

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Deividas Švėgžda

**Semantinių ir reliacinių užklausų vykdymo Oracle
DBVS tyrimas**

Magistro darbas

Vadovas: prof. dr. R. Butleris

Konsultantė: dokt. V. Uzdanavičiūtė

KAUNAS, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Deividas Švėgžda

**Semantinių ir reliacinių užklausų vykdymo Oracle
DBVS tyrimas**

Magistro darbas

Recenzentas:

doc. dr. Algimantas Venčkauskas

2012-05-24

Vadovas:

prof. dr. R. Butleris

2012-05-28

Konsultantė:

dokt. V. Uzdanavičiūtė

2012-05-20

Atliko:

IFM-0/4 gr. stud.

Deividas Švėgžda

2012-05-23

KAUNAS, 2012

Turinys

1. ĮVADAS.....	10
2. SEMANTINIŲ IR RELIACINIŲ UŽKLAUSŲ VYKDYMO ORACLE DBVS ANALIZĖ	12
2.1 SEMANTINIŲ IR RELIACINIŲ UŽKLAUSŲ VYKDYMO <i>ORACLE DBVS</i> ANALIZĖ, METODAI	13
2.1.1 <i>Informacijos klasifikavimo metodas</i>	13
2.1.2 <i>Informacijos palyginimo metodas</i>	15
2.2 ONTOLOGIJŲ IR JŲ APRAŠYMO KALBŲ ANALIZĖ	15
2.2.1 <i>Ontologijos sąvoka</i>	16
2.2.2 <i>Ontologijų kūrimas</i>	16
2.2.3 <i>Ontologijų aprašymo kalbos</i>	18
2.3 <i>SQL</i> IR <i>SPARQL</i> UŽKLAUSŲ ANALIZĖ	21
2.3.1 <i>SQL užklausų analizė</i>	21
2.3.2 <i>SPARQL užklausų analizė</i>	22
2.3.3 <i>SQL ir SPARQL užklausų palyginimas</i>	22
2.4 ORACLE SEMANTINĖS SAUGYKLOS ANALIZĖ.....	23
2.4.1 <i>RDF modelių įkėlimas</i>	24
2.4.2 <i>SQL, SPARQL užklausų vykdymo prototipas</i>	26
2.4.3 <i>RDF modelio papildymas</i>	28
2.4.4 <i>Oracle SPATIAL RDF duomenų modelis</i>	30
2.5 VARTOTOJŲ ANALIZĖ	31
2.6 PROGRAMINIŲ PAKETŲ, RELIACINIŲ IR SEMANTINIŲ SAUGYKLŲ DUOMENŲ INTEGRAVIMO SISTEMAI KURTI, ANALIZĖ.....	31
2.6.1 <i>Operacinė sistema</i>	31
2.6.2 <i>Modeliavimo įrankiai</i>	35
3. SEMANTINIŲ DUOMENŲ SAUGOJIMO METODAS	36
3.1 SEMANTINIŲ DUOMENŲ SAUGOJIMO METODO PROCESŲ MODELIAI	36
3.2 SEMANTINIŲ DUOMENŲ SAUGOJIMO METODO PANAUDOJIMO ATVEJAI	39
3.2.1 <i>Panaudojimo atvejis „kurti semantinius modelius“</i>	40
3.2.2 <i>Panaudojimo atvejis „įkelti semantinius duomenis“</i>	41
3.2.3 <i>Panaudojimo atvejis „įkelti semantinius duomenis INSERT metodu“</i>	41
3.2.4 <i>Panaudojimo atvejis „įkelti semantinius duomenis masiniu įkėlimo metodu“</i>	42
3.2.5 <i>Panaudojimo atvejis „sukurti taisyklių rinkinį“</i>	44

3.2.6	<i>Panaudojimo atvejis „sukurti taisyklę“</i>	44
4.	SEMANTINIŲ IR RELIACINIŲ DUOMENŲ INTEGRAVIMO METODAS	46
4.1	SEMANTINIŲ IR RELIACINIŲ DUOMENŲ INTEGRAVIMO METODO PROCESO MODELIAI	46
4.2	SEMANTINIŲ IR RELIACINIŲ DUOMENŲ INTEGRAVIMO METODO PANAUDOJIMO ATVEJAI	46
4.2.1	<i>Panaudojimo atvejis „Sukurti šabloną“</i>	46
4.2.2	<i>Panaudojimo atvejis „vykdyti duomenų paiešką iš šablono“</i>	47
5.	SUKURTŲ METODŲ TAIKYMO EKSPERIMENTINIS TYRIMAS	49
5.1	EKSPERIMENTO TIKSLAS	49
5.2	APRIBOJIMAI EKSPERIMENTUI	49
5.3	EKSPERIMENTO PLANAS	51
5.4	EKSPERIMENTO REZULTATAI	56
5.5	EKSPERIMENTO IŠVADOS	59
6.	IŠVADOS	60
	SUMMARY	62
	LITERATŪRA	63
	PRIEDAS 1	65
	PRIEDAS 2	65
	PRIEDAS 3	66

Paveikslėlių sąrašas

1 pav. Pagrindinių tyrimo tikslų modelis	12
2 pav. Informacijos skirstymo principinė schema.....	13
3 pav. Magistrinio darbo literatūros analizės schema, sudaryta pagal informacijos klasifikavimo metodą	15
4 pav. Ontologijos schema	17
5 pav. RDF grafas.....	18
6 pav. Ontologijų aprašymo kalbų plėtimosi principinis modelis	21
7 pav. <i>Oracle</i> semantinio tinklo architektūra [11].....	23
8 pav. Semantinis duomenų modelis „suvestinė“.....	27
9 pav. Taisyklių rinkinių schema.....	28
10 pav. <i>Oracle RDF</i> saugyklos duomenų modelis [15]	30
11 pav. Įmonės IT sistemos kaštų palyginimas	32
12 pav. <i>OS</i> lankstumą labiausiai lemiantys veiksniai.....	33
13 pav. Semantinių duomenų rašymo į <i>Oracle DBVS</i> proceso diagrama	36
14 pav. Taisyklių priskirimo proceso diagrama	37
15 pav. Semantinių duomenų įkėlimo proceso diagrama.....	38
16 pav. Duomenų įkėlimo masiniu būdu DB lygyje veiksmų schema.....	39
17 pav. Semantinių duomenų paruošimo panaudojimo atvejų diagrama	40
18 pav. Panaudojimo atvejo „kurti semantinius duomenis“ sekų diagrama.....	41
19 pav. Panaudojimo atvejo „Įkelti semantinius duomenis <i>INSERT</i> metodu“ sekų diagrama.....	42
20 pav. Panaudojimo atvejo „įkelti semantinius duomenis masiniu įkėlimo metodu“ sekų diagrama	43
21 pav. Panaudojimo atvejo „Sukurti taisyklių rinkinį“ sekų diagrama.....	44
22 pav. Panaudojimo atvejo „sukurti taisyklę“ sekų diagrama	45
23 pav. Semantinių ir reliacinių duomenų manipuliavimo procesų modelis	46
24 pav. Reliacinių ir semantinių duomenų integravimo panaudojimo atvejų modelis	46
25 pav. Šablono kūrimo procesas.....	47
26 pav. Panaudojimo atvejo „vykdyti duomenų paiešką iš šablono“ sekų diagrama.....	48
27 pav. Pavyzdinė <i>Oracle DBVS</i> „Human resource“ duomenų schema.....	50
28 pav. Eksperimento planas	52
29 pav. Modelių kūrimo programinio įrankio langas	56
30 pav. Taisyklių kūrimo programinis įrankio langas	56
31 pav. Semantinių duomenų įrašymo programinis įrankio langas	57
32 pav. Šablono „Darbuotojai“ duomenys	58
33 pav. Šablono „Darbai“ duomenys	59

Lentelių sąrašas

1 lentelė. Ontologijos schemos elementų reikšmės	17
2 lentelė. Ontologijų aprašymo kalbų palyginimas	20
3 lentelė. <i>SQL Loader</i> komandos aprašymas.....	25
4 lentelė. Pateikto 7 pvz. aprašymas.....	25
5 lentelė. Reliacinis duomenų modelis „suvestinė“	26
6 lentelė. <i>Linux</i> ir <i>Windows OS</i> palyginimas	35
7 lentelė. Panaudojimo atvejo „kurti semantinius modelius“ specifikacija.....	40
8 lentelė. Panaudojimo atvejo „įkelti semantinius duomenis" specifikacija	41
9 lentelė. Panaudojimo atvejo „įkelti semantinius duomenis INSERT metodu“ specifikacija	42
10 lentelė. Panaudojimo atvejo „įkelti semantinius duomenis masiniu įkėlimo metodu“ specifikacija	43
11 lentelė. Panaudojimo atvejo „sukurti taisyklių rinkinį“ specifikacija	44
12 lentelė. Panaudojimo atvejo „sukurti taisyklę“ specifikacija	45
13 lentelė. Panaudojimo atvejo „sukurti šabloną“ specifikacija.....	47
14 lentelė. Panaudojimo atvejo „vykdyti duomenų paiešką iš šablono“ specifikacija.....	48
15 lentelė. Sukurti semantinius modelius „Darbas“ ir „Darbuotojai“	52
16 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles	53
17 lentelė. Masiniu metodu įkelti semantinius duomenis į „Darbas“ modelį	53
18 lentelė. <i>INSERT</i> metodu įkelti semantinius duomenis į „Darbuotojai“ modelį.....	54
19 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles	54
20 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles	55
21 lentelė. Vykdyti duomenų paiešką šablonuose.....	55

Pavyzdžių sąrašas

1 pvz. RDF grafas užrašytas XML sintakse	19
2 pvz. RDFS grafas užrašytas XML sintakse	19
3 pvz. SQL užklauso fragmentas	22
4 pvz. SPARQL užklauso fragmentas.....	23
5 pvz. Tarpinės lentelės <i>stage</i> struktūra	24
6 pvz. <i>SQL Loader</i> iškviatimo komanda	24
7 pvz. Semantinių duomenų įrašymas <i>SQL Loader</i> metodu	25
8 pvz. Semantinių duomenų įrašymas <i>SQL INSERT</i> metodu.....	26
9 pvz. Darbuotojų atranka <i>SQL</i> notacija.....	26
10 pvz. Darbuotojų atranka <i>SPARQL</i> notacija	27
11 pvz. Darbuotojų atranka sudėtine <i>SQL-SPARQL</i> užklausa	28
12 pvz. <i>SPARQL</i> fragmentas, aprašytas taisyklėmis	29
13 pvz. <i>SQL-SPARQL</i> šablono „Darbas“ programnis kodas.....	57
14 pvz. <i>SQL-SPARQL</i> šablono „Darbuotojai“ programnis kodas.....	58

Terminų ir santrumpų žodynas

D2RQ – platforma, skirta reliacinių duomenų bazių virtualaus *RDF* grafo sluoksniui sukurti.

DB – duomenų bazė.

DBVS – duomenų bazės valdymo sistema.

DML (*Data Manipulation Language*) – duomenų manipuliavimo kalba (pvz: *SQL*), skirta duomenims apdoroti duomenų bazėse.

Duomenų bazių valdymo sistema – kompiuterinė programa ar programų paketas, skirtas duomenų bazei valdyti. Paprastai DBVS sugeba valdyti milžiniškus duomenų kiekius bei vienu metu palaiko daugelį lygiagrečiai dirbančių vartotojų.

Ekvivalentiškumas – lygiavertiškumas, lygiareikšmiškumas.

HTML (*Hypertext Markup Language*) – programavimo kalba, skirta pateikti turinį internete.

Kardinalumas – tokia ryšio charakteristika, kuri rodo maksimalų vieno objekto elementų kiekį, surištą su vienu kito objekto elementu.

Linux – laisvo kodo operacinė sistema.

MDSYS – daugiadimensinis sisteminis *Oracle* semantinės dalies vartotojas.

OS – operacinė sistema.

OWL (*Web Ontology Language*) – ontologijų aprašymo kalba.

Poklasis – klasės elementas, esantis žemesniame hierarchijos lygyje, lyginant su tiriama klase.

RDBVS – reliacinė duomenų bazės valdymo sistema.

RDF (*Resource Description Framework*) – konsorciumo *W3C* rekomenduojamas standartas semantikai, semantiniams ryšiams tarp duomenų aprašyti.

RDFS (*RDF Schema*) – standartas semantikai aprašyti, kuris praplečia *RDF* žodyną tam, kad apibūdintų klasių taksonomijas ir savybes.

Reliacinė duomenų bazių valdymo sistema – duomenų bazių valdymo sistema, kuri remiasi Edgardo Kodo (Edgar F. Codd) aprašytu reliaciniu modeliu [1]. Jos pagrindinis elementas yra reliacinė lentelė.

SDBVS – semantinė duomenų bazės valdymo sistema.

Semantinė duomenų bazių valdymo sistema – duomenų bazių valdymo sistema, kuri remiasi semantiniu duomenų modeliu [2].

SPARQL (*Simple Protocol and RDF Query Language*) – *RDF* užklausų kalba. Ji skirta duomenimis *RDF* grafuose manipuluoti semantinėse duomenų bazių valdymo sistemose.

SQL (*Structured Query Language*) – šiuo metu populiariausia duomenų aprašymo ir manipuliavimo kalba reliacinėse duomenų bazių valdymo sistemose.

SQLplus – standartinis *Oracle DBVS* klientas, galintis vykdyti *SQL* ir *PLSQL* komandas.

Superklasė – aukščiausias elementas hierarchinėje struktūroje.

Taisyklių rinkinys – taisyklių biblioteka, grupė, apjungianti dalykinės tam tikros srities taisykles.

URI (*Universal Resource Identifier*) – nusako, kur pasauliniame tinkle yra išsidėstę ištekliai ir padeda juos identifikuoti.

Viršklasis – klasės elementas, esantis aukštesnėje hierarchijos lygmenyje, lyginant su tiriama klase.

W3C – konsorciumas, leidžiantis programinės įrangos standartus žiniatinkliui.

Windows – *Microsoft* kompanijos kuriama ir plėtojama operacinė sistema.

XML (*eXtensible Markup Language*) – duomenų struktūrų bei jų turinio aprašomoji kalba.

1. Įvadas

Saugomos informacijos kiekiai su kiekviena diena nenumaldomai didėja. Duomenų suvaldymo ir tinkamo panaudojimo procesai lėtėja ir nebeatitinka šiuolaikinės visuomenės poreikių. Dabartiniam duomenų saugojimo metodui, pagrįstam Edgardo Kodo reliaciniu modeliu [1], ieškoma alternatyvų. Pasigirsta novatoriškų sprendimų, vadinamų semantika, semantiniiais tinklais, ontologijomis ir pan.

Siekiant sparčiau apdoroti informaciją, kuriami mechanizmai duomenims sisteminti, aprašant ontologijomis pagal semantinę prasmę. Tačiau ir ontologijų apimtys didėja, tuo pačiu metu duomenų interpretavimo procesai lėtėja. Todėl iškyla klausimai:

1. Kaip ir kur saugoti ontologijas?
2. Kaip efektyviai jose manipuluoti duomenimis?
3. Kaip integruoti semantinių saugyklų duomenis su anksčiau sukurtų reliacinių *DB* duomenimis?

Ontologijas galima saugoti tiek tekstiniuose failuose, tiek reliacinėse duomenų bazėse. Pirmuoju atveju ontologijoms valdyti reikia nemažų kompiuterio resursų. Nukenčia ontologijų saugojimo efektyvumas. Lyginant su pirmuoju antrasis būdas turi privalumų:

1. paprastas ir greitas *RDF* grafų saugojimas;
2. semantinių duomenų indeksacija, paspartinanti paiešką;
3. *SQL* ir *SPARQL* užklausų panaudojimas;
4. prieiga prie modelių;
5. centralizuotas ontologijų valdymas.

Todėl tikslinga išnagrinėti šiuolaikinės *DBVS*, užtikrinančias *RDF* grafų saugojimą. Koncernas *Oracle* sukūrė *RDF* valdymo platformą ir ją intensyviai plėtoja. Verta panagrinėti *Oracle* saugyklos galimybes ir sukurti sistemą, leidžiančią patogiai administruoti semantinius duomenis ir vykdyti integruotas *SQL* ir *SPARQL* užklausas vienu metu.

Tyrimo sritis

Oracle duomenų bazių valdymo sistemos (*DBVS*) galimybės duomenims integruoti, naudojant ontologijas ir semantinio aprašymo kalbas *RDF/OWL*.

Tyrimo objektas

Oracle DBVS semantinė saugykla ir jos galimybės.

Tyrimo tikslas ir uždaviniai

Tyrimo tikslas – ištirti *Oracle* semantinės duomenų saugyklos galimybes ir sukurti vartotojui patogią sistemą, metodus, leidžiančius:

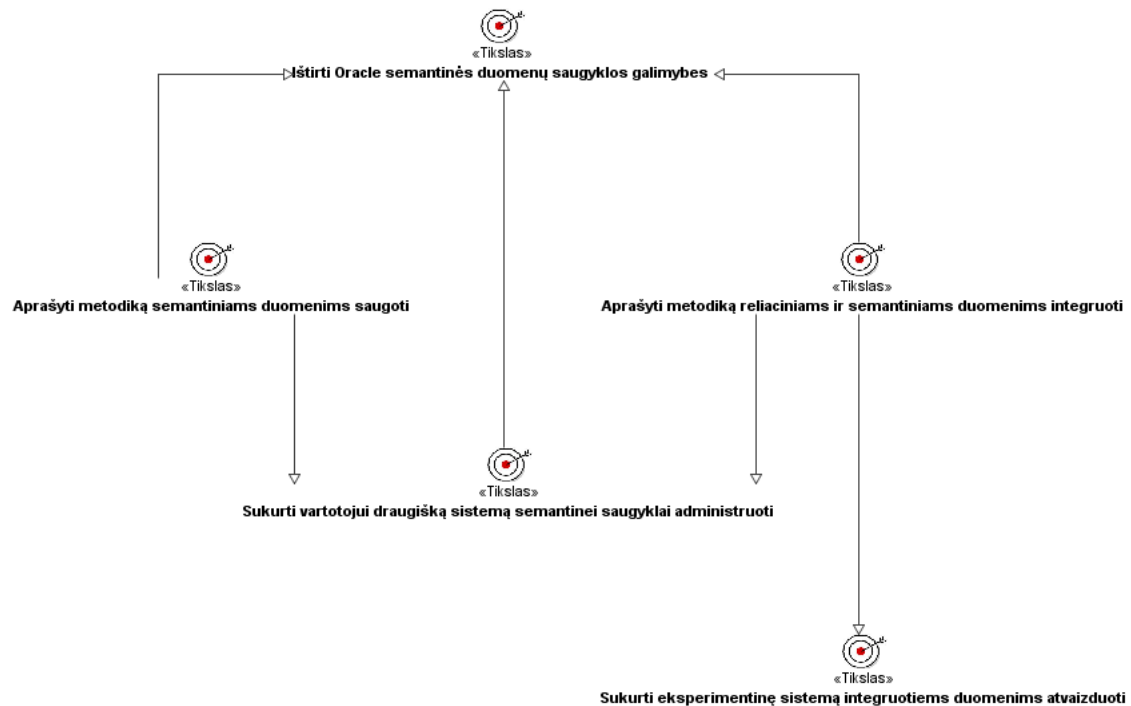
1. užkrauti, saugoti *RDF/RDFS/OWL* modelius;
2. vykdyti *DML* funkcijas iš *RDF/RDFS/OWL* modelių;
3. apibrėžti modeliams taisykles;
4. naudojant *SPARQL* grafinius šablonus, vykdyti *SQL* užklausas iš *RDF/OWL* duomenų ir ontologijų;
5. vykdyti ontologijomis grindžiamas užklausas iš reliacinės *DB*.

