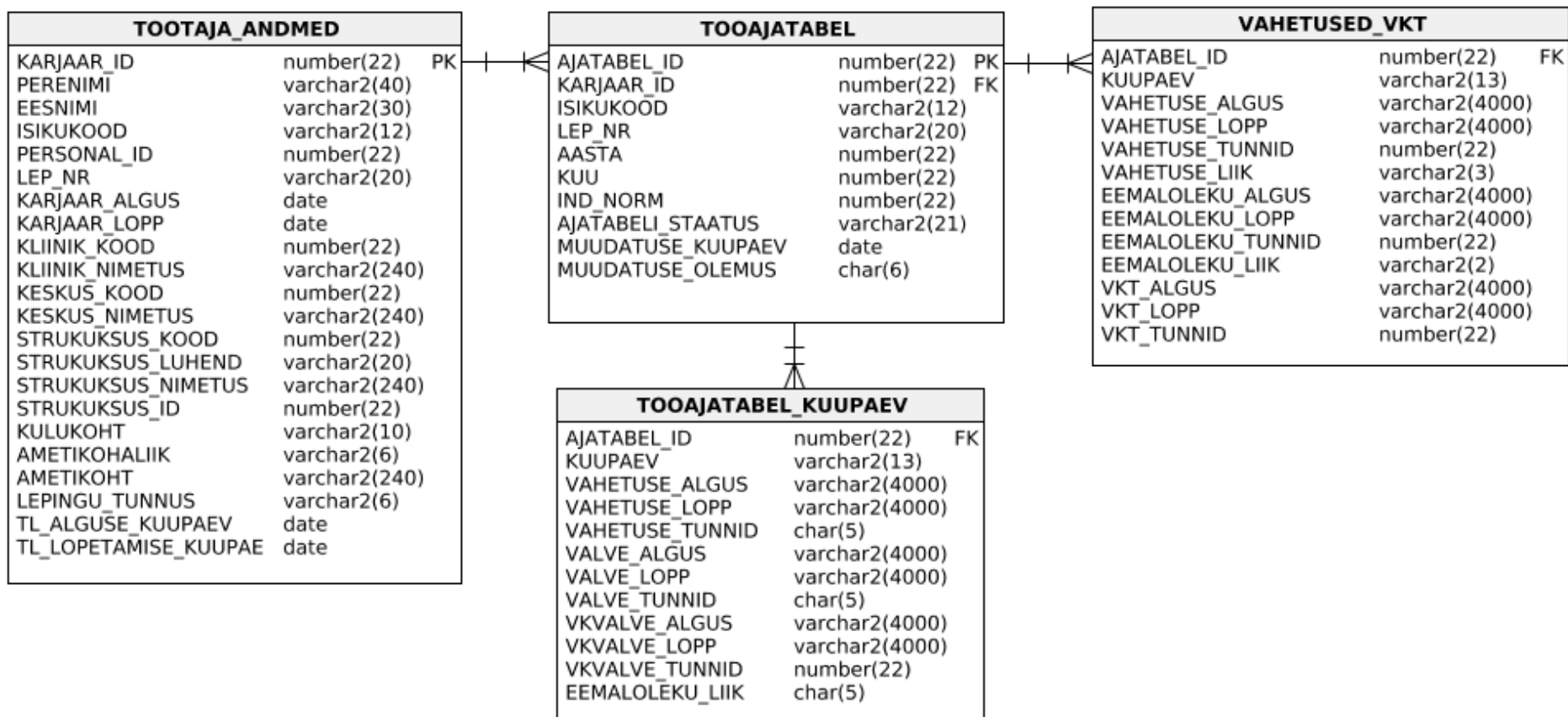
- TOOAJATABEL_KUUPAEV – tööajatabel kuupäevade lõikes, kus üks kuupäev kajastub ühekordselt. Näitab töötaja vahetuse, valvekorra või väljakutse kestust. Arvestatakse kõige varasemat tööpäeva algust ja kõige hilisemat tööpäeva lõppu ning tööaeg summeeritakse. Täpne töötundide kirjeldus on tabelis VAHETUSED_VKT. Töötaja puudumise korral märgitakse eemaloleku liik koodina (K - koolitus, P - korraline puhkus, AL - lähetus, ÕP - õppepuhkus, PP - puhkepäevad ja riigipühad, RP - palgata puhkus, VK - koolitus samal perioodil teises struktuuriüksuses kestusega alates 8h, H - haigusleht, jne).
- VAHETUSED_VKT – töötajate vahetused ning andmed lisatabelist ja väljakutsete tundide tabelist ehk need andmed, mis võivad ühe päeva kohta olla mitmerealised. Lisatabelist kajastub ainult koolitus kestusega vähem kui 8 tundi.

Enne tabelite käivitamist on vaja käivitada skript (Joonis 17), kus *p_kuu* ja *p_aasta* on tööajatabeli kuu ja aasta parameetrid, mida tuleb muuta käsitsi enne skripti käivitamist:

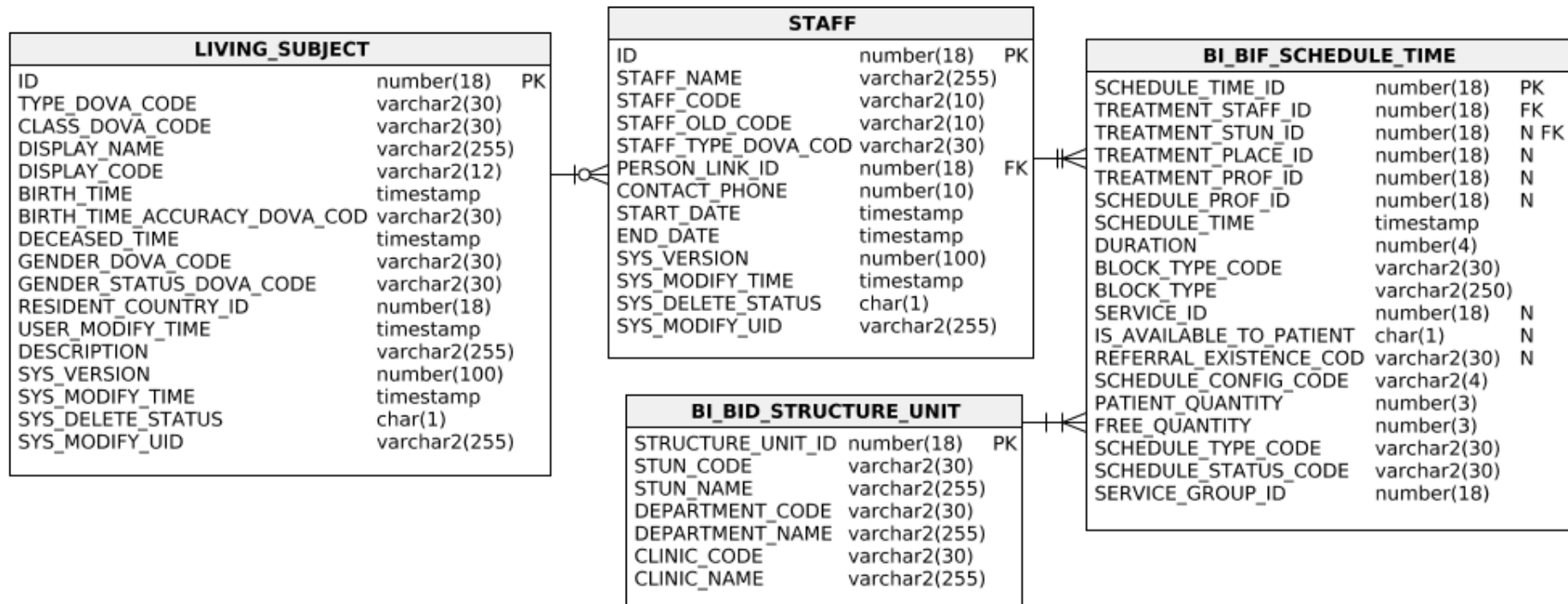
```
begin
view_parameters_pkg.set_mon_year(p_kuu,p_aasta);
end;
```

Joonis 17. Skript personali infosüsteemi tööajatabelite käivitamiseks.

Kõik vaated on omavahel seotud läbi võtme ning nende täpsemad kirjeldused on toodud Lisas 2. Joonisel 18 on esitatud personali infosüsteemi andmemudel, mis koosneb neljast vaatest. Joonisel 19 on esitatud HIS graafikute andmemudel. Põhitabeliks on BI.BIF_SCHEDULE_TIME, kuhu salvestatakse ambulatoorsete vastuvõttude ja uuringute graafikute ajaaknad. Tabel BI.BID_STRUCTURE_UNIT on sisestruktuuriüksuste tabel struktuuriüksuste, keskuste ja kliinikute nimedega ning koodidega. Tabelis STAFF on töötajate andmed (nimi, kood) ilma nende ametikohta täpsustuseta. Isikukoodid, mis on kahe infosüsteemi vahel ühenduslüliks, on talletatud tabelis LIVING_SUBJECT. Haigla infosüsteemi tabelid samuti paiknevad *Oracle* relatsioonilises andmebaasis.



Joonis 18. Personali infosüsteemi tööajatabelitega seotud vaated.



Joonis 19. HIS tööajatablete andmemudel.

4.7 Andmelao modelleerimise metoodika valik ja põhjendus

Tuginedes antud töö alapeatükis 2.3.4 tehtud võrdlusele, kasutati andmelao modelleerimises Kimballi lähenemist. Kimballi järgi võib andmelao arendamist alustada ühest andmevakast, mis on mõeldud ühe konkreetse osakonna või äriprotsessi jaoks. Kuna loomise hetkeks oli selgelt määratletud ainult üks äriprotsess ning teised protsessid ilmuvad andmelattu hiljem, siis oli mõistlik alustada andmevaka loomisest. See võimaldab kiiremini saada esimesi tulemusi.

Inmoni lähenemisviisi järgi võtavad andmelao esialgne ülesehitamine ja tarne rohkem aega ning selle edukaks kasutamiseks organisatsioonis peab olema suhteliselt suur spetsialistide meeskond, et hallata andmelao keskkonda [10, p. 18]. Magistritöö kirjeldatavas meditsiini-asutuses ei ole selleks vajalikke ressursse.

Kimballi tähtskeemi on aruandluses lihtne kasutada ja kasutajate jaoks on see arusaadavam. Enamik BI tööriistu töötavad tähtskeemiga hästi. Andmelao keskkond on väike ja võtab vähem ruumi. Keskkonna efektiivseks haldamiseks piisab väikesest arendajate ja arhitektide meeskonnast [10, p. 18].

Data Vault on semantiliselt keerukas. Traditsioonilise manuaalse kodeerimisega ei ole otstarbekas ehitada ETL protsessi. Jätkusuutlikuks hooldamiseks tuleb genereerida metaandmetega kood. Data Vault andmelao on tabelite vahel palju seoseid, mis mõjutavad jõudlust [20]. Asutuse äripoolne nõue oli kiire päringute vastuste saamine päringutele. Nimetatud põhjustel tehti valik Kimballi lähenemisviisi kasuks.

