



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Robertas Jankauskas

DUOMENŲ IŠGAVIMO, PERTVARKYMO IR ĮKĖLIMO

GREITAVEIKOS TYRIMAS

RESEARCH OF DATA EXTRACTION, TRANSFORMATION AND

LOADING SPEED

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004

Duomenų gavybos technologijos specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2017

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Robertas Jankauskas

**DUOMENŲ IŠGAVIMO, PERTVARKYMO IR ĮKĖLIMO
GREITAVEIKOS TYRIMAS**
**RESEARCH OF DATA EXTRACTION, TRANSFORMATION AND
LOADING SPEED**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004

Duomenų gavybos technologijos specializacija

Informatikos inžinerijos studijų kryptis

Vadovas	<u>Prof. Dmitrij Šešok</u> (Pedag. vardas, vardas, pavardė)	<u>(Parašas)</u>	<u>(Data)</u>
Konsultantas	<u>Dr. Vaida Buivydienė</u> (Pedag. vardas, vardas, pavardė)	<u>(Parašas)</u>	<u>(Data)</u>
Konsultantas	<u></u> (Pedag. vardas, vardas, pavardė)	<u>(Parašas)</u>	<u>(Data)</u>

Vilnius, 2017

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU

Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

BAIGIAMOJO MAGISTRO DARBO (PROJEKTO)

UŽDUOTIS

.....Nr.

Vilnius

Studentui (ei) Robertas Jankauskas

(Vardas, pavardė)

Baigiamojo darbo (projekto) tema: Duomenų išgavimo, pertvarkymo ir įkėlimo greಿತaveikos tyrimas

patvirtinta 2016 m. gruodžio 20 d. deкano potvarkiu Nr. 417fm

Baigiamojo darbo (projekto) užbaigimo terminas 2017 m. gegužės 29 d.

BAIGIAMOJO DARBO (PROJEKTO) UŽDUOTIS:

Duomenų saugojimo duomenų saugykloje analizė, ETL proceso analizė, ETL metodologijų, technologijų ir įrankių analizė, ETL proceso tyrimas naudojant skirtingas technologijas ir metodologijas

Baigiamojo magistro darbo (projekto) konsultantai:

(Pedag. vardas, vardas, pavardė)

Vadovas

(Parašas)

prof. Dmitrij Šešok

(Pedag. vardas, vardas, pavardė)

Užduotį gavau

(Parašas)

Robertas Jankauskas

(Vardas, pavardė)

2016-10-28

(Data)

Anotacija

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių technologijų katedra

ISBN ISSN
Egz. sk.
Data-.....-.....

Antrosios pakopos studijų **Informacinių technologijų** programos magistro baigiamasis darbas 4

Pavadinimas **Duomenų išgavimo, pertvarkymo ir įkėlimo greیتaveikos tyrimas**
Autorius **Robertas Jankauskas**
Vadovas **Dmitrij Šešok**

Kalba: lietuvių

Anotacija

Šio magistro baigiamojo darbo tikslas yra ištirti galimus duomenų išgavimo, pertvarkymo ir įkėlimo metodus bei sistemų teikiamas technologijas, kurios gali būti panaudotos duomenų išgavimo, pertvarkymo ir įkėlimo proceso metu iš transakcinės sistemos į analitinę sistemą. Analitinėje dalyje yra apžvelgiamos OLAP duomenų saugojimo sistemos, nagrinėjamas duomenų išgavimo, pertvarkymo ir įkėlimo procesas, įrankiai, metodai ir sistemų teikiamos technologijos, leidžiančios atlikti duomenų perkėlimą tarp esamų sistemų. Praktinėje dalyje yra atliekamas tyrimas imituojant duomenų išgavimo, pertvarkymo ir įkėlimo procesą tarp OLTP ir OLAP sistemų, kur OLTP sistema yra realizuota su „Oracle“ duomenų bazių valdymo sistema, o OLAP sistema realizuota su „Microsoft SQL Server“ duomenų bazių valdymo sistema, pasirinkus vieną iš duomenų išgavimo, pertvarkymo ir įkėlimo įrankių ir nagrinėtas technologijas. Pagal gautus rezultatus pateikiamos išvados ir rekomendacijos, kurias technologijas yra geriausia taikyti duomenų išgavimo, pertvarkymo ir įkėlimo proceso realizavimui.

Darbą sudaro 7 dalys: įvadas, duomenų saugojimo duomenų saugykloje metodologijos analizė, ETL procesas, duomenų įkėlimo metodai ETL procese, technologijos ir įrankiai, ETL proceso tyrimas, išvados ir siūlymai, literatūros sąrašas.

Darbo apimtis – 69 psl. teksto be priedų, 35 iliustr., 4 lent., 35 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

Prasminiai žodžiai: ETL, duomenų saugykla, „Oracle“, „Microsoft SQL Server“, SSIS, DBVS, OLTP, OLAP.

Anotacija anglų kalba

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Technologies

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Information Technologies** study programme Master Graduation Thesis

Title **Research of data extraction, transformation and loading speed**

Author **Robertas Jankauskas**

Academic supervisor **Dmitrij Šešok**

Thesis language: Lithuanian

Annotation

The main goal of master thesis is to research for possible data extraction, transformation and loading methods and system's provided technologies which can be used in data extraction, transformation and loading process from transactional to analytical system. On analytical part OLAP data storage systems are reviewed, data extraction, transformation and loading process tools, methods and system's provided technologies are considered which allow to transfer data between systems. On practical part research of data extraction, transformation and loading process is performed between OLTP and OLAP systems where OLTP system is implemented with „Oracle“ database management system, whereas OLAP system is implemented with „Microsoft SQL Server“ database management system choosing one of data extraction, transformation and loading tools and considered system's provided technologies. According to given results conclusion and recommendations are made which technologies are best to use in data extraction, transformation and loading process implementation.

Structure: introduction, methodology analysis of storing data in data warehouse, ETL process, data loading methods in ETL process, technologies and tools, research of ETL process, conclusions and suggestion, references.

Thesis consists of: 69 p. text without appendixes, 35 pictures, 4 tables, 35 bibliographical entries.

Appendixes included.

Keywords: ETL, data warehouse, „Oracle“, „Microsoft SQL Server“, SSIS, DBMS, OLTP, OLAP.

Sąžiningumo deklaracija

Vilniaus Gedimino technikos universiteto
egzaminų sesijų ir baigiamųjų darbų rengimo
bei gynimo organizavimo tvarkos aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Robertas Jankauskas, 20113557

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinės technologijos, DGTfm-15

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

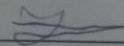
SĄŽININGUMO DEKLARACIJA

2017 m. gegužės 18 d.

Patvirtinu, kad mano baigiamasis darbas tema „Duomenų išgavimo, pertvarkymo ir įkėlimo greیتaveikos tyrimas“ patvirtintas 2016 m. gruodžio 20 d. dekanų potvarkiu Nr. 417fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas doc. dr. Dmitrij Šešok.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).



(Parašas)

Robertas Jankauskas

(Vardas ir pavardė)

Turinys

Įvadas.....	12
1. Duomenų saugojimo duomenų saugykloje metodologijos analizė	14
1.1. OLTP sistemos ir duomenų saugykla	14
1.2. Duomenų saugyklos ypatumai	15
1.3. Duomenų saugyklos architektūra.....	16
1.4. Duomenų saugyklos modeliavimas	18
2. ETL procesas	21
2.1. ETL svarba.....	21
2.2. ETL etapai.....	21
2.2.1. Duomenų išgavimas	21
2.2.2. Duomenų pertvarkymas	22
2.2.3. Duomenų įkėlimas.....	24
2.3. ETL sparta duomenų saugyklose	24
3. ETL metodai, technologijos bei įrankiai	26
3.1. Duomenų įkėlimo metodai ETL procese	26
3.2. Sistemų teikiamos technologijos ETL procese	27
3.2.1. Pakitusių duomenų registravimas	27
3.2.2. Duomenų bazės lentelių panaudojimas	29
3.2.3. Lygiagretus ETL procesas.....	31
3.2.4. Masinis duomenų įkėlimas.....	32
3.2.5. Indeksų panaudojimo galimybės	33
3.2.6. Vaizdinių panaudojimas	35
3.2.7. Duomenų suspaudimo metodika	36
3.2.8. Saugomų procedūrų panaudojimas	37
3.3. ETL įrankiai	38
3.3.1. „Microsoft SQL Server SSIS“ paketas.....	38
3.3.2. „Talend Open Studio“	39
3.3.3. „Informatica PowerCenter Express“	40
3.3.4. „Oracle Warehouse Builder“	41
3.3.5. „Oracle Data Integrator“	41
3.3.6. „IBM InfoSphere DataStage“	42
4. ETL proceso tyrimas	44

4.2. Duomenų įrašymo proceso tyrimas	45
4.2.1. Pasirinktos technologijos tyrimui	46
4.2.2. Duomenų įkėlimo parinkčių pasirinkimas	47
4.2.3. Nuoseklus duomenų įrašymas	49
4.2.4. Lygiagretus duomenų įrašymas	51
4.2.5. Duomenų įrašymas naudojant lentelių tipo struktūras	52
4.2.6. Duomenų įrašymas naudojant padalijimo metodologiją	52
4.2.7. Duomenų įrašymas naudojant laikinas lenteles	53
4.2.8. Duomenų įrašymas naudojant indeksuotus vaizdinius	53
4.2.9. Duomenų įrašymas naudojant atmintį taupančias lenteles	53
4.2.10. Duomenų įrašymas naudojant atmintį taupančių lentelių tipo struktūras	54
4.2.11. Duomenų įrašymas naudojant duomenų suspaudimo technologiją	54
4.3. Duomenų įrašymo ir atnaujinimo proceso tyrimas	54
4.4. ETL procesų tyrimų rezultatai	55
Išvados ir siūlymai	64
Literatūros ir šaltinių sąrašas	66
Priedai	70
A priedas. Pažyma dėl rezultatų panaudojimo VGTU	70
B priedas. Pažyma dėl dalyvavimo 20-ojoje jaunųjų mokslininkų konferencijoje	71

Paveikslukų sąrašas

1.1 pav. Duomenų saugyklos architektūra [5]	16
1.2 pav. „Žvaigždės“ schema [17]	19
1.3 pav. Duomenų kubas [17]	20
2.1 pav. Daugiapakopis duomenų keitimas [7]	23
2.2 pav. Grandininis duomenų keitimas [7]	24
3.1 pav. Duomenų pasikeitimų registravimo schema [4]	28
4.1 pav. Reliacinė užsakymų duomenų bazės schema	44
4.2 pav. Užsakymų duomenų saugyklos schema	45
4.3 pav. „OLE DB Destination“ parinktys	47
4.4 pav. Duomenų įkėlimo užduočių palyginimas	48
4.5 pav. Nuoseklus duomenų įrašymas	49
4.6 pav. Duomenų srauto įrašymo procesas	49
4.7 pav. Duomenų įrašymas į faktų lentelę	50
4.8 pav. Kiekvieno antrinio duomenų įrašymo proceso trukmė	50
4.9 pav. Viso duomenų įrašymo proceso trukmė	50
4.10 pav. Nuoseklus duomenų įkėlimas į tarpines lenteles	51
4.11 pav. Nuoseklus duomenų įkėlimas į duomenų saugyklos lenteles	51
4.12 pav. Lygiagretus duomenų įkėlimas į tarpines lenteles	52
4.13 pav. Lygiagretus duomenų įkėlimas į duomenų saugyklos lenteles	52
4.14 pav. Duomenų įkėlimas naudojant lentelių tipo struktūras	52
4.15 pav. Duomenų įkėlimas į padalytą faktų lentelę	53
4.16 pav. Duomenų įkėlimas į atmintį taupančias lenteles	54
4.17 pav. Duomenų įkėlimo proceso vykdymas naudojant SQL užduotis	55
4.18 pav. Nuoseklaus ir lygiagretaus ETL procesų rezultatai nenaudojant tarpinių lentelių	55
4.19 pav. Nuoseklaus ir lygiagretaus ETL procesų rezultatai naudojant tarpines lenteles	56
4.20 pav. ETL proceso rezultatai naudojant atmintį taupančių lentelių tipo struktūras	56
4.21 pav. ETL proceso rezultatai taikant lentelių tipo struktūras nenaudojant tarpinių lentelių	58

4.22 pav. ETL proceso rezultatai naudojant atmintį taupančių lentelių tipų struktūras.....	59
4.23 pav. ETL proceso rezultatai naudojant lentelių tipo struktūras.....	60
4.24 pav. Duomenų įrašymo į dimensijų lenteles rezultatai	60
4.25 pav. ETL proceso rezultatai naudojant skirtingų indeksų variantus tarpinėse lentelėse	61
4.26 pav. ETL proceso rezultatai naudojant duomenų suspaudimo technologiją.....	62
4.27 pav. ETL proceso rezultatai naudojant atmintį taupančias lenteles	62
4.28 pav. ETL proceso rezultatai naudojant skirtingas lentelių rūšis	63
4.29 pav. Duomenų įkėlimo rezultatai į faktų lentelę naudojant padalijimo metodologiją	63

Lentelių sąrašas

1.1 lentelė. Skirtumai tarp OLTP ir OLAP sistemų [8, 9]	14
1.2 lentelė. OLAP modelių palyginimas [5]	17
3.1 lentelė. ETL įrankių palyginimas [3, 35]	43
4.1 lentelė. Tyrimui pasirinktos technologijos.....	46

Santrumpų sąrašas

Darbas	Programa, vykdanči procedūras nustatytu laiku
Denormalizacija	Procesas, kurio metu atliekamas lentelių struktūrų pertvarkymas, kad lentelės būtų kuo mažiau susijusios dėl duomenų užklausos spartos gerinimo
DBVS	Duomenų bazių valdymo sistema
DDL	Duomenų struktūros apibrėžties žodynas (angl. <i>Data Definition Language</i>).
ETL	Duomenų išgavimas, pertvarkymas ir įkėlimas (angl. <i>Extract, Transform, Load</i>).
HOLAP	Hibridinis analitinis duomenų apdorojimas tinkle (angl. <i>Hybrid On-Line Analytical Processing</i>).
I/O	Įvestis ir išvestis (angl. <i>Input/Output</i>)
IBM	JAV kompiuterių gamintoja, programinės bei techninės įrangos bendrovė (angl. <i>International Business Machines Corporation</i>)
IT	Informacinės technologijos
MOLAP	daugiamatis analitinis duomenų apdorojimas tinkle (angl. <i>Multidimensional On-Line Analytical Processing</i>).
Normalizacija	Procesas, kurio metu sudaromos duomenų bazių lentelių struktūros taip, kad jose saugojami įrašai nesidubliuotų ir būtų nepriklausomi nuo kitų įrašų, esančių lentelėse, išskyrus tą lentelę identifikuojant į atributą
ODBC	Programavimo sąsaja, reikalinga prisijungimui prie kitų duomenų bazių (angl. <i>Open Database Connection</i>).
OLAP	Analitinis duomenų apdorojimas tinkle (angl. <i>On-Line Analytical Processing</i>)
OLE DB	Taikomoji programavimo sąsaja, leidžianti prieiti prie duomenų iš skirtingų šaltinių (angl. <i>Object Linking and Embedding Database</i>).
OLTP	Tiesioginių operacijų apdorojimas tinkle (angl. <i>On-Line Transactional Processing</i>).
RDBVS	Reliacinė duomenų bazių valdymo sistema
ROLAP	Reliacinis analitinis duomenų apdorojimas tinkle (angl. <i>Relational On-Line Analytical Processing</i>).
Sesija	Laikotarpis nuo prisijungimo pradžios iki pabaigos
SOA	Į paslaugas orientuota architektūra (angl. <i>Service Oriented Architecture</i>).
SQL	Standartinė užklausų kalba (angl. <i>Standard Query Language</i>).
T-SQL	Tiesioginės operacijos standartinė užklausų kalba (angl. <i>Transact - Standard Query Language</i>)
Trigeris	DBVS programinis kodas, kuris yra vykdomas esant atitinkamai duomenų manipuliavimo operacijai

Įvadas

Temos aktualumas

Organizacijoms dažnai tenka saugoti istorinius duomenis, kuriais naudojantis būtų galima priimti tam tikrus sprendimus. Jiems saugoti yra naudojamos specifinės duomenų bazės, vadinamos duomenų saugyklomis, kurios teikia verslo analitikams galimybes organizuoti, analizuoti bei panaudoti šiuos duomenis sprendimams priimti. Duomenys konkrečioje organizacijoje yra surenkami iš skirtingų duomenų šaltinių, tuomet atliekami šių duomenų pertvarkymo, koregavimo veiksmai ir tik tada vykdomas duomenų įkėlimo procesas į duomenų saugyklą. Naudojantis šiais duomenimis sukuriama ataskaitos galutiniam vartotojams, kuriais galima atlikti duomenų analizę be jokių techninių žinių. Vis dėlto, ar šie duomenys užima daug vietos atmintyje, ar jų skaičius yra didelis, įkelti juos į duomenų saugyklą iš skirtingų duomenų šaltinių užtrunka daug laiko, ypač, jei duomenų saugykla nėra tinkamai suprojektuota.

Tiek duomenų saugyklose, tiek ir kitose duomenų bazėse naudojamas duomenų išgavimo, pertvarkymo bei įkėlimo (ETL) procesas, kurio metu išgaunami duomenys iš skirtingų duomenų šaltinių, vykdomas duomenų išvalymas, filtravimas, tikrinimas, susiejimas, verslo taisyklių pritaikymas ir įkėlimas į duomenų saugyklą. Šis procesas yra sudėtingas ir reikalauja nemažai laiko ir pastangų, kad duomenys būtų įkelti į duomenų saugyklą, todėl organizacijoms svarbu pasirinkti, koku būdu jie norės atlikti šį procesą, bei įrankius, kurie leistų kuo trumpesniu laiku įkelti duomenis.

Darbo problematika

Šiuo metu yra sukurta daugybė technologijų, leidžiančių išgauti, pertvarkyti ir įkelti duomenis iš skirtingų duomenų šaltinių, tačiau nėra vieningo sprendimo, kurias iš esamų technologijų yra geriausia taikyti pagal tam tikrus apribojimus ar taisykles, todėl šiame darbe siekiama išnagrinėti esamas technologijas ir jas palyginti tarpusavyje, kurios iš jų leidžia perkelti duomenis pačiu sparčiausiu būdu.

Darbo tikslas

Šio magistro baigiamojo darbo tikslas yra ištirti galimus ETL būdus, sistemų teikiamas technologijas ir įrankius, kurie naudojami duomenims įkelti į duomenų saugyklą.

Darbo uždaviniai

Siekiant pasiekti tikslą reikia atlikti kelis uždavinius:

1. Išanalizuoti duomenų saugyklos metodologiją, duomenų saugojimo principus šioje sistemoje.

2. Detaliai išnagrinėti ETL procesą.
3. Išnagrinėti technologijas ir palyginti metodus bei įrankius, kurie naudojami ETL procese iš operacinių duomenų bazių į duomenų saugyklą.
4. Palyginti ETL proceso spartą taikant skirtingus duomenų įkėlimo metodus bei technologijas.

Tiriamąo darbo objektas.

Šio baigiamojo darbo tiriamasis objektas yra ETL procesas tarp OLTP ir OLAP sistemų.

Mokslinis naujumas.

Duomenų saugykla yra ganėtinai nauja duomenų saugojimo technologija, kuri atsirado maždaug devinto dešimtmečio pradžioje, tačiau šiais laikais duomenų saugykla yra taikoma ne tik įmonėse, kurios taiko IT technologijas, bet ir aukštosiose mokyklose, kuriose vykdomos finansinės, ekonominės, studijų ir kitos apskaitos. Viena iš šių aukštųjų mokyklų yra VGTU, kurioje yra atliekamas ETL procesas. Šioje institucijoje saugomi finansiniai, studijų ir kiti analizei aktualūs duomenys. Šiame darbe nagrinėjami jau egzistuojantys sprendimai, kuriais galima atlikti ETL procesą. Tam pasitelkta VGTU sukurta užsakymų valdymo sistema, kurioje šios institucijos darbuotojai gali užsisakyti jiems reikalingas prekes.

Praktinė vertė.

Išanalizuotus ETL proceso metodus, sistemų teikiamas technologijas bei įrankius bus galima pritaikyti ne tik VGTU, bet ir kitose organizacijose, kurioje įkeliami duomenys į duomenų saugyklą, jei šios sistemos teikia galimybes taikyti tokias technologijas. ETL greitaveikos tyrimas yra atliekamas iš „Oracle 12c“ į „Microsoft SQL Server“ DBVS, kurių pasirinkimą lėmė šių technologijų naudojimas VGTU, siekiant imituoti ETL procesą skirtingose aplinkose.

Darbo metodai.

Šiame darbe apžvelgiama mokslinė literatūra, elektroniniai leidiniai nagrinėjama tema, dokumentacijos, mokomoji medžiaga. Tiriant ETL procesą panaudoti „Oracle 12c Enterprise Edition Release“ ir „Microsoft SQL Server 2014 Enterprise Edition“ DBVS įrankiai, suprojektuota jau gamybinėje aplinkoje naudojama užsakymų veiklos duomenų bazė „Oracle“ sistemoje, bei naujai suprojektuota duomenų saugykla „Microsoft SQL Server“ aplinkoje.

Darbo aprobacija.

Šis darbas buvo pristatytas 20-ojoje jaunųjų mokslininkų konferencijoje „MOKSLAS – LIETUVOS ATEITIS“ tema „Duomenų užkrovimo greitaveikos tyrimas“.

1. Duomenų saugojimo duomenų saugykloje metodologijos analizė

1.1. OLTP sistemos ir duomenų saugykla

Populiariausios duomenų saugojimo sistemos yra OLTP sistemos. Jose atliekamos kiekvienos dienos organizacijos operacijos, tokios kaip inventORIZACIJA, bankinės operacijos, pirkimų bei pardavimų apskaita ir kt. Vis dėlto OLTP nėra tinkama duomenų analizei, nes tektų surinkti reikalingą informaciją iš skirtingų duomenų šaltinių, o tai yra sudėtingas ir daug kompiuterių išteklių reikalaujantis procesas [11]. Visų pirma, OLTP sistemose saugomi plataus masto duomenys arba ne visi duomenys, be to, skirtingose sistemose tie patys atributai gali turėti skirtingus pavadinimus, būti susiję skirtingais ryšiais, taip pat jose nuolatos turi būti įterpiami, atnaujinami šalinami duomenys realiu laiku, o tai gali apkrauti sistemos darbą. Visų antra, tokios sistemos dirba skirtingose platformose, duomenys yra saugomi skirtingose RDBVS, todėl norint išgauti reikalingą informaciją tenka pasinaudoti patyrusio specialisto paslaugomis, kuris sugebėtų gauti reikiamą informaciją iš skirtingų RDBVS. Visų trečia, OLTP sistemose duomenims gauti tenka naudoti SQL kalba parašytas užklausas. Vartotojui nėra prasmės pačiam rašyti tokias užklausas, todėl geriausia pateikti duomenis tik jam suprantamu formatu, pavyzdžiui, atvaizduoti duomenis ataskaitoje, „Microsoft Excel“ programinėje aplinkoje ir pan. Tokiems duomenims saugoti ir analizuoti yra naudojama duomenų saugykla ir ją papildančios OLAP sistemos. Duomenų saugykla vadinama centrine duomenų baze, kurioje saugomi integruoti duomenys iš skirtingų duomenų šaltinių, naudojamų analizei bei užklausoms vykdyti. Jame saugomi istoriniai bei archyvuoti duomenys, susidedantys iš operacinių duomenų ir kitų duomenų šaltinių, kurie leidžia verslo analitikams atlikti sprendimus konkrečioje organizacijoje. Šie duomenys yra vieni iš svarbiausių organizacijos turtų, kuriais remiantis galima atlikti konkrečios veiklos analizę. Priešingai nei OLTP sistemos, duomenų saugyklos sistemos aptarnauja vartotojus arba konkrečios srities žinovus analizuojant duomenis bei priimant sprendimus konkrečioje veikloje.

Pagrindiniai skirtumai tarp OLTP bei OLAP (duomenų saugyklos) sistemų pavaizduoti 1.1 lentelėje.

