

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kristjan Kolde 179837IADB

# **Eesti Covid-19 statistika ja avaandmete arendus tervise infosüsteemi andmelao põhjal**

Bakalaureusetöö

Juhendaja: Priit Rospel  
MSc.

Tallinn 2023

## **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: Kristjan Kolde

15. mai 2023

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on välja töötada lahendused Eestit tabanud Covid-19 tervisekriisi juhtimiseks vajaliku info tagamiseks ning avalikkuse teabevajaduse rahuldamiseks.

Töö käigus kirjeldatakse terviseandmetega seotud ökosüsteemi, analüüsitakse Covid-19 andmeteenuste jaoks vajalikke äri- ja süsteeminõudeid ning kirjeldatakse andmeteenuste arendusprotsess.

Andmeteenuste arendamiseks töötatakse tervise infosüsteemi andmelao tehnoloogial välja SQL protseduurid andmete rikastamiseks, puhastamiseks, ärireeglite rakendamiseks, andmete ajaloostatud laadimiseks ning automatiseeritakse avaandmete publitseerimine.

Arendused viiakse täide perioodil 2020 aasta märtsist kuni 2022 aasta lõpuni, seejuures suurem osa arendustöödest teostatakse tervisekriisi aegkriitilisusest lähtuvalt lühikeses ajaraamis, iteratiivset arendusprotsessi rakendades.

Arendustöö väljund annab tervisekriisi perioodil igapäevaselt automatiseeritud avaandmetena välja Covid-19 nakatumise ja vaktsineerimise statistikat. Antud lahendus on uudne võrreldes varasemaga, kus tervisestatistika koostamine toimus käsitööna.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 47 leheküljel, 8 peatükki, 18 joonist, 7 tabelit.

## **Abstract**

### **Development of Estonian COVID-19 Statistics and Open Data Solutions Based on the Data Warehouse of the National Health Information System**

The aim of the current Bachelor thesis is to devise possible solutions to provide necessary information relating to the Covid-19 crisis in Estonia to stakeholders and the public.

This piece of work describes the ecosystem of health data, analyses crucial business and system requirements for necessary data services and describes the development process of data services.

With the goal of devising data services, the work at hand develops SQL procedures for data enrichment, cleansing, business rule implementation, data versioning and automates open data publication based on the National Health Information System data warehouse.

The described development process lasts from March 2020 until the end of 2022 with majority of the work being realized within a short timeframe given the time pressure resulting from the Covid-19 crisis and following an iterative approach to development processes

The result of this work allows to publicize open data on Covid-19 infections and immunization statistics in an automated manner on a daily basis during the Covid-19 health crisis. The proposed solution is an improvement on and differs from previous methods of publishing health-care statistics which required manual compilation of data sets, by developing fully automated processes.

The thesis is in Estonian language and contains 47 pages of text, 8 chapters, 18 figures, 7 tables.

## Lühendite ja mõistete sõnastik

(SQL) INNER JOIN	SQL keeles päringu teostamisel INNER JOIN puhul pannakse tulemusse kui vasakpoolse tabeli vastav veerg = parempoolse tabeli vastav veerg ehk tagastatakse vasakpoolse ja parempoolse tabeli ühisosa.
(SQL) INSERT	SQL keeles andmete tabelisse lisamise operatsioon
(SQL) LEFT JOIN	SQL keeles päringu teostamisel väljastab kõik kirjed vasakpoolsest tabelist ja vastavalt JOIN tingimusele kirjed parempoolsest tabelist. Kui parempoolsest tabelist vastet ei leita, siis lisatakse vasakpoolsest tabelist võetud kirjele juurde parempoolse tulemuse tühjad NULL väljad.
(SQL) SELECT	SQL keeles andmete päringu teostamise operatsioon, mis tagastab andmed tabeli kujul.
(SQL) UPDATE	SQL keeles tabelis olevate andmete muutmise operatsioon
INF	Andmete esimene normaalkuju ( <i>1st normal form</i> )
Ag test	Antigeeni kiirtest ehk kiirtest on kiirdiagnostiline test, mis sobib ravipunktis tehtavaks testimiseks ja mis tuvastab otseselt antigeeni olemasolu või puudumise.
andmekvaliteet	Andmekvaliteet näitab, mil määral andmekarakteristikud rahuldavad teadaolevaid või eeldatavaid vajadusi kasutamisel ettemääratud tingimustes.
andmeladu andmeait	aruandluseks ja andmete analüüsimiseks kasutatav süsteem ning seda peetakse ärianalüüsi põhikomponendiks. Andmelaod on ühest või mitmest erinevast allikast pärinevate integreeritud andmete kesksed hoidlad
andmelett	data mart – on andmelaos kasutatav struktuuri ja andmete kättesaadavaks tegemise muster, mille eesmärk on kliendipoolsete päringute lihtsustamine
andmete esmane kasutus	isiku (tervise)andmete uurimine, mis seisneb otseselt eesmärgil kogutud andmete kasutamises – näiteks terviseandmete kasutamine raviteenuse osutamisel
andmete teisene kasutus	isiku (tervise)andmete uurimine, mis seisneb algselt muul eesmärgil kogutud andmete edasises töötlemises – näiteks terviseandmete analüütika ( <i>rahvusvaheliselt ja ka Eesti siseselt on erinevaid definitsioone, mis sõltuvad kasutusjuhust</i> )
ATC	<i>Anatomical Therapeutic Chemical Classification System</i> – ravimite klassifikaator

CDA	<i>Clinical Document Architecture</i> – meditsiinidokumentide andmevahetusstandard
Covid-19	akronüüm ingliskeelsest nimetusest <i>coronavirus disease</i> 2019 on rahvusvaheline lühend, millega tähistatakse koroonaviirushaigust, mida põhjustab koroonaviirus SARS-CoV-2.
CSV	<i>Comma Separated Values</i> . Komaga eraldatud väärtuste fail on piiritletud tekstifail, mis kasutab väärtuste eraldamiseks koma. Iga faili rida on andmekirje. Iga kirje koosneb ühest või mitmest komadega eraldatud väljast.
dBeaver	Andmebaasi haldustööriist ja SQL koodi arendusvahend
deserialiseerimine	Serialiseerimine on informaatikas objektide ja andmete teisendamine selliseks, et need oleks esitatavad järjest. Serialiseerimise vastand on deserialiseerimine, mille käigus taastatakse objektide ja andmete serialiseerimiselne seisund.
Docker	Teostab operatsioonisüsteemi tasemel virtualiseerimist ehk konteineriseerimist.
ECDC	<i>European Centre for Disease Prevention and Control</i> – Haiguste Ennetamise ja Tõrje Euroopa Keskus
EHIS	Eesti Hariduse Infosüsteem
EMA	<i>European Medicines Agency</i> – Euroopa Ravimiamet
Git	Vaba hajutatud versioonihaldustarkvara
GitLab	DevOps tarkvarapakett tarkvaraarenduses koodihalduseks, mis rakendab Git'i.
HL7	<i>Health Level 7</i>
immuniseerimine	protsess, kus vaksineerimise tulemusena saavutab inimkeha haiguse suhtes kaitse
JSON	<i>JavaScript Object Notation</i> ehk JSON on lihtsustatud andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeele alamhulgal. JSON on tekstvormingus ja programmeerimiskeelest sõltumatu. JSON-i looja on Douglas Crockford ja see on formaliseeritud kui RFC-4627
Kubernetes	avaliku lähtekoodiga konteinerite haldussüsteem, millega automatiseeritakse kasutuselevõttu, skaleeruvust ja konteineripõhiste rakenduste haldust
LOINC	Loogiliste vaatlusidentifikaatorite nimed ja koodid on andmebaas ja universaalne standard meditsiinilabori vaatluste tuvastamiseks.
LOINC	Logical Observation Identifiers Names and Codes – laialt kasutatav meditsiinivaldkonna terminoloogiastandard laboratoorsete testide ja kliiniliste toimingute kirjeldamiseks
materialiseerimine	Andmete salvestamine kettale.
MKM	Majandus- ja Kommunikatsiooniministeerium

ODS	Operational Data Store – andmelattu toodud alliksüsteemi andmete koopia kiht, reeglina ajaloostatud tabelid, kus andmeridadel on kehtivuse aeg.
OID	Rahvusvahelise Telekommunikatsiooniliidu ja ISO/IEC poolt standarditud identifikaatorimehhanismid mis tahes objektide, mõistete või "asjade" nimetamiseks globaalselt üheselt mõistetava püsiva nimega.
PCR	Polümeraasi ahelreaktsioon ehk PCR on meetod DNA või RNA järjestuse amplifikatsiooniks ehk kordistamiseks.
PHP	Skriptimiskeel, mida kasutatakse peamiselt serveripoolsetes lahendustes dünaamiliste veebilehtede loomisel.
Publitseerimiskeskus	Publitseerimiskeskus on tsentraalne meditsiinklassifikaatorite ja -standardite avaldamise keskkond, mida haldab Tervise ja Heaolu Infosüsteemide Keskus. Publitseerimiskeskuses avaldatakse Tervise infosüsteemis kasutusele võetavad standardid, klassifikaatorid ning loendid.
RHK10	Rahvusvaheline Haiguste Klassifikatsioon, versioon 10 ICD10
RR	Rahvastikuregister
SAP IQ	andmeanalüüsiks optimeeritud andmebaasitarkvara, endiste tootenimedega Sybase IQ ja SAP Sybase IQ
SARS	<i>Severe Acute Respiratory Syndrome</i> (SARS) (Raskekujuline äge respiratoorne sündroom – RÄRS)
SARS-CoV-2	Ägeda raskekujulise respiratoorse sündroomiga koroonaviirus 2 ehk SARS-CoV-2 on plussahelaline RNA-viirus, mis põhjustab inimestel koroonaviirushaigust COVID-19.
SNOMED CT	SNOMED CT või SNOMED Clinical Terms on süstemaatiliselt organiseeritud arvutiga töödeldav meditsiinitermine kogum, mis sisaldab koode, termineid, sünonüüme ja määratlusi, mida kasutatakse kliinilises dokumentatsioonis ja aruandluses.
SoM	Sotsiaalministeerium
SQL	<i>Structured Query Language</i> – programmeerimiskeel andmebaasipäringute ja andmetöötluse teostamiseks. Antud töös esitatud koodiviited kasutavad SAP IQ andmelao T-SQL (Transact SQL) dialekti.
STG	<i>Staging table</i> – andmelao tabel, kus toimub andmete töötlus ETL ja andmelaadimisprotsesside käigus
TA	Terviseamet
TAI	Tervise Arengu Instituut
TEHIK	Tervise ja Heaolu Infosüsteemide Keskus
TIS või TIKS	Tervise infosüsteem või tervise info kesksüsteem

vaktsineerimine	vaktsiini manustamine patsiendile, mille eesmärk on luua kehas vaktsiinivälditava haiguse suhtes kaitse
WHO	World Health Organization – Maailma Terviseorganisatsioon on ÜRO juures tegutsev rahvusvaheline tervishoiuorganisatsioon, mille eesmärk on inimeste tervise tagamine ja võitlus haiguste vastu.
XML	<i>Extensible Markup Language</i> – XML on W3C välja töötatud ja soovitatud standardne üldotstarbeline märgistuskeel, mille eesmärk on struktureeritud info jagamine infosüsteemide vahel, eelkõige Interneti veebipõhistes rakendustes. Märgistuste abil defineerib XML ühtseid dokumenditüüpe.
XQuery	Päringu- ja funktsionaalne programmeerimiskeel, mis teeb päringuid ja teisendab struktureeritud ja struktureerimata andmete kogusid, tavaliselt XML-vormingus, tekstina ja müüjaspetsiifiliste laiendustega muude andmevormingute jaoks. Keele on välja töötanud W3C XML Query töörühm.



## Sisukord

1	Sissejuhatus .....	13
2	Ülevaade terviseandmete ökosüsteemist .....	14
2.1	Tervise infosüsteem ja andmekogu ökosüsteem .....	14
2.2	Tervisedokumentide HL7 CDA standard .....	15
2.3	Tervise infosüsteemi andmeladu .....	16
2.4	Nakkushaiguste register (NAKIS).....	17
2.5	Andmeanalüüsi tarkvara .....	18
2.6	Riiklik avaandmete teabevärv .....	18
2.7	Arendusvahendid .....	19
3	Arendusvajaduste kaardistus .....	20
3.1	Koroonaviiruse SARS-CoV-2 testimine .....	21
3.2	Koroonaviirushaigusesse Covid-19 nakatumine .....	22
3.3	Covid-19 haiguse vastu vaktsineerimine .....	22
3.3.1	Immuunpreparaadi eristamine .....	23
3.3.2	Vaktsineerimise staatuse ja kuuri eristamine.....	23
4	Arendusprotsess, kaasatud osapooled ja arenduste skoop.....	25
4.1	Arendusprotsessi voog.....	25
4.2	Kaasatud osapooled .....	28
4.3	Töö skoop ja autori roll .....	29
5	Nõuete analüüs .....	30
5.1	Funktsionaalsed ärinõuded .....	30
5.1.1	SARS-CoV-2 viiruse testimine .....	30
5.1.2	Covid-19 haigusesse nakatumine .....	30
5.1.3	Covid-19 haiguse vastu vaktsineerimine .....	32
5.2	Mittefunktsionaalsed ärinõuded .....	32
5.2.1	Andmestiku seisu fikseerimine.....	32
5.2.2	Aruandluse valmimise aeg.....	32
5.2.3	Andmekvaliteedi probleemide lahendamine .....	32
5.2.4	Statistiline aruandlus aegridade alusel.....	32

5.2.5	Läbipaistvus aruandluse metoodika ja kokkupaneku osas .....	33
5.2.6	Isikuandmete kaitse ja kaudse tuvastamise riski maandamine .....	33
5.2.7	Aruannete dimensioonid ja asukohapõhine statistika, hõlmatus .....	33
5.3	Tehnilised mittefunktsionaalsed nõuded .....	33
5.4	Teostatavuse analüüs .....	33
6	Andmestike arendus .....	34
6.1	Koodi struktuur .....	34
6.2	Andmeletti andmete laadimise lahendus .....	34
6.3	Andmeleti ligipääsuõigused .....	36
6.4	Andmelaadimiste käiviti .....	36
6.5	Tõlketabeli lahenduse arendus .....	37
6.6	Aruannete alusandmete ajalise volatiilsuse vähendamine .....	39
6.7	Testide laadimine .....	41
6.8	Nakatumisandmestiku kokkupanek .....	42
6.8.1	Nakkusjuhtumite ajaline tekkimine .....	43
6.8.2	Nakkusjuhtumite tuvastamine nakatumise andmestiku pealt .....	44
6.8.3	Nakkusjuhtumite raporteerimise metoodika .....	47
6.9	Vaktsineerimisandmestiku kokkupanek .....	48
6.10	Andmestike ajaloostamine ja võrdlusalgoritm .....	50
6.11	Aruannete arendus .....	57
6.12	Avaandmete lahenduse arendus .....	58
6.13	Arendatud taaskasutatavate komponentide lähtekood .....	60
7	Kokkuvõte .....	61
	Kasutatud kirjandus .....	62
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	65
	Lisa 2 – Lähtekood .....	66

## Jooniste loetelu

Joonis 1. Tervise infosüsteemi andmearhitektuur (autor: TEHIK) .....	14
Joonis 2. Tervise infosüsteemi andmelao ülesehitus .....	17
Joonis 3. Andmeteenuste arendusprotsess .....	26
Joonis 4. Töö skoop ja andmeteenuste diagramm .....	29
Joonis 5. Andmeleti täitmise ja laadimise protsess .....	35
Joonis 6. Koodinäidis andmelaadimiste käivitusprotseduuri väljakutsungist .....	37
Joonis 7. Programmikoodi näide andmeelementide tõlkeprotseduuri välja kutsumiseks	39
Joonis 8. Koodinäidis funktsioon: f_Get_Statistics_Date .....	40
Joonis 9. Koodinäidis andmete esimese sisestamisaja leidmiseks .....	41
Joonis 10. Koodinäidis epikriiside leidmiseks, millel esineb vähemalt 2 konkreetset diagnoosi.....	43
Joonis 11. Nakatumise sündmuste grupeerimine nakkusjuhtudeks.....	44
Joonis 12. Koodinäide nakatumissündmuste määramine nakkusjuhtumitesse .....	47
Joonis 13. Nakatumise andmestiku statistilised hulgad koos kirjeldustega .....	47
Joonis 14. Vaktsineerimise dooside jaotus kuurideks .....	49
Joonis 15. Koodinäidis andmete võrdlevast ajaloostatud laadimisest JOIN meetodil ...	53
Joonis 16. Koodinäidis andmete võrdlevast ajaloostatud laadimisest UNION ALL meetodil .....	56
Joonis 17. Koodinäidis protseduuri p_IncrementalLoadFromLatest käivitamiseks .....	57
Joonis 18. Avaandmete avaldamise lahenduse arhitektuurijoonis .....	59

## **Tabelite loetelu**

Tabel 1. SARS-CoV-2 laboratoorsete analüüside valimi LOINC koodid.....	30
Tabel 2. Covid-19 nakatumise dokumenteerimise RHK-10 diagnoosikoodid.....	31
Tabel 3. Tõlketabeli struktuur .....	38
Tabel 4. Ajaloostatud laadimise näidistabeli "DMART" struktuur.....	50
Tabel 5. Näidistabeli "DMART" andmete lähteseis.....	50
Tabel 6. Näidistabeli "STG" viimase välja arvutatud seisu näidisandmed .....	51
Tabel 7. Ajaloostatud andmelaadimise "DMART" näidistabeli sisu pärast laadimist ...	51

# 1 Sissejuhatus

26. veebruaril 2020 tuvastati Eestis esimene SARS-CoV-2 nakkushaiguse juhtum laboratoorse testi teel [1]. Maailma oli tabanud koroonaviirusnakkus, mis kasvas kiirelt ülemaailmseks pandeemiaks [2].

