

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA



KAROLIS GRIGAITIS

**VERSLO DUOMENŲ INTELEKTUALIOS ANALIZĖS
DUOMENŲ KUBŲ KŪRIMO IR SAUGOJIMO
MODELIŲ TYRIMAS MS SQL SERVER 2005 BI
MANAGEMENT STUDIO PRIEMONĖMIS**

Magistro baigiamasis darbas

Vadovas:
doc. V. Šakys

KAUNAS, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA



VERSLO DUOMENŲ INTELEKTUALIOS ANALIZĖS
DUOMENŲ KUBŲ KŪRIMO IR SAUGOJIMO
MODELIŲ TYRIMAS MS SQL SERVER 2005 BI
MANAGEMENT STUDIO PRIEMONĖMIS

Informacinių sistemų inžinerijos magistro baigiamasis darbas
Studijų programa M405DF11

Recenzentas
prof. V. Jusas
2009 01

Vadovas:
doc. V. Šakys
2009 01

Atliko
IFM-3/4 gr. stud.
K. Grigaitis
2009 01 09

KAUNAS, 2009

TURINYS

SANTRAUKA.....	9
SUMMARY	10
1. ANALITINĖ DALIS.....	13
1.1. Analizės tikslas	13
1.2. Analizės metodai.....	13
1.3. Duomenų kubo analizė.....	13
1.4. Vartotojų analizė	23
1.4.1. Vartotojų aibė, tipai ir savybės	23
1.4.2. Vartotojų tikslai ir problemos	23
1.5. Esamų sprendimų analizė	24
1.6. Microsoft SQL Server 2005 apžvalga.....	26
1.6.1. Architektūra	27
1.6.2. Paslaugos (Services).....	28
1.6.3. Reikalavimai sistemai	30
1.7. ProClarity Desktop Professional 6.3 apžvalga.....	31
1.7.1. Reikalavimai sistemai	32
1.8. Siekiamas sprendimas	32
1.9. Analitinės dalies išvados	33
2. SISTEMOS PROJEKTAVIMO ESMINIAI SPRENDIMAI	34
2.1. Projektinės dalies tikslas.....	34
2.2. Bendra sistemos architektūra.....	34
2.3. Panaudojimo atvejai	36
2.4. Programinės įrangos kūrimo bei darbo pagrindiniai algoritmai.....	40
2.5. Reikalavimai sistemai.....	46
2.5.1. Funkciniai reikalavimai	46
2.5.2. Nefunkciniai reikalavimai	47
2.6. Esybių klasių diagrama.....	47
2.7. Sistemos projektavimo išvados	48
3. SISTEMOS REALIZACIJOS APRAŠAS	49
3.1. Sistemos realizacijos dalies tikslas.....	49
3.2. Duomenų saugyklos sukūrimas	49
3.3. Duomenų kubų realizacija	57
3.3.1. Pirmasis sukurtas duomenų kubas.....	57
3.3.2. Antrasis sukurtas duomenų kubas	60
3.3.3. Trečiasis sukurtas duomenų kubas	62

3.3.4.	<i>Ketvirtas sukurtas duomenų kubas</i>	65
3.3.5.	<i>Penktasis sukurtas duomenų kubas</i>	65
3.3.6.	<i>Šeštasis sukurtas duomenų kubas</i>	66
3.3.7.	<i>Septintasis sukurtas duomenų kubas</i>	68
3.3.8.	<i>Aštuntasis sukurtas duomenų kubas</i>	70
3.3.9.	<i>Devintasis sukurtas duomenų kubas</i>	71
3.3.10.	<i>Dešimtas sukurtas duomenų kubas</i>	72
3.4.	Testavimas	74
3.5.	Sistemos realizacijos išvados	75
4.	KTU DS INTELEKTUALI ANALIZĖ	76
4.1.	Intelektualios analizės tikslas.....	76
4.2.	Intelektualios analizės rezultatai	76
4.3.	Intelektualios analizės išvados	86
	IŠVADOS.....	87
	LITERATŪRA	89
	PRIEDAI	91

PAVEIKSLŲ TURINYS

1.1 pav. Importo kubo pavyzdys.....	14
1.2 pav. <i>Calendar Time</i> hierarchijos naudojimas <i>Time</i> dimensijoje.....	15
1.3 pav. Matų apibendrinimas naudojant skirtingas dimensijas.....	16
1.4 pav. Matų, atributų ir hierarchijos vietos nurodymas	16
1.5 pav. <i>ImportsFactTable</i> faktų eilutės kubo narvelis.....	17
1.6 pav. <i>AdventureWorksDW</i> duomenų bazės dalies pavyzdys	18
1.7 pav. Lentelės, reikalingos <i>Product</i> dimensijai aprašyti.....	19
1.8 pav. <i>MSND</i> svetainėje prieinamos legalios programinės įrangos sąrašas	26
1.9 pav. <i>SQL</i> serverio architektūra [15].....	27
1.10 pav. <i>Microsoft SQL Server</i> komponentai [16].....	30
1.11 pav. Paleistos <i>ProCrarity Desktop Professional 6.3</i> programos pirminis langas.....	31
2.1 pav. Projektuojamos sistemos įrangos diagrama	34
2.2 pav. Projektuojamos sistemos komponentų diagrama	35
2.3 pav. Principinė tinklo schema.....	36
2.4 pav. <i>SQL Server</i> administratoriaus panaudojimo atvejų diagrama.....	37
2.5 pav. <i>DB</i> Administratoriaus panaudojimo atvejų diagrama.....	38
2.6 pav. Analitiko panaudojimo atvejų diagrama.....	39
2.7 pav. Duomenų kubo kūrimo veiklos diagrama.....	41
2.8 pav. Duomenų kubo kūrimo sekų diagrama.....	43
2.9 pav. Sukinio kūrimo <i>Excel 2007</i> programa veiklos diagrama.....	43
2.10 pav. Sukinio kūrimo <i>Excel 2007</i> programa sekų diagrama	44
2.11 pav. Grafiko kūrimo <i>ProClarity</i> programa veiklos diagrama	45
2.12 pav. Grafiko kūrimo <i>ProClarity</i> programa sekų diagrama.....	46
2.13 pav. Esybių diagrama	48
3.1 pav. <i>Microsoft Access 2007</i> langas importavus palikiminę duomenų bazę	50
3.2 pav. <i>Microsoft SQL Server Management Studio</i> langas baigus importavimą.....	51
3.3 pav. Sutvarkyta duomenų bazė	55
3.4 pav. Sukurta duomenų bazės diagrama	55
3.5 pav. Pirmojo duomenų kubo vaizdinio schema.....	57
3.6 pav. Sukurto pirmojo <i>OLAP</i> kubo struktūra.....	58
3.7 pav. Sukinio peržiūra <i>Business Intelligence</i> priemonėmis	59
3.8 pav. Antrojo kubo dimensijos <i>DimStudentas</i> vaizdinys.....	60
3.9 pav. Antrojo kubo dimensijos <i>DimModulis</i> vaizdinys	60

3.10 pav. Antrojo kubo faktinės lentelės <i>FaktIndPlanas</i> vaizdinys	61
3.11 pav. Antrojo sukurto kubo struktūra	61
3.12 pav. Trečiojo kubo dimensijos vaizdiniai: iš eilės - <i>1DimGrupes</i> , <i>1DimStudentai</i> ir <i>1DimModuliai</i>	63
3.13 pav. Trečiojo kubo faktinės lentelės <i>1FaktStudentai</i> vaizdinys.....	63
3.14 pav. Trečiojo kubo faktinės lentelės <i>1FaktIndPlanas</i> vaizdinys	63
3.15 pav. Trečiojo sukurto kubo struktūra	64
3.16 pav. Ketvirtojo sukurto kubo struktūra.....	65
3.17 pav. Penktojo sukurto kubo struktūra.....	66
3.18 pav. Šeštojo sukurto kubo vaizdinys	67
3.19 pav. Šeštojo sukurto kubo struktūra.....	67
3.20 pav. Septintojo kubo dimensijos vaizdiniai: iš eilės - <i>2DimDestytojas</i> , <i>2DimFakultetas</i> , <i>2DimPareigos</i> , <i>2DimModulis</i> ir <i>2DimStudentas</i>	68
3.21 pav. Septintojo kubo faktinės lentelės <i>2FaktIndPlanas</i> vaizdinys	69
3.22 pav. Septintojo sukurto kubo struktūra.....	69
3.23 pav. Aštuntojo sukurto kubo vaizdinys	70
3.24 pav. Aštuntojo sukurto kubo struktūra	70
3.25 pav. Devintojo sukurto kubo vaizdinys	71
3.26 pav. Devintojo sukurto kubo struktūra.....	72
3.27 pav. Dešimtojo sukurto kubo vaizdinys	72
3.28 pav. Dešimtojo sukurto kubo struktūra	73
3.29 pav. <i>Deployment</i> proceso vykdymo laiko priklausomybė nuo techninės įrangos pajėgumo	75
4.1 pav. Vyrų bei moterų pažymių vidurkiai bakalauro bei magistratūros studijose pagal fakultetus	77
4.2 pav. Bakalauro bei magistro studijų vidurkiai fundamentaliųjų mokslų fakulteto katedrose	78
4.3 pav. Matematinės sistemos katedros modulių pažymių vidurkiai	79
4.4 pav. Daugiausiai procentaliai neigiamų pažymių parašę dėstytojai universitete	80
4.5 pav. Informacijos sistemų katedros dėstytojų rašomų pažymių kiekiai.....	81
4.6 pav. <i>Performance Map</i> vaizdavimo parametrų nustatymo langas.....	82
4.7 pav. Vyrų bei moterų pažymių vidurkiai bakalauro bei magistratūros studijose pagal fakultetus <i>Performance Map ProClarity Analytics</i> diagrama	82
4.8 pav. Studentų vidurkiai pagal jų lytį įvairiose katedrose diagrama	83

4.9 pav. Studentų bendras vidurkis įvairiose katedrose pagal studijų pirmą bei antrą kursą diagrama	84
4.10 pav. Išbrauktų studentų kiekiai įvairiose katedrose	85

LENTELIŲ TURINYS

<i>Lentelė 1.1 Serverių parametrų palyginimas</i>	24
<i>Lentelė 3.1 Universiteto darbuotojų pareigų lentelė</i>	52
<i>Lentelė 3.2 Duomenų bazės struktūra ir aprašymas.....</i>	56

Grigaitis K. Verslo duomenų intelektualios analizės duomenų kubų kūrimo ir saugojimo modelių tyrimas MS SQL Server 2005 BI Management Studio priemonėmis. Vadovas: V. Šakys. – Kaunas: Kauno Technologijos Universitetas, Informatikos fakultetas, Informacijos sistemų katedra, 2008. – 90 p.

SANTRAUKA

Šiuolaikiniame informacijos gausos amžiuje susiduriama su milžinišku duomenų kiekiu, kurį sunku tinkamai apdoroti, įvertinti bei analizuoti. Šiame magistriniame darbe nagrinėjama universiteto duomenų valdymo problema bei ieškoma sprendimų, kurie leistų pagerinti studijų proceso administravimą fakulteto ir rektoriato atsakingiems darbuotojams.

Pirmojoje darbo dalyje analizuojama duomenų kubo sąvoka, apžvelgiami dažniausiai naudojami intelektualios duomenų analizės kūrimo programinės įrangos paketai. Detaliai išnagrinėjamas *Microsoft SQL Server 2005* programinis paketas bei apžvelgiama *ProClarity Desktop Professional 6.3* programa. Antrojoje darbo dalyje buvo aptariama universiteto duomenų analizės sistemos architektūra, suprojektuojami pagrindiniai komponentai, pasiruošiama sistemos realizacijai. Trečiojoje darbo dalyje atliekama sistemos realizacija, kurios metu KTU palikiminės duomenų bazės pagrindu sukuriama duomenų saugykla bei įvairių rūšių kubai. Ketvirtojoje darbo dalyje *Excel 2007* programa braižomi įvairūs duomenų sukiniai bei *ProClarity Analytics* indikatorinės diagramos, kurių pagalba įvairiais pjūviais atlikta universiteto duomenų analizė.

KTU DS duomenų analizės sistema sukuria naujas galimybes universiteto veiklos gerinimo procese, analizuojant modulių, studentų, dėstytojų, studijų programų bei kitą informaciją.

Grigaitis K. Research of business data intelligence analysis and saving models using MS SQL Server 2005 BI Management Studio tools. Supervisor: V. Šakys. – Kaunas: Kaunas University of Technology, Faculty of Informatics, Department of Information Systems, 2008. – 90 p.

SUMMARY

In the century of fast growing information, we encounter with enormous amount of data, which is difficult to manage, evaluate and analyze. This Master's work analyses the problem of university data management and finds solution how to improve administration of studies process for employees of the faculty and rectorial.

First part analysis the concept of data cube, reviews the most usable software of business intelligence solutions. Describes *Microsoft SQL Server 2005* and *ProClarity Desktop Professional 6.3* software package in detail. Second part of this paper discusses architecture of university data analysis system, projects key components and prepares to realization of the system. In the third part realization of the system is performed, during this process data storage and various cubes are created based on KTU abandoned data base. In the fourth part of this paper various data diagrams are drawn using *Excel 2007* and performance map diagrams are created using *ProClarity Analytics*. These drawings enable to perform analysis of university data in many different dimensions.

System of KTU DS data analysis creates new opportunities in the process of university activities improvement by analyzing information of modules, students, lecturers, studies programs and other.

IVADAS

OLAP – tai betarpiškas duomenų analizės procesas, kuris leidžia valdyti informaciją, suteikia galimybę pažvelgti į verslo duomenis visais įmanomais pjūviais. Ar teko susidurti su problema, kai informacijos reikia dabar ir tuoj pat, o naudojamoje sistemoje dar nėra tokios ataskaitos? Tikriausiai, kad taip. Tokiu atveju tenka kviestis programuotoją, o jei jis negali padėti, tenka sėsti pačiam ir nuobodžiai pervedinėti skaičius iš kelių ataskaitų į *Excel* lenteles ar ant popieriaus. Verslo analizės sistemos (*Business Intelligence*) programų pakete yra įrankiai, kurie leidžia sukurti bet kokią ataskaitą, bet kokiu pjūviu ir tam visiškai nereikia programavimo įgūdžių. Šiandien verslo analizės sistema yra neišvengiama efektyvaus verslo valdymo priemonė, gebanti agreguoti iš įvairių šaltinių gaunamus didelius duomenų kiekius, pateikti juos lengvai suprantama forma ir užtikrinti informacijos sklaidą bei informacijos saugumą.

Šiame darbe yra nagrinėjama tyrimo problema, kaip verslo analizės sistemą efektyviai pritaikyti universiteto veiklos gerinimo procese. Dėl didžiulės duomenų gausos universiteto darbuotojams sunku tinkamai įvertinti ir analizuoti universiteto duomenis, daryti svarbias su studijų kokybe, mokymosi tendencijomis susijusias išvadas bei prognozes. Tyrimo objektas yra universiteto dėstytojų, studentų, modulių ir kitų objektų duomenų analizė. Tyrimo sritis – universiteto duomenų bazės modernizavimas bei *Business Intelligent* programų paketų pritaikymas analizuojant duomenis. Verslo duomenų intelektualios analizės duomenų kubų kūrimo ir saugojimo modelių tyrimas atliekamas naudojant *Microsoft SQL Server 2005 Business Intelligent Management Studio* priemones.

Tyrimo tikslas yra intelektualios analizės duomenų kubų ir *ProClarity* galimybių sukūrimas ir tyrimas, siekiant atlikti KTU studijų informacinės sistemos intelektualią analizę, kuri leistų pagerinti studijų proceso administravimą fakulteto ir rektoriato atsakingiems darbuotojams. Tyrimo uždaviniai yra šie:

1. Išanalizuoti duomenų kubų kūrimo bei saugojimo modelius.
2. Suprojektuoti sistemą bei duomenų bazę.
3. Sukurti duomenų saugyklą bei išsamiai duomenų analizei atlikti skirtus kubus.
4. Sukinių bei *ProClarity* analizės grafikų pagalba analizuoti universiteto duomenis.

Darbą sudaro keturios dalys. Pirmojoje dalyje analizuojama duomenų kubo sąvoka, jo struktūra bei elementai, apžvelgti dažniausiai naudojami intelektualios duomenų analizės kūrimo programinės įrangos paketai. Taip pat detaliam išnagrinėtas pasirinktas programinės įrangos paketas – *Microsoft SQL Server 2005* bei apžvelgta papildomai pasirinkta programinė įranga *ProClarity Desktop Professional 6.3*.

Antrojoje darbo dalyje buvo aptarta universiteto duomenų analizės sistemos architektūra, nubraižytos jos įrangos bei komponentų diagramos, atskleidžiančios sistemos struktūros ypatumus. Projektuojant sistemą buvo naudojama unifikuota modeliavimo kalba, kurios metodais nubraižytos panaudojimo atvejų diagramos, aprašyti duomenų kubų, *Excel 2007* sukinių bei *ProClarity* grafikų kūrimo algoritmai veiklos ir sekų diagramomis.

Trečiojoje darbo dalyje buvo atlikta sistemos realizacija, kurios metu buvo KTU palikiminės duomenų bazės pagrindu sukurta duomenų saugykla. Buvo sukurta įvairių rūšių kubų, pradedant paprastais, žvaigždės tipo ir baigiant sudėtingesniu – snaigės tipo, kurie leido atskleisti kubų kūrimo ypatybes.

Ketvirtojoje darbo dalyje panaudojant sudarytų kubų duomenis *Excel 2007* programa buvo kuriami įvairūs duomenų sukiniai, kurių pagalba įvairiais pjūviais atlikta universiteto duomenų analizė. Taip pat buvo naudojama papildoma *ProClarity* programinė įranga, kuria buvo nubraižytos indikatorinės diagramos.

Siekiant geresnio nagrinėjamo duomenų suvokimo, analizei naudojama medžiaga bei gauti rezultatai darbe pateikiami struktūrizuotose lentelėse, iliustruojami grafikais ir paveikslais.

1. ANALITINĖ DALIS

1.1. Analizės tikslas

Šios analizės tikslas yra išsiaiškinti kas yra duomenų kubai ir iš kokių elementų jie susideda. Svarbu žinoti bei suprasti jų kūrimo, veikimo, tvarkymo, priežiūros bei kitus susijusius principus. Išsiaiškinti esamų sprendimų taikymo pavyzdžius, suprasti kaip veikia sistema, kokia jos architektūra, rasti jos trūkumus bei ieškoti patobulinimo sprendimų. Išsiaiškinti vartotojų grupes, jų galimybes bei ribas, suprasti atliekamas funkcijas. Rasti šiandieną jau naudojamus programinius paketus, atlikti jų palyginimo analizę bei pasirinkti savo kuriamai sistemai labiausiai tinkantį produktą. Išsiaiškinti pasirinkto programinio paketo privalumus, trūkumus bei galimybes.

1.2. Analizės metodai

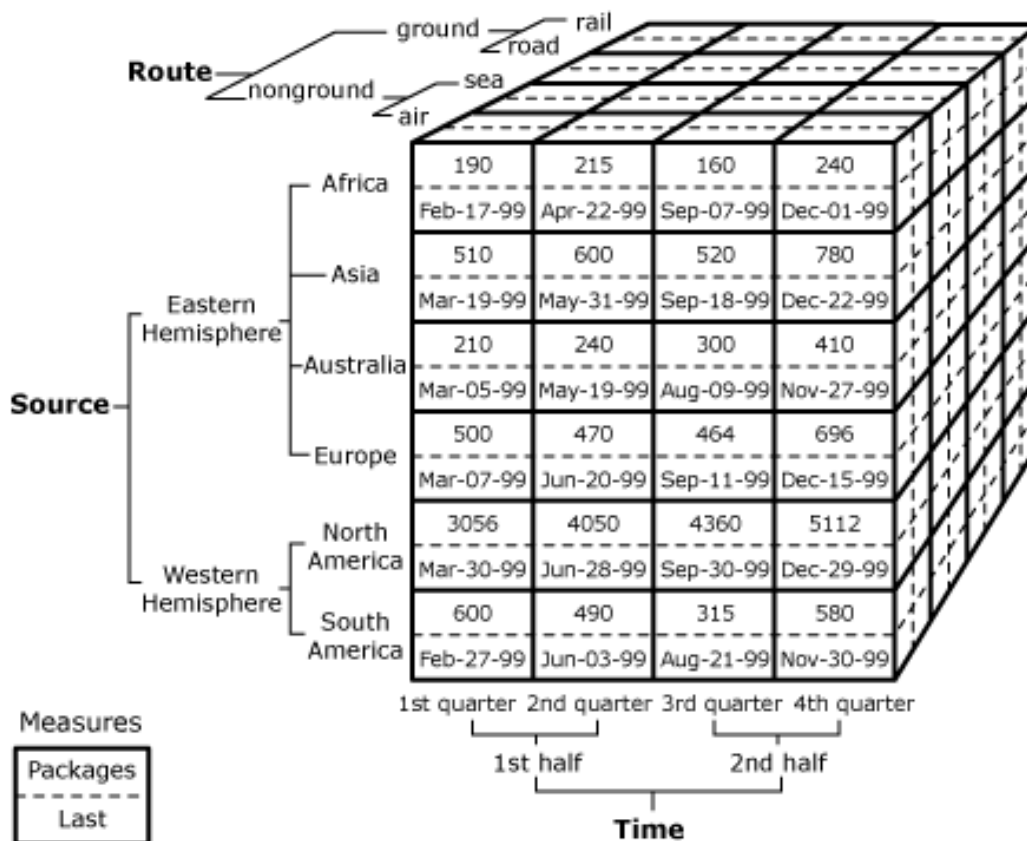
Siekiant užsibrėžto analizės tikslo darbo rašymo metu bus naudojami sistemos analizės metodai – tai mokslinės bei techninės literatūros ieškojimas, atrinkimas, skaitymas bei analizavimas.

1.3. Duomenų kubo analizė

Kubas apibrėžiamas savo matais (*measures*) ir dimensijomis (*dimensions*) [2, 3]. Matai ir dimensijos į kubą yra įkeliami iš lentelių (*tables*) ir atvaizdų (*views*), iš duomenų šaltinio atvaizdo, kuriuo kubas remiasi arba kuris yra generuojamas iš laipsnių ir dimensijų apibrėžimų ar nustatymų (*definitions*).

Išnagrinėsime 1.1 paveiksle pavaizduotą importo kubą, kuris turi du matus: *Packages* (paketai) ir *Last* (Paskutinė data), ir tris susijusias dimensijas, *Route* (maršrutas), *Source* (šaltinis), ir *Time* (laikas).

Mažesnės tekstinės reikšmės aplink kubą yra dimensijų nariai (*members of the dimensions*). Pavyzdyje nariai yra *Ground* (*Route* dimensijos narys), *Africa* (*Source* dimensijos narys) ir *Ist quoter* (*Time* dimensijos narys).



1.1 pav. Importo kubo pavyzdys

Kubo narveliuose esančios reikšmės rodo du matus: *Packages* ir *Last*. *Packages* matas parodo importuoto paketo numerį ir *Sum* funkcija yra naudojama apibendrinti (*aggregate*) faktus. *Last* matas parodo gavimo datą ir *Max* funkcija yra naudojama apibendrinti faktus.

Route dimensija rodo būdą, kuriuo importas pasiekė paskirties vietą. Dimensijos nariai yra:

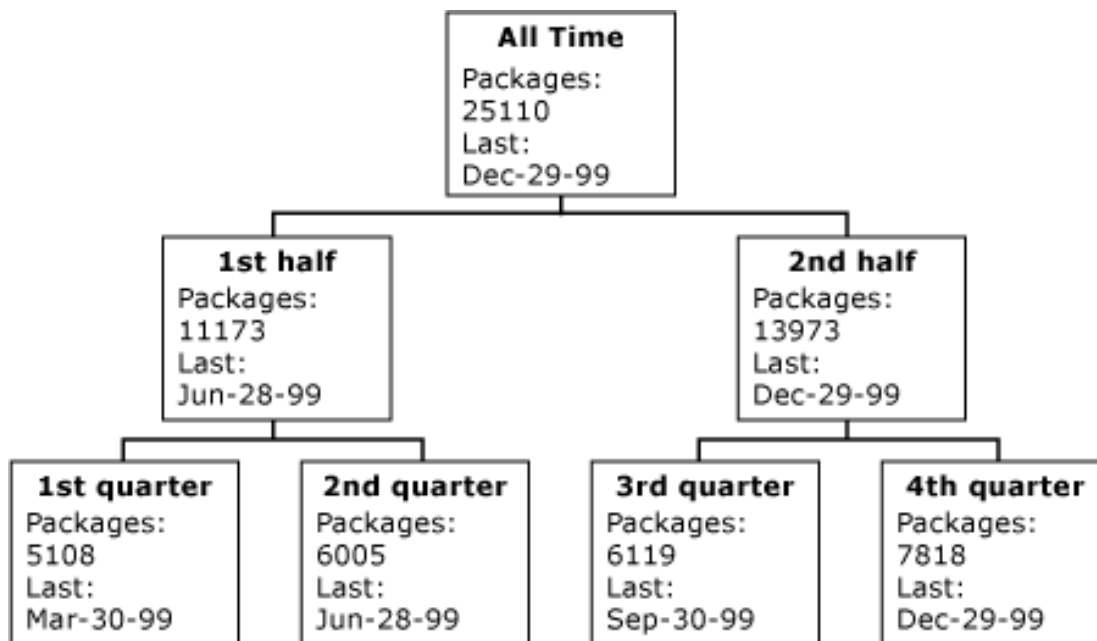
- ground (žemė),
- nonground (ne žemė),
- air (oras),
- sea (jūra),
- road (kelias) ir
- rail (geležinkelis).

Source dimensija rodo vietą, kur importas yra gaminamas, pavyzdžiui, Afrikoje ar Azijoje (Africa ar Asia).

Time dimensija rodo ketvirčius (*quarters*) ir pusmečius (*halves*).

Agregatai (*aggregates*) – apibendrintos reikšmės (apibendrinimai). Verslo vartotojai kube gali apibrėžti kiekvieno mato reikšmę (*value*) kiekvienam nariui ir kiekvienai dimensijai, priklausomai nuo nario lygmens dimensijoje, nes analizės paslaugos (*Analysis*

Services) apibendrina (*aggregate*) reikšmes aukštesniems lygmenims pagal poreikius. Pavyzdžiui, mato reikšmės mūsų iliustracijoje gali būti apibendrintos pagal standartinę kalendoriaus hierarchiją naudojant *Calendar Time* hierarchiją *Time* dimensijoje, kaip iliustruojame žemiau parodytame 1.2 paveiksle.



1.2 pav. *Calendar Time* hierarchijos naudojimas *Time* dimensijoje

Papildomai matų apibendrinimui naudojant vieną dimensiją, jūs galite matus apibendrinti naudodami kelių skirtingų dimensijų narių kombinacijas. Tai leidžia verslo vartotojams įvertinti matus daugelyje dimensijų tuo pačiu metu.

Pavyzdžiui, jei verslo vartotojas nori analizuoti ketvirčių importus, kurie atvežami oru iš Rytų pusrutulio ir Vakarų pusrutulio, verslo vartotojas gali parengti kubo užklausą, kuri išrinks tokius duomenis.

Kai kubas yra sudarytas, jūs galite kurti naujus apibendrinimus arba jūs galite keisti egzistuojančius apibendrinimus įrengdami parametrus tokius kaip ar apibendrinimai yra paskaičiuojami iki apdorojimo, ar skaičiuojami užklausoje metu, kaip parodyta 1.3 paveiksle.

