

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Virginija Uzdanavičiūtė

**Centralizuoto duomenų vientisumo apribojimų
valdymo sistema**

Magistro darbas

Darbo vadovė
prof. L. Nemuraitė

Kaunas, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Virginija Uzdanavičiūtė

**Centralizuoto duomenų vientisumo apribojimų
valdymo sistema**

Magistro darbas

Recenzentas

2007-01

doc. dr. S. Maciulevičius

Vadovė

prof. L. Nemuraitė
2007-01

Atliko

2007-01-08

IFM 1/4 gr. stud.
V. Uzdanavičiūtė

Kaunas, 2007

SUMMARY

System for centralist management of data integrity constraints

Integrity constraints comprise an urgent part of conceptual models. For this reason, they must be structured and represented in orderly manner. The purpose of this work is to solve the problems inherent in creation, storage and retrieval of centralized data integrity constraints.

This paper analyses the methods for implementation of data integrity constraints and describes a Oracle RDBMS-based technique that enables for impeccable functionality of information system and protection from invalid data.

After analysis of system requirements and currently available software solutions, UML models of system architecture, data and database were proposed. Suitable means were chosen for developing a system for centralist management of data integrity constraints.

The Constraint Editor geared towards designing, implementing and maintaining of procedures, integrity constraints throughout their lifecycle was developed. A simple, functional and intuitive graphical user interface makes data integrity constraints easy to story, manage and retrieve. The created software makes optimal use of RDBVS resources while performing ETL processes with multi – table insertions. The ability to insert data into multiple tables ensures further possibilities of supplying standard statistical analysis software with correct data and better performance than a procedure to perform multiple insertions using “IF..THEN” syntax.

Furthermore, the software verifies SQL DML functions system widethus protecting information system from invalid data.

TURINYS

LENTELIŲ SARAŠAS	6
PAVEIKSLŲ SARAŠAS	7
1. ĮVADAS	9
2. CENTRALIZUOTO VIENTISUMO APRIBOJIMŲ VALDYMO SISTEMOS ĮGYVENDINIMO GALIMYBIŲ ANALIZĖ	12
2.1. Vientisumo apribojimų klasifikacija ir savybės.....	12
2.2. Vientisumo apribojimų realizavimo būdų analizė	16
2.2.1. Vientisumo apribojimų realizavimo metodų analizė	16
2.2.2. Vientisumo apribojimus įgyvendinančių sistemų palyginimas	19
2.2.2.1. SQL įrankiai.....	19
2.2.2.2. Apribojimų DB sistemos	21
2.2.2.3. Taikomosios, verslo sistemos	27
2.2.2.4. Lyginamoji sistemų analizė	28
2.3. Veiklos tikslai	29
2.4. Vientisumo apribojimų įgyvendinimo sistemoje procesas	30
2.5. Vartotojų bei jų poreikių analizė	32
2.6. Vientisumo apribojimų valdymo sistemos veiklos sąveikų analizė	32
2.7. DBVS analizė, vientisumo apribojimų valdymo sistemos realizavimui	35
2.7.1. Oracle DBVS parinkimo pagrindimas	37
2.7.1.1. Oracle ETL procesas.....	38
2.7.1.2. Oracle9i DBVS objektų esminių bruožų analizė.....	39
2.8. Vientisumo apribojimų valdymo sistemos realizavimo architektūros parinkimas	42
2.8.1. Pasirinktas architektūros modelis	44
2.9. Vientisumo apribojimų valdymo sistemos realizavimo platformos, kalbos parinkimas.....	44
2.10. Projekto tikslai	47
2.11. Vientisumo apribojimų valdymo sistemos kokybės kriterijai	47
2.12. Sistemos įgyvendinimui galimų rizikos faktorių analizė.....	48
2.13. Projektavimo aplinka, metodai ir priemonės	49
2.14. Analizės išvados	50
3. CENTRALIZUOTO DUOMENŲ VIENTISUMO APRIBOJIMŲ VALDYMO KONCEPCINIS MODELIS	52
3.1. Vientisumo apribojimų valdymo sistemos reikalavimų modelis.....	52
3.1.1. Vientisumo apribojimų valdymo sistemai keliami funkciniai reikalavimai	52
3.1.2. Vientisumo apribojimų valdymo sistemos reikalavimų specifikacija	53
3.1.2.1. Vientisumo apribojimų redaktoriaus reikalavimų specifikacija	53
3.1.2.2. Apribojimų tikrinimo ir išgavimo serviso reikalavimų specifikacija	60
3.1.3. Apribojimų meta DB klasių modelis	61
3.1.4. Reikalavimai, skirti vientisumo apribojimų valdymo sistemos suderinamumui su įvairia sistemine PĮ.....	63
3.1.5. Vientisumo apribojimų valdymo sistemai keliami nefunkciniai reikalavimai	63
3.1.6. Reikalavimai vientisumo apribojimų valdymo sistemos duomenims	65
3.1.7. Reikalavimai programinei įrangai.....	65
3.1.8. Reikalavimai techninei įrangai	66
3.1.9. Reikalavimai IS apribojimų redaktoriaus vartotojo sąsajai	66
3.2. Centralizuoto vientisumo apribojimų valdymo sistemos projektas	67
3.2.1. Fizinė vientisumo apribojimų valdymo sistemos architektūra	67
3.2.2. Loginė vientisumo apribojimų valdymo sistemos architektūra	68
3.2.3. Detalaus apribojimo turinio tvarkymo projektas.....	72
3.2.4. Vientisumo apribojimų valdymo sistemos elgsenos modelis	73
3.2.5. Realizacijos modelis	75

3.2.5.1.	Programinių komponentų architektūra	75
3.2.5.2.	Vientisumo apribojimų valdymo sistemos diegimo modelis.....	75
3.2.5.3.	Duomenų bazės schemas	76
3.2.5.4.	Reikalavimai sistemos funkcionavimui	79
4.	EKSPERIMENTINIS VIENTISUMO APRIBOJIMŲ VALDYMO SISTEMOS TYRIMAS	80
4.1.	Centralizuoto vientisumo apribojimų valdymo sistemos aprašymas	80
4.1.1.	<i>Apribojimų redaktoriaus naudojimo aprašymas</i>	80
4.1.2.	<i>Apribojimų tikrinimo PĮ naudojimo aprašymas</i>	90
4.1.3.	<i>PĮ, įgyvendinančios apribojimų naudojimą Oracle ETL procese, aprašymas</i>	91
4.2.	Vientisumo apribojimų valdymo sistemos eksperimentinis tyrimas	92
4.2.1.	<i>Vientisumo apribojimų redaktoriaus kokybės tyrimas</i>	93
4.2.2.	<i>Vientisumo apribojimų tikrinimo serviso kokybės tyrimas</i>	99
4.2.3.	<i>Vientisumo apribojimų panaudojimo ETL procese tyrimas</i>	99
4.3.	Vientisumo apribojimų valdymo sistemos perspektyvos	108
5.	IŠVADOS IR REZULTATAI	110
	LITERATŪRA	111
	TERMINŲ IR SANTRUMPŲ ŽODYNAS	114
	PRIEDAI.....	116
1	PRIEDAS. ETL proceso įgyvendinimo Oracle9i DBVS veiklos diagramos	116
2	PRIEDAS. ETL proceso įgyvendinimo Oracle9i projekto diagramos	118
3	PRIEDAS. SQL tikrinimo projekto diagramos	121
4	PRIEDAS. Vientisumo apribojimų meta DB lentelių aprašymas	122
5	PRIEDAS. Struktūrų palyginimo skriptai	127
6	PRIEDAS. Straipsnis. Centralizuotos duomenų vientisumo apribojimų realizacijos metodika.....	129

LENTELIŲ SARAŠAS

1 lentelė. Vientisumo apribojimai esybių atributams	13
2 lentelė. Vientisumo apribojimai asociacijų ryšiams tarp esybių	14
3 lentelė. Vientisumo apribojimai specializacijos/apibendrinimo ryšio tipui	14
4 lentelė. GIS duomenų lentelė	22
5 lentelė. GIS duomenų aprašymas apribojimais	22
6 lentelė. CQA/CDB sistemos architektūros komponentų paskirties aprašymas	23
7 lentelė. Apribojimų sistemų analizė ir alternatyvos	25
8 lentelė. Apribojimų sistemų architektūros komponentų analizė ir alternatyvos	25
9 lentelė. Ontologijos aprašymas	27
10 lentelė. Apribojimo struktūros aprašymas	28
11 lentelė. Sistemų lyginamoji analizė	29
12 lentelė. Populiariausių DBVS lyginamoji analizė	36
13 lentelė. DB trigerių ir procedūrų palyginimas	39
14 lentelė. Procedūrų ir funkcijų palyginimas	40
15 lentelė. Procedūrų parametrų aprašymas	40
16 lentelė. Technologijų palyginimas	45
17 lentelė. .NET kalbų palyginimas	47
18 lentelė. Kuriamos sistemos kokybės parametrai	47
19 lentelė. Rizikos faktoriai	48
20 lentelė. Reikalavimo „Susieti apribojimus su prisijungimo taškais“ specifikacija	55
21 lentelė. Reikalavimo „Redaguoti apribojimo prisijungimą“ specifikacija	55
22 lentelė. Reikalavimo „Šalinti prisijungimo taško apribojimą“ specifikacija	56
23 lentelė. Reikalavimo „Tvarkyti apribojimo turinį“ specifikacija	57
24 lentelė. Reikalavimo „Tvarkyti apribojimų prisijungimo taškus“ specifikacija	59
25 lentelė. Veikimo reikalavimai	63
26 lentelė. Mobilumo reikalavimai	64
27 lentelė. Modifikavimo reikalavimai	64
28 lentelė. Kiti nefunkciniai reikalavimai	64
29 lentelė. Reikalavimai programinei įrangai	65
30 lentelė. Reikalavimai techninei įrangai	66
31 lentelė. Reikalavimai sistemos funkcionalumui	79
35 lentelė. Apribojimo sukūrimo eksperimentinio tyrimo apimtis	105
36 lentelė. SQL funkcijų eksperimentinio tyrimo apimtis	106
37 lentelė. ETL įgyvendinimo eksperimentinio tyrimo apimtis	107
38 lentelė. Nefunkcinio testavimo technikos	108
39 lentelė. Vartotojo lentelės struktūros aprašymas	122
40 lentelė. Vartotojo teisių lentelės struktūros aprašymas	122
41 lentelė. Teisių lentelės struktūros aprašymas	122
42 lentelė. DB lentelės struktūros aprašymas	123
43 lentelė. IS DB lentelės struktūros aprašymas	123
44 lentelė. Lauko lentelės struktūros aprašymas	123
45 lentelė. Apribojimo elemento lentelės struktūros aprašymas	123
46 lentelė. Procedūros lentelės struktūros aprašymas	124
47 lentelė. Operacijos lentelės struktūros aprašymas	124
48 lentelė. Operacijų klasifikatoriaus saugojimo lentelės struktūros aprašymas	124
49 lentelė. Parametrų lentelės struktūros aprašymas	124
50 lentelė. Pilno apribojimo lentelės struktūros aprašymas	125
51 lentelė. Reikšmės lentelės struktūros aprašymas	125
52 lentelė. Reikšmių intervalo saugojimo lentelės struktūros aprašymas	125
53 lentelė. Apribojimo ryšio saugojimo lentelės struktūros aprašymas	125

PAVEIKSLŲ SĄRAŠAS

1 pav. Vientisumo apribojimų klasifikacijos diagrama	13
2 pav. Trigerių metamodelis	18
3 pav. Vientisumo apribojimai	18
4 pav. SQL * PLUS redaktorius pagrindinis langas	20
5 pav. MySQL Query Browser redaktorius pagrindinis langas	20
6 pav. Lentelės DDL	21
7 pav. CQA/CDB komponentai	22
8 pav. RuleML architektūros modelis	24
9 pav. Apribojimų kalba aprašytas DTD	26
10 pav. SQL aprašas naudojant BNF notaciją	26
11 pav. MIDAS programos langas	27
12 pav. Veiklos tikslų modelis	30
13 pav. Pakeitimų valdymo proceso modelis	31
14 pav. Norimas įgyvendinti proceso modelis	31
15 pav. Aukščiausio lygio apribojimų valdymo kontekstinė diagrama	33
16 pav. Detalus veiklos panaudojimo atvejų modelis	34
17 pav. Detalizuotas veiklos panaudojimo atvejų modelis	35
18 pav. Procedūros realizavimo veiklos diagrama	41
19 pav. ODP.NET objektų modelis	46
20 pav. Apribojimų valdymo sistemos posistemiai	52
21 pav. Apribojimų pateikimo specialisto	54
funkcijų panaudojimo atvejų diagrama	54
22 pav. Apribojimų pateikimo specialisto papildomų	54
funkcijų panaudojimo atvejų diagrama	54
23 pav. Apribojimų redaktoriaus funkcijų panaudos atvejų diagrama	54
24 pav. Apribojimų susiejimo panaudojimo atvejų diagrama	55
25 pav. „Susieti apribojimą su prisijungimo taškais“ sekų diagrama	57
26 pav. Apribojimų turinio tvarkymo panaudojimo atvejų diagrama	57
27 pav. „Tvarkyti apribojimo turinį“ sekų diagrama	58
28 pav. „Tikrinti taisyklės teisingumą“ sekų diagrama	59
29 pav. Apribojimų tvarkymo panaudojimo atvejų diagrama	59
30 pav. „Prisijungimo taško tvarkymo“ sekų diagrama	60
31 pav. Kliento atliekamų funkcijų panaudojimo atvejų diagrama	60
32 pav. Apribojimų tikrinimo panaudojimo atvejų diagrama	61
33 pav. „Tikrinti SQL“ sekų diagrama	61
34 pav. Meta-DB klasių modelis	62
35 pav. Kliento/serverio modelis	67
36 pav. Apribojimų posistemio loginė architektūra	68
37 pav. Pagrindiniam posistemiiui priskirtos klasės	68
38 pav. Apribojimų valdymo sistemos loginė architektūra	68
39 pav. Teisių tikrinimo veiklos logikos klasių diagrama	69
40 pav. Apribojimo tvarkymo realizacijos klasės	70
41 pav. Prisijungimo realizacijos klasės	70
42 pav. Apribojimų tvarkymo projekto modelio trijų lygių klasių diagrama	71
43 pav. Apribojimo tvarkymo interfeiso navigavimo planas	71
44 pav. Detali apribojimų tvarkymo vartotojų paslaugų klasių diagrama	72
45 pav. DB prieigos modulis	72
46 pav. Apribojimo turinio tvarkymo detalaus projekto modelis	73
47 pav. Prisijungimo sekų diagrama	73
48 pav. Abstrakti apribojimų tvarkymo sekų diagrama	74

49 pav. Apribojimų sukūrimo pasirinkimo iš pagrindinio lango meniu, sekų diagrama	74
50 pav. Operacijų klasifikatoriaus parinkimo sekų diagrama	74
51 pav. Reikšmių intervalo įvedimo sekų diagrama	75
52 pav. Programinių komponentų architektūra.....	75
53 pav. Sistemos diegimo modelis	76
54 pav. Meta DB schemas diagrama	77
55 pav. Gamybinės DB schemas diagrama	78
56 pav. Pagrindinis redaktoriaus langas.....	81
57 pav. Vientisumo apribojimų redaktoriaus parametrų langas.....	81
58 pav. Prijungtų IS parametrų tvarkymo langas.....	82
59 pav. Oracle kliento parametrų peržiūros langas	83
60 pav. Vartotojų administravimo langas.....	83
61 pav. Vartotojų teisių kūrimo ir administravimo langas.....	84
62 pav. Apribojimų DB saugojamų lentelių ir jos laukų peržiūros, administravimo langas ...	84
63 pav. DB lentelių ir jų laukų pasirinkimo langas	85
64 pav. Apribojimo turinio tvarkymo langas	85
65 pav. Operacijų tvarkymo langas	86
66 pav. Reikšmės nuskaitymo langas	87
67 pav. Kelių reikšmių pasirinkimo langas	87
68 pav. Intervalo pasirinkimo, intervalo perdavimo į procedūrą langas	87
69 pav. Reikšmės pasirinkimo, perdavimo į procedūrą programos langas.....	88
70 pav. Pilno apribojimo peržiūros langas	88
71 pav. SQL kodo peržiūros langas	88
72 pav. IS redaktoriaus aprašymo langas	89
73 pav. Pagalbos langas	89
74 pav. Detalios vartotojo pagalbos paieškos langas	89
75 pav. SQL tikrinimo PĮ langas	90
76 pav. Įvedimo į daugelį lentelių PĮ pagrindinis langas.....	91
77 pav. IS parametrų suvedimo langas.....	93
79 pav. Perkeltų lentelių ir jų laukų peržiūros ir tvarkymo langas	94
80 pav. Prisijungimo prie SQL Serverio langas	95
81 pav. SQL Server schemas naršyklės langas.....	95
82 pav. Apribojimo sukūrimo langas.....	96
83 pav. SQL DML funkcijų tikrinimo langas.....	99
84 pav. SQL DML funkcijų tikrinimo langas.....	100
85 pav. ETL eksperimento vykdymo pagrindinis langas	101
86 pav. Laiko priklausomybės nuo duomenų kiekio ir apdorojimo struktūrų grafikas	102
87 pav. Laiko priklausomybės nuo duomenų kiekio ir apdorojimo grafikas.....	102
88 pav. Sudėtingesnių apribojimų struktūrų vykdymo laiko grafikas.....	103
89 pav. Sudėtingesnių apribojimų, struktūrų vykdymo laiko grafikas.....	103
90 pav. „IF-THEN“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas	104
91 pav. „INSERT FIRST“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas..	104
92 pav. „INSERT ALL“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas	105
93 pav. Besąlyginio įvedimo į kelias lenteles veiklos diagrama.....	116
94 pav. Sąlyginio įvedimo į kelias lenteles veiklos diagrama.....	116
95 pav. Sąlyginio pirminio įvedimo į kelias lenteles veiklos diagrama.....	117
96 pav. Sąlyginio pirminio įvedimo į kelias lenteles projekto sekų diagrama	118
97 pav. Sąlyginio įvedimo į kelias lenteles projekto sekų diagrama	119
98 pav. Besąlyginio įvedimo į kelias lenteles projekto sekų diagrama	120
99 pav. SQL tikrinimo veiklos diagrama	121
100 pav. SQL tikrinimo projekto sekų diagrama	121

1. ĮVADAS

Konceptualaus modelio, duomenų bazės projektavimas ir realizavimas – svarbus IS kūrimo etapas [9]. RDBVS teorijoje yra keletas reikalavimų, kuriuos turi atitikti DB duomenys. Vienas reikalavimų tipas – užtikrinti duomenų vientisumo sąlygas, kurioms esant, DB tam tikru laipsniu būtų teisinga, išsami ir pilna.

Šiuo metu trūksta patogių ir funkcionalių įrankių, leidžiančių greitai įgyvendinti duomenų vientisumo apribojimus, atlikti jų modifikavimą ar šalinimą nepažeidžiant duomenų ir jų tarpusavio ryšių, nes didelėse, sudėtingose dalykinėse srityse, turinčiose daug duomenų ir apribojimų, iškyla eilė problemų, susijusių su jų valdymu. Žinoma [9], kad pertekliniai apribojimai daro neigiamą įtaką veikimo greičiui, užima papildomą vietą kompiuterio atmintyje, o priešaringi ar konfliktiniai apribojimai iššaukia klaidas ir begalinius ciklus [1], [2], įvairias anomalijas. Norint, kad sistema užtikrintų vientisumą arba joje nebūtų klaidų, dažniausiai naudojamos tokios priemonės:

- tikrinami duomenys kas dieną, savaitę, mėnesį;
- išrenkami duomenys įvairiausiai pjūviais ir tikrinama, ar sueina galai, pavyzdžiui, ar suma pagal X sutampa su suma pagal Y.

Dėl to išauga sistemos palaikymo išlaidos, be reikalo eikvojamas darbo laikas. Duomenys vistiek būna nepatikimi, nes lieka rizika, kad reikiamu momentu jie nebus teisingi. Norint išvengti su vientisumo apribojimais susijusių problemų, reikia turėti tinkamą jų valdymo metodiką ir įrankį. Todėl šio tyrimo sritis – sistemų analizė ir projektavimas, vientisumo apribojimų įgyvendinimo DBVS būdai, procedūrų galimybės ir jų tobulinimas. Tyrimo objektas – vientisumo apribojimų įgyvendinimo ir keitimo sistema, apribojimų panaudojimo ETL procese galimybės. Darbe nagrinėjamos DBVS valdymo automatizavimo alternatyvos, vientisumo apribojimų įgyvendinimo procedūromis galimybės.

Analizei ir projektavimui pasirinktas objektinės analizės metodas ir unifikuota UML kalba. UML diagramos [3], [4] – savotiški objektinių sistemų žemėlapiai. Jie padeda greičiau ir paprasčiau perteikti informaciją apie sistemos struktūrą ir dinamines savybes.

Analizės metu išskirti dažniausiai su IS apribojimų valdymu susiję klausimai:

- kaip dažnai sistemoje atliekami pakeitimai susiję su apribojimais;
- kas lemia pasikeitimų atsiradimą – dažniausiai pasikeitimus ir keitimo dažnumą IS lemia kintantys reikalavimai, vartotojų poreikiai, verslo aplinka, įstatymai.
- kokia jų įtaka klaidų atsiradimui – dažnai apribojimų pakeitimas iššaukia naujas klaidas, ar anksčiau nenagrinėtas situacijas, problemas.

- ar lengvai valdomas šis procesas;
- ar pakanka priemonių, kokios jos.

Šie klausimai detalai nagrinėjami darbe. Minėtų problemų sprendimų radimui naudoti įvairių šalių informaciniai šaltiniai. Egzistuojančių programinės įrangos priemonių analizė parodė, kad nėra vieningo apribojimų realizavimo metodo, tačiau kai kurios esamos sistemos dalinai įgyvendina keliamus tikslus: greitai ir efektyviai realizuoja apribojimus ir jų pakeitimus. Darbe pasiūlytas inovatyvus sprendimas – atskirti apribojimus nuo IS ir struktūrizuoti juos saugoti atskiroje DBVS. Jam įgyvendinti suformuluoti tokie uždaviniai:

- apribojimų centralizuotam saugojimui ir išgavimui *sukurti vientisumo apribojimų meta DB*;
- apribojimų kūrimui ir modifikavimui – *vientisumo apribojimų redaktorių*;
- duomenų neprieštarinamumui ir teisingumui pasiekti – *tikrinimo servisą*.

Darbo naujumas yra tame, kad centralizuoto vientisumo apribojimų valdymo sistema įgyvendina duomenų išgavimo, transformavimo ir įkėlimo (ETL – angl. *Extract, Transform, Load*) procesą, kurio metu galima išgauti duomenis iš įvairių DBVS (MySQL, MS SQL Server 2005, Oracle) ir patikrinti jų vientisumą. Apribojimai gaunami iš centralizuotos meta duomenų bazės. Duomenų išgavimui iš įvairių DBVS naudojamos SQL transformacijos.

Apribojimų panaudojimas ETL procese leidžia sumažinti kreipinių į DBVS skaičių (nenaudojama loginė “IF-THEN” struktūra), atliekant duomenų analizę ar ruošiant saugoti juos paskirstytose, nutolusiose DBVS. Taip pat leidžia optimizuoti IS specialistų darbą, nes centralizacija suteikia galimybes vieningai peržiūrėti, saugoti ar vykdyti norimus apribojimus.

Darbas sudarytas iš 5 skyrių.

Įvade nagrinėjamas darbo aktualumas, sprendžiama problema.

Antrajame skyriuje nagrinėjami vientisumo apribojimai, jų klasifikacija, svarba IS kūrime, veiklos kontekstas, procesai ir esami sprendimai: apribojimus užtikrinantys metodai, egzistuojančios sistemos. Didelis dėmesys skirtas kūrimo aplinkos, priemonių, galimos architektūros, platformos parinkimo, programavimo kalbų analizei. Apibrėžti kuriamos sistemos tikslai, kokybės kriterijai ir identifikuota galima rizika.

Trečiajame skyriuje išnagrinėti reikalavimai programinei įrangai, parinkta redaktoriaus ir tikrinimo serviso architektūra, pateikiamos detalios apribojimų valdymo projekto diagramos.

Ketvirtajame skyriuje aprašoma realizuota apribojimų valdymo sistema, pateikiamas redaktoriaus, tikrinimo serviso, ETL proceso programinės įrangos eksperimentinio tyrimo aprašymas.

Darbo pabaigoje pateikiamos išvados ir tolesnės vientisumo apribojimų valdymo sistemos plėtimo, tobulinimo galimybės.

2. CENTRALIZUOTO VIENTISUMO APRIBOJIMŲ VALDYMO SISTEMOS ĮGYVENDINIMO GALIMYBIŲ ANALIZĖ

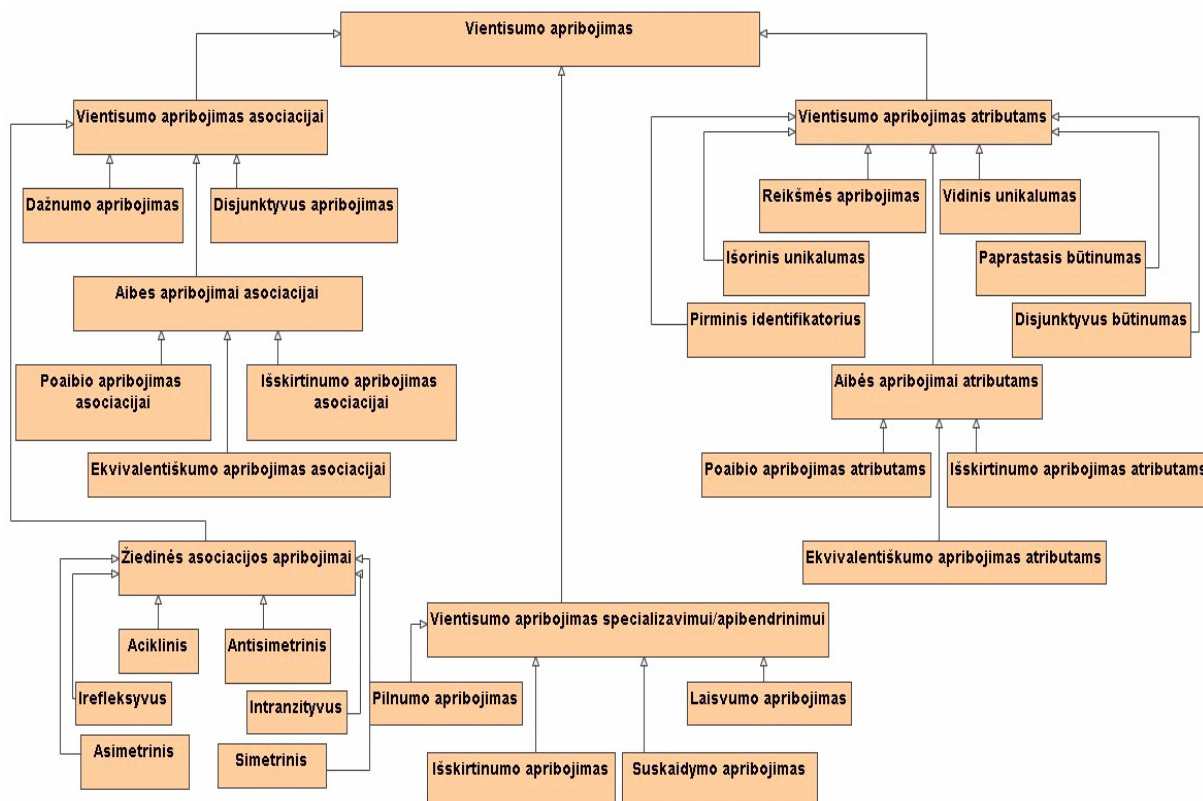
Skyriuje detaliai nagrinėjami vientisumo apribojimai, jų klasifikacija, svarba IS kūrime, esami įgyvendinimo būdai: metodai, sistemos ir jų sukūrimo technologijos, priemonės. Didelis dėmesys skiriamas metodų ir sprendimų pagal užsibrėžtus kriterijus lyginamajai, veiklos sąveikų analizei, naudojant UML veiklos panaudojimo atvejų diagramas. Analizuojami vartotojų poreikiai, formuojami veiklos ir sistemos tikslai. Sprendžiamos DBVS, sistemos architektūros pasirinkimo galimybės ir alternatyvos, privalumai ir trūkumai naudojant Oracle DBVS su J2EE, .NET programavimo technologijomis.

2.1. Vientisumo apribojimų klasifikacija ir savybės

Žinoma [1], kad vientisumo apribojimai – neatskiriama IS konceptualiojo modelio dalis, apimanti dalį probleminės srities semantikos. Todėl šiame skyriuje pateikiamas klasifikacijos modelis, pagrindinės savybės ir aprašymas.

Vientisumo apribojimai klasifikuojami (1 pav.) pagal tai, kokiems konceptualiojo modelio elementams priskirti:

- esybių atributams;
- asociacijų ryšiams tarp esybių;
- specializacijos/apibendrinimo ryšio tipui.



1 pav. Vientisumo apribojimų klasifikacijos diagrama

Lentelėse 1-3 pateikiami apribojimai ir jų paskirtys pagal klasifikaciją.

1 lentelė. Vientisumo apribojimai esybių atributams

Eil. Nr.	Pavadinimas	Paskirtis
1.	Pirminio identifikatoriaus apribojimas	Naudojamas unikaliai identifikuoti esybės tipo objektui iš tam tikro esybės tipo objektų grupės. Jis reikalauja, kad identifikuojantis atributas arba atributų grupė visada turėtų reikšmes ir kad šios reikšmės būtų unikalios.
2.	Paprastas būtinumo apribojimas	Naudojamas atributo arba esybės tipo objekto atliekamos rolės būtinumui pažymėti. Atributo būtinumas rodo, kad visose leistinose informacinės sistemos būsenose esybės tipo objekto atributas būtinai turi turėti reikšmę.
3.	Disjunktyvaus būtinumo apribojimas	Šis apribojimas rodo, kad esybės atributų būtinumas yra paskirstomas keliems atributams, t.y. visose leistinose IS būsenose bent vienas iš esybės atributų, kuriems nurodytas disjunktyvaus būtinumo apribojimas, būtinai turi turėti reikšmę.
4.	Unikalumo apribojimas	Unikalumo apribojimas rodo, kad esybės egzempliorių aibėje bet kurių dviejų egzempliorių atributų reikšmės yra skirtingos. Šis apribojimas naudojamas atributui arba jų

Eil. Nr.	Pavadinimas	Paskirtis
		grupei.
5.	Išorinio unikalumo apribojimas	Šis apribojimas neformaliai gali būti apibrėžiamas panašiai kaip unikalumo apribojimas skirtingų esybių atributams. Konceptualiai sujungtų atributų reikšmės turi būti skirtingos visoje konceptualiai sujungtų atributų reikšmių aibėje.
6.	Atributo reikšmės apribojimas	Apribojimas naudojamas apriboti atributo įgyjamų reikšmių aibę išvardinant visas galimas įgyti reikšmes, nurodant įgyjamų reikšmių intervalą specifikuojant intervalo pradžią ir pabaigą.

2 lentelė. Vientisumo apribojimai asociacijų ryšiams tarp esybių

Eil. Nr.	Pavadinimas	Paskirtis
1.	Ekvivalentiškumo apribojimas	Ekvivalentiškumo apribojimas tarp esybės ryšių su kitomis esybėmis nurodo, kad jeigu esybės egzempliorius dalyvauja viename ryšyje, tai jis būtinai turi dalyvauti ir kitame. Esybės egzemplioriui dalyvauja viename iš ryšių, kitas ryšys tampa privalomu. Analogiškai ekvivalentiškumo apribojimas tarp atributų nurodo, kad jeigu vienas iš atributų įgyja reikšmę tai būtinai ir kitas atributas turi įgyti reikšmę, t.y. įgijus reikšmę vienam iš atributų, kitas atributas irgi tampa privalomu.
2.	Poaibio apribojimas	Šis apribojimas naudojamas neprivalomiems esybės atributams arba nebūtiniesiems esybės ryšiams. Poaibio apribojimas tarp dviejų esybės ryšių su kitomis esybėmis nurodo, kad aibė esybės egzempliorių dalyvaujančių abiejuose ryšiuose yra poaibis esybės egzempliorių dalyvaujančių viename ryšyje. Analogiškai poaibio apribojimas tarp dviejų esybės egzempliorių nurodo, kad aibė esybės egzempliorių su atributų reikšmėmis yra poaibis esybės egzempliorių turinčių reikšmes tik dalies atributų.
3.	Išskirtinumo apribojimas	Šis apribojimas naudojamas neprivalomiems esybių atributams arba ryšiams. Išskirtinumo apribojimas tarp esybės ryšių su kitomis esybėmis rodo, kad tas pats esybės egzempliorius gali dalyvauti tik viename iš kelių ryšių su kitos esybės objektais. Analogiškai išskirtinumo apribojimas tarp esybės atributų rodo, kad iš kelių neprivalomų esybės egzemplioriaus atributų reikšmę gali turėti tik vienas.

3 lentelė. Vientisumo apribojimai specializacijos/apibendrinimo ryšio tipui

Eil. Nr.	Pavadinimas	Paskirtis
1.	Specializacijos/ apibendrinimo ryšio	Apribojimas naudojamas specializacijos/apibendrinimo ryšiui. <ul style="list-style-type: none"> ▪ Jeigu potipių esybių egzemplioriai tarpusavyje nepersidengia ir pilnai

Eil. Nr.	Pavadinimas	Paskirtis
	apribojimai	<p>apima esybės egzempliorių aibę, tuomet specializacijos/apibendrinimo ryšys vadinamas suskaidymo;</p> <ul style="list-style-type: none"> ▪ Jeigu potipių esybių egzemplioriai tarpusavyje nepersidengia, tačiau pilnai nepadengia esybės egzempliorių aibės, tuomet specializacijos/apibendrinimo ryšys vadinamas išskirtinumo; ▪ Jeigu potipių esybių egzemplioriai tarpusavyje persidengia ir pilnai padengia esybės egzempliorių aibę, tuomet specializacijos/apibendrinimo ryšys yra pilnasis; ▪ Jeigu potipių esybių egzemplioriai tarpusavyje persidengia ir pilnai nepadengia esybės egzempliorių aibę, tuomet specializacijos/apibendrinimo ryšys –laisvasis.
2.	Refleksyviojo ryšio apribojimai	<p>Šis apribojimas naudojamas refleksyviam ryšiui:</p> <ul style="list-style-type: none"> ▪ Apribojimas „Nerefleksyvus“ (angl. <i>Irreflexive</i>) draudžia tai pačiai esybei dalyvauti abiejose ryšio rolėse tuo pačiu metu, t.y., esybės egzempliorius A negali turėti ryšio pats su savimi. ▪ Apribojimas „Asimetrinis“ (angl. <i>Asymmetric</i>) apriboja priešingos rolės egzistavimą tarp dviejų skirtingų esybės egzempliorių. Jei esybės egzempliorius A turi ryšį su tos pačios esybės egzemplioriumi B, egzempliorius B negali turėti ryšio su tokia pat role su tuo pačiu esybės A egzemplioriumi (jei A yra tėvas, o B sūnus, tai A negali būti B sūnus). ▪ Apribojimas „Neciklinis“ (angl. <i>Acyclic</i>) nurodo, kad negali egzistuoti ciklą tarp esybės egzempliorių susietų refleksyviuoju ryšiu. Jeigu esybės egzempliorius A turi ryšį su tos pačios esybės B egzemplioriumi ir egzempliorius B turi ryšį su tos pačios esybės C egzemplioriumi, egzemplioriaus C ryšys su A egzemplioriumi yra negalimas. ▪ Apribojimas „Netranzityvus“ (angl. <i>Intransitive</i>) užtikrina, kad egzistuoja tik tiesioginis „tėvo“ – „vaiko“ ryšys. Jeigu esybės egzempliorius A turi ryšį su tos pačios esybės B egzemplioriumi ir egzempliorius B turi ryšį su tos pačios esybės C egzemplioriumi, egzemplioriaus A ryšys su C egzemplioriumi yra negalimas. ▪ Apribojimas „Simetrinis“ (angl. <i>Symmetric</i>) reikalauja, kad priešinga rolė taip pat būtų teisinga. Jeigu egzistuoja egzemplioriaus A ryšys su B, tuomet egzistuoja ir egzemplioriaus B ryšys su A. ▪ Apribojimas „Nesimetrinis“ (angl. <i>Antisymmetric</i>) kaip ir „Asimetrinis“ apribojimas neleidžia egzistuoti priešingai rolei tarp skirtingų esybės egzempliorių, tačiau priešingai negu „Nerefleksyvus“ apribojimas leidžia, kad ta pati esybė dalyvautų abiejose ryšio rolėse. Šis apribojimas leidžia esybės egzemplioriumi A

Eil. Nr.	Pavadinimas	Paskirtis
		turėti ryšį su tos pačios esybės A egzemplioriumi ir leidžia egzemplioriumi A turėti ryšį su egzemplioriumi B, tačiau draudžia egzemplioriumi B turėti ryšį su egzemplioriumi A.
3.	Egzempliorių susiejimo ryšiais apribojimas	Šis apribojimas ryšiui nurodo, kad ne visi esybės egzemplioriai gali dalyvauti ryšyje, o tik tie esybės egzemplioriai, kurie dalyvauja nurodytų ryšių aibėje.

Apribojimų klasifikacijos analizė parodė jų svarbą IS kūrimui – DB apibrėžime vientisumo apribojimai turi būti, kad papildyti duomenų bazę informacija apie realaus pasaulio duomenų apribojimus ir apsaugoti ją nuo neleistinų reikšmių įvedimo. Tai reiškia, kad turi būti kontroliuojamos ir, jei reikia, nutraukiamos DML (angl. Data Manipulation Language) t.y., duomenų įvedimo, atnaujinimo, pašalinimo operacijos. Todėl reikalinga išnagrinėti esamus vientisumo apribojimų realizavimo būdus, priemones, sistemas – kaip šiuo metu kuriami, modifikuojami ir saugojami apribojimai, kokiomis notacijomis aprašomi, priemonėmis ir technologijomis – išgaunami, ar užtenka įrankių.

2.2. Vientisumo apribojimų realizavimo būdų analizė

Veiklos tikslas – išanalizavus vientisumo apribojimų užtikrinimo metodus ir priemones, išskyrus jų trūkumus ir privalumus, parengti centralizuotos vientisumo apribojimų valdymo sistemos koncepciją ir modelį, kurie bus naudojami projekto ir realizacijos etapuose.

Planuojami rezultatai:

- atlikta vientisumo apribojimų esamų užtikrinimo metodų, sistemų privalumų ir trūkumų analizė;
- suformuotas sistemų palyginimo kriterijų rinkinys;
- suformuotas kokybės kriterijų būsimai sistemai rinkinys;
- pasiūlytas sprendimas – vientisumo apribojimų užtikrinimo metodas, sistemos koncepcija ir modelis.

2.2.1. Vientisumo apribojimų realizavimo metodų analizė

Poskyriuje analizuojami duomenų vientisumo apribojimus įgyvendinantys metodai [21].

Apribojimai lentelėms – tai sąlygos, kurios nurodomos kaip lentelės apibrėžimo dalis ir nusako reikalavimus atributų domenams. Tos sąlygos turi būti išpildytos visiems lentelės įrašams. Sąlyga gali būti nurodyta konkrečiam atributui ar visai lentelei. Jie gali būti užtikrinami:

- taikomosiomis programomis (programiniame kode);
- DBVS apribojimais;
- trigeriais.

Toliau nagrinėjamas kiekvienas metodas.

- ***Taikomosios programos (“kietas programinis kodas”)***

Taikomojoje programoje užtikrinamos sąlygos, kad įvedant, šalinant ar modifikuojant įrašus būtų išlaikomas duomenų korektiškumas, teisingumas, išvengta įvairių duomenų anomalijų. Sąlygas įgyvendinti gana sudėtinga, nes sunku įvertinti visus apribojimus kūrimo metu, pakeitimų atlikimas dažniausiai iššaukia didesnę klaidų atsiradimo tikimybę, kodo dubliavimą, anksčiau nenagrinėtas situacijas, problemas. Šis metodas nėra tinkamas didelių sistemų kūrimui ir palaikymui, todėl sekančioje dalyje analizuojamos trigerių galimybės.

- ***Trigeriai***

Trigeriai, kaip ir deklaratyvūs apribojimai yra naudojami duomenų neprieštarinamumui palaikyti, vykdant DML SQL *Delete*, *Update* ar *Insert* operacijas. Trigeriai gali būti naudojami ir kitų, tiesiogiai nedalyvaujančių operacijoje, lentelių modifikavimui priklausomai nuo atliekamos operacijos pagrindinėje lentelėje (pvz. renkant statistiką apie kiekvieną duomenų modifikavimą lentelėje, pakeitus bet kurį įrašą lentelėje, kitoje lentelėje įtraukiamas naujas įrašas). Toliau pateikiama trigerių naudojimo specifika:

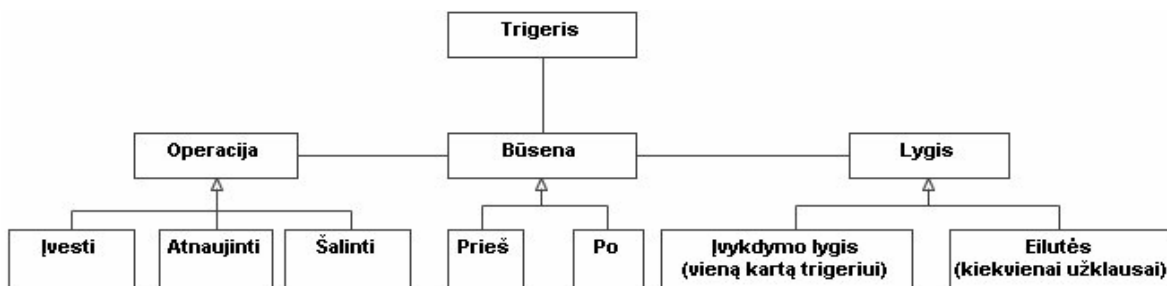
- koku momentu trigerį aktyvuoti: prieš vykdant modifikavimo operaciją ar po;
- ar jį aktyvuoti kiekvieną kartą vykdant SQL sakinį ar kiekvienai operacijoje dalyvaujančiai eilutei;
- vieną ar kelis sakinius, kuriuos reikia įvykdyti aktyvavus trigerį (trigerio kūnas);
- trigerį aktyvavus, jo kūne nurodomus sakinius vykdyti (be) sąlygiškai.

Pateikiamas trigerio operacijų, būsenų ir vykdymo metamodelis (2 pav.).
Metamodelyje, galimos operacijos – tai duomenų operacijos:

- įvesti;

- atnaujinti;
- šalinti.

Aprašomos dvi trigerio vykdymo būsenos – prieš arba po tam tikro veiksmo, situacijos. Toliau – vykdymo lygis: ar kartą suveikus trigeriui, ar kiekvienai lentelės eilutei – įrašui.



2 pav. Trigerių metamodelis

Norint naudoti trigerius, reikia tinkamai identifikuoti visas IS veiklos situacijas, nes netinkamai suveikusio trigerio atšaukti – neįmanoma, kas iššaukia duomenų atstatymo problemas, finansinius ir kitus nuostolius. Todėl tikslinga ieškoti kitų metodų, kurie leistų išvengti minėtų problemų.

▪ **DBVS apribojimai**

Dauguma DBVS gali palaikyti aibę (3 pav.) įvairių vientisumo apribojimų (jie apibrėžiami kuriant lenteles kartu su laukų tipais). Kiekviena apibrėžta taisyklė yra specifinė konkrečiai duomenų bazei. Tačiau, be specifinių taisyklių, reliaciniame modelyje yra dvi bendrosios vientisumo taisyklės, kurios privalomos visoms DBVS. Jos aprašo pirminius ir išorinius raktus.



3 pav. Vientisumo apribojimai

- Esybės integralumo užtikrinimo taisyklė – pirminio rakto reikšmė negali būti tuščia (angl. *null*).
- Išorinio rakto vientisumo taisyklė – turi būti ryšys tarp išorinio bei susieto pirminio raktų. Jei yra duomenų vaikinėje lentelėje, tai tėvinė lentelė negali būti pašalinta.

- Veiklos (angl. *business*) vientisumo užtikrinimo taisyklės skirtos IS dalykinės srities taisyklėms užtikrinti. DBVS jos užtikrinamos naudojant tikrinimo apribojimus (angl. *check constraint*).

DBVS apribojimai turi būti identifikuoti, projektuojami ir kuriami IS projektavimo metu. Dažniausiai jie neįvertinami tinkamai, o pakeisti jau egzituojančioje DBVS – sudėtinga: reikia gerų specialistų, sąlygų kaita neigiamai įtakoja duomenis, tenka spręsti istorinių (kurie neatitinka naujus apribojimus) duomenų saugojimo problemas.

Analizė parodė, kad nėra tinkamo metodo, todėl tolesniuose poskyriuose nagrinėjamos sistemos, apribojimų įgyvendinimui ir modifikavimui atlikti.

2.2.2. *Vientisumo apribojimus įgyvendinančių sistemų palyginimas*

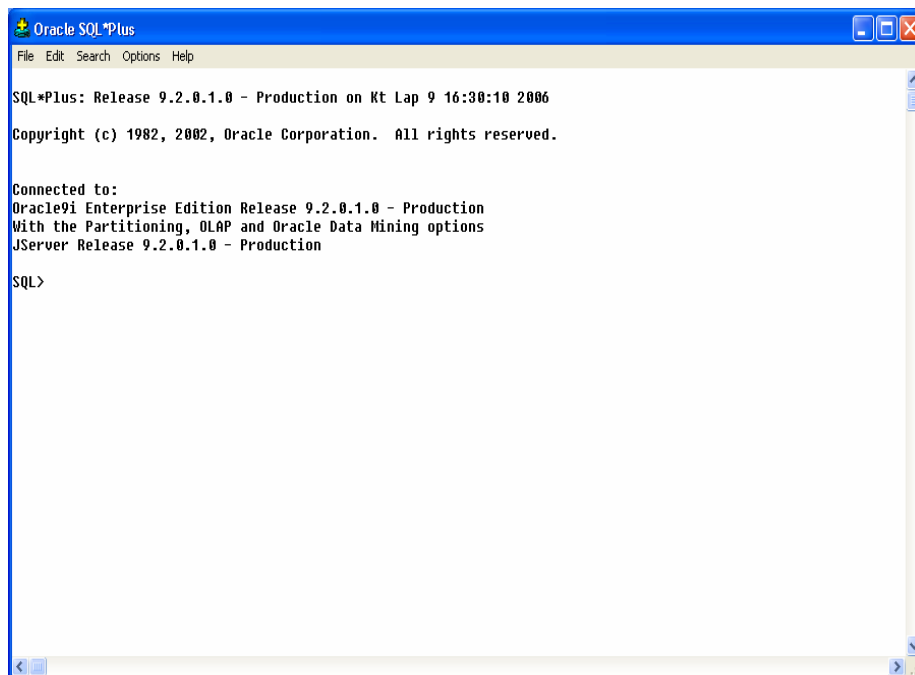
Poskyriuje nagrinėjamos sistemos, įrankiai, jų sukūrimo tikslai ir idėjos, principai. Tyrimo objektais pasirinkti:

- SQL įrankiai;
- Apribojimų DB sistemos (angl. *Constraint Databases*);
- Taikomosios sistemos, verslo sistemos;

Toliau nagrinėjami SQL įrankiai.

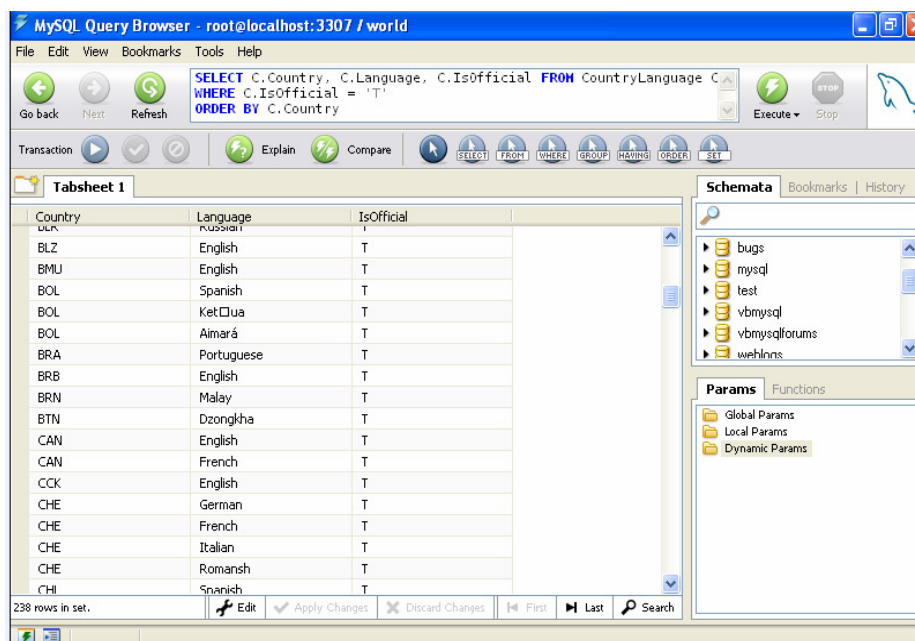
2.2.2.1. SQL įrankiai

SQL redaktoriai skirti darbui su DBVS, dažniausiai užklausių iš serverio vykdymui, duomenų išgavimui. SQL*Plus redaktorius (4 pav.), skirtas darbui su Oracle DBVS [30].



4 pav. SQL * PLUS redaktorius pagrindinis langas

MySQL Query Browser (5 pav.) – paprastas grafinis įrankis [31], skirtas darbui su MySQL duomenų bazių serveriu: kurti, vykdyti ir optimizuoti SQL užklausas.



