

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Giedrė Šimkevičiūtė

**Daugiaatribučių duomenų struktūrų sukūrimas ir jų
alternatyvų įvertinimas**

Magistro darbas

Darbo vadovas
doc. V. Jusas

Kaunas, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėjas
(parašas) doc. dr. E. Bareiša
2004 05

**Daugiaatribučių duomenų struktūrų sukūrimas ir jų
alternatyvų įvertinimas**

Magistro darbas

Kalbos konsultantė
Lietuvių kalbos katedros lektorė
(parašas) dr. J. Mikelionienė
2004 05

Vadovas
(parašas) doc. dr. V. Jusas
2004 05

Recenzentas
(parašas) doc. dr. A. Lenkevičius
2004 05

Atliko
IFM-8/2 gr. stud.
(parašas) G. Šimkevičiūtė
2004 05

Kaunas, 2004

Turinys

Santrauka.....	1
Abstract	1
Raktiniai žodžiai	2
Key words	2
Terminų ir santrumpų žodynas	3
1. Įvadas.....	4
2. Duomenų saugojimo ir apdorojimo principų analizė bei esminių projektavimo sąvokų apžvalga.....	5
2.1. Duomenų saugojimas.....	5
2.1.1. Loginė duomenų bazės struktūra	6
2.1.2. Įrašų saugojimas ir kreipiniai į juos	7
2.2. Duomenų tvarkymas bei užklauskos.....	9
2.2.1. Užklauskų tipai	9
2.2.2. Užklauskos vykdymo eiga	9
2.3. Veikimo greitinimo galimybės.....	11
2.3.1. Indeksavimas.....	11
2.3.2. Klasterių paskirtis ir naudojimas	12
2.3.3. Eilučių identifikatorių taikymas.....	12
2.3.4. Duomenų bloko pildymo valdymas	13
2.4. Duomenų struktūrų projektavimas: normalizavimo ir denormalizavimo procesai.....	14
2.4.1. Normalizavimo samprata	14
2.4.2. Normalizavimo pakopos	14
2.4.3. Denormalizavimo poreikis.....	15
2.4.4. Denormalizavimo taikymo atvejai	15
3. Sukurtos sistemos aprašymas	17
3.1. Sistemos paskirtis, vartotojai ir apribojimai	17
3.2. Veiklos sfera	17
3.2.1. Veiklos kontekstas.....	17
3.2.2. Veiklos padalinimas.....	18
3.3. Panaudojimo atvejai.....	20
3.4. Reikalavimai.....	28
3.4.1. Funkciniai reikalavimai	28
3.4.2. Nefunkciniai reikalavimai	31
3.5. Sistemos architektūra.....	32
3.5.1. Klasių paketų diagrama	33
3.5.2. Duomenų tvarkymo klasės	34
3.5.3. Dalykinės srities klasės	35

3.5.4.	Duomenų modelis.....	36
3.5.5.	Paskirstymo modelis.....	36
3.5.6.	Komponentų modelis.....	37
4.	Galimų duomenų struktūrų, užklausų ir jų greičio tyrimas	38
4.1.	<i>Dažniausiai pasitaikančios užklausos.....</i>	<i>38</i>
4.2.	<i>Galimi duomenų struktūrų variantai.....</i>	<i>38</i>
4.3.	<i>Užklausų greičio matavimo tyrimas</i>	<i>40</i>
4.4.	<i>Užklausų panaudojamumo įvertinimas</i>	<i>45</i>
5.	Išvados.....	47
6.	Literatūra	48

Ilustracijos

Paveikslas 1.	Duomenų bazės loginė struktūra. [8]	6
Paveikslas 2.	Duomenų bloko sandara. [8].....	7
Paveikslas 3.	Lentelės eilutė ir duomenų blokas.[9]	8
Paveikslas 4.	Užklauso vykdymas [10].....	10
Paveikslas 5.	Veiklos konteksto diagrama.....	18
Paveikslas 6.	Panaudojimo atvejų diagrama	21
Paveikslas 7.	Klasių paketų diagrama.	33
Paveikslas 8.	Vartotojo sąsajos klasių diagrama. Duomenų apdorojimo klasės.....	34
Paveikslas 9.	Vartotojo sąsajos klasių diagrama. Duomenų apdorojimo klasių detalizavimas. 35	
Paveikslas 10.	Dalykinės srities klasių diagrama.....	35
Paveikslas 11.	Duomenų modelis.....	36
Paveikslas 12.	Paskirstymo modelis.....	37
Paveikslas 13.	Komponentų modelis.....	37
Paveikslas 14.	Pradinė duomenų struktūra.	39
Paveikslas 15.	Normalizuota duomenų struktūra.....	40
Paveikslas 16.	Denormalizuota duomenų struktūra.	40
Paveikslas 18.	Vieno tyrimo etapo rezultatų įrašymo greičio priklausomybė nuo duomenų struktūros. 43	
Paveikslas 19.	Tyrimo rezultatų atnaujinimo greičio priklausomybė nuo duomenų struktūros. 43	
Paveikslas 20.	Vieno tyrimo etapo rezultatų atnaujinimo greičio priklausomybė nuo duomenų struktūros.	43
Paveikslas 21.	Tyrimo rezultatų šalinimo greičio priklausomybė nuo duomenų struktūros.....	44
Paveikslas 22.	Tyrimo rezultatų išrinkimo greičio priklausomybė nuo duomenų struktūros. ..	44
Paveikslas 23.	Vieno tyrimo etapo rezultatų išrinkimo greičio priklausomybė nuo duomenų struktūros. 44	

Lentelės

Lentelė 1. Veiklos įvykių sąrašas.	18
Lentelė 2. Užklausų panaudojamumo kriterijai. Užklausų panaudojamumo priklausomybė nuo duomenų struktūrų.	46

Santrauka

Šiame darbe sprendžiama problema, iškilusi kuriant medicininių tyrimų duomenų registravimo informacinę sistemą. Šioje srityje vieną tyrimą gali charakterizuoti net keli šimtai atributų. Taigi, problematiką sudaro duomenų bazės, kurios lentelės turi 200-300 laukų, projektavimas bei veikimas.

Darbui atlikti pasirinkta Oracle duomenų bazių valdymo sistema. Nagrinėjami pagrindiniai veiksniai, lemiantys duomenų apdorojimo spartą: duomenų saugojimo diske principai, užklausų vykdymo mechanizmas bei veikimo greitinimo galimybės.

Remiantis reliacinių duomenų bazių projektavimo principais, sukurtos galimos daugiaatribučių lentelių alternatyvos: normalizuota ir denormalizuota struktūros.

Eksperimento metu atliktas dažniausiai pasitaikančių užklausų greičio tyrimas bei užklausų sudėtingumo ir panaudojamumo analizė kiekvienos duomenų struktūros atveju.

Abstract

This paper is a research of problem which was faced during development of clinical information system. This is a scope, where one clinical test can be characterised by several hundreds of attributes. Thus the focus is on the design and performance of database tables with 200-300 columns.

To perform the research the Oracle Database Management System was chosen. There is the main factors, influencing performance, such as principles of data storage on disk, query processing also the main facilities to improve performance observed.

Based on the basic principles of relational database design some alternatives to tables with many attributes were created. It is normalized and denormalized structures.

There is performed the experiment to see how the speed of most used queries depends on data structure. Also there is done the analysis of complexity and usability of these queries.

Raktiniai žodžiai

Reliacinė duomenų bazė, daugiaatributė lentelė, užklausos vykdymas, normalizavimas, denormalizavimas, užklausos greičio priklausomybės, užklausų panaudojamumas.

Key words

Relational data base, table with many columns, query processing, normalizing, denormalizing, query speed dependencies, query usability.

Terminų ir santrumpų žodynas

Daugiaatributė lentelė – lentelė turinti daug (100 ir daugiau) stulpelių.

DBVS – duomenų bazių valdymo sistema.

Normalizavimas – reliacinių duomenų bazių projektavimo metu atliekama veiksmų seka, kurios tikslas – sutvarkyti duomenų bazę taip, kad joje nebūtų pasikartojančių duomenų.

Denormalizavimas – sąmoningas nukrypimas nuo normalizuotos schemos, siekiant pagerinti veikimą.

Užklausa – duomenų valdymo (įrašymo, atnaujinimo, šalinimo) ar išrinkimo iš duomenų bazės nurodymas.

SQL – *Structured Query Language*. Kalba, skirta užklausoms rašyti.

PL/SQL – *Programming Language / SQL*. SQL kalbos išplėtimas, leidžiantis kurti programuojamas struktūras.

1. Įvadas

Kiekvieną realaus pasaulio objektą galima apibūdinti tam tikrais požymiais. Pavyzdžiui, asmenį apibūdina vardas, pavardė, gimimo data, pastatą – vieta, tipas, užimamas plotas, vertė ir kt. Kompiuterizuojant dalykinę sritį, jei taikomos duomenų bazių valdymo sistemos (toliau – DBVS), kiekvienam objekto tipui sukuriama lentelė, kurioje požymiui skiriamas vienas stulpelis (atributas). Tačiau egzistuoja objektų, kuriems aprašyti neužtenka keleto požymių. Šis skaičius gali išaugti iki kelių dešimčių ar net kelių šimtų. Medicininių tyrimų atveju kiekvieno jų rezultatas gali būti 200-300 skaitinių reikšmių, taigi duomenų bazės lentelės tampa gigantiškos. Tačiau dauguma tyrimų yra atliekami etapais, tad reikšmių tipai periodiškai kartojasi. Tai suteikia galimybę modifikuoti lentelių struktūrą: jas suskaidyti į keletą ar keliolika mažesnių arba tiesiog skaidyti duomenis į keletą įrašų toje pačioje lentelėje.

Šis darbas sudarytas iš trijų dalių: analizės, sukurtos sistemos reikalavimų bei architektūros aprašymo ir eksperimento.

Kadangi sistemai realizuoti pasirinkta Oracle duomenų bazių valdymo sistema, analizės dalyje nagrinėsime įvairioje literatūroje aprašytus Oracle duomenų saugojimo, valdymo ir apdorojimo principus bei veikimo gerinimo galimybes. Taip pat apžvelgsime vieną pagrindinių reliacinių bazių projektavimo principų – normalizavimą bei nukrypimą nuo jo – denormalizavimą.

Sistemos aprašymui skirtoje dalyje pateikiamas jos sąsajos su aplinka, kelti reikalavimai, bei pagrindinį funkcionalumą atspindintys architektūros aspektai.

Eksperimentinėje dalyje, vadovaudamiesi analizės metu išdėstytomis normalizavimo ir denormalizavimo taisyklėmis, pateiksime galimų duomenų struktūrų fragmentų schemas, ištirsime sistemoje dažniausiai naudojamų užklausų veikimo greičio priklausomybes nuo duomenų struktūros, bei jų sudėtingumo ir panaudojamumo kitimą kiekvienu atveju.

Taigi šio darbo tikslas – atsižvelgiant į užklausų greitį ir jį lemiančius veiksnius, taip pat – į duomenų pasiekiamumą patogumą ir poreikius, parinkti nagrinėjamai informacinei sistemai tinkamiausią duomenų struktūrą.

2. Duomenų saugojimo ir apdorojimo principų analizė bei esminių projektavimo sąvokų apžvalga

Šiame skyriuje apžvelgsime įvairioje literatūroje aprašytas duomenų struktūras, paanalizuosime jų ypatumus. Taip pat panagrinėsime duomenų tvarkymo bei užklausų vykdymo mechanizmus, jų veikimo greitį ribojančius veiksnius bei optimizavimo galimybes.

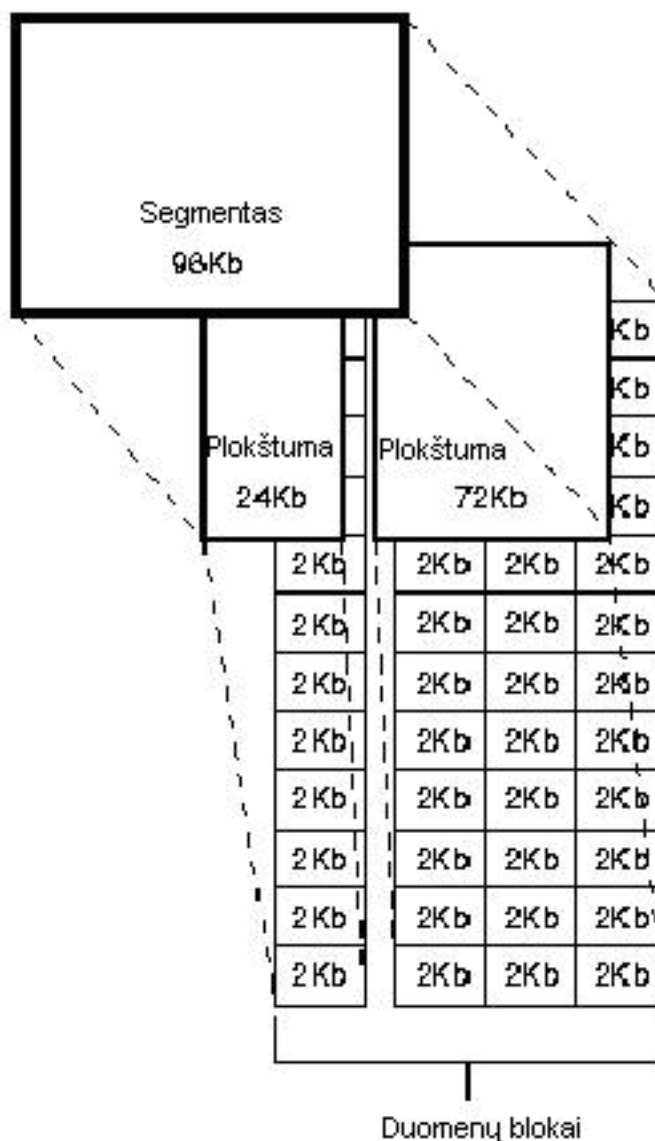
Kiekvienoje DBVS duomenų tvarkymo ir užklausų procesai fiziniame lygmenyje realizuoti skirtingai, todėl koncentruosimės ties viena – Oracle 9i – DBVS. Tokį pasirinkimą lėmė šios sistemos lankstumas ir suderinamumas su įvairiomis platformomis.

Skyriaus pabaigoje aptarsime pagrindines reliacinių duomenų bazių projektavimo sąvokas bei normalizavimo ir denormalizavimo principus, kuriais vadovausimės eksperimento metu.

2.1. Duomenų saugojimas

Norėdami suprasti, kas lemia operacijų su duomenimis greitį, turime pradėti aiškintis nuo paties žemiausio lygmens – loginio fizinės atminties suskirstymo bei gautų struktūrų tarpusavio ryšių.