			Packages	Last				
			All Sources	Eastern Hemisphere	Western Hemisphere	All Sources	Eastern Hemisphere	Western Hemisphere
All Time			25110	6547	18563	Dec-29-99	Dec-22-99	Dec-29-99
	1st half		11173	2977	8196	Jun-28-99	Jun-20-99	Jun-28-99
		1st quarter	5108	1452	3656	Mar-30-99	Mar-19-99	Mar-30-99
		2nd quarter	6065	1525	4540	Jun-28-99	Jun-20-99	Jun-28-99
	2nd half		13937	3570	10367	Dec-29-99	Dec-22-99	Dec-29-99
		3rd quarter	6119	1444	4675	Sep-30-99	Sep-18-99	Sep-30-99
		4th quarter	7818	2126	5692	Dec-29-99	Dec-22-99	Dec-29-99

1.3 pav. Matų apibendrinimas naudojant skirtingas dimensijas

Matų, atributų ir hierarchijos vietos nurodymas (*Measures, Attributes ir Hierarchies*). Matai, atributai ir hierarchijos pavyzdžio kube yra atkeliami iš žemiau pateikto 1.4 paveikslėlio lentelėje nurodytų stulpelių, esančių kubo faktų ir dimensijų lentelėse (*cube's fact and dimension tables*).

Measure or attribute (level)	Members	Source table	Source column	Sample column value
Packages measure	Not applicable	ImportsFactTable	Packages	12
Last measure	Not applicable	ImportsFactTable	Last	May-03-99
Route Category level in Route dimension	nonground,ground	RouteDimensionTable	Route_Category	Nonground
Route attribute in Route dimension	air,sea,road,rail	RouteDimensionTable	Route	Sea
Hemisphere attribute in Source dimension	Eastern Hemisphere,Western Hemisphere	SourceDimensionTable	Hemisphere	Eastern Hemisphere
Continent attribute in Source dimension	Africa,Asia,AustraliaEurope,N. America,S. America	SourceDimensionTable	Continent	Europe
Half attribute in Time dimension	1st half,2nd half	TimeDimensionTable	Half	2nd half
Quarter attribute in Time dimension	1st quarter,2nd quarter,3rd quarter,4th quarter	TimeDimensionTable	Quarter	3rd quarter

1.4 pav. Matų, atributų ir hierarchijos vietos nurodymas

Duomenys į vieną konkretų kubo narvelį paprastai įkeliami iš faktų lentelės apibendrinant daugelį eilučių. Pavyzdžiui, kubo narvelis esantis *air* nario, *Africa* nario ir *1st quarter* nario susikirtime, turi reikšmę, kuri yra įkelta apibendrinant šias lentelėje parodytas eilutes iš *ImportsFactTable* faktų eilutės.

Import_ReceiptKey	RouteKey	SourceKey	TimeKey	Packages	Last
3516987	1	6	1	15	Jan-10-99
3554790	1	6	1	40	Jan-19-99
3572673	1	6	1	34	Jan-27-99
3600974	1	6	1	45	Feb-02-99
3645541	1	6	1	20	Feb-09-99
3674906	1	6	1	36	Feb-17-99

1.5 pav. *ImportsFactTable* faktų eilutės kubo narvelis

Aukščiau parodyto 1.5 paveikslo lentelėje kiekviena eilutė turi tas pačias reikšmes *RouteKey*, *SourceKey* ir *TimeKey* stulpeliuose, tai rodo, kad šios eilutės teikia reikšmes tam pačiam kubo narveliui.

Čia pateiktame pavyzdyje parodytas labai parastas kubas, kuriame kubas turi tik vieną matų grupę ir visos dimensijų lentelės yra sujungtos į faktų lentelę pagal žvaigždės schemą (*star schema*). Kita bendra schema yra snaigės schema (*snowflake schema*), kurioje viena ar daugiau dimensijų lentelių jungiamos į kitas dimensijų lenteles, o ne tiesiai į faktų lentelę. Šis pavyzdys turi tik vienintelę faktų lentelę. Kai kubas turi kelias faktų lenteles, matai iš kiekvienos faktų lentelės yra organizuojami į matų grupes (*measure groups*) ir matų grupė yra susiejama su specialiu rinkiniu dimensijų, apibrėžiant dimensijų ryšius (*relationships*). Šie ryšiai yra apibrėžiami, nurodant dalyvaujančias lenteles duomenų šaltinio atvaizde ir ryšio pobūdį (*granularity*).

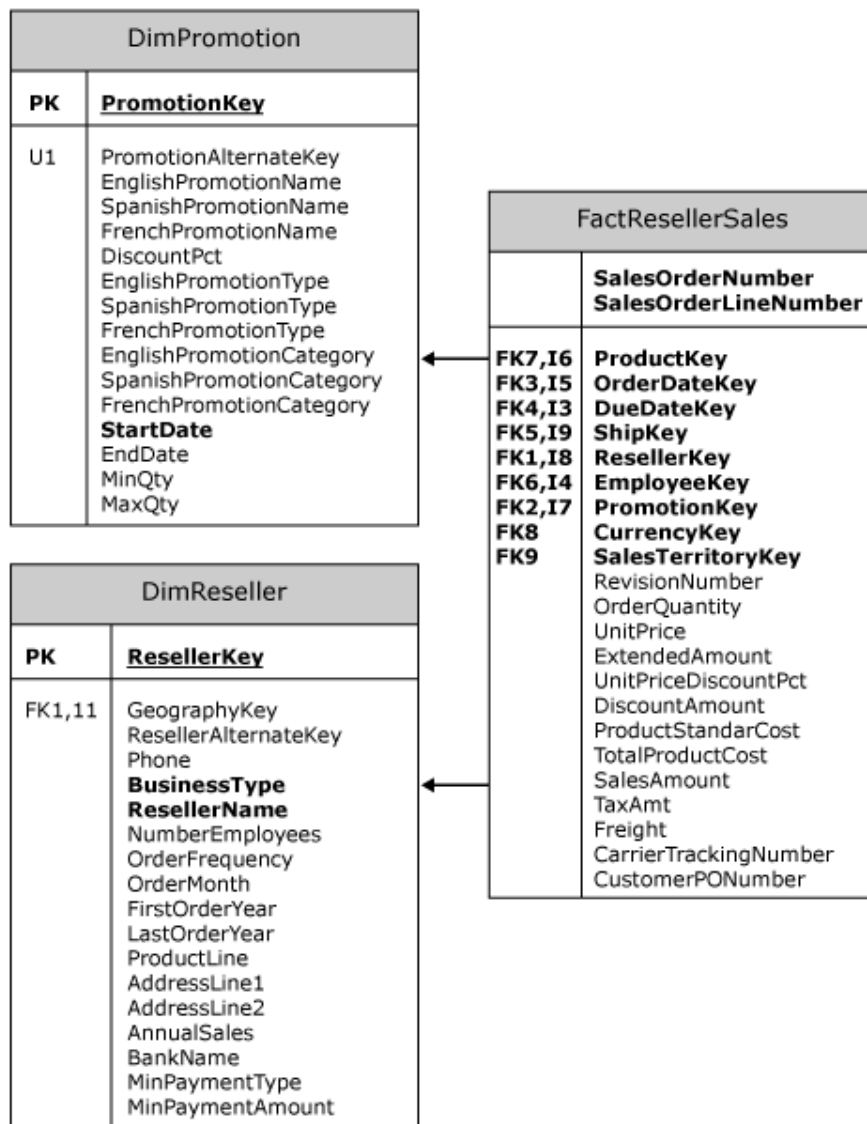
Dimensijų pagrindai. Visos *Microsoft SQL Server 2005* analizės paslaugų dimensijos yra atributų grupės, kurios remiasi lentelių ar atvaizdų stulpeliais iš duomenų šaltinio atvaizdo. Dimensijos egzistuoja nepriklausomai nuo kubo, gali būti naudojamos daugiamačiuose kubuose, gali būti naudojamos daug kartų viename kube ir gali būti susietos su analizės paslaugomis kitais dariniais (*instances*). Dimensijos yra dviejų tipų:

- dimensija, kuri egzistuoja nepriklausomai nuo kubo, yra vadinama duomenų bazės dimensija,
- dimensija, kuri egzistuoja kube, yra vadinama kubo dimensija.

Dimensijos struktūra yra labai priklausoma nuo dimensijų lentelės ar lentelių struktūros. Paprasčiausia struktūra yra vadinama žvaigždės schema, kur kiekviena dimensija remiasi vienintele dimensijos lentele, kuri tiesiogiai susieta su faktų lentele naudojant „pagrindinis raktas (*primary key*) - svetimasis raktas“ (*foreign key*) ryšiu (*relationship*).

1.6 paveikslo diagrama iliustruoja *AdventureWorksDW* duomenų bazės dalies pavyzdį, kurios *FactResellerSales* faktų lentelė yra surišta su dviem dimensijų lentelėmis *DimReseller*

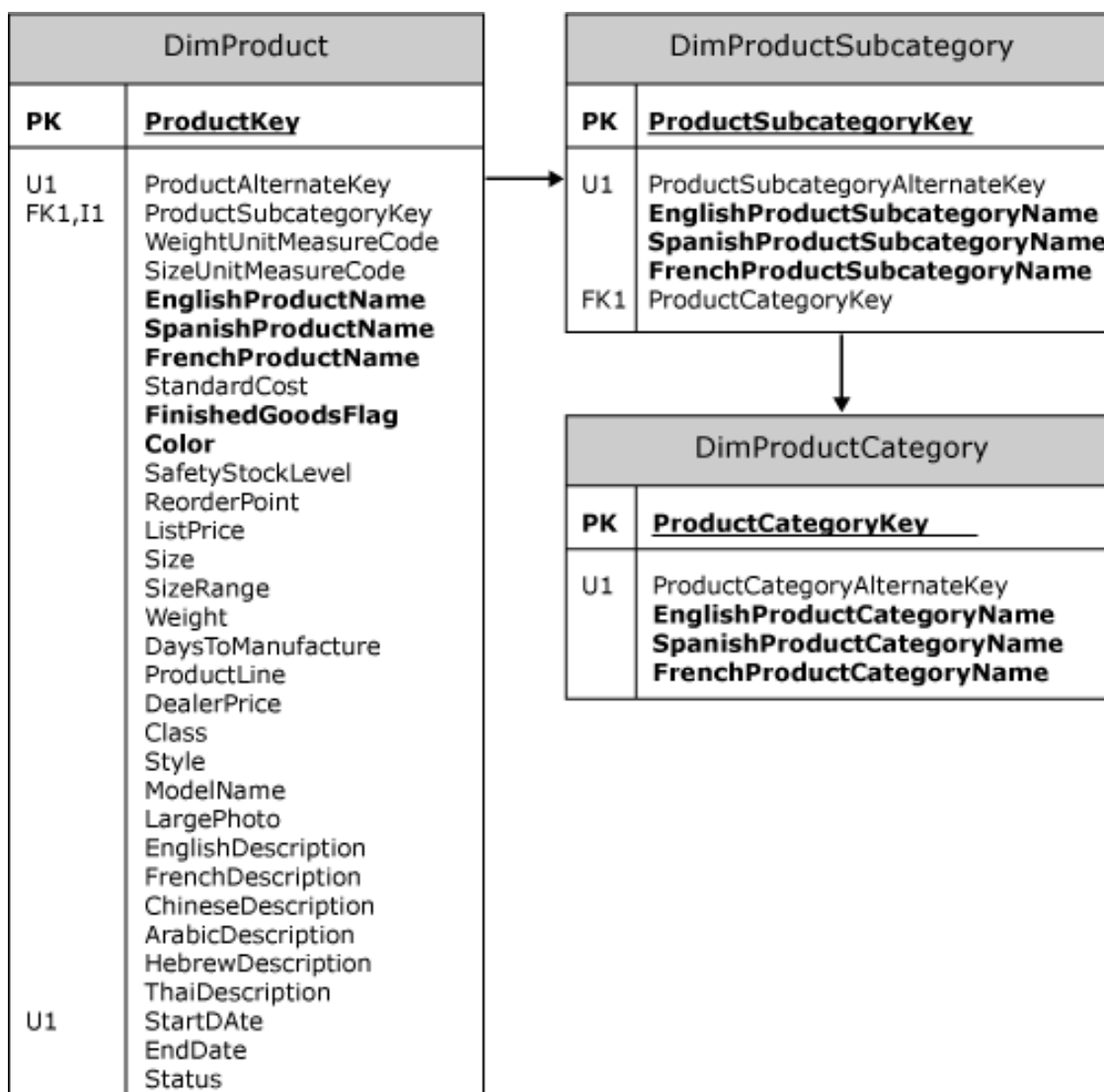
ir *DimPromotion*. *ResellerKey* stulpelis *FactResellerSales* faktų lentelėje apibrėžia svetimo rakto ryšį su *ResellerKey* pagrindinio rakto stulpeliu *DimReseller* dimensijos lentelėje. Panašiai *PromotionKey* stulpelis *FactResellerSales* faktų lentelėje apibrėžia svetimo rakto ryšį su *PromotionKey* pagrindinio rakto stulpeliu *DimPromotion* dimensijos lentelėje.



1.6 pav. *AdventureWorksDW* duomenų bazės dalies pavyzdys

Dimensijos, kuriamos snaigės schemos (*Snowflake schema*) pagrindu. Dažnai sudėtingesnės struktūros yra reikalingos, nes informacija iš daugelio lentelių yra reikalinga apibrėžiant dimensiją. Šioje struktūroje, vadinamoje snaigės schema, kiekviena dimensija remiasi atributais iš stulpelių daugelio lentelių, surištų viena su kita ir būtinai su faktų lentele „pagrindinį raktas – svetimas raktas“ ryšiais.

Pavyzdžiui, 1.7 paveikslo diagrama iliustruoja lenteles, reikalingas *Product* dimensijai visiškai aprašyti iš *AdventureWorksDW* projekto pavyzdžio:



1.7 pav. Lentelės, reikalingos *Product* dimensijai aprašyti

Snaigės schema ar nuorodos ryšys (*snowflake schema vs reference relationship*). Ankstesnėje diagramoje *FactResellerSales* faktų lentelė neturi svetimio rakto ryšio su *DimGeography* dimensijos lentele. Tačiau, *FactResellerSales* faktų lentelė turi svetimio rakto ryšį su *DimReseller* dimensijos lentele, kuri toliau turi svetimio rakto ryšį su *DimGeography* dimensijos lentele. Norint apibrėžti *Reseller* dimensiją, kuri turi geografinę informaciją apie kiekvieną perpardavinėtoją, jūs turite išrinkti šiuos atributus iš *DimGeography* ir *DimReseller* dimensijų lentelių. Tačiau, naudodami analizės paslaugas (*Analysis Services*), jūs galite gauti tuos pačius rezultatus sukurdami dvi atskiras dimensijas ir surišdami jas mato grupėje apibrėždami nuorodos dimensijos ryšį (*a reference dimension relationship*) tarp dviejų dimensijų.

Vienas pranašumų naudoti scenarijuje nuorodos dimensijos ryšius yra tai, kad jūs galite sukurti vienintelę geografinę dimensiją ir tada sukurti kelias kubo dimensijas, ant šios

geografinės dimensijos, nereikalaudami jokios papildomos erdvės duomenų saugojimui. Pavyzdžiui, jūs galite surišti vieną geografinio kubo dimensiją su perpardavėjo dimensija, o kitą geografinio kubo dimensiją su vartotojo dimensija.

Dimensijų apdorojimas. Po to, kai sukuriama dimensija, norint peržiūrėti atributus ir dimensijų hierarchiją ją privaloma įvykdyti. Jeigu dimensijos struktūra pasikeičia arba lentelių duomenys yra atnaujinami, reikia įvykdyti dimensiją iš naujo norint pamatyti pakeitimus. Kai po pakeitimų įvykdoma nauja dimensija, taip pat reikia iš naujo sugeneruoti visus kubus, kuriuose naudojama ši dimensija. Kitu atveju kubo negalėsime peržiūrėti.

Atributai ir atributų hierarchija (*Attributes and Attribute Hierarchies*). Dimensijos yra atributų rinkiniai, kurie ribojamas vienu ar keliais stulpeliais lentelėje arba atvaizde iš duomenų šaltinio atvaizdo.

Raktinis atributas (*key attribute*). Kiekviena dimensija turi raktinį atributą. Kiekvienas atributas ribojamas vienu ar daugiau stulpelių dimensijos lentelėje. Raktinis atributas yra dimensijos atributas, kuris identifikuoja stulpelius dimensijos pagrindinėje lentelėje, kurie yra naudojami svetimo rakto ryšiams su faktų lentele. Tipiška, raktinis atributas atstovauja pagrindinio rakto stulpelį ar stulpelius dimensijos lentelėje. Jūs galite apibrėžti lentelės loginį pagrindinį raktą duomenų šaltinio atvaizde, kuris neturi fizinio pagrindinio rakto pamatiniame duomenų šaltinyje.

Kai jūs apibrėžiate raktinius atributus, *Cube Wizard* ir *Dimension Wizard* stengiasi naudoti pagrindinio rakto stulpelius dimensijos lentelės duomenų šaltinio atvaizde. Jei dimensijos lentelė neturi loginio pagrindinio rakto ar apibrėžto fizinio pagrindinio rakto, vedliai gali nesugebėti teisingai apibrėžti raktinius atributus dimensijai.

Atributo susiejimas su stulpeliais duomenų šaltinio atvaizdo lentelėse ar atvaizduose (*Binding an Attribute to Columns*). Atributas yra susijęs (*bound*) su stulpeliais vienoje ar daugiau duomenų šaltinio atvaizdo lentelėje ar atvaizde. Atributas yra visada susijęs su vienu ar daugiau raktinių stulpelių, kurie apibrėžia narius (*members*), kurie yra tame attribute. Pagal numatytąją reikšmę, tai yra tik stulpelis su kuriuo atributas yra susietas. Atributas gali būti taip pat susietas su vienu ar daugiau papildomų stulpelių specialioms tikslams. Pavyzdžiui, atributo *NameColumn* savybė apibrėžia vardą, kuris rodomas vartotojui kiekvienam atributo nariui – ši atributo savybė gali būti susieta su konkrečiu dimensijos stulpeliu per duomenų šaltinio atvaizdą arba gali būti susietas su skaičiuojamuoju stulpeliu duomenų šaltinio atvaizde.

Atributų hierarchija (*Attribute Hierarchies*). Numatyta, kad atributų nariai yra organizuojami dviejų lygmenų hierarchija, susidedančia iš lakšto (*leaf*) lygmens ir lygmens Visi (*All*). Lygmuo Visi turi apibendrintą reikšmę atributų narių per visą matą kiekvienoje

matų grupėje tai dimensijai, kurios susietas atributas yra narys. Tačiau jei *IsAggregatable* savybė yra nustatyta *False*, lygmuo Visi nekuriamas.

Atributai gali būti ir taip paprastai būna, įrengti į vartotojo apibrėžtas hierarchijas, kurios teikia išskleidimo kelius (*drill-down paths*), kuriuose vartotojai gali naršyti matų grupės duomenis, su kuria atributas yra surištas. Kliento programose atributai gali būti naudojami teikti grupavimą ir konstruoti informaciją. Kai atributai yra įrengti su vartotojo apibrėžta hierarchija, jūs nustatote ryšius tarp hierarchijos lygmenų, kai lygmenys yra surišti daug – su vienu arba vienas – su vienu ryšiais (vadinamais natūraliais ryšiais). Pavyzdžiui, kalendorinio laiko hierarchijoje, *Day* (diena) lygmuo turi būti surišta su *Month* (mėnuo) lygmeniu, *Month* lygmuo su *Quarter* lygmeniu ir t.t. Nustatant ryšius tarp lygmenų vartotojo įrengtoje hierarchijoje įgalina analizės paslaugos (*Analysis Services*) apibrėžti daugiau naudingų apibendrinimų, padidinant užklausų našumą ir gali taip pat saugoti atmintį proceso metu, kas aktualu su dideliais ir sudėtingais kubais.

Atributų ryšiai, žvaigždės schemas ir snaigės schemas (*Attribute Relationships, Star Schemas, and Snowflake Schemas*). Numatyta, žvaigždės schemoje visi atributai yra tiesiogiai surišti su raktiniu atributu, kas įgalina vartotojus naršyti faktus kube, remiantis bet kurio atributo hierarchija dimensijoje. Snaigės schemoje atributas yra arba tiesiogiai surištas su raktiniu atributu, jei jų pamatinės lentelės yra tiesiogiai surištos su faktų lentele, arba yra netiesiogiai surištas su atributų reikšmėmis, kurios yra susiejamos su raktu pamatinėje lentelėje, kuri riša snaigės lentelę su tiesiogiai surišta lentele.

Apibendrinimai ir apibendrinimų konstravimas (*Aggregations and Aggregation Designs (SSAS)*). Apibendrinimai yra iš anksto suskaičiuotos sumos iš lakšto duomenų narvelių. Apibendrinimai pagerina užklausų parengimo laiką parengiant atsakymus prieš tai, kol klausimai bus pateikti. Pavyzdžiui, kada duomenų saugyklos faktų lentelė turi šimtus tūkstančių eilučių, užklausa reikalaujanti savaitinių pardavimų sumų kiekvieno produkto eilutei gali užimti daug laiko atsakyti, jei visos eilutės faktų lentelėje turi būti nuskenuotos ir susumuotos užklausos rengimo metu. Tačiau atsakymas gali būti pateiktas akimirksniu, jei sumuojami duomenys šios užklausos atsakyme būtų paskaičiuoti iš anksto. Šie iš anksto apskaičiuojami duomenys atsiranda apdorojimo metu ir yra greitos *TADA* (OLAP) technologijos pagrindas.

Kubai yra būdas, kuriuo *TADA* technologija organizuoja duomenų sumas į daugiamates struktūras. Dimensijos ir jų atributų hierarchijos atspindi užklausas, kurios gali būti pareikalautos iš kubo.

Apibendrinimai yra saugomi daugiamatės struktūros narveliuose, kurie nurodomi dimensijos koordinatėmis. Pavyzdžiui, klausimas “Kokie buvo pardavimai produkto X 1998

metais *Northwest* regionui?" įtraukia tris dimensijas (*Product*, *Time*, ir *Geography*) ir vieną matą (*Sales*). Kubo narvelio reikšmė nurodytose koordinatėse (produktas X, 1998, *Northwest*) ir yra atsakymas, tai viena skaitinė reikšmė.

Kiti klausimai gali gražinti daug reikšmių. Pavyzdžiui, "Kokio dydžio techninės įrangos (*Hardware*) pardavimai buvo keturiuose ketvirčiuose visuose regionuose 1998 metais?"

Tokios užklausos gražina rinkinį narvelių, kurių koordinatės atitinka nurodytas sąlygas. Užklausos gražinamų narvelių skaičius priklauso nuo elementų skaičiaus *Hardware* lygmenyje *Product* dimensijoje, keturiems ketvirčiams 1998 metais, ir nuo regionų skaičiaus *Geography* dimensijoje. Jei visi suminiai duomenys buvo iš anksto apskaičiuoti apibendrinimuose, užklausos atsakymo laikas priklausys tik nuo laiko, kurio reikia išrinkti nurodytus narvelius. Jokių skaičiavimų ar duomenų skaitymo iš faktų lentelės nebus atliekama.

Nors išankstiniai apskaičiavimai visų galimų apibendrinimų kube gali pateikti greitus galimus atsakymus į visas užklausas, analizės paslaugos (*Analysis Services*) gali lengvai paskaičiuoti keletą apibendrinamų reikšmių ir be išankstinių apskaičiavimų. Papildomi apskaičiavimai visų galimų apibendrinimų reikalauja žybaus apdorojimo laiko ir atminties. Todėl čia yra pasirenkamos galimybės tarp atminties reikalavimų ir procentų galimų išankstinių apskaičiavimų. Jei apibendrinimai neapskaičiuoti (0%), kubo parengimui reikalaujamas laiko kiekis ir atminties vieta yra minimizuoti, bet užklausos atsakymo laikas gali būti lėtas. Nes duomenys reikalingi atsakymui turi būti išrinkti iš lakšto narvelių ir tik po to apibendrinti užklausos rengimo metu, atsakant į kiekvieną užklausa. Pavyzdžiui, gražinant vienintelę reikšmę, kuri atsako į anksčiau pateiktą klausimą ("Kokie buvo pardavimai produkto X 1998 metais *Northwest* regionui?"), gali pareikalauti skaityti tūkstančius eilučių duomenų iš stulpelio, naudojamo teikti *Sales* matą kiekvienai eilutei ir tada apskaičiuoti sumą. Dar daugiau, laiko ilgumas reikalingas išrinkti duomenis labai priklausys nuo parinkto duomenų saugojimo modelio (*mode*) – *MOLAP*, *HOLAP*, ar *ROLAP*.

Apibendrinimų projektavimas (*Designing Aggregations*). *Microsoft SQL Server 2005* analizės paslaugos (*Analysis Services*) turi savyje sudėtingus algoritmus parinkti apibendrinimus išankstiniams apskaičiavimams, taigi todėl kiti apibendrinimai gali būti greitai apskaičiuoti iš išankstinių reikšmių. Pavyzdžiui, jei apibendrinimai yra iš anksto apskaičiuoti *Month* lygmeniui iš *Time* hierarchijos, apskaičiavimai *Quarter* lygmeniui reikalauja tik sudėti tris skaičius, kurie gali būti greitai apskaičiuoti pagal pareikalavimą. Ši technika taupo apdorojimo laiką ir sumažina atminties reikalavimus su minimaliu užklausos atsakymo laiko padidiniu.

Aggregation Design Wizard vedlys teikia parametrus jums nustatyti atminties ir procentų ribojimus algoritmams, kad pasiektumėte tinkamas proporcijas tarp užklauso atsakymo laiko ir atminties reikalavimų. Tačiau *Aggregation Design Wizard* vedlio algoritmas priima, kad visos galimos užklauso galimai vienodos. *Usage-Based Optimization Wizard* vedlys leidžia patenkinti kliento programas. Naudojantis vedliu suderinant kubo apibendrinimus, jūs galite pagreitinti atsakymus į dažnas užklauso ir sulėtinti atsakymus į retas užklauso be žymaus padidėjimo atminties, reikalingos kubui.

Apibendrinimai yra projektuojami naudojant vedlius, bet nėra faktiškai apskaičiuojami kol ta dalis, kuri reikalinga suprojektuotam apibendrinimui yra apdorojama. Kai apibendrinimas jau yra sukurtas, jei kubo struktūra kada nors pakeičiama arba jei duomenys yra pridėjami ar pakeičiami kubo šaltinio lentelėse, tai paprastai būtina peržiūrėti kubo apibendrinimus ir dar kartą gaminti kubą.

1.4. Vartotojų analizė

1.4.1. Vartotojų aibė, tipai ir savybės

Sistemoje egzistuos keletas vartotojų tipų:

- *SQL* serverio administratoriai
- Duomenų bazės administratoriai
- Analitikai

Kiekvienas jų turės savo paskirtį sistemoje bei prisijungęs prie jos galės atlikti jam priskirtus leistinus veiksmus. *SQL* serverio administratorius turės aukščiausio lygio teises. Jis administruoja kitus vartotojus, prižiūri *SQL* serverio veikimą. Duomenų bazės administratorius turi KTU duomenų bazės administratoriaus teises, kurios jam suteikia priėjimą prie duomenų ir leidžia juos administruoti, t.y. įvedinėti naujus įrašus, pašalinti nereikalingus, tvarkyti duomenų bazės struktūrą, kurti kubus. Analitikas gali prisijungti prie sistemos jam suteiktais programiniais paketais - *Microsoft Excel 2007* bei *ProClarity Professional 6.3*.

