

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Anne Siitan 164777IABB

**E-POE ANDMETE MIGREERIMINE POSTGRESQL
ANDMEBAASIST MICROSOFT SQL SERVERI ANDMEBAASI**

Bakalaureusetöö

Juhendaja: Erki Eessaar
PhD

Tallinn 2022

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: Anne Siitan

18.05.2022

Annotatsioon

Käesolev töö on osa tegeliku e-poe üleminekust uue tarkvara kasutamisele. Käesoleva bakalaureusetöö peamiseks eesmärgiks on migreerida, ehk kanda vajadusel teisendusi tehes üle, e-poe andmed PostgreSQL andmebaasist Microsoft (MS) SQL Serveri andmebaasi. Mõlemad andmebaasisüsteemid on SQL-andmebaasisüsteemid. Lähte ja sihtandmebaasi kontseptuaalne skeem on erinevad, mis muudab andmete ülekandmise keerukamaks.

Ülesande täitmiseks on vaja luua lähte- ja sihtandmebaasi füüsilise disaini seni puudunud kirjeldus nende tabelite osas, mida ülekandeks kasutatakse. See annab omakorda hea võimaluse leida ja analüüsida sihtandmebaasi tabelite disaini probleeme. Sihtandmebaasi tabelite dokumenteerimine ja nende disaini kvaliteedi analüüsimine on selle töö teised eesmärgid.

Lõputöö tulemusena valmisid andmete ülekandmiseks koostatud SQL laused. Andmeid kanti sihtandmebaasi nelja tabelisse. Kokku on nendes tabelites 269 veergu. Töös esitati nende nelja tabeli füüsilise disaini kirjeldus joonistena ning tabelite ja veergude sõnaliste kirjeldustena. Andmed võeti lähteandmebaasi üheksast tabelist. Töös esitati ka lähteandmebaasist kasutatud tabelite disaini kirjeldus joonisena ning nende tabelite veergude sõnalised kirjeldused. Varem selline dokumentatsioon lähte- ja sihtandmebaasi kohta ettevõttel puudus. Töös analüüsiti sihtandmebaasi nelja tabeli disaini kvaliteeti, toodi välja probleeme ja lisati autoripoolseid parandusettepanekuid. SQL-lausetes skriptfailidega saab tutvuda GitLabis aadressil: <https://gitlab.com/loputoo/Kolimine>

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 86 leheküljel, 6 peatükki, 22 joonist, 19 tabelit.

Abstract

Migration of the Data of an E-Store from a PostgreSQL Database to a Microsoft SQL Server Database

The current work is a part of an actual e-store migration process to a new platform. The main goal of this bachelor's thesis is to migrate data of the e-store from a PostgreSQL database to a MS SQL database. To achieve the goal, the author had to create the missing documentation of the physical design of the source and target databases in case of the tables that are involved in the migration. This in turn gave the opportunity to find and analyze the design flaws of the target database tables. Documenting the target database table design and analyzing the design quality of the target database tables are the secondary goals of the thesis.

Both database management systems are SQL-based. The conceptual schemas (schemas) of the source and target database, which describe the structure of base tables, are different and thus the data transfer is complex. The schemas are different because these were designed for different applications. The source database tables were created based on Django Oscar framework and slight modifications were made by the development team. The target database was designed by the development team of the company. Therefore, it was impossible to make one-to-one data transfer to the target database. Thus, there was the need to understand the structure of the databases and then convert the data during migration to make it compatible with the schema of the target database. Only a small number of tables from the source and target databases were used. The data was extracted from nine tables and had to be loaded to four tables.

The author chose a suitable approach for the data migration – the Big Bang data migration by writing and executing scripts. The decision was made by the author based on the requirements that were given by the company.

The requirements for the migration tool were put together by the author. The author compared different tools and chose the one that met all the requirements – Microsoft SQL Server Management Studio. This program can connect both to the PostgreSQL and MS

SQL Server databases. SQL statements can be executed there to load data from one database to another.

After that the author made the first mapping of the columns of the source and target databases in an Excel sheet. The mapping was created in collaboration with the company's software developer, because during the analysis it became clear that there was no documentation about the databases and the only hints were comments in the application source code.

SQL-sentences for the data transfer from a PostgreSQL database to a MS SQL database were compiled based on the compiled mapping. For each relevant table in the target database there is a SQL script file with *INSERT* and *UPDATE* statements. There are also two additional files with the statements that support the migration process. One of these must be executed before the migration. It changes data types of two columns and adds a new column to a source database table. The other script file must be executed after the migration. It reverses all the changes that were made in the source database with the execution of the statements in the first file.

The author analyzed the quality of the design of the target database tables, pointed out the problems, and made some recommendations for improvements. The author found 20 different types of problems in the design of tables.

The author created six SQL script files with in total 49 SQL-statements: four of these were *INSERT*, 26 of these were *UPDATE*, 15 of these were *ALTER*, and four of these were *SET*. SQL script files can be viewed at GitLab: <https://gitlab.com/loputoo/Kolimine>

The total data transfer based on test databases took about 19 minutes and 45 seconds. The scripts were also tested in the company's software developer's computer, with test databases and with a test application. The accuracy of the migrated data was checked. The actual migration by using the created scripts will take place after the completion of the thesis.

The thesis is in Estonian language and contains 86 pages of text, 6 chapters, 22 figures, 19 tables.

Lühendite ja mõistete sõnastik

.NET	Microsoft'i tarkvaraarenduse platvorm.
GDPR	<i>General Data Protection Regulation</i> , isikuandmete kaitse üldmäärus.
GUID	<i>Global Unique Identifier</i> , globaalne unikaalne identifikaator.
IBAN	<i>International Bank Account Number</i> , rahvusvaheline pangakonto number.
JSON	<i>JavaScript Object Notation</i> , andmete tekstina esitamise ja vahetamise lihtsustatud vorming.
LHV	Konkreetne finantsteenuste osutaja.
Kontseptuaalne skeem	Tuntakse ka nime all loogiline skeem. ANSI-SPARC andmebaaside kihilise arhitektuuri [1] kohaselt skeem, kus kirjeldatakse ühend kõigi andmebaasist huvitatud osapoolte vajadustest andmebaasis olevatele andmetele. SQL-andmebaasis moodustavad selle skeemi baastabelid ehk tabelid.
Lähteandmebaas	Tegelike andmetega PostgreSQL andmebaas, mida kasutab Leedu e-poe vana tarkvara ja kust võetakse ülekantavad andmed.
MS SQL Server	<i>Microsoft SQL Server</i> , Microsofti välja töötatud SQL-aluseks olevad andmemudelil põhinev andmebaasisüsteem ehk andmebaasihaldur. [2]
MS SQL Serveri testandmebaas	Autori arvutis loodud MS SQL Serveri andmebaas, mille põhjal ta kirjutas andmete ülekandmise lauseid. Selles oli samasugune tabelite struktuur nagu tööbaasis, kuid seal olid tegelike andmete asemel testandmed.
ODBC	<i>Open Database Connectivity</i> , avatud andmebaasipöördus.
PDF	<i>Portable Document Format</i> , porditav dokumendiformaat.
PostgreSQL	Tasuta vabavaraline (avatud lähtekoogiga) SQL-aluseks olevad andmemudelil põhinev andmebaasisüsteem ehk andmebaasihaldur. [3]
PostgreSQL testandmebaas	Autori arvutis loodud PostgreSQL andmebaas, mille põhjal ta kirjutas andmete ülekandmise lauseid. Selles oli samasugune tabelite struktuur nagu lähteandmebaasis, kuid seal olid tegelike andmete asemel testandmed.
SQL	<i>Structured Query Language</i> , standardiseeritud andmebaasikeel, mis realiseerib relatsioonilist andmemudelit ja mille abil saab teha andmebaasidega seotud toiminguid nagu näiteks hallata andmeid, andmebaasiobjekte, õiguseid ja tehinguid. [4]
SSMS	<i>Microsoft SQL Server Management Studio</i> .
Tööbaas, Sihtandmebaas	Tegelike andmetega MS SQL Serveri andmebaas, mida kasutab Leedu e-poe uus tarkvara ja kuhu lõpuks kantakse autori koostatud lauseid kasutades andmed üle lähteandmebaasist.

Unicode	Rahvusvaheline standard arvutites kirjasüsteemide kodeerimiseks. [5]
UPC	<i>Universal Product Code</i> , toodete vöotkoodi sümboolika, mida kasutatakse kaubaartiklite jälgimiseks kauplustes.
UML	<i>Unified Modeling Language</i> , üldotstarbeline modelleerimiskeel.
URL	<i>Uniform Resource Locator</i> , inernetiaadress.

Sisukord

1 Sissejuhatus	12
1.1 Töö taust ja lühikirjeldus	12
1.2 Probleem ja eesmärgid.....	12
1.3 Ülevaade töö struktuurist.....	13
2 Teoreetiline taust	14
2.1 Andmesiire.....	14
2.2 Andmesirdevahendid.....	17
2.3 SQL-andmebaasi ülesehituse probleemid	18
3 Metoodika.....	20
3.1 Objekti kirjeldus	20
3.2 Tööprotsessi kirjeldus.....	21
3.3 Tööriistade kirjeldus	22
3.4 Andmesirdevahendi valimine	24
3.4.1 Kriteeriumid	25
3.4.2 Kriteeriumite hindamine ja tööriistade võrdlemine.....	25
3.5 Metoodika kokkuvõte	27
4 Töö tulemused	28
4.1 Lähteandmebaasi füüsilise disaini mudel.....	28
4.1.1 <i>Accounts_user</i> tabel.....	28
4.1.2 <i>Address_useraddress</i> tabel	31
4.1.3 <i>Social_auth_usersocialauth</i> tabel.....	33
4.1.4 <i>Order_order</i> tabel.....	33
4.1.5 <i>Order_ordernote</i> tabel.....	38
4.1.6 <i>Order_line</i> tabel.....	38
4.1.7 <i>Order_billingaddress</i> tabel.....	41
4.1.8 <i>Order_shippingaddress</i> tabel	42
4.1.9 <i>Order_paymentevent</i> tabel.....	44
4.2 Sihtandmebaasi füüsilise disaini mudel.....	44
4.2.1 <i>User</i> tabel.....	45

4.2.2 <i>Mail_campaign_participant</i> tabel	52
4.2.3 <i>Order</i> tabel.....	53
4.2.4 <i>Order_product</i> tabel	61
4.3 Andmesiire.....	65
4.3.1 Planeerimine	65
4.3.2 Andmete ettevalmistamine	66
4.3.3 Andmesiirde kavandamine	66
4.3.4 Teostus.....	68
4.3.5 Testimine	71
5 Analüüs ja järeldused.....	72
5.1 Olemasoleva andmebaasi ülesehituse probleemid ja parandamise soovitusel	72
5.1.1 Probleemid tabelite ülesehituses.....	72
5.1.2 Kitsenduste vead.....	76
5.1.3 Maksimaalsed väljapikkused.....	77
5.1.4 Nimetamise probleemid.....	80
5.1.5 Üldised.....	81
5.2 Andmesiirde käigus esile kerkinud keerukused	82
5.2.1 Olemasolevate andmebaaside dokumenteerimine.....	82
5.2.2 Testandmebaasi tabelite tühjendamine.....	82
5.2.3 <i>User</i> tabelisse andmete ülekandmine	84
5.2.4 <i>Order</i> tabelisse andmete ülekandmine	88
5.2.5 <i>Order_product</i> tabelisse andmete ülekandmine.....	89
6 Kokkuvõte	91
Kasutatud kirjandus	93
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	98
Lisa 2 – Näide <i>User</i> tabeli kohta tehtud vastavustabelist Excelis	99
Lisa 3 - SSMS kaudu andmebaasi tabelite tühjendamise juhend.....	100

Jooniste loetelu

Joonis 1 Andmesiirde ettevalmistamise üldise protsessi tegevusdiagramm.	22
Joonis 2 SSMS tarkvara seos teiste töös kasutatud programmidega.	23
Joonis 3 Lähteandmebaasis kasutatud tabelite üldine struktuur.	28
Joonis 4 Lähteandmebaasi tabel <i>Accounts_user</i> ja sellega seotud tabelid.	29
Joonis 5 Lähteandmebaasi tabel <i>Order_order</i> ja sellega seotud tabelid.	34
Joonis 6 Lähteandmebaasi tabel <i>Order_oredernote</i> ja sellega seotud tabelid.	38
Joonis 7 Sihtandmebaasi andmesiirdes kasutatud tabelite üldine struktuur.	45
Joonis 8 Sihtandmebaasi tabel <i>User</i> ja selle seos <i>Mail_campaign_participant</i> tabeliga.	46
Joonis 9 Sihtandmebaasi tabel <i>Mail_campaign_participant</i> ja sellega seotud tabel.	52
Joonis 10 Sihtandmebaasi order tabel	54
Joonis 11 Sihtandmebaasi tabel <i>Order_product</i> ja sellega seotud tabel	62
Joonis 12 <i>User</i> tabelisse sisestatud kindlad numbrilised väärtused.	67
Joonis 13 SQL-lausetega skripti failide käivitamise tegevusdiagramm.	69
Joonis 14 Sihtandmebaasis <i>Order</i> tabeli välisvõtme kitsenduste nimed.	80
Joonis 15 <i>User</i> tabeli INSERT ja SET laused.	84
Joonis 16 <i>User</i> tabelis primaarvõtme väärtuste generaatori lähtestamine.	85
Joonis 17 Kliendi kõige uuema aadressi leidmine.	86
Joonis 18 Kliendi kõige uuema aadressi pealt lauatelefonide leidmine.	87
Joonis 19 <i>Before_migration</i> failis tellimuse numbri teisendamine.	88
Joonis 20 <i>Before_migration</i> failis veeru <i>message</i> andmetüübi muutmine.	89
Joonis 21 Veeru <i>message</i> andmetüübi tagasi muutmine.	89
Joonis 22 Ridade lisamine <i>order_product</i> tabelisse.	90

Tabelite loetelu

Tabel 1 Tööriistade kriteeriumite tähtsuse hindamine.	26
Tabel 2 Tööriistade vastavus kriteeriumitele.	26
Tabel 3 Lähteandmebaasi tabeli <i>Accounts_user</i> veergude kirjeldused.	29
Tabel 4 Lähteandmebaasi tabeli <i>Address_useraddress</i> veergude kirjeldused.....	31
Tabel 5 Lähteandmebaasi tabeli <i>Social_auth_usersocialauth</i> veergude kirjeldused.	33
Tabel 6 Lähteandmebaasi tabeli <i>Order_order</i> veergude kirjeldused.	34
Tabel 7 Lähteandmebaasi tabeli <i>Order_ordernote</i> veergude kirjeldused.	38
Tabel 8 Lähteandmebaasi tabeli <i>Order_line</i> veergude kirjeldused.	39
Tabel 9 Lähteandmebaasi tabeli <i>Order_billingaddress</i> veergude kirjeldused.	41
Tabel 10 Lähteandmebaasi tabeli <i>Order_shippingaddress</i> veergude kirjeldused.....	42
Tabel 11 Lähteandmebaasi tabeli <i>Order_paymentevent</i> veergude kirjeldused.	44
Tabel 12 Sihtandmebaasi tabelite üldinfo.	45
Tabel 13 Sihtandmebaasi tabeli <i>User</i> veergude kirjeldused.....	47
Tabel 14 Sihtandmebaasi tabeli <i>Mail_campaign_participant</i> veergude kirjeldused.	53
Tabel 15 Sihtandmebaasi tabeli <i>order</i> veergude kirjeldused.....	55
Tabel 16 Sihtandmebaasi tabeli <i>Order_product</i> veergude kirjeldused.	63
Tabel 17 Andmesiirde lausete kokkuvõte.	69
Tabel 18 Samasisulised aga erinevate veerupikkustega veerud.	78
Tabel 19 Vanad veerud, mida enam ei kasutata	81

1 Sissejuhatus

Sissejuhatuses esitatakse ülevaade töö taustast, kirjeldatakse probleemi ja pannakse paika töö eesmärgid ning antakse ülevaade kirjatöö struktuurist.

1.1 Töö taust ja lühikirjeldus

Ettevõttel on e-pood erinevates riikides. Eesti ja Läti poed on ühel platvormil, mis on loodud kasutades .NET raamistikku ning MS SQL Serveri andmebaasisüsteemi. Neil on erinevad andmebaasid, aga sarnane kontseptuaalne skeem (edaspidi skeem) ehk baastabelite struktuur. Kontseptuaalne skeem kirjeldab andmebaasi baastabelid (edaspidi tabelid). Leedu e-pood on teisel platvormil. See on loodud Django raamistikule ja rakendus kasutab PostgreSQL'i andmebaasisüsteemi. Nende andmebaasis on teistest riikidest erinev skeem.

Ettevõtte soovib Leedu e-poe teistega ühisele platvormile üle kolida. Seega Leedu andmebaas tuleb üle tuua PostgreSQL'st MS SQL Serverile.

1.2 Probleem ja eesmärgid

Töö peamine eesmärk on migreerida andmed PostgreSQL andmebaasist MS SQL Serveri andmebaasi. Andmete migreerimine tähendab andmete ülekannet koos nende vajadusel teisendamisega. Selle eesmärgi täitmiseks tuleb dokumenteerida osa sihtandmebaasi tabelitest. Seda pole varem tehtud. Ühtlasi annab see võimaluse analüüsida sihtandmebaasi tabelite disainiprobleeme. Sihtandmebaasi tabelite dokumenteerimine ja nende disaini kvaliteedi analüüsimine on töö teisesed eesmärgid.

Töö keerukus seisneb selles, et nende andmebaaside skeemid on erinevad, kuna need on loodud erinevate rakenduste jaoks. Lähteandmebaas on karbitoode ja sihtandmebaas ettevõtte arendajate käsitöö. Seega ei ole võimalik tabelleid sihtandmebaasi üks-ühele üle kolida. Järelikult on andmebaasi struktuurist vaja eelnevalt aru saada ja teisendada andmed ülekandmise käigus sihtandmebaasi skeemiga sobivaks.

1.3 Ülevaade töö struktuurist

Töö on jaotatud kaheks osaks – esimene pool on teoreetiline ja teine praktiline.

Teoreetiline osa jaguneb omakorda kaheks. Esimeses osas tutvustatakse andmesiidre teoreetilist tausta. Selgitatakse, mis on andmesiire. Selgitatakse andmesiidre tüüpe ja läbiviimise võimalusi ning millistest etappidest koosneb andmesiidre ja mida igas etapis tehakse. Peatüki lõpus kirjeldatakse erinevaid andmesiidre vahendite klasse ja viidatakse allikatele, kus antakse ülevaade SQL-andmebaasi ülesehituse probleemidest.

Teoreetilise osa teise poole võtab enda alla bakalaureusetöö meetodi kirjeldus. Esmalt kirjeldatakse detailselt töö objekti. Seejärel kirjeldatakse üldiselt käesolevas töös kasutatud andmesiidre protsessi ja tööriistu ning lõpuks valitakse töö jaoks sobivaim andmesiidrevahend ning põhjendatakse valikut.

Töö teine pool praktiline. Kõigepealt esitatakse lähte- ja sihtandmebaasi füüsilise disaini mudelid. Seejärel selgitatakse detailselt läbi viidud andmesiidre protsessi.

Järgnevalt analüüsitakse sihtandmebaasi tabelite ülesehituse probleeme ja tuuakse välja ka parandamise soovitused. Lõpuks võetakse kokku keerukused, millega autor andmesiidre käigus kokku puutus ja analüüsitakse, mida tehtust õppida.

2 Teoreetiline taust

Selles peatükis selgitatakse, mis on andmesiire. Kirjeldatakse andmesiirde tüüpe ja andmesiirde läbiviimise võimalusi ning selgitatakse andmesiirde läbiviimise erinevaid etappe. Tutvustatakse erinevaid andmesiirdevahendite klasse. Samuti viidatakse allikatele, kus käsitletakse SQL andmebaaside disainprobleeme.

2.1 Andmesiire

Andmesiire on sobivate andmete valimine, ettevalmistamine, kätte saamine, teisendamine ja lõplik liigutamine õigel ajahetkel teise andmehoidlasse ning lõpuks vana andmehoidla kasutusest kõrvaldamine. [6]

On kuus enamlevinud andmesiirde tüüpi. Nende jaotus ei ole väga range ja võib juhtuda, et mõni andmete ülekanne sobib mitme liigituse alla.

Andmesiirde tüübid on järgmised.

- Andmekandja migratsioon – Andmete ülekanne toimub ühelt füüsiliselt andmekandjalt teisele või füüsiliselt andmekandjalt virtuaalsele. Selle näiteks on üleminek paberkandjalt digitaalsele dokumendile. [7]
- Andmebaasi migratsioon – Andmebaas ei ole ainult pelgalt andmete talletamise/hoidmise koht. See pakub ka kindlat struktuuri, mis organiseerib informatsiooni kindlal viisil ja seda saab kontrollida andmebaasisüsteemi abil. Enamjaolt tähendab seda tüüpi andmesiire kas andmebaasisüsteemi uuemale versioonile üleminekut või üleminekut uut marki andmebaasisüsteemile. Uus andmebaasisüsteem võib olla sama ettevõtte mõni teine andmebaasisüsteem või mõne teise ettevõtte andmebaasisüsteem. Uus andmebaasisüsteem võib olla sama või ka erineva aluseks oleva andmemudeliga kui algne andmebaasisüsteem. Sellise ülemineku näiteks on üleminek MySQL'lt PostgreSQL'le. [7] Mõlema

näol on tegemist SQL-põhise andmebaasisüsteemiga, kuid neid arendavad erinevad ettevõtted. [7]

- Rakenduse migratsioon – Rakenduse migratsioon kujutab rakenduse ja sellega seotud andmete ning hostiserverite teisaldamist ühest keskkonnast teise. Rakenduse migreerimine uude keskkonda võib kaasa tuua mitmeid raskuseid. Algse keskkonna ja sihtkeskkonna vahel võib olla erinevusi, mis võivad nõuda rakenduse funktsioonide või arhitektuuri muutmist. Uus keskkond võib kaasa tuua ka ootamatuid turbe- ja vastavusprobleeme, mis võivad nõuda rakenduste ja andmete turvalisuse tagamiseks uute tööriistade kasutuselevõtmist ja võimaluste väljatöötamist. [8]
- Andmekeskuse migratsioon – Andmekeskus on füüsiline infrastruktuur ehk taristu, mida organisatsioonid kasutavad oma tähtsate rakenduste ja andmete hoidmiseks. See hõlmab endas servereid, arvutivõrke, kommutaatoreid ja muud IT varustust. Seega andmekeskuse kolimine võib tähendada eri asju: nii arvutite ümberpaigutamist kui ka digitaalsete varade ümberpaigutamist uutesse serveritesse ja mäluseadmetesse. [7]
- Äriprotsessi migratsioon – Selline andmesiirde vajadus tekib eelkõige seoses ühinemiste ja ülevõtmistega, ettevõtte optimeerimisega või reorganiseerimisega eesmärgiga konkurentsiprobleeme lahendada või uutele turgudele siseneda. Kõik need muutused võiva kaasa tuua ka vajaduse kolida ettevõtte rakendus ja andmebaas (koos klientide, toodete ja tehingute andmetega) uude keskkonda. [7]
- Pilve migratsioon – Kujutab endast andmete ja/või neid kasutavate rakenduste või teenuste liigutamist kohalikest serveritest teenusepakkuja hoole all olevatesse serveritesse. Andmed, rakendused ja teenused, mida hoitakse organisatsiooni valduses olevas serveris, kolitakse üle pilvepõhisele taristule. [9]

Andmesiirde läbiviimise võimalused on järgmised.

- Suure pauguga andmete üleviimine (*Big bang data migration*). Andmed viiakse üle ühekorraga ja lühema aja vältel. Sobib pigem väiksema mahuga andmete üleviimiseks. [10]

Plussid: Odavam, vähem keerukas, võtab vähem aega, kõik ülekande käigus vajalikuks osutuvad muudatused andmetes tehakse üks kord ja ühekorraga [7]

Miinused: Suur risk on teha suuri vigu. Eeldab andmeid kasutavate süsteemide tööseisakut ülekande perioodil. [7]

- Tilgutamine/niriseamine (*Trickle data migration*). Andmed viiakse üle jupikaupa, sel ajal töötavad lähte ja sihtandmebaas kõrvuti. [10] Sobib paremini suuremahuliste andmete liigutamiseks, samuti rakenduste puhul, mis peavad olema kättesaadavad 24/7. [7]

Plussid: Ei põhjusta tööseisakuid, kuna teostamise aega saab sättida nii, et see segaks tööd võimalikult vähe. Samuti esineb vähem ettenägematu vigu. [7]

Miinused: Kallim, võtab rohkem aeg, vajab rohkem ressursse, et hoida kahte süsteemi samaaegselt töös. [7]

Andmesiirde etapid on järgmised.

- Planeerimine. Selles etapis pannakse paika skoop, lähenemise viis ja täpsemad vahendid, eeldatav eelarve, realistlik teostuse aeg ning läbiviimise teostajad [10].
- Andmete ettevalmistamine. Ülekantavate andmete uurimine, üleliigsete andmete välja filtreerimine ja andmekvaliteedi parandamine. Selles etapis tehakse andmetest varukoopiad, märgitakse üles teostajate rollid, vajaminevad juurdepääsuõigused. [10]
- Andmesiirde kavandamine. Tehakse selges kasutatavad andmeväljad (sh milline on nendes olevate andmete tähendus). Koostatakse skriptifailid andmebaasikeele lausetega või valitakse ja kohandatakse kolmandate osapoolte migreerimise tööriista. [10]
- Teostus. Selles etapis toimub andmesiirde teostus. Toimub sobilike andmete väljaotsimine, teisendamine ja sihtandmebaasi laadimine. [10]
- Testimine. Sihtandmebaasi testimine, andmete õigsuse kontrollimine. Rakenduse käivitamine koos sihtandmebaasi andmetega. [10]

2.2 Andmesiiddevahendid

Andmesiidre tööriistad jagunevad laias laastus kolmeks.

- 1) Tööriistad, kus andmesiidre läbiviijad saavad ise kirjutada andmete migratsiooni skripte.
- 2) Enda taristu peal käitatavad tööriistad.
- 3) Pilvepõhised tööriistad. [11]

Isekoostatud skriptidega andmete migratsioon on ettevõttesisene lahendus, mis sobib väikestele projektidele, kiireteks parandusteks, või kui spetsiifilisemad tööriistad ei toeta siht- või lähteandmete allikaid. Sellise vahendi plussid on, et saab kiirelt arendada ja kui nõuded on lihtsad või andmesiidre on ühekordne ettevõtmine, siis võib lahendus osutada odavamaks kui muud lahendused. Miinusteks on, et selline lähenemine vajab koodi kirjutamise oskust. Skriptide muutmise vajadus võib kulusid suurendada, sest kui kood pole hästi dokumenteeritud, siis võivad muudatused kujuneda keeruliseks. Miinuseks on ka see, et antud lähenemine suunab arendajad oma põhiülesannete täitmisest kõrvale. [11]

Skriptide käivitamine ja enda taristul käitatavad spetsiaalsed tööriistad sobivad kasutamiseks siis, kui kolitavad andmed on delikaatsed. Tulenevalt isikuandmete kaitse üldmäärusest (GDPR) (2. jagu, artikkel 29) [12], pole lubatud kolmandatele osapooltele isikuandmeid edastada ilma isiku nõusolekuta.