1.1 lentelė. Skirtumai tarp OLTP ir OLAP sistemų [8, 9]

Požymis	OLTP	OLAP (duomenų saugykla)
Tikslas	Vykdyti ir kontroliuoti verslo užduotis	Išgauti ir analizuoti duomenis
Duomenų šaltiniai	Operaciniai duomenys	Istoriniai bei archyviniai duomenys iš skirtingų duomenų šaltinių
Duomenų bazės modelis	Reliacinis modelis	„Žvaigždės“ arba „Snaigės“ schema
Duomenų bazės struktūra	Normalinės formos (normalizuota duomenų bazės struktūra)	Denormalizacija (denormalizuota duomenų bazės struktūra)
Duomenų bazės dydis	Dydis svyruoja nuo 100 Mb iki 1 Gb	Dydis svyruoja nuo 100 Gb iki 1 Tb

1.1 lentelės tęsinys

Įrašų skaičius	Keli arba keliasdešimt tūkstančių įrašų	Daugiau nei milijonas įrašų
Duomenų bazės vykdymo sparta	Didelė	Dažniausiai priklauso nuo duomenų kiekio, užklausos sudėtingumo
Užklausų sudėtingumas	Paprastos	Sudėtingos, naudojamos grupinės arba agregavimo funkcijos
Atsarginės kopijos ir duomenų atstatymas	Nuolatiniai. Duomenys negali būti prarasti	OLTP perkrovimas

1.2. Duomenų saugyklos ypatumai

Kiekvienoje organizacijoje duomenų kiekis auga, todėl atsiranda poreikis saugoti šiuos duomenis, juos pasiekti, analizuoti ir pan. Duomenų saugyklos struktūra padeda saugoti tokius duomenis, todėl ji turi tenkinti šiuos reikalavimus, kad vartotojai galėtų greitai ir efektyviai gauti informaciją:

- Verslo intelektikos sistema turi leisti lengvai ir paprastai pasiekti informaciją. Verslo vartotojui informacija turi būti pasiekama taip, kad jam neprireiktų specialių ar minimalių žinių informacijos išgavimui. Taip pat ši informacija turi būti pasiekama kaip įmanoma spartesniu laiku [17].
- Verslo intelektikos sistema turi pateikti konkrečiam vartotojui tik jam aktualią ir reikalingą informaciją. Prieš įkeliant duomenis į duomenų saugyklą iš skirtingų duomenų šaltinių jie turi būti pertvarkomi, jų kokybė turi būti užtikrinta, ir tik tuomet juos galima perkelti į duomenų saugyklą. Reikalingą informacijos pateikimą taip pat nusako ir verslo intelektikos sistemų turinio žymėjimas bei apibrėžimai, pavyzdžiui, jei verslo intelektikos vienetai reiškia tą patį dalyką, jie turi būti žymimi vienodai, tačiau, jei šie vienetai reiškia skirtingus dalykus, tuomet jie turi būti žymimi skirtingai [17].
- Verslo intelektikos sistema turi prisitaikyti prie atitinkamų pokyčių. Vartotojų, verslo poreikiai, duomenys, technologijos nuolatos skiriasi. Ši sistema turi būti suprojektuota taip, kad ji galėtų suvaldyti neišvengiamus pokyčius nedarant įtakos jau esamiems duomenims [17].
- Verslo intelektikos sistema turi pateikti informaciją nustatytu laiku. Verslo intelektikos sistemą sudaro jau pertvarkyti duomenys, tačiau nauji, neapdoroti duomenys turi būti pertvarkomi ir įkeliami į duomenų saugyklą per kelias valandas, minutes ar net sekundes [17].
- Verslo intelektikos sistema turi apsaugoti informaciją nuo įsilaužimų. Visa svarbiausia informacija yra saugoma duomenų saugykloje, todėl visos

informacijos pateikimas į netinkamas rankas gali brangiai kainuoti organizacijai [17].

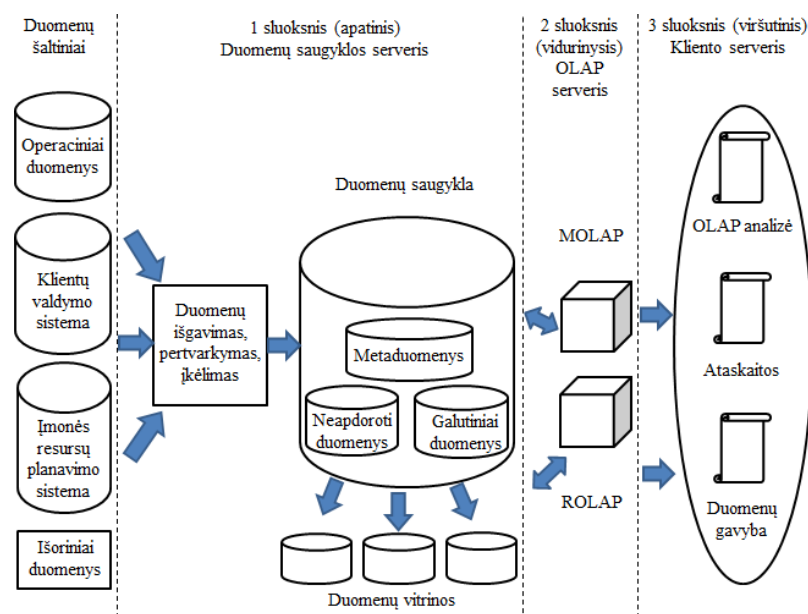
- Verslo intelektikos sistema privalo patenkinti vartotojus kaip patikimas duomenų šaltinio pagrindas sprendimų priėmimui vykdyti. Duomenų saugykloje turi būti saugomi tik teisingi duomenys, iš kurių būtų galima atlikti analizę, kuri būtų vertinga verslui [17].
- Verslo intelektikos sistema turi būti priimta verslo, jog ji tinkama naudoti. Nepaisant to, ar sistema sukurta nepriekaištingai, ji neteks vertės, jei ji bus beveik nebenaudojama [17].

1.3. Duomenų saugyklos architektūra

Klasikinė duomenų saugyklos architektūra susideda iš trijų lygmenų:

1. Apatinis lygmuo.
2. Vidurinis lygmuo.
3. Viršutinis lygmuo [5].

Duomenų saugyklos architektūra pavaizduota 1.1 pav.



1.1 pav. Duomenų saugyklos architektūra [5]

Į duomenų saugyklą gali būti įkeliami tiek operaciniai duomenys, tiek duomenys iš klientų valdymo sistemos, kuriame saugomi istoriniai duomenys apie organizacijos klientus, jų pirkimus, duomenys iš įmonės išteklių planavimo sistemos, kurioje saugomi verslo srities veiklų duomenys, pavyzdžiui, produktų planavimas, inventoriaus valdymas ir t. t., bei kiti išoriniai duomenys, kurie gali būti reikalingi organizacijai šių duomenų analizei. Apatinį lygmenį sudaro

duomenų saugyklos serveris, paremtas reliacinės duomenų bazės modeliu. Šiame lygmenyje įkeliami duomenys iš operacinių duomenų ar išorinių duomenų šaltinių (dažniausiai jie apima klientų valdymo sistemos, įmonės resursų planavimo duomenis) panaudojus atitinkamus įrankius. Šie įrankiai atlieka išgavimo, išvalymo bei pertvarkymo procesą. Duomenų išgavimas atliekamas panaudojant taikomasias programavimo sąsajas, kurios vadinamos tinklų sąsajomis. Tinklų sąsajos yra palaikomos DBVS ir leidžia pasiekti kitas nuotolines duomenų bazes. Keliais pavyzdžiais galėtų būti ODBC, OLE DB bei JDBC tinklų sąsajos. Pačioje duomenų saugykloje dažniausiai yra saugomi apibendrinantys duomenys, agreguoti iš skirtingų duomenų šaltinių panaudojus SQL kalbos agregavimo funkcijas, tokias kaip COUNT(), MAX(), AVERAGE() ir pan., ar kitas aritmetines operacijas, taip pat neapdoroti duomenys bei metaduomenys, kurioje saugomi duomenys apie duomenų saugyklą [12].

Vidurinį lygmenį sudaro OLAP serveris, kuris vykdomas panaudojus tiek reliacinį OLAP (ROLAP) modelį, tiek ir daugiamačių OLAP (MOLAP) modelį. Reliaciniame modelyje saugomi detalizuoti duomenys reliacinėse duomenų bazės lentelėse. Toks modelis kuriamas siekiant pagerinti ETL proceso laiką ir saugoti didelio kiekio duomenis, tačiau užklauskos dažniausios vykdomos lėtai. Daugiamačiame modelyje duomenys organizuojami duomenų kubo struktūroje, kuriuo galima vykdyti spartesnes užklauskas, tačiau tampa sudėtingiau išgauti didesnio kiekio duomenis. Šiuo metu egzistuoja ir hibridinis modelis (HOLAP), susidedantis iš ROLAP ir MOLAP modelių kombinacijos. Šis modelis leidžia saugoti detalius didelio kiekio duomenis paskirstant juos tiek ROLAP, tiek MOLAP sistemose. Siekiant šio modelio efektyvumo sutelkti duomenys yra saugomi MOLAP sistemoje, o detalizuoti duomenys saugomi ROLAP sistemoje. Pagrindinis šio modelio trūkumas yra jo struktūros sudėtingumas, nes HOLAP sistemos turi tikti tiek ROLAP, tiek MOLAP sistemoms bei įrankiams. OLAP modeliai palyginami 2 lentelėje (žr. 1.2 lentelė) [5].

1.2 lentelė. OLAP modelių palyginimas [5]

Modelis	ROLAP	MOLAP	HOLAP
Duomenų užklauskų greitis	Lėtai vykdomos užklauskos lyginant su kitais modeliais	Sparčiai vykdomos užklauskos lyginant su kitais modeliais	Sparčiai vykdomos užklauskos lyginant su kitais modeliais
Atliekami skaičiavimai su duomenimis	Atliekami sudėtingesni skaičiavimai gali sulėtinti visą procesą	Sudėtingi skaičiavimai atliekami greitai	Galima atlikti sudėtingus skaičiavimus
Duomenų saugojimo kiekis	Galima saugoti neriboto kiekio duomenis	Saugoti didelio kiekio duomenis yra sudėtinga, nes tenka saugoti dublikatus	Didesnio kiekio duomenys saugomi ROLAP sistemose
Pažeidžiamumas	Prastas duomenų saugyklos panaudojimas	Reliacinės duomenų bazės nėra tinkamos sprendimų priėmimo sistemoms	-

Viršutinį lygmenį sudaro kliento sluoksnis, kuriame vykdomos užklausos, suformuojamos ataskaitos, analizuojami ir išgaunami duomenys naudojantis duomenų gavybos įrankiais [12]. Naudojantis ataskaitų formavimo, užklausos įrankiais, vartotojai gali gauti jiems aktualius duomenis grafine atvaizdavimo forma, atlikti duomenų analizę, panaudoti duomenų gavybos įrankius atskleidžiant tam tikras tendencijas tarp duomenų ir t. t.

Pagal duomenų saugyklos architektūrą egzistuoja trys duomenų saugyklos modeliai:

- Įmonės duomenų saugykla.
- Duomenų vitrinos.
- Virtuali duomenų saugykla.

Įmonės duomenų saugykloje saugomi duomenys, kurie atitinka jos veiklą. Jame saugomi detalūs bei sutelkti duomenys ir gali užimti iki kelių šimtų gigabaitų ar terabaitų. Įmonės duomenų saugykla gali veikti tiek universaliuose kompiuteriuose, tiek ir kompiuterių serveriuose. Vienintelė problema, su kuria susiduriama, yra tai, jog šiai saugyklai reikalingas platus verslo modeliavimas, ir jo kūrimas gali trukti kelerius metus [12].

Duomenų vitrinose saugomi tik tam tikros srities duomenys, gaunami iš duomenų saugyklos, kurie tiekiami specifinių vartotojų grupių poreikiams patenkinti. Jos dažniausiai taikomos, kai reikia paspartinti užklausų vykdymą sumažinant gaunamų duomenų kiekį, taip pat skaidyti duomenis, kad būtų galima užtikrinti prieigą prie šių duomenų [12].

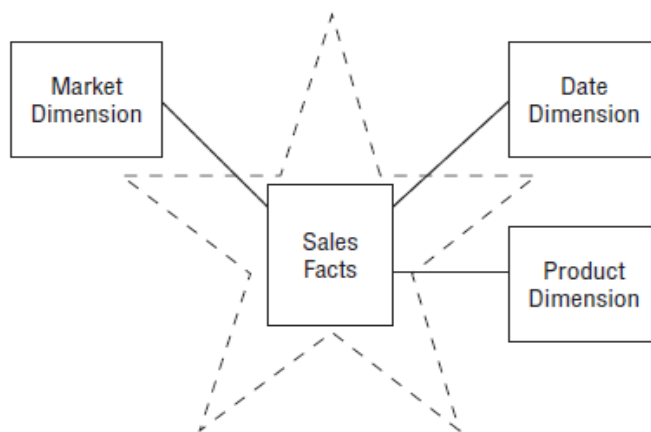
Virtualią duomenų saugyklą sudaro vaizdiniai, kuriame atrenkami duomenys iš operacinių duomenų bazių. Vienas iš dažniausiai naudojamų vaizdų yra materializuotas vaizdinys, kuriame periodiškai atnaujinamas saugomas SQL užklausos rezultatas. Vaizdinio atnaujinimo periodiškumas kinta jo kūrimo metu, o tai leidžia daug greičiau atlikti užklausas, kadangi vaizdinio rezultatai kartkartėmis saugomi diske, tačiau naudoja didelį kiekį atminties, o rezultatai tampa pasenę. Nors virtualią duomenų saugyklą sukurti paprasta, tačiau reikalauja didesnio kiekio atminties operacinėse duomenų bazėse [12].

1.4. Duomenų saugyklos modeliavimas

Kaip ir standartinėse duomenų bazėse, taip ir duomenų saugykloje taikomi duomenų bazių kūrimo metodai. Reliacinėse duomenų bazėse naudojamas reliacinis duomenų bazių modelis, kuriame duomenys bei ryšiai tarp jų yra organizuojami lentelėse. Duomenų saugyklos projektavimas paremtas daugiamačiu modeliavimu, kuriame naudojamos „žvaigždės“ arba „snaigės“ schemos, kur duomenys yra organizuojami į faktų bei dimensių lenteles. Faktų lentelė – tai centrinė lentelė, kuri jungia visas kitas lenteles, kurios vadinamos dimensių lentelėmis.

Dimensijų lentelėse yra saugomi duomenys, kurie kitaip dar vadinami verslo vienetais, apibūdinantys objektus faktų lentelėse [17].

Tradicinės duomenų bazės yra normalizuotos, o tai reiškia, jog lentelės ir tarpusavio ryšiai yra modeliuojami taip, jog lentelėse nebūtų dubliuojami duomenys, jų stulpeliai būtų nepriklausomi nuo kitų lentelėje esančių stulpelių, išskyrus pirminį raktą, kuris unikalčiai identifikuoja įrašą. Duomenų saugykloms projektuoti naudojama „žvaigždės“ schema nėra normalizuota (žr. 1.2 pav.). Priešingai, šiai schemai sumodeliuoti yra taikomas denormalizacijos principas. Denormalizacijos metodu siekiama optimizuoti duomenų išgavimo greitį bei efektyvumą iš duomenų bazės pridėdant pasikartojančius duomenis arba grupuojant duomenis, nes esant dideliui normalizuotų lentelių kiekiui jas tenka sujungti atliekant užklausas, o tai labai sulėtina sistemos darbą [17].



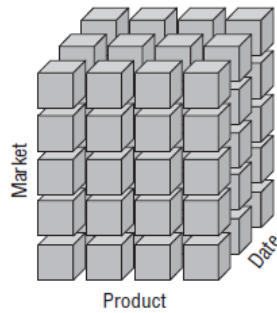
1.2 pav. „Žvaigždės“ schema [17]

Kartais siekiant sumažinti pasikartojančių įrašų kiekį naudojama „snaigės schema“, kuri yra „žvaigždės“ schemos išplėstinis modelis. Šios schemos principas yra toks, jog kai kurios dimensijų lentelės yra normalizuojamos, tačiau ji nėra taip dažnai naudojama kaip „žvaigždės“ schema dėl pernelyg sudėtingo užklausų formavimo [17].

Apibendrintas duomenų saugyklos modeliavimas apima tokius procesus:

1. Verslo proceso modeliavimas.
2. Verslo proceso duomenų nustatymas, kurie bus saugomi faktų lentelėje.
3. Faktų nustatymas, kurie atitiks verslo vienetus.
4. Dimensijų nustatymas, kurie atitiks kiekvieną įrašą, esantį faktų lentelėje [17].

Remiantis šiais modeliavimo etapais suprojektuojama duomenų saugykla. Suprojektavus ir sukūrus duomenų saugyklą, formuojama specializuota duomenų bazė, kurioje faktai sutvarkomi pagal matus, vadinama OLAP duomenų kubu (žr. 1.3 pav.).



1.3 pav. Duomenų kubas [17]

Taikant šią duomenų struktūrą galima atlikti duomenų analizę greitai ir efektyviai pasitelkiant jai būdingas operacijas:

1. Apžvalga (angl. *Roll up*). Jos metu atliekama agregavimo operacija perkeliant duomenis aukštesniu hierarchijos lygiu bei mažinant dimensijas, pavyzdžiui, viena iš duomenų kubo dimensijų yra diena, kuri gali būti vaizduojama hierarchiniu būdu tiek savaitės dienomis, tiek ir pačiomis savaitėmis. Šiuo atveju, turint duomenis, kurie atvaizduojami dienomis, bet reikia sužinoti savaitėmis, atliekamas grupavimas pagal savaites, ir gaunamas galutinis rezultatas.
2. Detalizavimas (angl. *Roll down*) yra atvirkštinė operacija apžvalgai. Jos metu duomenys perkeliama žemesniu lygiu, o kubo dimensijų nemažėja.
3. Pasukimas (angl. *Pivot*). Pasukimo metu keičiamos duomenų kubo dimensijų orientacijos, ir atliekamas duomenų grupavimas su skirtingomis dimensijomis.
4. Dalinimas (angl. *Slicing*). Šios operacijos metu pagal tam tikrus kriterijus atrenkama viena kubo dimensija, kurios rezultatas yra gautas mažesnis kubas.
5. Išpjovimas (angl. *Dicing*). Ši operacija panaši į dalinimą, tačiau atrenkamos kelios duomenų kubo dimensijos pagal tam tikrus kriterijus [31].

2. ETL procesas

2.1. ETL svarba

ETL vadinama duomenų išgavimo iš skirtingų duomenų šaltinių, jų transformavimo ir įkėlimo į numatytą duomenų saugyklą technologija. ETL procesas yra vienas svarbiausių užduočių verslo intelektikos srityje. Įmonėms labai svarbu įkelti duomenis kaip įmanoma sparčiau, kad būtų galima iš gautų duomenų formuoti ataskaitas, jas analizuoti. ETL procesas yra ganėtinai aktuali tema, nes šiais laikais duomenų kiekis bei dydis auga, todėl tenka taikyti metodus, kurie leistų įkelti didelio kiekio, skirtingo formato duomenis, ar kurti vartotojui patogius įrankius, kurie leistų automatiškai įkelti šiuos duomenis.

2.2. ETL etapai

ETL procesas susideda iš trijų etapų:

1. Duomenų išgavimas.
2. Duomenų pertvarkymas.
3. Duomenų įkėlimas.

2.2.1. Duomenų išgavimas

Duomenų išgavimo metu gaunami ir surenkami duomenys iš skirtingų duomenų šaltinių. Duomenų išgavimo procesas turi būti suprojektuotas taip, kad duomenų šaltiniai būtų kuo mažiau apkrauti, galėtų greičiau pateikti duomenis ir kt. Šiame procese taip pat svarbu integruoti sistemas, kurios yra naudojamos skirtingose platformose, tokiose, kaip skirtingos DBVS, operacinės sistemos, skirtingi bendravimo protokolai. Išskiriamos dvi duomenų išgavimo rūšys:

1. Loginis duomenų išgavimas.
2. Fizinis duomenų išgavimas [7].

Loginis duomenų išgavimas skaidomas į du etapus:

1. Pavienis duomenų išgavimas. Sistemos, kuriose nėra naudojamas pranešimų atnaujinimas, šiuo metodu gali atpažinti, kurie įrašai pasikeitė, bei juos išgauti. Tam, kad būtų galima atpažinti atliktus pakeitimus, turi būti įmanoma atpažinti visą pasikeitusią informaciją nuo šio įvykio. Dauguma duomenų saugyklų nenaudoja jokių metodų, leidžiančių fiksuoti pakeitimus. Vietoj to, likę duomenys iš duomenų sistemų lentelių yra išgaunami ir perkeliami į duomenų saugyklą ar parengimo sritį, ir duomenys yra palyginami su prieš tai išgautais duomenimis iš duomenų sistemų, kad būtų galima atpažinti duomenis [7].

2. Visiškas duomenų išgavimas. Kai kurios sistemos negali atpažinti duomenų, kurie pasikeitė, todėl tenka išgauti visus esamus įrašus. Taikant šį metodą dažnai reikalauja turėti paskutinių išgautų duomenų kopiją, kad būtų galima nustatyti pakeitimus [7].

Fizinis duomenų išgavimas taip pat skaidomas į du etapus:

1. Duomenų išgavimas prisijungus.
2. Duomenų išgavimas neprisijungus [7].

Duomenų išgavimo prisijungus metu duomenys išgaunami tiesiai iš duomenų šaltinio. Duomenų išgavimo proceso metu prisijungiama prie duomenų šaltinio, kad būtų galima prieiti tiesiogiai prie šaltinio lentelių ar prie tarpinės sistemos, kurioje saugomi duomenys. Išgaunant duomenis prisijungus būtina apsvastyti, ar paskirstytos transakcijos naudoja originalius objektų šaltinius ar jau paruoštus objektų šaltinius [7].

Duomenų išgavimo neprisijungus metu duomenys išgaunami ne tiesiogiai iš duomenų šaltinio, bet išoriniu būdu. Šis metodas dažnai taikomas, kai prieiti prie duomenų šaltinių lentelių negalima, todėl duomenys yra saugomi atskirai nuo duomenų šaltinio. Duomenys tuo metu jau yra paruošti, sudaryta egzistuojanti struktūra. Struktūrų pavyzdžiais gali būti:

1. Plokščiasis failas.
2. Duomenys apibrėžtame formate.
3. Iškelties failai.
4. Archyvų ir perdarymo registravimas.
5. Perkeliamos lentelių erdvės [7].

Prieš atliekant duomenų pertvarkymą atliekamas duomenų išvalymo procesas, kurio metu užtikrinama, jog į duomenų saugyklą bus įkelti kokybiški duomenys. Duomenų išvalymo metu atliekamos tokios operacijos, kaip tuščių laukų konvertavimas į standartizuotas reikšmes, laukų tikrinimas, identifikatorių unikalumo požymio nustatymas ir t. t.

2.2.2. Duomenų pertvarkymas

Duomenų pertvarkymo procesas dažniausiai vadinamas kaip sudėtingiausias ETL etapas vykdymo laiko atžvilgiu. Šiame procese gali būti atliekamos tiek pačios paprasčiausios duomenų transformacijos, tiek ir pačios sudėtingiausios duomenų pertvarkymo technikos. Duomenų pertvarkymas gali būti atliekamas toje pačioje duomenų bazėje, iš kurios išgaunami duomenys, ir tuomet perkelti duomenis į duomenų saugyklą, tačiau dažniausiai šis procesas atliekamas

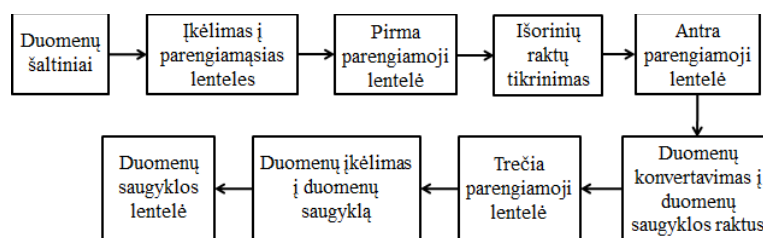
išoriniu būdu, pavyzdžiui, visą reikalingų duomenų lentelių arba iškelties failų turinį perkeltiant į parengiamąjį sluoksnį.

Duomenų pertvarkymo metu vykdomos taisyklės ar funkcijos, leidžiančios konvertuoti duomenis iš skirtingų duomenų šaltinių į standartinį formatą, tinkamą duomenų saugykloje ar kitoje duomenų bazėje. Šiame etape atliekamas įrašų konvertavimas, tikrinimas, atmetimas, rūšiavimas, filtravimas, besidubliuojančių įrašų šalinimas pagal atitinkamas verslo taisykles.

Duomenys dažniausiai transformuojami dviem būdais:

1. Daugiapakopiu būdu.
2. Grandininiu būdu [7].

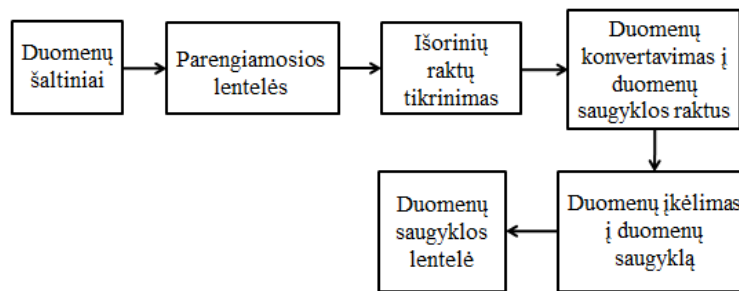
Daugiapakopiu būdu duomenų transformavimo logika daugumoje duomenų saugyklų susideda iš kelių etapų. Visų pirma, iš išorinio duomenų šaltinio (reliacinės duomenų bazių lentelės, iškelties failai ir pan.) gaunami duomenys ir perkeltami į parengiamąją lentelę. Tuomet atliekamas išorinių raktų tikrinimas. Patikrinti duomenys yra įkeliami į naują parengiamąją lentelę. Šioje lentelėje atliekamas duomenų iš išorinių duomenų šaltinių konvertavimas į duomenų saugyklos duomenis. Atlikus šį žingsnį, duomenys įkeliami į dar vieną parengiamąją lentelę. Iš šios lentelės vykdomas ETL procesas į duomenų saugyklos lentelę [7]. Detali schema pavaizduota 2.1 pav.



2.1 pav. Daugiapakopis duomenų keitimas [7]

Daugiapakopiu būdu siekiama išskirti kiekvieną transformacijos etapą ir gautus rezultatus įrašyti į naują laikinąją lentelę. Tai palengvina duomenų transformavimo fiksavimą ir proceso perkrovimą, tačiau tuo pačiu saugomas didesnis kiekis objektų reikalauja daugiau atminties kompiuteryje, taip pat padidėja šio proceso laikas.