Tervishoiukriisi juhtimiseks oli hädavajalik saada operatiivset ülevaadet nakkushaiguse levikust riigis. 2020-2022 perioodil tervisekriisi juhtimine eeldas ülevaate saamist nii viiruse levikust, nakatumisest kui ka nakkushaiguse vastu vaktsineerimisest. Kriisi algperioodil tuli statistika ja andmestike kokkupanekuks teha käsitööd kuna puudusid reaaliajalähedased lahendused, mis võimaldaksid andmeid automatiseeritult kokku tuua.

Erakordselt kõrge avaliku huvi ja nakkushaiguse kiire leviku tõttu oli vaja lühikese ajaga luua lahendusi, mis annaks nii kriisi juhtimiseks vajaliku info ning võimaldaks avalikkusele usaldusväärseid andmeid pakkuda. Antud töös arendan ja pakun välja võimaliku lahenduse.

Käesolev lõputöö käsitleb Covid-19 kriisi jaoks vajalike andmeteenuste arenduseks olulisi andmestike ning andmetöötlus- ja parendusmehhanismide realiseerimise võimalusi tuginedes tervise infosüsteemi andmelao tehnoloogiale.

Töö annab ülevaate terviseandmete ökosüsteemist, kaardistab andmelao arendusvajadusi, kirjeldab andmeteenuste arendusprotsessi ning analüüsib nõudeid. Töö käigus töötatakse välja andmelao SQL protseduurid ja teostatakse andmestike arendusi, mille tulemusena oleks võimalik esmakordselt automatiseeritult publitseerida Covid-19 nakatumise ja vaktsineerimise avaandmeid.

Töö autor on Tervise ja Heaolu Infosüsteemide Keskuses (TEHIK) tervisekriisi vältel lahendused arendanud, töötades seejuures koos klientidega Sotsiaalministeeriumist, Terviseametist ja teistest valdkonna asutustest.

## 2 Ülevaade terviseandmete ökosüsteemist

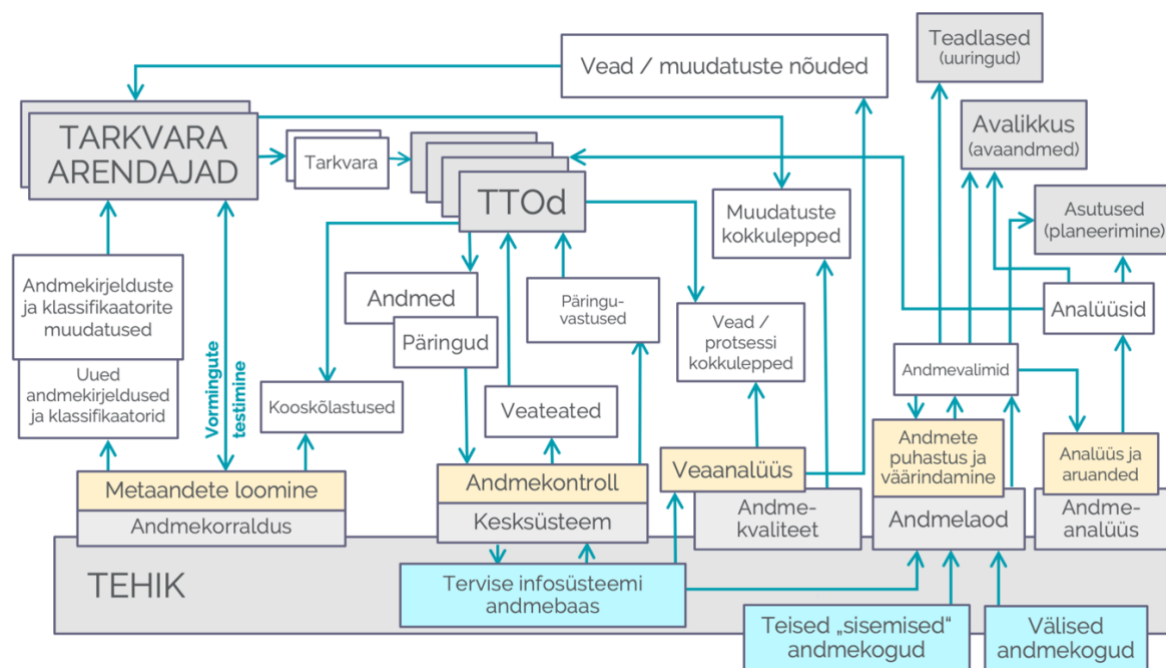
Käesolevas peatükis antakse ülevaade Eesti terviseandmete ökosüsteemist, olemasolevatest lahendustest, nendega seotud õiguslikest regulatsioonidest, tehnilistest lahendustest ning kitsendustest.

### 2.1 Tervise infosüsteem ja andmekogu ökosüsteem

Eestis tegutsevad tervishoiuteenuse osutajad (TTO) on tervishoiu teenuse korraldamise seaduse alusel kohustatud andmeid edastama Tervise infosüsteemi (TIS) [3]. Tervise infosüsteemi põhimääruse alusel on andmekogu vastutavaks töötajaks ehk lihtsustatult öeldes omanikuks Sotsiaalministeerium ja selle volitatud töötajaks ehk haldajaks on TEHIK [4]. Keskelt on terviseandmeid kogutud juba alates 2008 aastast [5]

Kõik TTO'd Eestis peavad tervishoiuteenuste korraldamise seaduse § 59<sup>2</sup> lõike 2 alusel esitama andmeid tervise infosüsteemi [3].

TIS arhitektuuri vaade on leitav arhitektuuridokumendist RIHA (Riigi Infosüsteemi Haldussüsteem) portaalis [6]. Terviseandmete ökosüsteemi kirjeldab järgnev joonis, mis võtab kokku TEHIK'u, TTO'de andmevahetuse ning andmekasutuse (Joonis 1).



Joonis 1. Tervise infosüsteemi andmearhitektuur (autor: TEHIK)



## 2.3 Tervise infosüsteemi andmeladu

Tervise infosüsteemi andmeladu on reguleeritud tervise infosüsteemi põhimääruse § 2<sup>1</sup> lõige 2 alusel [4].

Tervise infosüsteemi andmelao eesmärk on toetada riikliku tervisestatistika tegemist. Andmelattu laetakse andmeid tervise infosüsteemist, rahvastikuregistrist (RR), Eesti Hariduse Infosüsteemist (EHIS), Tervisekassa perearstide nimistut, Statistikaameti andmeid ning erinevaid avaandmete allikaid.

Andmelao andmelaadimised on realiseeritud valdavalt inkrementaalsete laadimistena. See tähendab, et andmelaos talletatakse muudatused ning regulaarsete laadimiste käigus laetakse andmelao andmebaasi vaid kirjete muutused.

Andmeladu koosneb järgnevatest kihtidest:

- andmete vastuvõtuala (*staging* – STG) – toimub viimaste muudatuste sisselaadimine, sobivale kujule viimine (transformatsioonid) ning andmete versioneerimine
- ajaloostatud andmete püsitalletusala (*Operational Data Store* – ODS) – andmed talletatakse alliksüsteemide andmemudeliga võimalikult sarnasel kujul, andmed on ajaloostatud – st. iga andmerida omab ajalist kehtivustempliit
- andmelettide ala (*Data Mart* – DM) – andmelettide koosseisu kaasatakse äriiselt defineeritud andmestike jaoks vajalikud allikad, viiakse läbi andmete transformatsioonid, andmete rikastamine erinevate allikate pealt, ärireeglite rakendamine
- presentatsioonikiht (*Presentation Layer*) – andmeanalüütikarakenduste poolt kasutatav ja andmete välistele teenustele avamise kiht

Tervise infosüsteemi andmelattu on andmete hõivamine ning nende pealt andmelettide ning presentatsioonikihi arendused tehtud vajadusepõhiselt (*on-demand*).

Andmete laadimise ETL (*Extract Transform Load*) protsessiks kasutatakse Hitachi Vantara Pentaho PDI<sup>1</sup> (Pentaho Data Integration) rakendusserverit, mis laeb

---

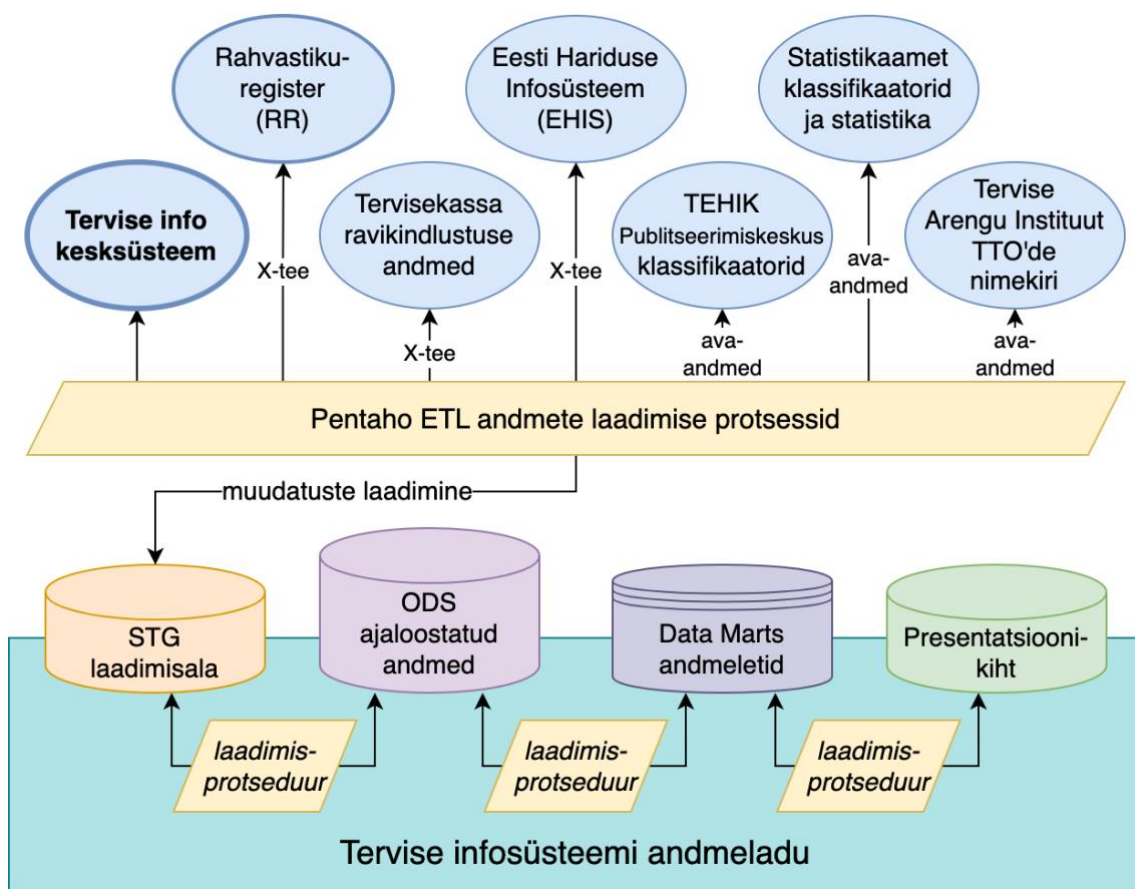
<sup>1</sup> <https://www.hitachivantara.com/en-us/products/dataops-software/data-integration-analytics.html>



alliksüsteemidest andmed, viib need andmelattu laadimiseks sobivale kujule ning seejärel kirjutab andmed baasi. Pentaho rakendus käivitab ka andmelao siseseid protseduure, mis kannavad andmeid edasi järgnevasse kihtidesse. Andmebaasiprotseduurid võimaldavad SQL käskude käivitamist juhtida andmebaasi serveri poolt.

ETL protseduuride käigus deserialiseeritakse XML dokumendid andmebaasi valdavalt 1 normaalkuju (1NF) vormi – sellest moodustub andmelao ODS kiht.

Tervise infosüsteemi andmelao ülesehitust andmeallikate ja andmearhitektuuri vaatest kirjeldab Joonis 2.



Joonis 2. Tervise infosüsteemi andmelao ülesehitus

## 2.4 Nakkushaiguste register (NAKIS)

Terviseameti haldusalasse kuulub Nakkushaiguste Register (NAKIS), mis on olemuselt veebipõhise kasutajaliidesega infosüsteem, kuhu TTO'd saavad sisestada nakkushaiguste teatisi. Nakkushaiguste register on kehtestatud põhimäärusega [9]. Nakkushaiguse

teatistele tuginedes on võimalik Terviseametil tegeleda põhimääruse alusel nakkushaiguste seire- ja tõrjeprotsessidega [10].

Nakkushaiguste register saab andmeid ka tervise infosüsteemist masinliidese kaudu [11].

## **2.5 Andmeanalüüsi tarkvara**

TEHIKus on terviseandmete analüüsiks kasutusel Tableau andmeanalüüsi tarkvara<sup>1</sup> ning tervise infosüsteemi andmelao „Statistikamoodul“ projektiga on varasemalt kasutusel iseteeninduslik analüütikalahendus, mis põhineb WebFocus tarkvaral.

Tableau tarkvara koosneb tööjaama rakendusest Tableau Desktop ning veebipõhisest rakendusest Tableau Server. Tableau Server põhjal pakutakse juhtimisinfo aruandeid ning ka avaandmeid tarbijatele. Tableau Server aruande loomiseks tuleb eelnevalt lahendus valmis arendada Tableau Desktop tarkvaras.

TEHIKus on rakendatud 2 Tableau Server keskkonda – esimene on ministeeriumi haldusala siseseks kasutamiseks ning kättesaadav ainult sisevõrgust. Teine Tableau Server teenindab avalikkust ning on mõeldud avaandmete põhjal tehtud aruannete avalikustamiseks.

## **2.6 Riiklik avaandmete teabevärv**

Majandus- ja Kommunikatsiooniministeeriumi eestvedamisel on loodud riiklik avaandmete teabevärv<sup>2</sup>. Lahendus on loodud eelkõige kataloogi vajadusi silmas pidades, võimaldades riigiasutustel viidata nende poolt publitseeritavatele avaandmetele [12].

Väiksematel asutustel on võimalik riikliku avaandmete teabevärava kaudu avaldada ka faile, neid käsitsi kasutajaliidese kaudu üles laadides [12]. Avaandmete teabevärv ei ole mõeldud suuremahuliste avaandmestike avaldamiseks vaid nendele viitamiseks.

---

<sup>1</sup> <https://www.tableau.com/products/server> - Tableau on andmeanalüüsi tarkvara

<sup>2</sup> <https://avaandmed.eesti.ee> – Eesti Avaandmete Teabevärv

## 2.7 Arendusvahendid

TEHIKus hallatakse loodava tarkvara ja konfiguratsiooni koodi GitLab keskkonnas. Git'i rakendates tagatakse koodi versioonihaldus. Git'i puhul kasutatakse *feature branch workflow* mustrit <sup>1</sup>.

Tööülesannete kirjeldamine ja jälgimine toimub Jira projektijuhtimistarkvara abil.

Koodikirjutamise vahenditeks on vabavaraline tarkvara Visual Studio Code. Andmebaasitoimingute jaoks on kasutusel dBeaver Enterprise rakendus.

---

<sup>1</sup> [https://docs.gitlab.com/ee/topics/gitlab\\_flow.html#feature-branch-workflow](https://docs.gitlab.com/ee/topics/gitlab_flow.html#feature-branch-workflow)

### 3 Arendusvajaduste kaardistus

Covid-19 tervishoiukriisi juhtimiseks perioodil 2020 – 2023 on olnud vajalik operatiivne juhtimisinfo nakkushaiguse leviku, haigestumise ja haiguse vastu vaktsineerimise vastu. Teabevajadus on olnud vajalik poliitika kujundamiseks, operatiivsete juhtimisotsuste langetamiseks ning avaliku huvi rahuldamiseks.

Viiruse leviku algperioodil sai Terviseamet, Sotsiaalministeerium ja seotud osapooled infot nakkushaiguste laboritest väljavõtete kujul, siis üsna peatselt sai selgeks, et testimise mahu kasvades ning operatiivsest andmevajadusest tingituna ei ole väljavõtete käsitööna kokkupanek ning nende analüüsimine enam jätkusuutlik.