Tööriista hinnapoliitika võiks soosida selle enda taristu peal käitamist. See tööriist tasub end ära siis, kui sellega ei kaasne ettevõttele püsikulusid, soetatakse pikaajaline litsents tööriista kasutamiseks, mitte ei maksta kasutamise eest kuutasu, ega näiteks ülekantud andmete mahu pealt. See omakorda tähendab, et andmesiidret on vaja teha korduvalt pikema aja vältel, mitte üks kord. Selliste tööriistade plussideks on lahenduse kasutuselevõtuks kuluv lühike aeg (eeldusel, et vahend pole nii keeruline, et seda peab pikalt õppima). IT meeskonnal on täielik kontroll nii lahenduse füüsilise kui rakenduse kihi üle. Miinusteks on see, et IT meeskond peab vastutama turvalisuse eest ja teostama tarkvara uuendusi. [11]

Enda taristul käivitatavate tööriistade näited on SQL Server Integration Services [13], Talend [14], IRI NextForm [15] ja Snaplex [16].

Pilvepõhised andmete migratsiooni tööriistad võivad olla parem valik organisatsioonidele, mis kas plaanivad viia andmed pilvepõhisesse andmebaasi või juba kasutavad seda. Tööriist sobib neil juhtudel kus lähteandmed ja sihtandmebaas asuvad mitmes andmekeskuses, kus on vaja kiiresti, kas suurendada või vähendada oma andmemahutu vastavalt vajadusele ja kus kohaliku tööriista kasutamisega kaasneks ettevõttele püsikulu, samas kui tarkvara kui teenuse eest tasutakse kuu ja mahupõhiselt. Miinusteks on sellise lahenduse korral võimalikud turvariskid, mis kaasnevad oma andmete usaldamisega kellegi teise hoole alla. Teenusepakkujaga võivad tekkida usaldus- või suhtlusprobleemid. Teenusepakkuja hinnapoliitika on prognoosimatu. Läbi tuleb mõelda, millise riigi seadused hakkaksid nendele andmetele kehtima ja kus toimuksid kohtuvaidlused. [11]

Pilvepõhiste tööriistade näited on Matillion [17] ja AWS Data Pipeline [18].

2.3 SQL-andmebaasi ülesehituse probleemid

Muster on struktuurse kirjutamise viis, et selgitada häid ja levinud lahendusi probleemidele. Antimuster kirjeldab probleemi lahendust, mis esmapilgul paistab hea, kuid mille kasutamine tekitab hiljem veel rohkem probleeme. Sellise kirjelduse osaks on ka algsest lahendusest parema probleemilahenduse kirjeldus. Bill Karwin seletab oma raamatus „SQL Antipatterns“ lahti levinumad vead, mida SQL-andmebaasi ja neid kasutavaid andmebaasirakendusi luues tehakse. Ta kutsub neid andmebaasi antimustriteks. Oma raamatus on ta jaganud need nelja kategooriasse: andmebaasi loogilise disaini antimustrid, andmebaasi füüsilise disaini antimustrid, päringute ja andmete muutmise antimustrid ja rakenduste arendamise antimustrid [19].

Phil Factor võtab raamatus „SQL Code Smells“ kokku SQL koodi halvad lõhnad ehk SQL koodi stiilid, mis pole otseselt vead, aga mis viitavad koodis olevatele andmebaasi disaini probleemidele. Ta on jaganud raamatu kaheksaks erinevaks peatükiks, kus on välja toodud erinevad SQL andmebaasi probleemid, mis viitavad andmebaasi refaktoreerimise vajadusele. [20] Refaktoreerimine tähendab andmebaasi ülesehituse muutmist, ilma, et muutuks erinevat tüüpi üldistatud väidete hulk, millele vastavaid tõeseid väiteid saab andmebaasis registreerida.

Blaha [21], Lapõnin [22] ja Eessaar [23] on näited uuringutest selle kohta, kui hästi on konkreetsed SQL-andmebaasid disainitud. Kõigis nendes uuringutes leiti uuritud andmebaasidest palju erinevat tüüpi vigu.

3 Metoodika

Selles peatükis antakse ülevaade lähteandmetest ja nende asukohast. Kirjeldatakse üldist tööprotsessi ja tutvustatakse andmesiirde jaoks kasutatud tööriistu. Selgitatakse andmesiirde vahendi valimise protsessi ja põhjendatakse tehtud valikuid.

3.1 Objekti kirjeldus

Ettevõttel on e-pood kolmes erinevas riigis – Eestis, Lätis ja Leedus. Eesti ja Läti e-pood on omavahel sarnased. Need on ühel platvormil, mis on loodud kasutades .NET raamistikku ning MS SQL Serveri andmebaasisüsteemi. Neil on riigiti erinevad andmebaasid, aga sarnane skeem ehk tabelite struktuur. Leedu e-pood on teistest erinev. See asub teisel platvormil ja on loodud Django raamistikule ning rakendus kasutab PostgreSQL'i andmebaasisüsteemi. Leedu andmebaasi skeem on teistest riikidest erinev.

Kuna ettevõtte soovib Leedu e-poe teistega ühisele platvormile üle kolida, siis tuleb Leedu PostgreSQL andmebaasist vajalikud andmed üle tuua uude Leedu MS SQL Serveri andmebaasi.

Ettevõttel on Leedus pood, kus toimub ka Leedu e-poe kauba väljastamine. Tavapoest tehtud tellimused ja maksed e-poe andmebaasis aga ei kajastu.

Sihtandmebaas on loodud kasutades MS SQL Serveri andmebaasisüsteemi. Andmebaasi skeem loodi umbes 2005. aastal ning see on ettevõtte arendusmeeskonna enda tehtud. Igal riigil on oma andmebaas, milles on 204 tabelit. Andmebaasi skeem on jäik, olemasolevaid tabeleid ja veerge kustutada ja muuta ei tohi. Mõjuval põhjusel võib andmebaasiobjekte lisada.

Lähteandmed asuvad PostgreSQL andmebaasis. Andmebaasi struktuur on genereeritud Django Oskar [24] raamistiku mudelite põhjal, millele ettevõtte arendusmeeskond on lisanud juurde vajalikku lisafunktsionaalsust. Andmebaas loodi umbes 2017. aastal ja selles on kokku 163 tabelit.

Lähteandmebaasi skeem erineb sihtandmebaasi omast. Kuna andmebaaside skeemid on erinevad, siis ei saa tabelites olevaid andmeid üks-ühele üle tuua. Tabeleid on palju, seega kogu andmebaasi andmekoosseisu ületoomine pole mõistlik, ega ka vajalik.

Töö teostamisel võeti ette sihtandmebaasi skeem ja selles olevad tabelid, mis on rakenduses toimimiseks ja toodete müügigarantii teostamiseks vajalikud. Ettevõtte selgitas välja, millised on nende jaoks olulised andmed, mida tuleb kindlasti uude andmebaasi üle tuua. Nendeks olid kasutajate andmed, tellimuste andmed ja tellimuste ajalugu, mis võimaldaks müüdnud toodetele garantiid teostada. Toodete UPC koodid (*Universal Product Code*) on mõlemas andmebaasis samad, seega tooteid üle tooma ei pea.

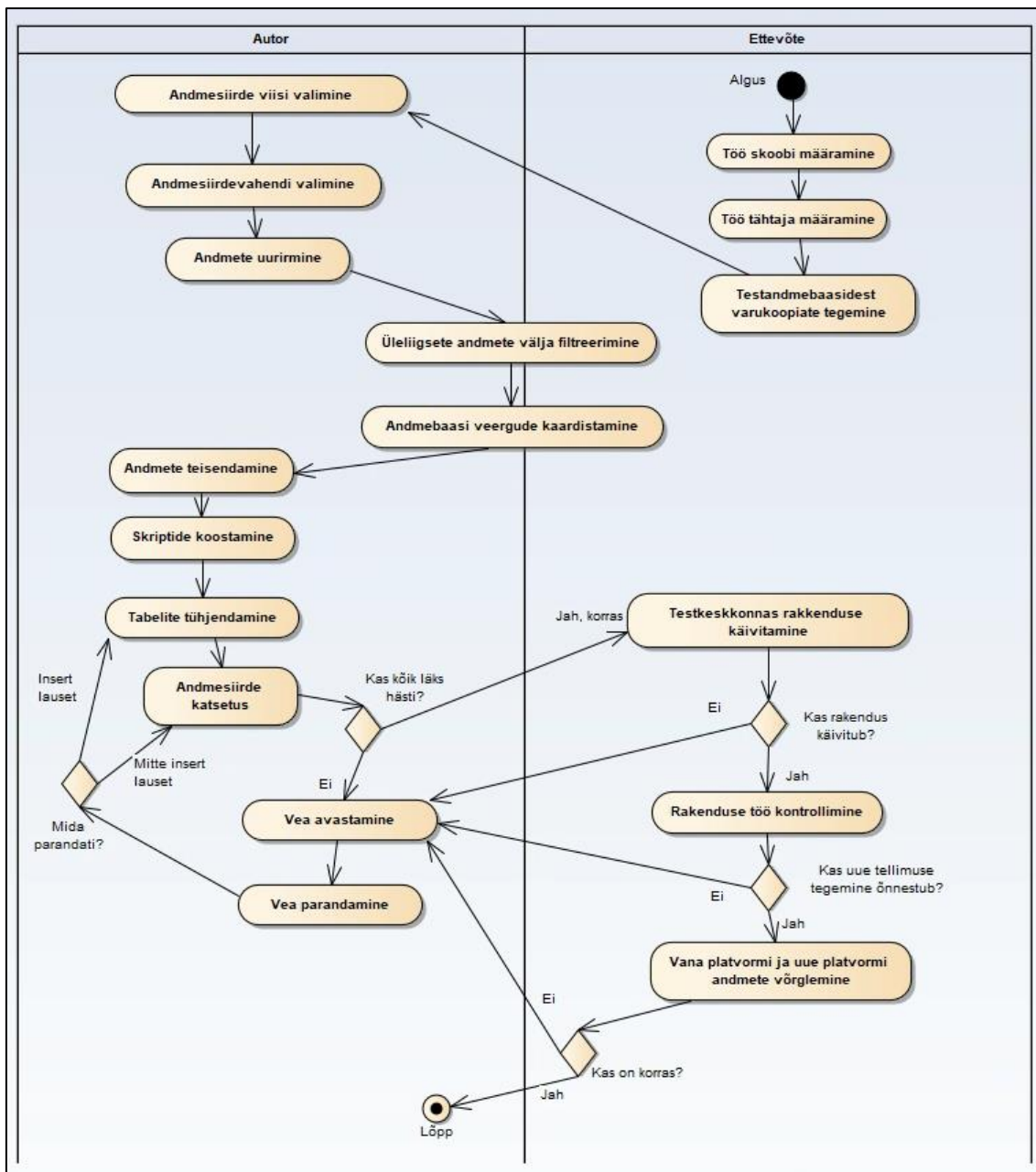
Seega on vaja sihtandmebaasis kanda andmed järgmistesse tabelitesse: *User*, *Order*, *Order_product* ja *Mail_campaign_participant* (kokku neli tabelit).

3.2 Tööprotsessi kirjeldus

Joonis 1 esitab antud töö jaoks vajalike toimingute UML tegevusdiagrammi. Diagramm on jagatud ära kahe ujumisraja vahel, et näidata toimingute jaotust autori ja ettevõtte vahel. Toimingud, mis jäävad kahe ujumisraja joone peale, kujutavad endast koostöö kohti. Töö teostamisel jälgiti andmesirde teostamise etappe (vt jaotis 2.1).

Nagu jooniselt näha siis töö skoobi ja tähtaja määras ettevõtte ning nende poolt tulid ka testandmebaaside varukoopiad, mille alusel autor lõi endale testandmebaasid. Andmesirde viis ja vahendi valimine oli autori otsustada. Üleliigsete andmete välja filtreerimine ja andmebaaside veergude kaardistamine toimus tihedas koostöös ettevõtte arendajaga. Andmesirdeks vajalike SQL-lausetega skriptide koostamine ja autori kohalikus arvutis andmesirde läbimängimine oli autori ülesanne. Testimise eesmärgil viidi samasugune andmesiire läbi ka ettevõtte arendaja arendusmasinas. Seejärel käivitati kohaliku masina külge ühendatud veebirakendus ja kontrolliti, kas see läheb käima. Pärast rakenduse käivitamist testis ettevõtte arendaja veebilehitsejaga veebirakenduses esitatavate andmete õigsust.

Tegelik andmesiire loodud skriptide alusel viiakse läbi peale töö valmimist.



Joonis 1 Andmesiirde ettevalmistamise üldise protsessi tegevusdiagramm.

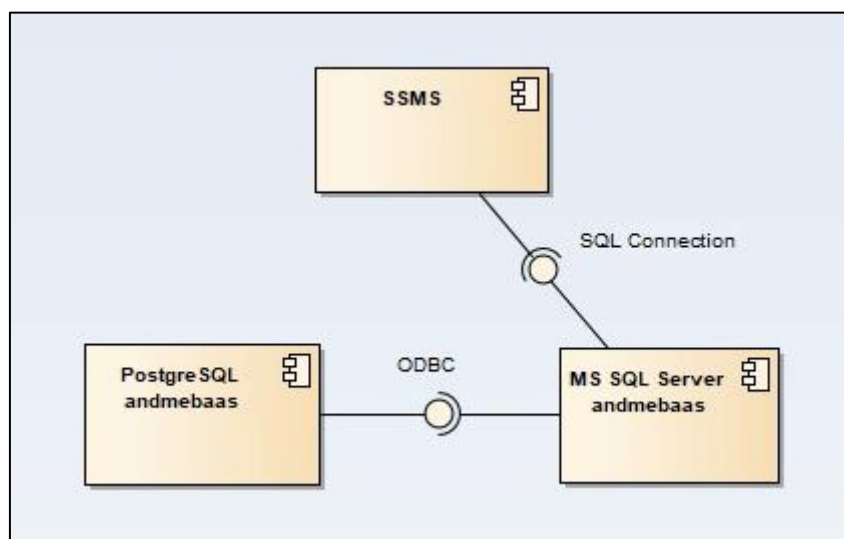
3.3 Tööriistade kirjeldus

Töös kasutati mitmeid programme. Testandmebaasid loodi MS SQL Serveri Express väljaande (versioon 15.0.2000.5) ja PostgreSQL (versioon 14.2-1) andmebaasisüsteemide abil. Ettevõtte kasutab MS SQL 2019 a versiooni 15.0.4198 ja PostgreSQL versiooni 13.2 kuid autori hinnangul (mida tõestas ka testimine arendaja arvutis) ei tekita versioonide erinevused tegelikul andmesiirdel probleeme. MS SQL Serveri Express väljaannet kasutati, sest see on tasuta.

Veergude esialgne kaardistamine ja vastavusse seadmine tehti Excelis. See vahend valiti, kuna see oli lihtsasti kättesaadav ja kohe autori arvutis olemas ning selle kasutamiskogemus oli autoril juba olemas.

Kuna lähteandmebaas on tehtud PostgreSQLis, siis nende andmete tutvumiseks kasutati tööriista pgAdmin 4. PgAdmin 4 on graafilist kasutajaliidest pakkuv PostgreSQL andmebaaside haldamise tööriist. Seda saab käivitada veebi- või töölauarakendusena. [25] Antud töös kasutati töölauarakendust. Kuna pgAdmin kippus tihti hanguma, siis selle kõrvale võeti töö lõpu poole kasutusse ka DataGrip. DataGrip on JetBrains'i loodud andmebaaside haldamise tööriist [26]. Seda kasutati peamiselt koodi vormindamiseks.

Sihtandmebaas on tehtud MS SQL Serveri andmebaasisüsteemis ja seal olevaid andmeid uuriti Microsoft SQL Server Management Studio (edaspidi SSMS) (versioon 18.11.1), vahendusel (vt Joonis 2).



Joonis 2 SSMS tarkvara seos teiste töös kasutatud programmidega.

SSMS on integreeritud keskkond, mis sobib erinevate SQL-andmebaasisüsteemide andmebaaside haldamiseks. See pakub tööriistu SQL Serveri konfigureerimiseks, jälgimiseks ja haldamiseks. Töös kasutati SSMS versiooni 18.11.1, mis oli töö koostamise hetkel kõige uuem kasutuses olev versioon, väljalaskekuupäevaga 08.03.2022. [27]

PostgreSQL andmebaasi ühendamiseks kasutati MS SQL Serveri Linked Serveri funktsionaalsust. Selleks tuli *Stack Builder*'ga installeerida PostgreSQL ODBC draiverid. Pärast draiverite installeerimist oli võimalik konfigureerida ODBC andmeallikas (*ODBC data source*). Edasi sai SSMS'i kaudu luua Linked Serveri objekti. Linked Server konfigureeriti SSMS'is interaktiivse abivahendiga ehk viisardiga (*Wizard*'ga). Pärast Linked Serveri lisamist oli SSMS valmis SQL päringute käivitamiseks üle kahe andmebaasi.

Kuna andmebaasid koos skeemidega olid olemas, siis polnud mõtet hakata nullist andmebaaside disaini esitavaid diagramme joonistama. Andmebaaside füüsilist disaini esitavad diagrammid koostati tööriistaga *Dbeaver* (versioon 22.0.2), mis võimaldas olemasolevate andmebaaside põhjal koostada automaatselt andmebaasi tabelite disaini esitavad joonised. Joonised on esitatud töö jaotistes 4.1 ja 4.2 .

Autori enda koostatud skeemide ehk diagrammide tegemiseks kasutati visuaalse modelleerimise tööriista *Enterprise Architect*.

3.4 Andmesirdevahendi valimine

Selles jaotises kirjeldatakse andmesirdevahendi valimise protsessi ja põhjendatakse tehtud valikuid.

Andmesirdevahendi valimisel lähtuti järgmistest aspektidest.

Sihtandmebaasi skeem on etteantud ja seda töö käigus muuta ei tohi. Põhjendatud vajadusel võib veerge juurde lisada, aga kustutada midagi ei tohi. Vahendite valikusse võeti ainult need tööriistad, mis toetavad etteantud jäika sihtandmebaasi skeemi.

Kuna tegu on ettevõttele oluliste andmetega, siis andmeleket ei tohi tekkida. Seega jäid valikust välja ka onlain tööriistad, kuna need pole autori hinnangul piisavalt turvalised.

Valikust jäid välja ka pilveteenused, kuna vastavalt isikuandmete kaitse üldmäärusele ei tohi delikaatseid isikuandmeid ilma isiku eelneva nõusolekuta kolmandatele osapooltele üle anda.

Kõike eelnevat arvesse võttes otsustati töö teostamiseks teha valik nende tööriistade hulgast, kus organisatsioonid saavad ise kirjutada andmete migratsiooni skripte.

Võrdlusesse võeti ainult need tööriistad, millega saab SQL skripte käsitsi kirjutada ja käivitada. Valikust jäid välja viisardi-põhised tööriistad, kuhu polnud võimalik käsitsi skripte lisada. Näiteks sel põhjusel jäi valikust välja *ESF Database Migration Toolkit* [28].

3.4.1 Kriteeriumid

Andmesirdevahendite valikul lähtuti järgnevatest kriteeriumitest.

- Peab toetama ühendumist PostgreSQL lähteandmebaasiga.
- Peab toetama ühendumist MS SQL Serveri sihtandmebaasiga.
- Hind – Raha on piiratud ressurss. Autoril on töö eesmärgil võimalik kasutada tasuta versioone või tasulise versiooni tasuta testperioodi vältel.
- Töövahendi keerukus ja selle õppimiskõver – Kuna andmesirde aeg on kindlaks määratud, siis on töövahendi selgeks saamise ja kasutamisele võtmiseks ajaline piirang.
- Dokumentatsioon – Kas rakenduse kohta on piisavalt informatsiooni selle kasutamise kohta?
- Andmevahetuse protsess – Kas vaja on laadida sisse mõlema andmebaasi varukoopia või saab rakendusega otse andmebaasi külge ühenduda? Eelistatum oleks, et poleks andmete failidesse eksportimist ja importimist. Süsteem peab ühenduma nii PostgreSQL andmebaasiga kui ka MS SQL Serveri andmebaasiga.

3.4.2 Kriteeriumite hindamine ja tööriistade võrdlemine

Andmesirde tööriista valimisel kasutati võrdlustel põhinevat meetodikat, mille autor ise välja mõtles. Selleks hindas autor koostöös ettevõtte arendajaga kõigepealt kriteeriumite tähtsust.

Kriteeriumite tähtsust hinnati kolme palli süsteemis (vt Tabel 1), kus

- 1 = pigem tähtis,
- 2 = tähtis,

- 3 = väga tähtis.

Tabel 1 Tööriistade kriteeriumite tähtsuse hindamine.

Kriteeriumid	Tähtsus
Kas toetab PostgreSQL	3
Kas toetab MS SQL Serverit	3
Hind	1
Töövahendi keerukus ja selle õppimiskõver	3
Dokumentatsioon	2
Andmevahetuse protsess	3

Valikusse võeti järgmised tööriistad: Microsoft SQL Server Management Studio (SSMS)(versioon 18.11.1), DataGrip (versioon 2021.3.4) ja Dbeaver.

Edasi vaadati, kas valikusse võetud tööriistad vastavad kriteeriumitele. Tabel 2 tähistab „v“ – kriteeriumi täitmist ja „-“ (miinus) tähistab kriteeriumi mitte täitmist. Kasutatava meetodi nõrkuseks võrreldes näiteks analüütiliste hierarhiate meetodi ehk Saaty meetodiga [29] on see, et kriteeriumi täitmise ulatust ehk suhtelist täidetust (näiteks vahend 1 täidab 50%, vahend 2 täidab 70%) ei hinnatud.

Tabel 2 Tööriistade vastavus kriteeriumitele.

Kriteeriumid	SSMS	DataGrip	Dbeaver
Kas toetab PostgreSQL	v	v	v
Kas toetab MS SQL Serverit	v	v	v
Hind	v	v	v
Töövahendi keerukus ja selle õppimiskõver	v	v	v
Dokumentatsioon	v	v	v
Andmevahetuse protsess	v	-	-

DataGrip’l ja Dbeaver’l ei ole Linked Serveri tuge. Seega ei täitnud need vahendid andmevahetuse protsessi kriteeriumit.

Tegelikult saab juba Tabel 2 põhjal öelda, et on vaid üks tööriist – SSMS, mis täidab kõiki kriteeriumeid.

Juhul kui erinevad tööriistad täidaksid erinevaid kriteeriumeid, tuleks võrrelda valikusse võetud tööriistade tulemust kriteeriumite tähtsusega ja valida tööriist, mis saab kõrgeima lõpphinde. Antud juhul on tulemus sama – parima hinnangu saab SSMS.

Lõplik hinne:

$$\text{SSMS} = 3 + 3 + 1 + 3 + 2 + 3 = 15$$

$$\text{DataGrip} = 3 + 3 + 1 + 3 + 2 + 0 = 12$$

$$\text{Dbeaver} = 3 + 3 + 1 + 3 + 2 + 0 = 12$$

3.5 Metoodika kokkuvõte

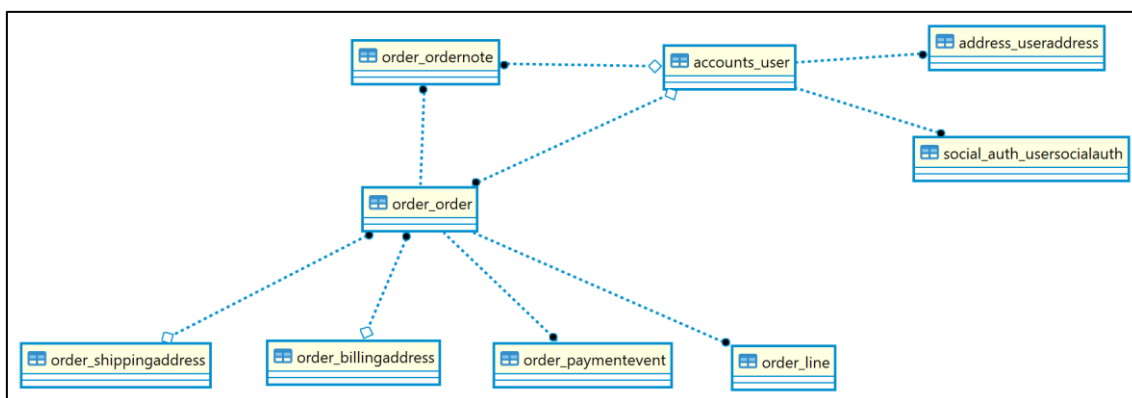
Kokkuvõttes keskendub antud töö andmebaasi migratsioonile. Andmed kantakse üle kahe SQL-andmebaasisüsteemi vahel, millel on erinev tootja. Lähteandmebaasi ja sihtandmebaasi struktuur on erinev. Tegemist on suure pauguga andmete ülekandmisega, kus andmed kantakse korraga uude andmebaasi üle ja siis võetakse see kohe kasutusele. Töös käsitletav andmete ülekanne on osa suuremast protsessist, mis hõlmab ka rakenduse migratsiooni. Andmete ülekandeks kasutatakse skriptide kirjutamist. Andmesiirdevahendiks valiti SSMS. Antud töö tulemusena valmivad ülekandeks vajalikud skriptid. Tegelik andmesiire loodud skriptide alusel toimub peale töö esitamist.

4 Töö tulemused

Selles peatükis kirjeldatakse töö tulemusi. Kirjeldatakse sihtandmebaasi neid tabeleid, kuhu andmeid üle kantakse. Esitatakse sihtandmebaasi nende tabelite veergude kirjeldused ja joonised tabelite omavaheliste seostega. Iga seos joonisel tähistab välisvõtme veergu ja sellele loodud välisvõtme kitsendust. Kirjeldatakse andmesiirde protsessi ja esitatakse erinevad arvulised näitajad, mis iseloomustavad loodud lahendust.

4.1 Lähteandmebaasi füüsilise disaini mudel

Selles jaotises tutvustatakse lähemalt lähteandmebaasi ja selle füüsilist disaini. Andmesiirde käigus võeti andmeid üheksast lähteandmebaasi tabelist. Joonis 3 kirjeldab lähteandmebaasis kasutatavate tabelite üldist struktuuri ja omavahelist seost.



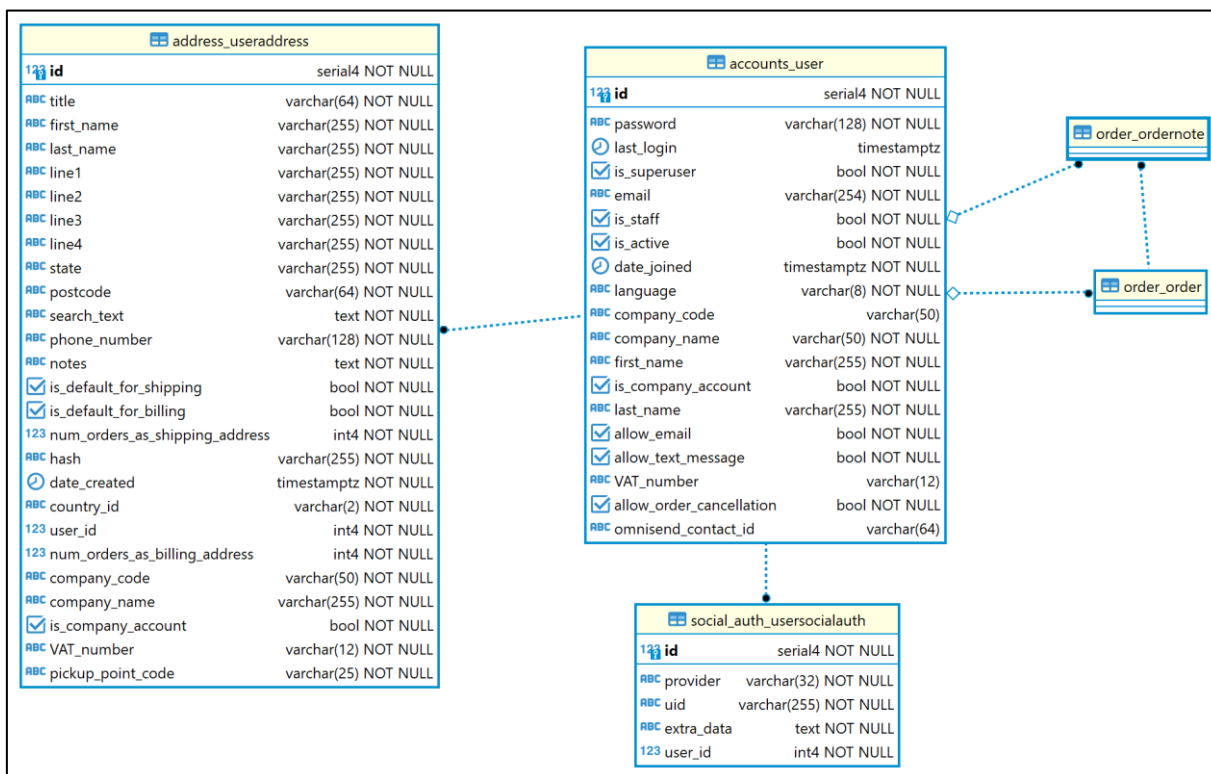
Joonis 3 Lähteandmebaasis kasutatud tabelite üldine struktuur.