5 pav. MySQL Query Browser redaktorius pagrindinis langas

Įrankyje įdiegta:

- Užklausos pateikimo langas – lengvai kurti ir vykdyti užklausas, gauti jų istoriją.
- SQL skriptų redaktorius – suteikia galimybes kurti ir modifikuoti SQL komandas.

- *Rezultatų langas* – skirtas darbui su keliomis užklausomis, jų rezultatų palyginimui.
- *Objektų naršyklė* – leidžia valdyti duomenų bases.
- *DB struktūros peržiūros langas*, suteiktos galimybės užklausų vykdymui išrinkti lenteles ir jų laukus, taip pat kurti ir šalinti lenteles.
- *Lentelių redaktorius pagalba* greitai ir lengvai kuriamos, modifikuojamos ir šalinamos lentelės.
- *Pagalbos sistema* vartotojui padeda susirasti reikiamą informaciją, naudojantis minėta sistema.

Priešingai nei minėtame įrankyje, SQL*Plus nėra grafinių komponentų DBVS objektų peržiūrai ir darbui su jais, todėl šis įrankis skirtas labiau patyrusiems specialistams, gerai žinantiems SQL ir objektinių – reliacinių DBVS principus. Oracle DBVS specialistai šį įrankį vertina už stabilumą, patikimumą, paprastumą (AB „Informacinės technologijos“ Oracle specializuotų mokymų vadovas).

2.2.2.2. Apribojimų DB sistemos

Apribojimų DB sistemų tikslas – išplėsti reliacinių DB paradigmą, leisti dirbti su įvairiu kiekiu netradicinių duomenų, skirtų GIS (angl. *Geographical Information Systems*), medicinos, mokslinėms sistemoms. Tokių bazių sąvoka ir nagrinėjimas atsirado 1990 metais ir jau yra sukurta keletas realizacijų (CQA/CDB, Dedale, C³, MLPQ/GIS, RuleML) [23], [24], [25], [26], [27], [28]. Toliau nagrinėjamos CQA/CDB, RuleML sistemos šiais aspektais:

- duomenų modelis;
- komponentinės architektūros modelis.

CQA/CDB sistemos apribojimų DB modelis – naujas ir perspektyvus erdviųjų objektų modeliavimo būdas GIS sistemose (angl. *Geographical Information Systems*), kuris leidžia atlikti operacijas ne baigtinėje erdvėje. Apribojimų DB aprašomi ir valdomi įvairaus masiškumo duomenys taikant vieningą strategiją ir įgalina vykdyti erdvines užklausas taikant realiaciones kalbas. Pateikiamas DB lentelės DDL (angl. *Data Definition Language*) skriptas, aprašantis žemės duomenų struktūrą (6 pav.), ir saugojamų duomenų pavyzdžiai 4-5 lentelės.

```
CREATE TABLE Land_use
  (Regiono_pavadinimas varchar(30),
  Žemės_tipas varchar(30),
  Koordinatės R(x, y),
  Primary Key (Regiono_pavadinimas))
```

6 pav. Lentelės DDL

4 lentelė. GIS duomenų lentelė

Regiono pavadinimas	Žemės tipas	Koordinatės
R1	Miškas	{(1, 1), (1, 1.0000001)..(1.000002)(3, 2) (3, 2.0000001)....(4, 4)}
R2	Žolė	{(12, 3), (12, 3.0000001)..(15, 6) (15, 16.0000006) }

Rinkinys
geografinių
duomenų

Pateikiama lentelė, kuri saugo GIS duomenis aprašytus apribojimais, naudojant CQA (angl. Constraint Query Algebra).

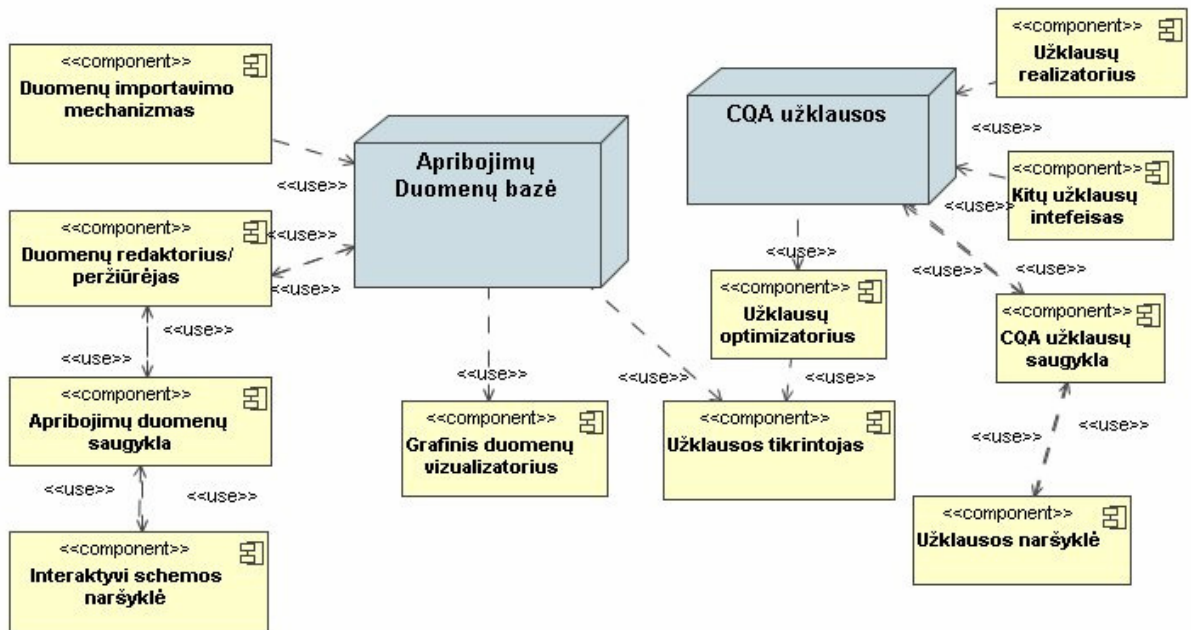
5 lentelė. GIS duomenų aprašymas apribojimais

Regiono pavadinimas	Žemės tipas	Koordinatės
R1	Miškas	$\{ x \leq 1 \wedge x \geq -1 \wedge y \leq 1 \wedge y \geq -1 \}$
R2	Žolė	$\{ x \geq 1 \wedge y \geq -1 \wedge x * y - 2 \leq 0 \}$

Apribojimas

CQA užklausa susideda iš primityvių operacijų, kurios prilyginamos reliacinės užklausių algebros operacijoms.

Toliau nagrinėjamas sistemos architektūros (7 pav.) modelis.



7 pav. CQA/CDB komponentai

Sistema kurta kaip komercinis produktas, todėl testuoti šios sistemos nepavyko. Žinoma, kad realizuota J2EE technologijos pagrindu. 6 lentelėje aprašomi komponentai, jų paskirtis.

6 lentelė. CQA/CDB sistemos architektūros komponentų paskirties aprašymas

Komponentas	Paskirtis
Interaktyvi schemas naršyklė (angl. <i>Interactive Schema Browser</i>)	Nuskaityti DBVS schemas objektus (lenteles, jų laukus ir kt.).
Apribojimų duomenų saugykla (angl. <i>Constraint Data Repository</i>)	Centralizuotai saugoti apribojimus.
CQA (angl. <i>Constraint Query Algebra</i>) užklausa	Apribojimai aprašomi CQA iš kurių formuojamos atitinkamos užklausa, tolesniems vykdymo etapams.
Grafinis duomenų vizualizatorius (angl. <i>Graphical Data Visualizer</i>)	Kadangi tai GIS sistema, tai realizuotas posistemis, leidžiantis atvaizduoti užrašytus apribojimus grafiškai (pvz. kur yra tam tikras objektas – žemė, ežeras, kelias ir kt.)
Užklauskų tikrintojas (angl. <i>Query evaluator</i>)	Apribojimo teisingumui atlikti, realizuotas užklauskų tikrinimo mechanizmas.
Duomenų importavimo mechanizmas	Suteiktos galimybės importuoti turimus duomenis į DBVS.
Duomenų redaktorius	Skirtas peržiūrėti užklauskoms, importuotiems duomenims.
Užklauskų optimizatorius	Mechanizmas leidžia optimizuoti užklauskas, greitesniam informacijos apdorojimui, išrinkimui pagal tam tikrus kriterijus.
CQA užklauskų saugykla	Centralizuotai saugoti CQA užklauskas.
Užklauskų naršyklė	Suteikti vartotojui galimybes greitai peržiūrėti ir redaguoti užklauskas.
Kitų užklauskų interfeisas	Leidžia peržiūrėti, redaguoti ne tik CQA užklauskas, bet ir SQL.

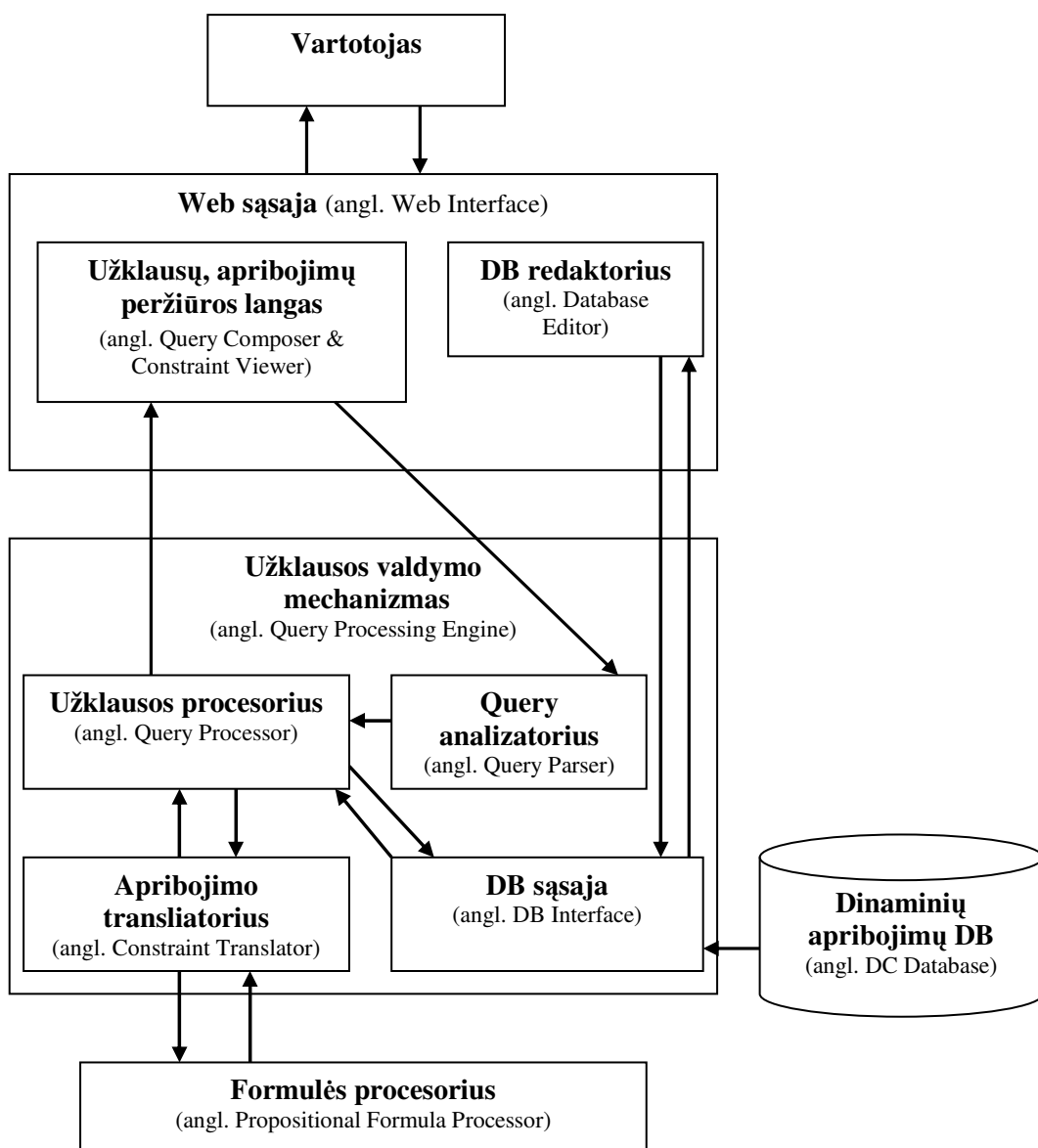
RuleML [28], skirta veiklos apribojimų kūrimui ir išgavimui, naudojant dinaminį apribojimų algebrą DCA (angl. Dynamic Constraint Algebra).

Toliau nagrinėsime šios sistemos architektūrą (8 pav.). Išskirdami nagrinėjamų komponentų sukūrimo tikslus, esminius bruožus, privalumus ir trūkumus.

Detalizuojamos komponentų funkcijos:

- *DB interfeisas* nagrinėja/apdoroja (angl. *Parses*) įvedimą į apribojimų DB, kur kiekviena lentelė yra formatuojama naudojant apribojimų kalbą, pateiktą 15 pav.
- *Užklauskų apdorotojas* (angl. *Query Parser*) – apdoroja apribojimų failų įvedimą ir atlieka aukšto lygio užklauskų valdymą, procesų hierarchijoje.
- *Užklauskų Procesorius* (angl. *Query Processor*) išskiria dinaminis apribojimus ir įveda tarpines (angl. *intermediate*) operacijas, perkelia į apribojimų transliatorių (angl. *Constraints Translator*).
- Tarpinės operacijos yra generuojamos iš DCA operacijų ir naudojamos patikrinimui dinaminį apribojimų dalių.
- Apribojimų transliatorius (angl. *Constraint Translator*) transformuoja dinaminis apribojimus į formules, generuoja logines (angl. *boolean*)

operacijas iš perkeltų tarpinių operacijų. Rezultatas siunčiamas į formulių procesorių (angl. Propositional Formula Processor), kur atliekamas testavimas. Iš ten duomenys siunčiami užklausos valdymo dali, sukurti ryšį.



8 pav. RuleML architektūros modelis

7-8 lentelėse nagrinėjami aukščiau aprašytų sistemų komponentai ir apribojimų aprašymo būdai, pateikiami alternatyvūs sprendimai vientisumo apribojimų sistemos sukūrimui.

7 lentelė. Apribojimų sistemų analizė ir alternatyvos

Sistema	Esamas sprendimas	Alternatyva
RULEML	Apribojimai aprašomi DCA (9 pav.)	Apribojimus aprašyti SQL loginėmis išraiškomis. Duomenų išrinkimui kartu su apribojimais naudoti SQL BNF notaciją (10 pav.).
CQA/CDB	Jungiami klasikiniai ir erdviniai (apribojimo kūrimui) duomenys.	Atskirti klasikinius duomenis, nuo apribojimus realizuojančių.
	Naudojama apribojimų užklausos kalba.	Naudoti SQL.
	Naudojamos notacijos DNF(angl. <i>Disjunctive Normal Form</i>).	SQL aprašyti naudojant BNF notaciją.
	DB lentelių rinkinys, kuriuose saugojami įrašai (taškai).	DB lentelėse galima saugoti apribojimų logines išraiškas, jų predikatus, jungimo operacijas.
	Apribojimo duomenys aprašomi I eilės polinomais.	Apribojimas aprašomas SQL operatoriais, matematiniais ir loginiais predikatais.

8 lentelė. Apribojimų sistemų architektūros komponentų analizė ir alternatyvos

Sistema	Komponentas	Alternatyva
CQA/CDB	Interaktyvi schemas naršyklė (angl. Interactive Schema Browser)	Sukurti įrankį, kuris automatiškai nuskaitytų IS DBVS schemas objektus (lenteles ir jų laukus, realizuotas procedūras ir jų parametrus)
	Apribojimų duomenų saugykla (angl. Constraint Data Repository)	Vientisumo apribojimus saugoti įvairiose DBVS, priklausomai nuo IS, kuriai aprašomi.
	CQA (angl. Constraint Query Algebra) užklausos	Vientisumo apribojimus aprašyti naudojant SQL operatorius, saugykloje saugoti kaip logines išraiškas (angl. boolean expressions).
	Grafinis duomenų vizualizatorius (angl. Graphical Data Visualizer)	Vientisumo apribojimus kurti patogioje grafinėje aplinkoje. Efektyvų įvedimą ir redagavimą suteikia dvimačių celių (angl. Grid) komponentas.
	Užklausos tikrintojas (angl. Query evaluator)	Vientisumo apribojimų teisingumą atlikti dinamiškai.

```

<!ELEMENT table (tuple)*>
<!ATTLIST table name CDATA #IMPLIED>
<!ELEMENT tuple ((value)*,(cond|condxml))>
<!ATTLIST tuple tid NMTOKEN #IMPLIED>
<!ELEMENT value (cons|var)>
<!ATTLIST value atrid NMTOKEN #IMPLIED atrname CDATA #IMPLIED>
<!ELEMENT cond (#PCDATA)>
<!ELEMENT condxml (expr)+>
<!ELEMENT expr (par|eq|neq|fdef|and|or|xor|imply|not|func|dva|bool)*>
<!ELEMENT fdef (func,expr)>
<!ATTLIST fdef delay (0|1) #REQUIRED>
<!ELEMENT eq ((var|cons),(var|cons))>
<!ELEMENT neq ((var|cons),(var|cons))>
<!ELEMENT par (par|eq|neq|fdef|and|or|xor|imply|not|func|dva|bool)*>
<!ELEMENT and EMPTY>
<!ELEMENT or EMPTY>
<!ELEMENT xor EMPTY>
<!ELEMENT imply EMPTY>
<!ELEMENT not EMPTY>
<!ELEMENT func EMPTY>
<!ATTLIST func name CDATA #REQUIRED>
<!ELEMENT var EMPTY>
<!ATTLIST var name CDATA #REQUIRED>
<!ELEMENT bool EMPTY>
<!ATTLIST bool value (0|1) #REQUIRED>
<!ELEMENT cons (#PCDATA)>
<!ELEMENT dva (func,(var|(bool|(var|bool)*)))>
<!ATTLIST dva type (bool|card|dep) #REQUIRED>

```

9 pav. Apribojimų kalba aprašytas DTD

1. Select_Išraiška	::= SELECT * FROM T ₁ , T ₂ , ..., T _n [WHERE Predikatas]
2. Predikatas	::= Lauko palyginimas su lauku NOT EXISTS (<i>Paprastas_Select</i>) Laukas NOT IN (<i>Paprastas_Select</i>) Predikatas AND Predikatas
3. Laukas {Item}	::= Lauko_Pavadinimas Konstanta
4. Lauko_Pavadinimas	::= [T].C
5. Palyginimas	::= <= = >= > <
6. Paprastas_Select	::= SELECT * FROM T [WHERE Paprastas_Predikatas]
7. Paprastas_Predikatas	::= Laukas = Laukas Paprastas_Predikatas AND Paprastas_Predikatas

10 pav. SQL aprašas naudojant BNF notaciją

SQL aprašo (10 pav.), naudojant BNF notaciją, žymėjimai:

- T₁, T₂, T₃, ..., T_n – DB lentelės;
- C – DB lentelės laukas; optimizuojant užklausoos vykdymą, geriausia nurodyti, iš kokios lentelės imamas laukas, t.y., [T₁].C, kur T₁ – lentelės pavadinimas, C – lauko pavadinimas.

- AND, OR – predikatai.
- <=, >=, =, <, > - palyginimo tipai.

Toliau poskyriuje nagrinėjama kaip taikomosios ir verslo sistemos, įgyvendina vientisumo apribojimus.

2.2.2.3. Taikomosios, verslo sistemos

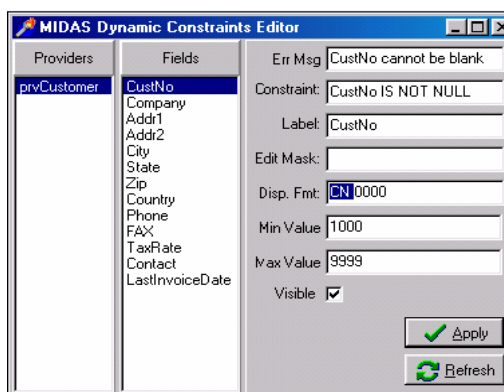
Šiame poskyriuje nagrinėjama kaip realizuojami vientisumo apribojimai sistemose, ar užtikrinamas juose duomenų vientisumas ir neprieštarinumas: Expression Filter, MIDAS, jų taikymo sritys.

PĮ *Expression Filter* [32] funkcionalumas prieinamas naudojant Oracle10g programinę įrangą. Grafinio interfeiso, skirto vartotojams kurti veiklos apribojimus nėra. Ši programinė įranga naudojama įvairių apribojimų kūrimui ir palaikymui, skirtų verslo spendimų priėmimui. Apribojimai s aprašomos verslo taisyklės. Veiklos logika pagrįsta ECA (angl. Event – Condition – Action) ontologija (9 lentelė).

9 lentelė. Ontologijos aprašymas

Ontologija	Expression Filter	Alternatyva/Naujas sprendimas
Įvykis	DML (duomenų įvedimo, pašalinimo, keitimo) funkcijos	
Taisyklės sąlygos	+	
Taisyklės veiksmas	Palaikyti verslo procesus (pasiūlyti skrydžio lėktuvu bilietą, viešbutį ir kt. paslaugas, pasiūlymus).	Užtikrinti duomenų vientisumą, leisdami /neleisdami atlikti DML operacijų.

MIDAS (angl. *Multi-Tier Distributed Application Services*) [7] (11 pav.) programinė įranga priešingai nei Expression Filter palaiko dinaminių duomenų bazių apribojimų įgyvendinimą.



11 pav. MIDAS programos langas

MIDAS programinės įrangos duomenų paketo sudėtis: metaduomenys, apribojimai, duomenys. Duomenų laukų apribojimai, jų savybės ir aprašymai pateikti 10 lentelėje.

10 lentelė. Apribojimo struktūros aprašymas

Apribojimo savybės	Aprašymas
Apribojimo klaidos pranešimas	Rodomas pranešimas jei įvesti duomenys negalimi.
Vartotojo apribojimas	Nurodomas apribojimas laukui, Pvz. CustNo negali būti tuščia. (angl. IS NOT NULL).
Lauko formatas	Laukas, duomenų formato kontrolės užtikrinimui.
Aprašymo laukas	Naudojamas duomenų bazės lentelės lauko pavadinimas. Pvz. Customer #
Redaguojamas laukas	Lauko formato valdymui Pvz: 0000
Maksimali reikšmė	Maksimali reikšmė skaitmeniniam laukui.
Minimali reikšmė	Minimali reikšmė skaitmeniniam laukui.
Matomumas	Nurodoma, ar laukas turi būti matomas ar ne.

Naudojantis šiuo įrankiu, vartotojai gali kurti ir modifikuoti vientisumo apribojimus esybių atributams: *pirminio identifikatoriaus, paprasto būtinumo, disjunktyvaus būtinumo, unikalumo, išorinio unikalumo, atributo reikšmės*. Programinė įranga nerealizuoja vientisumo apribojimų asociacijų ryšiams tarp esybių (*ekvivalentiškumo apribojimas, poaibio apribojimas, išskirtinumo apribojimas*), taip pat vientisumo apribojimų specializacijos/apibendrinimo ryšio tipui (*specializacijos/apibendrinimo ryšio apribojimai, refleksyviojo ryšio apribojimai, egzempliorių susiejimo ryšiais apribojimas*). Šių apribojimų neįvertinimas dažniausiai iššaukia duomenų nekoretiškumo, įrašų įvedimo, redagavimo ir šalinimo problemas.

Sekančiame poskyriuje apibendrinamos nagrinėtos sistemos (jų funkcijos, galimybės), kad priimti sprendimą naujos PĮ kūrimui, kuri įgyvendintų visus iškeltus kriterijus (11 lentelė).

2.2.2.4. Lyginamoji sistemų analizė

Poskyriuje lyginamos aukščiau nagrinėtos verslo, SQL įrankiai, taikomosios sistemos, apribojimų bazės tarpusavyje pagal pasirinktus kriterijus (11 lentelė).

Žymėjimo reikšmės:

- “+” – sistema įgyvendina pasirinktą kriterijų.
- “-“ – sistema neatitinka keliamą kriterijų.

Didžiausias dėmesys skiriamas šioms sistemų savybėms:

- Duomenų teisingumo ir neprieštaravimo užtikrinimas;
- Apribojimų įvedimo, modifikavimo, pašalinimo paprastumas;
- Centralizuotas ir automatizuotas apribojimų įgyvendinimas;
- Įvairių apribojimų DBVS panaudojimo galimybės.

11 lentelė. Sistemų lyginamoji analizė

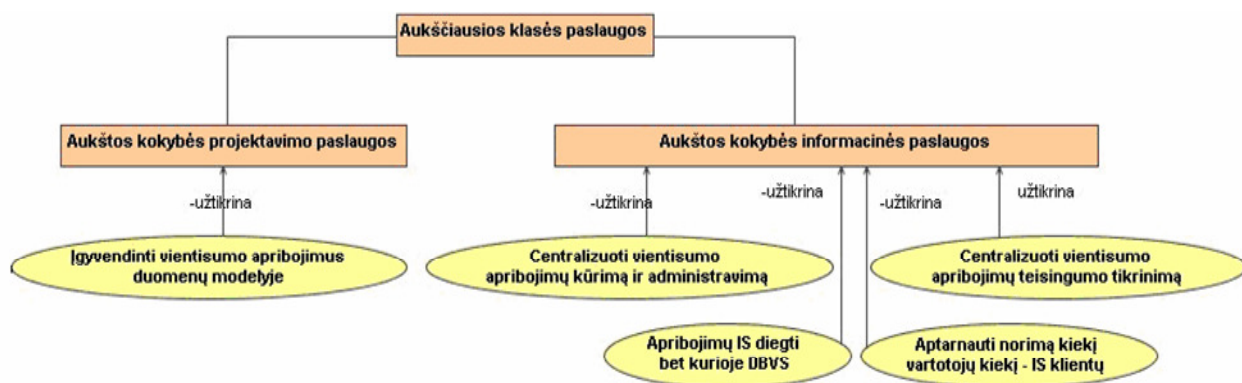
Kriterijus	Sistema	Expression Filter	Apribojimų DB	MIDAS	SQL įrankiai
Duomenų teisingumo ir neprieštaravimo užtikrinimas		-	-	+	+
Apribojimų įvedimo, modifikavimo, pašalinimo paprastumas		-	-	+	-
Struktūrizuotas apribojimų aprašymas		+	+	+	-
Apribojimų sąlygos duomenų bazėje saugojamos kaip išraiškos		+	+	-	-
Centralizuotas ir automatizuotas apribojimų įgyvendinimas		+	+	+	-
Apribojimų teisingumo dinaminis tikrinimas		-	-	+	-
Apribojimų parametrizavimas naudojant procedūras (stored –procedures)		+	-	-	-
SQL dinaminis naudojimas		+	-	-	-
Efektyvus apribojimų GUI		-	-	+	-
Verslo analizės sistemų integravimas		-	-	-	-
Web servisų įgyvendinimo galimybės apribojimų išgavimui		-	-	-	-
Greitas sistemos įdiegimas		-	-	+	+
Sistemos naudojimo paprastumas		-	+	+	+
Įvairių apribojimų DBVS panaudojimo galimybės		-	-	-	+
Sudėtinių užklausų palaikymas (sub-query)		-	+	-	+
Vartotojo teisių ribojimo galimybės		+	-	+	+

Lyginamoji analizė rodo, kad nei viena sistema netenkina iškeltų kriterijų ir reikalinga kurti naują sistemą. Todėl toliau poskyriuose aprašomi veiklos tikslai, vartotojų poreikiai ir įgyvendinamo proceso nauda.

2.3. Veiklos tikslai

Pagrindinis veiklos tikslas – kompiuterizuoti vientisumo apribojimų valdymo procesą, kurį taikant IS veiklos taisyklių, vientisumo apribojimų, jų pakeitimų realizavimas padarytų

kuo mažiau finansinės, struktūrinės žalos ir trukmę ku trumpesnį laiką. Tikslų modelyje (12 pav.) detalizuojami keliami veiklos tikslai.

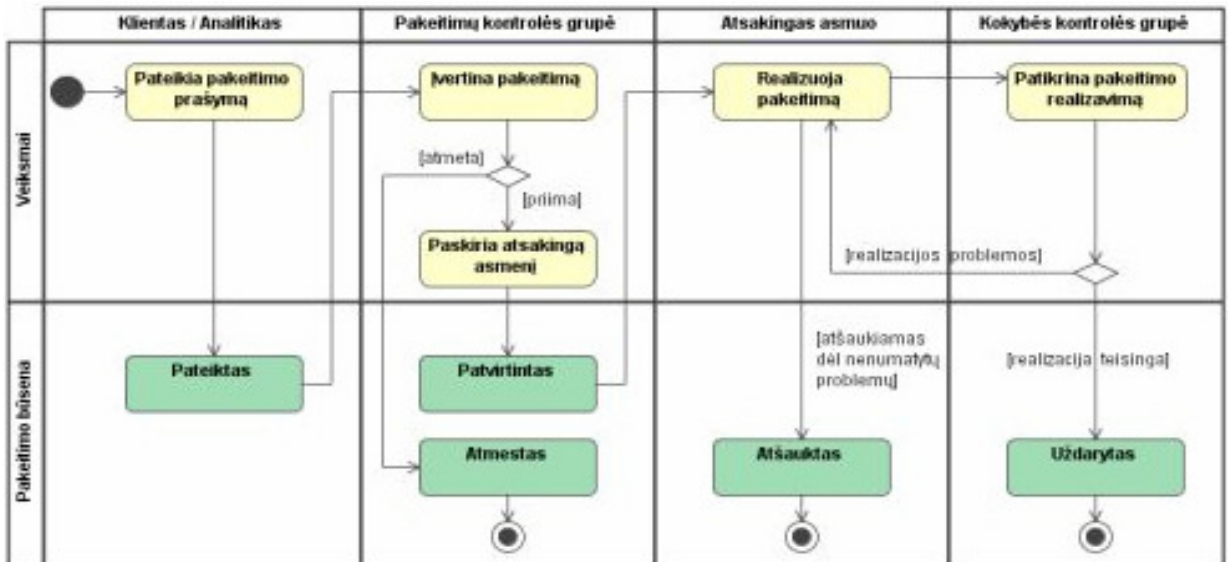


12 pav. Veiklos tikslų modelis

- užtikrinti aukščiausias projektavimo ir informacines paslaugas, užtikrinančias vientisumo apribojimų įgyvendinimą duomenų modelyje;
- teikti centralizuotas ir struktūrizuotas vientisumo apribojimų kūrimo ir administravimo funkcijas;
- centralizuoti apribojimų teisingumo tikrinimą;
- apribojimų IS įdiegti bet kurioje DBVS;
- aptarnauti norimą kiekį vartotojų – IS klientų vienu metu.

2.4. Vientisumo apribojimų įgyvendinimo sistemoje procesas

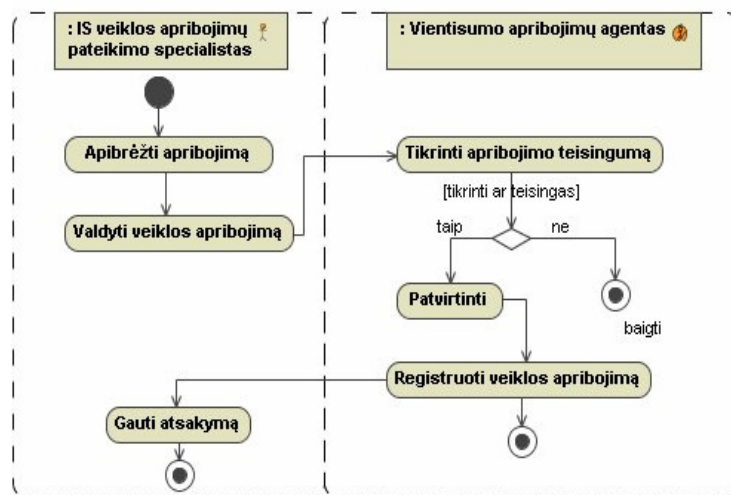
Vientisumo apribojimų valdymas [4] – neatsiejama IS kūrimo ir palaikymo proceso dalis, veikla, kurios metu vykdomi IS tobulinimai, keičiamasi informacija, užtikrinamas teisingas, nepriekaištingas įmonės IS funkcionalumas, duomenų teisingumas ir neprieštarinumas. Pateikiamas IS pakeitimų, apimančių ir vientisumo apribojimus, valdymo proceso modelis (13 pav.) kartu su būsenomis. Veikloje dalyvauja klientas ar analitikas, pakeitimų kontrolės grupė ir atsakingas asmuo.



13 pav. Pakeitimų valdymo proceso modelis

Siekiamo įgyvendinti vientisumo apribojimų valdymo modelio (14 pav.) tikslas – struktūrizuoti apribojimų kūrimo ir naudojimo procesą, taip kad būtų išskaidytas į lengvai kontroliuojamas ir nepriklausomai valdomas dalis.

Apribojimų valdymo kriterijai: tinkamumas, efektyvumas, vientisumas (pirminiai), prieinamumas (antrinis), patikimumas.



14 pav. Norimas įgyvendinti proceso modelis

Proceso modelyje *apribojimų pateikimo specialistas*:

- apibrėžtų apribojimo reikalingumą, sąlygas;
- valdytų (kurtų, modifikuotų, šalintų);

Vientisumo apribojimų agentas – sistema, tikrintų teisingumą, leistų valdyti apribojimų DB.

2.5. Vartotojų bei jų poreikių analizė

Esamo ir norimo įgyvendinti proceso analizė parodė, kad:

- Sistemų analitikams reikia metodo, padedančio užtikrinti vientisumo sąlygų tenkinimą būsimoje IS.
- Informacinių sistemų projektuotojams reikalingas įrankis, kuris užtikrintų duomenų modelio korektiškumą, išsamumą ir teisingumą.
- Vientisumo apribojimų pateikimo specialistams būtina įvesti ir koreguoti duomenų bazės vientisumo apribojimus, tikrinti jų sąlygas – ar teisingos, logiškos, taip apsaugojant nuo neleistinų, atsitiktinių duomenų keitimų būsimoje IS, patogaus įrankio pagalba.

2.6. Vientisumo apribojimų valdymo sistemos veiklos sąveikų analizė

Veiklos sąveikų analizė skirta procesams, funkcijoms, duomenų vienetams išanalizuoti, rasti reikalavimus būsimai sistemai. Todėl šiame skyriuje pateikiamos vientisumo apribojimus įgyvendinimui reikalingos sistemos ir vartotojų veiklos, esama situacija, procesai per veiklos panaudojimo atvejus, identifikuojami pagrindiniai veiklos subjektai.

▪ *Veiklos subjektai*

Pateikiami veiklos subjektai, dalyvaujantys vientisumo apribojimų valdymo sistemoje.

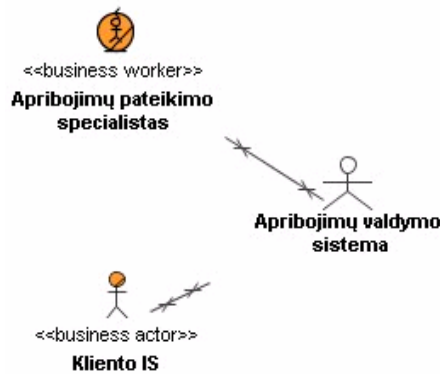
Veiklos subjektai:

- IS projektuotojas;
- Klientas;
- Apribojimų pateikimo specialistas;
- Apribojimų IS;
- Apribojimų tikrintojas;
- Apribojimų redaktorius.

Toliau analizuojamos sąveikos tarp jų ir veiklos panaudojimo atvejų, sudarant pagrindinius veiklos modelius.

- ***Veiklos sąveikų modelis***

Abstrakčiausiame lygyje, vientisumo apribojimų valdymo sistemą veiklą vaizduoja kontekstinė diagrama (15 pav.) – sistemos veiklos sąveikų modelis. Veiklos subjektai – vaizduojami veiklos aktoriais, centre rodoma norima kurti apribojimus įgyvendinančią sistemą.



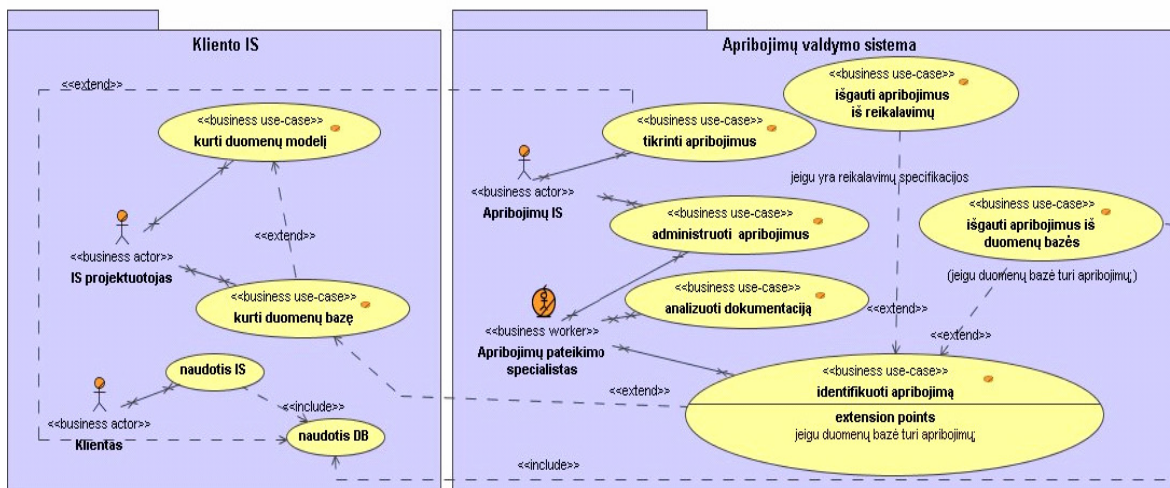
15 pav. Aukščiausio lygio apribojimų valdymo kontekstinė diagrama

Pateiktoje kontekstinėje diagramoje, vientisumo apribojimų valdymo sistema yra susijusi su tam tikra organizacijos IS, kuriai vientisumo apribojimus pateikia apribojimų pateikimo specialistas.

Toliau modeliuojamos vientisumo apribojimų valdymo sistemos detalesnė ir detaliausia sąveikų diagramos (16, 17 pav.), kurios rodo būsimos sistemos veiklos procesus, kuriuose dalyvauja 2.5.1. aprašyti veiklos subjektai.

- ***Detalesnis veiklos sąveikų modelis***

Vidiniai ir išoriniai sistemos aktoriai, jų veiklos, pateikiamos veiklos panaudojimo atvejų diagramoje (16 pav.).



16 pav. Detalus veiklos panaudojimo atvejų modelis

Modelyje organizacijos *IS projektuotojas* atsakingas už informacinės sistemos duomenų modelio, jo apribojimų ir duomenų bazės korektišką, išsamų, teisingą ir logišką identifikavimą ir projektavimą.

Klientas – organizacijos IS, kuri:

- naudojasi DB (duomenys struktūrizuotai saugojami tam tikroje DBVS);
- naudojasi vientisumo apribojimų užtikrinimo IS (atskira sistema, kur saugojami vientisumo ir veiklos apribojimai).

Apribojimų pateikimo specialistas atsakingas už informacinės sistemos veiklos apribojimų identifikavimą ir administravimą (įvedimą, koregavimą, šalinimą, peržiūrą). Norint įgyvendinti šias veiklas, šis specialistas turi analizuoti sistemos dokumentaciją, išgauti reikalavimus:

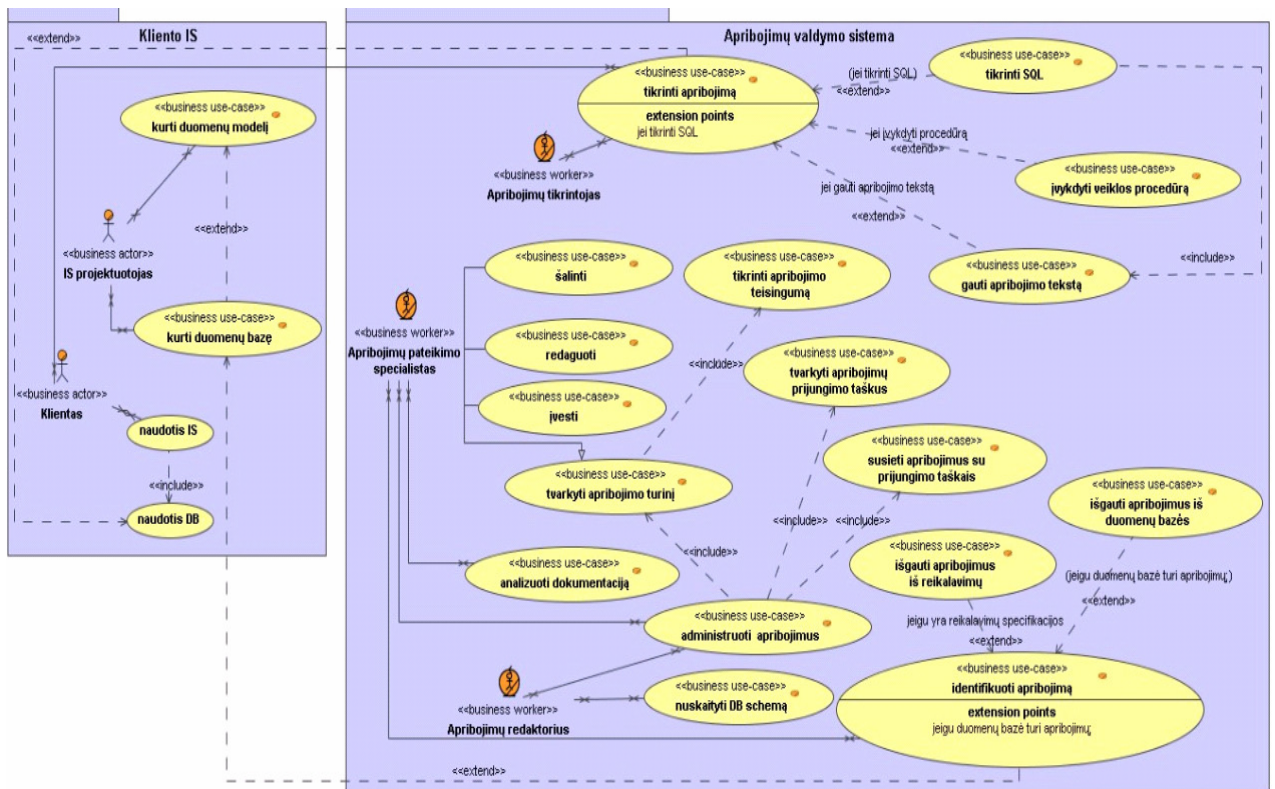
- iš reikalavimų dokumentų;
- nagrinėti fizinę duomenų bazės schemą;
- atlikti atvirkštinę inžineriją.

Apribojimų valdymo sistema pateikiama kaip vienetas, kurio veiklos: tikrinti apribojimų teisingumą ir leisti administruoti juos, naudojantis apribojimų DB.

Aktorių veiklos toliau detalizuojamos sekančioje dalyje.

▪ ***Detalizuotas veiklos sąveikų modelis***

Subjektų veiklos išplečiamos detalizuotoje veiklos panaudojimo atvejų diagramoje (17 pav.)



17 pav. Detalizuotas veiklos panaudojimo atvejų modelis

Pagrindinis veiklos panaudojimo atvejis – „administruoti apribojimus“. Veiklą išplečia šie veiklos panaudojimo atvejai:

- nuskaityti DB schemas objektus (lenteles ir jų laukus);
- tvarkyti apribojimų prijungimo taškus (tai IS DBVS lenteles ir jų laukai, kuriems kuriami apribojimai);
- susieti apribojimus su prijungimo taškais;
- tvarkyti (įvesti, redaguoti, šalinti, tikrinti) apribojimo turinį.

Veiklos panaudojimo atvejis – „tikrinti apribojimą“. Veiklą išplečia šie panaudojimo atvejai:

- naudotis DB (tai apribojimų saugojimo meta-DB);
- tikrinti SQL (tikrinamos DML (angl. *Data Manipulation Language*) funkcijos);
- įvykdyti veiklos procedūrą;
- gauti apribojimo tekstą.

2.7. DBVS analizė, vientisumo apribojimų valdymo sistemos realizavimui

Poskyriuje nagrinėjamos pačios populiariausios DBVS (MySQL, MS SQL Server 2005, Oracle9i), jų esminiai bruožai.

Visų minėtų DBVS duomenys prieinami per lentelių abstrakciją, kuri aprašo ryšius tarp skirtingų lentelių ar jų dalių. Duomenims įvesti, keisti, ieškoti bei lentelėms ir duomenų bazei valdyti – naudojama kalba SQL. 12 lentelėje pateikiamas populiariausių ir dažniausiai naudojamų mokymosi įstaigose, versle, tyrimuose DBVS palyginimas pagal kriterijus.

12 lentelė. Populiariausių DBVS lyginamoji analizė

Savybė	MySQL 5.1	Oracle9i	MS SQL Server 2005
DML funkcijos (INSERT, DELETE, UPDATE)	+	+	+
Sinonimų kūrimas	-	+	-
Duomenų įvedimas į kelias lenteles, prie tam tikrų apribojimų (angl. multi – table insert)	-	+	-
Turi procedūrinę kalbą	-	+ (PL/SQL)	+ (T-Transact)
Procedūros, trigeriai, atvaizdai, kursoriai	+	+	+
Sudėtinės užklausos	+	+	+
Apribojimai	+	+	+
Išoriniai raktai	+	+	+
Hierarchinės užklausos	-	+	+
Lentelių jungimai (angl. full outer join)	-	+	+
Veikia daugelyje platformų	+	+	-
Dažnai pasirenkama kuriant internetines svetaines	+	-	-
Taikoma didelių sistemų kūrimui	+-	+	+
Transakcijų valdymas	+	+	+
Dažniausiai naudojama priėjimo kalba	PHP	.NET palaikomos kalbos	Java, .NET palaikomos kalbos
Platus spektras duomenų tipų (subsekundžių laiko reikšmės, laiko zonos formatai)	-	+	-

MS SQL Server 2005 privalumai:

- pigesnė nei Oracle9i;
- greitesnis ir paprastesnis diegimas, palaikymas ir naudojimas.

Esminis trūkumas – DBVS palaikomos tik Windows OS platformoje.

MySQL privalumai:

- nemokama;
- gera sistema internetinių sistemų kūrimui;

- reikalauja mažiau techninių resursų lyginant su MS SQL Server 2005, ar Oracle9i.

Trūkumai:

- mažesnis duomenų tipų palaikymas ir didelis jų dydžių ribojimas, lyginant su Oracle9i;
- neturi eilutės identifikatoriaus, kuris reikalingas greitam duomenų modifikavimui, lyginant su Oracle9i.
- nėra rolių kūrimo ir administravimo funkcijų, priėjimui prie objektų riboti, lyginant su Oracle9i, MS SQL Server 2005.
- naujos versijos išleidžiamos gerai jų neištestavus, todėl nėra patikimos ir stabilios tam tikrą laiką. Tai pagrindinis kriterijus, kodėl retai naudojamos dideliems verslo projektams įgyvendinti.

Oracle9i privalumai:

- palaiko visas gerai žinomas OS (AIX, Compaq Tru64 UNIX, HP 9000 serija HP-UX, Linux Intel, Sun Solaris ir kt.), ne tik Windows.
- PL/SQL efektyvesnė procedūrinė kalba tegu T- SQL (MS SQL Server 2005).
- stabilus, patikimas, laiko patikrintas.

Trūkumai:

- brangesnė sistema, tačiau platus spektras galimybių kompensuoja kainų skirtumą lyginant su kitomis analizuotomis DBVS;

Visų minėtų DBVS galimybės – panašios. Pasirinkimą lemia – turimos lėšos, skirtos tokių sistemų įdiegimui, OS platformos, DBVS administravimo patirtis ir žinios, sprendžiamos problemos ir norimų įgyvendinti projektų apimtys, laikas. Taip pat DBVS stabilumas, patikimumas, lengvas ir patogus administravimas, sistemos vartotojų kiekis.

2.7.1. Oracle DBVS parinkimo pagrindimas

Oracle9i esminis pasirinkimo privalumas lyginant su kitomis DBVS – įvedimo į kelias lenteles funkcijų įgyvendinimo galimybės [5], [6], kurios yra viena iš technikų įgyvendinant SQL duomenų transformacijas naudojant tik vieną SQL užklausą (Veiklos diagramos pateikiamos 8.1 Priede). Šio funkcionalumo įgyvendinimas leis išplėsti vientisumo apribojimų naudojimo sritį, išvengiant daugelio INSERT sakinių rašymo. Oracle ETL procesas toliau detalizuojamas sekančiame poskyriuje.

2.7.1.1. Oracle ETL procesas

Anksčiau pagal nusistovėjusią ETL (angl. Extract, Transformation and Loading) paradigmą, šis procesas buvo vykdomas ne DBVS viduje, o tam tikrais nuo DBVS nepriklausomais įrankiais. Tačiau galima pastebėti tokio naudojimo keletą trūkumų:

- ETL įrankiai, išnaudoja didesnius CPU resursus, nes nėra jų kontrolės apsaugos.
- Įrankiai neintegruojami į DBVS.
- ETL programuotojai dažniausiai tampa naudojamų įrankių ekspertais, tačiau menkos jų žinios kaip įrankis susijęs su DBVS.
- Įrankiai nepalaiko arba nesuderinami su naujomis DBVS versijomis, neišnaudoja DBVS naujų versijų privalumų.

Todėl siekiant greito ir efektyvaus ETL proceso įgyvendinimo, reikalinga žinoti naudojamo įrankio ir DBVS teikiamus privalumus.