Eestis on juba aastast 2008 olemas riiklik tervise infosüsteem (TIS), kuhu kõik tervishoiuteenuse osutajad (TTO'd) andmeid esitavad. Samuti on tervise infosüsteemi juurde arendatud andmeladu ning -analüütikarakendus, kuid sealt kvaliteetse juhtimisinfo kättesaamine ilma täiendavate arendusteta ei ole võimalik.

Põhiliseks lahendatavaks probleemiks on operatiivse andmevajaduse rahuldamise jaoks valmislahenduste puudumine.

Alternatiivne lähenemine oleks käsitööna andmete analüüsimine, mis on arvestades teabevajaduse operatiivsust liialt ajamahukas ning veaaldis. Teabevajaduse operatiivsus tähendab, et iga päeva hommikul pidi viimase 24 tunni jooksul laekunud andmed fikseerima kella 07:00 seisuga ning kõik väljundid pidid hiljemalt kell 08:00 olema kättesaadavad Sotsiaalministeeriumi haldusala siseseks kasutamiseks.

Teabevajadus jaguneb kolmeks seotud alamvaldkonnaks:

- SARS-CoV-2 testimine ja testide tulemused
- Covid-19 haigusesse nakatumine ja selle diagnoosimine
- COVID-19 haiguse vastu vaktsineerimine

Iga nimetatud alamvaldkonna osas on vajadused spetsiifilised vastavalt ärinõuetele ja kasutusjuhtudele.

### 3.1 Koroonaviiruse SARS-CoV-2 testimine

Esimene SARS-CoV-2 koroonaviiruse nakatumine tuvastati Hiina Rahvavabariigi Hubei provintsis 2019 aasta lõpus [13].

Nakkushaiguse kiire leviku tõttu kuulutas Maailma Terviseorganisatsioon (WHO – World Health Organization) 11. märtsil 2020 välja Covid-19 pandeemia [14].

Nakkushaiguse tuvastamiseks oli välja töötatud PCR test, mida hakkasid Eestis läbi viima Terviseameti nakkushaiguste labor ja Synlab Eesti ning teised tervishoiuteenuse osutajad, sealhulgas mitmed haiglad [15] [16].

Info testimise ja laborianalüüsi tulemuse kohta laekus riikliku terviseandmete ökosüsteemi kolmel viisil:

- Tervise infosüsteemi saatekirja vastuse dokumendina [17]
- Nakkushaiguste registrisse (NAKIS) nakkushaiguse teatisena [9]
- labori koostatud tabel-väljavõtetena

Testimise algusperioodil saatsid analüüse läbi viivad laborid andmed Terviseametisse tabelväljavõtete kujul, kus oli teada proovivõtu koht, aeg ning laborianalüüsi tulemus [18].

Nakkushaiguste registrisse edastati nakkushaiguse teatis vaid positiivsete testitulemuste kohta, samas testimismahu hindamiseks oli riiklikus statistikas vaja teada ka negatiivseid testitulemusi.

Algne statistika koostati Terviseameti nakkushaiguste tõrje osakonna spetsialistide poolt käsitsi, agregeerides statistika erinevate laborite andmete pealt. Paraku aga testimismahu kasvades ning laborite lisandumisel ei ostunud enam otstarbekaks ega jätkusuutlikuks käsitööna statistika koostamine.

Samuti käsitsi statistika koostamine võib põhjustada vigu ning eksitada avalikkust, seeläbi vähendades avaldatud andmete usaldusväärsust. Kõrgenenud avaliku huvi ning algse statistika avaldamisega seotud probleeme kajastasid ka mitmed meediaväljaanded [19] [20].

Ühe alternatiivina oli variant kasutada tervise infosüsteemi andmelao analüütikakeskkonda, kuid toorandmete pealt otse analüütika tegemine osutus tehniliselt liialt keerukaks.

### **3.2 Koroonaviirushaigusesse Covid-19 nakatumine**

Riikliku testimisstrateegia muutmise tulemusena 2022 aastal hakkasid tervishoiuteenuse osutajad dokumenteerima Covid-19 haigusesse nakatumist ka sümptomaalse pildi olemasolul. Testimist oli võimalik läbi viia ka SARS-CoV-2 Ag antigeenitestide abil. Kui SARS-CoV-2 PCR testimist viisid läbi laborid ning andmed laekusid tervise infosüsteemi automatiseeritud liideste abil, siis koduseks ja meditsiiniliseks kasutamiseks mõeldud SARS-CoV-2 Ag kiirtestide edastamine ei olnud automatiseeritud. Kiirtesti tulemus saab olla aga diagnoosi määramise üheks aluseks.

TTO'd dokumenteerisid nakatumist ravikokkuvõtte ehk epikriisi dokumendil diagnooside kombinatsioonina. Terviseameti poolt väljastatud juhendites oli toodud kirjeldus, milliste RHK10 (rahvusvaheline haiguste klassifikatsioon, 10. versioon) diagnoosi klassifikaatorile vastavate diagnooside alusel Covid-19 nakatumist dokumenteerida [21].

Tervikliku nakatumise kohta ülevaate saamiseks lahendust ei olnud ning see tuli arenduste käigus realiseerida.

### **3.3 Covid-19 haiguse vastu vaktsineerimine**

Covid-19 haiguse vastu alustati vaktsineerimist 27. detsembril 2020 2 [22]. Vaktsineerimine dokumenteeriti TTO'de poolt immuniseerimise teatise dokumendil, mis laekub tervise infosüsteemi [23].

Immuniseerimise teatise dokumendil on võimalik dokumenteerida muuhulgas:

- Vaktsiinvälditav haigus (kood 219, nimetus „Covid-19“)
- Immuunpreparaadi (vaktsiini) ATC kood
- Immuunpreparaadi (vaktsiini) nimetus ja partii number
- Manustatud doosi kordsuse number

- Märge, kas immuniseerimine on lõpetatud või on kokku lepitud uus immuniseerimise aeg

Vaktsineerimise sündmused moodustavad vaktsineerimise kuuri. Haiguse vastu esmane vaktsineerimine võib toimuda sõltuvalt patsiendi vanusest, immuunpreparaadist ja patsiendi varasemast või esimese doosi järgest läbipõdemisest vastavalt kas 1, 2 või 3 doosiga. Kui vaktsineerimise kuur on läbitud loetakse patsient selles kuuris täielikult vaktsineerituks.

Hilisemad vaktsineerimise käigus manustatud doosid loetakse tühustusdoosideks, mille eesmärk on immuunsuse tugevdamine.

Esmase andmeanalüüsi käigus tuvastati, et vaktsineerimise teatiste dokumenteerimise kvaliteet on ebahühtlane.

### **3.3.1 Immuunpreparaadi eristamine**

Immuunpreparaadi eristamise vajadus on tingitud vajadusest eristada vastavalt immuniseerimise reeglitele eranditega, kus teatud vaktsiinide korral on vaktsineerimise 1 kuur 1 doosiline (Covid-19 Vaccine Janssen) või 3 doosiline (Comirnaty väikelaste vaktsiin).

Immuunpreparaadi eristamiseks on andmetele tuginedes kaks võimalust – lähtuvalt immuunpreparaadi ATC koodist või immuunpreparaadi nimetusest. Covid-19 vastu vaktsineerimise algperioodil oli kõigi Eestis kasutatavate vaktsiinide ATC kood sama. ATC kood antakse ravimi müügiloa alusel Euroopa Ravimiameti (EMA – European Agency of Medicines) poolt.

Immuunpreparaadi eristamiseks jääb seega võimalus kasutada ainult selle nime. Paraku nimekuju edastamine on mitmetes terviseteenuse osutajate infosüsteemides realiseeritud vabatekstina ja seetõttu esineb vabateksti sisestamisel vigu.

### **3.3.2 Vaktsineerimise staatuse ja kuuri eristamine**

Immuniseerimise teatisel dokumenteeritakse mitmenda doosi on patsient saanud konkreetse haiguse vastu vaktsineerimisel. Dokumenteerimise nõue eeldab aga seda, et TTO'd on dokumenteerimise hetkel kindlaks teinud, mitmes doos patsiendile manustati.

Meditsiinilises vaates teostatakse kõik Covid-19 vastu vaktsineerimised sama immuunpreparaadiga, see tähendab, et näiteks nii 1, 2 kui 3 doosi manustamiseks saab kasutada sama vaktsiini. Eraldi dooside jaoks ei ole eraldi vaktsiine. Arsti ja patsiendi nõusolekul võib vaktsineerija küll manustada patsiendile erinevaid vaktsiine näiteks tõhustusdoosiks.

Arvestades suuremahulise vaktsineerimiskampaaniaga 2022 aasta esimeses pooles esines dokumenteerimisel mitmeid puudujääke – seejuures doosi numbri määramisel. Näiteks võis patsient saada esimese doosi oma perearsti juures ja seejärel minna ise vaktsineerimiskeskusesse, kus manustati teine doos. Suure dokumenteerimise mahu juures võis tekkida olukordi, kus teine doos dokumenteeriti vigaselt kui 1. doos.

Andmekvaliteedi probleemidest lähtuvalt tuleb seetõttu statistika jaoks dooside jaotus üle arvutada.



## 4 Arendusprotsess, kaasatud osapooled ja arenduste skoop

Arvestades arenduste aegkriitilist iseloomu, mis oli tingitud kriisiolukorras kõrgendatud andmevajadusest tuli rakendada lühikese tagasisideahelaga arendusprotsessi. Arendusprotsessi oli kaasatud osapooli nii Sotsiaalministeeriumi haldusalast kui ka väliseid osapooli, kellega lahendusi valideeriti. Sotsiaalministeeriumi haldusalas ning tervise infosüsteemi volitatud töötajana on arendusi läbiviivaks asutuseks Tervise ja Heaolu Infosüsteemide Keskus (TEHIK), mis tegutseb põhimääruse alusel [24].

### 4.1 Arendusprotsessi voog

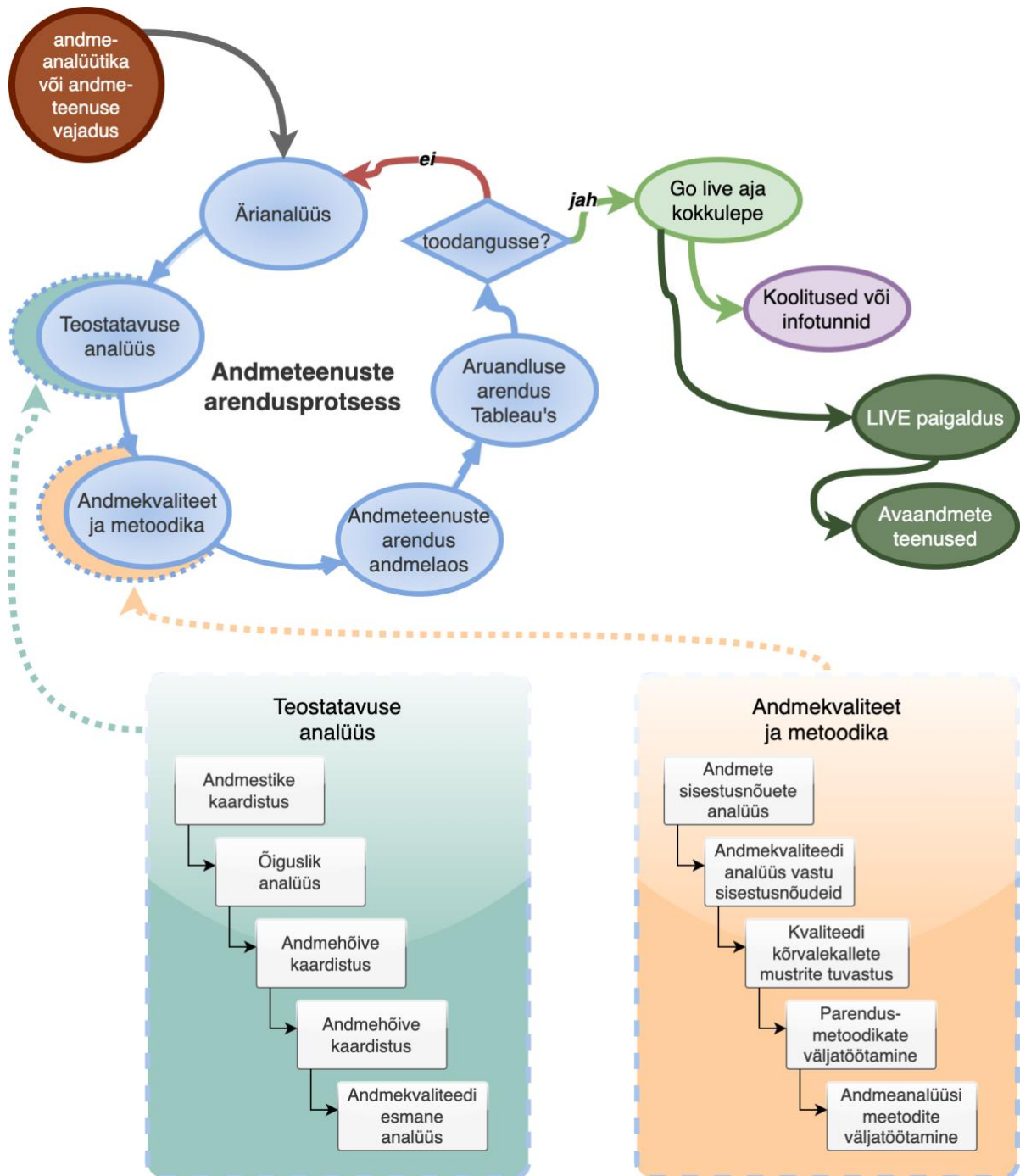
Analüüsi- ja arendustöö toimub põimunult agiilsele tarkvaraarendusmetoodikale toetudes. Arendusmuustrite praktikatest võib arendusprotsessi võrrelda DataOps praktikatega [25], [26]. Arendusprotsessi eesmärk on lühikeste iteratsioonide kaupa luua väärtust kliendile, seejuures on kvaliteedi tagamiseks arendusprotsessis kontrollpunktid, kus veendutakse, kas on jõutud tulemuseni, mis on kasutuselevõtuks piisavalt küps.

Arendusprotsessi üks väljakutseid on andmekvaliteedi võimalike probleemidega tegelemine, kuna olemuslikult selguvad need arendustöö käigus, siis nende parendusmeetmete rakendamisele kuluvat aega on esmase ärianalüüsi käigus raske kindlaks teha.

Andmekvaliteedi probleemide lahendamine ja andmestike analüüsimeetoodika välja töötamine on oluline tagamaks väljundandmete ja statistika usaldusväärsust.

Andmeteenuste arendusprotsessiga paralleelselt toimuvad ka teised terviseandmete ökosüsteemi protsessid nagu andmekvaliteedi nõuete arendus, andmevahetusvormingute ja -standardite arendus, mis on siinse töö skoopist väljas, kuid on ülevaatlilikult kirjeldatud diagrammil Joonis 1.

Arendusprotsessi üldistatud toimemudel on kirjeldatud järgnevalt diagrammil Joonis 3:



Joonis 3. Andmeteenuste arendusprotsess

1. Ärianalüüs: lähteülesande kokkupanek ja sisend tellijalt

- Andmevajaduse (andmeanalüütika, andmeteenused) kindlaks tegemine
- Lähteülesande püstitamine
- Lähteülesande analüüs
- Olemasolevate lahenduste kõrvutamise, arendusvajaduse kindlaks tegemine
- Aruandlusvajaduse väljundi analüüs

2. Teostatavuse analüüs
  - Andmestike kaardistus – kas algandmed on olemas ja millistes andmekogudes
  - Õiguslik analüüs – kas arendusülesande jaoks on õiguslik alus lähteandmetele
  - Andmehõive ja olemasolu kaardistus – mis koosseisus on lähteandmed olemas
  - Andmekvaliteedi esmane analüüs
3. Andmekvaliteet ja meetodika
  - Algsed dokumenteerimismõõded, mille põhjal andmeid hõivata
  - Andmekvaliteedi analüüs dokumenteerimismõõdet vastu
  - Andmekvaliteedi kõrvalekalde mustrite tuvastamine
  - Kõrvalekallete mustritele parendusmeetodite leidmine
4. Andmete arendus andmelao platvormil
  - ETL protsesside arendus
  - Andmete modelleerimine
  - Andmete puhastamis- ja rikastamisalgoritmide välja töötamine
  - Andmelaadimisprotseduuride realiseerimine
  - Aegriidide ja statistiliste väljundite arendus
  - Aruannete sisendandmestike arendus (sh. avaandmete jaoks vajadusel)
5. Aruandluslahenduse arendus Tableau analüütikatarkvaras
  - Tableau aruannete visuaalide loomine
  - Koostöös kliendiga aruannete täiendamine
6. Kontrollpunkt – kas arendatud lahenduses on jõutud soovitud tulemusteni ja sellega liigutakse toodangusse?
  - „Go live“ aja kokku leppimine, millal lahendus toodangusse läheb
  - Tarbijatele koolitused ja infotunnid, et andmeväljunditest arusaamist luua
  - LIVE paigaldus – millal muudatused avalikult rakenduvad
  - Avaandmete teenused – avaandmete kättesaadavaks muutmine

## 4.2 Kaasatud osapooled

Arendusprotsessi on kaasatud teabevajadusega osapooled, kellega koostöö toimub koosolekute formaadis, mille käigus kaardistatakse nõuded, analüüsitakse andmekvaliteeti ja kellega valideeritakse arendatud lahenduste tulemeid.