1.4.2. Vartotojų tikslai ir problemos

SQL serverio administratoriaus pagrindinis tikslas yra užtikrinti serverio stabilų darbą bei šalinti sistemos sutrikimus. Kad tai pasiekti, jis turi laiku įrašyti saugumo spragas

užtaisančius atnaujinimus. *SQL* serverio administratorius taip pat atsakingas ir už kitus vartotojus. Jis juos sukuria, ištrina arba modifikuoja esamus, nustato jiems priskirtas teises.

Duomenų bazės administratorius gali prisijungti prie KTU duomenų bazės bei gali disponuoti šiais duomenimis. Jis turi teisę įrašyti naujus, trinti nereikalingus arba modifikuoti klaidingus. Duomenų bazės administratorius projektuoja vaizdinius bei jų pagrindu kuria duomenų kubus.

Analitikų tikslai yra pasinaudojant jiems suteiktais įrankiais - *Microsoft Excel 2007* bei *ProClarity Professional 6.3* programomis kurti sukinius bei kitaip analizuoti duomenų bazėje esančius duomenis.

1.5. Esamų sprendimų analizė

Norint atlikti verslo duomenų intelektualios analizės duomenų kubų kūrimo ir saugojimo modelių tyrimą, labai svarbus veiksmas yra tinkamos programinės įrangos pasirinkimas. Ši programinė įranga turi turėti duomenų bazių serverį bei palaikyti *OLAP* (angl. *Online Analytical Processing* - betarpiška duomenų analizė) technologiją. Šiuo metu patys populiariausi duomenų bazių kūrimo įrankiai yra šie:

- *Microsoft SQL Server 2005*
- *MySQL Enterprise Server 5.0*
- *IBM DB2 Data Warehouse Enterprise Edition 9.5*
- *Oracle Database 11g*

Visi šie įrankiai yra komerciniai bei tarpusavyje konkuruojantys, todėl pasižymi skirtingomis savybėmis. Vienintelis *MySQL* yra atviro kodo ir platinamas nemokamai pagal *GNU General Public License* licenziją, todėl yra laisvai prieinamas kiekvienam vartotojui. Žemiau pateiktoje lentelėje matome šių serverių kai kurių parametrų palyginimus.

Lentelė 1.1 Serverių parametrų palyginimas

	<i>Microsoft SQL Server 2005</i> [4, 5]	<i>MySQL Enterprise Server 5.0</i> [6, 7]	<i>IBM DB2 Data Warehouse Enterprise Edition 9.5</i> [8, 9]	<i>Oracle Database 11g</i> [10, 11]
Palaikomos operacinės sistemos	<ul style="list-style-type: none"> • <i>Windows 2000 (SP4)</i> • <i>Windows XP (SP2)</i> • <i>Windows Server 2003 (SP1)</i> • <i>Windows Vista</i> 	<ul style="list-style-type: none"> • <i>AIX</i> • <i>BSDi</i> • <i>FreeBSD</i> • <i>HP-UX</i> • <i>i5/OS</i> • <i>Linux</i> • <i>Mac OS X</i> • <i>NetBSD</i> • <i>Novell</i> 	<ul style="list-style-type: none"> • <i>OS/2</i> • <i>UNIX</i> • <i>Windows</i> • <i>Linux (including Linux on zSeries)</i> • <i>PDA</i> • <i>i5/OS</i> • <i>z/VSE</i> • <i>z/VM</i> 	<ul style="list-style-type: none"> • <i>Apple Mac OS X Server:</i> • <i>PowerPC</i> • <i>HP HP-UX: PA-RISC, Itanium</i> • <i>HP Tru64 UNIX: Alpha</i> • <i>HP OpenVMS: Alpha, Itanium</i> • <i>IBM AIX5L:</i>

		<ul style="list-style-type: none"> <i>NetWare</i> • <i>OpenBSD</i> • <i>OS/2 Warp</i> • <i>QNX</i> • <i>IRIX</i> • <i>Solaris</i> • <i>SunOS</i> • <i>SCO</i> <i>OpenServer</i> • <i>SCO</i> <i>UnixWare</i> • <i>Tru64</i> • Visos <i>Windows</i> versijos 		<ul style="list-style-type: none"> <i>IBM POWER</i> • <i>IBM z/OS: zSeries</i> • <i>Linux: x86, x86-64, PowerPC, zSeries, Itanium</i> • <i>Microsoft Windows: x86, x86-64, Itanium</i> • <i>Sun Solaris: SPARC, x86, x86-64</i>
Licenzija	<p>Standartinės versijos kainos prasideda nuo 5999\$ už vieną procesorių arba 1849\$ su 5 CALs. <i>Enterprise</i> versijos kainos prasideda nuo 24999\$ vienam procesoriui arba 13969\$ su 25 CALs. Taip pat veikia <i>MSDN</i> akademinis aliansas, kuriame universitetų studentai gali būti užregistruojami bei gali nemokamai parsisiųsti <i>Microsoft</i> programinę įrangą.</p>	<p><i>MySQL</i> serveris platinamas nemokamai pagal <i>GNU General Public License (GPL)</i> licenziją. Taip pat yra mokama <i>MySQL Enterprise</i> versija, kurios kaina svyruoja nuo 479€ iki 3999€ metams vienam serveriui, priklausomai nuo versijos.</p>	<p>Egzistuoja viso keturios <i>IBM DB2</i> serverio versijos - trys mokamos: <i>Express</i>, <i>Workgroup</i> ir <i>Enterprise</i>, kurių kainos yra 4874\$ vienam procesoriui arba 165\$ vienam vartotojui <i>Express</i> versijoje, bei ketvirtoji nemokama: <i>Express-C</i> versija.</p>	<p><i>Oracle Database Enterprise Edition</i> kaina vienam procesoriui 40000\$, <i>Oracle Database Standart Edition</i> kaina vienam procesoriui 15000\$, <i>Oracle Database Standart Edition One</i> kaina vienam procesoriui 4995\$.</p>
OLAP	<p><i>SQL Server 2005</i> turi analizės paslaugas (angl. <i>Analysis Services</i>), kurios turi visus reikalingus <i>OLAP</i> įrankius.</p>	<p><i>MySQL</i> serveris neturi integruotų <i>OLAP</i> įrankių, tačiau egzistuoja daug papildomų programų bei įskiepių, dirbančių su šiuo serveriu bei leidžiančių atlikti <i>OLAP</i> analizę.</p>	<p><i>IBM DB2 Data Warehouse Enterprise Edition</i> versija turi integruotus įrankius, skirtus dirbti su <i>ETL (Extract, Transform, and Load - išskleisti, transformuoti ir užkrauti)</i>, <i>Data mining</i> (duomenų gavyba), <i>OLAP</i> ir <i>in-line</i> analize.</p>	<p><i>Oracle Database 11g Enterprise Edition</i> turi įrankį, pavadintą <i>Oracle OLAP</i>. Šis įrankis yra skaičiavimų variklis, kuris turi visą spektrą išplėstinių analitinių aplikacijų įskaitant planavimo, biudžetinimo (angl. <i>budgeting</i>), prognozavimo, pardavimų ir marketingo.</p>

Microsoft SQL Server 2005 šiuo metu yra labai populiarus, pažangus bei galingas duomenų bazių kūrimo programinis įrankis. Šiame pakete yra mano magistrinio darbo temai analizuoti reikalingi intelektualūs duomenų analizės bei kubų kūrimo įrankiai. Kauno Technologijos Universitetas bendradarbiauja su *Microsoft* korporacija, todėl savo studentus gali užregistruoti *MSDN* akademinio aljanso duomenų bazėje [12], ko pasekoje studentams suteikiamas vartotojo vardas bei slaptažodis, kuriuo jie gali prisijungti prie sistemos. Čia

jiems nemokamai bei legaliai prieinama didžioji dauguma *Microsoft* komercinių programinės įrangos produktų. Dėl šių priežasčių mano magistrinio darbo realizacija bus atliekama būtent *Microsoft* programomis. 1.8 paveiksle matome *MSND* svetainės langą su prieinamos legalios programinės įrangos sąrašu.

1.8 pav. *MSND* svetainėje prieinamos legalios programinės įrangos sąrašas

1.6. *Microsoft SQL Server 2005* apžvalga

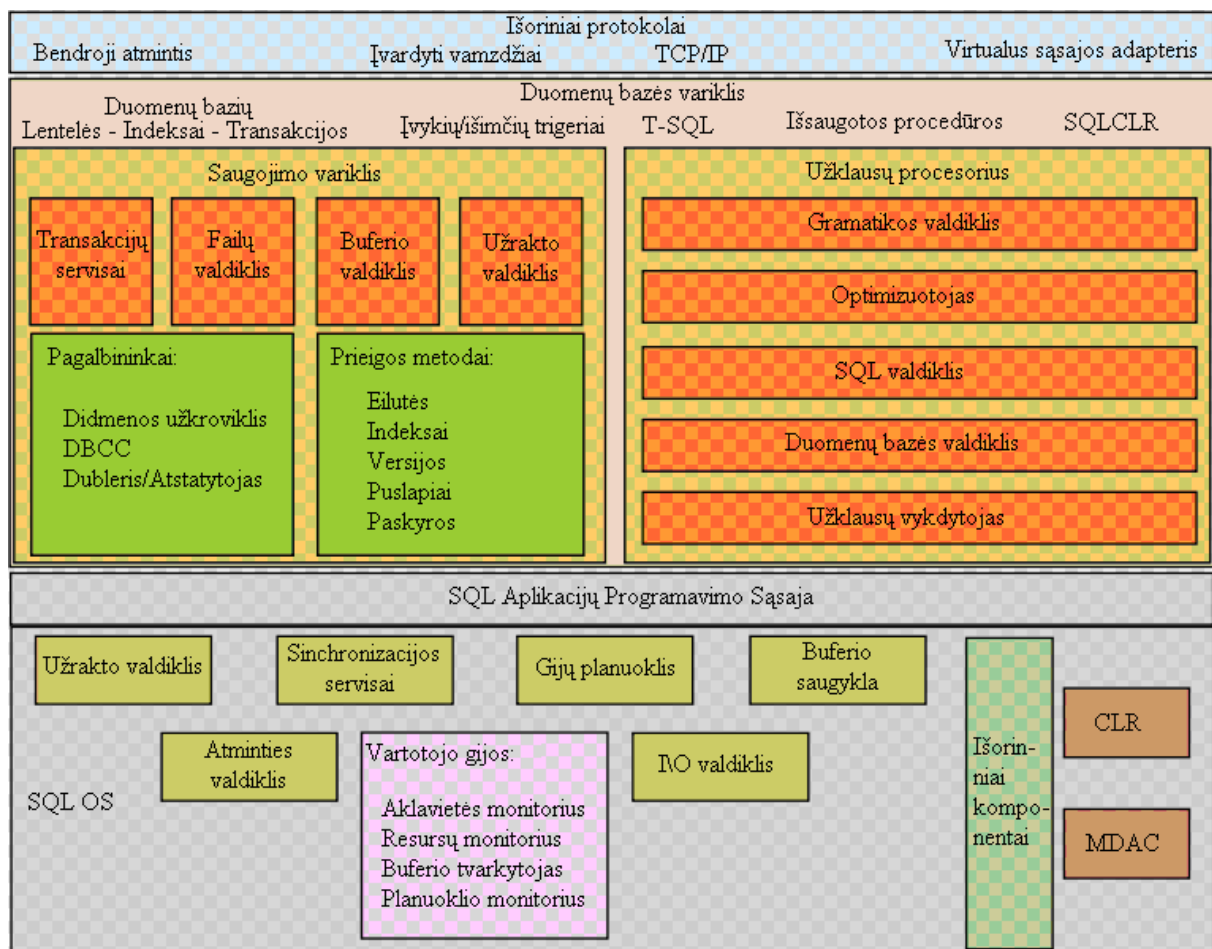
Microsoft SQL Server 2005 yra reliacinė duomenų bazių valdymo sistema (*RDBMS*). Jos pirminė užklausų (angl. *query language*) kalba yra *Transact-SQL*, kuri yra *ANSI/ISO* standarto struktūrizuotos užklausų kalbos (angl. *Structured Query Language – SQL*) įgyvendinimas [13].

Egzistuoja keletas skirtingų *Microsoft SQL Server 2005* programų paketo versijų [14]: *SQL Server 2005 Enterprise Edition* – pati didžiausia ir galingiausia versija, turinti maksimalų kiekį įrankių bei galimybių, *SQL Server 2005 Standard Edition* – versija, skirta mažoms bei vidutinėms įmonėms, *SQL Server 2005 Workgroup Edition* – versija, skirta duomenų valdymo sprendimams mažoms įmonėms bei darbo grupėms, *SQL Server 2005*

Express Edition – nemokamai platinama versija, neturinti didelio funkcionalumo, *SQL Server 2005 Compact Edition* – versija, skirta kurti mobilioms, darbalaukio įterptinėm (angl. *Embeddable*) ir mažasvorėms programoms, ir *SQL Server 2005 Developer Edition* – versija, skirta mokomiesiems tikslams dirbti su duomenų bazėmis ir turi tokį patį funkcionalumą, kaip ir *enterprise* versija. Taip pat egzistuoja *Standard, Enterprise, and Developer editions* 64-bitų *SQL Server* versijos.

1.6.1. Architektūra

Microsoft SQL 2005 serverio architektūra susideda iš trijų pagrindinių dalių: *SQL OS*, kuriame realizuoti pagrindiniai *SQL* serverio servisi, įskaitant gijų planavimą (angl. *thread scheduling*), atminties valdymą ir *I/O* valdymą; reliacinio variklio, kuriame realizuoti reliacinės duomenų bazės komponentai, įskaitant suderinamumą su duomenų bazėmis, lentelėmis, užklausomis ir serverio procedūromis; ir protokolo sluoksnio (angl. *Protocol Layer*), kuris atskleidžia *SQL* serverio funkcionalumą. *SQL* serverio architektūra pateikta 1.9 paveiksle.



1.9 pav. *SQL* serverio architektūra [15]

1.6.2. Paslaugos (Services)

Microsoft SQL Server 2005 turi nemažai pridedamųjų (angl. *add-on*) paslaugų. Jos nėra privalomos ir nesusiję su duomenų bazės veikimu, tačiau jos turi didelę reikšmę prisidedant prie duomenų bazės valdymo sistemos. Šios paslaugos vykdomos kaip kai kurių *SQL* serverio komponentų dalis arba kaip atskiros *Windows* operacinės sistemos paslaugos, turinčios savo nuosavas aplikacijų programavimo sąsajas (angl. *API (Application programming interface)* - aplikacijų programavimo sąsaja). 1.10 paveiksle parodyti *Microsoft SQL Server* komponentai.

- **Service Broker** – paslaugų tarpininkas, kuris vykdomas kaip duomenų bazės variklio dalis ir tiekia patikimą susirašinėjamą bei žinučių eiliškumo platformą *SQL* serverio programai. Pavyzdžiui yra naudojamas suteikti asinchroninio programavimo aplinką.
- **Replication Services** – replikavimo paslaugos yra naudojamos replikuoti ir sinchronizuoti duomenų bazės objektus, apimant visumą ar tik tų objektų grupę, per replikavimo agentus, kurie gali būti kiti duomenų bazių serveriai tinkle, ar duomenų bazės kliento pusėje. Replikavimas atitinka leidėjo/prenumeratoriaus (angl. *publisher/subscriber*) modelį, pvz., pakeitimai, išsiųsti vieno duomenų bazės serverio (leidėjo) ir yra gaunami kitų serverių (prenumeratorių). *SQL Server* palaiko tris skirtingus replikavimo tipus:
 - **Transaction replication** – transakcijos replikavimas. Kiekviena transakcija, sukurta leidėjo duomenų bazėje yra sinchronizuojama su prenumeratorių duomenų bazių serveriais, kurie šia transakcija atnaujina savo duomenų bazes.
 - **Merge replication** – suliejimo replikavimas. Pakeitimai, padaryti tiek leidėjo, tiek prenumeratoriaus serveriuose yra sekami ir šie pakeitimai yra periodiškai sinchronizuojami tarpusavyje šių serverių. Jeigu tie patys duomenys yra skirtingai modifikuojami abiejuose leidėjo ir prenumeratoriaus serveriuose, sinchronizacija aptiks konfliktą, kuris turi būti išspręstas – arba rankiniu būdu, arba iš anksto apibrėžtais veiksmiais.
 - **Snapshot replication** – momentinis replikavimas. Momentinis duomenų replikavimas kopijuoja visos duomenų bazės kopiją ir replikuoja ją prenumeratoriams.

- **Analysis Services** – analizės paslaugos, turinčios suderinamumą su *OLAP* ir duomenų gavybos technologijom *SQL Server* duomenų bazėse. *OLAP* variklis suderinamas su *MOLAP*, *ROLAP* ir *HOLAP* duomenų saugojimo modeliais. Analizės servिसai suderinti su *XML* analizės (*XML for Analysis*) standartu kaip pagrindiniu komunikacijos protokolu. Kubo duomenys gali būti pasiekiami naudojant *MDX* užklausas. Duomenų gavybos specifinis funkcionalumas palaikomas *DMX* užklausų kalba. Analizės servिसai apima įvairius algoritmus – sprendimų medžius (angl. *Decision trees*), grupavimo algoritmus (angl. *clustering algorithm*), *Naive Bayes* algoritmą, laiko eilių analizę, sekų grupavimo algoritmą ir neuronų tinklus – naudojamus duomenų gavyboje
- **Reporting Services** – ataskaitų paslaugos – tai ataskaitų generavimo terpė. Ji administruojama per *web* sąsają. Ataskaitų servिसų *web* sąsaja suderinama su skirtingom ataskaitų plėtojimo aplikacijom. Ataskaitos kuriamos *RDL* failais, kurie gali būti konvertuojami į įvairius paplitusius formatus, tokius kaip *Excel*, *PDF*, *CSV*, *XML*, *TIFF* (taip pat kitais paveiksliukų formatais) ir *HTML*.
- **Notification Services** – pranešimų paslaugos, skirtos kurti pranešimus, kurių pagrindas yra užsakymai (*subscription based*) apie įvykius. Šiuos servिसus galima atsisiųsti ir naudoti su senesnėmis versijomis. *Subscription Management Objects* – tai *API* (angl. *advanced programming interface*) skirta kurti užsakymų valdymo programoms. Įvykiai gali būti iš keturių skirtingų šaltinių : valdoma *API*, *COM* (angl. *Common Object Model*) *API*, *XML API* arba procedūrinė *API*. Pranešimai siunčiami *XSL* formatu, todėl juos gali priimti įvairūs įrenginiai. Servिसų kopijos apibrėžiamos *XML* konfigūracinėje byloje, o duomenys saugomi *SQL* duomenų bazėje pagal šabloną *kopijos_vardasNSMain*. Paslaugos realizuotos pasinaudojant standartinėmis Microsoft Windows paslaugomis ir yra identifikuojamos pagal šabloną *NS\$kopijos_vardas*. Kiekviena iš kopijų gali kontroliuoti daug programinių produktų, kurie apibrėžti *XML* byloje (*application definition file - ADF*) ir saugo informaciją apie įvykius, užsakymus ir pranešimus.
- **Integration Services** – integravimo paslaugos, kurios skirtos integruoti duomenims iš skirtingų tipų duomenų šaltinių. Integravimo paslaugos turi savo grafinius vartotojo sąsajos įrankius, skirtus atlikti tokius duomenų integravimo funkcionalumus kaip duomenų išskleidimas iš įvairių duomenų šaltinių, duomenų užklauskimas, duomenų transformavimas įskaitant agregavimą, duplikavimą ir suliejimą, tuomet duomenų užkrovimą į kitus šaltinius, o taip

pat elektroninio pašto žinučių siuntimas, kuriose informuojama apie atliekamas operacijas.

- **Full Text Search Service** - pilnatekstė paieška skirta ieškoti duomenų pagal duotą raktinę frazę tekstiniuose duomenyse. Dabar galima vykdyti užklausas prijungtose tarnybinėms stotims (angl. *linked servers*). Jei anksčiau be didesnių sunkumų buvo galima naudoti vieną arba tik visus iškart stulpelius, dabar lengvai pasirenkame tuos stulpelius, kurių reikia paieškai. Paieška gali būti vykdoma skirtinga nei nurodyta pagal nutylėjimą stulpeliui kalba (*LocaleID*). Pilnateksčiai katalogai lengvai archyvuojami ir atstatomi su duomenimis ar be jų. Galima naudoti *attach* ir *detach* operacijas. Pilnateksčiai katalogai gali būti sukurti iš *XML* duomenų. Labai padidintas pilnateksčių indeksų našumas.



1.10 pav. Microsoft SQL Server komponentai [16]

1.6.3. Reikalavimai sistemai

Pagrindiniai techniniai reikalavimai sistemai yra tokie:

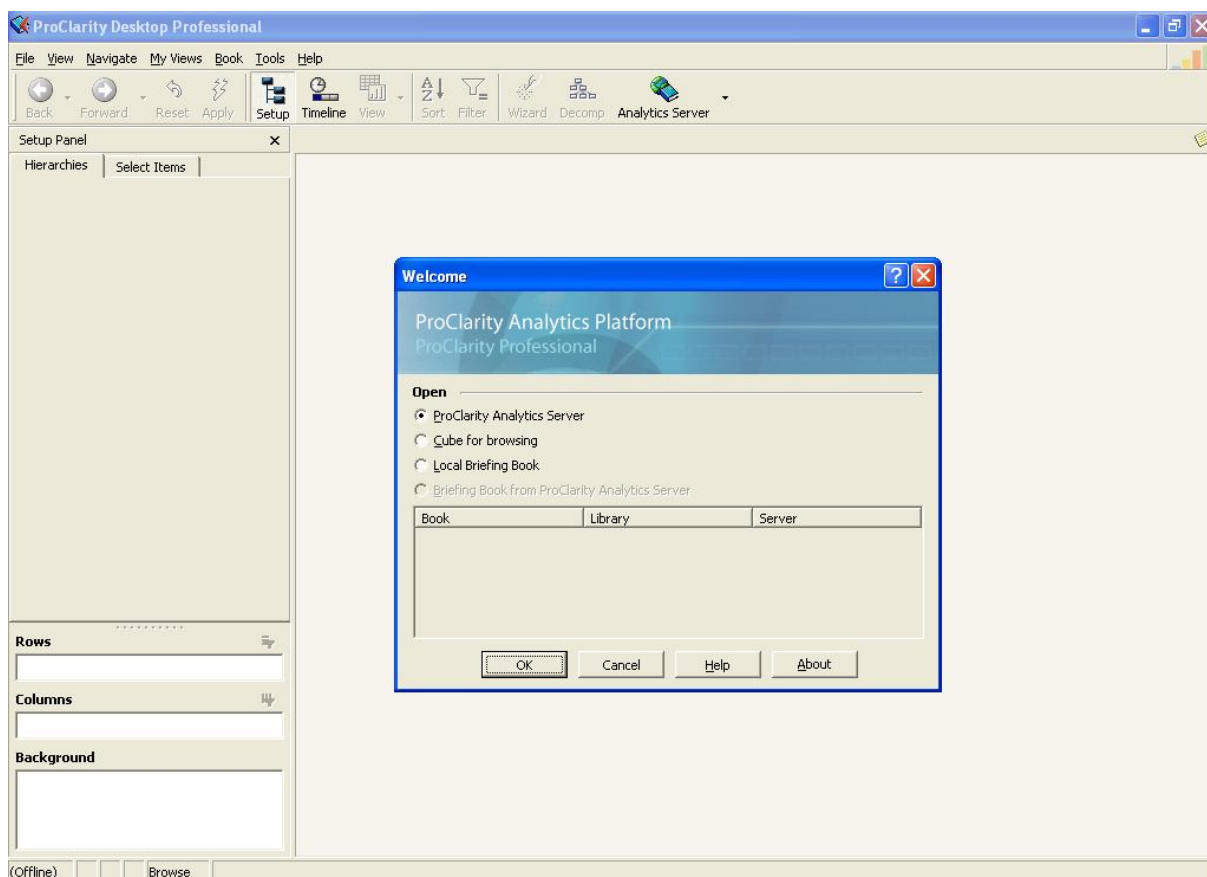
- *Pentium III* arba geresnis (bent 1GHz taktinio dažnio procesorius rekomenduojamas)
- 192MB RAM (512MB rekomenduojama)
- 525MB kietojo disko atminties
- *Windows 2000 (SP4)*, *Windows XP (SP2)* arba *Server 2003 (SP1)*
- *Net Framework 2.0*

- *Windows Installer 3.0*

1.7. *ProClarity Desktop Professional 6.3* apžvalga

ProClarity Analytics organizacijoms teikia galingus bei lengvai naudojamus programinius įrankius, kurie apima viską nuo *ad-hoc* užklausų iki sudėtingų analizės modelių [17]. Šis programinis sprendimas prisitaiko prie unikalių verslo ir techninių reikalavimų, praplėsdamas *Microsoft* intelektualaus verslo (*Business intelligence*) platformą paprastu duomenų prieigos sprendimu *SQL Server* analizės paslaugoms (2000 ir 2005 versijoms). 1.11 paveiksle pateiktas paleistos *ProClarity Desktop Professional 6.3* programos pirminis langas.

ProClarity supaprastina tai, kaip žmonės gauna duomenis bei bendrauja su jais – padedant tašką ataskaitų generavimo problemoms. Informacinės sistemos gali suteikti žmonėms, priimantiems sprendimus, turtingesnę, labiau koncentruotą informaciją, o galutiniai vartotojai gali laisvai tirti besikeičiančios informacijos savybių problemas patys.



1.11 pav. Paleistos *ProClarity Desktop Professional 6.3* programos pirminis langas

Naudojantis *ProClarity*, analitikai sukuria analitinius pradžios taškus įskaitant grafikus, pagrindinius vykdymo indikatorius (*KPIs*), duomenų lenteles ir pažangesnius

vizualizacijos būdus, tokius kaip dekompozicijos medis (*Decomposition Tree*), charakteristikų žemėlapis (*Performance Map*) ir perspektyvinis vaizdas (*Perspective View*). Šie vizualizacijos įrankiai leidžia greitai identifikuoti tendencijas, galimybes ir grėsmes esančias dideliuose duomenų kiekiuose. Tokie pradžios taškai yra išdėstyti keliais lengvai naudojamais kliento tipo programiniais paketais, suprojektuotais patenkinti specifinius vartotojo poreikius, įskaitant tinklo (*Web*) klientus, portalus, ataskaitas, publikacijas ir taškų lenteles.

ProClarity Analytics, kuris dabar yra *Microsoft* intelektualaus verslo (*Business intelligence*) koncepcijos vienas pagrindinių komponentų, teikia unikalius įrankius kurie padeda verslo sprendimų kūrėjams ištyrinėti didelius kiekius duomenų greitos išvalgos būdu. *ProClarity* supaprastina galutinio vartotojo *UDM* navigaciją parodant ryšius tarp dimensijų hierarchijų, atributų hierarchijų ir matų. Tai atliekama per konteksto filtrus, kurie sutelkia informaciją, kurią gauną galutiniai vartotojai. *ProClarity Analytics* yra glaudžiai integruotas su *Microsoft Office* paketu, įskaitant *Excel*, *PowerPoint* ir *Exchange* programas. Organizacijoje galutiniai vartotojai gali lengvai siųsti elektroniniu paštu savo išvalgas, įterpti interaktyvias diagramas į *PowerPoint* programą arba kurti *Excel* programos pagrindu aplikacijas įskaitant *ProClarity Analytics*.