Tiksams pasiekti suformuluoti šie uždaviniai:

1. Atlikti ontologijų ir jų aprašymo kalbų analizę:
 - 1.1. *OWL*;
 - 1.2. *RDF* grafai;
 - 1.3. *RDFS*.
2. Atlikti *Oracle DBVS* reliacinės ir semantinės duomenų saugyklos analizę:
 - 2.1 struktūros;
 - 2.2 *Oracle SPATIAL RDF* duomenų modelio;
 - 2.3 *SQL* kalbos, užklausų formavimo principų;
 - 2.4 *SPARQL* kalbos, užklausų formavimo principų.
3. Atlikti programinių paketų analizę reliacinių ir semantinių saugyklų duomenų integravimo sistemai kurti:
 - 3.1 darbinės aplinkos (*Windows, Linux, kita OS*);
 - 3.2 sistemos kūrimo įrankių;
 - 3.3 kitų.
4. Atlikti sistemos kūrimo paketų, programavimo kalbų – technologijų analizę:
 - 4.1 programavimo kalbų;
 - 4.2 programavimo įrankių.
5. Sukurti sistemą, leidžiančią patogiai administruoti semantinius duomenis:
 - 5.1 kurti modelius;
 - 5.2 kurti taisykles;
 - 5.3 įterpti semantinius duomenis.
6. Sukurti sistemą, leidžiančią integruoti reliacinius ir semantinius duomenis, vienu metu vykdyti *SQL* ir *SPARQL* užklausas.
7. Atlikti eksperimentą.

2. Semantinių ir reliacinių užklausų vykdymo *Oracle* DBVS analizė

Šios analizės tikslai – suprasti ir įvaldyti svarbiausias technologijas, metodikas, susijusias su problemine sritimi, ištirti technologijų veikimo principus, tarpusavio sąryšius ir integravimo galimybes. Susipažinti su kitų mokslininkų, dirbančių panašioje srityje, publikuota literatūra, pasiektais rezultatais. Pagrindiniai tikslai pateikti 1 pav.



1 pav. Pagrindinių tyrimo tikslų modelis

Analizės tikslams įgyvendinti keliami šie uždaviniai:

1. literatūros šaltinių analizė;
2. probleminės srities (semantinių ir reliacinių duomenų integravimas) analizė;
3. problemos sprendimo naudingumo ir pagrįstumo analizė;
4. ontologijų ir jų aprašymo kalbų (*OWL*, *RDF*) analizė;
5. *SQL* ir *SPARQL* užklausų analizė;
6. *Oracle* reliacinės duomenų saugyklos analizė;
7. *Oracle* semantinės duomenų saugyklos analizė;
8. programavimo technologijų (kalbų) analizė;
9. programinių paketų, reliacinių ir semantinių saugyklų duomenų integravimo sistemai kurti, analizė;
10. projektavimo įrankių analizė.

2.1 Semantinių ir reliacinių užklausų vykdymo *Oracle DBVS* analizė, metodai

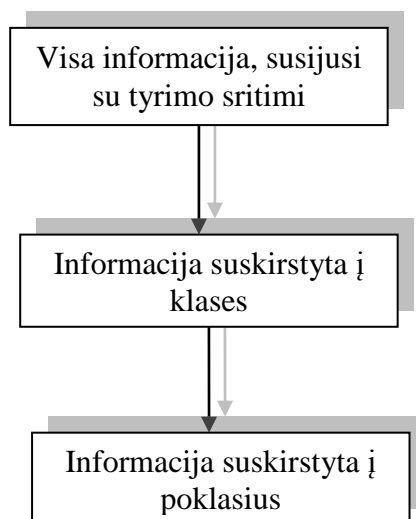
Oracle DBVS semantinės saugyklos ir jos galimybės duomenims integruoti, naudojant ontologijas ir semantinio aprašymo kalbas *RDF/OWL*, analizei naudojami šie metodai:

1. informacijos klasifikavimo metodas;
2. informacijos palyginimo metodas.

Nenuoseklus informacijos analizavimas neefektyvus. Tai lemia netinkamas informacijos atrinkimas: informacijos šaltiniai neinformatyvūs, nevisiškai atitinka tyrimo sritį, kartojasi, pateikiama pasenusi ar net klaidinga informacija. Analizės efektyvumą padidina struktūrinis, sistemingas požiūris į analizuojamą informaciją, taikant mokslinius informacijos atrinkimo metodus.

2.1.1 Informacijos klasifikavimo metodas

Informacija klasifikuojama į tam tikras klases ir poklasius pagal pasirinktus kriterijus. Principinė schema pavaizduota 2 pav.



2 pav. Informacijos skirstymo principinė schema

Šiame darbe į klases skirstoma pagal technologijas ir temas. Iš viso sudarytos 9 klasės:

1. probleminė sritis;
2. problemos sprendimo naudingumas ir pagrįstumas;
3. ontologijos;
4. *SQL* ir *SPARQL* užklausos;
5. *Oracle* reliacinės duomenų saugyklos;
6. *Oracle* semantinės duomenų saugyklos;
7. programavimo technologijos;
8. programavimo įrankiai;
9. projektavimo įrankiai.

Kiekviena klasė turi po 3 vienodus poklasius, kurie nurodo atitinkamos klasės informacijos aktualumą. Informacija įvardijama kaip labai aktuali, aktuali ir mažiau aktuali.

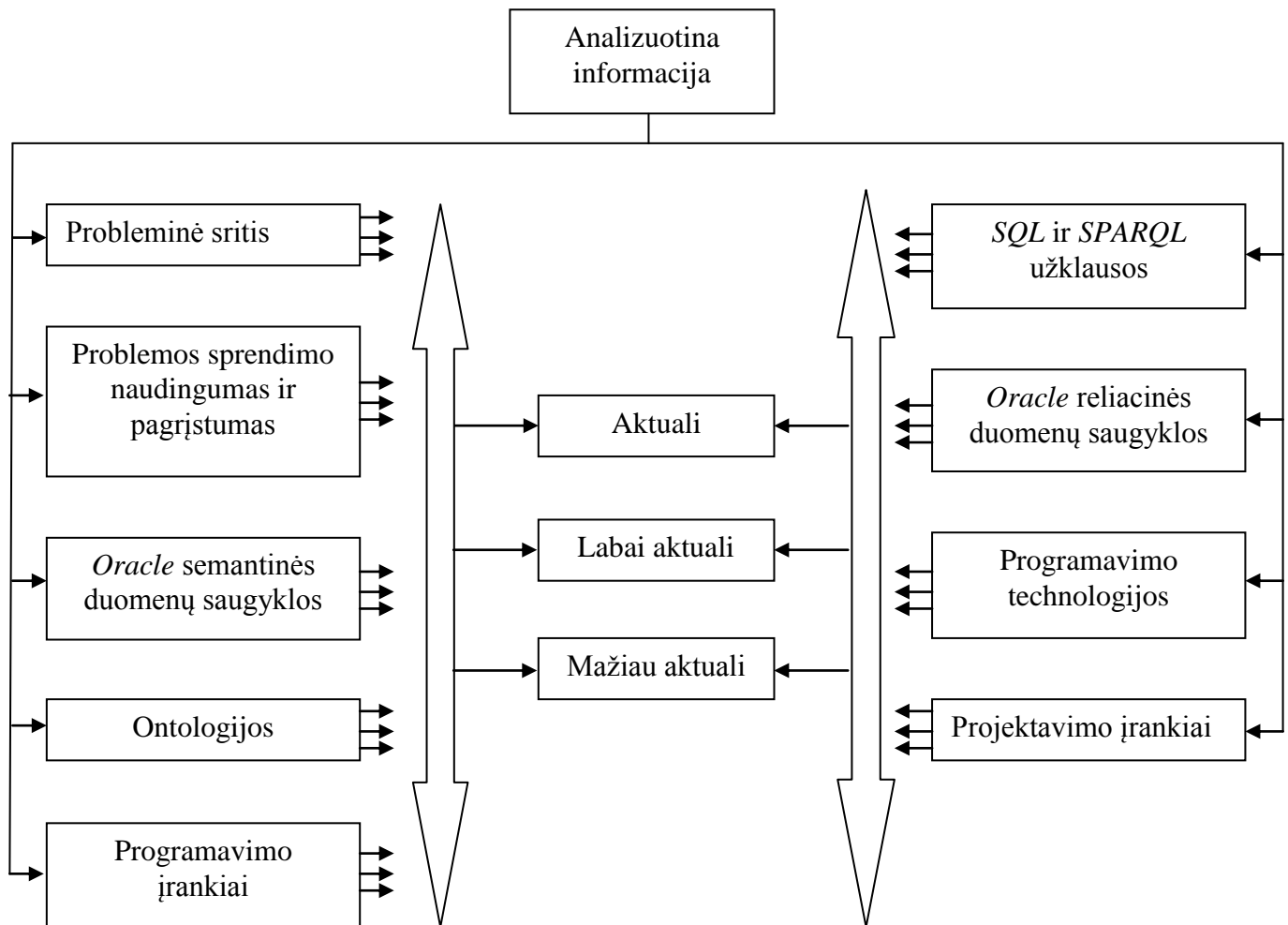
Labai aktuali informacija laikoma ta, kurią būtina išanalizuoti ir įvaldyti, norint sėkmingai išspręsti problemą. Tarkime, turime *SQL* ir *SPARQL* klasę. Tai informaciją apie *SQL* užklausų notaciją laikysime labai aktuali informacija, kadangi neįvaldžius jos uždavinio išspręsti nepavyks.

Aktuali informacija laikoma ta, kurią reikėtų išanalizuoti ir įvaldyti. Tai informacija, kuri nedaro esminės įtakos sprendžiamai problemai, t. y. ir be jos galima išspręsti tiriamą problemą. Tačiau informacija yra svarbi, gali praturtinti žinių bazę, paspartinti tyrimo eigą ir pan. Kaip pavyzdį paimkime du straipsnius ta pačia tematika. Pirmasis publikuojamas vienoje iš didžiausių ir labiausiai vertinamų duomenų bazių, parašytas garsaus ir pripažinto universiteto mokslininkų. Antrasis parašytas mažiau vertinamo universiteto mokslininkų ir publikuojamas asmeninėje internetinėje svetainėje. Be abejonės, pirmasis straipsnis laikomas labai aktuali, tačiau vienu šaltiniu pasikliauti nevertėtų, tad reikėtų išanalizuoti ir antrąjį.

Taip pat į aktualų poklasį įtraukiami senesni, mažiau informatyvūs šaltiniai.

Mažiau aktuali informacija laikoma ta, kurią vertėtų išanalizuoti turint laisvo laiko. Į šią poklasę įtraukiami skirtingų sričių šaltiniai, turintys panašių aspektų su tiriama sritimi. Taip pat šiam poklasiui priklauso senesni, ne tokie patikimi šaltiniai.

3 pav. pateikta magistrinio darbo literatūros šaltinių analizės schema, sudaryta pagal informacijos klasifikavimo metodą.



3 pav. Magistrinio darbo literatūros analizės schema, sudaryta pagal informacijos klasifikavimo metodą

2.1.2 Informacijos palyginimo metodas

Metodas remiasi paprastu dviejų ar daugiau šaltinių tarpusavyje palyginimu. Antrinių šaltinių panaudojimas užtikrina aukštą analizės kokybę, nes skirtingų mokslininkų atlikti tyrimai ta pačia metodika užtikrina duomenų patikimumą. Taip pat lyginant tarpusavyje skirtingus metodus, notacijas, realizacijas, programinius paketus ir t. t., randamas optimalus būdas rezultatui pasiekti.

2.2 Ontologijų ir jų aprašymo kalbų analizė

Ontologijos – stipriai besivystanti, vis plačiau pritaikoma technologija. Ši technologija itin reikšminga siekiant suvaldyti vis sparčiau augančius informacijos kiekius, intelektualizuojant įvairių rūšių informacines sistemas. Taip pat ontologijos vaidina didelį vaidmenį pasaulinio semantinio tinklo koncepcijoje. Semantinis tinklas, kuris dar vadinamas *ateities interneto tinklu*, suvokiamas kaip globali duomenų bazė. Būtent ontologijos, kurios šiuo atveju laikomos informacijos interpretavimo bibliotekomis, leidžia žmonėms ir mašinoms daryti sprendimus, kaip interpretuoti ir panaudoti atitinkamą informaciją. Įvairius informacijos resursus sujungus viena ar keliomis ontologijomis ir apibrėžus reikiama taisyklių rinkiniais sukurama sistema, galinti

logiškai ir savarankiškai disponuoti informacija, t. y. analizuoti ir suvokti tekstą, generuoti atsakymus į užklausas suvokiant generuoto teksto prasmę, daryti išvadas, gauti papildomos informacijos ir t. t.

2.2.1 Ontologijos sąvoka

Ontologija (iš graikų kalbos *onto* – būtis, *logija* – žodis) – filosofinės būties teorija. Terminu *ontologija* vadinama svarbiausia metafizikos šaka, kurioje nagrinėjama būtis ir egzistencija [3]. Taip pat nagrinėjamos pagrindinės kategorijos, bandant išaiškinti, kokios ir kokių tipų esybės egzistuoja.

Kompiuterijoje *ontologijos* apibrėžiamos kaip tam tikros srities sąvokų visumos specifikuojamas išreikštu pavidalu (angl.: *An ontology is an explicit specification of a conceptualization*). Pasak N. Guarino [4], ontologijos yra ypatingos rūšies žinių bazė, aprašančios faktus, kuriuos tam tikra naudotojų grupė laiko visada teisingais, remiantis sutartomis naudojamu žodyno terminų prasmėmis.

2.2.2 Ontologijų kūrimas

Prieš kuriant ontologijas visų pirma reikia apibrėžti, kam ji bus naudojama, kokio detalumo reikalauja. Taip pat reikia suprasti, kad ontologija yra realus pasaulio modelis, todėl jos konstrukcija turi atspindėti realybę. Visi šie aspektai turi reikšmingos įtakos kuriamam modeliui. Apsibrėžti ontologijos sritį galima atsakius į kelis esminius klausimus [5]:

1. Kokią sritį turi apimti ontologijos?
2. Kodėl kuriamos ontologijos? Kokiu tikslu?
3. Į kokius klausimus turi atsakyti ontologijos?
4. Kas naudosis ontologijomis?

Ontologijos kūrimas yra iteracinis¹ procesas. Todėl apibrėžus pradinę ontologijos versiją, ją visada galima patobulinti.

Kitas žingsnis yra pačios ontologijos konstrukcijos kūrimas, t.y. apibrėžtų klasių išdėstymas į taksonominę² hierarchiją bei sąvokų, savybių, atributų aprašymas. Ontologijos konstrukciją sudaro:

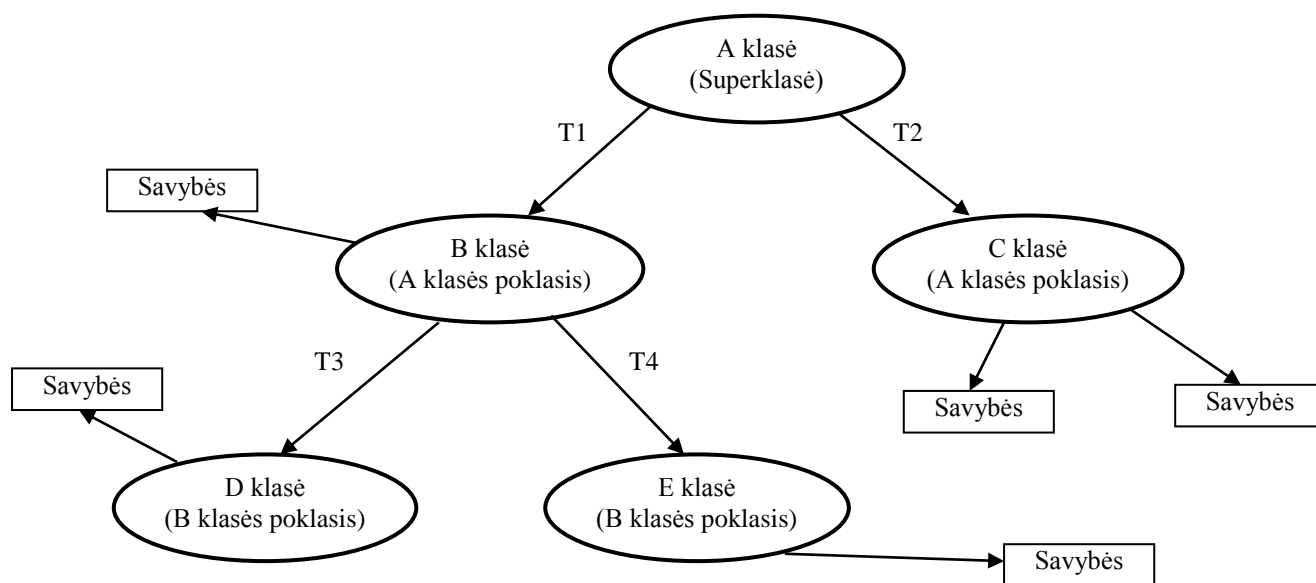
1. klasės (kitaip koncepcijos) – apibūdina dalykinės srities supratimą;
2. poklasiai – detalizuoja klasių supratimą;
3. klasės savybės – aprašo klasės savybes ir atributus;
4. ryšiai;

¹ Iteracija – operacijos pakartotinio panaudojimo rezultatas.

² Taksonomija – klasių išdėstymas poklasiais – viršklasiais.

5. Apribojimai.




Principinė sukonstruotos ontologijos schema pavaizduota 4 pav.



4 pav. Ontologijos schema

Kaip pavaizduota 4 pav., viršūnėje yra *A klasė* dar vadinama *superklase* dėl savo padėties hierarchijoje. Žemiau esančios klasės yra aukštesniųjų klasių poklasiai arba žemesniųjų viršklasiai. Pavyzdžiui D klasė yra B klasės poklasis arba B yra D klasės viršklasis. Ontologijos schemos elementų reikšmės pateiktos 1 lentelėje.

1 lentelė. Ontologijos schemos elementų reikšmės

Elementas	Reikšmė
	Klasė
	Klasės savybės ir atributai
	Ryšys
T1 ÷ T4	Apribojimai

Kuo toliau, tuo daugiau ontologijų sukuriama įvairioms sritims, todėl, prieš konstruojant ontologijas, reikėtų atkreipti dėmesį, ar nėra sukurtų analogiškų reikiamoms. Daug racionaliau panaudoti jau sukurtas ontologijas ar koreguoti jas pagal savus poreikius, nei kurti analogiškas naujai. Analizė parodė, kad yra internetinių ontologijų bibliotekų [6] (*ontology library*), kurios skirtos pakartotinam ontologijų panaudojimui. Pasinaudojant atitinkamais programiniais paketais (tokiais kaip *protege* ir panašiais) paprasta iš bibliotekų persikelti reikiamas ontologijas ir jas koreguoti.

2.2.3 Ontologijų aprašymo kalbos

Pirmos kartos internetas paremtas daugiausiai hiperteksto žymėjimo kalba *HTML* parašytais dokumentais. Tačiau semantiniam pasauliniam tinklui kaip standartas pasirinkta *XML* kalba. *XML* pasirinkta dėl to, kad galima užkoduoti platų duomenų diapazoną ir kartu su ontologijų aprašymo kalbomis – *RDF*, *RDFS*, *OWL* – susieti tarpusavyje suteikiant prasmę. Tuo tarpu *HTML* galimybės labiau ribotos. *HTML* kalba leidžia suteikti tam tikrą struktūrą išvaizdai, duomenims atvaizduoti, tačiau klasifikuoti teksto bloką, sukurti semantikos nepavyks.

2.2.3.1 Resursų aprašymo sistemos (*RDF*) analizė

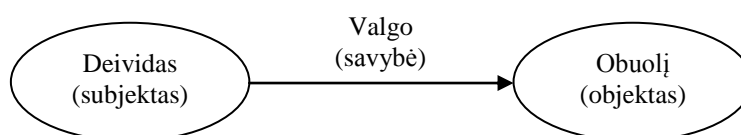
RDF konsorciumo *W3C*³ rekomenduojamas standartas semantikai, semantiniams ryšiams tarp duomenų aprašyti. Šis standartas suteikia paprastą bei efektyvų būdą aprašyti duomenų išteklius, remiantis universaliu resursų identifikatoriumi *URI*.

URI nusako, kur pasauliniame tinkle yra išsidėstę ištekliai ir padeda juos identifikuoti.