Teabevajadusega osapooled jagunevad vastavalt rollile järgnevalt:

- Sotsiaalministeerium – tervisekriisi juhtimine
- Terviseamet – tervisekriisi juhtimine, riiklike Covid-19 ülevaadete tegemine, rahvusvahelise aruandluse teostamine:
  - *European Centre for Disease Prevention and Control* (ECDC)
  - *European Medicines Agency* (EMA)
  - *World Health Organization* (WHO)
- Tervisekassa (endise nimega Haigekassa) – tervisekriisi korraldus, riiklik Covid-19 vaktsiinivaru juhtimine
- Riigikantselei – vabariigi valitsuse informeerimine tervisekriisi alasel
- Vabariigi valitsuse teadusnõukoda (Irja Lutsar, Krista Fischer jt.) – Covid-19 leviku prognoosmudelite tegemine
- Avaandmete tarbijad:
  - Koroonakaart.ee (Keegan McBride, MTÜ Open Knowledge Estonia)<sup>1</sup>
  - Tartu Ülikooli koroonaviiruse analüüsivahendid<sup>2</sup>
  - Postimees
  - Eesti Rahvusringhääling (ERR)
  - jt. meediaväljaanded ning andmehuvilised

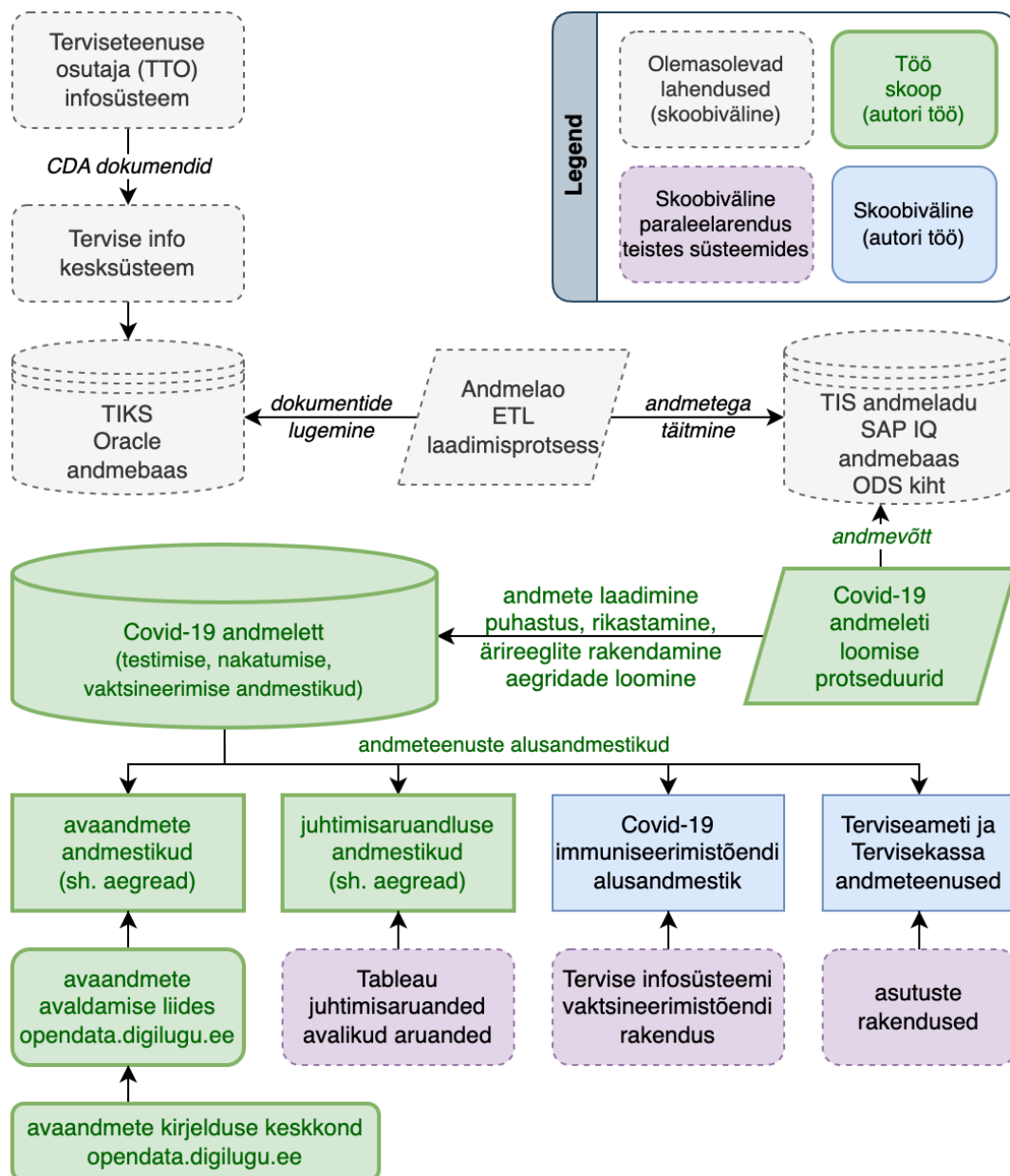
---

<sup>1</sup> <https://koroonakaart.ee/et> - MTÜ Open Knowledge Estonia ja kogukonna poolt arendatud Covid-19 statistika esitluse keskkond

<sup>2</sup> <https://koroona.ut.ee/> - Tartu Ülikooli teadlaste arendatud koroonaviiruse analüüsivahendid

### 4.3 Töö skoop ja autori roll

Töö autor on TEHIK’u andmeanalüüsi- ja andmekorralduse osakonna töötaja, vastutades andmeladude arenduste eest ning Covid-19 tervisekriisi vältel arendanud andmelao lahendused. Töö skoobi kitsendamiseks on võetud fookusesse Covid-19 andmeleti arenduse, mis on statistika ja juhtimisinfo väljundite aluseks. Skoobist jääb välja Tableau aruandlus, mida arendasid TEHIKu andmeanalüüsi talituse töötajad. Samuti ei ole skoobis andmevahetusteenused ega juba eksisteerivad lahendused tervise infosüsteemi andmelao andmelaadimiste „ODS“ kihini. Skoopi illustreerib Joonis 4, kus roheline värviga on toodud autori arendatud osa, mis kuulub töö skoopi.



Joonis 4. Töö skoop ja andmeteenuste diagramm

## 5 Nõuete analüüs

Äriliste nõuete ja vajaduste kaardistus ning analüüs toimus koostöös Terviseameti ja Sotsiaalministeeriumi valdkonnaspetsialistide ning -juhtidega. Koostöövormiks olid analüüsikoosolekud kui ka e-kirja vahetus. Täiendavaid nõudeid kaardistati erinevatest allikatest nagu Terviseameti juhendmaterjalid, WHO (World Health Organization), ECDC (European Centre for Disease Control), EMA (European Medicines Agency) allikatest. Analüüsitulemid valideeriti koostöös tellijatega.

Ärinõuete analüüs on kirjeldatud analüüsi tulemina ning hõlmab endas esmase andmekvaliteedi analüüsi käigus selgunud probleeme.

### 5.1 Funktsionaalsed ärinõuded

#### 5.1.1 SARS-CoV-2 viiruse testimine

Aruandlusesse tuleb hõlmata SARS-CoV-2 viiruse testimise andmed laborianalüüside tulemustelt, mis edastatakse tervise infosüsteemi saatekirja vastuse dokumendi kujul.

Hõlmamise aluseks on kõik analüüsid, mille LOINC (*Logical Observation Identifiers Names and Codes*) koodiks üks järgnevatest:

Tabel 1. SARS-CoV-2 laboratoorsete analüüsides valimi LOINC koodid

<b>Analüüs</b>	<b>LOINC kood</b>
SARS koroonaviirus 2 RNA	94309-2
SARS koroonaviirus 2 RNA kurguloputusvedelikust	L-5422
SARS koroonaviirus 2 RNA (seireuuring)	A-5509
SARS koroonaviirus 2 Ag (ribaanalüüs)	94558-4
SARS koroonaviirus 2 Ag	96119-3

#### 5.1.2 Covid-19 haigusesse nakatumine

Covid-19 haigusesse nakatumist dokumenteeritakse TTOde poolt ambulatoorse epikriisi dokumendil vähemalt 2 diagnoosikoodi kombinatsiooniga. Diagnoosikoodid on

määratletud Rahvusvahelise Haiguste Klassifikaatori versioon 10 (RHK-10) alusel, mida Eestis haldab Sotsiaalministeerium<sup>1</sup>. RHK-10 põhineb rahvusvahelisel International Classification of Diseases (ICD-10) põhjal, mida haldab Maailma Terviseorganisatsioon (WHO)<sup>2</sup>. Diagnoosikoodid on klassifikaatoris hierarhilised ning hierarhiat väljendatakse klassifikaatori koodis.

Tabel 2. Covid-19 nakatumise dokumenteerimise RHK-10 diagnoosikoodid

<b>RHK-10 diagnoos loetelust</b>	<b>kombinatsioonis mistahes RHK-10 diagnoosiga</b>	
U07.1	J*	R07*
U07.2	B34.9	R09*
B34.2 (ainult perioodil 2020-01-01 kuni 2020-05-01)	B97.8	R43*
B97.2 (ainult perioodil 2020-01-01 kuni 2020-05-01)	M25.5	R50*
	M79.1	R51*
	R05*	R52*
	R06*	R53*
U07.1	B33.8	B34.2
U07.2	B33	B34.8
	B34	B97.2
U07.1	A08.4	R10.3
	A09	R10.4
	B34.1	R10
	G93.3	R11
	M54	Z22.8
	R10.1	Z22

\* tärniga märgitud diagnoosikoodid hõlmavad hierarhias ka kõiki alamdiagnoose

Covid-19 diagnoos võib seega määratud olla näiteks kombinatsioonis: U07.1 ja mistahes J diagnoos; või 2020 aasta alguses näiteks B97.2 koos M79.1 diagnoosiga jne.

Epikriiside otsingukombinatsioone on vastavalt  $4*14 + 2*6 + 1*12 = 80$ , seejuures on osad koodid laiemal ulatusega hõlmates RHK-10 hierarhias ka alamdiagnoose.

<sup>1</sup> Eesti RHK-10 koodide kataloog asub internetiaadressil <https://rhk.sm.ee/>

<sup>2</sup> <https://www.who.int/standards/classifications/classification-of-diseases> WHO International Classification of Diseases

### **5.1.3 Covid-19 haiguse vastu vaksineerimine**

Covid-19 haiguse vastu vaksineerimist dokumenteerivad TTO'd immuniseerimise teatise dokumentidel, mis edastatakse tervise infosüsteemi.

Eristamaks immuniseerimise teatistelt Covid-19 haiguse vastu vaksineerimist tuleb immuniseerimise teatise filtreerida vaktsiinvälditava haiguse klassifikaatori alusel. Covid-19 klassifikaatori väärtuseks on kood 219.

## **5.2 Mittefunktsionaalsed ärinõuded**

### **5.2.1 Andmestiku seisu fikseerimine**

Andmestiku seis tuleb igapäevaselt fikseerida kell 07:00 hommiku seisuga. Seejuures eelneva kuupäeva aruandlusesse kuuluvad andmed, mis on laekunud vahemikus eelmise kuupäeva 07:00:00 (k.a.) kuni käesoleva kuupäeva 07:00:00ni (v.a.).

Hommikuse 07:00 seisu fikseerimise vajadus oli tingitud sellest, et laborites analüüsiti suur hulk eelneval päeval kogutud proovimaterjali tihti öhtusel ja öisel ajal.

### **5.2.2 Aruandluse valmimise aeg**

Tingituna Vabariigi Valitsuse komisjoni istungitest ning Terviseameti nõuetest pidi hommikune aruandlus olema valmis hiljemalt kell 8:00. Lisaks sellele pidi olema täiendavat puhveraega, et analüütik saaks aruande üle vaadata – selleks oli hinnanguliselt ajaaken kuni 20 minutit. Aruande üle vaatamine oli vajalik selleks, et veenduda, kas kõik laborianalüüsid on tähtaegselt laekunud. Võis esineda juhtumeid, kus mõne TTO andmete edastamise liides tõrkus ning öisel ajal andmed ei laekunud plaanipäraselt.

### **5.2.3 Andmekvaliteedi probleemide lahendamine**

Käsitsi dokumenteeritud andmete puhul eksisteeris risk, et osa andmeid oli kas vigaselt või ebatäpselt dokumenteeritud. Seetõttu tuli välja töötada meetodika, mis kõrvaldaks enamlevinud vead dokumenteerimisel.

### **5.2.4 Statistiline aruandlus aegridade alusel**

Statistilise aruandluse väljundis peaks aruandluse dimensioonide lõikes iga kuupäeva kohta kajastuma statistika, isegi, kui sellel kuupäeval ei ole ühtegi sündmust aset leidnud.



Vajadus on tingitud sellest, et iga päeva kohta peab näitama ka kumulatiivselt kogunenud sündmuste hulka.

#### **5.2.5 Läbipaistvus aruandluse metoodika ja kokkupaneku osas**

Andmestike kokkupaneku metoodika peab olema avaandmete keskkonnas dokumenteeritud, tagamaks suuremat läbipaistvust ja usaldust aruandluse kohta. Samuti on metoodika kirjeldamine vajalik andmete võrreldavuse tagamiseks rahvusvahelisel tasandil. Dokumentatsioon tuleb seetõttu avaldada nii eesti kui inglise keeles.

#### **5.2.6 Isikuandmete kaitse ja kaudse tuvastamise riski maandamine**

Avaandmete ja aruannete koosseisus ei tohi olla ei otseselt tuvastatavaid isikuandmeid ning tuleb vältida isiku kaudse tuvastamise riski. Seetõttu tuleb avaandmestikest isikuline seos kõrvaldada ning avaldada ainult agregeeritud statistikat.

#### **5.2.7 Aruannete dimensioonid ja asukohapõhine statistika, hõlmatus**

Enamike statistilisi aruandeid tuleb esitada statistiliste agregaatidena

### **5.3 Tehnilised mittefunktsionaalsed nõuded**

Arendustööde teostamisel peab arvestama TEHIK’u mittefunktsionaalsete nõuete [27] ja IT profiilist [28] lähtuvate nõuetega.

### **5.4 Teostatavuse analüüs**

Kõigi aruannete alusandmestike andmed kuuluvad tervise infosüsteemi andmekoosseisu ning on laetud tervise infosüsteemi andmelattu.

Teostatavuse keerukust kasvatab kõrvalekalded andmekvaliteedist. Andmekvaliteediks loetakse andmete vastavust kokkulepitud standarditele. Andmekvaliteedi kõrvalekalletega tegeletakse arendustöö etapis.

## 6 Andmestike arendus

Andmestike arendus toimus tervise infosüsteemi andmelao SAP IQ platvormil. Andmete laadimise ja transformeerimise lahendus on realiseeritud SQL protseduuridena.

### 6.1 Koodi struktuur

Võimaldamaks SQL koodi paigaldust erinevatesse andmebaasikeskkondadesse ja skeemidesse on SQL koodis andmebaasi skeemi nimetus parametrizeeritud `{schema}` muutuja abil. *dBeaver* rakenduses koodi paigaldades küsib rakendus kasutajalt muutuja väärtustamist <sup>1</sup>.

### 6.2 Andmeleti andmete laadimise lahendus

Andmelao arhitektuurist lähtuvalt arendati Covid-19 andmestikud andmeleti lahendusena. Andmeleti arenduse meetodika tähendab, et andmed kopeeritakse andmelattu sisse laetud andmete (ODS) kihist edasi järgmisesse skeemi, kus andmed viiakse sobivale kujule ja kus rakendatakse andmetöötlusalgoritme.

Füüsiliselt on andmelett eraldi skeemis, mis võimaldab protseduure, tabeleid, andmeid ja andmetöötlust hoida väljaspool andmelao teisi kihte, tagades seeläbi isoleerituse.

Andmeleti skeemis paiknevad tabelid, kuhu on laetud andmestikud materialiseeritakse. Andmed denormaliseeritakse esimese normaalkuju (1NF) kujule, kuna see on andmeanalüütikas kasutamiseks optimaalsem – lühendades päringute aega, vähendades JOIN operatsioonide hulka [29].