Tabelite veergude juures tähistatakse „V“ abil veerge, milles olevaid andmeid on vaja sihtandmebaasi üle kanda ja „-“, (miinus) abil tähistatakse veerge, milles olevaid andmeid ei ole vaja üle kanda.

4.1.1 *Accounts_user* tabel

E-poe kasutajate andmeid hoitakse lähteandmebaasis tabelis nimega *Accounts_user*. Joonis 4 on näha tabeli *Accounts_user* seoseid teiste andmesiirde käigust kasutatud tabelite vahel. Joonis 5 on näha tabelite *Accounts_user*, *Address_useraddress* ja

Social_auth_usersocialauth veergude andmetüüpe, väljapikkuseid ja kas *NULL* on veerus lubatud ehk kas sellele veerule vastavates väljades võib väärtus puududa või mitte.



Joonis 4 Lähteandmebaasi tabel *Accounts_user* ja sellega seotud tabelid.

Tabelis *Accounts_user* on 19 veergu. Tabel 3 esitab *Accounts_user* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 3 Lähteandmebaasi tabeli *Accounts_user* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
V	id	Kasutaja numbriline identifikaator. Primaarvõti.
V	password	Parool ehk salasõna, mida kasutatakse isikusamasuse tõendamiseks ja juurdepääsu saamiseks.
V	last_login	Kuupäev ja kellaaeg, millal kasutaja viimati e-poodi sisselogis.
-	is_superuser	Kas kasutajal on õigus kogu süsteemi hallata ehk kas ta on ülikasutaja? Veerg on mõeldud omatöötajatele.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
V	email	Kasutajaga ühenduse saamiseks kasutatav e-posti aadress.
-	is_staff	Kas kasutaja on omatöötaja?
V	is_active	Kas kasutajakonto on aktiveeritud? Kasutajakonto on aktiveeritud siis, kui ta e-posti aadress on kinnitatud.
V	date_joined	Kuupäev ja kellaaeg, millal kasutaja registreeris end e-poe kasutajaks.
V	language	Kasutaja keeleline eelistus e-poes.
-	company_code	Juriidilise isiku registreerimise kood äriregistris. Väärtus registreeritakse siis, kui kasutajaks on juriidiline isik.
V	company_name	Juriidilise isiku korral ettevõtte nimi. Väärtus registreeritakse siis, kui kasutajaks on juriidiline isik.
V	first_name	Kasutaja eesnimi. Kui kasutajaks on juriidiline isik, siis täidetakse veerg kontaktisiku eesnimega.
V	is_company_account	Kas tegu on juriidilise isiku kontoga?
V	last_name	Kasutaja perekonnanimi. Kui kasutajaks on juriidiline isik, siis täidetakse veerg kontaktisiku perekonnanimega.
V	allow_email	Kas turunduslikud e-kirjad kliendi e-posti aadressile on kliendi poolt lubatud?
V	allow_text_message	Kas turunduslikud SMS'id kliendi mobiiltelefonile on kliendi poolt lubatud?
V	VAT_number	Juriidilise isiku käibemaksukohuslase registreerimise number (KMKR). Väärtus registreeritakse siis, kui kasutajaks on juriidiline isik.
-	allow_order_cancellation	Kas poetöötajal on õigus tagakontoris tellimusi tühistada. Veerg on mõeldud oma töötajatele.
V	omnisend_contact_id	Turundussüsteemi kasutaja identifikaator.

4.1.2 Address_useraddress tabel

Lähteandmebaasi kasutajate aadresse hoitakse tabelis nimega *Address_useraddress*, milles on 26 veergu. Tabel 4 esitab *Address_useraddress* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 4 Lähteandmebaasi tabeli *Address_useraddress* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Kasutaja aadressi numbriline identifikaator. Primaarvõti.
-	title	Nime ees käiv tiitel. Näited on Mr, Miss, Mrs, Ms ja Dr.
V	first_name	Kasutaja eesnimi. Kui kasutajaks on juriidiline isik, siis täidetakse veerg kontaktisiku eesnimega.
V	last_name	Kasutaja perekonnanimi. Kui kasutajaks on juriidiline isik, siis täidetakse veerg kontaktisiku perekonnanimega.
V	line1	Kasutaja elukoha aadressi tänava nimi ja maja number või talu nimi. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi tänava nimi ja maja number või talu nimi.
V	line2	Kasutaja elukoha aadressi korteri number. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi korteri number. Kui kasutaja aadressis ei ole korteri numbrit, siis seda veergu ei täideta.
-	line3	Vana veerg, seda ei kasutata.
V	line4	Kasutaja elukoha aadressi küla või linn. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi küla või linn.
V	state	Kasutaja elukoha aadressi maakond. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi maakond.
V	postcode	Kasutaja elukoha aadressi postiindeks. Juriidilise isiku korral kirjutatakse tema

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
		Äriregistris registreeritud aadressi postii indeks.
-	search_text	Veergu salvestatakse aadress kokku ühe väärtusena, et lihtsustada otsimist. Antud projektis seda veergu ei kasutata.
V	phone_number	Isikuga ühenduse saamiseks kasutatav telefoninumber.
-	notes	Märkmed, juhised kullerile kauba toimetamiseks kauba kättesaamise asukohta.
-	is_default_for_shipping	Kas kasutaja aadress on vaikimisi ka kauba kättesaamise aadress?
-	is_default_for_billing	Kas kasutaja aadress on vaikimisi ka arve saaja aadress?
-	num_orders_as_shipping_address	Mitu korda on seda kasutaja elukoha aadressi tarneaadressina kasutatud?
-	hash	Iga aadressi kohta arvutatakse räsiväärtus, et samale isikule ei registreeritaks korduvaid aadresse.
V	date_created	Kuupäev ja kellaaeg, millal sisestati süsteemi kasutaja aadress.
V	country_id	Kasutaja elukoha aadressi riigi identifitseerimise kood. Riigi tunnus kahetähelise lühendina, mis vastab ISO 3166-1 alpha-2 standardile.
V	user_id	Kasutaja numbriline identifikaator. Viide kasutajate (<i>Accounts_user</i>) tabelile. Välisvõti.
-	num_orders_as_billing_address	Mitu korda on seda kasutaja elukoha aadressi arve saaja aadressina kasutatud?
-	company_code	Kasutaja ettevõtte äriregistri kood. Kasutatakse siis, kui kasutajaks on juriidiline isik.
V	company_name	Kasutaja ettevõtte nimi. Kasutatakse siis, kui kasutajaks on juriidiline isik.
-	is_company_account	Kas kasutaja on juriidiline isik?

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	vat_number	Juriidilise isiku käibemaksukohuslase registreerimise number (KMKR). Väärtus registreeritakse siis, kui kasutaja on juriidiline isik.
-	pickup_point_code	Kauba väljastuspunkti kood.

4.1.3 Social_auth_usersocialauth tabel

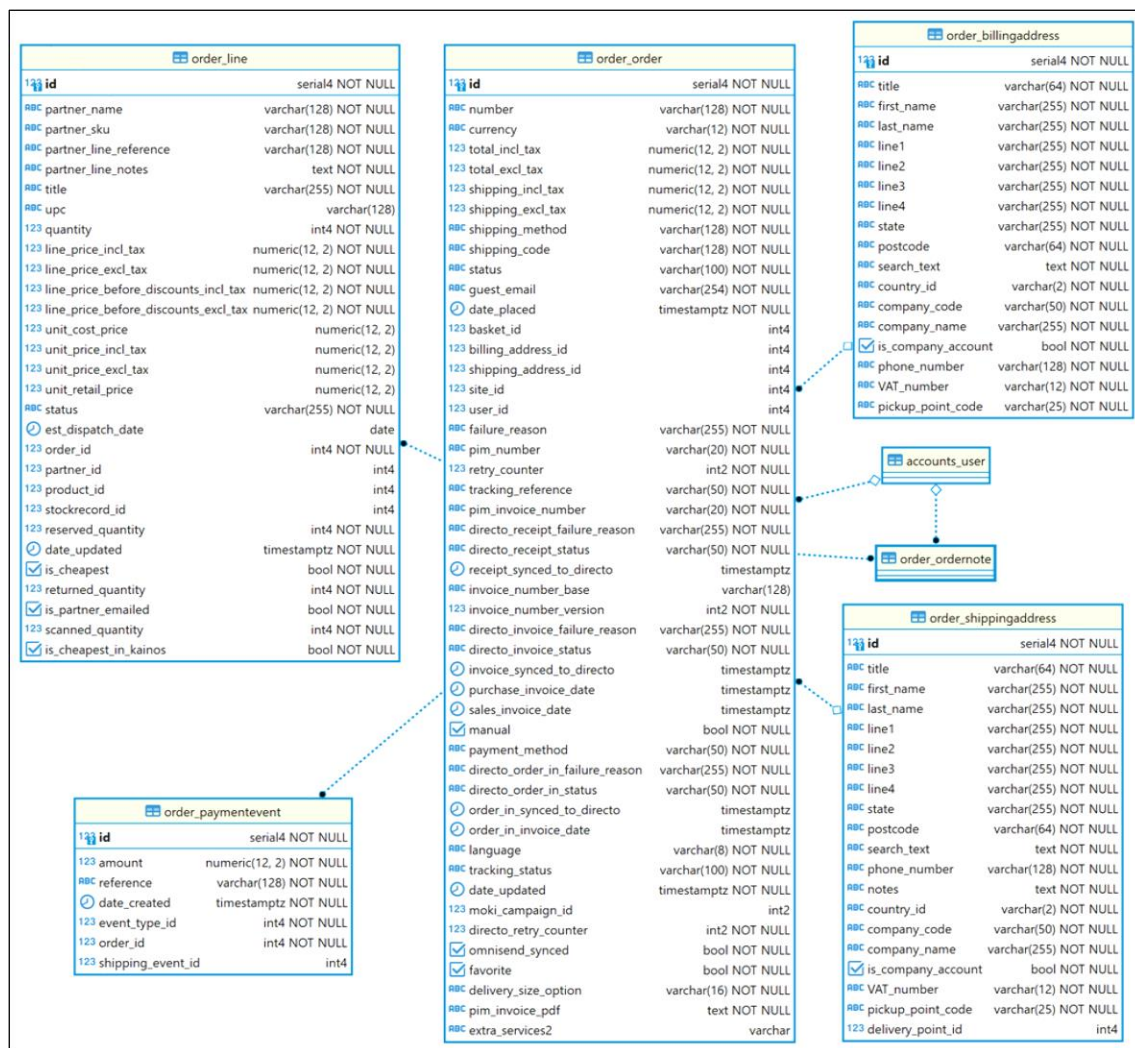
Lähteandmebaasi tabelis nimega *Social_auth_usersocialauth* hoitakse kasutaja sotsiaalmeediaga seotud infot. Selles tabelis on 5 veergu. Tabel 5 esitab *Social_auth_usersocialauth* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 5 Lähteandmebaasi tabeli *Social_auth_usersocialauth* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Numbriline identifikaator. Primaarvõti.
-	provider	Kasutaja süsteemi sisenemisel kasutatav enesetuvastamise viis. Kas Google kontoga või Facebooki kontoga?
V	uid	Kasutaja identifikaator valitud süsteemis. Facebook'il on kasutaja numbriline identifikaator ja Google'l on e-posti aadress.
-	extra_data	Kasutaja konto sidumisel sotsiaalmeedia platvormiga selle platvormi poolt saadetud päringu vastus e-poe süsteemile. Info salvestatakse veergu <i>JSON</i> formaadis.
-	user_id	Kasutaja numbriline identifikaator. Viide kasutajate(user) tabelile

4.1.4 Order_order tabel

Lähteandmebaasi tabelis nimega *Order_order* hoitakse tellimustega seotud andmeid. Joonis 5 on kujutatud *Order_order* tabeli seoseid teiste töös kasutatud lähteandmebaasi tabelitega, veergude andmetüüpe, väljapikkuseid ja kas *NULL* on veerus lubatud ehk kas sellele veerule vastavates väljades võib väärtus puududa või mitte.



Joonis 5 Lähteandmebaasi tabel *Order_order* ja sellega seotud tabelid.

Order_order tabelis on 49 veergu. Tabel 6 esitab *Order_order* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 6 Lähteandmebaasi tabeli *Order_order* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Tellimuse numbriline identifikaator. Primaarvõti.
V	number	Tellimuse number, mida kasutatakse arvetel. Kliendile näidatakse arvel tellimuse numbrit, mitte tellimuse id väärtust.
V	currency	Kolmetäheline valuuta tähis, mis vastab ISO 4217 standardile. Näide on EUR.

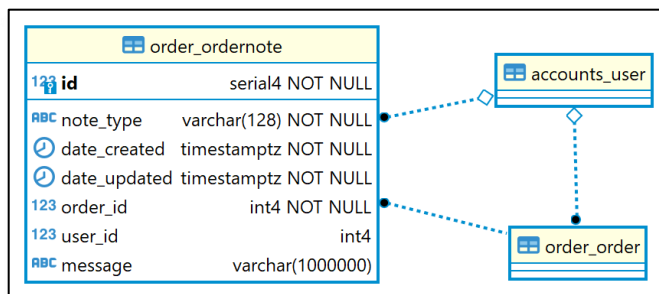
Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	total_incl_tax	Tellimuse kogumaksumus koos käibemaksuga tellimuse tegemise hetkel.
-	total_excl_tax	Tellimuse kogumaksumus ilma käibemaksuta tellimuse tegemise hetkel.
V	shipping_incl_tax	Tarne kogumaksumus koos käibemaksuga tellimuse tegemise hetkel.
-	shipping_excl_tax	Tarne kogumaksumus ilma käibemaksuta tellimuse tegemise hetkel.
-	shipping_method	Tarne meetod.
V	shipping_code	Tarne meetodi kood.
V	status	Tellimuse seisund ehk olek inimloetavas keeles, sõnena.
V	guest_email	Mitteregistreeritud kasutaja e-posti aadress. Mitteregistreeritud kasutaja on süsteemi kasutaja, kes pole ennast e-poe süsteemis kasutajaks registreerinud.
V	date_placed	Kuupäev ja kellaaeg, millal kasutaja läheb ostu eest tasuma.
-	basket_id	Ostukorvi identifikaator. Välisvõti. Viitab tabelile <i>Basket_basket</i> .
-	billing_address_id	Arve saaja andmete identifikaator. Välisvõti. Viitab tabelile <i>Order_billingaddress</i> .
-	shipping_address_id	Kauba kättesaaja identifikaator. Välisvõti. Viitab tabelile <i>Order_shippingaddress</i> .
-	site_id	E-poe identifikaator. Välisvõti. Viitab tabelile <i>Django_site</i> .
V	user_id	Kasutaja numbriline identifikaator. Viitab kasutajate (<i>User</i>) tabelile.
-	failure_reason	Kui tellimusega tekib probleem, siis süsteem kirjutab sellesse veergu probleemi põhjuse. Kui probleem kõrvaldatakse, siis seda väärtust ära ei kustutata.
V	pim_number	Raamatupidamissüsteemi tellimuse identifikaator.
-	retry_counter	Probleemide korral proovitakse toimingut uuesti teha. Süsteem kirjutab veergu, mitu korda prooviti

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
		toimingut uuesti teha enne kui asi laabus või kuni maksimaalne proovide arv sai täis.
-	tracking_reference	Pakiautomaadi või kulleri poolt saadetud paki jälgimise kood.
V	pim_invoice_number	Ostuarve number raamatupidamissüsteemis, mille väljastab hulгимүтја.
-	directo_receipt_failure_reason	Ostuarve sünkroniseerimisel müügisüsteemiga tekkinud vea korral salvestatakse veergu sünkroniseerimise vea põhjus.
-	directo_receipt_status	Ostuarve sünkroniseerimisel müügisüsteemiga määratud sünkroniseerimise seisund ehk olek.
-	receipt_synced_to_directo	Kuupäev ja kellaaeg, millal ostuarve sünkroniseeriti müügisüsteemiga.
-	invoice_number_base	Arvnumber.
-	invoice_number_version	Näitab, mitu korda on arvet pärast tellimuse tegemist muudetud. Veerus olev väärtus näitab mitmes versioon praegu arvest kasutusel on.
-	directo_invoice_failure_reason	Müügiarve sünkroniseerimisel müügisüsteemiga tekkinud vea korral salvestatakse sünkroniseerimise vea põhjus.
-	directo_invoice_status	Müügiarve sünkroniseerimisel müügisüsteemiga määratud sünkroniseerimise seisund ehk olek.
-	invoice_synced_to_directo	Kuupäev ja kellaaeg, millal müügiarve sünkroniseeriti müügisüsteemiga.
-	purchase_invoice_date	Kuupäev ja kellaaeg, millal kaup laost välja saadetakse.
-	sales_invoice_date	Kuupäev ja kellaaeg, millal ostja kauba kätte sai.
-	manual	Kas käesolev tellimus on tehtud poe töötaja poolt, et täiendada poe laoseisu või tellida kliendi soovil talle tooteid?
-	payment_method	Tellimuse koostamisel valitud makse meetod.
-	directo_order_in_failure_reason	Laoseisu täiendamiseks tellitud kauba sisseostuarve müügisüsteemiga sünkroniseerimisel tekkinud vea korral salvestatakse veergu sünkroniseerimise vea põhjus.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	directo_order_in_status	Laoseisu täiendamiseks tellitud kauba sisseostuarve müügisüsteemiga sünkroniseerimise seisund ehk olek.
-	order_in_synced_to_directo	Kuupäev ja kellaaeg, millal laoseisu täiendamiseks tellitud kauba sisseostuarve sünkroniseeriti müügisüsteemiga.
-	order_in_invoice_date	Kuupäev ja kellaaeg, millal loodi laoseisu täiendamiseks tellitud kauba sisseostuarve.
V	language	Keel, milles tellimus koostati. Märgitakse ISO standardiga määratud kahetähelise koodina.
-	tracking_status	Tarne seisund ehk olek paki teekonna jälgimisel kauba saajale.
-	date_updated	Kuupäev ja kellaaeg, millal uuendati tellimust.
-	moki_campaign_id	Liisingüsteemi kampaania identifikaator. Välisvõti.
-	directo_retry_counter	Mitu korda on üritatud käesoleva tellimusega müügisüsteemi teenuse poole pöörduda.
V	omnisend_synced	Kas tellimuse andmed on turundustööriistaga sünkroniseeritud?
-	favorite	Kas antud tellimusele on märk külge pandud? Tagakontoris on võimalus tellimusele märk külge panna, et seda hiljem jälgida.
-	delivery_size_option	Arvutuslik paki suurus. Väline teenus ütleb, mis suurusega pakki tellimus mahutada kas S, M, L või XL.
-	pim_invoice_pdf	E-poe ostuarve hulgemüüjalt PDF faili kujul.
-	extra_services	Lisateenused. Kas klient soovib tuppavedu või kas soovitakse, et kaup pakitakse kingituseks? Väärtuse näol on tegemist massiiviga, kuhu süsteem salvestab kliendi poolt tehtud valikud ette antud võimalikest väärtustest.

4.1.5 Order_ordernote tabel

Order_ordernote tabelis on 7 veergu ja sinna salvestatakse andmed tellimuste märkmete kohta (Joonis 6).



Joonis 6 Lähteandmebaasi tabel *Order_oredernote* ja sellega seotud tabelid.

Tabel 7 esitab *Order_ordernote* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 7 Lähteandmebaasi tabeli *Order_ordernote* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Tellimuse märkme identifikaator. Primaarvõti.
-	note_type	Märkme tüüp.
-	date_created	Kuupäev ja kellaaeg, millal märke tehti.
-	date_updated	Kuupäev ja kellaaeg, millal märget uuendati ehk muudeti.
-	order_id	Tellimuse identifikaator, mille kohta märke tehti. Välisvõti. Viide tellimuste (<i>Order_order</i>) tabelile.
-	user_id	Kasutaja numbriline identifikaator. Välisvõti. Viide kasutajate (<i>User</i>) tabelile. Kasutaja, kelle tellimuse kohta märke tehti.
V	message	Märkme sisu. Vabatekstiline väärtus.

4.1.6 Order_line tabel

Order_line tabelis on 29 veergu, Sellesse salvestatakse andmeid tellimusridade kohta. Tabel 8 esitab *Order_line* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 8 Lähteandmebaasi tabeli *Order_line* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Tellimuses oleva tellimusrea numbriline identifikaator.
-	partner_name	Lao nimi kust toode tuleb.
-	partner_sku	Toote kood partnerlaos. Üldjuhul on see sama kui UPC (<i>Universal Product Code</i>) kood, aga mitte alati.
-	partner_line_reference	Partnerladu võib tootele oma tootekoodi panna. Kui nii on tehtud, siis see tootekood salvestatakse sellesse veergu.
-	partner_line_notes	Partnerlao poolt tehtud märkmed.
-	title	Nime ees käiv tiitel. Näited on Mr, Miss, Mrs, Ms ja Dr.
V	upc	Tellimusreal oleva toote UPC ehk triipkood. See on kood, mida tootjad kasutavad oma toodete elektrooniliseks tuvastamiseks.
V	quantity	Ühe toote tükiarv tellimuses.
-	line_price_incl_tax	Tellimuse rea summa (ühe toote eksemplari hind korrutatuna tükiarvuga) koos käibemaksuga. Kui tootel on allahindlus, siis see on juba siia sisse arvatud.
-	line_price_excl_tax	Tellimuse rea summa (ühe toote hind korrutatuna tükiarvuga) ilma käibemaksuta. Kui tootel on allahindlus, siis see on juba siia sisse arvatud.
-	line_price_before_discounts_incl_tax	Tellimuse rea summa täishind koos käibemaksuga. Enne allahindluse arvutamist.
-	line_price_before_discounts_excl_tax	Tellimuse rea summa täishind ilma käibemaksuta. Enne allahindluse arvutamist.
V	unit_cost_price	Tellimusreal oleva toote sisseostu hind hulgimüüjalt ilma käibemaksuta.
V	unit_price_incl_tax	Tellimusreal oleva ühe toote hind käibemaksuga ostu sooritamise hetkel.
V	unit_price_excl_tax	Tellimusreal oleva ühe toote käibemaksuta hind ostu sooritamise hetkel.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
	unit_retail_price	Tellimusreal oleva toote sisseostu hind hulгимүүжalt ilma käibemaksuta. Veerus olev väärtus on võrdne sama tabeli veerus <i>unit_cost_price</i> oleva väärtusega, sest kuna e-pood müüb tooteid alati tüki kaupa, siis on kogus alati tükide
	status	Tellimusrea seisund ehk olek.
-	est_dispatch_date	Eeldatav väljasaatmise aeg.
-	order_id	Tellimuse identifikaator. Välisvõti. Viitab tabelile <i>Order_order</i> .
-	partner_id	Partneri identifikaator. Välisvõti. Viitab tabelile <i>Partner_partner</i> .
V	product_id	Toote identifikaator. Välisvõti. Viitab tabelile <i>Product_product</i> .
-	stockrecord_id	Laoseisu kirje identifitseerimise identifikaator. Viitab tabelile <i>Partner_stocrecord</i> .
-	reserved_quantity	Hulgilaos reserveeritud toote kogus tükiarvuna.
V	date_updated	Kuupäev ja kellaeg, millal teostati tellimusreal viimane muudatus.
-	is_cheapest	Kas toode oli ostuhetkel turul kõige odavam? (Esimene kontrollimise koht) Väärtus saadakse ühest hinnavaatlusportaalist tehtud päringuga.
-	returned_quantity	Tagastatud toote kogus tükides.
-	is_partner_emailed	Kas partnerladu on tellimusrea tühistusest teavitatud?
-	scanned_quantity	Skaneeritud toodete kogus. Väljastatud tooted skaneeritakse kassas, veendumaks, et väljastatud kauba kogus oleks õige.
-	is_cheapest_in_kainos	Kas toode oli ostuhetkel turul kõige odavam? (Teine kontrollimise koht) Väärtus saadakse ühest hinnavaatlusportaalist tehtud päringuga.

4.1.7 Order_billingaddress tabel

Order_billingaddress tabelis on 18 veergu, Sellesse salvestatakse andmeid arve saaja aadressi kohta. Tabel 9 esitab Order_billingaddress tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 9 Lähteandmebaasi tabeli Order_billingaddress veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Arve saaja numbriline identifikaator.
-	title	Nime ees käiv tiitel. Näited on Mr, Miss, Mrs, Ms ja Dr.
V	first_name	Arve saaja eesnimi.
V	last_name	Arve saaja perekonnanimi.
V	line1	Arve saaja elukoha aadressi tänava nimi ja maja number või talu nimi. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi tänava nimi ja maja number või talu nimi.
V	line2	Arve saaja elukoha aadressi korteri number. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi korteri number. Kui kasutaja aadressis ei ole korteri numbrit, siis seda väärtust ei registreerita.
-	line3	Vana veerg, seda ei kasutata enam.
V	line4	Arve saaja elukoha aadressi linn. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi linn. Kui arve saaja aadress ei sisalda linna, siis on kasutaja sunnitud improviseerima, millisele väljale ta oma aadressi paigutab.
V	state	Arve saaja elukoha aadressi maakond. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi maakond.
V	postcode	Arve saaja elukoha aadressi postiindeks. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi postiindeks.
-	search_text	Veergu salvestatakse aadressi andmed ühe väärtusena, et lihtsustada otsimist. Antud projektis seda veergu ei kasutata.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
V	country_id	Arve saaja elukoha aadressi riigi identifitseerimise kood. Riigi tunnus kahetähelise lühendina, mis vastab ISO 3166-1 alpha-2 standardile.
V	company_code	Ettevõtte registreerimise kood äriregistris. Väärtus registreeritakse siis, kui arvesaaja on juriidiline isik.
V	company_name	Ettevõtte nimi. Väärtus registreeritakse siis, kui arvesaaja on juriidiline isik.
V	is_company_account	Kas arvesaaja on juriidiline isik?
V	phone_number	Arve saajaga ühenduse saamiseks kasutatav telefoninumber.
-	VAT_number	Juriidilise isiku käibemaksukohustlase registreerimise number (KMKR). Väärtus registreeritakse siis, kui kaubasaaja on juriidiline isik.
-	pickup_point_code	Kauba väljastuspunkti kood.