ProClarity, sprendimus priimantiems žmonėms teikia unikalius įrankius, kurie leidžia greitos išvalgos būdu iširti didelius kiekius duomenų. Paprasta duomenų navigacija, galingi skaičiavimai bei patentuoti pažengę vizualizacijos įrankiai yra pagrindiniai komponentai padarantys *ProClarity Analytics* geriausiu atsakant į besikeičiančio verslo klausimą „Kodėl?“.

1.7.1. Reikalavimai sistemai

Pagrindiniai techniniai reikalavimai sistemai yra tokie:

- Procesorius: *Pentium* šeimos;
- Operacinės sistemos: *Windows 2000*, *Windows XP Professional*, *Windows Server 2003*;
- Atmintis: 128MB *RAM* atminties.

1.8. Siekiamas sprendimas

Šiuo darbu bus siekiama sukurti informacinę sistemą, kurios pagrindas yra *Microsoft SQL Server 2005* programine įranga sukurta KTU duomenų bazė. Šioje bazėje bus saugomi visi su universitetu susiję duomenys: studentų, dėstytojų, modulių, pareigų ir kita informacija. Pasinaudojant šiais duomenimis bus kuriami vaizdiniai, faktinės bei dimensinės lentelės,

kurių pagrindu galiausiai kuriami duomenų kubai. Duomenų kubai leis sukinių kūrimo pagalba analizuoti duomenų bazėje esančius duomenis įvairiais pjūviais, ko pasekoje galėsime atlikti išsamią intelektualią duomenų analizę. *ProClarity Desktop Professional 6.3* programinė įranga bus taip pat naudojama analizuojant duomenų kubais sukurtą informaciją.

1.9. Analitinės dalies išvados

- Naudojant aprašomosios literatūros analizės ir apibendrinimo metodus, apibrėžta kubo sąvoka, jo struktūra bei elementai, išnagrinėtas konkretus pavyzdys, atskleidžiantis jo teikiamas galimybes bei naudą.
- Siekiant išsiaiškinti sistemos įvairiapusiškas vartojimo galimybes, aptarti galimi sistemos vartotojai, aprašytos jų paskirtys sistemoje bei tikslai.
- Tyrimo objekto analizės metu, naudojantis įvairia užsienio ir lietuvių autorių literatūra, apžvelgti dažniausiai naudojami intelektualios duomenų analizės kūrimo programinės įrangos paketai. Atlikta apžvalga suteikė galimybę pasirinkti kuriamam darbui labiausiai tinkantį programinės įrangos paketą.
- Detaliai išnagrinėtas pasirinktas programinės įrangos paketas – *Microsoft SQL Server 2005*, aptarta jo architektūra, paslaugos bei techniniai reikalavimai sistemai. Apžvelgta papildomai pasirinkta programinė įranga *ProClarity Desktop Professional 6.3*, kuri išplečia duomenų kubų analizės galimybes.

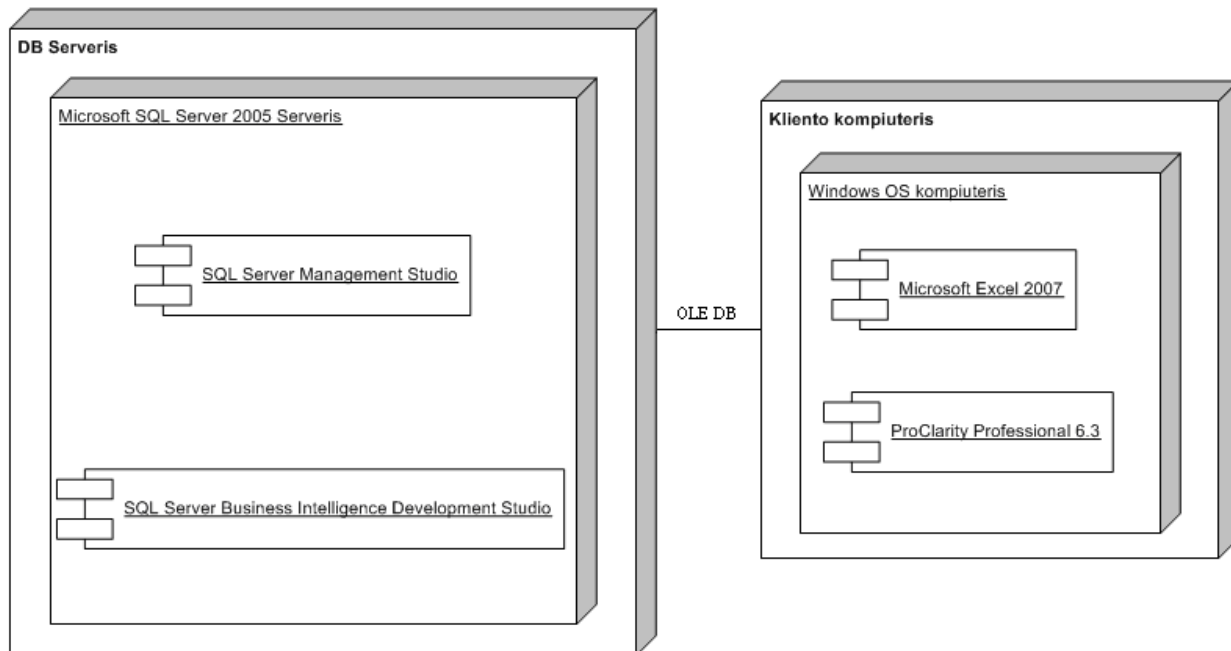
2. SISTEMOS PROJEKTAVIMO ESMINIAI SPRENDIMAI

2.1. Projektinės dalies tikslas

Projektinės dalies tikslas yra numatyti projektuojamos sistemos esminius aspektus. Tai yra suprojektuoti KTU palikiminės duomenų bazės duomenimis paremtą šiuolaikinę duomenų bazę, veikiančią *Microsoft SQL Server 2005* programiniame pakete. Tiksliai apibrėžti sistemos komponentų tarpusavio sąveiką bei ryšius. Aprašyti kiekvieno sistemoje esančio vartotojo galimus veiksmus ar jų sekas. Aprašyti esminius sistemos kūrimo algoritmus, numatyti sistemos trūkumus ar apribojimus.

2.2. Bendra sistemos architektūra

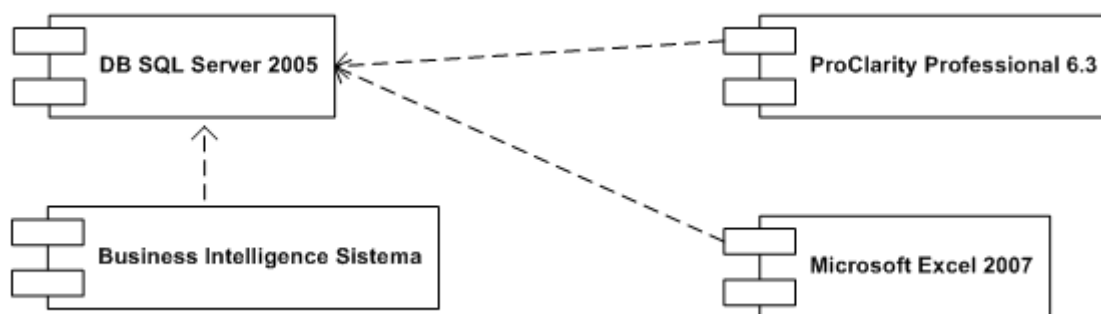
Bendrą sistemos architektūrą aiškiausiai apibrėžia įrangos diagrama, pateikta 2.1 paveiksle. Įrangos diagrama parodo sąryšį tarp sistemos programinės įrangos ir aparatūrinės įrangos.



2.1 pav. Projektuojamos sistemos įrangos diagrama

Kaip matome iš nubraižytos diagramos, sistemą sudaro duomenų bazės serveris bei kliento kompiuteris. Jie tarpusavyje komunikuoja *OLE DB* protokolu. Duomenų bazės serveris sukurtas *Microsoft SQL Server 2005* programinės įrangos pagrindu. Jis turi du

pagrindinius projektuojamoje sistemoje naudojamus komponentus: *SQL Server Management Studio* ir *SQL Server Business Intelligence Development Studio* įrankius. Pirmasis skirtas administruoti duomenų bazę, tvarkyti duomenų lenteles, ryšius, administruoti vartotojus ir t.t. Antrasis įrankis yra skirtas duomenų analizei. Jo pagalba kuriamos dimensijos, matai bei kubai. Kliento kompiuteris aprūpintas *ProClarity Professional 6.3* bei *Microsoft Excel 2007* programine įranga. Duomenų bazės serverio ir kliento kompiuterio komponentų tarpusavio ryšius parodo komponentų diagrama, pateikta 2.2 paveiksle.



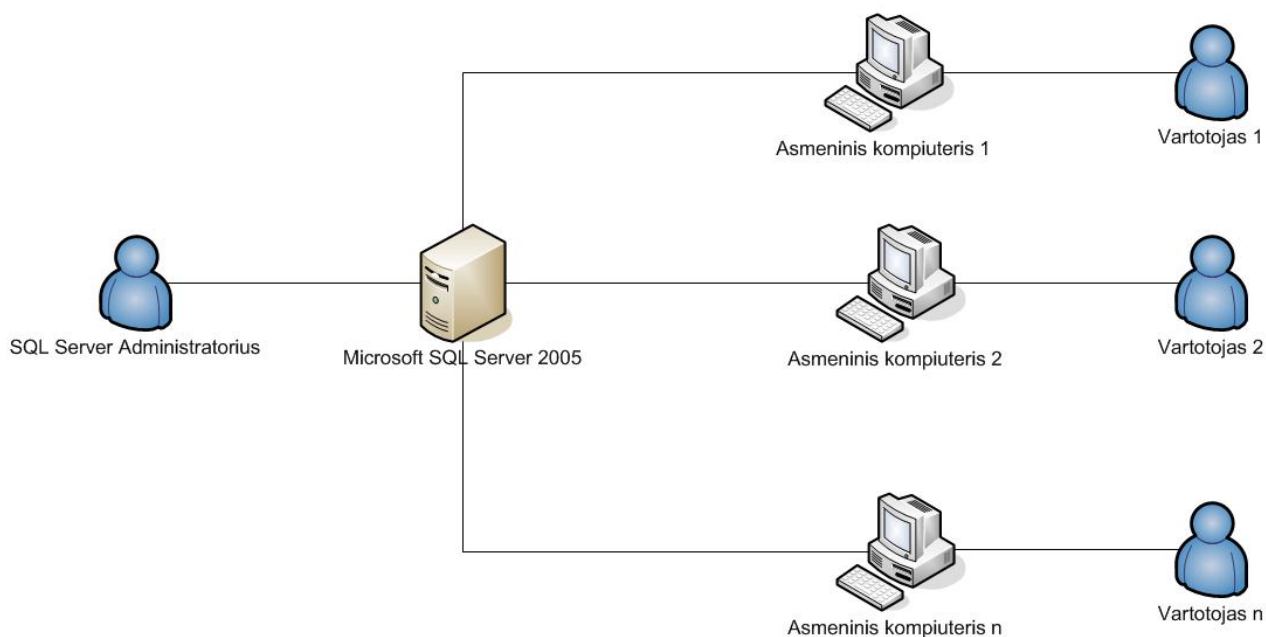
2.2 pav. Projektuojamos sistemos komponentų diagrama

Komponentų diagrama parodo sistemos programinės įrangos komponentus ir sąryšius tarp jų. Kaip matome iš diagramos, sistemoje svarbiausią vaidmenį atlieka duomenų bazės serveris. *Business Intelligence* sistema yra *SQL* Serverio integruotas komponentas, todėl tai yra glaudžiai susijusi programinė įranga. *ProClarity* bei *Microsoft Excel* programos naudojamos tik kliento kompiuteryje. Klientas naudodamas šias programas jungiasi prie *SQL Server 2005* programinės įrangos bei gauna priėjimą prie reikalingų atlikti analizei duomenų.

Projektuojama sistema veikia tinkle ir yra kuriama kliento – serverio architektūros būdu. Tokios architektūros didžiausias privalumas yra tas, kad turint vieną serverį, galime turėti praktiškai neribotą skaičių klientų. Klientų skaičiaus ribojimas atsiranda tik dėl techninių sistemos pajėgumų aptarnauti klientus bei dėl tinklo apkrovos ribotumo. Sistemos tinklinė principinė schema pavaizduota 2.3 paveikslo diagramoje.

Kliento dalis – tai lokalus asmeninis kompiuteris su Windows šeimos operacine sistema bei specialia programine įranga - *ProClarity Professional 6.3* bei *Microsoft Excel 2007*.

Serverio dalis – tai serveris, kuriame instaliuota *Microsoft Server 2003* operacinė sistema bei *SQL Server 2005* programų paketas.



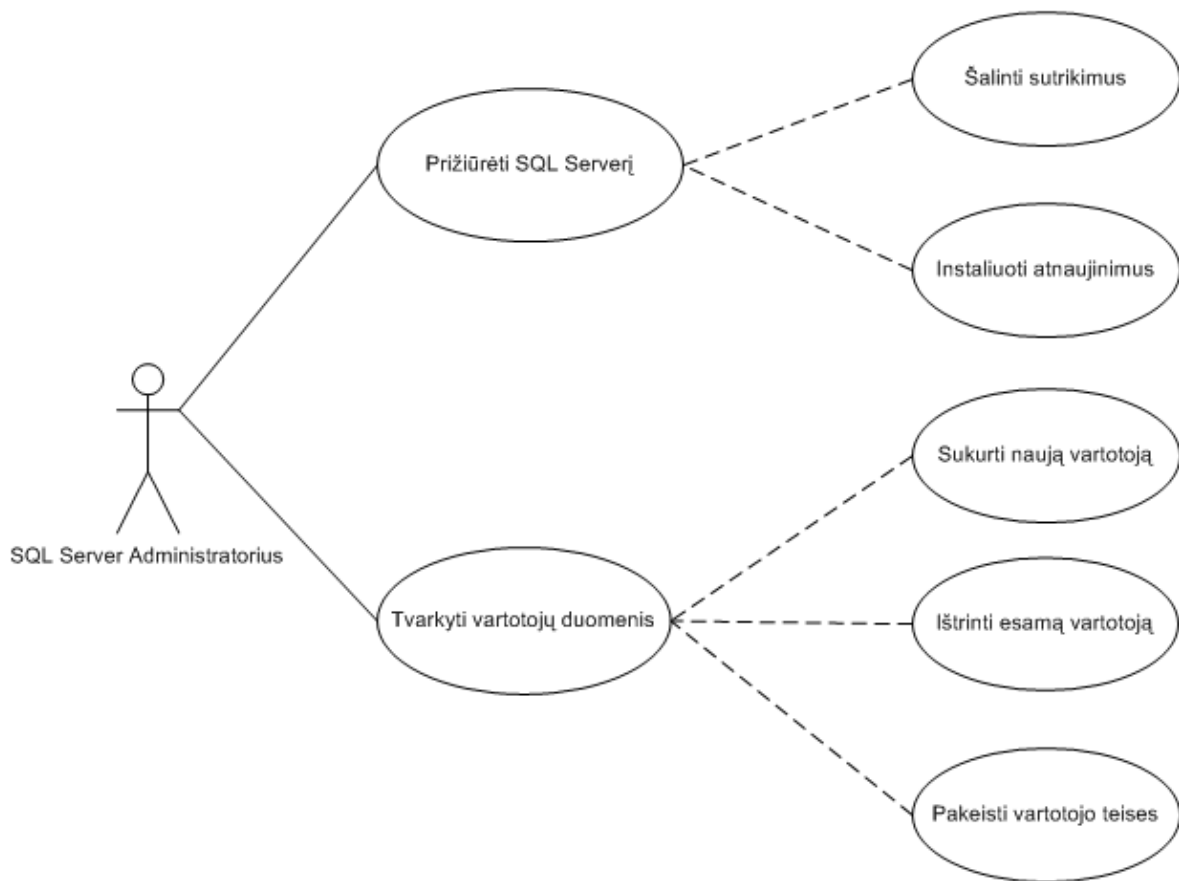
2.3 pav. Principinė tinklo schema

Pats svarbiausias sistemoje žmogus yra *SQL Server* administratorius. Nuo jo priklauso visos sistemos darbas. Sistemos vartotojais gali būti tiek analitikai, tiek duomenų bazės administratoriai. Plačiau apie kiekvieno iš jų veiksmus bei galimybes aptarsime nagrinėdami kiekvieno jų panaudojimo atvejus.

2.3. Panaudojimo atvejai

Kaip jau aptarėme analizėje, sistemoje egzistuos trys vartotojų tipai: *SQL Server* administratorius, duomenų bazės administratorius bei analitikas. Kiekvienas iš jų gali atlikti tik jam priskirtus leidžiamus veiksmus. Panaudojimo atvejų diagrama specifikuoja aukšto lygio vartotojo tikslus, kuriuos sistema turės įgyvendinti. Šie tikslai nebūtinai turi būti užduotys ar veiksmai. Kitaip sakant, panaudojimo atvejų diagrama aprašo sistemos funkcionalumą.

Pirmoje 2.4 paveiksle pavaizduotoje diagramoje matome *SQL Server* administratoriaus panaudojimo atvejų diagramą.



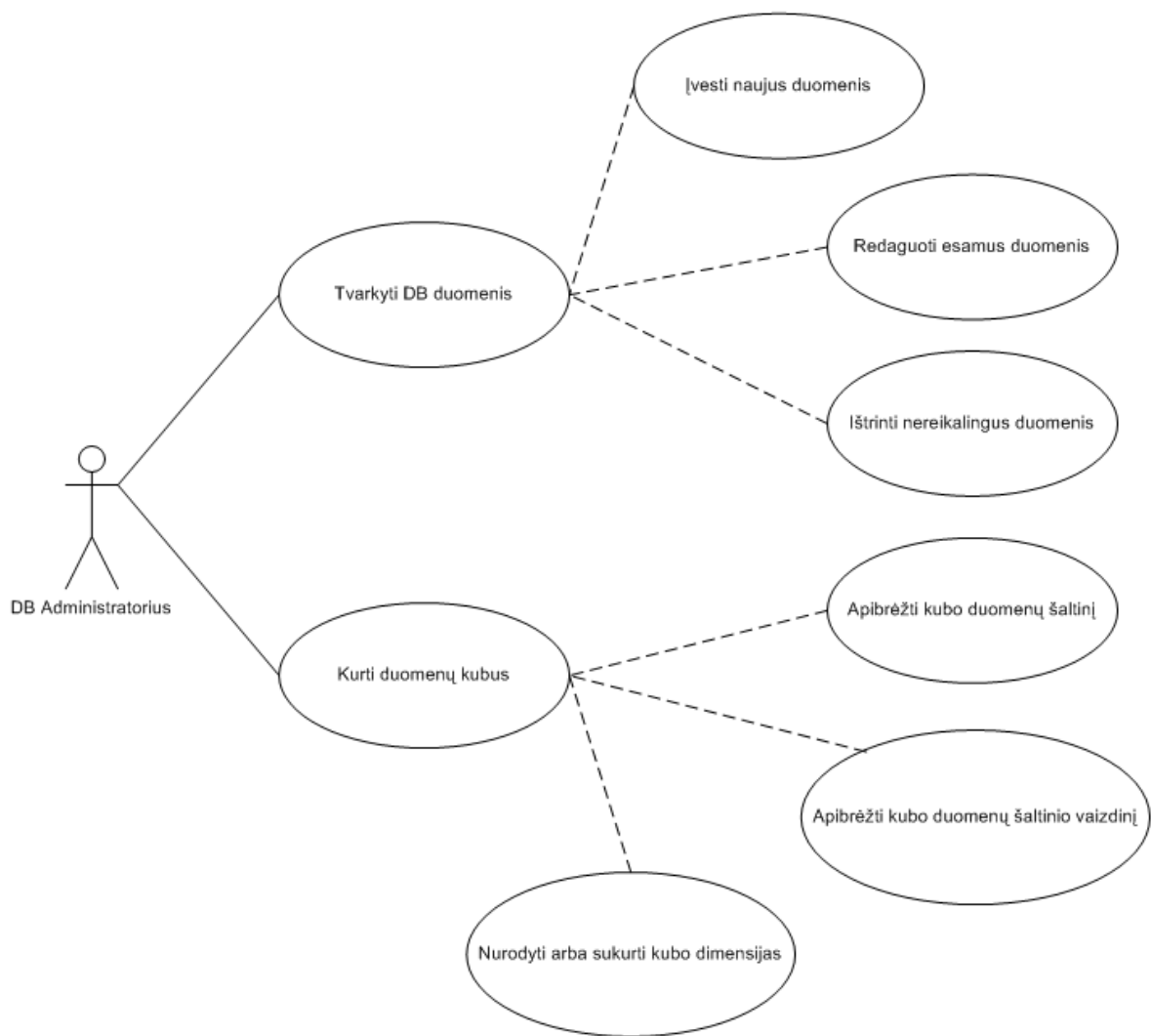
2.4 pav. *SQL Server* administratoriaus panaudojimo atvejų diagrama

Kaip matome iš diagramos, *SQL Server* administratorius turi dvi pagrindines užduotis: prižiūrėti *SQL* serverį bei tvarkyti vartotojų duomenis.

Prižiūrėti *SQL Serverį* – tai labai plati sąvoka, apimanti daug veiksmų. Visų pirma, tai neapsiriboja vien *SQL Server 2005* programinės įrangos priežiūra, o taip pat apima ir serverio operacinės sistemos priežiūrą. Sąvoka „priežiūra“ reiškia, kad *SQL Server* administratorius atsakingas už tinkamą visos sistemos konfigūravimą, atnaujinimų, atsakingų už sistemos saugumą, laiku instaliavimą, greitą sutrikimų bei gedimų šalinimą bei bendrą sklandų sistemos darbą.

Tvarkyti vartotojų duomenis – apima vartotojų, turinčių galimybę prisijungti prie *SQL Server 2005* administravimą. Administratorius atsakingas už naujų vartotojų sukūrimą, nereikalingų vartotojų šalinimą bei tinkamą šiems vartotojams teisių priskyrimą sistemoje.

Antroje 2.5 paveiksle pavaizduotoje diagramoje matome duomenų bazių administratoriaus panaudojimo atvejų diagramą.



2.5 pav. DB Administratoriaus panaudojimo atvejų diagrama

Kaip matome iš diagramos, duomenų bazių administratorius turi dvi pagrindines užduotis: tvarkyti duomenų bazės duomenis bei kurti duomenų kubus.

Tvarkyti duomenų bazės duomenis – apima veiksmus, galimus atlikti su duomenų baze bei jos duomenimis: tai naujų duomenų įvedinėjimas, esamų duomenų tvarkymas bei redagavimas ir nereikalingų duomenų pašalinimas.

Kurti duomenų kubus – apima visus veiksmus, reikalingus sukurti duomenų kubui. Tai duomenų kubo duomenų šaltinio apibrėžimas, duomenų kubo šaltinio vaizdinio apibrėžimas, dimensijų, matų bei hierarchijų kūrimas, galiausiai pačio kubo kūrimas bei jo duomenų peržiūrėjimas.

Trečioje 2.6 paveikslo diagramoje matome analitiko panaudojimo atvejų diagramą.



2.6 pav. Analitiko panaudojimo atvejų diagrama

Kaip matome iš diagramos, analitikas turi 4 pagrindines užduotis: Atvaizduoti duomenis *ProClarity* programoje, tvarkyti duomenis, kurti sukinius ir kurti ataskaitas.

Atvaizduoti duomenis *ProClarity* programoje – apima analitiko darbą *ProClarity* programine įranga. Tam reikia nurodyti prisijungimo prie domenų bazės parametrus bei nurodyti norimos diagramos vaizdavimo parametrus bei duomenų šaltinius.

Tvarkyti duomenis – apima veiksmus, galimus atlikti su duomenų baze bei jos duomenimis: tai naujų duomenų įvedinėjimas, esamų duomenų tvarkymas bei redagavimas ir nereikalingų duomenų pašalinimas.

Kurti sukinius – apima veiksmus, reikalingus atlikti norint sukurti sukinį: tai yra prisijungimo prie *SQL Server* parametrų nurodymas bei sukinio projektavimo veiksmai – reikiamų duomenų šaltinių išrinkimas, vaizdavimo formos parinkimas ir t.t.

Kurti ataskaitas – apima reikiamų ataskaitų kūrimą panaudojant duomenis, gautus analizės metu sukurtais duomenų sukiniais.

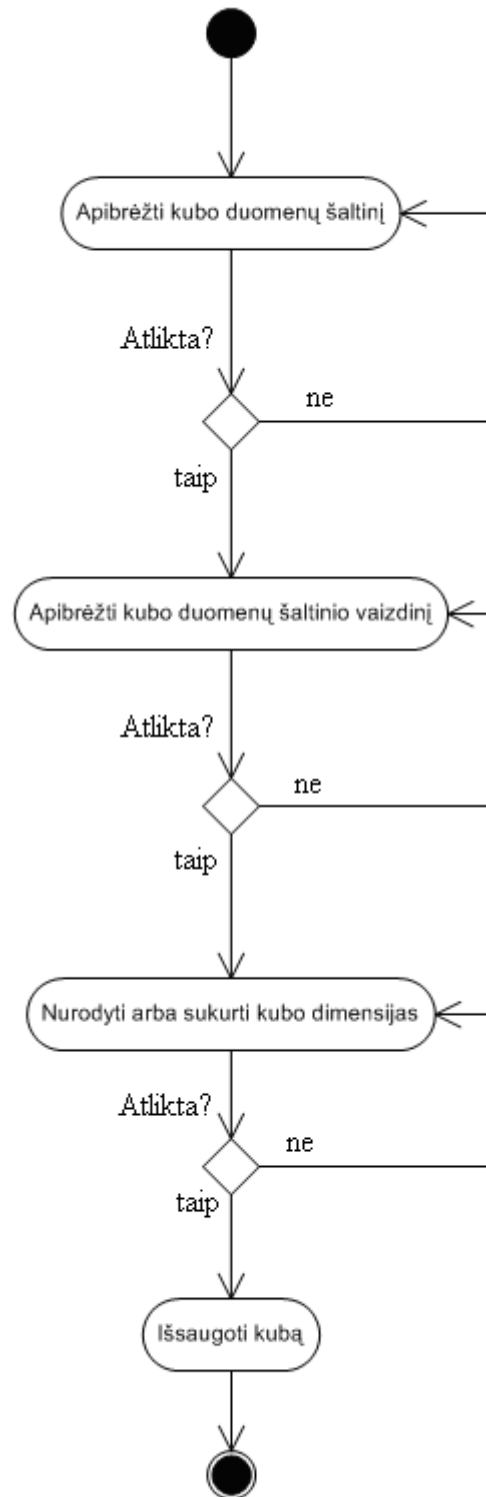
2.4. Programinės įrangos kūrimo bei darbo pagrindiniai algoritmai

Programinės įrangos kūrimo bei darbo pagrindinius algoritmus geriausiai aprašo veiklos bei sekų diagramos.

Veiklos diagramos parodo veiklas ir veiksmus, kurie nusako darbų sekas. Unifikuotoje modeliavimo kalboje (*UML*) veiklos diagramos atstovauja verslo bei žingsnis-po-žingsnio operacijų darbų sekas sistemoje. Kitaip sakant, veiklos diagrama parodo bendrą veiksmų seką.