Valik 1 normaalkuju kuju kasuks osutus mitmel põhjusel:

- andmeleti andmed peavad olema loogiliselt seotud, nende tihe sidumine annab andmelao vaatest optimeeritud lähenemise, kus kõik äri loogika rakendamiseks ning aruandluse ehitamiseks vajalikud andmed on ühe juhtumina ühel real olemas

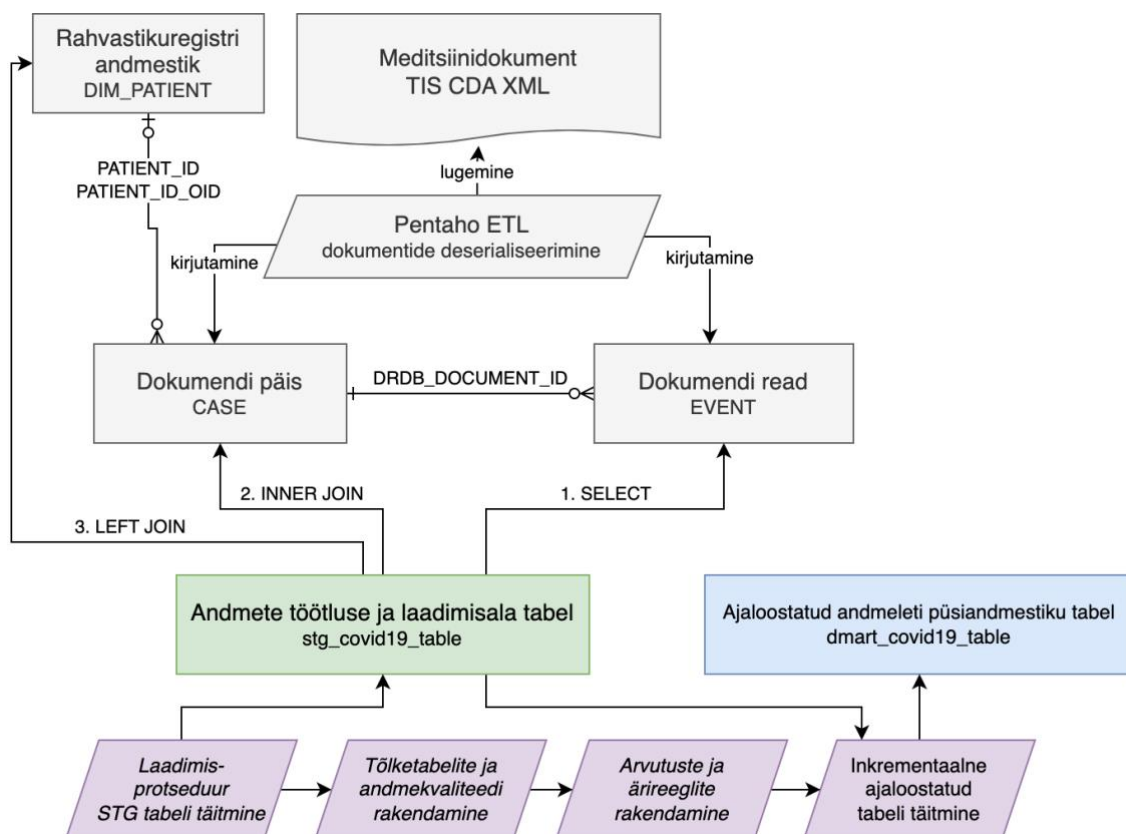
---

<sup>1</sup> <https://dbeaver.com/docs/wiki/SQL-Execution/#dynamic-parameter-bindings>

- andmete versioneerimine on lihtsam, kuna on võimalik võrrelda tervet rea koosseisu tervikuna
- aruandlusrakendustes sobib 1 normaalkuju kuju hästi tarbimiseks, kuna kõik vajalikud dimensioonid on ühes tabelis esindatud ning analüütilalahenduse arendaja ei pea täiendavaid andmeid JOIN operatsioonide abil ühendama.

Joonis 5 kirjeldab andmeleti tabeli täitmist. Sarnane protsess toimub kõigi kolme andmestiku täitmisel.

Andmeleti tabelites on esindatud veerud alliktabelitest, mida kasutatakse hiljem andmeanalüüsi või andmetöötlemise eesmärkidel. Täiendavalt on tabelites olemas arvutuslikud veerud, mida allikandmetes ei olnud. Tekstiliste väärtuste parandamiseks on tabelites sihtmärkveerud, kuhu andmekvaliteedi parandatud väärtused määratakse – seejuures säilivad ka originaalväärtused, et hiljem oleks võimalik analüüsida andmekvaliteedi kõrvalekaldeid.



Joonis 5. Andmeleti täitmise ja laadimise protsess

1. Andmeleti töötlusalas on kasutatakse STG (staging, laadimisala) tabelit, mis protseduuri käivitades tühjendatakse varasematest tulemustest.

2. INSERT FROM SELECT päringuga laetakse andmed alliktabelitest. SELECT päringu sihtmärk on sündmuste tabel, millele INNER JOIN abil liidetakse ühisosa hulk dokumendi päisest, millega sündmus andmelattu laeti. Seejärel liidetakse LEFT JOIN abil kirje Rahvastikuregistri (RR) andmestiku tabelist DIM\_PATIENT patsiendi välisvõtme alusel. Kuna meditsiinidokument võib kirjeldada ka välismaalasega toimunud sündmust, keda RR tabelis ei eksisteeri, siis ei pea vastet leiduma.
3. SELECT päringu tulemus on olemuslikult denormaliseeritud kuju ning see lisatakse INSERT käsu abil STG tabelisse.
4. Rakendatakse tõlketabelite käivitust.
5. Rakendatakse arvutusi ja ärireeglite rakendamist.
6. Kui STG alal on andmed valmis, siis tuvastatakse muudatused võrreldes kehtiva seisuga ODS tabelis ning selle sisu uuendatakse.

### **6.3 Andmeleti ligipääsuõigused**

Andmeleti jaoks on loodud teenuskonto, millel on „SELECT“ ehk lugemisõigusega ligipääs allikandmete tabelitele läbi vastava rolli.

Teenuskontol on kirjutamisõigusega ligipääs andmeleti skeemile.

Andmeleti kasutamiseks on loodud rollid, millele määratud andmeleti tabelitele ja vaadetele vastavalt lugemis või kirjutamisõigusega ligipääs. Kirjutamisõigus on vaid teenusehalduri rollil.

### **6.4 Andmelaadimiste käiviti**

Andmelaadimiste käivitamiseks loodi protseduur võimaldab käivitada andmebaasis olevaid protseduure, seejuures kirjutades logitabelisse käivituse algus- ja lõpuaja, kas käivitus oli edukas. Logitabeli põhjal on võimalik analüüsida protseduuride käivitust, töötamiseks kulunud aega. Seeläbi on võimalik tuvastada esinenud vigu ning leida kohti optimeerimiseks.

```

BEGIN
  -- Initialize parameters for logging calls to sub-procedures
  SET batch_id      = COALESCE(batch_id      , HASH(cast(now() as varchar(64)) || '' ||
rand()*100000, 'SHA-256'));
  SET parent_instance_id = COALESCE(parent_instance_id, HASH(cast(now() as varchar(64)) || '' ||
rand()*100000, 'SHA-256'));
  SET parentProcedure  = COALESCE(parentProcedure  , '${schema}.dwh_load');

  -- Initiates call to a anonymous sub-procedure and logs
  call ${schema}.p_LogProcessExecution(
    '${schema}',          -- schemaName
    'p_ct_dmart_covid19_vaccination', -- procedureName
    1,                   -- procedureAcceptsParams
    batch_id,            -- batch_id
    parent_instance_id,  -- parent_instance_id
    parentProcedure      -- parentProcedure
  );

  -- call other sub-procedures
END;

```

Joonis 6. Koodinäidis andmelaadimiste käivitusprotseduuri väljakutsungist

## 6.5 Tõlketabeli lahenduse arendus

Andmestikes, kus sisendandmed on käsitsi sisestatud, esineb kõrvalekaldeid nimetuse kokkulepitud kirja pildist. Kuigi erinevad kirja pildid võivad inimloetavalt mõistetavad olla, siis andmeanalüütikas grupeerimiste ja päringute tegemiseks vigane kirja pilt ei sobi. Seetõttu tuli valmis arendada lahendus, mis võimaldaks tuvastada kõrvalekaldeid kokkulepitud kujust ning viia need väärtused õigele kujule – seda võib nimetada tõlketabeli lahenduseks.

Arendatud lahendus koosneb kahest osast – tõlketabelist ja protseduurist.

Tõlketabeli eesmärk on võimaldada andmekvaliteedi kõrvalekallete parandamist läbi lahenduse haldaja poolt loodud tõlgete.

Protseduur täidab kahte ülesannet:

- leiab lähteveerust tõlke sisendväärtuse ja määrab tõlgitud väärtuse sihtveergu

- tõlke puudumisel lisab tõlketabelisse puuduva sisendi, misjärel saab kasutaja perioodiliselt tõlkeväärtusi sisestada tõlketabelisse

Protseduur teeb seejuures tekstilähedast otsingut ning suudab lahendada ära enamlevinud väiksemad kõrvalekalded lähteväärtuste hulgas, täpsemalt:

- tõlke lähteväärtus on tõusutundetu
- ignoreeritakse liigseid tühikuid.

Tabel 3. Tõlketabeli struktuur

<b>Veeru nimetus</b>	<b>Andme- tüüp</b>	<b>Suurus</b>	<b>NOT NULL</b>	<b>Vaikeväärtus</b>	<b>Kirjeldus</b>
id	int	10	TRUE	autoincrement	Rea tunnus, primaarvõti
Schema	varchar	256	FALSE	[NULL]	Skeemi nimi, kus objekt asub
Table	varchar	256	TRUE	[NULL]	Objekti ehk tabeli nimi, kus tõlgitakse
Column	varchar	256	TRUE	[NULL]	Veeru nimi tabelis, kuhu tõlkeväärtus pannakse
TranslateFrom	varchar	512	FALSE	[NULL]	Originaalvärtus, mida tõlgitakse
TranslateTo	varchar	512	FALSE	[NULL]	Uus, parandatud värtus, milleks originaalvärtus tõlgitakse
Description	varchar	1024	FALSE	[NULL]	Vabatekstiline kirjeldus meie enda jaoks, kes, millal, miks lisas
dwh_valid	bit	1	FALSE	1	Kas kirje on kehtiv? 0 = kehtetu 1 = kehtiv
dwh_valid_from	timestamp	23	FALSE	NOW()	Millal kirje tekkis (automaatselt väärtustatud)
dwh_valid_to	timestamp	23	FALSE	[NULL]	Kuni mis ajani tõlge kehtis (käsitsi väärtustamine, kui vaja tõlge kehtetuks muuta)
dwh_created_by	varchar	256	FALSE	current user	Kes on tõlke lisanud - tavaliselt automaatselt

					laadimisteenuse teenuskonto nimi
--	--	--	--	--	-------------------------------------

Tõlketabeli põhjal tõlgete määramist võimaldab protseduur p\_TranslateColumn, mille väljakutsumise näidiskood on järgnev:

```
-- Call p_TranslateColumn to assign translations from SOURCE_COLUMN to TARGET_COLUMN
call ${schema}.p_TranslateColumn (
  '${schema}'          , -- sourceSchemaName
  'TABLE_NAME'        , -- sourceObjectName
  'SOURCE_COLUMN'     , -- sourceValueColumnName
  'TARGET_COLUMN'     , -- sourceTranslatedColumnName
  '${schema}'         , -- translateTableSchemaName
  'RULES_TRANSLATE'   , -- translateTableObjectName
  0                   , -- fallbackToOriginalValue
  1                   , -- insertMissingTranslation
  'valid'             , -- invalidateColumnName
  'ValidationErrors'  -- invalidateErrorColumnName
);
```

Joonis 7. Programmikoodi näide andmeelementide tõlkeprotseduuri välja kutsumiseks

## 6.6 Aruannete alusandmete ajalise volatiilsuse vähendamine

Tervise infosüsteemi laekuvad andmed tervishoiuteenuse osutajate infosüsteemidest. Analüüsis selgus, et SARS-CoV-2 testid võivad laekuda ajalise hilinemisega. Testide puhul on teada nende proovivõtu aeg, laborianalüüsi läbi viimise aeg ning dokumendi laekumise aeg tervise infosüsteemi. Analoogselt on ka epikriiside ja immuniseerimistestatistega.

Kui statistika põhineks sündmuse tegelikul aset leidmise ajal, siis tuleks igapäevaselt statistikasse raporteerida tagantjärele toimunud sündmusi. Segaduse vähendamiseks ning andmete äriiselt arusaadavamaks tegemiseks otsustati, et statistilise raporteerimise alus on sündmuse dokumenteerimise, mitte aset leidmise aeg. Arvestades seda, et reeglina laekub info vähese viiteajaga hiljemalt järgmise tööpäeva alguseks, fikseeriti statistilise raporteerimise jaoks kuupäev seisuga eelmise kuupäeva kella 07:00st kuni käesoleva päeva 07:00ni (v.a.).

Arvestades seda, et dokumenteerimisvigade tuvastamisel on TTO'del kohustus andmed korrastada, siis tehniliselt toimub protsess järgnevalt:

1. TTO poolt saadetakse tervise infosüsteemi dokumendi 1. versioon
2. TTO poolel tuvastatakse viga
3. TTO poolel viiakse sisse parandused, millega tekib dokumendi 2. versioon, mis saadetakse tervise infosüsteemi, sellega muudatakse tervise infosüsteemi poolel dokumendi 1. versioon kehtetuks

Võimalik on ka protsess, kus dokumendi 1. versioon tühistatakse ja 2. versiooni peale ei laeku – st. seni kehtinud dokument loogiliselt kustutati.

Kuna andmelao vaatest on dokumendi sisestamise ajaks alati selle viimase kehtiva versiooni sisestamise aeg, siis ainult dokumendi laekumise aja peale statistika üles ehitamine oleks eksitav. Statistikas samas aga ei tohi kajastada andmeid, mis värskema teadmise järgi ei ole enam kehtivad. Seetõttu leitakse alati andmetest viimane versioon, kuid selle statistikasse kajastamise kuupäev võetakse sama sündmuse esmakordse dokumenteerimise aja alusel.

Metoodika rakendamiseks loodi SQL funktsioon, mis sündmuse esmase dokumenteerimise laekumise ajast tervise infosüsteemi lahutab maha 7 tundi ning leiab kuupäeva. Seeläbi on kuupäev +7 tunni võrra nihkes.

```
CREATE OR REPLACE FUNCTION ${schema}.F_GET_STATISTICS_DATE(  
    statsTime TIMESTAMP  
    ) RETURNS DATE  
SQL SECURITY INVOKER DETERMINISTIC  
BEGIN  
    DECLARE hourShift INTEGER;  
    -- kui avaldamise aeg on kell 07:00 hommikuse seisuga siis nihe on 7t  
    SET hourShift = -7;  
    RETURN CAST(DATEADD(HOUR, hourShift, statsTime) AS DATE);  
END;
```

Joonis 8. Koodinäidis funktsioon: f\_Get\_Statistics\_Date



Sündmuse esmane dokumenteerimise aeg leitakse ODS tabelist, kus on leitavad kõik dokumendi versioonid, nende hulgast leitakse analüütikafunktsiooni MIN abil esimene.

```
-- Leiame sündmuste tabelist dokumendi esimese sisestamise aja
SELECT
  event.dwh_id,
  MIN(document.INSERT_TIME) over(
    PARTITION BY          -- Esimese aja leidmise otsinguakna definitsioon
      document.DOCUMENT_NR, -- Dokumendi number
      document.DOCUMENT_NR_OID, -- Dokumendi numbri OID
      document.PATIENT_ID, -- Patsiendi id
      document.PATIENT_ID_OID, -- Patsiendi id OID
      document.INSTIT_REG_CODE -- Dokumendi koostanud tervishoiuasutuse (TTO) reg. kood
    ) as min_INSERT_TIME
  INTO   #tmp_lookup_INSERT_TIME
  FROM   event_table as "event"
  INNER JOIN case_table as "document" ON event.DRDB_DOCUMENT_ID =
  document.DRDB_DOCUMENT_ID;

-- Uuendame andmeleti tabelit leitud ajaga
UPDATE stg_dmart_table
SET   stg.FirstInsertTime = tmp.min_INSERT_TIME
FROM   stg_dmart_table stg
INNER JOIN #tmp_lookup_INSERT_TIME tmp ON stg.dwh_id = tmp.dwh_id;
```

Joonis 9. Koodinäidis andmete esimese sisestamisaja leidmiseks

## 6.7 Testide laadimine

Testide laadimine toimub saatekirja vastuse dokumentidega (SKV) laekunud laborianalüüsi tulemuste andmestikust.

SKV dokumendil on laborianalüüsi tulemus dokumenteeritud kasutades SNOMED klassifikaatorit. Klassifikaatorist eksisteerib kasutusel paralleelselt mitu versiooni, seetõttu võib näiteks positiivset testitulemust dokumenteerida erinevalt. Järgnevalt toodud erinevad näidisväärtused:

- P
- Positiivne

- 10828004

Analüütika tarbeks tuleb erinevad tulemused viia ühtsele kujule ehk integreerida. Seda saavutatakse tõlketabeli rakendamise abil.