4.1.8 Order_shippingaddress tabel

Order_shippingaddress tabelis on 20 veergu, sellesse salvestatakse andmeid kauba kättesaamise aadressi kohta. Tabel 10 esitab *Order_shippingaddress* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 10 Lähteandmebaasi tabeli *Order_shippingaddress* veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Kaubasaaja numbriline identifikaator.
-	title	Nime ees käiv tiitel. Näited on Mr, Miss, Mrs, Ms ja Dr.
V	first_name	Kaubasaaja eesnimi.
V	last_name	Kaubasaaja perekonnanimi.
V	line1	Kauba kättesaamise asukoha aadressi tänava nimi ja maja number või talu nimi.
V	line2	Kauba kättesaamise asukoha aadressi korteri number. Kui asukoha aadressis ei ole korteri numbrit, siis seda veergu ei täideta.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	line3	Vana veerg, seda ei kasutata enam.
V	line4	Kauba kättesaamise asukoha aadressi linn. Kui asukoha aadress ei sisalda linna, siis on kasutaja sunnitud improviseerima millisele väljale ta asukoha aadressi paigutab.
V	state	Kauba kättesaamise asukoha aadressi maakond.
V	postcode	Kauba kättesaamise asukoha aadressi postin indeks.
-	search_text	Veergu salvestatakse aadress ühe väärtusena, et lihtsustada otsimist. Antud projektis seda veergu ei kasutata.
V	phone_number	Kauba saajaga ühenduse saamiseks kasutatav telefoninumber.
-	notes	Märkmed, juhised kullerile kauba toimetamiseks kauba kättesaamise asukohta.
V	country_id	Kauba kättesaamise asukoha aadressi riigi identifitseerimise kood. Riigi tunnus kahetähelise lühendina, mis vastab ISO 3166-1 alpha-2 standardile.
V	company_code	Ettevõtte registreerimise kood äriregistris. Väärtus registreeritakse siis, kui kauba kättesaaja on juriidiline isik.
V	company_name	Ettevõtte nimi. Väärtus registreeritakse siis, kui kauba kättesaaja on juriidiline isik.
-	is_company_account	Kas kauba kättesaaja on juriidiline isik?
-	VAT_number	Juriidilise isiku käibemaksukohuslase registreerimise number (KMKR). Väärtus registreeritakse siis, kui kaubasaaja on juriidiline isik.
V	pickup_point_code	Kauba väljastuspunkti kood. Kui tarneviisiks on valitud „Tulen ise järele“, siis on seal poe kood, kuhu klient kaubale järele läheb. Muul juhul on see tühi sõne.
-	delivery_point_id	Tarnepunktide tabeli indikaator. Viitab tabelile, kus salvestatakse pakiautomaatide ja poodide aadresse. Välisvõti.

4.1.9 Order_paymentevent tabel

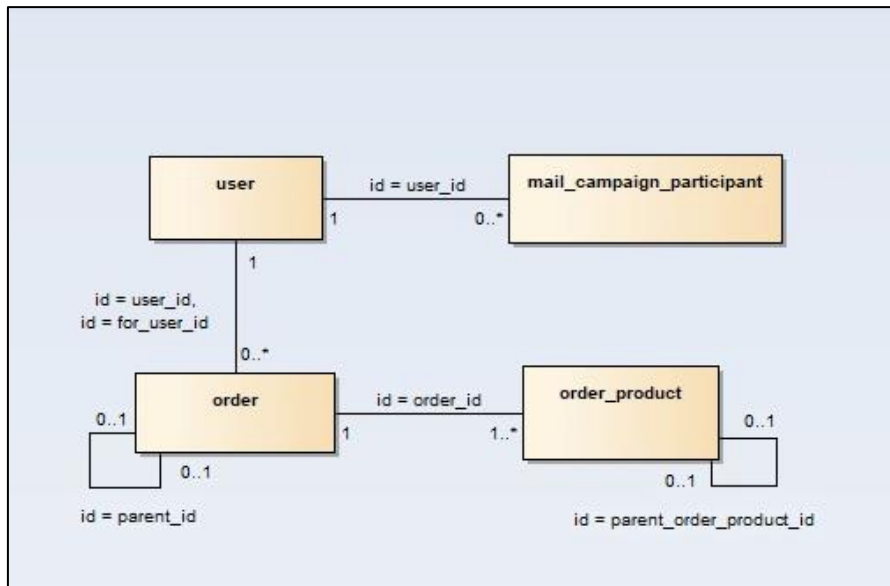
Order_paymentevent tabelis on 7 veergu, sellesse salvestatakse andmeid maksesündmuste kohta. Tabel 11 esitab Order_paymentevent tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 11 Lähteandmebaasi tabeli Order_paymentevent veergude kirjeldused.

Kasutatakse andmesiirdel	Veeru nimi	Kirjeldus
-	id	Tellimuse tasumise identifikaator.
-	amount	Makse suurus. Kahe koma täpsusega arv.
-	reference	Makse viitenumber.
V	date_created	Kuupäev ja kellaaeg, millal saabus maksete vahendajalt (Paysera LT) teavitus, et makse on tehtud.
-	event_type_id	Maksesündmuse tüübi identifikaator. Välisvõti.
-	order_id	Tellimuse numbriline identifikaator.
-	shipping_event_id	Tarne sündmuse identifikaator. Veergu ei kasutata.

4.2 Sihtandmebaasi füüsilise disaini mudel

Selles jaotises tutvustatakse lähemalt andmesiirde käigus sihtandmebaasis andmetega täidetud tabelleid ja nende füüsilist disaini. Joonis 7 on näha sihtandmebaasi tabelite omavahelised seosed.



Joonis 7 Sihtandmebaasi andmesiirde kasutatud tabelite üldine struktuur.

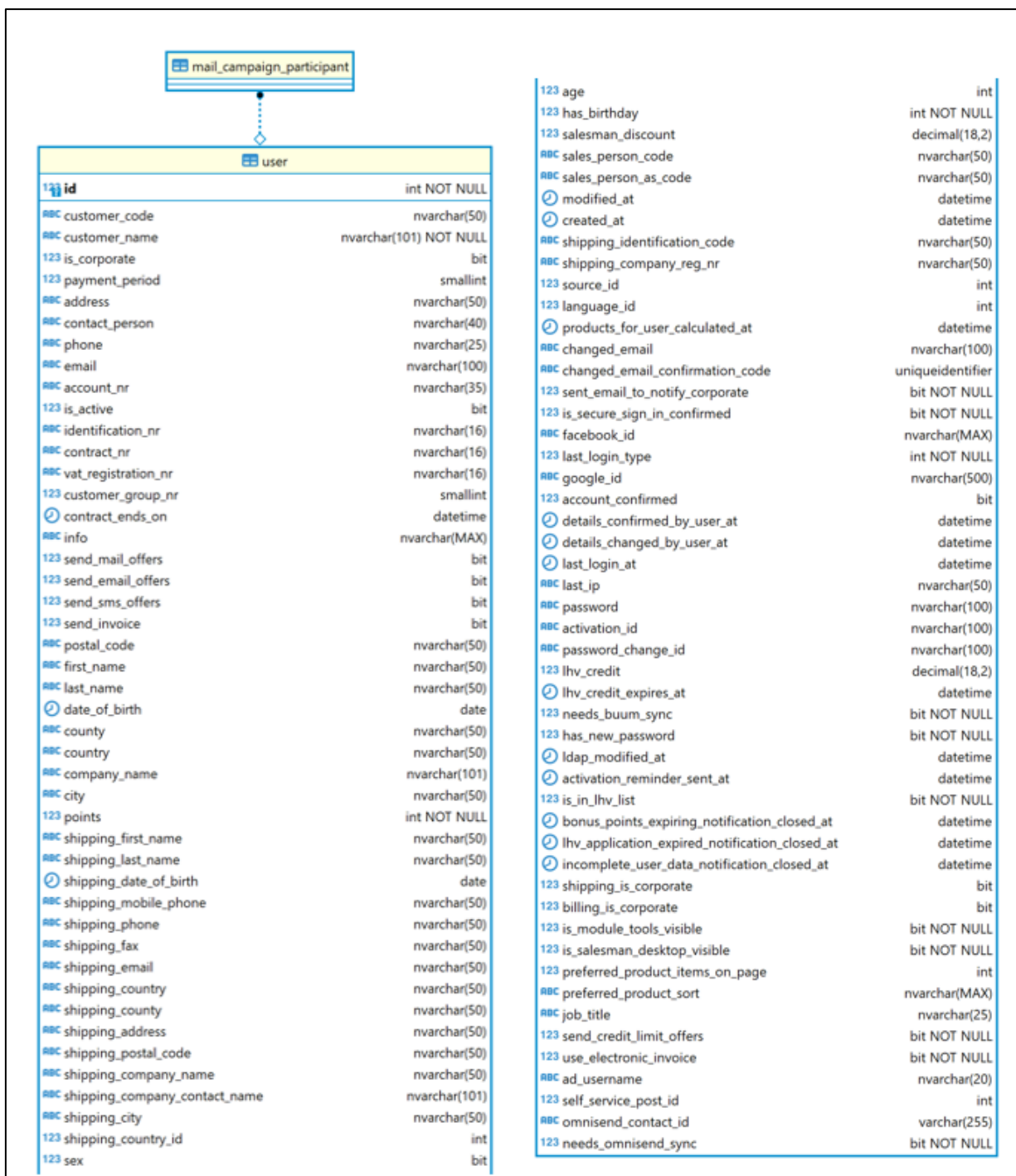
Sihtandmebaasis täideti andmesiirde tulemusena andmetega veerud neljas tabelis: *User*, *Order*, *Order_product* ja *Mail_campaign_participant*. Tabel 12 esitatakse sihtandmebaasi tabelite kirjeldused ja veergude arvud.

Tabel 12 Sihtandmebaasi tabelite üldinfo.

Tabeli nimi	Tabeli kirjeldus	Veergude arv
Mail_campaign_participant	Uudiskirjaga liitunud kliendid.	10
Order	E-poeest tehtud tellimused.	123
Order_product	Tellimuse read, kuna ühe tellimusega võib tellida mitu toodet.	40
User	Süsteemi kasutajad.	96

4.2.1 User tabel

E-poe kasutajate andmeid hoitakse sihtandmebaasis tabelis nimega *User*. Järgmiselt jooniselt on näha veergude andmetüüpe, väljapikkuseid ja kas *NULL* on veerus lubatud ehk kas sellele veerule vastavates väljades võib väärtus puududa või mitte (Joonis 8).



Joonis 8 Sihtandmebaasi tabel *User* ja selle seos *Mail_campaign_participant* tabeliga.

Tabel nimega *User* on suur, selles on 96 veergu. Tabel 13 esitab *User* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 13 Sihtandmebaasi tabeli *User* veergude kirjeldused.

Veeru nimi	Kirjeldus
id	Isiku numbriline identifikaator <i>User</i> tabelis. Primaarvõti.
customer_code	Kliendi identifikaator müügisüsteemis.
customer_name	Kliendi nimi. Kui kliendiks on juriidiline isik, siis tuleb veergu ettevõteteniimi, füüsilise isiku korral aga kliendi eesnimi ja perekonnanimi ühe sõnena. Veerus olevad andmed on sünkroonis müügisüsteemiga.
is_corporate	Kas kasutajaks on juriidiline isik?
payment_period	Kliendi makseperiood kuudes arvulise väärtusena
address	Kasutaja alalise elukoha aadress – näiteks tänav maja korter.
contact_person	Kontaktisiku eesnimi ja perekonnanimi ühe sõnena. Veerg täidetakse siis, kui kasutajaks on juriidiline isik.
phone	Isikuga ühenduse saamiseks kasutatav telefoninumber.
email	Isikuga ühenduse saamiseks kasutatav meiliaadress. Unikaalsete väärtustega veerg, väärtused on mõeldud kasutaja identifitseerimiseks.
account_nr	Kasutaja pangakonto number (<i>IBAN</i>).
is_active	Kas kliendi konto on aktiveeritud?
identification_nr	Kasutaja isikukood. Riigi poolt isikule väljaantud identifikaator, mis on väljaandva riigi piires unikaalne. Veerg võimaldab eristada samanimelisi inimesi.
contract_nr	Kliendilepingu number. Täidetakse ainult juriidiliste isikute korral.
vat_registration_nr	Käibemaksukohustuslase registreerimise (KMKR) number.
customer_group_nr	Kassasüsteemi kliendigrupi number.
contract_ends_on	Lepingu lõpukuupäev (ainult juriidiliste isikute jaoks).
info	Lisainfo väli. Vabateksti väli lisainfo sisestamiseks.
send_mail_offers	Kas pakkumiste saatmine kliendi postiaadressile on lubatud?
send_email_offers	Kas pakkumiste saatmine kliendi e-postiaadressile on lubatud?
send_sms_offers	Kas pakkumiste saatmine kliendile SMS'i teel on lubatud?
send_invoice	Kas arve saatmine on lubatud?

Veeru nimi	Kirjeldus
postal_code	Kasutaja alalise elukoha aadressi postiindeks.
first_name	Kasutaja eesnimi. Täidetakse siis, kui kasutajaks on füüsiline isik.
last_name	Kasutaja perekonnanimi. Täidetakse siis, kui kasutajaks on füüsiline isik.
date_of_birth	Kasutaja sünni kuupäev sünnikoha kohaliku aja järgi.
county	Kasutaja alalise elukoha aadressi maakond.
country	Kasutaja alalise elukoha aadressi riik.
company_name	Ettevõtte nimi. Lahter täidetakse ainult juriidiliste isikute korral.
city	Kasutaja alalise elukoha aadressi linn.
points	Boonuspunktid.
shipping_first_name	Kauba kättesaaja isiku eesnimi.
shipping_last_name	Kauba kättesaaja isiku perekonnanimi.
shipping_date_of_birth	Kauba kättesaaja isiku sünnikuupäev.
shipping_mobile_phone	Kauba kättesaajaga ühenduse saamiseks kasutatav mobiiltelefoni number.
shipping_phone	Kauba kättesaajaga ühenduse saamiseks kasutatav lauatelefoni number.
shipping_fax	Kauba kättesaajaga ühenduse saamiseks kasutatav faksinumber.
shipping_email	Kauba kättesaajaga ühenduse saamiseks kasutatav e-posti aadress.
shipping_country	Kauba kättesaamise asukoha aadressi riik.
shipping_county	Kauba kättesaamise asukoha aadressi maakond.
shipping_address	Kauba kättesaamise asukoha postiaadress.
shipping_postal_code	Kauba kättesaamise asukoha aadressi sihtnumber.
shipping_company_name	Kaubasaaja ettevõtte nimi.
shipping_company_contact_name	Kaubasaaja ettevõtte kontaktisik.
shipping_city	Kauba kättesaamise asukoha aadressi linn.

Veeru nimi	Kirjeldus
shipping_country_id	Kauba kättesaamise asukohta aadressi riigi numbriline identifikaator. Välisvõti. Viitab tabelile <i>dbo.Classificator_value</i> .
sex	Kas kasutaja on meessoost? Esitatud tõeväärtusena – True/1= mees, False/0= naine.
age	Kasutaja vanus aastates arvuna päringu teostamise hetkel. Virtuaalne veerg, milles olev väärtus arvutatakse siis, kui veeru kohta päring tehakse.
has_birthday	Kas kasutajal on sel kuul sünnipäev? Sisemiselt arvutatud väärtus, mis tagastab arvu, mille põhjal rakenduse koodis tuvastatakse, kas kasutajal on sel kuul sünnipäev. Veeru andmetüübiks on <i>INTEGER</i> ja väärtused selles on kas 1 (on sellel kuul sünnipäev) või 0 (ei ole sellel kuul sünnipäeva).
salesman_discount	Müügimehe maksimaalne allahindlus protsentides. Veergu kasutatakse omatöötajate puhul.
sales_person_code	Müügisüsteemis oleva müüja identifikaator, kes selle kliendi registreeris.
sales_person_as_code	Müüja identifitseerimise kood müügisüsteemis. Veerg täietakse siis, kui kasutaja on omatöötaja.
modified_at	Kuupäev ja kellaeg, millal tehti kasutajaandmetes viimane muudatus.
created_at	Kuupäev ja kellaeg, millal kasutaja süsteemis registreeriti.
shipping_identification_code	Tarne identifitseerimise kood. Tegemist on vana veeruga, enam seda ei kasutata.
shipping_company_reg_nr	Kaubasaaja ettevõtte registrikood. Seda veergu rakenduse koodis ei kasutata.
source_id	Kliendi kasutajaks registreerimise allika identifitseerimise kood. Kood väljendab seda, mille kaudu või kus klient kasutajaks registreeriti.
language_id	Kasutaja eelistatava kasutuskeele numbriline identifikaator. Välisvõti. Viide tabelile <i>dbo.Classificator_value</i> .
products_for_user_calculated_at	Kuupäev, millal protseduur <i>CalculateRecommendationsForUsers</i> arvutas kasutajale toote soovitusel tabelisse <i>Product_for_user</i> . Tegemist on vana veeruga, enam seda ei kasutata.

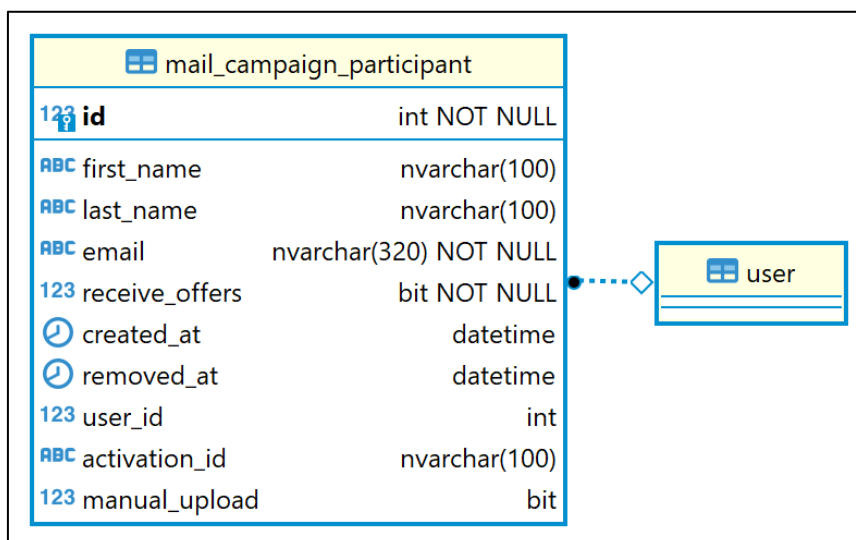
Veeru nimi	Kirjeldus
changed_email	Kasutaja e-posti aadressi vahetamisel kasutaja uus e-posti aadress.
changed_email_confirmation_code	Kasutaja e-postiaadressi vahetamisel kasutatav kinnituskood. Selle alusel kinnitatakse vana e-posti aadress ja siis saab uue aadressi sisestada.
is_secure_sign_in_confirmed	Kas turvaline sisselogimine on kinnitatud? Kasutatakse telefoniga sisse logimisel, kui klient endast isikukoodiga tuvastab.
facebook_id	Facebooki kasutaja identifitseerimise kood. Kasutatakse süsteemi sisselogimisel.
last_login_type	Mis tüüpi oli kasutaja viimane süsteemi sisenemine? Võimalikud väärtused on kirjeldatud loendustüübiga rakenduse lähtekoodis.
google_id	Google konto identifitseerimise kood.
account_confirmed	Kas kasutajakonto andmed on kinnitatud?
details_confirmed_by_user_at	Kuupäev ja kellaaeg, millal kasutaja kasutajaandmete üksikasjad kinnitas.
details_changed_by_user_at	Kuupäev ja kellaaeg, millal kasutaja viimati enda kasutajaandmeid muutis.
last_login_at	Kuupäev ja kellaaeg, millal toimus viimane sisselogimine e-poodi.
last_ip	Viimase sisselogimise IP aadress.
password	Parool ehk salasõna, mida kasutatakse isikusamasuse tõendamiseks või juurdepääsu saamiseks.
activation_id	Kasutaja konto aktiveerimisel kasutatav e-postiga registreerimise identifitseerimise kood. See kood on süsteemi poolt genereeritud <i>GUID (Globally Unique Identifier)</i> kood. Veerg täidetakse kasutajakonto loomisel kui klient registreerib end e-posti aadressiga.
password_change_id	Parooli vahetamisel kasutaja e-posti aadressile saadetakse identifitseerimise kood. See kood on süsteemi poolt genereeritud <i>GUID (Globally Unique Identifier)</i> kood. Veergu kasutatakse siis, kui kasutaja ei mäleta e-poodi sisenemiseks enda parooli ja ta soovib endale uut parooli salvestada.
lhv_credit	LHV (konkreetse ettevõtte) krediidimäär. Väärtust hoitakse kümnendarvuna.

Veeru nimi	Kirjeldus
lhv_credit_expires_at	Kuupäev ja kellaaeg, millal aegub kliendile tehtud LHV järelmaksu pakkumine. Veerg on seotud kampaaniatega ja turundajate poolt kliendile saadetavate pakkumistega.
needs_buum_sync	Kas kliendi andmeid on vaja müügisüsteemiga sünkroniseerida?
has_new_password	Kas kasutajal on uus parool?
ldap_modified_at	Kuupäev ja kellaaeg, millal ID-kaardilt isikuandmeid küsitakse (ID-kaardiga sisselogimisel). Protokoll, mille kaudu ID-kaardi sertifikaatidele ligipääsetakse.
activation_reminder_sent_at	Kuupäev ja kellaaeg, millal saadeti aktiveerimise meeldetuletus.
is_in_lhv_list	Kas klient on LHV poolt saadetud nimekirjas, st kas ta on LHV klient?
bonus_points_expiring_notification_closed_at	Kuupäev ja kellaaeg, millal suleti teavitus boonuspunktide aegumisest.
lhv_application_expired_notification_closed_at	Kuupäev ja kellaaeg, millal LHV avalduse vorm kliendi poolt suleti.
incomplete_user_data_notification_closed_at	Kuupäev ja kellaaeg, millal suleti teavitus, et kasutaja andmed on poolikud.
shipping_is_corporate	Kas kaubasaaja on juriidiline isik?
billing_is_corporate	Kas arve tasub juriidiline isik?
is_module_tools_visible	Kas näidata veebilehele ilmuvas kasutajaliideses poe töötajatele riba, mille kaudu klientide eest oste sooritada? Tegemist on vana veeruga, enam seda ei kasutata.
is_salesman_desktop_visible	Kas kasutaja arvuti ekraanil on programm nimega müüja töölaud (<i>Salesman desktop</i>) nähtav?
preferred_product_items_on_page	Mitu toodet kuvatakse korraga lehel? Milline on kasutaja eelistus, mitu toodet ta tahab, et talle e-poes korraga kuvatakse?
preferred_product_sort	Eelistatud toodete sorteerimine. Mille järgi soovib klient oma tooteid sorteerida – nt kas hinna, boonuse või millegi muu järgi? Tegemist on vana veeruga, enam seda ei kasutata.
job_title	Töötaja ametinimetus. Väärtus registreeritakse siis, kui kasutaja on oma töötaja.
send_credit_limit_offers	Kas kliendile võib saata järelmaksu limiidi pakkumisi?

Veeru nimi	Kirjeldus
use_electronic_invoice	Kas kasutatakse elektroonilist arvet?
ad_username	<i>Active directory</i> (lühidalt AD) kasutajanimi/ kasutajatunnus omatöötajatel. Väärtus registreeritakse siis, kui kasutaja on omatöötaja.
self_service_post_id	Iseteeninduse eelistatava pakiautomaadi kood (nt kas omniva või dpd). Viitab tabelile <i>Self_service_post</i> . Välisvõti.
omnisend_contact_id	Kliendi identifikaator turundustööriistas.
needs_omnisend_sync	Kas kliendiandmed vajavad turundustööriistaga sünkroniseerimist?

4.2.2 Mail_campaign_participant tabel

Tabelis *Mail_campaign_participant* hoitakse uudiskirjaga liitunud klientide andmeid. Selle tabeli andmeid kasutatakse turunduslike eesmärkida saavutamiseks. Tabel on väike (Joonis 9), selles on 10 veergu.



Joonis 9 Sihtandmebaasi tabel *Mail_campaign_participant* ja sellega seotud tabel.

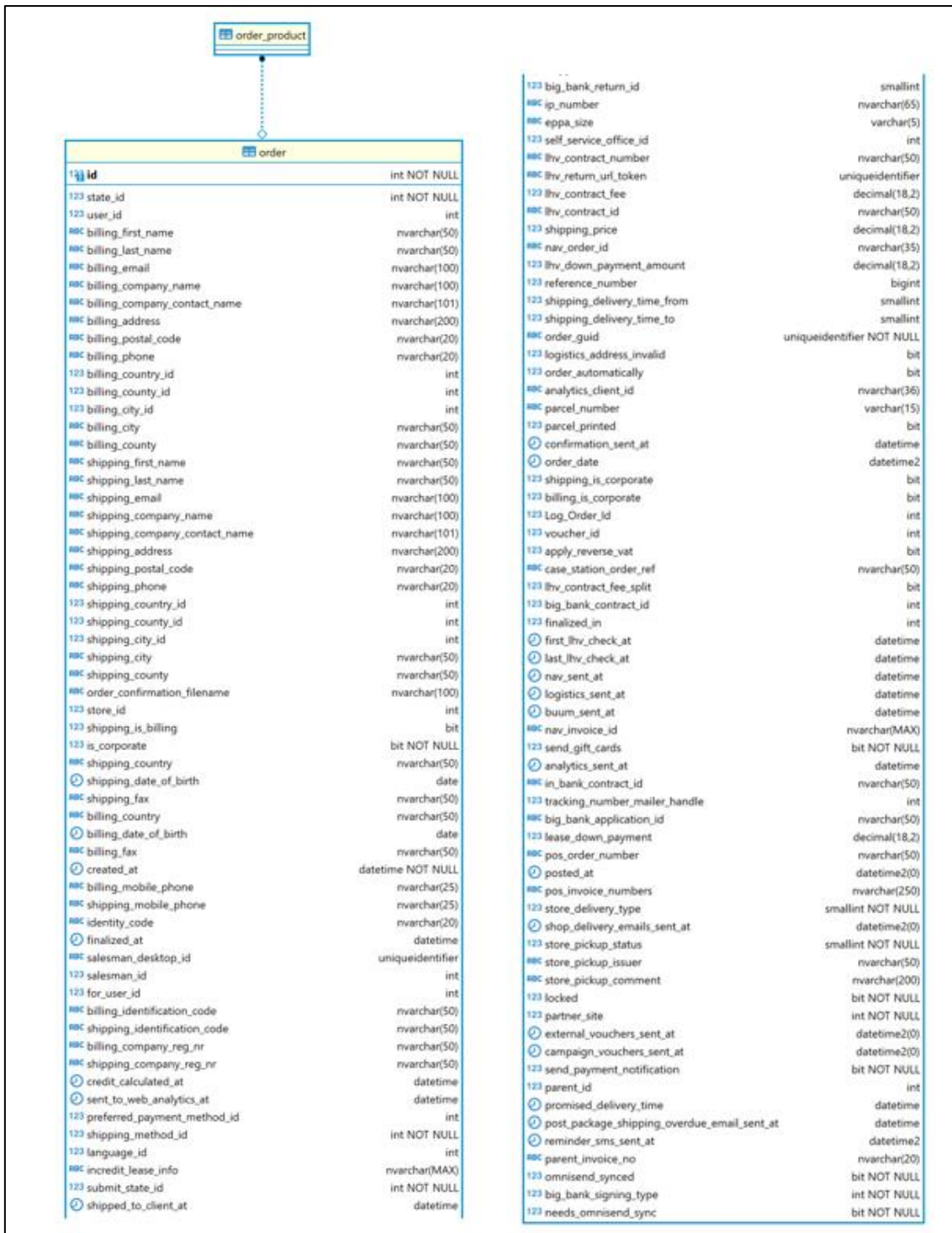
Tabel 14 esitab *Mail_campaign_participant* tabeli veergude nimetused koos kirjeldusega.