Grandininiu būdu atliekamas duomenų pertvarkymas įkėlimo metu. Tokiu būdu išvengiama problemų, kurios atsiranda naudojant daugiapakopį duomenų pertvarkymą. Duomenų pertvarkymas atliekamas įkeliant duomenis iš skirtingų duomenų šaltinių į atskiras parengiamąsias lenteles. Tuomet tikrinami išoriniai raktai, konvertuojami duomenys iš išorinių duomenų šaltinių į duomenų saugyklos išorinius raktus, ir galiausiai atliekamas įterpimas į duomenų saugyklos lentelę [7]. Detali šio proceso schema pavaizduota 2.2 pav.



2.2 pav. Grandininis duomenų keitimas [7]

ETL proceso laikas taip pat priklauso nuo to, kaip duomenys gali būti nuskaitymi iš parengiamosios vietos ir įterpiami į duomenų saugyklą. Dėl ETL spartos reikalaujama, kad ši vieta būtų naudojama toje pačioje duomenų bazėje, kuriame saugoma duomenų saugyklos struktūra, kadangi duomenų nuskaitymas iš jos paspartina ETL. Dažniausiai rekomenduojama, jog duomenys būtų parengiami tarp kaip įmanoma daugiau fizinių diskų, kad būtų galima užtikrinti, jog duomenų nuskaitymas nebus kliūtis ETL procesui [33].

2.2.3. Duomenų įkėlimas

Duomenų įkėlimo metu išgauti bei pertvarkyti duomenys yra įkeliami į duomenų saugyklą ar kitą duomenų bazę. Duomenų įkėlimo procesas dažniausiai apima kiekvieno įrašo įterpimą kaip naują eilutę duomenų saugyklos lentelėje naudojant SQL įterpimo komandą ar ETL įrankį. Kiti įkėlimo procesai apima masinių duomenų įterpimą panaudojus masinio įkėlimo rutiną. Ši rutina veikia sparčiau įkeliant didelio kiekio duomenis, bet neleidžia tikrinti kiekvieno įrašo integravimo šio proceso metu, todėl atsiradus tam tikrai klaidai nebus galimybės atsekti konkretaus įrašo, dėl kurio galėjo būti nutrauktas procesas. SQL įterpimo komanda, priešingai nei masinė įkėlimo rutina, įgalina tikrinti kiekvieno įrašo integravimą, tačiau yra lėtesnė įkeliant duomenis.

2.3. ETL sparta duomenų saugyklose

Duomenų saugyklose duomenys yra įkeliami į juos sudarančių faktų ir jas jungiančių dimensijų lenteles. Dimensijų lentelėse gali būti saugomas ne didesnis kiekis duomenų nei faktų lentelėse, tuo tarpu faktų lentelėse saugomi keli, keli šimtai milijonų ar daugiau įrašų, todėl daugiausiai laiko panaudojama perkeliant duomenis į šias lenteles. Visų pirma, ETL procesas atliekamas dimensijų lentelėse, nes faktų lentelėse yra saugomos nuorodos (raktai) į dimensijų lentelių įrašus. Tuomet turi būti atliekamas naujų raktų generavimas iš dimensijų lentelių. Dažniausiai tai būna pirminis šių lentelių raktas, tačiau gali būti naudojamas ir kitas unikalus raktas. Kai tikrai duomenys į skirtingas dimensijas yra įkelti, atliekamas duomenų įkėlimas į faktų lenteles.

Bendras ETL į duomenų saugyklą laikas skaičiuojamas tokiu būdu:

$$T_i = \sum(T(i,j)_{Dim_E} + T(i,j)_{Dim_{TL}}) + \sum(T(i,j)_{Fakt_E} + T(i,j)_{Fakt_{TL}}) +$$

$T(i)_{kitos\ operacijos}$, kur $\sum(T(i,j)_{Dim_E} + T(i,j)_{Dim_{TL}})$ yra ETL laikas išgavus duomenis iš duomenų šaltinių ir pertvarkius bei įkėlus duomenis į kiekvieną dimensijų lentelę, $\sum(T(i,j)_{Fakt_E} + T(i,j)_{Fakt_{TL}})$ yra ETL laikas išgavus duomenis iš duomenų šaltinių ir pertvarkius bei įkėlus duomenis į faktų lenteles, $T(i)_{kitos\ operacijos}$ – laikas, skirtas kitų operacijų vykdymui, susijusių su ETL procesu [32]. Naudojantis šia formule galima nustatyti, kiek laiko trunka ne tik visas ETL procesas, bet ir individualaus proceso laikas: atitinkamai galima apskaičiuoti, kiek laiko trunka duomenų išgavimas, atliekami pertvarkymai ir galutinis duomenų įkėlimas.

3. ETL metodai, technologijos bei įrankiai

ETL sudaro trys operacijos, tačiau yra metodų, leidžiančių vienaip ar kitaip įkelti duomenis kokybiškiau, spartesniu būdu. Dauguma metodų yra realizuoti duomenų integravimo įrankiuose, tačiau tokiose DBVS, kaip „Oracle“, gali būti naudojamas PL/SQL kodas, „Microsoft SQL Server“ naudojama T-SQL, taip pat SQL užklausos. Šis būdas gali užtikrinti geresnę ETL kokybę bei spartą negu naudojant daugumą automatizuotų įrankių puikiai išmanant šių procedūrinių kalbų galimybes, tačiau programuotojui pačiam rašyti programinį kodą, kuriuo būtų galima atlikti duomenų išgavimą, pertvarkymą ir įkėlimą, yra daug laiko reikalaujantis darbas. Be to, programuotojui reikia pačiam kurti lenteles ir sekti ETL proceso metu esančias operacijas ir klaidas.

3.1. Duomenų įkėlimo metodai ETL procese

ETL procesas vykdomas nuolatos, o ne vieną kartą. Kaskart atliekamos duomenų įrašymo, atnaujinimo ir šalinimo operacijos operacinėse duomenų bazėse, todėl turi būti panaudojamas tinkamas duomenų perkrovimo metodas norint matyti atliktus pakeitimus duomenų saugyklose. Kadangi duomenų saugyklose saugomi istoriniai duomenys, ir jie naudojami siekiant greičiau vykdyti užklausas, realiu laiku juos sinchronizuoti nėra praktiška, nes atliekamos transakcijos didžiuliu mastu, kurios gali apkrauti sistemas, todėl siekiant išlaikyti duomenų sinchronizavimą tarp dviejų duomenų bazių, tenka apsvarstyti duomenų šaltinių atnaujinimo metodus, kurie būtų kuo geriau suderinami su duomenų perkrovimo rūšimis.

ETL išskiriami du duomenų įkėlimo metodai:

1. Visiškas duomenų įkėlimo procesas.
2. Pavienis duomenų įkėlimo procesas [32].

Visišku ETL proceso metu atliekamas visų esamų duomenų įkėlimas į tarpinę duomenų bazę, į kurią įrašomi, pertvarkomi ir įkeliami duomenys iš šios duomenų bazės į duomenų saugyklą. Pavienio duomenų įkėlimo metu duomenys yra atnaujinami sparčiau lyginant su visiško duomenų įkėlimo metodu, kadangi atliekamos papildomos operacijos su didesniais duomenų kiekiais, t. y. visų duomenų šalinimo ir naujų įkėlimo operacijos. Dažniausiai šis metodas yra panaudojamas pirmą kartą atkėlus duomenis į duomenų saugyklą, tačiau gali būti atliekamas viso ETL proceso metu. Tokiu atveju iš skirtingų duomenų šaltinių atrenkami visi, net ir neatnaujinti duomenys, perkeliama į tarpinį sluoksnį, kuriame atliekami pertvarkymai, ir tuomet vykdomas duomenų įkėlimas į duomenų saugyklą. Kadangi duomenys gali kartotis tiek duomenų šaltiniuose, tiek duomenų saugykloje, reikalingos atitinkamos operacijos, kurios gebėtų nustatyti, ar duomenys jau egzistuoja duomenų saugykloje, ir jų neįkėlinėti iš naujo.

Pavienis duomenų įkėlimo procesas atliekamas tikrai su tais duomenimis, kurie buvo keičiami tam tikru laikotarpiu. Tai yra operacija, apimanti tik senų duomenų atnaujinimą ir naujų įrašų įterpimą. Kai tikrai įrašai yra pašalinami arba tampa nebereikalingi, tuomet pavienio sinchronizavimo metu duomenų saugykloje esantys įrašai dažniausiai išlieka, tačiau gali būti pažymimi atitinkamu požymiu, jog jie yra nebereikalingi, labai retais atvejais gali būti pašalinami. Atsižvelgiant į duomenų saugyklos struktūrą (faktų bei dimensių lentelės) iš pradžių pavienio duomenų atnaujinimo metu išgaunami duomenys iš dimensių bei faktų lentelių ir įkeliami į laikinąją duomenų bazę, tuomet pertvarkomi ir įkeliami duomenys iš dimensių lentelių, vėliau ir duomenys iš faktų lentelių [32].

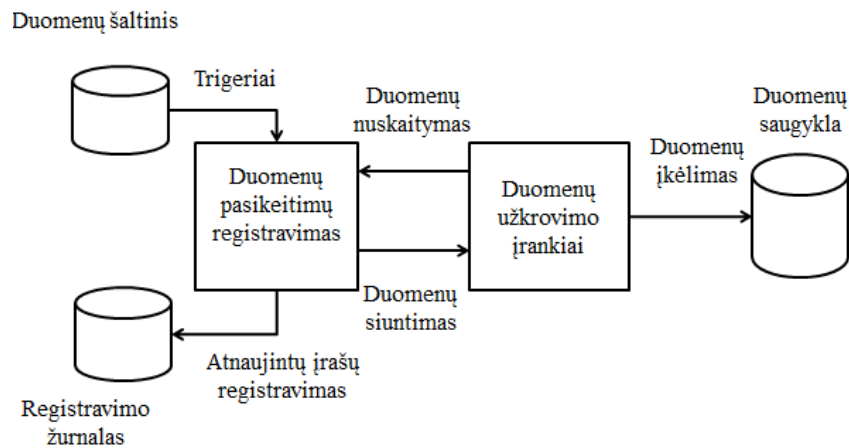
Pavienis duomenų įkėlimo procesas gali būti realizuojamas tiek naudojantis DBVS sistemos teikiamomis technologijomis, tiek atskira programine įranga, leidžiančiomis realiu laiku gauti keičiamus duomenis ir juos saugoti atskiroje registravimo saugykloje.

3.2. Sistemų teikiamos technologijos ETL procese

3.2.1. Pakitusių duomenų registravimas

Pakitusių duomenų registravimo metu remiamasi principu, jog būtų įkeliami tik tie duomenys, kurie buvo pakeisti realiu laiku tam tikru laikotarpiu. Tai yra pavienio duomenų įkėlimo metodologijos esmė. ETL proceso metu iš operacinių sistemų ištraukiami tik pakitę duomenys atlikus tam tikras transakcines operacijas. Išskiriamos dvi šios metodologijos rūšys: esamų pakitimų registravimas (angl. Change Tracking) ir visų pakitusių duomenų registravimas (angl. Change Data Capture). Esamų pakitimų registravimo metu naudojamas požymis, dažniausiai įrašo keitimo laikas, kuris nurodo, ar įrašas buvo koreguojamas, ar ne. Šis metodas tinka tuomet, jei duomenys nėra šalinami, tačiau nesaugoma įrašų keitimo istorija, kas leidžia sumažinti duomenų kiekį lentelėse. Visų pakitusių duomenų registravimo metu saugomi visi esminiai duomenų pokyčiai: naujai įrašyti, atnaujinti ir pašalinti duomenys. Pagrindinis šio metodo privalumas yra tai, kad registruojami absoliučiai visi atlikti pakeitimai, iš kurių galima paprastai atsekti, koku metu buvo atliktos modifikacijos. Visgi duomenų kiekis yra daug didesnis lyginant su esamų pakitimų registravimu, todėl gauti konkrečius duomenis yra sudėtingiau ir užtrunka ilgiau.

Dažnai tokiems įrašams saugoti naudojamos duomenų bazių lentelės, registravimo žurnalai ir pan. Visų pakitimų registravimo realizacija pateikta 3.1 pav.



3.1 pav. Duomenų pasikeitimų registravimo schema [4]

Iš duomenų bazės, kuriame vykdomi pakeitimai, gaunami reikalingi įrašai ir įrašomi į atskirą registravimo žurnalą ar duomenų bazės lentelės kopiją. Šiems pakeitimams registruoti naudojami trigeriai, kurie įrašo pakeitimus į esamą registravimo žurnalą ar lentelę. Tuomet pasitelkiant ETL įrankį ar panaudojant savarankiškai parašytą programos kodą šie duomenys yra išgaunami, pertvarkomi ir įkeliami į atitinkamą duomenų saugyklą.

Yra keletas metodų, kaip būtų galima registruoti pakeitimus realiu laiku:

- Duomenų šaltinių įrašų tarpusavio palyginimas. Šiuo metodu nuosekliai lyginami du duomenų šaltiniai: lyginami tiek įrašai, tiek ir laukai su ankstesne kopija, todėl ankstesni duomenys turi būti saugojami parengiamajame sluoksnyje. Šis metodas užtikrina, jog bus palyginti visi pakitę duomenų įrašai, tačiau reikalauja papildomų kompiuterių išteklių ir laiko [1].
- Duomenų bei numatytų šaltinių palyginimas naudojant ciklinę pasikartojimų kontrolinę sumą. Taikant šį metodą naudojamas algoritmas, kuris suskaičiuoja tam tikrą unikalią reikšmę iš skirtingų laukų. Lygiai taip pat, kaip ir visų duomenų bei numatytų šaltinių palyginimo atveju, lyginami dabartiniai įrašai su ankstesniais įrašais, tačiau filtravimui naudojamas tik vienas skaičius, sugeneruotas kontrolinės sumos funkcijos, vietoj visų skirtingų laukų [1].
- Transakcijų registravimo failai. Daugumoje DBVS šie failai yra integruoti, kuriuose įrašomi visi įrašyti, atnaujinti bei pašalinti įrašai kiekvienos transakcijos metu. Šie failai gali būti nuskaitymi tam tikru dažnumu (pavyzdžiui, kas parą) arba galima nuolatos juos peržiūrėti realiu laiku. Pagrindinis šių failų privalumas yra tai, kad atliekant transakcijas ir įrašant pakeitimus į šiuos failus duomenų bazė neapkraunama papildomomis operacijomis. Visgi naudojant šiuos failus reikia nuolatos juos išvalyti, kad senų ir nereikalingų duomenų neliktų [1].

- DBVS trigerių panaudojimas. Tai yra vienas iš dažniausių naudojamų metodų, kuris vykdomas tada, kai tikrai atliekami įrašymo, atnaujinimo ar šalinimo veiksmai lentelėje. Paleidus triggerį, vykdomas jame esantis programinis kodas, įrašantis duomenų pakeitimus į atskirą lentelę, registravimo failą arba jei lentelė turi auditinį lauką, kuriame nustatoma, kada paskutinį kartą buvo modifikuotas įrašas, tuomet į šį lauką įrašoma šio įrašo pakeitimo data, tačiau pasirenkant šį metodą turi būti apsvairstyta, ar verta jį naudoti, nes trigeriai papildomai apkrauna ir stabdo duomenų bazės darbą [1].
- Programų, kuriose yra įdiegtas duomenų pasikeitimų registravimas, naudojimas. Tai yra pats paprasčiausias metodas, nereikalaujantis jokių techninių žinių, kad būtų galima pačiam registruoti pakeitimus. Vis dėlto kai kurios programos gali kainuoti labai brangiai, taip pat tam, kad būtų galima naudotis šiuo įrankiu, reikia apmokėti darbuotoją, kuris gebėtų dirbti su šiuo įrankiu [1].

3.2.2. Duomenų bazės lentelių panaudojimas

Duomenys dažniausiai saugomi lentelėse, iš kurių yra gana paprasta vykdyti užklausas. Vis dėlto galima pasitelkti DBVS teikiamomis lentelių galimybėmis, leidžiančiomis vienokiu ar kitokiu būdu paspartinti ETL procesą.

ETL procesas naudojant laikinas lenteles

Laikinos lentelės apibrėžiamos kaip tarpinės rezultatų lentelės, leidžiančios paspartinti SQL užklausas įrašant didelio kiekio duomenis. DBVS sistemose išskiriamos dvi šių lentelių grupės: lokalsios bei globalios laikinosios lentelės. Lokalsios lentelės yra matomos tik jų kūrėjams konkrečioje sesijoje, kai tik jos sukuriamos. Kai tikrai sesija pasibaigia, duomenys iš šių lentelių yra pašalinami. Jose dažniausiai saugojami laikini duomenys, kai jie yra paruošiami duomenų įkėlimui. Globalios lentelės yra matomos bet kokiam vartotojui bet kurios sesijos metu, kai tikrai jos sukuriamos, o duomenys yra pašalinami, kai tikrai nebelieka vartotojų, naudojančių šias lenteles [25]. „Oracle“ DBVS laikinosios lentelės, pasibaigus sesijai, yra paliekamos duomenų bazėje, o jose esantys duomenys yra pašalinami. „Microsoft SQL Server“ DBVS pasibaigus sesijai, pašalinami ne tik duomenys, bet ir pačios lentelės, nes jos yra saugomos atskiroje „tempdb“ duomenų bazėje, kurioje visi vartotojų sukurti esantys objektai yra pašalinami, kai tik pasibaigia sesija. Šioje duomenų bazėje minimaliai registruojamos visos transakcinės operacijos, išvalomi šių operacijų registravimo failai kaskart perkrovus duomenų bazę, todėl neatsiranda papildomų sistemos apkrovų atliekant šias operacijas. Bene pagrindinis skirtumas tarp globalių ir laikinių lentelių yra tas, jog lokalsios laikinos lentelės yra žymiai geriau apsaugotos, nes jų negali matyti kiti vartotojai, tik jų kūrėjas.

Padalytos lentelės, padalijimo metodologija

Esant dideliam duomenų kiekiui duomenys gali būti įkeliami į padalytas lenteles. Padalytomis lentelėmis vadinamos lentelės, kurios duomenys yra paskirstomos skirtingose lentelės dalyse, o pačios lentelės yra paskirstomos skirtingose failų grupėse vienoje duomenų bazėje. Šių lentelių struktūra nesiskiria nuo įprastų lentelių, tačiau duomenys yra paskirstomi šių lentelių dalyse pagal atitinkamus kriterijus [25]. Tokie kriterijai nustatomi pagal vieno lentelės atributo reikšmes, pavyzdžiui, apribojimai gali būti formuojami pagal datą: vienoje lentelės dalyje bus paskirstyti duomenys iki 2010 m. gruodžio 31 d., kiti duomenys bus saugomi nuo 2011 m. sausio 1 d. iki 2011 m. gruodžio 31 d. ir t. t. Lentelių padalijimas traktuojamas kaip pagrindinės lentelės padalijimas į kelias logiškai sugrupuotas dalis tokiu būdu, kad kiekviena šios lentelės dalis būtų valdoma nepriklausomai nuo pagrindinės lentelės. Tokiu būdu padalijimo galimybė leidžia greitai ir paprastai valdyti lenteles, nes esant dideliam duomenų kiekiui vieną didelę lentelę valdyti yra sudėtinga.

Kiekvienoje lentelėje gali būti kuriami indeksai, taip pat ir atskirose jos dalyse. Naudojant šiuos indeksus galima greičiau pasiekti duomenis nei pagrindinėje lentelėje, taip pat juos valdyti nedarant įtakos šiai lentelei. Dėl tos pačios priežasties yra paprasčiau juos kurti ant padalytų lentelių negu ant vienos lentelės.

Atmintį taupančios lentelės

Viena iš naujausių DBVS savybių yra į atmintis orientuotos transakcijos (angl. In-Memory OLTP). Pagrindinis šios savybės privalumas yra duomenų užklausų ir manipuliavimo spartos pagerinimas. Šių sistemų realizavimui naudojamos atmintį taupančios lentelės ir sukompiliuotos saugomos procedūros, vartotojo sukurtos funkcijos bei trigeriai. Atmintį taupančios lentelės (angl. Memory optimized tables), priešingai, nei kietuosiuose diskuose saugomos lentelės, yra saugomos operatyvinėje atmintyje. Tai yra „Microsoft SQL Server“ DBVS naudojamos lentelės, kurios yra palaikomos nuo 2014 m. versijos [22]. Pagrindinis skirtumas tarp kietuosiuose diskuose saugomų lentelių yra tai, kad atmintyje saugomose lentelėse duomenys saugomi eilučių struktūroje. Kietuosiuose diskuose naudojamos fundamentinės duomenų saugojimo struktūros, vadinamos puslapiais, kuriuose atliekamos I/O operacijos. Pagrindinis privalumas naudojantis į eilutes orientuota duomenų saugojimo metodologija yra tai, kad nereikia papildomai nuskaityti puslapių į atmintį norint pasiekti atmintį taupančias lenteles. Tam yra sukuriami duomenų failai atskirose, atmintį taupančiose failų grupėse, kurie užtikrina duomenų ilgalaikiškumą, panašiai, kaip ir duomenų failai, naudojami kietuosiuose diskuose saugomose lentelėse. Tačiau šie failai yra pridėtiniai, o tai reiškia, jog pakitę duomenys negali būti atnaujinami, tik pridedami. Tokiu būdu duomenų bazėje išvengiama

I/O operacijų, taip pat palaikomas visiškas duomenų ilgalaikiškumas, dažnai naudojamas duomenų atstatymui sistemos ar duomenų bazės perkrovimo metu [2]. Kadangi atmintį taupančios lentelės ir jų įrašai yra saugomi atmintyje, tokiu būdu pagerinamos duomenų manipuliavimo operacijos ir pasiekiamumo vykdymas.

Išskiriamos dvi šių lentelių rūšys: ilgalaikės ir trumpalaikės lentelės. Ilgalaikės lentelės garantuoja duomenų ilgalaikiškumą sistemoje. Trumpalaikės lentelės apima pačių duomenų, ne lentelių saugojimo trukmę. Šiuo atveju duomenys saugomi lentelėse tol, kol nepasibaigia prisijungimo prie duomenų bazės sesija. Tokios lentelės yra naudojamos vietoj laikinų lentelių arba kaip pakaitalas laikinoms lentelėms ETL proceso realizavimo parengiamajame sluoksnyje metu [22].

Sukompiliuotos saugomos procedūros, vartotojo sukurtos funkcijos bei trigeriai yra duomenų bazės objektai, kurie sukompiliuojami mašininiame kode pradiniam naudojimui tik tada, kai jie yra sukuriami, ir saugomi DDL faile. Šie DDL failai yra perkompiliuojami tik tuo atveju, kada perkraunamos duomenų bazės ar jų egzemplioriai. Taip užtikrinama, kad kiekvieną kartą nereikės perkompiliuoti šių objektų, kai tiksliai jie vykdomi [22].

Vartotojo apibrėžtos lentelių tipo struktūros

Nuo „Microsoft SQL Server“ DBVS 2008 m. versijos leidžiama vartotojams kurti tipus, kurie apibrėžia lentelių struktūras. Naudojantis šiais tipais galima apibrėžti lentelės tipo parametrus tiek saugomose procedūrose, tiek ir funkcijose, arba jų viduje. Naudojantis šia savybe galima perduoti didelio kiekio duomenis į „SQL Server“ DBVS panašiai, kaip ir masinio įkėlimo būdu. Pagrindiniai šių struktūrų privalumai yra tai, kad sumažinamas perduodamų parametrų kiekis procedūrose bei duomenų paketų perdavimo laikas tarp kliento ir serverio. Tokios rūšies struktūrose galima kurti indeksus, tačiau turi būti deklaruojamas lentelės tipo kintamasis, kuriame sukuriami indeksai ir į kurį perduodami duomenys per procedūrą gaunamus duomenis, taip pat galima kurti pirminio rakto arba unikalumo apribojimus, tačiau jų negalima perkompiliuoti, atlikti duomenų manipuliavimo operacijas, jei lentelės tipo kintamasis perduodamas per procedūrą ar funkciją, nepalaikoma statistikos peržiūra [29].

3.2.3. Lygiagrečias ETL procesas

ETL procesas gali būti vykdomas lygiagrečiai siekiant vykdyti kelias operacijas vienu metu bei sumažinti ETL laiką esant labai dideliame duomenų kiekiui. Tokiu būdu įmanoma šio proceso srautą padalinti į kelias dalis ir jas vykdyti lygiagrečiai siekiant sumažinti vykdymo laiką. Procesai yra padalijami į kelias dalis taip, jog šios dalys būtų nepriklausomos, pavyzdžiui, įkeliant duomenis į lenteles, jei bet kokios lentelės įrašas negali būti apdorojamas iki tol, kol ankstesnis įrašas nebus apdorotas, vadinasi, šie įrašai priklausomi vienas nuo kito. Tuo atveju, jei

procesas yra vykdomas nuosekliai, jis negali būti padalijamas į atskirus antrinius procesus siekiant įkelti duomenis lygiagrečiai, tačiau, jei visi lentelės įrašai gali būti apdorojami nepriklausomai vienas nuo kito, tuomet lentelė gali būti padalijama ir vykdoma lygiagrečiai.

Lygiagrečiai galima vykdyti tiek duomenų išgavimą iš skirtingų duomenų šaltinių, tiek įkeliant duomenis į kelias lenteles, tiek ir antrinius procesus. Dažnai eksperimentiniais tyrimais sprendžiama, ar efektyviau būtų lygiagrečiai vykdyti visą ETL proceso srautą ar tik jo dalis [33].

Dar viena galimybė optimizuoti ETL procesą yra paskirstyti keliems procesoriams vykdyti kelis smulkesnius procesus vienu metu. Tokiu būdu galima paspartinti ETL laiką, tačiau šis metodas nebūtinai yra optimalus, jei procesas nevykdomas lygiagrečiai, nes kompiuteris tokiu atveju gali būti apkrautas, ir tai gali pristabdyti proceso vykdymą. Jeigu antriniai procesai vykdomi lygiagrečiai paskirstant juos keliems procesoriams, tuomet galima gauti didesnę naudą optimizuojant visą ETL procesą, kas yra dažniausiai naudojamas ir rekomenduotinas metodas [33].