Sekų diagramos parodo kaip objektai bendrauja tarpusavyje laike ir yra naudojamos sudėtingų vartojimo atvejų ar veiksmų detalizavimui. Vertikali ašis žymi laiką, o horizontalioje išdėstomi objektai.

Pirmiausiai panagrinėkime duomenų kubo kūrimo algoritmą. Šiam veiksmui atlikti geriausiai tinka duomenų kubo kūrimo veiklos diagrama, kuri pavaizduota 2.7 paveiksle.

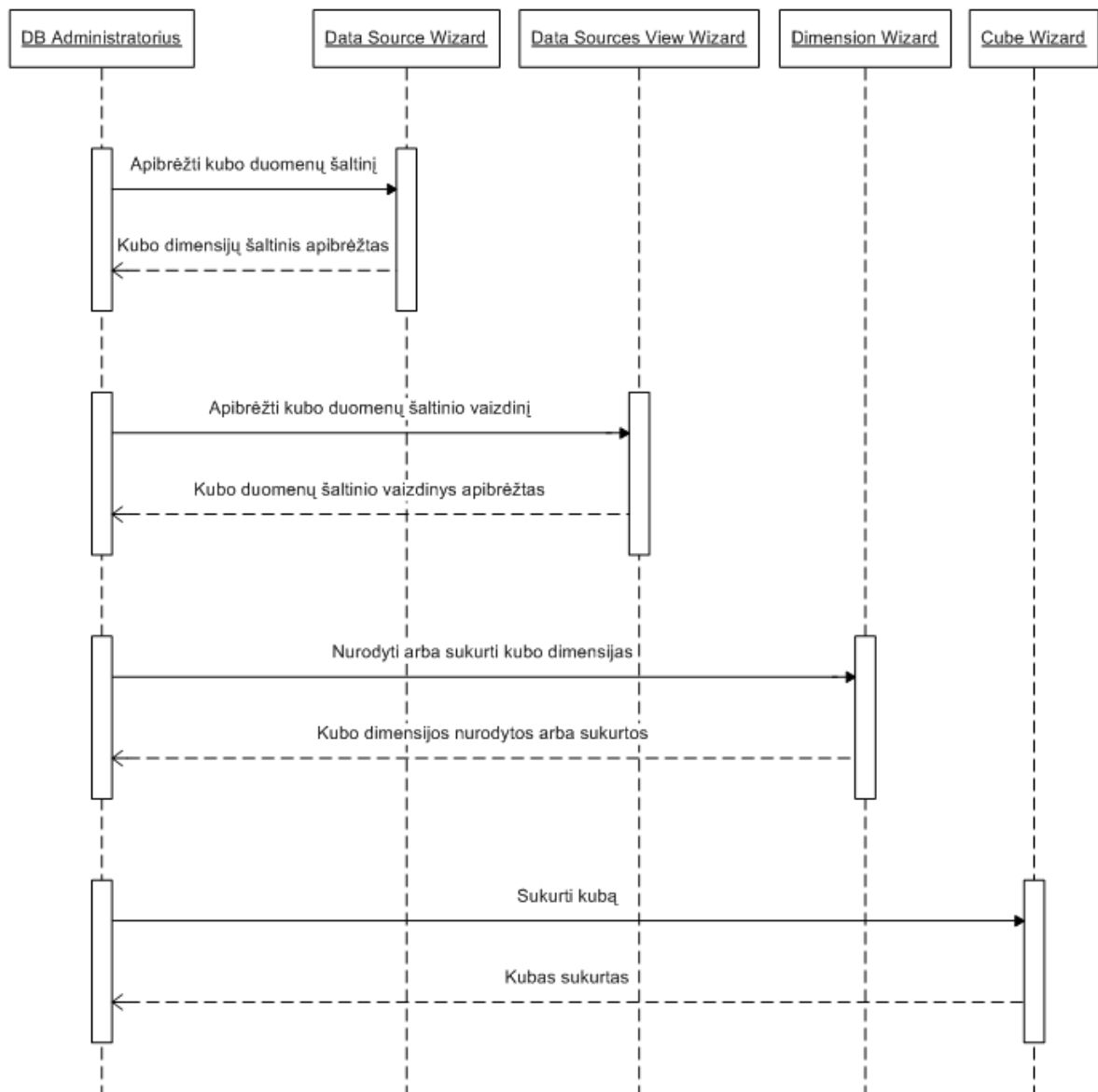


2.7 pav. Duomenų kubo kūrimo veiklos diagrama

Kubas kuriamas *SQL Server Business Intelligence Development Studio* programa. Kuriant kubą, pirmiausia reikia apibrėžti kubo duomenų šaltinį, kitaip tariant nurodyti prisijungimo parametrus prie *SQL Server 2005* bei duomenų bazės, kuri bus naudojama. Tam naudojamas *Data Source Wizard* įrankis. Esant nesėkmingam šios užduoties užbaigimui, veiksmas kartojamas kol bus įvykdytas kubo duomenų šaltinio apibrėžimas. Sėkmingai

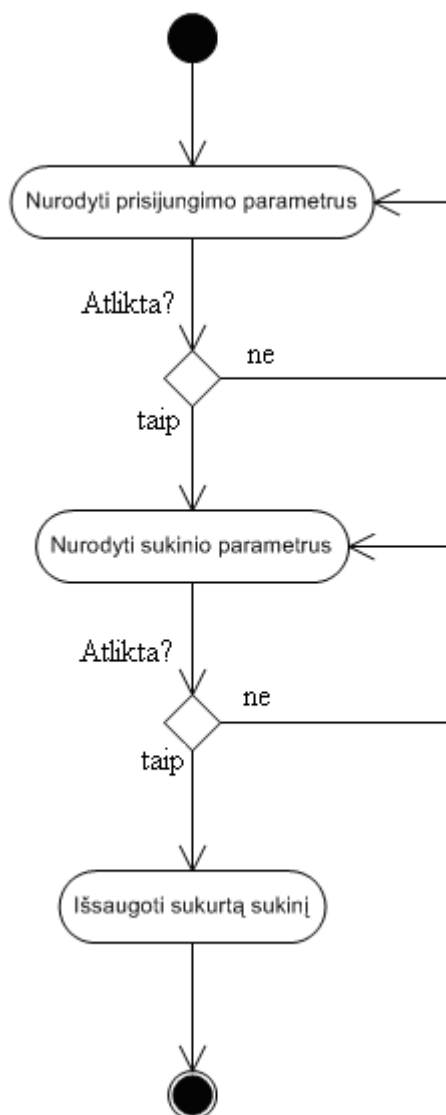
užbaigus veiksmą – vykdomas kitas žingsnis. Šiuo atveju tai. apibrėžiamas kubo duomenų šaltinio vaizdinys, kurio duomenimis bus kuriamas duomenų kubas. Šiam veiksmui įvykdyti naudojamas *Data Source View Wizard* įrankis. Kaip ir praeitame žingsnyje, esant nesėkmingai veiksmo pabaigai, veiksmų seka kartojama, o esant sėkmingai pabaigai – pereinama prie kito žingsnio. Dar kitas žingsnis yra naudoti *Dimension Wizard*, kurio pagalba nurodomos kubo dimensijos. Esant nesėkmingam veiksmui, grįžtama atgal ir kartojamas veiksmas iš naujo, o esant sėkmingai pabaigai, pereinama prie kito žingsnio – *Cube wizard*. Tai paskutinis įrankis kubo kūrimo procese. Įvykdžius sėkmingą kubo kūrimo procesą, gaunamas duomenų kubas, o esant nesėkmingam bandymui, kaip ir visus praeitus kartus, veiksmas kartojamas iš naujo.

Šie veiksmai taip pat atspindi duomenų kubo kūrimo sekų diagramoje, pavaizduotoje 2.8 paveiksle.



2.8 pav. Duomenų kubo kūrimo sekų diagrama

Dabar panagrinėkime sukinių kūrimo *Excel 2007* programa algoritmą. Šiam veiksmui atlikti geriausiai tinka sukinių kūrimo *Excel 2007* programa veiklos diagrama, pavaizduota 2.9 paveiksle.

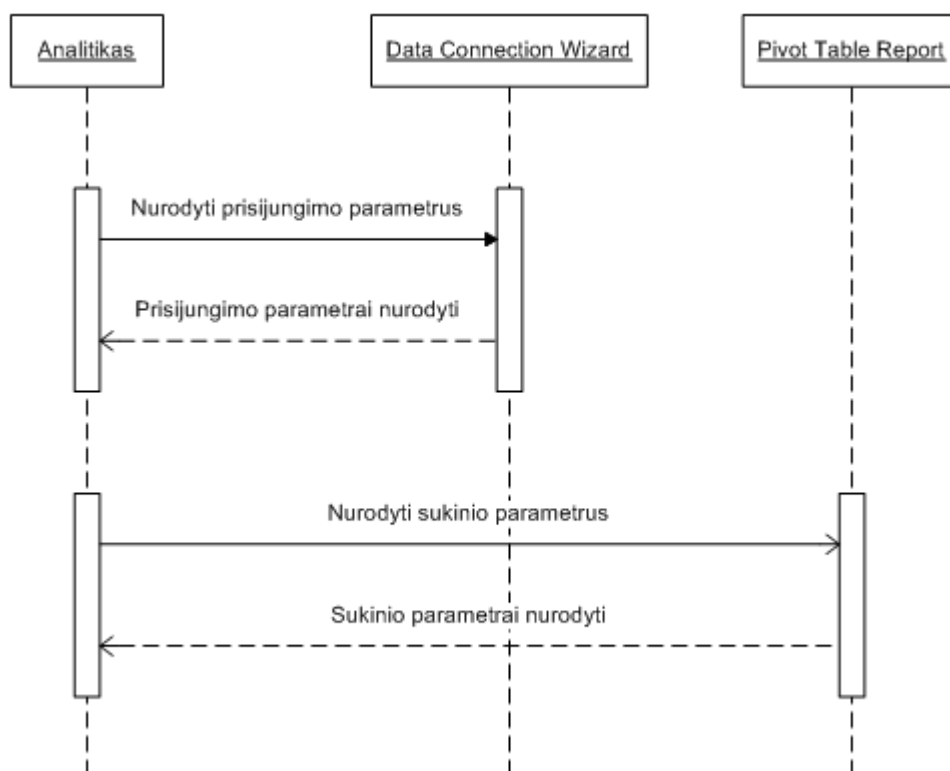


2.9 pav. Sukinių kūrimo *Excel 2007* programa veiklos diagrama

Taigi, sukinyi kuriamas naudojant *Excel 2007* programą. Pradėjus darbą šia programa, pirmiausia reikia nurodyti prisijungimo parametrus prie duomenų šaltinio, šiuo atveju tai yra *Microsoft SQL Server 2005*. Tam tikslui naudojamas *Data Connection Wizard*. Nurodžius prisijungimo prie serverio parametrus, taip pat reikia pasirinkti duomenų bazę bei vaizdinį, kurio duomenis naudosim sukinių kūrimo. Nesėkmingai atlikus šį veiksmą, jį reikia kartoti iš naujo, kol pavyks prisijungti, nes kitu atveju negalėsime prieiti prie reikalingų

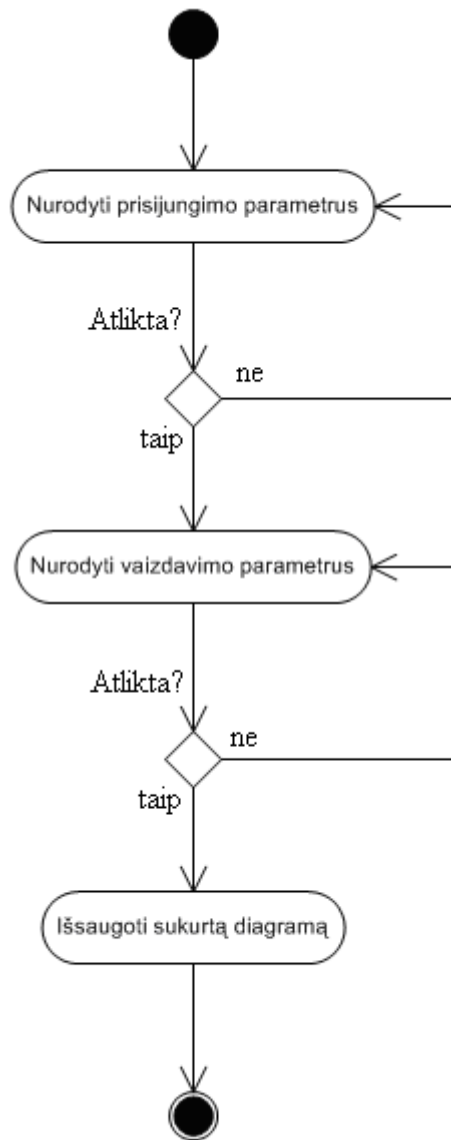
sukiniui kurti duomenų. Sėkmingai įvykdžius užduotį, pereiname prie sekančio veiksmo, t.y. sukinio parametrų nurodymo. Šiam tikslui įgyvendinti *Excel 2007* programa turi savus įrankius, tokius kaip *Pivot Table Report*. Šio įrankio pagalba galime nagrinėti duomenis kurdami sukinius įvairiausiomis dimensijomis. *Pivot Chart* įrankio pagalba galime pasirinkti norimą sukinio grafiko vaizdavimo tipą. Sėkmingai atlikus veiksmą, sukurtą sukinį galime išsaugoti, dėti į reikalingas sukurti ataskaitas bei analizuoti gautąją informaciją. Nesėkmingai įvykdžius veiksmą, jis kartojamas iš naujo, kol sėkmingai bus sukurtas sukinys.

Šie veiksmai taip pat atsispindi sukinio kūrimo *Excel 2007* programa sekų diagramoje, kuro pavaizduota 2.10 paveiksle.



2.10 pav. Sukinio kūrimo *Excel 2007* programa sekų diagrama

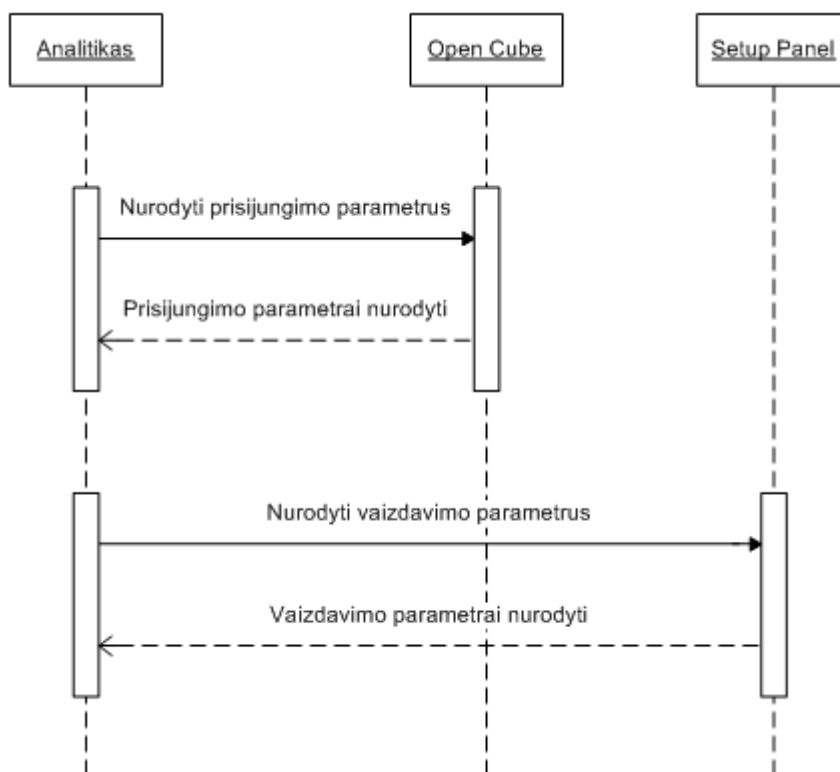
Dabar panagrinėkime grafiko kūrimo *ProClarity* programa algoritmą. Šiam veiksmui atlikti geriausiai tinka grafiko kūrimo *ProClarity* programa veiklos diagrama, pavaizduota 2.11 paveiksle.



2.11 pav. Grafiko kūrimo *ProClarity* programa veiklos diagrama

Taigi, grafikas kuriamas *ProClarity Desktop Professional 6.3* programa. Pradėjus darbą pirmiausia reikia nurodyti prisijungimo prie *SQL Server 2005* parametrus – nurodyti serverį ir prisijungus nurodyti duomenų bazę bei vaizdinį, kurie bus naudojami dirbant programa. Esant nesėkmingai veiksmo pabaigai, veiksmų seka kartojama, o esant sėkmingai pabaigai – pereinama prie kito žingsnio. Tas žingsnis yra nurodyti vaizdavimo parametrus, t.y. kaip ir kuriant sukinius, nurodyti kokiais pjūviais nagrinėsime duomenų bazėje esančią informaciją. Sėkmingai įvykdžius užduotį, ekrane gauname norimą diagramą, kurią galime išsisaugoti bei nagrinėti. Esant nesėkmingam ar neteisingam vaizdavimo parametru nurodymui, veiksmas kartojamas iš naujo, kol gaunama norima nagrinėti diagrama.

Šie veiksmai taip pat atsispindi grafiko kūrimo *ProClarity* programa sekų diagramoje, pavaizduotoje 2.12 paveiksle.



2.12 pav. Grafiko kūrimo *ProClarity* programa sekų diagrama

2.5. Reikalavimai sistemai

2.5.1. Funkciniai reikalavimai

Panaudojamumas – kad būtų galimybė naudotis sistema, reikalingas kompiuteris su specialia programine įranga bei tinklinė prieiga prie serverio (internetas arba vietinis tinklas).

Naudingumas - sukurtais duomenų kubais analizuojant duomenų saugykloje esančius duomenis, turi aiškiai matytis analizės teikiama nauda.

Stabilumas bei tikslumas - sistema turi sklandžiai veikti, todėl reikia stengtis kuriamoje duomenų saugykloje nepalikti klaidų, dėl kurių analizės metu gautume neteisingus rezultatus.

Praplečiamumas - realizavus projektą, duomenų kiekis sistemoje pastoviai didės, todėl reikia numatyti, kad sistema galėtų prijungti vis naujus duomenis ar duomenų grupes (lenteles).

2.5.2. Nefunkciniai reikalavimai

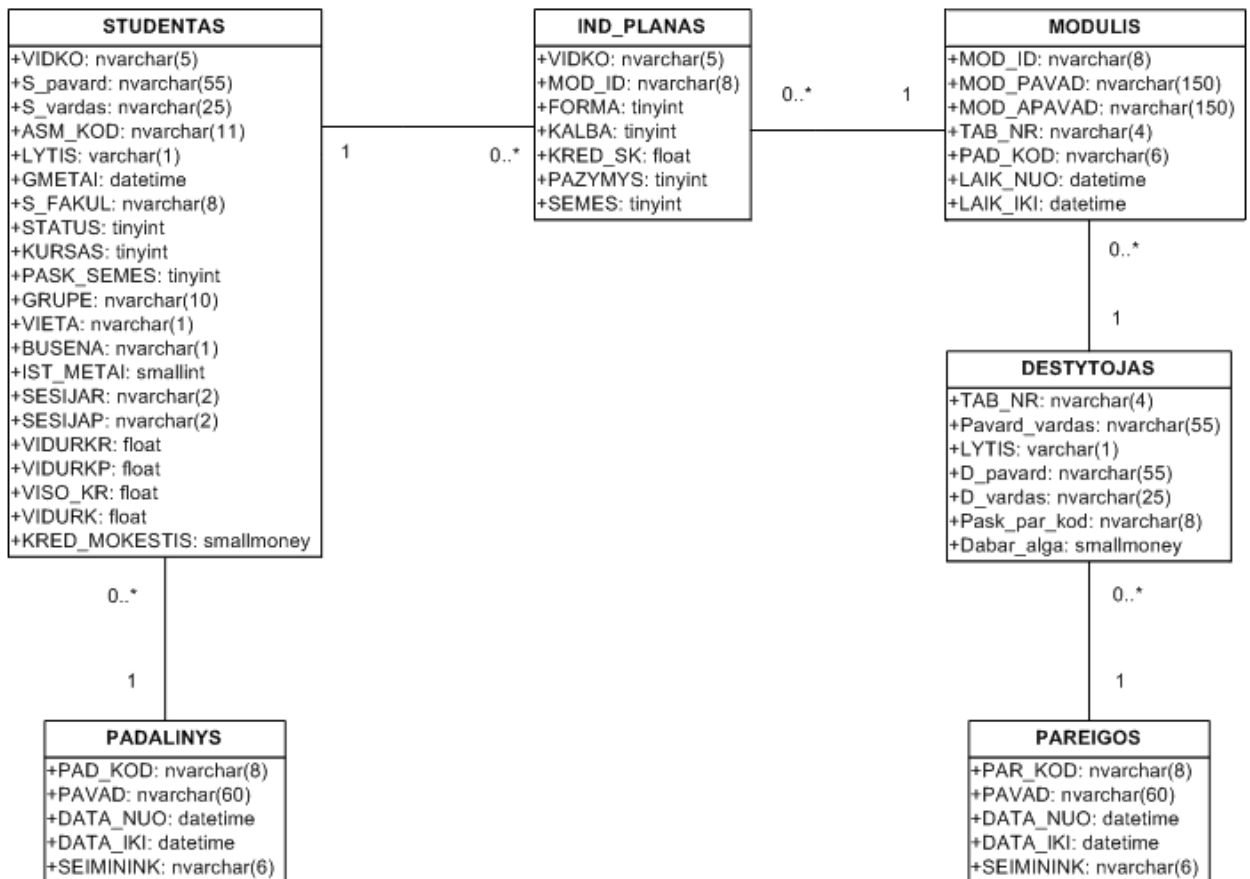
Naujumas - šiuolaikinio sėkmingo verslo etiketė - verslo analizės sistema. Savo darbo metu aš pritaikysiu duomenų analizės metodus universiteto veiklai gerinti, kas yra nauja šiai sričiai.

Išbaigtumas - sistemos išbaigtumas nėra pats svarbiausias reikalavimas, nes didėjant duomenų kiekiui bei atsirandant naujiems duomenims, bus kuriami nauji duomenų kubai, kuriais atliekama duomenų analizė.

Saugumas - sistemoje saugoma įvairi slapta informacija, pradedant studentų bei dėstytojų sąrašais, baigiant studentų bei dėstytojų asmenine informacija, tokia kaip gyvenamosios vietos adresai, asmens kodai, dėstytojų algų sumos ir t.t. Tai svarbi universitetui informacija, todėl jos apsauga yra labai svarbus reikalavimas šiai sistemai.

2.6. Esybių klasių diagrama

Šioje diagramoje matome dalykinės srities esybių klasių modelį. Šios klasės yra skirtos tik duomenims saugoti ir neturi metodų. Šis 2.13 paveiksle pavaizduotas esybių klasių modelis yra transformuojamas į duomenų bazės schemą. Klasių atributai paaiškinti 7 punkto antroje lentelėje.



2.13 pav. Esybių diagrama

2.7. Sistemos projektavimo išvados

- Šitoje darbo dalyje, siekiant suprojektuoti kuriamos sistemos modelį, buvo aptarta šios sistemos architektūra, nubraižytos jos įrangos bei komponentų diagramos, atskleidžiančios sistemos struktūros ypatumus.
- Projektuojant sistemą, buvo naudojama unifikuota modeliavimo kalba (*UML*), kuri leido profesionaliai suprojektuoti ir dokumentuoti sistemą dar prieš realizaciją. Buvo nubraižytos panaudojimo atvejų diagramos, kurios tiksliai apibrėžia kiekvieno vartotojo veiksmus sistemoje.
- Siekiant išsiaiškinti, kaip kuriama sistema, taip pat buvo panaudota *UML* kalba, kurios pagalba buvo aprašyti duomenų kubų, *Excel 2007* sukinių bei *ProClarity* grafikų kūrimo algoritmai veiklos ir sekų diagramomis.
- Nubraižyta dalykinės srities esybių klasių modelis, kuris bus panaudotas tolesnėje dalyje, kuriant duomenų bazę.

3. SISTEMOS REALIZACIJOS APRAŠAS

3.1. Sistemos realizacijos dalies tikslas

Šio darbo sistemos realizacijos dalies tikslas yra sukurti mano suprojektuotą veikiančią sistemą, kurios pagalba bus atliekama universiteto duomenų nuodugni analizė. Pirmiausia iš palikiminės KTU duomenų bazės bus kuriama *Microsoft SQL Server 2005* programų pakete veikianti duomenų saugykla. Šios saugyklos duomenų pagrindu bus kuriami duomenų kubai. Siekiant atlikti kuo išsamesnį duomenų kubų kūrimo bei saugojimo modelių tyrimą, bus kuriama kuo įvairesnių, pasižyminčių skirtingomis savybėmis bei architektūra kubų, kurie atskleistų duomenų analizės galimybes pasirinktoje nagrinėjamoje srityje.

3.2. Duomenų saugyklos sukūrimas

Kaip jau minėjome analitinėje dalyje, duomenų bazės kūrimui panaudosime KTU palikiminę duomenų bazę. Šioje duomenų bazėje yra saugoma archyvinė studentų, dėstytojų, absolventų, modulių, individualių planų ir kita universiteto informacija. KTU duomenų saugykla apima universiteto veiklos duomenis nuo 1999 m. iki 2002 m. Ji yra sukurta *dBase* programų paketu [18, 19]. Tai yra vienas iš seniausių ir pirmųjų pripažintų duomenų bazių sistemų kūrimo įrankių, kurio atsiradimo data siekia 1980 metus. Ji veikė daugumoje to meto operacinių sistemų, tokių kaip *Apple*, *UNIX*, *DOS* ir t.t. Tačiau dėl lėtos integracijos su *Windows* šeimos operacinėmis sistemomis, prarado rinką ir buvo nukonkuruota kitų produktų. *dBase* duomenų bazės failo formatas yra *.dbf*.

Taigi, norėdami sukurti savo duomenų bazę, paremtą palikiminiais duomenimis, pirmiausia turime juos apdoroti bei konvertuoti į *SQL Server 2005* programai suprantamą formatą. Visi šie žingsniai aprašyti sekančiame skyrelyje.