Tabel 14 Sihtandmebaasi tabeli *Mail_campaign_participant* veergude kirjeldused.

Veeru nimi	Kirjeldus
id	Uudiskirjaga liitunud klientide identifikaator. Selles tabelis on kõik kasutajad, nii registreeritud kui ka anonüümsed, kes on tellinud endale uudiskirjad. Primaarvõti.
first_name	Kliendi eesnimi.
last_name	Kliendi perekonnanimi.
email	Kliendiga ühenduse saamise eesmärgil kasutatav e-posti aadress.
receive_offers	Kas klient on nõus saama e-postiga turunduslike pakkumisi?
created_at	Kuupäev ja kellaeg, millal klient tellis/nõustus saada uudiskirja.
removed_at	Kuupäev ja kellaeg, millal klient loobus uudiskirja saamisest.
user_id	Kasutaja identifikaator. Viitab tabelile <i>user</i> . Välisvõti.
activation_id	Uudiskirja aktiveerimisel kasutatav identifitseerimise kood. Sellega valideeritakse kasutaja e-posti aadress.
manual_upload	Kas kliendi andmed imporditi uudiskirja tabelisse Exceli failist? Süsteem võimaldab tagakontoris Exceli faili kaudu andmete üleslaadimist ja lisamist selle tabelisse.

4.2.3 Order tabel

Tabelis *Order* hoitakse e-poe tellimustega seotud andmeid. Tabel on suur, selles on 123 veergu (Joonis 10).



Joonis 10 Sihtandmebaasi order tabel

Tabel 15 esitab *order* tabeli veergude nimetused koos kirjeldusega.

Tabel 15 Sihtandmebaasi tabeli *order* veergude kirjeldused

Veeru nimi	Kirjeldus
id	Tellimuse numbriline identifikaator. Primaarvõti.
state_id	Tellimuse seisundi ehk oleku numbriline identifikaator.
user_id	Kasutaja numbriline identifikaator. Viide tabelile <i>dbo.User</i> .
billing_first_name	Arve saaja eesnimi.
billing_last_name	Arve saaja perekonnanimi.
billing_email	Arve saajaga ühenduse saamiseks kasutatav e-posti aadress.
billing_company_name	Kui arve saaja on juriidiline isik, siis tema äriregistris olev nimi.
billing_company_contact_name	Kui arve saaja on juriidiline isik, siis tema kontaktisiku eesnimi ja perekonnanimi.
billing_address	Arve saaja elukoha aadress. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadress.
billing_postal_code	Arve saaja elukoha aadressi postiindeks. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi postiindeks.
billing_phone	Arve saajaga ühenduse saamiseks kasutatav telefoninumber.
billing_country_id	Arve saaja riigi numbriline identifikaator. Viitab tabelile <i>dbo.Classificator_value</i> . Välisvõti.
billing_county_id	Arve saaja maakonna numbriline identifikaator. Viitab tabelile <i>dbo.Classificator_value</i> . Välisvõti.
billing_city_id	Arve saaja linna numbriline identifikaator. Viitab tabelile <i>dbo.Classificator_value</i> . Välisvõti.
billing_city	Arve saaja elukoha aadressi linn. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi linn.
billing_county	Arve saaja elukoha aadressi maakond. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi maakonda.
shipping_first_name	Kauba kättesaaja eesnimi.
shipping_last_name	Kauba kättesaaja perekonnanimi.

Veeru nimi	Kirjeldus
shipping_email	Kauba kättesaajaga ühenduse saamiseks kasutatav e-posti aadress
shipping_company_name	Kui kaubasaaja on juriidiline isik, siis tema äriregistris olev nimi
shipping_company_contact_name	Kui kaubasaaja on juriidiline isik, siis ettevõtte poolse kontaktisiku eesnimi ja perekonnanimi.
shipping_address	Kauba kättesaamise aadress. Tarne aadress.
shipping_postal_code	Kauba kättesaamise aadressi postindeks.
shipping_phone	Kauba kättesaajaga ühenduse saamiseks kasutatav telefoninumber.
shipping_country_id	Kauba kättesaamise aadressi riigi numbriline identifikaator. Viitab tabelile <i>classifier_value</i> . Välisvõti.
shipping_county_id	Kauba kättesaamise aadressi maakonna identifikaator. Viitab tabelile <i>classifier_value</i> . Välisvõti.
shipping_city_id	Kauba kättesaamise aadressi linna numbriline identifikaator. Viitab tabelile <i>classifier_value</i> . Välisvõti.
shipping_city	Kauba kättesaamise aadressi linn.
shipping_county	Kauba kättesaamise aadressi maakond.
order_confirmation_filename	Tellimuse kinnituse PDF faili nimi. Tellimuse kinnitusel saadetakse kliendile e-post, milles on ka PDF fail kaasas.
store_id	Kaupluse numbriline identifikaator. Kauba tellimisel poodi pannakse sellesse välja kaupluse number, kuhu kaup telliti. Viide <i>dbo.Store</i> tabelile. Välisvõti.
shipping_is_billing	Kas kauba kättesaamise aadress on sama, mis arve saaja elukoha aadress?
is_corporate	Kas klient on juriidiline isik?
shipping_country	Kauba kättesaamise aadressi riik.
shipping_date_of_birth	Kauba kättesaaja sünnikuupäev.
shipping_fax	Kauba kättesaajaga ühenduse saamiseks kasutatav faksi number.
billing_country	Arve saaja elukoha aadressi riik. Juriidilise isiku korral kirjutatakse tema äriregistris registreeritud aadressi riik.
billing_date_of_birth	Arve saaja sünnikuupäev.

Veeru nimi	Kirjeldus
billing_fax	Arve saajaga ühenduse saamiseks kasutatav faksi number.
created_at	Tellimuse loomise kuupäev ja kellaaeg. Tellimus luuakse siis, kui lähed ostukorvist edasi lehele „Vormista tellimus“. Vajalik selleks, et kui ostuprotsess jäi poolikuks, siis saab kliendile meeldetuletuse saata.
billing_mobile_phone	Arve saajaga ühenduse saamiseks kasutatav mobiiltelefoni number.
shipping_mobile_phone	Kauba saajaga ühenduse saamiseks kasutatav mobiiltelefoni number.
identity_code	Kasutaja isikukood. Veerg peab olema täidetud, kui tellimuse tegi kasutaja, kes valideeris ennast sisselogimisel kas ID-kaardiga või Smart-ID'ga. Hetkel seda veergu veel ei kasutata.
finalized_at	Ostuprotsessi lõppemise kuupäev ja kellaaeg. Registreeritakse pärast maksmist või tellimuse kinnitamist.
salesman_desktop_id	Tagakontori ehk <i>salesman_desktop_id</i> . <i>Salesman'is</i> võis töötaja valida tarne, mida kliendil ei ole võimalik veebis ise valida. Sellisel juhul pannakse juurde <i>salesman_desktop_id</i> ning siis sellest lähtuvalt arvutatakse või ei arvutata tarne hind. See, kas arvutatakse või mitte, sõltub olukorrast. Näiteks kui enne sooviti kaupa kätte saada pakiautomaadist, aga nüüd poest, siis ei tule mingit tasu juurde.
salesman_id	Klienditeenindaja/müüja kui e-poe kasutaja numbriline identifikaator (<i>user_id</i>). Viide <i>User</i> tabelile. Kliendid ja omatöötajad on kõik e-poe kasutajatena <i>User</i> tabelis. Välisvõti.
for_user_id	Veeru selgitus ei ole teada, kuna dokumentatsioon puudub. Tegemist on vana veeruga, enam seda ei kasutata. Eesti tabelis on selles veerus kõikjal <i>NULL</i> . Välisvõti.
billing_identification_code	Arve saaja isikukood, mis on riigi poolt kodanikele ja alaliselt riigis elavatele isikutele antav ainulaadne märkide kombinatsioon.
shipping_identification_code	Kaubasaaja isikukood.
billing_company_reg_nr	Kui arve saaja on juriidiline isik, siis tema äriregistri registreerimise number.
shipping_company_reg_nr	Kui kaubasaaja on juriidiline isik, siis tema äriregistri registreerimise number .

Veeru nimi	Kirjeldus
credit_calculated_at	Krediidi arvutamise kuupäev ja kellaaeg.
sent_to_web_analytics_at	Kuupäev ja kellaaeg, millal tellimus saadeti <i>Google Analytics</i> 'sse. Suurim kuupäev selles veerus on 9.01.2015. Tegemist on vana veeruga, enam seda ei kasutata.
preferred_payment_method_id	Eelistatud maksemeetodi identifikaator. Välisvõti.
shipping_method_id	Tarne meetodi identifitseerimise kood. Võimalikud väärtused on kirjeldatud loendustüübiga rakenduse lähtekoodis.
language_id	Mis keeles on tellimus tehtud? Keele numbriline identifikaator. Veeru andmetüüp on <i>INTEGER</i> . Võimalikud väärtused on kirjeldatud loendustüübiga rakenduse lähtekoodis.
incredit_lease_info	Liisingut osutava ettevõtte poolt saadetud päringu vastus. Vastus salvestatakse <i>JSON</i> formaadis.
submit_state_id	Esitatud tellimuse seisundi ehk oleku numbriline identifikaator.
shipped_to_client_at	Kuupäev ja kellaaeg, millal kaup kliendile välja saadeti.
big_bank_return_id	Big Banki tagastatav numbriline identifikaator
ip_number	Kliendi arvuti IP aadress. Tegemist on vana veeruga, enam seda ei kasutata.
eppa_size	Eeldatav pakiautomaadi kapi suurus. Väline teenus ütleb, mis suurusega pakki tellimus mahutada – kas näiteks S, M, L või XL.
self_service_office_id	Pakiautomaadi identifikaator. Veerus <i>shipping_address_id</i> olev väärtus loob seose <i>order_shippingaddress</i> tabelis oleva reaga ja seal on kirjas, milline paikaautomaat täpselt. Välisvõti.
lhv_contract_number	LHV poolt e-poele saadetud lepingu number.
lhv_return_url_token	E-poe poolt genereeritud salavõti, millega saab LHV hiljem päringu vastuse tagasi e-poele saata
lhv_contract_fee	LHV lepingutasu.
lhv_contract_id	LHV lepingu identifikaator.
shipping_price	Tarne hind koos käibemaksuga.
nav_order_id	Raamatupidamissüsteemi tellimuse identifikaator.
lhv_down_payment_amount	LHV sissemaksu summa.

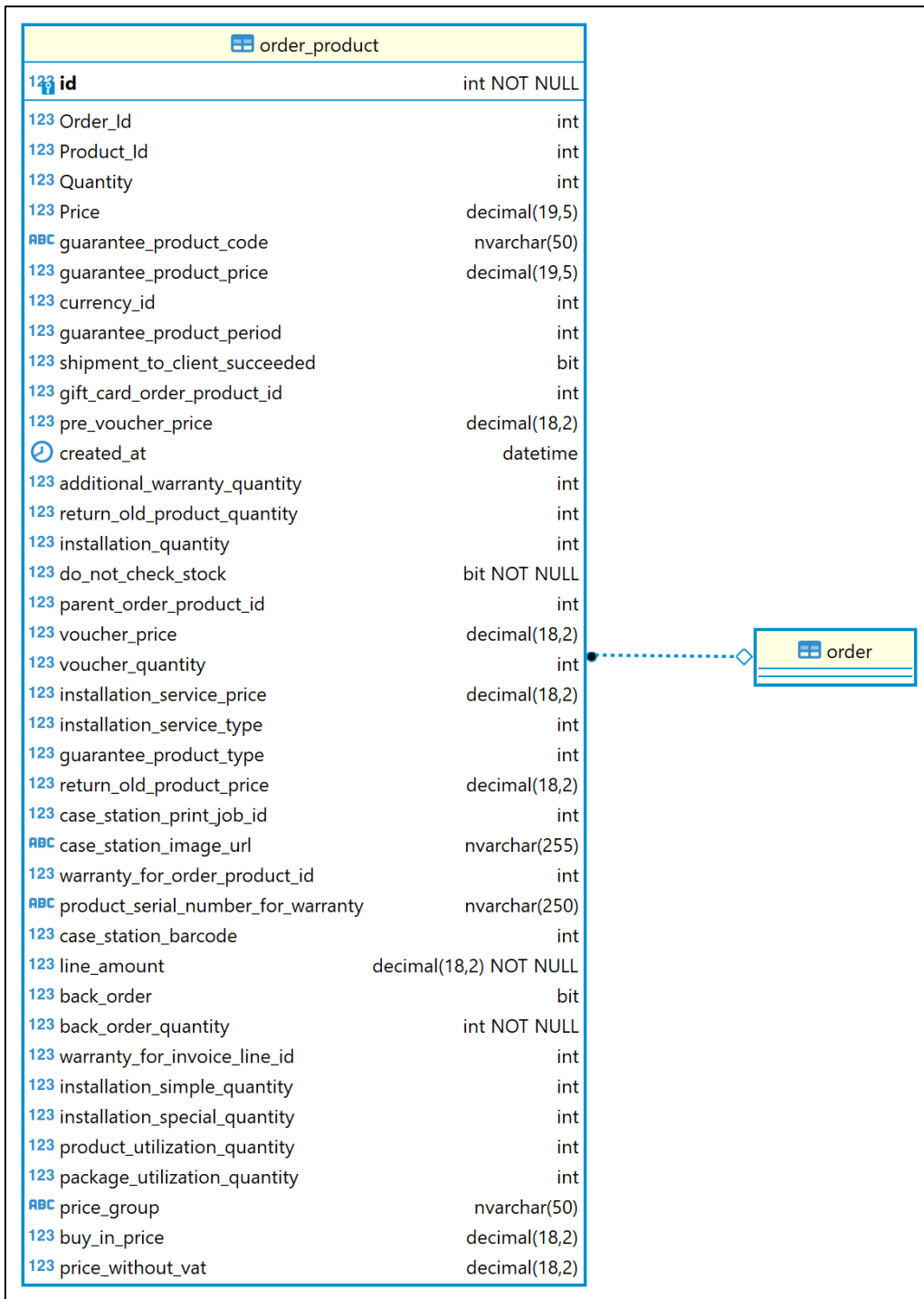
Veeru nimi	Kirjeldus
reference_number	Makse viitenumber. See võib olla kas järelmaksu viitenumber või pangalingi viitenumber.
shipping_delivery_time_from	Minimaalne eeldatav tarneaeg päevades.
shipping_delivery_time_to	Maksimaalne eeldatav tarneaeg päevades.
order_guid	Tellimuse identifikaator. Automaatselt genereeritav.
logistics_address_invalid	Kas logistika süsteemis olev aadress on kehtetu?
order_automatically	Kas kaup tellitakse sihtkohta automaatselt?
analytics_client_id	<i>Google Analytics</i> kliendi identifikaator.
parcel_number	Paki number. Tegemist on vana veeruga, enam seda ei kasutata.
parcel_printed	Kas pakikleebis on prinditud?
confirmation_sent_at	Kuupäev ja kellaeg, millal saadeti ostu kinnituse e-kiri.
order_date	Tellimuse kuupäev. Pannakse esimese makse laekumisel või sünkroniseerimisel.
shipping_is_corporate	Kas tarnija on juriidiline isik?
billing_is_corporate	Kas arve saaja on juriidiline isik?
Log_Order_Id	Logistika süsteemis tellimuse numbriline identifikaator.
voucher_id	Kinkekaardiga tasumisel, kinkekaardi numbriline identifikaator. Välisvõti.
apply_reverse_vat	Kas kliendist juriidilisele isikule rakendub pöördkäibemaks?
case_station_order_ref	Mingi müügiallikas, mida enam pole.
lhv_contract_fee_split	Kas LHV lepingutasu poolitamine on lubatud?
big_bank_contract_id	Big Bank'i lepingu number.
finalized_in	Numbriline viide selle kohta, kus kohas tellimus lõpule viidi. Näited on veebis, telefonis ja tagakontor (<i>backoffice</i>). Võimalikud väärtused on kirjeldatud loendustüübiga rakenduse lähtekoodis.
first_lhv_check_at	Kuupäev ja kellaeg, millal käidi esimest korda LHV's kontrollimas, kas leping on allkirjastatud.
last_lhv_check_at	Kuupäev ja kellaeg, millal käidi viimati LHV süsteemis kontrollimas, kas leping on allkirjastatud.

Veeru nimi	Kirjeldus
nav_sent_at	Kuupäev ja kellaeg, millal saadeti viimati tellimuse andmed raamatupidamissüsteemi.
logistics_sent_at	Kuupäev ja kellaeg, millal saadeti viimati tellimuse andmed logistikasüsteemi.
buum_sent_at	Kuupäev ja kellaeg, millal viimati saadeti tellimuse andmed müügisüsteemi.
nav_invoice_id	Raamatupidamissüsteemi arve identifikaator.
send_gift_cards	Kas kinkekaartide saatmine on lubatud? Kui tellitakse mingi toode, millel on kinkekaart kaasas, siis vaadatakse, kas see väärtus on 1 ja kui on, siis saadetakse kliendile kinkekaardiga seotud toode.
analytics_sent_at	Kuupäev ja kellaeg, millal tellimus saadeti <i>Google Analytics</i> 'sse?
in_bank_contract_id	Inbanki lepingu identifikaator.
tracking_number_mailer_handle	Saadetise jälgimise e-posti Maileri identifikaator. Mailer on e-posti teenus.
big_bank_application_id	Big Bank'i finantseerimise id (laenu/järelmaksu taotluse id).
lease_down_payment	Liisingu sissemakse suurus.
pos_order_number	Tellimuse number kassasüsteemis (<i>point of sale</i>).
posted_at	Tellimuse arve konteerimise aeg.
pos_invoice_numbers	Arvete numbrid kassasüsteemis (<i>point of sale</i>).
store_delivery_type	Poodi tarne viis. Valikus on kolm varianti: <i>none</i> , poodi või <i>drive-in</i> . Võimalikud väärtused on kirjeldatud loendustüübiga rakenduse lähtekoodis.
shop_delivery_emails_sent_at	Kuupäev ja kellaeg, millal saadeti kiri selle kohta, et toode telliti poodi ja kas toode on poes olemas.
store_pickup_status	Kauba poest kättesaamise seisund ehk olek.
store_pickup_issuer	Poest kauba väljaandja nimi vaba tekstiväljana. See isik, kes poest kauba kliendile kätte andis.
Store_pickup_comment	Kommentaari poest kauba väljastamise kohta.
Locked	Kas tellimus on lukustatud? Tellimus lukustatakse siis, kui kasutaja läheb maksma.

Veeru nimi	Kirjeldus
Partner_site	SEB grupi leht, millelt algatati tellimus meie süsteemis. Seda veergu ei kasutata.
External_vouchers_sent_at	Partnerite loodud kupongide (<i>voucher</i>) väljasaatmise aeg.
Campaign_vouchers_sent_at	Kampaania kupongide (<i>voucher</i>) väljasaatmise aeg.
Send_payment_notification	Kas saata tasumise teavitust?
Parent_id	Seotud tellimus eraldi transpordile, toote või teenuse müügi jaoks. Välisvõti.
Promised_delivery_time	Kuupäev ja kellaeg, millal on tellimuse tarne tähtaeg.
Post_package_shipping_overdue_email_sent_at	Pakiautomaadi tarne hilinemise teavituse saatmise aeg.
Reminder_sms_sent_at	Millal saadeti tellimuse kohale jõudmise meeldetuletuse lühisõnum ehk SMS?
Parent_invoice_no	Seotud arve eraldi transpordi, toote või teenuse müügi jaoks.
Omnisend_synced	Kas tellimus on turundussüsteemiga sünkroniseeritud?
Big_bank_signing_type	BIG Bank'i sisselogimise tüüp.
Needs_omnisend_sync	Kas tellimus vajab turundussüsteemiga sünkroniseerimist? Sünkroniseerimised toimuvad teatud aja tagant. Kui väärtus on 1, siis lisatakse tellimus järgmisesse sünkroniseerimisse.

4.2.4 Order_product tabel

Tabelis *Order_product* hoitakse tellimusreaga seotud andmeid. Joonis 11 on näha tabeli *Order_product* veergude andmetüüpe, väljapikkuseid ja kas *NULL* on veerus lubatud ehk kas sellele veerule vastavates väljades võib väärtus puududa või mitte.



Joonis 11 Sihtandmebaasi tabel *Order_product* ja sellega seotud tabel

Order_product tabelis on 40 veergu. Tabel 16 esitab *Order_product* tabeli veergude nimetused koos nendes hoitavate andmete tähenduse vabatekstilise kirjeldusega.

Tabel 16 Sihtandmebaasi tabeli *Order_product* veergude kirjeldused.

Veeru nimi	Kirjeldus
id	Tellimusrea numbriline identifikaator. Unikaalne väärtus <i>order_product</i> tabelis. Primaarvõti.
Order_Id	Tellimuse numbriline identifikaator. Välisvõti. Viitab sihtandmebaasis tabelile <i>dbo.Order</i> .
Product_Id	Toote numbriline identifikaator. Välisvõti. Viitab sihtandmebaasis tabelile <i>dbo.Product</i> .
Quantity	Tellimuses oleva toote kogus tükkides.
Price	Toote hind koos käibemaksuga tellimuse sooritamise hetkel.
guarantee_product_code	Toote garantii kood. Viitab tabeli <i>Product</i> , veerule <i>code</i> . Tegemist on vana veeruga, enam seda ei kasutata.
guarantee_product_price	Toote garantii maksumus toote ostmise hetkel. Tegemist on vana veeruga, enam seda ei kasutata.
currency_id	Valuuta identifikaator. Rakenduses defineeritud <i>Currency</i> loendustüübis on kaks võimalikku väärtust 4 = EUR ja 2 = bonus points.
guarantee_product_period	Tootele eraldi juurde müüdava lisagarantii kehtivuse pikkus aastates.
shipment_to_client_succeeded	Näitab, kas kaupa saab kliendile välja saata. Näitab, kas rida on täielikult raamatupidamissüsteemist tellitud. Kui see on 0, siis pakki teele ei panda.
gift_card_order_product_id	Kinkekaardi identifikaator. Välisvõti. Viitab tabelile <i>Gift_card_order_product</i> . Kui see kinkekaart on kingitus, siis on seal tabelis kirjas kinkekaardi saaja ja sõnum jne. Kinkekaardid on ka <i>Product</i> tabelis. Välisvõti.
pre_voucher_price	Toote hind koos käibemaksuga enne kupongi kasutust.
created_at	Loomise kuupäev. See hetk, kui toode tellimusele külge pannakse.
additional_warranty_quantity	Lisagarantiikogus tükkides. Näitab, mitmele tootele võeti lisagarantiid.
return_old_product_quantity	Tagastatud vana toote kogus. Tegemist on vana veeruga, enam seda ei kasutata.
installation_quantity	Paigaldusteenuse kogus tükkides.
do_not_check_stock	Ära vaata laoseisu. Nt kinkekaartidel pole vaja laoseisu vaadata, neid jagub lõputult.

Veeru nimi	Kirjeldus
parent_order_product_id	Toode, mis sõltub teisest tootest. Näited on kampaania lisatooted ja kingitused. Välisvõti.
voucher_price	Toote hind koos käibemaksuga pärast sooduskupongi rakendamist.
voucher_quantity	Mitu toodet saadi sooduskupongi olemasolul soodsamalt?
installation_service_price	Paigaldusteenuse hind koos käibemaksuga.
installation_service_type	Paigaldusteenuse tüüp.
guarantee_product_type	Toote garantii jaoks. Toote tüüp. Vana veerg.
return_old_product_price	Tagastatud vana toote äraveo hind koos käibemaksuga.
case_station_print_job_id	Telefoni kesta kavandamiseks kasutatava printimise töö identifitseerimisekood.
case_station_image_url	Telefoni kesta kavandamiseks kasutatava pildi URL (<i>Uniform Resource Locator</i>) aadress.
warranty_for_order_product_id	Viide veebis tehtud tellimuse reale, millele lisagarantii juurde müüdi. Välisvõti.
product_serial_number_for_warranty	Toote seerianumber garantii jaoks.
case_station_barcode	Triipkood, mille alusel maksta telefoni kesta kavandamise eest iseteeninduse kioskis. Tegemist on vana veeruga, enam seda ei kasutata.
Line_amount	Arveridade arv. Vana veerg, enam seda ei kasutata.
Back_order	Kas toode on tellimisel? Toodet hetkel pole, aga kui tellida, siis see tuleb paari päevaga kohale. Kaupa pole hetkel olemas, aga seda saab ette tellida, kuna on teada, et see tuleb lähiajal.
back_order_quantity	Tellitud toodete kogus tükiarvuna.
warranty_for_invoice_line_id	Viide müügisüsteemis tehtud tellimuse reale, millel lisagarantii juurde müüdi. Välisvõti.
installation_simple_quantity	Lihtpaigalduse kogus tükiarvuna. Lihtpaigaldust teostatakse hetkel ainult Eestis.
installation_special_quantity	Eripaigalduse kogus tükiarvuna. Eripaigaldust teostatakse hetkel ainult Eestis.
product_utilization_quantity	Vana toote utiliseerimise kogus tükiarvuna. Vana toodet utiliseeritakse hetkel ainult Eestis.

Veeru nimi	Kirjeldus
package_utilization_quantity	Pakendi utiliseerimise kogus tükiarvuna. Vana toodet utilitseeritakse hetkel ainult Eestis.
price_group	Maaletooja hinnagrupi nimi ostu hetkel. Näited on kampaania hinnagrupp või sünnipäeva hinnagrupp.
buy_in_price	Toote sisseostu hind ilma käibemaksuta.
price_without_vat	Toote käibemaksuta hind ostu hetkel.

4.3 Andmesiire

Andmesiire viidi läbi vastavalt peatükis 3 kirjeldatud meetodile. Selles jaotises kirjutatakse täpsemalt kõigist andmesiire etappidest. Samuti esitatakse erinevad arvulised näitajad, mis iseloomustavad loodud lahendust.

4.3.1 Planeerimine

Ettevõtte määras töö skoobi ja pani paika ka esialgse töö tähtaja, milleks oli 2022. aasta maikuu lõpp. Ettevõtte poolt tehti testandmebaasidest varukoopiad ja anti need mälu pulgal autorile.

PostgreSQL testandmebaas oli tööandmebaasi üsna lähedane koopia, aga isikute/organisatsioonide nimed, kontaktid jms oli muudetud, et nende alusel ei saaks isikuid ja organisatsioone tuvastada. Andmete hulk selles andmebaasis oli üsna sarnane tegelikult ülekandmist vajavale andmete hulga. MS SQL Serveri testandmebaasis oli vähe andmeid ja ka need ei olnud tegelike isikute ja organisatsioonide kohta.

Autor valis andmesiirde läbiviimise viisiks suure pauguga andmete üleviimise, et andmesiire oleks tehtav ühekorraga ja lühikese ajaga vältel. Valiku põhjuseks oli ettevõtte soov teha autori koostatud lausete abil andmesiire ka tööandmebaasis. Töö teostajaks määrati autor ja ettevõtte poolt määrati autorile ka tarkvara arendaja, kellega koostööd teha.

Selles etapis valiti välja ka lähenemise viis ja teostamise vahendid. Vahendite valimisest kirjutati põhjalikumalt jaotises 3.4. Tööks kasutas autor *SQL Server Management Studio* (SSMS) rakendust. Andmesiire plaaniti realiseerida autori poolt kirjutatud skriptidena,

mis käivitatakse SMSS programmis. Selle programmiga saab siduda nii PostgreSQL kui ka MS SQL Serveri andmebaasi ja andmed ühest andmebaasist teise üle tõsta.