RDF naudoja šias pagrindines sąvokas [7]:

1. sukurti paprastą duomenų modelį – duomenys talpinami į grafo tipo struktūrą, kurioje aprašomi objektai susieti su reikšmėmis prasminiu ryšiu;
2. *URI* paremtas žodynas – visi resursai grafe identifikuojami *URI* identifikatoriumi;
3. duomenų tipai – apibrėžia tipų aibę;
4. raidės – paprasta eilutė, sauganti resurso reikšmę (gali būti tipizuota);
5. į *XML* verčiama sintaksė;
6. paprastas faktų išreiškimas – struktūra išlaikoma kaip įmanoma paprastesnė;
7. paveldėjimas – hierarchinės struktūros sudarymas.

RDF kalboje sąryšiai tarp išteklių nusakomi trejetais: subjektas/išteklis (*subject*), predikatas/savybė (*predicate*) ir objektas/reikšmė (*object*). Šiuos trejetus galima interpretuoti kaip trumpus sakinius. Pavyzdžiui, Deividas (subjektas) valgo (savybė) obuolį (objektas).



5 pav. RDF grafas

5 pav. pateikiamas principinis *RDF* grafas, kurį sudaro tik vienas trejetas. Paprastai *RDF* grafas sudarytas iš aibės tarpusavyje susijusių trejetų.

³ <http://www.w3.org>

RDF grafai užrašomi išplėstine *XML* (duomenų struktūrų bei jų turinio aprašomąja kalba), vadinama *XML/RDF*. 1 pvz. pateiktas *XML/RDF* sintakse užrašytas fragmentas, atspindintis 5 pav. pateiktą grafą.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://Deividas.lt">
  <rdf:Description rdf:about="http://Deividas.lt">
    <dc:valgo>rdf:about: obuoli</dc:valgo>
  </rdf:Description>
</rdf:RDF>
```

1 pvz. *RDF* grafas užrašytas *XML* sintakse

2.2.3.2 *RDF* schema (*RDFS*) analizė

RDFS praplečia *RDF* žodyną tam, kad apibūdintų klasių taksonomijas ir savybes [8]. Taip pat ji išplečia apibrėžimus kai kuriems iš *RDF* elementų. Pavyzdžiui, nustato sritis ir savybių diapazoną ir susieja *RDF* klases bei ypatybes į taksonomijas, pasinaudojant *RDFS* žodynu. Šis praplėtimas suteikia didesnes galimybes duomenų prasmėms analizuoti. Dėl šių galimybių *RDFS* dar vadinama paprasta ontologijų kalba.

Nagrinėjant tą patį pavyzdį, kur Deividas valgo obuolį, pritaikant *RDFS* notaciją galima būtų suprasti, kad Deividas yra žmogus. 2 pvz. pateikiamas kodo fragmentas, kaip tai atrodo atvaizdavirus *RDFS* notacija.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.Deividas.lt"
  <rdfs:Class rdf:ID="Žmogus" />
  <rdfs:Class rdf:ID="Deividas">
    <rdfs:subClassOf rdf:resource="#zmogus"/>
  </rdfs:Class>
</rdf:RDF>
```

2 pvz. *RDFS* grafas užrašytas *XML* sintakse

2.2.3.3 *OWL* analizė

Analizuojant ontologijomis sprendžiamus uždavinius pastebėta, kad dabartiniams uždaviniams išspręsti nebepakanka *RDF* ir *RDFS* ontologijų aprašymo kalbų. Informacijos kiekiai didėja ir jiems suvaldyti keliami vis sudėtingesni uždaviniai. Šiems uždaviniams išspręsti reikalinga išraiškingesnė ontologijų kalba. Tam pasitelkiama *OWL* [9]. *OWL* palengvina masinį turinio interpretavimą dar labiau praplėsdama *RDF* žodyną ir semantiką. *OWL* suteikia galimybę detaliau aprašyti turinį, savybių charakteristikas. Taip pat ja aprašomi tarp klasių atsirandantys ryšiai ir nusakomi jų kardinalumas bei ekvivalentiškumas.

OWL kalba pagal sudėtingumą ir išraiškingumą suskirstyta į lygius:

1. *OWL Lite* – palaiko pirmines klasifikavimo ir kardinalumo (0...1) aprašymo galimybes. Naudojamas su paprastais įrankiais.
2. *OWL DL (description logics)* – suteikia maksimalią išraišką, tuo pačiu metu užtikrinant skaičiavimų baigtinumą (visi skaičiavimai bus atlikti), skaičiavimų trukmę (visi skaičiavimai bus atlikti per tam tikrą baigtinį laiką). Šiame kalbos lygmenyje gali būti naudojami visi *OWL* kalbos loginiai elementai, struktūros, tačiau su tam tikrais apribojimais. Pavyzdžiui, klasė *C* gali būti kitų klasių poklasis, bet ne atskiras atvejis.
3. *OWL Full* – suteikia maksimalią išraišką bei laisvą sintaksę. Tokiu būdu skaičiavimų baigtinumas per atitinkamą laikotarpį nėra užtikrinamas.

Kiekvienas toliau einantis *OWL* kalbos lygis yra ankstesnės plėtinys, todėl turi paveldėjęs visas savybes iš ankstesnės. Iš to galime daryti išvadą, kad paprastesne *OWL* lygio kalba aprašyta ontologija bus suprantama ir aukštesnio lygio kalbai.

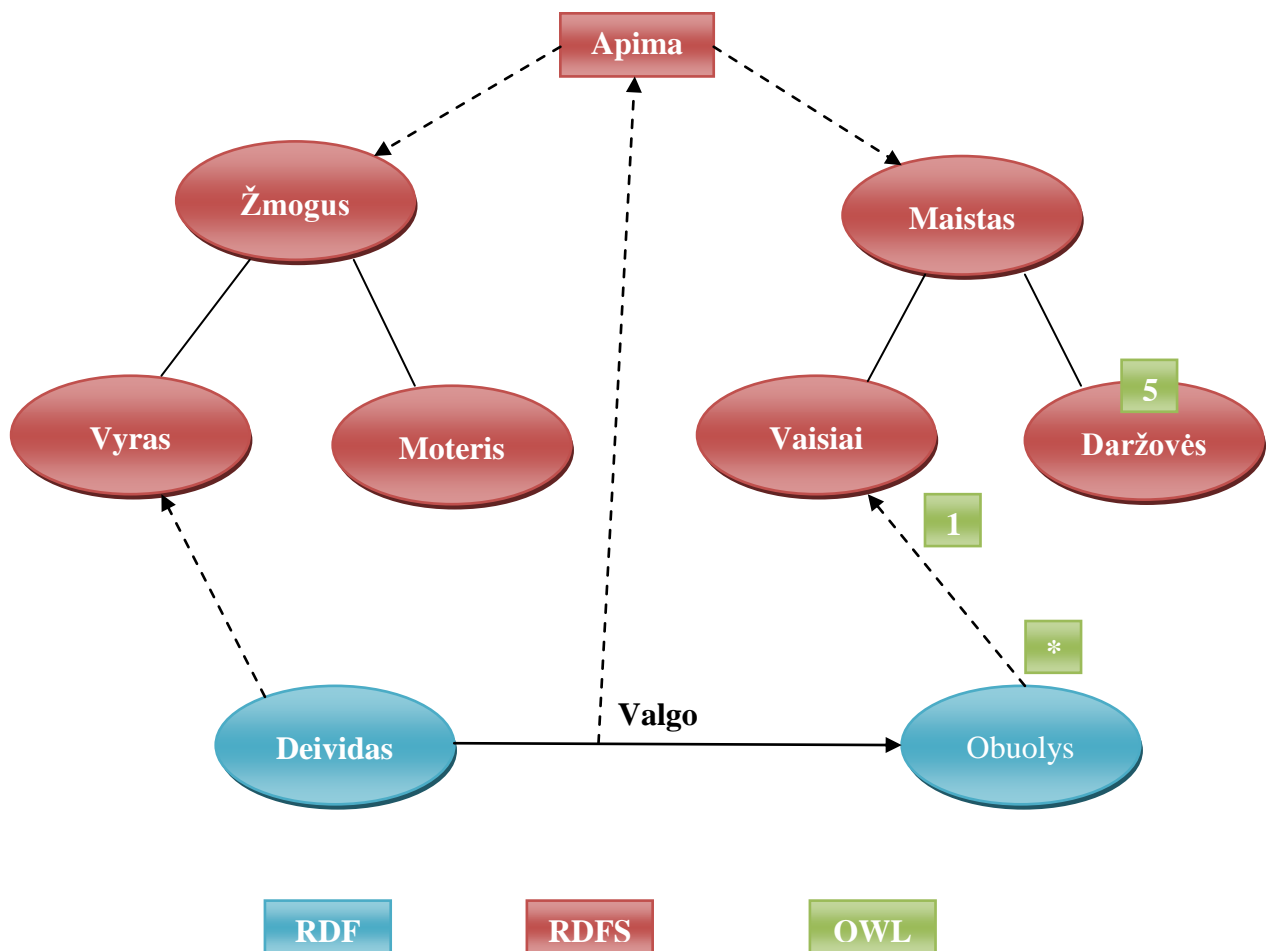
2.2.3.4 *OWL, RDF, RDFS* palyginimas

Analizės metu nustatyta, kad ontologijų aprašymo pradine kalba laikoma *RDF*. Didėja poreikis mašinoms (kompiuteriams) sugebėti efektyviau interpretuoti tekstus, todėl sukurtos kitos ontologijų aprašymo kalbos. Jos sukurtos *RDF* kalbos pagrindu praplėtus biblioteką. 2 lentelėje pateikiamas ontologijų kalbų *RDF/RDFS/OWL* išraiškingumo palyginimas.

2 lentelė. Ontologijų aprašymo kalbų palyginimas

Savybė	RDF	RDFS	OWL
Subjektas	+	+	+
Predikatas	+	+	+
Objektas	+	+	+
Klasės		+	+
Poklasiai		+	+
Sąryšiai tarp klasių		+	+
Nusako kardinalumą			+
Nusako ekvivalentiškumą			+

6 pav. pavaizduota, kaip atskira ontologijų aprašymo kalba lemia bendrą ontologijos modelį.



6 pav. Ontologijų aprašymo kalbų plėtimosi principinis modelis

Mėlynos spalvos elementai vaizduoja *RDF* kalbos elementus, kur $\langle \text{subjektas/Deividas} \rangle \langle \text{predikatas/valgo} \rangle \langle \text{objektas/obuolys} \rangle$. Raudonais elementais pavaizduoti *RDFS* elementai, kurie papildo *RDF* biblioteką. Šiuo atveju, žmogus ir maistas yra klasės, o jų poklasiai atitinkamai – vyras, moteris ir vaisiai, daržovės. Žalios spalvos elementai vaizduoja, kokiais elementais dar papildyta *RDF/RDFS* biblioteka *OWL* ontologijų aprašymo kalboje.

2.3 *SQL* ir *SPARQL* užklausų analizė

2.3.1 *SQL* užklausų analizė

Struktūrizuota užklausų kalba *SQL* – šiuo metu populiariausia duomenų aprašymo ir manipuliavimo kalba reliacinėms duomenų bazių sistemoms valdyti. *SQL* notacija remiasi keliomis pagrindinėmis komandomis, kurias galima suskirstyti į šias grupes:

1. Objektų valdymas:
 - 1.1 CREATE – komanda objektams kurti (pvz.: lentelėms);
 - 1.2 DROP – komanda objektams naikinti.
2. Duomenų atranka:

- 2.1 SELECT – komanda įrašams atrinkti iš vienos ar kelių lentelių;
- 2.2 FROM – komanda nurodo, kokioje lentelėje atrenkami ar ieškomi duomenys;
- 2.3 WHERE – komanda, nurodanti sąlygą, kurią turi tenkinti atrenkama informacija;
- 3. Duomenų valdymas:
 - 3.1 INSERT – komanda naujiems įrašams įterpti;
 - 3.2 UPDATE – komanda vieno ar daugiau įrašų reikšmėms pakeisti;
 - 3.3 DELETE – komanda įrašams iš lentelių pašalinti.

2.3.2 SPARQL užklausų analizė

SPARQL – *RDF* užklausų kalba, skirta duomenimis *RDF* grafuose manipuliuoti semantinėse duomenų bazių valdymo sistemose. Ši kalba kurta 2006 m., o 2008 m. sausio mėn. *W3C* konsorciumo rekomenduota ontologijų užklausų kalba. *SPARQL* sukurta tam, kad vartotojai, turintys žemą techninį žinių lygį ir menką supratimą apie duomenų saugojimą ir struktūrizavimą, galėtų formuoti užklausas. Metodas nepriklausomas nuo programavimo kalbos ir naudojamos platformos.

SPARQL palaiko 4 užklausų formas [10]:

1. *SELECT* – gražina visus arba dalį kintamųjų, kurie buvo susieti užklausos šablone;
2. *CONSTRUCT* – gražina *RDF* grafą, sukonstruotą pagal užklausos šablone paminėtus kintamuosius;
3. *ASK* – nustato, ar egzistuoja faktinė informacija pagal užklausą. Gražinama loginė reikšmė;
4. *DESCRIBE* – gražina *RDF* grafą, kuris aprašo rastus resursus.

SPARQL leidžia užrašyti paprastas užklausas, išreiškiant jas paprastu grafo šablonu: <subjektas><predikatas><objektas>. Kiekvienas trejeto elementas gali būti kintamasis ir sudėtingesnės, naudojančios agreguotus šablonų rinkinius ir papildomus ribojimus, sistemos dalis.

2.3.3 SQL ir SPARQL užklausų palyginimas

Palyginimui analizuojamas uždavinys: reikia rasti, kokią algą uždirba Deividas *Google* koncerne. Paieška atliekama *SQL* užklausa iš reliacinės duomenų lentelės ir *SPARQL* užklausa iš *RDF* grafo.

SQL užklausa:

```
SELECT alga
FROM darbdaviai
WHERE darbdavys = 'Google'
```

3 pvz. *SQL* užklausos fragmentas

Reliacinėje duomenų bazės lentelėje *darbdaviai* ieškomas algos laukelis, kuris priklauso lentelės eilutei, kurioje nurodytas darbdavys *Google*.

SPARQL užklausa:

```
SELECT ?alg
WHERE { darbd:Google ab:alga ?alg . }
```

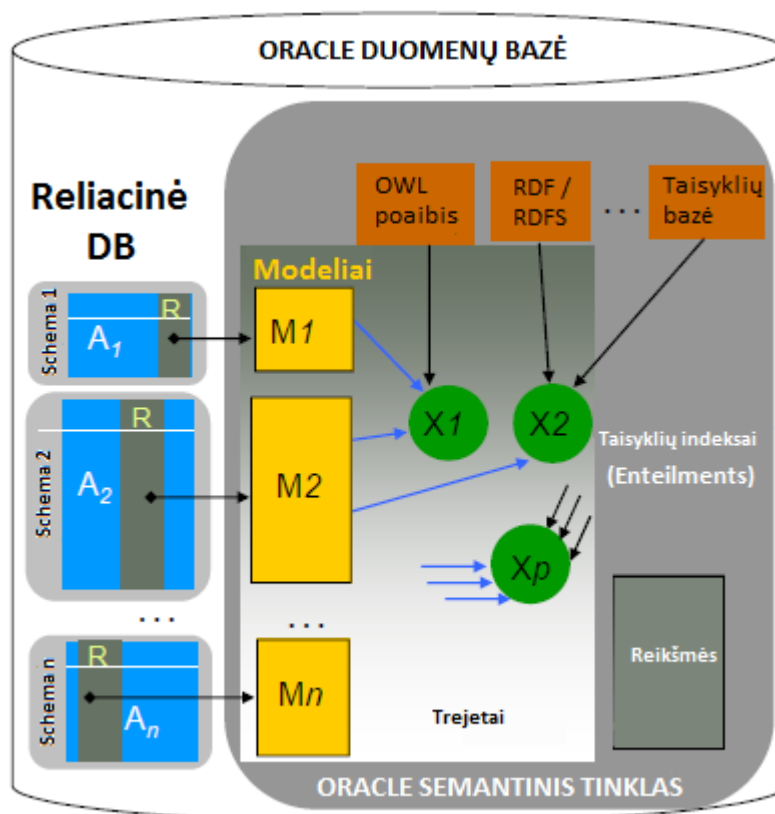
4 pvz. SPARQL užklauso fragmentas

SPARQL užklausoje nurodomi 2 šaltiniai *dardb* ir *ab*. Pagal atitinkamus *RDF* grafo elementus (klases, poklasių) *Google* ir *alga* surandama, kiek Deividas uždirba *Google* kompanijoje.

2.4 Oracle semantinės saugyklos analizė

Oracle semantinė saugykla susideda iš:

1. *RDF* trejetų;
2. duomenų modelių;
3. taisyklių rinkinių;
4. taisyklių indeksų (*Entailments*).



7 pav. Oracle semantinio tinklo architektūra [11]

Semantinis duomenų modelis gali atspindėti reliacinį duomenų modelį ar jo dalį. Modeliuose saugoma semantinė informacija (*RDF* trejetai).

Oracle semantinėje saugykloje naudojami standartiniai taisyklių rinkiniai tokie kaip: *RDF*, *OWL* ir kiti. Taip pat galima kurti ir savo taisyklių rinkinius ir jais apibrėžti semantinius modelius. Taisyklių rinkiniai naudojami apibrėžti duomenis taisyklėmis ir taip duomenų tarpusavio sąryšiams suteikti semantinę prasmę.

Taisyklių indeksai naudojami modelių duomenims indeksuoti specifiniais taisyklių rinkiniais. Kaip ir reliaciniuose duomenų modeliuose indeksai pagreitina duomenų paiešką.

2.4.1 *RDF* modelių įkėlimas

Įkėlti saugoti *RDF* modelius *Oracle* duomenų bazės valdymo sistemoje galima 3 metodais [11]:

1. Masiniu įkėlimu (*Bulk load*), naudojant *SQL* įkelėją (*SQL Loader*);
2. *SQL INSERT* komanda;
3. *Java API* (*Jena*, *Sesame*).

2.4.1.1 Masinio įkėlimo metodas

Semantinius duomenis galima įkelti masiniu metodu, naudojant *staging* lentelę. Šį metodą rekomenduojama naudoti užkraunant didelį kiekį semantinių duomenų trejetų.

Prieš pradėdant naudotis šiuo metodu visų pirma reikia patikrinti duomenų teisingumą, sintaksę. Patikrinus semantiniai duomenys *SQL Loader* įrankiu įkeliami į *staging* lentelę. Tada galima kviesti:

SEM_APIS.BULK_LOAD_FROM_STAGING_TABLE procedūrą, kuri įkelia duomenis iš *stage* lentelės.

5 pavyzdyje atvaizduota *stage* lentelė, įskaitant visus lentelės stulpelius, stulpelių pavadinimus, įrašų parametrus.

```
CREATE TABLE stage_table (  
    RDF$STC_sub varchar2(4000) not null,  
    RDF$STC_pred varchar2(4000) not null,  
    RDF$STC_obj varchar2(4000) not null  
);
```

5 pvz. Tarpinės lentelės *stage* struktūra

SQL Loader procesas iškviečiamas panaudojant komandą:

```
Sqllldr userid=vartotojo_vardas/slaptazodis control=control.ctl data=data.csv
```

6 pvz. *SQL Loader* iškvietimo komanda

Komanda	Aprašymas
<i>control</i>	Komanda nurodo nustatymų failą.
<i>data</i>	Komanda nurodo duomenų failą.
<i>Data.csv</i>	Semantinių duomenų failas su reikiama įkelti duomenimis. Galimi įvairūs duomenų failo tipai.
<i>Controlctl</i>	Duomenų įkėlimo nustatymų failas, kurio struktūra pavaizduota žemiau: <pre>LOAD DATA INFILE * INTO TABLE lenteles_pavadinimas FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' (1 stulpelio pavadinimas, 2 stulpelio pavadinimas, 3 pavadinimas įrašas)</pre>

Sukėlus duomenis į *stage* lentelę, sukuriama semantinis modelis ir paleidžiamas masinis įkėlimas iš *stage* lentelės:

```
Create table manordf_lentele (triple sdo_rdf_triple_s);

Exec sem_apis.create_sem_model('manordf', 'manordf_lentele' , 'triple');

Exec sem_apis.bulk_load_from_staging_table(manordf, 'schemas_pavadinimas',
'staging', flags=>'parallel_create_index parallel=4');
```

7 pvz. Semantinių duomenų įrašymas SQL Loader metodu

4 lentelė. Pateikto 7 pvz. aprašymas

Komanda	Aprašymas
<i>sdo_rdf_triple_s</i>	Semantinių duomenų tipas, konstruktorius.
<i>sem_apis.create_sem_model</i>	Komanda, sukurianti semantinį modelį.
<i>manordf</i>	Semantinio modelio pavadinimas.
<i>manordf_lentele</i>	Lentelė, kurioje saugomos nuorodos į semantinius duomenis.
<i>triple</i>	Stulpelio <i>sdo_rdf_triple_s</i> tipo pavadinimas.
<i>sem_apis.bulk_load_from_staging_table</i>	
<i>parallel_create_index</i>	Aktyvuoja vidinį Oracle indeksą, kuris sukuria galimybę duomenis įkelti lygiagrečiai, taip paspartindamas įkėlimo procesą.