Oracle 9i DBVS galimybės keičia ETL paradigmą, nes palaiko šia svarbią savybę:

- *Sąlyginį kelių lentelių įvedimą (angl. Condition multi-table insert).*

Naudojant Oracle9i DBVS, paprasta įvedimo užklausa gali būti tinkamai panaudojama sąlyginio ar besąlyginio įvedimo į kelias lenteles metu. Tai ypač efektinga nei vykdyti keletą įvedimo sakinių. Naudojama struktūra “INSERT...SELECT” – formato išplėtimas.

Įvedimo į kelias lenteles tipai:

- besąlyginis įvedimas (angl. unconditional insert);
- sąlyginis įvedimas (angl. conditional all insert);
- sąlyginis pirminis įvedimas (angl. conditional first insert);

Sąlyginis įvedimo sakiny bus įvedama, jei nurodyta “WHEN” sąlyga – teisinga. Įvedimas galimas naudojant Oracle9i struktūras “FIRST INSERT” arba “ALL WHEN”, jei jos teisingos:

- Struktūros “FIRST INSERT” reikšminis žodelis “FIRST” reiškia, kad kiekvienas įrašas tikrinamas, kol įrašas yra teisingas su “WHEN” sąlyga.
- Struktūros “INSERT ALL” reikšminis žodelis “ALL” naudojamas specifikuojant įvedimo sakinį, kuris atitinka visas “WHEN” teisingas sąlygas.

Įterpimo į kelias lenteles privalumai:

- leidžia efektyviau įterpti duomenis į daugiau kaip 1 lentelę, naudojant SQL.

- suteikia geresnes galimybes duomenų transformavimui iš vienos sistemos į kitą, ar toje pačioje sistemoje atliekant tam tikrą jų analizę.
- optimizuojamas vykdymo greitis ir sumažinamas transakcijų skaičius duomenų šaltiniui.
- nereikalauja laikinų lentelių kūrimo, kadangi duomenų šaltinis peržiūrimas vieną kartą tam tikrai operacijai, išvengiant kelių lentelių su įvairiomis sąlygomis, išraiškomis panaudojimo.

Be to, Oracle9i yra platus spektras objektų naudojimo galimybių, kurios aptiriamos tolesniame poskyriuje.

2.7.1.2. Oracle9i DBVS objektų esminių bruožų analizė

Poskyriuje nagrinėjamos Oracle9i DBVS objektų galimybės.

Oracle9i yra įdiegtas objektinis – objektinis – reliacinis duomenų valdymo modelis, kuris leidžia naudotis plačiausiai pasaulyje paplitusiomis DBVS priemonėmis ir standartais, bei neribotai juos plėsti, naudojant DBVS saugomus objektus. Oracle DBVS objektai palaiko daugelį klasikinės objektinės orientacijos savybių:

- atributai;
- metodai;
- paveldimumas;
- inkapsuliacija;
- kompozicija;
- perdengimas.
- ir kt.

Plačiau pateikiamos procedūrų charakteristikos ir jų esminiai bruožai [5], [6], realizavimo diagrama (18 pav.), palyginimas su kitais objektais. Skirtumai tarp DB trigerių ir procedūrų pateikiami 13 lentelėje.

13 lentelė. DB trigerių ir procedūrų palyginimas

Požymis	Objektas	Trigeris	Procedūra
Apibrėžimas		angl. Create Trigger;	angl. Create Procedure;
Kodo saugojimas		angl. USER_TRIGGERS;	angl. USER_SOURCE;
Iškvietimas		Netiesiogiai;	Tiesiogiai;
Transakcijų valdymas		Nėra (angl. <i>commit</i> , <i>savepoint</i> , <i>rollback</i> negalimi);	Yra (angl. <i>commit</i> , <i>savepoint</i> , <i>rollback</i> galimi);

Iš 4 lentelės galime daryti išvadą, kad procedūrų išskirtinis bruožas lyginant su trigeriais – transakcijų valdymas, t.y., įvykdę klaidingą komandą, galime atšaukti ją, naudodami komandą „*rollback*“. Atlikus trigeriuose aprašytus veiksmus, informacijos atstatomumas praktiškai neįmanomas, išskyrus tuos atvejus, kai naudojamos duomenų audito galimybės lentelėms (saugojant buvusius prieš tam tikrą operaciją, duomenis).

Procedūrų privalumas – kiek norima reikšmių gražinimas (14 lentelė). Naudojant funkcijas, procedūras, galima naudoti įvairaus tipo įėjimo, išėjimo arba kartu įėjimo/išėjimo parametrus.

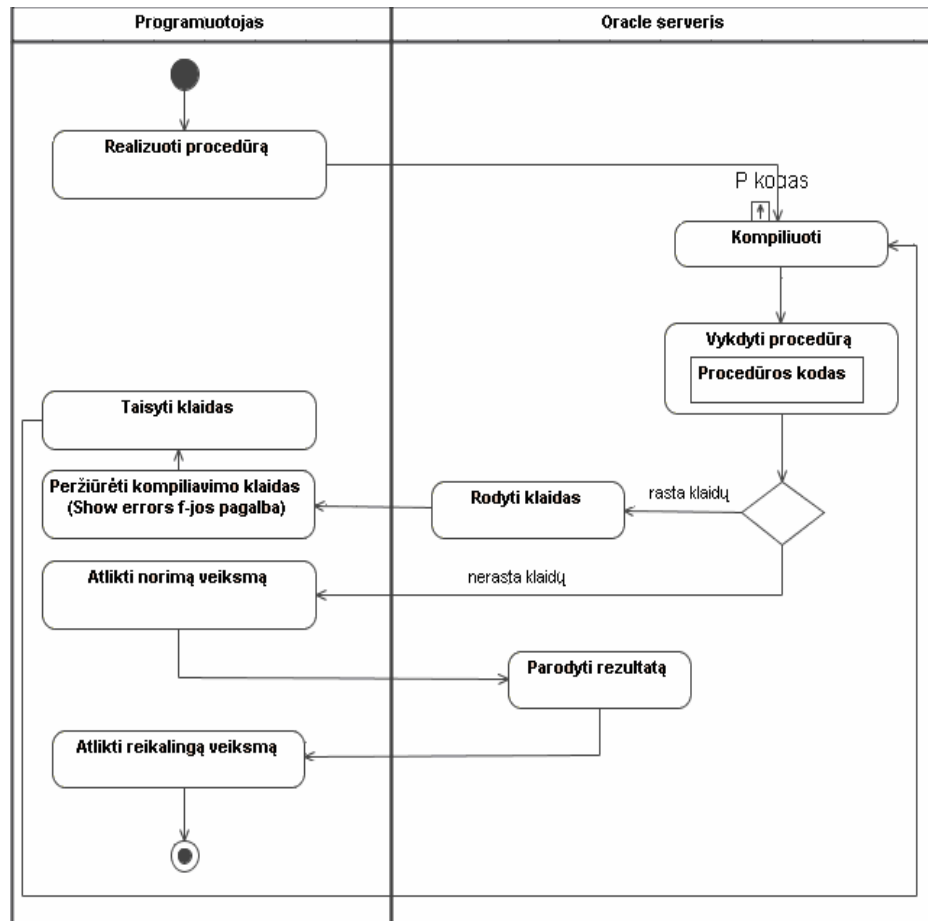
14 lentelė. Procedūrų ir funkcijų palyginimas

Požymis	Objektas	Procedūra	Funkcija
Vykdomas		Vykdoma kaip PL/SQL sakinytis;	Vykdoma kaip PL/SQL sakinytis;
Reikšmių gražinimas		Neturi gražinimo (angl. Return) sakinio, bet per išorės parametrus gali perduoti reikšmę ar keletą jų.	Privalo turėti gražinimo sakinį (angl. Return);
Gražinamų reikšmių skaičius		<ul style="list-style-type: none"> ▪ <i>vieną</i> ▪ <i>daug</i> ▪ <i>nė vienos</i> 	Privalo gražinti vieną reikšmę;

Procedūrų, funkcijų parametrai ir jų esminiai bruožai pateikiami 15 lentelėje.

15 lentelė. Procedūrų parametrų aprašymas

Įėjimo parametras (IN)	Išėjimo parametras (OUT)	Įėjimo, išėjimo parametras (IN-OUT)
Režimas pagal nutylėjimą;	Turi būti specifikuota;	Turi būti specifikuota;
Reikšmė perduodama į procedūrą;	Gražinama reikšmė į kviečiamą programinę įrangą;	Reikšmė perduodama į procedūrą ir gražinama į programinę įrangą;
Formalūs parametrai veikia kaip konstantos;	Inicializuojamas kintamasis;	Inicializuojamas kintamasis;
Parametrai gali būti: <ul style="list-style-type: none"> ▪ <i>simbolinio tipo;</i> ▪ <i>konstantos;</i> ▪ <i>tam tikra išraiška;</i> ▪ <i>inicializuotas kintamasis.</i> 	Turi būti kintamasis;	Turi būti kintamasis;



18 pav. Procedūros realizavimo veiklos diagrama

Realizuojant vientisumo apribojimus, naudojant standartines ir vartotojo sukurtas Oracle9i procedūras (Stored – Procedures) išryškėtų jų privalumai:

Pagerintas našumas

- Išvengiama kodo tikrinimo, naudojant daugeliui naudotojų paskirstytoje SQL srityje;
- Išvengiama PL/SQL kodo tikrinimo realiu laiku, nes klaidų tikrinimas atliekamas procedūros kompiliavimo metu;
- Naudojant komandas, sumažinamas duomenų srauto apimtis;
- Sumažėja tinklo apkrovimas, nes tinklu perduodamas tik kreipinys į procedūrą;
- Padidėja darbo našumas, kadangi šios procedūros yra sukompilijuotos.

Lengvas palaikymas

- Leidžia atlikti pakeitimus šablone tiesiogiai, netrukdant kitiems naudotojams;
- Vienos procedūros kodo keitimas paveikia daugelį taikomųjų programų;
- Vienos procedūros kodo keitimas leidžia išvengti analogiško testavimo;

- Suteikia galimybę vienos taikomosios programos logiką paskirstyti daugeliui klasterio serverių.

Pagerintas duomenų saugumo ir integralumo užtikrinimas

- Leidžia kontroliuoti vartotojų priėjimo prie duomenų bazės objektų granuliškumą;
- Veiklos taisyklių vieningos saugyklos organizavimo būdas, nepaskirstant jų daugelyje taikomųjų kliento programų.
- Suteikia tranzakcijų valdymo galimybes (kas neįmanoma naudojant triggerius).

Aiškus kodas

- Naudojant logiškus, aiškius ir paskirtį atitinkančius procedūrų pavadinimus, išvengiama daugelio komentarų rašymo.

2.8. Vientisumo apribojimų valdymo sistemos realizavimo architektūros parinkimas

Šioje dalyje nagrinėjamos šiuolaikinės programinės įrangos architektūros pasirinkimo strategijos. Sistemos architektūra – svarbus PĮ projektavimo metu priimamas sprendimas. Nuo to priklausys sistemos veikimas, kokios techninės ir programinės priemonės bus reikalingos realizuojant, taip pat kaip efektyviai bus panaudojami sistemos resursai.

Poskyriuje aptariamos atskirai veikiančios programos, paskirstytos sistemos, žiniatinkliui orientuotos sistemos, žiniatinklio servisai.

- ***Atskirai veikiančios programos***
 - Instaliuojamos ir veikia viena kompiuteryje;
 - Paprastai yra pagrįstos grafine vartotojo sąsaja;
- ***Paskirstytos sistemos***

Paskirstytą sistemą sudaro tam tikroje organizacijoje išbarstyti kompiuteriai, kurių kiekvienas gali savarankiškai atlikti įvairias duomenų apdorojimo operacijas, vykdyti informacinės sistemos užduotis.

- Yra įdiegiamos ir veikia keliuose kompiuteriuose;

- Naudoja kliento/serverio modelį;
- Paskirstyti komponentai naudoja bendravimo protokolus;

Paskirstytą sistemą sudaro tam tikroje organizacijoje išbarstyti kompiuteriai, kurių kiekvienas gali savarankiškai atlikti įvairias duomenų apdorojimo operacijas, vykdyti informacinės sistemos užduotis. Atskiri paskirstytų sistemų kompiuteriai dažniausiai yra arba specializuojami tam tikros klasės uždaviniams spręsti, arba yra skiriami konkrečioms darbo vietoms aptarnauti. Pavyzdžiui, kompiuteriai gali būti skiriami konstravimo darbams, finansiniams skaičiavimams, įvairiems planavimo ir apskaitos darbams. Trūkumai:

- **Sunkesnis gedimų diagnozavimas.** Paskirstytose sistemose, kuriose sprendžiami uždaviniai naudoja iš karto keletą kompiuterių ir kitokių sistemos elementų, dažnai būna sunku pasakyti, kokio sistemos elemento gedimas yra blogo sistemos darbo priežastis.
- **Duomenų nesuderinamumas.** Jeigu atskiros sistemos dalys yra projektuojamos nepriklausomai, o tokia pagunda paskirstytose sistemose būna didelė, šiose dalyse parengti duomenys gali būti nesuderinami.
- **Sudėtingas bendrų informacinių resursų valdymas.** Paskirstytose sistemose, kurios dažniausiai yra konstruojamos iš atskirų dalių, yra sunkiau patenkinti bendruosius įmonės informacinės sistemos saugumo reikalavimus.

- *Žiniatinkliui orientuotos sistemos*

Web tinklui orientuotos sistemos tinkamiausios tais atvejais, kai pagrindinis tikslas - sąlyginai nedidelio kiekio duomenų atranka pagal tam tikrus požymius bei jų peržiūra. Pateikiamos pagrindinės charakteristikos:

- Specialus paskirstytų sistemų atvejis;
- Kliento sąsaja paprastai pasiekama per Internet naršyklę;
- Bendrauja per HTTP protokolą;

- *Žiniatinklio servisas*

Web Servisas – specialiu formatu pateikiama paslauga, funkcijų rinkinys (nuotolinė RPC realizacija), apiformintas kaip savarankiškas programinis vienetas ir padėtas tinkle taip, kad juo galėtų pasinaudoti kitos programos. Ji maksimizuoja žiniatinklio efektyvų panaudojimą. Web servisas leidžia: ne tik pasinaudoti tiesiogiai funkcijomis esančiomis tinkle (pvz.: funkcija teikia vertimo iš vienos kalbos į kitą paslaugą), bet ši technologija atveria ir

naujas elektroninio verslo galimybes. Iš tinkle esančių Web servisų galime sukurti naują paslaugą. Pagrindinės savybės:

- Neturi vartotojo sąsajos;
- Bendrauja keisdami XML formato duomenimis HTTP protokolu;

Tarkime yra atskiros svetainės, kurios teikia transporto, maitinimo, apgyvendinimo paslaugas. Iš jų galima sukurti naują paslaugą – keliavimo po įvairias šalis, vystyti e-verslą.

2.8.1. Pasirinktas architektūros modelis

Išnagrinėjus visas galimas architektūros pasirinkimo galimybes, nuspręsta kurti paskirstytą sistemą, išnaudojant žiniatinklio serviso galimybes. Paskirstytą sistemą sudarytų:

- IS, kuriai kuriami vientisumo apribojimai;
- Apribojimų valdymo sistema, kurios komponentai:
 - Redakatoriaus PĮ;
 - Tikrinimo servisas (žiniatinklio servisas);
 - ETL proceso įgyvendinimas (žiniatinklio servisas).

Žiniatinklio serviso sukūrimas leis pasinaudoti tiesiogiai IS veiklos funkcijas įgyvendinančiomis procedūromis tinkle greitai, efektyviai ir dinamiškai. Bus išplėsta sukurtos PĮ integracija per internetinį tinklą, naudojant šią technologiją. Žiniatinklio technologijų įdiegimas maksimizuos tinklo panaudojimą, užtikrins centralizuotą IS vientisumo apribojimų įgyvendinimą ir pakeitimą. Taip pat leis kurti naujas sistemas bei tobulinti esamas, nepriklausomai nuo esamų sprendimų bei turimos įrangos, neįtakojant vartotojų darbo. Web Servisai veiks nepriklausomai nuo įrangos, OS. XML – bus naudojamas kaip duomenų perdavimo standartas.

2.9. Vientisumo apribojimų valdymo sistemos realizavimo platformos, kalbos parinkimas

Siekiant parinkti optimaliausią kūrimo platformą, buvo nagrinėjami variantai: .NET ir J2EE technologijos (16 lentelė). Pagrindiniai kriterijai parinkimui yra šie:

- naudojamos programavimo kalbos patogumas, naujumas, didelės galybės;
- platformos bibliotekų apimtis, įvairių DBVS schemų nuskaitymo priemonių galybės, prieinamumas bei dokumentacijos kokybė;
- numanoma realizavimo sparta, platformos taikomųjų programų kūrimo priemonių įvairovė ir kokybė;

- programos architektūros paprastumas, išplečiamumas, integracijos su kitomis PĮ galimybės;
- mobilumas;
- suderinamumas.

16 lentelė. Technologijų palyginimas

Kriterijus	Oracle9i ir J2EE	Oracle9i ir .NET
Kūrimo platformos paprastumas	-	+
Patirtis, žinios	-	+
Priemonės darbui su Oracle DBVS	+	+
Įvairių DBVS schemų nuskaitymo technologijos palaikymas	-	+
Mobilumas	+	-
Suderinamumas	+	+
Integracija su kitomis PĮ	-	+
Didesnė platformos bibliotekų apimtis	-	+
Programavimo kalbos paprastumas, naujumas	-	+

Oracle9i pasižymi plačiu galimybių spektru, todėl idealios sąlygos vientisumo apribojimų automatizavimui kompiuterizuoti, naudojant .NET technologijas [11], [12], [13], [14], [15], [16], [17], [18]. Pateikiamos esminės savybės, kurios paskatino priimti šį sprendimą:

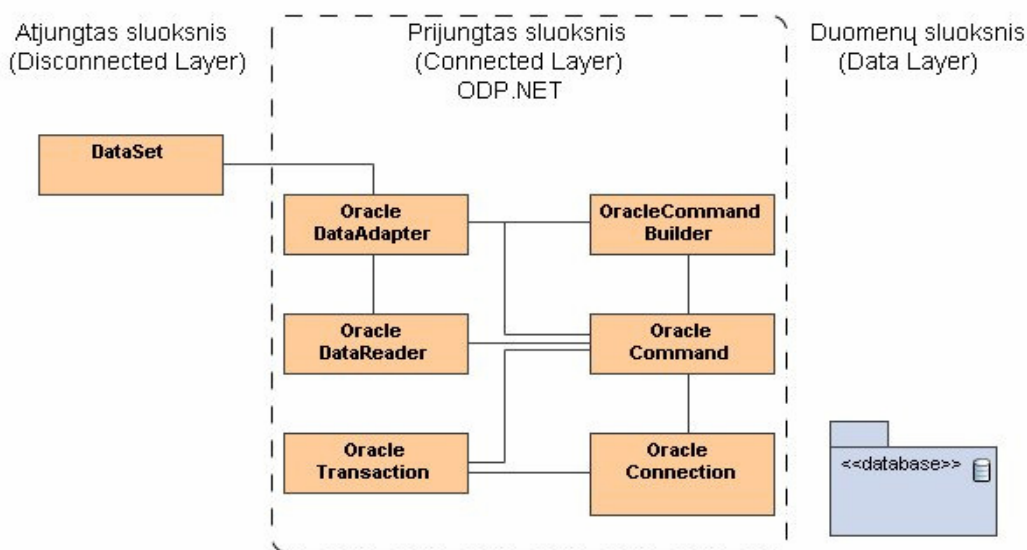
- **Oracle yra atviras**
 - DB serverį palaiko Windows, Linux, Unix operacinės sistemos;
 - sistemų kūrėjai gali naudoti .NET, neprarasdami DB teikiamų galimybių.

- **.NET privalumai**
 - didelis programavimo kalbų pasirinkimas;
 - milžiniška klasių biblioteka;
 - didelis dėmesys saugumui;
 - žiniatinklio paslaugos;
 - ODP.NET priėjimas prie duomenų;
 - vieninga darbo aplinka;
 - kodo derinimas;
 - įvairių redaktorių naudojimo galimybės (kodo, formų, duomenų schemų).

- **.NET technologijos kartu su Oracle9i duoda šiuos privalumus:**
 - patikimumą;

- prieinamumą;
 - galimybes naudoti visas išskirtines DB savybes per Oracle duomenų teikėją (ODP.NET).
- **ODP.NET**
- gali būti naudojamas su Oracle8, Oracle8i, Oracle9i, Oracle10g duomenų bazių serveriais;
 - palaiko VS.NET 2002, 2003, 2005;
 - palaiko .NET Framework 1.0, 1.1, 2.0;
 - pilnos PL/SQL naudojimo galimybės;
 - palaiko Oracle duomenų tipus;
 - dinaminis prisijungimo kintamųjų valdymas;
 - transakcijų valdymas;
 - pilnas Unicode kodavimo palaikymas;
 - apsaugos (Windows, proxy serverio ir SSL) autentifikavimo palaikymas;
 - duomenų išgavimas ir modifikavimas (įrašų kiekio valdymas, skirtingų užklausų vykdymas, tos pačios komandos vykdymas keletą kartų);
 - greitis;
 - patikimumas;

Pateikiamas ODP.NET objektų modelis (19 pav.).



19 pav. ODP.NET objektų modelis

Įvairių DBVS schemų nuskaitymui, palaikoma galimybė naudoti OLE DB duomenų tiekėją.

Kadangi apribojimų IS realizavimui pasirinkta .NET kūrimo platforma, analizuojamos kalbų pasirinkimo alternatyvos, esminės savybės, sėkmingam sprendimo priėmimui.

17 lentelė. .NET kalbų palyginimas

Savybės	C#	VB	C/C++
Objektinė programavimo kalba	+	+	+
Sintaksinių struktūrų lankstumas	+	-	+
Programavimo kalbos patogumas	+	+	+
Didelės kalbos galimybės	+	-	-

Atsižvelgiant į didesnę galimybių spektrą (17 lentelė), pasirinkta C# programavimo kalba.

2.10. Projekto tikslai

Sukurti lanksčią ir adaptyvią vientisumo apribojimus įgyvendinančią IS, kur apribojimai, kaip žinios būtų saugojamos centralizuotai ir struktūrizuotai, atskirai nuo IS duomenų, vartotojo norimoje DBVS. Autonomiškos saugyklos įgyvendinimo dėka, suteikti didesnes galimybes įvairių IS integravimo procesams įgyvendinti, apribojimų sąlygas neišbarstyti programiniame kode, išvengti saugojimo failų sistemose, lengviau atlikti pakeitimus. Žiniatinklio technologijų dėka, apribojimus išgauti greitai, efektyviai, atlikti apribojimų dinaminį tikrinimą, SQL sakinių atitikimą apribojimams.

2.11. Vientisumo apribojimų valdymo sistemos kokybės kriterijai

Pasirinkti sistemos kokybės parametrai pateikiami 18 lentelėje.

18 lentelė. Kuriamos sistemos kokybės parametrai

Savybės	Kokybės savybių aprašymas
Korektiškumas	PĮ atitinka savo funkcinę specifikaciją.
Pilnumas	Realizuoti visi numatyti vientisumo apribojimai.
Patikimumas	Sukurta PĮ atlieka visas specifikacijoje numatytas funkcijas ir neatlieka nenumatytų funkcijų pasikeitus aplinkos veiksniais. (Įdiegus/pašalinus kitas sistemas)

Savybės	Kokybės savybių aprašymas
Integralumas	PĮ užtikrina duomenų teisingumą vartotojui atlikus neleistinus veiksmus.
Panaudojamumas	<ul style="list-style-type: none"> ▪ Sistema yra intuityviai suvokiama. ▪ Lengva ir paprasta išmokyti dirbti su ja. (Paprastumas leis vartotojams greičiau įsisavinti sistemą, pajusti jos galimybes, teikiamus privalumus)
Sąveikos su kitomis sistemomis galimybės	Lengva sistemą integruoti su kitomis PĮ.
Išbaigtumas	PĮ įgyvendina visus iškeltus tikslus.
Paprasta, logiška ir reikalavimus įgyvendinanti vartotojo sąsaja redaktoriui	Paprasta, aiški ir funkcionali vartotojo sąsaja leis vartotojui įgyvendinti norimą tikslą – įvesti, redaguoti duomenų bazės vientisumo apribojimus.
Aukštas sistemos naudojimo lygmuo.	Daug sistemos vartotojų.
Palankus požiūris į IS turimas galimybes	Matoma reali įrankio ir gaunamo serviso nauda apribojimų valdymo, verslo procesuose.

2.12. Sistemos įgyvendinimui galimų rizikos faktorių analizė

Identifikuoti būsimo projekto rizikos faktoriai ir jų eliminavimo būdai pateikiami 19 lentelėje.

19 lentelė. Rizikos faktoriai

Rizikos faktorius	Rizikos faktoriaus eliminavimo būdas
Vientisumo apribojimų sudėtingas sąlygų identifikavimas ir aprašymas, realizavimas ar modifikavimas.	<ol style="list-style-type: none"> 1) Nerealizuoti. 2) Ieškoti kitų būdų. 3) Keisti aprašymo formą.
Grafinė vartotojo sąsaja neatitinka reikalavimų, yra per sudėtinga vartotojui.	Modeliuoti kelis variantus galimos grafinės vartotojo sąsajos prototipus.
Programinė įranga per lėtai veikia ar reaguoja į vartotojo veiksmus	<ol style="list-style-type: none"> 1) Numatyti kritines programos dalis; 2) Numatyti kritines funkcijas, užklausas; 3) Numatyti sistemos našumo problemas; <p>Optimizuoti funkcijas, užklausas, ar dalį net pašalinti. Vengti ar sumažinti kritinių sistemos situacijų skaičių. Testuoti kiekvieną modulį, įvairiose kūrimo stadijose.</p>
Pastebėta daug sistemos išplėtimo galimybių.	Suskirstyti projektą į modulius. Realizuoti svarbiausius.
Viršytas programos kūrimo laikas	Numatyti savo tikslui priimtina mažesnę sistemos variantą, atsisakyti tam tikrų funkcijų realizavimo.

2.13. Projektavimo aplinka, metodai ir priemonės

Programinės įrangos įgyvendinimui naudojamas RUP proceso modelis. Modelio privalumas – iteratyvus, architektūra grindžiamas, panaudojimo atvejų valdomas PĮ kūrimas. Sistemos įgyvendinimui, naudojamos RUP pagrindinės veiklos:

- verslo modeliavimas;
- reikalavimų apibrėžimas;
- analizė ir projektavimas;
- realizacija;
- testavimas;
- įdiegimas.

Toliau minimos savybės lėmusios šio projektavimo proceso pasirinkimą:

- veiklos modeliavimo galimybės, reikalavimų kuriamai IS suradimui, veiklos procesams apibrėžti;
- veiklos, klasių stereotipų palaikymas.

Visiems IS kūrimo proceso etapų sprendimams priimti, naudojama projektavimo kalba – UML. Vaizdinio modeliavimo privalumai, palyginti su formaliomis specifikacijomis (kai pritaikomos specialiai tam skirtos matematinės formulės ar specializuota kalba, pavyzdžiui PDL, OCL), programos tekstu (kuria nors programavimo kalba, pavyzdžiui C++), arba aprašymu natūralia kalba (pvz., lietuvių), yra šie:

- lengviau suprasti sudėtingas sistemas;
- paprasčiau suformuoti loginį užduoties sprendimą;
- lengviau aptikti ribines programos darbo sąlygas ir “siaurąsias vietas”;
- išvengiama dviprasmiško reikalavimų aprašymo.

Pasirinktas įrankis – Magic Draw (10.0 versija) [29]. Tai įvairiapusis, patogus, galingas CASE įrankis. Magic Draw 10.0 privalumas – diagramas galima naudoti visose programinės įrangos kūrimo ciklo fazėse. Pradedant nuo reikalavimų apibrėžimo, UML gali būti naudojama specifikuojant, vizualizuojant ir dokumentuojant tuos reikalavimus įgyvendinančios programinės įrangos modelį, jo struktūrą, naudojant grafines diagramas. Palaiko Rational korporacijos unifikuoto kūrimo procesą – RUP.

Vartotojo sąsajos projektui realizuoti pasirinktas *Microsoft Visio 2003* CASE įrankis. Tai geras įrankis Windows formų efektyviam ir vaizdžiam projektavimui atlikti.

2.14. Analizės išvados

Iš visų nagrinėtų egzistuojančių sistemų, nė viena neatitinka visų pasirinktų kriterijų. Heterogeniškas, nesisteminis vientisumo apribojimų įgyvendinimas nagrinėtais metodais, iššaukia:

- kodo dubliavimą;
- ilgą kūrimo laiką;
- didesnę klaidų atsiradimo tikimybę;
- sudėtingą apribojimo pakeitimo vykdymą;
- struktūrizacijos trūkumą;
- apribojimo taikymo verslo procesuose negalimumą;
- papildomas išlaidas samdant IT specialistus pakeitimų realizavimui;
- integracijos su kitomis sistemomis problemas.

Siekiant numatyto tikslo, – kompiuterizuoto vientisumo apribojimų valdymo proceso, – reikalingas struktūrizuotas metodas, siekiamos sistemos įgyvendinimui.

SQL įrankiai yra skirti DBVS specialistams, nes reikalinga:

- mokėti SQL užklausų kalbą;
- išmanyti pagrindinius DB kūrimo ir palaikymo principus, taisykles;
- norint išgauti vientisumo apribojimus, kuriuos saugo DBVS, reikia būti susipažinus su tos DBVS galimybėmis: ar turi žodynų (angl. data dictionary) lenteles ar atvaizdus (angl. views), sisteminės lenteles;

Duomenų žodynuose saugojama informacija apie DB struktūras ir objektus: lenteles, laukus, vartotojus, duomenų failus ir kt. Ši informacija gali būti pateikiama vartotojui tik skaitymo režime, be modifikavimo ar šalinimo galimybių, DBA suteikiant papildomas teises.

- apribojimus kuriamai IS dažniausiai pateikia dalykinės srities specialistai, kurie neturi DBVS žinių, kaip realizuoti. Apribojimus pateikia sava kalba. Norėdami pasinaudoti sistema, reikia juos apmokinti.

Įrankiuose nerasta vientisumo apribojimų patogaus peržiūros, kūrimo ir modifikavimo redaktoriaus, skirto greitai ir paprastai peržiūrėti sukurtus apribojimus tam tikrai IS.

Apribojimų DB sukūrimo analizė leido priimti tinkamą sprendimą:

- atskirti apribojimus nuo IS, juos struktūrizuoti ir centralizuotai saugoti atskiroje DBVS.
- apribojimus išskaidyti iki elementų (laukas, reikšmė, operacija), panaudoti predikatus, logines ir aritmetines operacijas.

Apribojimų DB komponentų analizė parodė, kad:

- tikslinga sukurti universalų mechanizmą įvairių DBVS schemos objektų (lentelės, laukai) nuskaitymui ir pateikimui patogiai vartotojui forma (medžio struktūra).
- įvairių IS parametrus saugoti meta DB.
- Apriboti vartotojo įvedimo ir modifikavimo galimybes, išvengiant klaidų (nurodyti, kur dėti skliaustus, kaip pasirinkti DB lentelės lauką, operacijas, nurodyti ryšius).
- vartotojo dokumentacijos pateikimui, geriausi naudoti naršyklę.

Sistemos projektui įgyvendinti pasirinktas Magic Draw 10.0 projektavimo įrankis, UML modeliavimo kalba, RUP – veiklos panaudojimo atvejams identifikuoti, sistemos sąveikoms, procesams grafiškai atvaizduoti.

Sistemos realumui, efektyvumui, naudingumui ir teisingumui įrodyti bus atliktas eksperimentas – sukurta vientisumo apribojimų valdymo sistema, naudojant .NET platformą, C# kalbą. Ši sistema leis sėkmingai kurti ir tvarkyti apribojimus patogioje grafinėje aplinkoje. Apribojimai bus saugojami struktūrizuoti ir centralizuotai Oracle9i DBVS, suteiktos galimybės per žiniatinklį išgauti jų turinį, atlikti dinaminį teisingumo ir DML sakinių tikrinimo funkcijas. Taip pat apribojimai bus panaudoti įgyvendinant Oracle ETL procesą, naudojant SQL.

3. CENTRALIZUOTO DUOMENŲ VIENTISUMO APRIBOJIMŲ VALDYMO KONCEPCINIS MODELIS

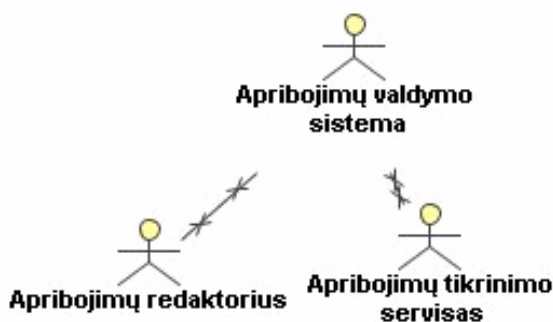
Siame skyriuje suformuoti funkciniai ir nefunkciniai vientisumo apribojimų redaktoriaus ir tikrinimo serviso įgyvendinimui keliami reikalavimai ir jų specifikacijos, naudojant UML panaudojimo atvejų diagramas. Sistemos sąveikoms su vartotoju – apribojimų pateikimo specialistu atvaizduoti, sumodeliuotos detalios sekų diagramos. Centralizuoto vientisumo apribojimų valdymo sistemos įgyvendinimui, pateikiamas duomenų klasių modelis – leidžiantis išskirti apribojimus iki elementų (laukas, reikšmė, operacija). Skyriuje aprašomas sistemos projektas: fizinė ir loginė architektūra, reikalavimai jos veikimo aplinkai, techninei ir programinei įrangai, vartotojo, veiklos, duomenų paslaugų, sistemos elgsenos, realizacijos (programinių komponentų, diegimo, DB) UML modeliai, vartotojo sąsajos reikalavimai ir išsami sistemos naudojimo instrukcija.

3.1. Vientisumo apribojimų valdymo sistemos reikalavimų modelis

Reikalavimo modelyje detalai pateikiami vientisumo apribojimų valdymo sistemai keliami funkciniai ir vartotojo sąsajos reikalavimai, jų specifikacijos, nefunkciniai reikalavimai veikimui, duomenims, techninei įrangai, detalizuojamas apribojimų meta DB duomenų modelis.

3.1.1. Vientisumo apribojimų valdymo sistemai keliami funkciniai reikalavimai

Poskyriuje pateikiami vientisumo apribojimų valdymo sistemos struktūros modelis (20 pav.) ir kiekvienai posistemei keliami funkciniai reikalavimai.



20 pav. Apribojimų valdymo sistemos posistemiai

IS apribojimų redaktoriui keliami šie funkciniai reikalavimai:

- įvesti, peržiūrėti, modifikuoti, šalinti ir saugoti klientų IS parametrus;

- vartotojui leisti testuoti prijungtą prie redaktoriaus klientų, IS ryšį;
- nuskaityti kliento IS schemos objektus – lenteles ir jų laukus;
- saugoti, šalinti prisijungimo taškus, t.y. lenteles ir jų laukus meta-DB;
- suteikti galimybes kurti, modifikuoti, išrinkti, šalinti apribojimus patogioje grafinėje formoje;
- sieti, šalinti prisijungimo taškus ir jų laukus su apribojimais;
- pateikti ir leisti įvesti, redaguoti sistemoje esančius predikatus;
- kurti, šalinti apribojimų IS redaktoriaus vartotojus;
- kurti, šalinti vartotojus ir jų priėjimą, teises prie apribojimų meta-DB.
- redaktorius turi automatiškai užtikrinti, kad duomenys esantys tarp apribojimo sąlygas aprašančių lentelių ir lentelės, kuri saugo visą apribojimą turi būti tapatūs.

IS apribojimų tikrinimo Web servisui keliami šie funkciniai reikalavimai:

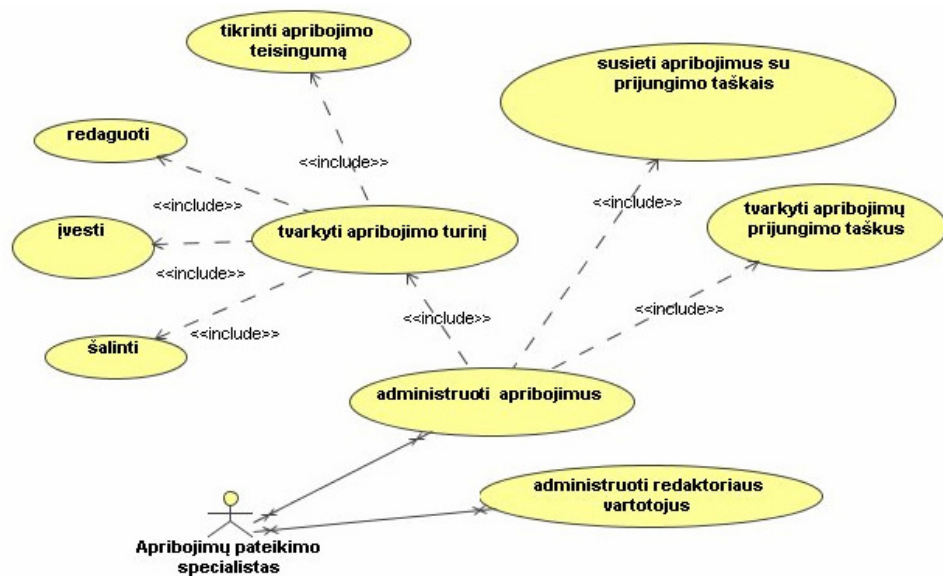
- testuoti ryšį su klientine IS.
- tikrinti SQL DML funkcijų teisingumą ir pateikti kliento IS ar paduota užklausa yra teisinga;
- gražinti norimo apribojimų turinį;
- gražinti pranešimus apie tam tikras operacijas, ar klaidas;
- vykdyti veiklos procedūras ir gauti rezultatus.

3.1.2. Vientisumo apribojimų valdymo sistemos reikalavimų specifikacija

Reikalavimų panaudojimo atvejų modelis suformuotas iš veiklos panaudojimo atvejų modelio (15, 16, 17 pav.) atrenkant panaudojimo atvejus, kuriuos tikslinga kompiuterizuoti. Pateikiami panaudojimo atvejų modeliai išskaidyti į „Apribojimų redaktoriaus“ (3.1.2.1.) ir „Apribojimų tikrinimo ir išgavimo serviso“ (3.1.2.2.) kompiuterizuojamus panaudojimo atvejus.

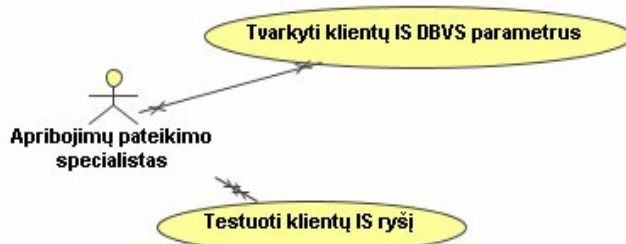
3.1.2.1. Vientisumo apribojimų redaktoriaus reikalavimų specifikacija

Sudarant redaktoriaus reikalavimų specifikaciją buvo peržiūrėti ir detalizuoti panaudos atvejai. Pateikiama apibendrinta apribojimų pateikimo specialisto panaudojimo atvejų diagrama (21 pav.).



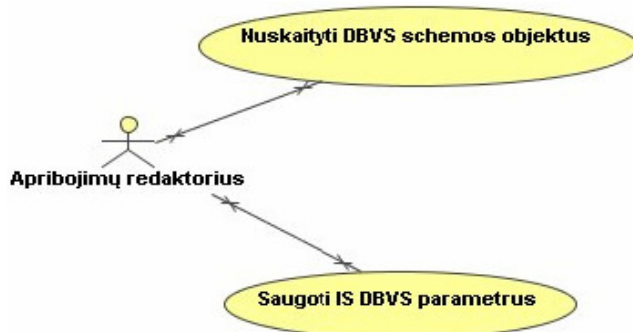
21 pav. Apribojimų pateikimo specialisto funkcijų panaudojimo atvejų diagrama

Kadangi kuriamas universalus apribojimų redaktorius, sistemoje bus saugojami IS DBVS parametrai, kuriuos tvarkys apribojimų pateikimo specialistas (22 pav.).



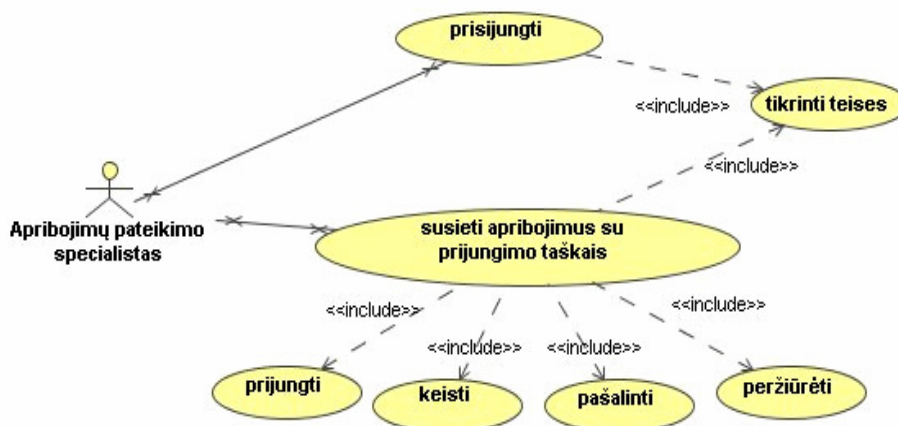
22 pav. Apribojimų pateikimo specialisto papildomų funkcijų panaudojimo atvejų diagrama

Apribojimų redaktoriaus programinė įranga turi galėti nuskaityti DBVS schemas objektus, t.y. lenteles ir jų laukus, saugoti, redaguoti, šalinti sistemų parametrus. Pateikiama (23 pav.) panaudojimo atvejų diagrama.



23 pav. Apribojimų redaktoriaus funkcijų panaudos atvejų diagrama

Toliau pateikiamos detalios kompiuterizuojamų panaudojimų atvejų diagramos (24 pav.), jų sekų diagramos kartu su detaliais pagrindinių scenarijų aprašymais.



24 pav. Apribojimų susiejimo panaudojimo atvejų diagrama

Panaudojimo atvejo „Susieti (prijungti) apribojimus su prijungimo taškais“ pagrindinis scenarijus (20 lentelė) .

20 lentelė. Reikalavimo „Susieti apribojimus su prisijungimo taškais“ specifikacija

Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises.
Įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas prisijungia prie apribojimų redaktoriaus (šis žingsnis apima panaudojimo atvejį „prisijungti“).	1.1. Sistema tikrina ar vartotojas turi apribojimų pateikimo specialisto teises (panaudojimo atvejis „tikrinti teises“). Jei vartotojas neturi tokių teisių, sistema baigia darbą.
2. Vartotojas pasirenka prisijungimo tašką (db lentelę, jos laukus).	2.1. Jei tokio prisijungimo taško nėra, sistema baigia darbą.
3. Vartotojas pasirenka apribojimo pavadinimą.	
4. Pasirenka operaciją „prijungti“.	4.1. Sistema išsaugo apribojimo prijungimo duomenis.
Po sąlyga	Duomenų bazėje išsaugoti apribojimo prijungimo prie db lenteles ar jos lauko duomenys su juos redagavusio vartotojo identifikatoriumi.

Panaudojimo atvejo „Susieti (redaguoti) apribojimus su prijungimo taškais“ pagrindinis scenarijus (21 lentelė) .

21 lentelė. Reikalavimo „Redaguoti apribojimo prisijungimą“ specifikacija

Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises. Sistemoje yra įvestų apribojimų ir susietų su prijungimo taškais (lentele ir jos laukais).
Įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas prisijungia prie apribojimų redaktoriaus (šis žingsnis apima panaudojimo	1.1. Sistema tikrina ar vartotojas turi apribojimų pateikimo specialisto teises (panaudojimo atvejis

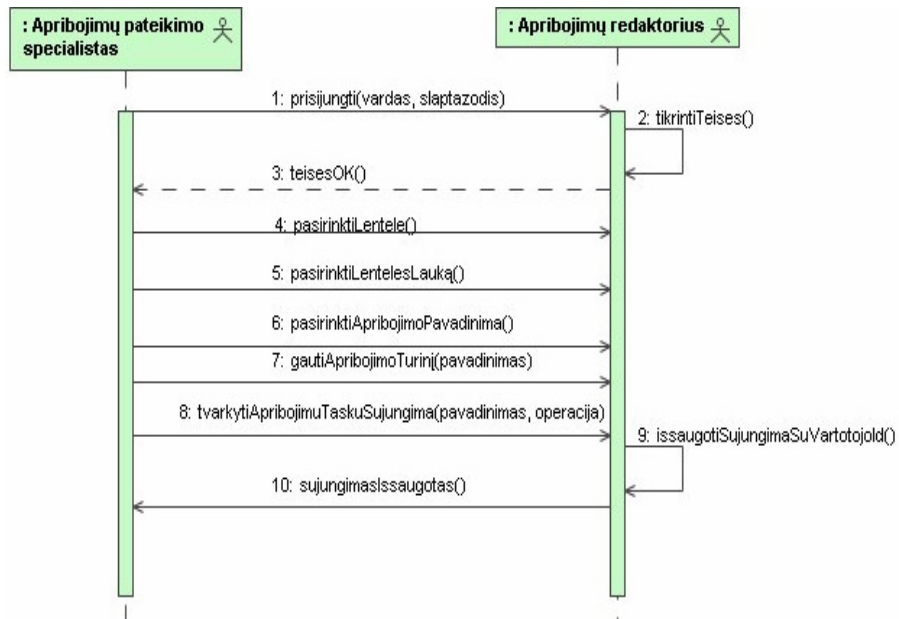
Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises. Sistemoje yra įvestų apribojimų ir susietų su prijungimo taškais (lentelė ir jos laukais).
atvejį „prisijungti“).	„tikrinti teises“). Jei vartotojas neturi tokių teisių, sistema baigia darbą.
2. Vartotojas pasirenka prisijungimo tašką (db lentelę, jos laukus)	2.1. Jei tokio prisijungimo taško nėra, sistema baigia darbą. 2.2. Sistema pateikia sąrašą apribojimų tai lentelei ar laukui.
3. Vartotojas pasirenka norimo apribojimo pavadinimą	
4. Pasirenka operaciją „redaguoti“.	4.1. Sistema leidžia redaguoti apribojimo prijungimo duomenis.
Po sąlyga	Duomenų bazėje išsaugoti apribojimo prijungimo pakeitimai su juos įvedusio vartotojo identifikatoriumi.

Panaudojimo atvejo „Susieti (pašalinti) apribojimus su prijungimo taškais“ pagrindinis scenarijus (22 lentelė).

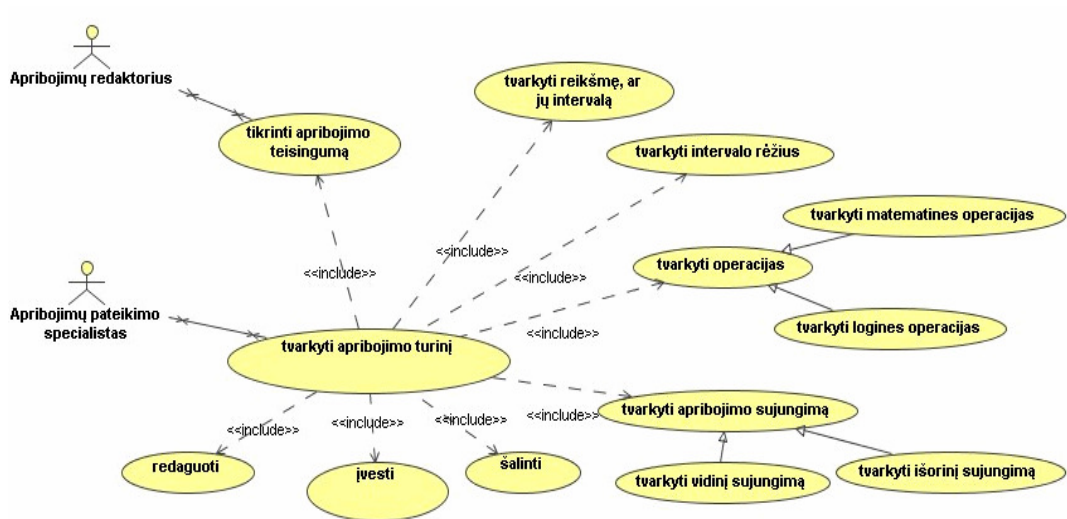
22 lentelė. Reikalavimo „Šalinti prisijungimo taško apribojimą“ specifikacija

Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises. Sistemoje yra įvestų apribojimų ir susietų su prijungimo taškais (lentelė ir jos laukais).
Įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas prisijungia prie apribojimų redaktoriaus (šis žingsnis apima panaudojimo atvejį „prisijungti“).	1.1. Sistema tikrina ar vartotojas turi apribojimų pateikimo specialisto teises (panaudojimo atvejis „tikrinti teises“). Jei vartotojas neturi tokių teisių, sistema baigia darbą.
2. Vartotojas pasirenka prisijungimo tašką (db lentelę, jos laukus)	2.1. Jei tokio prisijungimo taško nėra, sistema baigia darbą.
3. Vartotojas pasirenka apribojimo pavadinimą.	
4. Vartotojas pasirenka lenteles laukus.	
5. Pasirenka operaciją „pašalinti“.	5.1. Sistema pašalina ryšį tarp apribojimo ir prijungimo taško.
Po sąlyga	Duomenų bazėje pašalinti apribojimo prijungimo prie db lenteles ar jos lauko duomenys su juos pašalinusio vartotojo identifikatoriumi.