4.3.2 Andmete ettevalmistamine

Selles etapis pandi paika, et andmesiirde laused kirjutatakse autori kohalikus masinas. Lepiti kokku, et kui autoril andmesiire katsetus õnnestub, siis tehakse samasugune katsetus ka ettevõtte arendaja arendusmasinas. Kui ka seal peaks andmesiire õnnestuma, siis järgmine samm oleks andmesiire PostgreSQL tööandmebaasist MS SQL Serveri tööandmebaasi. Juurdepääsuõiguseid antud töö puhul ei jagatud, kuna töö tegi autor ise, kohalikus masinas talle antud varukoopiate andmebaaside põhjal. Varukoopiaid testandmebaasidest selles etapis ei tehtud, kuna need olid ettevõtte poolt juba varem tehtud ja autorile üle antud.

Selles etapis pandi üles ka andmesiirde tegemise testkeskkond, milles autor kasutas ettevõtte arendaja abi. Kõigepealt installeeriti autori arvutisse MS SQL Server Express väljaanne, SSMS ja PostgreSQL. Seejärel ühendati SSMS MS SQL Serveriga. SSMS ja MS SQL Serveri omavaheliseks autentimise meetodiks oli Windows Authentication. Kuna MS SQL Server installeeriti sama kasutajaga, millega SSMS'is ühendatakse MS SQL Serveri külge, siis oli lõputöö autoril ligipääsuõigused juba olemas. Siis imporditi SSMS'iga andmebaasi varukoopia MS SQL Serverisse. Seejärel loodi käsurea tööriistaga psql tühi PostgreSQL andmebaas. Käsurea tööriistaga imporditi värskelt loodud tühja andmebaasi ettevõtte poolt tehtud PostgreSQL andmebaasi varukoopia.

4.3.3 Andmesiirde kavandamine

Kui andmebaasid olid tööle saadud, siis hakati uurima sihtandmebaasi tabelite sisu. Seejärel uuriti koostöös ettevõttepoolse arendajaga ülekantavaid andmeid. Sihtandmebaasis oli vaja kanda andmed järgmistesse tabelitesse: *User*, *Order*, *Order_product*, ja *Mail_campaign_participant*.

Andmebaaside kohta puudus dokumentatsioon, info andmebaasi veergude tähenduse kohta saadi rakenduse koodist ja koodis olevatest kommentaaridest. Kuna autoril puudus juurdepääs rakenduse koodile, siis tehti koostööd ettevõttepoolse arendajaga. Kui veeru väärtused tulid rakenduse lähtekoodis kasutatavast loendustüübist (*ENUM*), siis need olid ainukesed osad, mida autor rakenduse koodist nägi. Kõigepealt uuriti andmebaaside

tabelite sisu ja koostati veergude definitsioonid. Need on esitatud jaotistes 4.1 ja 4.2. Seejärel vaadati üle sihtandmebaasi veerud. Koostöös ettevõttega valiti välja kõige vajalikumad sihtandmebaasi veerud ja leiti neile veergudele lähteandmebaasist vastavused. Andmed sihtandmebaasi vajalikesse veergudesse leiti lähteandmebaasi järgnevatest tabelitest.

- *Accounts_users*
- *Address_useraddress*
- *Order_billingaddress*
- *Order_order*
- *Order_orderline*
- *Order_ordernote*
- *Order_paymentevent*
- *Order_shippingaddress*
- *Social_auth_usersocialauth*

Selle põhjal koostati lähte- ja sihtandmebaasi veergude vastavustabel Excelis. Näide *User* tabeli vastavusest, mis Excelis kirja pandi, on Lisa 2 – Näide *User* tabeli kohta tehtud vastavustabelist Excelis.

Kui lähteandmebaasis puudus veerule vaste, aga veeru täitmine oli vajalik, siis prooviti leida alternatiivne vaste. Kui veerg aga polnud rakenduse tööks hädavajalik, siis sellesse väärtust ei pandud. Kõik püsiprogrammeeritud (*hardcoded*) veeru väärtused on ettevõttega kooskõlastatud (Joonis 12).

```
UPDATE [target_db].[dbo].[user]
SET    [target_db].[dbo].[user].is_secure_sign_in_confirmed = 0,
       [target_db].[dbo].[user].needs_buum_sync = 0,
       [target_db].[dbo].[user].has_new_password = 0,
       [target_db].[dbo].[user].is_in_lhv_list = 0,
       [target_db].[dbo].[user].is_module_tools_visible = 0,
       [target_db].[dbo].[user].is_salesman_desktop_visible = 0,
       [target_db].[dbo].[user].send_credit_limit_offers = 0,
       [target_db].[dbo].[user].use_electronic_invoice = 0,
       [target_db].[dbo].[user].needs_omnisend_sync = 0,
       [target_db].[dbo].[user].send_invoice = 1,
       [target_db].[dbo].[user].last_login_type = 1,
       [target_db].[dbo].[user].source_id = 116
FROM [target_db].[dbo].[user];
```

Joonis 12 *User* tabelisse sisestatud kindlad numbrilised väärtused.

Andmete ülekandel lähtuti põhimõttest, et andmeid tule üle kanda nii vähe kui võimalik ja nii palju kui vajalik.

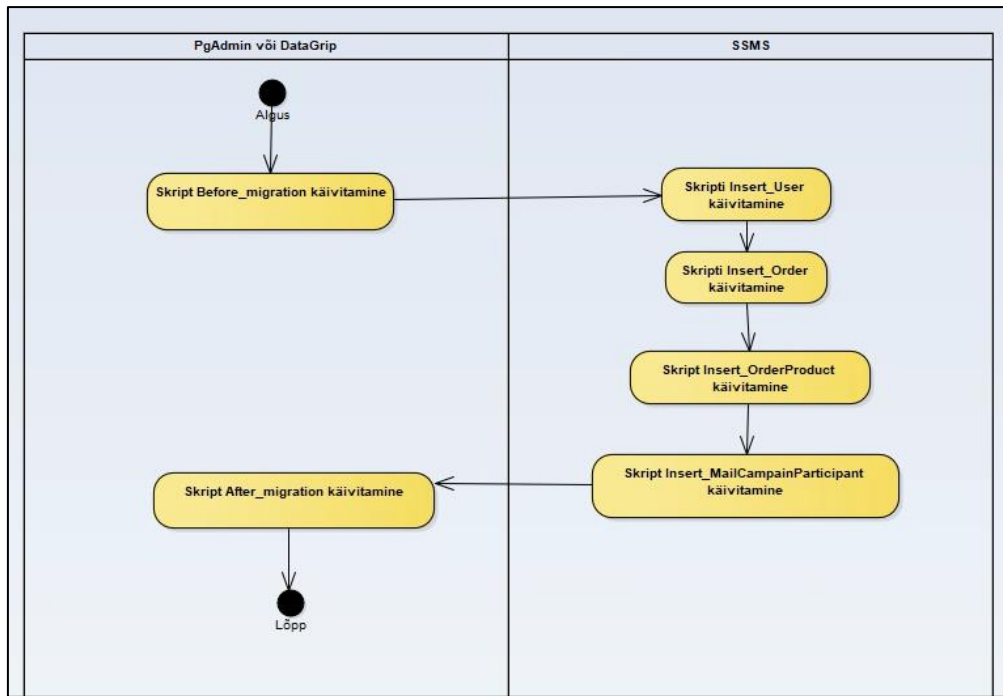
Telefoninumbritega tehti nii, et *User* tabelisse kantud telefoninumbrid jaotati lauatelefonideks ja mobiilinumbriteks ning sisestati vastavatesse veergudesse. *Order* tabelis telefoninumbreid mobiiltelefonideks ja tavatelefonideks jagama ei hakatud, kuna see ei olnud ettevõtte arvates esialgu oluline.

Vastavalt vastavustabelile, mille näide asub Lisa 2, koostati andmete ülekandmiseks mõeldud SQL laused. Igale sihtandmebaasi tabelile, kuhu andmeid kantakse, koostati eraldi fail. Failide nimed olid *Insert_User*, *Insert_Order*, *Inser_OrderProduct* ja *Inser_MailCampaignParticipant*. Need failid sisaldavad andmesiidiks vajalikke lauseid. Tühjadesse tabelitesse andmete lisamiseks kirjutati INSERT ja UPDATE laused. SQL lausete koostamisel lähtuti soovist, et neid saaks hiljem pärisandmetega tehtava tegeliku andmesiidide käigus samuti kasutada. Parema koodi hallatavuse jaoks tehti esmalt INSERT laused, kus lisati andmed kohustuslikesse (NOT NULL kitsendusega) veergudesse. Seejärel lisati ülejäänud veergude väärtused UPDATE lausetega.

Lisaks koostati veel kaks faili lähteandmebaasis (*PostgreSQL*) käivitamiseks: *Before_migration* ja *After_migration*. Esimene fail sisaldab abiveergude loomist ja veergude andmetüüpide muutmise lauseid. Teine fail on mõeldud loodud abiveergude kustutamiseks ja muudetud veeru tüüpide tagasi muutmiseks.

4.3.4 Teostus

Töö tulemusena valmisid andmesiidide läbiviimiseks SQL-lausetega skriptifailid. Kui laused olid koostatud, siis tegi autor kohalikus masinas MS SQL Serveri testandmebaasi neli tabelit tühjaks. Testandmebaasi tabelite tühjendamisest kirjutatakse pikemalt jaotises 5.2.1. Kui tabelid olid tühjendatud, siis testiti andmesiidret. Selleks käivitati SSMS'is ja pgAdminis andmesiidide jaoks koostatud failid. Joonis 13 esitab nende skriptifailide käivitamise järjekorra. Tegevusdiagrammi ujumisrajal näidatakse, millises programmis neid käivitati.



Joonis 13 SQL-lausetega skripti failide käivitamise tegevusdiagramm.

Peale andmesirde katsetust hinnati andmete õigsust ja päringu täitmise kiirust. Failide käivitamise järjekord on tähtis. Tabel 17 on välja toodud failide käivitamise järjekord, ülekantavate ridade arv, lausete arv failis, faili füüsiliste koodiridade arv ja skriptis sisalduvate lausete täitmise kiirus autori arvutis.

Tabel 17 Andmesirde lausete kokkuvõte.

Failide käivitamise järjekorra nr	Andmesirde faili nimi	Ridade arv test-andmebaasis	Lausete arv failis	Füüsiliste koodiridade arv failis	Failis olevate lausete täitmise kiirus
1	Insert_User	22 100	14	260	35 sek
2	Insert_Order	85 898	11	199	14 min 39sek
3	Inser_OrderProduct	100 023	2	36	47 sek
4	Inser_MailCampainParticipant	22100	1	18	u 1ms

Kuna ridade arv testandmebaasis on lähedane tegelikult ülekandmist vajavate andmete hulgale, siis võib eeldada tegelikult andmesiidel lähedast läbiviimise kiirust.

Koodiridade loendamisel lähtuti Source Lines of Code põhimõttest [30]. Koodiridu loendati spetsiaalse programmiga *cloc* [31].

Enne koodiridade arvu leidmist vormindati kood. Ettevõtte anonüümsuse säilitamiseks nimetati andmebaasid ümber. Lähteandmebaasi nimi asendati aliasega „*source_db*“ ja sihtandmebaasi nimi asendati aliasega „*target_db*“.

Koodi vormistamise põhimõtted olid järgmised.

Suurtähtedega on:

- võtmesõnad SQL lause alamosades ehk klauslites (nt INSERT, SELECT, FROM, WHERE),
- andmetüüpide nimed (nt VARCHAR, BIT, DATETIME),
- funktsioonide nimed (nt CONCAT, IIF, SUBSTRING).

Väiketähtedega on:

- veergude nimed,
- tabelite nimed.

Reavahetused ja joondamine:

- Klauslid algavad uuel realt.
- Kui veergudele viitamine on esitatud nimekirjana, siis pannakse veergude nimed eraldi ridadele ja koma on veeru nime järel.
- Kommentaarid on eraldi ridadel.
- SQL laused on üksteisest eraldatud tühja reaga.

Skriptides on kokku 49 SQL lauset:

- 4 *INSERT* lauset,

- 26 *UPDATE* lauset,
- 15 *ALTER* lauset,
- 4 *SET* lauset.

Kood on üles laetud GitLab'i koodihoidlasse aadressil:
<https://gitlab.com/loputoo/Kolimine>

Kõikide skriptide täitmine võttis kokku aega 19 minutit ja 45 sekundit, sellest andmete ülekanne 15 minutit ja 15 sekundit.

Autori arvuti, kus mõõdeti andmete ülekandeks kuluvat aega, tehnilised parameetrid on Protsessor on Intel® Core™ i7-4770HQ CPU, 4 tuuma, 8 lõime. Mälu on 16 GB DDR3 ja mälu kiirus on 1600 MHz. Arvuti on MacBook Pro 2015. aastast. Operatsioonisüsteemiks on Windows10. SSD ketas on Apple SM0256G.

4.3.5 Testimine

Testimise eesmärgil viidi andmesiire läbi ettevõtte arendaja arendusmasinas. Arendusmasinas võeti samad varukoopiad ja käivitati skriptid ehk tehti andmesiire PostgreSQL testandmebaasist MS SQL Serveri testandmebaasi. Seejärel käivitati veebirakendus, mis oli ühendatud kohaliku masina andmebaasi külge. Pärast rakenduse käivitamist kontrolliti veebilehitsejaga veebirakenduses olevate andmete õigsust.

Arendajal oli MS SQL Serveri versioon sama, mis ettevõttel tööbaasi jaoks kasutusel ja PostgreSQL versioon oli 9.5, mis oli vanem kui kasutas autor.

Andmesiirde katsetus õnnestus. Testmasina külge ühendatud veebirakendus käivitus ja testrakenduse kasutaja logis enda nimel sisse ning nägi enda andmeid ja tellimusi.

Tegelikku andmebaasi migreerimist ja skriptide kasutamist autor oma töös ei kajasta, kuna seda teeb ettevõtte peale töö kirjutamise lõppu.

5 Analüüs ja järeldused

Selles peatükis analüüsitakse töö tulemusi ja tehakse järeldusi. Peatükis kirjeldatakse milliseid probleeme sisaldab sihtandmebaasi disain ja kuidas neid võiks parandada. Analüüsitakse andmesiirde käigus tekkinud keerukusi ja mida neist õppida võiks.

5.1 Olemasoleva andmebaasi ülesehituse probleemid ja parandamise soovitused

Käesolevas jaotises analüüsitakse sihtandmebaasis olevat nelja tabelit, kuhu andmesiirde tulemusel andmeid üle kantakse. Analüüsi käigus leitud vigade ja probleemide kõrvaldamiseks tehakse parandusettepanekuid.

Töö käigus analüüsiti neid nelja tabelit Bill Karwini raamatus „SQL Antipatterns“ [19] esitatud SQL-andmebaaside disaini antimustrite suhtes ja Phil Factor'i raamatus „SQL Code Smells“ [20] välja toodud probleemide suhtes. Lisaks osutatakse ka muudele tabelite disainiprobleemidele. Järgnevalt käsitletakse leitud probleeme täpsemalt.

Indeksite hindamise jätab autor oma tööst välja, sest indeksid ei ole osaks andmebaasi kontseptuaalsest skeemist vaid siseskeemist. Probleemide tüüpide korral ei ole eesmärgiks loetleda kõiki seda tüüpi probleemi esinemisi, vaid tuuakse näiteid.

Kokkuvõttena võib öelda, et nii nagu ka teistes olemasolevate SQL-andmebaaside disaini uuringutes [21] [22] [23], leiti ka siin palju erinevat tüüpi vigu. Paljud nendest vigadest olid sellised, mida täheldati ka nimetatud teistes uuringutes.

5.1.1 Probleemid tabelite ülesehituses

Andmebaasi füüsilise disaini antimustritest esines tabelites antimuster nimega *Phantom File*, mis kujutab endast olukorda, kus faile, nt pilte, hoitakse väljaspool andmebaasi ja andmebaasis hoitakse ainult faili aadressi. Viide failile asub tekstitüüpi veerus. Probleeme sellise lahenduse puhul on mitu. Kui andmebaasis kustutatakse rida, mis viitab suurele failile, siis seda faili automaatselt ei kustutata. Failisüsteemi võib jääda suur kogus sellised faile, mida ei kasutata, aga mis võtavad suure hulga ruumi. Sama kehtib ka vastupidi. Kui kustutatakse fail, siis viide jääb ikka andmebaasi ja viitab ikka faili endisele

asukohale. Ka ei allu väljaspool andmebaasi asuvad failid andmebaasis toimuvatele transaktsioonidele. Andmebaasi varukoopiat tehes ei lisata väljaspool andmebaasi asuvaid faile varukoopiasse. Samuti on andmebaasis keeruline kontrollida faili aadressi õigsust. [22]

Sihtandmebaasis on *Order_product* tabelis veerg *case_station_image_url*, milles hoiustatakse telefoni kestade piltide asukoht serveris, mitte pilti ennast. E-poel on palju pildifaile. Tabelis *Order* on tellimuse kinnituse PDF faili nimi veerus *order_confirmation_filename*.

Soovitus oleks failide hoiustamiseks tutvuda Erle Maido bakalaureusetöös välja toodud nelja (*bytea*, suurobjekt, failisüsteem ja pilvelahendus) failide salvestamist toetavat SQL-andmebaasi disainimustriga ja valida vastavalt süsteemi nõuetele sobivaim [32]. Selles töös kirjeldatakse kahte disaini, kus fail on andmebaasisüsteemi kontrolli all (failide hoidmine andmebaasis *bytea* väärtustena ja suurobjektidena). Ka MS SQL Serveri puhul on kaks võimalust, kuidas hoida faile andmebaasisüsteemi kontrolli all [33]. Väiksemate failide jaoks sobib *VARBINARY* tüüpi veerg ja suuremate failide korral sobib *VARBINARY(MAX)* tüüpi veerg, millel on *FILESTREAM* omadus. Viimasel juhul salvestatakse väärtused kettale failisüsteemi failidena, kuid nende andmetega saab töötada läbi andmebaasisüsteemi ja need alluvad andmebaasisüsteemide transaktsioonide ja varundamise mehhanismidele [34].

Liiga suure veergude arvuga tabelite loomine. Suur hulk erinevaid andmeid on paigutatud kõik ühte tabelisse ehk tabelid ei lähtu otstarbe lahususe disainiprintsiibist ja on madala tasemeni normaliseeritud. See viitab normaliseerimise veale või pigem soovile tabelleid denormaliseerida. Seda põhjustab ka arvamus, et tabelite ühendamise päringud on kallid ehk nende täitmine võtab palju aega. Tavaliselt saab selliseid (baas)tabelleid asendada vaadete, hetktõmmiste ehk materialiseeritud vaadetega või tabelifunktsioonidega, mis opereerivad kõrge tasemeni normaliseeritud tabelite põhjal. [20]

User tabelis on 96 veergu ja *Order* tabelis on 123 veergu. Soovituseks oleks tabelid lahku lüüa ja teha eraldi väiksemad tabelid. Näiteks tabelis *User* registreeritakse nii eraisikutest kui ka juriidilistest isikutest kasutajate andmed. Parem lahendus oleks kasutada mustrit *Party* (osapool) [35] ning luua eraldi tabel eraisikute andmete jaoks, eraldi tabel

juriidiliste isikute (organisatsioonide) andmete jaoks ning lisaks ühine tabel *Party*, kus on sellistele atribuutidele ja seosetüüpidele vastavad andmed, mis on ühised nii era- kui ka juriidilistele isikutele. Töötaja-spetsiifilised andmed võiksid olla eraldi tabelis, mis on seotud eraisikute andmetega tabeliga.

Kasutatakse ühist- ja üldist klassifikaatorite tabelit. See tähendab, et erinevat tüüpi klassifikaatorite andmed on koos samades tabelites. Probleem seisneb lahenduse keerukuses ja päringute kirjutamise ebamugavuses. Selline klassifikaatorite tabel paisub suureks ja erinevat tüüpi klassifikaatorite erinevate atribuutide tõttu võib tabel jääda pooltühjaks. Aleksandra-Salome Vellemaa uuris oma bakalaureusetöös „Mõned disainimustrid klassifikaatorite esitamiseks SQL andmebaasides“ klassifikaatorite tabelite mustreid ja soovitas klassifikaatorite puhul kasutada disainimustrit „Kunstlik ühendaja“ [36]. Iga klassifikaatori tüüpi kohta tuleks luua eraldi tabel, kus on klassifikaatori kood ja nimetus. Ühises klassifikaatori tabelis, millega need tabelid on seotud, registreeritakse näiteks klassifikaatori väärtuse hetke seisund ja väärtuse registreerimise ning viimase muutmise aeg.

Klassifikaatorite puudus. Mitmetes kohtades, kus saaks kasutada klassifikaatoreid, on andmebaasis klassifikaatorite tabelite defineerimise asemel defineeritud rakenduse koodis loendustüübid, mille väärtused määravad võimalikud väärtused tabeli veerus. Klassifikaatorite lisamine ja uuendamine (nt nimetuse muutmine) tähendab rakenduse lähtekoodi muutmist. Samuti puudub võimalus hakata teatud klassifikaatori tüüpide korral andmebaasis vajalikke lisaandmeid registreerima. Näiteks tabelis *User* võiksid klassifikaatorid olla: Maksemeetod (*payment method*), Kliendigrupp (*customer group*), Sugu (*sex*), Kasutajaks registreerimise allikas (*source*), Viimase sisselogimise tüüp (*last login type*), Ametikoht (*job title*).

Rahvusvaheliste standardite eiramine. Tabelis *Order* registreeritakse keel numbrilise identifikaatorina ja keeled on defineeritud rakenduse koodis loendustüübina. Rahvusvaheline keelte klassifikaator [37] kirjeldab igale maailma keelele unikaalse koodi, kuid sellest standardist tulenevaid koodi ei kasutata.

Andmete dubleerimine samas tabelis. Andmete dubleerimine võib küll kiirendada teatud päringuid, kuid andmete kooskõla tagamiseks tuleb teha lisatööd ning alati on võimalus, et andmebaasi jõuavad üksteisega mittekooskõlas olevad väited. Näiteks *User*

tabelis on nii veerg *shipping_country* vabatekstilise riigi identifikaatori jaoks kui ka *shipping_country_id*, mis on viide klassifikaatori tabelile.

Tabelis *User* on veerg *age*, kus andmebaasi on salvestatud isiku vanus, mis on arvutatud veerg ja milles olev väärtus arvutatakse iga teatud aja tagant uuesti. Mõistlik oleks salvestada isiku sünnikuupäev ja vanus arvutada jooksvalt koodis. Tegelikult ongi kasutaja tabelis salvestatud ka sünniaeg. Järelikult on tegemist andmete dubleerimisega, sest vanuse saab arvutada nii sünniajast kui lugeda eraldi veerust. Kui kõigi kasutajate vanust iga päev ümber ei arvutata, siis võivad vanuse veerus olla ebaõiged andmed.

Tabelist *User* saab teada seda, kas isik on eraisik näiteks nii selle järgi, et *is_corporate* väärtus näitab, et isik pole juriidiline isik kui ka selle järgi, et on registreeritud *identification_nr*.

Andmete dubleerimine üle erinevate tabelite. Näiteks tabelis *Mail_campaign_participant* registreeritakse kliendi nimi ja e-posti aadress, mis on ka tabelis *User*. Info selle kohta, kas kasutaja on juriidiline isik või mitte, on tabelis *User*. Tellimus on seotud kasutajaga ning tabelis *Order* korratakse infot selle kohta, kas klient on juriidiline isik või mitte. Tellimuse juures registreeritakse poest kauba väljaandja nimi vabatekstina (*store_pickup_issuer*), mitte viitena *User* tabelis olevale kasutajale.

Perioodi salvestamise eelistamine täpsele ajahetkele. Perioodi salvestamine andmebaasi teeb programmeerija töö keerulisemaks. Perioodi pikkus on lõpu hetke ja alguse hetke vahe. Seda saab mõõta erinevates ühikutes ja võib tekkida vajadus päringute järele, mis vajavad tulemust teistsugustes ühikutes kui on kasutatud andmebaasis salvestatud andmete juures. Kasutades perioodi talletamiseks algushetke ja lõpphetke eraldi salvestamist, on päringute tegemine kergem ja erinevate ühikute kasutamise valikuvabadus on suurem. [20]

Sihtandmebaasis esineb selline viga *Order_product* tabelis veerul *guarantee_product_period*, kus andmebaasi on salvestatud tootele müüdü lisagarantii pikkus perioodina.

5.1.2 Kitsenduste vead

Tabelites ei ole sisulisi unikaalseid võtmeid. Seda probleemi kirjeldab B. Karwin muustriga „Üks suurus kõigile“ [19] Tabelites on unikaalne süsteemi genereeritud väärtus veerus nimega *id*, kuid ei ole muid unikaalsuse kitsendusi. Seega saab tegelikult registreerida sama olemi andmeid korduvalt, erinevate *id* väärtustega – andmebaasisüsteemi poolt selleks takistust ei ole.

Tabelis *User* ei lubata ühesuguseid e-posti aadresse sisestada, aga samas pole see veerg ka kohustuslik. Seega mitu korda sama kasutaja sisestamine, ilma e-posti aadressita, on lubatud.

Soovituseks on leida ja jõustada sisulised unikaalsuse kitsendused, deklareerides veergudele NOT NULL ja UNIQUE kitsendused.

Tabelites on väga palju mittekohustuslikke veerge, mis lubavad NULLe. Järgnevalt tuuakse välja, kui mitu protsenti veergudest igas tabelis lubavad *NULLe*:

- 84,37% *User* tabelist,
- 86,99% *Order* tabelist,
- 90% *Order_product* tabelist,
- 70% *Mail_campaign_participant* tabelist.

Soovituseks on deklareerida kõigile veergudele, kus peab olema väärtus, NOT NULL kitsendus. Kui veerul on vaikimisi väärtus, siis see ei taga seda, et väljas peab väärtus olema, sest vaikimisi väärtuse saab alati ise üle kirjutada.

Üheski tabelis ei ole ühtegi CHECK kitsendust. Andmebaasis saab näiteks registreerida negatiivseid hindu, koguseid ja vanuseid, @ märgita meiliaadresse ja sündmuste toimumise aegu ebareaalselt kauges minevikus ja tulevikus. Isegi kui andmeid kontrollib rakendus, siis a) reeglid pole dokumenteeritud (andmebaasis olevad kitsendused oleksid ühtlasi ka nende reeglite ühes kohas koos oleva dokumentatsiooni eest) ja ainus viis neid teada saada on vaadata lähtekoodi b) kui andmeid ei hallata läbi rakenduse, siis reeglid ei rakendu ja c) rakenduse uuendamisel tuleb reeglite kontroll uuesti teha.

Soovituseks on kõigepealt dokumenteerida andmetele kehtivad reeglid ja siis kaaluda nende alusel CHECK kitsenduste loomist.

Vale andmetüübi kasutamine. Veerule andmetüübi määramine on kitsenduse defineerimine, sest andmetüüp määrab ära võimalikud väärtused, mida veerus saab hoida. Andmete hoidmine valel kujul põhjustab probleeme andmete tähendusest arusaamises, aga ka andmete kodeerimises, sorteerimises, indekseerimises ja muudes toimingutes. Sihtandmebaasis esineb selline viga näiteks *User* tabelis veerul *sex*. Veerus olev väärtus peaks näitama kasutaja sugu, aga antud juhul see nii pole. Veeu andmetüübiks on *BIT*. Tegemine on tõeväärtustüübi asemel kasutatava andmetüübiga, kus 1 tähistab tõeväärtust *TRUE* ja 0 tähistab tõeväärtust *FALSE*. Antud juhul näitab 1, et kasutaja on meessoost ja 0 tähistab, et kasutaja on naissoost. Antud lahendusega aga ei saa katta kõiki soo andmete registreerimisega seotud olukordi.