Šiuo metu yra daugybė ETL įrankių, leidžiančių lygiagrečiai vykdyti ETL procesą. Šie įrankiai dažniausiai turi patogią grafinę vartotojo sąsają, leidžiančią lengvai ir paprastai sukurti ETL procesą, bei nustatyti parametrus, leidžiančius atlikti pasirinktus procesus ar antrinius procesus lygiagrečiai.

Lygiagretus ETL proceso vykdymas gali būti naudojamas kartu su padalytomis lentelėmis, jeigu jose saugomas labai didelis duomenų kiekis. Tokiu būdu galima pagerinti didelių lentelių skanavimą bei sujungimą, indeksų kūrimą, padalintų indeksų skanavimą, masinį duomenų įkėlimą, agregavimą ir kopijavimą [28].

3.2.4. Masinis duomenų įkėlimas

Masinio duomenų įkėlimo metu, skirtingai nei įkeliant duomenis įprastiniu būdu, įrašai yra keliami į vieną ar kelias kolekcijas, ir per jas įkeliami duomenys į duomenų bazę. Šis metodas pagerina proceso vykdymą, spartą, nes įprastinio įkėlimo metu kiekvienam įrašui tenka registruoti klaidas atskiruose failuose, kas sulėtina ETL spartą. Masinio įkėlimo metu mažinamas transakcijų registravimas, todėl šis metodas yra plačiai naudojamas. „Oracle“ naudojama „BULK COLLECT“ sintaksė, kuri leidžia įrašyti daug įrašų į vieną kolekcijos tipo kintamąjį. „Microsoft SQL Server“ DBVS kolekcijos nepalaikomos, tačiau naudojama „OPENROWSET“ funkcija su „BULK INSERT“ pasirinkimu, leidžianti išgauti duomenis iš skirtingų duomenų šaltinių per OLE DB tinklo sąsają, o rezultatą atvaizduoti kaip vieną lentelės stulpelį užklauso metu. Siekiant dar labiau pagerinti ETL proceso vykdymą, galima „užrakinti“ lenteles panaudojant užuominą „TABLOCK“, užtikrinanti, jog nebus atliekamos jokios duomenų manipuliavimo operacijos su duota lentele. Dažniausiai ši užuomina naudojama užklausoje,

tačiau galima naudoti ir su „OPENROWSET“ funkcija [28]. Vis dėlto „BULK INSERT“ pasirinkimas leidžia masiškai įkelti duomenis tik iš tekstinių duomenų failų šaltinių, tiesiogiai iš duomenų bazės lentelių išgauti duomenis galima pasinaudojus „SELECT...INTO“ operacija, kuri prieš įrašydama duomenis sukuria naują lentelę, į kurią bus įkeliami duomenys, todėl dažniausiai pasinaudojama vienu iš ETL įrankiu, leidžiančiu pasiekti duomenų šaltinio lenteles, saugomas kitoje DBVS, ir palaikančias masinio duomenų įkėlimo operaciją.

3.2.5. Indeksų panaudojimo galimybės

Indeksu duomenų bazėse vadinama diske saugoma struktūra, leidžianti žymiai sparčiau išgauti duomenis iš lentelės ar vaizdinio. Indeksas apima raktus, susidedančius iš vieno ar daugiau lentelės ar vaizdinio stulpelių. Kiekvienoje DBVS naudojamos skirtingos šių raktų saugojimo struktūros. „Microsoft SQL Server“ DBVS raktai saugomi B- medžio struktūroje, kur duomenys yra saugomi panašiai, kaip ir dvejetainiame medyje, tačiau B- medyje galimi daugiau nei du vaikiniai mazgai [21]. „Oracle“ DBVS duomenys saugomi ne tik B- medžio struktūroje, tačiau ir taškinėje indekso struktūroje, kur kiekvieno įrašo stulpeliui yra sukuriamas dvimatis bitų masyvas, atvaizduojantis indekse esančią reikšmę, padaugintą iš eilučių skaičiaus lentelėje [10].

„Microsoft SQL Server“ DBVS naudojami keli indeksų tipai:

- Klasteriniai indeksai. Šie indeksai surūšiuoja ir saugo duomenys lentelėje arba vaizdinyje pagal indeksų raktines reikšmes. Indeksai organizuojami B- medžio pavidalu, susidedančiu iš šakninio elemento, nuo kurio pradedama duomenų paieška, tarpinio lygio, kuriame saugomos indeksų raktinės reikšmės, ir iš žemiausiame lygmenyje esančių vaikinių elementų, kuriuose yra saugomi lentelių duomenys. Kiekvienas eilutės indeksas savyje turi raktinę reikšmę ir rodyklę į žemesniame lygyje esantį puslapį ar duomenų eilutę, pagal kurią galima atrinkti reikiamus duomenis. Pagrindinė šio indekso savybė yra ta, kad jis surūšiuoja atitinkamos lentelės puslapius bei eilutes pagal jo reikšmę, todėl atliekant atitinkamą DML operaciją duomenys yra kaskart pertvarkomi ir perrūšiuojami. Kadangi klasterinis indeksas surūšiuoja duomenis lentelėje, joje gali būti sukurtas tik vienas šios rūšies indeksas [21, 26].
- Neklasteriniai indeksai. Neklasteriniai indeksai organizuojami toje pačioje B- medžio struktūroje, tačiau jie apima raktines reikšmes ir įrašų vietas, kurios nurodo lentelės įrašų saugojimo vietą, žemiausiame lygmenyje esantys vaikiniai elementai susideda iš puslapiuose esančių indeksų, skirtingai nei klasteriniai indeksai, kur vaikiniai elementai susideda iš duomenų puslapių, ir nerūšiuoja

duomenų eilučių. Šių indeksų struktūra yra atskirta nuo duomenų eilučių, todėl klasterinių indeksų raktinės reikšmės nurodo į duomenų eilutes, kurios apima šias reikšmes [21, 26].

- Filtruoti indeksai. Filtruoti indeksai yra optimizuoti neklasteriniai indeksai, naudojami indeksuoti tik dalį lentelės įrašų, atitinkančių tam tikrus kriterijus, pavyzdžiui, atrinkti tik užpildytas reikšmes, reikšmes, kurios atitinka tam tikrus kriterijus ir pan. Gerai sudarytas indeksas gali pagerinti užklausos vykdymą, sumažinti indekso valdymo bei saugojimo vietas, kurioje yra saugomi šie indeksai, kaštus [21].
- Unikalūs indeksai. Unikalūs indeksai nusako, jog lentelėse negali būti dubliuojamų reikšmių ir kad lentelės įrašai tam tikru būdu yra unikalūs. Tokiu būdu siekiama užtikrinti duomenų vientisumą [21].
- Maišos indeksai. Maišos indeksai naudojami siekiant pasiekti duomenis iš lentelių, saugomų atmintyje, todėl jie naudojami atmintį taupančiose lentelėse. Šie indeksai organizuojami specialių talpyklų, kurios vadinamos tikslinėmis atminties vietomis, masyve, kuriame sutapatinami indeksų raktai su atitinkama tiksline atminties vieta naudojant atitinkamą maišos funkciją. Maišos funkcija yra deterministinė (apskaičiuojama naudojantis indeksų raktų reikšmėmis ir tikslinių atminties vietų kiekiu pagal atitinkamą funkciją) ir visada priskiria vienodas indeksų raktines reikšmes tokioms pačioms tikslinėms atminties vietoms [21].
- Stulpelių formato saugojimo indeksai (angl. *Columnstore*). Šie indeksai saugo stulpelių tipo duomenis skirtingai nei B- medžio struktūra paremtais indeksais, kurie saugo duomenis eilučių formatu. Kuriant šios rūšies indeksus pirmiausia lentelės įrašai skaidomi į įrašų grupes, kurie susideda iš stulpelių segmentų. Tuomet kiekvienas segmentas yra suspaudžiamas įrašų grupėje ir perkeliamas į vietą, kurioje saugomas stulpelis. Tiek klasteriniai, tiek neklasteriniai indeksai gali būti kuriami kaip stulpelių formato indeksai, kur neklasteriniai indeksai gali būti naudojami tik esant duomenų gavimui, o klasterinis indeksas naudojamas kaip kartu su lentele saugomas indeksas, kurioje galima atlikti DML operacijas. Šie indeksai yra populiarūs, kai siekiama sparčiau išgauti tik tam tikrus duomenis, pavyzdžiui, duomenų saugyklose esamų duomenų gavimui [18, 21].

3.2.6. *Vaizdinių panaudojimas*

Vaizdinių duomenų bazėse vadinama iš kelių sujungtų lentelių gaunamų duomenų reprezentacija. Tokio vaizdinio paskirtis yra supaprastinti duomenų užklausas vartotojui, paslėpti šių užklausių bei duomenų sudėtingumą, apsaugoti tam tikrus lentelių stulpelius nuo neteisėtos prieigos. Išskiriamos dvi vaizdinių rūšys: įprasti ir materializuoti (naudojami „Oracle“ DBVS) arba indeksuoti vaizdiniai (naudojami „Microsoft SQL Server“ DBVS). Įprasti vaizdiniai naudojami siekiant paslėpti duomenų užklauskos sudėtingumą, kad vartotojas galėtų paprasčiau parašyti užklausas. Materializuotu vaizdiniu vadinama fizine lentele, kuri naudojama siekiant pagerinti užklauskos vykdymo efektyvumą bei teikti sinchronizuotus duomenis. Šie vaizdiniai gali būti naudojami kaip alternatyva ETL įrankiams. Tokiu atveju, tenka pasinaudoti SQL kalba norint įkelti duomenis iš duomenų šaltinio į galutinę duomenų saugyklą. Materializuoti vaizdiniai taip pat gali būti plačiai naudojami siekiant saugoti suminius arba sutelktus rezultatus duomenų saugyklose, kadangi dažniausiai renkami suminiai duomenys, kuriuos perskaičiuoti esant dideliui duomenų kiekiui užtrunka. Tokie duomenys išgaunami greičiau, nes materializuotuose vaizduose yra perskaičiuojamos sudėtingos užklauskos ir suminiai rezultatai, saugomi atskiroje lentelėje.

Materializuoti vaizdiniai dažniausiai naudojami tose DBVS, kurios teikia didžiausią funkcionalumą duomenų bazių programuotojams. Populiariausios DBVS, kuriose naudojami materializuoti vaizdiniai, yra „Oracle“ ir „Microsoft SQL Server“ DBVS.

Materializuoti vaizdiniai „Oracle“ DBVS

Materializuoti vaizdiniai yra naudojami „Oracle“ DBVS, „Microsoft SQL Server“ DBVS alternatyva materializuotiems vaizdams yra indeksuoti vaizdai, kai kurios kitos DBVS nepalaiko šių vaizdinių, todėl jie turi būti kuriami pačių programuotojų. Šie vaizdiniai apibrėžiami kaip duomenų bazių objektai, kurie saugo rezultatus iš gautų užklausių. Materializuotuose vaizdiniuose, kaip ir įprastose duomenų bazės lentelėse, galima vykdyti užklausas, įterpti, atnaujinti bei šalinti duomenis, kurti indeksus. Materializuoti vaizdai taip pat gali būti skaidomi į atskiras dalis, suspaudžiami, vykdomi lygiagrečiai siekiant efektyvesnio užklauskos vykdymo [6]. Duomenų saugyklose materializuoti vaizdiniai gali būti naudojami kaip apibendrinančios lentelės, kuriose yra atliekami skaičiavimai, lentelių sujungimai duomenų analizei bei ataskaitų kūrimui.

„Oracle“ DBVS teikiami keli materializuotų vaizdinių privalumai:

1. Materializuotuose vaizdiniuose yra perskaičiuojami apibendrinti duomenys, ir perrašomos sudėtingos užklauskos, iš kurių užklausiama sparčiau nei naudojant įprastas užklausas.

2. Galutiniams vartotojams atliekami keitimai materializuotuose vaizdiniuose yra nematomi.

Materializuoto vaizdinio trūkumai:

- Užima nemažai vietos diske.
- Saugojami duomenys būna pasenę tam tikru laikotarpiu iki sekančio atnaujinimo.

Indeksuoti vaizdiniai „Microsoft SQL Server“ DBVS

Indeksuotais vaizdiniais vadinami „Microsoft SQL Server“ DBVS vaizdiniais, kuriuos sudaro lygiai vienas unikalų klasterinį indeksą turintis stulpelis ir objektai, iš kurių atliekamos užklausos, ir kurie saugomi toje pačioje duomenų bazėje. Tokios rūšies vaizdiniai leidžia pagerinti duomenų išgavimą, nes galutiniai rezultatai, kurie yra perskaičiuojami, yra saugomi, o ne tiesiogiai išgaunami. Dažniausiai naudojantis šiais vaizdiniais siekiama gauti apibendrintus duomenis, o tiesioginis jų išgavimas naudojantis užklausa sulėtina sistemos darbą, todėl indeksuotais vaizdiniais, kaip ir materializuotais vaizdiniais, siekiama šiuos rezultatus, gautus iš duomenų, saugoti kietajame diske, nes rezultatai yra perskaičiuojami, kas sumažina užklausos vykdymo kaštus [20].

Teisingai pritaikius indeksuotus vaizdus galima gauti maksimaliai efektyviausius rezultatus, tačiau šių vaizdinių kūrimas yra apribotas: negalimos įdėtinės užklausos, neleidžiama užklausti duomenų iš bent dviejų tų pačių lentelių, kiekvienas duomenų bazės objektas, naudojamas indeksuotame vaizdinyje, turi būti nurodytas konkrečioje schemoje, t. y. jame negali būti užklausiama duomenys iš kitoje sistemoje esančių lentelių, arba negali priklausyti kitai schemai, kurioje nėra sukurtas indeksuotas vaizdinyje, išorinės lentelių sujungimo sąlygos, lentelių užuominos, tokios grupavimo funkcijos, kaip COUNT(*), MAX(), MIN(), ir t. t.

3.2.7. Duomenų suspaudimo metodika

Viena iš DBVS naudojamų savybių yra duomenų suspaudimas. Ši metodika yra naudojama, kai norima pagerinti užklausos vykdymą ir sumažinti fizinių duomenų bazėse saugomų objektų dydžius. „Microsoft SQL Server“ DBVS išskiriamos kelios duomenų suspaudimo rūšys: puslapių ir eilučių [19]. Puslapių suspaudimo metu optimizuojama eilučių saugojimo vieta puslapyje sumažinant pasikartojančius duomenis. Tam yra sudaroma speciali suspaudimo informacinė struktūra, kurioje išskiriamos ir saugomos pasikartojančios reikšmės stulpeliuose ar visuose puslapiuose, bei jų nuorodos į pasikartojantį šabloną. Tuomet vykdoma atitinkamų šablonų paieška, ir radus atitikimus pasikartojančios reikšmės yra pakeičiamos nuorodomis į pasikartojantį šabloną [23]. Eilučių suspaudimo metu duomenys saugomi eilutėje,

kur pastovaus dydžio duomenų tipai saugomi kintamojo tipo formatu sumažinant saugojamų duomenų tipus arba pašalinant tuščius simbolius, jei duomenys yra eilutės tipo [24].

Pagrindinis šios metodikos trūkumas yra tai, kad tiek duomenų suspaudimui, tiek duomenų suspaudimo sumažinimui panaudojamas didesnis kiekis procesorių, tačiau žymiai sumažinamas I/O operacijų kiekis, kai norima gauti duomenis.

3.2.8. Saugomų procedūrų panaudojimas

ETL procesas gali būti atliekamas kviečiant T-SQL, PL/SQL ar kita kalba parašytas procedūras, funkcijas. Tai yra vienas iš dažniausiai naudojamų būdų organizacijose, kurios nenaudoja ETL įrankių. Įkeliant duomenis aprašomi žymekliai (angl. *Cursor*), ciklai ar kiti programavimo elementai, naudojami kartu su SQL kalbos komandomis. Vis dėlto ETL sparta priklauso ne tik nuo parašyto kodo, bet ir nuo SQL kodo, kuris neretai sulėtina duomenų užklausos operacijas, jei yra parašytas netinkamai, todėl norint užklausti duomenis efektyviau, tenka optimizuoti SQL užklausos kodą. Keletas šaltinių teikia rekomendacijas, kaip galima optimizuoti SQL užklausos kodą. Vienas iš dažniausiai naudojamų būdų yra analizuoti užklausos planą, kuriuo galima nustatyti, kiek yra sunaudojama kompiuterio išteklių įvykdžius tam tikrą užklausą. Jeigu užklausos planas ne visada pateikia tinkamas rekomendacijas, tuomet galima naudoti užuominas, kuriais remiantis pateikiamos efektyvesnės instrukcijos užklausos planui. Vis dėlto rekomenduojama naudoti tam sukurtus įrankius, kurie leidžia aptikti problemas užklausose, kurių negali aptikti užklausos planas.

DBVS yra išskiriamos dvių rūšių procedūros:

- Interpretuotos saugomos procedūros.
- Sukompiliuotos saugomos procedūros.

Interpretuotos saugomos procedūros yra standartinės procedūros, kurios saugomos diskuose, kaip ir standartinės duomenų bazių lentelės. Jos yra interpretuojamos ir perkompiliuojamos kiekvieną kartą, kai tik yra paleidžiamos vykdymui.

Sukompiliuotos procedūros saugomos ne diskuose, tačiau operatyvinėje atmintyje. Kaip minėta anksčiau, jos yra naudojamos į atmintį orientuotose OLTP sistemose siekiant pasiekti atmintį taupančias lenteles. Pirmą kartą sukūrus šias procedūras, jų keisti negalima, tik šalinti arba perkurti. Tai paaiškinama tuo, jog jos yra sukompiliuojamos tik vieną kartą jas iškart įvykdžius. Sekantį kartą įvykdžius šias procedūras jos nėra perkompiliuojamos, dėl to neatliekamos papildomos operacijos, dėl kurių gali sulėtėti sistemos darbas [22].

Kartais procedūros gali būti kviečiamos ir iš ETL įrankių, jei norima atlikti veiksmus, kurių negalima atlikti su atitinkamu įrankiu ar norima paspartinti ETL, jei įrankyje naudojamas analogas nėra pakankamai efektyvus.

3.3. ETL įrankiai

ETL procesui įgyvendinti sukurti įrankiai, kurie leidžia patogiai, naudojant grafinę vartotojo sąsają, ir efektyviai įkelti duomenis į duomenų saugyklą. Šiuose įrankiuose dažniausiai yra integruoti ETL optimizavimo metodai, kurių nereikia kurti patiems. Populiariausi ir dažniausiai naudojami ETL įrankiai yra:

1. „Microsoft SQL Server SSIS“ paketas.
2. „Talend Open Studio“.
3. „Informatica PowerCenter“.
4. „Oracle Data Integrator“.
5. „Oracle Warehouse Builder“.
6. „IBM InfoSphere DataStage“.

3.3.1. „Microsoft SQL Server SSIS“ paketas

„Microsoft SQL Server SSIS“ paketas yra platforma, skirta įmonių lygio duomenims integruoti bei pertvarkyti. Šis paketas leidžia atlikti duomenų išgavimo, pertvarkymo ir įkėlimo operacijas, spręsti sudėtingas verslo problemas kopijuojant ar atsisiunčiant failus, siunčiant žinutes el. paštu esant tam tikram įvykiui, atnaujinant duomenų saugyklas, išvalant bei išgaunant duomenis ir valdant „SQL Server“ objektus ir duomenis [15, 27].

„Microsoft SQL Server SSIS“ paketas leidžia:

- Išgauti duomenis iš bet kokių išorinių duomenų šaltinių.
- Pertvarkyti duomenis, pavyzdžiui, konvertuoti duomenų tipus, tikrinti išorinius raktus ir t. t.
- Įkelti duomenis į bet kokį duomenų šaltinį.
- Apibrėžti procesų srautus.

„Microsoft SQL Server SSIS“ privalumai:

- Duomenų saugyklos projektavimas remiasi geriausiomis praktikomis.
- Didelė duomenų įkėlimo iš nuotolinių sistemų vykdymo sparta.
- Standartizuota duomenų integracija.

- Duomenų išgavimas iš bet kurių duomenų šaltinių.
- Puikus palaikymo ir paskirstymo modelis [27].

„Microsoft SQL Server SSIS“ trūkumai:

- Galima naudoti tik „Microsoft Windows“ operacinėse sistemose.
- Grafinė vartotojo sąsaja turi defektų [27].

3.3.2. „Talend Open Studio“

„Talend Open Studio“ yra atvirojo kodo duomenų integravimo platforma, leidžianti patenkinti organizacijos duomenų integravimo poreikius. „Talend Open Studio“ leidžia atlikti duomenų bazių sinchronizacijas, duomenų mainus, ETL verslo intelektikos ar analizės srityse, duomenų migravimą, sudėtingų duomenų pertvarkymą ir įkėlimą [34]. Įrankį sudaro trys pagrindiniai komponentai:

- „Business Modeler“. Ši programa leidžia kurti integravimo procesus ir apžvelgti šių procesų kūrimo vystymąsi. „Business Modeler“ taip pat pergrupuoja visas susijusias dokumentacijas, palaikančias duomenų integravimo procesą patogioje diagramoje [34].
- „Job Designer“. „Job Designer“ pateikia grafinį bei funkcinį integravimo procesų vaizdą naudojant komponentų ir jungčių paletę. Duomenų integravimo procesai kuriami užvedus komponentus ir jungtis į darbo vietą [34].
- „Metadata Manager“. Kadangi „Talend Open Studio“ yra į metaduomenis paremta programinė įranga, jie yra saugomi ir vykdomi pagrindinėje saugykloje – „Metadata Manager“. Ši saugykla centralizuoja visą projektų informaciją ir užtikrina visų duomenų integravimo procesų nuoseklumą. „Metadata Manager“ paremta reliaciniu modeliu, kuriame darbų priklausomybės gali būti lengvai atpažintos palengvinant duomenų integravimo vykdymą, duomenų migravimo ir duomenų sinchronizavimo darbus [34].

„Talend Open Studio“ privalumai:

- Atvirojo kodo programinė įranga, kurią galima atsisiųsti nemokamai.
- Leidžiama prieiti prie šaltinio kodo ir jį plėtoti.
- Paprasta naudotis tiems, kurie dirba JAVA platformoje, ir gali būti lengvai vykdomi procesai vietiniame serveryje.
- Draugiška vartotojo sąsaja ir kitos funkcijos, kurių nėra kitose ETL įrankiuose.

- Teikia lanksčius programinės įrangos sprendimus pagal skirtingus verslo modelius nepaisant jų apimties [34].

„Talend Open Studio“ trūkumai:

- Įrankis gali būti sudėtingas tiems vartotojams, kurie nedaug išmano apie duomenų integracijos procesą.
- Įrankis nėra tinkamas, kai norima apdoroti itin didelio kiekio duomenis [34].

3.3.3. „Informatica PowerCenter Express“

„Informatica PowerCenter Express“ yra įmonėse naudojama duomenų integravimo platforma, teikianti patikimus sprendimus IT valdymui, globalioms IT komandoms, kūrėjams ir verslo analitikams, kadangi ji ne tik teikia patikimus duomenis, bet ir palaiko įvairiausius duomenų integravimo projektus ir bendradarbiavimą tarp IT ir verslo. Šio įrankio dėka galima prisijungti prie bet kokių duomenų šaltinių, atlikti dinaminį skaidymą, atstatymą, metaduomenų valdymą, duomenų maskavimą [14]. Kaip ir „Talend Open Studio“, „Informatica PowerCenter Express“ susideda iš trijų pagrindinių komponentų:

- „Informatica PowerCenter Client Tools“ kliento įrankiai. Naudojantis šiais įrankiais kūrėjai gali valdyti saugyklą, prižiūrėti sesijų vykdymą, apibrėžti objektų sutapatinimo ir vykdymo ypatumus ir t. t [14].
- „Informatica PowerCenter Repository“ saugykla. Tai yra centrinė vieta, kurioje saugomi visi duomenys. Visi kliento įrankiai, kaip ir „Informatica“ serveris, naudoja saugyklą duomenų gavimui. Saugykla gali būti interpretuojama kaip kietasis diskas arba kompiuterio atmintis [14].
- „Informatica PowerCenter Server“ serveris. Šiame serveryje atliekami visi su duomenų integravimu susiję veiksmai. Serveris fiziškai prisijungia prie duomenų šaltinių ir kitų duomenų saugyklų, kad gautų duomenis, atliktų reikiamus pakeitimus ir įkeltų duomenis į duomenų sistemas [14].

„Informatica PowerCenter“ privalumai:

- „Informatica PowerCenter“ yra labai specializuota ties ETL.
- Gali veikti tiek „Windows“, tiek „Unix“ sistemose.
- Labai našus įrankis.
- Naudojamas pakitusių duomenų fiksavimas – išgaunami tik pakitę duomenys [14].

„Informatica PowerCenter“ trūkumai:

- Labai brangi programinė įranga.
- Nėra tinkama mažoms arba vidutinio dydžio įmonėms [14].

3.3.4. „Oracle Warehouse Builder“

„Oracle Warehouse Builder“ įrankis leidžia kurti verslo intelektikos programas, atlikti ETL proceso projektavimą, išgauti duomenis iš heterogeninių sistemų ir teikti metaduomenų pranešimo funkcijas. Vartotojai gali kurti tiek dimensinius, tiek reliacinius duomenų modelius, suprojektuoti duomenų saugyklos architektūrą remiantis „žvaigždės“ schemas principu, tarpines duomenų saugyklas, ETL procesus, vartotojų priejimo sluoksnius [16].