2.1.1. Loginė duomenų bazės struktūra

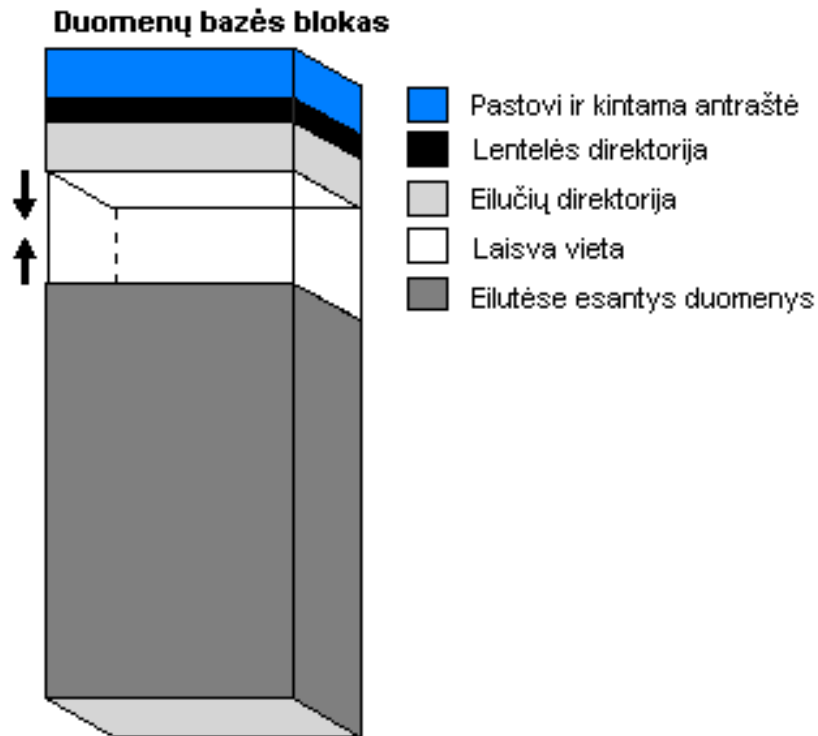


Paveikslas 1. Duomenų bazės loginė struktūra. [8]

Taigi pagal pirmąjį paveikslą matome, kad smulkiausia struktūros dalis yra duomenų blokas (puslapis), kurio dydis gali būti nustatomas duomenų bazės kūrimo metu. Vienas blokas atitinka tam tikrą baitų skaičių diske.

Gretimi blokai sujungiami į grupes – plokštumas. Kelias plokštumas, skirtas tam pačiam objektui saugoti, jungia segmentai (pvz., lentelės segmentas, indekso segmentas ir pan.). Pradinis segmento dydis yra lygus vienos plokštumos dydžiui, tačiau, esant reikalui, kuriamos naujos plokštumos, kurios gali būti ir ne gretimos diske. Segmentas gali apimti plokštumas netgi iš skirtingų fizinių failų.

Panagrinėkime duomenų bloko sandarą. Kadangi nagrinėsime veikimo greitinimo galimybes, ji mums yra svarbi.



Paveikslas 2. Duomenų bloko sandara. [8]

Taigi bloką sudaro tarnybinių bei vartotojo duomenų sritys. Tarnybiniai duomenys yra:

- Antraštė – joje saugoma bendra bloko informacija: adresas ir segmento tipas.
- Lentelės direktorija – čia saugoma informacija apie lentelę, kurios eilutės yra šiame duomenų bloke.
- Eilutės direktorija – informacija apie bloke saugomas eilutes.

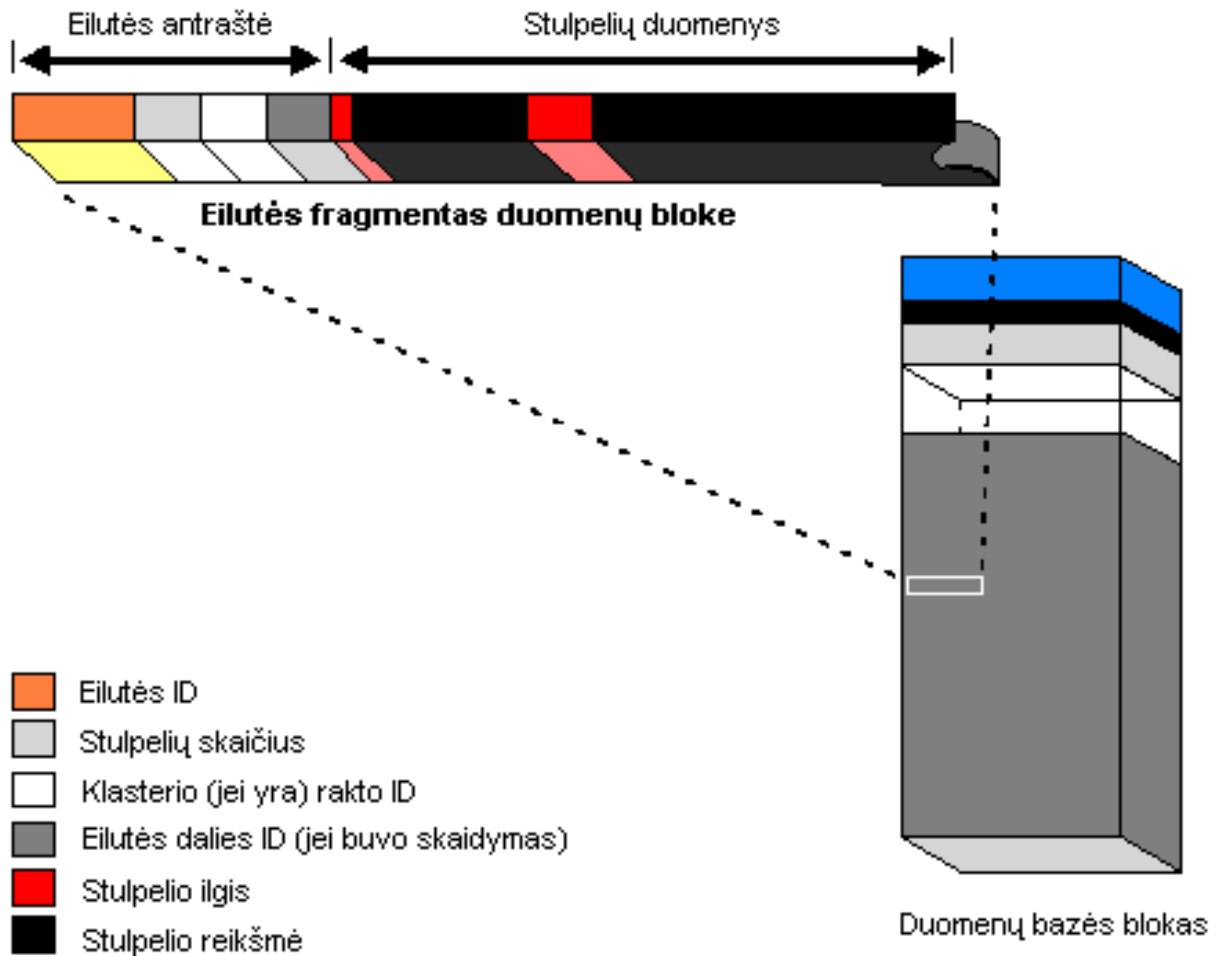
Oracle leidžia valdyti vartotojo srities panaudojimą. Reguluojant laisvos ir užpildytos vietos bloke santykį pagal prognozuojamą duomenų kintamumą, galima didinti sistemos darbo greitį ir vietos diske panaudojimo efektyvumą. Plačiau tai bus aptarta 2.3. skyriuje.

Sekančiame skyrelyje bus nagrinėjama, kaip bloke saugomi vartotojo duomenys.

2.1.2. Įrašų saugojimas ir kreipiniai į juos

Lentelės eilutė, jei jos ilgis neviršija 255 stulpelių, gali būti saugoma viename bloke, tačiau, jei visa netelpa, likusi jos dalis keliama į kitą. Šios eilučių dalys susiejamos tuo pačiu identifikaciniu kodu. Jei eilutės stulpelių skaičius viršija 255, ji skaidoma ir, jei bloke telpa visa,

atskiros jos dalys susiejamos bloko viduje. Pastaruoju atveju vartotojas greičio sumažėjimo nepajus.



Paveikslas 3. Lentelės eilutė ir duomenų blokas.[9]

Trečiajame paveiksle matome, jog eilutė visada turi antraštę, kurią sudaro identifikacinis numeris, stulpelių skaičius, klasterio (jei jis yra – apie tai taip pat kalbėsime vėliau) rakto identifikacinis numeris ir, jei eilutė buvo skaidyta, jos dalies identifikacinis numeris. Viename bloke telpančios eilutės antraštė užima mažiausiai 3 baitus.

Po antraštės paeiliui seka stulpelio ilgis ir stulpelio reikšmė. Jų seka dažniausiai yra beveik tokia pati, kaip ir aprašoma kuriant lentelę. Į pabaigą paprastai nukeliami tik stulpeliai, kurių reikšmės gali užimti labai daug vietos (pvz., „long“ tipo duomenys). Modifikavimo metu pridėdant naujus stulpelius, jie taip pat jungiami prie eilutės pabaigos. Stulpelio ilgiui, jei stulpelyje saugomi duomenys neviršija 250 baitų, aprašyti užtenka 1 baito. Priešingu atveju reikia 3. Stulpelio reikšmės ilgis priklauso nuo duomenų tipo. Kintamo ilgio duomenų tipo reikšmėms

vieta skiriama dinamiškai. Jei duomenų ilgis fiksuotas – jiems rezervuojama maksimalaus ilgio erdvė. Tačiau jei stulpelio reikšmė yra *null*, atminties jai neskiriama. Jei šis stulpelis įrašė ne paskutinis ir už jo esantys turi priskirtas reikšmes, naudojamas vienas baitas aprašyti stulpelio ilgiui, kuris lygus nuliui. Jei toliau sekančių stulpelių reikšmės yra *null*, nei jų ilgiams, nei reikšmėms saugoti atminties neskiriama – tai parodo naujos eilutės pradžia. Iš čia seka išvada, jog „daug stulpelių turinčias lenteles reikėtų sudarinėti taip, jog stulpeliai, kurių reikšmės gali būti *null*, būtų pabaigoje“ [9].

2.2. Duomenų tvarkymas bei užklausos

2.2.1. Užklausų tipai

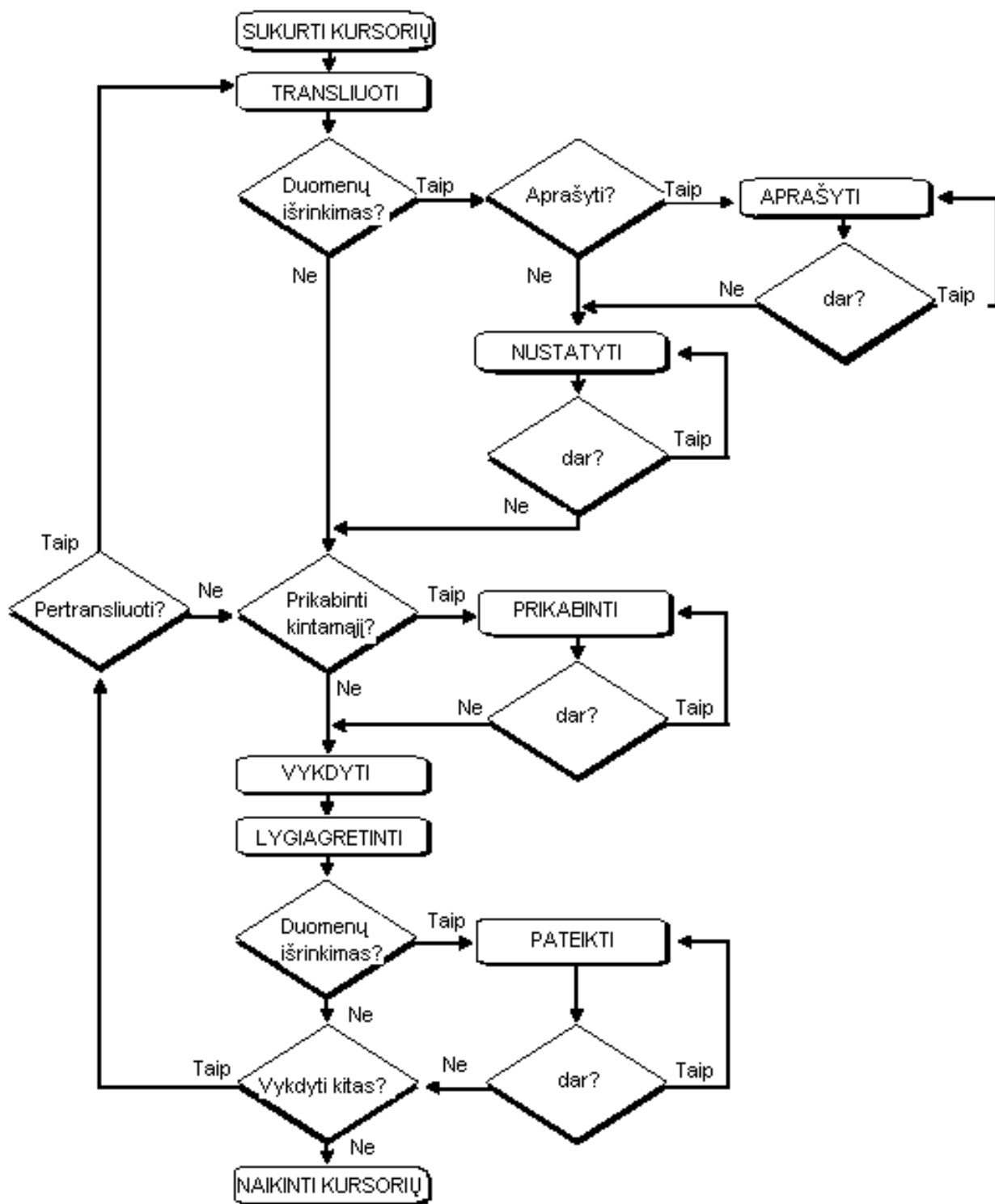
Oracle galimos įvairių paskirčių užklausos: duomenų valdymo (*Data Manipulation Language, DML*), duomenų nustatymo (*Data Definition Language, DDL*), transakcijų, vartotojo seansų, sistemos valdymo ir kitos. Šiame straipsnyje visų išnagrinėti nepajėgtume, todėl koncentruosimės į duomenų valdymo užklausas. Jos skirstomos taip:

- duomenų išrinkimas (SELECT);
- naujų eilučių įrašymas (įterpimas) (INSERT);
- esamų duomenų modifikavimas (UPDATE);
- esamų eilučių šalinimas (DELETE);
- duomenų modifikavimas arba įrašymas pagal tam tikras sąlygas (MERGE);
- užklausos vykdymo plano peržiūra (EXPLAIN PLAN);
- laikinas lentelės ar vaizdinio pasiekiamumo apribojimas (užrakinimas) (LOCK TABLE).

Dažniausiai naudojamos 4 pirmųjų tipų užklausos. Taigi toliau nagrinėsime tik jas.

2.2.2. Užklausos vykdymo eiga

Bendra užklausos vykdymo schema pateikiama ketvirtajame paveiksle:



Paveikslas 4. Užklauso vykdomas [10]

Kiekviena užklausa vykdoma penkiais būtinais etapais:

- sukuriamas kursorius;
- užklausa transliuojama;
- prikabinami kintamieji;

- užklausa vykdoma;
- kursorius naikinamas.

Papildomai užklauskos vykdymas gali būti lygiagretinamas. Tačiau „duomenų išrinkimas skiriasi nuo kitų užklauskų tuo, kad jos rezultatas, skirtingai negu kitų – pavyko arba nepavyko, - yra įrašų aibė“ [10]. Todėl papildomai turi būti apibrėžta ir nustatyta rezultatų saugojimo vieta laikinojoje atmintyje ir rezultatai nukreipti į ją. Iš čia jie siunčiami vartotojui.