## 6.8 Nakatumisandmestiku kokkupanek

Nakatumisandmestik pannakse kokku ambulatoorse epikriisi dokumentidelt, mis vastavad analüüsinõuetes esitatud diagnoosikombinatsioonidele. SQL päringuks selle vormistamine on aga keerukas, kuna tuleb leida üles epikriisid, mis vastavad kahele diagnoosikoodile.

Lähtudes andmemudelist, kus epikriisi dokumendi päis on eraldi tabelis ning selle tabeli primaarvõtmele viitab epikriisi diagnooside tabel, lahendati päring optimeeritud kujul järgnevalt:

1. Diagnooside tabelist leitakse kõik unikaalsed epikriisi primaarvõtmed, kus diagnoos vastab vähemalt ühele diagnoosikoodidest, mida kombineeritakse teiste diagnoosikoodidega.
2. UNION ALL operatsiooniga liidetakse unikaalsed primaarvõtmed diagnooside tabelist, mis vastavad vähemalt ühele diagnoosikoodidest, millega kombineeritakse esimesi diagnoosikoode.
3. Ühendandmestikust leitakse kõik epikriisi primaarvõtmed, mis esinevad vähemalt 2 korda.

Seeläbi on leitud üles epikriisid, milles esineb kombineeritult kaks kombineeritavat diagnoosikoodi. Joonis 10 kirjeldab päringu realisatsiooni SQL keeles.

```

-- Looime ajutise tabeli, kuhu leiame epikriiside primaarvõtmed
CREATE TABLE #find_epicrisis (DRDB_DOCUMENT_ID bigint not null);

-- Epikriisid, millel on kaks diagnoosi (üks esimesest, teine teisest grupeeringust)
INSERT INTO #find_epicrisis (DRDB_DOCUMENT_ID)
SELECT DRDB_DOCUMENT_ID
FROM (
  -- leiame epikriisid diagnoosidega: U07.1, U07.2
  SELECT DISTINCT DRDB_DOCUMENT_ID
  FROM DWH_EPICRISIS_DIAGNOSIS_AMB
  WHERE RHK10 IN (
    'U07.1',
    'U07.2'
  )
  UNION ALL
  -- liidame epikriisid, millel esineb vähemalt üks valitud B diagnoos
  SELECT DISTINCT DRDB_DOCUMENT_ID
  FROM DWH_EPICRISIS_DIAGNOSIS_AMB
  WHERE RHK10 IN (
    'B33.8',
    'B33',
    'B34',
    'B34.2',
    'B34.8',
    'B97.2'
  )
) combination
GROUP BY DRDB_DOCUMENT_ID -- grupeerime primaarvõtme alusel
HAVING count(*) > 1 -- jätame alles epikriisid, mis esinesid mõlemas grupis
;

```

Joonis 10. Koodinäidis epikriiside leidmiseks, millel esineb vähemalt 2 konkreetset diagnoosi

### 6.8.1 Nakkusjuhtumite ajaline tekkimine

Nakkusjuhtumiks loetakse isiku nakatumist Covid-19 haigusesse, mis on tuvastatud kas ainult epikriisi dokumendilt – sel juhul on tegemist laboratoorselt kinnitamata

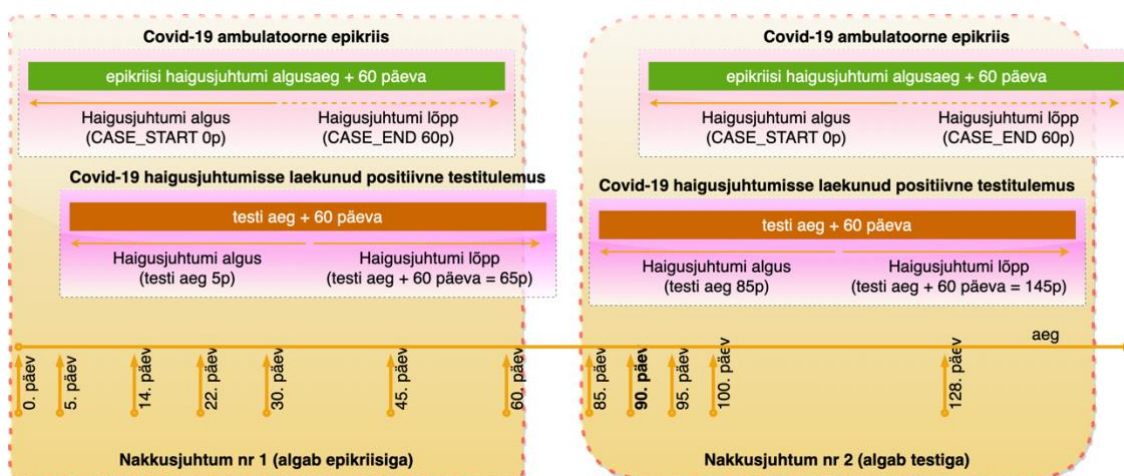
nakatumisega või SARS-CoV-2 positiivse testiga, misjuhul on tegemist laboratoorselt kinnitatud nakkusega.

Sündmused laekuvad tervise infosüsteemi aja jooksul ning seega võib esineda volatiilsus – näiteks võidakse nakatumine esmalt dokumenteerida epikriisil ning seejärel testiga, misjuhul nakatumine kinnitati laboratoorselt.

Patsient võib haigusse ka korduvalt nakatuda, seetõttu loetakse nakatumiseks vastavalt ECDC soovitudele 60 päevast ajaakent, kus nakatumise esimene sündmus loetakse nakkuse algusajaks ning sellele 60 päeva jooksul järgnevad sündmused loetakse korduvateks faktideks, mis uut nakkusjuhtumit statistilises mõttes ei loo [30].

### 6.8.2 Nakkusjuhtumite tuvastamine nakatumise andmestiku pealt

Info nakatumise kohta võib jõuda tervise infosüsteemi andmestikku kui positiivne testitulemus või kui epikriisil dokumenteeritud diagnoosina. Covid-19 nakatumise statistika vaates ei ole oluline, mitu korda on nakatumist ühe nakkusjuhtumi jooksul dokumenteeritud vaid millal on nakkusjuhtum tuvastatud.



Joonis 11. Nakatumise sündmuste grupeerimine nakkusjuhtudeks

Realiseerimiseks optimaalse lahenduse leidmine SQL koodis on võrreldav Gaps-and-Islands<sup>1</sup> probleemiga, kus ajalisse gruppi kuuluvad sündmused on defineeritud grupi esimese sündmuse raames [31].

Analüütikaandmebaasides nagu SAP IQ on FOR tsüklite või CURSOR rakendamine äärmiselt ebaefektiivne ning aeglane, seetõttu patsiendipõhine iteratsioon lahendusena oli välistatud [32].

Lahendusalgorithm:

- Luuakse tsükkel, mille maksimaalne iteratsioonide arv on maksimaalne nakkusjuhtumite võimalik arv ühel patsiendil, sõltumata nakatumissündmuste või -juhtumite koguarvust.
- Tsükli alguses esimene vaba sündmus, mis ei ole määratud nakkusjuhtumi alla.
- Sündmustest leitakse ajaliselt esimene kasutades DENSE\_RANK() analüütikafunktsiooni.
- Kõik vabad sündmused, mis ei kuulu nakkusjuhtumi alla, kuid mis ajaliselt mahuvad eelnevalt leitud esimese vaba sündmuse poolt defineeritud nakkusjuhtumi ajalisse aknasse, määratakse sellesse leitud nakkusjuhtumisse.
- Iteratsiooni korratakse kuni kõik sündmused on paigutatud nakkusjuhtumite akendesse.

Joonis 12 esitatud koodinäidis kirjeldab lahendusalgorithmi realiseerimise SQL keeles.

---

<sup>1</sup> <https://livebook.manning.com/book/sql-server-mvp-deep-dives/chapter-5/1>

```

BEGIN
-- Iteratsioon üle nakkusjuhtumite
caseAssignmentLoop:
LOOP
-- Väljume tsüklis, kui määramata sündmusi pole
IF (SELECT 1 FROM infection_event WHERE CaseDefinedById IS NULL)
IS NULL THEN LEAVE caseAssignmentLoop; END IF ;

-- Leiame patsiendi lõikes esimese vaba sündmuse
SELECT
id,          -- sündmuse primaarvõti
PATIENT_ID, -- patsiendi primaarvõti
EventStartDate, -- sündmuse alguskuupäev
EventEndDate, -- otsinguakna lõpuaeg
dense_rank() over (
partition by -- mitmes sündmus patsiendil
PATIENT_ID
order by
EventStart ASC, -- ajaliselt esimene vaba
EventType DESC, -- 'test' saab prioriteedi
id
) as CaseResultRank
INTO #tmp_FindFirstEventForCase
FROM infection_event
WHERE CaseDefinedById IS NULL; -- otsitakse määramata sündmusi

-- CaseResultRank = 1 esimene vaba sündmus määrab juhtumi
UPDATE infection_event event
SET event.CaseDefinedById = definer.id,
event.CaseStart = definer.EventStartDate,
event.CaseEnd = definer.EventEndDate
FROM infection_event as event
INNER JOIN #tmp_FindFirstEventForCase as definer
ON definer.CaseResultRank = 1
AND event.PATIENT_ID = definer.PATIENT_ID
AND event.EventStart >= definer.EventStartDate
AND event.EventStart <= definer.EventEndDate
WHERE event.CaseDefinedById IS NULL;

```

```

DROP TABLE IF EXISTS #tmp_FindFirstEventForCase;

END LOOP caseAssignmentLoop;

END;

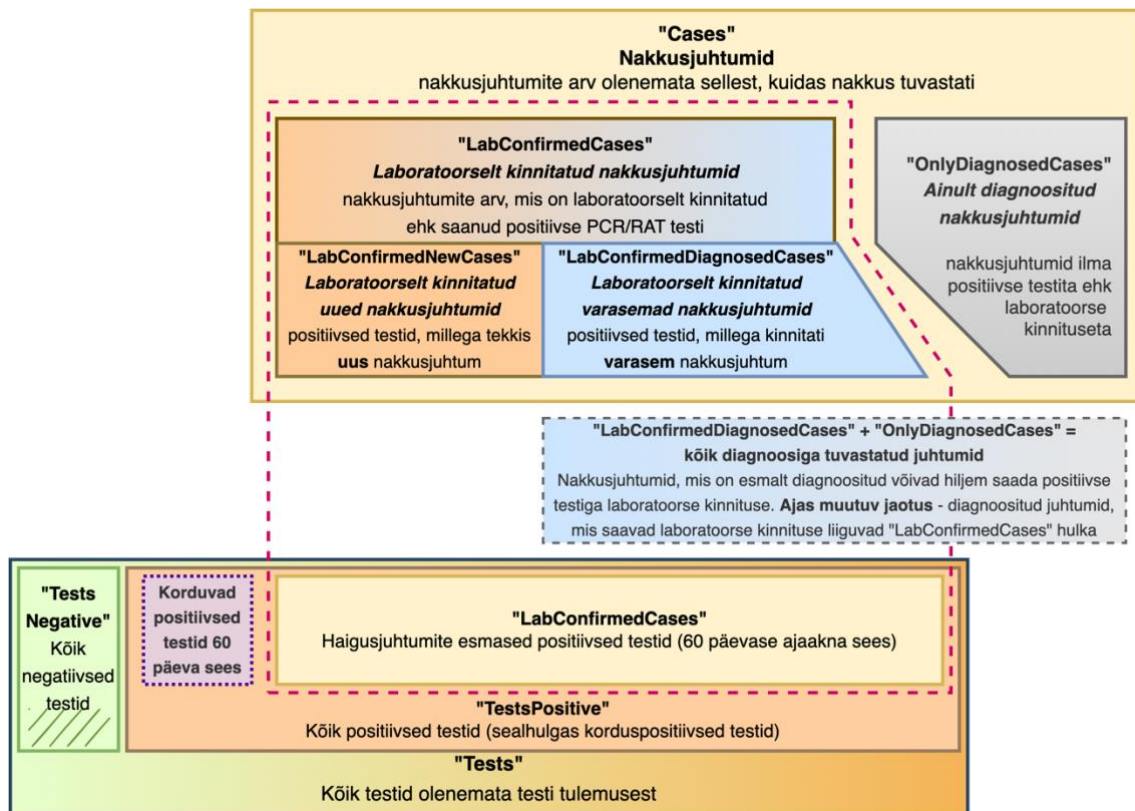
```

Joonis 12. Koodinäide nakatumissündmuste määramine nakusjuhtumitesse

### 6.8.3 Nakusjuhtumite raporteerimise meetodika

Avaandmetes ja andmeanalüütikatarkvaras tarbitavas andmestikus oli vaja luua agregeeritud alusandmestik, kus kajastuvad erinevad statistilised hulgad, mille põhjal oleks võimalik aru saada, kuidas nakatumine on tuvastatud.

Joonis 13 kirjeldab arendatud statistilise raporteerimise hulkade meetodikat, mis võimaldab ära katta kõik nakatumise statistika andmevajadused ning võimaldab teha hulgaarvutusi, et optimaalselt andmeid edastada.



Joonis 13. Nakatumise andmestiku statistilised hulgad koos kirjeldustega

## 6.9 Vaktsineerimisandmestiku kokkupanek

Covid-19 vaktsineerimise jaoks vajalikud andmed on dokumenteeritud immuniseerimise teatistel, kus vaktsiinvälditava haiguse koodiks on 219 (Covid-19) vastavalt vaktsiinvälditavate haiguste klassifikaatoritele.

Vaktsineerimise andmeid analüüsid selgus andmekvaliteedis mitmeid probleeme, nendest olulisemad olid:

1. Vaktsiini nime erinevad esinemisvormid – juhtumitel, kus vaktsiini nimi oli käsitsi kirjutatud esines tihti erinevaid kirjavigu nime ülesmärkimisel.
2. Dokumenteeritud doosi number ei olnud loogiliselt õige – täheldati juhtumeid, kus ajalises loogilises jaotuses oli patsiendile varasemalt dokumenteeritud näiteks 1. ja 2. doos, kuid nendele järgnes 1. doos. Analüüsi käigus selgus, et osad vaktsineerimist läbiviivad TTOd ei olnud tõlgendanud vaktsiini doosi numbrit õigesti.
3. Vaktsineerimise kuuri lõpetamise seis oli ebakorrektselt dokumenteeritud – kuigi andmetele tuginedes võis leida, et patsient oli Covid-19 haiguse läbipõdemise järel saanud 1 doosi, siis ärireeglite alusel tuli lugeda vaktsineerimine lõpetatuks, siis oli immuniseerimise teatisel dokumenteeritud, et vaktsineerimise kuur ei ole lõpetatud

Vaktsineerimise osas tuli kuuride läbimine vastavalt ärireeglitele ümber arvutada.

Kui kõik järgnevad kuurid pärast 1. vaktsineerimise kuuri on vastavalt reeglitele 1 doosilised, siis 1. kuuri dooside arv sõltub mitmetest aspektidest:

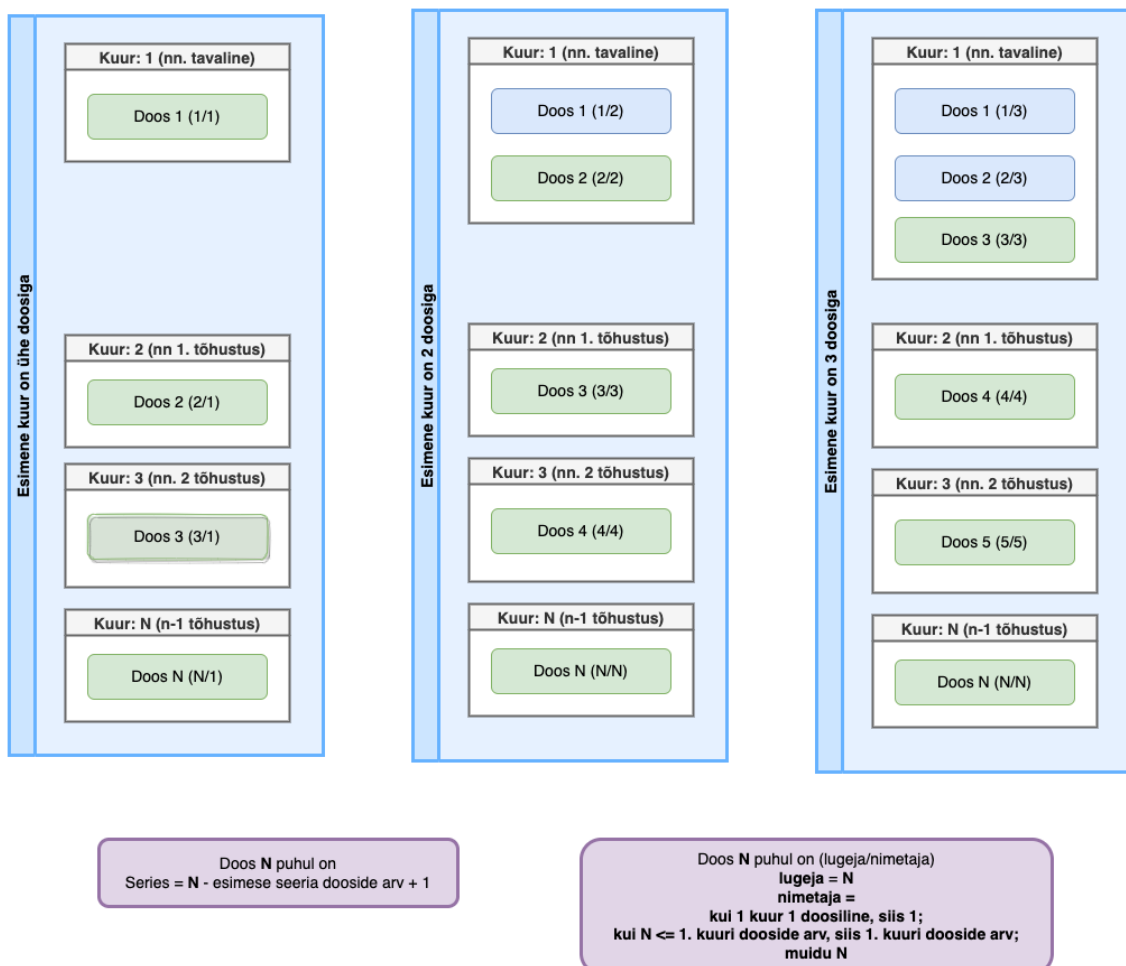
1. Kui patsienti on vaktsineeritud 1. kuuris 1 doosilise vaktsiiniga (nt. Janssen), siis on 1. kuur automaatselt 1 doosiline.
2. Kui patsient on enne 1. doosi manustamist Covid-19 haiguse läbi põdenud või nakatunud Covid-19 haigusse 14 päeva pärast 1. doosi manustamist aga enne 2. doosi manustamist, siis on 1. kuur 1 doosiline, kui muidu oleks pidanud olema 2 doosiline.