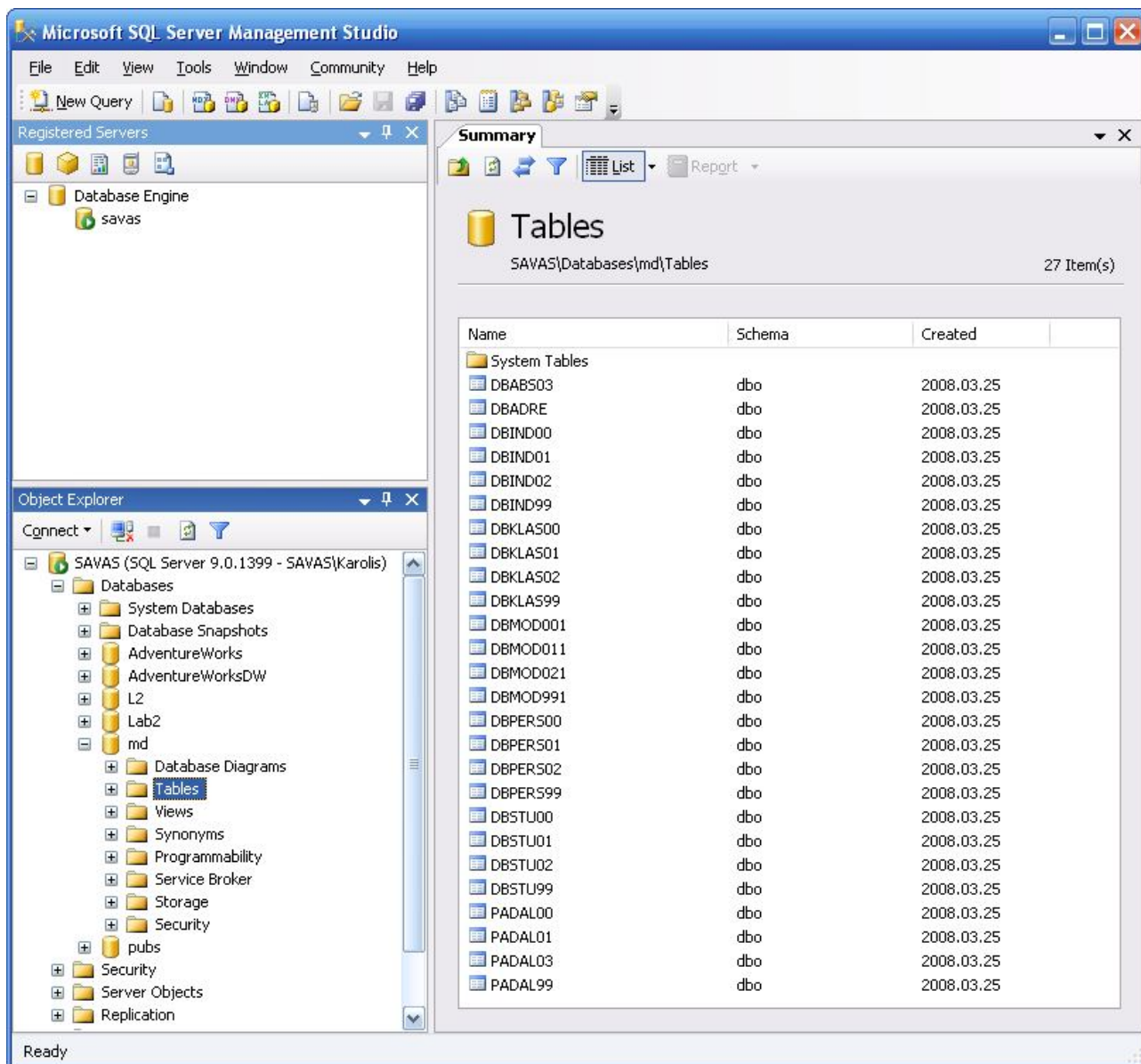
Pirmiausia turimą duomenų bazę importuosime į *Microsoft SQL Server 2005*. Turime šiuos *dBase* failus: *DBABS03.DBF* - Absolventai; *DBADRE.DBF* – Adresai; *DBIND00.DBF*, *DBIND01.DBF*, *DBIND02.DBF*, *DBIND99.DBF* – Individualūs planai; *DBKLAS00.DBF*, *DBKLAS01.DBF*, *DBKLAS02.DBF*, *DBKLAS99.DBF* – Klasės; *DBMOD001.DBF*, *DBMOD011.DBF*, *DBMOD021.DBF*, *DBMOD991.DBF* – Moduliai; *DBPERS00.DBF*, *DBPERS01.DBF*, *DBPERS02.DBF*, *DBPERS99.DBF* – Dėstytojai; *DBSTU00.DBF*, *DBSTU01.DBF*, *DBSTU02.DBF*, *DBSTU09.DBF* – Studentai; *PADAL00.DBF*, *PADAL01.DBF*, *PADAL03.DBF*, *PADAL99.DBF* - Padaliniai. *SQL Server 2005* negali tiesiogiai importuoti *dBase* duomenų bazės failo, todėl šiam veiksmui atlikti naudojame

tarpinę programinę įrangą – *Microsoft Access 2007*. Importavus šiuos failus, *Microsoft Access 2007* programos langas pateikiamas 3.1 paveiksle.

VIDKO	PAVAR	VARDAS	SKYRP	STATUS	VIETA	STUD_PR	SE
44608	Dyburytė	Indrė	1	2	V	D713DA01	03
44553	Jurevičiūtė	Živilė	1	2	V	D713DA01	03
46177	Kopteva	Natalija	1	2	V	D713DA01	03
43587	Malanova	Ala	1	2	V	D713DA01	03
45612	Pasmokytė	Asta	1	2	V	D713DA01	03
47225	Sauspreikšytė	Violeta	1	2	V	D713DA01	03
35899	Stancikas	Saulius	1	2	V	D713DA01	03
41907	Vaikutienė	Daiva	1	2	V	D713DA01	03
45779	Venslova	Dainius	1	2	V	D713DA01	03
45758	Zizonytė	Kristina	1	2	V	D713DA01	03
45491	Baltuškaitė	Kristina	1	2	V	D701DA01	03
46909	Barzdonytė	Renata	1	2	V	D701DA01	03
45607	Griciūtė	Jurgita	1	2	V	D701DA01	03
38346	Lekerauskas	Erikas	1	2	V	D701DA01	03
44285	Pabricienė	Kristina	1	2	V	D701DA01	03
45025	Stankevičiūtė	Kristina	1	2	V	D701DA01	03
45313	Zubinaitė	Lina	1	2	V	D701DA01	03
32589	Norkevičius	Rimas	1	3	P	M701DA01	04
38824	Adašiūnaitė	Jolanta	2	2	P	D713VA01	03
37578	Maleckas	Artūras	2	2	V	D713VA01	03
44277	Lukšaitė	Inga	2	2	V	D701VA01	03
45330	Martinaitytė	Vita	2	2	V	D701VA01	03
45456	Naujokaitė	Rūta	2	2	V	D701VA01	03

3.1 pav. *Microsoft Access 2007* langas importavus palikiminę duomenų bazę

Tiek *SQL Server 2005*, tiek *Excel 2007* programos yra sukurtos *Microsoft* kompanijos ir yra glaudžiai susiję, todėl pasinaudodami *SQL Server Import And Export Wizard* vedlio pagalba, sukurtos bylos duomenis galime nesunkiai perkelti į *Microsoft SQL Server 2005* serverį. Atlikus importavimo veiksmą, rezultatai pateikiami 3.2 paveiksle.



3.2 pav. Microsoft SQL Server Management Studio langas baigus importavimą

Taigi, turime duomenis, kurių pagrindu projektuojame savo duomenų bazę. Šiuos duomenis dar turime sutvarkyti. *Microsoft Excel 2007* įrankiu iš duomenų bazės failų *DBPERS00.dbf*, *DBPERS01.dbf*, *DBPERS02.dbf* ir *DBPERS99.dbf* duomenis kopijuojame į vieną darbo lapą. Iš šio darbo lapo filtruojame duomenis į naują personalo darbo lapą *Pers_pareigos*. Šis veiksmas reikalingas norint pašalinti keturiuose failuose esančius pasikartojančius dėstytojų pareigų įrašus (nes didelė dalis dėstytojų dėstė keliuose fakultetuose). Galiausiai iš parengto unikalių duomenų darbo lapo *Pers_pareigos* dėstytojų svarbiausieji stulpeliai, kurie bus naudojami kubams kurti, filtruojami į kitą darbo lapą *Dėstytojai*. Šiuo veiksmu sukursime naują dėstytojų failą, kuriame kiekvienam dėstytojui bus įrašytas tik vienas įrašas su jo svarbiausiais laukais. Darbo lape *Dėstytojai* duomenys sumaišomi, kad užtikrintume asmens duomenų konfidencialumą. Šiuos du *Microsoft Excel*

2007 programos darbo lapus perkeliame į *Microsoft SQL Server 2005*. Tikslui pasiekti naudojame *Import Spreadsheet Wizard* vedlį, kurio pagalba reikiamus *Microsoft Excel 2007* programos darbo lapus importuojame į *Microsoft Access 2007* programą. Dar kartą naudojame *SQL Server Import And Export Wizard* vedlį, kurio pagalba sukurtos bylos duomenis perkeliame į *Microsoft SQL Server 2005*. Sukurtos lentelės pavadinamos *PERS_PAREIGOS* ir *DESTYTOJAS*. Šiose lentelėse atliekami galutiniai redagavimo veiksmai – pašalinami nereikalingi atributai, pertekliniai stulpeliai, atliekami duomenų korektiškumo tvarkymo veiksmai.

Taip pat lentelėje *DESTYTOJAS* sukuriame *Pask_Par_kod* (paskutinių pareigų kodas) ir *Dabar_alga* (dabartinė alga) atributai. Pirmajame kiekvienam iš darbuotojų įrašytas paskutiniųjų jo pareigų kodas, o antrajame atribute kiekvienam iš darbuotojų apskaičiuota jo dabartinė alga. Bazinė pareigų alga yra nurodyta pareigų lentelėje. Jeigu darbuotojo stažas (skaičiuojant nuo pradinės įdarbinimo datos) yra didesnis nei 25 metai, tai jam bazinė einamų pareigų alga padidinta 40%. Jeigu darbuotojo stažas didesnis nei 10 metų, tai jam bazinė alga padidinta 20%.

Lentelė 3.1 Universiteto darbuotojų pareigų lentelė

Pareigų pavadinimas	Pareigų kodas	Bazinė alga litais
Profesorius	20870	4500
Docentas	20863	3000
Profesorius - lektorius	20872	2400
Docentas – lektorius	20865	2100
Kitos pareigos		1900

Taip pat reikia *Microsoft SQL Server 2005* duomenų bazėje sulieti tos pačios struktūros failų grupes: iš *PADAL00*, *PADAL01*, *PADAL03* ir *PADAL99* - į *PADAL*; iš *DBSTU00*, *DBSTU01*, *DBSTU03* ir *DBSTU99* - į *DBSTU*; iš *DBIND00*, *DBIND01*, *DBIND02* ir *DBIND99* - į *DBIND*; iš *DBMOD001*, *DBMOD011*, *DBMOD021* ir *DBMOD991* - į *DBMOD*; iš *DBKLAS00*, *DBKLAS01*, *DBKLAS02* ir *DBKLAS99* - į *DBKLAS*. Šie veiksmai atliekami naudojant *SQL* užklausų (*SQL Query*) kalbą. Pvz.: sujungiant *PADAL* lenteles, atliekami tokie veiksmai: pirmiausia sukuriamą *PADAL* lentelė su tokiais pačiais laukais ir laukų tipais kaip ir *PADAL00*; tuomet įvykdome *SQL* užklausą:

```
INSERT INTO PADAL
SELECT * FROM PADAL00
UNION ALL
SELECT * FROM PADAL01
UNION ALL
SELECT * FROM PADAL03
UNION ALL
```

*SELECT * FROM PADAL99*

Ši užklausa įterpia visus įrašus iš lentelių *PADAL00*, *PADAL01*, *PADAL03* ir *PADAL99* į lentelę *PADAL*. Analogiški veiksmai atliekami ir su kitomis lentelių grupėmis.

Kuriama lentelė *STUDENTAS* iš *DBSTU* lentelės. *STUDENTAS* lentelėje yra šie atributai: *VIDKO*, *PAVAR* pavadinimą keičiame į *S_pavard*, *VARD* pakeičiame į *S_vardas*, *ASM_KOD*, *LYTIS*, *GMETAI*, *FAKUL* keičiame į *S_FAKUL*, *STATUS*, *KURSAS*, *SEMES* pakeičiame į *PASK_SEMES*, *GRUPE*, *VIETA*, *BUSENA*, *IST_METAI*, *SESIJAR*, *SESIJAP*, *VIDURKR*, *VIDURKP*, *VISO_KR*, *VIDURK*. *LYTIS* stulpelio reikšmes keičiame tokia tvarka: jeigu reikšmė *I*, tuomet pakeičiame į *V*, kitu atveju – į *M*. Prie *GMETAI* stulpelio reikšmių iš kairės pusės pridėdame *19* ir turimą eilutę konvertuojame į datą. *STATUS*, *KURSAS*, *SEMES* stulpelių reikšmes konvertuojame į *Tinyint* formatą. *IST_METAI* stulpelio reikšmes konvertuojame į *Smallint* formatą. Išrenkame tik įrašus, su unikaliomis *VIDKO* stulpelio reikšmėmis, imdami atitinkamo studento įrašą su didžiausia stulpelio *VISO_KR* (kreditų skaičius) reikšme bei įrašome juos į sukuriamą *STUDENTAS* lentelę.

Kuriama lentelė *MODULIS* iš *DBMOD* lentelės. *MODULIS* lentelėje yra šie atributai: *MODULIS* pavadinimą keičiame į *MOD_ID*, *MOD_PAVAD*, *MOD_APAVAD*, *TAB_NR*, *PAD_KOD*, *LAIK_NUO*, *LAIK_IKI*. *LAIK_NUO* ir *LAIK_IKI* stulpelių reikšmes konvertuojame į *Datetime* formatą. Išrinkame tik įrašus, su unikaliomis *MODULIS* stulpelio reikšmėmis, imdami pirmąjį atitinkamo modulio įrašą bei įrašome juos į sukuriamą *MODULIS* lentelę.

Kuriama lentelė *IND_PLANAS* iš *DBIND* lentelės. *IND_PLANAS* lentelėje yra šie atributai: *VIDKO*, *MODULIS* pavadinimą keičiame į *MOD_ID*, *FORMA*, *KALBA*, *KRED_SK*, *PAZYMYS*, *SEMES*. *PAZYMYS* stulpelio reikšmes keičiame tokia tvarka: jeigu reikšmės *AT*, *BR*, *NE*, *NM*, *NS*, tuomet pakeičiame į *0*, jei reikšmė *IS*, tai keičiame į *5*, kitu atveju – naudojame esamą *PAZYMYS* stulpelio reikšmę. *FORMA*, *KALBA*, *PAZYMYS*, *SEMES* stulpelių reikšmes konvertuojame į *Tinyint* formatą. Išrenkame visus įrašus iš lentelės *DBIND* ir atliekame jiems nurodytus pakeitimus bei įrašome juos į sukuriamą *IND_PLANAS* lentelę.

Kuriame lentelę *PADALINYS* iš *PADAL* lentelės. *PADALINYS* lentelėje yra šie atributai: *KODAS* pavadinimą keičiame į *PAD_KOD*, *PAVADINIM* pakeičiame į *PAVAD*, *DATA_NUO*, *DATA_IKI*, *SEIMININK*. *DATA_NUO* ir *DATA_IKI* stulpelių reikšmes konvertuojame į *Datetime* formatą. Išrenkame tik įrašus, su unikaliomis *KODAS* stulpelio reikšmėmis (tik inicializuotos), imdami pirmąjį atitinkamo padalinio įrašą bei įrašome juos į sukuriamą *PADALINYS* lentelę.

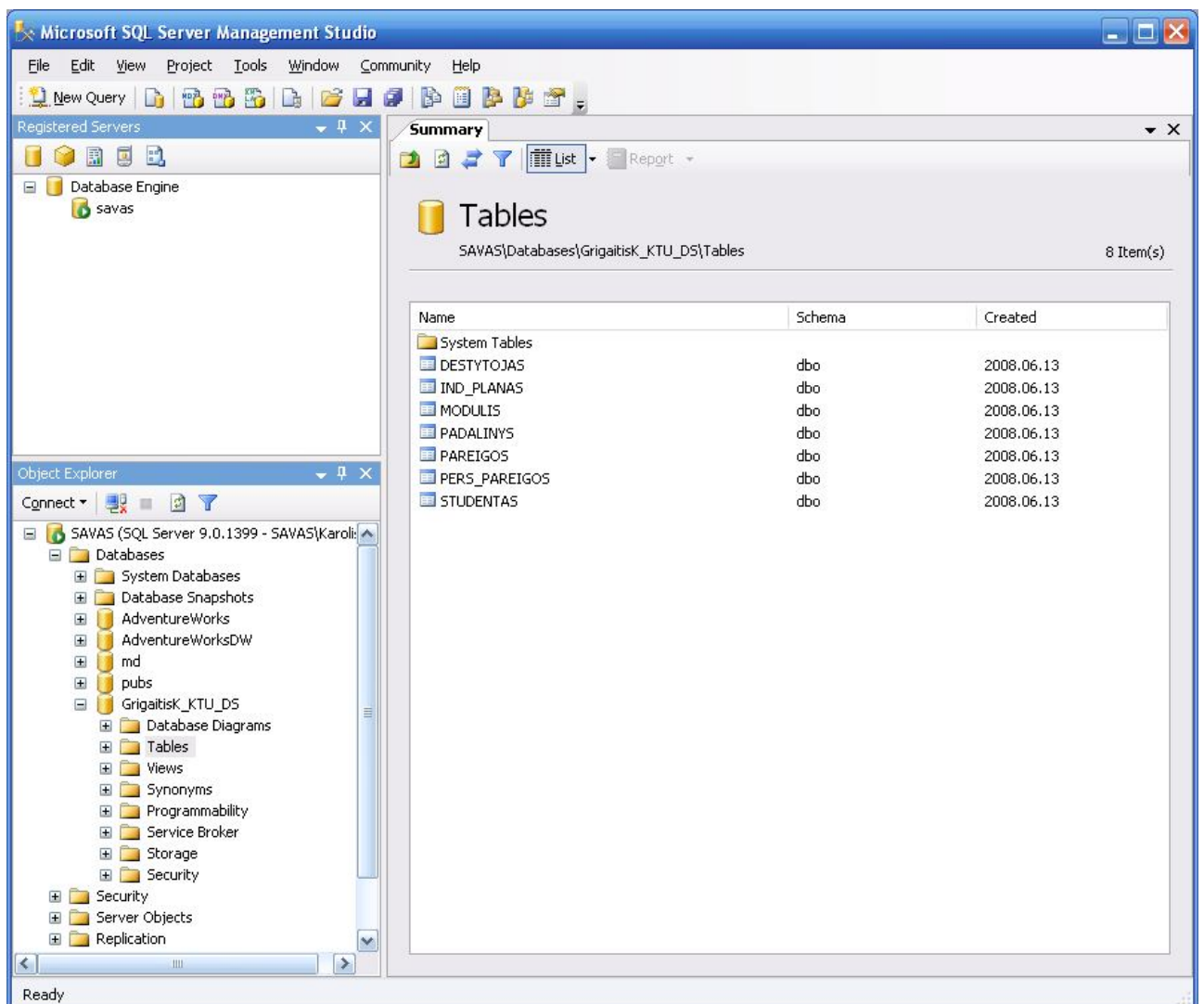
Kuriame lentelę *PAREIGOS* iš *DBKLAS* lentelės. *PAREIGOS* lentelėje yra šie atributai: *KODAS* pavadinimą pakeičiame į *PAR_KOD*, *PAVADINIM* pakeičiame į *PAVAD*,

DATA_NUO, *DATA_IKI*, *SEIMININK*. *DATA_NUO* ir *DATA_IKI* stulpelių reikšmes konvertuojame į *Datetime* formatą. Išrenkame tik įrašus, su unikaliomis *KODAS* stulpelio reikšmėmis (tik inicializuotos), imdami pirmąjį atitinkamų pareigų įrašą bei įrašome juos į sukuriamą *PAREIGOS* lentelę.

Į lentelę *STUDENTAS* įterpiame naują stulpelį, kurį pavadiname *KRED_MOKESTIS*. Šiame stulpelyje yra saugoma studento mokesčio už vieną modulio kreditą reikšmė. Atnaujiname (*UPDATE*) stulpelio *KRED_MOKESTIS* reikšmes lentelėje *STUDENTAS* tokiomis sąlygomis:

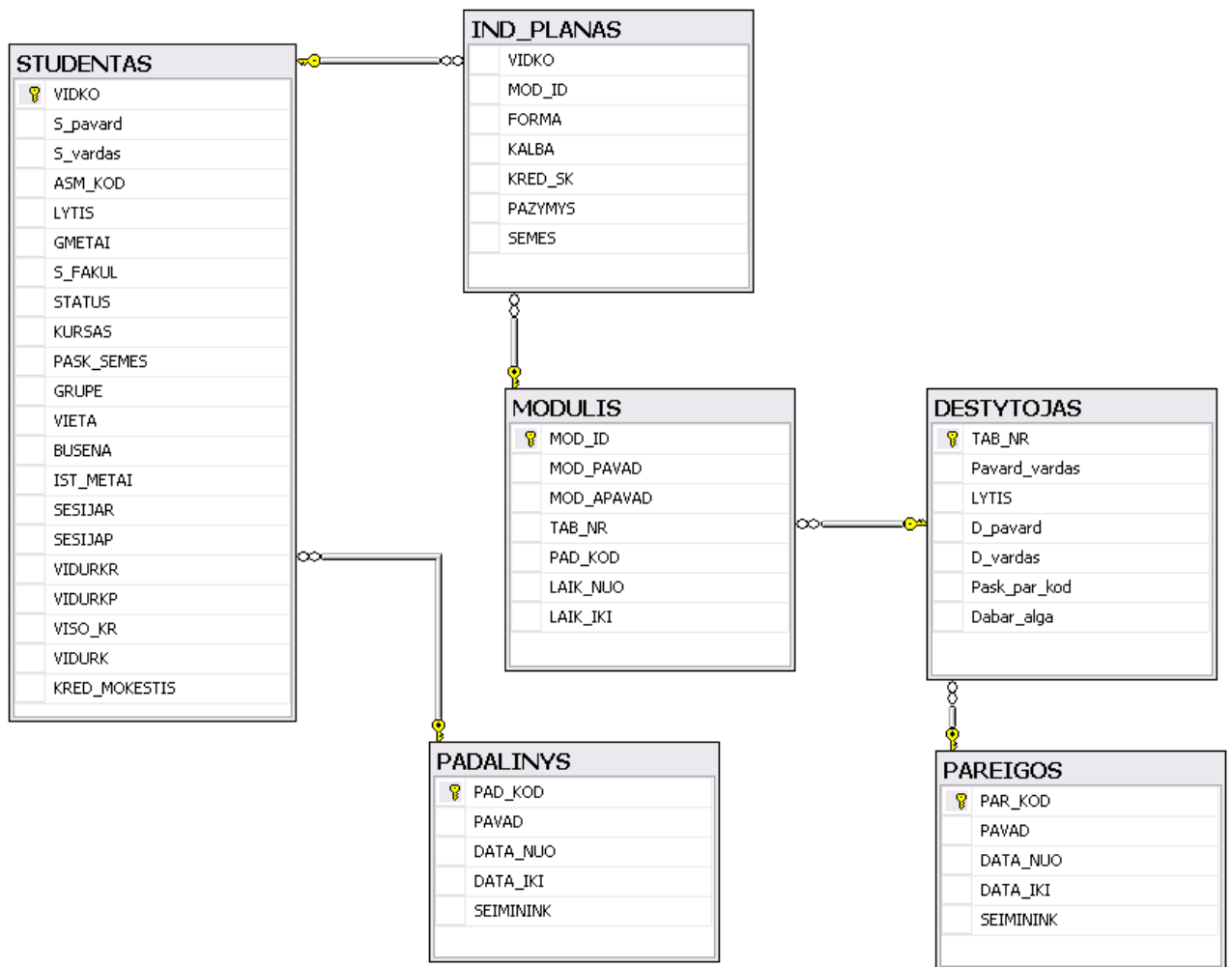
- a) Jeigu studento vidurkis (*VIDURK* stulpelio reikšmė) yra mažesnis už 8,5, tuomet *KRED_MOKESTIS* stulpeliui priskiriame 200 Lt.
- b) Kitu atveju - *KRED_MOKESTIS* stulpelio reikšmę imame 40% mažesnę (nuo 200 Lt).

Sutvarkius visas lenteles bei jų duomenis, nereikalingos lentelės iš duomenų bazės pašalinamos. Gauti rezultatai pateikti 3.3 paveiksle.



3.3 pav. Sutvarkyta duomenų bazė

Kuriame sutvarkytos duomenų bazės diagramą (*Database Diagrams*), kurios schema pavaizduota 3.4 paveiksle. Sudarytoje diagramoje sukeliamos 6 sutvarkytos lentelės: *DESTYTOJAS*, *STUDENTAS*, *MODULIS*, *IND_PLANAS*, *PADALINYS*, *PAREIGOS*. Pažymime lentelėse (kur reikia) *Primary key* atributus ir sujungiame lenteles ryšiais.



3.4 pav. Sukurta duomenų bazės diagrama

Duomenų bazės struktūra bei aprašymas pateikti lentelėje.