Dabar panagrinėkime užklauskos transliavimo mechanizmą. Tarkime, vartotojo aplikacija inicijuoja duomenų išrinkimą. Užklausa siunčiama transliuoti ir įkeliama į privačią SQL sritį. Čia tikrinama, ar joje nėra klaidų. Jei užklausa teisinga, ji imama transliuoti, o sutransliuotas ruošinys perkeliamas į visiems pasiekiamą bendrą atminties zoną. Tokiu būdu šis, vieną sykį apdorotas, ruošinys gali būti vykdomas daug kartų, nes kaskart, vykdant užklausa, ieškoma, ar bendrojoje srityje nėra atitinkamo ruošinio. Šitaip darbas žymiai pagreitinamas. Tačiau atminties dydis yra fiksuotas, todėl seniausiais naudoti užklauskų ruošiniai iš jos šalinami.

Už užklauskos vykdymo greitį Oracle atsakingas optimizavimo mechanizmas. Pagal kaupiamas statistikas yra sprendžiama, ar indeksuotai lentelei taikyti indeksus, ar atlikti pilną jos peržiūrą, sudėtingos SQL sakinių struktūros transformuojamos į paprastesnes. Taigi kuo daugiau susietų lentelių, tuo daugiau painumo įneša ryšiai. Išrinkimas iš vienos lentelės šiuo požiūriu yra kur kas paprastesnis.

2.3. Veikimo greitinimo galimybės

2.3.1. Indeksavimas

Indeksas – duomenų bazėje saugoma struktūra, leidžianti rūšiuoti duomenis. Rūšiavimas gali būti atliekamas pagal vieną ar kelis lentelės stulpelius, taip pat indeksų vienai lentelei gali būti ne vienas. Esant daug duomenų, išrinkimas iš surūšiuoto jų sąrašo žymiai pagreitėja. Tačiau didelis indeksų kiekis stabdo duomenų atnaujinimą, įterpimą ir šalinimą, nes po kiekvienos tokios operacijos, indeksai turi būti atnaujinti.

Senesnėse Oracle DBVS versijose didelis indeksų kiekis buvo neišvengiamas, nes, sukūrus sudėtinį (kelis stulpelius apimantį) indeksą, „užklausoje, kurios WHERE dalyje nenaudoja pirmojo indekso stulpelio, jis būdavo netaikomas, ir užklausa tapdavo pasmerkta pilnai lentelės peržiūrai“ [4]. Oracle 9i šios problemos nebeliko. „Sudėtinis indeksas skaidomas į subindeksus,

kurių kiekis lygus pirmojo indekso stulpelio reikšmių kiekiui.“ [4] Rūšijimas atliekamas kiekvieno subindekso viduje.

Jei lentelės neindeksuotos, duomenys išrenkami pilnos peržiūros būdu, tačiau D. Aldridge rekomenduoja klasifikatorių lentelių neindeksuoti.

2.3.2. Klasterių paskirtis ir naudojimas

Klasterių kūrimas ir naudojimas – alternatyvus Oracle duomenų saugojimo būdas. Klasterio esmė – sujungti keletą lentelių, turinčių tokius pačius stulpelius ir bendrą informaciją saugoti tik vienoje disko srityje. Dažniausiai tai naudojama raktų ir išorinių raktų reikšmių saugojimui. Nenaudojant klasterių ta pati informacija saugoma keliose vietose.

Kiekvienas klasteris turi raktą – „stulpelį, ar stulpelių grupę, bendrą sujungiamoms lentelėms. <...> Kiekviena klasterio rakto reikšmė, nepriklausomai nuo to, kiek įrašų sujungtose lentelėse ją turės, klasteryje ir jo indekse saugoma tik vieną kartą.“ [11]

Tačiau nepaisant šių privalumų, Oracle 9i administratoriaus žinyne patariama klasterius taikyti tik toms lentelėms, iš kurių:

- „išrinkimas vykdomas žymiai dažniau, nei įrašymas, atnaujinimas ar šalinimas;
- įrašai dažniausiai išrenkami kartu, t.y. retai reikalinga informacija iš kaž kurios lentelės atskirai.“[11]

2.3.3. Eilučių identifikatorių taikymas

Kaip jau buvo rašyta anksčiau, įrašų išrinkti galima keliais būdais: pilnos peržiūros arba naudojant indeksą. Tačiau greičiausias išrinkimo būdas yra naudojant eilutės identifikatorių (*ROWID*). Šie identifikatoriai gali būti fiziniai arba loginiai.

„Fizinis identifikatorius – fizinis įrašo adresas, kuris nuo sukūrimo iki pašalinimo išlieka nepakitęs. Naudojamas neindeksuotose lentelėse, klasteriais sujungtose lentelėse, indeksuose o taip pat – lentelių ar indeksų dalyse (*partitions*).“[12]

„Loginis identifikatorius skirtas saugoti įrašų adresus indeksuotose lentelėse“. [12] Šis adresas nėra pastovus. Jis kinta priklausomai nuo įrašų rūšijavimo metu įrašo užimtos pozicijos.

ROWID yra kiekvienos duomenų bazės lentelės pseudostulpelis. Jis nėra saugomas duomenų bazėje, tačiau gali būti gautas *SELECT* sakinyje prirašius jo pavadinimą: *SELECT *, ROWID*.

Eilučių identifikatorius pravartu naudoti tada, kai duomenys pradžioje yra išrenkami, o paskui atnaujinami, šalinami arba išrenkami pakartotinai. Tačiau būtina užtikrinti, jog darbo eigoje *ROWID* nebūtų keičiami.

2.3.4. Duomenų bloko pildymo valdymas

Priklausomai nuo prognozuojamų duomenų ir kreipinių į juos tipo, kuriant loginį duomenų bazės objektą (lentelę, indeksą, klasterį) galima nustatyti jo naudojamų blokų pildymo eigą. Tam skirti parametrai PCTFREE ir PCTUSED. Jų vertės reiškia vietą bloke procentais ir nei kiekviena atskirai, nei abi kartu negali viršyti 100.

PCTFREE skirtas išlaikyti bloke tam tikrą kiekį laisvos vietos, kuri rezervuojama atnaujinimo operacijų metu galimiems duomenų ilgio padidėjimams. Kai bloke daugėja sukauptos informacijos ir laisvos vietos kiekis sumažėja iki šio parametro nustatytos ribos, įrašymas į bloką nebeleidžiamas. „Maža PCTFREE reikšmė leidžia taupyti vietą ir įrašymo metu bloką užpildyti taupiau, tačiau atnaujinant didesnė tikimybė, jog įrašas bus paskirstytas per kelis blokus. Didelė PCTFREE reikšmė, atvirkščiai, leidžia atnaujinti duomenis efektyviau, tačiau visai segmento informacijai saugoti prireiks daugiau vietos.“ [13] Pradinė šio parametro reikšmė yra 10.

PCTUSED paskirtis – mažėjant bloke esančios informacijos apimčiai, rezervuoti bloką tik atnaujinimo operacijoms kol užimtos vietos kiekis netaps mažesnis, nei šiuo parametru nustatyta. Taigi duomenų atnaujinimas vyks efektyviau ir įrašai bus mažiau suskaidyti, jei ši reikšmė bus mažesnė, tačiau nepanaudojamos laisvos vietos diske bus daugiau.

Tad parenkant šių parametrų reikšmes reikia atsižvelgti į duomenų kitimo tendenciją – kaip stipriai jų ilgai ir užimama vieta linkusi keistis, ar dažnai bus vykdomas atnaujinimas ir pan. Vistik D. Aldridge siūlo klasifikatorių lentelėms nustatyti PCTFREE=0, o duomenų lentelėms – PCTFREE=1. Abiem atvejais PCTUSED siūloma reikšmė 99. Tiesa, autorius rekomenduoja panagrinėti tarnybinės lentelės DBA_TABLES saugomą statistiką.

2.4. Duomenų struktūrų projektavimas: normalizavimo ir denormalizavimo procesai.

Iš ankstesnių skyrelių tampa aišku, jog projektuojant duomenis, išskyrus reikiamas esybes, jas charakterizuojančius atributus bei nustačius tarpusavio ryšius, būtina sukurti reikiamus raktus, indeksus taip pat reikia surūšiuoti atributus, atsižvelgiant į jų būsimas reikšmes. Toliau tęsiant darbą, būtina užtikrinti kiek galima geresnį duomenų panaudojamumą bei vientisumą, apsaugą nuo netyčinio sugadinimo ir pan. Tam reikia atlikti duomenų struktūros normalizavimą – „sutvarkymą“ pagal tam tikras taisykles. Šiame skyriuje trumpai apžvelgsime normalizavimo sampratą ir pakopas, o taip pat panagrinėsime išimtis – kada normalizuotoje schemoje galimas tyčinis duomenų dubliavimas.

2.4.1. Normalizavimo samprata

Normalizavimas – procesas, taikomas projektuojant reliacines duomenų bazes. Jo tikslai (pagal R. E. Wyllys):

1. Duomenų bazėje turi būti saugomi visi reikalingi duomenys.
2. Duomenų perteklius turi būti kiek galima mažesnis.
3. Visos reikšmės rašomos į joms skirto duomenų tipo laukus.
4. Duomenų atnaujinimas turi būti vykdomas efektyviai.
5. Vengiama netyčinio duomenų praradimo pavojaus.

Paprastai yra normalizuojama mažiausiai iki 3 normalinės formos (3NF) (normalizavimo lygmenys bus aprašyti vėliau). Tokiu būdu duomenų bazė tampa apsaugota nuo „galimų duomenų sugadinimų, vadinamųjų „įterpimo anomalijų“, „šalinimo anomalijų“ ir „atnaujinimo anomalijų“ [7]. Šio tipo klaidos dažnos nepakankamai normalizuotose bazėse, kuriose duomenys gali būti vienaip ar kitaip sudubliuoti (pertekliniai). Pavyzdžiui, duomenų įrašymo atveju gali pasitaikyti taip, jog duomenis reikės įrašyti daugiau nei į vieną vietą, šalinimo metu gali būti pašalinti ne tik reikiami, bet ir tėviniai duomenys, o atnaujinimo veiksmas apima tiek įrašymą, tiek šalinimą, todėl galimos abiejų tipų anomalijų kombinacijos.

2.4.2. Normalizavimo pakopos

Normalizavimo procesas yra atliekamas pažingsniui. Kiekvieno žingsnio rezultatas – vis aukštesnė normalinė forma. Apžvelkime šių lygmenų apibrėžimus (pagal R. E. Wyllys).

1-oji normalinė forma (1NF)

Lentelė yra 1NF, jei :

- joje nėra pasikartojančių eilučių;
- nėra pasikartojančių duomenų grupių ar masyvų;
- kiekvieno stulpelio visi duomenys yra to paties tipo.

Reikalavimas, kad duomenys nesikartotų, reiškia, jog lentelė turi turėti raktą, kuris gali būti sudarytas iš vieno ar kelių stulpelių (pastaruoju atveju jis vadinsis sudėtinis).

2-oji normalinė forma (2NF)

Lentelė yra 2 NF, jei ji yra 1NF bei joje nėra dalinių priklausomybių. Tai reiškia, jog sudėtinio rakto atveju, visi kiti lentelės stulpeliai turi priklausyti nuo viso rakto, o ne nuo atskirų jo dalių.

3-oji normalinė forma (3NF)

Lentelė yra 3NF, jei ji yra 2NF, bei joje nėra tranzityvių priklausomybių. Tranzityvią priklausomybę galima paaiškinti taip: jei B priklauso nuo A, o C priklauso nuo B, tai C priklauso nuo A. Šią situaciją galima iliustruoti trumpu pavyzdžiu: knyga priklauso bibliotekos padaliniui, padalinys – bibliotekai, vadinasi: knyga priklauso bibliotekai. Lentelę, kurios duomenys pasižymi tokia priklausomybe, reikėtų skaidyti.

Kaip jau buvo minėta, lentelei, normalizuotai iki 3NF, duomenų sugadinimo anomalijos jau nebegresia, tačiau tobulumas dar nepasiektas ir galimas tolesnis normalizavimas, kuris šiame darbe nagrinėjamas nebus.

2.4.3. Denormalizavimo poreikis

„Denormalizavimas suteikia galimybę pagerinti sistemos veikimą, leidžiant duomenims sistemingai kartotis. Tačiau ši metodika taikytina tik kraštutiniais atvejais, tada, kai kitos veikimo gerinimo galimybės, tokios kaip indeksavimas ir pan., jau išsemtos.“[1]

„Atlikus denormalizavimą, kai kurių duomenų pasiekiamumas gali stipriai pagreitėti, tačiau tai – visuomet papildomos išlaidos: sulėtėja duomenų apdorojimas, atsiranda anksčiau minėtų įrašymo, šalinimo anomalijų rizika, kiekviena denormalizuota lentelė reikalauja papildomo programavimo.“[1] Tačiau, atsižvelgiant į tai, jog toliau nagrinėsime įvairių struktūrų veikimą, apžvelgti pagrindinius denormalizavimo taikymo atvejus reikia.

2.4.4. Denormalizavimo taikymo atvejai

Šis skyrelis parengtas pagal J. Speelpenning, P. Daux, J. Gallus.

1. Skaičiuojamųjų reikšmių saugojimas

Gali būti taikomas tada, kai duomenys ir rezultatai saugomi skirtingose lentelėse, rezultato reikia dažnai, jis retai kinta. Tačiau šiuo atveju turi būti užtikrintas rezultato perskaičiavimas kiekvienu duomenų pakeitimo atveju.

2. *Ryšio atsisakymas*

Šis būdas naudojamas tada, kada yra dažnas poreikis kartu su vaikinės lentelės duomenimis išgauti ne raktinių tėvinės lentelės laukų reikšmes. Tokiu atveju šie laukai perkeliama į vaikinę lentelę. Tačiau tai vis tik yra „balastinių“ duomenų saugojimas, be to, reikalaujantis papildomų jų atnaujinimo.

3. *Fiksuotos reikšmės*

Tai denormalizavimo būdas, galimas naudoti klasifikatoriams kurti. Klasifikatorių lentelėse įrašų dažniausiai būna nedaug, jų reikšmės kinta retai. Jei klasifikatoriaus duomenys imami iš kelių lentelių, kartais patogiau lentelių skaičių sumažinti, tėvinės lentelės įrašų reikšmes fiksuojant greta vaikinės lentelės įrašų. Tačiau tokiu atveju bus reikalingos apsaugos priemonės, neleidžiančios įvesti kitokių negu fiksuotos reikšmių.

4. *Vaikinių laukų saugojimas kartu su tėviniais*

Jei pasitaiko atvejis, kada yra dažnas poreikis tėvinės lentelės duomenų ir, greta, vaikinės lentelės laukų reikšmių, galima šiuos laukus perkelti į tėvinę lentelę. Tačiau tai naudinga tik tais atvejais, kada vaikinėje lentelėje įrašų kiekis nedidelis, tarkime, mažesnis už 30. Šiuo veiksmu sutaupome vietos ir laiko, nes nebelieka ryšių ir bereikalingų raktų.

5. *Trumpų lentelių grandinių raktų saugojimas*

Duomenų struktūrose gana dažnai pasitaiko trumpų grandinių formos dariniai. Tarkime, A yra tėvinis B, o B yra tėvinis C. Tačiau dažnai reikalingas ryšys „senelis – vaikai“. Tokiu atveju, į „vaikaičio“ lentelę galime įkomponuoti „senelio“ raktą. Taip sutrumpėja užklausa, tačiau, kaip ir visais denormalizavimo atvejais, modifikuojant duomenis reikalingi papildomi veiksmai, užtikrinantys jų vientisumą.

3. Sukurtos sistemos aprašymas

Kaip jau buvo minėta, šio darbo problematikos šaltinis yra atliktas projektas „Kardiovaskulinės reabilitacijos informacinė sistema“. Šiame skyriuje nagrinėsime sistemai keltus reikalavimus, jos architektūrą bei su tuo susijusius duomenų struktūros ypatumus.