3. Vaksineerimise 1 kuur on 2 doosiline kõigil patsientidel, kes ei ole olnud nakatunud ja keda on vaksineeritud 2 doosilise vaktsiiniga.
4. Vaksineerimise 1 kuur võib 0..4 aastastel väikelastel olla 3 doosiline, kui neid on vaksineeritud 3 doosilise vaktsiiniga.

Arendatud lahendus taandas äriloogika matemaatilisele või arvutuslikule loogikale, mis võimaldas ülesannet tehniliselt optimeeritult lahendada. Arvutuslikku lahendust kirjeldab Joonis 14.

Vaksineerimise kuuride arvutus võeti hiljem ka Covid-19 vaksineerimise tõendi väljastamisel arvesse. Vaksineerimise tõendi lugeja ja nimetaja arvutus on esitatud Joonis 14 diagrammil.



Joonis 14. Vaksineerimise dooside jaotus kuurideks

## 6.10 Andmestike ajaloostamine ja võrdlusalgoritm

Andmelaadimise tulemusena on tekkinud viimase kehtiva seisuga andmestik, mis on laetud STG (*staging table*) tabelisse. Aruandluse ja andmete hilisema liigutamise jaoks on vajalik, et püsiandmestik oleks versioneeritud – st. oleks teada, milline oli kehtiv seis minevikus. Selleks kasutatakse ajaloostatud andmestiku tabelit.

Näidislahenduse kirjeldamiseks kasutame andmebaasi ajaloostatud tabelit „DMART“, kuhu laetakse andmeid ajutisest „STG“ tabelist. Tabel 4 kirjeldab „DMART“ näidistabeli struktuuri.

Tabel 4. Ajaloostatud laadimise näidistabeli "DMART" struktuur

Veerg	Andmetüüp	Kirjeldus
dwh_id	identity	Automaatselt kasvav andmelao tabeli primaarvõti
dwh_valid_from	timestamp	Rea kehtivuse algusaeg
dwh_valid_to	timestamp	Rea kehtivuse lõppaeg, kui tühi on hetkel ääriselt kehtiv
pid	bigint	Isiku primaarvõti
weight	integer	Isiku kaal

Andmelaadimisprotsessis arvutatakse „STG“ tabelisse välja viimane kehtiv seis kõigist andmetest (Tabel 6). Seda tuleb võrrelda „DMART“ tabeli lähteseisuga (Tabel 5).

Tabel 5. Näidistabeli "DMART" andmete lähteseis

dwh_id	dwh_valid_from	dwh_valid_to	pid	weight	Kirjeldus
1	2023-01-01 00:00	2023-01-01 15:00	1	150	Ei ole enam kehtiv.
2	2023-01-01 00:00		2	80	Hetkel kehtiv
3	2023-01-01 00:00		3	69	Hetkel kehtiv
4	2023-01-01 00:00		4	70	Hetkel kehtiv

Tabel 6. Nädistabeli "STG" viimase välja arvutatud seisu nädisandmed

pid	weight	Kirjeldus
2	80	Ei ole muutunud pärast viimast laadimist
3	65	On muutunud pärast viimast laadimist
NULL	30	Uus kirje

Ajaloostatud tabeli „DMART“ uuendamiseks „STG“ lähtetabelist, tuleb:

- tuvastada, millised seni kehtinud read „DMART“ tabelis on muutunud
  - vanad kirjed tuleb muuta kehtetuks
  - uued versioonid tuleb lisada
- tuvastada, millised seni kehtinud ridu enam ei eksisteeri ehk on kustutatud – need read tuleb muuta kehtetuks
- tuvastada, millised uued read on tekkinud ja need lisada DMART tabelisse

Algoritmi tulemust kirjeldab Tabel 7.

Tabel 7. Ajaloostatud andmelaadimise "DMART" nädistabeli sisu pärast laadimist

dwh_id	dwh_valid_from	dwh_valid_to	pid	weight	Kirjeldus
1	2023-01-01 00:00	2023-01-01 15:00	1	150	Ei võrreldud, kuna pole kehtiv
2	2023-01-01 00:00		2	80	Ei ole muutunud
3	2023-01-01 00:00	2023-01-02 10:05	3	69	Muutus pärast viimast laadimist
4	2023-01-01 00:00	2023-01-02 10:05	4	70	„source“-st kadunud, järelilikult „kustutatud“
5	2023-01-02 10:05		3	65	Uus versioon lisati uue reana
6	2023-01-02 10:05		5	30	Uus kirje

Kõik andmeleti andmestikud peavad olema ajaloostatud ja andmelaadimiste käigus toimuv võrdlev andmete uuendamine peab toimuma suurte andmemahtude korral kiirelt ja optimeeritult.

Varasemalt oli kasutatud andmete võrdlemiseks JOIN meetodit, mille jõudlus suurte andmehulkade korral oli liialt madal. JOIN meetodil tabelite uuendamine toimus järgnevalt (illustreerib koodinäidis Joonis 15):

1. STG tabelile LEFT JOIN abil leitakse vasted DMART tabelist, leidmaks, mis kirjed on erinevad – seejuures on STG tabelis olemas, kuid ei ole DMART tabelis – need kirjed tuleb lisada DMART tabelisse
2. DMART tabelile LEFT JOIN abil leitakse vasted STG tabelist, tuvastades mis kirjed on erinevad – seejuures on DMART tabelis olemas, kuid ei ole STG tabelis – need kirjed tuleks DMART tabelis muuta kehtetuks
3. Seejärel UPDATE käsu abiga muudetakse DMART tabelis kehtetuks kirjed, mis enam STG tabelis ei eksisteeri
4. INSERT käsu abiga lisatakse DMART tabelisse kirjed STG tabelist, mida DMART tabelis kehtiva seisuna ei olnud

```

-- dmart tabelis muudetakse kehtetuks read, mida STG tabelis ei ole
UPDATE dmart
SET   dwh_valid_to = CURRENT_TIME
FROM   dmart
LEFT JOIN stg
      ON dmart.pid = stg.pid
      AND dmart.weight = stg.weight
-- loetelu kõigist veergudest, mida võrreldakse
WHERE  dmart.dwh_valid_to IS NULL -- võrreldakse kehtivaid kandeid
      AND stg.pid IS NULL -- kehtetuks muudetakse kirjed, millele vastet "stg" tabelis ei ole
;

-- dmart tabelis muudetakse kehtetuks read, mida STG tabelis ei ole
INSERT INTO dmart (dwh_valid_from, pid, weight)
SELECT
  CURRENT_TIME as dwh_valid_from,
  pid,
  weight
-- loetelu kõigist veergudest, mida lisatakse
FROM   stg
LEFT JOIN dmart
      ON dmart.dwh_valid_to IS NULL
      AND dmart.pid = stg.pid
      AND dmart.weight = stg.weight
WHERE  dmart.dwh_valid_from IS NULL
;

```

Joonis 15. Koodinäidis andmete võrdlevast ajaloostatud laadimisest JOIN meetodil

Sellise meetodiga lähenemine on suuremahulistes tabelites (enam kui 1mln rida, >30 veergu) SAP IQ andmebaasis aeglane – LEFT JOIN algoritmiline keerukus on ligikaudselt  $n^2$ .

Täitmaks mittefunktsionaalseid nõudeid tuli leida optimeeritum lähenemine andmestike võrdluseks. Arendustöö käigus töötati välja uus meetod, mille hinnanguline algoritmiline keerukus on  $n$ .

1. STG ja DMART tabeli read liidetakse ühte ajutisse tabelisse UNION ALL käsuga, lisatakse virtuaalsed veerud \_\_origin (väärtus 1 STG puhul, 0 DMART puhul) ja \_\_id (originaaltabeli primaarvõti).

2. Üle kõigi veergude (v.a. unikaalsed identifikaatorid) käivitatakse GROUP BY käsk
3. Grupeeritud andmestikust jäetakse käsuga HAVING COUNT(\*) = 1 alles vaid ühekordselt esinevad grupeeringud – kui grupeering esines 2 korda, siis järelikult eksisteeris sama rida nii STG kui DMART tabelites, kui aga rida esines 1 korra, siis esines ta kas STG või DMART tabelis.
4. MIN() agregeerimiskäsuga võetakse välja \_\_origin ja \_\_id veergude väärtused. Kuna eelnevalt oli grupeeringust alles jäänud ainult ühekordselt esinevad read, siis agregeerimisfunktsiooni tulemuseks on alati konkreetse rea veeru väärtus.
5. DMART tabelis loetakse kehtetuks kirjed, mis ei esinenud STG tabelis ja UPDATE käsuga muudetakse DMART tabelis need read kehtetuks.
6. INSERT käsuga lisatakse DMART tabelisse STG tabelist need read, mis ei esinenud DMART tabelis.

Lahendust illustreerib koodinäidis (Joonis 16). Kokkuvõttes võib sellise lähenemise algoritmilist keerukust hinnata lineaarseks.

```

SELECT
  MIN("__origin") as "__origin",
  MIN("__key") as "__key"
INTO #tmp_comparision
FROM (
  SELECT -- Hetkel kehtiv seis "dmart" tabelist
    cast(0 as bit) as "__origin", -- sihtmärktabel
    ROWID("target") as "__key", -- rea "primaarvõti"
    pid, weight -- loetelu kõigist veergudest, mida võrreldakse
  FROM dmart as "target"
  WHERE dwh_valid_to IS NULL -- seni kehtivad kirjed
  UNION ALL
  SELECT -- Hetkel kehtiv seis "stg" tabelist
    cast(1 as bit) as "__origin", -- alliktabel
    ROWID("source") as "__key", -- rea "primaarvõti"
    pid, weight -- loetelu kõigist veergudest, mida võrreldakse
  FROM stg as "source"
) target_vs_source
GROUP BY
  pid, weight -- loetelu kõigist veergudest, mida võrreldakse
HAVING
  count(*) = 1 -- leiame read, mis eksisteerivad ainult ühes tabelitest
;

-- Määrame "DMART" tabelis kehtetuks kirjed, mida "STG" tabelis ei ole
UPDATE dmart "target"
SET "target".dwh_valid_to = CURRENT_TIME
FROM dmart as "target"
INNER JOIN #tmp_comparision as "target_vs_source"
  ON "target_vs_source"."__origin" = 0 -- ainult "DMART"
  AND "target_vs_source"."__key" = ROWID("target")
WHERE "target".dwh_valid_to IS NULL
;

-- Lisame "DMART" tabelisse uued ja muutunud kirjed
INSERT INTO dmart (dwh_valid_from, pid, weight)
SELECT
  CURRENT_TIME as dwh_valid_from,
  pid, weight -- loetelu kõigist veergudest, mida lisatakse

```

```
FROM stg as "source"  
-- lisame vaid need read, mis eksisteerisid ainult "STG" tabelis  
INNER JOIN #tmp_comparision as "target_vs_source"  
  ON "target_vs_source"."__origin" = 1 -- ainult "STG"  
  AND "target_vs_source"."__key" = ROWID("source")  
;
```

Joonis 16. Koodinäidis andmete võrdlevast ajaloostatud laadimisest UNION ALL meetodil

Arvestades kirjeldatud meetodi koodimahukust, siis meetodi implementeerimise lihtsustamiseks sai loodud protseduur `p_IncrementalLoadFromLatest`, mis võimaldab parametrizeeritult anda ette allika ja sihtmärktabeli nimed, loetelu veergudest, mida võrreldakse. Protseduuri käivitades genereeritakse dünaamiliselt SQL, mis viib andmete võrdleva ajaloostamise täide. Protseduuri väljakutsumise näidist illustreerib Joonis 17.



```

call ${schema}.p_IncrementalLoadFromLatest (
--* source of latest data - schema name
    '${schema}',
--* source of latest data - object (table or view) name
    'stg',
-- source of latest data - column name for unique key (or NULL)
    "",
--* target of versioned data - schema name
    '${schema}',
--* target of versioned data - object (table or view) name
    'dmart',
-- target of versioned data - column name for unique/primary key (or NULL)
    'dwh_id',
-- name of timestamp field in target table which marks since when a row is valid from
    null,
--* name of timestamp field in target table which marks since when a row is valid from
    'dwh_valid_from',
--* name of timestamp field in target table which marks until when a row is valid to
    'dwh_valid_to',
-- exclude these columns from comparision (separate by comma)
    NULL,
-- limit comparision only to these columns (separate by comma)
    NULL,
-- invalidate rows from TARGET that do not exist in SOURCE
    1,
-- insert rows to TARGET that only exist in SOURCE
    1,
-- timestamp to assign to expiring old entries and new valid entries
    NULL
);

```

Joonis 17. Koodinäidis protseduuri p\_IncrementalLoadFromLatest käivitamiseks

## 6.11 Aruannete arendus

Andmeleti tabeli pealt on võimalik andmeanalüütikutel luua Tableau aruandlustarkvaras juba andmeanalüüsi hõlbustamiseks visualiseeritud aruandeid. Samas mitmete aegrea graafikute jaoks on parem kasutada juba andmebaasi pool välja arvutatud aruandeid.

Andmelaos on agregeeritud statistilised realiseeritud täielike aegridadena, see tähendab, et iga dimensiooni kombinatsiooni kohta esineb andmestikus rida, kus on toodud nende dimensioonide lõikes välja päevane sündmuste arv (DailyCount) ning jooksev sündmuste summa (TotalCount).

Dimensioonide kombinatsiooni puhul on matemaatiliselt tegemist maatrikskorrutisega, dimensioonide hulka võib kuuluda:

- Statistika kuupäev (*StatisticsDate*)
- Asukohapõhisel aruandel maakond (*County*), haldusüksus (*Commune*), asutusüksus (*Village*)
- Statistiline mõõdik (*MeasurementType*)

## 6.12 Avaandmete lahenduse arendus

Andmelao poolel ette valmistatud aruanded, mis on olemuslikult statistilised agregeeritud aruanded, on valmis avaandmetena publitseerimiseks.

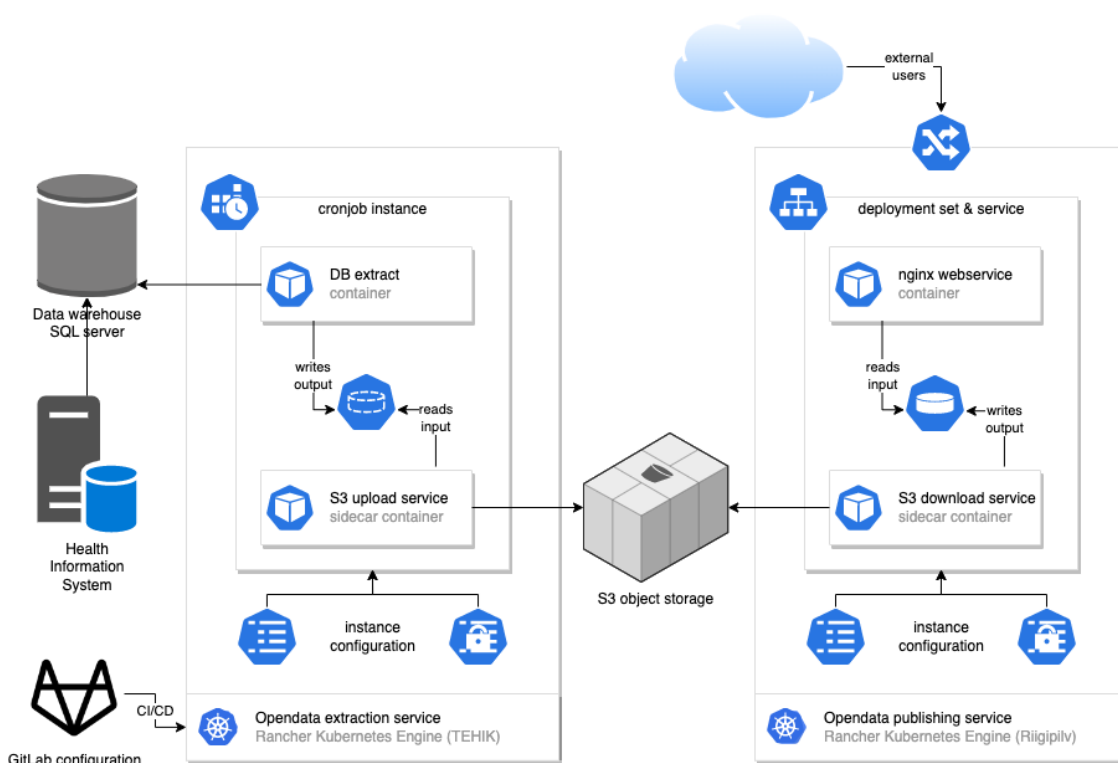
Avaandmete tarbijad soovisid andmeid nii CSV formaadis, et mugavalt neid Microsoft Excel jm. sarnase tarkvaraga analüüsida ning ka JSON formaadis, et arendajatel oleks võimalik andmeid tarbida masinmõistetava liidese kaudu.