Sekų diagramomis (25, 27, 28, 30 pav.) atvaizduojamos apribojimų pateikimo specialisto sąveikos su apribojimų redaktoriumi.



25 pav. „Susieti apribojimą su prisijungimo taškais“ sekų diagrama



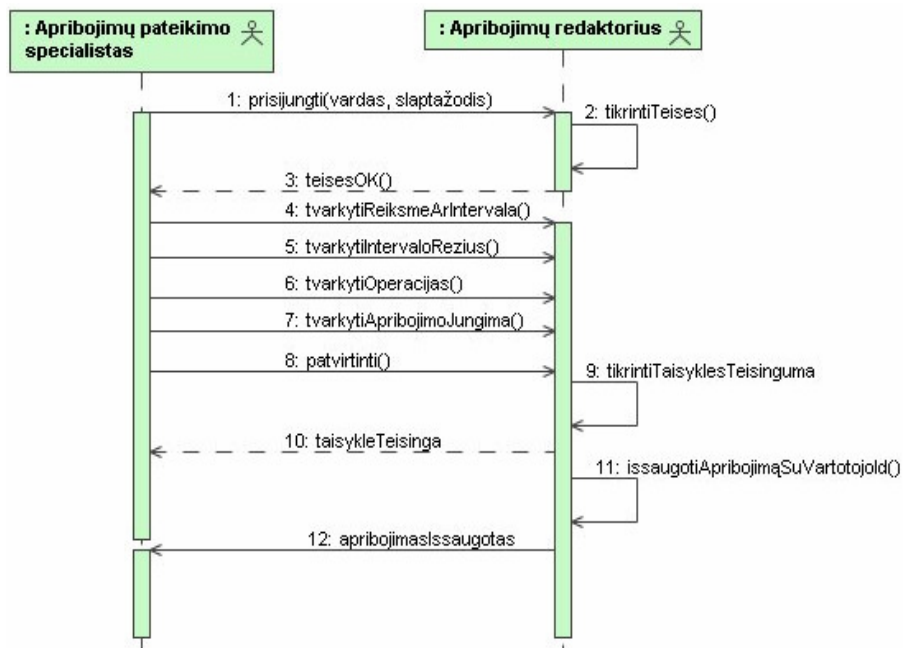
26 pav. Apribojimų turinio tvarkymo panaudojimo atvejų diagrama

Panaudojimo atvejo „Tvarkyti apribojimo turinį“ pagrindinis scenarijus (23 lentelė) .

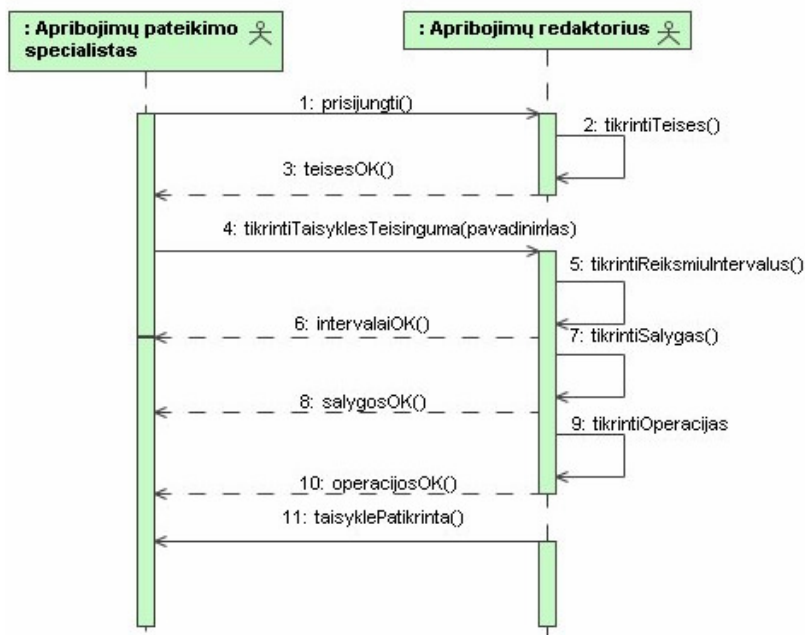
23 lentelė. Reikalavimo „Tvarkyti apribojimo turinį“ specifikacija

Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises.
Įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas prisijungia prie apribojimų redaktoriaus (šis žingsnis apima panaudojimo atvejį „prisijungti“).	1.1. Sistema tikrina ar vartotojas turi apribojimų pateikimo specialisto teises (panaudojimo atvejis „tikrinti teises“). Jei vartotojas neturi tokių teisių, sistema baigia darbą.

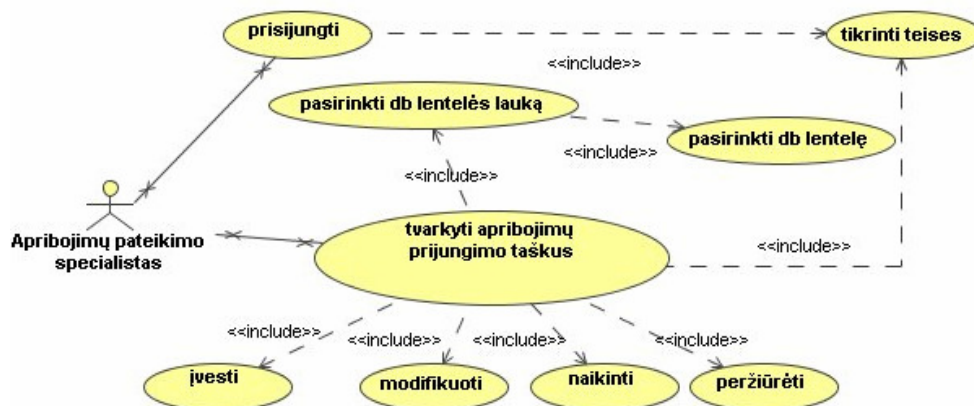
Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises.
2. Vartotojas įveda reikšmes ar jų intervalus.	
3. Vartotojas pasirenka reikšmių intervalo režius (uždaras, atviras)	
4. Vartotojas pasirenka operacijas (matematinės, logines).	
5. Vartotojas pasirenka apribojimo sujungimą (vidinis, išorinis).	
6. Patvirtina	6.1. Sistema tikrina taisykles teisingumą ir logiškumą. Jei neteisinga, sistema informuoja vartotoją išsamiu klaidos pranešimu. Priešingu atveju atlieka žingsnį 6. 6.2. Sistema išsaugo apribojimo turinio duomenis.
Po sąlyga	Duomenų bazėje išsaugoti apribojimo turinys kartu su jį įvedusio vartotojo identifikatoriumi.



27 pav. „Tvarkyti apribojimo turinį“ sekų diagrama



28 pav. „Tikrinti taisykles teisingumą“ sekų diagrama



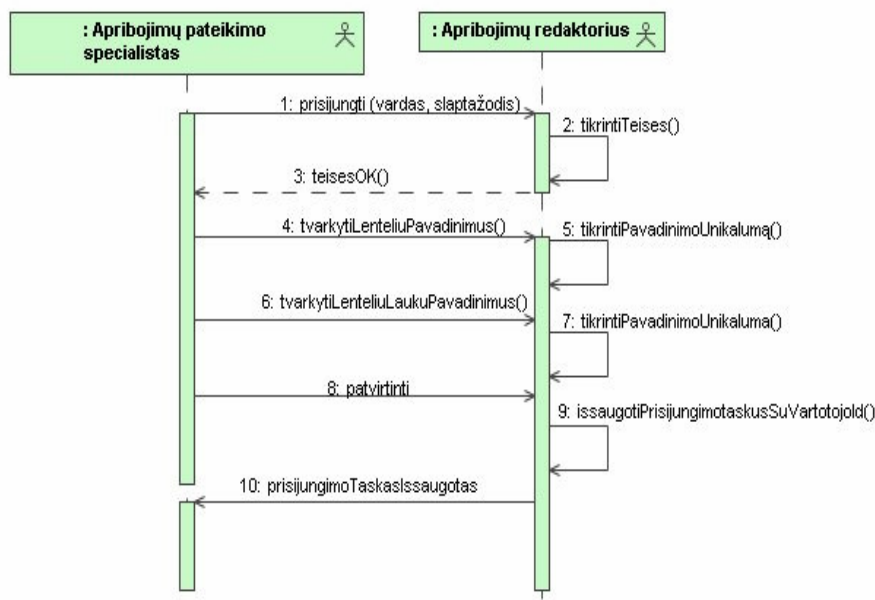
29 pav. Apribojimų tvarkymo panaudojimo atvejų diagrama

Panaudojimo atvejo „Tvarkyti apribojimų prijungimo taškus“ pagrindinis scenarijus (24 lentelė).

24 lentelė. Reikalavimo „Tvarkyti apribojimų prisijungimo taškus“ specifikacija

Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises.
Įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas prisijungia prie apribojimų redaktoriaus (šis žingsnis apima panaudojimo atvejį „prisijungti“).	1.1. Sistema tikrina ar vartotojas turi apribojimų pateikimo specialisto teises (panaudojimo atvejis „tikrinti teises“). Jei vartotojas neturi tokių teisių, sistema baigia darbą.
2. Vartotojas tvarko (įveda, tvarko, redaguoja, šalina) kliento IS duomenų bazės lenteles,	2.1. Sistema tikrina lentelių pavadinimų unikalumą. Jei pavadinimai vienodi – pranešti

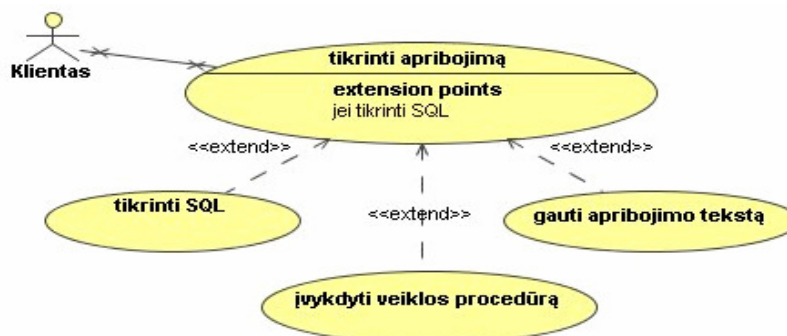
Prieš sąlyga	Vartotojas turi apribojimų pateikimo specialisto teises.
naudodamasis redaktoriumi.	virtotojui išsamiau klaidos pranešimu. Sistema pereina į 3 žingsnį.
3. Vartotojas tvarko (įveda, tvarko, redaguoja, šalina) kliento IS duomenų bazės lenteles)	3.1. Sistema tikrina lentelių laukų unikalumą. Jei pavadinimai vienodi – pranešti vartotojui išsamiau klaidos pranešimu. Priešingu atveju pereina į 4 žingsnį.
4. Patvirtinti	4.1. Sistema išsaugo duomenų bazėje kliento IS lentelių pavadinimus, jų laukus.
Po sąlyga	Duomenų bazėje išsaugotas apribojimo turinys kartu su jį įvedusio vartotojo identifikatoriumi.



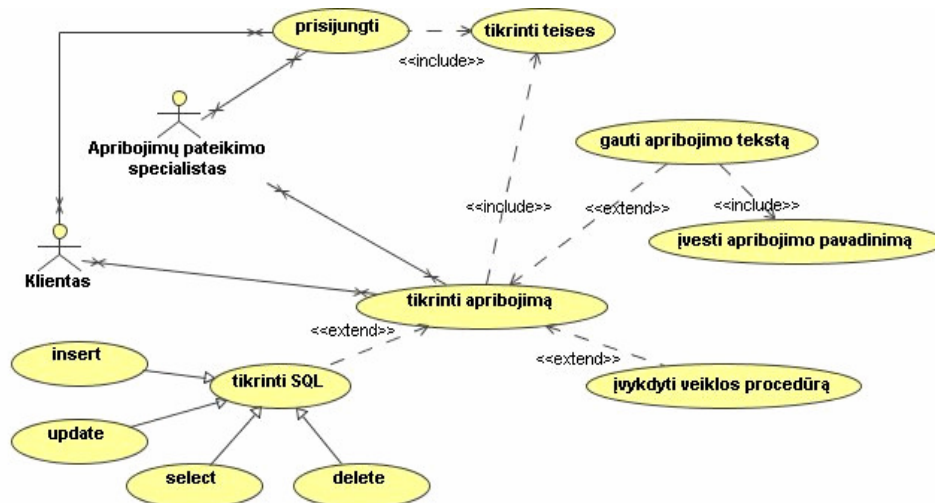
30 pav. „Prisijungimo taško tvarkymo“ sekų diagrama

3.1.2.2. Apribojimų tikrinimo ir išgavimo serviso reikalavimų specifikacija

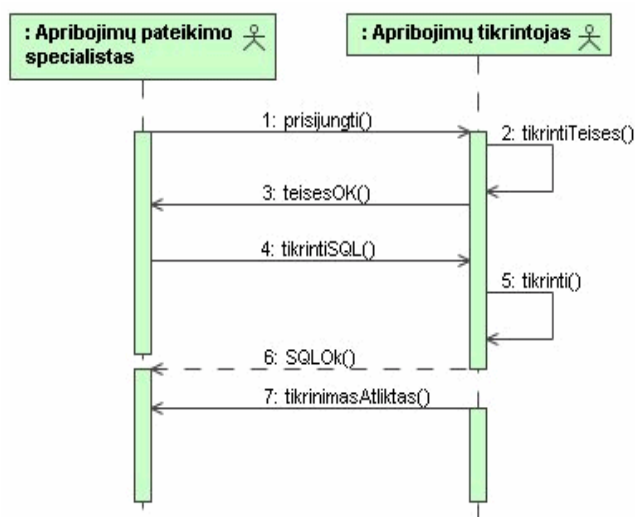
Apribojimų tikrinimo serviso panaudojimo atvejų diagrama (30, 31 pav.).



31 pav. Kliento atliekamų funkcijų panaudojimo atvejų diagrama



32 pav. Apribojimų tikrinimo panaudojimo atvejų diagrama

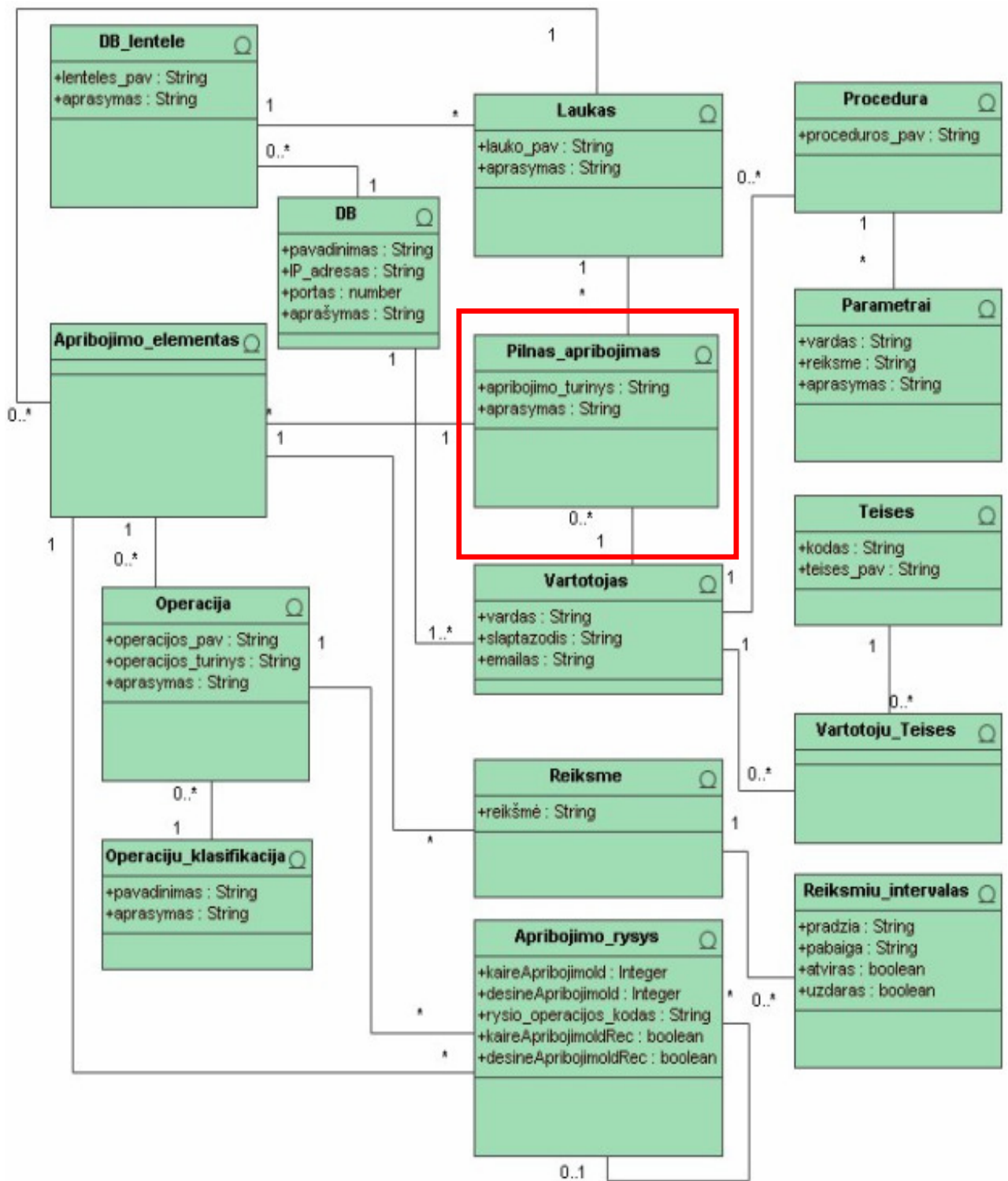


33 pav. „Tikrinti SQL“ sekų diagrama

3.1.3. Apribojimų meta DB klasių modelis

Modelio (34 pav.) esmė – saugoti IS vientisumo apribojimus loginėmis išraiškomis, DBVS lenteles, jų laukus, operacijas, reikšmes. Taip pat fiksuoti informaciją kokie ryšiai tarp apribojimų, jei apribojimas yra sudėtinis, kokie galimi reikšmių intervalai, jų sujungimo operacijos, kurios procedūros ir jų parametrai realizuoja jas pagal vartotojų teises.

Meta DB klasių modelyje apribojimai yra išskaidomi į elementus: laukas, operacija ir reikšmė. Apibrėžiamos saugomos operacijos, predikatai ir jų aprašymai: matematinės (>, <, >=, <=, =, <>) ir loginės (or, and, xor). Intervalai – uždari [] ar atviri (). Taip pat ar laukams galimos „null“, „not null“ reikšmės ir kitos SQL funkcijos (angl. like, between ir kt.).



34 pav. Meta-DB klasių modelis

Efektyvesniam ir greitesniam sistemos veikimui realizuojama papildoma klasė (*apvesta modelyje raudonai*), kurioje bus saugojama pilna, logiška ir teisinga tam tikro apribojimo sąlyga, suformuota tam tikros operacijos (duomenų įterpimo, atnaujinimo, trynimo) atlikimui patikrinti. Ši klasė naudojama kaip veiklos apribojimų greitinančioji (angl. *cache*) esybė. Operacijų vykdymui naudojama struktūrizuota užklausa kalba (SQL).

Esybės įgyvendinimui keliamas reikalavimas, kad kliento posistemė turi automatiškai užtikrinti, kad duomenys esantys tarp apribojimo sąlygas aprašančių lentelių ir lentelės, kuri saugo visą apribojimą turi būti teisingi ir logiški, redaktoriuje atlikus tam tikrus pakeitimus ar įvedus papildomas sąlygas (angl. „*synchronization between sources*“).

Modelio realizavimo privalumai:

- duomenys saugojami struktūrizuoti;
- struktūrizacija įgalina efektyvų redagavimą įprastais vartotojų įrankiais;
- paprastesnis vientisumo apribojimų administravimas, palaikymas;
- modelis atitinka reliacinių duomenų bazių teoriją.

Trūkumai:

- ilgesnis verslo taisyklės vykdymo laikas, kurį kompensuoja esybės sukūrimas „Pilnas_ apribojimas“ .

3.1.4. Reikalavimai, skirti vientisumo apribojimų valdymo sistemos suderinamumui su įvairia sistemine PĮ

Vientisumo apribojimų redaktorius turi turėti galimybes nuskaityti IS, kuriai kuriami apribojimai, DB schemą, naudojant prieigos parametrus (serverio IP adresas ar vardas, duomenų bazės, prisijungimo vardas ir slaptažodis).

3.1.5. Vientisumo apribojimų valdymo sistemai keliami nefunkciniai reikalavimai

Keliamus nefunkcinius reikalavimus galima suskirstyti į grupes: programinės įrangos veikimo, jos modifikavimo ir mobilumo. Toliau lentelėse pateikiamos reikalaujamos savybės ir jų aprašymai.

Programinės įrangos veikimo grupei apibrėšime šiuos 25 lentelėje pateikiamus kokybės parametrus.

25 lentelė. Veikimo reikalavimai

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
Efektyvumas	Sistema veikia efektyviai išnaudodama visus turimus sisteminius resursus.
Palaikomumas	Programinė įranga turi būti sukurta taip, kad ji galėtų patenkinti nuolat besikeičiančius vartotojo reikalavimus.
Patikimumas	Programinė įranga turi funkcionuoti taip, kad funkcionavimo sutrikimo atveju savo veikimo aplinkai nepadarytų sisteminės ir ekonominės žalos.

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
Našumas	Programinė įranga turi efektyviai išnaudoti jai skirtus resursus (procesoriaus darbo greitis, atminties kiekis ir t.t.)
Panaudojamumas	Programinė įranga turi būti patogi naudoti savo vartotojams, t.y. turėti intuityvią vartotojo sąsają ir gerą dokumentaciją.
Sistemos išplėtimas	Turi būti suteikta galimybė išplėsti programinę įrangą nauju funkcionalumu.
Stabilumas	Apribojimų IS turi palaikyti stabilų ryšį su klientinėmis IS.

Programinės įrangos mobilumo grupei priklauso 26 lentelėje pateikiami kokybės parametrai.

26 lentelė. Mobilumo reikalavimai

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
Pernešamumas	PĮ lengvai pernešama į kitą platformą (OS, naujesnę DBVS versiją).
Pakartotinis panaudojamumas	Atskiras PĮ dalis galima pritaikyti kitose sistemose ir procesuose (ETL).
Sąveikos galimybės	Lengva sistemą integruoti su kitomis PĮ.
Prieinamumas	Duomenys turi būti prieinama tuo metu, kai jų prireikia.

Programinės įrangos modifikavimo grupei priskiriami 27 lentelėje aprašyti kokybės parametrai.

27 lentelė. Modifikavimo reikalavimai

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
Informatyvumas	Turi būti pateikiami informatyvūs pranešimai atliekant tam tikras funkcijas ar negalint jų įvykdyti.
Testuojamumas	PĮ turi būti lengvai testuojama.
Lankstumas	Sistema turi būti suvokiama ir keičiama ne tik jų kūrėjų, bet ir kitų asmenų.

Kitos PĮ savybės pateikiamos 28 lentelėje.

28 lentelė. Kiti nefunkciniai reikalavimai

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
Paprasta, logiška ir reikalavimus įgyvendinanti vartotojo sąsaja	Paprasta, aiški ir funkcionali vartotojo sąsaja leis vartotojui įgyvendinti norimą tikslą - įvesti, redaguoti duomenų bazės vientisumo apribojimus. Sistemos vartotojas turi būti informuojamas apie klaidingai

Reikalavimas/Savybė	Kokybės reikalavimų aprašymas
	atliekamus veiksmus. Klaidų pranešimai turi būti informatyvūs, su nuorodomis apie teisingą tam tikros funkcijos atlikimą;
Informatyvi vartotojo dokumentacija	Pagalbos sistemos įgyvendinimas leis išvengti sistemos ir vartotojo klaidų, veiksmai bus atliekami greitai ir sklandžiai.
Išsami produkto dokumentacija	Išsami sistemos dokumentacija bus naudinga produkto išplėtimui nauju funkcionalumu.
Vartotojų autentifikacija ir autorizacija	Turi būti užtikrintas vartotojų autentikavimas ir autorizavimas (prisijungimo vardas/slaptažodis, vartotojų teisės ir privilegijos);
Apribojimų sistemos kopijos saugojimas	Sistema turi užtikrinti, kad kasdien būtų atliekamas visos IS veiklos apribojimų sistemos atsarginės kopijos realizavimas. Saugoti 3 paskutiniąsias kopijas.

3.1.6. Reikalavimai vientisumo apribojimų valdymo sistemos duomenims

Poskyriuje detalizuojami vientisumo apribojimų valdymo sistemos duomenims keliami reikalavimai.

- Sistemos įėjimo duomenys vienu iš toliau minimu būdu:
 - turi būti SQL formato (Insert, Update, Delete, Select sakiniai);
 - kitai funkcijai galiojantis apribojimo turinys;
 - egzistuojančios Oracle procedūros pavadinimas ir tinkamo tipo parametrai.
- Vartotojų prisijungimo prie apribojimų redaktoriaus slaptažodžiai turi būti saugojami DB koduoti MD5 algoritmu.
- Duomenys suvesti į DB turi būti logiški.

3.1.7. Reikalavimai programinei įrangai

Reikalavimai, ODP.NET įdiegimui pateikti 29 lentelėje.

29 lentelė. Reikalavimai programinei įrangai

Reikalavimo aprašymas	Reikalavimas
Microsoft .NET Framework versija	Microsoft .NET Framework 1.0 ar aukštesnė
Operacinė sistema	Windows 98, Windows NT, Windows XP, Windows 2000, Windows Server 2003
Oracle DB versija	Oracle9i ar žemesnės versijos duomenų bazė
Oracle kliento versija	Oracle9i ar žemesnės versijos klientas
Servisai	.Net servisas, kartu su ODP.NET programine įranga

3.1.8. Reikalavimai techninei įrangai

Apribojimų valdymo sistemai reikalingi (minimalūs, rekomenduotini) techninei įrangai resursai aprašyti 30 lentelėje.

30 lentelė. Reikalavimai techninei įrangai

Resursas	Minimalus	Rekomenduojamas
CPU	Pentium III 750	Pentium 4 1500
Atmintis	256 MB	512MB; 1GB rekomenduojamas didesniems projektams
Disko talpa	200 MB	200 MB ar daugiau
Video režimas	800*600 @ 64k spalvos	1280*1024 @ 64k spalvos
Operacinė sistema	Windows 98/2000/XP	

3.1.9. Reikalavimai IS apribojimų redaktoriaus vartotojo sąsajai

Aiškumas. Vartotojo sąsaja turi būti aiški ir nereikalaujanti iš vartotojo keisti jam jau įprastų uždavinio sprendimo būdų. Sistemos vartotojas turi būti informuojamas apie klaidingai atliekamus veiksmus. Klaidų pranešimai turi turėti nuorodas apie teisingą tam tikros funkcijos atlikimą. Turi būti naudojami dalykinės srities terminai, sąvokos ir žymėjimai.

Suderinamumas. Vartotojo sąsaja turi būti suderinta, t.y. leistų vartotojams jau sukauptas darbo su programa žinias panaudoti naujų uždavinių sprendimui (apribojimų įvedimui ir modifikavimui), greičiau įsisavinti naujas sistemos galimybes ir dėka to, susikoncentruoti ne į naujų mygtukų, komandų ar meniu punktų perpratimą, o į sprendžiamas problemas.

Draugiškumas. Siekti, kad dirbant su programa būtų galima ištaisyti klaidas, užtikrinti, kad atlikus neteisingą veiksmą nebūtų išgadinti duomenys ar pati sistema.

Grižtamasis ryšys. Svarbus vartotojo veiksmas turi būti vizualiai patvirtintas tam, kad vartotojas žinotų, jog jo pasirinkta komanda buvo įvykdyta.

Paprastumas. Vartotojo sąsaja turi būti kiek įmanoma paprastesnė, kad būtų ją lengviau įsisavinti ir naudoti. Darbiniame programos lange turi būti rodoma informacija, kuri reikalinga tam tikros funkcijos atlikimui. Turi būti vengiama ilgų meniu punktų, mygtukų ir

formos laukų pavadinimų. Atlikti vartotojo sąsajos elementų grupavimą pagal prasmę, kad leistų išnaudoti vartotojo asociatyvinį mąstymą.

Estetiškumas. Teisingai parinkti vartotojo sąsajos spalvas, elementų išvaizdą, kas leistų vartotojui lengviau perprasti sąsają, susikoncentruoti į savo sprendžiamus uždavinius.

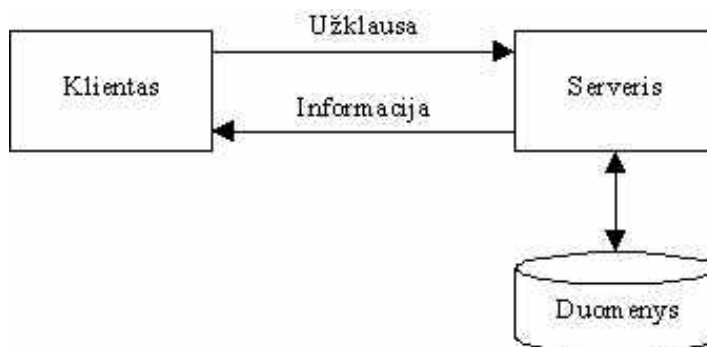
3.2. Centralizuoto vientisumo apribojimų valdymo sistemos projektas

Šiuolaikinės duomenų bazės paprastai yra didelės apimties ir sudėtingos, norint išsamiai su jomis susipažinti, reikia daug laiko. Todėl ypač svarbu, kad būtų galima greitai nuspręsti, ar duomenys yra visiškai tinkami norimam tikslui, yra teisingi ir nėra prieštaravimų.

Apribojimų redaktorius užtikrins greitą ir efektyvų apribojimų įgyvendinimą ir modifikavimą, o tinklu teikiamos paslaugos (angl. Web Services) leis patikrinti IS duomenų korektiškumą, teisingumą, naudojant SQL DML sakinius, ar atitinka IS duomenų reikalavimus (pvz. kad alga negali būti mažesnė nei nustatytas valstybės minimumas, darbuotojo įdarbinimo data turi būti ankstesnė nei atleidimo data ir kt.).

3.2.1. Fizinė vientisumo apribojimų valdymo sistemos architektūra

Sistemos projektui įgyvendinti pasirinktas kliento/serverio architektūros modelis. Veikimo principas pateiktas grafiškai (35 pav.). Klientas atsakingas už ryšį su vartotoju. Serveris atsakingas už užduočių, kurias pateikia klientas, vykdymą. Serveris gaus iš klientų užklausas ar kitus duomenis, apdoros juos, vykdydamas tikrinimo komandas ir grąžins rezultatus klientui.



35 pav. Kliento/serverio modelis

Kliento/serverio sistemos sukūrimui naudojama objektiškai orientuota programų sistemų inžinerija.

3.2.2. Loginė vientisumo apribojimų valdymo sistemos architektūra

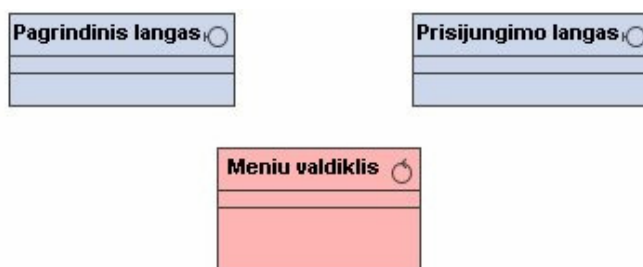
Pasirinktas trijų lygių architektūros modelis (36 pav.). Remiantis šiuo modeliu, vientisumo apribojimų valdymo sistemą sudaro vartotojo, veiklos ir duomenų paslaugos, sugrupuotos į atitinkamus paketus.



36 pav. Apribojimų posistemio loginė architektūra

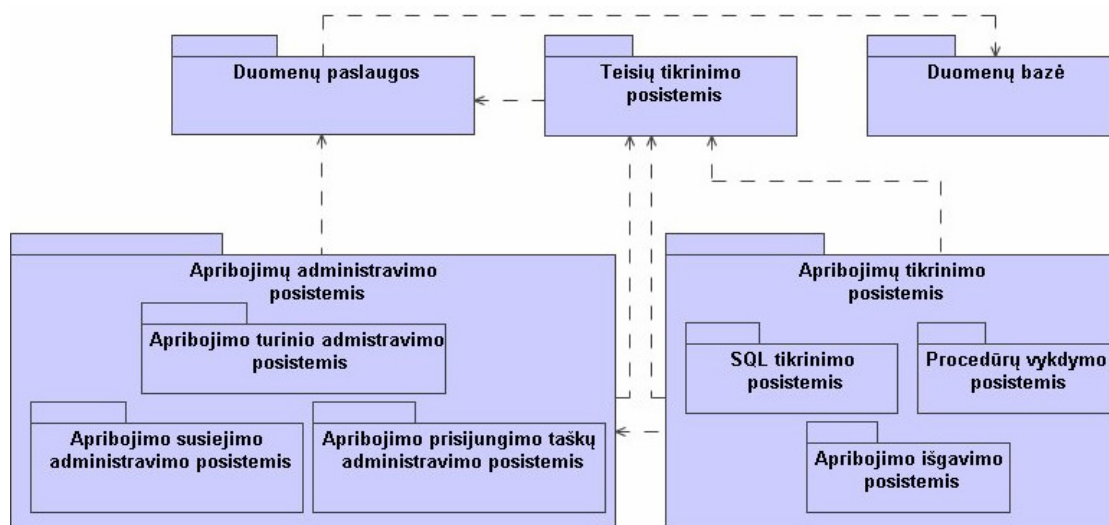
Kadangi sistema – didelė, ji išskaidyta į posistemius (37 pav.). Įtrauktas papildomas posistemis „Pagrindinis posistemis“, kuriame sukoncentruotas visas vientisumo apribojimo valdymas – pagrindinis langas, meniu valdiklis, kuris nukreipia vartotojų įvykius į atitinkamus posistemius. Šiam posistemiiui (37 pav.) priskirtas ir prisijungimo langas, kuris leidžia patekti į pagrindinį langą.

Pateikiamos pagrindiniam posistemiiui priskirtos klasės (37 pav.).



37 pav. Pagrindiniam posistemiiui priskirtos klasės

Apribojimų valdymo sistema išskaidyta į posistemius (38 pav.).



38 pav. Apribojimų valdymo sistemos loginė architektūra

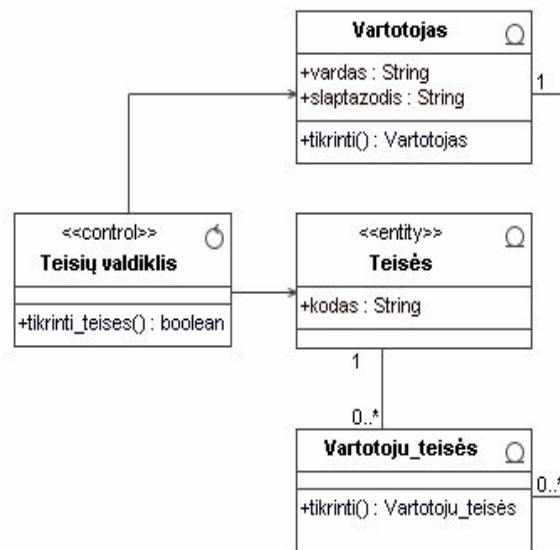
Modelyje (38 pav.) visi posistemiai naudoja duomenų prieigos klasę, turi priklausomybės ryšius su teisių tikrinimo posistemiu, kadangi vartotojų galimybės naudoti tam tikras funkcijas, priklauso nuo jų teisių. Todėl sistema tikrina vartotojo teises ne tik jam prisijungiant, bet ir pasirinkus kitą meniu punktą. Išskirtos paslaugos:

- **Vartotojo paslaugos**

Vartotojo paslaugos atitiks ribines klases, kurios buvo identifikuotos, remiantis panaudojimo atvejų specifikacijų analize bei vartotojo interfeiso eskizais.

- **Veiklos paslaugos**

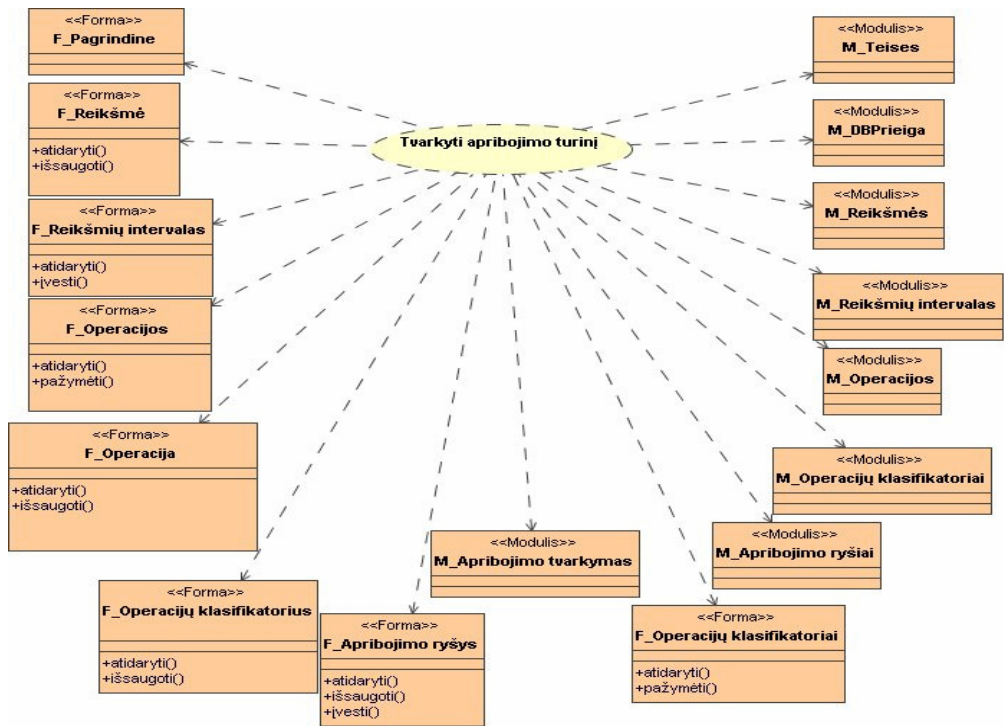
Veiklos paslaugos išvedamos iš analizės etapo valdiklių ir esybių klasių (39 pav.). Teisės tikrinimo posistemio projekto modelį sudaro veiklos logikos (valdiklių ir esybių) klasės. Vartotojo interfeiso šis posistemis neturi, jį naudoja kiti posistemiai.



39 pav. Teisių tikrinimo veiklos logikos klasių diagrama

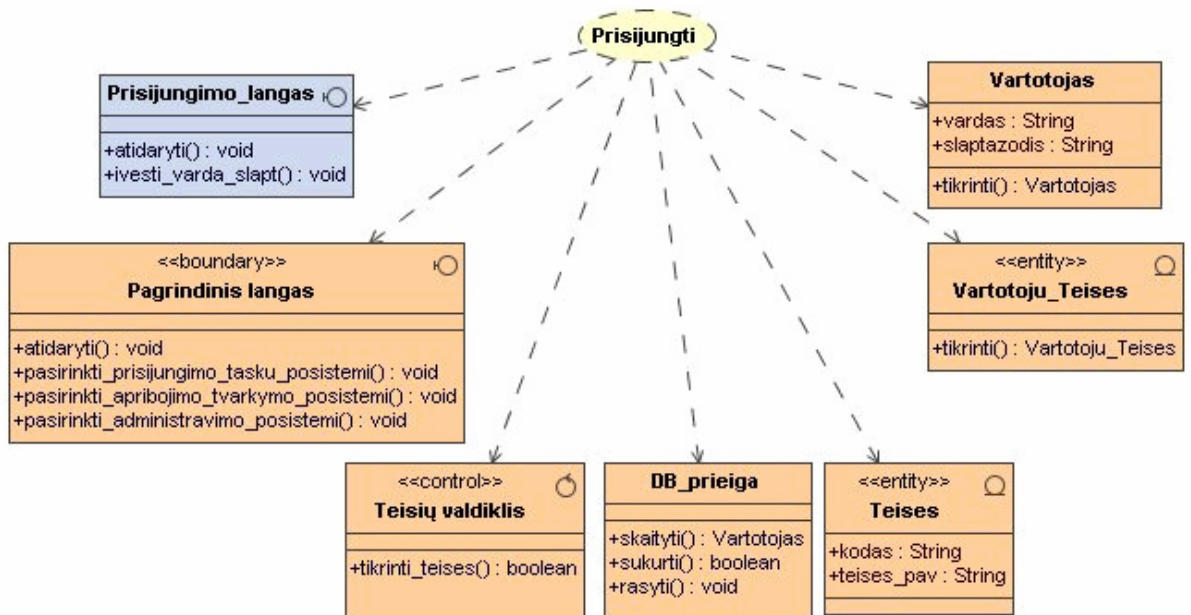
Apribojimo tvarkymo panaudojimo atvejo realizacijos klasės (40, 41 pav.)

Šios klasės rodo, kokios klasės reikalingos realizuojant kiekvieną panaudojimo atvejį. Leidžia pamatyti, ar yra visos reikiamos klasės, ar nėra nereikalingų.



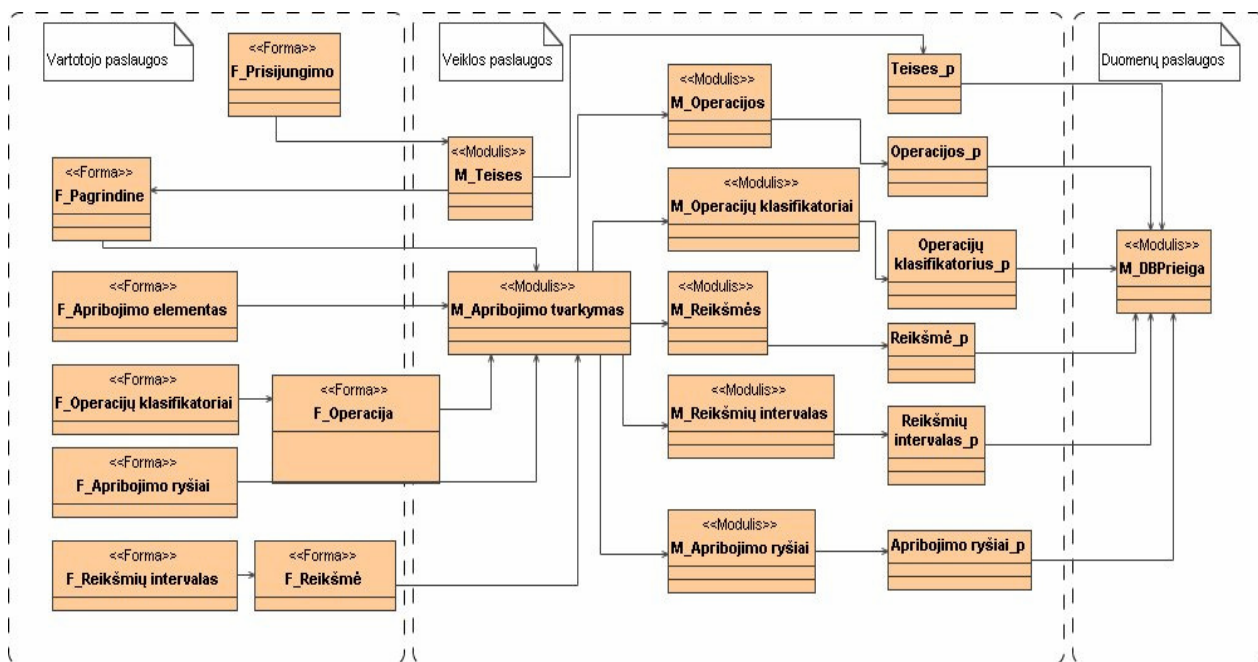
40 pav. Apribojimo tvarkymo realizacijos klasės

Prisijungimo panaudojimo atvejo realizacijos klasės (41 pav.)



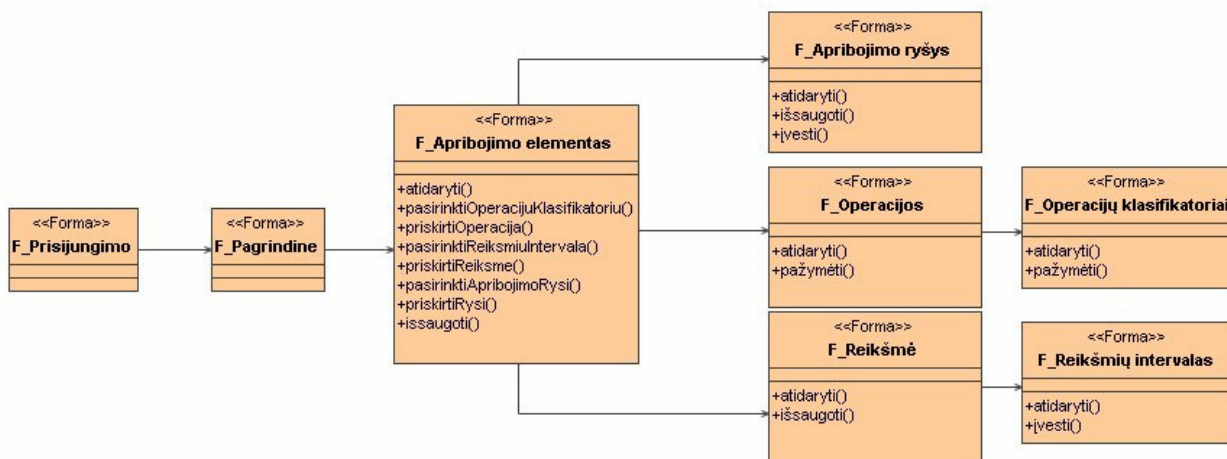
41 pav. Prisijungimo realizacijos klasės

Pateikiama apribojimų turinio tvarkymo projekto modelio trijų lygių klasių diagrama (42 pav.).



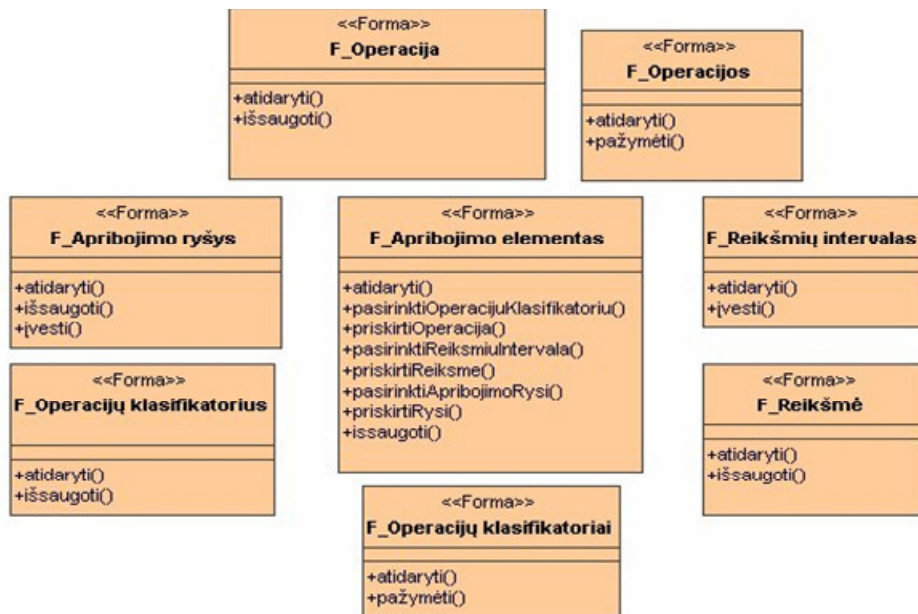
42 pav. Apribojimų tvarkymo projekto modelio trijų lygių klasių diagrama

Pateikiama apribojimų tvarkymo interfeiso navigavimo planas (43 pav.), kaip galima pereiti iš vieno lango šį kitą.



43 pav. Apribojimo tvarkymo interfeiso navigavimo planas

Toliau modeliujame detalų apribojimų tvarkymo vartotojų paslaugų klasių modelį (44 pav.).



44 pav. Detali apribojimų tvarkymo vartotojų paslaugų klasių diagrama

▪ **Duomenų paslaugos**

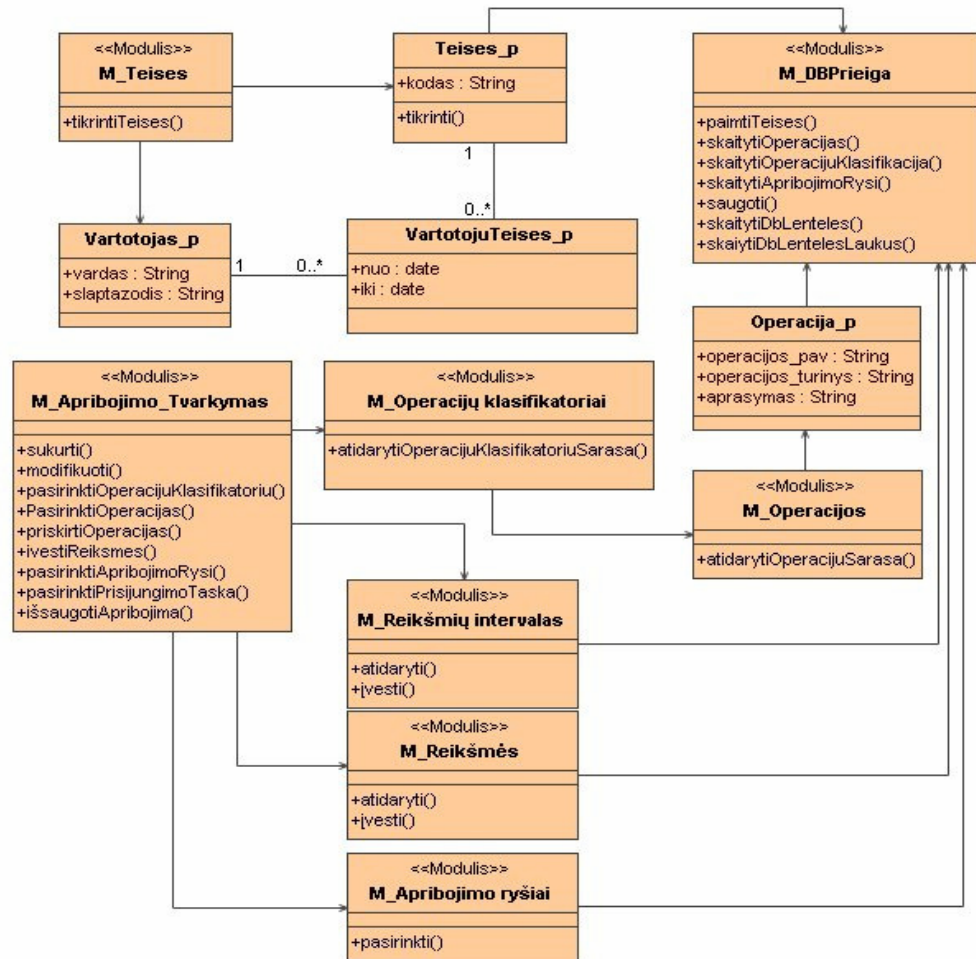
Sąveikos su duomenų baze operacijas realizuoja vienas modulis, kuris atlieka prisijungimo, užklausų formavimo, duomenų skaitymo – rašymo operacijas (45 pav.).