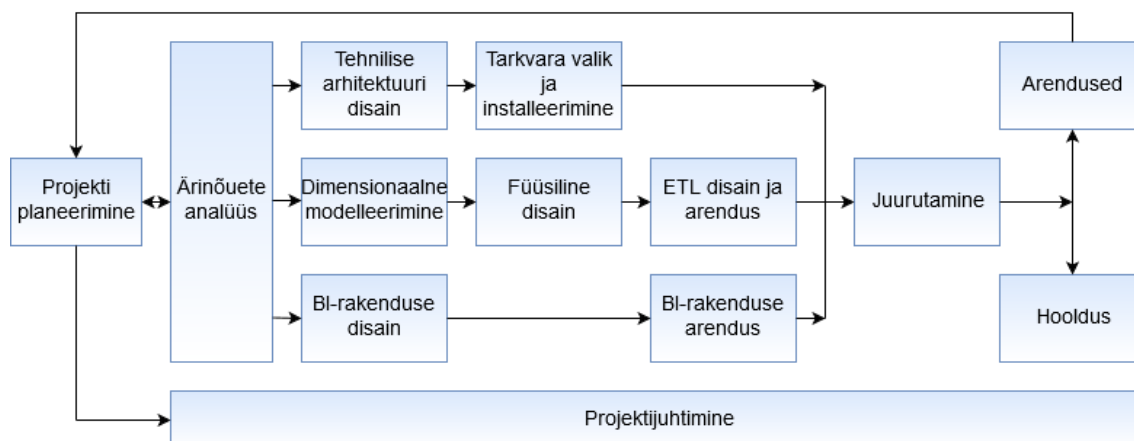
4.8 Andmelao arendusprotsess

Andmelao arendamiseks on palju erinevaid metoodikaid. Neist üks tuntuim on Kimball Lifecycle metoodika (Joonis 20), mida kasutati antud projektis. Kimballi elutsükli metoodika lõi 80ndate keskpaigas Kimballi grupp ja Metaphor Systems, ning seda tutvustati üldsusele 90ndatel [14].

Metoodika kolm peamist põhimõtet on [21]:

- Keskendumine ettevõttele lisandväärtuse loomisele;
- Andmete dimensionaalne modelleerimine;

- Iteratiivne andmelao/BI keskkonna arendamine elutsükli sammudega.



Joonis 20. Kimballi elutsükli arendusmetoodika.

Projekti planeerimisfaasis määratleti selle skoop ja komplekteeriti töögrupp. Käesoleva töö autor osales projektis nii arendaja kui ka analütiku rollis: suhtles IT- ja ärikasutajatega, realiseeris andmemudeli ja arendas ETL protsessi. Välispartneritega suhtles ettevõtte projektijuht, kes aitas ka personali infosüsteemi andmeid mõista. Samuti oli tema vastutusallas projekti eelarve. IT-osakonna spetsialist tegeles infrastruktuuriga, paigaldas tarkvarad ja haldas serveri. Andmelao grupijuht aitas nii korralduslikes küsimustes kui ka tehnilistes. Arenduse käigus konsulteeriti välisspetsialistidega.

Ettevõttele lisandväärtuse loomiseks selgitati ärinõuded, määratleti süsteemi tehnilised nõuded, disainiti dimensionaalne mudel ja selgitati, milliseid aruandeid on vaja arendada. Seejärel valiti ja installeeriti kõik vajalikud tarkvarad, koostati andmemudeli füüsiline disain, arendati ETL protsess ja aruanded. Süsteemi vastavust kontrolliti seatud funktsionaalsetele ja mittefunktsionaalsetele nõuetele. Nõuded olid süsteemis korrektselt realiseeritud.

Andmelao projekti ülesanded olid jagatud vastavalt Kimballi elutsükli arendusmetoodika tööriistakastile [21]. Joonisel 21 on esitatud projekti etapid ning erinevate isikute tegevused: vastutamine, sisendi andmine, tulemustest informeeritud saamine. Töö autori ülesanded on märgitud helesinisega.

	Ärikasutajad	Ärijuht	Analüütik & arendaja (töö autor)	Andmelao grupijuht	Andmelao spetsialist	Ettevõtte projekti juht	IT-osakonna spetsialist	IT-osakonna juhataja	Välisspetsialistid			
Projekti määratlus	◆	○	◆	○		●		○			LEGEND:	
Projekti planeerimine	□	◆	○	○		●		○			●	Vastutaja
Ärinõuete analüüs	◆		●	○		○					○	Kaasatud
Tehnilise arhitektuuri disain			●	○					○		◆	Annab sisendi
Tarkvara valik ja installeerimine		□	○	○		○	●	●			□	Informeeritakse tulemustest
Dimensionaalne modelleerimine			●	○					○			
Füüsiline disain			●	○					○			
ETL disain ja arendus			●	○					○			
BI-rakenduse disain	◆		●	○					○			
BI-rakenduse arendus			●	○					○			
Juurutamine	□		●	○			●	○				
Arendused	◆		●	○	●				○			
Hooldus			●	○	●		○					

Joonis 21. Andmelao projekti ülesanded.

Kuna andmelao projektid on küllalt keerukad, mistõttu võivad kergesti tekkida viivitused, hoiti kogu projekti vältel tihedat suhtlust IT- ja ärikasutajate vahel, ning jälgiti tekkivaid probleeme.

4.9 Andmelao projekti juurutamine

Tihti eeldatakse, et andmelao esimese versiooni saab kasutusele võtta vaid mõne nädalaga, sest piisab sellest, kui õpetada kasutajaid uusi vigadest teatamise vahendeid rakendama. Tegelikult tuleb neid koolitada ka andmelaost saadavate andmetega töötama, ning mis veel olulisem, kohanema uue tööprotsessiga, mille andmelao kasutamine kaasa toob. Parimate tulemuste saavutamiseks planeeritakse koolitada kasutajaid.

Lisaks koolitusele on andmelao juurutamise etapis plaanis:

- võtta kasutusele vajalikud turvameetmed (piirata ligipääs andmelao rakendustele ja andmetele);
- ette valmistada tugiuksus, kes hakkab vastama küsimustele andmelao kasutamise ja andmete kohta, sest tõenäoliselt leiavad kasutajad anomaaliaid isegi põhjalikult kontrollitud andmetes, ning vajavad nende tõlgendamiseks ja eelnevate andmetega kooskõlastamiseks abi;

- pakkuda tulevastele spetsialistidele põhjalikumad koolitust aruannete koostamiseks.

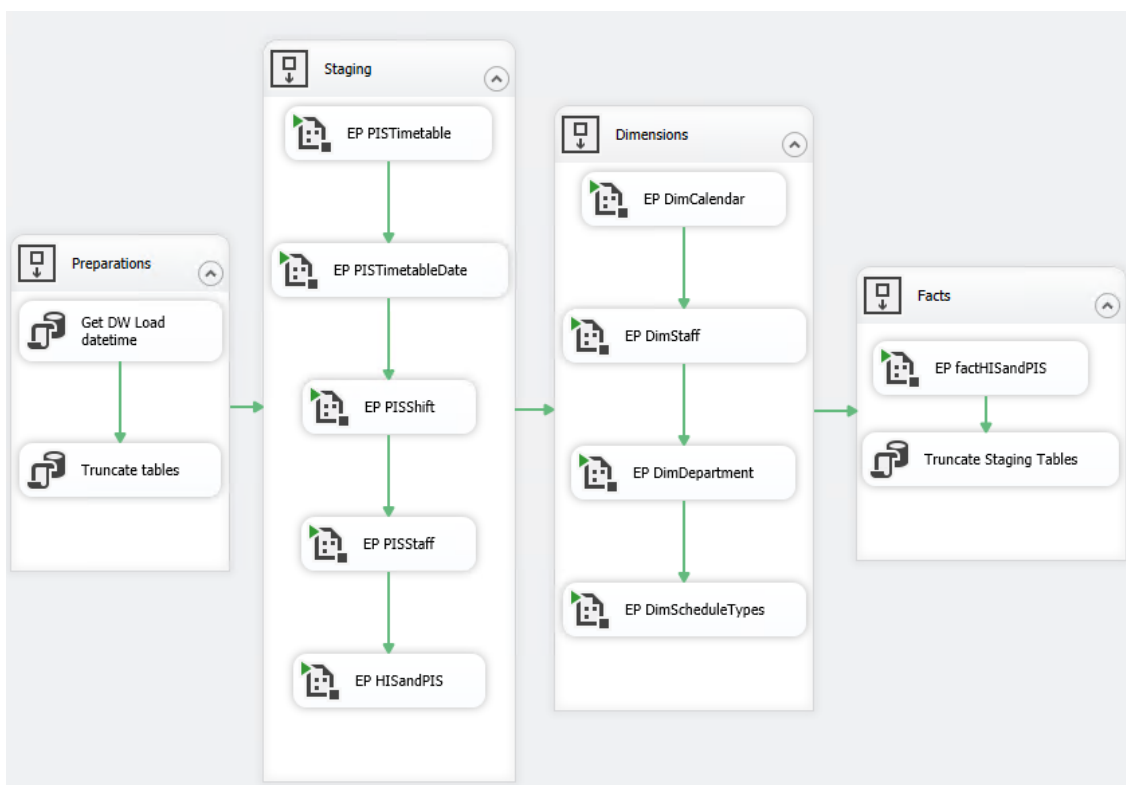
Andmetes vigade ennetamiseks on plaanis:

- oodata andmelao juurutamisega, kuni andmed on valmis (kättesaadavad ja kontrollitud) ning tööriistu ja ajakohastamisprotseduure on testitud;
- katsetada süsteemi enne selle juurutamist väikese testgrupiga, kelle moodustavad erinevate oskuste ja ootustega kasutajad;
- võimaldada ligipääs andmelaole vaid neile kasutajatele, keda on juba koolitatud.

4.10 ETL protsessi koostamine

Andmelao arendustsüklis on oluliseks komponendiks andmete teisenduskiht ehk ETL protsess, mis koosneb andmete pärimisest algallikast, töötlemisest ja andmelattu laadimisest.

Protsessi arenduseks kasutati SSIS (*SQL Server Integration Services*) platvormi vahendis *Microsoft Visual Studio 2015*, kus oli loodud projekt 1 põhipaketiga, mis käivitab 10 alampaketti (Joonis 22). Pakett on sisuliselt ülesannete kogum, mida täidetakse järkjärguliselt [22].



Joonis 22. ETL protsess SSIS platvormis.

Järgnevalt kirjeldatakse ETL protsessi alampakette.

Get DW Load datetime. Viimase andmete hõivamise ajatempli salvestamiseks metaandmete baasi loodi tabel *Data_Flow* (Lisa 4, SQL 1), veergudega *Name* ja *CET*, väärtustega *LastExtraction* ja parameetriga, mis määrab viimase laadimise ajahetke (Lisa 4, SQL 2).

Truncate tables. Andmelao tabelite tühjendamine (Lisa 4, SQL 3).

Personali infosüsteemi andmed (alampakettides *EP PISTimetable*, *EP PISTimetableDate*, *EP PISShift*) hõivati kuude kaupa. Enne vaadete käivitamist loodi ja omistati muutujatele väärtused (Joonis 23).

Name	Data type	Value	Expression
p_aasta	Int32	2018	DATEPART("yy",GETDATE())
p_aasta1	Int32	2018	(@[User::p_kuu1] == 1 ? DATEPART("yy",DATEADD("yy", 1, GETDATE())) : DATEPART("yy",GETDATE())
p_aasta2	Int32	2018	(@[User::p_kuu2] == 12 ? DATEPART("yy",DATEADD("yy", -1, GETDATE())) : DATEPART("yy",GETDATE())
p_kuu	Int32	4	DATEPART("mm",DATEADD("mm",0, GETDATE()))
p_kuu1	Int32	5	DATEPART("mm",DATEADD("mm",1, GETDATE()))
p_kuu2	Int32	3	DATEPART("mm",DATEADD("mm",-1, GETDATE()))

Joonis 23. Muutujate loomine ja omistamine.

Ülesandes *Execute SQL Task* kirjutati päring, kus küsimärgid kutsuvad välja muutujaid p_kuu ja p_aasta (Joonis 24):

```
begin view_parameters_pkg.set_mon_year(?,?);
end;
```

Joonis 24. Skripti käivitamine *Execute SQL Task* komponendis.

EP PISStaff. Käivitati pakett PISStaff, kuhu laeti töötajate andmed personali infosüsteemist.

EP HISandPIS. Tabelist BIF_SCHEDULE_TIME laetakse kolme kuu töögraafikute andmed (eelmine kuu, käesolev kuu ja järgmine kuu) (Lisa 4, SQL 4). Seejärel toimub andmete töötlemine *Derived Column* komponendiga (Joonis 25). Luuakse uued veerud SCHEDULE_DATE, SCHEDULE_TIMESTAMP ja SCHEDULE_END_TIME. Viimane on tähtis ajaperioodi võrdlemiseks personali infosüsteemi tööajatabeliga, kuna selles talletatakse vahetuste algus- ja lõpuajad.