„Oracle Warehouse Builder“ palaiko tokias duomenų sistemas kaip: „Oracle“ duomenų bazė, iškelties failai, „SQL Server“, „Informix“, „Sybase“, „SAP“ ir kitos sistemos prieinant prie šių šaltinių per „Oracle“ atvirąją tinklų sąsają, ODBC, universalius kompiuterius.

„Oracle Warehouse Builder“ privalumai:

- Stipri sąsaja tarp „Oracle“ programų.
- Visų įrankių integracija į vieną programą ir aplinką [16].

„Oracle Warehouse Builder“ trūkumai:

- Skiriama daugiau dėmesio ETL, o ne duomenų valdymui.
- Įrankiai daugiausiai naudojami įkeliant duomenis neprisijungus [16].

3.3.5. „Oracle Data Integrator“

„Oracle Data Integrator“ yra „Oracle“ kompanijos sukurtas duomenų išgavimo, pertvarkymo bei įkėlimo įrankis, leidžiantis kurti, vykdyti, valdyti duomenų integravimo procesus verslo intelektikos sistemose. Šis įrankis geba įkelti didelių apimčių duomenis, pertvarkyti, įkelti duomenis į duomenų saugyklas. Viena iš šio įrankio savybių yra tai, jog atliekamas ne nuoseklus ETL procesas, o duomenų išgavimo, tuomet duomenų įkėlimo ir pertvarkymo procesas, kas leidžia papildomai nenaudoti atskiro ETL serverio norint optimizuoti ETL procesą [30].

„Oracle Data Integrator“ architektūra susideda iš kelių komponentų: „Repository“ (saugykla), „Topology Manager“, „Designer“, „Security Manager“, „Operator“, „Agent“, „Scheduler“ ir „Metadata Navigator“.

Pagrindinis „Oracle Data Integrator“ architektūros komponentas yra saugykla, kuri prieinama kliento-serverio ar „plono“ kliento režime. Ši saugykla yra visiškai atvira, lanksti,

prieinama programoms, kurios atsakingos už pakeitimus, projektų vystymą, integraciją ir informacijos dalijimąsi. „Topology Manager“ yra grafinės vartotojo sąsajos modulis, valdantis visos fizinės bei loginės sistemos infrastruktūrą. „Designer“ įrankis skirtas kurti duomenų pertvarkymo procesą ir tikrinti duomenų integraciją. Vienos iš pačių svarbiausių funkcijų šiame įrankyje yra duomenų srauto atvaizdavimas, kūrimas, duomenų integracijos bei pertvarkymų palaikymas, saugyklos duomenų arba programų atbulinė inžinerija. „Security Manager“ įrankis valdo vartotojų profilius, prieigos prie duomenų teises, privilegijas. „Operator“ modulis valdo bei prižiūri duomenų integravimo procesus gamybinėje aplinkoje. Juo galima matyti vykdymo pranešimus su klaidomis ar be jų, kiek įrašų buvo įkelta, koks kodas buvo vykdomas ir t. t. „Agent“ yra autonominė programa, kuri vykdo programavimo sąsajas, kurias „Scheduler“ modulis suplanuoja, kada jas vykdyti. „Metadata Navigator“ atliekama saugyklos turinio paieška. Šis navigatorius yra orientuotas į „ploną“ klientą paieškos vykdymui bei analizei, be to, jis gali valdyti visus produkcijos procesus, gali sekti ir dalinai nurodyti duomenų gyvavimo ciklą [30].

„Oracle Data Integrator“ privalumai:

- Didelis našumas.
- Paprastesnė ETL architektūra.
- Duomenų kokybės ugniasienė, leidžianti automatiškai registruoti klaidingus duomenis [30].

„Oracle Data Integrator“ trūkumai:

- Kaip ir „Oracle Warehouse Builder“, daugiausia dėmesio skiriama ETL, o ne duomenų valdymui, bei daugiausiai naudojama įkeliant duomenis neprisijungus.
- Gali užtrukti nemažai laiko norint įgusti dirbti su šiuo įrankiu [30].

3.3.6. „IBM InfoSphere DataStage“

„IBM InfoSphere DataStage“ yra pagrindinis „IBM InfoSphere Information Server“ duomenų integravimo platformos produkto modulis. Šis įrankis leidžia prisijungti prie operacinių sistemų, duomenų bazių ir įmonių programų per universaliuosius kompiuterius ir paskirstytas sistemas, naudoja metaduomenų saugyklą, leidžiančią integruoti kitus „InfoSphere Information Server“ modulius, palaiko SOA architektūros metodologiją, leidžia kurti dinaminėje lygiagrečioje vykdymo infrastruktūroje, leidžiančioje vartotojams projektuoti ir įdiegti procesus vykdymo metu nekeičiant jokių integravimo darbų, pristato duomenis realiu laiku siekiant optimizuoti išgavimo, pertvarkymo bei įkėlimo procesus [13].

„IBM Infosphere DataStage“ privalumai:

- Lanksti kūrimo aplinka.
- Lygiagretus procesų vykdymas (galima tik „Enterprise“ versijoje) [13].

„IBM Infosphere DataStage“ trūkumai:

- Dideli architektūriniai skirtumai tarp „Server“ ir „Enterprise“ versijų, todėl migravimas iš vienos versijos į kitą gali reikalauti daug išteklių ir laiko.
- Nėra automatinio klaidų valdymo ir atstatymo mechanizmo.
- Ganėtinai brangi mažoms arba vidutinio dydžio kompanijoms [13].

3.1 lentelė. ETL įrankių palyginimas [3, 35]

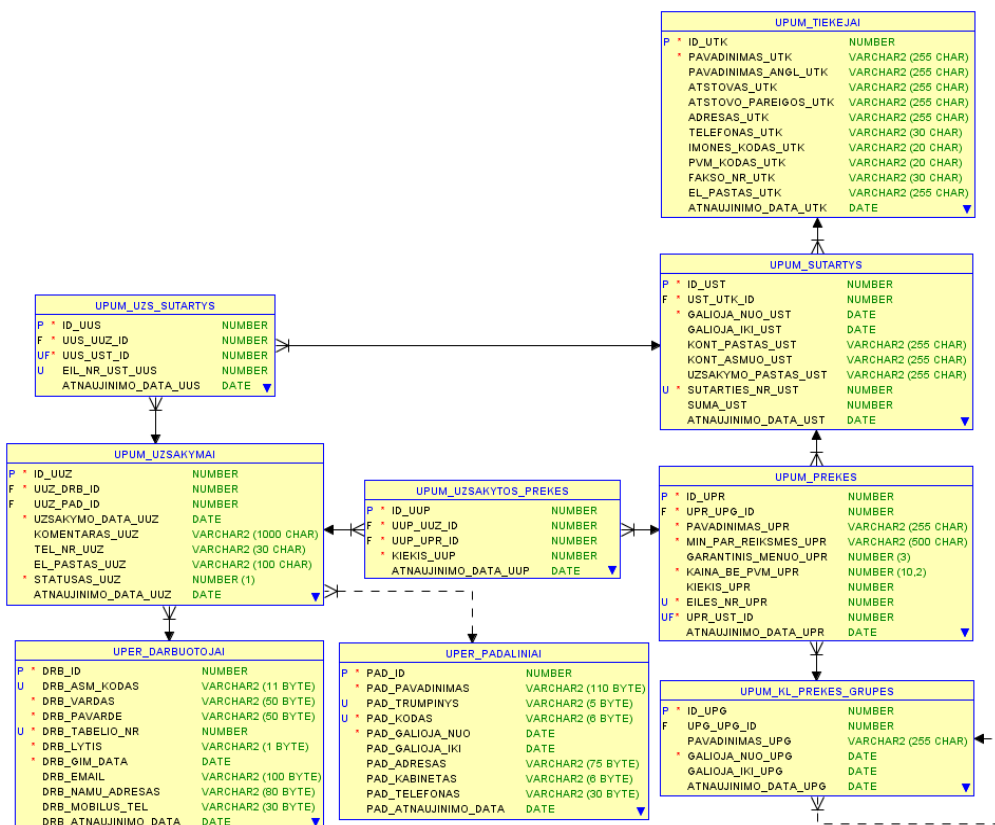
	„Microsoft SQL Server SSIS“ paketas	„Oracle Warehouse Builder“	„Oracle Data Integrator“	„Talend Open Studio“	„Informatica PowerCenter“	„IBM InfoSphere DataStage“
OS palaikymas	„Microsoft“ OS	Daugiaplatformė	Daugiaplatformė	Daugiaplatformė	Daugiaplatformė	Daugiaplatformė
PĮ rūšis	Komercinė	Komercinė	Komercinė	Atvirojo kodo	Komercinė	Komercinė
Didelių duomenų palaikymas	Ne	Taip	Taip	Taip	Taip	Taip
Naudojimasis programa	Aukšto lygio. „Microsoft Visual Studio“ vartotojo sąsaja	Vidutinio lygio.	Vidutinio lygio. Gali užtrukti perprasti įrankį	Labai aukšto lygio, patogi vartotojo sąsaja	Labai aukšto lygio. Aiški grafinė vartotojo sąsaja, paprasta naudotis įrankiu	Aukšto lygio. Nesudėtinga išmokyti naudotis įrankiu
Versijų kontrolės palaikymas	Yra	Yra	Yra	Yra	Yra	Yra
Licencija	Dalis „SQL Server“, licencijos (pagal procesoriaus branduolių kiekį)	Licencija pagal procesoriaus branduolių kiekį/metus	Vartotojo licencija arba licencija pagal procesoriaus branduolių kiekį	Atvirojo kodo licencija („Apache“ licencija)	Licencija kiekvienam serveriui	Licencija pagal procesoriaus branduolių kiekį
Derinimo režimo galimybės	Lūžio taškai, duomenų peržiūra, klaidų kontrolė	Lūžio taškai	Simuliacijos režimas	Derinimo režimas realiu laiku	Lūžio taškai, duomenų analizė, sesijos derintuvas	Palaiko derinimo įrankius testavimui
Darbų kontrolė ir vykdymas	Darbo agentas (angl. <i>Job Agent</i>)	Nėra (Reikalingas „Oracle scheduler“)	Yra	„Talend Open Studio“ darbų eksportavimas, „kevalo“ įrenginio rašmenų paleidimas	Yra	Nėra (Galima kviesti programas, kai darbas įvykdytas)
Netinkamų įrašų kontrolė	Klaidos fiksuojamas „*.csv“ failuose arba lentelėse	Klaidų lentelės	Integravimo žinių moduliai, tikrinimo žinių moduliai	„tMap“ komponentas	Netinkami duomenys (failai) atmetami, atmesti įrašai fiksuojami	Yra

4. ETL proceso tyrimas

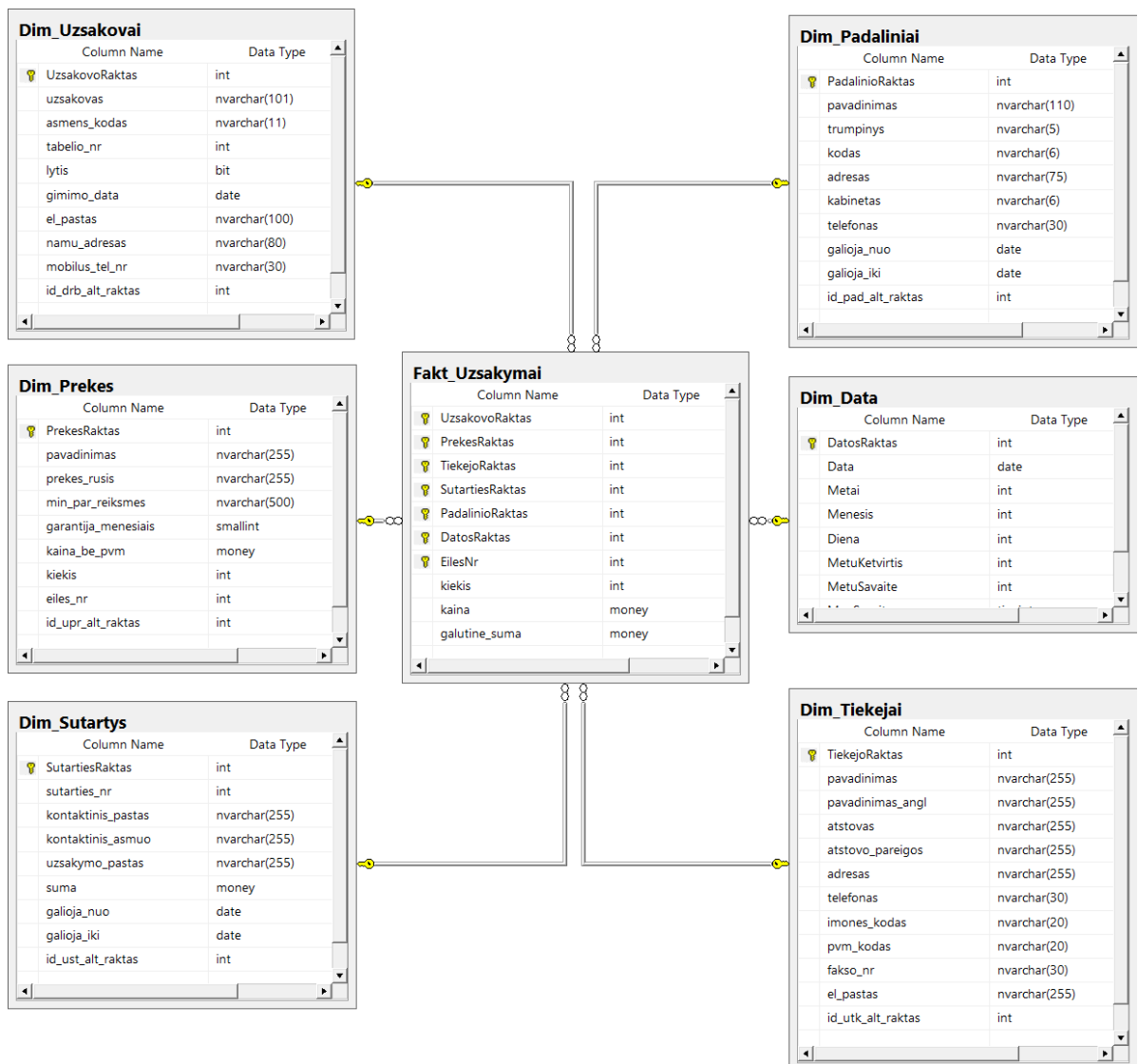
Kaip jau buvo minėta anksčiau, šiam tyrimui yra panaudotos dvi RDBVS technologijos: „Oracle“ bei „Microsoft SQL Server“, – kadangi tai yra plačiausiai naudojamos sistemos IT ir kitose įmonėse. Pirmajame etape atliekamas duomenų įrašymo tyrimas iš „Oracle“ DBVS į „Microsoft SQL Server“ DBVS. Antrajame etape atliekamas ETL procesas tiek įrašant naujus, tiek atnaujinant esamus duomenis iš duotų DBVS sistemų. Galutiniame etape palyginami skirtingi duomenų įkėlimo metodai ir technologijos su skirtingais duomenų kiekiais.

Duomenys yra įkeliami naudojant „Microsoft SQL Server SSIS“ paketą, kuris yra integruotas kartu su „Microsoft Visual Studio 2015“ vartotojo aplinka, kadangi tai yra universalus įrankis, galintis įkelti duomenis iš bet kokių duomenų šaltinių ir įkelti duomenis į bet kokią duomenų saugyklą. Prieš vykdant ETL procesą, sukuriama naujas Microsoft Visual Studio projektas ir jame esantys paketai, kuriais atliekamas duomenų išgavimas, pertvarkymas bei įkėlimas.

ETL testavimui panaudotos VGTU atliekamų užsakymų verslo modelio duomenų bazės: užsakymų duomenų bazė kaip duomenų šaltinis bei užsakymų duomenų saugykla kaip šaltinis, į kurį įkeliami užsakymų duomenys (žr. 4.1, 4.2 pav.).



4.1 pav. Reliacinė užsakymų duomenų bazės schema



4.2 pav. Užsakymų duomenų saugyklos schema

4.2. Duomenų įrašymo proceso tyrimas

Duomenų įrašymas yra viena iš svarbiausių duomenų manipuliavimo operacijų. Nauji duomenys visą laiką įrašomi į duomenų saugyklą, todėl iš pradžių atliekamas duomenų įrašymo tyrimas: atliekamas pradinis naujų duomenų įkėlimas. Vėliau atliekamas ETL proceso tyrimas perkeltiant tiek naujus, tiek atnaujintus duomenis.

„Microsoft SQL Server“ DBVS yra sukurtos dvi duomenų bazės: viena iš jų yra tarpinė duomenų bazė (parengiamasis sluoksnis), skirta atlikti duomenų pertvarkymui ir paruošimui įkelti į duomenų saugyklą, antra yra duomenų saugyklos struktūra, kurioje saugomos dimensijų ir faktų lentelės, į kurias įkeliami galutiniai duomenys.

Tarpinėje duomenų bazėje yra sukurti duomenų bazės objektai, kurie paruošti duomenų išgavimui, pertvarkymui ir įkėlimui į duomenų saugyklą: sukurtos lentelės, atitinkančios duomenų šaltinyje esančias lenteles, vaizdiniai, kurie išgauna duomenis iš tarpinio sluoksnio

lentelių, saugomos procedūros, kurios įrašo duomenis į parengiamąjį sluoksnį ir įkelia juos į duomenų saugyklą.

Atskirai sukurta ir dar viena duomenų bazė, kurioje saugomos dimensijų ir faktų lentelės. Šioje duomenų bazėje vienintelis skirtumas yra tas, kad faktų lentelė „Fakt_Uzsakymai“ yra padalyta, t. y. atliekamas tyrimas ir įkeliant duomenis į padalytą faktų lentelę.

Pakitusių duomenų registravimui naudojamas kiekvienos lentelės paskutinį kartą modifikuoto įrašo datos laukas ir jį generuojantis trigeris. Ši pasirinkimą sąlygojo kelios priežastys: visų pirma, pritaikant šią metodologiją esamai duomenų basei duomenys nebus šalinami, todėl neatsiras problemų dėl to, jog naujas įrašas nebus įkeltas į duomenų saugyklą (ypač tai įtakoja faktų užsakymų lentelę); visų antra, tokiu būdu iš tos pačios lentelės duomenis bus paprasčiau atrinkti, ypač, jei filtravimui bus naudojamas prie šio lauko pridėtas indeksas. Trigerių panaudojimas įtakoja duomenų bazės vykdymą, tačiau, jei keičiamas minimalus laukų kiekis arba vykdomas paprastas kodas, tuomet trigeris duomenų bazės vykdymą įtakos minimaliai. Nepaisant šio metodo privalumų, šis laukas nėra patikimas esant duomenų šalinimui, todėl tokiose sistemose, kur atliekamos visos galimos duomenų manipuliavimo operacijos, naudingiausia registruoti visus esamus pakeitimus atskirose lentelėse, ir jomis naudojantis išgauti paskutinį kartą modifikuotus duomenis.

„Microsoft SQL Server SSIS“ yra sukurtas projektas, kuriame įgyvendinti duomenų įrašymo ir duomenų įrašymo bei atnaujinimo procesai. ETL tyrimui naudojami įrašai siekia 100 000 tūkst. Duomenų įrašymui ir atnaujinimui panaudota 50 000 naujų bei 50 000 esamų modifikuotų įrašų.

Duomenys įkeliami dviem būdais: tiek visišku, tiek pavieniu būdu. Visų pirma, įkeliami visi duomenys, kur tarpusavyje yra palyginamos skirtingos naudotos technologijos. Tuomet atliekamas pavienis duomenų įkėlimo būdas. Pavienis duomenų įkėlimas atliekamas tiek nenaudojant tarpinių lentelių, tiek naudojant tarpines lenteles, tačiau, kai duomenys yra įkeliami į parengiamąsias lenteles, pakanka ištirti panaudojus vieną duomenų įkėlimo metodą, kol naudojant konkrečias technologijas šis metodas neturės jokios įtakos, skiriasi tik įkeliamų duomenų kiekis į šias lenteles. Tyrimo metu yra pateikiami grafikai taikant skirtingas metodologijas ir technologijas, ir pateikiami komentarai, kuriais atvejais duomenys yra įkeliami sparčiau.

4.2.1. Pasirinktos technologijos tyrimui

4.1 lentelė. Tyrimui pasirinktos technologijos

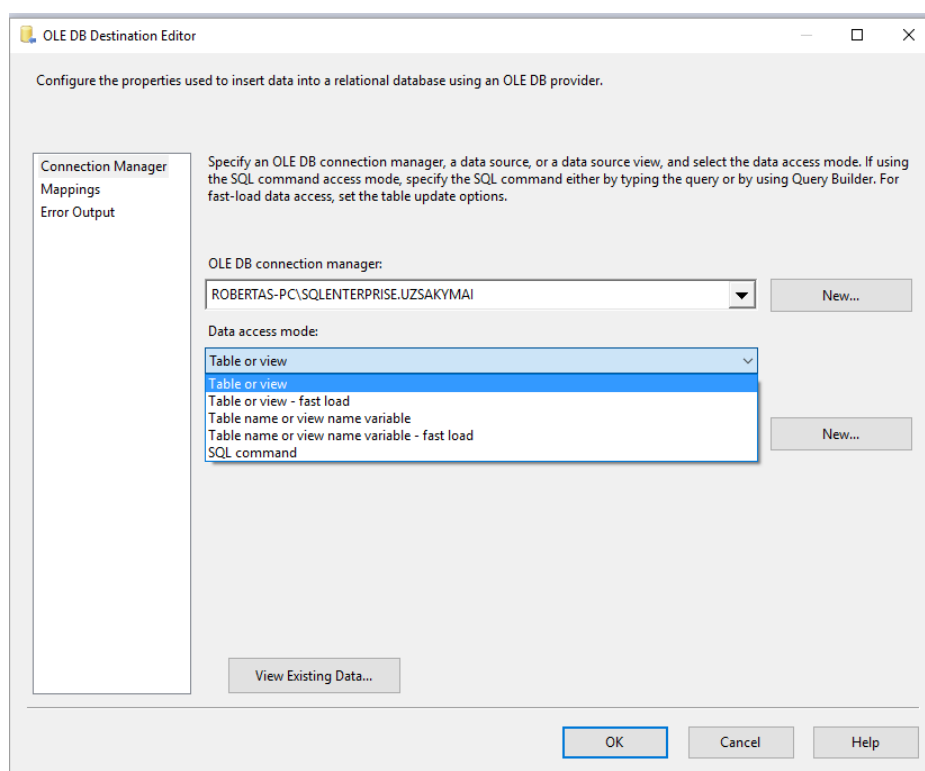
Kategorija	Programinė įranga
Operacinė sistema	„Windows 10 64bit“

4.1 lentelės tęsinys

Techninė įranga	„Intel(R) Core(TM) i5-2410M CPU @ 2.30Ghz“ procesorius, „4,00 Gb RAM“ operatyvinė atmintis
Duomenų šaltinio DBVS	„Oracle 12c Enterprise Edition 12.1.0“
Duomenų saugyklos kūrimo ir įgyvendinimo DBVS	„Microsoft SQL Server 2014 Enterprise Edition“
ETL įrankis	„Microsoft SQL Server Integration Services“ paketas
ETL vartotojo aplinka	„Microsoft Visual Studio 2015 Community Edition“
Duomenų pakitimų registravimo realizacijai pasirinkti įrankiai	Auditinis datos laukas, nurodantis, kada paskutinį kartą buvo keičiamas įrašas, ir trigeriai, kurie nustato paskutinio įrašo keitimo datą

4.2.2. Duomenų įkėlimo parinkčių pasirinkimas

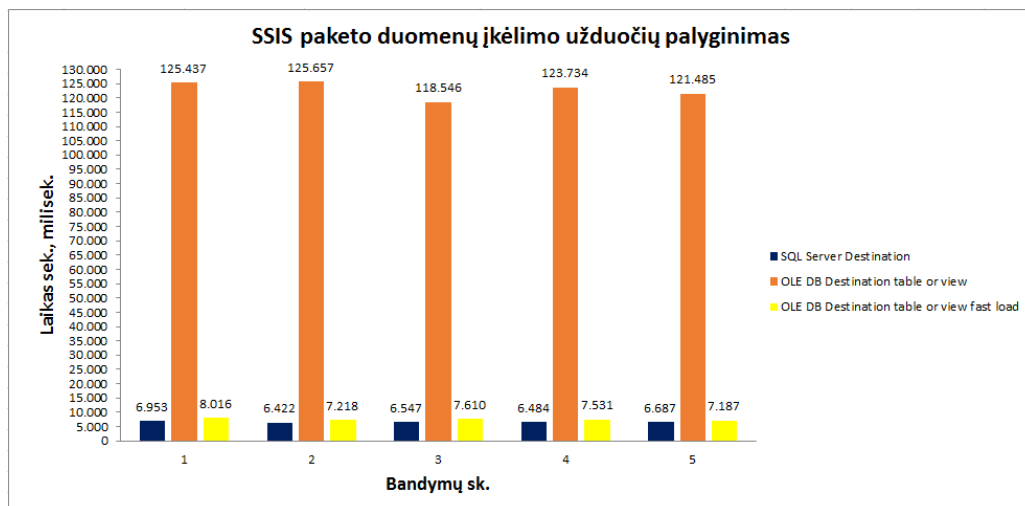
Duomenys „Microsoft SSIS“ pakete gali būti įkeliami keliais būdais: naudojantis tiek „OLE DB Destination“, tiek „SQL Server Destination“ užduotimis. Naudojant „OLE DB Destination“ galimi keli pasirinkimai: „Table or view“ arba „Table or view – fast load“ (žr. 4.3 pav.).