45 pav. DB prieigos modulis

3.2.3. Detalaus apribojimo turinio tvarkymo projektas

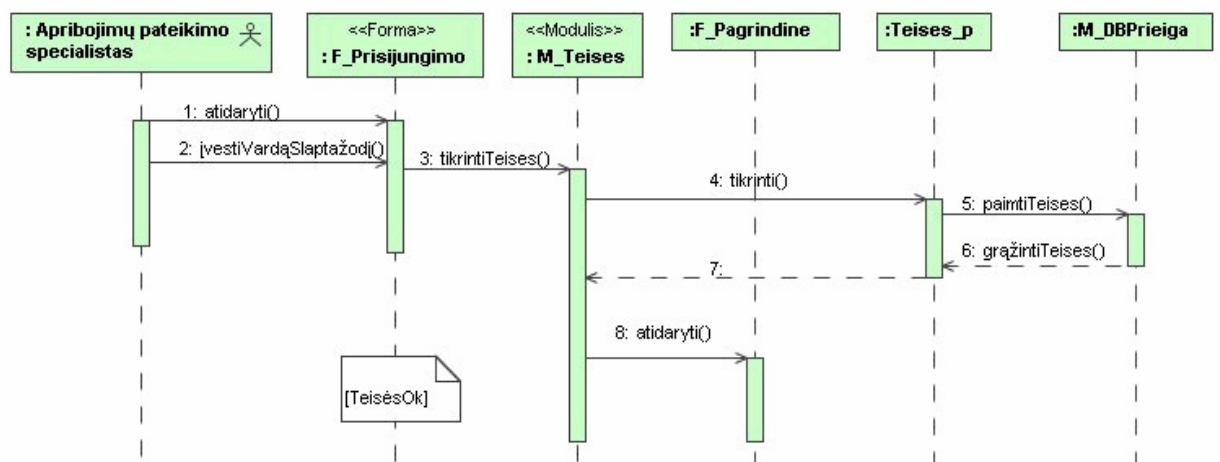
Apribojimo turinio tvarkymo projekte (46 pav.) pateikiamos su apribojimų kūrimu ir tvarkymu susijusios klasės ir metodai.



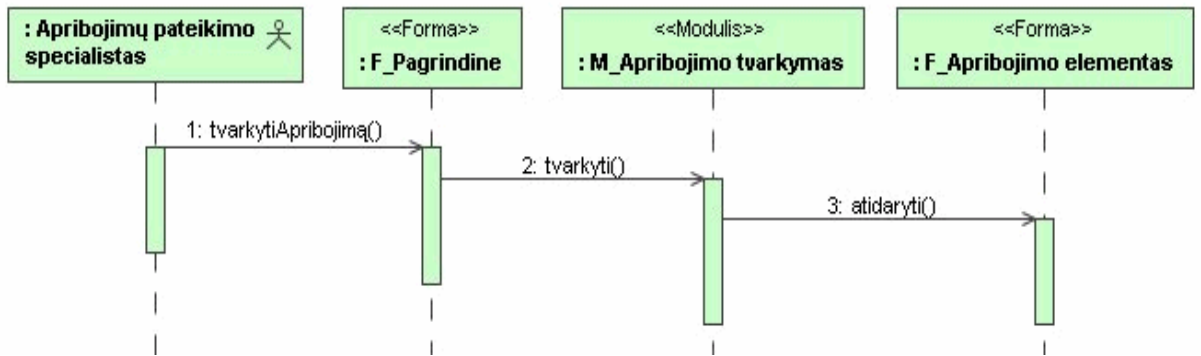
46 pav. Apribojimo turinio tvarkymo detalaus projekto modelis

3.2.4. Vientisumo apribojimų valdymo sistemos elgsenos modelis

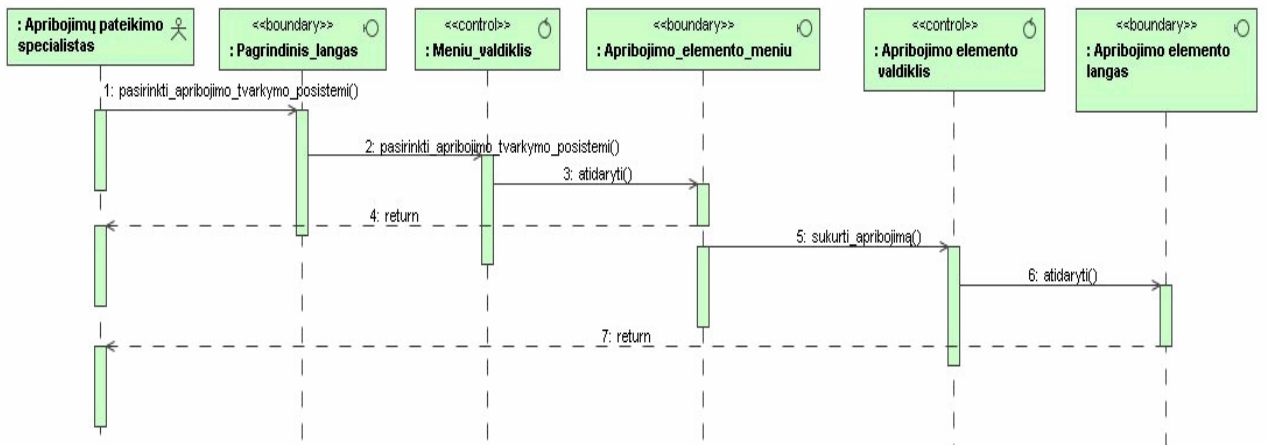
Kiekvienai panaudojimo atvejo realizacijai sukurtos sekų diagramos (47 - 51 pav.), kurios atspindi pagrindines sąveikas tarp architektūros elementų.



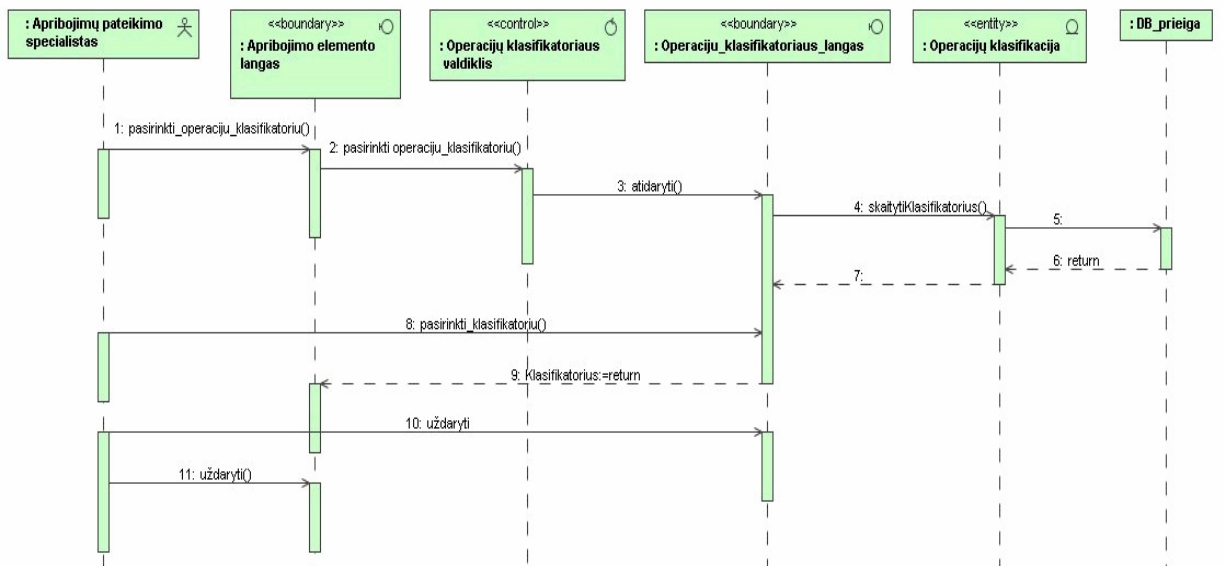
47 pav. Prisijungimo sekų diagrama



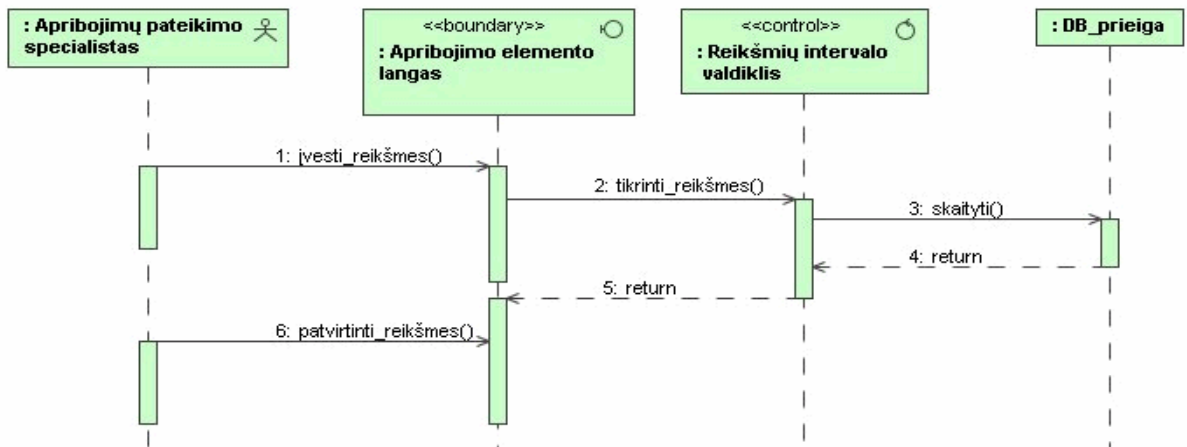
48 pav. Abstrakti apribojimų tvarkymo sekų diagrama



49 pav. Apribojimų sukūrimo pasirinkimo iš pagrindinio lango meniu, sekų diagrama



50 pav. Operacijų klasifikatoriaus parinkimo sekų diagrama



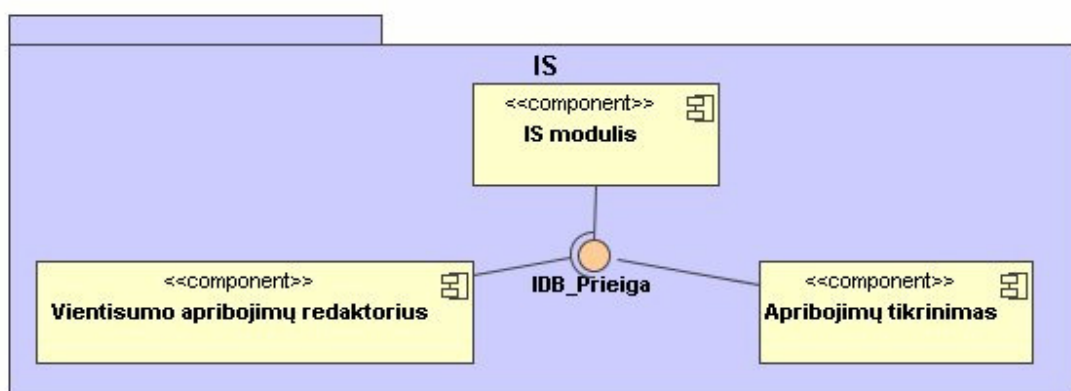
51 pav. Reikšmių intervalo įvedimo sekų diagrama

3.2.5. Realizacijos modelis

Realizacijos modeliuose pateiksime programinių komponentų diagramą (52 pav.) ir įdiegimo modelį (53 pav.).

3.2.5.1. Programinių komponentų architektūra

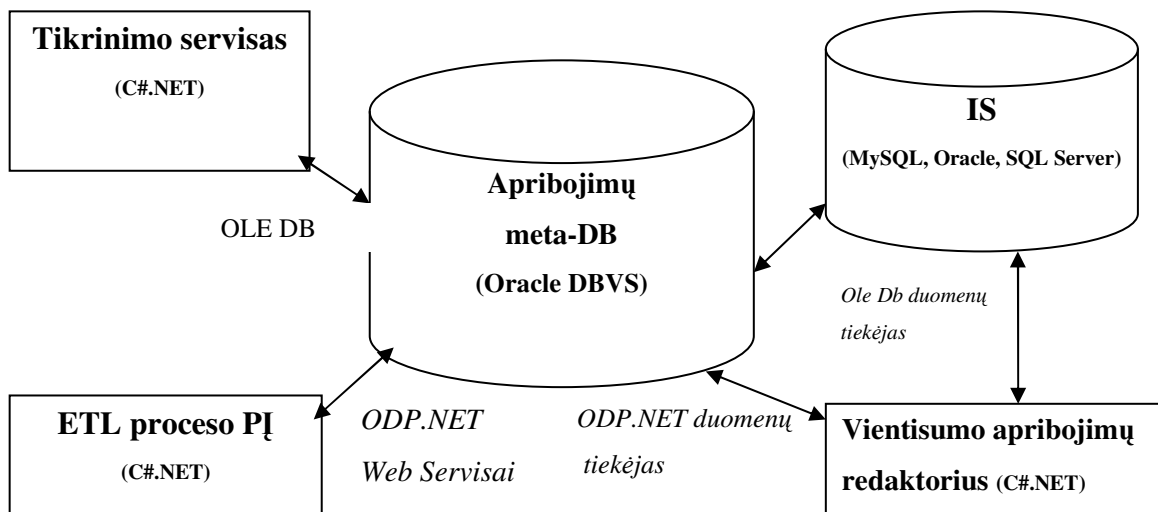
Modelyje (52 pav.) pateikiami pagrindiniai komponentai ir jų sąveikos tarpusavyje. Vientisumo apribojimų redaktorius rodomas, kaip su IS sąveikaujantis per DB prieigą komponentas, papildoma PI, apribojimų kūrimui ir tvarkymui įgyvendinti. Apribojimų tikrinimo servisas – komponentas, užtikrinantis, kad duomenys patenkantys į IS būtų teisingi, neprieštaringi, tenkintų meta DB saugojamų apribojimų sąlygas.



52 pav. Programinių komponentų architektūra

3.2.5.2. Vientisumo apribojimų valdymo sistemos diegimo modelis

Pateikiamas abstraktus (53 pav.) vientisumo apribojimų valdymo sistemos diegimo modelis, kuris parodo pasiskirstymą tarp techninės ir programinės įrangos (komponentų).



53 pav. Sistemos diegimo modelis

Šio diegimo modelio (53 pav.) ašis – apribojimų meta DB, kuri realizuojama Oracle9i DBVS. Joje saugojami loginėmis išraiškomis aprašyti apribojimai ir jų sąlygos, IS objektai (lentelių ir laukų vardai), prisijungimo parametrai. Taip pat redaktoriaus ir tikrinimo serviso vartotojai, jų prieigos teisės ir veiklos procedūros su parametrais. Apribojimų modelis pateiktas 3.1.3. poskyriuje.

Vientisumo apribojimų redaktorius sukurtas .NET technologijomis, įgyvendina apribojimų kūrimą ir tvarkymą. Redaktoriaus PĮ su Oracle DBVS bendrauja, naudojant ODP.NET duomenų tiekėją, kurio esminės charakteristikos ir naudojimo inovatyvumas pateikiamas poskyriuje 2.7.1.

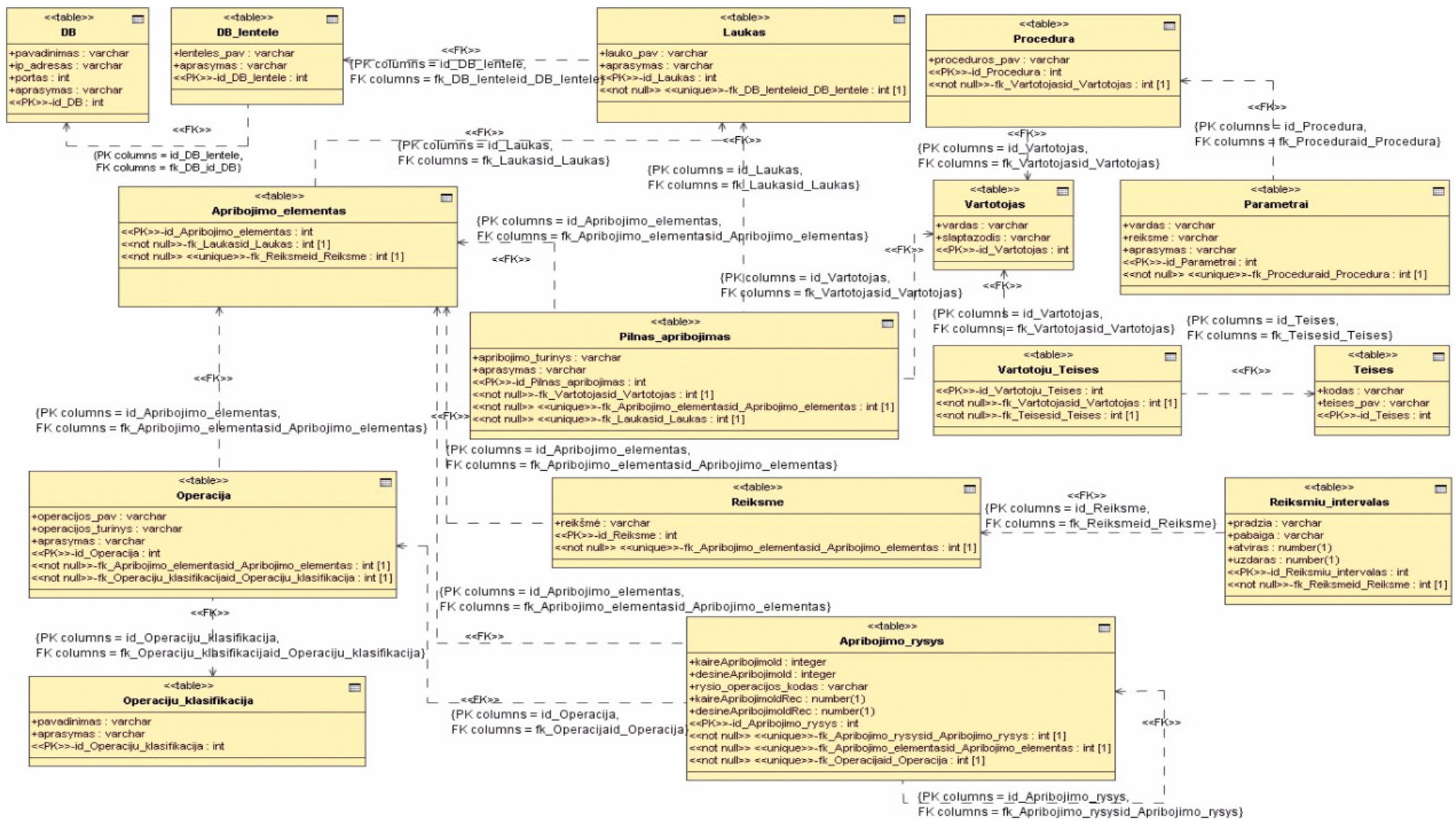
Vientisumo apribojimų redaktoriaus PĮ, naudojant OLE DB duomenų tiekėją, nuskaito IS, kurių DBVS gali būti (MySQL, SQL Server, Oracle) objektus (lenteles ir jų laukus), apribojimų kūrimui. OLE DB pagalba užtikrinamas vieningas schemų nuskaitymas.

Tikrinimo servisas ir ETL PĮ įgyvendinta išnaudojant .NET Web Servisų galimybes kartu su ODP.NET, Ole DB duomenų tiekėjais. Priklausomai nuo IS DBVS parametrų parenkamas tinkamas.

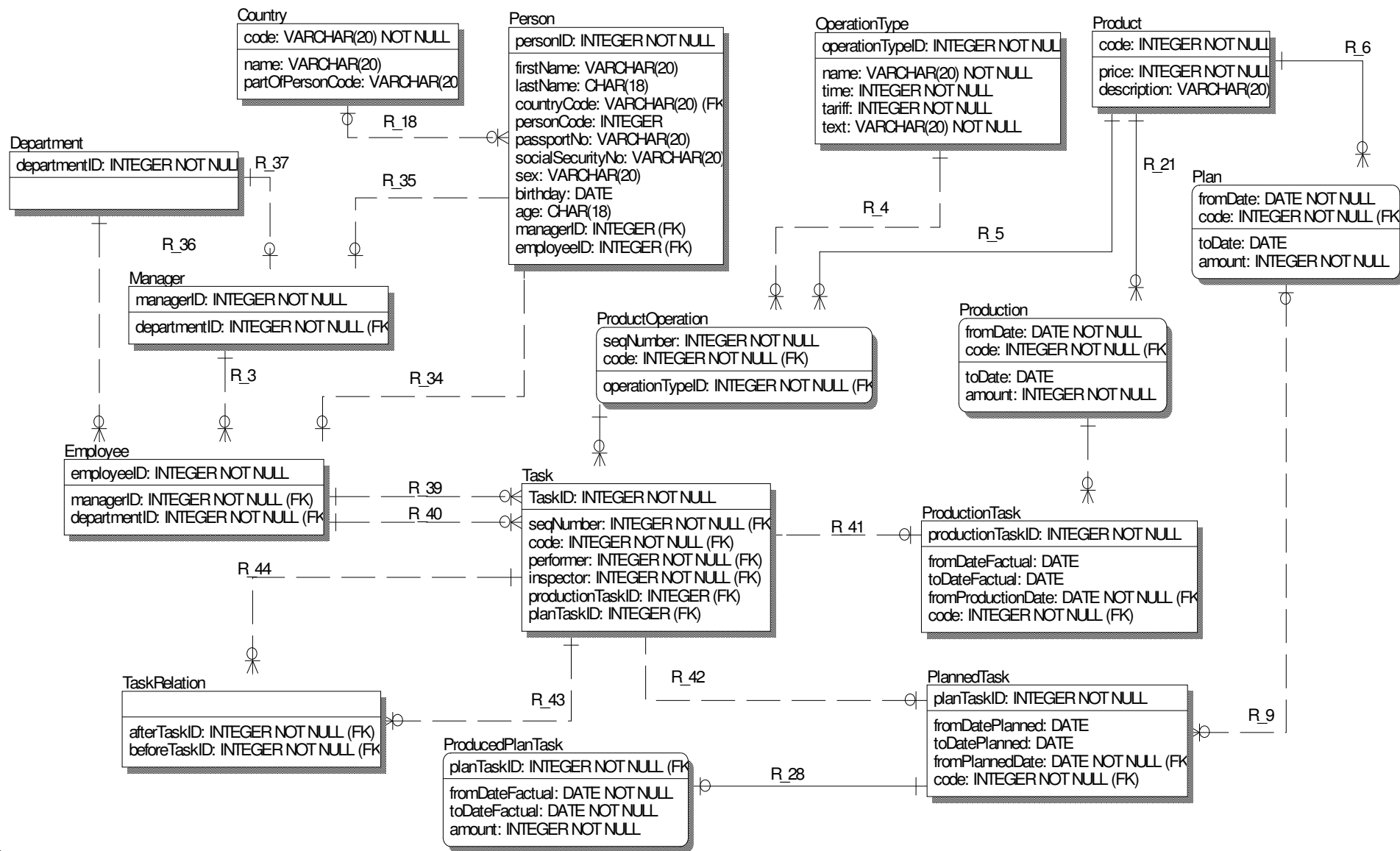
3.2.5.3. Duomenų bazės schemos

Pateikiamos 2 duomenų bazės schemos: apribojimų meta DB (54 pav.) ir eksperimentui naudojama gamybinės IS DB schema (55 pav.).

Meta – db schema gauta įrankio Magic Draw 10.0 suteiktomis galimybėmis – transformuotas klasių modelis į duomenų bazės schemas modelį. Pasirinktas Oracle DBVS profilis. Sumodeliuotos 25 duomenų bazės lentelės, jų pirminiai ir išoriniai raktai. Pagal nutylėjimą – tai sveikėjo tipo kintamieji. Meta - db lentelių struktūra pateikiama 4 Priede.



54 pav. Meta DB schemas diagrama



55 pav. Gamybinės DB schemas diagrama

3.2.5.4. Reikalavimai sistemos funkcionavimui

31 lentelėje pateikiami reikalavimai apribojimų valdymo sistemos funkcionalumui.

31 Lentelė. Reikalavimai sistemos funkcionalumui

Reikalavimo aprašymas	Reikalavimas
Microsoft .NET Framework versija	Microsoft .NET Framework 1.0 ar aukštesnė
Operacinė sistema	Windows 98, Windows NT, Windows XP, Windows 2000, Windows Server 2003
Oracle DB versija	Oracle9i ar žemesnės versijos duomenų bazė
Oracle kliento versija	Oracle9i ar žemesnės versijos klientas
Servisai	.Net servisas, kartu su ODP.NET programine įranga
Duomenų tiekėjai	ODP.NET OleDB, skirtas MySQL, MS SQL Server 2005, Oracle DBVS.

4. EKSPERIMENTINIS VIENTISUMO APRIBOJIMŲ VALDYMO SISTEMOS TYRIMAS

Sukurtą redaktorių, tikrinimo servisą ir Oracle ETL įgyvendinantį servisą pritaikysime gamybinei IS, kurios DB modelis pateikiamas 55 paveiksle, taip pat įmonės biudžeto valdymo IS.

Šiame skyriuje – sudarytas eksperimento modelis, kad įvertinti, ar PĮ atitinka apibrėžtus kokybės kriterijus ir nustatyti problemas (defektus), dėl kurių sistema negali atitikti vartotojo poreikių.

Eksperimentui atlikti, pirmiausia pateikiamos sukurtos PĮ programiniai langai su aprašymais.

4.1. Centralizuoto vientisumo apribojimų valdymo sistemos aprašymas

Tolesniuose poskyriuose aprašoma sukurta centralizuoto vientisumo apribojimų valdymo sistemos dalys: redaktorius, tikrinimo servisas ir Oracle ETL vykdymą, įgyvendinanti PĮ. Detalus aprašymas reikalingas sistemos efektyviam eksperimentui atlikti, patikrinti ar sistema atitinka keliamus kokybės ir funkcionalumo reikalavimus, ar vartotojas gali atlikti norimus veiksmus, kaip nurodyta aprašymuose.

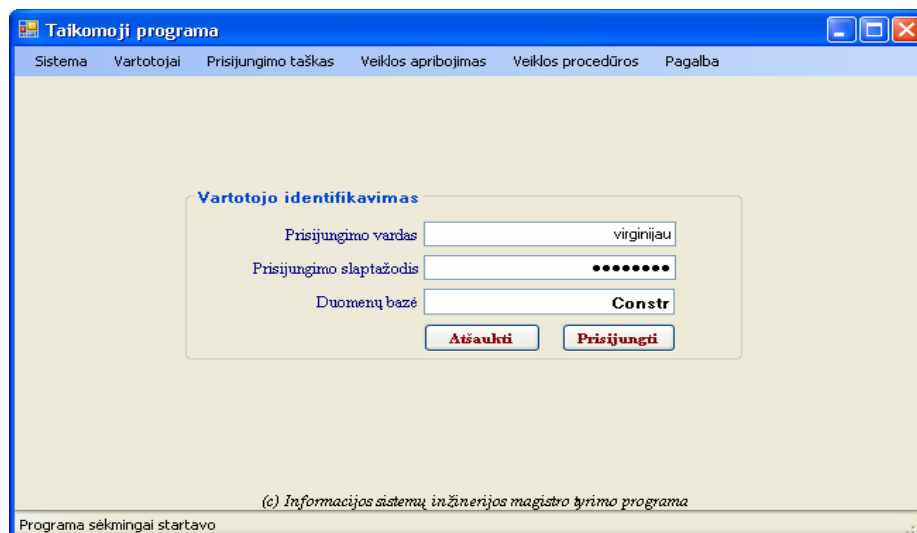
4.1.1. Apribojimų redaktoriaus naudojimo aprašymas

Šiame poskyriuje detalai aprašomi apribojimų kūrimą ir modifikavimą įgyvendinančio redaktoriaus programos langai ir pagrindinės funkcijos.

Pagrindinis darbo su IS vientisumo apribojimų redaktoriumi langas (56 pav.). Norėdamas dirbti su šiuo redaktoriumi vartotojas turės autentifikuotis sistemoje (vardas/slaptažodis/bazė) ir turėti teises prie DBVS, kad atlikti norimus veiksmus. Visose programos languose rodomas prisijungusio vartotojo vardas.

Pagrindiniai meniu punktai:

- *Sistema*
- *Vartotojai*
- *Prijungimo taškas*
- *Veiklos apribojimas*
- *Veiklos procedūros*
- *Pagalba*

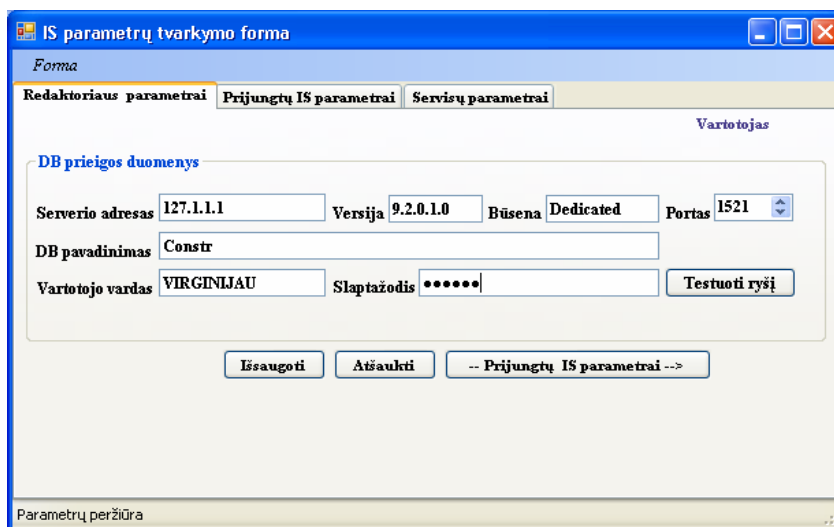


56 pav. Pagrindinis redaktoriaus langas

Meniu „Sistema“ sub - meniu punktai

Sistemos parametrai:

- Redaktoriaus parametrų langas (57 pav.), skirtas peržiūrėti DBVS parametrus. Testuoti ryšį kliento ryšį su apribojimų serveriu (*virtotojui pateikiami informaciniai arba klaidos pranešimai*).



57 pav. Vientisumo apribojimų redaktoriaus parametrų langas

Redaktoriaus parametrai:

- *serverio adresas;*
- *porto numeris;*
- *duomenų bazės pavadinimas;*
- *virtotojo vardas, slaptažodis;*
- *versija.*

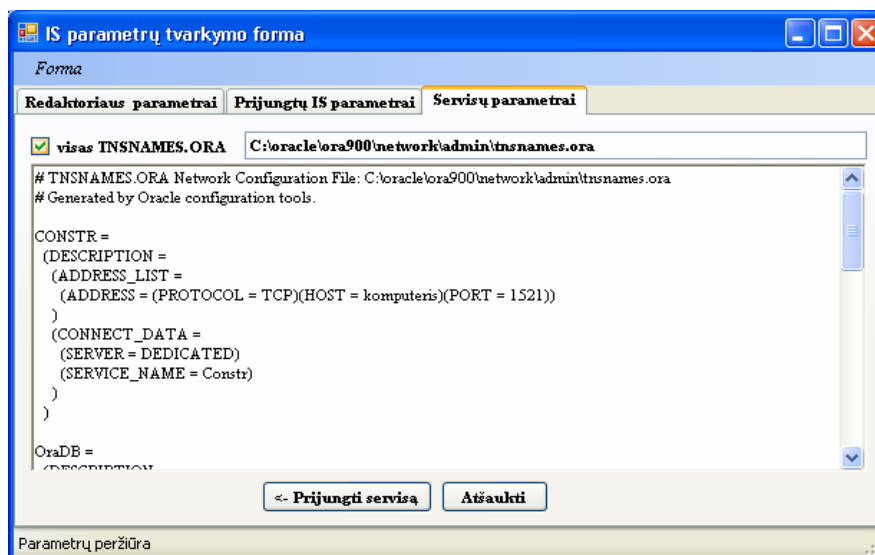
- Prijungtų IS parametrai – suteikti galimybes redaktoriaus vartotojui prijungti, atjungti, saugoti, peržiūrėti, atrinkti IS, kuriai bus kuriami vientisumo apribojimai, DBVS parametrus (58 pav.). Testuoti ryšį su serveriu (vartotojui pateikiami informaciniai arba klaidos pranešimai). Vartotojo patogumui, realizuota navigacija tarp kitų uždavinių atlikimo.

58 pav. Prijungtų IS parametru tvarkymo langas

IS parametrai:

- *serverio adresas;*
- *porto numeris;*
- *duomenų bazės pavadinimas;*
- *vartotojo vardas, slaptažodis;*
- *versija.*

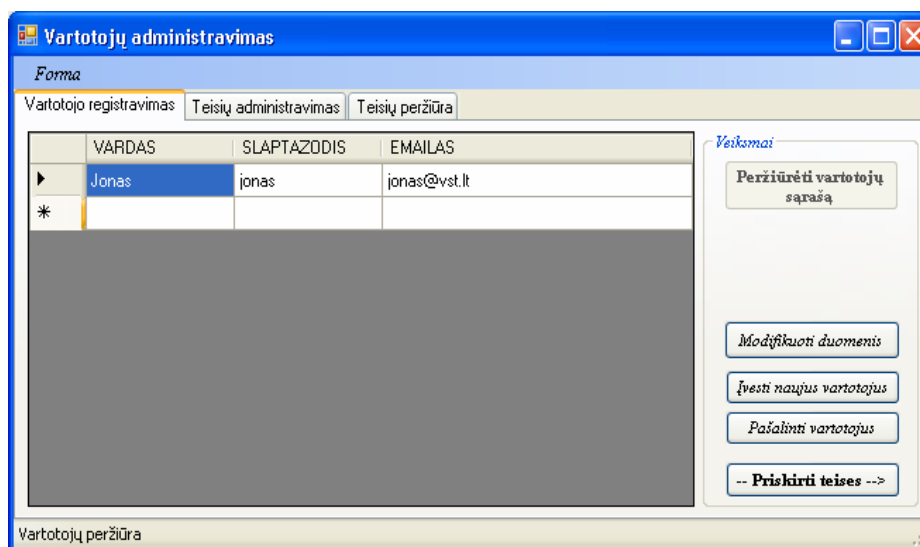
Toliau modeliuojama forma (59 pav.), skirta Oracle Servisu peržiūrai iš kliento. Oracle DBVS servिसai aprašomi TNSNAMES.ORA faile. Tam reikės nurodyti kelią iki servisu aprašomo failo.



59 pav. Oracle kliento parametru peržiuros langas

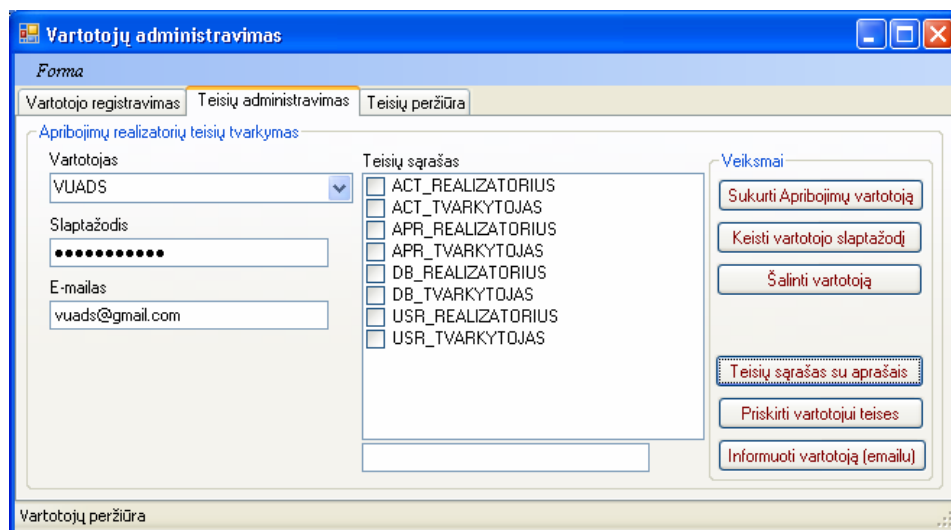
- *Vartotoju, teisiu tvarkymas - suteikti galimybes sistemos administratoriui sukurti naujus vartotojus, juos administruoti, taip pat tvarkyti ju teises (60, 61 pav.).*

Pagrindiniame lange (60 pav.) administratorius gali kurti vartotoja, perziureti ju sarsa, modifikuoti asmens duomenis, filtruoti pagal prisijungimo varda.



60 pav. Vartotoju administravimo langas

Sekančiame formos lange (61 pav.), administratorius gali priskirti, modifikuoti, atimti teises. Įdiegta funkcija – informuoti vartotoją e-mailu, kad sukurtos teisės, pateikti prisijungimo duomenis.

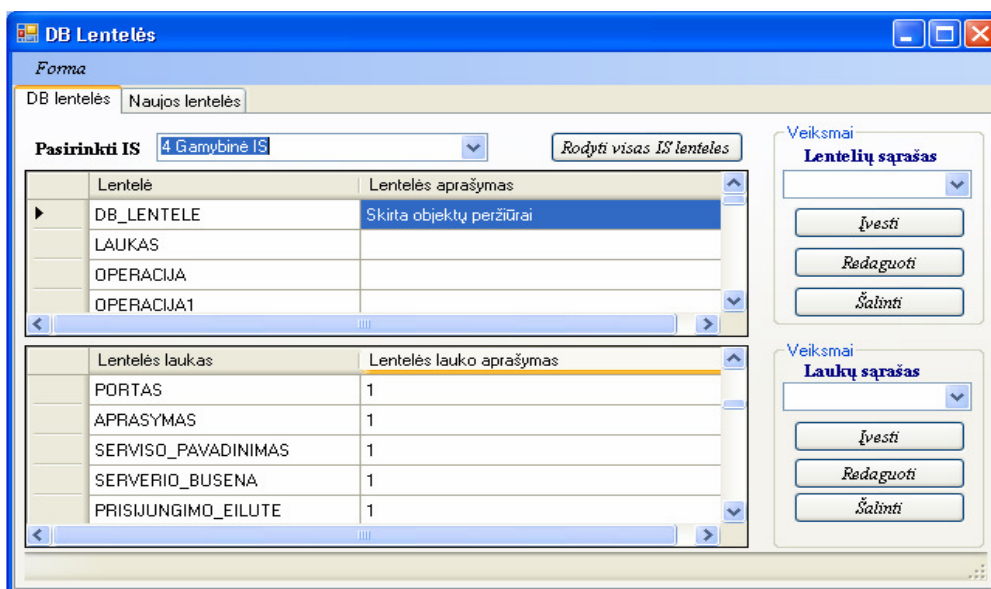


61 pav. Vartotojų teisių kūrimo ir administravimo langas

Prijungimo taškas sub - meniu punktai:

- *DB lentelių tvarkymas (62, 63 pav.) – suteikia galimybes vartotojui sukurti peržiūrėti, atsirinkti, šalinti norimą lentelę, jų laukus ir aprašymus.*

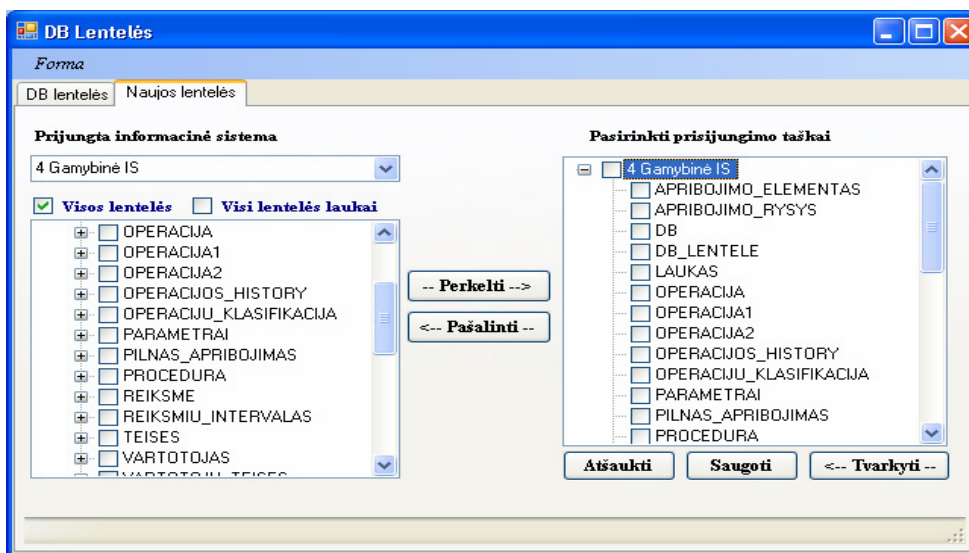
Pateikiama forma (62 pav.) – „DB lentelės“. Šioje formoje vartotojas, gali pasirinkti norimą schemos lentelę, peržiūrėti, šalinti jos laukus. Pereiti į kitą langą, naujų lentelių išsaugojimui. Redaguoti bus galima tik aprašymo lauką.



62 pav. Apribojimų DB saugojamų lentelių ir jos laukų peržiūros, administravimo langas

Sub-meniu – Nuskaityti apribojimų prisijungimo taškus

Kitame formos lange (63 pav.) – vartotojui patogiai pateikiami pasirinktos IS schemos objektai ir jų laukai. Galima pasirinkti išsaugoti visą schemą, kelias lenteles, ar išskirtinai tik tam tikrus jų laukus apribojimų meta-db.

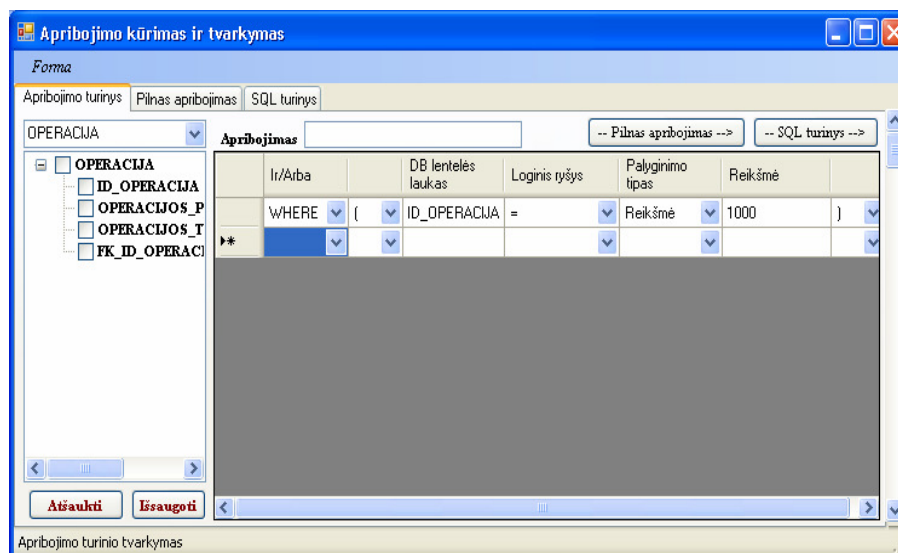


63 pav. DB lentelių ir jų laukų pasirinkimo langas

Veiklos apribojimas sub - meniu punktai:

- *Turinio tvarkymas (64, 65, 66, 67 pav.) – sumodeliuota forma, skirta apribojimo turinio įgyvendinimui ir modifikavimui. Apima (reikšmes, operacijas, db lenteles ir jų laukus). Tvarkymas - tai naujų įvedimas, esamų redagavimas ir šalinimas, peržiūra.*

Forma (64 pav.) – „Apribojimo turinio tvarkymas“. Leidžia susikurti apribojimą lentelei, ar lentelės laukui.



64 pav. Apribojimo turinio tvarkymo langas

Dešinėje – medžio struktūra atvaizduojama pasirinktos IS, lentelė ir jos laukai, kurie saugojami apribojimų meta DB. Jei norimo lauko nėra, apribojimų pateikimo specialistas gali nusiskaityti, naudodamasis forma (64 pav.).

Viršuje – tekstiniame lauke įvedamas norimas apribojimo pavadinimas, atitinkantis sukūrimo reikalingumą.

Toliau gride, jei tai pirmas apribojimo sakiny, tai lauke automatiškai suformuojamas WHERE sakiny, nurodantis, kad tai bus įvedama tam tikras apribojimas, su loginėmis sąlygomis.

Vartotojo patogumui, sekančioje celėje vaizduojamos 4 lygių skliaustai. T.y. ar skliaustai yra, ar reikalingas 1, 2, 3, 4.

2 kartus spragtelėję pele (angl. double-click) ant norimo lauko, automatiškai laukas pateikiamas 3 grido celėje.

Toliau pasirenkamas loginis ryšys iš sąrašo. Ryšių sąrašą galima redaguoti, pašalinti, peržiūrėti ir papildyti formoje „Operacijų tvarkymas“ (65 pav.). Kur pateikiamas viršuje – pirmame gride operacijų klasifikatorius, o žemiau – galimos operacijos.

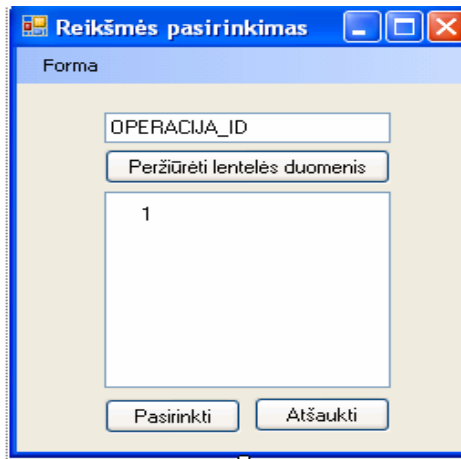
Pavadinimas	Aprašymas
Ekspontinės	Skirtos matematinėms operacijoms aprašyti.
Loginės	
Matematinės	Skirtos matematinėms operacijoms aprašyti.
Palyginimo	

Pavadinimas	Operacijos turinys	Aprašymas
Apytiksliai lygu	~=	"Apytiksliai lygu" operatorius
ARBA	OR	Loginis <ARBA>
Atimtis	-	Aritmetinė atimtis
Dalyba	/	Aritmetinė dalyba
Daugiau	>	"Daugiau" operatorius

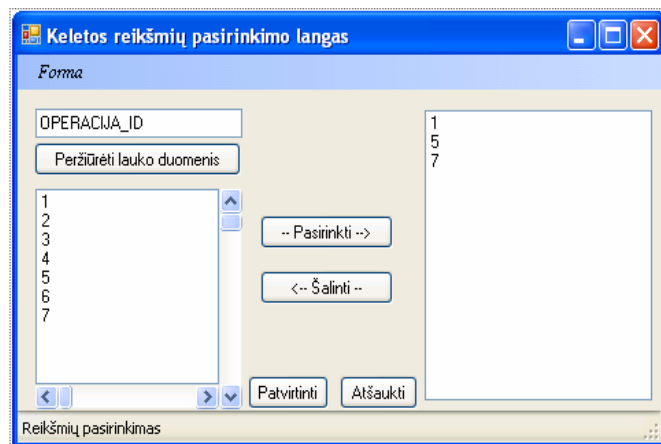
65 pav. Operacijų tvarkymo langas

Pasirenkamas palyginimo tipas: ar reikšmė, parametras, laukas, ar kita. Jei reikšmė – įvedama rankomis norima reikšmė. Jei parametras – tai perduodamas parametras iš veiklos procedūrų. Veiklos procedūros skirtos, tam tikrų skaičiavimų realizavimui. Sakykime, paskaičiuoti algą su priedu. Tai procedūroje bus realizuotas algoritmas, kuriam parametrus perduos apribojimų kūrėjas.

Jei laukas – tai bus pateikiamas to lauko reikšmės iš duomenų bazės. Reikšmę galima nuskaityti vieną (66 pav.) arba kelias (67 pav.), arba pasirinkti norimą intervalą (68 pav.), paskaičiuoti reikšmę procedūroje (69 pav.).