Derived Column Name	Derived Column	Expression
SCHEDULE_DATE	<add as new c...	(DT_DBDATE)SCHEDULE_TIME
SCHEDULE_TIMESTAMP	<add as new c...	(DT_DBTIME2,0)(RIGHT("0" + (DT_STR,2,1252)DATEPART("HH",SCHEDULE_TIME),2) + ":" + RI.
SCHEDULE_END_TIME	<add as new c...	DATEADD("minute",(DT_I8)DURATION,SCHEDULE_TIME)

Joonis 25. Andmete töötlemine *Derived Column* komponendiga SSIS platvormis.

Tabeli BIF_SCHEDULE_TIME seostatakse tabelitega STAFF, LIVING_SUBJECT ja BI.BID_STRUCTURE_UNIT. *Derived Column* komponendis luuakse veerg YEAR_MONTH, et võrrelda andmeid personali infosüsteemi tööajatabelitega, kus on talletatud väärtused kuu ja aasta. Personali infosüsteemi tabelis TOOAJATABEL veerus AJATABELI_STAATUS asendati NULL väärtused funktsiooniga REPLACENULL (AJATABELI_STAATUS, "Tabel puudub"). Andmete ühildamiseks teisendati andmetüüpe, näiteks veerg KUU teisendati DT_NUMERIC andmetüübiks ning ühendati veeruga AASTA:

```
LEN((DT_WSTR,12)MONTH_NR) == 1 ? (DT_WSTR,12) aasta + "-" + "0" +
(DT_WSTR,12)MONTH_NR : (DT_WSTR,12) aasta + "-" + (DT_WSTR,12) kuu.
```

Lisati uus veerg PIS_IS_BUSY väljendiga NULL(DT_I4), mis hiljem täidetakse UPDATE lausega faktitabelis. Tabelist TOOTAJA_ANDMED hõivati kulukoht, mille esimesed neli sümbolit LEFT(KULUKOHT,4) vastavad BI.BID_STRUCTURE_UNIT veerule DEPARTMENT_CODE. Eestikeelsed veergude nimetused muudeti ingliskeelseteks ning ühendati komponendi *Merge Join* abil haigla infosüsteemi andmetega läbi töötaja isikukoodi ja keskuse.

EP DimCalendar. Kuupäevade dimensioon tabelist BID_CALENDAR (Lisa 4, SQL 5).

EP DimStaff. Dimensioon töötaja andmetega (Lisa 4, SQL 6). Andmed hõivatakse tabelitest LIVING_SUBJECT ja STAFF tingimusel, et tabeli STAFF veerg STAFF_CODE ei ole tühi (IS NOT NULL), kuna võrreldakse ainult meditsiinitöötajaid, kellel peab olema kood. Tabelis LIVING_SUBJECT talletatakse isikukode ning tabelis STAFF töötajate koode. Komponendi *Merge Join* abil seostatakse kaks tabelit läbi STAFF.PERSON_LINK_ID ja LIVING_SUBJECT.ID. Kuna mõned nimed olid väiketähtedega, mõned suurtähtedega, lisati komponent *Derived Column* funktsiooniga UPPER(), mis muudab kõik nimed suurtähelisteks.

EP DimDepartment. Keskuste ja kliinikute dimensiooni laeti tabelitest BI.BID_STRUCTURE_UNIT ja TOOTAJA_ANDMED (Lisa 4, SQL 7). Automaatselt genereeriti unikaalne surrogaatvõti DEPARTMENT_ID INT IDENTITY(1,1) [23, p. 223].

EP DimScheduleTypes. Laeti ambulatoorsete vastuvõttude ja uuringute graafikute blokeeringute tüüpide dimensioon tabelist BI.BIF_SCHEDULE_TIME (Lisa 4, SQL 8). SQL päringuga lisati tunnused HIS_IS_BUSY ja HIS_IS_BUSY_TEXT, mis näitavad, kas antud kombinatsioonis on graafik blokeeritud või mitte (Lisa 4, SQL 9). Erandiks on haigla infosüsteemi töögraafikute blokeeringu tüüp „Koosolek“, kuna personali infosüsteemi jaoks see on tööaeg, siis koosoleku korral veerul HIS_IS_BUSY näidatakse, et ajaaken ei ole blokeeritud. Automaatselt genereeriti unikaalne surrogaatvõti SCHEDULE_TYPE_ID INT IDENTITY(1,1) [23, p. 223].

EP factHISandPIS. Pärast dimensioonide laadimist toimub faktitabeli laadimine andmelattu. Enne laadimist komponendis *Execute SQL Task* lausega UPDATE kontrollitakse, kas personali infosüsteemis vahetuste tabelis PISShift esineb kirje sama ajaperioodiga või mitte ning vastavalt sellele uuendatakse veerg PIS_IS_BUSY

väärtustega 1 või 0. Erandiks on koolitused, mida samuti kajastatakse tabelis PISShift. Päring kontrollib, et koolitus ei toimuks antud ajavahemikul. Eellaadimise alast tõstetakse kirjed faktitabelisse. Viited dimensioonidele asendatakse vastavate väärtuste ID-ga.

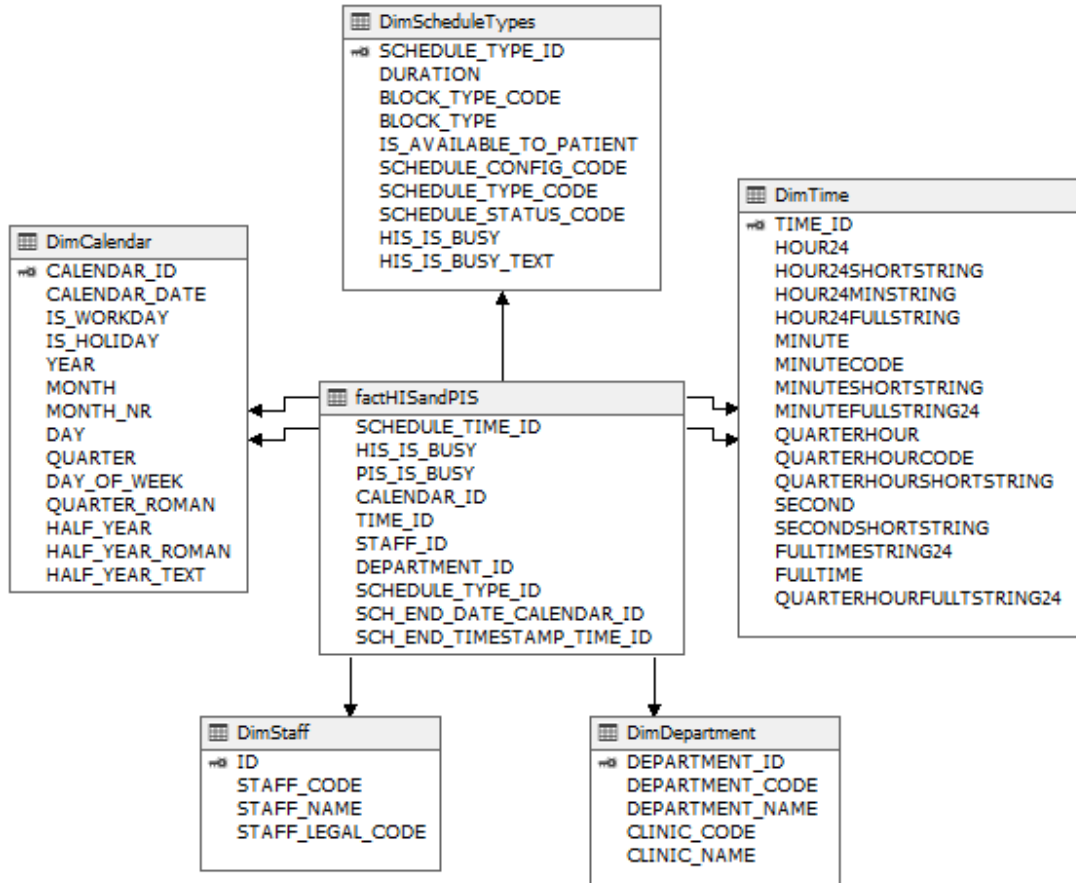
Truncate Staging Tables. Eellaadimisala tabelite tühjendamine TRUNCATE lausega.

ETL protsessi automaatseks käivitamiseks kasutati komponente *SQL Server Agent Jobs* ja *Schedules*. *SQL Agent Job* on konkreetne tegevus, mida *SQL Agent* täidab. Komponent *Schedules* määrab, millal *Job* peab olema täidetud. Projekti jaoks oli loodud üks *Job*, mis öösel kell 1:00 käivitab põhipaketi andmete automaatseks laadimiseks andmelattu.

4.11 Dimensionaalne andmemudel

Käesoleva töö projektis koostati andmemudel lähtudes dimensionaalse modelleerimise põhimõtetest. Andmemudel koosneb faktitabelist, kuhu kogutakse mõõdikud, ja neljast fakti kirjeldavast dimensioonitabelist (Joonis 26), mis on seotud faktitabeliga läbi võtme:

- *DimCalendar* – graafikute algus- ja lõpukuupäeva dimensioon
- *DimTime* – graafikute algus- ja lõpukellaaja dimensioon, mida genereeriti ja täideti skriptiga (Lisa 4, SQL 10 ja SQL 11) [24]
- *DimStaff* – meditsiinitöötaja dimensioon (ametikohta täpsustuseta)
- *DimDepartment* – haigla keskuste ja kliinikute dimensioon
- *DimScheduleType* – ambulatoorsete graafikute blokeeringute tüüpide dimensioon
- *FactHISandPIS* – haigla infosüsteemi töötajate graafikute ajaslottide faktitabel, mis sisaldab tunnust (PIS_IS_BUSY), kas antud ajaslot on personali infosüsteemis hõivatud (näiteks puhkusega) või mitte. Antud tunnuse väärtused on 0 ja 1, mis on faktitabeli mõõdiku rollis. Teine tunnus on HIS_IS_BUSY, mille väärtus on 1 – kui kliinilise infosüsteemi graafiku ajaslot on blokeeritud, näiteks koolituse jaoks, ja 0 – kui aeg on blokeerimata.



Joonis 26. Töö- ja vastuvõtugraafikute andmemudel.

Dimensionaalse andmemudeli disainimisel saadud tähtskeem on lihtsa struktuuriga ja võimaldab kiiresti analüüsida suuri andmekogusid. Seda seetõttu, et tähtskeemi tabelite vahel on vähe seoseid.

4.12 Ärianalüütika vahendi juurutamine ja aruannete loomine

Andmetega efektiivselt töötamiseks tuleb teada, millist aruandlusvahendit kasutada, ning kuidas teha seda parimal võimalikul viisil. Erinevad aruandlusvahendid mõjutavad ettevõtete analüüsi- ja otsustusprotsesse erinevalt. On hulgaliselt tarnijaid, kes selliseid vahendeid pakuvad.