4.3 pav. „OLE DB Destination“ parinktys

Kadangi duomenų įkėlimas vykdomas į lenteles, kitos parinktys nėra svarstomos. Pirmoji parinktis nusako, kad kiekvieną kartą, kai tam tikras įrašas yra įkeliamas į duomenų saugyklą, atliekamas transakcijos patvirtinimas. Antroji parinktis nusako, kad duomenys bus įkeliami masiškai, kurio metu galima nurodyti, kiek įrašų gali būti perduodama ir patvirtinta vienos

transakcijos metu. „SQL Server Destination“ vykdo masinio duomenų įkėlimo operaciją. Lyginant su „OLE DB Destination“ naudojantis šia užduotimi duomenys yra įkeliami sparčiau, nes jungiantis prie „SQL Server“ duomenų bazės naudojama bendroji atmintis, kuri nenaudoja papildomų konfigūracijų, apkraunančių prisijungimą prie sistemos nuotoliniu būdu. Pagrindinis šios atminties požymis yra tai, kad duomenys į ją perduodami be papildomų komunikavimo apkrovų, todėl programavimo sąsaja, kuri realizuoja duomenų įkėlimą į esamą sistemą naudojant bendrąją atmintį, ir esama sistema turi būti viename serveryje. „OLE DB Destination“ naudojami TCP/IP bei „išvardintų kanalų“ protokolai, kuriais SSIS paketas gali įrašyti duomenis ir į nuotolinę duomenų bazę. Taigi „SQL Server Destination“ užduotis yra naudinga, jei paketas ir duomenų saugykla, į kurią įrašomi duomenys, yra tame pačiame serveryje, o „OLE DB Destination“ naudinga tuomet, kai duomenų saugykla ir paketas nebūtinai yra tame pačiame serveryje.



4.4 pav. Duomenų įkėlimo užduočių palyginimas

4.4 pav. esantis grafikas parodo, koks yra skirtumas tarp duomenų įkėlimo užduočių, kai įkeliami 100 000 įrašų. Pagal pateiktą grafiką duomenų įrašymo tyrimams panaudota „SQL Server Destination“ užduotis.

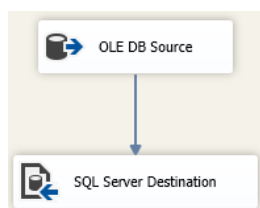
Duomenų įrašymui ir atnaujinimui naudojama T-SQL operacija „MERGE“, kuri palygina visus esamus įrašus ir nustato, ar turi būti atnaujinamas, ar įrašomas naujas įrašas. Ši operacija naudojama ir SSIS pakete, tačiau remiantis atliktais bandymais nustatyta, kad atnaujinant vienos lentelės duomenis, kai atnaujinama 50 000 įrašų, paketo vykdymas užtrunka keliolika minučių, kadangi naudojama OLE DB komanda, kurioje vykdoma atnaujinimo operacija, atliekanti kiekvieno įrašo atnaujinimą, kas sulėtina viso proceso darbą, todėl naudojama procedūra, kuri vykdo T-SQL kalba parašytą „MERGE“ operaciją.

4.2.3. Nuoseklus duomenų įrašymas

Nuoseklus duomenų įrašymo metu duomenys į lenteles yra įrašomi pasibaigus ankstesniam duomenų įrašymo procesui į konkrečią lentelę. Įkeliant duomenis į dimensių lenteles duomenų srautų vykdymo tvarka nesvarbi, tačiau duomenis į faktų lentelę yra privaloma įkelti tik tuo atveju, jei visi duomenys yra įkelti į dimensių lenteles. Duomenų įrašymo į duomenų saugyklą proceso realizacija pateikta 4.5 ir 4.6 pav.

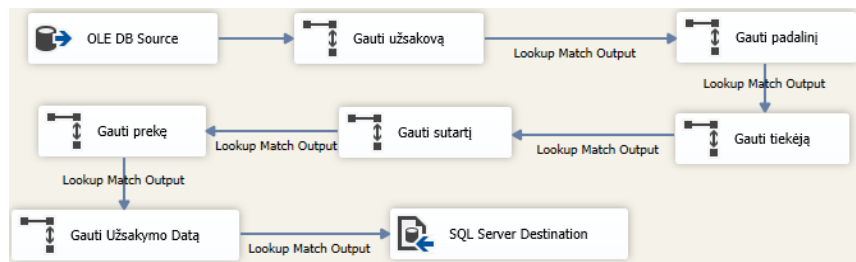


4.5 pav. Nuoseklus duomenų įrašymas



4.6 pav. Duomenų srauto įrašymo procesas

Prieš įkeliant duomenis į užsakymų faktų lentelę vykdoma atitinkamų išorinių raktų paieška iš dimensių lentelių, kad būtų galima susieti tarpusavio ryšius (žr. 4.7 pav.).



4.7 pav. Duomenų įrašymas į faktų lentelę

Įvykdžius procesą SSIS pakete galima matyti, kokie procesai buvo įvykdyti sėkmingai, o kurie – nesėkmingai, ir per kiek laiko užtruko šie procesai (žr. 4.8, 4.9 pav.).

```

Task Duomenų įrašymas
  Validation has started (2)
  [SSIS.Pipeline] Information: Validation phase is beginning.
  Progress: Validating - 0 percent complete
  Progress: Validating - 50 percent complete
  Progress: Validating - 100 percent complete
  Validation is completed (2)
  Start, 09:17:43
  [SSIS.Pipeline] Information: Validation phase is beginning.
  Progress: Validating - 0 percent complete
  Progress: Validating - 50 percent complete
  Progress: Validating - 100 percent complete
  [SSIS.Pipeline] Information: Prepare for Execute phase is beginning.
  Progress: Prepare for Execute - 0 percent complete
  Progress: Prepare for Execute - 50 percent complete
  Progress: Prepare for Execute - 100 percent complete
  [SSIS.Pipeline] Information: Pre-Execute phase is beginning.
  Progress: Pre-Execute - 0 percent complete
  Progress: Pre-Execute - 50 percent complete
  Progress: Pre-Execute - 100 percent complete
  [SSIS.Pipeline] Information: Execute phase is beginning.
  [SSIS.Pipeline] Information: Post Execute phase is beginning.
  Progress: Post Execute - 0 percent complete
  Progress: Post Execute - 50 percent complete
  Progress: Post Execute - 100 percent complete
  [SSIS.Pipeline] Information: "SQL Server Destination" wrote 2556 rows.
  [SSIS.Pipeline] Information: Cleanup phase is beginning.
  Progress: Cleanup - 0 percent complete
  Progress: Cleanup - 50 percent complete
  Progress: Cleanup - 100 percent complete
  Finished, 09:17:45, Elapsed time: 00:00:02.046
  Task Padalinių duomenų įrašymas
  Task Prekių įrašymas
  Task Sutarčių duomenų įrašymas
  Task Tiekėjų duomenų įrašymas
  
```

4.8 pav. Kiekvieno antrinio duomenų įrašymo proceso trukmė

```

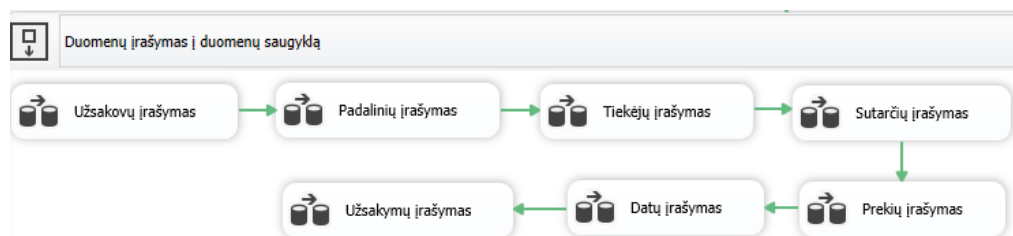
  Validation has started
  Duomenų įrašymas į duomenų sandėlį
  Validation has started
  Task Duomenų įrašymas
  Task Padalinių duomenų įrašymas
  Task Prekių įrašymas
  Task Sutarčių duomenų įrašymas
  Task Tiekėjų duomenų įrašymas
  Task Užsakymų įrašymas
  Task Užsakovų duomenų įrašymas
  Validation is completed
  Start, 09:17:27
  Finished, 09:17:57, Elapsed time: 00:00:30.297
  Validation is completed
  Start, 09:17:27
  Finished, 09:17:57, Elapsed time: 00:00:31.000
  
```

4.9 pav. Viso duomenų įrašymo proceso trukmė

Duomenys nuosekliai įkeliami ir naudojant tarpines lenteles. Šiuo atveju duomenys iš „Oracle“ DBVS yra įkeliami į „Microsoft SQL Server“ DBVS parengiamąjį sluoksnį, kuriame apibrėžtos tokios pačios struktūros, kaip ir „Oracle“ DBVS, lentelės, atliekami atitinkami duomenų pakeitimai, ir jų įkėlimas į duomenų saugyklą (žr. 4.10, 4.11 pav.).



4.10 pav. Nuoseklus duomenų įkėlimas į tarpines lenteles

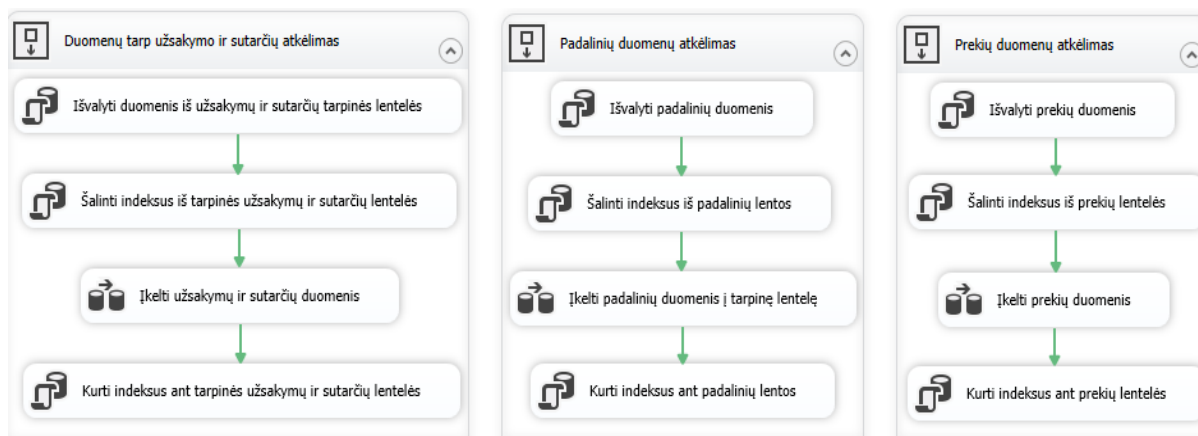


4.11 pav. Nuoseklus duomenų įkėlimas į duomenų saugyklos lenteles

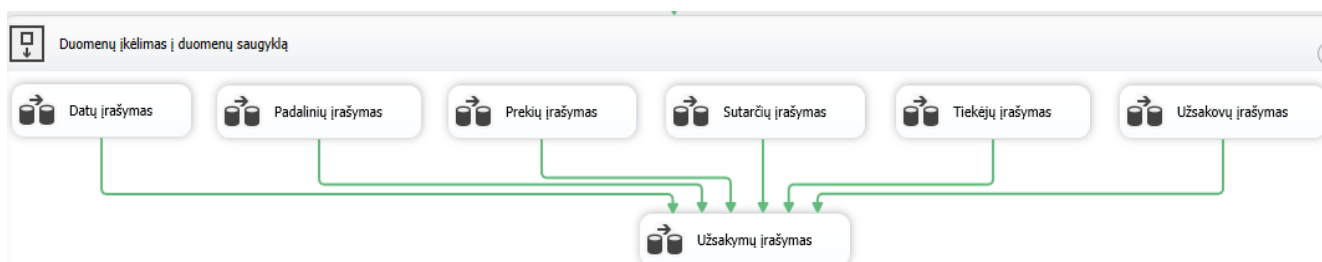
Pateiktame pavyzdyje prieš įkeliant duomenis šalinami lentelėse sukurti indeksai, kurie paspartina duomenų išgavimo užklausas. Tuomet įkėlus duomenis indeksai yra perkuriami sekančiam etapui. Tyrimo metu yra tikrinama, kai duomenys yra įkeliami į lenteles su konkrečios rūšies indeksais, jų nenaudojant ir pan.

4.2.4. Lygiagrečius duomenų įrašymas

Šio proceso įgyvendinimui sukuriamas SSIS paketas, kuriame atliekamas duomenų įrašymas į atskiras lenteles lygiagrečiai. Pagrindinis skirtumas projektuojant procesą yra tai, kad tarp užduočių nėra sujungtos rodyklės, nurodančios, kokia seka turi būti vykdomos užduotys (žr. 4.12, 4.13 pav.). Taip pat tam, kad duomenų įkėlimas būtų vykdomas lygiagrečiai, turi būti nustatytas paketo parametras „MaxConcurrentExecutables“, kuris nurodo, kiek vienu metu bus vykdomos užduotys. Šis parametras susideda iš to, kiek procesorių veikia kompiuteryje pridėdant 2. Kadangi šiame kompiuteryje yra keturi procesoriai, tuomet galima vykdyti 6 procesus vienu metu. Lygiai toks pat procesas atliekamas, kai duomenys iš pradžių yra įkeliami į parengiamąjį sluoksnį. Kadangi lentelių, iš kurių gaunami duomenys, yra daugiau nei esamų procesorių kompiuteryje, lygiagrečiai bus vykdomos daugiausiai 6 užduotys. „Microsoft SSIS“ paketas pats nusprendžia, kokios užduotys bus vykdomos lygiagrečiai.



4.12 pav. Lygiagretus duomenų įkėlimas į tarpines lenteles

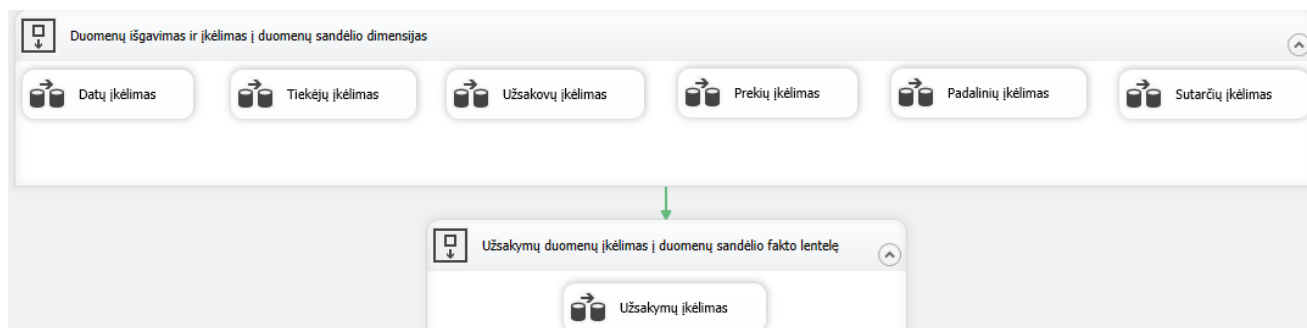


4.13 pav. Lygiagretus duomenų įkėlimas į duomenų saugyklos lenteles

4.2.5. Duomenų įrašymas naudojant lentelių tipo struktūras

Parengiamajame sluoksnyje yra sukurti vartotojų apibrėžti lentelių tipai, kuriais remiantis galima perduoti gaunamus duomenis iš skirtingų duomenų šaltinių į lentelių tipo kintamąjį, ir procedūros, kuriose apibrėžtas perduodamas lentelių tipo parametras, ir naudojantis šio parametro rezultatais atliekamas duomenų įkėlimas į duomenų saugyklą.

Duomenų įkėlimo į parengiamąjį sluoksnį procesas yra identiškas procesui, kuris vykdomas įkeliant duomenis lygiagrečiai (žr. 4.14 pav.). Visi kiti procesai, kuriuose vykdomas duomenų įkėlimas į tarpinę duomenų bazę, yra identiški, nes duomenų įkėlimo į duomenų saugyklą procesas nepriklauso nuo to, kaip įkeliami duomenys į parengiamąsias lenteles.

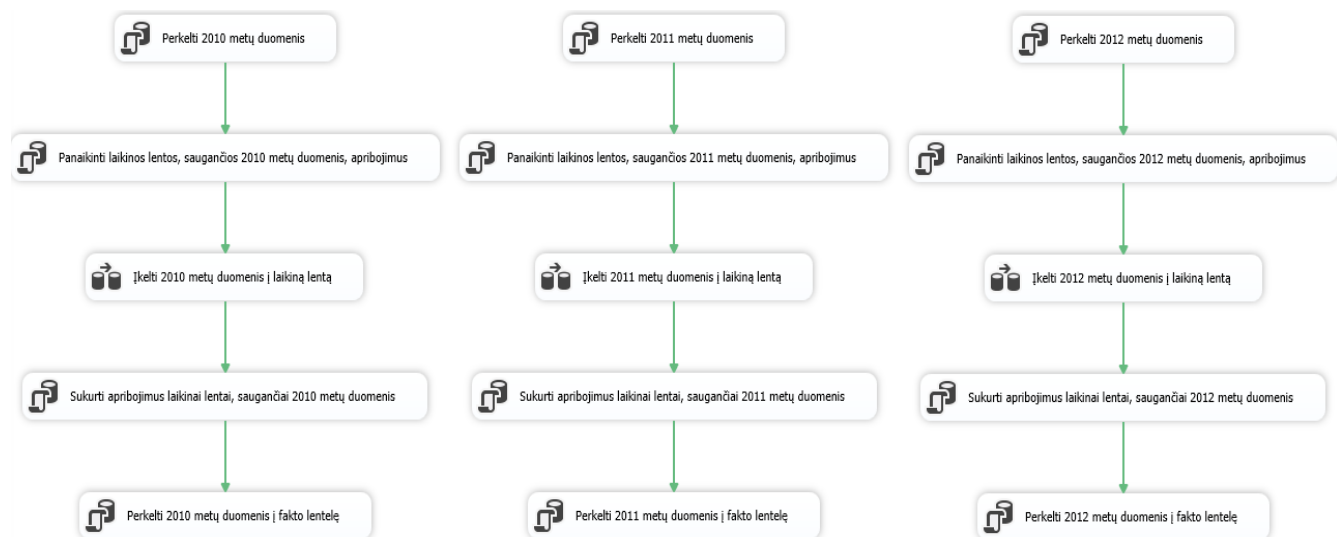


4.14 pav. Duomenų įkėlimas naudojant lentelių tipo struktūras

4.2.6. Duomenų įrašymas naudojant padalijimo metodologiją

Lentelių padalijimo metodologija yra dažnai naudojama tuomet, jei jose esančių duomenų kiekis visą laiką sparčiai auga ir užima kelis šimtus gigabaitų ar terabaitų. Šiuo atveju

naudingiausia yra padalyti faktų lenteles, nes jose saugoma daugiausia duomenų. Šios metodologijos realizacijai sukurtos papildomos lentelės, kuriose yra įkeliami duomenys pagal atitinkamus kriterijus. Šios lentelės naudojamos tam, kad būtų galima atlikti duomenų įkėlimo operaciją naudojant komandą „ALTER TABLE „faktu_lentele“ SWITCH PARTITION X TO tarpine_lentele“. Įkėlus visus duomenis į papildomas lenteles su ta pačia komanda atliekamas duomenų įkėlimas į faktų lentelės dalį. Šie duomenys į lenteles yra įkeliami lygiagrečiai, kaip ir į parengiamąsias lenteles (žr. 4.15 pav.).



4.15 pav. Duomenų įkėlimas į padalytą faktų lentelę

4.2.7. Duomenų įrašymas naudojant laikinas lenteles

Šis procesas yra identiškas duomenų įkėlimo į tarpinę duomenų bazę lygiagrečiai procesui, tačiau vienintelis skirtumas yra tai, kad lentelės yra saugomos „tempdb“ duomenų bazėje, ir kurios, pasibaigus sesijai, yra pašalinamos.

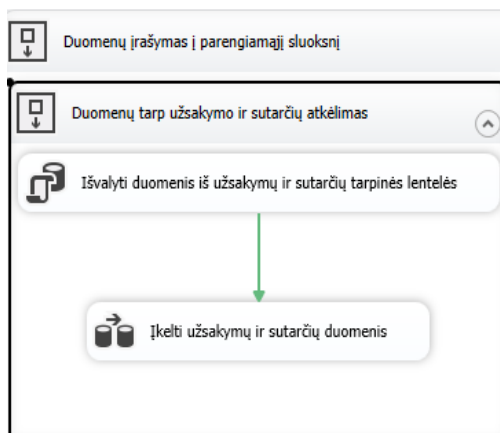
4.2.8. Duomenų įrašymas naudojant indeksuotus vaizdinius

Šiame procese yra naudojami indeksuoti vaizdiniai, kuriuose yra parašytos užklausos iš parengiamųjų lentelių. Naudojantis šiais vaizdiniais gaunami šių užklausų rezultatai. Visgi prieš įkeliant duomenis į parengiamąsias lenteles jose turi būti išvalomi duomenys. Indeksuoti vaizdiniai neleidžia naudoti komandos „TRUNCATE“, kuri išvalo visus lentelės duomenis. Tam, kad būtų galima panaudoti šią komandą, prieš tai galima pašalinti indeksuotame vaizdinyje sukurtą indeksą. Tuomet išvalius parengiamųjų lentelių duomenis galima perkurti indeksus šiuose vaizdiniuose.

4.2.9. Duomenų įrašymas naudojant atmintį taupančias lenteles

Šiame procese duomenys yra įkeliami į parengiamojo sluoksnio atmintį taupančias lenteles vietoje kietajame diske saugomų lentelių. Šiose lentelėse yra keletas apribojimų: išvalyti šių lentelių turinį galima naudojant tik „DELETE“ komandą, kadangi „TRUNCATE“ komanda

šiose lentelėse yra nepalaikoma, taip pat turi būti vienas pirminio rakto apribojimas konkrečiame lauke su neklasteriniu indeksu. Indeksus kurti leidžiama, tačiau jie turi būti iš anksto aprašyti, todėl indeksų šalinimo ir perkūrimo užduočių nėra (žr. 4.16 pav.).



4.16 pav. Duomenų įkėlimas į atmintį taupančias lenteles

4.2.10. Duomenų įrašymas naudojant atmintį taupančių lentelių tipo struktūras

Į atmintį orientuotos transakcijos gali būti naudojamos ir lentelių tipo struktūrose. Sintaksė nesiskiria nuo tos, kurioje kuriamos ir atmintį taupančios lentelės, tiktai nurodoma, kad lentelių tipo struktūra yra traktuojama kaip atmintį taupanti lentelių tipo struktūra.

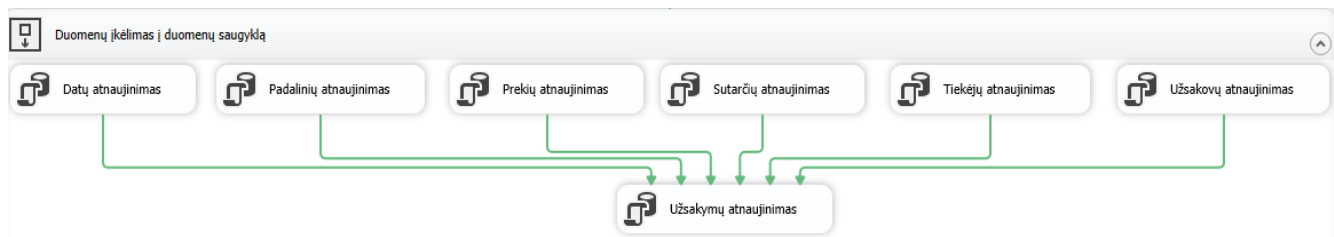
4.2.11. Duomenų įrašymas naudojant duomenų suspaudimo technologiją

Duomenų suspaudimas atliekamas parengiamojo sluoksnio lentelėse: tiek puslapių, tiek eilučių. ETL procesas yra identiškas lygiagretaus proceso vykdymui į kietajame diske esančias lenteles šalinant ir perkuriant indeksus, tačiau prieš įkeliant duomenis yra atliekama atitinkamo duomenų suspaudimo parinktis, tuomet ir indeksų šalinimas. Tada įkėlus duomenis į parengiamąsias lenteles vykdomas duomenų įkėlimas į duomenų saugyklą iš suspaustų parengiamųjų lentelių duomenų ir indeksų kūrimas. Taip pat yra palyginami skirtingi ETL procesai, kai papildomai suspaudžiami lentelėse esantys indeksai.

4.3. Duomenų įrašymo ir atnaujinimo proceso tyrimas

Šiame procese duomenys ne tik įkeliami, bet ir atnaujinami duomenų saugykloje. Verta paminėti, kad šioje sistemoje gali būti atnaujinamos tik dimensijų lentelės – faktų lentelėje esantys pakeitimai gali būti įkeliami, tačiau užsakymų sistemoje įkeliami tik įvykdyti užsakymai, todėl pakeitimų joje neatliekama.

Kaip jau minėta anksčiau, SSIS pakete procesas atliekamas naudojantis T-SQL operacija „MERGE“. Visi ETL paketai yra identiški, išskyrus tai, kad kviečiama SQL užduotis, kurioje kviečiama procedūra, tiek įrašanti, tiek atnaujinanti duomenis iš parengiamųjų lentelių (žr. 4.17 pav.).