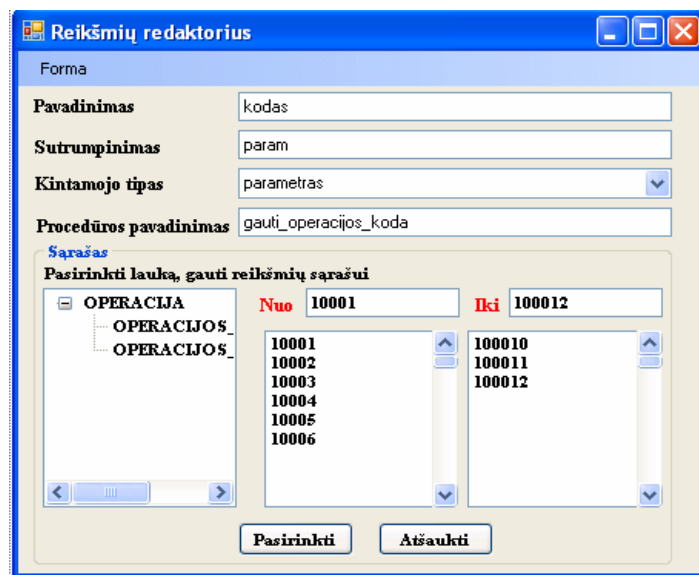


66 pav. Reikšmės nuskaitymo langas



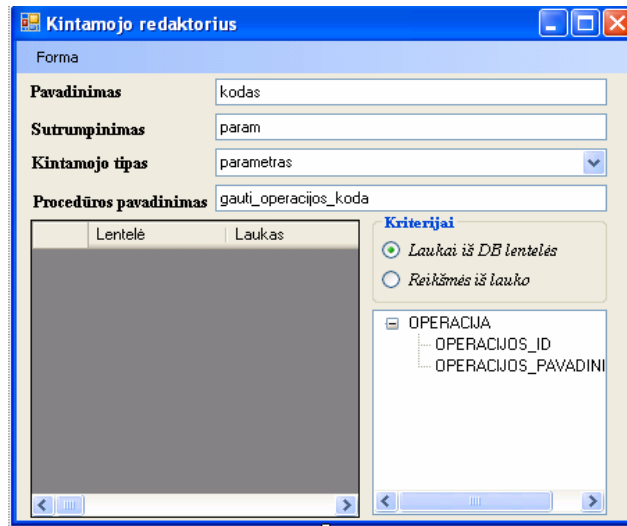
67 pav. Kelių reikšmių pasirinkimo langas

Sudėtiniam dinaminiam apribojimui įgyvendinti, naudojama forma (68 pav.).



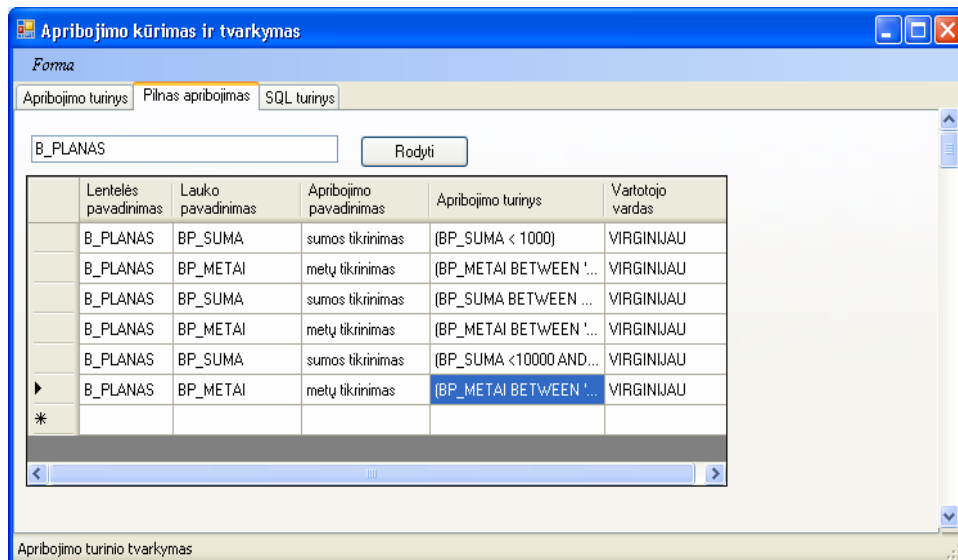
68 pav. Intervalo pasirinkimo, intervalo perdavimo į procedūrą langas

Dinaminiam apribojimui realizuoti, sukurta forma (69 pav.), kuri leidžia, aprašyti parametrus, jų tipus, procedūros pavadinimą.



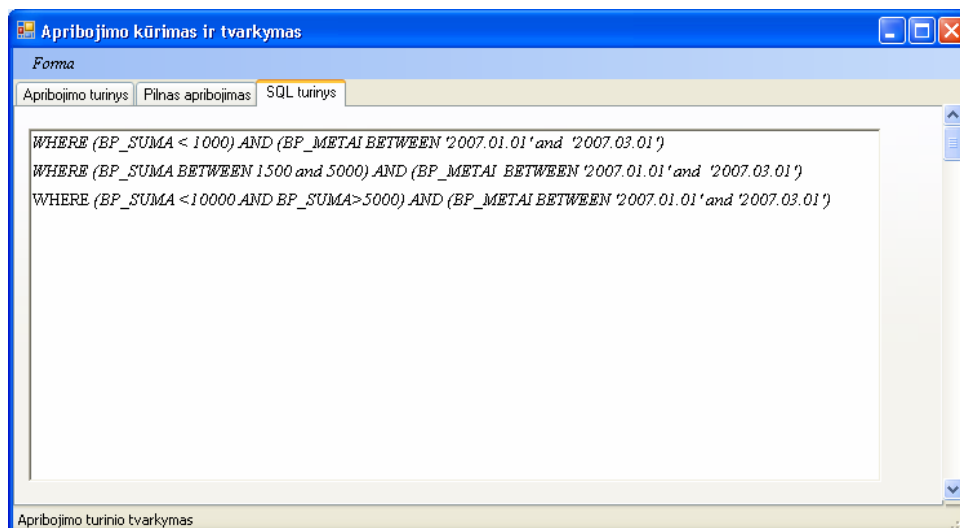
69 pav. Reikšmės pasirinkimo, perdavimo į procedūrą programos langas

Forma (70 pav.), skirta lentelės laukų apribojimų peržiūrai, paieškai pagal tam tikrą kriterijų atlikti.



70 pav. Pilno apribojimo peržiūros langas

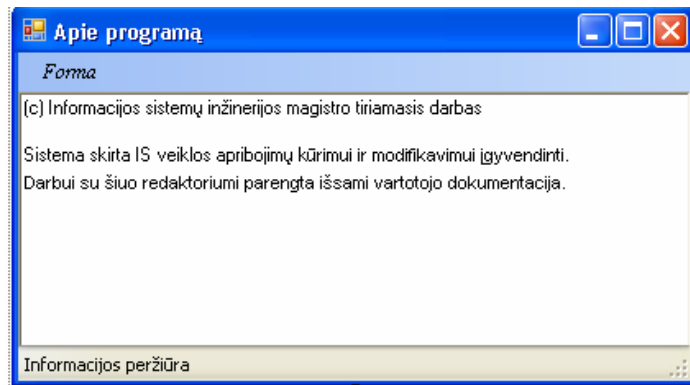
Forma (71 pav.) skirta kuriamo SQL apribojimo greitam peržiūrėjimui.



71 pav. SQL kodo peržiūros langas

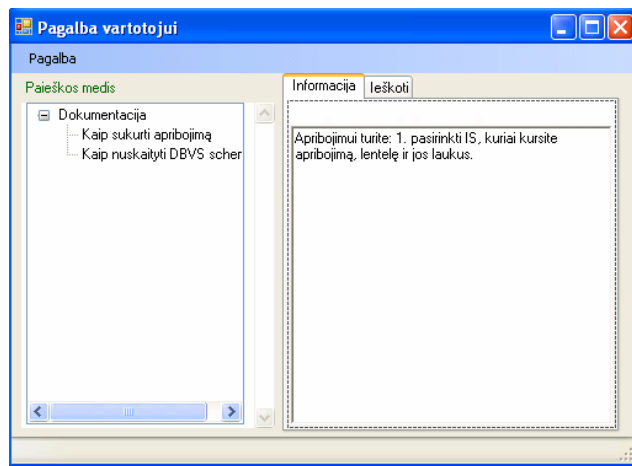
Pagalba sub-meniu punktai:

- Apie programą. Forma skirta supažinti vartotoją su sistemos galimybėmis (72 pav.).

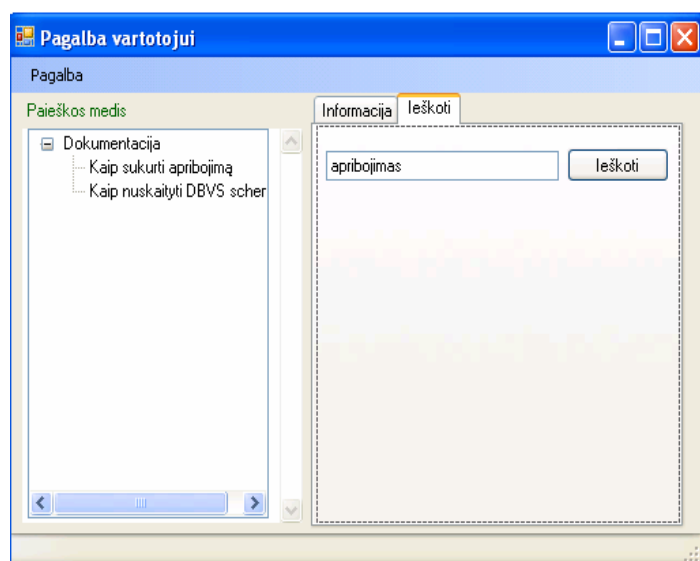


72 pav. IS redaktoriaus aprašymo langas

- Vartotojo pagalbos dokumentacijos langas (73 pav.) Patogiam informacijos paieškai atlikti, visa reikiama dokumentacija pateikiama nuorodomis Web naršyklės pagalba. Galima atlikti ir detalesnę paiešką pagal norimą kriterijų (74 pav.).



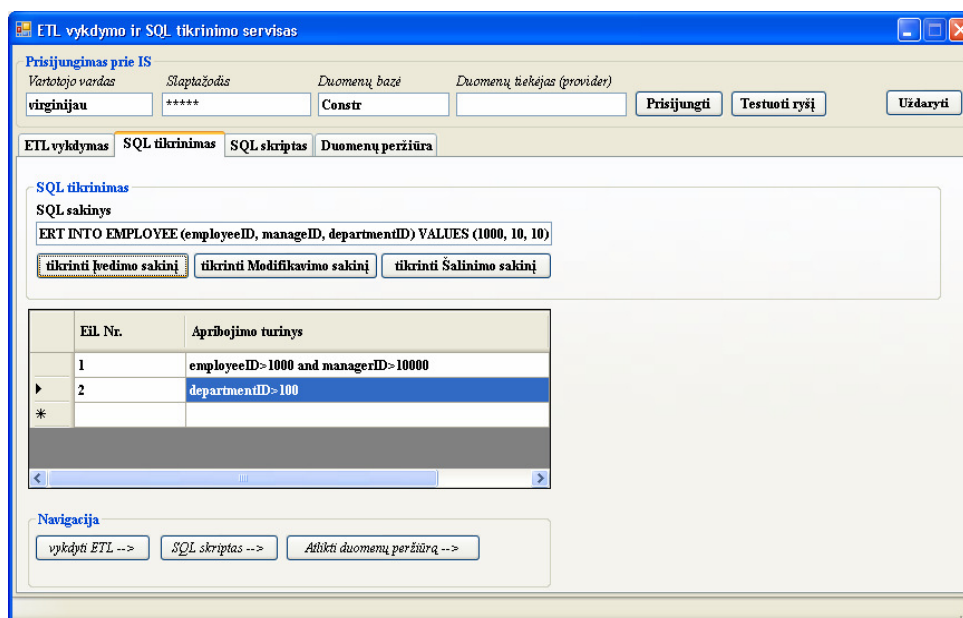
73 pav. Pagalbos langas



74 pav. Detalios vartotojo pagalbos paieškos langas

4.1.2. Apribojimų tikrinimo PĮ naudojimo aprašymas

Tikrinimo PĮ užtikrina, kad įvedami, modifikuojami, šalinami IS duomenys yra korektiški, teisingi, nėra prieštaravimų. Pagrindinis tikrinimo langas (75 pav.).



75 pav. SQL tikrinimo PĮ langas

Apribojimų pateikimo specialisto teises turintis vartotojas gali patikrinti ar norimi įvesti/redaguoti duomenys atitinka apribojimų meta DB saugojamus apribojimus. Tam reikalinga atlikti šiuos veiksmus:

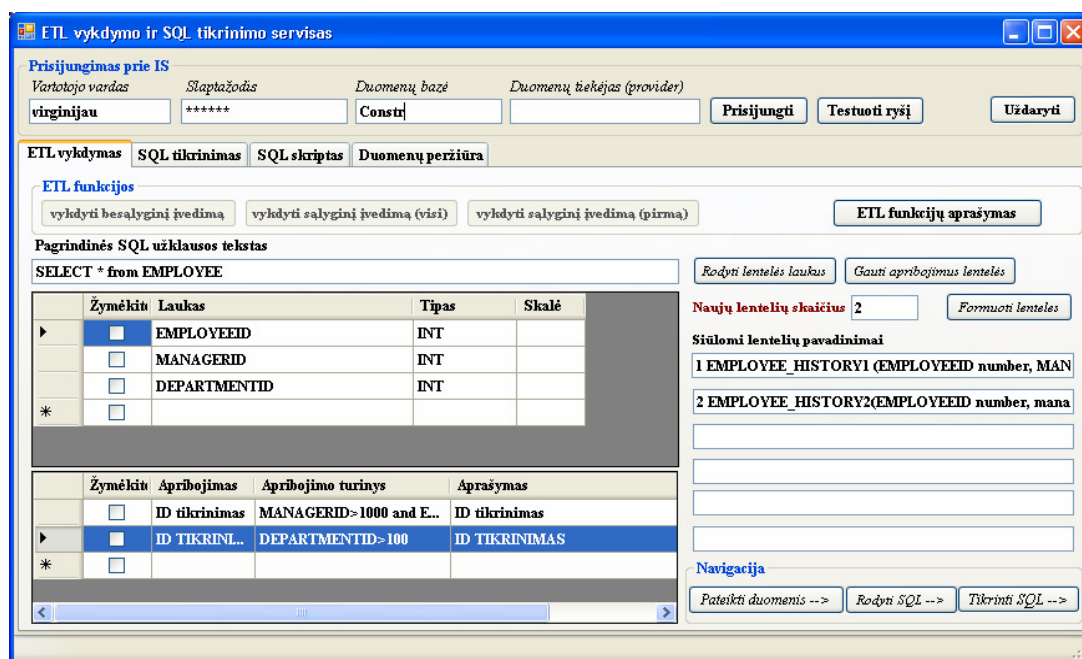
1. Prisijungti prie apribojimų meta – DB. Nurodomas prisijungimo vardas, slaptažodis ir DB vardas. Jei prisijungimo duomenys bus teisingi, vartotojas bus klausiamas, prie kokios IS bus norima jungtis. IS parametrai turi būti aprašyti IS parametrų formoje. Suteiktos galimybės vartotojui patikrinti ar geras ryšys su meta DB.
2. Vartotojas tekstiniame lange turi įvesti norimą tikrinti SQL sakinį.
3. Sekantis žingsnis – pasirinkti kokį tikrinimą atlieka (įvedimo, šalinimo, ar modifikavimo). SQL užklausa turi atitikti SQL standartą. (PĮ automatiškai patikrina, ar SQL sakiny atitinka standartą, ar nėra kokių sintaksinių klaidų. Taip pat į kokią lentelę norima atlikti veiksmą).
4. Toliau, jei prieš tai veiksmai atlikti teisingai, PĮ pateikia sąrašą apribojimų, jei tokie saugojami apribojimų meta DBVS saugykloje.
5. Jei užklausa duomenys atitinka apribojimų sąlygas, duomenys žiniatinklio servisu pagalba įvedami ar atliekami modifikavimai pasirinkus vieną iš

mygtukų: „tikrinti įvedimo sakinį“, „tikrinti modifikavimo sakinį“, „tikrinti šalinimo sakinį“.

SQL užklausų tikrinimo veiklos ir vykdymo sekų diagramos pateikiamos 3 priede. SQL tikrinimo projekto diagramos.

4.1.3. PĮ, įgyvendinančios apribojimų naudojimą Oracle ETL procese, aprašymas

PĮ (76 pav.) įgyvendina vartotojui galimybę atlikti Oracle DBVS saugojamų duomenų analizę, išsaugoti duomenis, tenkinančius apribojimus aprašytus meta DB į atskiras lenteles.



76 pav. Įvedimo į daugelį lentelių PĮ pagrindinis langas

Tam reikalinga atlikti tokius veiksmus:

1. Prisijungti prie apribojimų meta DB, nurodant vartotojo vardą, slaptažodį ir DB vardą. (Vartotojas turėti turėti prieigos teises, su apribojimų pateikimo specialisto role.) Jei autentifikacijos duomenys – teisingi, sistema paprašys nurodyti IS, iš kurios bus imami duomenys pavadinimą, iš saugojamo sąrašo. Vartotojui suteiktos galimybės testuoti PĮ su serveriu ryšį, išvedami atitinkami pranešimai, ar prisijungti galima ar ne.
2. Toliau tekstiniame lange vartotojas turi suvesti SQL užklausą, pagal kurią bus analizuojami ir imami iš DB lentelės duomenys, pasirinkdamas mygtuką „Rodyti lentelės laukus“. (PĮ automatiškai patikrina užklausos sintaksės teisingumą, jei teisinga - vartotojui pateiks analizuojamos lentelės laukus, duomenų tipus, tam kad būtų galima pačiam susikurti naujas lenteles išvengiant klaidų).

3. Vartotojui suteiktos galimybės peržiūrėti kokie tos lentelės apribojimai saugojami meta DB. Reikia tik pasirinkti mygtuką: „gauti apribojimus lentelės“. Sistema automatiškai pateiks tos lentelės apribojimų sąrašą: apribojimas, apribojimo turinys, aprašymas.
4. Tam, kad duomenys tenkantys tam tikrų apribojimų sąlygas būtų išsaugoti į atskiras lenteles, vartotojui reikia įvesti kiek naujų lentelių jam bus reikalinga. (Nurodomas skaičius.) Tuomet sistema automatiškai pateiks siūlomus pavadinimus. Pavadinimas suformuojamas iš lentelės pavadinimo, pridėdam žodelį „_history“ ir eilės numerį. Toks lentelių pavadinimų suformavimas leis greitai ir efektyviai nuspręsti, iš kokios lentelės imami duomenys.
5. Sekančiame žingsnyje vartotojui reikia nurodyti kokie laukai ir jų duomenų tipai bus reikalingi kiekvieno lentelėje. Kad išvengti klaidų – vartotojas turi pasirinkti iš pateikiamo sąrašo dvigubu pelės paspaudimu.
6. Vartotojas turi nurodyti kokie apribojimai bus reikalingi kiekvienai lentelei. Leidžiama pasirinkti iš celių sąrašo dvigubu pelės paspaudimu. (Tekstiniuose languose turi atsirasti skaičiai – apribojimo identifikatoriai).
7. Paskutinis žingsnis – nurodyti kokį veiksmą atlikti. T.y, ar sąlyginis – pirminis įvedimas, ar sąlyginis, ar besąlyginis.
8. Jei visi 1-7 veiksmai buvo atlikti kaip nurodyta, sistema atliks duomenų pagal apribojimus analizę, išsaugant juos į atitinkamas lenteles, leis atlikti peržiūrą, lange „Duomenų peržiūra“.

Įvedimo į kelias lenteles metodų veiklos diagramos pateikiamos 1 priede. ETL proceso įgyvendinimo Oracle9i DBVS veiklos diagramos, projekto sekų diagramos – 2 priede. ETL proceso įgyvendinimo Oracle9i projekto diagramos.

4.2. Vientisumo apribojimų valdymo sistemos eksperimentinis tyrimas

Eksperimente apribojimų redaktorių naudosime gamybinės, įmonės biudžeto valdymo IS apribojimų kūrimui ir modifikavimui įgyvendinti, bus patikrinta ar PĮ nuskaito Oracle, MySQL, MS SQL Server DBVS schemas.

Aprašyti IS apribojimai bus taikomi kitose veiklos procesuose: SQL tikrinimo ir ETL įgyvendinimo. Todėl šiame poskyriuje aprašomos eksperimento apimtys apribojimo turinio tvarkymui (35 lentelė), SQL įvedimo, šalinimo, redagavimo funkcijų teisingumo tikrinimui (36 lentelė), ETL daugelio įvedimo į lenteles (angl. multi – table insert) įgyvendinimui (37 lentelė).

ETL eksperimento metu, įvertinsime iš DBVS imamų ir analizuojamų duomenų pagal apribojimų sąlygas, laiko priklausomybę nuo įrašų kiekio, naudojant įprastinę „IF-THEN“ struktūrą ir Oracle9i DBVS teikiamas galimybes.

4.2.1. Vientisumo apribojimų redaktoriaus kokybės tyrimas

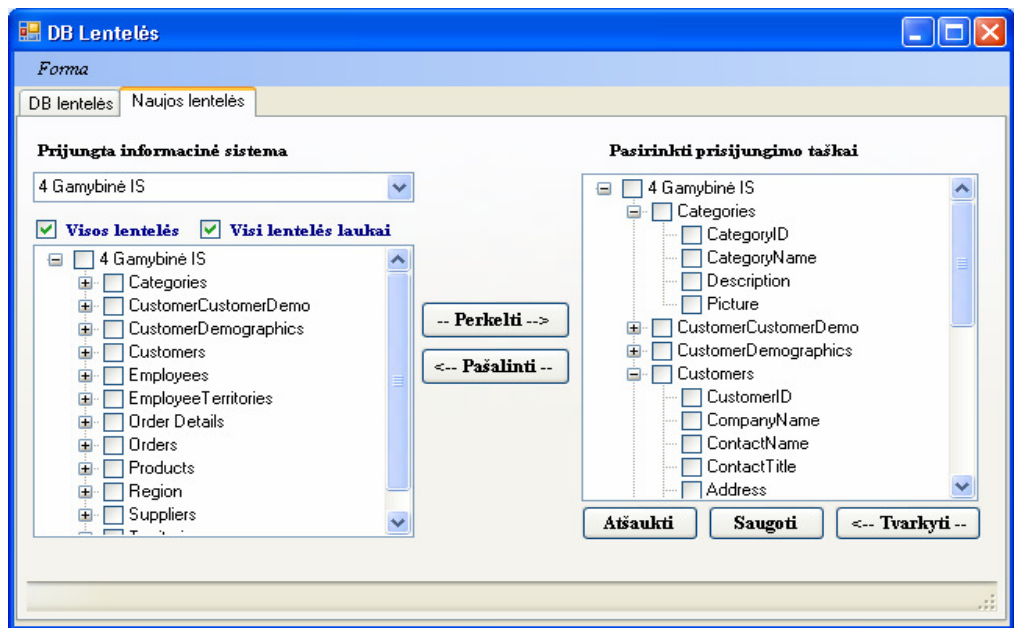
Tyrimas bus atliekamas šiais aspektais:

- Ar PĮ universali, t.y. ar nuskaito Oracle, MySQL, MS SQL Server DBVS schemas objektus (lenteles ir jų laukus).
- Leidžia kurti apribojimus bet kuriai minėtai DBVS patogioje grafinėje aplinkoje.
- Ar patogi vartotojo sąsaja.
- Ar sukurtas įrankis atitinka iškeltus kokybės reikalavimus, įgyvendina visus iškeltus tikslus.
- Suvedame ir išsaugojame IS, kurios DBVS – SQL Server parametrus formoje pavaizduotoje 77 pav.

77 pav. IS parametru suvedimo langas

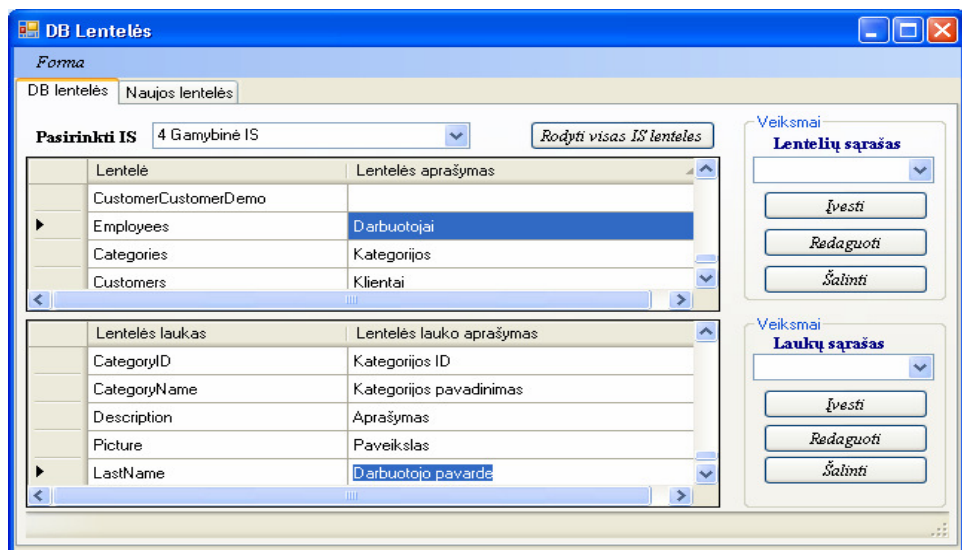
Tuomet iš meniu „Prisijungimo taškas“ pasirenkame punktą „Nuskaityti prisijungimo taškus“.

Formoje (78 pav.) pasirenkame įvestos IS pavadinimą, ir automatiškai turi būti nuskaityta jos schema. Kairėje matome nuskaitytas lenteles ir jų laukus. Tuomet galime pasirinkti norimas, lenteles ir jas perkelti į meta DB.



78 pav. Gamybinės IS schemos nuskaitymo pateikimo langas

Išsaugotas lenteles ir jų laukus galima peržiūrėti, papildyti meta informacija formoje (79 pav.).

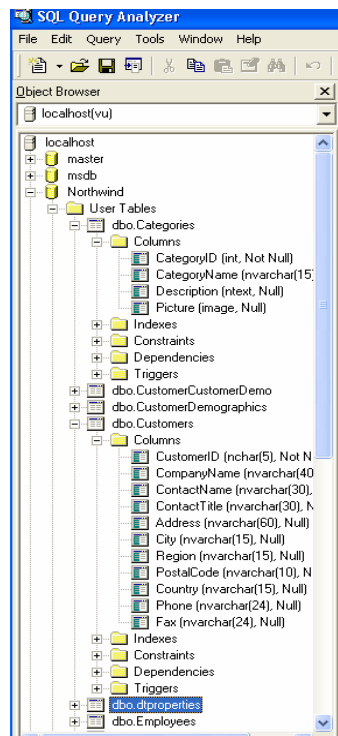


79 pav. Perkeltų lentelių ir jų laukų peržiūros ir tvarkymo langas

Pateiksime ir SQL Server prisijungimo langą (80 pav.) ir schemos, kurios objektus į meta DB perkėlėme vaizdą (81 pav.)



80 pav. Prisijungimo prie SQL Serverio langas



81 pav. SQL Server schemas naršyklės langas

Analogišku principu yra nuskaitomos ir MySQL, Oracle DBVS schemas.

Toliau pateikiami įgyvendinti eksperimentiniai pavyzdžiai, įmonės biudžeto valdymo IS.

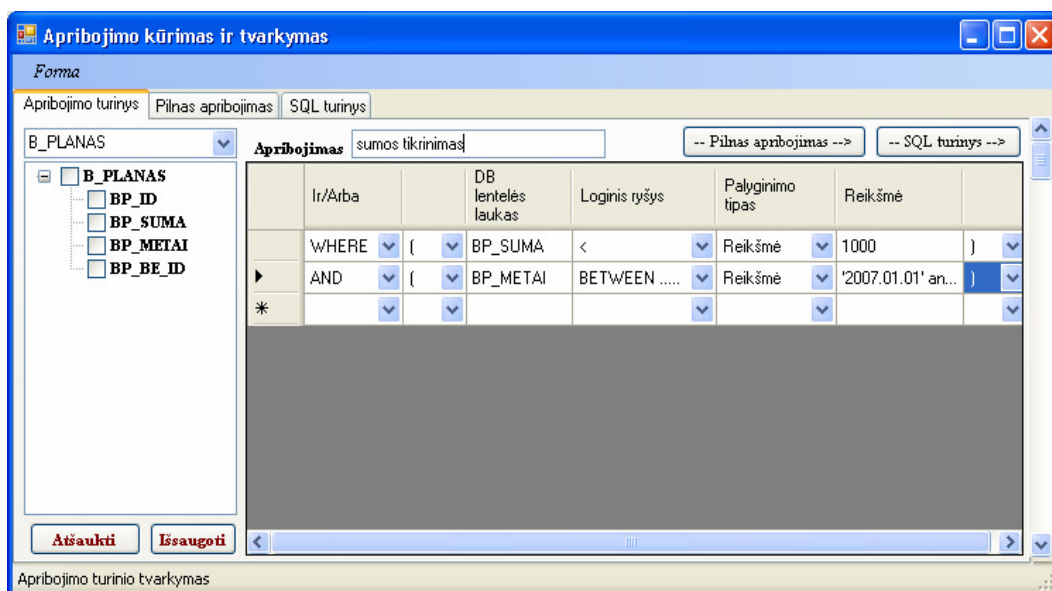
1. **Tikslas.** Sukurti apribojimą, kuris patikrintų kokioms biudžeto eilutėms yra skirta mažiau nei 1000 2007 metų I ketvirčiui.

2. **Vartotojas,** turintis Apribojimų pateikimo specialisto teises ir prisijungimo duomenis: virginijau, vestas.

Veiklos žingsniai:

1. pasirenkame biudžeto plano lentelę.
2. pasirenkame lauką BP_SUMA, dvigubu pelės spragtelėjimu.
3. pasirenkame „WHERE sąlygą“.
4. pasirenkame atidarymo skliaustą.
5. pasirenkame operatorių “<“.

6. nurodome, kad tai bus reikšmė, iš tos lentelės lauko sąrašo.
7. iš BP_SUMA lauko reikšmės pasirinkimo lango, pasirenkame 1000.
8. analogiškai žingsniais nurodome ir datos apribojimą, tik čia reikšmės parenkamos iš intervalo.
9. kartodami 1-7 žingsnius sukuriame dar 2 apribojimus, pavaizduojamus programos lange (82 pav.).



82 pav. Apribojimo sukūrimo langas

Toliau aprašomi pavyzdžiai kitos IS apribojimams sukurti.

1. **Tikslas.** Sukurti apribojimą, kuris užtikrintų gamybos IS operacijos atlikimo trukmę.
2. **Vartotojas,** turintis Apribojimų pateikimo specialisto teises ir prisijungimo duomenis: virginijau, vestas.

Veiklos žingsniai:

1. Susikurti naują procedūrą – „tikrinti_operacijos_trukmę“.
2. Nurodyti procedūros parametrus. Šiuo atveju bus – trukmė.
3. Išsaugoti apribojimo taikymo lentelės pavadinimą – „OperationType“;
4. Išsaugoti apribojimo taikymo lentelės laukų pavadinimus – „time“.
5. Tuomet kursime apribojimo elementą, kurio sudėtinės dalys:
 - ID (sistema automatiškai generuos)
 - Lauko „time“ ID.
 - Reikšmė arba reikšmių intervalas. Šiuo atveju bus reikšmės. Sakykime, kad time > 0 t.y. trukmė neturi būti neigiama ir negali būti =0. Norėdami pasirinkti operacijas, pirmiau turime pasirinkti operacijų klasifikatorių „Matematinės“ ir gausime >, =.

Apribojimo sąlygas sujungsime tarpusavyje, nes yra dvi sąlygos. Pasirenkame „Logines operacijas“ ir „and“ operatorių. Tuomet galutinė išraiška, kurią saugosime lentelėje „Pilnas apribojimas“ atrodys taip:

time > 0 and time <> 0.

1. **Tikslas.** Sukurti apribojimą, kuris užtikrintų gamybos IS atlyginimų teisingą įvedimą.
2. **Vartotojas,** turintis Apribojimų pateikimo specialisto teises ir prisijungimo duomenis: virginijau, virginijau.

Veiklos žingsniai:

6. Susikurti naują procedūrą – „tikrinti_algą“.
7. Nurodyti procedūros parametrus. Šiuo atveju bus – alga.
8. Išsaugoti apribojimo taikymo lentelės pavadinimą – „Person“;
9. Išsaugoti apribojimo taikymo lentelės laukų pavadinimus – „salary“.
10. Tuomet kursime apribojimo elementą, kurio sudėtinės dalys:
 - ID (sistema automatiškai generuos)
 - Lauko „salary“ ID.
 - Reikšmė arba reikšmių intervalas. Šiuo atveju bus reikšmių intervalas. Sakykime, kad salary > 0 t.y. alga neturi būti neigiama ir negali būti = 0. Taip pat, jos minimumas - 550 ir yra intervale nuo [550; 1800]. Norėdami pasirinkti operacijas, pirmiau turime pasirinkti operacijų klasifikatorių „Matematinės“ ir gausime >, =. Reikšmių intervale reikės nurodyti ribas, t.y., kad uždaras intervalas. Ribinės reikšmės priklauso intervalui.

Apribojimo sąlygas sujungsime tarpusavyje, nes yra 5 sąlygos. Pasirenkame „Logines operacijas“ ir „and“ operatorių. Tuomet galutinė išraiška, kurią saugosime lentelėje „Pilnas apribojimas“ atrodys taip:

salary > 0 and salary <> 0 and salary between (550 and 1800).

„Between“ operatorius intervalo pradžią ir pabaigą priskiria tam intervalui, tai nereikės atskirai nurodyti sąlygos, kad salary > 550.

Galime daryti išvadas, kad apribojimo aprašymas:

- patogus;
- vaizdus;
- minimizuotas klaidų kiekis. Nes vartotojui nurodoma kur, ką įvesti. Pateikiami reikšmių sąrašai.

Toliau tirsime ar apribojimų redaktorius atitinka iškeltus kokybės reikalavimus (32 lentelė).

Žymėjimų reikšmės:

“+” – atitinka.

“-” – neatitinka.

32 lentelės. Apribojimų redaktoriaus kokybės įvertinimas

Savybės	Kokybės savybių aprašymas	Patvirtinimas/Paneigimas
Korektiškumas	PĮ atitinka savo funkcinę specifikaciją.	+
Pilnumas	Realizuoti visi numatyti vientisumo apribojimai.	+
Patikimumas	Sukurta PĮ atlieka visas specifikacijoje numatytas funkcijas ir neatlieka nenumatytų funkcijų pasikeitus aplinkos veiksniams. (Įdiegus/pašalinus kitas sistemas)	+
Integralumas	PĮ užtikrina duomenų teisingumą vartotojui atlikus neleistinus veiksmus.	+
Panaudojamumas	<ul style="list-style-type: none"> ▪ Sistema yra intuityviai suvokiama. ▪ Lengva ir paprasta išmokti dirbti su ja. (Paprastumas leis vartotojams greičiau įsisavinti sistemą, pajusti jos galimybes, teikiamus privalumus) 	<p>+</p> <p>+</p>
Sąveikos su kitomis sistemomis galimybės	Lengva sistemą integruoti su kitomis PĮ.	+
Išbaigtumas	PĮ įgyvendina visus iškeltus tikslus.	+
Paprasta, logiška ir reikalavimus įgyvendinanti vartotojo sąsaja redaktoriui	Paprasta, aiški ir funkcionali vartotojo sąsaja leis vartotojui įgyvendinti norimą tikslą – įvesti, redaguoti duomenų bazės vientisumo apribojimus.	+
Aukštas sistemos naudojimo lygmuo.	Daug sistemos vartotojų.	Šiuo metu dar negalima patvirtinti, nei paneigti.
Palankus požiūris į IS turimas galimybes	Matoma reali įrankio ir gaunamo serviso nauda apribojimų valdymo, verslo procesuose.	+

Toliau poskyriuje tirsime tikrinimo serviso galimybes, ar užtikrina, kad į IS DB nebūtų įvesti apribojimų sąlygas netenkinantys duomenys.

4.2.2. Vientisumo apribojimų tikrinimo serviso kokybės tyrimas

Atliekamas eksperimentas, ar bus leista įvesti į gamybos IS, įrašą, kurio reikšmės netenkina, aprašytų meta DB sąlygų.

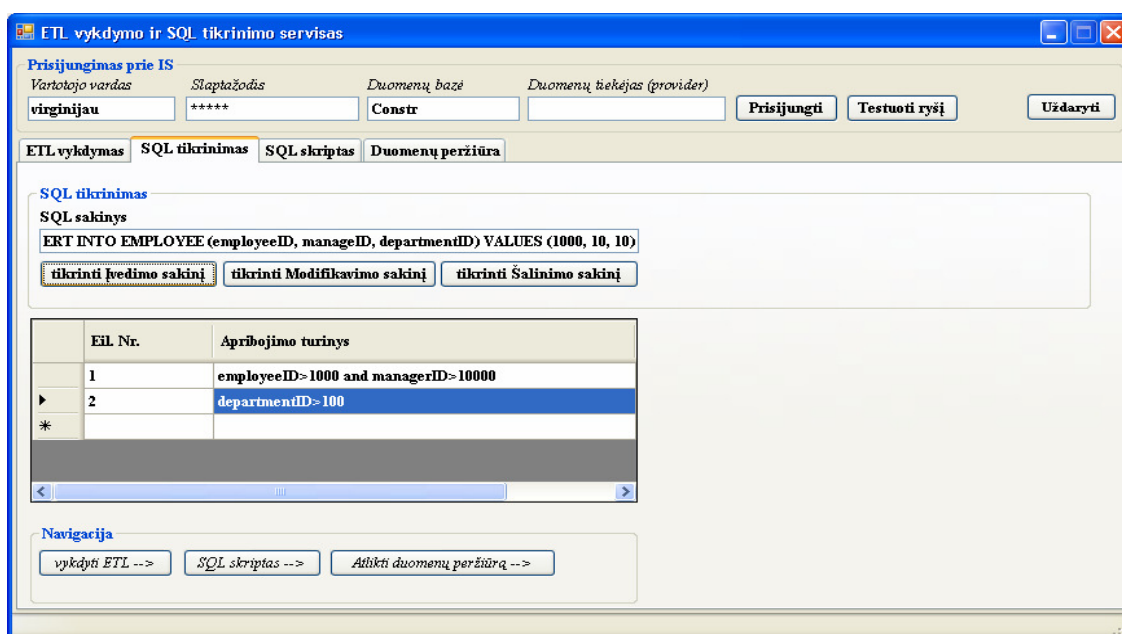
Prisijungėme prie tikrinimo PĮ, nurodydami prijungimo vardą, slaptažodį ir DB vardą.

Kadangi prisijungimo duomenys – teisingi, sistema leido prisijungti prie gamybinės IS. Pasirinkome SQL Insert sakinio vykdymą, kurio duomenų turėtų neleisti įvesti į DBVS.

SQL sakinio struktūra: INSERT INTO EMPLOYEES (employeeID, managerID, departmentID) values (1000, 10, 10). Kadangi meta DB aprašyti apribojimai:

Departamento ID turi būti > 1000, darbuotojo ID > 1000, vadovo ID > 1000, tai sistema užtikrino, kad vartotojo įvesti duomenys nepatektų į IS, vykdant tikrinimą ir įvedimą.

Programos langas pateikiamas 83 paveiksle.



83 pav. SQL DML funkcijų tikrinimo langas

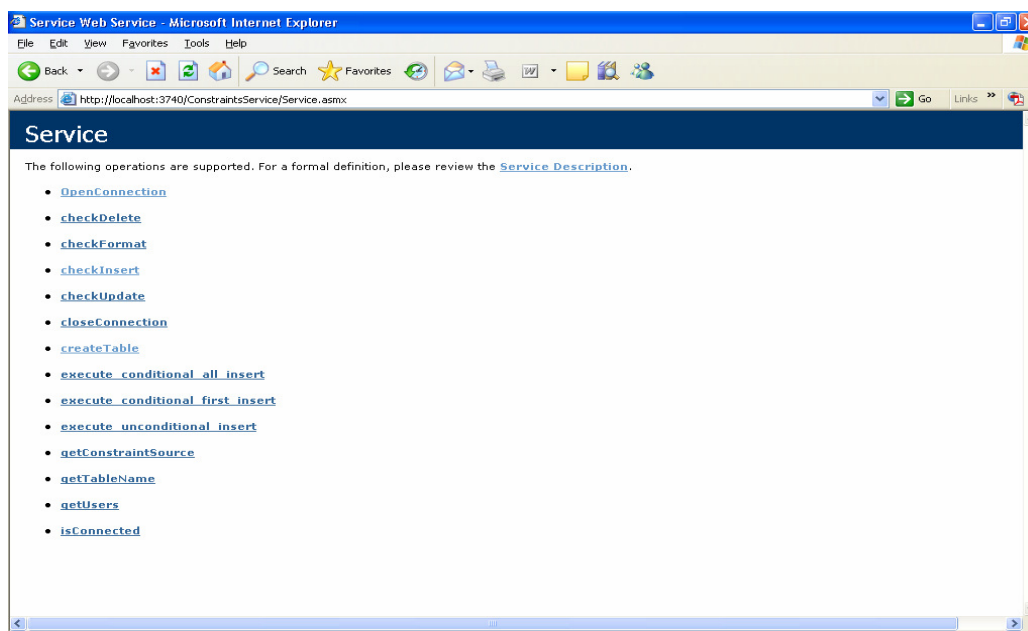
Išvada: sistema užtikrino duomenų nepatekimą į IS.

4.2.3. Vientisumo apribojimų panaudojimo ETL procese tyrimas

Šiame poskyriuje aprašomas tyrimas dviem aspektais:

- Kokybės;
- Vykdyto laiko priklausomybės nuo analizuojamų pagal apribojimų sąlygas duomenų kiekio ir naudojamos struktūros (angl. „IF-THEN“, „INSERT ALL“, „FIRST INSERT“).

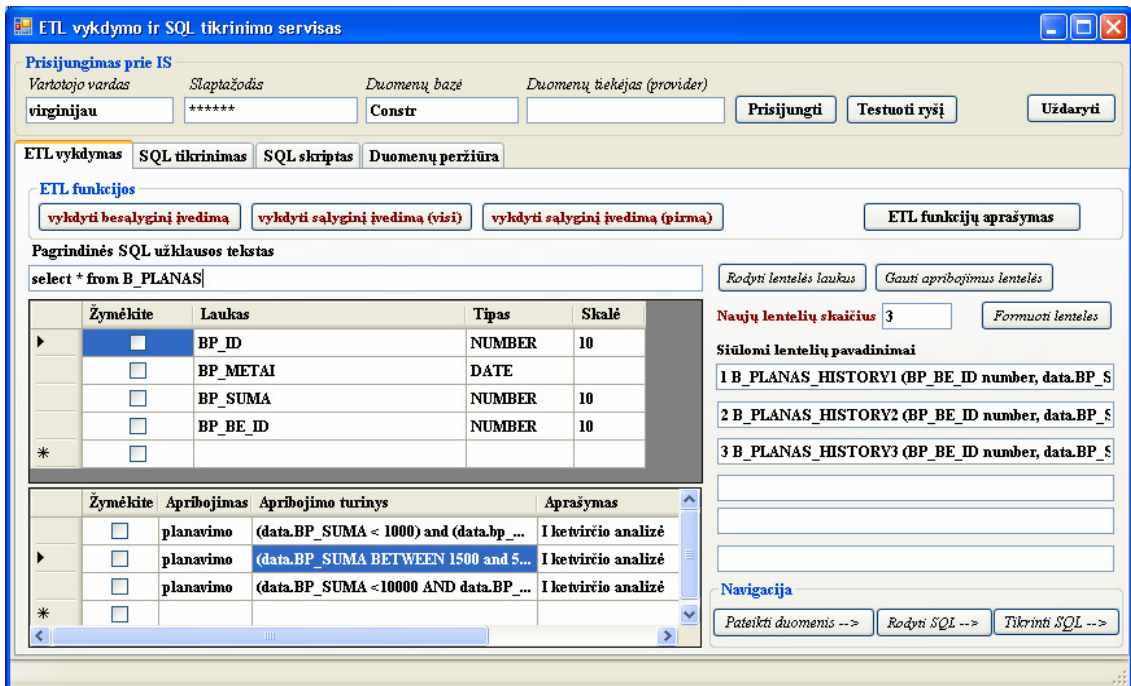
Pateikiamas (84 pav.) Web Servisų sukūrimo langas. Šios funkcijos panaudotos ETL ir SQL tikrinimo serviso PĮ, įgyvendinti.



84 pav. SQL DML funkcijų tikrinimo langas

Eksperimente patikrinsime, ar leidžia atlikti tam tikros duomenų lentelės analizę – t.y. pagal pasirinktus apribojimus, išrinkti duomenis ir juos išsaugoti atskirose lentelėse. Atlikus visus veiksmus, aprašytus poskyriuje 4.1.3:

- Prisijungiame prie programos (85 pav.) vardu: virginijau/vestas@Constr. Sistema paprašė nurodyti su kuria IS bus atliekamas eksperimentas, pasirinkome biudžeto planavimo sistemą.
- Įvedėme norimą užklausą . “Select * from B_PLANAS.” T.y. šios lentelės duomenys bus analizuojami.
- Pasirinkome “rodyti lentelės laukus.” Sistema pateikė sąrašą tos lentelės laukų ir jų duomenų tipus.
- Toliau pasirinkome mygtuką “gauti apribojimus lentelės”. Sistema išvedė visus tos lentelės apribojimus, saugojamus apribojimų meta DB.
- Duomenys bus keliami į 3 lenteles. Sistema sėmingai suformavo tos lentelės pavadinimus. Jai nurodome kokius laukus bus reikalingi ir kokius tai lentelei bus taikomi apribojimai. Skaičiukas priekyje žymi apribojimo identifikatorių.
- Toliau pasirinkome eksperimentui šiuos mygtukus, atitinkančius metodus: “vykdyti sąlyginį įvedimą (visi)”, “vykdyti sąlyginį įvedimą (pirma)” apribodami kas kart duomenų iš lentelės B_PLANAS įrašų kiekį (1000, 3000, 5000 ir t.t.).
- (Struktūrų vykdymo skriptai pateikiami 5 priede. Struktūrų palyginimo skriptai.)



85 pav. ETL eksperimento vykdymo pagrindinis langas

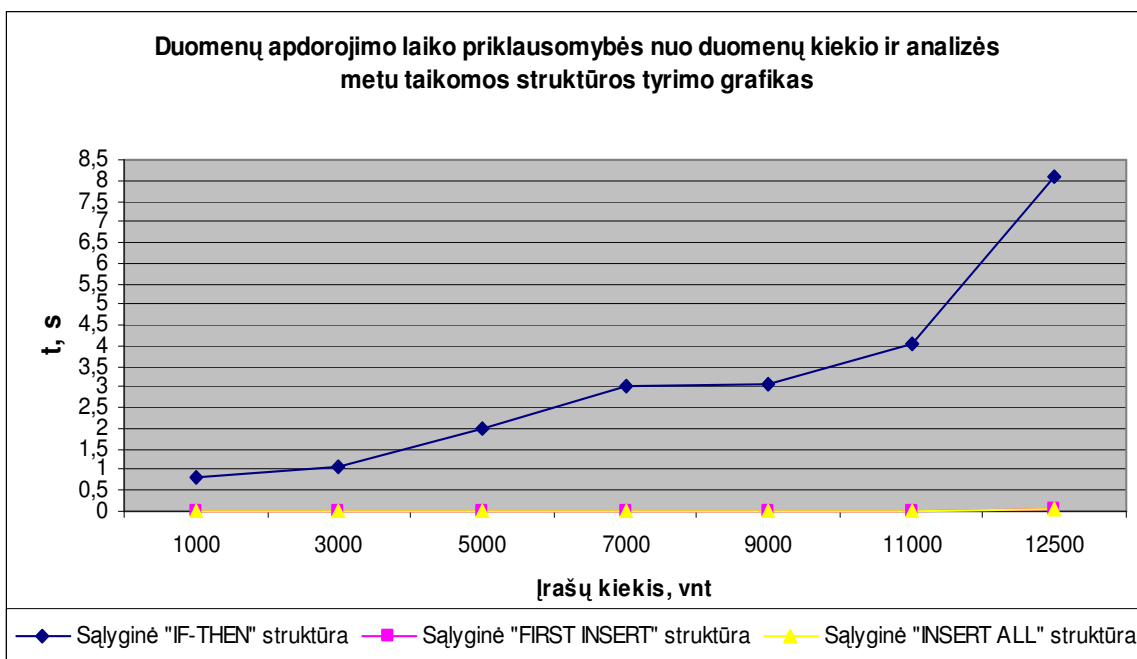
33, 34 pateikiami laiko priklausomybės nuo analizuojamų lentelės B_PLANAS (įmonės biudžeto valdymo IS) duomenų kiekio ir naudojamos struktūros, gauti rezultatai

33 lentelė, Laiko priklausomybės nuo analizuojamo duomenų kiekio, rezultatai

Įrašų kiekis, vnt	Struktūros panaudojimo vykdymo t, s	"IF-THEN"	"INSERT FIRST"	"INSERT-ALL"
		struktūra	struktūra	struktūra
1000		0,8	0,01	0,01
3000		1,05	0,01	0,01
5000		2,02	0,01	0,01
7000		3,02	0,02	0,01
9000		3,08	0,02	0,02
11000		4,03	0,02	0,03
12500		8,07	0,03	0,03

Gautus rezultatus atvaizduosime grafiškai (86-92 pav.), kur:

- X ašyje pažymėsime analizuojamų iš lentelės B_PLANAS įrašų kiekį. Analizuojama 12500 įrašų.
- Y ašyje – struktūros vykdymo laiką, sekundėmis, išskirdami kartu ir milisekundes (8.5 – tai 8 sekundės ir 5 milisekundes).