Masiniu įkėlimu (*Bulk load*) įkelti semantinius duomenis taip pat galima kitais Java paketais:

1. *Jena* (*OracleBulkUpdateHandler.addInBulk(...)* metodu);
2. *Sesame* (*OracleBulkUpdateHandler.addInBulk* metodu).

2.4.1.2 SQL INSERT komanda

Norint įkelti nedidelį kiekį semantinių duomenų trejetų, tai atlikti rekomenduojama naudojant *SQL INSERT* komandą. Duomenys turi atitikti tam tikrus kriterijus:

1. *URI* duomenų įrašas turi prasidėti ir baigtis laužtiniais skliausteliais <>;
2. tušti mazgai išskiriami brūkšneliu ir dvitaškiu _;
3. tekstas išskiriamas kabutėmis "";
4. tarpo simboliai yra negalimi *URI* duomenų įrašuose ir tuščiuose mazguose.

Užkraunant semantinius trejetus šiuo metodu taip pat naudojamas *SDO_RDF_TRIPLE_S* konstruktorius. Duomenys įkeliami naudojant standartinę *SQL INSERT* metodą, reikšmėms priskiriant konstruktorių su semantiniiais duomenimis kaip parodyta 8 pavyzdyje.

```
INSERT INTO RDF_DUOMENYS VALUES (SDO_RDF_TRIPLE_S('RDF_MODELIO_ID',  
'<SUBJEKTAS>', '<OBJEKTAS>', '"PREDIKATAS"'));
```

8 pvz. Semantinių duomenų įrašymas *SQL INSERT* metodu

2.4.2 SQL, SPARQL užklausų vykdymo prototipas

Norint manipuluoti duomenimis reliacinėje ir semantinėje saugyklose reikia vykdyti 2 užklausas:

1. *SQL* reliacinėje duomenų bazėje;
2. *SPARQL* semantinėje duomenų bazėje.

5 lentelėje, 8 pav. ir 9-10 pvz. pateikti įvardintų saugyklų duomenų modeliai su duomenimis ir užrašytomis užklausomis. *SQL* ir *SPARQL* užklausos rezultatai vienodi. Jie atrinks, koks darbuotojas dirba *Oracle* kompanijoje ir kiek uždirba.

- Reliacinis duomenų modelis **suvestinė**:

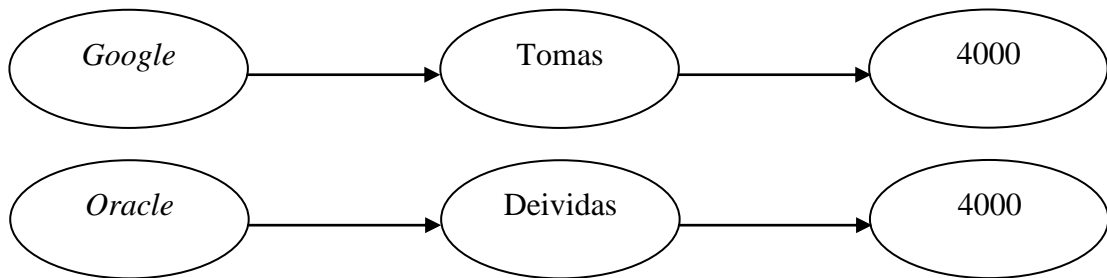
5 lentelė. Reliacinis duomenų modelis „suvestinė“

Darbdavys	Darbuotojas	Alga
Google	Tomas	4000
Oracle	Deividas	4000

```
SELECT darbuotojas, alga  
FROM suvestinė  
WHERE darbdavys = 'Oracle';
```

9 pvz. Darbuotojų atranka *SQL* notacija

- Semantinis duomenų modelis **suvestinė** (2 *RDF* trejetai):



8 pav. Semantinis duomenų modelis „suvestinė“

```

SELECT ?darbuotojas, ?alga
WHERE {
  darbdavys:Oracle
  ab:darbuotojas ?darbuotojas
  ab:alga ?alg
}

```

10 pvz. Darbuotojų atranka *SPARQL* notacija

Vykdam bet kurią iš aukščiau aprašytų užklausų atrenkamas atsakymas: **Deividas 4000**. Tačiau siekiant vykdyti abi užklausas vienu metu susiduriama su problemomis:

1. Nepatogu, kadangi reikia vykdyti 2 atskiras užklausas;
2. *SQL* interpretatoriai nepalaiko *SPARQL*, tad tenka užklausas vykdyti per skirtingus interpretatorius;
3. Sudėtingas duomenų integravimas.

Šias problemas iš esmės galima išspręsti pasinaudojus viena iš *Oracle* semantinio tinklo procedūrų *SEM_MATCH* [12]. *Oracle* duomenų bazės valdymo sistemoje *SQL* užklausose galima kreiptis į numatytas procedūras. *SEM_MATCH* procedūra *SQL* užklausų kalba praplečia galimybę manipuluoti duomenimis semantinėse saugyklose. Ši funkcija turi šiuos atributus:

- | | |
|------------------|-----------------------------|
| 1. užklausa | <code>VARCHAR2;</code> |
| 2. modelis | <code>SEM_MODELS;</code> |
| 3. taisyklės | <code>SEM_RULEBASES;</code> |
| 4. slapyvardžiai | <code>SEM_ALIASES;</code> |
| 5. filtras | <code>VARCHAR2;</code> |
| 6. parinktis | <code>VARCHAR2.</code> |

Užklausos atributas funkcijoje yra privalomas, kiti neprivalomi (gali būti *NULL*). Pasinaudojant *SEM_MATCH* funkcija *SPARQL* užklausa galima įterpti į *SQL*. Kaip tai atrodo, pateikta 11 pavyzdyje.

```

SELECT darbuotojas, alga
FROM table (sem_match(
  `SELECT ?darbuotojas, ?alga
WHERE { darbdavys:Google
        ab:darbuotojas ?darbuotojas
        ab:alga ?alg
      }`,
  SEM_MODEL(`suvestinė`)
))

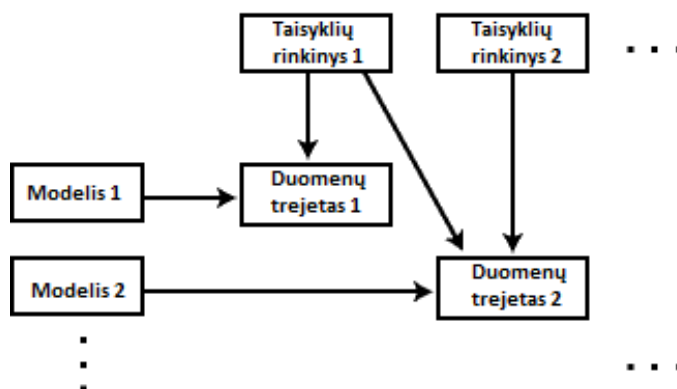
```

11 pvz. Darbuotojų atranka sudėtine *SQL-SPARQL* užklausa

2.4.3 *RDF* modelio papildymas

RDF trejetai nepasižymi jokiais semantiniiais ryšiais. *RDF* trejetais paremtą modelį reikia aprašyti taisyklėmis (*rules*) ir taisyklių rinkiniais (*rulebases*). Šie objektai saugomi *SPATIAL RDF* duomenų modelyje (aprašytas 2.4.4 skyriuje).

Taisyklėmis aprašytame modelyje galima konstruoti užklausas, kurios vykdys semantinę paiešką, remiantis prasmingais ryšiais tarp duomenų, o ne tik nurodant konkrečius ieškomus stulpelius [13]. 9 pav. pateikta principinė schema, kur pavaizduoti duomenų trejetai aprašyti taisyklių rinkiniais. Duomenų bazės gali turėti bet kokių skaičių semantinių modelių, duomenų trejetų ir taisyklių rinkinių. Užklaustos interpretatorius gali padaryti išvadas ir atrinkti duomenų trejetus, esančius keliuose semantiniuose modeliuose pagal tą patį taisyklių rinkinį ar keletą taisyklių rinkinių.



9 pav. Taisyklių rinkinių schema

Taisyklės yra objektas, kuris gali būti pritaikytas daryti išvadas iš semantinių duomenų. Taisyklės identifikuojamos pagal vardą ir susideda iš:

1. *IF* šalutinio modelio praeičiai;

2. laisvai pasirenkamo filtro salygos, kuri toliau suvaržo subgrafus, priderintus prie *IF* šalutinio modelio;
3. *THEN* šalutinio modelio rezultatams.

Žemiau pateiktas 12 pvz, kur konferencijai pirmininkaujantis asmuo yra taip pat ir konferencijos recenzentas.

```

('PrimininkaujantisAsmuoTaisykle', -- Taisykles pavadinimas
 '(?r :PirmininkaujantisAsmuoOf ?c)', -- IF šalutinis modelis
 NULL, -- filtro salyga
 '(?r :RecenzentasOf ?c)', -- THEN šalutinis modelis
 SEM_ALIASES (SEM_ALIAS('', 'http://test.lt/test/'))
 )

```

12 pvz. SPARQL fragmentas, aprašytas taisyklėmis

Šiuo atveju taisyklė neturi filtravimo. Tad atvaizdavimo komponentas yra *NULL*. Geresniam rezultatui pasiekti reikia naudoti vieno trejeto modelį *THEN* šalutinei taisyklei. Jei *THEN* šalutinėje taisyklėje yra kelių trejetų modelis, galima tokį modelį išskaidyti į keletą taisyklių su vieno trejeto *THEN* modeliu.

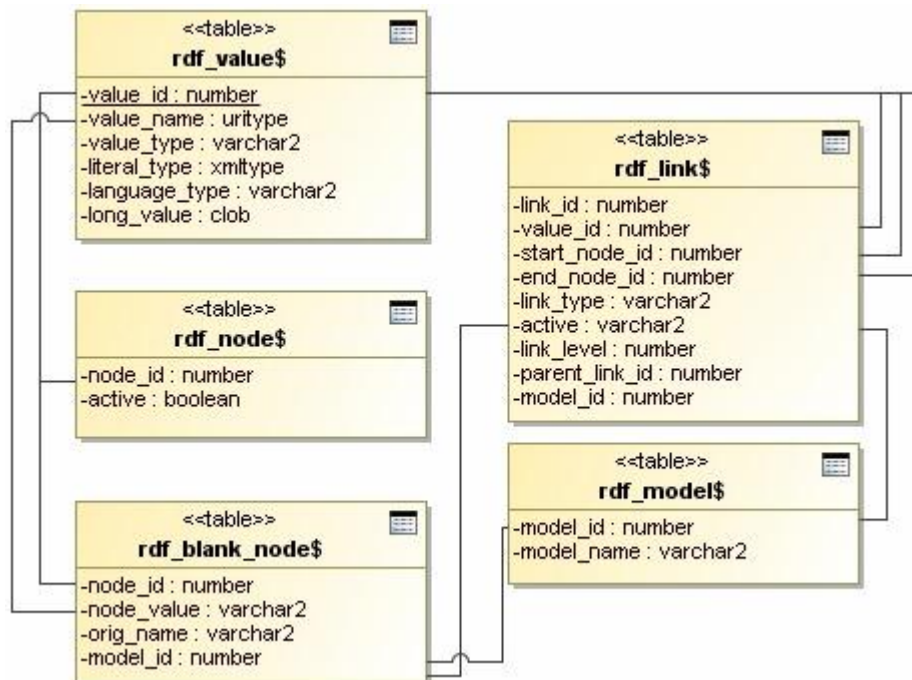
Taisyklių rinkiniai (*rulebases*) yra objektas, turintis daugiau nei vieną taisyklę. Žemiau pateikiami *Oracle* siūlomi taisyklių rinkiniai:

1. *RDFS*;
2. *RDF* (*RDFS* pogrupyje);
3. *OWLSIF*;
4. *RDFS++*;
5. *OWLPrime* ;
6. *SKOSCORE*.

RDF ir *RDFS* taisyklių rinkiniai sukuriama iškvietus *SEM_APIS.CREATE_SEM_NETWORK* procedūrą semantiniam tinklui sukurti. *RDFS* taisyklių rinkinys įgyvendina *RDFS* paveldėjimo (*entailment*) taisykles, apibūdintas *W3C* konsorciumo. *RDF* taisyklių rinkinys atstovauja *RDF* paveldėjimo taisyklėms, kurios yra *RDFS* paveldėjimo taisyklių poaibyje. Pamatyti minėtų taisyklių rinkinių turinį galima nagrinėjant *MDSYS.SEMR_RDFS* ir *MDSYS.SEMR_RDF* vaizdus. Taip pat galima sukurti savo vartotojo apibrėžtą taisyklių rinkinį, panaudojant *SEM_APIS.CREATE_RULEBASE* procedūrą. Sukurtas taisyklių rinkinys leidžia papildomai apibrėžti išvadų darymo galimybes.

2.4.4 Oracle SPATIAL RDF duomenų modelis

Semantini duomenys saugomi *Oracle Spatial RDF* duomenų modelyje, *MDSYS* sistemoje, pagal 10 pav. pavaizduotą schemą. Vartotojo duomenų bazės sistemoje saugomos tik nuorodos į duomenų elementus [14].



10 pav. Oracle RDF saugyklos duomenų modelis [15]

Kiekvienas *RDF* trejetas yra išskaidomas ir saugomas *MDSYS* sistemoje [15]. Lentelėje „rdf_model\$” saugoma informacija (modelio pavadinimas ir identifikacinis numeris) apie visus *RDF* ir *OWL* modelius duomenų bazėje.

Kuriant naują *RDF* modelį, visų pirma automatiškai sukuriamas unikalus modelio identifikatorius – „model_id” ir kartu su modelio pavadinimu įrašas išsaugomas „rdf_model\$” lentelėje. Įrašant semantinius duomenis kiekvienas tekstinis *RDF* trejeto elementas saugomas lentelėje „rdf_value\$”. Tekstinės reikšmės išsaugomos *value name* lauke. Kiekvienas įrašas saugomas unikaliai, jeigu tokios *RDF* grafo reikšmės jau yra lentelėje, tuomet nauji įrašai nekuriami, o naudojami esami. *RDF* trejetai duomenys surašami į grafą: subjektai ir objektai saugojami kaip mazgai, o savybės kaip nuorodos. *RDF* trejetas (subjektas, predikatas, objektas) – traktuojamas kaip vienas *DB* objektas. Semantinių duomenų valdymas vyksta centralizuotai – pagrindinėje schemeje, kartu kur yra realizuoti duomenų apdorojimo procedūros, funkcijos ir šablonai duomenų paieškos užklausoms formuoti.

Lentelės „rdf_links“ ir „rdf_node” skirtos *Spatial* duomenų modelio tinklui palaikyti. Šis tinklas užtikrina *RDF* duomenų palaikymą grafe. Subjekto mazgai yra pradiniai nuorodų mazgai, o objekto mazgai – galiniai. Ši informacija saugoma lentelėje „rdf_links“, o lentelėje „rdf_blank_node\$” yra saugoma informacija apie tuščius mazgus. „rdf_node” – saugo tik

identifikatorių „node_id”, kuris yra toks pat kaip mazgų (subjektų, objektų) identifikatorius „value_id”.

2.5 Vartotojų analizė

Šiame tyrime analizuojami ir modeliuojami metodai, darbui su semantiniiais ir reliaciniais duomenimis *Oracle DBVS* saugykloje, yra naudingi asmenims, administruojantiems duomenis. Pagrindinis šių vartotojų tikslas – patogiai, mažomis techninėmis, materialinėmis ir techninių žinių sąnaudomis tinkamai administruoti ir pateikti duomenis kitai interesantų grupei.

Taip pat galima išskirti ir kitą vartotojų aibę – asmenis, kurie suinteresuoti pasinaudoti duomenų bazėse sukaupta informacija. Pagrindinis šių vartotojų tikslas – iš sistemos pagal reikiamus kriterijus gauti informaciją nepaisant to, kokioje realizacijoje (reliacinėse lentelėse ar ontologijose) ji yra saugoma.

2.6 Programinių paketų, reliacinių ir semantinių saugyklų duomenų integravimo sistemai kurti, analizė

2.6.1 Operacinė sistema

Analizuojant operacines sistemas, kuriose dažniausiai realizuojami sprendimai, taikant *Oracle DBVS*, pastebėta, kad dauguma realizuotų sprendimų apsiriboja *Linux* ir *Windows* šeimų operacinėmis sistemomis. Tolimesnėje analizėje pateikiamas šių *OS* palyginimas.

6 lentelėje palyginamos *Linux* ir *Windows* operacinės sistemos. Operacinė sistema, neturinti kriterijuose nurodytos funkcijos ar funkcionalumo, vertinama 0 taškų. Jei atitinkamą funkciją turi abi operacinės sistemos tai tokiu atveju vertinimas priklauso nuo funkcijos ar funkcionalumo panaudojimo efektyvumo. *OS*, efektyviau panaudojanti funkciją, vertinama 2 taškais, atitinkamai 1 tašku vertinama ne taip efektyviai funkciją panaudojanti *OS*.

2.6.1.1 Microsoft Windows ir Linux OS palyginimas

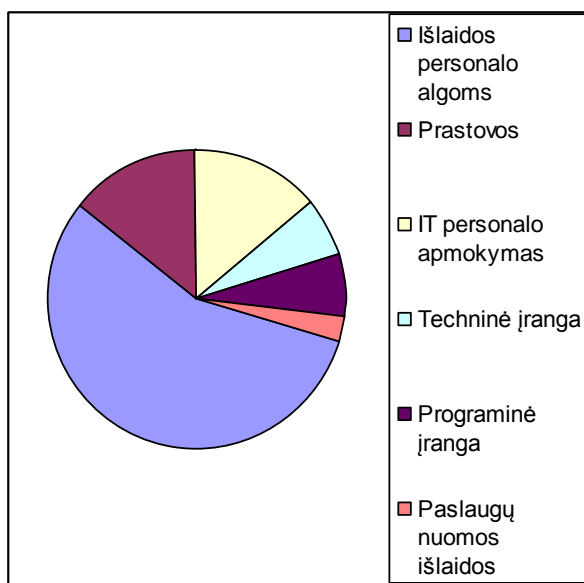
Linux ir *Windows* operacinės sistemos analizuojamos šiais aspektais:

1. kaina;
2. greitaveika;
3. saugumas;
4. lankstumas;
5. valdymas;
6. suderinamumas.

Kaina

Dauguma *Linux* distribucijų yra nemokamos. Tokių operacinių sistemų verslas pagrįstas mokamomis metinėmis *OS* priežiūros prenumeratomis. Tad sakyti, kad bus taupoma diegiant sistemas su *Linux OS*, yra klaidinga. Tokių sistemų priežiūrai ir IT specialistams paprastai skiriami didesni finansiniai ištekliai nei *Windows OS*.

Bendroji nuosavybės kaina yra naudingas matas, skaičiuojant IT sprendimo išlaidas per tam tikrą laiką. Tačiau, kaip teigia tyrimų bendrovė *IDC* 2007 metais atliktame tyrime „*Three Year Server TCO*“ [16], programinės įrangos kaina trijų metų laikotarpyje sudaro tik 7 % sistemos kaštų. Didžiąją dalį išlaidų sudaro sistemos palaikymas.