3.1. Sistemos paskirtis, vartotojai ir apribojimai

Sukurtos sistemos paskirtis internetu registruoti ir kaupti klinikos pacientų ir jiems atliktų tyrimų informaciją bei pateikti ją suderinamu su statistikos programų paketais formatu. Sistemos vartotojai – klinikos personalas: gydytojai bei medicinos seserys.

Kadangi sistema yra prieinama naršyklės pagalba, išskirtinių reikalavimų vartotojų aplinkai nėra: tinka Windows arba Linux operacinė sistema bei bet kuri jų palaikoma naršyklė. Realizuoti projektavimo metu iškelti reikalavimai sąsajos su kitomis posistemėmis galimybei.

3.2. Veiklos sfera

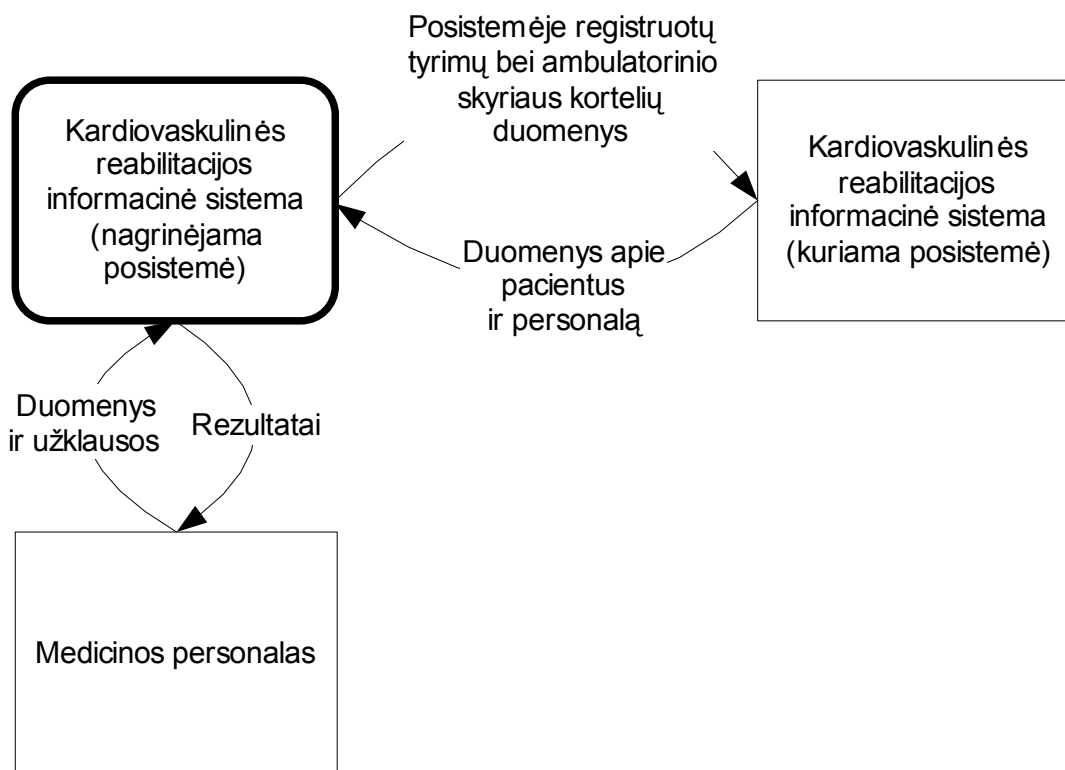
Veiklos sferos nustatymas – pirminė reikalavimų išgavimo ir formulavimo proceso stadija. Šiame skyriuje pateiktas sistemos ir ryšių su aplinka bei pagrindinių veiklos įvykių aprašymas.

3.2.1. Veiklos kontekstas

Penktajame paveiksle vaizduojama šiame darbe nagrinėjamos kardiovaskulinės reabilitacijos informacinės sistemos aplinka ir ryšiai su ja.

Medicinos personalas – gydytojai, med. seserys – įveda pacientų tyrimų rezultatus bei ambulatorinio skyriaus pacientų duomenis. Pagal poreikį duomenys gali būti peržiūrimi ir redaguojami. Kiekvieno paciento tyrimo duomenys gali būti pateikti kableliniu formatu. Toks pateikimas reikalingas, norint juos nukreipti statistikos paketams.

Kadangi vartotojų valdymas bei klinikos gydomojo skyriaus pacientų registracija planuojama realizuoti kitoje posistemėje, yra užtikrinta šios sąsajos galimybė.



Paveikslas 5. Veiklos konteksto diagrama.

3.2.2. Veiklos padalinimas

Pirmojoje lentelėje pateikti veiklos įvykiai – vartotojo „bendravimo“ su sistema atvejai. Kiekvieno jų metu sistema su aplinka keičiasi duomenimis: vienus gauna, kitus formuoja bei pateikia vartotojui.

Lentelė 1. Veiklos įvykių sąrašas.

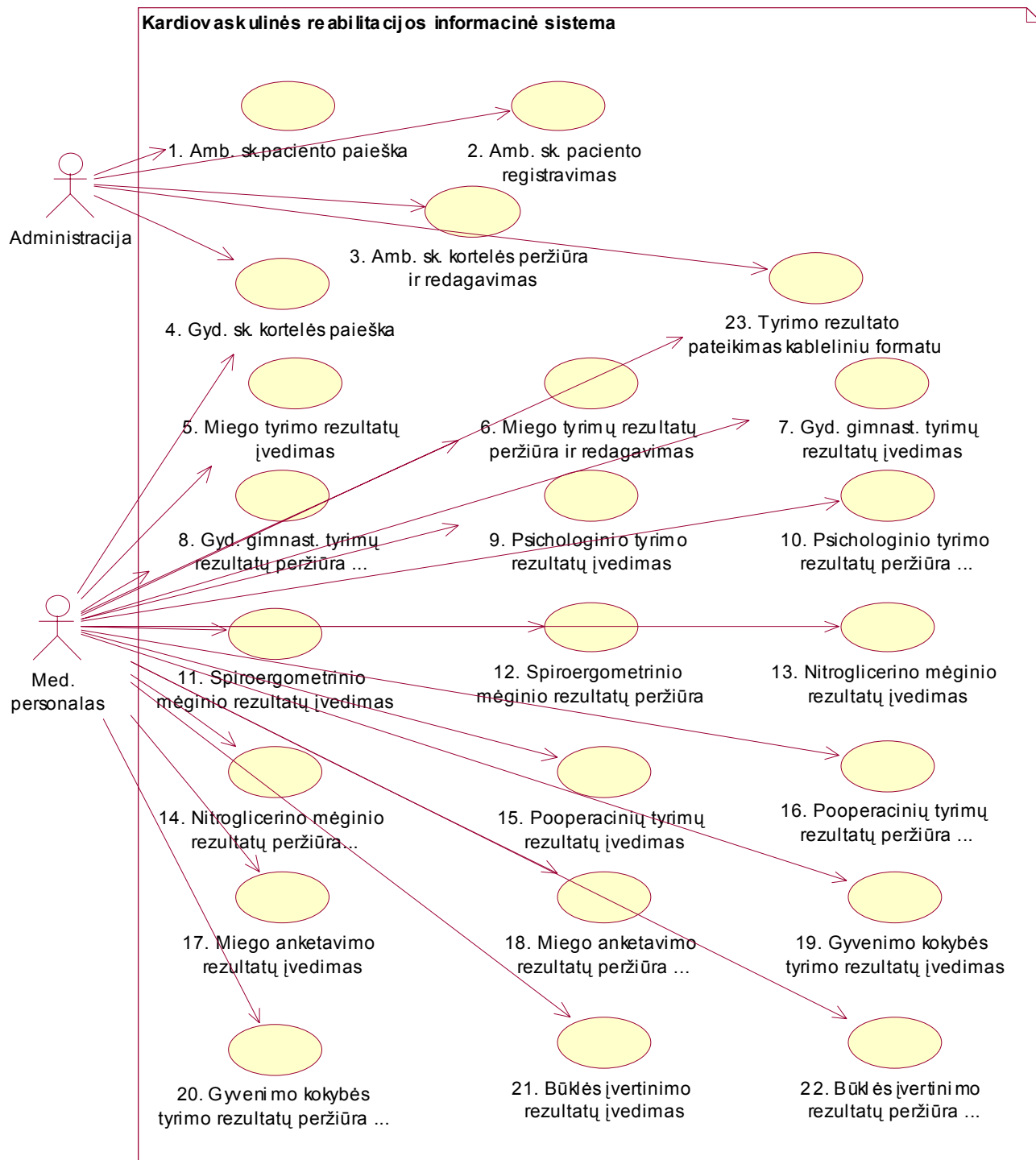
<i>Eil. Nr.</i>	<i>Įvykio pavadinimas</i>	<i>Įeinantys/Išeinantys informacijos srautai</i>
1	Administratorė atlieka ambulatorinės kortelės paiešką.	Paieškos užklausa (<i>in</i>). Duomenys apie užklausą tenkinančias korteles (<i>out</i>).
2	Administratorė registruoja naują ambulatorinio skyriaus pacientą arba atlieka jau esančių duomenų redagavimą ar peržiūrą.	Duomenys apie pacientą (<i>in</i>). Duomenys apie pacientą (<i>out</i>).
3	Gydytojas pradeda darbą – atlieka paciento paiešką.	Paieškos užklausa (<i>in</i>).

		Duomenys apie užklausa tenkinančius pacientus bei prieiga prie jų tyrimų duomenų (<i>out</i>).
4	Gydytojas įveda konkretaus paciento miego tyrimo rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji miego tyrimo rezultatai (<i>in</i>). Atlikto miego tyrimo duomenys (<i>out</i>).
5	Gydytojas įveda konkretaus paciento tyrimų gydomosios gimnastikos metu rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji tyrimų gyd.gimnastikos metu rezultatai (<i>in</i>). Atlikto gyd. gimnastikos tyrimo rezultatai (<i>out</i>).
6	Gydytojas įveda konkretaus paciento psichologinio tyrimo rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji psichologinio tyrimo rezultatai (<i>in</i>). Atlikto psichologinio tyrimo rezultatai (<i>out</i>).
7	Gydytojas įveda konkretaus paciento spiroergometrinio mėginio rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji spiroergometrinio mėginio rezultatai (<i>in</i>). Atlikto spiroergometrinio mėginio rezultatai (<i>out</i>).
8	Gydytojas įveda konkretaus paciento nitroglicerino mėginio rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji nitroglicerino mėginio rezultatai (<i>in</i>). Atlikto nitroglicerino mėginio rezultatai (<i>out</i>).
9	Gydytojas įveda konkretaus paciento pooperacinių tyrimų rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji pooperacinių tyrimų rezultatai (<i>in</i>). Atliktų pooperacinių tyrimų rezultatai (<i>out</i>).
10	Gydytojas įveda konkretaus paciento subjektyvaus miego anketavimo rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji subjektyvaus miego anketavimo rezultatai (<i>in</i>). Atlikto subjektyvaus miego anketavimo rezultatai (<i>out</i>).
11	Gydytojas įveda konkretaus paciento gyvenimo	Nauji gyvenimo kokybės tyrimo

	kokybės tyrimo rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	rezultatai (<i>in</i>). Atlikto gyvenimo kokybės tyrimo rezultatai (<i>out</i>).
12	Gydytojas įveda konkretaus paciento bendro būklės įvertinimo rezultatus arba atlieka jau esančių duomenų redagavimą, peržiūrą ar išvedimą kableliniu formatu.	Nauji bendro būklės įvertinimo rezultatai (<i>in</i>). Atlikto bendro būklės įvertinimo rezultatai (<i>out</i>).

3.3. Panaudojimo atvejai

Panaudojimo atvejai – tolimesnis žingsnis formalizuojant vartotojo reikalavimus bei transformuojant juos į konkrečius techninės užduoties punktus. Šeštajame paveiksle pavaizduojami pagrindiniai panaudojimo atvejai, atspindintys dalykinę sritį, bei vartotojų ryšiai su jais.



Paveikslas 6. Panaudojimo atvejų diagrama.

Panaudojimo atvejų aprašymas

1. Ambulatorinio skyriaus paciento kortelių paieška

Vartotojas/Aktorius:	Administratorė.
Aprašas:	Apima ambulatorinio skyriaus paciento kortelės paieškos procesą.
Prieš sąlyga:	Yra įvestų kortelių.
Sužadinimo sąlyga:	Atsirado poreikis surasti kortelę.
Po sąlyga:	Surasta viena arba kelios kortelės. Gali būti nerasta nei vienos.

2. Ambulatorinio skyriaus paciento duomenų registravimas

Vartotojas/Aktorius:	Administratorė.
Aprašas:	Apima naujo ambulatorinio skyriaus paciento registravimo procesą.
Prieš sąlyga:	Pacientas dar neužregistruotas.
Sužadinimo sąlyga:	Atvyko naujas pacientas.
Po sąlyga:	Pacientas užregistruotas.

3. Ambulatorinio skyriaus paciento duomenų peržiūra ir redagavimas

Vartotojas/Aktorius:	Administratorė.
Aprašas:	Apima ambulatorinio skyriaus paciento kortelės peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas užregistruotas.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

4. Gydomojo skyriaus paciento duomenų paieška.

Vartotojas/Aktorius:	Administratorė, med. personalas.
Aprašas:	Apima gydomojo skyriaus paciento duomenų paieškos procesą.
Prieš sąlyga:	Yra užregistruotų gydomojo skyriaus pacientų.
Sužadinimo sąlyga:	Atsirado poreikis įvesti, redaguoti, peržiūrėti arba kableliniu formatu gauti paciento tyrimo rezultatus.
Po sąlyga:	Surastas vienas arba keli pacientai. Gali būti nerasta nei vieno.