Avaandmete avaldamiseks sai loodud mikroteenuste arhitektuuriprintsiibil põhinev teenuste kiht, mis koosneb osadest:

- *db\_extract\_service* – võtab ühendust andmelao baasiga, teeb päringu objekti pihta ning kirjutab tulemuse CSV ja JSON failideks
- *s3\_upload\_service* – töötab *sidecar* mustriis *db\_extract\_service*'ga ning kirjutab andmed kesksesse S3 failihoidlasse
- *s3\_download\_service* – töötab avaandmete klastris riigipilves ning laeb andmed alla kesksest S3 failihoidlast, kirjutades need kettapinnale
- *nginx* – levinud veebiserver, mis muudab avaandmete failid kättesaadavaks tarbijatele; *nginx* serverib staatilist sisu (faile)

Infoturbe kaalutlustel toimib kogu opendata.digilugu.ee keskkond kui staatiline veeb. Andmed tuuakse väliste teenustega sisse ning isegi, kui ründe käigus peaks keskkonna üle kontrolli saavutama, siis sealt TEHIK’u keskkonda tagasi liikumine ei ole võimalik.

Teenused on orkestreeritud Kubernetes platvormil, kahes eraldiseisvas klastris. TEHIK’u sisemises klastris toimub andmebaasiga ühenduse loomine ning andmefailide kirjutamine kesksesse S3 hoidlasse. Riigipilve Kubernetes klastris toimivad mikroteenused sünkroniseerivad andmed staatiliste failide kujul avalikku pilve, kust kaudu nendele ka ligi pääseb. Arendatud lahenduse arhitektuuri kujutab Joonis 18.



Joonis 18. Avaandmete avaldamise lahenduse arhitektuurijoonis

Kogu lahendus on täielikult automatiseeritud ning töötab Kubernetes *cron-job* teenusena ajastatult. See tähendab, et andmete tekkimisel uuendatakse neid regulaarse graafiku alusel ka avaandmete keskkonnas.

Avaandmed on kättesaadavad TEHIKu avaandmete keskkonnas aadressil: <https://opendata.digilugu.ee/>

Keskkonnas on dokumenteeritud andmestikud, kasutusjuhend<sup>1</sup>, nende kokkupanemise metoodika<sup>2</sup> ning viited failidele, mida on võimalik kasutajal kas alla laadida või kasutada andmeallikana oma analüüsitarkvaras (nt. Tableau, MS Excel vm).

### **6.13 Arendatud taaskasutatavate komponentide lähtekood**

Taaskasutatavate komponentide lähtekood on avalikustatud Koodivaramus ning on kasutatav SAP IQ andmebaasi tehnoloogial ning mikroteenused vastavalt Kubernetes platvormil orkestreeritavate konteineritena. Viited lähtekoodile on esitatud Lisas 2.

---

<sup>1</sup> <https://opendata.digilugu.ee/docs/#/et/howto> avaandmete kasutusjuhend

<sup>2</sup> <https://opendata.digilugu.ee/docs/#/et/methodology> avaandmete üldine metoodika

## 7 Kokkuvõte

Lõputöö eesmärk oli tervise infosüsteemi andmelao põhjal arendada välja lahendused, mis annaks Covid-19 tervisekriisi juhtimiseks vajaliku info ning võimaldaks avalikkusele usaldusväärseid andmeid pakkuda.

Vajaduste analüüs ning nõuete kaardistus andis ülevaate Covid-19 andmestike arendamiseks vajalikest nõuetest, mida arendusprotsessi käigus täiendati tingituna vajadusest lahendada andmekvaliteedi probleeme.

Töö skoop hõlmas Covid-19 andmeleti arendust tervise infosüsteemi andmelaoos koos toetavate protseduuridega andmete rikastamiseks, puhastamiseks, ärireeglite rakendamiseks, ajaloostatud laadimiseks ning automatiseeritud lahendust avaandmete publitseerimiseks.

Arendused viidi täide pikema ajaperioodi vältel alates 2020 aasta märtsist kuni 2022 aasta lõpuni. Suurem osa arendustöödest tuli siiski teostada lühikeses ajaraamis, iteratiivset agiilset arendusprotsessi rakendades, mis on lõputöös kirjeldatud.

Projekti võib lugeda õnnestunuks, kuna välja arendatud lahendused on olnud kasutuses ning on pälvinud positiivset tagasisidet nii klientidelt kui avalikkuselt. Välja töötatud arendusmustreid ja lahendusi laiendatakse tulevikus teiste terviseandmete andmete arendamiseks, mis võimaldab oluliselt hoida kokku aega varasemalt tehtud käsitöö arvelt ning parandada andmete usaldusväärset ja kättesaadavust.

## Kasutatud kirjandus

- [1] Terviseamet, "Eestis diagnoositi esimene COVID-19 haigusjuhtum," 27 02 2020. [Online]. Available: <https://www.terviseamet.ee/et/uudised/eestis-diagnoositi-esimene-covid-19-haigusjuhtum>. [Accessed 01 05 2023].
- [2] WHO, "WHO Timeline - COVID-19," 27 04 2020. [Online]. Available: <https://www.who.int/news/item/27-04-2020-who-timeline---covid-19>. [Accessed 01 05 2023].
- [3] Riigi Teataja, "Tervishoiuteenuste korraldamise seadus (lühend - TTKS)," 01 04 2023. [Online]. Available: <https://www.riigiteataja.ee/akt/117052020013>. [Accessed 01 05 2023].
- [4] Riigi Teataja, "Tervise infosüsteemi põhimäärus," 05 04 2023. [Online]. Available: <https://www.riigiteataja.ee/akt/105042023019>. [Accessed 01 05 2023].
- [5] Riigi Teataja, "Tervise infosüsteemi põhimäärus," 01 09 2008. [Online]. Available: <https://www.riigiteataja.ee/akt/13015769>. [Accessed 01 05 2023].
- [6] Tervise ja Heaolu Infosüsteemide Keskus, "Tervise infosüsteemi kesksüsteemi spetsifikatsioon," 07 11 2020. [Online]. Available: <https://www.riha.ee/api/v1/systems/digilugu/files/44ed0500-2e2f-4ed6-85d6-38cb772f6996>. [Accessed 01 05 2023].
- [7] Riigi Teataja, "Tervise infosüsteemi edastatavate dokumentide andmekoosseisud ning nende esitamise tingimused ja kord," 07 02 2022. [Online]. Available: <https://www.riigiteataja.ee/akt/104022022007>. [Accessed 01 05 2023].
- [8] Tervise ja Heaolu Infosüsteemide Keskus, "Standardite ja klassifikaatorite publitseerimiskeskus," [Online]. Available: <https://pub.e-tervis.ee/standards2/Standards>. [Accessed 01 05 2023].
- [9] Riigi Teataja, "Nakkushaiguste registri põhimäärus," 11 02 2022. [Online]. Available: <https://www.riigiteataja.ee/akt/113032019242>. [Accessed 01 05 2023].
- [10] Riigi Teataja, "Terviseameti põhimäärus," 04 07 2022. [Online]. Available: <https://www.riigiteataja.ee/akt/101072022038>. [Accessed 01 05 2023].
- [11] Tervise ja Heaolu Infosüsteemide Keskus, "Nakkushaiguste Register," 03 08 2018. [Online]. Available: <https://www.riha.ee/Infos%C3%BCsteemid/Vaata/nhr>. [Accessed 01 05 2023].
- [12] Majandus- ja Kommunikatsiooniministeerium, Eesti avaandmete teabevärava üldjuhend , [Online]. Available: <https://avaandmed.eesti.ee/instructions/teabevarava-uldjuhend>. [Accessed 01 05 2023].
- [13] ECDC, "Coronavirus disease 2019 (COVID-19) in the EU/EEA and the UK – eighth update," 14 03 2020. [Online]. Available: <https://www.ecdc.europa.eu/sites/default/files/documents/covid-19-rapid-risk-assessment-coronavirus-disease-2019-eighth-update-8-april-2020.pdf>. [Accessed 01 05 2023].
- [14] WHO, "Director-General's opening remarks at the media briefing on COVID-19," 11 03 2020. [Online]. Available: <https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>. [Accessed 01 05 2023].

- [15] Terviseamet, "Laboratoorsed testid COVID-19 haiguse diagnoosimisel ja SARS-CoV2 tekitajaga kokkupuute hindamisel," 02 02 2021. [Online]. Available: [https://www.terviseamet.ee/sites/default/files/Nakkushaigused/Juhendid/COVID-19/laboratoorsed\\_testid\\_covid-19\\_diagnoosimisel\\_-\\_luhiinfo\\_02.02.21.pdf](https://www.terviseamet.ee/sites/default/files/Nakkushaigused/Juhendid/COVID-19/laboratoorsed_testid_covid-19_diagnoosimisel_-_luhiinfo_02.02.21.pdf). [Accessed 01 05 2023].
- [16] Terviseamet, "Testimine koroonaviiruse suhtes," [Online]. Available: <https://www.terviseamet.ee/et/uuskoroonaviirus/testimine>. [Accessed 01 05 2023].
- [17] Riigi Teataja, "Tervise infosüsteemi edastatavate dokumentide andmekoosseisud ning nende esitamise tingimused ja kord - Lisa 4 Saatekirja vastuse andmekoosseis," 17 08 2021. [Online]. Available: [https://www.riigiteataja.ee/aktiivisa/1040/2202/2007/Lisa\\_4.pdf](https://www.riigiteataja.ee/aktiivisa/1040/2202/2007/Lisa_4.pdf). [Accessed 01 05 2023].
- [18] A. S. Loora-Elisabet Lomp, "Koroonakriis sundis testitulemused liikuma reaalselt," Postimees, 20 04 2020. [Online]. Available: <https://leht.postimees.ee/6941742/koroonakriis-sundis-testitulemused-liikuma-reaalajas>. [Accessed 01 05 2023].
- [19] O. K. H. R. S. V. T. V. Martin Laine, "Täielik korralagedus koroonaepeemia andmetega: e-riik Eestis tõrjutakse viirust Exceli tabelite ja iganenud süsteemide abil," Eesti Ekspress, 29 04 2020. [Online]. Available: <https://ekspress.delfi.ee/artikkel/89697991/taielik-korralagedus-koroonaepeemia-andmetega-e-riik-eestis-torjutakse-viirust-exceli-tabelite-ja-iganenud-susteemide-abil>. [Accessed 01 05 2023].
- [20] O. Kund, "Uskumatu, aga tõsi: „Me ei tea, mitut inimest on kokku testitud“ (intervjuu Krista Fischer'iga)," Eesti Ekspress, 29 04 2020. [Online]. Available: <https://ekspress.delfi.ee/artikkel/89691671/uskumatu-aga-tosi-me-ei-tea-mitut-inimest-on-kokku-testitud>. [Accessed 01 05 2023].
- [21] Terviseamet, "Juhis tervishoiuteenuse osutajale esmatasandil (COVID-19)," 31 10 2022. [Online]. Available: [https://www.terviseamet.ee/sites/default/files/Nakkushaigused/Juhendid/COVID-19/juhis\\_tervishoiuteenuse\\_osutajale\\_esmatasandil\\_covid-19\\_alates\\_31.10.2022.pdf](https://www.terviseamet.ee/sites/default/files/Nakkushaigused/Juhendid/COVID-19/juhis_tervishoiuteenuse_osutajale_esmatasandil_covid-19_alates_31.10.2022.pdf). [Accessed 01 05 2023].
- [22] Sotsiaalministeerium, "Eestis algab COVID-19 vaksineerimine," 27 12 2020. [Online]. Available: <https://www.sm.ee/uudised/eestis-algab-covid-19-vaksineerimine>. [Accessed 01 05 2023].
- [23] Riigi Teataja, "Tervise infosüsteemi edastatavate dokumentide andmekoosseisud ning nende esitamise tingimused ja kord - Lisa 10 Immuniseerimise teatise andmekoosseis," 07 02 2022. [Online]. Available: [https://www.riigiteataja.ee/aktiivisa/1040/2202/2007/SOM\\_m53\\_lisa10.pdf](https://www.riigiteataja.ee/aktiivisa/1040/2202/2007/SOM_m53_lisa10.pdf). [Accessed 01 05 2023].
- [24] Sotsiaalministeerium, "Tervise ja Heaolu Infosüsteemide Keskuse põhimääruse kinnitamine," 28 08 2019. [Online]. Available: [https://www.tehik.ee/sites/default/files/2020-12/Tervise\\_ja\\_Heaolu\\_Infosusteemide\\_Keskuse\\_pohimaaerus.pdf](https://www.tehik.ee/sites/default/files/2020-12/Tervise_ja_Heaolu_Infosusteemide_Keskuse_pohimaaerus.pdf). [Accessed 01 05 2023].
- [25] H. Atwal, Practical DataOps, Springer, 2020.
- [26] J. Ereth, "DataOps - Towards a Definition," University of Stuttgart, Stuttgart, Germany, 09 2018. [Online]. Available: <https://ceur-ws.org/Vol-2191/paper13.pdf>.

- [27] Tervise ja Heaolu Infosüsteemide Keskus, "Mittefunktsionaalsed nõuded," 05 2021. [Online]. Available: <https://www.tehik.ee/sites/default/files/2021-05/Mittefunktsionaalsed%20n%C3%B5uded.pdf>. [Accessed 01 05 2023].
- [28] Tervise ja Heaolu Infosüsteemide Keskus, "IT Profiil," 04 2021. [Online]. Available: <https://www.tehik.ee/sites/default/files/2021-04/IT-Profiil%20viimane.pdf>. [Accessed 01 05 2023].
- [29] G. L. S. Seung Kyoon Shin, "Denormalization strategies for data retrieval from data warehouses," *Decision Support Systems*, vol. 42, no. 1, pp. 267-282, 2006.
- [30] European Centre for Disease Prevention and Control , "Surveillance definitions for COVID-19," 15 03 2021. [Online]. Available: <https://www.ecdc.europa.eu/en/covid-19/surveillance/surveillance-definitions>.
- [31] I. Ben-Gan, *SQL Server MVP Deep Dives*, Manning Publications Co., 2009.
- [32] H. Guenal, "Performance: Why you should avoid using cursors!," SAP, 26 02 2020. [Online]. Available: <https://blogs.sap.com/2020/02/26/performance-why-you-should-avoid-using-cursors/>. [Accessed 01 05 2023].



# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Kristjan Kolde

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Eesti Covid-19 statistika ja avaandmete arendus tervise infosüsteemi andmelao põhjal", mille juhendaja on Priit Rospel
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15. mai 2023

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## **Lisa 2 – Lähtekood**

Avalikustatav lähtekood asub Eesti riiklikus Koodivaramus “TEHIK” grupi Andmeladude avalike projektide all ning on ligipääsetav siit:

### **SAP IQ andmelaadimisprotseduurid**

[https://koodivaramu.eesti.ee/tehik/dwh/common/sap\\_iq/](https://koodivaramu.eesti.ee/tehik/dwh/common/sap_iq/)

### **SAP IQ andmebaasist andmete CSV ja JSON failidesse kirjutamise mikroteenus**

<https://koodivaramu.eesti.ee/tehik/dwh/common/kubernetes/export-sapiq-data>

### **SAP IQ andmebaasist andmete CSV ja JSON failidesse kirjutamise mikroteenus**

<https://koodivaramu.eesti.ee/tehik/dwh/common/kubernetes/minio-sync-files-to-s3>

### **Avaandmete publitseerimise mikroteenus – S3 failihoidlast failide allalaadija**

<https://koodivaramu.eesti.ee/tehik/dwh/common/kubernetes/minio-sync-files-from-s3>