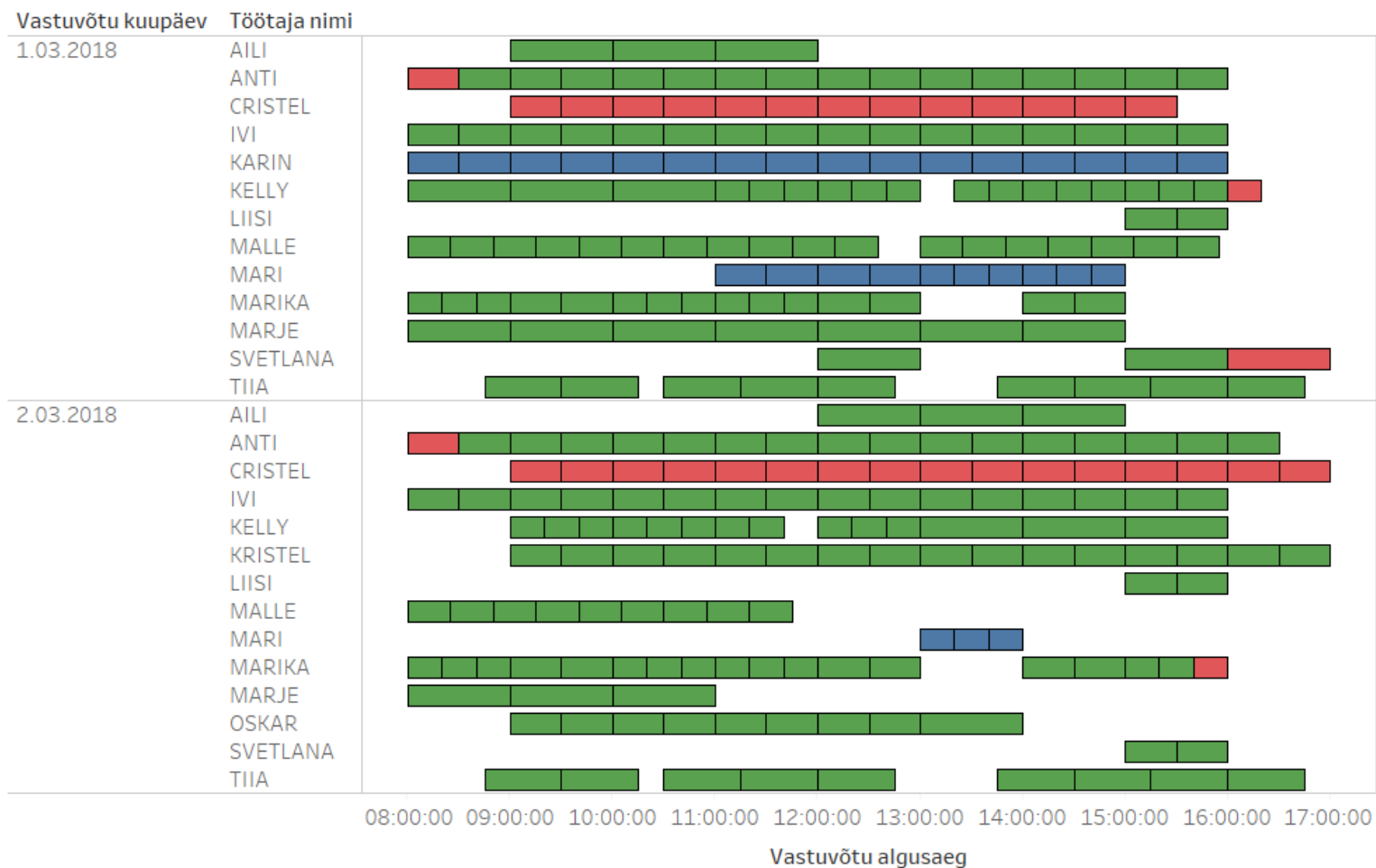
Täiusliku analüüsi- ja aruandlusvahendi omadused võiksid olla järgmised [25]:

- *Lihtsus.* Lihtne tavakasutaja jaoks.
- *Ühenduvus.* Ühendub sujuvalt paljude andmeallikatega.
- *Visuaalsus.* Esitab vaikumisi sobivaid graafikuid.

- *Jagamine*. Lihtne andmete jagamine.
- *Skaleeritavus*. Töötleb suuri andmekogumeid.

Disaini etapis prooviti erinevaid aruandlusvahendeid. Parimaks osutus *Tableau*, mis vastab ülaltoodud omadustele ning suudab rahuldada ärikasutajate vajadusi ja nõudeid (alapeatükk 4.2). Loodi võrdlusaruannete komplekt, mis toob välja andmete vasturääkivused kahes infosüsteemis. Joonisel 27 on näide ühest aruandest, milles on vasturääkivused erinevate värvidega märgistatud. Selleks kasutati horisontaalset tulpdiagrammi Gantt. Aruandes on sinisega märgistatud graafikuajad, mis on blokeeritud haigla infosüsteemis, kuid personali infosüsteemis mitte. Punasega tähistatud graafikuajad on personali infosüsteemis blokeeritud, kuid haigla infosüsteemis mitte. Rohelisega on märgistatud need graafikuajad, kus konflikti pole.

Personali- ja haigla infosüsteemi graafikute võrdlusaruanne

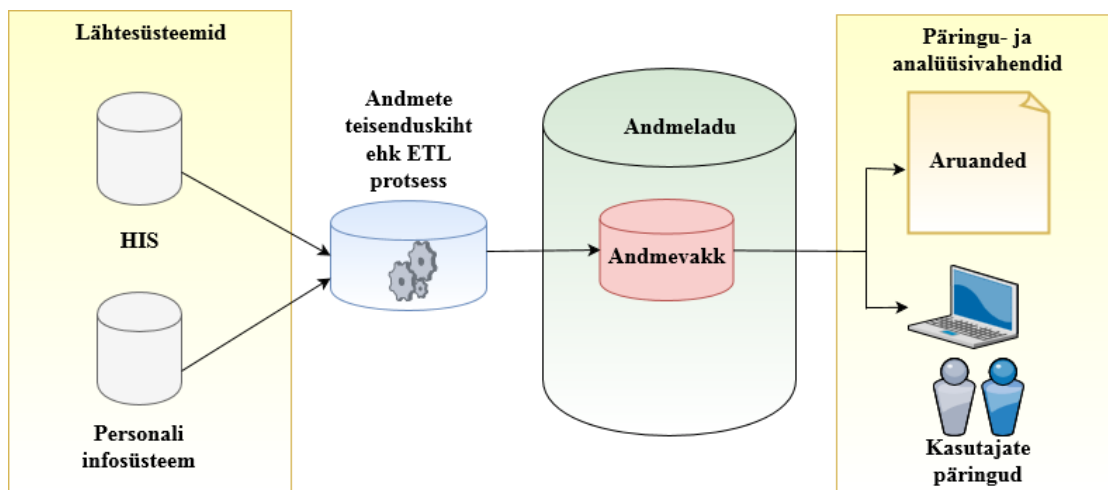


■ Blokeering HIS'is, personaliinfosüsteemis mit.. ■ Blokeering personaliinfosüsteemis, HIS'is mit.. ■ Konflikti pole

Joonis 27. Personali- ja haigla infosüsteemi graafikute võrdlusaruanne.

4.13 Lahenduse arhitektuur

Käesoleva lõputöö tulemuseks on andmelao lahendus, mis ühendab kahest erinevast lähtesüsteemist pärit andmed (Joonis 28). Lähtesüsteemideks on personali infosüsteem ja haigla infosüsteem (HIS).



Joonis 28. Andmelao lahenduse arhitektuur.

Andmed hõivatakse algallikatest, neid töödeldakse ja teisendatakse ETL protsessi abil. Andmeladu on keskne hoidla, mis hetkel koosneb ühest andmevakast. Andmevakk on orienteeritud ühe äriprotsessi aruannetele ehk tööajagraafikutele.

Äriksutajate jaoks loodi aruandlusvahendis graafikute võrdlusaruanne. Samuti on võimalus kirjutada analüütilisi SQL päringuid. Lahendus tagab kiire andmete töötlemise ja info liikumise kahe infosüsteemi vahel.

4.14 Ülevaade tööriistadest

Arenduseprotsessi käigus kasutati mitmeid tarkvaravahendeid. Protsessijoonised koostati kasutades vabavaralist tarkvara *Bizagi Process Modeler* [19] ning lähtudes parimatest praktikatest BPMN standardis [26]. BPMN on modelleerimiskeel, mis võimaldab esitada organisatsioonidele äriprotsesse ja töövooge.

Andmebaasi mudelite jaoks kasutati veebipõhist tööriista *Vertabelo*, mis on üliõpilastele tasuta. Joonised olid samuti tehtud veebipõhises tarkvaras *draw.io* [27], mis on mõeldud vooskeemide, protsessidiagrammide, graafikute ja UML skeemide jaoks.

Andmelao ETL protsess on loodud *Microsoft Visual Studio Community 2015's* kasutades *SQL Server Integration Services (SSIS)* platvormi. SQL kontrollpäringuid teostati *Microsoft SQL Server Management Studio's*. Aruandlusvahendina kasutati *Tableau* tööriista, mis võimaldab luua interaktiivseid aruandeid.

5 Tulemuste analüüs

Käesoleva töö üks hüpotees oli, et kahe infosüsteemi graafikute andmed pole vastavuses ja sisaldavad palju ebatäpsusi. Aruanne kinnitas seda hüpoteesi. Aruandest selgus tendents vigade hulga suurenemisele käesoleva ja tuleviku kuu graafikutes, kus haigla infosüsteemi järgi on töötaja koolitusel, puhkusel või lähetusel, aga personali infosüsteemi järgi hoopis tööl. Kõige vähem vigu on möödunud perioodi graafikutes. See näitab, et personali infosüsteemi sisestatakse andmed hiljem kui haigla infosüsteemi. Kuu lõpul tehakse palju lisatööd, et kahe programmi andmeid võrrelda ja vigu parandada.

Ka kuude puhul, mille kohta on palgaarvestus juba tehtud, leidub vastuolusid. Andmete uurimise käigus leiti, et nende vastuolude põhjus võib olla infosüsteemi eripära või tööprotsessi korraldus, kuna graafikuid ja töötajaid identifitseeritakse kahes infosüsteemis erinevalt. Osad juhid (sh juhatuse liikmed, kliinikute juhid) ei kajastu personali infosüsteemis vastuvõttude poolel vaid on ainult administratiivses rollis. Näiteks on inimene, kes töötab nii meditsiinilises osakonnas arstina kui ka tugiosakonnas juhatajana, personali infosüsteemis kirjas juhatajana ning tema tööajad on süsteemis täidetud vastavalt sellele ametikohale 8 tundi päevas, kuigi ambulatoorseid vastuvõtte teeb ta ehk vaid paar tundi päevas. Sellised erandid on mõistlik aruandes välistada, või muuta personali infosüsteemi funktsionaalsemaks, et saaks täpsemalt märkida ambulatoorsete või uuringute vastuvõttude aegu. On olnud juhtumeid, kus haigla infosüsteemis oli ambulatoorne graafik blokeeritud, kuid personali infosüsteemi järgi oli arst/õde tööl statsionaarses osakonnas. Selliseid juhtumeid pole võimalik omavahel võrrelda, kuna personali infosüsteemis ei kajastu, kas tegemist on ambulatoorse või statsionaarse osakonnaga. Teine võimalik põhjus vasturääkivuste tekkimiseks on haigla infosüsteemis avatuks jäetud uuringute graafikuajad, mis peaksid olema blokeeritud. Need jäetakse aga avatuks, kuna patsient ei saa ise endale aegu broneerida. Kui avatud ajad siiski broneeritakse haigla töötaja poolt, siis uuringu teostab keegi teine. Aruande järgi on tüüpiline, et haigla infosüsteemis on tööpäev pikem kui personali infosüsteemis. Oma rolli mängib ka inimfaktor, näiteks unustab töötaja teavitada tehtud muudatustest klienditeenindust või graafiku koostajat.