11 pav. Įmonės IT sistemos kaštų palyginimas

Sudėjus išlaidas personalo algoms, prastovoms ir apmokymams, susidaro 83 % visų sistemos išlaidų. Tad esminis kaštų mažinimas galimas optimizuojant minėtas sritis.

Greitaveika

Linux OS paremta komandinės eilutės vartotojo sąsaja, o *Windows OS* – grafine vartotojo sąsaja. Pastaroji reikalauja daugiau kompiuterio resursų, todėl *Oracle DBVS* efektyvumas mažėja.

Windows palaiko daugiaprogramį [17] (*multitasking*) režimą, bet nėra kolektyvioji (*multiuser*) *OS*. Tuo tarpu *Linux* palaiko daugiaprogramį (*multitasking*) režimą ir yra kolektyvioji (*multiuser*) bei daugiagijė (*multithreading*) *OS*.

Saugumas

Didesnė tikimybė susidurti su grėsmėmis yra *Windows OS*, kadangi dauguma kenkėjiškų sprendimų realizuota *Windows* sistemoms pakenkti, todėl naudojant *Linux OS* tikimybė susidurti su kenkėjiškais puolimais yra mažesnė.

Tačiau kai kalbama apie grėsmes saugai didelėse sistemose, jokių ypatingų apsaugos priemonių nėra. Bet kuriam nuolat veikiančiam sprendimui reikia kokybiškos technologijos, kruopščių saugos valdymo procesų ir visapusiškų pastangų kovoti su grėsmėmis, naudojantis identifikavimo valdymu ir prieigos kontrole.

Lankstumas

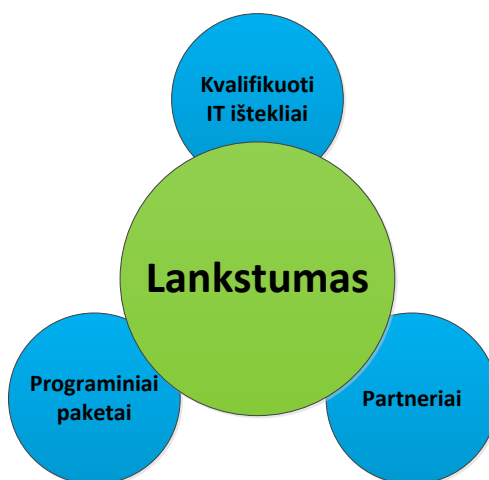
Planuojant ir vykdam bet kokius IT sprendimus, susiduriama su įvairiomis problemomis ir išskyla įvairių klausimų:

1. Kuris papildomas programinis paketas labiausiai atitiks organizacijos verslo reikalavimus?
2. Ar turime savų įgūdžių tai įgyvendinti? Ar galime samdyti tinkamos kvalifikacijos specialistus?
3. Ar galime rasti partnerių, kurie padėtų planuoti ir diegti IT sprendimus, atsižvelgdami į realią patirtį su tokiomis kaip mūsų įmonėmis ir scenarijais?
4. Ar sistema pakankamai lanksti ateities pokyčiams ir iššūkiams?

Serverio OS platforma, kuri pasirenkama, turi tiesioginės įtakos kiekvienam paskesniai organizacijos priimtam sprendimui – tiek taikant naują programą, tiek ieškant partnerio ar rūpinantis kvalifikuotais specialistais. Taip pat tai turi įtakos ateities sistemoms, patobulinimams, praplėtimams.

Sistemai vystyti ir ilgalaikei panaudai didžiausią įtaką turi 3 komponentai (12 pav.):

1. kvalifikuoti IT ištekliai – gebantys užtikrinti sistemos nepertraukiamą veiklą;
2. partneriai – sustiprinantys organizacijos silpnąsias vietas;
3. programiniai paketai – pasiekiami, atsinaujinantys, gebantys užtikrinti organizacijos poreikius.



12 pav. OS lankstumą labiausiai lemiantys veiksniai

Analizė parodė, kad *Microsoft* kompanija, subūrusi didžiausią partnerių ir aukštos kvalifikacijos IT išteklius, leidžiančius klientams turėti didžiausią savų bei trečiųjų šalių sukurtų programinių paketų pasirinkimą. Kaip teigia *Microsoft* kompanija, *Windows* platformą palaiko:

1. tūkstančiai įvairių taikomųjų programinių paketų;
2. daugiau kaip 750 000 *Microsoft* partnerių;
3. daugiau nei 450 000 specialistų, turinčių *Microsoft* sertifikuoto sistemų inžinieriaus (*MCSE*) laipsnį;
4. daugiau nei 6 milijonai programų kūrėjų;
5. didžiausia pasaulyje nepriklausomų programinės įrangos pardavėjų bendruomenė.

Lyginant su *Microsoft*, konkreti *Linux* distribucija negali užtikrinti tokios gausos papildomų programinių paketų palaikomumo. Taip pat dauguma šiuolaikinių sprendimų nėra realizuojami *Linux* platforma arba realizuojama vėliau. Aktualių programinės įrangos paketų nebuvimas arba vėlesnis realizavimas sumažina organizacijos konkurencingumą.

Valdymas

Lyginant *Windows OS* aplinką su *Linux OS* pastebėta, kad *Windows OS* aplinkos valdymas yra paprastesnis. Dauguma nustatymų ar parametrų pakeitimai atliekami interaktyvioje vartotojo sąsajoje, trečiųjų šalių programinių paketų įdiegimas ir valdymas paprastas, nereikalauja išskirtinių žinių. Tuo tarpu šiuos veiksmus atlikti *Linux OS* aplinkoje kur kas sudėtingiau. Dauguma veiksmų atliekama komandinės eilutės vartotojo sąsajoje, o siekiant įdiegti pašalinius programinius paketus dažnai reikia papildomų programinių paketų, branduolio (*kernel*) bei daug kitų nustatymų. Šie veiksmai atliekami mažiau interaktyvioje aplinkoje ir paprastai reikalauja didesnių gebėjimų ir didesnės kompetencijos IT specialistų.

Suderinamumas

Windows šeimos *OS* užima daugiau kaip 85 % [18] operacinių sistemų rinkos. Todėl taikomųjų programinių paketų kūrėjai visų pirma savo programas realizuoja *Windows* aplinkoje. Be to, *Microsoft* padeda programas suderinti su *Windows OS*, siūlydami nemokamus testavimo įrankius, kurie padeda įmonės IT specialistams, programų kūrėjams ir nepriklausomiems programinės įrangos pardavėjams testuoti trečiųjų šalių programinius paketus ir suderinamumą su *Windows OS*. Taip pat *Microsoft* yra sukūrusi funkcijas, gebančias palaikyti senesnių versijų programinius paketus. Tokio suderinamumo *Linux OS* neturi.

Išvados

Išanalizavus *Linux* ir *Windows* operacinių sistemų esminius aspektus (kainą, greitaveiką, saugumą, lankstumą, valdymą ir suderinamumą), kiekvienas aspektas įvertintas taškais. Vertinimai pateikti 6 lentelėje.

6 lentelė. *Linux* ir *Windows OS* palyginimas

Kriterijus	<i>Linux</i>	<i>Windows</i>
Kaina	1	1
Greitaveika	2	1
Saugumas	2	2
Lankstumas	1	2
Valdymas	1	2
Suderinamumas	1	2
Iš viso:	8	10

Rezultatai rodo, kad *OS* kaina labai mažai lemia sistemos kaštus. Taip pat ir saugumas didžiaja dalimi priklauso nuo technologinių sprendimų ir IT specialistų kompetencijos, kuriant ir administruojant sistemas. Šiose srityse tiek *Linux*, tiek *Windows* sistemos išskirtinumų neturi. *Linux OS* dėl savo technologinių sprendimų išsiskiria tik greitaveika. Tačiau nepasižymi tokiu lankstumu, valdymu ir suderinamumu kaip *Windows* operacinės sistemos.

2.6.2 Modeliavimo įrankiai

Tyrimo modeliams braižyti pasirinkti 2 modeliavimo paketai:

1. *Oracle Data Modeler* – duomenų modeliams vizualizuoti ir modeliuoti;
2. *MagicDraw* paketai – metodų modeliams modeliuoti ir vizualizuoti.

Šių paketų pasirinkimą lėmė darbo patirtis, pakankamas funkcionalumas reikiamiems modeliams modeliuoti. Dėl šių priežasčių kiti modeliavimo paketai neanalizuoti.

2.7 Analizės išvados

1. Atlikta *Oracle* semantinės saugyklos analizė parodė, kad gerai išvystyta *Oracle* semantinės dalies infrastruktūra suteikia geras galimybes saugoti semantinius duomenis ir jais manipuluoti;
2. analizės metu nustatyti tinkama programinė aplinka ir taikomieji paketai eksperimentui vykdyti;
3. išanalizavus *Oracle* semantinę ir reliacinę saugyklas, nuspręsta sumodeliuoti 2 metodus:
 - 3.1 semantinių duomenų saugojimo;
 - 3.2 semantinių ir reliacinių duomenų integravimo.

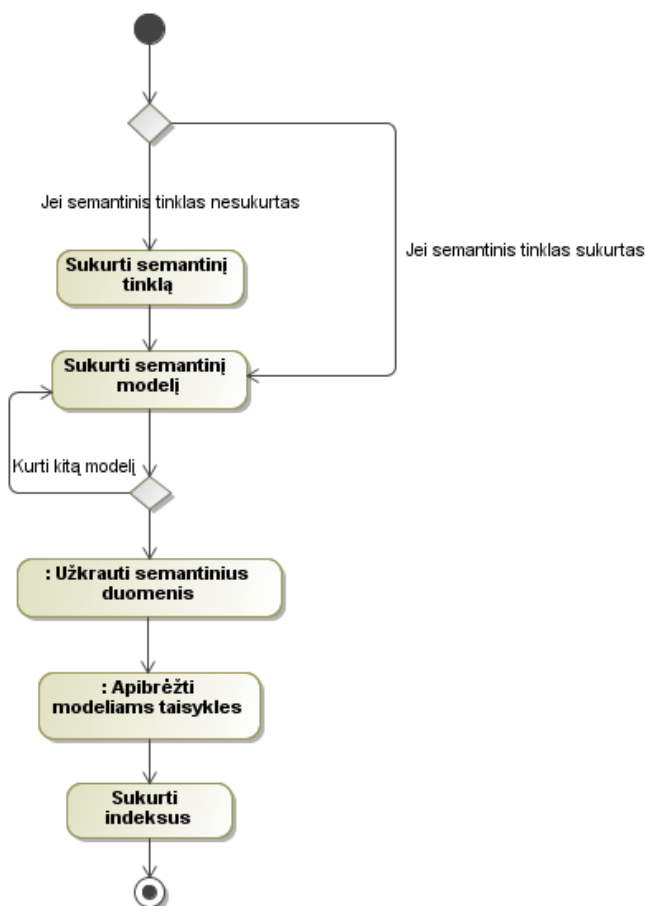
3. Semantinių duomenų saugojimo metodas

3.1 Semantinių duomenų saugojimo metodo procesų modeliai

Semantinių duomenų saugojimo metodas sumodeliuotas organizacijoms, siekiančioms įsidiegti ir eksploatuoti semantines saugyklas *Oracle* duomenų bazės valdymo sistemoje.

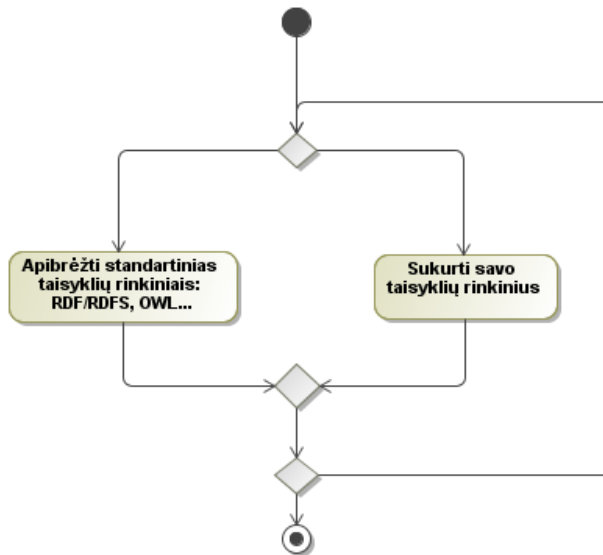
Vienas svarbiausių procesų semantinės saugyklos eksploatavime yra duomenų paruošimas. Duomenų įrašymo ir paruošimo procese (13 pav.) yra svarbūs 5 elementai:

1. semantinis tinklas;
2. semantiniai modeliai;
3. semantinių duomenų įrašymas;
4. interpretavimo taisyklės;
5. duomenų indeksavimas.



13 pav. Semantinių duomenų rašymo į *Oracle DBVS* proceso diagrama

Galima naudoti standartinius taisyklių rinkinius: *RDF*, *OWL* ir kitus, arba sukurti savo (14 pav.).



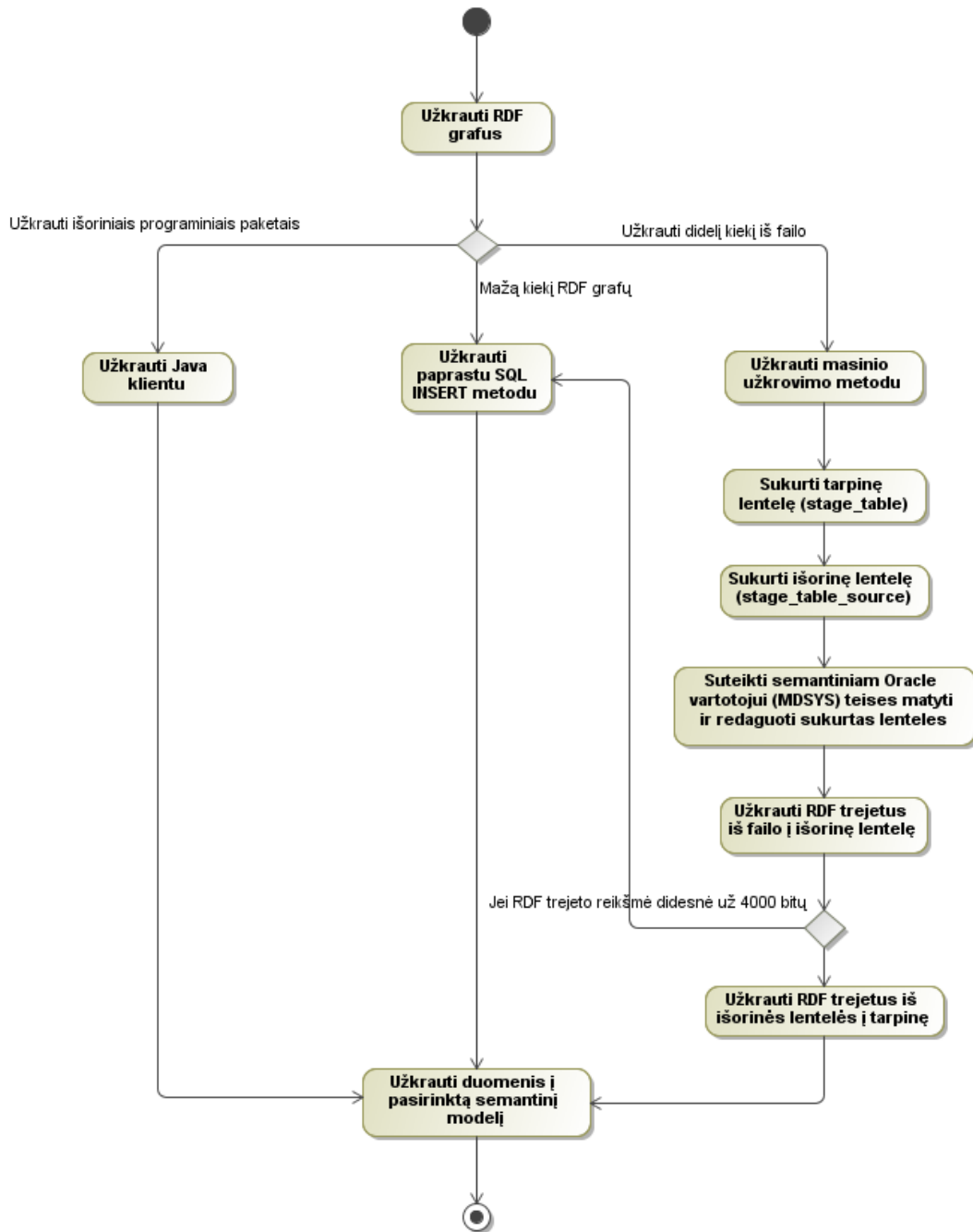
14 pav. Taisyklių priskirimo proceso diagrama

Kaip aptarta 2.4.1 skyriuje, *RDF* modelių įkėlimas vykdomas 3 metodais:

1. *RDF* įrašomi *Java* klientu;
2. *RDF* įrašomi paprastu *SQL INSERT* metodu;
3. *RDF* įrašomi masinio užkrovimo metodu.

Kaip pateikta semantinių duomenų užkrovimo procese (15 pav.), sudėtingiausiai realizuojamas masinio užkrovimo metodas. Tačiau jis yra sparčiausias nei kiti įvardinti. Didesnę spartą lemia šie veiksniai:

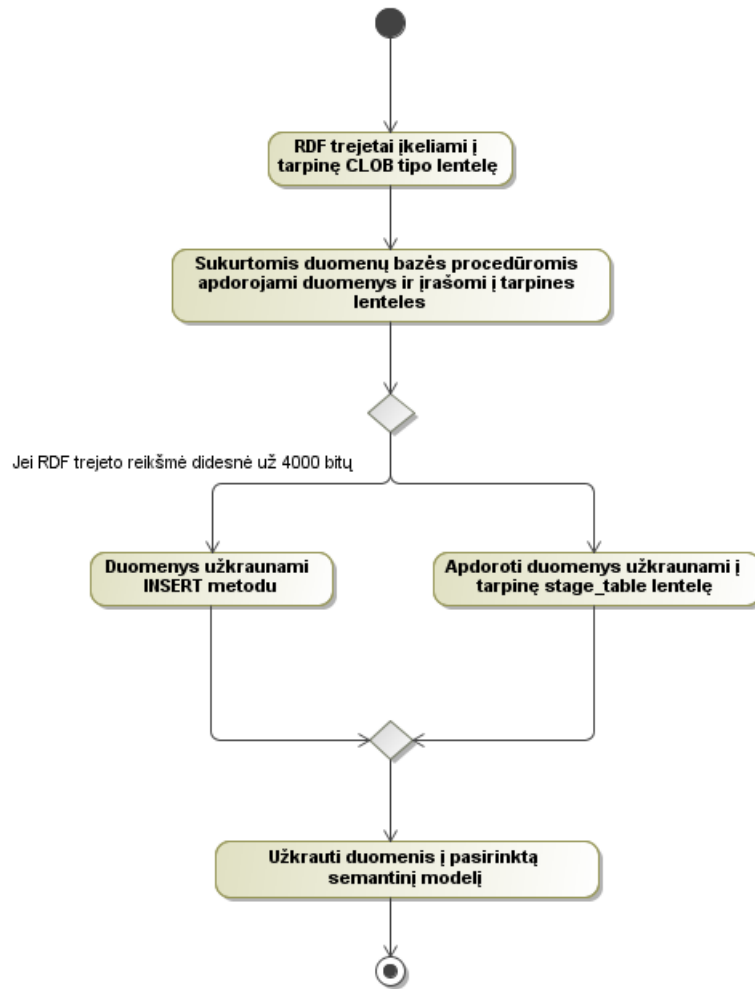
1. lygiagreti semantinių trejetų įrašymo galimybė;
2. tiesioginiame įrašyme nedalyvauja trečiųjų šalių programinė įranga. Todėl neprarandama laiko prisijungimams prie duomenų bazės.



15 pav. Semantinių duomenų įkėlimo proceso diagrama

Analizės metu nustatytas masinio įkėlimo metodas *SQL LOADER* įrankiu yra pakankamai senas. Šiuolaikinės *DBVS* funkcionalumas palaiko išorinių duomenų apdorojimą *DB* lygyje. Todėl nuspręsta patobulinti masinio įkėlimo metodą, atsisakant *SQL LOADER* įrankio ir sukūrus šį funkcionalumą *DB* lygyje.