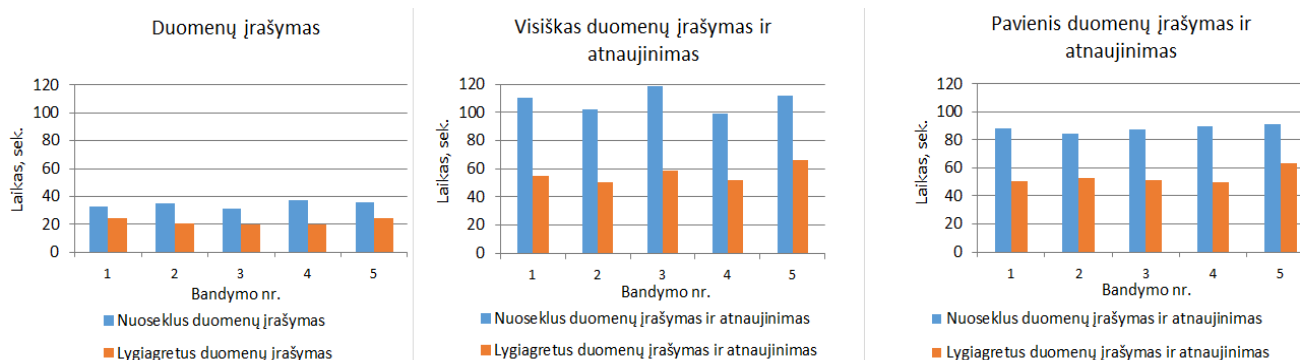


4.17 pav. Duomenų įkėlimo proceso vykdymas naudojant SQL užduotis

Jei atliekamas duomenų įrašymas ir atnaujinimas nenaudojant tarpinių lentelių, tuomet pasinaudojama lentelių tipo kintamaisiais, nes tik tokiu būdu įmanoma perduoti rezultatus norint vykdyti SQL užduotį.

4.4. ETL procesų tyrimų rezultatai

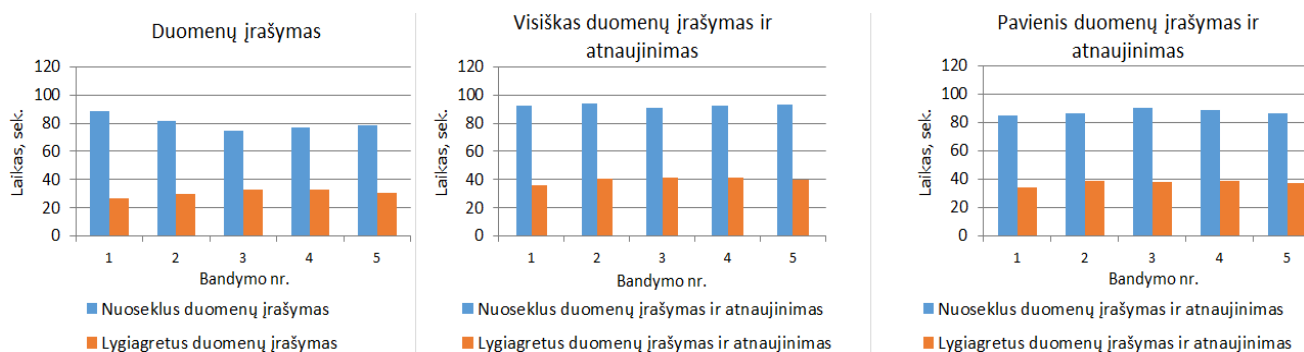
Galutiniai duomenų įrašymo bei duomenų įrašymo ir atnaujinimo rezultatai naudojant skirtingas technologijas yra pateikti grafikuose. Taip pat pavaizduoti rezultatai įkeliant duomenis į parengiamąjį sluoksnį naudojant visišką bei pavienį duomenų įkėlimo metodus, tiek įkeliant duomenis pavieniu būdu nenaudojant tarpinių lentelių.



4.18 pav. Nuoseklaus ir lygiagreto ETL procesų rezultatai nenaudojant tarpinių lentelių

4.18 pav. pateikti skirtingi duomenų išgavimo ir įrašymo į dimensijų bei faktų lenteles laikai naudojant skirtingus duomenų įkėlimo būdus, kai nenaudojamos tarpinės lentelės. Iš šių grafikų galima spręsti, kad duomenys į duomenų saugyklą yra įrašomi daug sparčiau, jei procesas vykdomas lygiagrečiai: pirmame grafike duomenys įkeliami vidutiniškai 33 % greičiau nei įkeliant duomenis nuosekliai, antrame grafike – 47,87 %, trečiame grafike naudojamas pavienis duomenų įkėlimo metodas, kurio metu duomenys buvo įkeliami 5 % greičiau nei naudojant visišką duomenų įkėlimo metodą. Geriausias duomenų įrašymo laikas – 19,671 sek., įkeliant tiek naujus, tiek atnaujintus duomenis taikant visišką duomenų įkėlimo metodą – 50,594 sek., taikant pavienį duomenų įkėlimo metodą – 50,078 sek.

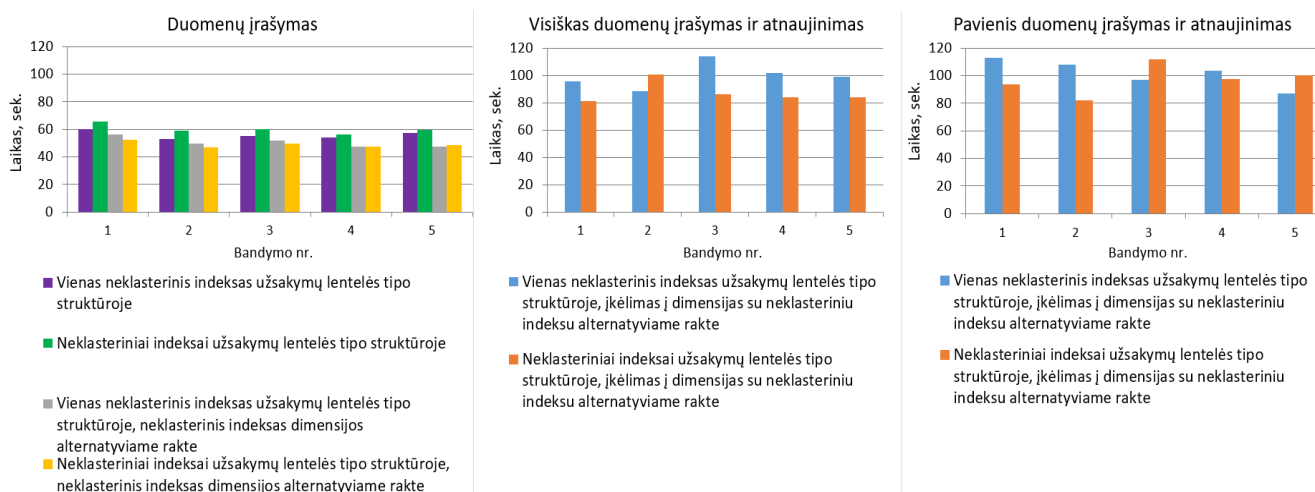
4.19 pav. pateiktas identiškas duomenų įkėlimas, tačiau naudojamos tarpinės lentelės.



4.19 pav. Nuoseklaus ir lygiagreto ETL procesų rezultatai naudojant tarpines lenteles

Lyginant su ankstesniais grafikais duomenų įrašymas nenaudojant tarpinių lentelių yra vykdomas sparčiau nei naudojant tarpines lenteles, kadangi atliekamas papildomas duomenų įkėlimas į šias lenteles. Visgi vykdant duomenų įrašymo ir atnaujinimo procesą duomenys sparčiau yra įkeliami naudojant tarpines lenteles: taikant visišką duomenų įkėlimo metodą, duomenys lygiagrečiai yra įkeliami vidutiniškai 40 % sparčiau negu įkeliant nuosekliai. Geriausias duomenų įrašymo laikas iš visų atliktų bandymų – 26,516 sek., įkeliant tiek naujus, tiek atnaujintus duomenis – 35,688 sek. Pavieniu duomenų įkėlimo būdu lyginant su visišku duomenų įkėlimo būdu duomenys įkeliami vidutiniškai 6% sparčiau, geriausias duomenų įkėlimo laikas – 34,578 sek.

Sekančiuose grafikuose pavaizduoti įvairiausi duomenų įkėlimo laikai nenaudojant tarpinių lentelių pritaikius atitinkamų rūšių indeksus ir technologijas.



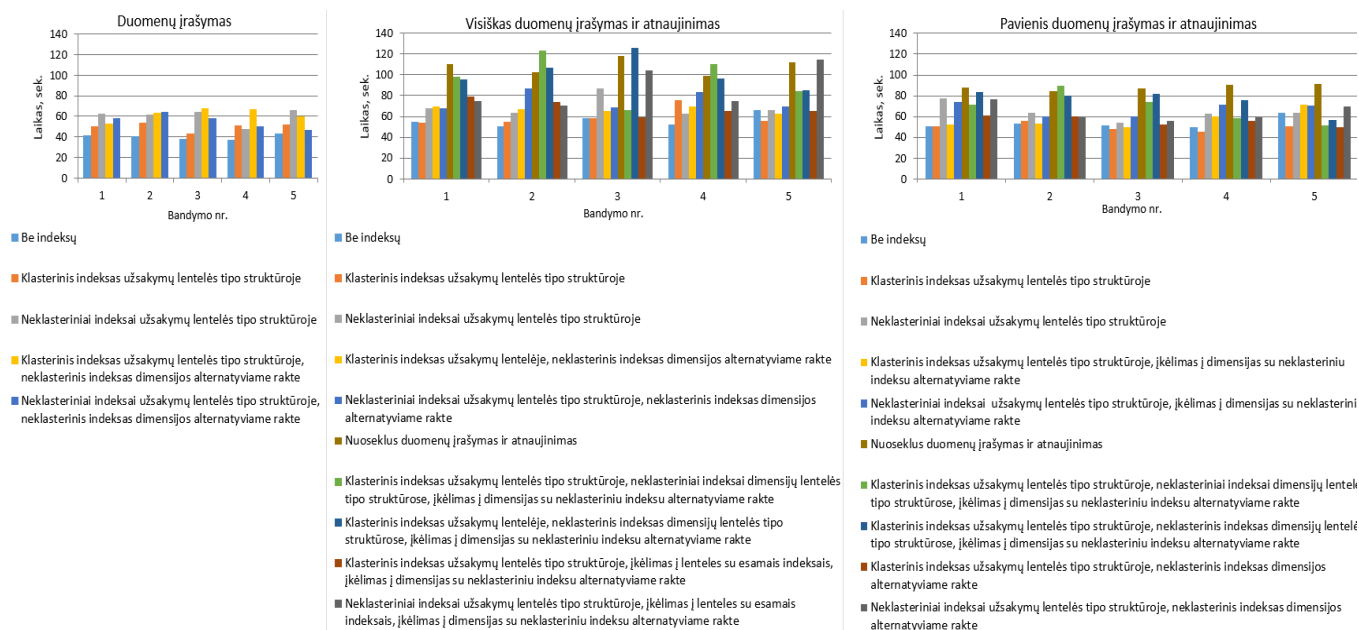
4.20 pav. ETL proceso rezultatai naudojant atmintį taupančių lentelių tipo struktūras

4.20 pav. pateikti ETL procesų rezultatai naudojant atmintį taupančių lentelių tipo struktūras su galimais atitinkamų rūšių indeksų panaudojimo galimybėmis. Duomenys yra įkeliami į dimensijų lenteles, kuriose yra sukurti neklasteriniai indeksai dimensijų lentelių alternatyviuose raktuose, kurie unikaliai identifikuoja įrašą, saugomą tarpinėse lentelėse, kai vykdomas duomenų įrašymo ir atnaujinimo procesas, kitokiu atveju, duomenų įkėlimas užtrunka daugiau nei 15 min.

Kiekvienose atmintį taupančių lentelių tipo struktūrose yra sukurti vienas arba keli neklasteriniai indeksai, kurie yra privalomi jose, atitinkamai laukuose, turinčiuose išorinio rakto apribojimus.

Iš gautų rezultatų galima pastebėti, kad įrašymo metu duomenys yra įkeliami sparčiausiai, jei sukuriama neklasteriniai indeksai užsakymų lentelės tipo struktūroje ant iš anksčiau minėto apribojimo turinčių laukų, ir egzistuoja neklasterinis indeksas kiekvienoje dimensijų lentelėje: vidutiniškai duomenys buvo įkeliami per 48,944 sek. Geriausias duomenų įkėlimo laikas – 47,235 sek. Duomenų įrašymo ir atnaujinimo metu duomenys taikant visišką duomenų įkėlimo metodą įkeliami sparčiausiai, jei kiekviename iš anksčiau minėtų laukų yra sukuriama neklasterinis indeksas. Vidutiniškai duomenys buvo įkeliami per 87,165 sek., kol naudojant vieną klasterinį indeksą duomenys buvo įkeliami per 99,472 sek. Geriausias užfiksuotas duomenų įkėlimo laikas naudojant vieną klasterinį indeksą yra 81,438 sek. Pavienio duomenų įkėlimo metu gauti rezultatai panašūs: duomenys taikant įvairiausias indeksų rūšis įkeliami atitinkamai per 101,738 sek. ir 97,153 sek. Geriausias užfiksuotas duomenų įkėlimo laikas taikant pavienį duomenų įkėlimo metodą – 82,016 sek.

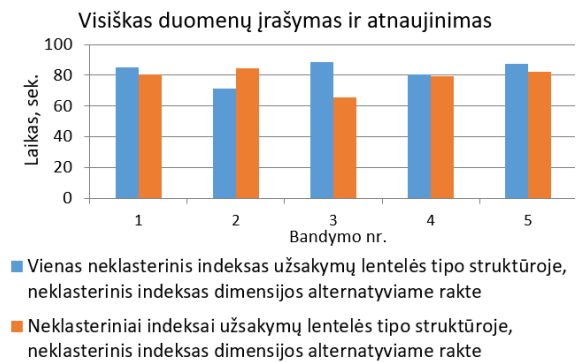
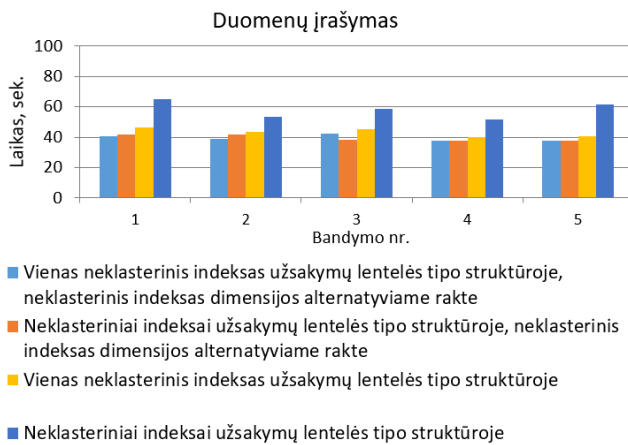
4.21 pav. pateikti duomenų įkėlimo naudojant lentelių tipo struktūras laikų grafikai. Kiekvieno duomenų įkėlimo metu taikomi skirtingi indeksų panaudojimo būdai. Duomenų įrašymo metu sukurti indeksai tik užsakymų lentelės tipo struktūrose, kadangi neatliekamas lentelių sujungimas tarpusavyje. Visa kita yra identiška, kaip ir su atmintį taupančių lentelių tipo struktūromis, duomenų įrašymo ir atnaujinimo metu testuojami įvairiausių rūšių indeksai.



4.21 pav. ETL proceso rezultatai taikant lentelių tipo struktūras nenaudojant tarpinių lentelių

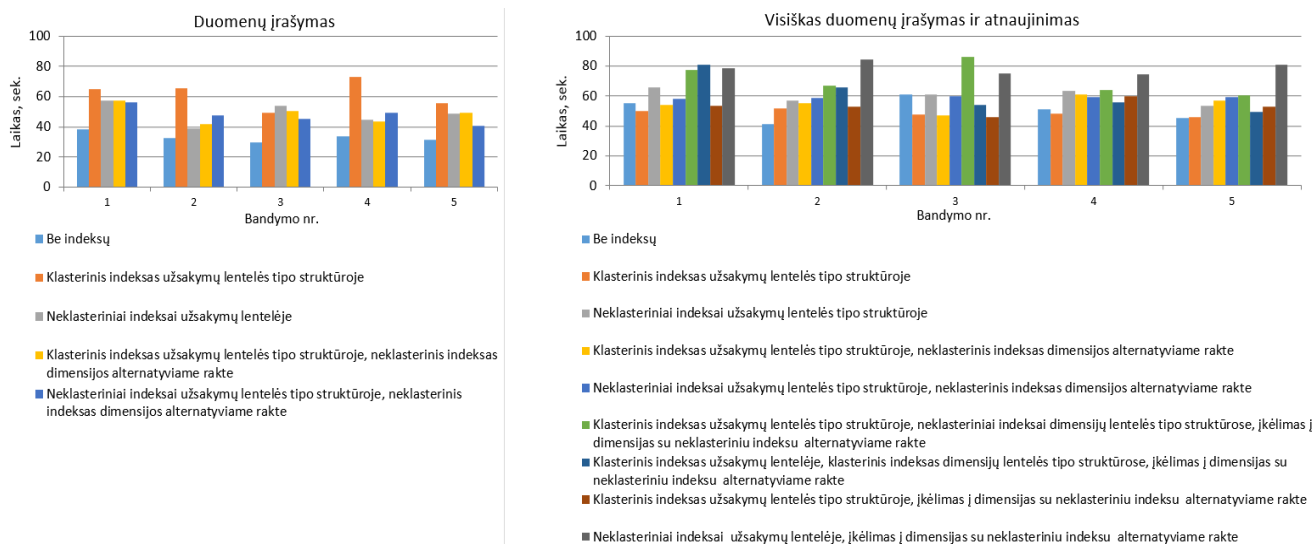
Iš pateiktų grafikų galima spręsti, kad duomenys sparčiausiai įrašomi, kai nėra apibrėžti indeksai lentelių tipo struktūrose: vidutiniškai duomenys buvo įkeliami per 40,244 sek., geriausias duomenų įkėlimo laikas – 37,125 sek. Lėčiausiai duomenys įrašomi, jei yra sukurtas klasterinis indeksas užsakymų lentelės tipo struktūroje ir neklasterinis indeksas kiekvienos dimensijų lentelės alternatyviame rakte: duomenų įkėlimo procesas truko vidutiniškai 62,338 sek. Duomenų įrašymo ir atnaujinimo procesas naudojant lentelių tipo struktūras vykdomas sparčiausiai, kai, kaip ir duomenų įrašymo metu, nenaudojami indeksai: duomenų įkėlimas truko vidutiniškai 56,469 sek. Taip pat yra palyginami duomenų įkėlimo laikai įrašant bei atnaujinant duomenis pavieniu būdu. Skirtumai tarp visiško duomenų įrašymo ir atnaujinimo naudojant skirtingus indeksų variantus yra panašūs, tačiau nežymiai sparčiau duomenys yra įkeliami, kai sukuriamas klasterinis indeksas užsakymų lentelės tipo struktūroje: duomenų įkėlimo procesas truko vidutiniškai 50,125 sek., o geriausias užfiksuotas duomenų įkėlimo laikas – 45,453 sek.

Likusiųse grafikuose yra pavaizduojami duomenų įkėlimo laikai taikant atitinkamų rūšių indeksus ir technologijas naudojant tarpines lenteles lygiai taip pat, kaip ir nenaudojant tarpinių lentelių.



4.22 pav. ETL proceso rezultatai naudojant atmintį taupančių lentelių tipų struktūras

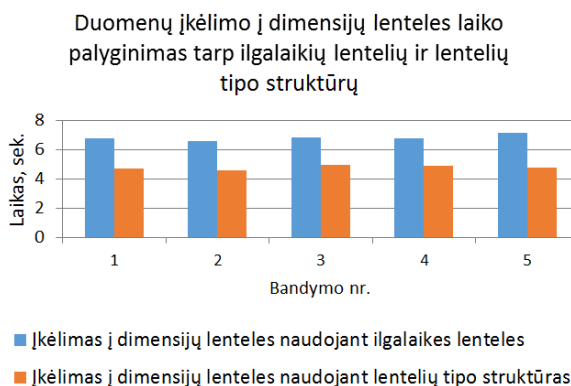
4.22 pav. pateikti duomenų įkėlimo laikai taikant atmintį taupančių lentelių tipo struktūras. Remiantis atliktais tyrimais nustatyta, kad įrašymo metu duomenys yra įkeliami sparčiausiai, jei sukuriamas vienas indeksas užsakymų lentelės tipo struktūroje ant išorinio rakto apribojimus turinčių laukų, ir egzistuoja neklasterinis indeksas kiekvienoje dimensijų lentelėje: vidutiniškai duomenys buvo įkeliami per 39,353 sek. Labai panašūs rezultatai gauti ir naudojant atskirus indeksus kiekvienam iš anksčiau minėtų laukų: vidutiniškai duomenys buvo įkeliami per 39,561 sek. Iš atliktų bandymų geriausi duomenų įkėlimo laikai irgi panašūs – duomenys atitinkamai buvo įkeliami per 37,406 sek. ir 37,750 sek. Duomenų įrašymo ir atnaujinimo metu duomenys įkeliami sparčiausiai, jei kiekviename iš anksčiau minėtų laukų yra sukuriamas neklasterinis indeksas. Vidutiniškai duomenys buvo įkeliami per 78,478 sek., kol naudojant vieną neklasterinį indeksą duomenys buvo įkeliami per 82,494 sek. Geriausias užfiksuotas duomenų įkėlimo laikas naudojant vieną neklasterinį indeksą yra 65,453 sek.



4.23 pav. ETL proceso rezultatai naudojant lentelių tipo struktūras

4.23 pav. pateikti duomenų įkėlimo laikai naudojant lentelių tipo struktūras. Tiek duomenų įrašymo, tiek duomenų įrašymo ir atnaujinimo procesai buvo vykdomi sparčiausiai, kai nebuvo naudojami indeksai: duomenys buvo įrašomi vidutiniškai per 33,255 sek., įrašomi bei atnaujinami – per 40,752 sek., geriausi užfiksuoti duomenų įrašymo bei duomenų įrašymo ir atnaujinimo laikai atliekant penkis bandymus atitinkamai yra 29,953 sek. bei 45,437 sek.

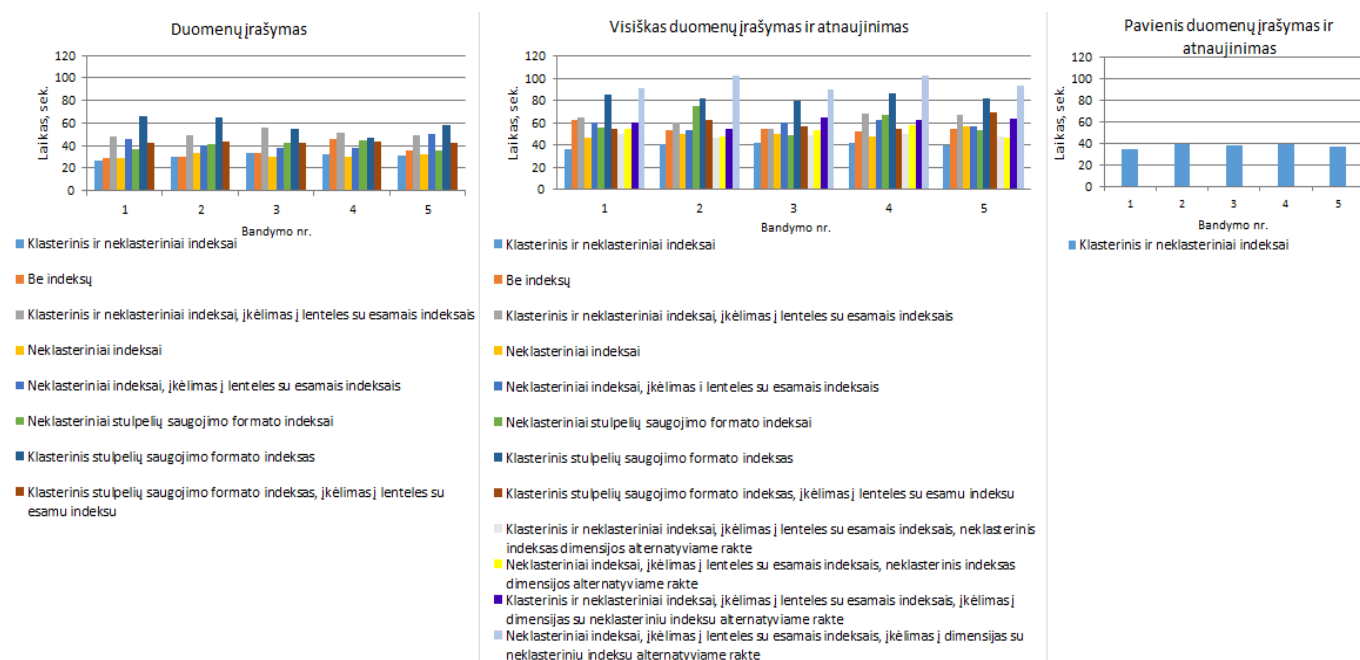
Atliekant eksperimentus pastebėta, kad naudojant lentelių tipo struktūras duomenys į dimensijų lenteles buvo įrašomi sparčiau nei naudojant ilgalaikes lenteles, tačiau tiek įrašant, tiek atnaujinant duomenis naudojant ilgalaikes lenteles buvo įkeliami sparčiau nei naudojant lentelių tipo struktūras. 4.24 pav. pateiktas grafikas, kuriame pavaizduoti skirtumai įrašant duomenis naudojant abiejų rūšių lenteles. Atlikus visus penkis bandymus nustatyta, kad duomenys yra įkeliami sparčiau negu naudojant ilgalaikes lenteles – vidutiniškai 40 % sparčiau. Geriausias duomenų įkėlimo laikas į dimensijų lenteles naudojant lentelių tipo struktūras – 4,578 sek., naudojant ilgalaikes lenteles – 6,578 sek.