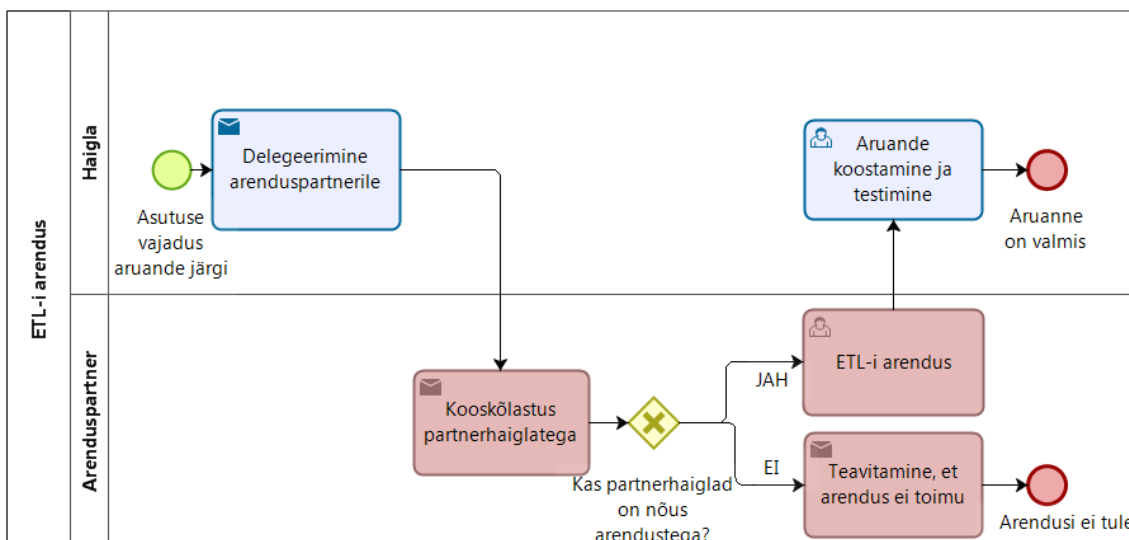
Teine hüpotees oli, et pärast vastava andmelao ja ärianalüütika vahendite juurutamist hakkab info liikuma senisest operatiivsemalt ja veatumalt. Seda kinnitas TO-BE mudeli simuleerimine, mille käigus selgus, andmelao kasutamise puhul vähenes lähteallikate vastuolude arv märkimisväärselt. AS-IS simulatsioonis tehtud muudatused kliinilises infosüsteemis polnud personali infosüsteemi sisestatud 63 korral, TO-BE protsessis vaid 14 korral.

5.1 Kasu organisatsioonile

Ressursside kokkuhoid. Töögraafikute muutmise võtab keskmiselt 35 minutit. TO-BE simulatsioonis vähenes graafikus muudatuste tegemiseks kuluv aeg 23 minutini. Suure hulga graafikute parandamise korral ulatub ajaline niisiis võit kümnete ja sadade töötundideni.

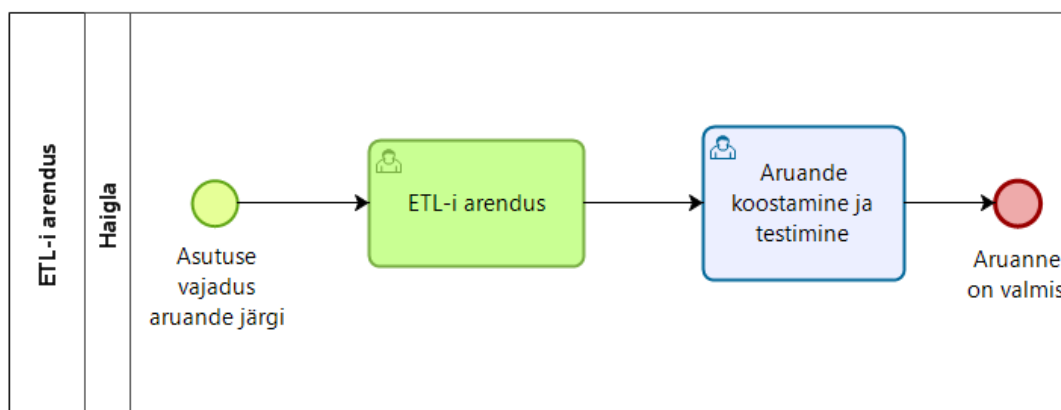
Andmete kvaliteedi parendamine. Lisaks väheneb lähteallikate vastuolude arv, kuna aruanded aitavad neid leida. AS-IS simulatsioonis tehtud muudatused kliinilises infosüsteemis polnud personali infosüsteemi sisestatud 63 korral, TO-BE protsessis vaid 14 korral. Seda tuhande muudatuse puhul.

Lahendus ettevõtte haldusalas. Samuti toob andmeladu teenuse ettevõtte haldusalasse, mis aitab vähendada sõltuvust arenduspartneritest, ning hoida ära suurte kulutuste tegemist kahe infosüsteemi vahele liidese loomiseks. Joonisel 29 on esitatud hetkeseis AS-IS mudelina: igasugune aruande arendus kooskõlastatakse partnerhaiglatega. See tähendab, et iga kord tuleb partnerhaiglat veenda arenduste vajalikkuses ja kasulikkuses. Kui partnerhaigla pole arendustega nõus, ei saa neid ette võtta. Arendusi teostab partner. Joonisel 29 on punasega märgistatud tegevus, mis on plaanis uue lahenduse abil parandada.



Joonis 29. AS-IS ETL arenduse protsess.

Joonisel 30 on esitatud tulevikuvaade TO-BE mudelina: arendusi saab teha iseseisvalt ja arengupartnerita, neid partnerhaiglatega kooskõlastamata. Rohelisega tähistatud tegevus aitaks parandada hetkeolukorda.



Joonis 30. TO-BE ETL arenduse protsess.

AS-IS ja TO-BE protsesside simulatsioon (Lisa 5) näitab, et andmelao kasutusele võtmisega võidab asutus nii ajaliselt kui ka rahaliselt. Tuleb märkida, et ETL arendusele kulub haiglal veidi enam aega, kuna asutusel pole selles palju kogemust (arenduspartneril kulub keskmiselt 672 minutit, haiglal 900 minutit), ent seevastu säästetakse aega, mis kuluks partnerhaigla vastuse ootamisele ning nendega info kooskõlastamisele (tavaliselt 2-14 päeva).

Rahaliselt poolelt on näha, et 100 simuleeritud juhtumist 92 puhul võtab partnerhaigla arenduse ette, ning selle eest tuleb maksta arenduspartnerile umbes 23 000 eurot. Kui arendusi tehakse haiglas sees, kulub 100 arenduse kohta umbes 7000 eurot.

Lisaks näitab simulatsioon, et asutus saab andmeladu kasutades vajalikud aruanded tehtud, kuna pole tarvis partnerhaigla nõusolekut. AS-IS protsessis realiseeriti 92 arendust; TO-BE protsessi puhul olid realiseeritud kõik arendused (100 arendust).

Kokkuvõttes võib öelda, et andmelao kasutamisest võidab asutus nii ajalises kui ka rahalises mõttes; samuti saab sellisel juhul realiseerida kõik soovitud arendused. Ühtlasi paraneb andmete kvaliteet.

5.2 Autori ettepanekud

Infosüsteemide funktsionaalsust puudutavad ettepanekud:

Ennekõike suurtes ettevõtetes on oluline, et töötajaid puudutavad põhiaandmed oleksid õiged ja ühtsed kõikides infosüsteemides. Selleks tuleb personali infosüsteemi struktuuriüksuste nimetused ühtlustada haigla infosüsteemis kasutatavate nimetustega või luua andmeloendite haldamiseks *Master Data* süsteem, kust teised süsteemid andmeid võtavad. Korralik loend on kõige tähtsam andmekvaliteedis ja seeläbi ka juhtimisinfot mõjutav faktor [28].

Kahe infosüsteemi täielikuks ja täpsemaks võrdlemiseks tuleb luua personali infosüsteemi tööajatabeli moodulisse ambulatoorse töö eraldi märkimise võimalus.

Tööprotsesse puudutavad ettepanekud:

Personaliinfosüsteemi graafikud peavad olema adekvaatsed, et saaks aegsasti vastuvõtnud sulgeda.

5.3 Autori tulemus

Autori põhitulemuseks on magistritöö eesmärkide saavutamine:

- Kahe infosüsteemi tarbeks on olemasolevate vahendite abil arendatud välja alternatiivne süsteem (andmeladu ja aruandluskeskkond), mis tuvastab lähteallikate andmekvaliteedi probleeme andmete vastuolude seire abil.
- On avastatud varjatud ebakõlasid töögraafikute moodulite kasutamisel.
- Erinevate lähtesüsteemide töögraafikute täielikuks võrdluseks on leitud muutmist vajavad kohad ja tehtud ettepanekud nende parendamiseks.

Eesmärkide saavutamiseks uuris töö autor, milliseid probleeme esineb graafikute kasutamisel praegu ning mida oodatakse uuest lahendusest. Autor suhtles asutuse töötajatega, kelle tööülesanded on seotud töögraafikutega ning nende haldamisega. Ka analüüsis autor kahe infosüsteemi andmekoosseisu. Teadusartikleid kasutades andis autor ülevaate andmelaost ja erinevatest disaini kontseptsioonidest, valis antud ettevõtte jaoks sobivaima disaini („*alt-üles*“) ja põhjendas seda. Arendas andmeladu ja aruandeid, mis toovad välja vasturääkivused kahes infosüsteemis. Andis asutusele omapoolsed soovitused, kuidas arendada infosüsteeme ja selle kaudu vähendada ebatäpsusi. Praeguse ja tuleviku olukorra võrdlemiseks kasutas autor AS-IS ja TO-BE mudeleid, mis võimaldavad hinnata pakutud lahenduse efektiivsust. Viis BMPN tehnikat kasutades läbi protsesside simuleerimise *Bizagi* töövahendis ning tõi välja ajalise võidu protsesside vahel. Arendusprotsessis kasutas *Kimball Lifecycle* metoodikat.

6 Kokkuvõte

Käesoleva magistritöö eesmärk oli analüüsida ja võrrelda andmelao disaini kontseptsioone ning luua valitud kontseptsiooni järgi andmeladu ja aruandluskeskkond infosüsteemidele, mis tuvastab lähteallikate andmekvaliteedi probleeme. Selle eesmärk on aidata kaasa nii andmekvaliteedi paranemisele kui ka ettevõtte tööprotsessi parendamisele.

Koostati riskianalüüs, ning viidi läbi graafikutes muudatuste tegemise simulatsioon. Loodi aruannete komplekt, ärikasutajate nõuetele vastav dimensionaalne andmemudel ja selleks vajalik ETL protsess. Parima andmelao loomise praktika järgimiseks konsulteeriti välisspetsialistiga (andmelao arhitektiga ja ärianalüüsi vahendite juurutajatega), ning tehti intervjuud meditsiinasutuse juhatusega, et kaardistada käimasolevad protsessid, koguda juhatuse nõuded andmelaole ning analüüsida võimalikke riske. Arendusprotsessis kasutati *Kimball Lifecycle* meetodikat.

Aruannete komplekti abil tõestati hüpotees, et kahe infosüsteemi graafikute andmed pole vastavuses. TO-BE simulatsiooniga tõestati ka teine hüpotees, et pärast vastava andmelao ja ärianalüütika vahendite juurutamist info hakkab liikuma senisest operatiivsemalt ja veatumalt.

Selgus, et andmelao ja aruannete kasutuselevõtt aitab töötajatel graafikutes muudatuste tegemisel oluliselt aega võita; samuti paranes andmete kvaliteet.

Lähtesüsteemise võrdlemiseks anti soovitusi infosüsteemi funktsionaalsuse parandamiseks (kuna funktsionaalsuse erinevuse tõttu polnud võimalik andmeid täielikult kontrollida).

Projekt andis meditsiinasutusele võimaluse proovida uut tehnoloogiat, ning andis olulise tõuke ettevõtte infotehnoloogilise arengule. Samuti tuleb asutusele kasuks, et teenus on nende haldusalas, mis aitab arenduspartneritest vähem sõltuda. Andmelao

loomine aitab ka ära hoida suurte kulutuste tegemist kahe erineva infosüsteemi vahele liidese loomiseks.

Kuigi töömahu suuruse tõttu ei jõutud analüüsida kõiki probleeme, said tähtsamad probleemid siiski analüüsitud ning neile esialgsed lahendused leitud. Edaspidi plaanitakse kasutajatelt tagasisidet koguda ja arendada andmeladu, näiteks võrrelda mitte ainult graafikuid vaid ka tegelikke toimunud vastuvõtte.

Tulevikus kaasatakse andmelattu ka raamatupidamise, kõnekeskuse, apteegisüsteemi ja siseveebiga seotud infosüsteeme. Töö tulemusi saab kasutada erinevate projektide jaoks, et andmelao kasutamise abil andmete kvaliteeti parandada.