Funkcionalumas realizuotas „duomenų įkėlimas“ pakete. Sukurtų procedūrų atliekamų veiksmų schema pateikta 16 pav.

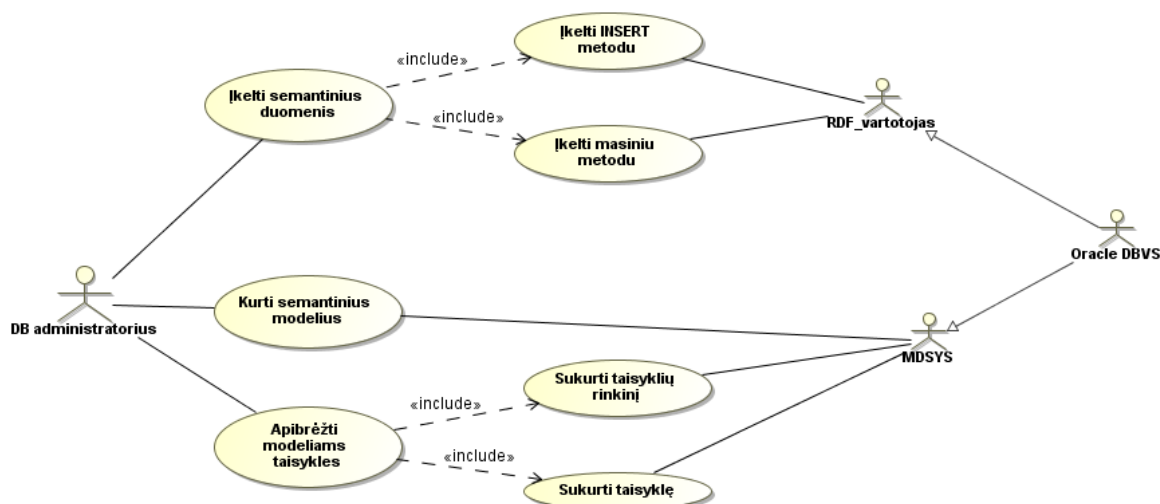


16 pav. Duomenų įkėlimo masiniu būdu DB lygyje veiksmų schema

3.2 Semantinių duomenų saugojimo metodo panaudojimo atvejai

Semantinių duomenų saugojimo panaudojimo atvejų modelyje (17 pav.) pateikti visi galimi sistemos panaudojimo atvejai. Semantinius duomenis administruojančiam asmeniui aktualūs trys pagrindiniai panaudojimo atvejai:

1. semantinių modelių kūrimas;
2. semantinių duomenų įkėlimas;
3. taisyklių apibrėžimas *RDF* modeliams.



17 pav. Semantinių duomenų paruošimo panaudojimo atvejų diagrama

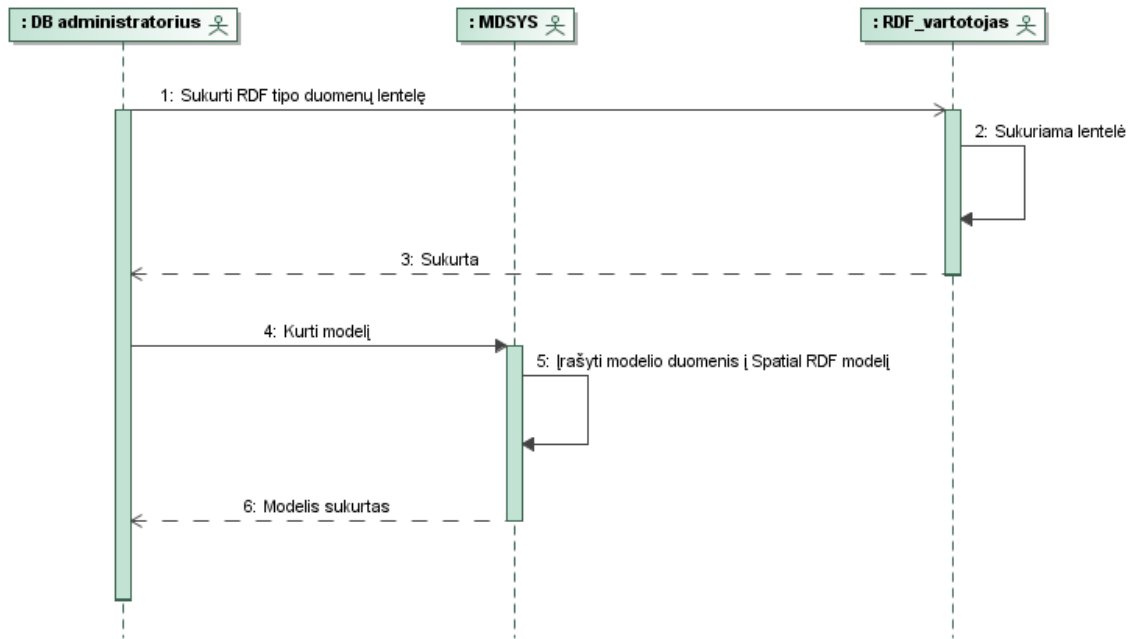
Tolimesniuose punktuose aprašomas kiekvieno panaudojimo atvejo specifika ir elgsena.

3.2.1 Panaudojimo atvejis „kurti semantinius modelius“

7 lentelėje pateikta panaudojimo atvejo „kurti semantinius modelius“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 18 pav.

7 lentelė. Panaudojimo atvejo „kurti semantinius modelius“ specifikacija

PA „Kurti semantinius modelius“		
Tikslas. Sukurti <i>RDF</i> modelį semantiniams duomenims saugoti		
Aprašymas. Semantiniai duomenys saugomi <i>RDF</i> modeliuose. Semantinius duomenis galima įrašyti esamuose modeliuose arba sukurti naujus.		
Prieš sąlygą	Vartotojas įveda modelio pavadinimą	
Aktorius	Duomenų bazės administratorius	
Sužadinimo sąlyga	Kuriant naują <i>RDF</i> modelį	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda modelio pavadinimą	Sistema priskiria įvestą modelio pavadinimą modelio pavadinimo ir <i>RDF</i> duomenų lentelės kintamiesiems	
2. Aktyvuojamas <i>RDF</i> duomenų lentelės kūrimo procesas	Sukuriamas <i>RDF</i> tipo duomenų lentelė	
3. Aktyvuojamas <i>RDF</i> modelio kūrimo procesas	Sukuriamas <i>RDF</i> modelis	



18 pav. Panaudojimo atvejo „kurti semantinius duomenis“ sekų diagrama

3.2.2 Panaudojimo atvejis „įkelti semantinius duomenis“

8 lentelėje pateikta panaudojimo atvejo „įkelti semantinius duomenis“ detali specifikacija.

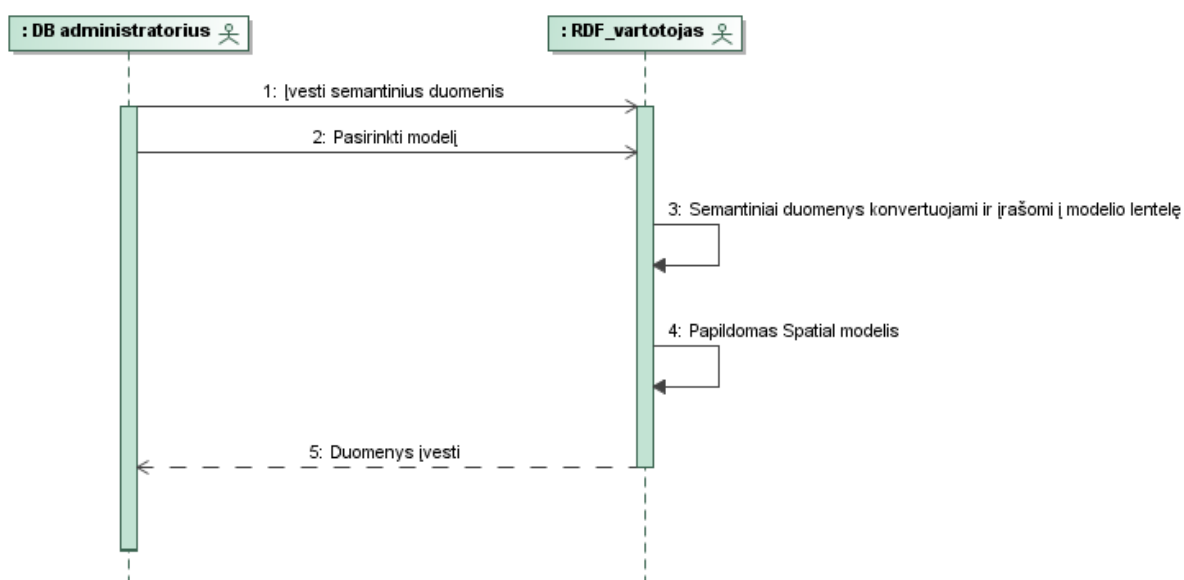
8 lentelė. Panaudojimo atvejo „įkelti semantinius duomenis“ specifikacija

PA „Įkelti semantinius duomenis“		
Tikslas. Įkelti <i>RDF</i> trejetus		
Aprašymas. Įkeliami semantiniai duomenys.		
Prieš sąlyga	Įvesti <i>RDF</i> trejetą arba įkelti failą	
Aktorius	Duomenų bazės administratorius	
Sužadinimo sąlyga	Įvesti <i>RDF</i> duomenis	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	

3.2.3 Panaudojimo atvejis „įkelti semantinius duomenis *INSERT* metodu“

9 lentelėje pateikta panaudojimo atvejo „įkelti semantinius duomenis *INSERT* metodu“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 19 pav.

PA „Įkelti semantinius duomenis <i>INSERT</i> metodu“	
Tikslas. Įkelti <i>RDF</i> trejetus	
Aprašymas. Įkeliami semantiniai duomenys po vieną <i>RDF</i> trejetą	
Prieš sąlyga	Įvesti <i>RDF</i> trejetą
Aktorius	Duomenų bazės administratorius
Sužadinimo sąlyga	Įvesti <i>RDF</i> duomenis
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai



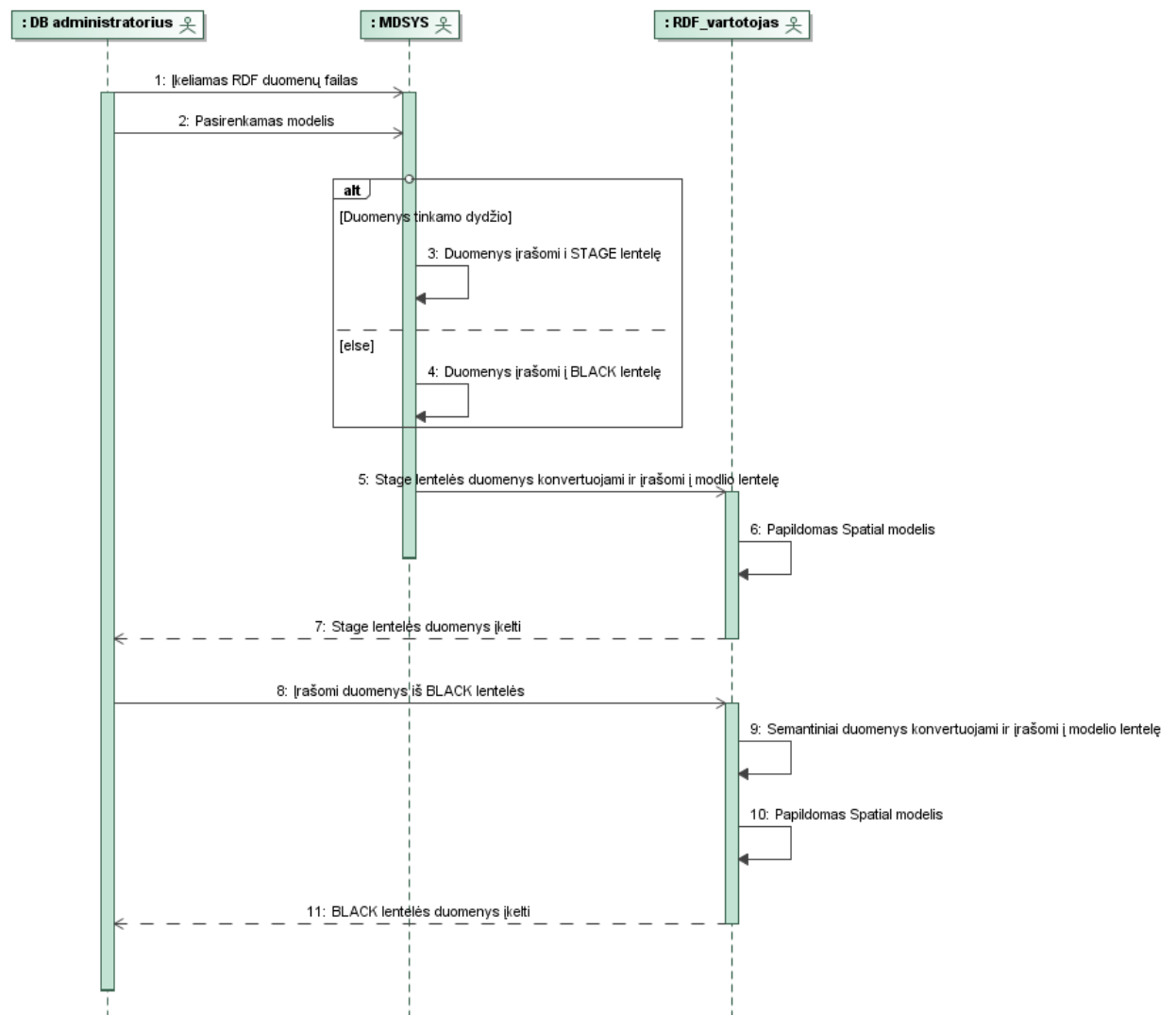
19 pav. Panaudojimo atvejo „įkelti semantinius duomenis *INSERT* metodu“ sekų diagrama

3.2.4 Panaudojimo atvejis „įkelti semantinius duomenis masiniu įkėlimo metodu“

10 lentelėje pateikta panaudojimo atvejo „įkelti semantinius duomenis masiniu įkėlimo metodu“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 20 pav.

10 lentelė. Panaudojimo atvejo „Įkelti semantinius duomenis masiniu įkėlimo metodu“ specifikacija

PA „Įkelti semantinius duomenis masiniu įkėlimo metodu“		
Tikslas. Įkelti <i>RDF</i> trejetus		
Aprašymas. Įkeliami semantiniai duomenys iš failo		
Prieš sąlyga	Įkelti failą su <i>RDF</i> trejetais	
Aktorius	Duomenų bazės administratorius	
Sužadinimo sąlyga	Įkelti failą	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	



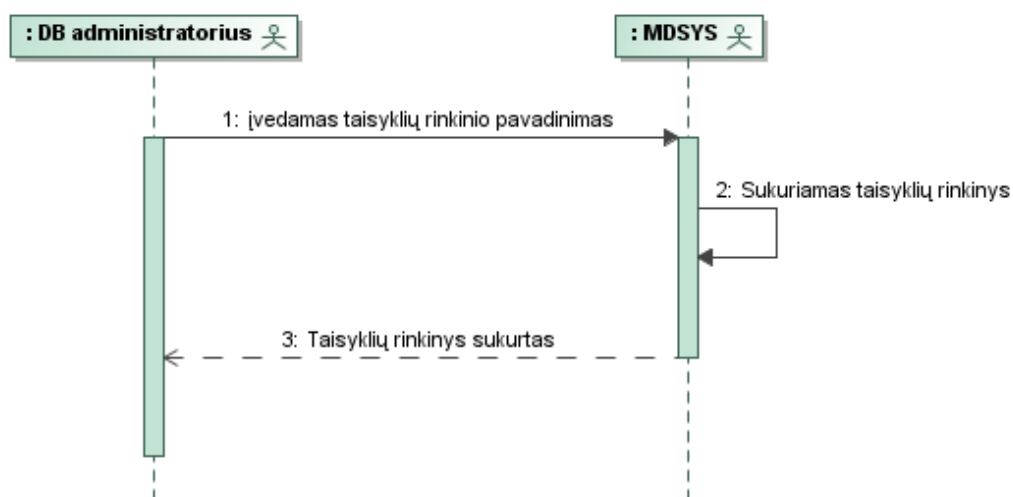
20 pav. Panaudojimo atvejo „įkelti semantinius duomenis masiniu įkėlimo metodu“ sekų diagrama

3.2.5 Panaudojimo atvejis „sukurti taisyklių rinkinį“

11 lentelėje pateikta panaudojimo atvejo „įkelti semantinius duomenis masiniu įkėlimo metodu“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 21 pav.

11 lentelė. Panaudojimo atvejo „sukurti taisyklių rinkinį“ specifikacija

PA „Sukurti taisyklių rinkinį“		
Tikslas. Sukurti taisyklių rinkinį		
Aprašymas.		
Prieš sąlygą	Įvesti taisyklių rinkinį	
Aktorius	Duomenų bazės administratorius	
Sužadinimo sąlyga	Sukurti taisyklių rinkinį	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	

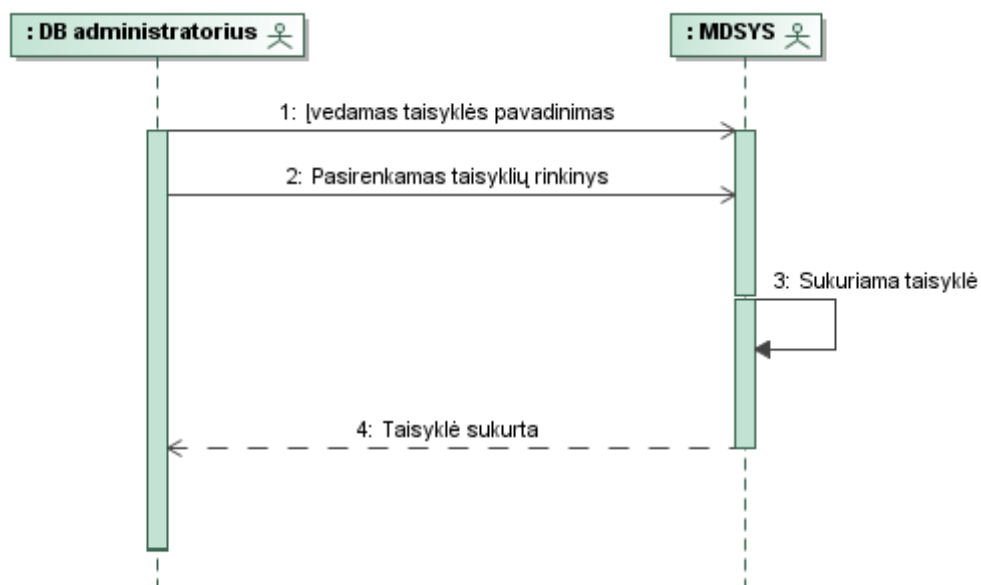


21 pav. Panaudojimo atvejo „Sukurti taisyklių rinkinį“ sekų diagrama

3.2.6 Panaudojimo atvejis „sukurti taisyklę“

12 lentelėje pateikta panaudojimo atvejo „sukurti taisyklę“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 22 pav.

PA „Sukurti taisyklę“		
Tikslas. Sukurti taisyklę		
Aprašymas.		
Prieš sąlygą	Pasirinkti taisyklių rinkinį	
Aktorius	Duomenų bazės administratorius	
Sužadinimo sąlyga	Sukurti taisyklę	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	



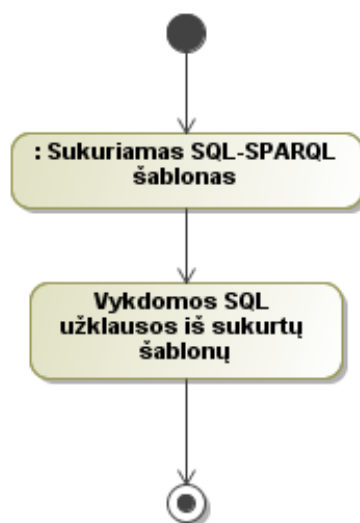
22 pav. Panaudojimo atvejo „sukurti taisyklę“ sekų diagrama

4. Semantinių ir reliacinių duomenų integravimo metodas

4.1 Semantinių ir reliacinių duomenų integravimo metodo proceso modeliai

Duomenys talpinami skirtingose duomenų saugyklose: semantinėje ir reliacinėje. Šiose skirtingose duomenų saugyklose duomenys interpretuojami skirtingai, taip pat skiriasi jų metamodeliai bei duomenų manipuliavimo sintaksė. Skirtingų saugyklų duomenims integruoti sumąstytas sprendimas užrašyti integruotus *SQL* ir *SPARQL* užklausų šablonus. Šablonai saugomi duomenų bazės vaizduose (*views*).