Lentelė 3.2 Duomenų bazės struktūra ir aprašymas

Lentelės pavadinimas	Atributo pavadinimas	Atributo tipas	Aprašas
STUDENTAS	VIDKO	nvarchar(5)	Studento numeris
	S_pavard	nvarchar(55)	Studento pavardė
	S_vardas	nvarchar(25)	Studento vardas
	ASM_KOD	nvarchar(11)	Asmens kodas
	LYTIS	varchar(1)	Studento lytis
	GMETAI	datetime	Studento gimimo metai
	S_FAKUL	nvarchar(8)	Studento fakultetas
	STATUS	tinyint	Studento statusas
	KURSAS	tinyint	Studento kursas
	PASK_SEMES	tinyint	Studento paskutinis semestras
	GRUPE	nvarchar(10)	Studento grupė
	VIETA	nvarchar(1)	Studento vieta
	BUSENA	nvarchar(1)	Studento būseną
	IST_METAI	smallint	Studento įstojimo metai
	SESIJAR	nvarchar(2)	Rudens sesijos statusas
	SESIJAP	nvarchar(2)	Pavasario sesijos statusas
	VIDURKR	float	Studento rudens vidurkis
	VIDURKP	float	Studento pavasario vidurkis
	VISO_KR	float	Studento kreditų skaičius
	VIDURK	float	Bendras vidurkis
KRED_MOKESTIS	smallmoney	Kredito mokestis	
IND_PLANAS	VIDKO	nvarchar(5)	Studento numeris
	MOD_ID	nvarchar(8)	Modulio ID
	FORMA	tinyint	Mokymosi forma
	KALBA	tinyint	Mokymosi kalba
	KRED_SK	float	Kreditų skaičius
	PAZYMYS	tinyint	Pažymys
	SEMES	tinyint	Semestras
PADALINYS	PAD_KOD	nvarchar(8)	Padalinio kodas
	PAVAD	nvarchar(60)	Padalinio pavadinimas
	DATA_NUO	datetime	Padalinio data nuo
	DATA_IKI	datetime	Padalinio data iki
	SEIMININK	nvarchar(6)	Kam priklauso
MODULIS	MOD_ID	nvarchar(8)	Modulio ID
	MOD_PAVAD	nvarchar(150)	Modulio pavadinimas
	MOD_APAVAD	nvarchar(150)	Modulio anglų kalboje pavadinimas
	TAB_NR	nvarchar(4)	Numeris
	PAD_KOD	nvarchar(6)	Padalinio kodas
	LAIK_NUO	datetime	Modulio laikas nuo
	LAIK_IKI	datetime	Modulio laikas iki
DESTYTOJAS	TAB_NR	nvarchar(4)	Numeris
	Pavard_vardas	nvarchar(55)	Dėstytojo pavardė ir vardas
	LYTIS	varchar(1)	Dėstytojo lytis
	D_pavard	nvarchar(55)	Dėstytojo pavardė
	D_vardas	nvarchar(25)	Dėstytojo vardas
	Pask_par_kod	nvarchar(8)	Paskutinių pareigų kodas
	Dabar_alga	smallmoney	Dėstytojo dabartinė alga
PAREIGOS	PAR_KOD	nvarchar(8)	Pareigų kodas
	PAVAD	nvarchar(60)	Pareigų pavadinimas
	DATA_NUO	datetime	Pareigų data nuo
	DATA_IKI	datetime	Pareigų data iki
	SEIMININK	nvarchar(6)	Kam priklauso

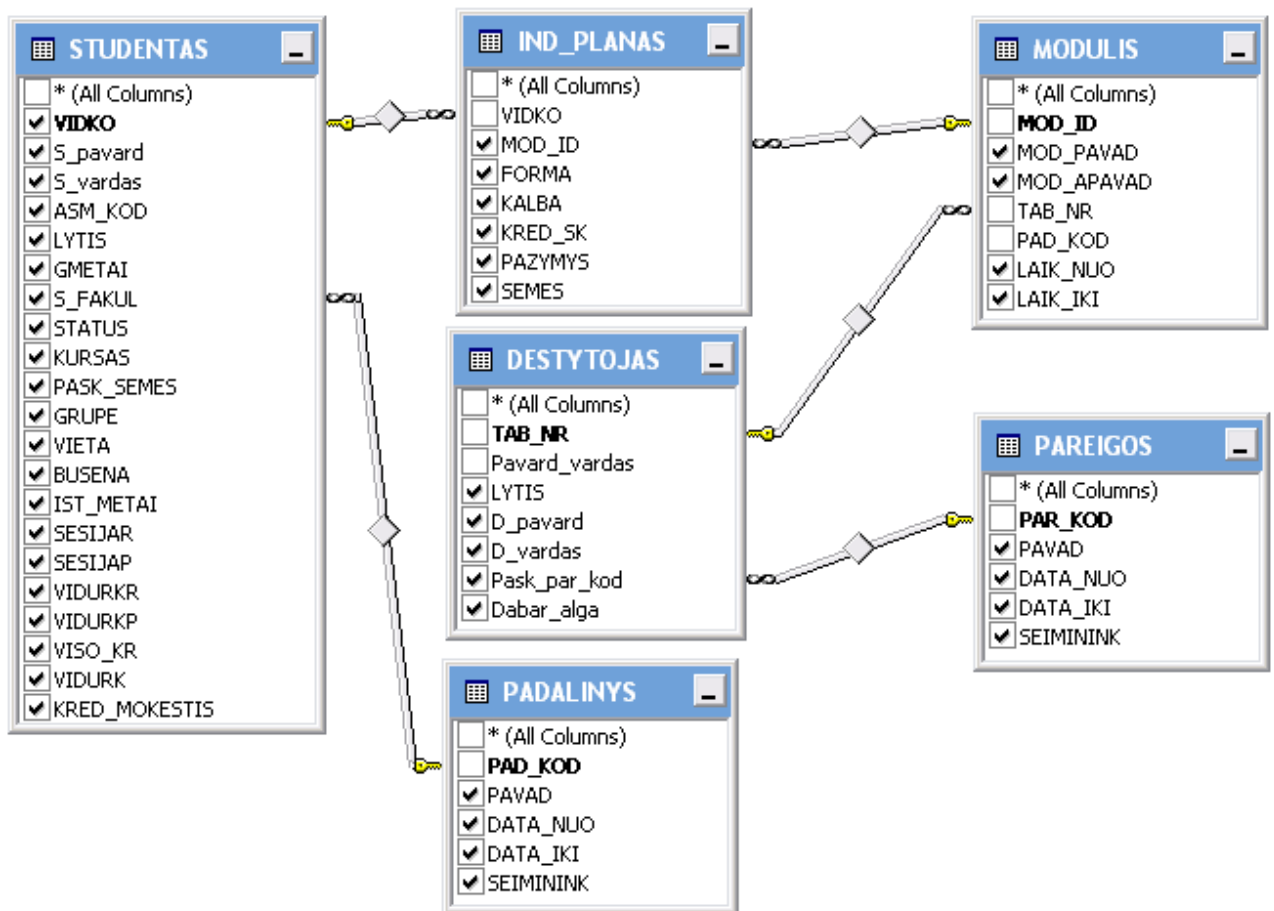
Šiais veiksmais buvo sukurta KTU duomenų bazė, paremta senąja palikimine duomenų baze ir veikianti *Microsoft SQL Server 2005* serveryje. Toliau savo darbe naudosiu šią duomenų bazę kaip pagrindą kurti duomenų kubams, sukiniams bei indikatorinėms diagramoms.

3.3. Duomenų kubų realizacija

Duomenų kubai kuriami *SQL Server Business Intelligence Development Studio* programa. Ši programa įdiegiama kartu su *SQL Server 2005* programų paketu.

3.3.1. Pirmasis sukurtas duomenų kubas

Pirmąjį vaizdinį, kurio schemą matome 3.5 paveiksle, kuriame iš visų šešių duomenų saugyklos lentelių. Vaizdinį galime kurti ir iš mažesnio kiekio lentelių, tačiau tuo atveju duomenų kubas bus sudarytas iš mažiau duomenų.

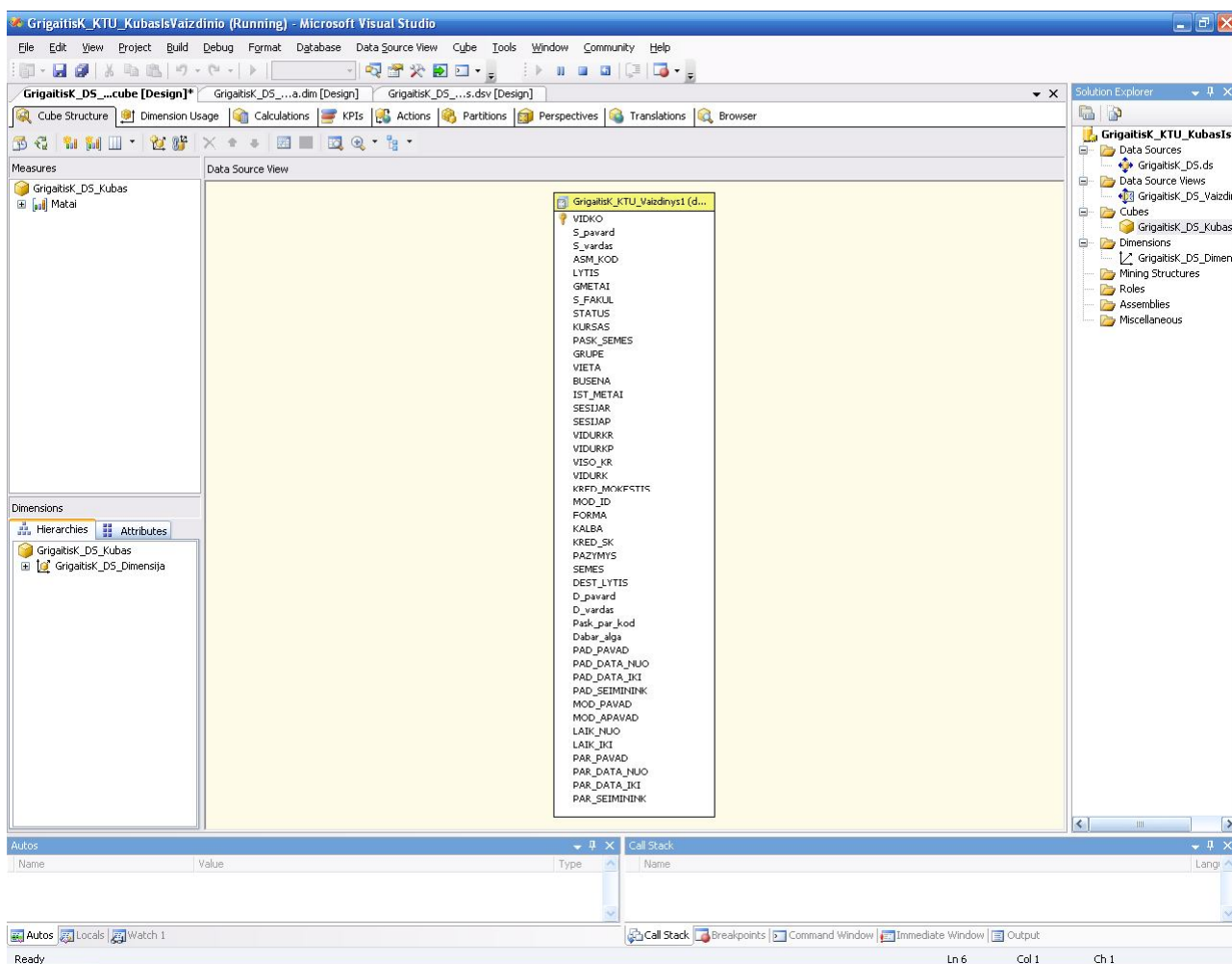


3.5 pav. Pirmojo duomenų kubo vaizdinio schema

Pirmąjį kubą kuriame panaudodami prieš tai aprašytą vaizdinį, kuris apima visas duomenų bazėje esančias lenteles. Kubas kuriamas projekcinėje dalyje aprašytu algoritmu. Kitaip sakant *OLAP* duomenų kubo kūrimas susideda iš šių žingsnių:

- a) Apibrėžti duomenų šaltinį (nustatyti prisijungimo prie duomenų šaltinio parametrus).
- b) Apibrėžti duomenų šaltinio vaizdinį, kurio duomenys bus naudojami kubo atvaizdavime.
- c) Sukurti kubo dimensijas.
- d) Mokymosi forma-Semestras-Modulio pavadinimas hierarchijos sukūrimas.
- e) Kubo sukūrimas.
- f) *Deployment* proceso įvykdymas.

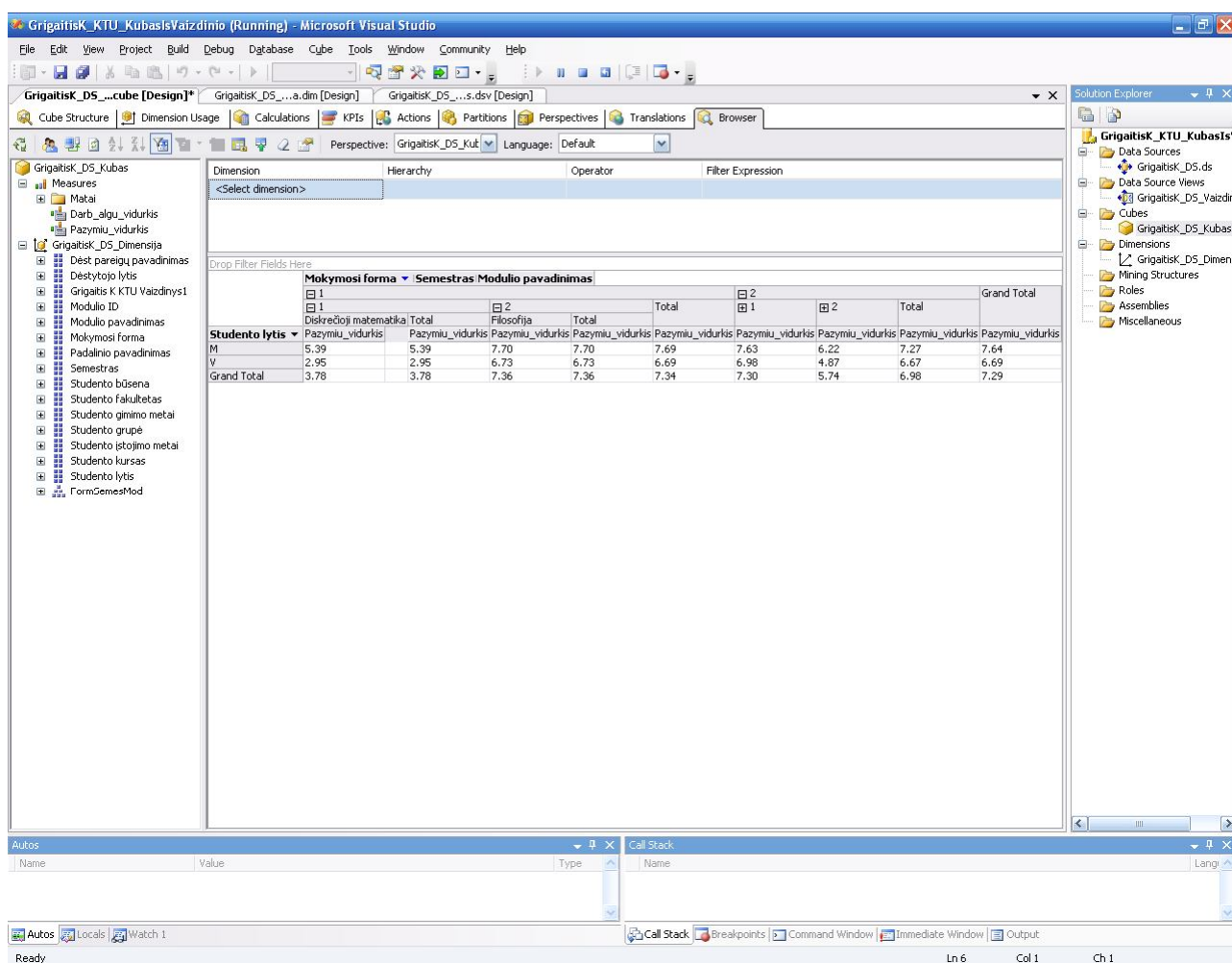
Atlikus šiuos veiksmus gauname rezultatus, pavaizduotus 3.6 paveiksle.



3.6 pav. Sukurto pirmojo *OLAP* kubo struktūra

Business Intelligence priemonėmis kuriame sukini, kuriame atsispindėtų pasirinktų iš variantų lentelės modulių vidurkiai pagal studento lytį. Pirmiausia pagrindiniame *Cube*

structure lange sukuriame du naujus matus: *pažymių suma* bei *pažymių skaičius*. Tuomet einame į *Calculations* langą, jame sukuriame pažymių vidurkio skaičiuojamąjį matą bei įvykdome *deployment* procesą. Tuo pačiu principu galime sukurti ir darbuotojų algų vidurkio skaičiuojamąjį matą. Tam veiksmui įvykdyti vėl reikia *Cube structure* lange sukurti dar du naujus matus: *darbuotojų algų suma* ir *darbuotojų algų skaičius*, *Calculations* lange sukurti darbuotojų algų vidurkio skaičiuojamąjį matą ir dar kartą įvykdyti *deployment* procesą. Galiausiai einame į *Browser* langą, kuriame nurodome norimus peržiūrėti duomenis pagal mūsų pasirinktas dimensijas bei matus. 3.7 paveiksle pateiktas sukinys, kuriame matome pirmos bei antros mokymosi formos pasirinktų modulių vidurkius pagal studento lytį.

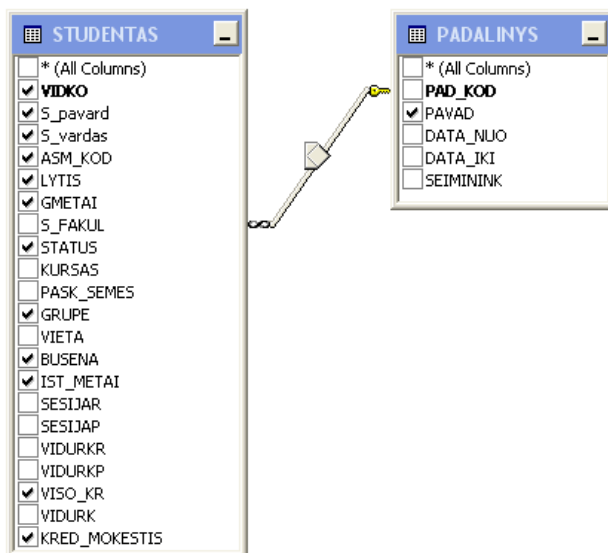


3.7 pav. Sukinio peržiūra *Business Intelligence* priemonėmis

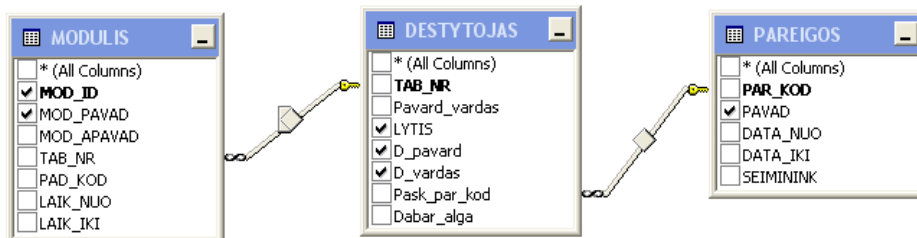
Sukinių peržiūra *Microsoft Visual Studio 2005* programa nėra labai lanksti bei negali grafiškai atvaizduoti sukurtus sukinius, todėl atlikdami intelektualią duomenų analizę, naudosime *Microsoft Excel 2007* priemones. Ši programa, naudodama *OLE DB* protokolą, gali prisijungti prie sukurto duomenų kubo bei grafiškai atvaizduoti sukinius pagal sukurto duomenų kubo dimensijas bei matus.

3.3.2. Antrasis sukurtas duomenų kubas

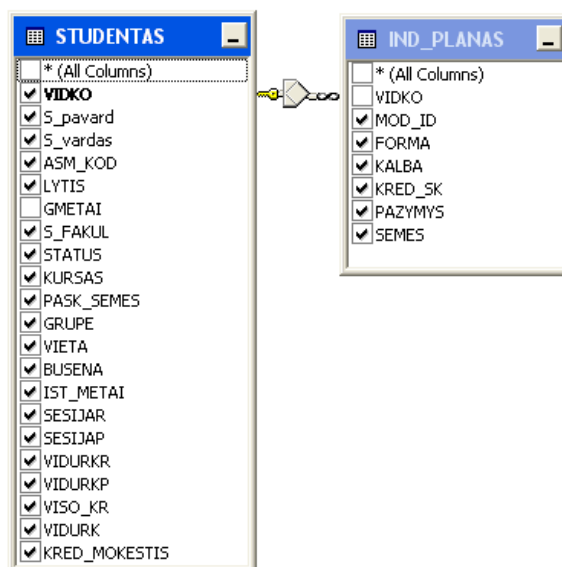
Antrasis kubas sukurtas panaudojant 3.10, 3.11 ir 3.12 paveiksluose pateiktas vaizdinių schemas, kurių kiekviena atspindi kuriamo kubo dimensinę arba faktinę lentelę.



3.8 pav. Antrojo kubo dimensijos *DimStudentas* vaizdinys

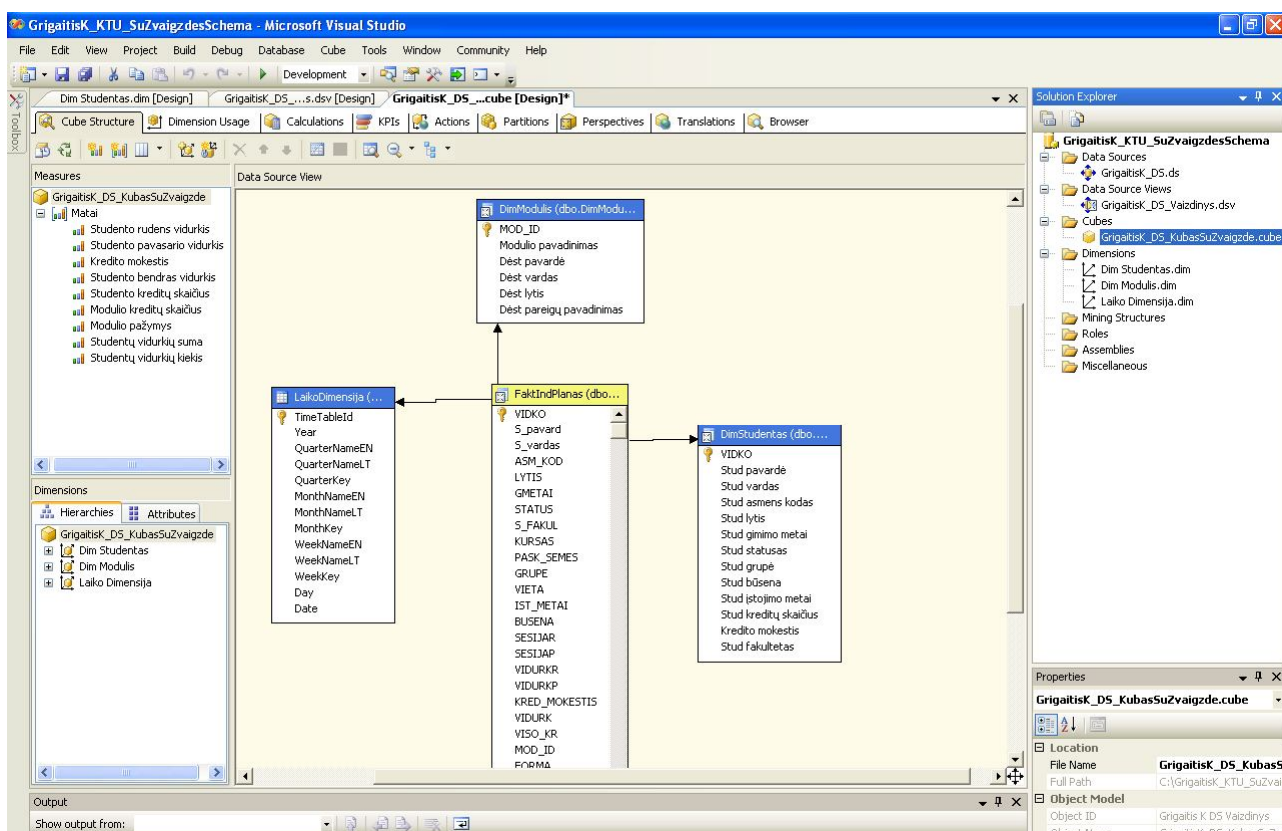


3.9 pav. Antrojo kubo dimensijos *DimModulis* vaizdinys



3.10 pav. Antrojo kubo faktinės lentelės *FaktIndPlanas* vaizdinys

Sukurto antrojo kubo struktūra pateikta 3.13 paveiksle.



3.11 pav. Antrojo sukurto kubo struktūra

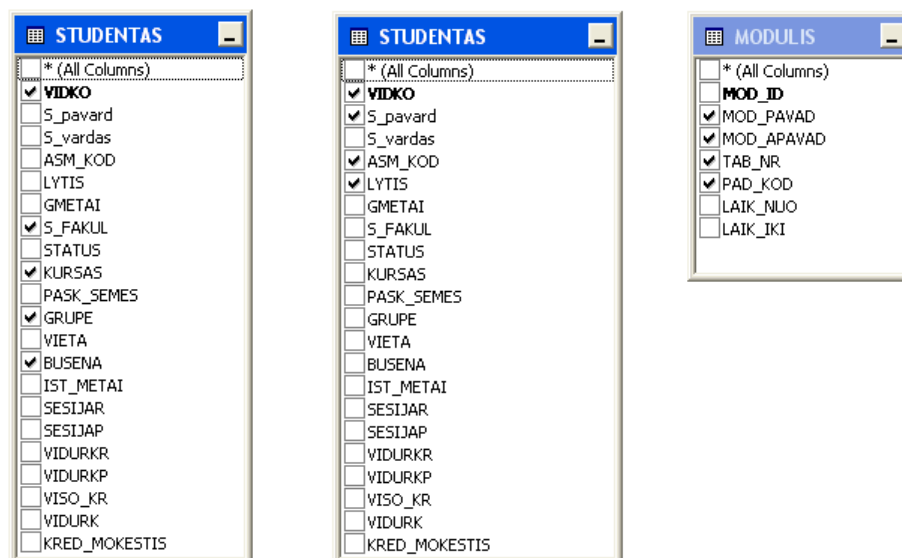
Šis kubas sukurtas pagal žvaigždės tipo schemą, t.y. centre yra faktinė lentelė ir aplinkui ją prijungtos dimensinės lentelės. Kaip matome iš kubo struktūros, jis turi tris dimensines bei vieną faktinę lentelę. Dimensijoje *DimModulis* saugoma informacija apie

modulio pavadinimą, dėstytojo vardą, pavardę, lytį bei pareigų pavadinimą. Dimensijoje *DimStudentas* laikoma informacija apie studento vardą, pavardę, asmens kodą, lytį, gimimo metus, statusą, grupę, būseną, įstojimo į universitetą metus, studento kreditų skaičių, kredito mokesčių bei fakultetą. Taip pat panaudota laiko dimensija, kuri susieta ryšiu su studento gimimo metais. Iš faktinės duomenų lentelės matams kurti panaudojam šiuos duomenis: studento rudens vidurkis, studento pavasario vidurkis, kredito mokestis, studento bendras vidurkis, studento kreditų skaičius, modulio pažymys. Sukuriami du nauji matai studentų vidurkių suma ir studentų vidurkių kiekis, kurių duomenimis sukuriamas skaičiuojamasis matas bendras studentų vidurkis.

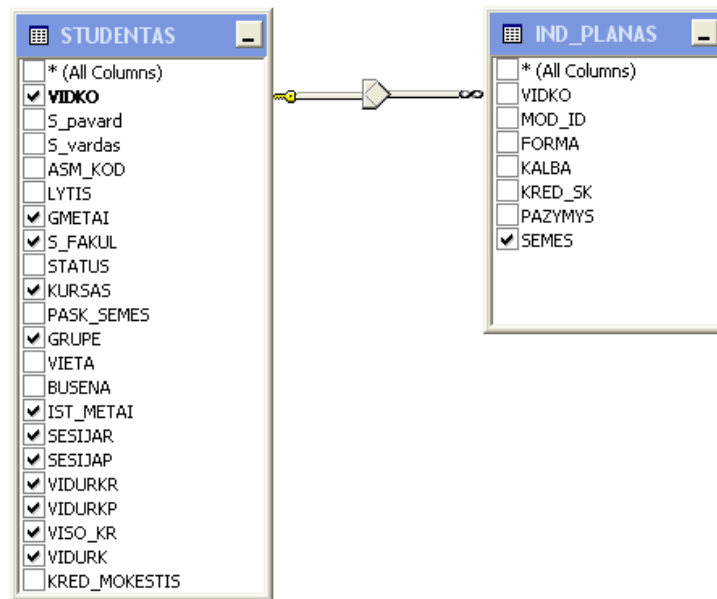
Šis kubas skirtas analizuoti studentų mokymosi informaciją. Galime analizuoti duomenis pagal studento asmeninę informaciją, tokią kaip vardas, pavardė, asmens kodas, lytis, gimimo metai ar grupė. Tačiau didžiausią naudą teikia tokios dimensijos kaip studento būseną, įstojimo į universitetą metai, studento kreditų skaičius ir fakultetas. Visus šiuos duomenis galime analizuoti su mokymosi modulių susijusios informacijos dimensijų lentele, kuri leistų studentų informaciją peržiūrėti modulio pavadinimo, dėstytojo vardo, pavardės ir lyties dimensijomis. Taip pat sukurtos laiko dimensijos dėka galime filtruoti duomenis pagal studento gimimo metus. Rezultatus apibendrinti galime pagal sukurtos faktinės lentelės matus.