5. Miego tyrimo rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima polisomnografinio miego tyrimo rezultatų įvedimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

6. Miego tyrimo rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima polisomnografinio miego tyrimo rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

7. Tyrimų gydomosios gimnastikos metu rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima tyrimų, atliktų gydomosios gimnastikos metu, rezultatų įvedimo procesą. Tyrimą sudaro: <ul style="list-style-type: none"> • Sinusinio širdies ritmo stebėjimas. • Gyd. gimnastikos poveikio vertinimas. Įvedimo metu turi būti skaičiuojami poveikio vertinimą nusakantys išvestiniai rodikliai. • Oksimetrija;
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

8. Tyrimų gydomosios gimnastikos metu rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima tyrimų, atliktų gydomosios gimnastikos metu, rezultatų peržiūros ir redagavimo procesą. Redagavimo metu turi būti perskaiciuojami poveikio vertinimą nusakantys išvestiniai rodikliai.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo

	duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

9. Psichologinio tyrimo rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima psichologinio tyrimo rezultatų įvedimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

10. Psichologinio tyrimo rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima psichologinio tyrimo rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

11. Spiroergometrinio mėginio rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima spiroergometrinio mėginio rezultatų įvedimo procesą. Įvedimo metu pagal įvestus požymius turi būti apskaičiuojami išvestiniai rodikliai.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

12. Spiroergometrinio mėginio rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima spiroergometrinio mėginio rezultatų peržiūros ir redagavimo procesą. Redagavimo metu turi būti

	perskaičiuojamos išvestinių rodiklių reikšmės.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

13. Nitroglicerino mėginio rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima nitroglicerino mėginio rezultatų įvedimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

14. Nitroglicerino mėginio rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima nitroglicerino mėginio rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

15. Pooperacinių tyrimų rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima pooperacinių tyrimų rezultatų įvedimo procesą. Tyrimai: <ul style="list-style-type: none"> • koronarografija; • revaskuliarizacija.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

16. Pooperacinių tyrimų rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
-----------------------------	------------------

Aprašas:	Apima pooperacinių tyrimų rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

17. Subjektyvaus miego anketavimo rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima subjektyvaus miego anketavimo rezultatų įvedimo procesą. Anketos: <ul style="list-style-type: none"> • Pitsburgo miego kokybės indeksas; • miego logaritmas; • žodinis miego vertinimas; • Teilorio metodika; • Spilbergerio skalė; • Epworth mieguistumo skalė.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

18. Subjektyvaus miego anketavimo rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima subjektyvaus miego anketavimo rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

19. Gyvenimo kokybės tyrimo rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima gyvenimo kokybės tyrimo rezultatų įvedimo procesą. Anketos: <ul style="list-style-type: none"> • SF-36 klausimynas; • katalikybės anketa;

	• HAD klausimynas.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

20. Gyvenimo kokybės tyrimo rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima gyvenimo kokybės tyrimo rezultatų peržiūros ir redagavimo procesą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

21. Bendro būklės įvertinimo rezultatų įvedimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima bendro būklės įvertinimo rezultatų įvedimo procesą. Įvedimo metu reikia atlikti būklės įvertinimo skaičiačiavimą.
Prieš sąlyga:	Pacientas surastas paieškos metu.
Sužadinimo sąlyga:	Atliktas naujas tyrimas ir gauti jo rezultatai.
Po sąlyga:	Tyrimo rezultatai įvesti.

22. Bendro būklės įvertinimo rezultatų peržiūra ir redagavimas

Vartotojas/Aktorius:	Med. personalas.
Aprašas:	Apima bendro būklės įvertinimo rezultatų peržiūros ir redagavimo procesą. Redagavimo metu reikia perskaičiuoti būklės įvertinimą.
Prieš sąlyga:	Pacientas surastas paieškos metu, tyrimo duomenys įvesti ir išsaugoti.
Sužadinimo sąlyga:	Poreikis peržiūrėti duomenis ir, jei jie pasikeitė arba buvo įvesti klaidingai, redaguoti.
Po sąlyga:	Paciento duomenys pateikti ekrane ir, jei buvo redaguoti, atnaujinti.

23. Paciento konkretaus tyrimo rezultatų pateikimas kableliniu formatu.

Vartotojas/Aktorius:	Med. personalas, administratorius.
-----------------------------	------------------------------------

Aprašas:	Surasto paciento konkretūs duomenys pateikiami kableliniu formatu. Jie gali būti nukreipiami į ekraną arba išsaugomi tekstiniame faile.
Prieš sąlyga:	Pacientas surastas paieškos metu. Įvesti ir išsaugoti reikiamo tyrimo rezultatai.
Sužadinimo sąlyga:	Atsirado poreikis gauti tyrimo rezultatus kableliniu pavidalu.
Po sąlyga:	Tyrimo rezultatai pateikti ekrane arba išsaugoti faile, kurio vardas sudarytas iš tyrimo pavadinimo fragmento, paciento kodo, bei tyrimo eilės numerio (jei toks yra).

3.4. Reikalavimai

3.4.1. Funkciniai reikalavimai

Reikalavimai paieškai

- 1) Turi būti realizuota ambulatorinio skyriaus paciento kortelės paieška pagal jo asmens kodą ir/arba registravimo datą ir/arba vardą ir/arba pavardę.
- 2) Turi būti galimas visų ambulatorinio skyriaus kortelių sąrašas, surūšiuotas pagal paciento asmens kodą ir atvykimo datą.
- 3) Turi būti leidžiama surasti gydomojo skyriaus paciento duomenis pagal jo registracijos numerį ir/arba asmens kodą ir/arba vardą ir/arba pavardę.

Reikalavimai duomenų įvedimui ir taisymui

- 4) Turi būti leidžiama įvesti ir koreguoti ambulatorinio skyriaus paciento duomenis. Lovadienių skaičius ir suma už visus lovadienius turi būti skaičiuoja automatiškai.
- 5) Turi būti leidžiama įvesti ir koreguoti polisomnografinio miego tyrimo rezultatus. Užtikrinti, kad tyrimo kiekis vienam pacientui neviršytų 9.
- 6) Turi būti leidžiama įvesti ir koreguoti sinusinio širdies ritmo duomenis. Šis tyrimas atliekamas gydomosios gimnastikos metu. Užtikrinti maksimalų tyrimų kiekį vienam pacientui – 9. Tyrimas atliekamas 7 etapais. Pagal vartotojo pasirinkimą turi būti leidžiamas įvedimas ir koregavimas kiekvieno etapo rezultatams atskirai arba viso tyrimo rezultatams iš karto.

- 7) Turi būti leidžiama įvesti ir koreguoti gydamosios gimnastikos poveikio vertinimo rezultatus.
- 8) Gydamosios gimnastikos poveikio vertinimo išvestiniai rodikliai turi būti apskaičiuojami automatiškai pagal įvestus požymius.
- 9) Turi būti leidžiama įvesti ir koreguoti oksimetrijos duomenis. Tai – gydamosios gimnastikos metu atliekas tyrimas. Jį sudaro 16 etapų. Pagal vartotojo pasirinkimą turi būti leidžiamas įvedimas ir koregavimas kiekvieno etapo rezultatams atskirai arba viso tyrimo rezultatams iš karto.
- 10) Turi būti leidžiama įvesti ir koreguoti psichologinio tyrimo rezultatus. Šio tyrimo kiekis vienam pacientui negali viršyti 3.
- 11) Turi būti leidžiama įvesti ir koreguoti spiroergometrinio mėginio rezultatus. Šis tyrimas atliekamas 34 etapais. Pagal vartotojo pasirinkimą turi būti leidžiamas įvedimas ir koregavimas kiekvieno etapo rezultatams atskirai arba viso tyrimo rezultatams iš karto.
- 12) Pagal kiekvieno spiroergometrinio mėginio etapo rezultatus turi būti automatiškai skaičiuojami išvestiniai rodikliai.
- 13) Turi būti leidžiama įvesti ir koreguoti nitroglicerino mėginio pirmos dalies rezultatus.
- 14) Turi būti leidžiama įvesti ir koreguoti nitroglicerino mėginio antros dalies rezultatus.
- 15) Turi būti leidžiama įvesti ir koreguoti koronarografijos rezultatus. Vienam pacientui tyrimas gali būti atliekamas vieną kartą.
- 16) Turi būti leidžiama įvesti ir koreguoti revaskuliarizacijos rezultatus. Vienam pacientui tyrimas gali būti atliekamas vieną kartą.
- 17) Turi būti leidžiama įvesti ir koreguoti anketos „Pitsburgo miego kokybės indeksas“ rezultatus. Vienam pacientui tyrimas gali būti atliekamas tik vieną kartą.
- 18) Turi būti leidžiama įvesti ir koreguoti anketos „Miego logaritmas“ rezultatus. Vienam pacientui tyrimų skaičius neribojamas.
- 19) Turi būti leidžiama įvesti ir koreguoti anketos „Žodinis miego vertinimas“ rezultatus. Vienam pacientui tyrimų skaičius neribojamas.
- 20) Turi būti leidžiama įvesti ir koreguoti anketos „Teilorio metodika“ rezultatus. Vienam pacientui tyrimų skaičius neribojamas.
- 21) Turi būti leidžiama įvesti ir koreguoti anketos „Spilbergerio skalė“ rezultatus. Vienam pacientui tyrimų skaičius neribojamas.
- 22) Turi būti leidžiama įvesti ir koreguoti anketos „Epworth mieguistumo skalė“ rezultatus. Vienam pacientui tyrimų skaičius neribojamas.

- 23) Turi būti leidžiama įvesti ir koreguoti anketos „SF-36 klausimynas“ rezultatus. Tai – gyvenimo kokybės tyrimo dalis. Vienam pacientui gali būti atliekama tik vieną kartą.
- 24) Turi būti leidžiama įvesti ir koreguoti katalikybės anketos rezultatus. Tai – gyvenimo kokybės tyrimo dalis. Vienam pacientui gali būti atliekama tik vieną kartą.
- 25) Turi būti leidžiama įvesti ir koreguoti anketos „HAD klausimynas“ rezultatus. Tai – gyvenimo kokybės tyrimo dalis. Vienam pacientui gali būti atliekama tik vieną kartą.
- 26) Turi būti leidžiama įvesti ir koreguoti būklės tyrimo rezultatus. Tyrimas atliekamas dviem etapais. Bendras būklės įvertinimas turi būti skaičiuojamas automatiškai.
- 27) Privalo būti užtikrinta įvedamų duomenų kontrolė:
 - asmens kodo teisingumo tikrinimas;
 - datos vientisumo tikrinimas;
 - laikotarpį nusakančių datų palyginimas;
 - skaitinių reikšmių vientisumo tikrinimas;
 - vardą, pavardę nurodančių reikšmių vientisumo tikrinimas;
 - kita.

Reikalavimai duomenų išvedimui

Visų tyrimų atvejais, jei yra jau įvestų paciento tyrimų bei apskaičiuotų rezultatų, turi būti galima jų peržiūra.

- 28) Ambulatorinio skyriaus paciento kortelės peržiūra. Pateikiama tame pačiame naršyklės lange.
- 29) Polisomnografinio miego tyrimo rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys turi būti sugrupuoti pagal matavimų seką.
- 30) Sinusinio širdies ritmo duomenų peržiūra. Pateikiama naujame naršyklės lange. Duomenys turi būti sugrupuoti etapais.
- 31) Gydomosios gimnastikos poveikio vertinimas. Pateikiama naujame naršyklės lange. Rodomi vartotojo įvesti požymiai bei suskaičiuoti išvestiniai rodikliai.
- 32) Oksimetrijos rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys turi būti sugrupuoti etapais.
- 33) Psichologinio tyrimo rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys turi būti sugrupuoti pagal vertinimo skales.

- 34) Spiroergometrinio mėginio rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys sugrupuoti etapais. Šalia vartotojo įvestų duomenų turi būti spausdinami to paties tyrimo išvestiniai rodikliai.
- 35) Nitroglicerino mėginio pirmos dalies rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys grupuojami pagal matavimų seką.
- 36) Nitroglicerino mėginio antros dalies rezultatų peržiūra. Pateikiama naujame naršyklės lange. Duomenys grupuojami pagal matavimų seką.
- 37) Koronarografijos rezultatų peržiūra. Duomenys pateikiami tame pačiame naršyklės lange.
- 38) Revaskuliarizacijos rezultatų peržiūra. Duomenys pateikiami tame pačiame naršyklės lange.
- 39) Anketos „Pitsburgo miego kokybės indeksas“ rezultatai. Pateikiami tame pačiame naršyklės lange.
- 40) Anketos „Miego logaritmas“ rezultatai. Pateikiami naujame naršyklės lange.
- 41) Anketos „Žodinis miego vertinimas“ rezultatai. Pateikiami naujame naršyklės lange.
- 42) Anketos „Teilorio metodika“ rezultatai. Pateikiami naujame naršyklės lange.
- 43) Anketos „Spilbergerio skalė“ rezultatai. Pateikiami naujame naršyklės lange.
- 44) Anketos „Epworth mieguistumo skalė“ rezultatai. Pateikiami naujame naršyklės lange.
- 45) Anketos „SF-36 klausimynas“ rezultatai. Pateikiami tame pačiame naršyklės lange.
- 46) Katalikybės anketos rezultatai. Pateikiami tame pačiame naršyklės lange.
- 47) Anketos „HAD klausimynas“ rezultatai. Pateikiami tame pačiame naršyklės lange.
- 48) Būklės įvertinimo rezultatai. . Pateikiama naujame naršyklės lange. Duomenis grupuoti pagal etapus. Virš vartotojo įvestų duomenų spausdinti būklės įvertinimą balais.
- 49) Visų įvestų tyrimų rezultatus turi būti galima išvesti kableliniu formatu į tekstinį failą arba į ekraną.

3.4.2. Nefunkciniai reikalavimai

Reikalavimai sistemos išvaizdai

- 50) Vartotojo sąsaja turi būti lengvai suprantama, maloni, netrukdyti darbui.
- 51) Informacijos įvedimo laukai turi būti išdėstyti patogiai.

Reikalavimai panaudojamumui

- 52) Privalomas teisingas lietuviškų simbolių apdorojimas.

Reikalavimai veikimo sąlygoms

53) Sistema turi turėti galimybę būti integruota su kitomis posistemėmis.

Reikalavimai saugumui

54) Turi būti prašoma operacijų patvirtinimo.

55) Duomenis turėtų būti leidžiama įvesti tik asmeniui, turinčiam prisijungimo vardą ir slaptažodį.

56) Turi būti nustatytas sesijos laikas – kiek ilgai po paskutinio vartotojo veiksmo sistemoje prisijungimas išliks aktyvus.

Reikalavimai sistemos priežiūrai

57) Turi būti galimybė keisti klasifikatorių reikšmes.

58) Turi būti galimybė administruoti vartotojus.

Kultūriniai – politiniai reikalavimai

Nėra.

Teisiniai reikalavimai

Nėra.

3.5. Sistemos architektūra

Sistemos architektūrą, remiantis *Rational Unified Process (RUP)*, įprasta vaizduoti keturiais skirtingais pjūviais (*view*):

1. Panaudojimo atvejų modelis (*Use Case View*). Jis skirtas parodyti pagrindinius funkcinius reikalavimus.
2. Loginis modelis (*Logical View*). Jo esmė – klasių diagramos, kurių pagrindu kuriamas duomenų modelis bei sistemos posistemės. Sistemos loginį vaizdavimą papildo sekų, bendradarbiavimo, būsenų bei veiklos diagramos.
3. Paskirstymo modelis (*Deployment View*) vaizduoja, kokie aparatūriniai komponentai sudarys kuriamą sistemą.
4. Komponentų (*Component View*) modelis skirtas parodyti, kokie programiniai komponentai sudarys kuriamą sistemą.

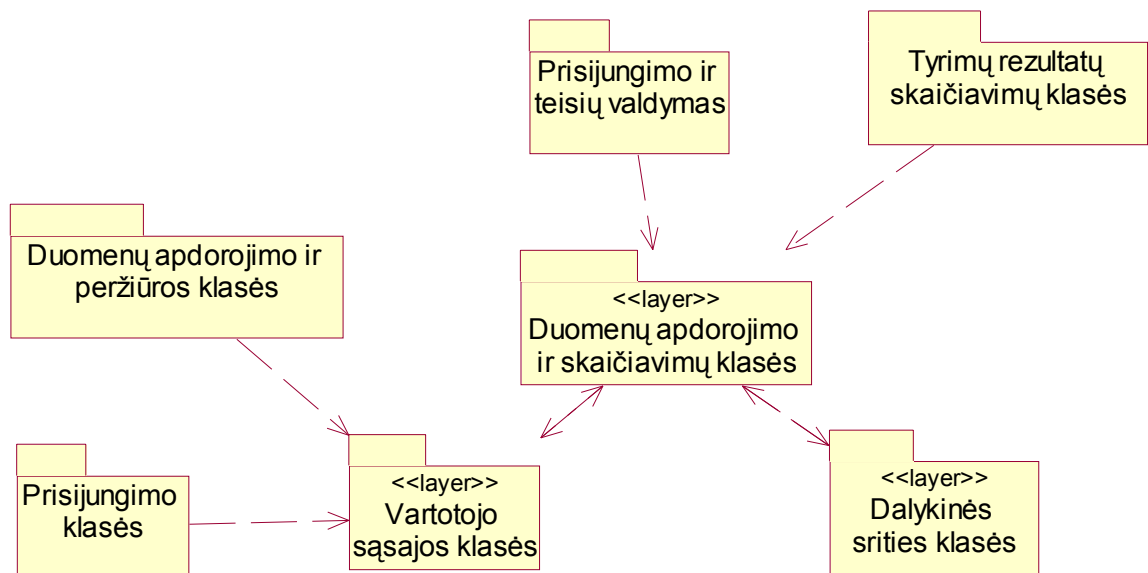
Drauge su šiais modeliais, jei reikia, pateikiamas duomenų modelis.