Kasutatud kirjandus

- [1] E. Eessaar, „Andmebaasid II. Andmeaidad ja andmevakad.“ 2011. [Võrgumaterjal]. Available: http://maurus.ttu.ee/ained/IDU0230_2012/doc/9/Teema_IDU0230_14_2011_slaidid.pdf. [Kasutatud 1 Aprill 2018].
- [2] V. Kotkas, H.-M. Haav, J. Tepandi, E. Öunapuu ja J. Grauberg, „Uurimisprojekti „Andmeaitade (teiseste andmekogude) loomise põhimõtete väljatöötamine“ lõpparuanne,“ Märts 2013. [Võrgumaterjal]. Available: <https://www.riigikantselei.ee/valitsus/valitsus/et/riigikantselei/strateegia/poliitika-analuusid-ja-uuringud/tarkade-otsuste-fondi-uuringute-kokkuvotted>. [Kasutatud 26 Märts 2018].
- [3] V. Rainardi, *Building a Data Warehouse With Examples in SQL Server*, 2010.
- [4] P. Ponniah, *Data Warehousing Fundamentals For IT Professionals Second Edition*, A John Wiley & Sons, Inc.m Hoboken, New Jersey, 2010.
- [5] D. Linnstedt ja M. Olschimke, *Building a Scalable Data Warehouse with Data Vault 2.0*, Todd Green, 2016.
- [6] F. Hayes, „The story so far – Computerworld,“ 2002. [Võrgumaterjal]. Available: <http://www.computerworld.com/databasetopics/data/story/0%2C10801%2C70102%2C00.html>. [Kasutatud 10 märts 2018].
- [7] L. Yessad ja A. Labiod, „Comparative Study of Data Warehouses Modeling Approaches: Inmon, Kimball and Data Vault,“ *International Conference on System Reliability and Science*, 2016.
- [8] R. Kimball ja M. Ross, *The Data Warehouse Toolkit, Third Edition. The Definitive Guide to Dimensional Modeling.*, Wiley, 2013.
- [9] C. Ballard, D. Herreman, D. Schau, R. Bell, E. Kim ja A. Valencic, *Data modeling techniques for data warehousing*, IBM, 1998.
- [10] M. Breslin, „Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models,“ *Business Intelligence Journal*, p. 18, 2004.
- [11] M. Schreuder, „Dimensional modeling and Data Vault – a happy marriage?,“ 2011. [Võrgumaterjal]. Available: <http://blog.in2bi.com/business-intelligence/dimensional-modeling-and-data-vault-ndash-a-happy-marriage/>.

- [12] V. Schalkwyk, „A comparison of the impact of Data Vault and dimensional modelling on data warehouse performance and maintenance.“ *These de Magister*, Potchefstroom University, South of Africa, 2014.
- [13] M. Raud, „Andmelao koostamine laboriettevõttele,“ Tallinna Tehnikaülikooli Raamatukogu Digikogu, Tallinn, 2017.
- [14] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy ja B. Becker, *The Data Warehouse Lifecycle Toolkit*, Wiley Publishing, 2008.
- [15] S. Bongsik, „Risk Assessment Of Data Warehousing, A Case Study“.
- [16] „ISO/IEC 25010:2011,“ 2011. [Võrgumaterjal]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>. [Kasutatud 15 Märts 2018].
- [17] J. Tepandi, „Tarkvara protsessid, kvaliteet ja standardid,“ 13 Detsember 2017. [Võrgumaterjal]. Available: <http://tepani.ee/tns-loeng.pdf>. [Kasutatud 12 Aprill 2018].
- [18] M. Sillaots, „Kvalitatiivne riskianalüüs (Qualitative Risk Analysis),“ Tallinna Ülikool, 2004. [Võrgumaterjal]. Available: <http://www.tlu.ee/~martins/pj/2plan/209risk/2093000riskkvalanal.html>. [Kasutatud 2017-2018].
- [19] „Bizagi,“ Bizagi Lmted, 2002-2018. [Võrgumaterjal]. Available: <https://www.bizagi.com/uk/products/bpm-suite/modeler>.
- [20] D. Andriyenko, „Data Vault: the good, the bad and the ugly,“ 2016.
- [21] R. Kimball, „Kimball DW/BI Lifecycle Methodology,“ Kimball Group, [Võrgumaterjal]. Available: <https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dw-bi-lifecycle-method/>. [Kasutatud 10 Veebruar 2018].
- [22] B. Knight, D. Knight, M. Davis ja W. Snyder, *Knight's Microsoft SQL Server 2012 Integration Services 24-Hour Trainer*, John Wiley & Sons, Inc., 2013.
- [23] J. Mundy, W. Thornthwaite ja R. Kimball, *The Microsoft Data Warehouse Toolkit: With SQL Server 2008 R2 and the Microsoft Business Intelligence Toolset. Second Edition.*, Wiley Publishing, Inc., 2011.
- [24] R. Rad, „Script for Creating and Generating members for Time,“ 14 Juuli 2014. [Võrgumaterjal]. Available: <http://www.rad.pasfu.com/index.php?/archives/122-Script-for-Creating-and-Generating-members-for-Time-Dimension.html>. [Kasutatud Detsember 2017].
- [25] D. G. Murray, *Tableau Your Data! : Fast and Easy Visual Analysis with Tableau Software. Second Edition.*, John Wiley & Sons, Inc., 2016.
- [26] „Best practices in process modeling,“ Bizagi, [Võrgumaterjal]. Available:

http://help.bizagi.com/bpm-suite/en/index.html?what_to_avoid_when_process_model.htm. [Kasutatud Aprill 2018].

- [27] „Draw.io Diagrams,“ Artisan, [Võrgumaterjal]. Available: www.draw.io. [Kasutatud 2017-2018].
- [28] Optime, „Optime MDM - master data haldus,“ [Võrgumaterjal]. Available: http://optime.ee/index.php?option=com_content&view=article&id=40&Itemid=80&lang=et. [Kasutatud Aprill 2018].

Lisa 1 – HIS andmebaasi mudeli dokumenteerimine

Mudeli detailid

Mudeli nimetus: HIS

Andmebaasimootor: Oracle andmebaas

Tabelid

Tabel STAFF

Kirjeldus: Asutuse töötajad (üldine mõiste, roll täpsustuseta)

Veeru nimetus	Tüüp	Omadused	Kirjeldus
ID	number(18)	PK	Töötaja identifikaator
STAFF_NAME	varchar2(255)		Töötaja nimi
STAFF_CODE	varchar2(10)		Töötaja kood
STAFF_OLD_CODE	varchar2(10)		Töötaja vana kood
STAFF_TYPE_DOVA_CODE	varchar2(30)		Töötaja tüüpi kood
PERSON_LINK_ID	number(18)		
CONTACT_PHONE	number(10)		Töötaja telefoninumber
START_DATE	timestamp		Algusaeg
END_DATE	timestamp		Lõpuaeg
SYS_VERSION	number(100)		Versioon
SYS_MODIFY_TIME	timestamp		Viimase muutmise aeg
SYS_DELETE_STATUS	char(1)		Kas on kustutatud?
SYS_MODIFY_UID	varchar2(255)		Viimane muutja

Tabel BI_BIF_SCHEDULE_TIME

Kirjeldus: Ambulatoorse ja diagnostika graafikuaegade pesad (saadaval, kasutatud, blokeeritud, kuid mitte tühistatud).

Veeru nimetus	Tüüp	Omadused	Kirjeldus
SCHEDULE_TIME_ID	number(18)	PK	Graafikuaegade identifikaator
TREATMENT_STAFF_ID	number(18)		Viide töötajale
TREATMENT_STUN_ID	number(18)	null	Viide struktuuriüksusele
TREATMENT_PLACE_ID	number(18)	null	Viide ruumile
TREATMENT_PROF_ID	number(18)	null	Viide töötaja erialale
SCHEDULE_PROF_ID	number(18)	null	Viide graafiku erialale
SCHEDULE_TIME	timestamp		Viide kuupäevale ja kellaajale graafikus
DURATION	number(4)		Graafikus märgitus vastuvõtu kestus
BLOCK_TYPE_CODE	varchar2(30)		Blokeeringu tüübi kood
BLOCK_TYPE	varchar2(250)		Blokeeringu tüüp
SERVICE_ID	number(18)	null	Täidetud ainult siis, kui selle broneeringu aja jooksul saab läbi viia ühe väga spetsiifilise teenuse
IS_AVAILABLE_TO_PATIENT	char(1)	null	Indikaator, mis näitab kas aeg on patsiendiportaalis kättesaadav
REFERRAL_EXISTENCE_CODE	varchar2(30)	null	Kas on vajalik saatekiri

SCHEDULE_CONFIG_CODE	varchar2(4)		Vastuvõtuliik (esmane, korduv jne)
PATIENT_QUANTITY	number(3)		Patsientide arv vastuvõtul
FREE_QUANTITY	number(3)		Vabade aegade arv (tavaliselt 0 või 1)
SCHEDULE_TYPE_CODE	varchar2(30)		Vastuvõtutüüp (AMB - ambulatoorne, OBS - uuring)
SCHEDULE_STATUS_CODE	varchar2(30)		Atribuut, mis näitab, kas antud aeg kuulub graafikule, mis on avatud või suletud
SERVICE_GROUP_ID	number(18)		Viide teenusegrupile

Tabel LIVING_SUBJECT

Kirjeldus: Haigla operatiivbaasi tabel töötajate nimedega ja isikukoodidega

Veeru nimetus	Tüüp	Omadused	Kirjeldus
ID	number(18)	PK	Identifikaator
TYPE_DOVA_CODE	varchar2(30)		
CLASS_DOVA_CODE	varchar2(30)		
DISPLAY_NAME	varchar2(255)		Nimi
DISPLAY_CODE	varchar2(12)		Isikukood
BIRTH_TIME	timestamp		Sünniaeg
BIRTH_TIME_ACCURACY_DOVA_CODE	varchar2(30)		
DECEASED_TIME	timestamp		
GENDER_DOVA_CODE	varchar2(30)		Sugu (M/N)
GENDER_STATUS_DOVA_CODE	varchar2(30)		
RESIDENT_COUNTRY_ID	number(18)		

USER_MODIFY_TIME	timestamp		
DESCRIPTION	varchar2(255)		Kirjeldus
SYS_VERSION	number(100)		
SYS_MODIFY_TIME	timestamp		Viimase muutmise aeg
SYS_DELETE_STATUS	char(1)		Kas on kustutatud?
SYS_MODIFY_UID	varchar2(255)		Viimane muutja

Tabel BI_BID_STRUCTURE_UNIT

Kirjeldus: Sisestruktuuri üksused, keskused ja kliinikud

Veeru nimetus	Tüüp	Omadused	Kirjeldus
STRUCTURE_UNIT_ID	number(18)	PK	Struktuuriüksuse identifikaator
STUN_CODE	varchar2(30)		Struktuuriüksuse kood
STUN_NAME	varchar2(255)		Struktuuriüksuse nimi
DEPARTMENT_CODE	varchar2(30)		Keskuse kood
DEPARTMENT_NAME	varchar2(255)		Keskuse nimi
CLINIC_CODE	varchar2(30)		Kliiniku kood
CLINIC_NAME	varchar2(255)		Kliiniku nimi