Siin oleks hea lähtuda standardist ISO/IEC 5218, mis on mõeldud inimsugude kajastamiseks andmebaasides. See soovib soo kajastamiseks kasutada nelja spetsiifilist koodi, kus: 0 = Teadmata, 1 = Meessoost, 2 = Naissoost, 9 = Soolist määramist ei kohaldata. [38] Kuna sihtandmebaasis registreeritakse eraisikute ja juriidiliste isikute andmed ühes tabelis, siis peaks juriidiliste isikute puhul kasutama koodi 9. Ühtlasi võimaldaks sellise klassifikaatori kasutamine veeru *sex* kohustuslikuks muuta ehk deklareerida sellele *NOT NULL* kitsenduse.

Tabeli *User* veerg *has_birthday* on tüüpi *INTEGER*, kuid tegelikult piisaks tüübist *BIT*, sest registreeritakse väärtuseid 0 ja 1.

5.1.3 Maksimaalsed väljapikkused

Ka maksimaalsed väljapikkused on tegelikult kitsendused andmetele.

Erinevad maksimaalsed väljapikkused ehk väljapikkused samasisuliste andmetega veergudel. Andmebaasis puudub ühtne lähenemine sama sisuga andmetega veergude väljapikkuste määramisele. Selline lahendus tekitab küsimusi, miks ühes veerus on samale infole jäetud rohkem ruumi kui teises. See tekitab andmete kasutajas segadust. Tabel 18 toob välja mõned näited sihtandmebaasi tabelites esinenud väljapikkuste erinevustes.

Tabel 18 Samasisulised aga erinevate väljapikkustega veerud.

	Tabeli nimi	Veeru nimi	Andmetüüp	Välja- pikkus
Aadress	order	billing_address	NVARCHAR	200
	order	shipping_address	NVARCHAR	200
	user	address	NVARCHAR	50
	user	shipping_address	NVARCHAR	50
E-posti aadress	mail_campaign_ participant	email	NVARCHAR	320
	order	billing_email	NVARCHAR	100
	order	shipping_email	NVARCHAR	100
	user	email	NVARCHAR	100
	user	shipping_email	NVARCHAR	50
	user	changed_email	NVARCHAR	100
Eesnimi	mail_campaign_ participant	first_name	NVARCHAR	100
	order	billing_first_name	NVARCHAR	50
	order	shipping_first_name	NVARCHAR	50
	user	first_name	NVARCHAR	50
	user	shipping_first_name	NVARCHAR	50
Ettevõtte- nimi	order	billing_company_name	NVARCHAR	100
	order	shipping_company_ name	NVARCHAR	100
	user	company_name	NVARCHAR	101
	user	shipping_company_ name	NVARCHAR	50
Mobiil- telefoni number	order	billing_mobile_phone	NVARCHAR	25
	order	shipping_mobile_phone	NVARCHAR	25
	user	shipping_mobile_phone	NVARCHAR	50
Perekonna- nimi	mail_campaign_ participant	last_name	NVARCHAR	100

	Tabeli nimi	Veeru nimi	Andmetüüp	Välja- pikkus
	order	billing_last_name	NVARCHAR	50
	order	shipping_last_name	NVARCHAR	50
	user	last_name	NVARCHAR	50
	user	shipping_last_name	NVARCHAR	50
Postiindeks	order	billing_postal_code	NVARCHAR	20
	order	shipping_postal_code	NVARCHAR	20
	user	postal_code	NVARCHAR	50
	user	shipping_postal_code	NVARCHAR	50
Telefoni- number	order	billing_phone	NVARCHAR	20
	order	shipping_phone	NVARCHAR	20
	user	phone	NVARCHAR	25
	user	shipping_phone	NVARCHAR	50

Soovituseks on ühtlustada väljapikkuste kasutus.

Ebaõiged väljapikkused.

Tabel 18 on näha postiindekseid sisaldavate veergude väljapikkusi *User* ja *Order* tabelites. Postiindeks on veergudes maksimaalse väljapikkusega 20 kuni 50. Antud pikkus tundub olevat ülehinnatud. Näiteks siin [39] pakuvad erinevad autorid maksimaalset pikkust vahemikus 10 kuni 20.

E-posti aadressi pikkus ei tohi ületada 254 märki, kuid andmebaasis olevate veergude väljapikkused ei arvesta sellega nagu on näha Tabel 18. [40]

Andmebaasides olevate isikunimede veergude väljapikkused ei arvesta väga pikkade nimedega [41]. Tabel 18 on näha, et eesnime ja perekonnanime väljapikkuseks on määratud maksimaalselt ainult 100.

Aadressi veerud ei arvesta väga pikkade kohanimede võimalusega. Pikim kohanimi on Wikipedia [42] kohaselt 85 märki. Ühe sõnena kirja pandav aadress koosneb mitmest komponendist ehk kohanimest.

5.1.4 Nimetamise probleemid

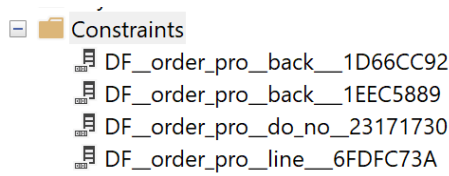
Reserveeritud sõnade kasutamine nimedes. Võimalusel vältida nimetamisel selliste sõnade kasutamist. Nende sõnade kasutamine halvendab koodi loetavust, võib tekitada probleeme koodi vormindajale ja võib tekitada vigu koodi kirjutamisel. [20]

Sihtandmebaasis on selliste nimedega kaks tabelit: *User* ja *Order*.

Numbrite kasutamine andmebaasiobjektide nimes. [20] Sihtandmebaasis on mõnedel välisvõtme kitsendustel nimede lõpus numbritest ja tähtedest koosnev kood (Joonis 14).

Süsteemi poolt genereeritud nimede kasutamine kitsenduste puhul. See juhtub näiteks primaarvõtmetega ja välisvõtmetega, kui andmekirjelduskeele lausetes (*Data Definition Language* ehk *DDL*) ei panda kitsendustele nime. Sellisel juhul annab nimed andmebaasisüsteem. Automaatselt genereeritud nimesid on hiljem keerulisem kasutada ja need tekitavad kergesti segadust. [20]

Sihtandmebaasis on mõnede välisvõtme kitsenduste nimede lõpus automaatselt genereeritud numbritest ja tähtedest koosnev kood (Jooni 14).



Joonis 14 Sihtandmebaasis *Order* tabeli välisvõtme kitsenduste nimed.

Isikunimede hoidmisel andmebaasis ei arvestata kultuuriliste eripäradega. Eeldatakse, et isikul on esimene nimi ja teine nimi (veerud *first_name* ja *last_name*). Selline lahendus ei kajasta kultuure, kus kasutusel on rohkem kui üks eesnimi ja/ või perekonnanimi, perekonnanime puudumist või suuremat hulka erinevaid

nimekomponente. Samuti ei arvestata kultuuridega, kus isikunimede struktuuris on esimesel kohal on hoopis perekonnanimi ja viimasel kohal eesnimi.

Isikunimede hoidmisel ja nende veergude nimetamisel andmebaasides on soovitatav tutvuda Mairi Jõgi bakalaureusetöoga „Mõned disainimustrid isikunimede hoidmiseks SQL-andmebaasides“, kus saab vastavalt vajadustele ja valida sobivaima disainimustri [43].

Sihtandmebaasis nimetatakse isiku nimede hoidmise veerge *first_name* ja *last_name*. Antud juhul oleks parem kasutada veeru nimedena *given_name* ja *surname* [43].

Tõstutundlikud nimed. Tabelis *Order_product* on neli veergu, mille nimed on tõstutundlikud, st eristatakse suur ja väiketähti (*Order_Id, Product_Id, Quantity, Price*). Tõstutundlikud nimed muudavad koodi, sh SQL koodi, kirjutamise keerulisemaks. [44] Nende asemel peaks eelistama tõstutundetuid nimesid.

5.1.5 Üldised

Andmebaasis on tabelites mitu sellist veergu, mis on vanad ehk mida rakenduses enam ei kasutata. Need veerud võiks andmebaasist kustutada (Tabel 19).

Tabel 19 Vanad veerud, mida enam ei kasutata.

Tabeli nimi	Veeru nimi
user	shipping_identification_code
user	shipping_company_reg_nr
user	products_for_user_calculated_at
user	sent_email_to_notify_corporate
user	is_module_tools_visible
user	preferred_product_sort
order	for_user_id
order_product	guarantee_product_code
order_product	guarantee_product_price

5.2 Andmesiirde käigus esile kerkinud keerukused

Selles jaotises tutvustatakse andmesiirde käigus ettetulnud keerukusi ja probleeme ning seda, kuidas neid lahendati.

5.2.1 Olemasolevate andmebaaside dokumenteerimine

Autoril ei olnud tööd alustades kasutada lähteandmebaasi ja sihtandmebaasi dokumentatsiooni, sest ettevõttel need puudusid. Autor sai vaadata tegelike andmebaaside struktuuri, testandmeid ja konsulteerida arendajaga. Andmestruktuuridest arusaamine on eduka andmesiirde vältimatu eeldus. Autor ei loendanud täpselt töötunde, kuid hindab tagantjärele, et andmebaaside dokumenteerimisele kulus umbes sama palju aega (umbes üks kuu) kui andmesiirde SQL lausete koostamiseks. Autori soovitus on arendamise käigus andmebaase dokumenteerida ja anda andmebaasiobjektidele (nt tabelid, veerud) nimed, mis lihtsustavad nende tähenduse mõistmist. Käesolevates andmebaasides oli tabelite ja veergude nimetamine arusaadavuse mõttes üsna hea ja lihtsustas tabelitest ning veergudest aru saamist.

Kõige raskem oli nende veergudega, mis olid vanad ja mida praegu enam ei kasutata. Sihtandmebaasi kasutatud tabelites oli kümme sellist veergu. Neile oli selgituste leidmine raskem ja pidi rohkemate inimeste käest uurima. Ühe tähendust ei õnnestunudki välja selgitada. Lähteandmebaasis selliseid veerge ei olnud, sest see on tegelikult väljatöötamise mõttes uuem andmebaas.

5.2.2 Testandmebaasi tabelite tühjendamine

Andmete tegelikul ülekandmisel kantakse andmed tühja andmebaasi. Selleks, et andmete ülekandeskriptid oleksid kasutatavad hiljem ka päris andmetega, tuli andmete ülekandeks luua samasugused tingimised ka testbaasis. MS SQL Serveri testandmebaasis olid testandmed ka nendes neljas tabelis, kuhu andmed üle kantakse.

Tabeli sisu kustutamisel tuleb enne kustutada sellest välisvõtmete kaudu sõltuvuses olevate tabelite sisu. Alles siis saab tabelit tühjendada.

Sihtandmebaasi on suur, 204 tabelit ja seal on palju tabelite vahelisi seoseid. Ainuüksi tabeliga *User* on seotud 83 tabelit ja *Order* tabeliga on seotud 30 tabelit. Seega kogu andmebaasi tühjaks tegemine ja mitteseotud tabelitesse andmete uuesti sisestamine on

liiga suur töö. Autor otsustas selle asemel valida neli tabelit kuhu andmesiirdega peab andmeid lisama ja neist välisvõtmete kaudu sõltuvad tabelid (võib olla mitmetasemeline sõltuvus, kus sõltuval tabelil on omakorda sõltuvused), eemaldada valitud tabelitega seotud kitsendused, kustutada valitud tabelid ning taastada eemaldatud kitsendused ja tabelid.

Tabeliga seotud objektide leidmiseks proovis autor internetist leitud avatud lähtekoodiga skripti [45], aga selgus, et salvestatud protseduuride rekursiooni limiit MS SQL'is on 32 ja see ei sobinud, sest seoste puu läks veel sügavamaks.

Sellejärel proovis autor ise tabelitelt kitsendusi eemaldada. Kuid tabelitel on palju kitsendusi ning nende eemaldamine ja uuesti loomine on mahukas töö. Seetõttu otsustas autor töö lihtsustamiseks kasutada tööriista.

SSMS'is on selleks funktsionaalsus olemas. Täpsem andmebaasi tabelite tühjendamise juhend esitatakse Lisa 3 - SSMS kaudu andmebaasi tabelite tühjendamise juhend.

SSMS koostab ise vajalike SQL lausetega faili, mille käivitamisel kitsendused eemaldatakse, tabelid kustutatakse (*DROP TABLE*) ning seejärel tabelid (sh kitsendused) ja indeksid taastatakse. Selle lahenduse puuduseks on, et MS SQL Serveri poolt pannakse osadele kitsenduste nimedele juurde süsteemi poolt genereeritud numbriline kood ja see on iga kord erinev. Loodud skriptiga kustutati ja loodi uuesti kokku 83 tabelit.

Tööandmebaasis seda skripti käivitama ei pea, kuna see andmebaas on juba tühi. Kui ettevõtte otsustab sama vahendit kasutada oma testandmebaasi tühjendamiseks, siis peab ta kogu tühjendamise protsessi uuesti tegema, genereerides tabelite tühjendamiseks uued skriptid. MS SQL Server genereerib kitsenduste nimedele järelliited ja seetõttu ei saa sama faili mitu korda kasutada. Veel üks põhjus, miks sama faili tööandmebaasis kasutada ei saa on see, et tööandmebaas on pidevas arenduses ja sellepärast võib seal tabelite arv teine olla.

Õppida tasuks seda, et testbaasist andmete kustutamine ei pruugi olla nii lihtne kui algul tundub. Kõik oleneb kustutatava tabeli seoste arvust teiste tabelitega ja sellest, millised on välisvõtme kitsendustega seotud kompenseerivad tegevused. Kui need oleksid *ON DELETE CASCADE*, siis oleks sellisel otstarbel kustutamine lihtne. Samas tööbaasis

oleks *ON DELETE CASCADE* liigne kasutamine ohtlik, sest võib tingida ootamatu suure andmehulga kustutamise.

5.2.3 User tabelisse andmete ülekandmine

Kasutajate tabelisse andmete ülekandmine tehti nii, et ettevõtte soovil kanti koos ülejäänud andmetega üle ka lähteandmebaasi kasutajate tabeli identifikaatorid (lähteandmebaasi *Accounts_user* tabeli *id* veerus olev väärtus).

MS SQL Serveris võib tabeli veerg olla määratud identiteedi veeruks, mis tähendab, et sellesse veergu genereerib süsteem sisemist arvujada generaatorit kasutades ise unikaalseid täisarvulisi väärtuseid. Selleks, et tabeli identiteedi veergu saaks kasutaja ise väärtuseid sisestada, peab olema sellele tabelile määratud *IDENTITY_INSERT* omadus.

MS SQL Serveril on piirang, et sessiooni piires saab kasutajal olla ainult ühel tabelil *IDENTITY_INSERT* omadus sisse lülitatud [46]. Primaarvõtme väärtuste ise tabelisse sisestamiseks tuli see omadus enne sisse lülitada ja peale INSERT lauset jälle välja lülitada (Joonis 15).

```
-- enne primaarvõtme väärtuste sisestamist
SET IDENTITY_INSERT [target_db].[dbo].[user] ON;

-- primaarvõtme väärtuste sisestamine
INSERT INTO [target_db].[dbo].[user]
    (id,
     is_corporate,
     first_name,
     last_name,
     company_name,
     customer_name,
     email)
SELECT au.id,
       au.is_company_account,
       au.first_name,
       au.last_name,
       au.company_name,
       (SELECT IIF(au.is_company_account = 'TRUE', au.company_name, (
           SELECT CONCAT(au.first_name, ' ', au.last_name)))) AS customer_name,
       au.email
FROM [POSTGRES].[source_db].[public].[accounts_user] AS au;

--pärast primaarvõtme väärtuste sisestamist
SET IDENTITY_INSERT [target_db].[dbo].[user] OFF;
```

Joonis 15 *User* tabeli INSERT ja SET laused.

Peale andmete tabelisse lisamist tuli ka muuta primaarvõtme väärtuste generaatori juures seda arvu, millest alates see uusi väärtuseid genereerib (Joonis 16). Järgmised genereeritavad arvud peavad olema suuremad kui andmebaasis võtmeveerus peale andmete ülekannet olev kõige suurem arv.

```
DECLARE @NewSeed NUMERIC(10)
SELECT @NewSeed = MAX(id) FROM [target_db].[dbo].[user];
DBCC CHECKIDENT([user], RESEED, @NewSeed);
```

Joonis 16 *User* tabelis primaarvõtme väärtuste generaatori lähtestamine.

Kõik see ei olnud otseselt probleem, aga kuna töö autoril puudus selle teostamise kogemus, siis oli see autori jaoks õppimiskoht.

5.2.3.1 Kliendi aadress

Probleem oli selles, et lähteandmebaasis võib kasutajal aadresside tabelis olla mitu aadressi, aga sihtandmebaasis saab kasutajal olla ainult üks aadress. Lähteandmebaasist õige rea leidmine ei olnud nii lihtne, kui algul tundus. Kui kasutaja on määranud mõne oma aadressi vaikimisi aadressiks, siis tuli just see aadress sihtandmebaasi üle kanda. Kui vaikimisi aadressi polnud määratud, siis tuli leida kõige uuema kuupäevaga aadressi rida ja võtta sealt vajalikud andmed (Joonis 17). Ka telefoninumber tuli võtta samalt realt.

```
UPDATE [target_db].[dbo].[user]
SET
  [target_db].[dbo].[user].address = (SELECT CONCAT(join_result.line1, ' ',
join_result.line2)),
  [target_db].[dbo].[user].phone = join_result.phone_number,
  [target_db].[dbo].[user].postal_code = join_result.postcode,
  [target_db].[dbo].[user].county = join_result.state,
  [target_db].[dbo].[user].country = join_result.country_id,
  [target_db].[dbo].[user].city = join_result.line4,
  [target_db].[dbo].[user].shipping_first_name = join_result.first_name,
  [target_db].[dbo].[user].shipping_last_name = join_result.last_name,
  [target_db].[dbo].[user].shipping_country = join_result.country_id,
  [target_db].[dbo].[user].shipping_county = join_result.state,
  [target_db].[dbo].[user].shipping_address = (
    SELECT CONCAT(join_result.line1, ' ', join_result.line2)),
  [target_db].[dbo].[user].shipping_postal_code = join_result.postcode,
  [target_db].[dbo].[user].shipping_company_name = join_result.company_name,
  [target_db].[dbo].[user].shipping_city = join_result.line4
FROM [target_db].[dbo].[user] AS u
RIGHT JOIN
  (SELECT DISTINCT(au.id),
    au.date_created,
    au.line1,
    au.line2,
    au.phone_number,
```

```

        aua.postcode,
        aua.state,
        aua.line4, --city,
        aua.first_name,
        aua.last_name,
        aua.country_id,
        aua.company_name
FROM [POSTGRES].[source_db].[public].[accounts_user] AS au
RIGHT JOIN
--millisel kuupäeval loodi iga kasutaja kõige uuem aadressi rida
(SELECT user_id,
      MAX(date_created) MaxDate
FROM [POSTGRES].[source_db].[public].[address_useraddress]
GROUP BY user_id) AS MaxDates ON au.id = MaxDates.user_id
LEFT JOIN
--kõige uuem rida
[POSTGRES].[source_db].[public].[address_useraddress] AS aua
ON MaxDates.user_id = aua.user_id AND
MaxDates.MaxDate = aua.date_created) AS join_result
ON u.id=join_result.id;

```

Joonis 17 Kliendi kõige uuema aadressi leidmine.

Teine suurem probleem oli selles, et lähteandmebaasi aadresside sisestamine oli tehtud nii, et väljade täitmisel ei arvestatud juhtudega, kui vajalik aadress ei asu linnas. Kui aadress ei asu linnas ja vaja on sisestada küla ja vald, siis selleks kindlaid välju pole ja sisestaja peab ise andmete registreerimisel loominguliselt lähenema.

5.2.3.2 Telefoni number

Lähteandmebaasis on veerg „telefoni number“ (*phone_number*), kuhu võib sisestada lauatelefonide või mobiiltelefoni numbreid. Sihtandmebaasis on selleks aga kaks eraldi veergu. Mobiiltelefoni numbreid kasutatakse näiteks turunduses SMS'i teel pakkumiste saatmiseks. Kuna tegu on Leedu e-poega ja kaupa tarnitakse ainult Leedu piires, siis võib eeldada, et enamuse kasutajaid on Leedu numbritega.

Telefoninumbrite eristamisel lähtuti teadmistest, et Leedu suunakood on +370 ja Leedu lauatelefonide numbrid algavad numbritega 3, 4 või 5. [47] Selle alusel eraldati telefoninumbritest lauatelefonid (Joonis 18), ülejäänud numbrid tõsteti mobiiltelefonide alla.

```

UPDATE [target_db].[dbo].[user]
SET
    [target_db].[dbo].[user].shipping_phone = landline_nrs.phone
FROM [target_db].[dbo].[user] AS u
RIGHT JOIN
    (SELECT DISTINCT(au.id) AS old_user_id,
        aua.phone_number AS phone
    FROM [POSTGRES].[source_db].[public].[accounts_user] AS au
    LEFT JOIN
        --millisel kuupäeval loodi iga kasutaja kõige uuem aadressi rida
        (SELECT user_id,
            MAX(date_created) MaxDate
        FROM [POSTGRES].[source_db].[public].[address_useraddress]
        GROUP BY user_id) MaxDates
    ON au.id = MaxDates.user_id
    LEFT JOIN
        --kõige uuem rida
        [POSTGRES].[source_db].[public].[address_useraddress] aua
    ON MaxDates.user_id = aua.user_id AND
        MaxDates.MaxDate = aua.date_created
    WHERE aua.is_default_for_shipping='false' AND
        (SUBSTRING(aua.phone_number, 1,4)='+370') AND
        (SUBSTRING(aua.phone_number, 5,1) IN ('3', '4', '5')))) as
landline_nrs
ON u.id=landline_nrs.old_user_id;

```

Joonis 18 Kliendi kõige uuema aadressi pealt lauatelefonide leidmine.

Selle ülesande lahendamiseks mõeldud SQL kood läks pikaks, kuna number tuli võtta aadressi tabelist, kas kliendi vaikimisi määratud aadressi realt või siis kliendi kõige uuema sisestuse pealt.

5.2.3.3 Riigi identifikaator

Lähteandmebaasis on riigi identifitseerimise kood veerus *country_id* kahetähelise koodina. Sihtandmebaasis on see aga numbriline kood, mille sisu tuleb tabelist *Classifier*, kus on kirjas klassifikaatorite tüübid. Klassifikaatorite väärtused tulevad tabelist *Classifier_value*.

Selgus, et PostgreSQL testandmebaasis on *county_id* veerus ühes reas väärtus „GB“, aga MS SQL Serveri testandmebaasi tabelis *Classifier_value* puudus sellele vastav väärtus „Great Britain“.

Tööandmebaasis ei tohiks sellist viga tekkida, kuna arendusmeeskond lisab Leedu ja Suurbritannia riikide andmed enne andmesiiret klassifikaatorite väärtuste tabelisse.

5.2.4 Order tabelisse andmete ülekandmine

Selles jaotises kirjeldatakse probleeme, mis tekkisid seoses sihtandmebaasi tabelisse *Order* andmete ülekandmisega.

5.2.4.1 Tellimuse identifikaator

Lähteandmebaasis on tellimuste tabelis nimega *Order_order* veerud *id* ja *number*, mis oma sisult on erinevad. *Id* on andmebaasis tellimuste identifikaator. Tegemist on surrogaatvõtmega – süsteemi genereeritud unikaalne täisarv, mis on mõeldud tellimustele viitamiseks süsteemis, kuid mida ei näidata lõppkasutajatele. Veeu andmetüüp on *INTEGER*. Number on tellimuse number, mida näidatakse arvel kliendile ja see on andmetüübiga *CHARACTER VARYING*. Tellimuse number koosneb ühest tähes, millel järgneb numbrimärkidest koosnev osa.

Ettevõtte soovis, et sihtandmebaasis oleks tellimuste tabeli primaarvõtmeks ja tellimuste identifikaatoriks lähteandmebaasi veeru *number* sisu, ilma ees oleva täheta. Edaspidi hakataks sellesse veergu arvujada generaatorit kasutades genereerima uusi arvulisi väärtuseid.

Selleks teisendati kõigepealt *number* veerus olevad andmed õigele kujule ehk eemaldati väärtuse algusest tähemärk ning siis teisendati allesjäänud väärtus *INTEGER* andmetüüpi väärtuseks (Joonis 19).

```
ALTER TABLE order_order
ADD order_number_int INT;

UPDATE order_order
SET order_number_int = CAST(RIGHT(number, LENGTH(number) - 1) AS INT)
WHERE number NOT LIKE 'DEV-JU%';

UPDATE order_order
SET order_number_int = CAST(RIGHT(number, LENGTH(number) - 6) AS INT)
WHERE number LIKE 'DEV-JU%';
```

Joonis 19 *Before_migration* failis tellimuse numbri teisendamine.

Pärast tellimuste numbrite ülekannet muudeti ka selle veeruga seotud arvujada generaatorit nii, et järgmine väljastatav arv oleks viimasele sisestusele järgnev arv.

5.2.4.2 Sõnum

Lähteandmebaasis on veerg *message* andmetüübiga *TEXT*. SSMS aga ei toeta sellist tüüpi veerge. Selleks, et päringuid saaks teha ja andmeid üle kanda, tehti PostgreSQL testandmebaasi *Order_order* tabelisse ajutiselt juurde uus veerg *message2* andmetüübiga *CHARECTER VARYING* ja pikkusega 1 000 000. Tööandmebaasis on kõige pikem teade pikkusega 531 824 märki. Veergu *message2* kopeeriti veeru *message* sisu teisendatud kujul. Seejärel esialgne veerg *message* kustutati ja veerg *message2* nimetati ümber *message* (Joonis 20).

```
ALTER TABLE order_ordernote
ADD message2 VARCHAR(1000000);

UPDATE order_ordernote
SET message2= message;

ALTER TABLE order_ordernote
DROP COLUMN message;

ALTER TABLE order_ordernote
RENAME COLUMN message2 TO message;
```

Joonis 20 *Before_migration* failis veeru *message* andmetüübi muutmine.

Töö lõppedes muudetakse skriptifailiga *After_migration* kõik jälle endiseks tagasi (Joonis 21).

```
ALTER TABLE order_ordernote
ADD message2 TEXT;

UPDATE order_ordernote
SET message2= message;

ALTER TABLE order_ordernote
DROP COLUMN message;

ALTER TABLE order_ordernote
RENAME COLUMN message2 TO message;
```

Joonis 21 Veeru *message* andmetüübi tagasi muutmine.

5.2.5 *Order_product* tabelisse andmete ülekandmine

Lähteandmebaasi ja sihtandmebaasi toodete tabelid sisaldavad samu andmeid. Seetõttu saab andmed omavahel siduda UPC koodi kaudu. Kuigi esialgu tundus, et veergudes

oleva andmed on samasugused, siis tuleb ikkagi PostgreSQL andmebaasist UPC veerus loetud väärtused teisendada NVARCHAR tüüpi väärtuseks, kuna MS SQL üritab muidu UPC koode, mis sisaldavad ainult numbreid, teisendada *INTEGER* tüüpi väärtuseks. Sel juhul ei anna kahe andmebaasi *Product* tabelite ühendamine soovitud tulemust (Joonis 22).