23 pav. atvaizduotas semantinių ir reliacinių duomenų manipuliavimo proceso modelis.



23 pav. Semantinių ir reliacinių duomenų manipuliavimo procesų modelis

4.2 Semantinių ir reliacinių duomenų integravimo metodo panaudojimo atvejai

Semantinių ir reliacinių duomenų manipuliavimo metode numatomi 2 panaudojimo atvejai (24 pav.)



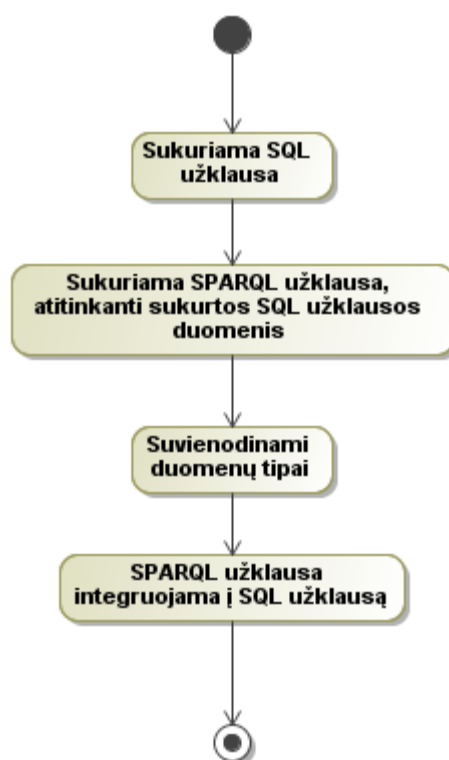
24 pav. Reliacinių ir semantinių duomenų integravimo panaudojimo atvejų modelis

4.2.1 Panaudojimo atvejis „Sukurti šablona“

13 lentelėje pateikta panaudojimo atvejo „sukurti šablona“ detali specifikacija.

PA „Sukurti šabloną“	
Tikslas. Sukurti šabloną reliacinių ir semantinių užklausų vykdymui	
Aprašymas. Šablonas susideda iš integruotų reliacinių ir semantinių užklausų. Šablonai naudojami semantiniams ir reliaciniams duomenims manipuluoti SQL užklausa.	
Prieš sąlyga	
Aktorius	Duomenų bazės administratorius
Sužadinimo sąlyga	
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai

Šablono kūrimo procesas susideda iš 4 etapų, kurie pavaizduoti 25 pav.

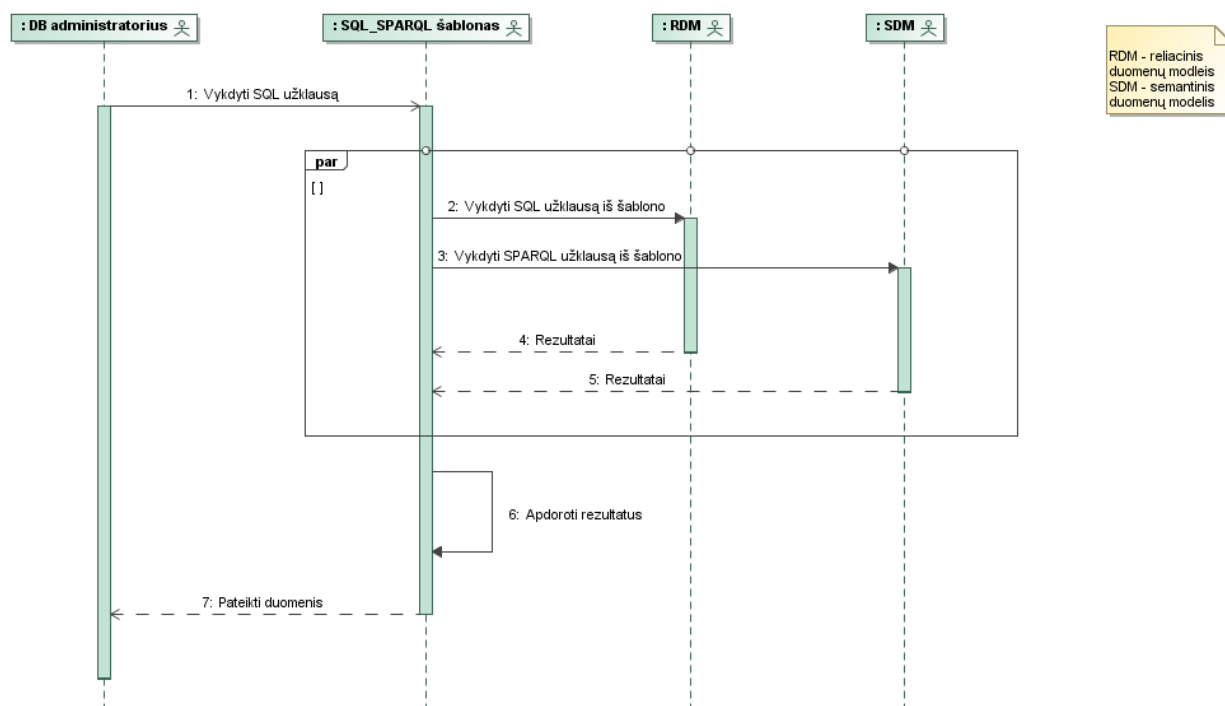


25 pav. Šablono kūrimo procesas

4.2.2 Panaudojimo atvejis „vykdyti duomenų paiešką iš šablono“

14 lentelėje pateikta panaudojimo atvejo „sukurti taisyklę“ detali specifikacija. Šio panaudojimo atvejo elgsena iliustruojama 26 pav.

PA „Vykdyti duomenų paiešką iš šablono“	
Tikslas. SQL užklausa vykdyti paiešką semantinėje ir reliacinėje saugykloje	
Aprašymas.	
Prieš sąlygą	
Aktorius	Duomenų bazės administratorius
Sužadinimo sąlyga	Vykdyti duomenų paiešką
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai



26 pav. Panaudojimo atvejo „vykdyti duomenų paiešką iš šablono“ sekų diagrama

5. Sukurtų metodų taikymo eksperimentinis tyrimas

5.1 Eksperimento tikslas

Eksperimento tikslas – sukurti eksperimentinę aplinką ir joje imituoti metodikose numatytus veiksmus:

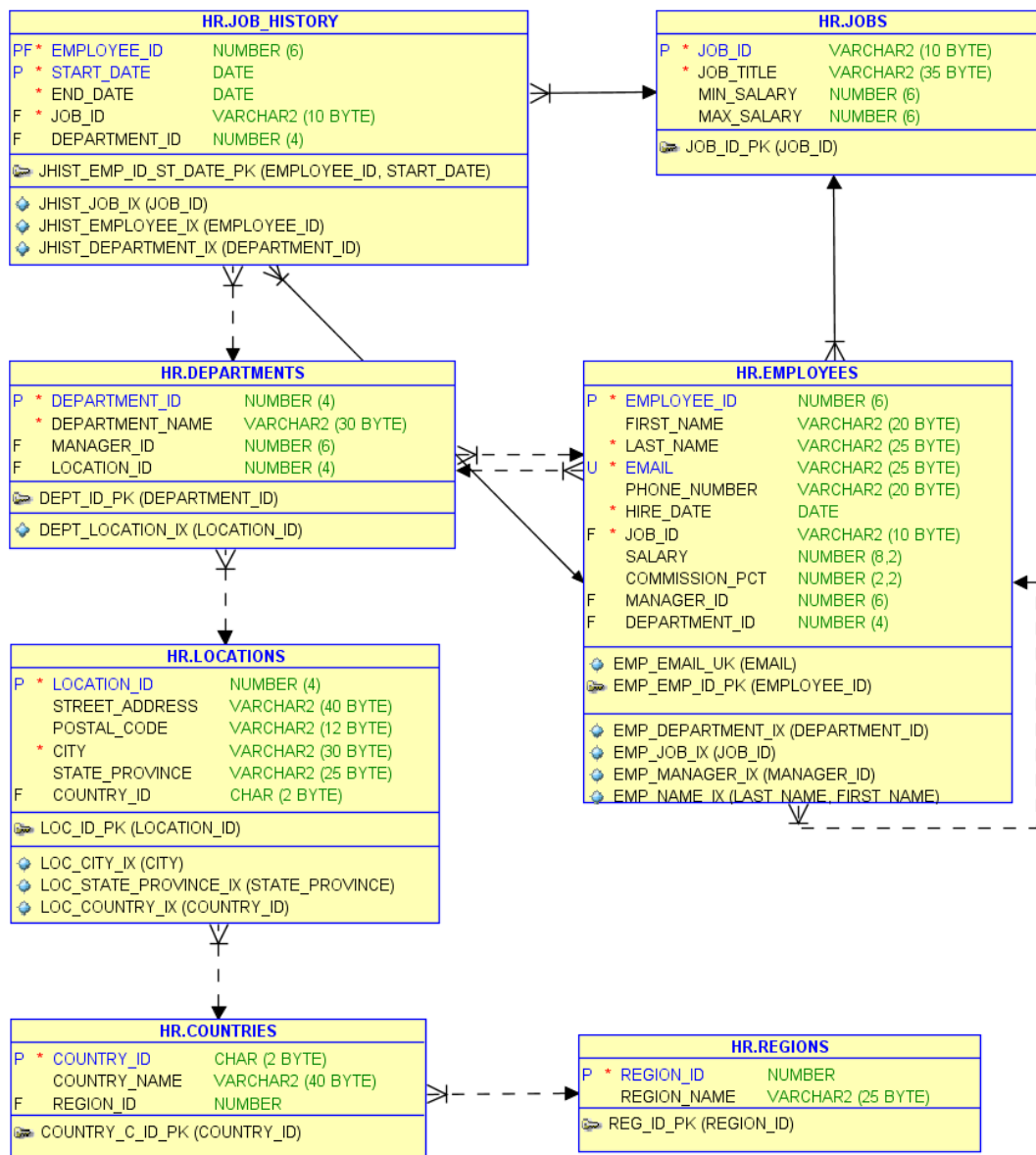
1. sukurti semantinius modelius;
2. sukurti taisyklių rinkinius ir taisykles;
3. įkelti semantinius duomenis;
4. sukurti *SQL-SPARQL* šablonus;
5. įvykdyti duomenų paiešką per semantinius šablonus.

5.2 Apribojimai eksperimentui

Eksperimentui naudojami 6 apribojimai:

1. Apribojimas duomenų bazei;

Eksperimentui naudojama pavyzdinė *Oracle* „*Human resource*“ schema (27 pav.) su duomenimis.



27 pav. Pavyzdinė Oracle DBVS „Human resource“ duomenų schema

2. Apribojimai semantiniam modeliui:

Eksperimentui sukurti 2 semantiniai modeliai:

1. „Darbas“ – modelio duomenys saugomi DARBAS_RDF duomenų lentelėje;
2. „Darbuotojai“ – modelio duomenys saugomi DARBUOTOJAI_RDF duomenų lentelėje.

3. Apribojimai taisyklėms:

Eksperimentui naudojami taisyklių rinkiniai:

1. *RDFS* rinkinys ir standartinės taisyklių rinkinio taisyklės;
2. „Darbuotojai“ rinkinys ir taisyklės:
 - Darbas;
 - Darbovietė;
 - Įdarbintas;

- Uždarbis.

4. Apribojimai indeksams:

Eksperimentui sukurtas indeksas *darbuotojai_indx*, kuris suindeksuoja duomenis pagal „darbuotojai“ taisyklių rinkinį.

5. Apribojimai semantiniams duomenims:

Semantiniam modeliui „Darbas“ *RDF* trejetai sugeneruoti iš reliacinės pavyzdinės *Oracle* duomenų bazės „*Human resource*“ schemos. Duomenys sugeneruoti *D2RQ*⁴ paketo pagalba. Kad duomenys skirtųsi nuo reliacinės duomenų bazės, visi įrašai sukurti su galūne „_sem“. Sugeneruoti *RDF* trejetai pateikti 1 priede. Duomenys įkelti masinio metodo būdu.

Semantiniam modeliui „Darbuotojai“ duomenys įkelti *INSERT* metodu, apibrėžiant *RDFS* ir „Darbuotojai“ taisyklių rinkiniais. Įrašai taip pat paimti iš reliacinės pavyzdinės *Oracle* duomenų bazės „*Human resource*“ schemos pridėdant galūnę „_rdf“. *RDF* trejetai pateikti 2 priede.

6. Apribojimai *SQL-SPARQL* šablonui:

Kiekvienam modelio eksperimentui sukurta po vieną *SQL-SPARQL* šabloną:

1. „Darbas“ modeliui sukuriamas šablonas, atvaizduojantis darbo pavadinimą ir to darbo minimalaus ir maksimalaus užmokesčio ribas.
2. „Darbuotojai“ modeliui sukuriamas šablonas, atvaizduojantis darbuotojo vardą, darbo pavadinimą, darbovietę, įdarbinimo datą ir uždarbį.

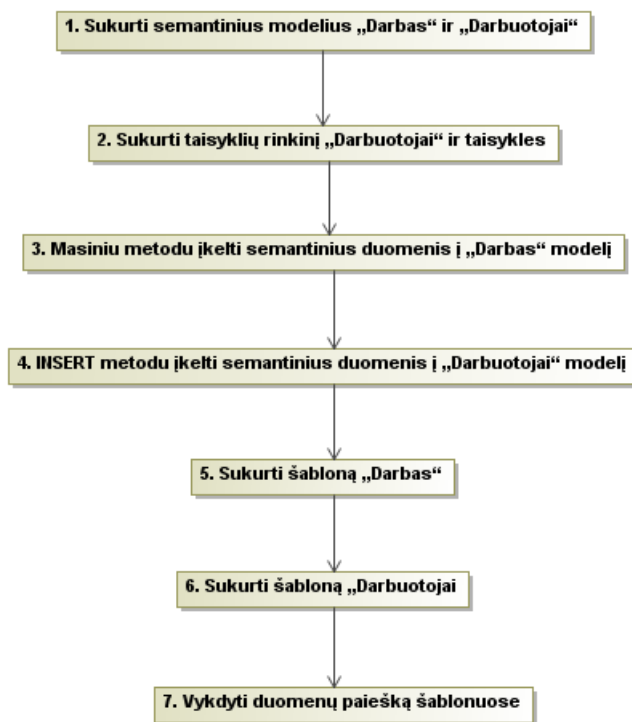
5.3 Eksperimento planas

Semantinių duomenų saugojimo bei semantinių ir reliacinių duomenų integravimo metodų eksperimentui sukurti 2 internetiniai įrankiai:

1. semantinių duomenų administravimo įrankis;
2. reliacinių ir semantinių duomenų integravimo internetinė sistema.

Pirmame įrankyje 1-4 etapai numatyti eksperimento plane (28 pav.).

⁴ <http://d2rq.org/>



28 pav. Eksperimento planas

15 – 21 lentelėse pateikta eksperimento etapų specifikacija.

15 lentelė. Sukurti semantinius modelius „Darbas“ ir „Darbuotojai“

Eksperimento etapo pavadinimas		1. Sukurti semantinius modelius „Darbas“ ir „Darbuotojai“
Panaudos atvejis / reikalavimas(-ai):		Kurti semantinius modelius
Aprašymas	Semantinių duomenų administravimo įrankiu sukurti semantinius modelius „Darbas“ ir „Darbuotojai“	
Sąlygos "Prieš"	Prie sistemos prisijungęs duomenų administratorius	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Įvesti modelio pavadinimą „Darbas“ ir spausti „Sukurti“ mygtuką.	Automatiškai atliekami visi veiksmai modeliui sukurti. Apie sėkmingą modelio sukūrimą sistema pateikia pranešimą.
2.	Įvesti modelio pavadinimą „Darbuotojai“ ir spausti „Sukurti“ mygtuką.	

16 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles

Eksperimento etapo pavadinimas		2. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles
Panaudos atvejis / reikalavimas(-ai):		Sukurti taisyklių rinkinį; Sukurti taisyklę.
Aprašymas	Semantinių duomenų administravimo įrankiu sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles	
Sąlygos "Prieš"	Prie sistemos prisijungęs duomenų administratorius	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Įvesti taisyklių rinkinio pavadinimą „Darbuotojai“ ir spausti „Sukurti“ mygtuką.	Automatiškai atliekami visi veiksmai taisyklių rinkiniui sukurti. Apie sėkmingą taisyklių rinkinio sukūrimą sistema pateikia pranešimą.
2.	Įvesti taisyklės pavadinimą „111111“ ir spausti „Sukurti“ mygtuką.	
3.	Įvesti taisyklės pavadinimą „222222“ ir spausti „Sukurti“ mygtuką.	

17 lentelė. Masiniu metodu įkelti semantinius duomenis į „Darbas“ modelį

Eksperimento etapo pavadinimas		3. Masiniu metodu įkelti semantinius duomenis į „Darbas“ modelį.
Panaudos atvejis / reikalavimas(-ai):		Įkelti masiniu metodu
Aprašymas	Semantinių duomenų administravimo įrankiu įkelti semantinius duomenis į „Darbas“ modelį.	
Sąlygos "Prieš"	Prie sistemos prisijungęs duomenų administratorius. Sukurtas semantinis modelis „Darbas“.	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Internetinėje duomenų įkėlimo formoje pasirenkamas tekstinis dokumentas su <i>RDF</i> duomenimis.	Automatiškai atliekami visi veiksmai duomenims apdoroti ir įkelti į semantinę saugyklą. Apie sėkmingą duomenų įkėlimą sistema pateikia pranešimą.

18 lentelė. *INSERT* metodu įkelti semantinius duomenis į „Darbuotojai“ modelį

Eksperimento etapo pavadinimas	4. <i>INSERT</i> metodu įkelti semantinius duomenis į „Darbuotojai“ modelį.	
Panaudos atvejis / reikalavimas(-ai):	Įkelti <i>INSERT</i> metodu	
Aprašymas	Semantinių duomenų administravimo įrankiu įkelti semantinius duomenis į „Darbuotojai“ modelį.	
Sąlygos "Prieš"	Prie sistemos prisijungęs duomenų administratorius. Sukurtas semantinis modelis „Darbuotojai“.	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Įvedamas <i>URI</i> adresas	
2.	Įvedamas subjektas	
3.	Įvedamas predikatas	
4.	Įvedamas objektas	
5.	Įrašomi įvesti duomenys	Automatiškai atliekami visi veiksmai duomenims apdoroti ir įkelti į semantinę saugyklą. Apie sėkmingą duomenų įkėlimą sistema pateikia pranešimą.