Seos BI_BIF_SCHEDULE_TIME_STAFF

STAFF	1..*	BI_BIF_SCHEDULE_TIME
ID	<->	TREATMENT_STAFF_ID

Seos STAFF_LIVING_SUBJECT

LIVING_SUBJECT	0..*	STAFF
-----------------------	-------------	--------------

ID	<->	PERSON_LINK_ID
----	-----	----------------

Seos BI_BIF_SCHEDULE_TIME_BI_BID_STRUCTURE_UNIT

BI_BID_STRUCTURE_UNIT	1..*	BI_BIF_SCHEDULE_TIME
STRUCTURE_UNIT_ID	<->	TREATMENT_STUN_ID

Lisa 2 – Personali infosüsteemi vaadete dokumenteerimine

Mudeli detailid

Mudeli nimetus: Personali infosüsteem

Andmebaasimootor: Oracle andmebaas

Tabelid

Tabel TOOAJATABEL

Kirjeldus: Tööajatabel põhiantmetega kuude lõikes

Veeru nimetus	Tüüp	Omadused	Kirjeldus
AJATABEL_ID	number(10)	PK	Tööajatabeli identifikaator
KARJAAR_ID	number(10)	FK	Karjääri identifikaator
ISIKUKOOD	varchar2(12)		Töötaja isikukood
LEP_NR	varchar2(20)		Lepingu number
AASTA	number(4)		Tööajatabeli aasta
KUU	number(2)		Tööajatabeli kuupäev
IND_NORM	number(22)		Kuu individuaalsete töötundide norm
AJATABELI_STAATUS	varchar2(21)		Planeerimisel, tabel on kinnitatud, parandustabel
MUUDATUSE_KUUPAEV	date		Tööajatabeli viimase muudatuse kuupäev
MUUDATUSE_OLEMUS	char(6)		Muutmine, kustutamine, lisamine

Tabel TOOTAJA_ANDMED

Kirjeldus: Töötaja andmed karjääride lõikes.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
KARJAAR_ID	number(22)	PK	Karjääri identifikaator
PERENIMI	varchar2(40)		Töötaja perekonnanimi
EESNIMI	varchar2(30)		Töötaja eesnimi
ISIKUKOOD	varchar2(12)		Töötaja isikukood
PERSONAL_ID	number(22)		Töötaja identifikaator
LEP_NR	varchar2(20)		Lepingu number
KARJAAR_ALGUS	date		Karjääri alguskuupäev
KARJAAR_LOPP	date		Karjääri lõpukuupäev
KLIINIK_KOOD	number(22)		Kliiniku kood
KLIINIK_NIMETUS	varchar2(240)		Kliiniku nimetus
KESKUS_KOOD	number(22)		Keskuse kood
KESKUS_NIMETUS	varchar2(240)		Keskuse nimetus
STRUKUKSUS_KOOD	number(22)		Struktuuriüksuse kood
STRUKUKSUS_LUHEND	varchar2(20)		Struktuuriüksuse lühend
STRUKUKSUS_NIMETUS	varchar2(240)		Struktuuriüksuse nimetus
STRUKUKSUS_ID	number(22)		Struktuuriüksuse identifikaator
KULUKOHT	varchar2(10)		Kulukoht koodina
AMETIKOHALIIK	varchar2(6)		Ametikohaliik (õen, arst jne)
AMETIKOHT	varchar2(240)		Ametikoht (kardioloog, õde jne)
LEPINGU_TUNNUS	varchar2(6)		PÕH/SISKOH/RESI/VLK OH
TL_ALGUSE_KUUPAEV	date		Töölepingu algus
TL_LOPETAMISE_KUUPAEV	date		Töölepingu lõpetamine

Tabel TOOAJATABEL_KUUPAEV

Kirjeldus: Tööajatabel kuupäevade lõikes – 1 kuupäev ühekordselt, kõige varasem tööpäeva algus, kõige hilisem tööpäeva lõpp, summeeritud tööaeg, täpne töötundide kirjeldus tabelis VAHETUSED_VKT.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
AJATABEL_ID	number(10)	FK	Tööajatabeli identifikaator
KUUPAEV	varchar2(13)		Vahetuse/Valve/ Väljakutse/ Koolituse kuupäev
VAHETUSE_ALGUS	varchar2(4000)		Vahetuse algusaeg
VAHETUSE_LOPP	varchar2(4000)		Vahetuse lõpuaeg
VAHETUSE_TUNNID	char(5)		Vahetuse kestus tundides
VALVE_ALGUS	varchar2(4000)		Valve algusaeg
VALVE_LOPP	varchar2(4000)		Valve lõpuaeg
VALVE_TUNNID	char(5)		Valve kestus
VKVALVE_ALGUS	varchar2(4000)		Väljakutsevalvealgus
VKVALVE_LOPP	varchar2(4000)		Väljakutsevalvelõpp
VKVALVE_TUNNID	number(22)		Väljakutsevalvetunnid
EEMALOLEKU_LIIK	char(5)		P - korraline puhkus, H - haigus, ÕP - õppepuhkus jne

Tabel VAHETUSED_VKT

Kirjeldus: Töötajate vahetused, andmed lisatabelist ja väljakutsete tundide tabelist ehk need andmed, mis võivad olla mitmerealised ühe päeva kohta. Lisatabelist kajastub ainult koolitus kestusega vähem kui 8 tundi.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
AJATABEL_ID	number(10)	FK	Tööajatabeli identifikaator

KUUPAEV	varchar2(13)		Vahetuse/Koolituse kuupäev
VAHETUSE_ALGUS	varchar2(4000)		Vahetuse algusaeg
VAHETUSE_LOPP	varchar2(4000)		Vahetuse lõpuaeg
VAHETUSE_TUNNID	number(22)		Vahetuse kestus tundides
VAHETUSE_LIIK	varchar2(3)		Vahetuse liik
EEMALOLEKU_ALGUS	varchar2(4000)		Eemaloleku algusaeg
EEMALOLEKU_LOPP	varchar2(4000)		Eemaloleku lõpuaeg
EEMALOLEKU_TUNNID	number(22)		Eemaloleku kestus tundides
EEMALOLEKU_LIIK	varchar2(2)		Eemaloleku liik (nt K)
VKT_ALGUS	varchar2(4000)		Väljakutse algusaeg
VKT_LOPP	varchar2(4000)		Väljakutse lõpuaeg
VKT_TUNNID	number(22)		Väljakutse kestus tundides

Seos TOOAJATABEL_TOOTAJA_ANDMED

TOOTAJA_ANDMED	1..*	TOOAJATABEL
KARJAAR_ID	<->	KARJAAR_ID

Seos TOOAJATABEL_KP_TOOAJATABEL

TOOAJATABEL	1..*	TOOAJATABEL_KUUPAEV
AJATABEL_ID	<->	AJATABEL_ID

Seos VAHETUSED_VKT_TOOAJATABEL

TOOAJATABEL	1..*	VAHETUSED_VKT
AJATABEL_ID	<->	AJATABEL_ID

Lisa 3 – Andmelao dimensioonitabelite dokumenteerimine

Dimensioon DimCalendar

Kirjeldus: Graafikute algus- ja lõpukuupäeva dimensioon.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
CALENDAR_ID	numeric(18)	PK	Kalendri identifikaator
CALENDAR_DATE	date		Kuupäev
IS_WORKDAY	nvarchar(1)		Kas tööpäev?
IS_HOLIDAY	nvarchar(1)		Kas riigipüha?
YEAR	numeric(4,0)		Aasta
MONTH	nvarchar(30)		Kuu
MONTH_NR	numeric(2,0)		Kuu number
DAY	numeric(2,0)		Päev
QUARTER	numeric(1,0)		Kvartal
QUARTER_ROMAN	nvarchar(3)		Kvartali number rooma numbritega (nt IV)
HALF_YEAR	numeric(1,0)		Poolaasta
HALF_YEAR_ROMAN	nvarchar(3)		Poolaasta rooma numbritega (I ja II)
HALF_YEAR_TEXT	nvarchar(30)		Poolaasta tekst rakenduse aktiivkeeles
DAY_OF_WEEK	nvarchar(30)		Nädalapäev

Dimensioon DimTime

Kirjeldus: Graafikute algus- ja lõpukellaaja dimensioon.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
---------------	------	----------	-----------

TIME_ID	int	PK	Aja identifikaator
HOUR24	int		Täistunnid (nt 23)
HOUR24SHORTSTRING	varchar(2)		Täistunnid (nt 23:00)
HOUR24MINSTRING	varchar(5)		Täistunnid (nt 23:00)
HOUR24FULLSTRING	varchar(8)		Täistunnid (nt 23:00:00)
MINUTE	int		Minutid (nt 59)
MINUTECODE	int		Minutite kood (nt 2359)
MINUTESHORTSTRING	varchar(2)		Minutid (nt 59)
MINUTEFULLSTRING24	varchar(8)		Kellaaeg minutite täpsustusega (nt 23:59:00)
QUARTERHOUR	int		Veerand tundi (nt 3)
QUARTERHOURCODE	int		Veerand tundi kood (nt 2345)
QUARTERHOURSHORTSTRING	varchar(2)		Kellaaeg 15 minuti kaupa (nt 45)
SECOND	int		Sekundid (nt 29)
SECONDSHORTSTRING	varchar(2)		Sekundid (nt 29)
FULLTIMESTRING24	varchar(8)		Kellaaeg (tunnid, minutid, sekundid, nt 23:59:29)
FULLTIME	time(7)		Kellaaeg (tunnid, minutid, sekundid, nt 23:59:29)
QUARTERHOURFULLSTRING24	varchar(8)		Kellaaeg minutite täpsustusega 15 minuti kaupa (nt 23:45:00)

Dimensioon DimScheduleTypes

Kirjeldus: Ambulatoorsete vastuvõtude ja uuringute graafikute tüüpide dimensioon.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
SCHEDULE_TYPE_ID	int	PK	Graafiku tüübi identifikaator
DURATION	numeric(4,0)		Vastuvõtu kestus (graafikus)
BLOCK_TYPE_CODE	nvarchar(120)		Blokeeringu tüübi kood
BLOCK_TYPE	nvarchar(120)		Blokeeringu tüüp

IS_AVAILABLE_TO_PATIENT	nvarchar(1)		Tunnus, kas on patsiendiportaalis nähtav
SCHEDULE_CONFIG_CODE	nvarchar(16)		Vastuvõtu liik (nt E - esmane, K - korduv)
SCHEDULE_TYPE_CODE	nvarchar(30)		Vastuvõtu tüüp (nt AMB – ambulatoorne)
SCHEDULE_STATUS_CODE	nvarchar(30)		Tunnus, kas graafikuaeg on avatud või suletud
HIS_IS_BUSY	nvarchar(1)		Tunnus, mis näitab, kas antud kombinatsioonis on graafiku aeg blokeeritud või mitte (1 või 0)
HIS_IS_BUSY_TEXT	nvarchar(11)		Tunnus, mis näitab, kas antud kombinatsioonis on graafiku aeg blokeeritud või mitte (tekstina)

Dimensioon DimStaff

Kirjeldus: Haigla töötaja dimensioon.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
ID	numeric(18)	PK	Töötaja identifikaator
STAFF_CODE	nvarchar(40)		Töötaja kood
STAFF_LEGAL_CODE	nvarchar(1020)		Töötaja isikukood
STAFF_NAME	nvarchar(120)		Töötaja nimi

Dimensioon DimDepartment

Kirjeldus: Haigla keskuste ja kliinikute dimensioon.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
DEPARTMENT_ID	int	PK	Keskuse identifikaator
DEPARTMENT_CODE	nvarchar(30)		Keskuse kood
DEPARTMENT_NAME	nvarchar(255)		Keskuse nimetus
CLINIC_CODE	nvarchar(30)		Kliiniku kood
CLINIC_NAME	nvarchar(255)		Kliiniku nimetus

Faktitabel factHISandPIS

Kirjeldus: Faktitabel haigla infosüsteemi töö- ja vastuvõtugraafikutega ja tunnusega, kas personaliinfosüsteemis on see graafiku aeg blokeeritud või mitte.