4.24 pav. Duomenų įrašymo į dimensijų lenteles rezultatai

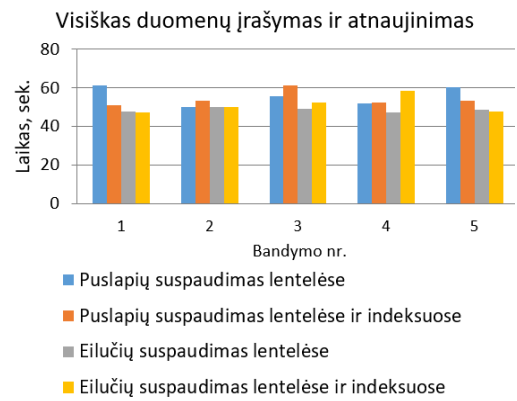
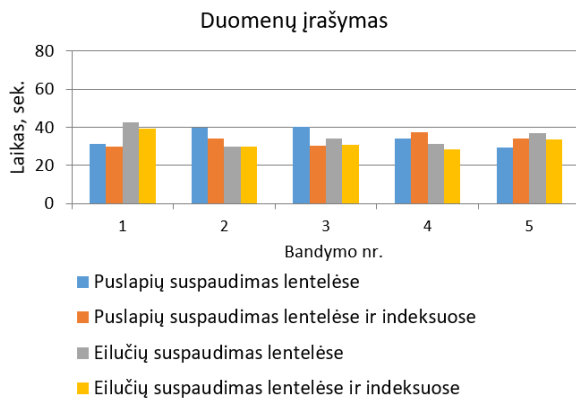
4.25 pav. pateikti duomenų įkėlimo laikai naudojant tarpines lenteles su skirtingais naudojamais indeksais. Tiek duomenų įrašymo, tiek duomenų įrašymo ir atnaujinimo procesai

buvo vykdomi sparčiausiai naudojant vieną klasterinį indeksą lauke, kuris unikaliai identifikuoja įrašą lentelėje, ir neklasterinius indeksai laukuose, kurie apibrėžia sąryšius tarp lentelių: pirmo proceso metu duomenys buvo įkeliami vidutiniškai per 30,516 sek., antro proceso metu – per 39,784 sek. Geriausias užfiksuotas duomenų įrašymo laikas – 30,891 sek., duomenų įrašymo ir atnaujinimo – 35,688 sek. Taip pat galima pastebėti, kaip duomenys buvo įkeliami naudojant pavienį duomenų įkėlimo metodą. Duomenys vidutiniškai buvo įkeliami per 41,475 sek., o tai yra lėčiau lyginant su visišku duomenų įkėlimu naudojant klasterinį ir neklasterinius indeksus lentelėse, tačiau iš visų atliktų bandymų naudojant šį metodą duomenys buvo įkeliami sparčiausiai – per 34,578 sek.



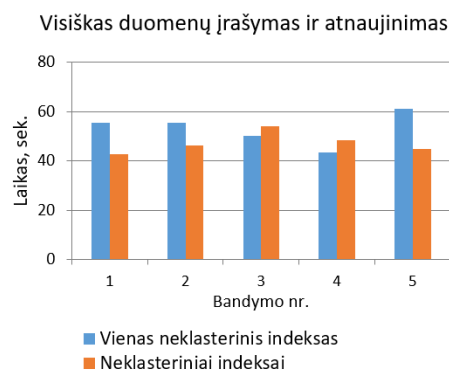
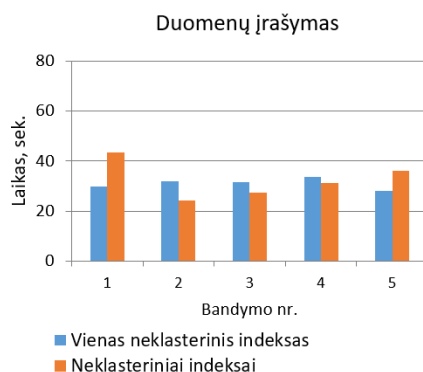
4.25 pav. ETL proceso rezultatai naudojant skirtingų indeksų variantus tarpinėse lentelėse

4.26 pav. pateikti duomenų įkėlimo laikai, kai tarpinėse lentelėse ir jose esančiuose indeksuose duomenys buvo suspaudžiami naudojant puslapių ir eilučių suspaudimą. Duomenų įrašymo metu efektyviausias metodas buvo eilučių suspaudimas lentelėse ir indeksuose, kur duomenys buvo įkeliami vidutiniškai per 28,453 sek. Skirtumai tarp kitų naudojamų metodų yra nežymūs. Duomenų įrašymo ir atnaujinimo metu efektyviausias metodas buvo eilučių suspaudimas lentelėse: duomenys vidutiniškai buvo įkeliami per 48,644 sek.



4.26 pav. ETL proceso rezultatai naudojant duomenų suspaudimo technologiją

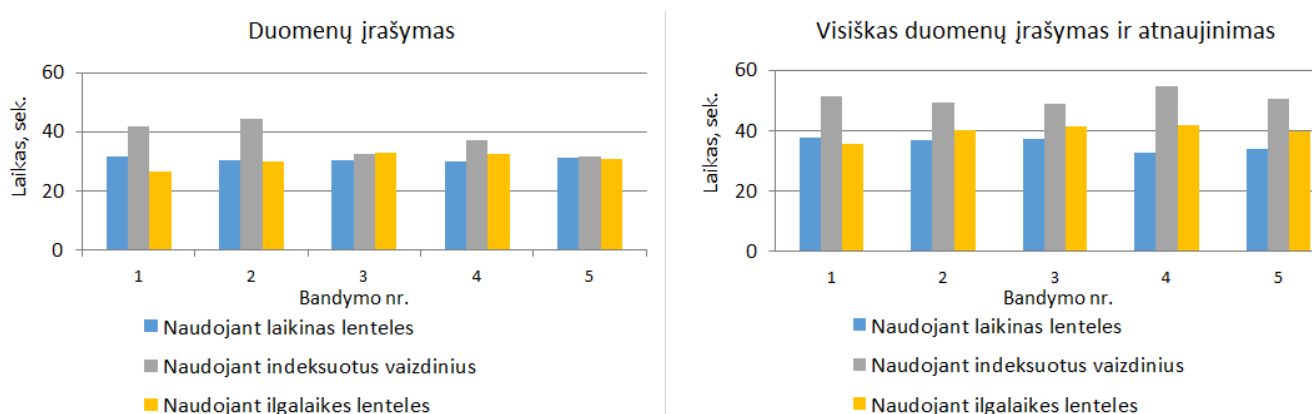
4.27 pav. pateikti grafikai, kurie apibrėžia duomenų įkėlimo laiką naudojant atmintį taupančias lenteles. Šiame procese naudojamos dvi indeksų kombinacijos: vienas neklasterinis indeksas tuose laukuose, kurie apibrėžia išorinio rakto apribojimus, bei neklasteriniai indeksai atskirai kiekviename iš šių laukų. Naudojant neklasterinius indeksus duomenų įrašymo bei duomenų įrašymo bei atnaujinimo procesai buvo vykdomi sparčiausiai: pirmas procesas vidutiniškai truko 31,037 sek., antras procesas – 47,188 sek. Geriausias užfiksuotas duomenų įrašymo laikas – 24,328 sek., duomenų įrašymo bei atnaujinimo – 42,766 sek. Tokius skirtumus tarp geriausio ir blogiausio rezultatų pateiktuose grafikuose sąlygoja operatyviosios atminties panaudojimas: naudojant didesnę kiekį operatyviosios atminties apkraunamas ir duomenų įkėlimo procesas į šias lenteles.



4.27 pav. ETL proceso rezultatai naudojant atmintį taupančias lenteles

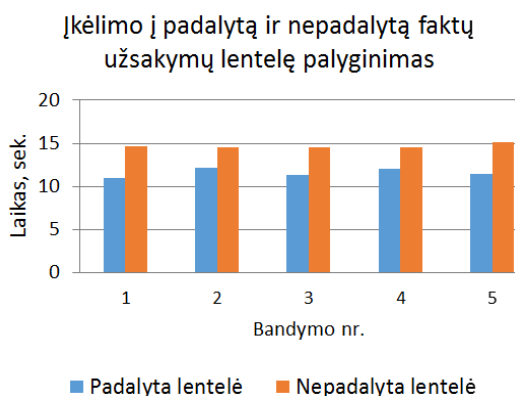
4.28 pav. pateikti grafikai apibrėžia skirtingus duomenų įkėlimo laikus naudojant ilgalaikes lenteles, laikinas lenteles bei indeksuotus vaizdinius. Tarpinėse lentelėse buvo naudojamas vienas klasterinis indeksas lauke, kuris unikaliai identifikuoja įrašą lentelėje, bei neklasteriniai indeksai išorinio rakto apribojimus turinčiuose laukuose. Pagal atliktus tyrimus duomenų įrašymo procesas buvo vykdomas sparčiausiai naudojant ilgalaikes lenteles: duomenys buvo įkeliami vidutiniškai per 30,716 sek., geriausias duomenų įkėlimo laikas – 26,516 sek. Duomenų įrašymo bei atnaujinimo procesas buvo vykdomas sparčiausiai naudojant laikinas

lenteles: duomenys buvo įkeliami vidutiniškai per 35,750 sek., geriausias duomenų įkėlimo laikas – 32,688 sek.



4.28 pav. ETL proceso rezultatai naudojant skirtingas lentių rūšis

4.29 pav. pateikti skirtingi duomenų įkėlimo į padalytą ir nepadalytą užsakymų faktų lentelę laikai. Iš grafiko galima spręsti, kad, jei užsakymų faktų lentelė yra padalyta, duomenys visais atvejais yra įkeliami sparčiau nei įkeliant į nepadalytą lentelę: duomenys yra įkeliami vidutiniškai 21 % sparčiau negu į nepadalytą lentelę.



4.29 pav. Duomenų įkėlimo rezultatai į faktų lentelę naudojant padalijimo metodologiją

Išvados ir siūlymai

1. Atlikus duomenų saugojimo duomenų saugykloje metodologiją pastebėta, kad saugoti duomenis duomenų saugykloje yra efektyviau nei OLTP duomenų bazėse, nes neatliekamos duomenų manipuliavimo operacijos duomenų saugyklose realiu laiku, tačiau sudėtingiausias procesas yra ETL procesas, kuris apima didžiąją dalį laiko išgaunant duomenis iš skirtingų duomenų šaltinių, juos pertvarkant ir įkeliant į duomenų saugyklą.
2. Išanalizavus ETL procesą nuspręsta naudoti grandininį duomenų pertvarkymo būdą, kur duomenys yra įkeliami į parengiamąsias lenteles tik vieną kartą, kol daugiapakopiu būdu įkeliami į kelias parengiamąsias lenteles kiekvieno pertvarkymo metu, nes tokiu būdu sunaudojama mažiau kompiuterio išteklių ir duomenų bazės objektų.
3. Atlikus ETL metodologijas, technologijas ir įrankius galima teigti, kad nėra vieningo metodo, leidžiančio panaudoti technologijas ETL procese – dažniausiai tenka atlikti eksperimentinius tyrimus, kuriais remiantis galima nuspręsti, kurias metodologijas ar technologijas naudoti ETL procese. Pasirinktas ETL įrankis „Microsoft SQL Server SSIS“ labiausiai tenkina duomenų integracijos proceso vykdymo reikalavimus, kuris geriausiai suderintas su „Microsoft SQL Server“ DBVS.
4. Atlikus išsamius ETL tyrimus naudojant skirtingas metodologijas bei technologijas, nustatyta, kad sparčiausiai duomenys buvo įrašomi nenaudojant tarpinių lentelių vykdant duomenų įrašymo procesą lygiagrečiai: duomenys įrašomi mažiausiai 33 % sparčiau nei taikant kitas technologijas. Naudojant tarpines lenteles nustatyta, kad duomenys sparčiausiai buvo įkeliami naudojant klasterinį indeksą unikaliame lauke, kuris identifikuoja įrašą, ir neklasterinius indeksus laukuose, kurie nustato išorinio rakto apribojimus tarp lentelių: duomenys buvo įkeliami mažiausiai 0,65 % sparčiau negu taikant kitas technologijas, – tačiau iš visų atliktų bandymų geriausias rezultatas buvo gautas naudojant atmintį taupančias lenteles. Duomenų įrašymo ir atnaujinimo procesas buvo vykdomas sparčiausiai, kai buvo naudojamos laikinosios lentelės, kur duomenys įkeliami mažiausiai 10 % sparčiau nei taikant kitas technologijas. Mažiausiai efektyvus ETL procesas buvo vykdomas naudojant atmintį taupančių lentelių tipo struktūras dėl apribojimų, taikomų šioms struktūroms. Duomenų įkėlimas į padalytą lentelę taip pat pasiteisino, todėl esant dideliame duomenų kiekiui galima naudoti šią metodologiją.

Pavieniam ETL procesui buvo naudojamas auditinis laukas, tačiau rekomenduotina naudoti auditines lenteles, kurios saugo ne tik įrašo keitimo laiką, tačiau ir visus pakitusius įrašus, kadangi tokiu būdu bus galima gauti ir tuos įrašus, kurie buvo įrašyti ir pašalinti tuo pačiu

metu. Vis dėlto pavieniam ETL proceso realizavimui, jei lentelių kiekis didėja, užklausų formavimas, kuriais išgaunami tik pakitę duomenys, sudėtingėja, ir jos vykdymas gali tapti lėtesnis nei įkeliant ar išgaunant visus esamus įrašus duomenų bazėse.

Nerekomenduotina naudoti atmintį taupančias lenteles ar šių lentelių tipus, kadangi dažnai buvo susiduriama su problemomis, jog užklausų vykdymui neužtekdavo operatyvinės atminties. Didesnį atminties kiekį turinčiam kompiuteriui galima naudoti šias lenteles, tačiau įkėlimas į jas užtrunka ilgiau nei į kietajame diske saugomas lenteles.

Taip pat pastebėta, kad duomenų įkėlimas naudojant lentelių tipo kintamuosius buvo vykdomas sparčiau į dimensių lenteles negu naudojant „SQL Server Destination“ duomenų įkėlimo parinktį, tačiau įkėlimas į faktų lentelę buvo lėtesnis negu naudojant „SQL Server Destination“ parinktį, nes papildomai buvo vykdomas išorinių raktų iš dimensių lentelių atrinkimas, todėl rekomenduojama naudoti šią technologiją, kai vykdomas duomenų įkėlimas į lenteles, kurioms nereikia gauti atitinkamų raktų iš kitų lentelių.

Nors sparčiausiai duomenys buvo įkeliami naudojant laikinąsias lenteles, tačiau SSIS paketas, jei atsijungiama nuo duomenų bazės, neranda šių lentelių, todėl rekomenduotina naudoti ilgalaikes lenteles su klasteriniu ir neklasteriniais indeksais, kurioms galima taikyti papildomas optimizavimo technikas, pavyzdžiui, sumažinti transakcijų fiksavimą ir pan., tuo pačiu palaikant paskutinę duomenų įkėlimo versiją.

Literatūros ir šaltinių sąrašas

1. BRALGIN, Igor. *Change Data Capture (CDC) Methods* [interaktyvus]. 2011 [žiūrėta 2017 m. kovo 5 d.]. Prieiga per internetą: <http://www.dwh-club.com/dwh-bi-articles/change-data-capture-methods.html>.
2. BRUIJN, de Jos; AGARWAL, Sunil; KOMO, Darmadi; BORDIA, Badal ir kt. *Microsoft SQL Server In-Memory OLTP and Columnstore Feature Comparison* [interaktyvus]. 2016, Microsoft Corporation [žiūrėta 2017 m. kovo 20 d.]. Prieiga per internetą: http://download.microsoft.com/download/D/0/0/D0075580-6D72-403D-8B4D-C3BD88D58CE4/SQL_Server_2016_In_Memory_OLTP_and_Columnstore_Comparison_White_Paper.pdf.
3. *Commercial ETL Tools: ETL Tools Comparison* [interaktyvus]. [žiūrėta 2016 m. spalio 20 d.]. Prieiga per internetą: <http://www.etltools.net/etl-tools-comparison.html>.
4. DARSHAN M. Tank., AMIT Ganatra., Y. P. Kosta., C. K. Bhensdadia. *Speeding ETL Processing in Data Warehouses Using High-Performance Joins for Changed Data Capture (CDC)* [interaktyvus]. 2010, p. 365-368. [žiūrėta 2017 m. kovo 5 d.]. ISBN 978-0-7695-4201-0. Prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5656810>.
5. *Data Warehousing Tutorial* [interaktyvus]. [žiūrėta 2016 m. sausio 17 d.]. Prieiga per internetą: <http://www.tutorialspoint.com/dwh/>.
6. *Database Advanced Replication. Materialized View Concepts and Architecture* [interaktyvus]. 2015 [žiūrėta 2017 m. kovo 10 d.]. Prieiga per internetą: <https://docs.oracle.com/database/121/REPLN/repview.htm#REPLN266>.
7. *Database Data Warehousing Guide. Loading and Transformation* [interaktyvus]. 2017 [žiūrėta 2017 m. sausio 19 d.]. Prieiga per internetą: http://docs.oracle.com/cd/B19306_01/server.102/b14223/transform.htm.
8. *DataOnFocus. OLTP vs OLAP* [interaktyvus]. 2015 [žiūrėta 2016 m. sausio 17 d.]. Prieiga per internetą: <http://www.dataonfocus.com/oltp-vs-olap/>.
9. *Datawarehouse4u.Info* [interaktyvus]. 2009 [žiūrėta 2016 m. sausio 17 d.]. Prieiga per internetą: <http://datawarehouse4u.info/OLTP-vs-OLAP.html>.
10. *Difference between b-tree and bitmap index tips* [interaktyvus]. 2015 [žiūrėta 2017 m. kovo 19 d.]. Prieiga per internetą: http://www.dba-oracle.com/t_difference_between_btree_and_bitmap_index.htm.

11. GALLO, Crescenzo; BONIS, de Michelangelo; PERILLI, Michele. *Data warehouse design and management: Theory and practice* [interaktyvus]. 2010 [žiūrėta 2016 m. sausio 3 d.]. Prieiga per internetą: <https://core.ac.uk/download/pdf/6342303.pdf>.
12. HAN, Jiawei.; KAMBER, Micheline.; PEI, Jian. *Data Mining. Concepts and Techniques*, 3rd ed. [interaktyvus]. Elsevier Inc., 2012. ISBN 978-0-12-381479-1 [žiūrėta 2017 m. kovo 13 d.]. Prieiga per internetą: <http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>.
13. *IBM InfoSphere DataStage* [interaktyvus]. [žiūrėta 2016 m. spalio 17 d.]. Prieiga per internetą: <https://www.ibm.com/us-en/marketplace/datastage>.
14. *Informatica* [interaktyvus]. 2017 [žiūrėta 2016 m. spalio 17 d.]. Prieiga per internetą: <https://www.informatica.com/products/data-integration/powercenter.html#fbid=ov14KmEQxeG>.
15. KATRAGADDA, Ranjith; TIRUMALA, Sremath Sreenivas; NANDIGMA, David. *ETL tools for Data Warehousing: An empirical study of Open Source Talend Studio versus Microsoft SSIS* [interaktyvus]. [žiūrėta 2017 m. sausio 25 d.]. Prieiga per internetą: <http://unitec.researchbank.ac.nz/bitstream/handle/10652/3366/ETL%20tools%20for%20Data%20Warehousing%20An%20empirical%20study.pdf?sequence=1>.
16. KHOSLA, Reema. *Oracle Warehouse Builder. Concepts 11g Release 2 (11.2)*, 6th ed. [interaktyvus]. 2014 [žiūrėta 2016 m. spalio 19 d.]. Prieiga per internetą: http://docs.oracle.com/cd/E11882_01/owb.112/e10581.pdf.
17. KIMBALL, Ralph; ROSS, Margy. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. [interaktyvus]. Indianapolis: John Wiley & Sons, 2013. ISBN 978-1-118-73219-9 [žiūrėta 2015 m. gruodžio 23 d.]. Prieiga per internetą: <http://www.essai.rnu.tn/Ebook/Informatique/The%20Data%20Warehouse%20Toolkit,%203rd%20Edition.pdf>.
18. *Microsoft. Columnstore indexes – data loading guidance* [interaktyvus]. 2017 [žiūrėta 2017 m. balandžio 2 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-data-loading-guidance>.
19. *Microsoft. Data Compression* [interaktyvus]. 2016 [žiūrėta 2017 m. balandžio 4 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/data-compression>.

20. *Microsoft. Designing Indexed Views* [interaktyvus]. 2017 [žiūrėta 2017 m. balandžio 7 d.]. Prieiga per internetą: [https://technet.microsoft.com/en-us/library/ms187864\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187864(v=sql.105).aspx).
21. *Microsoft. Indexes* [interaktyvus]. 2016 [žiūrėta 2017 m. balandžio 2 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/indexes>.
22. *Microsoft. In-Memory OLTP (In-Memory Optimization)* [interaktyvus]. 2016 [žiūrėta 2017 m. kovo 30 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp>.
23. *Microsoft. Page Compression Implementation* [interaktyvus]. 2016 [žiūrėta 2017 m. balandžio 4 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/page-compression-implementation>.
24. *Microsoft. Row Compression Implementation* [interaktyvus]. 2016 [žiūrėta 2017 m. balandžio 4 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/row-compression-implementation>.
25. *Microsoft. Special Table Types* [interaktyvus]. 2017 [žiūrėta 2017 m. balandžio 5 d.]. Prieiga per internetą: [https://technet.microsoft.com/en-us/library/ms186986\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms186986(v=sql.105).aspx).
26. *Microsoft. SQL Server Index Design Guide* [interaktyvus]. 2017 [žiūrėta 2017 m. balandžio 2 d.]. Prieiga per internetą: [https://technet.microsoft.com/en-us/library/jj835095\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/jj835095(v=sql.110).aspx).
27. *Microsoft. SQL Server Integration Services* [interaktyvus]. 2017 [žiūrėta 2017 m. kovo 5 d.]. Prieiga per internetą: <https://msdn.microsoft.com/en-us/library/ms141026.aspx>.
28. *Microsoft. The Data Loading Performance Guide* [interaktyvus]. 2009 [žiūrėta 2017 m. sausio 27 d.]. Prieiga per internetą: [https://technet.microsoft.com/en-us/library/dd425070\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/dd425070(v=sql.100).aspx).
29. *Microsoft. Use Table-Valued Parameters (Database Engine)* [interaktyvus]. 2017 [žiūrėta 2017 m. balandžio 5 d.]. Prieiga per internetą: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/use-table-valued-parameters-database-engine>.
30. MIQUEL, Laura Hofman; STANLEY, Joshua. *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator 12c (12.1.2)*, 3rd ed. [interaktyvus]. 2014, Oracle [žiūrėta 2016 m. spalio 19 d.]. Prieiga per internetą: <https://docs.oracle.com/middleware/1212/odi/ODIDG.pdf>.

31. *OLAP Online Analytical Processing* [interaktyvus]. [žiūrėta 2016 m. sausio 17 d.]. Prieiga per internetą: <http://athena.ecs.csus.edu/~olap/olap/OLAPoperations.php>.
32. SHITAO, Tu; LANJUAN, ZHU. *An optimized ETL fault-tolerant algorithm in data warehouses* [interaktyvus]. 2013, Information Science and Technology (ICIST) [žiūrėta 2017 m. vasario 25 d.]. ISBN 978-1-4673-2764-0 Prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6747594>.
33. SIMITSIS, Alkis; WILKINSON, Kevin; CASTELLANOS, Malu; DAYAL, Umeshwar. *QoX-Driven ETL Design: Reducing the Cost of ETL Consulting Engagements* [interaktyvus]. 2009, p. 953-960 [žiūrėta 2017 m. balandžio 1 d.]. ISBN 978-1-60558-551-2. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.367.3287&rep=rep1&type=pdf>.
34. *Talend Products Data Integration* [interaktyvus]. 2017 [žiūrėta 2017 m. spalio 19 d.]. Prieiga per internetą: <https://www.talend.com/products/data-integration>.
35. VILINSKI, Vladimir. *ETL tool comparison* [interaktyvus]. 2014 [žiūrėta 2017 m. spalio 20 d.]. Prieiga per internetą: <http://vilinski.de/etl-tool-comparison/>.

Priedai

A priedas. Pažyma dėl rezultatų panaudojimo VGTU

PATVIRTINIMAS

2017 m. gegužės 24 d.

Vilnius

Šiuo aktu patvirtinu, jog Vilniaus Gedimino technikos universiteto, Duomenų gavybos technologijų II kurso studentas Robertas Jankauskas pristatė savo darbą ITSC Programavimo skyriaus Informacinių sistemų grupei. Tyrimo rezultatai bus panaudoti VGTU siekiant įkelti duomenis iš „Oracle“ sistemos.

VGTU ITSC vadovas

Paulius Nomgaudas
VGTU ITSC vadovas

(Vardas, pavardė)



(Parašas)



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

PAŽYMĖJIMAS

Nr. 17-0412-...

Patvirtinantis, kad 2017 m. balandžio 12 d.

Robertas Jankauskas

dalyvavo VGTU 20-osios jaunųjų mokslininkų konferencijos
„**MOKSLAS – LIETUVOS ATEITIS**“ teminėje konferencijoje
INFORMATIKA. INFORMACINĖS TECHNOLOGIJOS,
parengė ir pristatė pranešimą
„**Duomenų užkrovimo greیتaveiklos tyrimas**“.

Mokslo komiteto pirmininkas

prof. dr. Genadijus Kulvietis

Organizacinio komiteto pirmininkas

doc. dr. Jelena Mamčenko

Vilnius, 2017