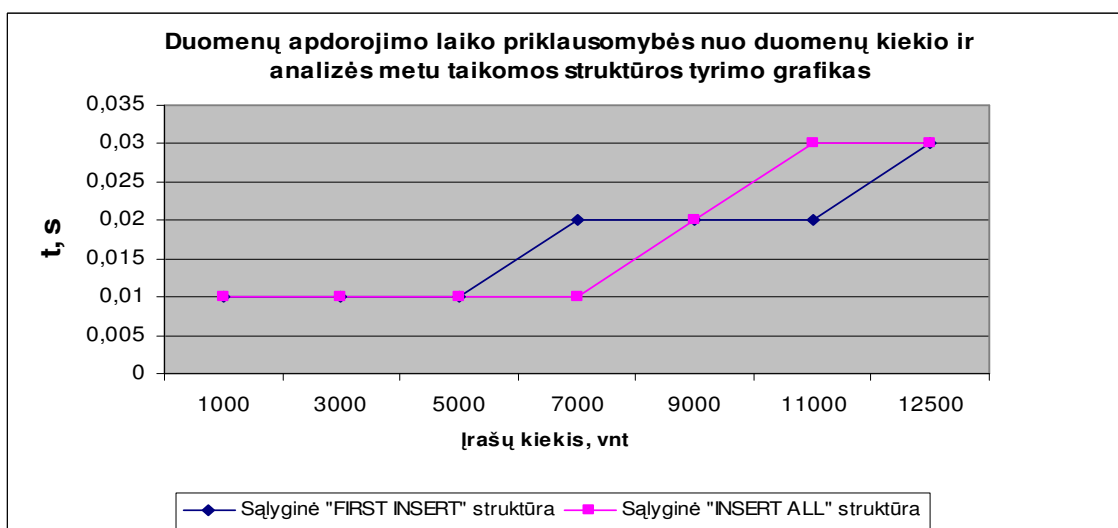


86 pav. Laiko priklausomybės nuo duomenų kiekio ir apdorojimo struktūrų grafikas

Iš 86 paveiksle pateikto grafiko galime daryti išvadą, kad norint atlikti duomenų apdorojimą pagal užsibrėžtus apribojimus, reikia pirmiausia:

- juos aprašyti ir saugoti centralizuotai;
- naudoti Oracle DBVS duomenų transformavimo į kelias lenteles teikiamas galimybes, kas leidžia ryškiai sumažinti vykdymo laiką.

Kadangi „IF-THEN“ struktūros naudojimo laikas ypač didelis lyginant su kitomis dviem struktūromis, toliau pateikiamas laiko vykdymo palyginimas tarp jų.

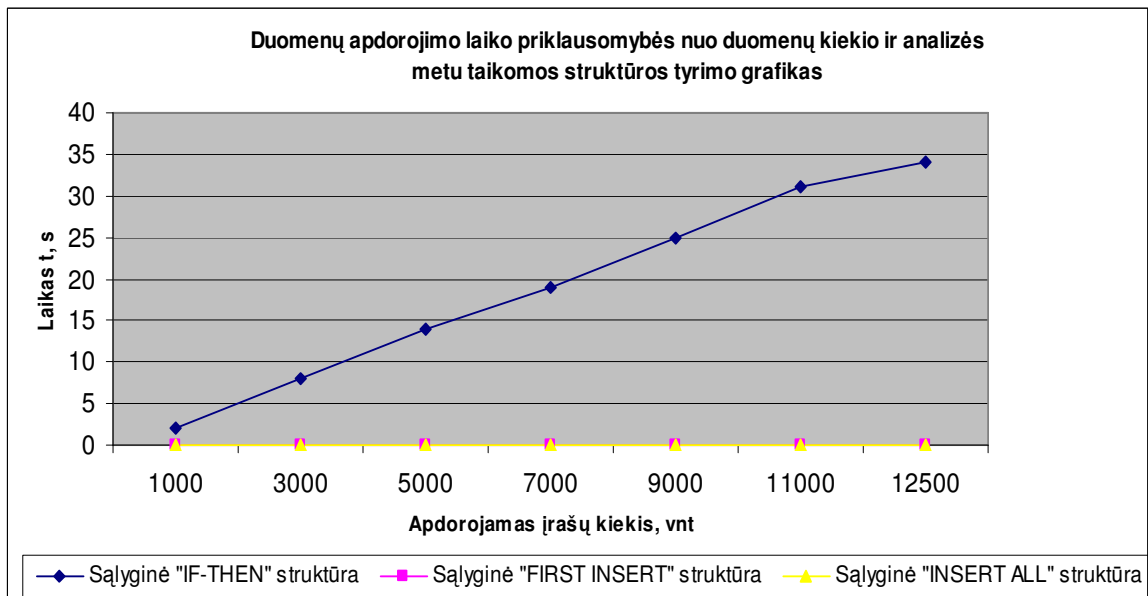


87 pav. Laiko priklausomybės nuo duomenų kiekio ir apdorojimo grafikas

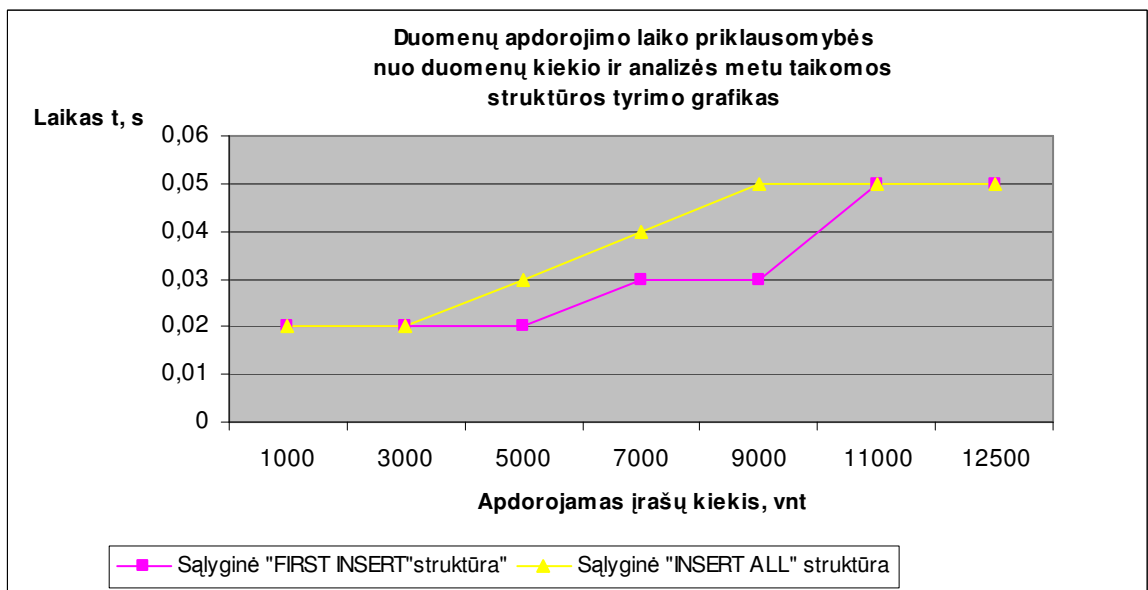
Pastebime, kad pastarųjų vykdymo laikas – labai mažas, t.y. net milisekundės.

Toliau eksperimentuojama su sudėtingesnėmis apribojimų struktūromis: ilgesniais biudžeto plano laikotarpiais. Tyrimo rezultatai pateikiami 34 lentelėje, grafikai – 88, 89 pav.

<i>Įrašų kiekis, vnt</i>	<i>Struktūros panaudojimo vykdymo t,s</i>	<i>"IF-THEN" struktūra</i>	<i>"INSERT FIRST" struktūra</i>	<i>"INSERT-ALL" struktūra</i>
1000	2,08	0,02	0,02	0,02
3000	8,02	0,02	0,02	0,02
5000	14,02	0,02	0,02	0,03
7000	19,08	0,03	0,03	0,04
9000	25,02	0,03	0,03	0,05
11000	31	0,05	0,05	0,05
12500	34,03	0,05	0,05	0,05



88 pav. Sudėtingesnių apribojimų struktūrų vykdymo laiko grafikas

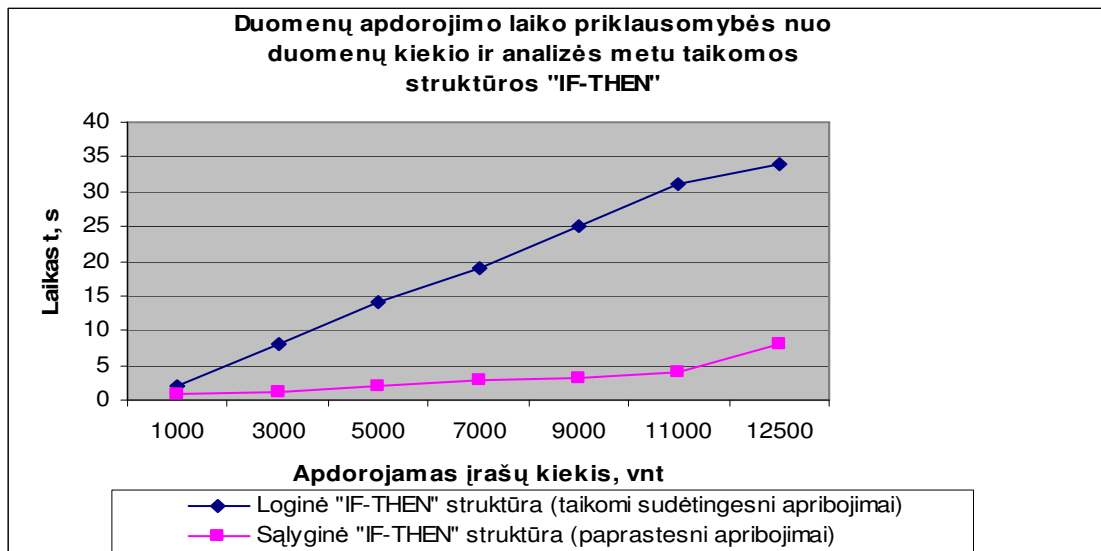


89 pav. Sudėtingesnių apribojimų, struktūrų vykdymo laiko grafikas

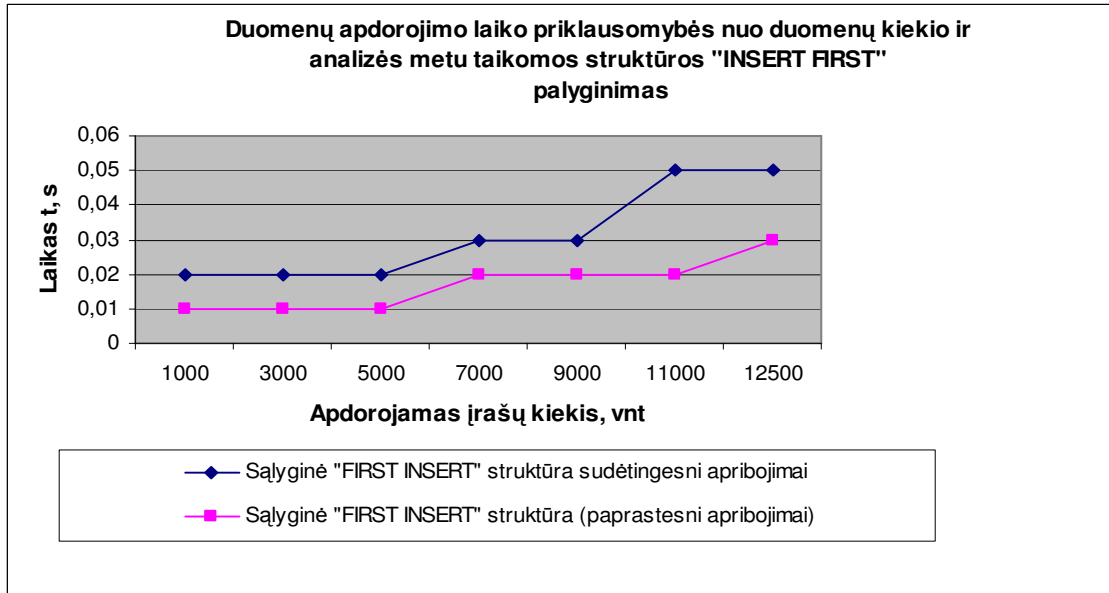
Iš grafikų 88, 89 pav. pastebime, kad Oracle struktūrų panaudojimas analizuojant duomenis su sudėtingesniais apribojimais aprašytais meta DB ir išsaugojant juos į atskiras lenteles, laikas nežymiai

padidėjo. O jau sąlyginės struktūros „IF-THEN“ laikas – padidėjo ryškiai. Todėl toliau grafikuose pateiksime abiejų eksperimentų metodų lyginamuosius grafikus (90-92 pav.).

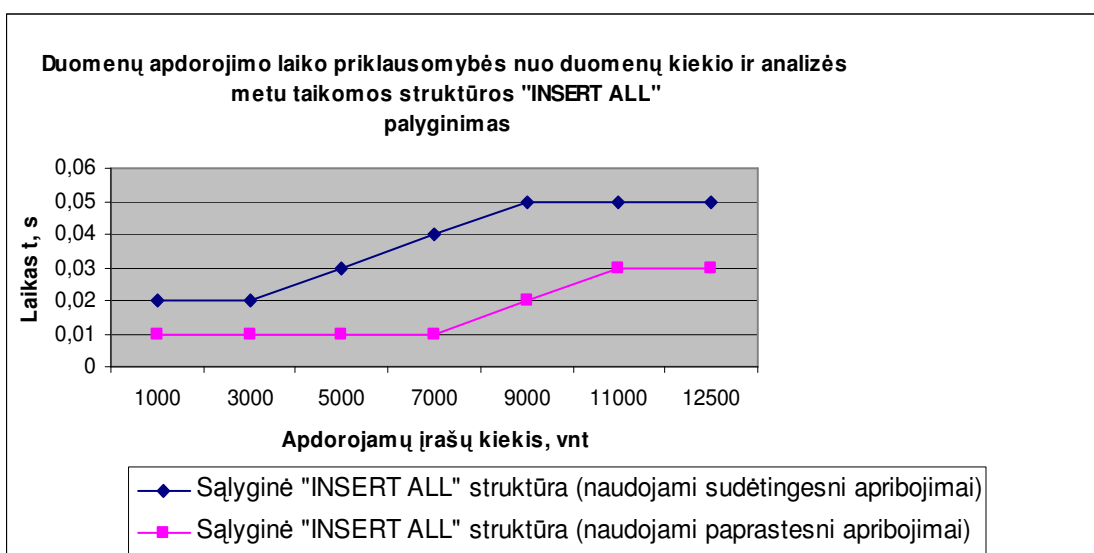
90 paveiksle – pateikiamas palyginimas tarp „IF-THEN“ struktūros naudojimo su paprastesniais ir sudėtingesniais apribojimais. Galime daryti išvadą, kad kuo sudėtingesni apribojimai, tuo vykdymo laikas didėja. Todėl verta išnagrinėti vykdymo laiko pokyčius (90 - 92pav.) tarp Oracle9i DBVS struktūrų galimybių.



90 pav. „IF-THEN“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas



91 pav. „INSERT FIRST“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas



92 pav. „INSERT ALL“ vykdymo laiko palyginimas su įvairiais apribojimais, grafikas

ETL proceso tyrimo išvados:

- Sistema veikia be klaidų (eksperimento apibendrinimas 35-37 lentelėse);
- Duomenys atrenkami pagal apribojimus, teisingai ir kokybiškai;
- Išnaudojamos visos Oracle DBVS naudojimo galimybės (sumažintas duomenų išsaugojimo į atskiras lenteles vykdymo laikas, kreipinių į duomenų šaltinius skaičius, sistemos apkrovimus.)
- Pateisino apribojimų valdymo sistemos kūrimas Oracle DBVS pagrindu. Įrodyta apribojimų saugyklos nauda apribojimo kūrimo, modifikavimo bei duomenų analizės procesuose.

35 lentelė. Apribojimo sukūrimo eksperimentinio tyrimo apimtis

EKSPERIMENTO KOMPONENTŲ/VEIKSMŲ SĄRAŠAS				
<i>Eil. Nr.</i>	<i>Vykdoma funkcija (eksperimento objektas)</i>	<i>Situacija prieš</i>	<i>Situacija po</i>	<i>Rezultatas</i>
1.	Pasirinkti tam tikros IS duomenų lentelę.	IS turi būti aprašyti parametrai meta-db. Turi būti geras ryšys su IS. Turi būti geras ryšys su meta-db.	Medžio struktūros komponente atvaizduojama lentelės struktūra – lentelės pavadinimas ir jos laukai.	Veikia.
2.	Pasirinkti norimą lauką, naudodami pelę (angl. double-click)	Sėkmingai nuskaityta lentelės struktūra iš meta-db.	Pasirinktas lentelės laukas atvaizduojamas celėje.	Veikia.
3.	Vartotojui turi būti pateiktas loginių ryšių sąrašas.	Sėkmingai pasirinktas db lentelės laukas, kuris atvaizduojamas celėje.	Pasirinkta operacija pateikiama celėje.	Veikia.
4.	Pasirinkti palyginimo tipą.	<ul style="list-style-type: none"> ▪ Jei vartotojas pasirenka „Reikšmė“ – tai pateikti vartotojui langą, kur pasirinktą tam tikro lauko reikšmę. Jei operacijos tipas („=“, „in“), tai vartotojui leisti pasirinkti kelias reikšmes. 	Suformuojama reikšmės ar reikšmių eilutė, kuri atvaizduojama celėje.	Veikia.

EKSPERIMENTO KOMPONENTŲ/VEIKSMŲ SĄRAŠAS				
<i>Eil. Nr.</i>	<i>Vykdoma funkcija (eksperimento objektas)</i>	<i>Situacija prieš</i>	<i>Situacija po</i>	<i>Rezultatas</i>
		<ul style="list-style-type: none"> ▪ Jei pasirinktas „Parametras“, tai vartotojui pateikti procedūrų pasirinkimo langą ir leisti suvesti reikiamus parametrus, reikšmės ar reikmių apskaičiavimui. ▪ Jei pasirinktas „Laukas“, tai vartotojui pateikti lauko reikšmių pasirinkimo langą. ▪ Jei vartotojas pasirenka „Kita“ vartotojui leisti suvesti norimas reikšmes ar kintamuosius. 	<p>Vartotojui apskaičiuojama reikšmė, tam tikros procedūros pagalba ir pateikiama vartotojui celėje.</p> <p>Vartotojui pateikiamas lauko reikšmės ar keletos reikšmių langas, vartotojas pasirenka norimą reikšmę ar kelias ir suformuota eilutė pateikiama celėje.</p> <p>Norimas kintamasis ar reikšmė atvaizduojama celėje.</p>	
5.	Pasirinkti skliaustus.	Vartotojui turi būti leista, bet kokiu momentu bet kokiam apribojimo elementui, jo sąlygoms uždėti skliaustus.	Uždėti skliaustai.	Veikia.
6.	Jei apribojimas sudėtinis, tai sekantys elementai turi prasidėti „and“, „or“ junginiais.	Pirmasis apribojimo elementas turi prasidėti „Where“ sąlyga, sekančios „and“, „or“.		Veikia.

36 lentelė. SQL funkcijų eksperimentinio tyrimo apimtis

EKSPERIMENTO KOMPONENTŲ/VEIKSMŲ SĄRAŠAS				
<i>Eil. Nr.</i>	<i>Vykdoma funkcija (eksperimento objektas)</i>	<i>Situacija prieš</i>	<i>Situacija po</i>	<i>Rezultatas</i>
1.	Įvesti SQL DML funkcija (INSERT, UPDATE, DELETE, INSERT).	Turėti teises darbu su apribojimų valdymo sistema. Geras redaktoriaus PĮ ryšys su IS.	Tikrinama ar atitinka SQL sakinių struktūrą.	Veikia.
2.	Iš SQL sakinio formuojamas lentelės pavadinimas	Įvesta teisinga DML funkcija.	Sistema pateikia lentelės pavadinimą.	Veikia.
3.	PĮ gražina tos lentelės apribojimų sąrašą.	Lentelė yra tam tikroje IS, kurios parametrai saugojami meta-db.	Pateikiamas vartotojui patogia forma apribojimų sąrašas.	Veikia.
4.	Iš SQL DML funkcijų suformuojamos loginės išraiškos.	Teisingos struktūros DML funkcijos.	SQL DML funkcija išskaidyta į logines išraiškas.	Veikia.
5.	Įstatomos reikšmės į apribojimo sąlygas.	Sėkmingai suformuotos loginės išraiškos.	Gaunama simbolinio tipo eilutė. Pasiruošiama tikrinimui.	Veikia.
6.	Atlikti Tikrinimą, spausdami mygtuką „Tikrinti“.		Jei apribojimus atitinka – leidžiam atlikti DML operaciją. Priešingu atveju – gražinamas „false“ pranešimas.	Veikia.
7.	Vartotojui turi būti pateikiamas pranešimas			Veikia.

	apie atliktą/neatliktą veiksmą.			
--	---------------------------------	--	--	--

37 lentelė. ETL įgyvendinimo eksperimentinio tyrimo apimtis

EKSPERIMENTO KOMPONENTŲ/VEIKSMŲ SĄRAŠAS				
<i>Eil. Nr.</i>	<i>Vykdoma funkcija (eksperimento objektas)</i>	<i>Situacija prieš</i>	<i>Situacija po</i>	<i>Rezultatas</i>
<i>Besąlyginis įvedimas</i>				
1.	Įvesti SQL DML komandą (INSERT, UPDATE, DELETE, tekstiniame lauke.	Sėkmingas prisijungimas prie sistemos. Geras PĮ ryšys su IS. Geras tinklo ryšys.	Paspausti įvedimo mygtuką. Atliekamas SQL DML funkcijos sintaksės tikrinimas. Išgaunamas lentelės pavadinimas. Pateikiamas tos lentelės laukų sąrašas.	Veikia.
2.	Įvesti naujų lentelių kieki.	Sėkmingai suformuotas naujų lentelių sąrašas su pavadinimų pabaiga - „history_nr“	Lentelių sąrašai atvaizduojami tekstiniuose laukuose.	Veikia.
3.	Leisti pasirinkti lentelėms norimus laukus.		Sukuriamos lentelės su norimais laukų pavadinimais.	Veikia.
4.	Vykdyti duomenų nukrovimą.		Duomenys sėkmingai išsaugojami norimoje lentelės struktūroje. Pateikiamas vartotojui pranešimas apie atliktą operaciją.	Veikia.
<i>Salvinis įvedimas, Salvinis – pirminis įvedimas</i>				
1.	Įvesti SQL DML komandą (INSERT, UPDATE, DELETE, tekstiniame lauke.	Sėkmingas prisijungimas prie sistemos. Geras PĮ ryšys su IS. Geras tinklo ryšys.	Paspausti įvedimo mygtuką. Atliekamas SQL DML funkcijos sintaksės tikrinimas. Išgaunamas lentelės pavadinimas. Pateikiamas tos lentelės laukų sąrašas.	Veikia.
2.	Įvesti naujų lentelių kieki.	Sėkmingai suformuotas naujų lentelių sąrašas su pavadinimų pabaiga - „history_nr“	Lentelių sąrašai atvaizduojami tekstiniuose laukuose.	Veikia.
3.	Leisti pasirinkti lentelėms norimus laukus.		Sukuriamos lentelės su norimais laukų pavadinimais.	Veikia.
4.	Pagal lentelės pavadinimą, išgauti iš meta – db apribojimų sąrašą.		Pateikiamas vartotojui patogia forma apribojimų sąrašas.	Veikia.
5.	Nurodyti kiekvienai lentelei norimus apribojimų elementus.	Suformuotas sąrašas apribojimo elementų. Sukurtos lentelės;		Veikia.
6.	Vykdyti duomenų nukrovimą.		Duomenys sėkmingai išsaugojami norimoje lentelės struktūroje. Pateikiamas vartotojui pranešimas apie atliktą operaciją.	Veikia.

Pateikiama struktūrizuota lentelė nefunkcinio testavimo būdų aprašymai, siūlomi metodai (38 lentelė) ir galimi eksperimento rezultatai (taip/ne).

„**Taip**“: **Teigiamas** – skirtas parodyti, kad sistema daro tai, kas nurodyta reikalavime.

„**Ne**“: **Neigiamas** – parodyti, kad sistema nedaro to, ko, remianti reikalavimu, neturėtų daryti.

38 Lentelė. Nefunkcinio testavimo technikos

Nefunkcinis testavimas	Reikalingumo aprašymas	Siūlomas metodas/būdas	Taip/Ne
<i>Suderinamumo testavimas</i>	Patvirtinti, kad sistema nedaro žalos kitoms suinstaliuotoms sistemoms ir atvirkščiai.	Dirbti su sistema ir fiksuoti ar sistema nepateikia tam tikrų sisteminių klaidos pranešimų.	NE
<i>Veikimo aplinkos testavimas</i>	Ar IS veikia įvairiose Windows operacinėse sistemose.	Tikrinti įvairiose Windows versijose (98, 2000, XP)	TAIP
<i>Apsaugos testavimas</i>	Atlikti priėjimo prie resursų kontrolę.	<i>Funkcijų lygmenyje</i> – ar vartotojui prieinamos tik tos funkcijos ir duomenys, kurie yra leidžiami tokio tipo vartotojui.	TAIP
		<i>Sistemos lygmenyje</i> – ar sistema prieinama tik tiems vartotojams, kuriems suteiktos tokios teisės.	TAIP
<i>Patogumo testavimas</i>	Patikrinti ar sistemos vartotojai norimus veiksmus gali atlikti efektyviai. („Sistema neveikia, jei aš negaliu atlikti norimo veiksmo“).	<ul style="list-style-type: none"> ▪ Formalus patogumo testavimo metodas – paprašyti vartotojo su sistema atlikti eilę veiklų. ▪ Patogumo peržiūra – patogumą įvertinti neįtraukiant sistemos vartotojų. 	TAIP
<i>Atsistatymo po klaidos testavimas</i>	Patikrinti, ar sistemos atsistatymas vyksta sklandžiai, ar pats procesas yra efektyvus.	Klaidos į sistemą įvedimas ir sistemos atsiskaitymo galimybių nustatymas.	TAIP

Sistemos testavimo tikslas – įvertinti, ar sistema bus priimta vartotojų ir išvengti klaidų, galinčių turėti neigiamą įtaką vartotojo funkciniai, finansinei veiklai ar įvaizdžiui.

Patvirtinimo testavimo tikslas – įvertinti, ar sistema atitinka veiklos reikalavimus ir patikrinti, ar sistema veikia teisi ngai ir yra patogi, prieš ją formaliai atiduodant vartotojui.

4.3. Vientisumo apribojimų valdymo sistemos perspektyvos

Vientisumo apribojimų valdymo sistemą, galėtume išplėsti šiuo funkcionalumu:

- automatizuoti didelio kiekio SQL užklausų tikrinimą, ar atitinka apribojimų meta DB saugojamus apribojimus.

- išnagrinėti kitų DBVS galimybes įgyvendinti ETL tiesiogiai per SQL, išvengiant daugelio SQL sakinių rašymo, kreipinių į DB serverį sumažinimo.

5. IŠVADOS IR REZULTATAI

- Esamų duomenų vientisumo apribojimų valdymo metodų ir įrankių analizė parodė, kad yra daug apribojimų realizavimo galimybių, tačiau trūksta vientisos apribojimų realizacijos metodikos.
- Sukurtas duomenų modelis leidžia struktūrizuoti ir centralizuotai saugoti įvairių tipų IS apribojimus.
- Remiantis nagrinėtais geriausių įrankių pavyzdžiais, suprojektuotas ir sukurtas apribojimų redaktorius, kuris leidžia:
 - kurti ir valdyti vientisumo apribojimus patogioje grafinėje formoje;
 - išgauti duomenis iš praktikoje paplitusių DBVS (Oracle, MS SQL Server, MySQL), naudojant vieningą schemas apdorojimo technologiją.
- Tikrinimo servisas centralizuotai užtikrina teisingą, vientisą duomenų įvedimą, redagavimą, šalinimą paskirstytose DBVS.
- Eksperimentiškai patikrinta, kad vientisumo apribojimus galima naudoti ne tik duomenų vientisumo užtikrinimo, bet ir išgavimo, transformavimo, įkėlimo (ETL) procese, išnaudojant Oracle9i DBVS teikiamą funkcionalumą. Laiko priklausomybės nuo duomenų kiekio ir struktūrų panaudojimo tyrimo metu įrodyta, kad Oracle9i savybių naudojimas yra:
 - Veiksmingas;
 - Efektyvus;
 - Išnaudojamos visos reikiamos DBVS galimybės, nei įgyvendinant šiuos procesus kitomis priemonėmis.
- Tyrimo metu patvirtinta, kad tikslinga vientisumo apribojimus saugoti atskirai nuo duomenų bazės schemas, ypač jei dirbama su paskirstytomis DBVS.
- Centralizuota duomenų vientisumo apribojimų realizacijos metodika buvo pristatyta 11-oje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinės Technologijos 2006“ ir išspausdinta straipsnyje konferencijos leidinyje.

LITERATŪRA

- [1] Miliauskaitė, Elita; Nemuraitė, Lina. Representation of integrity constraints in conceptual models // Informacinės technologijos ir valdymas = Information technology and control / Kauno technologijos universitetas. ISSN 1392-124X. 2005, T. 34, nr. 4. p. 355-365.
- [2] Nemuraitė, Lina; Miliauskaitė, Elita. Konceptualiojo modelio vientisumo apribojimų taksonomija // Informacinės technologijos 2005: konferencijos pranešimų medžiaga. T. 2. - ISBN 9955-09-788-4. - Kaunas, 2005, p. 552-565.
- [3] Saulius Kaukėnas, Darius Šilingas. UML panaudojimas IT specialistų darbo optimizavimui. “Informacinės technologijos verslui — 2003” pranešimų medžiaga, VU KHF, 2003 05 23, pp. 39-44.
- [4] Darius Šilingas. Programinės įrangos projektavimas naudojant UML. Prieiga per internetą: <http://www.bpi.lt/text.php?lang=1&item=210&arg=189>
- [5] Nagavali Pataballa, Priya Nathan. Learn Oracle from Oracle. Oracle9i: Program with PL/SQL, Volume 1. – 2001
- [6] Nagavali Pataballa, Priya Nathan. Learn Oracle from Oracle. Oracle9i: Program with PL/SQL, Volume 2. – 2001
- [7] Dynamic Constraints in MIDAS. Prieiga per internetą: <http://bdn.borland.com/article/26409>.
- [8] Redaktorių aprašymai. Prieiga per internetą: <http://www.sharewareconnection.com/titles/schema-editor.htm>
- [9] Conceptual Modeling and Markup Languages. Prieiga per internetą: <http://xml.coverpages.org/conceptualModeling.html>
- [10] XML representation of a relational database. Prieiga per internetą: <http://www.w3.org/XML/RDB.html>
- [11] Introducing Oracle Regular Expressions, An Oracle Technical White Paper, September 2003.
- [12] Sorting Your Linguistic Data inside the Oracle9i Database, An Oracle Technical White Paper, September 2002.
- [13] John Paul Cook. Build a .NET Application on the Oracle Database. Prieiga per internetą: http://www.oracle.com/technology/pub/articles/cook_dotnet.html
- [14] John Paul Cook. Securing a .NET Application on the Oracle Database. Prieiga per internetą: http://www.oracle.com/technology/pub/articles/mastering_dotnet_oracle/cook_masteringdotnet.html
- [15] Mark A. Williams. Cursor In, Cursor Out. Prieiga per internetą: <http://www.oracle.com/technology/oramag/oracle/06-jan/o16odpnet.html>
- [16] Mark A. Williams. The Values That Bind. Prieiga per internetą: <http://www.oracle.com/technology/oramag/oracle/05-sep/o55odpnet.html>

[17] Mark A. Williams. Optimize Result Set Retrieval Using ODP.NET and Ref Cursors. Prieiga per internetą:

http://www.oracle.com/technology/pub/articles/mastering_dotnet_oracle/williams_refcursors.html

[18] Podlipensky Paul Alexandrovich. .Net Regular Expressions. Prieiga per internetą:

<http://www.csharp-home.com/articles/184/>

[19] Shuxin Yin, Indrakshi Ray. Relational Database Operations Modeling with UML. 19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers) pp. 927-932.

[20] WebFOCUS sistemos dokumentacija. Prieiga per internetą.

https://wwws7.its.unc.edu/doc/pdf/wf_DDgt_53.pdf#search=%22Expression%20Biulder%20WebFocus%22

[21] Uzdnavičiūtė, Virginija. Centralizuotos duomenų vientisumo apribojimų realizacijos metodika // Informacinės technologijos 2006: 11-iosios tarpuniversitetinės doktorantų ir magistrantų konferencijos pranešimų medžiaga, II dalis [Kaunas, 2006 m. balandžio 28 d.]. - Vilnius, 2006, p. 127 – 132 nr. 2.

[22] Dina Goldin, Ayferi Kutlu, Mingjun Song, Extending the constraint database framework, Proceedings of the Paris C. Kanellakis memorial workshop on Principles of computing & knowledge: Paris C. Kanellakis memorial workshop on the occasion of his 50th birthday, p.42-54, June 08-08, 2003, San Diego, California, USA

[23] M. Iwaihara, M. Kozawa, J. Narazaki, Y. Kambayashi, „A System for Querying and Viewing Business Constraints,“ Proc. Int. Workshop on Rule Markup Languages for Business Rules on the Semantic Web, Michael Schroeder and Gerd Wagner (Eds.), Sardinia, Italy, June 14, 2002. CEUR-WS Prieiga per internetą: <http://CEUR-WS.org/Vol-60/>

[24] Alexander Brodsky, Victor E. Segal, Jia Chen, Pavla A. Exarkhopoulo, The CCUBE constraint object-oriented database system, Proceedings of the 1999 ACM SIGMOD international conference on Management of data, p.577-579, May 31-June 03, 1999, Philadelphia, Pennsylvania, United States.

[25] Stephane Grumbach, Philippe Rigaux, Luc Segoufin, The DEDALE system for complex spatial queries, Proceedings of the 1998 ACM SIGMOD international conference on Management of data, p.213-224, June 01-04, 1998, Seattle, Washington, United States.

[26] M. Iwaihara, „Matching and Deriving Dynamic Constraints for E-Commerce Negotiations,“ Proc. Workshop on Technologies for E-Services, Co-located with VLDB2000, Cairo, Sep. 2000.

[27] M. Iwaihara, „Supporting Dynamic Constraints for Commerce Negotiations,“ 2nd Int. Workshop in Advanced Issues of E-Commerce and Web-Information Systems (WECWIS), IEEE Press, pp. 12--20, June 2000.

[28] Chutiporn Anutariya, Somchai Chatvichienchai, Mizuho Iwaihara, Vilas Wuwongse, and Yahiko Kambayashi, „A Rule-Based Access Control Model,“ Proc. 2nd Int. Workshop on Rules and Rule

Markup Languages for the Semantic Web, RuleML 2003, Sanibel Island, Springer LNCS2876, pp. 35-48, Oct. 2003.

[29] No Magic, Inc. MagicDraw UML. Prieiga per internetą: <http://www.magicdraw.com/>.

[30] Įrankis SQL* Plus. Prieiga per internetą: <http://orafaq.com/faqplus.htm>.

[31] Įrankis MySQL Query Browser. Prieiga per internetą: <http://dev.mysql.com/doc/query-browser/en/>.

[32] Expression Filter. Prieiga per internetą:

http://www.oracle.com/technology/products/database/rules_manager/pdf/twp_AppDev_ExpressionFilter_10gR2.pdf

TERMINŲ IR SANTRUMPŲ ŽODYNAS

DBVS – duomenų bazių valdymo sistema.

RDBMS (angl. *Relational Database Management System*) – reliacinė duomenų bazių valdymo sistema.

DUOMENŲ ANOMALIJA (angl. *Data Anomaly*) – tai vienareikšmiškai negalima nustatyti duomenų.

DUOMENŲ ATNAUJINIMO ANOMALIJA – tai duomenų prieštarumas, atsirandantis dėl duomenų pertekliaus, atnaujinus tik dalį jų.

DUOMENŲ PAŠALINIMO ANOMALIJA – nenumatytas duomenų praradimas, susijęs su kitų duomenų pašalinimu.

DUOMENŲ ĮVEDIMO ANOMALIJA – nebuvimas galimybės įvesti duomenis dėl to, kad trūksta kitų duomenų.

SQL (angl. *Structured Query Language*) – struktūrizuota užklausų kalba.

PĮ – programinė įranga.

UML (angl. *Unified Modeling Language*) – unifikauta modeliavimo kalba.

GUI (angl. *Graphical User Interface*) – grafinis vartotojo interfeisas.

IT (angl. *Information Technology*) – informacinės technologijos.

IS (angl. *Informatikon System*) – informacijos sistema.

DB (angl. *Database*) – duomenų bazė.

CASE (angl. *Computer Aided Software Engineering*) – integruota IS kūrimo aplinka.

RUP (angl. *Rational Unified Process*) – PĮ kūrimo procesas.

OCL (angl. *Object Constraint Language*) – objektų apribojimų aprašymo kalba.

DML (angl. *Data Manipulation Language*) – duomenų manipuliavimo kalba.

PL/SQL (angl. *Procedural Language/SQL*) – standartizuota reliacinių duomenų bazių procedūrinė kalba, SQL išplėtimui.

ETL (angl. *Extract, Transform, Load*) – duomenų saugyklų procesas, duomenų į/iš DBVS išrinkimui, transformavimui ir užkėlimui atlikti.

CDBs (angl. *Constraint Databases*) – apribojimų duomenų bazės.

CQA (angl. *Constraints Query Algebra*) – apribojimų užklausų algebra.

CDA (angl. *Constraints Dinamic Algebra*) – apribojimų dinaminė algebra.

ODP.NET (angl. *Oracle Data Provider*) – Oracle DBVS duomenų tiekėjas, skirtas .NET programinei įrangai.

GIS (angl. *Geographical Information Systems*) – geografinės informacinės sistemos.

J2EE (angl. *Java 2, Platform, Enterprise Edition*) – standartinė programų kūrimo Java kalba platforma.

DNF (angl. *Disjunctive Normal Form*) – apribojimų modeliavimo taškų aibės simboliu santykiu forma.

BNF (angl. *Boyso Normal Form*) – Boyso kodo normalinė forma.

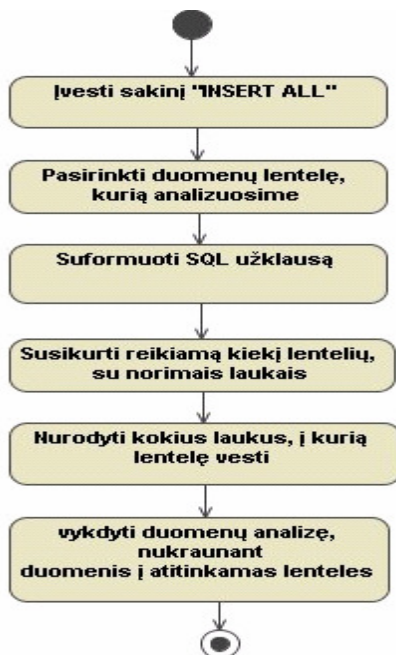
MVC (angl. *Model-View-Controller*) – architektūros šablonas.

DBA (angl. *Database Administrator*) – DB administratorius, turintis aukščiausias DBVS administravimo teises.

PRIEDAI

1 PRIEDAS. ETL proceso įgyvendinimo Oracle9i DBVS veiklos diagramos

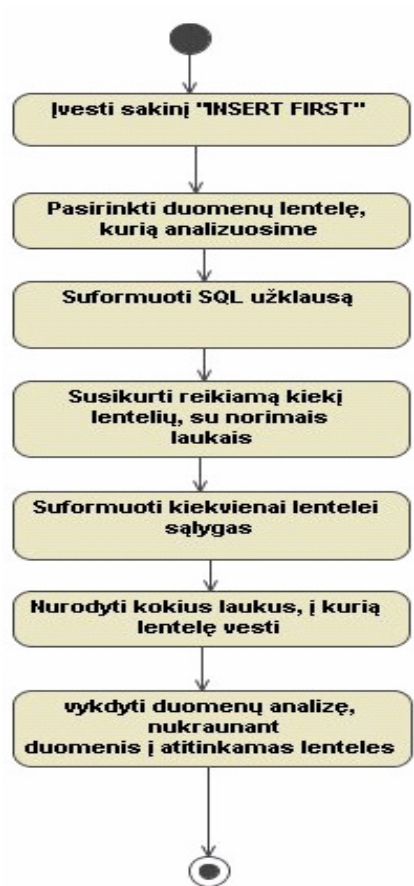
93, 94, 95 paveiksluose pateikiamos besąlyginio, sąlyginio, sąlyginio-pirminio įvedimo veiklos diagramos, norint įgyvendinti ETL procesą, išvengiant daugelio SQL sakinių rašymo. Apribojimų sąlygos nurodytai lentelei gaunamos iš centralizuotos apribojimų meta DB.



93 pav. Besąlyginio įvedimo į kelias lenteles veiklos diagrama



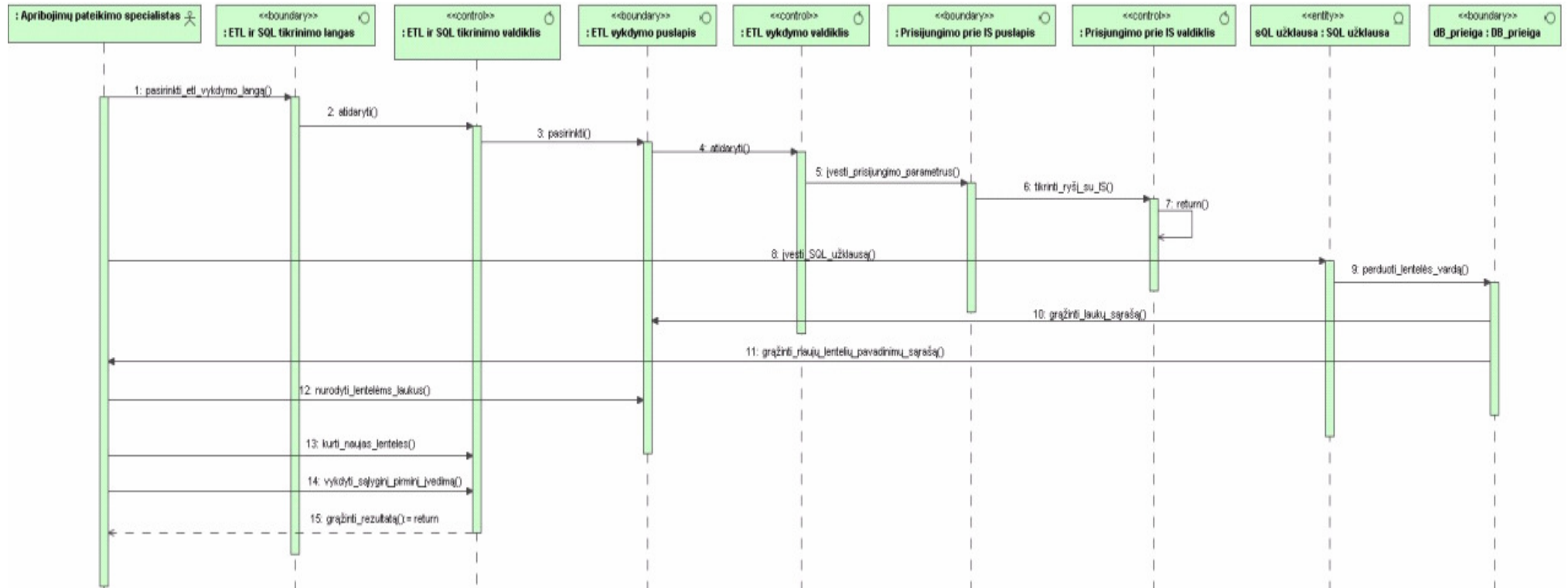
94 pav. Sąlyginio įvedimo į kelias lenteles veiklos diagrama



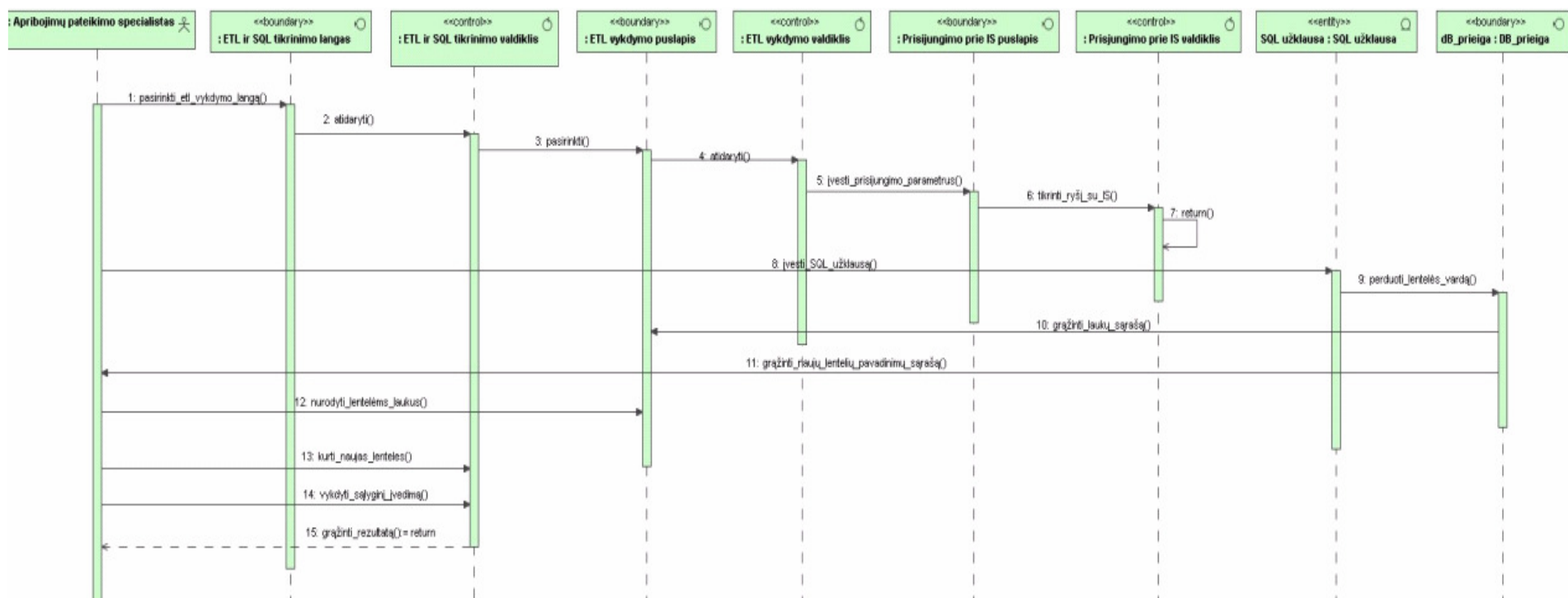
95 pav. Sąlyginio pirminio įvedimo į kelias lenteles veiklos diagrama

2 PRIEDAS. ETL proceso įgyvendinimo Oracle9i projekto diagramos

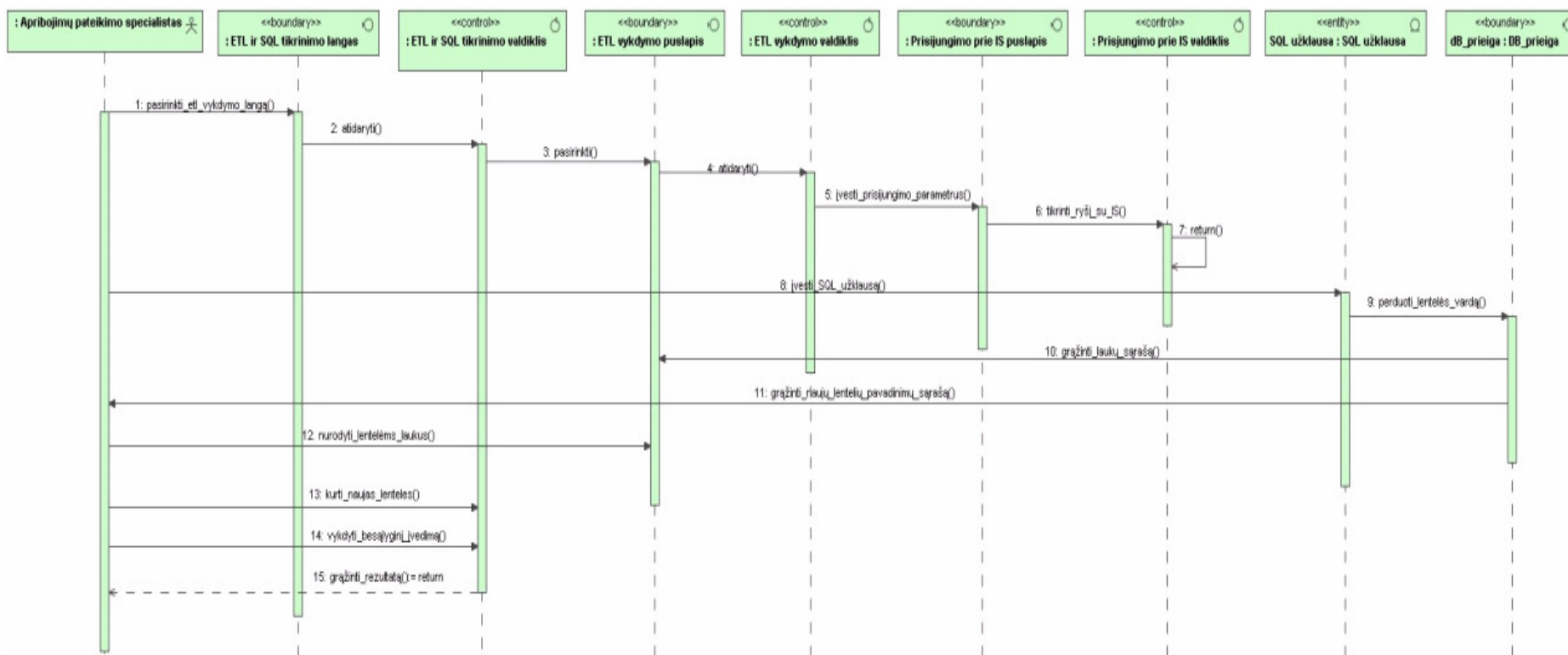
Šiame priede pateikiamos projekto sekų diagramos (96, 97, 98 pav.), įvedimo į daugelį lentelių realizavimui.