Veeru nimetus	Tüüp	Omadused	Kirjeldus
SCHEDULE_TIME_ID	int	PK	Graafiku aja identifikaator
HIS_IS_BUSY	int		Tunnus, kas HIS-is on blokeering
PIS_IS_BUSY	int		Tunnus, kas personali infosüsteemis on blokeering
CALENDAR_ID	int		Kuupäeva ID andmelaos
TIME_ID	int		Kellaaja ID andmelaos
STAFF_ID	int		Töötaja ID andmelaos
DEPARTMENT_ID	int		Keskuse ID andmelaos
SCHEDULE_TYPE_ID	int		Graafiku tüübi ID andmelaos
SCH_END_DATE_CALENDAR_ID	int		Lõpukuupäeva ID andmelaos
SCH_END_TIMESTAMP_TIME_ID	int		Lõpukellaaja ID andmelaos

Lisa 4 – ETL protsessiga seotud SQL laused

SQL 1. Viimase andmete hõivamise ajatempel metaandmetes

```
USE META
GO

IF EXISTS
    (SELECT * FROM SYS.TABLES
     WHERE NAME = 'DATA_FLOW')
DROP TABLE DATA_FLOW
GO
CREATE TABLE DATA_FLOW
    ( ID int          NOT NULL IDENTITY(1,1),
      NAME varchar(20) NOT NULL,
      CET datetime,
      CONSTRAINT PK_DATA_FLOW
      PRIMARY KEY CLUSTERED (ID)
    )
GO

CREATE INDEX DATA_FLOW_NAME
ON DATA_FLOW (name)
GO

DECLARE @CET DATETIME
SET @CET='2018-01-01 03:00:00'
INSERT INTO DATA_FLOW (NAME, CET)
    VALUES ('LASTEXTRACTION', @CET)
GO
```

SQL 2. Viimase laadimise ajahetk SSIS Execute SQL komponendis

```
UPDATE DATA_FLOW SET CET = GETDATE()
WHERE NAME = 'LASTEXTRACTION'
```

SQL 3. Tabelite tühjendamine

```

TRUNCATE TABLE dbo.PISTimetable;
TRUNCATE TABLE dbo.PISTimetableDate;
TRUNCATE TABLE dbo.PISShift;
TRUNCATE TABLE dbo.PISStaff;
TRUNCATE TABLE dbo.HISandPIS;
TRUNCATE TABLE dbo.factHISandPIS;
TRUNCATE TABLE dbo.DimCalendar;
TRUNCATE TABLE dbo.DimStaff;
TRUNCATE TABLE dbo.DimDepartment;
TRUNCATE TABLE dbo.DimScheduleTypes;

```

SQL 4. Haigla infosüsteemi kolme kuu graafikud

```

SELECT *
FROM BIF_SCHEDULE_TIME
WHERE SCHEDULE_TIME BETWEEN ADD_MONTHS(TRUNC(SYSDATE, 'MONTH'), - 1)
AND ADD_MONTHS(TRUNC(SYSDATE, 'MONTH'), 2)

```

SQL 5. Dimensiooni DimCalendar loomine

```

CREATE TABLE "DimCalendar" (
    "CALENDAR_ID" numeric(18),
    "IS_WORKDAY" nvarchar(1),
    "IS_HOLIDAY" nvarchar(1),
    "YEAR" numeric(4,0),
    "MONTH" nvarchar(30),
    "MONTH_NR" numeric(2,0),
    "DAY" numeric(2,0),
    "QUARTER" numeric(1,0),
    "DAY_OF_WEEK" nvarchar(30),
    "QUARTER_ROMAN" nvarchar(3),
    "HALF_YEAR" numeric(1,0),
    "HALF_YEAR_ROMAN" nvarchar(3),
    "HALF_YEAR_TEXT" nvarchar(30),
    "CALENDAR_DATE" date,
    CONSTRAINT pk_DimCalendar PRIMARY KEY (CALENDAR_ID)
)

```

SQL 6. Dimensiooni DimStaff loomine

```

CREATE TABLE DimStaff (
    [ID] numeric(18,0),
    [STAFF_CODE] nvarchar(40),
    [STAFF_NAME] nvarchar(1020),
    [STAFF_LEGAL_CODE] nvarchar(120),
    CONSTRAINT pk_DimStaff PRIMARY KEY (ID)
);

```

SQL 7. Dimensiooni DimDepartment loomine

```
CREATE TABLE "DimDepartment" (  
    "DEPARTMENT_ID" int IDENTITY(1,1),  
    "DEPARTMENT_CODE" nvarchar(30),  
    "DEPARTMENT_NAME" nvarchar(255),  
    "CLINIC_CODE" nvarchar(30),  
    "CLINIC_NAME" nvarchar(255),  
    CONSTRAINT pk_DimDepartment PRIMARY KEY (DEPARTMENT_ID)  
);
```

SQL 8. Dimensiooni DimScheduleTypes loomine

```
CREATE TABLE "DimScheduleTypes" (  
    "SCHEDULE_TYPE_ID" int IDENTITY(1,1),  
    "DURATION" numeric(4,0),  
    "BLOCK_TYPE_CODE" nvarchar(120),  
    "BLOCK_TYPE" nvarchar(1020),  
    "IS_AVAILABLE_TO_PATIENT" nvarchar(1),  
    "SCHEDULE_CONFIG_CODE" nvarchar(16),  
    "SCHEDULE_TYPE_CODE" nvarchar(30),  
    "SCHEDULE_STATUS_CODE" nvarchar(30),  
    "HIS_IS_BUSY" nvarchar(1),  
    "HIS_IS_BUSY_TEXT" nvarchar(11),  
    CONSTRAINT pk_DimScheduleTypes PRIMARY KEY (SCHEDULE_TYPE_ID)  
);
```

SQL 9. Päring, mis lisab juurde tunnused HIS_IS_BUSY ja HIS_IS_BUSY_TEXT

```
SELECT DISTINCT DURATION  
    ,BLOCK_TYPE_CODE  
    ,BLOCK_TYPE  
    ,IS_AVAILABLE_TO_PATIENT  
    ,SCHEDULE_CONFIG_CODE  
    ,SCHEDULE_TYPE_CODE  
    ,SCHEDULE_STATUS_CODE  
    ,CASE WHEN BLOCK_TYPE = '-'  
        AND SCHEDULE_STATUS_CODE = 'OPENED'  
        THEN 0  
        ELSE 1  
        END AS HIS_IS_BUSY  
    ,CASE WHEN BLOCK_TYPE = '-'  
        AND SCHEDULE_STATUS_CODE = 'OPENED'  
        THEN 'Avatud'  
        ELSE 'Blokeeritud'  
        END AS HIS_IS_BUSY_TEXT  
FROM BIF_SCHEDULE_TIME;
```

SQL 10. Ajadimensiooni struktuuri loomise skript

```
CREATE TABLE [dbo].[DimTime] (  
    [TIME_ID] [int] NOT NULL,  
    [HOUR24] [int] NULL,  
    [HOUR24SHORTSTRING] [varchar](2) NULL,  
    [HOUR24MINSTRING] [varchar](5) NULL,  
    [HOUR24FULLSTRING] [varchar](8) NULL,  
    [MINUTE] [int] NULL,  
    [MINUTECODE] [int] NULL,  
    [MINUTESHORTSTRING] [varchar](2) NULL,  
    [MINUTEFULLSTRING24] [varchar](8) NULL,  
    [QUARTERHOUR] [int] NULL,  
    [QUARTERHOURCODE] [int] NULL,  
    [QUARTERHOURSHORTSTRING] [varchar](2) NULL,  
    [QUARTERHOURFULLSTRING24] [varchar](8) NULL,  
    [SECOND] [int] NULL,  
    [SECONDSHORTSTRING] [varchar](2) NULL,  
    [FULLTIMESTRING24] [varchar](8) NULL,  
    [FULLTIME] [time](7) NULL,  
    CONSTRAINT [PK_DimTime] PRIMARY KEY CLUSTERED ([TIME_ID] ASC) WITH (  
        PAD_INDEX = OFF,  
        STATISTICS_NORECOMPUTE = OFF,  
        IGNORE_DUP_KEY = OFF,  
        ALLOW_ROW_LOCKS = ON,  
        ALLOW_PAGE_LOCKS = ON,  
    ) ON [PRIMARY]  
) ON [PRIMARY];
```

SQL 11. Ajadimensiooni kirjete genereerimise skript

```
declare @hour int  
declare @minute int  
declare @second int  
set @hour=0  
while @hour<24  
begin  
    set @minute=0  
    while @minute<60  
    begin  
        set @second=0  
        while @second<60  
        begin  
            INSERT INTO [dbo].[DimTime]  
            ([TIME_ID],  
            [HOUR24],  
            [HOUR24SHORTSTRING],  
            [HOUR24MINSTRING],  
            [HOUR24FULLSTRING],  
            [MINUTE],  
            [MINUTECODE],
```



```

[MINUTESHORTSTRING],
[MINUTEFULLSTRING24],
[QUARTERHOUR],
[QUARTERHOURCODE],
[QUARTERHOURSHORTSTRING],
[QUARTERHOURFULLSTRING24],
[SECOND],
[SECONDSHORTSTRING],
[FULLTimestRING24],
[FULLTIME])
select
(@hour*10000) + (@minute*100) + @second as TIME_ID,
@hour as [HOUR24],
right('0'+convert(varchar(2),@hour),2) [HOUR24SHORTSTRING],
right('0'+convert(varchar(2),@hour),2)+':00' [HOUR24MINSTRING],
right('0'+convert(varchar(2),@hour),2)+':00:00' [HOUR24FULLSTRING],
@minute as [MINUTE],
(@hour*100) + (@minute) [MINUTECODE],
right('0'+convert(varchar(2),@minute),2) [MINUTESHORTSTRING],
right('0'+convert(varchar(2),@hour),2)+':'+
right('0'+convert(varchar(2),@minute),2)+':00' [MINUTEFULLSTRING24],
@minute/15 as [QUARTERHOUR],
(@hour*100) + ((@minute/15)*15) [QUARTERHOURCODE],
right('0'+convert(varchar(2),((@minute/15)*15)),2) [QUARTERHOURSHORTSTRING],
right('0'+convert(varchar(2),@hour),2)+':'+
right('0'+convert(varchar(2),((@minute/15)*15)),2)+':00'
[QUARTERHOURFULLSTRING24],
@second as [SECOND],
right('0'+convert(varchar(2),@second),2) [SECONDSHORTSTRING],
right('0'+convert(varchar(2),@hour),2)+':'+
right('0'+convert(varchar(2),@minute),2)+':'+
right('0'+convert(varchar(2),@second),2) [FULLTimestRING24],
convert(time,right('0'+convert(varchar(2),@hour),2)+':'+
right('0'+convert(varchar(2),@minute),2)+':'+
right('0'+convert(varchar(2),@second),2)) as [FULLTIME]
set @second=@second+1
end
set @minute=@minute+1
end
set @hour=@hour+1
end;

```

Lisa 5 – ETL arenduse AS-IS ja TO-BE protsessi simulatsioon

Joonis 1. ETL arenduse AS-IS protsessi simulatsioon

Name	Type	Instances completed	Instances started	Min. time (m)	Max. time (m)	Avg. time (m)	Total time (m)	Total fixed cost
ETL-i arendus	Process	100	100	125	720	672.4	67240	23000
Asutuse vajadus aruande järgi	Start event	100						
Delegeerimine arenduspartnerile	Task	100	100	60	60	60	6000	0
Kas partnerhaiglad on nõus arendustega?	Gateway	100	100					
Aruande koostamine ja testimine	Task	92	92	300	300	300	27600	0
ETL-i arendus	Task	92	92	300	300	300	27600	23000
Kooskõlastus partnerhaiglatega	Task	100	100	60	60	60	6000	0
Aruanne on valmis	End event	92						
Teavitamine, et arendus ei toimu	Task	8	8	5	5	5	40	0
Arendusi ei tule	End event	8						

Joonis 2. ETL arenduse TO-BE protsessi simulatsioon

Name	Type	Instances completed	Instances started	Min. time (m)	Max. time (m)	Avg. time (m)	Total time (m)	Total fixed cost
ETL-i arendus	Process	100	100	900	900	900	90000	7000
Asutuse vajadus aruande järgi	Start event	100						
Aruanne on valmis	End event	100						
ETL-i arendus	Task	100	100	600	600	600	60000	7000
Aruande koostamine ja testimine	Task	100	100	300	300	300	30000	0