19 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles

Eksperimento etapo pavadinimas	5. Sukurti „Darbas“ šabloną.	
Panaudos atvejis / reikalavimas(-ai):	Sukurti šabloną	
Aprašymas	Užrašyti <i>SQL-SPARQL</i> šabloną „Darbas“ <i>DB</i> lygyje.	
Sąlygos "Prieš"	Sukurtas semantinis modelis „Darbas“ ir įkelti duomenys	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Užrašoma <i>SQL</i> užklausa, kuri ieško duomenų reliacinėje saugykloje	Reliacinėje duomenų bazėje atrenkami duomenys apie darbus.
2.	Užrašoma <i>SPARQL</i> užklausa, kuri ieško duomenų semantinėje saugykloje	Semantinėje duomenų bazėje atrenkami duomenys apie darbus.
3.	Integruojama <i>SPARQL</i> užklausa į <i>SQL</i> ir suvienodinami duomenų tipai.	
4.	Šablonas sukuriamas kaip duomenų bazės objektas – vaizdas (<i>view</i>)	Duomenų vaizdas sukurtas be klaidų

20 lentelė. Sukurti taisyklių rinkinį „Darbuotojai“ ir taisykles

Eksperimento etapo pavadinimas	Sukurti „Darbuotojai“ šabloną.	
Panaudos atvejis / reikalavimas(-ai):	Sukurti šabloną	
Aprašymas	Užrašyti <i>SQL-SPARQL</i> šabloną „Darbuotojai“ <i>DB</i> lygyje.	
Sąlygos "Prieš"	Sukurtas semantinis modelis „Darbuotojai“ ir įkelti duomenys	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Užrašoma <i>SQL</i> užklausa, kuri ieško duomenų reliacinėje saugykloje	Reliacinėje duomenų bazėje atrenkami duomenys apie darbus.
2.	Užrašoma <i>SPARQL</i> užklausa, kuri ieško duomenų semantinėje saugykloje	Semantinėje duomenų bazėje atrenkami duomenys apie darbus.
3.	Integruojama <i>SPARQL</i> užklausa, į <i>SQL</i> ir suvienodinami duomenų tipai.	
4.	Šablonas sukuriamas kaip duomenų bazės objektas – vaizdas (<i>view</i>)	Duomenų vaizdas sukurtas be klaidų

21 lentelė. Vykdyti duomenų paiešką šablonuose

Eksperimento etapo pavadinimas	7. Vykdyti duomenų paiešką šablonuose.	
Panaudos atvejis / reikalavimas(-ai):	Vykdyti paiešką iš šablonų	
Aprašymas	Duomenų iš šablonų atvaizdavimas ir manipuliavimas duomenų atvaizdavimo lygyje	
Sąlygos "Prieš"	Sukurti <i>SQL-SPARQL</i> šablonai	
Sąlygos "Po"		
Vartotojas:	Duomenų administratorius	
Žingsnio Nr.	Žingsnis	Laukiamas rezultatas
1.	Duomenų atvaizdavimo aplikacijos pritaikymas prie šablonų	Atvaizduojami visi duomenys, pasiekiami per šablonus
2.	Vykdoma duomenų paieška	Atvaizduojami ieškomi duomenys, pasiekiami per šablonus

5.4 Eksperimento rezultatai

Vadovaujantis darbe sukurtu metodu, visų pirma sukuriami modeliai „Darbas“ ir „Darbuotojai“ (29 pav.).

Sukūrė	Modelio pavadinimas	Modelio lentelės pavadinimas	Modelio tipas	Panaikinti
SEM	DARBAS	DARBAS_RDF	TRIPLE	Ištrinti
SEM	DARBUOTOJAI	DARBUOTOJAI_RDF	TRIPLE	Ištrinti

29 pav. Modelių kūrimo programinio įrankio langas

Kitame žingsnyje taisyklių rinkiniais apibrėžiami modeliai. Tai galima atlikti sukūrus taisyklių indeksą, panaudojant standartinius taisyklių rinkinius ar anksčiau sukurtus, arba sukurti naujus taisyklių rinkinius.

Pagal eksperimento planą kuriamas naujas taisyklių rinkinys „Darbuotojai“ (30 pav.).

Schemas vartotojas	Taisyklių rinkinio pavadinimas	Taisyklių rinkinio vaizdas	Statusas
SYS	RDF	RDFR_RDF	VALID

30 pav. Taisyklių kūrimo programinis įrankio langas

Vykdamas eksperimentinį tyrimą, rasta *Oracle DBVS* teisių interpretavimo klaida. Kuriant naują taisyklių rinkinį, pasinaudojant darbo metu sukurtu įrankiu, *Oracle* teisių interpretatorius klaidingai interpretuoja šį įvykį ir nesuteikia reikiamų teisių kūrimo procesui užbaigti. Eksperimento metu nustatyta, kad taisyklių rinkinį galima sukurti prisijungus prie vidinio *Oracle* užklausių rašymo įrankio *sqlplus*. Nuspręsta tyrimo metu kurtam įrankiui realizuoti funkcionalumą, kuris pateiktų vartotojui instrukciją, kaip sukurti taisyklių rinkinį *sqlplus* įrankiu.

Kitas etapas – modelių užpildymas duomenimis. Modelis „Darbas“ užpildomas šiame darbe patobulintu masinio užkrovimo metodu, semantinius duomenis įrašant iš *RDF* trejetų failo. Modelis „Darbuotojai“ užpildomas paprastu *INSERT* metodu, duomenis įrašant į tam pritaikytą formą (31 pav.).

31 pav. Semantinių duomenų įrašymo programinis įrankio langas

Kai modeliai užpildomi duomenimis, sukuriama *SQL-SPARQL* šablonai. Šablonų išeities kodai pateikti 13 – 14 pavyzdžiuose.

```

create or replace view darbas as
select JOB_TITLE, MIN_SALARY, MAX_SALARY
from HR.JOBS
union
select JOB_TITLE, to_number(MIN_SALARY), to_number(MAX_SALARY)
from table(sem_match(
'(?JOB_ID <http://localhost:2020/vocab/resource/JOBS_JOB_TITLE> ?JOB_TITLE)
(?JOB_ID <http://localhost:2020/vocab/resource/JOBS_MIN_SALARY> ?MIN_SALARY)
(?JOB_ID <http://localhost:2020/vocab/resource/JOBS_MIN_SALARY> ?MAX_SALARY)',
sem_models ('DARBAS'),
null,
SEM_ALIASES(SEM_ALIAS('','http://test.lt/HRS.EMPLOYEES/')),
NULL));

```

13 pvz. *SQL-SPARQL* šablono „Darbas“ programinis kodas

```

CREATE OR REPLACE VIEW DARBUOTOJAI AS
SELECT  first_name,  job_title,  department_name,  to_char(hire_date,  'dd-mon-yy')
hire_date,  salary
from hr.employees, hr.departments, hr.jobs
where hr.employees.department_id = hr.departments.department_id AND
hr.employees.job_id = hr.jobs.job_id
union
SELECT first_name, job_title, department_name, hire_date, to_number(salary) salary
FROM TABLE(SEM_MATCH(
  '(?first_name :darbas ?job_title)
  (?first_name :darboviete ?department_name)
  (?first_name :idarbintas ?hire_date)
  (?first_name :uzdarbis ?salary)',
  SEM_Models('DARBUOTOJAI'),
  null,
  SEM_ALIASES(SEM_ALIAS('', 'HTTP://WWW.TEST.LT/HRS.EMPLOYEES/')),
  null));

```

14 pvz. *SQL-SPARQL* šablono „Darbuotojai“ programnis kodas

Duomenų manipuliavimas ir atvaizdavimas vartotojams vyksta *SQL* užklausoms kreipiantis į *SQL-SPARQL* šablonus. Eksperimentinio tyrimo metu sukurtų šablonų duomenų atvaizdavimas pateiktas 32 – 33 paveikslėliuose.

Vardas	Darbo pavadinimas	Padalinys	Įdarbinimo data	Alga
Adam	Stock Manager	Shipping	10-apr-05	8200
Adam_RDF	Stock Manager_RDF	Shipping_RDF	10-APR-05	8200
Alana	Shipping Clerk	Shipping	24-apr-06	3100
Alberto	Sales Manager	Sales	10-mar-05	12000
Alexander	Programmer	IT	03-jan-06	9000
Alexander	Purchasing Clerk	Purchasing	18-may-03	3100
Alexander_RDF	Purchasing Clerk_RDF	Purchasing_RDF	18-MAY-03	3100
Alexis	Shipping Clerk	Shipping	20-feb-05	4100
Allan	Sales Representative	Sales	01-aug-04	9000
Alyssa	Sales Representative	Sales	19-mar-05	8800

32 pav. Šablono „Darbuotojai“ duomenys

Informacija apie darbus		
Darbo pavadinimas	Minimali alga	Maksimali alga
Accountant	4200	9000
Accountant_sem	4200	4200
Accounting Manager	8200	16000
Accounting Manager_sem	8200	8200
Administration Assistant	3000	6000
Administration Assistant_sem	3000	3000
Administration Vice President	15000	30000
Administration Vice President_sem	15000	15000
Finance Manager	8200	16000
Finance Manager_sem	8200	8200

33 pav. Šablono „Darbai“ duomenys

5.5 Eksperimento išvados

Eksperimentinio tyrimo metu patikrinti darbe sukurti metodai. Eksperimento rezultatai rodo, kad metodai veikia tinkamai ir gali būti pritaikomi semantiniams duomenims administruoti ir integruoti su reliaciniais duomenimis. Darbe išskelti uždaviniai sėkmingai įgyvendinti.

6. Išvados

1. Atlikus analizę nustatyta, kad *Oracle DBVS* ontologijas aprašyti galima ne tik populiariomis notacijomis, tokiomis kaip *RDF*, *RDFS*, *OWL* ir kitos, bet ir sukurti savo taisykles ir taisyklių rinkinius.
2. Išanalizavus semantinę *Oracle* saugyklą nustatyta, kad *SPARQL* užklausų kalba užrašytas užklausas galima integruoti į *SQL* užklausas ir sukurti vientisą integruotą užklausa reliaciniams ir semantiniams duomenims manipuluoti tuo pačiu metu.
3. Vertinant analizę nustatyta, kad *Oracle DBVS* užtikrina funkcionalumą semantiniams duomenims saugoti, jiems analizuoti ir jais manipuluoti. Taip pat yra galimybė duomenis *Oracle DBVS* integruoti su reliaciniais duomenimis. Semantinis funkcionalumas realizuotas *MDSYS* schemeje.
4. Sukurtas semantinių duomenų saugojimo metodas funkcionuojantis *Oracle DBVS*. Šis metodas aprašo svarbiausius *RDF* duomenų saugojimo aspektus:
 - 4.1 semantinių modelių kūrimą ir panaudojimą;
 - 4.2 *RDF* trejetų saugojimą *Oracle DBVS* patogiais metodais;
 - 4.3 taisyklių ir taisyklių rinkinių kūrimą ir jų panaudojimą modeliams apibrėžti;
 - 4.4 paieškos indeksų kūrimą ir jų panaudojimą, sujungiant modelius ir taisykles.
5. Pagal semantinių duomenų saugojimo metodą sukurtas semantinės saugyklos administravimo programinis prototipas. Šis įrankis suteikia galimybę dalykinės srities specialistams – ne informatikams – kurti ir valdyti modelius, taisyklių rinkinius, įkelti *RDF* trejetus.
6. Sukurtas semantinių ir reliacinių duomenų integravimo metodas grindžiamas *SQL-SPARQL* šablonais, užtikrinantis vientisą semantinių ir reliacinių duomenų integravimą. Naudojant *SQL-SPARQL* šablonus semantiniams ir reliaciniais duomenimis tuo pačiu metu manipuluojama užrašant *SQL* užklausas.
7. Šiame tyrime sukurti ir eksperimentiškai išbandyti semantinių ir reliacinių duomenų integravimo bei semantinių duomenų saugojimo metodai gali būti naudojami:

- 7.1 semantinėms saugykloms organizacijose kurti, diegti ir administroti savo turimose *Oracle DBVS*;
 - 7.2 reliaciniams ir semantiniams duomenims integruoti;
 - 7.3 semantinėms saugykloms *Oracle DBVS* eksploatuoti;
 - 7.4 semantinių ir reliacinių duomenų integravimo įrankiams kurti;
 - 7.5 semantinių duomenų saugojimo *Oracle DBVS* įrankiams kurti.
8. Surasta ir *Oracle* techniniame pagalbos centre užregistruota privilegijų semantiniame pakete interpretavimo klaida, kuri turėjo įtakos semantinės saugyklos administravimo programinio prototipo taisyklių kūrimo funkcionalumui užtikrinti. T. y. bandant programiniu prototipu sukurti taisyklių rinkinį *Oracle DBVS* privilegijų interpretavimo mechanizmas suveikia neteisingai ir nesuteikia reikiamų teisių atlikti numatytą veiksmą. Tačiau sukurtas sprendimas, kuris, kuriant taisyklių rinkinį, pateikia instrukciją, kaip pabaigti veiksmą be programinio prototipo.

Research of the semantic and relational queries processing in Oracle DBMS

Deividas Švėgžda

Master degree graduation carried out in Kaunas University of Technology, Department of
Information Systems

SUMMARY

In the modern world it is necessary to save a larger amount of data in order to meet the needs. More difficult systems are created in order to save and process the data. However, a big challenge for ordinary systems is to control and appropriately offer for consumers such huge amounts of data.

In order to process information quicker, mechanisms are created to systemize the data, and using ontologies to describe them according to the semantic meaning. But the amount of ontologies are also growing, and at the same time data interpretation processes are getting slow. Therefore, the question arises how and where to save ontologies and how to integrate the data with the information which is in relational data bases in order the processes to become quicker.

In the research semantic storage opportunities of *Oracle* data bases management systems are analysed. The methods of semantic data storage, semantic and relational data integration are patterned. The method of the semantic data storage solves semantic data storage tasks such as the creation of semantic patterns and rules, *RDF* storage in *Oracle* data bases management systems. The method of the semantic and relational data integration solves different storage data integration tasks. Thus, integrated SQL and SPARQL queries patterns are created which let us manipulate SQL queries language and semantic and relational data at the same time.

There is also created the prototype of semantic data administrative tool by using a semantic data storage method. The prototype gives for a consumer a convenient connection to administrate semantic data.

Keywords: ontologies, semantic and relational data integration, *Oracle SPATIAL RDF* pattern, *RDF*, *OWL*.

Literatūra

1. Codd E.F., A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 1970. 13(6): p. 377.
2. Hammer, M., and D. McLeod, The semantic data model: a modelling mechanism for data base applications, in Proceedings of the 1978 ACM SIGMOD international conference on management of data 1978, ACM: Austin, Texas. p. 26-36.
3. Smith B., Blackwell Guide to the Philosophy of Computing and Information, Oxford: Blackwell, 2003, p. 155-166.
4. Guarino N., Formal ontology and information systems. Frontiers in artificial intelligence and applications, 1998. 46: p. 3-15.
5. Noy N. F., McGuinness D.L. Ontology development 101: a guide to creating your first ontology, 2001 [žiūrėta 2010 m. spalio 24d.]. Prieiga per internetą: <http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>.
6. *OWL ontologies library* [žiūrėta 2010 m. lapkričio 7d.]. Prieiga per internetą: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library.
7. *RDF Semantics* [žiūrėta 2011 m. sausio 21d.]. Prieiga per internetą: <http://www.w3.org/TR/2004/REC-rdf-mt-20040210>.
8. *RDF Vocabulary Description Language 1.0: RDF Schema*. [žiūrėta 2011 m. sausio 24d.]. Prieiga per internetą: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
9. *OWL Web Ontology Language* [žiūrėta 2011 m. kovo 5d.]. Prieiga per internetą: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
10. *SPARQL Query Language for RDF* [žiūrėta 2011 m. kovo 19d.]. Prieiga per internetą: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
11. Beaugard B., Das S., Perry M., Rieb K., Sundara S., et al., Oracle Database Semantic Technologies. Konferencijos "Semantic Technologies Conference" pranešimų medžiaga. San Franciskas, JAV, 2011.
12. Sundara, S., ir kiti. Visualizing large-scale RDF data using Subsets, Summaries, and Sampling in Oracle. in Data Engineering (ICDE), 2010 IEEE 26th tarptautinė konferencija on. 2010, p. 1048-1059.
13. Zhe W., ir kiti. Implementing an Inference Engine for RDFS/OWL Constructs and User-Defined Rules in Oracle. in Data Engineering, 2008. ICDE 2008. IEEE 24th tarptautinė konferencija 2008, p. 1239-1248
14. Alexander N., Ravada S., RDF Object Type and Reification in the Database. in Data Engineering, 2006. ICDE '06. Proceedings of the 22nd tarptautinė konferencija. 2006.

15. Uzdaviciūtė V., Butleris R., Oracle DBVS galimybės ontologijomis grindžiamo duomenų integravimo procese, Informacinės technologijos = Proceedings of Master and PhD students conference on informatic : 15-osios tarpuniversitetinės magistrantų ir doktorantų konferencijos "Informacinė visuomenė ir universitetinės studijos" (IVUS 2010) medžiaga, Kaunas: Vytauto Didžiojo universitetas. ISSN 2029-249X. 2011, nr. 15, p. 169-174.
16. Perry R., Gillen A., Demonstrating Business Value: Selling to Your C-Level Executives. White paper, 2007.
17. Trivedi G., Oracle on Windows vs Unix and Linux - Differences [žiūrėta 2011 m. rugsėjo 27d.]. Prieiga per internetą: <http://www.dbamatrix.com/ora-unix-linux-win.html>.
18. W3C, OS Platform Statistics. [žiūrėta 2011 m. rugsėjo 28d.]. Prieiga per internetą: http://www.w3schools.com/browsers/browsers_os.asp.

Priedas 1

Dėl didelės apimties pateikiamas kompaktiniame diske.

Priedas 2

Subjektas	Predikatas (taisyklė)	Objektas
Jennifer_RDF	darbas	Administration Assistant_RDF
Pat_RDF	darbas	Marketing Representative_RDF
Michael_RDF	darbas	Marketing Manager_RDF
Den_RDF	darbas	Purchasing Manager_RDF
Guy_RDF	darbas	Purchasing Clerk_RDF
Sigal_RDF	darbas	Purchasing Clerk_RDF
Shelli_RDF	darbas	Purchasing Clerk_RDF
Alexander_RDF	darbas	Purchasing Clerk_RDF
Karen_RDF	darbas	Purchasing Clerk_RDF
Susan_RDF	darbas	Human Resources Representative_RDF
Matthew_RDF	darbas	Stock Manager_RDF
Adam_RDF	darbas	Stock Manager_RDF
Payam_RDF	darbas	Stock Manager_RDF
Shanta_RDF	darbas	Stock Manager_RDF
Kevin_RDF	darbas	Stock Manager_RDF

Subjektas	Predikatas (taisyklė)	Objektas
Jennifer_RDF	darboviete	Administration_RDF
Pat_RDF	darboviete	Marketing_RDF
Michael_RDF	darboviete	Marketing_RDF
Den_RDF	darboviete	Purchasing_RDF
Guy_RDF	darboviete	Purchasing_RDF
Sigal_RDF	darboviete	Purchasing_RDF
Shelli_RDF	darboviete	Purchasing_RDF
Alexander_RDF	darboviete	Purchasing_RDF
Karen_RDF	darboviete	Purchasing_RDF
Susan_RDF	darboviete	Human Resources_RDF
Matthew_RDF	darboviete	Shipping_RDF
Adam_RDF	darboviete	Shipping_RDF
Payam_RDF	darboviete	Shipping_RDF
Shanta_RDF	darboviete	Shipping_RDF
Kevin_RDF	darboviete	Shipping_RDF

Subjektas	Predikatas (taisyklė)	Objektas
Jennifer_RDF	idarbintas	17-SEP-03
Pat_RDF	idarbintas	17-AUG-05
Michael_RDF	idarbintas	17-FEB-04
Den_RDF	idarbintas	07-DEC-02
Guy_RDF	idarbintas	15-NOV-06
Sigal_RDF	idarbintas	24-JUL-05
Shelli_RDF	idarbintas	24-DEC-05
Alexander_RDF	idarbintas	18-MAY-03
Karen_RDF	idarbintas	10-AUG-07
Susan_RDF	idarbintas	07-JUN-02
Matthew_RDF	idarbintas	18-JUL-04
Adam_RDF	idarbintas	10-APR-05
Payam_RDF	idarbintas	01-MAY-03
Shanta_RDF	idarbintas	10-OCT-05
Kevin_RDF	idarbintas	16-NOV-07

Subjektas	Predikatas (taisyklė)	Objektas
Jennifer_RDF	uzdarbis	4400
Pat_RDF	uzdarbis	6000
Michael_RDF	uzdarbis	13000
Den_RDF	uzdarbis	11000
Guy_RDF	uzdarbis	2600
Sigal_RDF	uzdarbis	2800
Shelli_RDF	uzdarbis	2900
Alexander_RDF	uzdarbis	3100
Karen_RDF	uzdarbis	2500
Susan_RDF	uzdarbis	6500
Matthew_RDF	uzdarbis	8000
Adam_RDF	uzdarbis	8200
Payam_RDF	uzdarbis	7900
Shanta_RDF	uzdarbis	6500
Kevin_RDF	uzdarbis	5800

Priedas 3

Kompaktiniame diske pateikiami programinio įrankio išeities kodai.