PostgreSQL testandmebaasis oli *Order_line* tabelis viiteid sellistele toodetele, mida MS SQL Serveri testandmebaasi *Product* tabelis pole. Selgus, et see on testandmebaaside probleem ja tööbaasis sellist probleemi ei ole. Toodete haldussüsteemis on olemas kõik tooted ja need kantakse ka üle sihttööandmebaasi.

```
INSERT INTO [target_db].[dbo].[order_product] (  
    Order_Id,  
    Product_Id,  
    Quantity,  
    Price,  
    currency_id,  
    created_at,  
    do_not_check_stock,  
    line_amount,  
    back_order_quantity,  
    buy_in_price,  
    price_without_vat,  
    old_product_id  
)  
SELECT  
    oo.order_number_int,  
    p.id,  
    ol.quantity,  
    ol.unit_price_incl_tax,  
    4,  
    ol.date_updated,  
    1,  
    0,  
    0,  
    ol.unit_cost_price,  
    ol.line_price_excl_tax,  
    ol.product_id  
FROM target_db.dbo.product AS p  
RIGHT JOIN  
    [POSTGRES].[source_db].[public].[order_line] AS ol  
ON (p.code)=CAST(ol.upc AS NVARCHAR)  
LEFT JOIN  
    [POSTGRES].[source_db].[public].[order_order] AS oo  
ON oo.id=ol.order_id;
```

Joonis 22 Ridade lisamine *order_product* tabelisse.

6 Kokkuvõte

Käesoleva bakalaureusetöö põhiliseks eesmärgiks oli migreerida, ehk vajadusel teisendusi tehes üle kanda, e-poe andmed PostgreSQL andmebaasist Microsoft SQL Serveri andmebaasi. Mõlemad andmebaasisüsteemid on SQL-andmebaasisüsteemid.

Töö raskus oli selles, et nende andmebaaside kontseptuaalsed skeemid ehk baastabelite struktuurid on erinevad, kuna need on loodud erinevate rakenduste jaoks. Lähteandmebaas on osa valmislahendusest ehk karbitootest ja sihtandmebaas on ettevõtte arendajate käsitöö. Seega ei olnud võimalik kanda lähteandmebaasi tabelitest andmeid üks-ühele üle sihtandmebaasi. Kasutatud tabelite struktuurist oli vaja aru saada ja teisendada andmed ülekandmise käigus sihtandmebaasi skeemiga sobivaks.

Ettevõtte poolt anti ette kriteeriumid, millest lähtuvalt valiti sobivaim andmesiirde viis. Töös sõnastati nõuded andmesiirde tööriistale ja võrreldi kolme tööriista. Parimaks tööriistaks osutus Microsoft SQL Server Management Studio (SSMS), kuna see võimaldab käivitada SQL lauseid nii lähte- kui ka sihtandmebaasi põhjal. Autor kasutab skriptipõhist lahendust, kus andmete ülekandmiseks kasutatakse ise kirjutatud SQL lauseid.

Andmed võetakse lähteandmebaasi üheksast tabelist ja pannakse sihtandmebaasi nelja tabelisse. Tegelik tabelite hulk nendes andmebaasides on suurem. Autor töötas lähte- ja sihtandmebaasi kohaliku koopiaga, milles oli originaalidega samasugune struktuur kuid kus olid testandmed. Töös tehti kõigepealt esmane kaardistus Excelis, kus sihtandmebaasi nelja tabeli veergudele otsiti vasteid lähteandmebaasi veergudelt. Veergude kirjeldamisel ja kaardistamisel kasutati ettevõtte tarkvaraarendaja abi, kuna analüüsi käigus selgus, et andmebaaside dokumentatsioon puudub.

Kaardistuse põhjal koostati SQL laused andmete üleviimiseks. Igale sihtandmebaasi sisestatavale tabelile koostati eraldi fail, mis sisaldab andmesiirdeks vajalikke SQL lauseid. Andmete ülekandmiseks koostati kokku kuus SQL-lausetega skripti faili, mis sisaldavad kokku 49 SQL lauset. Autori arvutis andmesiiret katsetades võttis kõigi nende skriptide täitmine umbes 19 minutit ja 45 sekundit. SQL-lausetega skriptfailidega saab tutvuda GitLabis aadressil: <https://gitlab.com/loputoo/Kolimine>

Töö käigus analüüsi ka sihtandmebaasi disaini probleeme ja tehti parandusettepanekuid. Autor leidis sihtandmebaasi tabelite disainis 20 erinevat tüüpi probleemi.

Lõpetuseks saab öelda, et töö oli edukas, väljatöötatud lahendus toimis. Ühe andmebaasisüsteemi andmebaasist teise andmebaasisüsteemi andmebaasi andmete ülekandmise realiseerimine õnnestus. Skriptid SQL-lausetega on ettevõtte tarkvaraarendaja masinas testandmetega testitud. Selle käigus veenduti ülekande tulemuse õigsuses. Tegelik andmesiire toimub peale lõputöö esitamist.

Kasutatud kirjandus

- [1] Wikipedia, the free encyclopedia, „ANSI-SPARC Architecture,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/ANSI-SPARC_Architecture. [Kasutatud 17.05.2022].
- [2] Wikipeedia vaba entsüklopeedia, „Microsoft SQL Server,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Microsoft_SQL_Server. [Kasutatud 25.03.2022].
- [3] The PostgreSQL Global Development Group, „Postgresql,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/>. [Kasutatud 02.04.2022].
- [4] P. Loshin ja J. Sirk, „TechTarget. Structured Query Language (SQL),“ 02 2022. [Võrgumaterjal]. Available: <https://www.techtarget.com/searchdatamanagement/definition/SQL>. [Kasutatud 03.04.2022].
- [5] Wikipeedia Vaba entsüklopeedia, „Unicode,“ [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Unicode>. [Kasutatud 08.04.2022].
- [6] J. Morris, Practical data migration, Bcs Learning & Development Lim, 2012.
- [7] AltexSoft, „Data Migration: Process, Types, and Golden Rules to Follow,“ Altexsoft, 23.11.2020. [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/data-migration/>. [Kasutatud 01.04.2022].
- [8] Sumo Logic, „Application Migration,“ [Võrgumaterjal]. Available: <https://www.sumologic.com/glossary/application-migration/>. [Kasutatud 14.05.2022].
- [9] Sumo Logic, „DevOps and Security Glossary Terms. Cloud Migration,“ [Võrgumaterjal]. Available: <https://www.sumologic.com/glossary/cloud-migration/>. [Kasutatud 15.05.2022].
- [10] K. Prohorchik, „A data migration strategy in six steps,“ 22.04.2021. [Võrgumaterjal]. Available: <https://www.itransition.com/blog/data-migration-strategy>. [Kasutatud 01.04.2022].

- [11] Talend, Inc, „What is Data Migration? Guide to Data Migration Tools and Planning,“ [Võrgumaterjal]. Available: <https://www.stitchdata.com/resources/data-migration>. [Kasutatud 16.05.2022].
- [12] Euroopa Parlament ja Nõukogu, „Isikuandmete kaitse üldmäärus,“ 27.04.2016. [Võrgumaterjal]. Available: <https://eur-lex.europa.eu/legal-content/ET/TXT/HTML/?uri=CELEX:02016R0679-20160504>. [Kasutatud 17.05.2022].
- [13] Microsoft, „SQL Server Integration Services,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15>. [Kasutatud 10.05.2022].
- [14] Talend, „Data integration,“ [Võrgumaterjal]. Available: <https://www.talend.com/products/integrate-data/>. [Kasutatud 20.05.2022].
- [15] Innovative Routines International (IRI), Inc., „Unlock the Business Value of Application Data,“ [Võrgumaterjal]. Available: <https://www.iri.com/products/nextform>. [Kasutatud 20.05.2022].
- [16] Snaplogic, „On-premises Snaplex/Groundplex,“ [Võrgumaterjal]. Available: <https://docs-snaplogic.atlassian.net/wiki/spaces/SD/pages/1438278/On-premises+Snaplex+Groundplex>. [Kasutatud 20.05.2022].
- [17] Matillion Ltd, „Modern analytics made easy with,“ [Võrgumaterjal]. Available: <https://www.matillion.com/>. [Kasutatud 20.05.2022].
- [18] Amazon Web Services, Inc., „AWS Data Pipeline. Easily automate the movement and transformation of data,“ [Võrgumaterjal]. Available: <https://aws.amazon.com/datapipeline/>. [Kasutatud 20.05.2022].
- [19] B. Karwin, SQL Antipatterns Avoiding the Pitfalls of Database Programming, Raleigh(North Carolina), Dallas(Texas): The Pragmatic Bookself, 2010.
- [20] P. Factor, „SQL Code Smells,“ 2014. [Võrgumaterjal]. Available: <https://assets.red-gate.com/community/books/sql-code-smells.pdf>. [Kasutatud 07.04.2022].
- [21] M. Blaha, „A retrospective on industrial database reverse engineering projects - part 2,“ Proceedings Eighth Working Conference on Reverse Engineering, 2001, pp. 147-153.
- [22] A. Lapõnin, „Andmebaasi loogilise ja füüsilise disaini antimustrite esinemine mõnedes vaba tarkvara poolt kasutatavates SQL-andmebaasides,“ 10 06 2015.

- [Võrgumaterjal]. Available: <https://digikogu.taltech.ee/et/Item/5876763e-b6a7-42cc-a5e1-68123916c884>. [Kasutatud 01.04.2022].
- [23] E. Eessaar, „The Usage of Declarative Integrity Constraints in the SQL Databases of Some Existing Software.“ %1 *Software Engineering and Algorithms, Vol 1: 10th Computer Science On-line Conference 2021*, Springer International Publishing, 2021, pp. 375-390.
- [24] Django Oscar kogukond, „Oscar is an open-source ecommerce framework for Django,“ [Võrgumaterjal]. Available: <http://oscarcommerce.com/>. [Kasutatud 02.05.2022].
- [25] Postgres, „pgAdmin 4,“ [Võrgumaterjal]. Available: <https://www.pgadmin.org/docs/pgadmin4/6.8/index.html>. [Kasutatud 2022 03 15].
- [26] JetBrains, „DataGrip,“ [Võrgumaterjal]. Available: <https://www.jetbrains.com/datagrip/>. [Kasutatud 01.05.2022].
- [27] Microsoft, „Download SQL Server Management Studio (SSMS),“ 08 03 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>. [Kasutatud 15.03.2022].
- [28] DBSofts, „ESF Database Migration Toolkit,“ [Võrgumaterjal]. Available: <https://www.dbsofts.com/>. [Kasutatud 20.03.2022].
- [29] Wikipedia, the free encyclopedia, „Analytic hierarchy process,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Analytic_hierarchy_process. [Kasutatud 21.05.2022].
- [30] Wikipedia, the free encyclopedia, „Source lines of code,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Source_lines_of_code. [Kasutatud 08.05.2022].
- [31] AlDanial, „cloc, Count Lines of Code,“ 05.12.2021. [Võrgumaterjal]. Available: <https://github.com/AlDanial/cloc>. [Kasutatud 18.05.2022].
- [32] E. Maido, „Mõned failide salvestamise disainimustrid SQL-andmebaasi kasutavate andmebaasirakenduste jaoks,“ 08.06.2021. [Võrgumaterjal]. Available: <https://digikogu.taltech.ee/et/Item/409ee372-2a60-4b69-8cd7-d0683b887c84>. [Kasutatud 15.05.2022].

- [33] C. v. I. J. G. Russell Sears, „To BLOB or Not To BLOB: Large Object Storage in a Database or a Filesystem?“, 2006. [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/research/wp-content/uploads/2006/04/tr-2006-45.pdf>. [Kasutatud 2. 05.2022].
- [34] Microsoft, „FILESTREAM (SQL Server)“, 29.11.2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-US/sql/relational-databases/blob/filestream-sql-server?view=sql-server-ver15>. [Kasutatud 21.05.2022].
- [35] J. Newcum, „A Universal Person and Organization Data Model“, TDAN.com, 07 2002. [Võrgumaterjal]. Available: <https://tdan.com/a-universal-person-and-organization-data-model/5014>. [Kasutatud 21.05.2022].
- [36] A.-S. Vellemaa, „Mõned disainimustrid klassifikaatorite esitamiseks SQL andmebaasides“, 12.06.2015. [Võrgumaterjal]. Available: <https://digikogu.taltech.ee/et/Item/fb7e7116-a202-4fd3-a7bc-be5134d9a975>. [Kasutatud 15.05.2022].
- [37] Wikipedia, the free encyclopedia, „List of ISO 639-1 codes“, [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes. [Kasutatud 21.05.2022].
- [38] Wikipedia, the free encyclopedia, „ISO/IEC 5218“, [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/ISO/IEC_5218. [Kasutatud 12.05.2022].
- [39] Erinevad autorid, „I need to store postal codes in a database. How big should the column be?“, STACK OVERFLOW, [Võrgumaterjal]. Available: <https://stackoverflow.com/questions/325041/i-need-to-store-postal-codes-in-a-database-how-big-should-the-column-be>. [Kasutatud 21.05.2022].
- [40] RFC Editor, „RFC 3696, "Application Techniques for Checking and Transformation of Names",“ 02 2004. [Võrgumaterjal]. Available: https://www.rfc-editor.org/errata_search.php?rfc=3696&eid=1690. [Kasutatud 16.05.2022].
- [41] „Top 5 People With the Longest Names“, History Rundown, [Võrgumaterjal]. Available: <https://www.historyrundown.com/top-5-people-with-the-longest-names/>. [Kasutatud 16.05.2022].

- [42] Wikipedia, the free encyclopedia, „List of long place names,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/List_of_long_place_names. [Kasutatud 21.05.2022].
- [43] M. Jõgi, „Mõned disainilahendused isikunimede hoidmiseks SQL-andmebaasides,“ [Võrgumaterjal]. Available: <https://digikogu.taltech.ee/et/Item/ab7e8f7e-ce28-4cc1-bb78-40c624d19bf6>. [Kasutatud 25.04.2022].
- [44] J. Atwood, „The Case For Case Insensitivity,“ CODING HORROR programming and human factors, 04.12.2005. [Võrgumaterjal]. Available: <https://blog.codinghorror.com/the-case-for-case-insensitivity/>. [Kasutatud 21.05.2022].
- [45] J. Yao, „Script to Delete Data from SQL Server Tables with Foreign Key Constraints,“ 15.10.2015. [Võrgumaterjal]. Available: <https://www.mssqltips.com/sqlservertip/4059/script-to-delete-data-from-sql-server-tables-with-foreign-key-constraints/>. [Kasutatud 15.05.2022].
- [46] Microsoft , „SET IDENTITY_INSERT (Transact-SQL),“ 30.1.2012. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/statements/set-identity-insert-transact-sql?view=sql-server-ver15>. [Kasutatud 30.04.2022].
- [47] Wikipedia, the free encyclopedia, „Telephone numbers in Lithuania,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Telephone_numbers_in_Lithuania. [Kasutatud 01.04.2022].
- [48] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Anne Siitan

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „E-poe andmete migreerimine PostgreSQL andmebaasist Microsoft SQL Serveri andmebaasi“, mille juhendaja on Erki Eessaar
 - 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.

18.05.2022

¹ 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 tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

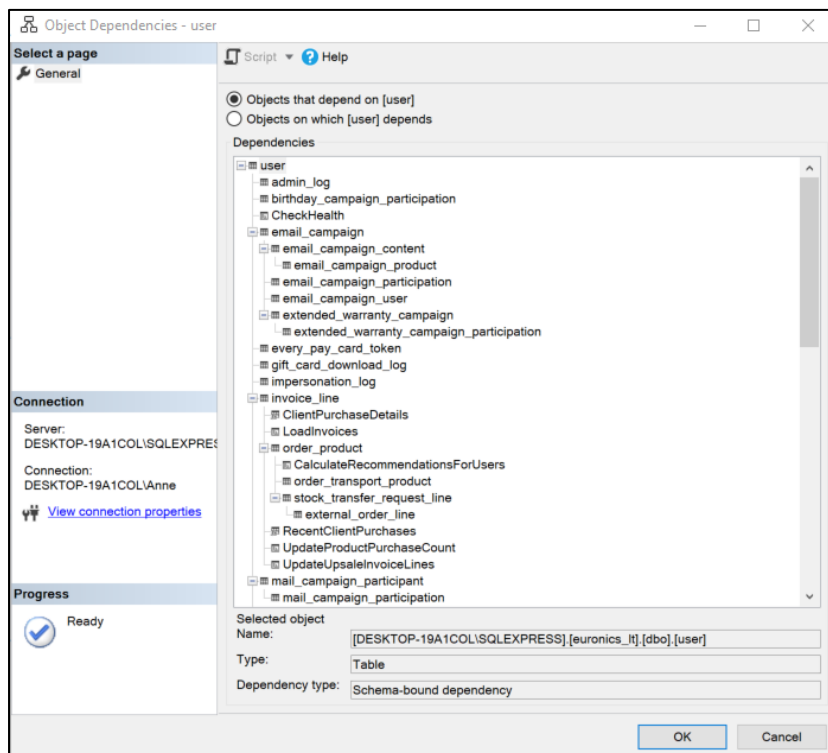
Lisa 2 – Näide *User* tabeli kohta tehtud vastavustabelist

Excelis

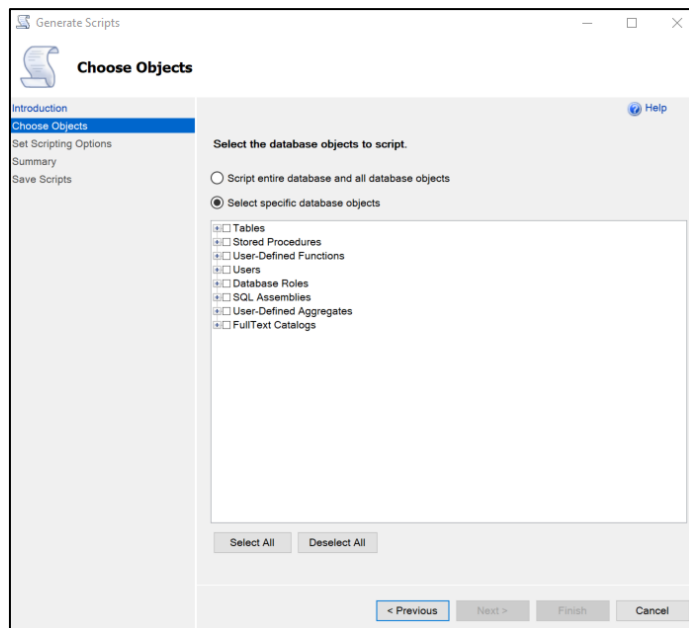
Eesti keelne tõlge	Uus süsteem user tabel	DATA	IS_	Vana süsteem (tabel)	(veerg)	DATA_TYPE	kommentaarid
Kauba kättesaaja isiku eesnimi	shipping_first_name	nvarchar	YES	address_useraddress	first_name	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja isiku perenimi	shipping_last_name	nvarchar	YES	address_useraddress	last_name	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja isiku sünnikuupäev	shipping_date_of_birth	date	YES	Neil pole, tuleb NULL			
Kauba kättesaaja mobiiltelefoninumber	shipping_mobile_phone	nvarchar	YES	address_useraddress	phone_number (ainult mobilid)	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja telefoninumber	shipping_phone	nvarchar	YES	address_useraddress	phone_number (lautelefone nr)	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja faksinumber	shipping_fax	nvarchar	YES	Neil pole, tuleb NULL			
Kauba kättesaaja e-posti aadress	shipping_email	nvarchar	YES	accounts_user	email		
Kauba kättesaaja riik	shipping_country	nvarchar	YES	address_useraddress	country_id	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja maakond	shipping_county	nvarchar	YES	address_useraddress	state	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja posti aadress	shipping_address	nvarchar	YES	address_useraddress	line1 + line 2	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja sihtnumber	shipping_postal_code	nvarchar	YES	address_useraddress	postcoode	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)
Kauba kättesaaja Ettevõtte nimi	shipping_company_name	nvarchar	YES	address_useraddress	company_name	charater varying	Kui is_default_for_shipping = true siis sealt, muidu kõige uuemalt realt (date_created järgi kõige uuem)

Lisa 3 - SSMS kaudu andmebaasi tabelite tühjendamise juhend

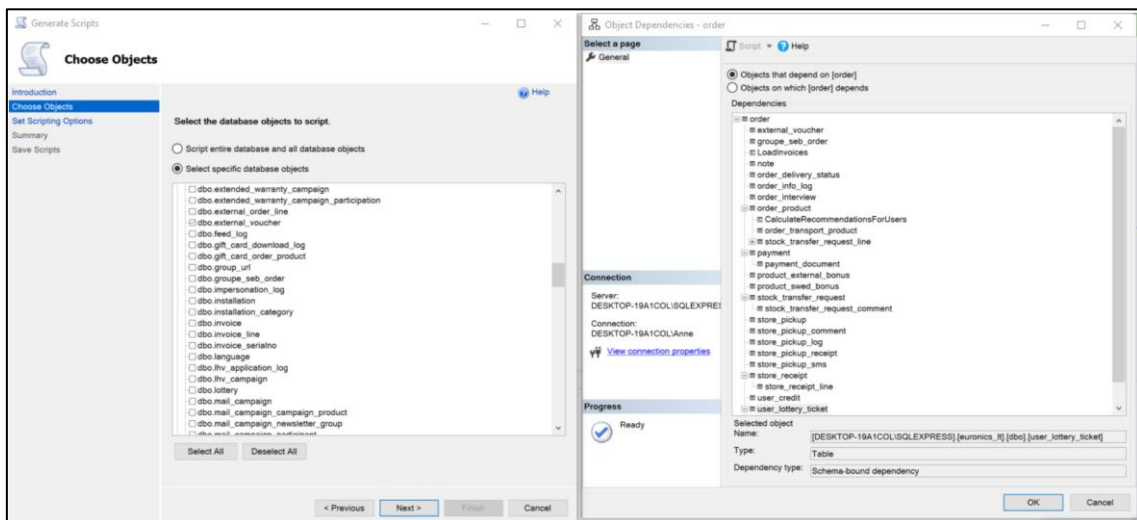
Juhend põhineb tarkvara versioonil Microsoft SQL Server Management Studio 18.11.1. Kõigepealt tuli vaadata üle täidetavate tabelite sõltuvused. Selleks tuli *Object Explorer* aknas tabelil teha paremkliik ja valida „*View Dependencies*“. Avaneb uus aken „*Object Dependencies*“. Valida „*Objects that depend on*“. Avades kõik plussmärgid kuvatakse puustruktuuris tabelid, mis on valitud tabelist sõltuvuses.



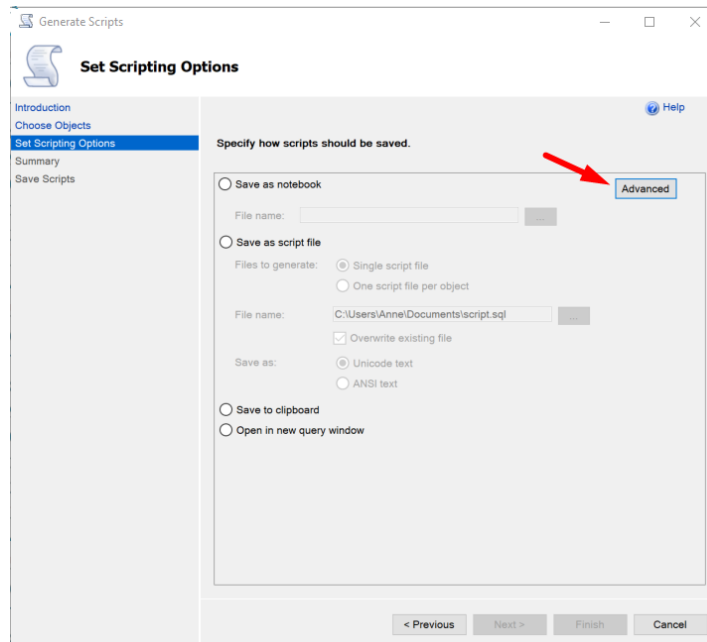
Kuna soov oli teha mitu tabelit tühjaks, siis tuleb *Object Explorer* aknas andmebaasi nime peal teha paremkliik ja valida „*Tasks*“ ja „*Generate Scripts*“. Edasi tuleb valida „*Select specific database objects*“ ja avada plussmärgist „*Tables*“ ja „*Stored Procedures*“.



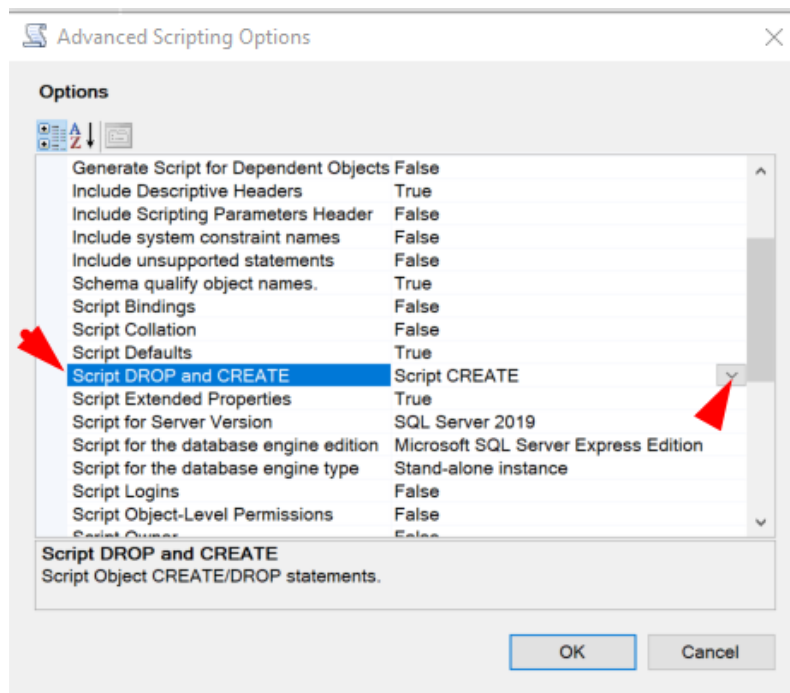
Selles aknas tuleb ära linnutada kõik need tabelid ja protseduurid, mis on sõltuvuste aknas puustruktuurina esitatud. Nii tuleb teha kõikide tabelitega, mida peab tühjaks tegema.



Kui vajalikud tabelid on valitud, siis „Set Scripting Options“ aknas tuleb valida „Advanced“.



Tuleb otsida üles rida „Script DROP and CEATE“ ja valida nimekirjast „Drop and Create“. Seejärel tuleb otsida rida „Script indexes“ ja märkida sinna „True“.



Antud töö puhul tähendas *User* tabeli tühjendamine tegelikult kõigi nelja sihtandmebaasi tabeli tühjendamist, kuna, tabelil on sõltuvused kõigi neljaga tabeliga.

SSMS'ga *User* tabeli tühjendamiseks genereeritud skriptifail oli 4650 rida pikk (koos tühikute ja kommentaaridega). Allpool on väike näide genereeritud koodist.

```
237 ALTER TABLE [dbo].[note] DROP CONSTRAINT [FK_note_user]
238 GO
239 ALTER TABLE [dbo].[note] DROP CONSTRAINT [FK_note_order]
240 GO
241 ALTER TABLE [dbo].[note] DROP CONSTRAINT [FK_note_classifier_value]
242 GO
243 ALTER TABLE [dbo].[news_tag] DROP CONSTRAINT [FK_news_tag_news_tag_category]
244 GO
245 ALTER TABLE [dbo].[news_tag] DROP CONSTRAINT [FK_news_tag_news]
246 GO
247 ALTER TABLE [dbo].[news_revision] DROP CONSTRAINT [FK_news_id]
248 GO
249 ALTER TABLE [dbo].[news_product] DROP CONSTRAINT [FK_news_product_product]
250 GO
251 ALTER TABLE [dbo].[news_product] DROP CONSTRAINT [FK_news_product_news]
252 GO
253 ALTER TABLE [dbo].[news_comment] DROP CONSTRAINT [FK_news_comment_user]
254 GO
```