Toliau apžvelgsime tik tuos architektūrinius aspektus, kurie turi sąryšį su darbo problematika.

3.5.1. Klasių paketų diagrama

Siekiant aiškumo, loginio modelio klases galima suskirstyti sluoksniais:

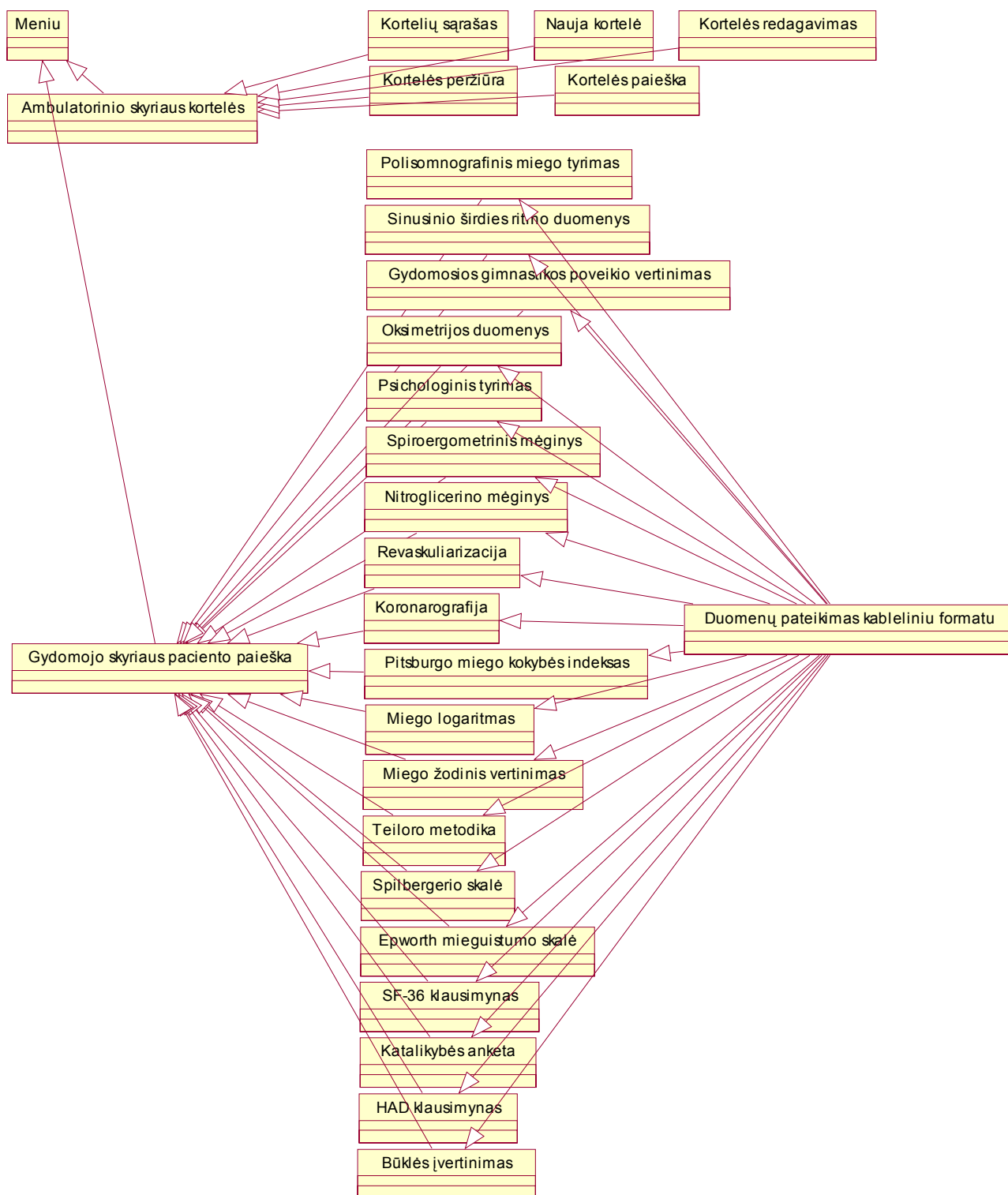
- Vartotojo sąsajos klasės – “ribinės” klasės (*interfaces*), turinčios ryšį su vartotoju. Tai – duomenų įvedimo, pateikimo, prisijungimo klasių paketai.
- Skaičiavimų klasės – valdymo. Jos atlieka duomenų valdymo, skaičiavimų funkcijas. Šiame sluoksnyje yra prisijungimo ir teisių valdymo klasių paketas bei tyrimų rezultatų skaičiavimų klasių paketas.
- Dalykinės srities klasės aprašo dalykinės srities objektus. Jų pagrindu projektuojama duomenų bazė.



Paveikslas 7. Klasių paketų diagrama.

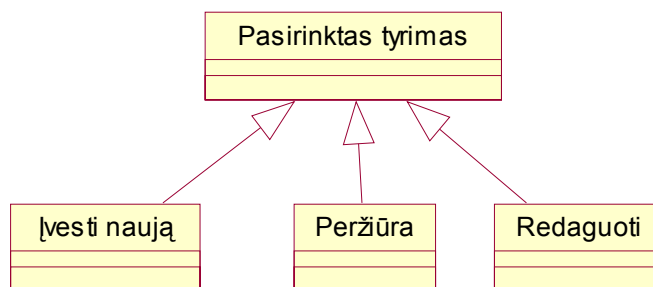
Toliau pateiksime labiausiai susijusias su šio darbo problematika klasių diagramas. Diagramos braižytos naudojant Rational Rose 2000e programinę įrangą.

3.5.2. Duomenų tvarkymo klasės



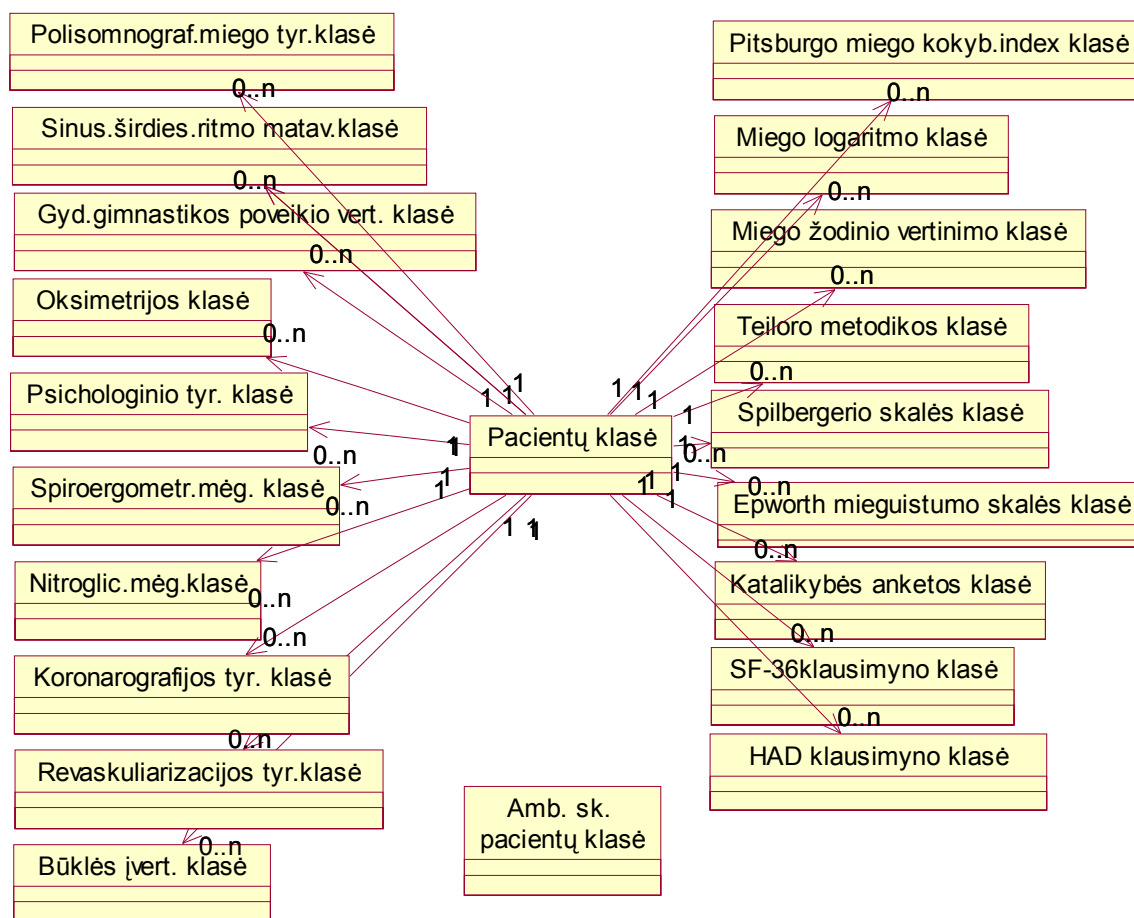
Paveikslas 8. Vartotojo sąsajos klasių diagrama. Duomenų apdorojimo klasės.

Kiekvieno tyrimo klasė jungia konkrečiam tyrimui pritaikytas įvedimo, peržiūros bei redagavimo klases.



Paveikslas 9. Vartotojo sąsajos klasių diagrama. Duomenų apdorojimo klasių detalizavimas.

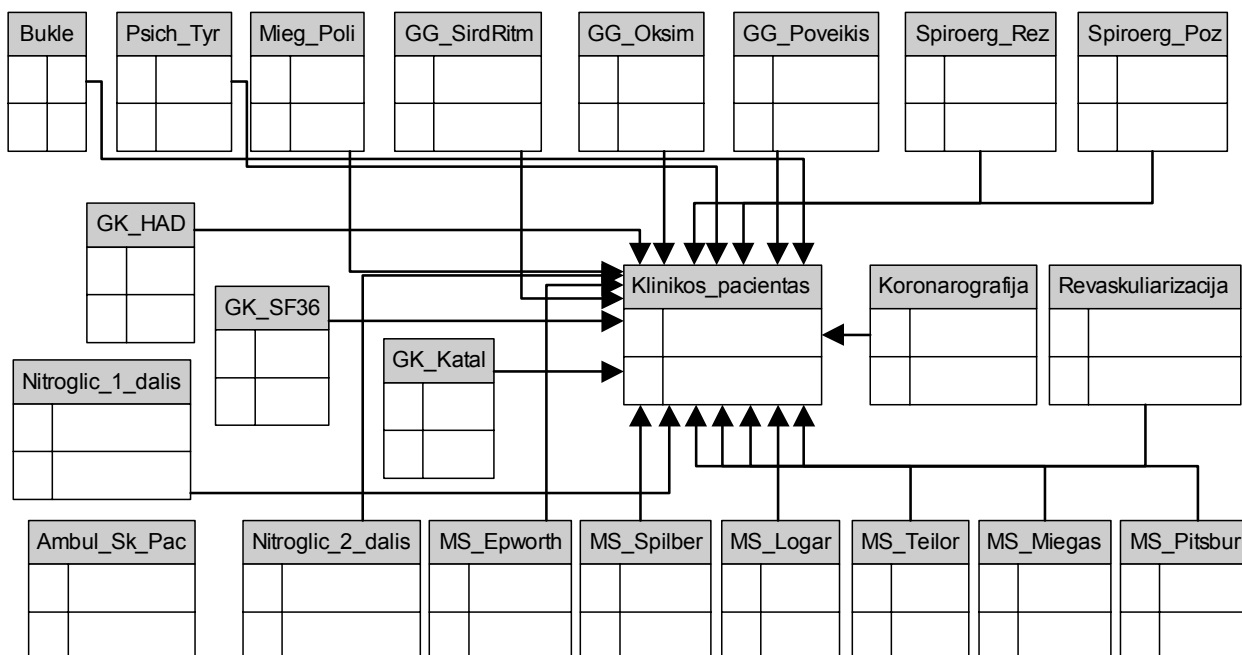
3.5.3. Dalykinės srities klasės



Paveikslas 10. Dalykinės srities klasių diagrama.

3.5.4. Duomenų modelis

Šiame skyriuje duomenų modelis pateikiamas eskizo pavidalu, nesigilinant į jame būsiančią informaciją. Šis modelis sukurtas pagal dalykinės srities klasių diagramą.¹

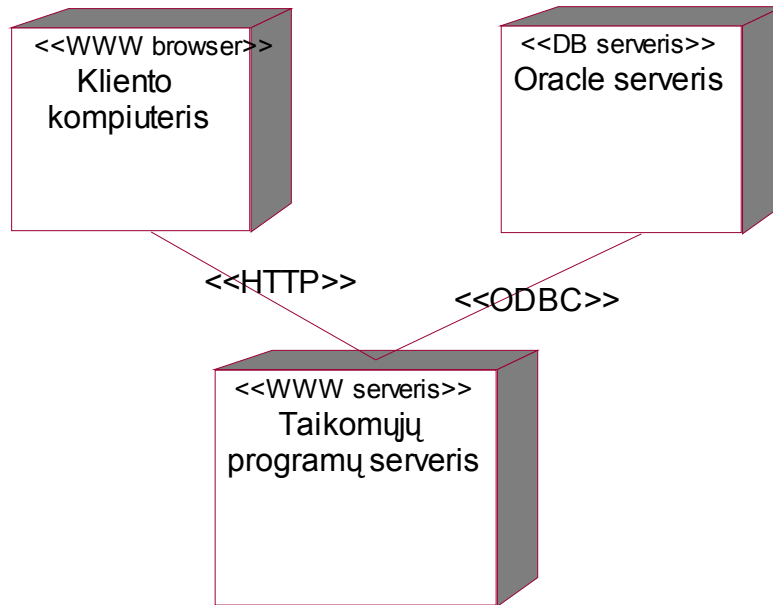


Paveikslas 11. Duomenų modelis.

3.5.5. Paskirstymo modelis

Šiame modelyje parodyta aparatūrinė sistemos pusė. Kliento kompiuteriu per naršyklę įvedami duomenys, matomi rezultatai. Taikomųjų programų serveryje atliekami skaičiavimai, patikrinimai ir t.t. Duomenų bazės serveryje saugoma pati duomenų bazė.

¹ Ši ir kitos duomenų diagramos braižytos su MS Visio 2002 programų paketu. Pagal jo duomenų bazės schemos braižymo šabloną, rodyklė, reiškianti ryšį „vienas su daug“, nukreipiama iš vaikinės lentelės į tėvinę.