96 pav. Sąlyginio pirminio įvedimo į kelias lenteles projekto sekų diagrama



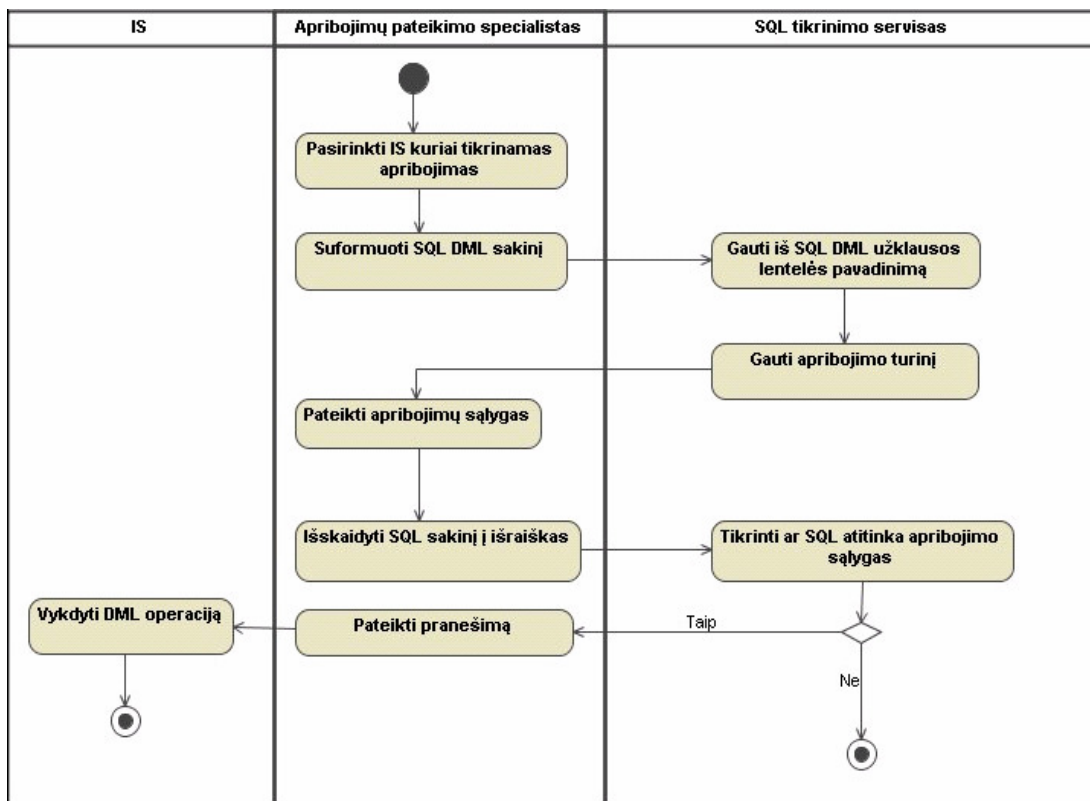
97 pav. Sąlyginio įvedimo į kelias lenteles projekto sekų diagrama



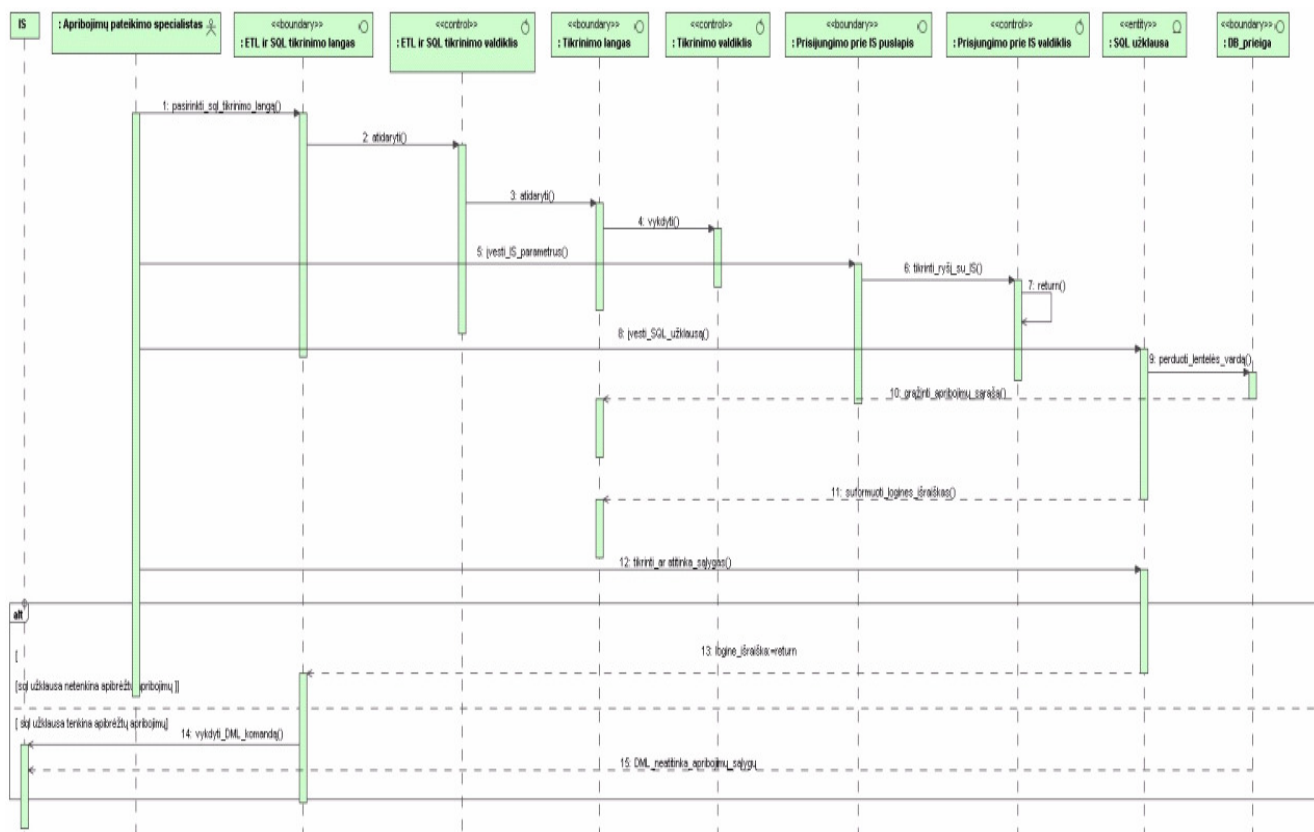
98 pav. Besąlyginio įvedimo į kelias lenteles projekto sekų diagrama

3 PRIEDAS. SQL tikrinimo projekto diagramos

Priede pateikiamos DML sakinių tikrinimo veiklos (99 pav.) ir projekto sekų diagramos (100 pav.).



99 pav. SQL tikrinimo veiklos diagrama



100 pav. SQL tikrinimo projekto sekų diagrama

4 PRIEDAS. Vientisumo apribojimų meta DB lentelių aprašymas

Pateikiamas vientisumo apribojimų valdymo sistemos meta - duomenų bazės schemas lentelių aprašymas, naudojant Oracle SQL duomenų tipus.

Žymėjimas:

✓ – kriterijų atitinkantis laukas.

T – kriterijų atitinkantis laukas. Atitikmuo „Taip“.

N – kriterijų atitinkantis laukas. Atitikmuo „Ne“.

39 lentelė – „Vartotojas“. Paskirtis – saugoti duomenis apie vartotoją.

39 lentelė. Vartotojo lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_VARTOTOJAS	Number(10)	✓	T	Vartotojo unikalus identifikatorius
VARDAS	Varchar2(75)		T	Vartotojo prisijungimo vardas prie apribojimų redaktoriaus
SLAPTAŽODIS	Varchar2(75)		T	Vartotojo prisijungimo slaptažodis prie apribojimų redaktoriaus
SMAILAS	Varchar2(100)		N	Vartotojo elektroninio pašto adresas

40 lentelė – „Vartotojų teisės“. Paskirtis – saugoti duomenis apie vartotojų teises.

40 lentelė. Vartotojo teisių lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_VARTOTOJU_TEISES	Number(10)	✓	T	Vartotojo teisių unikalus identifikatorius
FK_ID_VARTOTOJAS	Number(10)		T	Vartotojo unikalus identifikatorius – išorinis raktas
FK_ID_TEISES	Number(10)		T	Teisių unikalus identifikatorius – išorinis raktas

41 lentelė – „Teisės“. Paskirtis – saugoti informaciją apie teises.

41 lentelė. Teisių lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_TEISES	Number(10)	✓	T	Teisės unikalus identifikatorius
KODAS	Varchar2(150)		T	Teisės kodas
TEISES_PAV	Varchar2(50)		T	Teisės pavadinimas

42 lentelė – „DB“. Paskirtis – saugoti prijungtų informacinių sistemų parametrus.

42 lentelė. DB lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_DB	Number(10)	✓	T	Duomenų bazės unikalus identifikatorius
PAVADINIMAS	Varchar2(50)		T	Duomenų bazės pavadinimas
IP_ADRESAS	Varchar2(50)		T	Duomenų bazės IP adresas
PORTAS	Number(5)		T	Porto numeris
APRAŠYMAS	Varchar2(150)		N	Duomenų bazės paskirties, pastabų laukas

43 lentelė – „DB_lentelė“. Paskirtis – saugoti prijungtų informacinių sistemų duomenų bazių objektų, t.y. lentelių informaciją.

43 lentelė. IS DB lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_DB_LENTELE	Number(10)	✓	T	Duomenų bazės lentelės identifikatorius
LENTELES_PAV	Varchar2(50)		T	Duomenų bazės lentelės pavadinimas
APRASYMAS	Varchar2(50)		N	Duomenų bazės lentelės paskirties aprašymo laukas
FK_ID_DB	Number(10)		T	Duomenų bazės identifikatorius – išorinis raktas

44 lentelė – „Laukas“. Paskirtis – saugoti prijungtų informacinių sistemų duomenų bazių objektų, t.y. lentelių laukų informaciją.

44 lentelė. Lauko lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_LAUKAS	Number(10)	✓	T	Duomenų bazės lentelės lauko unikalus identifikatorius
LAUKO_PAV	Varchar2(100)		T	Duomenų bazės lentelės lauko pavadinimas
APRASYMAS	Varchar2(100)		T	Duomenų bazės lentelės lauko paskirties aprašymo laukas
FK_ID_DB_LENTELE	Number(5)		T	Duomenų bazės lentelės unikalus identifikatorius – išorinis raktas

45 lentelė – „Apribojimo elementas“. Paskirtis – saugoti prijungtų informacinių sistemų duomenų bazių objektų, t.y. lentelių laukų informaciją.

45 lentelė. Apribojimo elemento lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_APRIBOJIMO_ELEMENTAS	Number(10)	✓	T	Apribojimo elemento unikalus identifikatorius
FK_ID_LAUKAS	Number(10)		T	Duomenų bazės lentelės lauko unikalus identifikatorius – išorinis raktas
FK_ID_REIKSME	Number(10)		T	Duomenų bazės lentelės lauko reikšmės unikalus

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
				identifikatorius – išorinis raktas

46 lentelė – „Procedūra“. Paskirtis – saugoti sukurtų procedūrų pavadinimus duomenų bazėje.

46 lentelė. Procedūros lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_PROCEDURA	Number(10)	✓	T	Procedūros unikalus identifikatorius
PROCEDŪROS_PAV	Varchar2(75)		T	Procedūros pavadinimas
FK_ID_VARTOTOJAS	Number(10)		T	Vartotojo unikalus identifikatorius – išorinis raktas

47 lentelė – „Operacija“. Paskirtis – saugoti apribojimų operacijas, jų turinį.

47 lentelė. Operacijos lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_OPERACIJA	Number(10)	✓	T	Apribojimo operacijos unikalus identifikatorius
OPERACIJOS_PAV	Varchar2(75)		T	Apribojimo operacijos pavadinimas
OPERACIJOS_TURINYS	Number(10)		T	Apribojimo operacijos turinys
APRASYMAS			N	Apribojimo operacijos aprašymas
FK_ID_OPERACIJU_KLASIFIKACIJA			T	Operacijų klasifikatoriaus unikalus identifikatorius – išorinis raktas

48 lentelė – „Operacijų klasifikacija“. Paskirtis – saugoti operacijų klasifikatorius.

48 lentelė. Operacijų klasifikatoriaus saugojimo lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_OPERACIJU_KLASIFIKACIJA	Number(10)	✓	T	Operacijų klasifikacijos unikalus identifikatorius
PAVADINIMAS	Varchar2(50)		T	Operacijos klasifikatoriaus pavadinimas
APRASYMAS	Varchar2(150)		N	Operacijos klasifikatoriaus aprašymas

49 lentelė – „Parametrai“. Paskirtis – saugoti procedūrų parametrus.

49 lentelė. Parametrų lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_PARAMETRAI	Number(10)	✓	T	Procedūros parametrų unikalus identifikatorius
VARDAS	Varchar2(50)		T	Parametro pavadinimas
REIKSME	Varchar2(150)		T	Parametro reikšmė
APRASYMAS	Varchar2(150)		N	Parametro paskirties aprašymas
FK_ID_PROCEDURA	Number(10)		T	Procedūros unikalus identifikatorius – išorinis raktas

50 lentelė – „Pilnas apribojimas“. Paskirtis – saugoti sudėtinį vientisumo apribojimą.

50 lentelė. Pilno apribojimo lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_PILNAS_APRIBOJIMAS	Number(10)	✓	T	Apribojimo unikalus identifikatorius
APRIBOJIMO_TURINYS	Varchar2(255)		T	Apribojimo turinys
APRASYMAS	Varchar2(150)		T	Apribojimo paskirties aprašymas
FK_ID_VARTOTOJAS	Number(10)		T	Vartotojo unikalus identifikatorius – išorinis raktas
FK_ID_APRIBOJIMO_ELEMENTAS	Number(10)		T	Apribojimo elemento unikalus identifikatorius – išorinis raktas
FK_ID_LAUKAS	Number(10)		T	Lauko unikalus identifikatorius – išorinis raktas

51 lentelė – „Reikšmė“. Paskirtis – saugoti vientisumo apribojimo reikšmes.

51 lentelė. Reikšmės lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_REIKSME	Number(10)	✓	T	Reikšmės unikalus identifikatorius
REIKSME	Varchar2(150)		N	Reikšmės turinys
FK_ID_APRIBOJIMO_ELEMENTAS	Number(10)		T	Apribojimo elemento unikalus identifikatorius – išorinis raktas

52 lentelė – „Reikšmių intervalas“. Paskirtis – saugoti vientisumo apribojimo reikšmių intervalus.

52 lentelė. Reikšmių intervalo saugojimo lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_REIKSMIU_INTERVALAS	Number(10)	✓	T	Reikšmių intervalo unikalus identifikatorius
PRADŽIA	Varchar2(150)		N	Reikšmės pradžios turinys
PABAIGA	Varchar2(150)		N	Reikšmės pabaigos turinys
ATVIRAS	Number(10)		N	Intervalo atvirumo laukas
UZDARAS	Number(10)		N	Intervalo uždarumo laukas
FK_ID_REIKSME	Number(10)		T	Reikšmės unikalus identifikatorius – išorinis raktas

53 lentelė – „Apribojimo ryšys“. Paskirtis – saugoti sudėtinių vientisumo apribojimų sujungimo duomenis.

53 lentelė. Apribojimo ryšio saugojimo lentelės struktūros aprašymas

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
ID_APRIBOJIMO_RYSYS	Number(10)	✓	T	Apribojimo ryšio

Lentelės laukas	Duomenų tipas	Pirminis raktas	Privalomas?	Aprašymas
				unikalus identifikatorius
KAIREAPRIBOJIMOID	Varchar2(150)		N	Apribojimo jungimo iš kairės simbolis
DESINEAPRIBOJIMOID	Varchar2(150)		N	Apribojimo jungimo iš dešinės simbolis
RYSIO_OPERACIJOS_KODAS	Varchar2(150)		N	Operacijos turinys
KAIREAPRIBOJIMOIDREC	Number(10)		N	
DESINEAPRIBOJIMOIDREC	Number(10)		N	
FK_ID_APRIBOJIMO_RYSYS	Number(10)		T	Apribojimo ryšio unikalus identifikatorius – išorinis raktas
FK_ID_APRIBOJIMO_ELEMENTAS	Number(10)		T	Apribojimo elemento unikalus identifikatorius – išorinis raktas
FK_ID_OPERACIJA	Number(10)		T	Apribojimo operacijos unikalus identifikatorius – išorinis raktas

5 PRIEDAS. Struktūrų palyginimo skriptai

Pateikiami struktūrų panaudojimo skriptai, naudojant Oracle PL/SQL procedūrinę kalbą, apribojimų sąlygas aprašytas apribojimų redaktoriumi ir saugojamas meta DB.

1: Įprastos loginės struktūros “IF-THEN” panaudojimas DBVS, duomenų pagal užsibrėžtus redaktoriumi ir saugojamus meta DB apribojimus, išrinkimui ir saugojimui į atskiras lenteles atlikti.

```
begin
  for data in ( SELECT * FROM B_PLANAS)
  loop
    if (data.BP_SUMA < 1000) and (data.bp_metai between '2007.01.01' and '2007.03.01') THEN
      INSERT INTO B_PLANAS_HISTORY1 VALUES (data.BP_BE_ID, data.BP_SUMA, data.BP_METAI);
    end if;
    if (data.BP_SUMA BETWEEN 1500 and 5000) and (data.bp_metai between '2007.01.01' and '2007.03.01')
  THEN
    INSERT INTO B_PLANAS_HISTORY2 VALUES (data.BP_BE_ID, data.BP_SUMA, data.BP_METAI);
    end if;
    if (data.BP_SUMA <10000 AND data.BP_SUMA>5000)and (data.bp_metai between '2007.01.01' and
'2007.03.01') THEN
      INSERT INTO B_PLANAS_HISTORY3 VALUES (data.BP_BE_ID, data.BP_SUMA, data.BP_METAI);
    end if;
  end loop;
end;
```

2: Struktūros “FIRST INSERT” panaudojimas Oracle9i DBVS, duomenų pagal užsibrėžtus redaktoriumi ir saugojamus meta DB apribojimus, išrinkimui ir saugojimui į atskiras lenteles atlikti.

```
begin
  INSERT FIRST
    WHEN (BP_SUMA < 1000) and (bp_metai between '2007.01.01' and '2007.03.01') THEN
      INTO B_PLANAS_HISTORY1 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)
    WHEN (BP_SUMA BETWEEN 1500 and 5000) and (bp_metai between '2007.01.01' and '2007.03.01') THEN
      INTO B_PLANAS_HISTORY2 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)
    WHEN (BP_SUMA <10000 AND BP_SUMA>5000) and (bp_metai between '2007.01.01' and '2007.03.01')
  THEN
    INTO B_PLANAS_HISTORY3 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)
  SELECT * FROM B_PLANAS;
end;
```

3: Struktūros “INSERT ALL” panaudojimas Oracle9i DBVS, duomenų pagal užsibrėžtus redaktoriumi ir saugojamus meta DB apribojimus, išrinkimui ir saugojimui į atskiras lenteles atlikti.

begin

INSERT ALL

WHEN (BP_SUMA < 1000) and (bp_metai between '2007.01.01' and '2007.03.01') THEN

INTO B_PLANAS_HISTORY1 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)

WHEN (BP_SUMA BETWEEN 1500 and 5000) and (bp_metai between '2007.01.01' and '2007.03.01') THEN

INTO B_PLANAS_HISTORY2 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)

WHEN (BP_SUMA <10000 AND BP_SUMA>5000)and (bp_metai between '2007.01.01' and '2007.03.01') THEN

INTO B_PLANAS_HISTORY3 VALUES (BP_BE_ID, BP_SUMA, BP_METAI)

SELECT * FROM B_PLANAS;

end;

4: Vykdomi 1, 2, 3 aprašyti algoritmai, tačiau su sudėtingesne užklausa – aprašytų apribojimų didesnis paieškos intervalas.

(BP_SUMA < 1000) and (bp_metai between '2007.04.01' and '2007.08.01')

(BP_SUMA BETWEEN 1500 and 5000) and (bp_metai between '2007.04.01' and '2007.08.01')

(BP_SUMA <10000 AND BP_SUMA>5000)and (bp_metai between '2007.04.01' and '2007.08.01').

6 PRIEDAS. Straipsnis.

CENTRALIZUOTOS DUOMENŲ VIENTISUMO APRIBOJIMŲ REALIZACIJOS METODIKA

Virginija Uzdanavičiūtė

Kauno Technologijos Universitetas, Informatikos fakultetas, Informacijos sistemų katedra

Straipsnyje nagrinėjami duomenų vientisumo apribojimų įgyvendinimo DBVS metodai, realizacijos Oracle DBVS procedūromis metodika, kuri leidžia užtikrinti nepriekaištingą informacinės sistemos funkcionavimą ir apsaugą nuo nekorektiškų duomenų. Pateikiami procedūrų modeliai UML kalba bei aprašomas redaktorius, kuris padėtų projektuoti, realizuoti ir palaikyti procedūras viso jų gyvavimo ciklo metu.

1 Konceptualaus modelio projektavimo svarba IS kūrime

Konceptualaus modelio, duomenų bazės projektavimas ir realizavimas – svarbus IS kūrimo etapas. RDBVS teorijoje yra keletas reikalavimų, kuriuos turi atitikti duomenų bazės duomenys. Viena reikalavimų rūšis – užtikrinti duomenų vientisumo sąlygas, kurioms esant, DB tam tikru laipsniu yra teisinga, išsami ir pilna.

Didelėse, sudėtingose dalykinėse srityse, turinčiose daug duomenų ir apribojimų, iškyla eilė problemų, susijusių su jų valdymu, kadangi pertekliniai apribojimai neigiamai įtakoja veikimo greitį, užima papildomą vietą kompiuterio atmintyje, o prieštaringi ar konfliktiniai apribojimai iššaukia klaidas ir begalinius ciklus [1], anomalijas. *Atnaujinimo anomalija* – tai duomenų prieštaringumas, atsirandantis dėl duomenų pertekliaus, atnaujinus tik dalį jų. *Pašalinimo anomalija* – nenumatytas duomenų praradimas, susijęs su kitų duomenų pašalinimu. *Įvedimo anomalija* – nebuvimas galimybės įvesti duomenis dėl to, kad trūksta kitų duomenų.

Apribojimų įvertinimas konceptualiajame modelyje leidžia adekvačiai atvaizduoti dalykinę sritį [2], verifikuoti modelį ankstyvojoje kūrimo stadijoje ir naudoti ne tik duomenų bazių, bet ir kitų tipų schemas, programų kodui generuoti.

Išvengti daugumos su vientisumo apribojimais susijusių problemų padeda tinkamai parinktas jų užtikrinimo metodas.

2 Vientisumo apribojimų realizavimo metodai

Apribojimai lentelėms – tai sąlygos, kurios nurodomos kaip lentelės apibrėžimo dalis ir nusako reikalavimus atributų domenams. Tos sąlygos turi būti išpildytos visoms lentelės eilutėms. Sąlyga gali būti nurodyta konkrečiam atributui ar visai lentelei. Pateikiama detali apribojimų klasifikacija 1 pav.



1 pav. Vientisumo apribojimai

Esybės integralumo užtikrinimo taisyklė – pirminio raktų reikšmė negali būti tuščia („null“). Išorinio raktų vientisumo taisyklė – turi būti ryšys tarp išorinio bei susieto pirminio raktų. Jei yra duomenų vaikinėje lentelėje, tai tėvinė lentelė negali būti pašalinta. Veiklos (*business*) vientisumo užtikrinimo taisyklės skirtos IS dalykinės srities taisyklėms užtikrinti.

Žinoma, kad vientisumo apribojimai gali būti realizuoti šiais metodais:

- taikomosiomis programomis;
- duomenų bazės trigeriais;
- duomenų bazės palaikomais apribojimais.

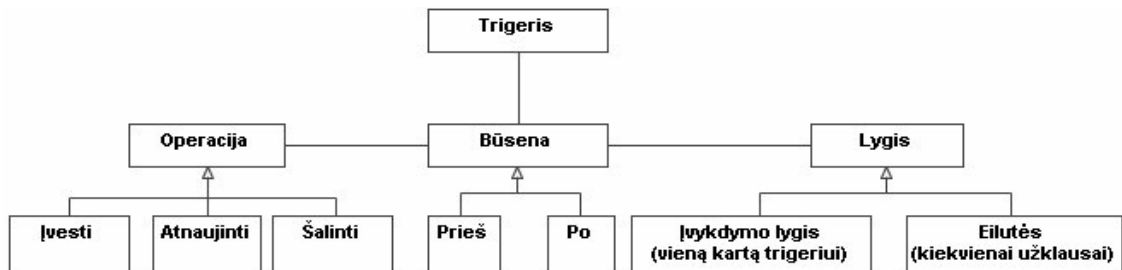
Taikomojoje programoje užtikrinamos sąlygos, kad įvedant, šalinant ar modifikuojant įrašus būtų išlaikomas duomenų korektiškumas, teisingumas, išvengta įvairių duomenų anomalijų.

Trigeriai, kaip ir deklaratyvūs apribojimai yra naudojami duomenų neprieštaringumui palaikyti, vykdant SQL **Delete**, **Update** ar **Insert** operacijas. Trigeriai gali būti naudojami ir kitų, tiesiogiai nedalyvaujančių operacijose, lentelių modifikavimui priklausomai nuo atliekamos operacijos pagrindinėje lentelėje (pvz. renkant statistiką apie kiekvieną

duomenų modifikavimą lentelėje, pakeitus bet kurią eilutę lentelėje, kitoje lentelėje įtraukiama nauja eilutė). Trigeriai kuriami sakiniu **Create trigger**, kuriame nurodoma:

- kurį SQL duomenų modifikavimo sakinį (Insert, Update ar Delete) vykdant, kviesti triggerį;
- kokių momentu triggerį aktyvuoti: prieš vykdant modifikavimo operaciją ar po;
- ar jį aktyvuoti kiekvieną kartą vykdant SQL sakinį ar kiekvienai operacijoje dalyvaujančiai eilutei;
- vieną ar kelis sakinius, kuriuos reikia įvykdyti aktyvavus triggerį (triggerio kūnas);
- triggerį aktyvavus, jo kūne nurodomus sakinius vykdyti (be) sąlygiškai.

Triggerio operacijų, būsenų ir vykdymo metamodelis pateikiamas 2 pav.



2 pav. Trigerių metamodelis

Apribojimų įgyvendinimas taikomųjų programų kode ar triggerių pagalba iššaukia:

- kodo dubliavimą;
- ilgą kūrimo laiką;
- didesnę klaidų atsiradimo tikimybę;
- sudėtingą apribojimo pakeitimo vykdymą.

Siekiant numatyto tikslo, – kompiuterizuoto vientisumo apribojimų valdymo proceso, kurio kontekste IS veiklos pakeitimų realizavimas padarytų kuo mažiau žalos ir truktų kuo trumpesnę laiką, – siūlomas struktūrizuotas metodas.

3 IS veiklos apribojimų realizavimo metodas

IS efektyviam veikimui užtikrinti, siūlomas metodas – IS veiklos taisyklės saugoti struktūrizuotoje saugykloje („repository“) pagrįstoje duomenų bazės valdymo sistema Oracle. Šių taisyklių ir apribojimų automatizuotam ir centralizuotam įgyvendinimui bei administravimui bus sukurtas patogus įrankis.

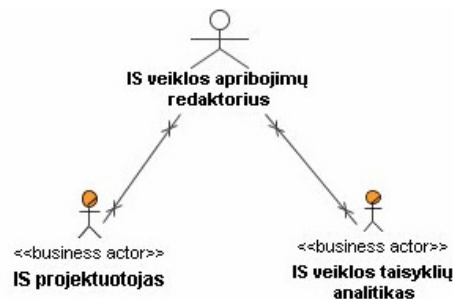
IS veiklos apribojimų įrankio - redaktoriaus realizavimui turi būti taikomas **unifikavimo principas**:

- sistema turi būti taip projektuota, kad ji būtų vienalytė;
- visos sistemos dalys turi būti suprojektuotos ir sukonstruotos vienodai;
- sistema turi būti suvokiama ir keičiama ne tik jų kūrėjų, bet ir kitų asmenų.

Programinės įrangos veiklos modeliams ir reikalavimams specifikuoti, vizualizuoti ir dokumentuoti pasirinkta priemonė [3], [4] – Magic Draw UML 10.0. Tai – įvairiapusis CASE įrankis, kuris realizuoja UML 2.0.

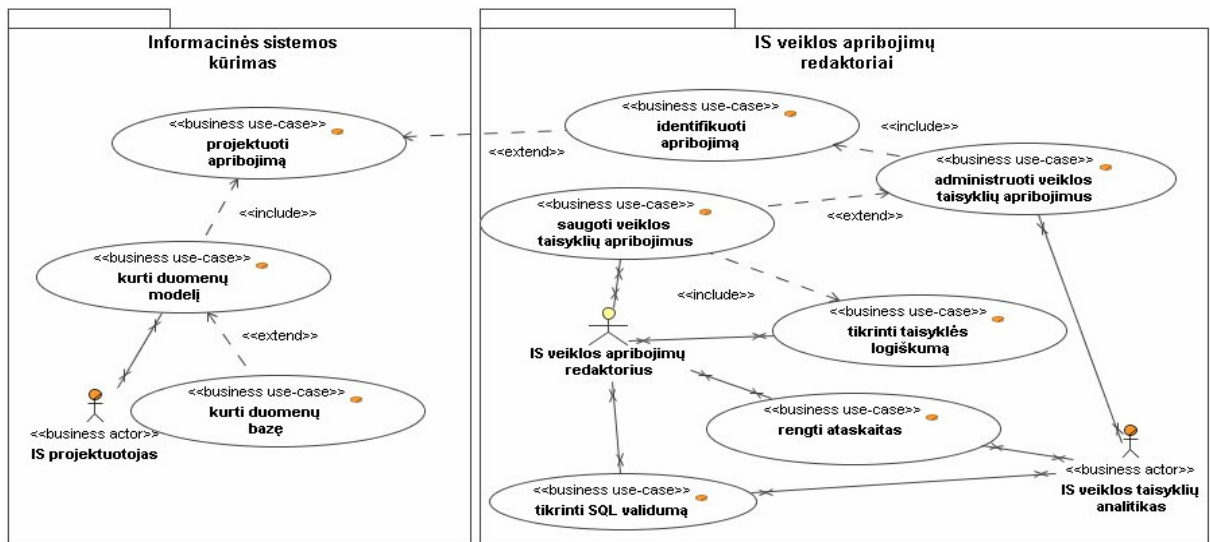
Sudarant sistemos veiklos modelius, naudojami veiklos panaudojimo atvejų stereotipai: <<Business use case>>. Šie stereotipai yra dalis Rational organizacijos unifikavimo kūrimo proceso (RUP) specifikos.

Pateikiama realizuojamo IS veiklos apribojimų redaktoriaus kontekstinė diagrama (3 pav.). Veiklos subjektai – asmenys vaizduojami veiklos aktoriais; centre rodoma kuriama sistema.



3 pav. Veiklos subjektų diagrama

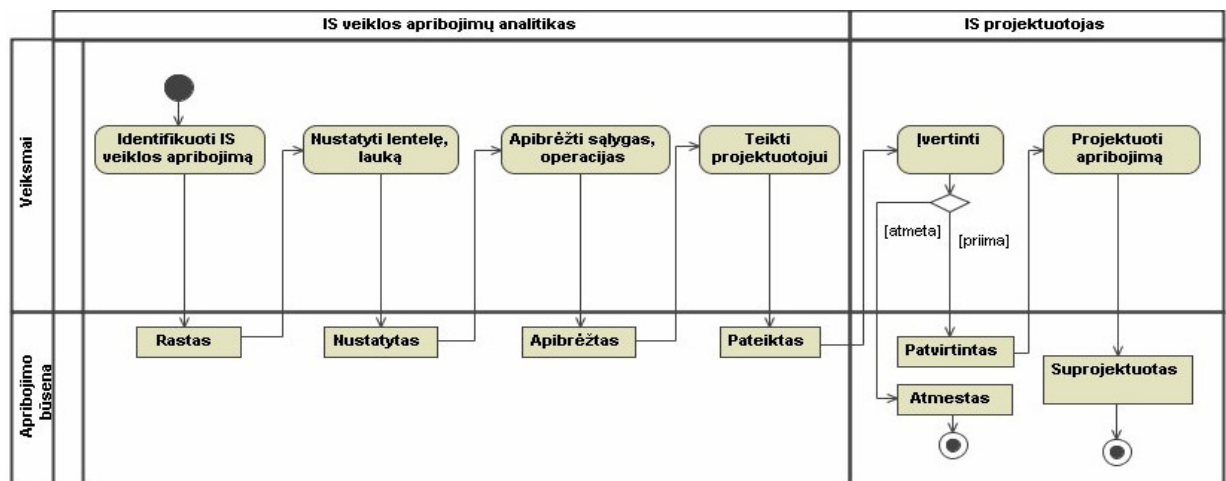
Pateikiamas IS veiklos apribojimų sistemos veiklos sąveikų modelis (4 pav.), kuris rodo būsimos sistemos veiklos procesus, kuriuose dalyvauja išoriniai aktoriai („business actors“).



4 pav. IS veiklos apribojimų sistemos veiklos sąveikų modelis

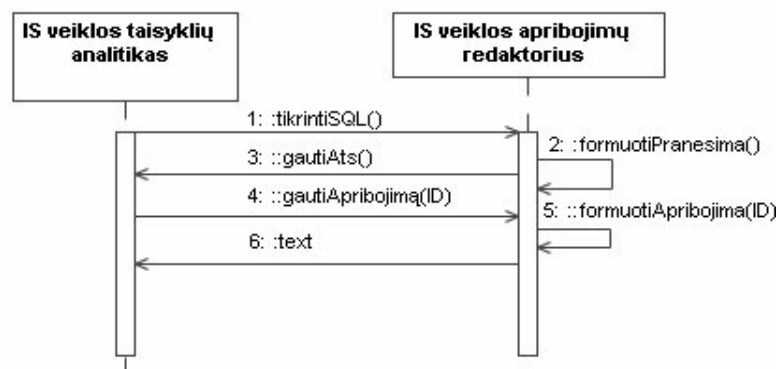
IS veiklos taisyklių analitiko rolės: identifikuoti apribojimus duomenų modelyje, administruoti (įvesti/redaguoti/šalinti/išrinkti) apribojimus ir veiklos taisykles, rengti ir analizuoti sistemos ataskaitas.

IS projektuotojas atsakingas už informacinės sistemos duomenų modelio, jo apribojimų ir duomenų bazės korektišką, išsamų, teisingą ir logišką projektavimą. IS analitiko ir projektuotojo veiklos diagrama, įgyvendinant duomenų modelyje apribojimus, nustatant jų būsenas kiekviename etape, pateikiama (5 pav.)



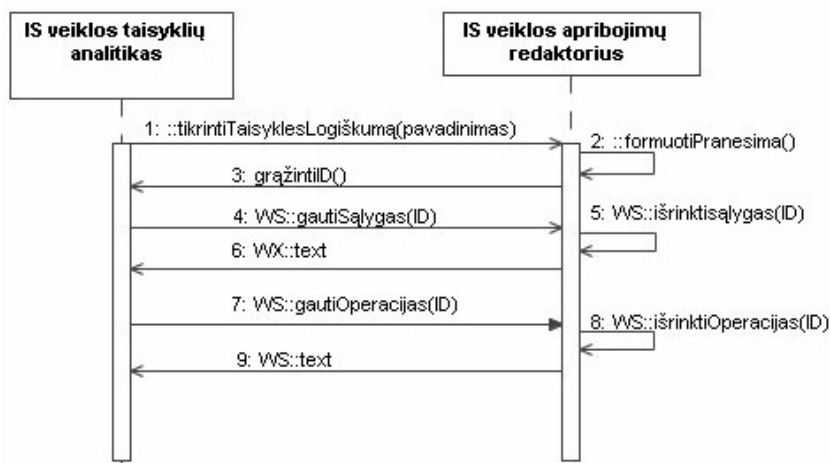
5 pav. Apribojimo realizavimo veiklos ir būsenų perėjimo diagrama

IS veiklos taisyklių redaktoriaus programinė įranga atlieka šias funkcijas: tikrina veiklos taisyklių logiškumą, teikia ataskaitas apie apribojimų naudojimo dažnumą (galime identifikuoti sistemos veikimo riziką), leidžia administruoti ir saugoti IS veiklos taisykles DBVS'e. Sumodeliuotos IS veiklos apribojimų redaktoriaus ir IS analitiko sekų diagramas (6 pav.), (7 pav.).



6 pav. Panaudojimo atvejo „tikrinti SQL“ sekų diagrama

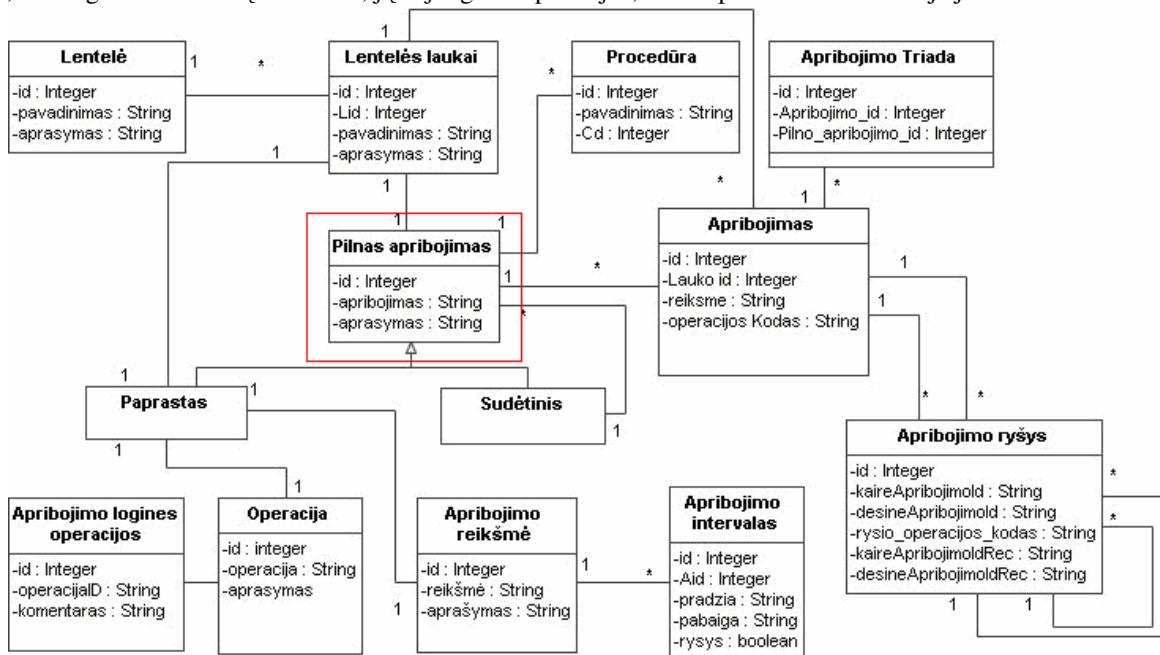
Kompiuterizuota sistemos funkcija „tikrinti SQL“ leidžia IS veiklos analitikui patikrinti SQL DML operacijas perduodant tam tikrus duomenis pačiam (pagal nutylėjamą sistema atlieka automatiškai) ir gauti tam tikrą pranešimą (klaidos arba teisingumo), apribojimo ar taisyklės eilutę.



7 pav. Panaudojimo atvejo „tikrinti taisyklės logiškumą“ sekų diagrama

Taisyklės logiškumo tikrinimas bus naudingas, norint įsitikinti sąlygų teisingumu ir pilnumu, išvengti klaidingų duomenų išrinkimo. Paprasčiausias tikrinimo pvz. – ar $90 > 100 < 112$?

Pateikiamas dalykinės srities klasių modelis (8 pav.) Modelio esmė – saugoti IS veiklos taisyklių taikymo DB lentelės laukus, operacijas, reikšmes. Taip pat fiksuoti informaciją kokie ryšiai tarp apribojimų, jei apribojimas yra sudėtinis, kokie galimi reikšmių intervalai, jų sujungimo operacijos, kurios procedūros realizuoja jas.



8 pav. Klasių modelis

Apibrėžiamos saugomos operacijos ir jų aprašymai: matematinės ($>$, $<$, $>=$, $<=$, $=$, $<>$) ir loginės (or, and, xor). Intervalai – uždari $[]$ ar atviri $()$. Taip pat ar laukams galimos „null“, „not null“ reikšmės.

Efektyvesniam ir greitesniam sistemos veikimui realizuojama papildoma klasė (apvesta modelyje raudonai), kurioje bus saugojama pilna, logiška ir teisinga tam tikro apribojimo sąlyga, suformuota tam tikros operacijos (duomenų įterpimo, atnaujinimo, trynimo) atlikimui patikrinti. Ši klasė naudojama kaip veiklos apribojimų greitinančioji („cache“) esybė. Operacijų vykdymui naudojama struktūrizuota užklausų kalba (SQL).

Kliento posistemė turi automatiškai užtikrinti, kad duomenys esantys tarp apribojimo sąlygas aprašančių lentelių ir lentelės, kuri saugo visą apribojimą turi būti teisingi ir logiški, redaktoriuje atlikus tam tikrus pakeitimus ar įvedus papildomas sąlygas („synchronization between sources“).

Klasių modelio realizavimo privalumai:

- klaidų tikimybė mažesnė;

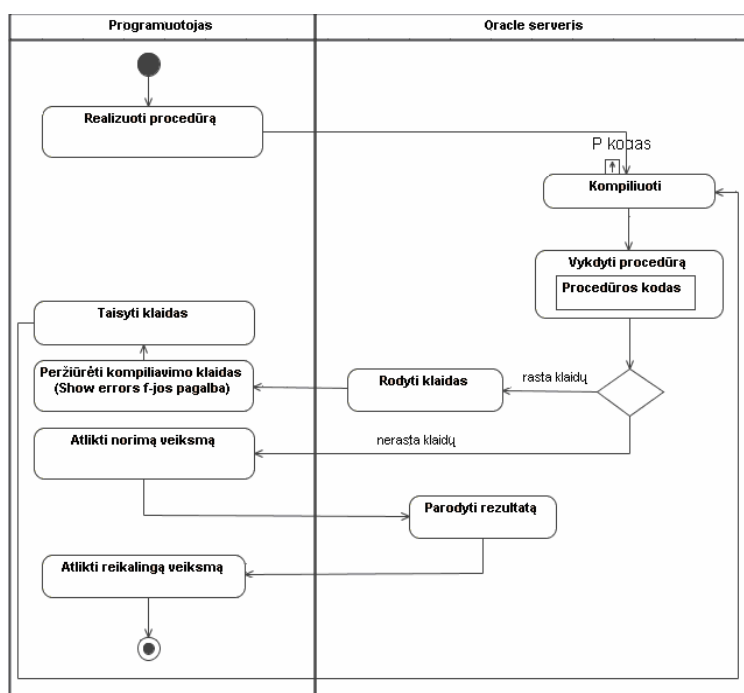
- duomenys saugojami struktūrizuoti;
- struktūrizacija įgalina efektyvų redagavimą įprastais vartotojų įrankiais;
- paprastesnis IS veiklos taisyklių administravimas, palaikymas;
- modelis atitinka reliacinių duomenų bazių teoriją.

Trūkumai:

- ilgesnis IS veiklos taisyklės vykdymo laikas, kurį kompensuos greitinančioji („cache“) esybė.

4 Oracle DBVS procedūrų bruožai

Oracle duomenų bazėse yra įdiegtas objektinis – reliacinis duomenų valdymo modelis, kuris leidžia naudotis plačiausiai pasaulyje paplitusiomis reliacinių duomenų bazių priemonėmis ir standartais, bei neribotai juos plėsti, naudojant duomenų bazėje saugomus objektus. Oracle objektai palaiko daugelį klasikinės objektinės orientacijos savybių: atributai, metodai, paveldimumas, inkapsuliacija, kompozicija ir perdengimas. Plačiau pateikiamos procedūrų charakteristikos ir jų esminiai bruožai [5], [6], realizavimo diagrama (9 pav.)



9 pav. Procedūros realizavimo veiklos diagrama

Pagerintas našumas

- Išvengiama kodo tikrinimo, naudojant daugeliui naudotojų paskirstytoje SQL srityje;
- Išvengiama PL/SQL kodo tikrinimo realiu laiku, nes klaidų tikrinimas atliekamas procedūros kompiliavimo metu;
- Naudojant komandas, sumažinamas duomenų bazės užklausų skaičius;
- Sumažėja tinklo apkrovimas, nes tinklu perduodamas tik kreipinys į procedūrą (o ne visas *Transact-SQL* programos tekstas);
- Padidėja darbo našumas, kadangi šios procedūros yra sukompilijuotos, ir jas įvykdžius vieną kartą, pasilieka atmintyje kitam vykdymui.

Lengvas palaikymas

- Leidžia atlikti pakeitimus šablone tiesiogiai, netrukdant kitiems naudotojams;
- Vieno šablono keitimas paveikia daugelį aplikacijų;
- Vieno šablono keitimas leidžia išvengti analogiško testavimo;
- Suteikia galimybę vienos taikomosios programos logiką paskirstyti daugeliui programų.

Pagerintas duomenų saugumo ir integralumo užtikrinimas

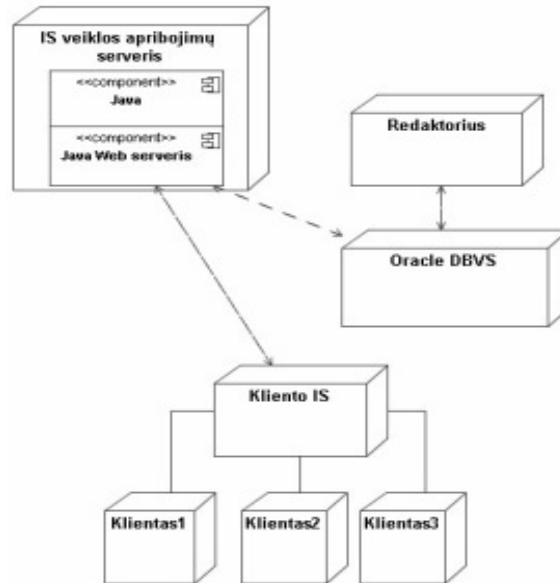
- Leidžia kontroliuoti neprivilegijuotų vartotojų priejimą prie duomenų bazės objektų;
- Veiklos taisyklių vieningos saugyklos organizavimo būdas, nepaskirstant jų daugelyje taikomųjų kliento programų.

Aiškus kodas

- Naudojant logiškus, aiškius ir paskirtį atitinkančius procedūrų pavadinimus, išvengiama daugelio komentarų rašymo.

5 Web servais – sistemos galimybių išplėtimas

Pateikiamas IS veiklos taisyklės įgyvendinančios sistemos architektūra (10 pav.) Java Web servisų sukūrimas leis pasinaudoti tiesiogiai IS veiklos funkcijas įgyvendinančiomis Oracle procedūromis tinkle greitai, efektyviai ir dinamiškai (perduodant tam tikrus parametrus). Žiniatinklio technologijų įdiegimas maksimizuos tinklo panaudojimą, užtikrins centralizuotą IS veiklos taisyklių įgyvendinimą ir pakeitimą. Taip pat iš tinkle esančių Web Servisų bus galima susikurti naujas paslaugas, adaptuotas konkrečiam organizacijos IS uždaviniui atlikti.



10 pav. Sistemos architektūra

6 Išvados

Vientisumo apribojimų įgyvendinimas duomenų baze pagrįstoje saugykloje išryškina šiuos IS veikimo privalumus:

- užtikrinamas duomenų vientisumas (nėra dubliuotų, neteisingų duomenų);
- optimizuojamos užklausos;
- efektyvesnis indeksų kūrimas ir duomenų išrinkimas;
- apribojimų parametrizavimas leis atnaujinti apribojimus, nekeičiant procedūrų kodo.

Literatūra

- [1] Elita Miliauskaitė, Lina Nemuraitė. Representation of integrity constraints in conceptual models.
- [2] Lina Nemuraitė, Elita Miliauskaitė. Konceptualiojo modelio vientisumo apribojimų taksonomija.
- [3] Saulius Kaukėnas, Darius Šilingas. UML panaudojimas IT specialistų darbo optimizavimui.
- [4] Darius Šilingas. Programinės įrangos projektavimas naudojant UML.
- [5] Nagavalli Pataballa, Priya Nathan. Learn Oracle from Oracle. Oracle9i: Program with PL/SQL, Volume 1. – 2001
- [6] Nagavalli Pataballa, Priya Nathan. Learn Oracle from Oracle. Oracle9i: Program with PL/SQL, Volume 2. – 2001

CENTRALIZED IMPLEMENTATION METHODOLOGY OF DATA INTEGRITY CONSTRAINTS

This paper analyses the methods of implementation of data integrity constraints describes a Oracle RDBMS-based technique that allows for impeccable functionality of information system and protection from invalid data. UML models of procedures discussed are supplied. The editor for designing, implementing and maintaining of procedures through their lifecycle is described.