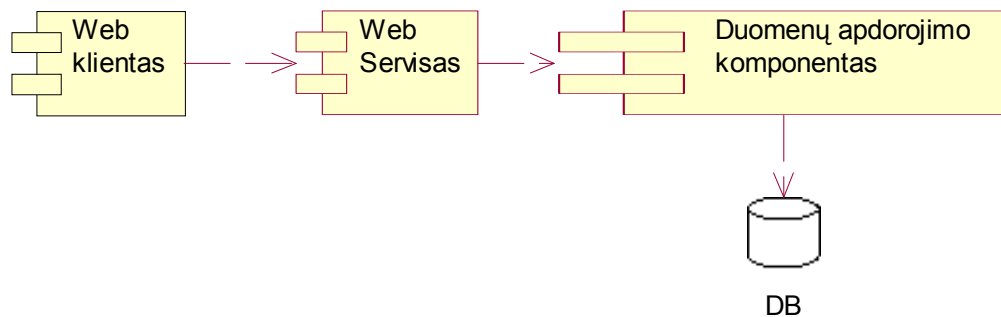


Paveikslas 12. Paskirstymo modelis.

3.5.6. Komponentų modelis

Šiame modelyje vaizduojami sistemos programiniai komponentai:

- Web klientas – vartotojo kompiuteryje esanti naršyklė
- Web servisas – taikomųjų programų serveryje vykdomos programos.
- Duomenų komponentas – atsakingas už duomenų išrinkimą iš duomenų bazės, jų įrašymą ir keitimą.



Paveikslas 13. Komponentų modelis.

4. Galimų duomenų struktūrų, užklausų ir jų greičio tyrimas

Šios, eksperimentinėje, dalyje bus išskirti ir aprašyti nagrinėjamoje sistemoje dažniausiai pasitaikančių užklausų tipai bei, atsižvelgiant į tyrimų etapiškumą, suprojektuoti galimi duomenų struktūrų variantai. Taip pat bus pateikti atlikto minėtų užklausų greičio matavimo eksperimento rezultatai bei nagrinėjamas užklausų sudėtingumas kiekvienos duomenų struktūros atveju.

4.1. Dažniausiai pasitaikančios užklausos

Pagal 3.4.1. skyrelyje išvardintus funkcinius reikalavimus bei 3.5.2. skyrelyje pateiktą vartotojo sąsajos klasių diagramą nesunku nustatyti dažniausiai sistemoje vartojamų užklausų tipus. Tai:

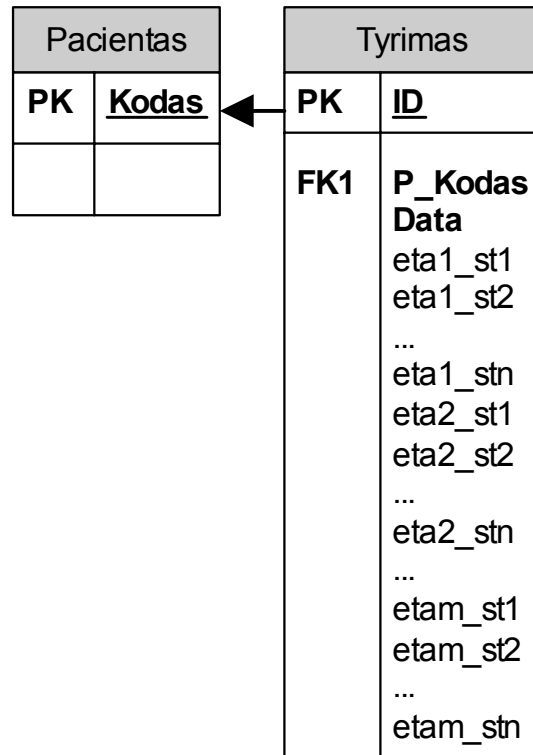
- 1) vieno tyrimo rezultatų įvedimas;
- 2) vieno tyrimo etapo rezultatų įvedimas;
- 3) vieno tyrimo rezultatų atnaujinimas;
- 4) vieno tyrimo etapo rezultatų atnaujinimas;
- 5) vieno tyrimo rezultatų šalinimas;
- 6) vieno tyrimo rezultatų išrinkimas;
- 7) vieno tyrimo etapo rezultatų išrinkimas.

Šios užklausos atliekamos pagal paieškos metu surasto paciento kodą. Šios užklausos bus tiriamos visų vėliau aprašytų duomenų struktūrų atvejais.

4.2. Galimi duomenų struktūrų variantai

Kaip buvo minėta anksčiau, vieno tyrimo rezultatus gali sudaryti iki 200-300 reikšmių. Tačiau, kai kurie tyrimai atliekami etapais, tad atitinkamos lentelės laukų pavadinimai, duomenų tipai periodiškai kartojasi po keletą, keliolika ar keliasdešimt kartų. Iš čia išplaukia galimybė tyrimą nusakančią duomenų bazės lentelę skaidyti į keletą mažesnių: tėvinėje lentelėje liktų tyrimo identifikacinė ir kita nesikartojanti informacija, o vaikinės atitiktų tyrimo etapus. Ši struktūra turėtų būti normalizuota iki trečiosios normalinės formos (žr. 2.1.1. ir 2.1.2. skyrelius). Kita alternatyva – vadovaujantis denormalizavimo taisyklėmis, aprašytomis 2.1.3. ir 2.1.4. skyreliuose, vieną tyrimą atitinkantį lentelės įrašą skaidyti į keletą tos pačios lentelės įrašų (jų kiekis atitiktų tyrimo etapų kiekį). Šiuo atveju identifikacinė ir kita nesikartojanti tyrimo informacija turėtų būti pakartojama kiekviename tyrimo etapą atitinkančiame įrašė.

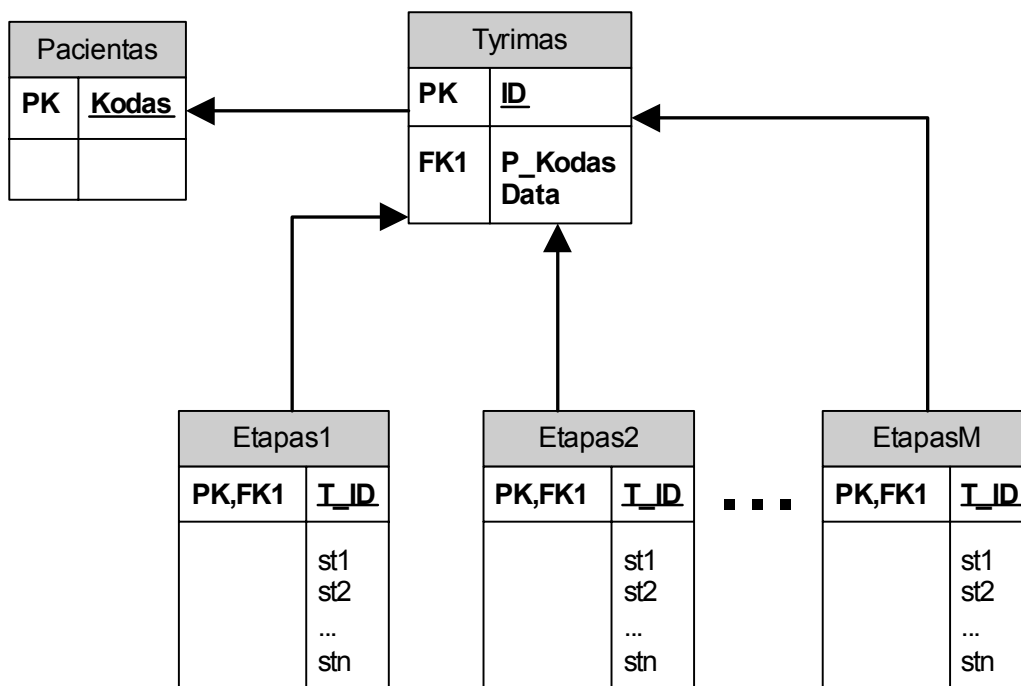
Aptartų duomenų struktūrų schemas pateiktos 14, 15 ir 16 paveiksluose.²



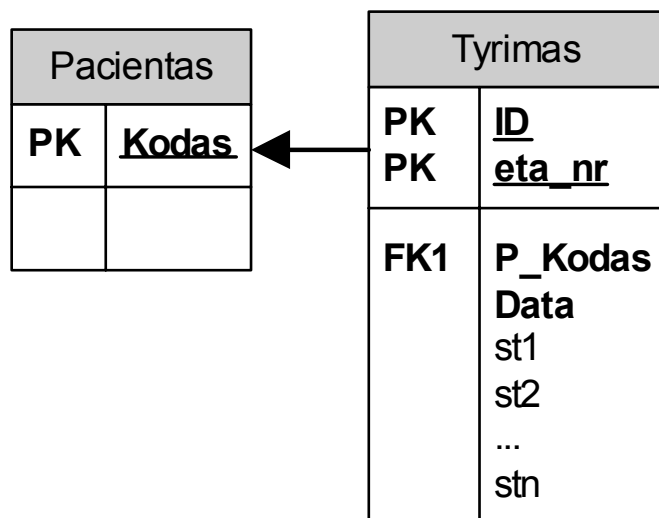
Paveikslas 14. Pradinė duomenų struktūra.

² Santrumpos:

- lauke P_Kodas saugoma paciento kodo reikšmė;
- T_ID – tyrimo identifikacinio numerio reikšmė;
- pradinėje schemoje: eta1_st1 – pirmojo etapo pirmoji reikšmė, etam_stn – m-tojo etapo n-oji reikšmė ir pan.;
- normalizuotoje ir denormalizuotoje schemose: st1 – pirmoji tyrimo etapo reikšmė, stn – n-oji tyrimo etapo reikšmė ir pan.;
- denormalizuotoje schemoje: eta_nr – etapo numerio reikšmė.



Paveikslas 15. Normalizuota duomenų struktūra.



Paveikslas 16. Denormalizuota duomenų struktūra.

4.3. Užklausų greičio matavimo tyrimas

Pagal antrame šio darbo skyriuje atliktą literatūros apžvalgą galime daryti išvadas, jog ilgų įrašų apdorojimas dėl galimo jų dalių išsibarstymo po skirtingus duomenų blokus trunka ilgiau,

nei trumpų. Užklausų vykdyme svarbus aspektas – rezultatams išskiriamo (*fetch*) buferio dydis. Ilgiems įrašams atminties rezervuoti reikia daugiau, nei trumpiems. Tokiu atveju lentelių suskaidymas atrodo tikslingas. Taip pat, remiantis minėta apžvalga, normalizuota duomenų struktūra yra patikimesnė už denormalizuotą. Tačiau ryšiai tarp lentelių yra darbą lėtinantis veiksnys.

Remiantis šiais faktais, norint tiksliau įvertinti kiekvienos nagrinėjamos duomenų struktūros privalumus bei trūkumus nagrinėjamos dalykinės srities kontekste, buvo atliktas užklausų greičio matavimo eksperimentas.

Eksperimento užduotį galima suformuluoti taip: tyrimą charakterizuoja identifikatorius, paciento kodas, data bei 270 penkiaženklis skaičiaus tipo reikšmių. Datos ir paciento kodo pora yra unikali. Toks duomenų tipas kuriamoje sistemoje užfiksuotas kaip dažniausiai pasitaikantis.

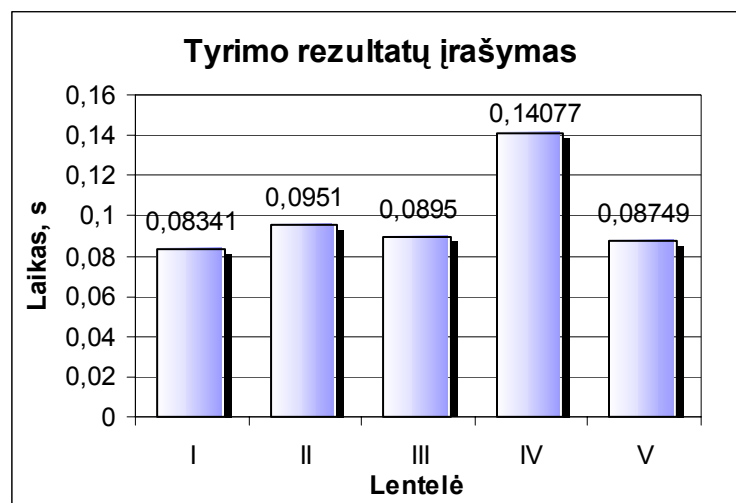
Eksperimentui atlikti sukurtos tokios duomenų struktūros:

1. Vieną tyrimo tipą nusako viena lentelė. Vieną konkretų tyrimą atitinka vienas lentelės įrašas (pagal 14 pav.).
2. Tarkime, tyrimas atliekamas 9 etapais ir kiekvieno jų metu gaunama po 30 įvertinimų. Vieną tyrimo tipą nusako viena tėvinė lentelė, kurioje saugoma tyrimų identifikavimo informacija. Kiekvieno tyrimo etapo rezultatai rašomi į atskiras lenteles, kurių iš viso yra 9 (pagal 15 pav.).
3. Tarkime, tyrimas atliekamas 9 etapais ir kiekvieno jų metu gaunama po 30 įvertinimų. Vieną tyrimo tipą nusako viena lentelė, kurios vienas įrašas atitinka vieną tyrimo etapą. Tokiu būdu vienam konkrečiam tyrimui skiriami 9 įrašai. Siekiant išsaugoti unikalumo apribojimus, pirminis raktas papildomas etapo numeriu (pagal 16 pav.).
4. Tarkime, tyrimas atliekamas 30 etapų ir kiekvieno jų metu gaunama po 9 įvertinimus. Vieną tyrimo tipą nusako viena tėvinė lentelė, kurioje saugoma tyrimų identifikavimo informacija. Kiekvieno tyrimo etapo rezultatai rašomi į atskiras lenteles, kurių iš viso yra 30 (pagal 15 pav.).
5. Tarkime, tyrimas atliekamas 30 etapų ir kiekvieno jų metu gaunama po 9 įvertinimus. Vieną tyrimo tipą nusako viena lentelė, kurios vienas įrašas atitinka vieną tyrimo etapą. Tokiu būdu vienam konkrečiam tyrimui skiriami 9 įrašai. Siekiant išsaugoti unikalumo apribojimus, pirminis raktas papildomas etapo numeriu (pagal 16 pav.).

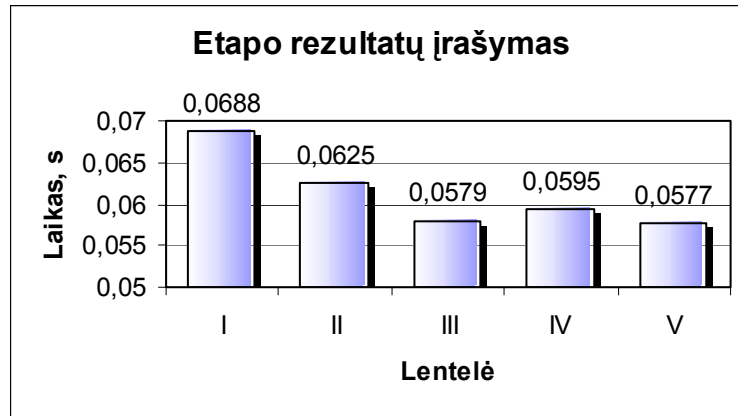
Tyrimas buvo atliekamas vykdant ankstesniame skyriuje išvardintas užklausas. Kadangi SQL sakiniai, esantys bloke, siunčiamame iš kliento kompiuterio, siunčiami ir vykdomi po vieną, tai lemia didelį vykdymo laiką ir neatspindi realaus sistemos vartojimo. Todėl įrašymo, atnaujinimo, šalinimo užklausos buvo vykdomos procedūrų pagalba. Išrinkimo užklausos buvo siunčiamos iš kliento kompiuterio. Vykdamas procedūrų pagalba būtų prirėkę duomenų formatavimo ir pateikimo koku nors pavidalu.

Kadangi užklausos apdorojimo greičiui turi įtakos interneto kanalo pralaidumas, jo ir serverio apkrautumas, kreipiniai į kiekvieno tipo procedūras ir užklausas buvo atliekami po dešimt ir daugiau kartų ir imamas laikų vidurkis. Eksperimentui atlikti buvo pasirinkta aplinka, panaši į vartotojo: geografiškai nuo serverio nutolęs kompiuteris, interneto greitis – 64kbit/s.

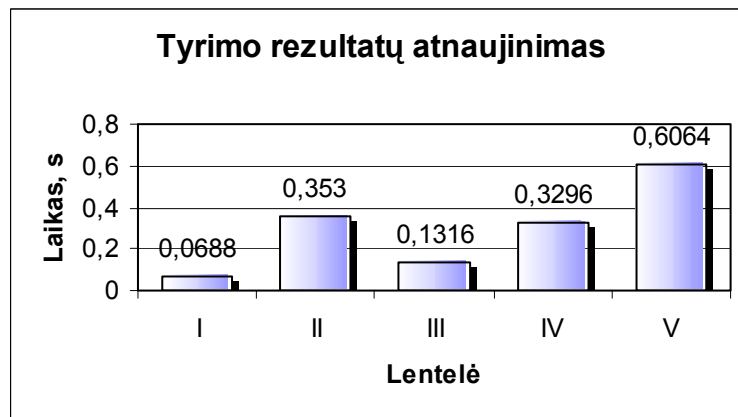
Tyrimo rezultatai vaizduojami 17-23 paveiksluose. Diagramose esantys romėniški skaičiai atitinka šiame skyriuje išvardintų sukurtų duomenų struktūrų numerius.



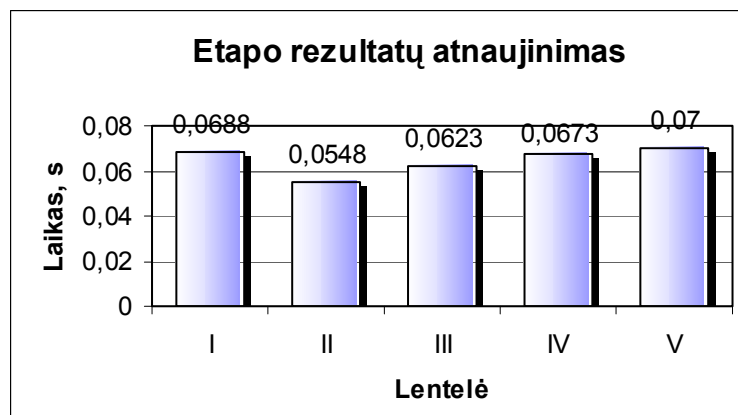
Paveikslas 17. Tyrimo rezultatų įrašymo greičio priklausomybė nuo duomenų struktūros.



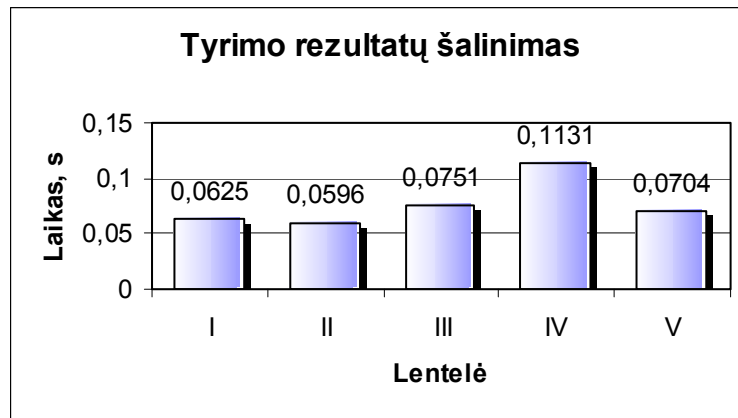
Paveikslas 18. Vieno tyrimo etapo rezultatų įrašymo greičio priklausomybė nuo duomenų struktūros.



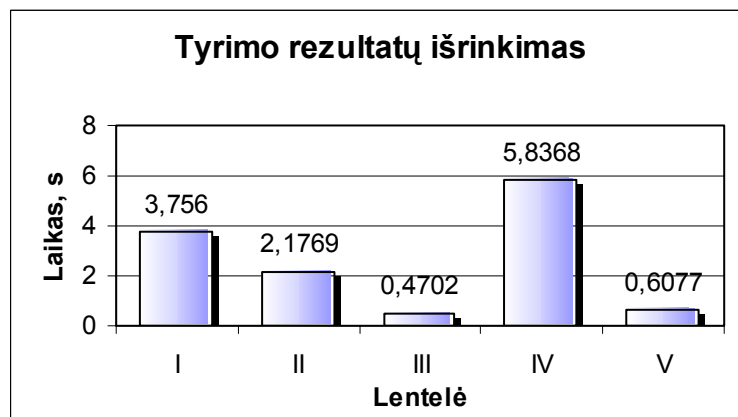
Paveikslas 19. Tyrimo rezultatų atnaujinimo greičio priklausomybė nuo duomenų struktūros.



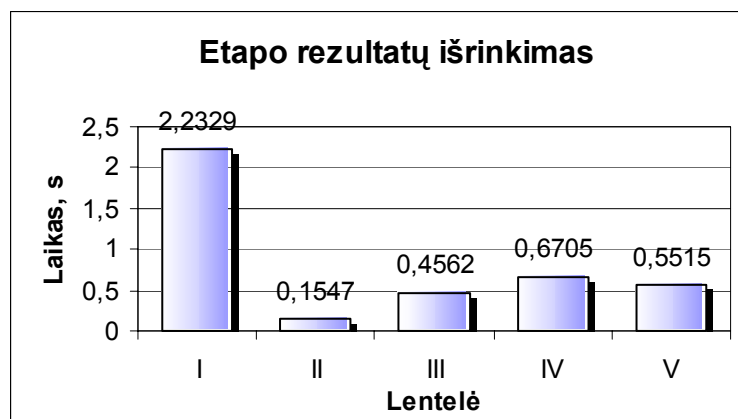
Paveikslas 20. Vieno tyrimo etapo rezultatų atnaujinimo greičio priklausomybė nuo duomenų struktūros.



Paveikslas 21. Tyrimo rezultatų šalinimo greičio priklausomybė nuo duomenų struktūros.



Paveikslas 22. Tyrimo rezultatų išrinkimo greičio priklausomybė nuo duomenų struktūros.



Paveikslas 23. Vieno tyrimo etapo rezultatų išrinkimo greičio priklausomybė nuo duomenų struktūros.

4.4. Užklausų panaudojamumo įvertinimas

Keičiantis duomenų struktūrai iš esmės keičiasi ir pačių užklausų sudėtingumas bei taikymas. Šiame skyriuje panagrinėsime užklausų panaudojamumo kitimą kiekvienos, 4.3. skyriuje aprašytos, duomenų struktūros atveju.

Pirmuoju atveju užklausa tiesinė. Įrašymo, atnaujinimo, šalinimo metu keičiamas tik vienas įrašas. Jei atliekamas išrinkimas pagal pirminį arba unikalų raktą, gražinamas taip pat tik vienas įrašas. Esminė problema programuotojams – užklausų ilgis. Programavimo metu tokios užklausa paprastai formuojamos dinamiškai simbolių eilučių pavidalu. Tačiau PL/SQL kalboje simbolių eilutės maksimalus galimas ilgis yra 4000 simbolių. Jei užklausa formavimo metu ši riba pasiekama ir bandoma viršyti, iškyla vykdymo klaida ir programos darbas nutrūksta. Kompiliavimo metu neįmanoma nustatyti tokios klaidos galimybės. Taip pat išlieka rizika, jog sistemos tobulinimo ar plėtros metu pakeitus lentelės laukų duomenų tipus (padidinus jų maksimalų ilgį), programos veikimo prognozuojamumas stipriai sumažės arba ji taps netgi nebeįnaudojama.

Antruoju ir ketvirtuoju atveju rizika, jog užklausa dinaminio formavimo metu bus suformuota per ilga, žymiai sumažėja, tačiau, priklausomai nuo vaikinių lentelių kiekio, padidėja ir pačių užklausų kiekis. Tai aktualu įrašymo, keitimo, šalinimo metu. Be to, kiekvienoje šių užklausų atsiranda ryšys su tėvine lentele. Išrinkimo atveju užtenka vienos užklausa, tačiau joje esančių ryšių skaičius išauga proporcingai lentelių skaičiui. Taip pat, norint iš rezultatų pašalinti besikartojančią informaciją, užklausoje reikia išvardinti norimus gauti laukus arba jau išrinktus duomenis papildomai apdoroti.

Trečiuoju ir penktuoju atveju, skirtingai negu antruoju ir ketvirtuoju, papildomo ryšio su tėvine lentele nebus, tačiau atsiras būtinybė kopijuoti tyrimo identifikacinę ir kitą neperiodinę informaciją. Kuo daugiau etapų, tuo daugiau perteklinės informacijos. Taip pat, skirtingai nei kitais atvejais, duomenų išrinkimo metu duomenys gaunami ne vienoje eilutėje, bet atskirose.

Apibendrinant, duomenų struktūrų nulemtus užklausų privalumus ir trūkumus galima aprašyti lentele. Žvaigždučių kiekis rodo pranašumą arba trūkumą kitimą. Kuo mažiau žvaigždučių, tuo duomenų struktūra pranašesnė pagal konkretų kriterijų.

Lentelė 2. Užklausų panaudojamumo kriterijai. Užklausų panaudojamumo priklausomybė nuo duomenų struktūrų.

<i>Vertinimo kriterijus</i>	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>
„Per ilgos užklausos“ rizika	***	**	**	*	*
Ryšių su kitomis lentelėmis skaičius	*	**	*	***	*
Užklausų kiekis vienam tyrimui atnaujinti / įterpti / trinti	*	**	**	***	***
Išrinktų tyrimo rezultatų eilučių kiekis	*	*	**	*	***
Išrinktų tyrimo rezultatų papildomo apdorojimo poreikis (išrinkimo užklausos „nepatogumas“)	*	***	**	***	**
Tinkamumas vieno etapo rezultatams apdoroti	***	*	**	*	**
Perteklinės informacijos kiekis	*	**	***	**	***

5. Išvados

1. Pagal 17-21 paveikslus matome, jog duomenų tvarkymo veiksmai (įrašymas, atnaujinimas, šalinimas), nepriklausomai nuo duomenų struktūros, atliekami greičiau, negu per sekundę, tad vartotojas skirtumų greičiausiai nepajus.
2. 22 paveiksle matome, jog tyrimo duomenų išrinkimo greitis ženkliai skiriasi. Kadangi vartotojui perduotų duomenų kiekis visais atvejais yra panašus, pagal 2.2.2. skyrių galime teigti, jog tai lemia atminties kiekis, reikalingas išskirti vienu metu. Kuo „platesnis“ gaunamas rezultatas – sudėtinė duomenų eilutė, – tuo ilgiau vykdoma užklausa. Kitas veiksnys, turintis įtakos duomenų išrinkimo greičiui – ryšių tarp lentelių kiekis. Ši priklausomybė paveiksle aiškiai matoma. Pagal 2.3.2. skyrių, yra galimybė šią situaciją gerinti klasterių pagalba, tačiau reikia atsižvelgti į tai, jog toks veiksmas turės neigiamos įtakos duomenų tvarkymo spartai.
3. Iš greičio matavimo eksperimento išplaukia, jog nors atnaujinimas denormalizuotose struktūrose vyksta lėčiau, jos yra žymiai pranašesnės išrinkimo atveju. Pagal antrąją lentelę, ypač jei tyrimas atliekamas mažiau negu dešimčia etapų, denormalizuotas struktūras galima būtų įvertinti vidutiniškai.
4. Daugiaatributes lenteles užklausų sudėtingumo aspektu galima būtų vertinti palankiai, tačiau „per ilgos užklausos“ rizika ir jos vengimas, priklausomai nuo stulpelių kiekio ir maksimalaus duomenų ilgio, gali sukelti nemažų nepatogumų, kodo painumo ir pan. Be to, jos visai netinkamos valdyti ir išrinkti atskirų tyrimo etapų duomenims.
5. Žvelgiant į ankstesnes išvadas, mažiausiai kritinių problemų kelia denormalizuota duomenų struktūra. Pagal 2.3.3. ir 2.3.4. skyrius, duomenų atnaujinimo spartą būtų galima pagreitinti naudojant eilučių identifikatorius bei manipuliuojant segmento parametru PCTFREE. Indeksavimas šiuo, kaip ir visais kitais aptartais, atveju yra būtinas.

6. Literatūra

- [1] Speelpenning J., Daux P., Gallus J. **Data Modeling and Relational Database Design, Volume 1, Student Guide.** Oracle Corporation, 2001. – 320p.
- [2] Papaj R., Burleson D.K. **Oracle Databases on the Web.** The Coriolis Group, 1997. ISBN: 1576100995.
- [3] Antoshenkov G., Ziauddin M. **Query processing and optimization in Oracle Rdb.** The VLDB Journal (1996) 5:229-237.
- [4] Schumacher R. **It's The Little Things That Count.** Oracle Technology Network [žiūrėta 2004.05.20]. Prieiga per internetą:
http://otn.oracle.com/oramag/webcolumns/2003/techarticles/schumacher_skipscan.html
- [5] Aldridge D. **A Practical Guide to Data Warehousing in Oracle, Part 2.** Database Journal. 2004 sausis [žiūrėta 2004.05.20]. Prieiga per internetą:
<http://databasejournal.com/features/oracle/article.php/3304791>
- [6] Jacobs K. **Query Optimization.** Oracle Technology Network [žiūrėta 2004.05.20]. Prieiga per internetą:
<http://otn.oracle.com/oramag/oracle/02-jul/o42dba.html>
- [7] Wyllys R.E. **Steps in Normalization.** The University of Texas at Austin Graduate School of Library and Information Science [žiūrėta 2004.05.19]. Prieiga per internetą:
<http://www.gslis.utexas.edu/~l384k11w/normstep.html#Section%201.%20Introduction>
- [8] **2. Data Blocks, Extents, and Segments.** Oracle 9i Database Concepts, Release 2 (9.2) [žiūrėta 2004.05.19]. Prieiga per internetą:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96524/c03block.htm
- [9] **10. Schema Objects.** Oracle 9i Database Concepts, Release 2 (9.2) [žiūrėta 2004.05.19]. Prieiga per internetą:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96524/c11schem.htm
- [10] **14. SQL, PL/SQL, and Java.** Oracle 9i Database Concepts, Release 2 (9.2) [žiūrėta 2004.05.19]. Prieiga per internetą:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96524/c16sqlpl.htm
- [11] **18. Managing Clusters.** Oracle 9i Database Administrator's Guide, Release 2 (9.2) [žiūrėta 2004.05.19]. Prieiga per internetą:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96521/clustrs.htm

[12] **12. Native Datatypes.** Oracle 9i Database Concepts, Release 2 (9.2) [žiūrēta 2004.05.19].

Prieiga per internetą:

http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96524/c13datyp.htm#6538

[13] **14. Managing Space for Schema Objects.** Oracle 9i Database Administrator's Guide,

Release 2 (9.2) [žiūrēta 2004.05.19]. Prieiga per internetą:

http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96521/schema.htm#949