3.3.3. Trečiasis sukurtas duomenų kubas

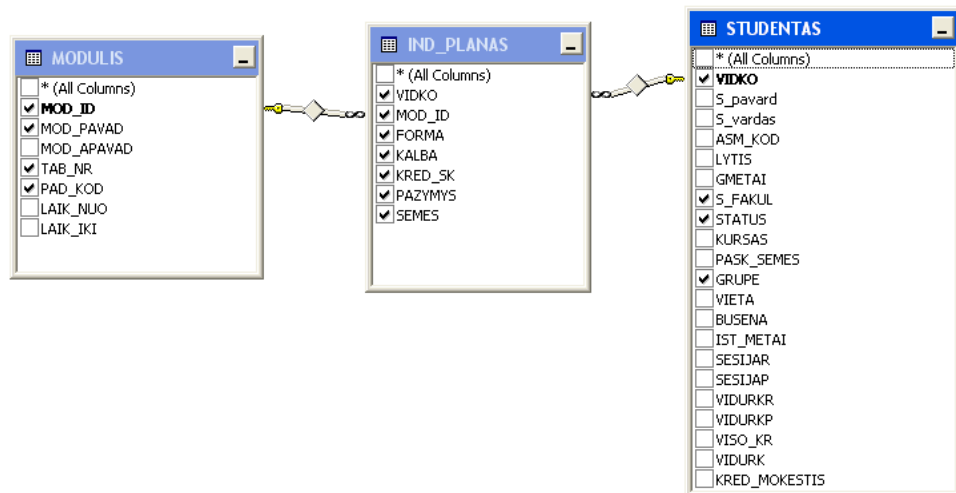
Trečiasis kubas sukurtas panaudojant 3.14, 3.15 ir 3.16 paveiksluose pateiktas vaizdinių schemas, kurių kiekviena atspindi kuriamo kubo dimensinę arba faktinę lentelę.



3.12 pav. Trečiojo kubo dimensijos vaizdiniai: iš eilės - *IDimGrupės*, *IDimStudentai* ir *IDimModuliai*

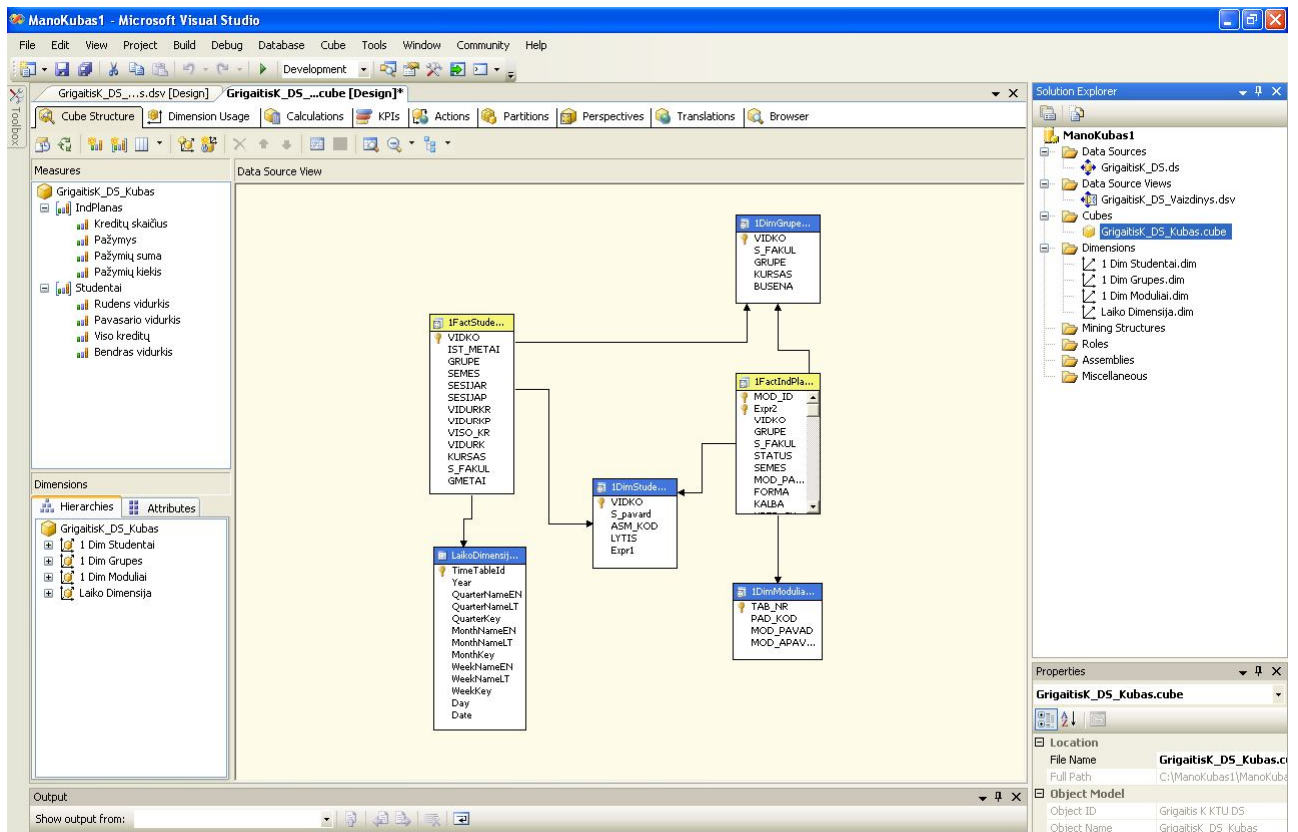


3.13 pav. Trečiojo kubo faktinės lentelės *IFaktStudentai* vaizdinys



3.14 pav. Trečiojo kubo faktinės lentelės *IFaktIndPlanas* vaizdinys

Sukurto trečiojo kubo struktūra pateikta 3.17 paveiksle.



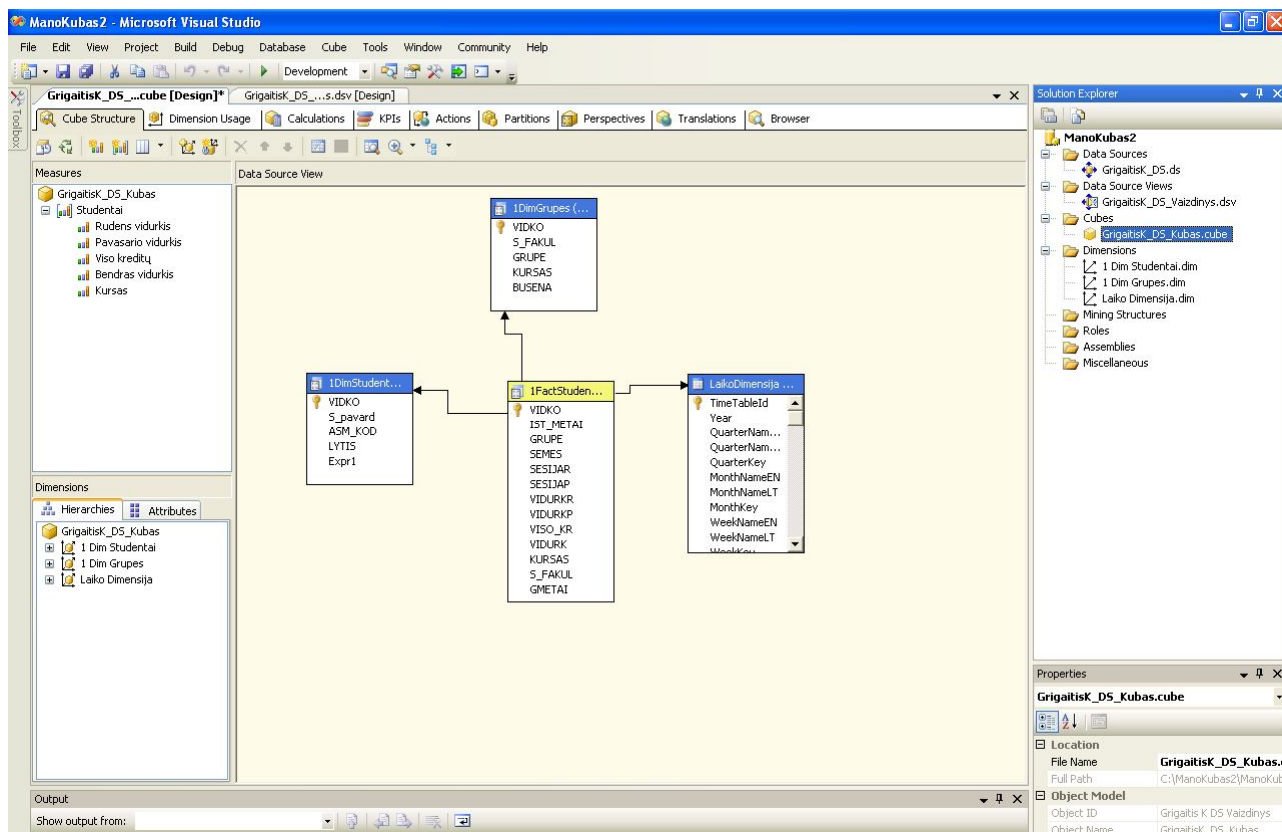
3.15 pav. Trečiojo sukurtą kubo struktūra

Trečiasis kubas įdomus tuo, kad jis taip pat sukurtas žvaigždės tipo schemas pagrindu, tačiau jame panaudotos dvi skirtingos faktinės lentelės *IFaktIndPlanas* ir *IFaktStudentai*. Iš pirmosios faktinės lentelės *IFaktIndPlanas* sukuriama šie matai: kreditų skaičius bei modulio pažymys. Taip pat skaičiuojant modulio pažymių kiekį bei sumą, sukuriama skaičiuojamasis matas studentų pažymių vidurkis. Iš antrosios faktinės lentelės *IFaktStudentai* sukuriama šie matai bendras studento vidurkis, studento pavasario vidurkis, studento rudens vidurkis bei studento kreditų skaičius.

Naudojantis šiuo kubu patogu analizuoti duomenis pagal sukurtų dimensinių lentelių grupes, kuriose sugrupuoti konkrečios srities dimensijos, pavyzdžiui dimensinėje lentelėje *IDimGrupes* yra informacija apie fakultetą, kursą, grupę bei būseną; dimensinėje lentelėje *IDimStudentai* yra dimensijos apie studento vardą, pavardę, asmens kodą bei lytį; galiausiai *IDimModuliai* dimensinėje lentelėje yra informacija apie universiteto modulius – t.y. modulio pavadinimas, modulio anglišką pavadinimą ir padalinio kodą. Analogiškai egzistuoja dvi grupės matų, kurių viename yra informacija apie studento vidurkius, o kitoje apie kreditų skaičius bei pažymius.

3.3.4. Ketvirtas sukurtas duomenų kubas

Ketvirtajam kubui sukurti panaudota dalis tų pačių dimensinių bei faktinių lentelių kaip ir prieš tai kurtam kubui. Sukurto ketvirtojo kubo struktūra pateikta 3.18 paveiksle.

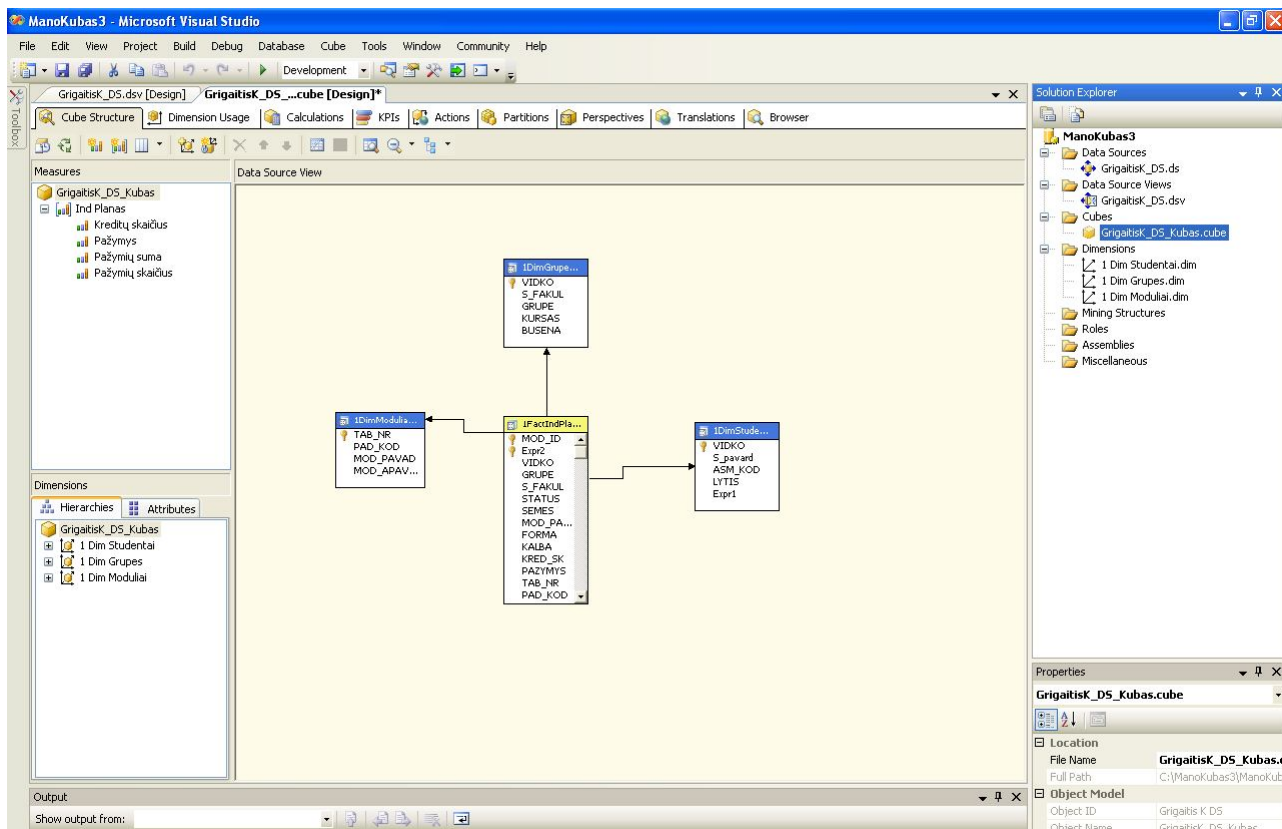


3.16 pav. Ketvirtojo sukurto kubo struktūra

Tai yra toks pat kubas kaip ir trečiasis, tik apkarpytas. Tiksliau tariant tai yra viena jo dalis. Palikta tik viena faktinė lentelė *IFaktStudentai* bei dimensijos, su kuriomis jis turėjo ryšius prieš tai sukurtame kube: *1DimGrupe*, *1DimStudentai* bei laiko dimensija. Jame yra mažiau duomenų, todėl *deployment* procesas vykdomas sparčiau. Šis kubas galėtų būti panaudotas ataskaitoms bei sukiniams kurti, kurioms naudojami tik *IFaktStudentai* faktinės lentelės sukuriama matai – studento rudens vidurkis, studento pavasario vidurkis, studento kreditų skaičius bei bendras studento vidurkis.

3.3.5. Penktasis sukurtas duomenų kubas

Penktajam kubui sukurti taip pat panaudota dalis tų pačių dimensinių bei faktinių lentelių kaip ir trečiajam kubui, tačiau su skirtinga faktine lentele nei ketvirtajam kube. Sukurto penktojo kubo struktūra pateikta 3.19 paveiksle.



3.17 pav. Penktojo sukurtos kubo struktūra

Kaip ir anksčiau aptartu atveju (ketvirto kubo) penktasis kubas yra taip pat trečiojo kubo dalis, tačiau kita. Palikta tik viena faktinė lentelė *IFaktIndPlanas* bei dimensijų lentelės, su kuriomis jis turėjo ryšius prieš tai sukurtame kube: *IDimGrupe*, *IDimStudentai* ir *IDimModuliai*. Šiose dimensijų lentelėse galime analizuoti duomenis tokiomis dimensijomis: modulio pavadinimas, modulio kodas, modulio angliškasis pavadinimas, fakultetas, grupė, kursas būseną, studento vardas pavardė, asmens kodas, bei lytis. Jame yra mažiau duomenų negu trečiajame kube, todėl *deployment* procesas irgi vykdomas sparčiau. Šis kubas galėtų būti panaudotas ataskaitoms bei sukiniams kurti, kurioms naudojami tik *IFaktIndPlanas* faktinės lentelės sukuriama matai – studento kreditų skaičius, modulio pažymys, pažymių suma bei kiekis, kurie sukurti norint apskaičiuoti bendrą studentų vidurkį.

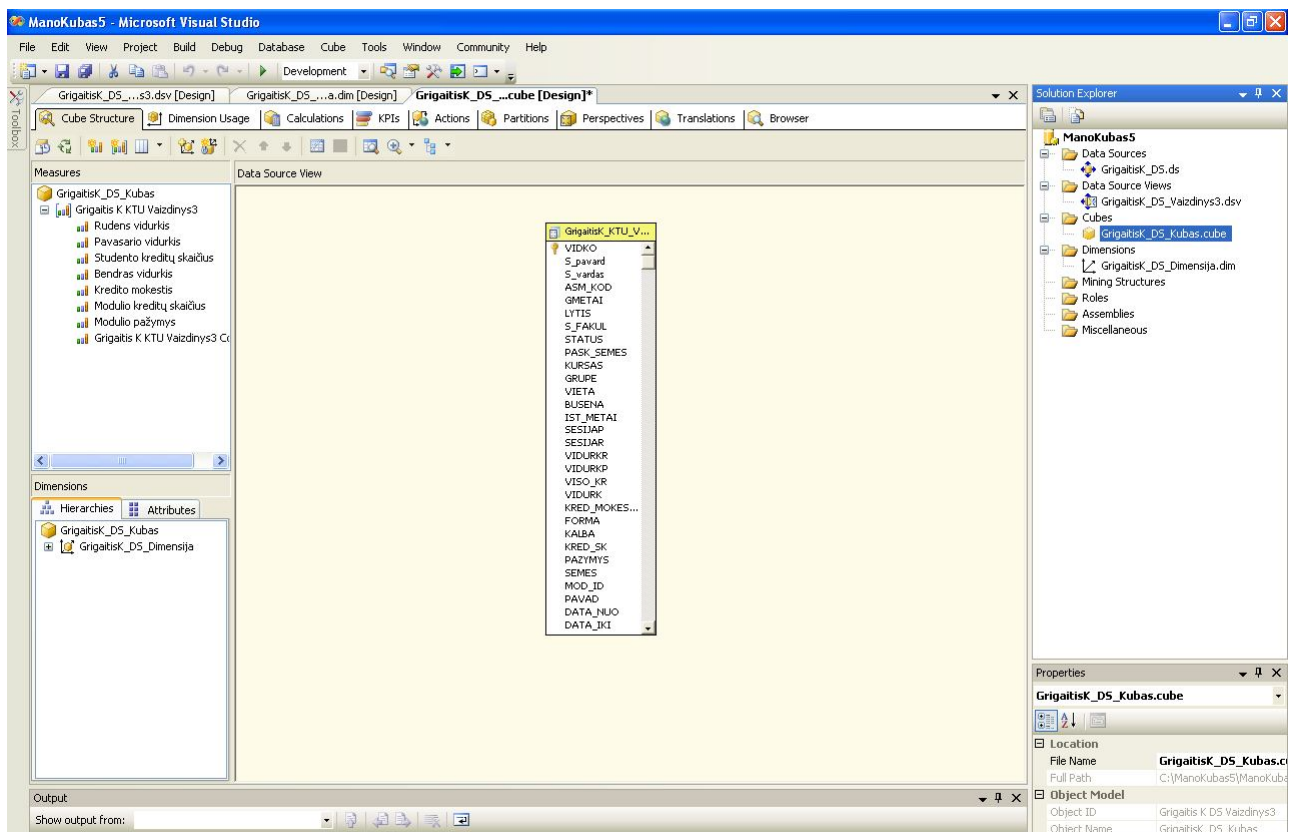
3.3.6. Šeštasis sukurtas duomenų kubas

Šeštasis kubas sukurtas pagal 3.20 paveiksle pateikto vaizdinio schemą.



3.18 pav. Šeštojo sukurto kubo vaizdinys

Sukurto šeštojo kubo struktūra pateikta 3.21 paveiksle.



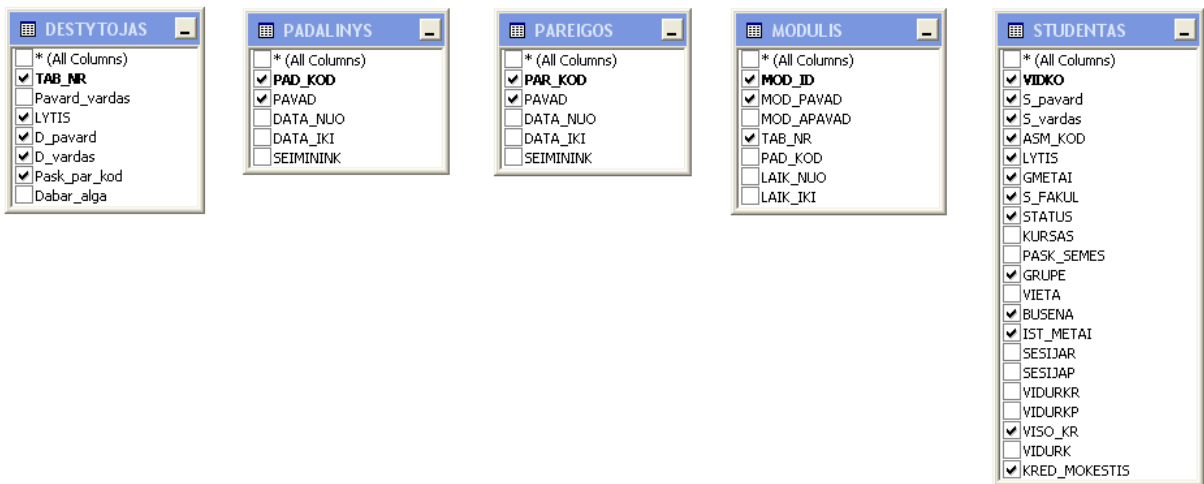
3.19 pav. Šeštojo sukurto kubo struktūra

Šis kubas sukurtas pagal vieną vaizdinį, kuris apima tarpusavyje sujungtas tris skirtingas duomenų lenteles. Kuriant tokį kubą yra nenurodomos konkrečios dimensinės bei faktinės lentelės, o tiesiog atskiros eilutės priskiriamos matų arba dimensijų grupei. Todėl, kaip matome X paveiksliuke, sukurto kubo struktūra vaizduojama kaip viena faktinė lentelė.

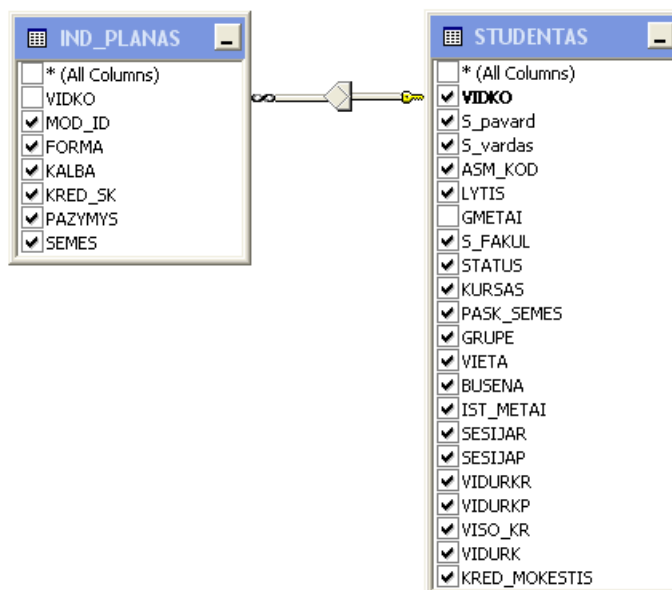
Šiame kube realizuotos šios dimensijos: studento asmens kodas, modulio pavadinimas, mokymosi forma, mokymosi kalba, semestras, studento būseną, studento fakultetas, studento gimimo metai, studento įstojimo metai, studento lytis ir vardas su pavarde. Analizuojant galime naudoti šiuos matavimus: studento bendras vidurkis, studento kredito mokestis, modulio kreditų skaičius, modulio pažymys, studento rudens bei pavasario vidurkiai ir studento kreditų skaičius.

3.3.7. Septintasis sukurtas duomenų kubas

Septintasis kubas sukurtas panaudojant 3.22 ir 3.23 paveiksluose pateiktas vaizdinių schemas, kurių kiekviena atspindi kuriamo kubo dimensinę arba faktinę lentelę:

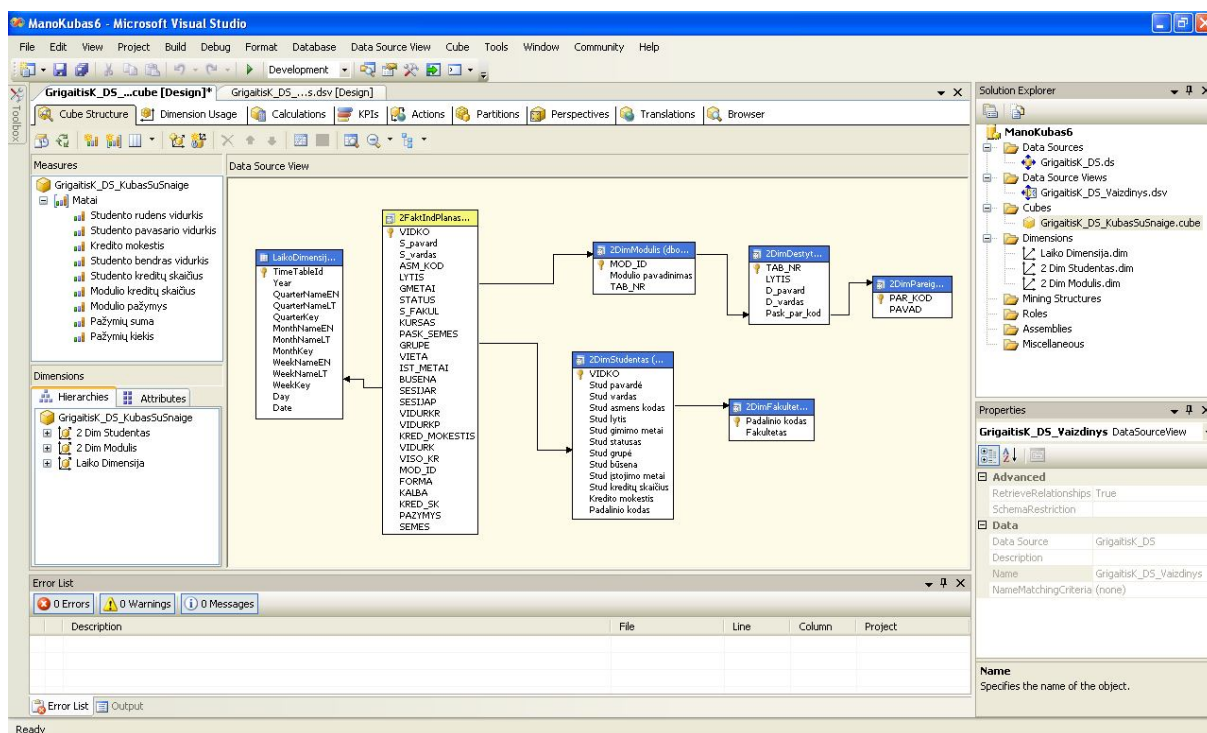


3.20 pav. Septintojo kubo dimensijos vaizdiniai: iš eilės - *2DimDestytojas*, *2DimFakultetas*, *2DimPareigos*, *2DimModulis* ir *2DimStudentas*



3.21 pav. Septintojo kubo faktinės lentelės *2FaktIndPlanas* vaizdinys

Sukurto septintojo kubo struktūra pateikta 3.24 paveiksle.

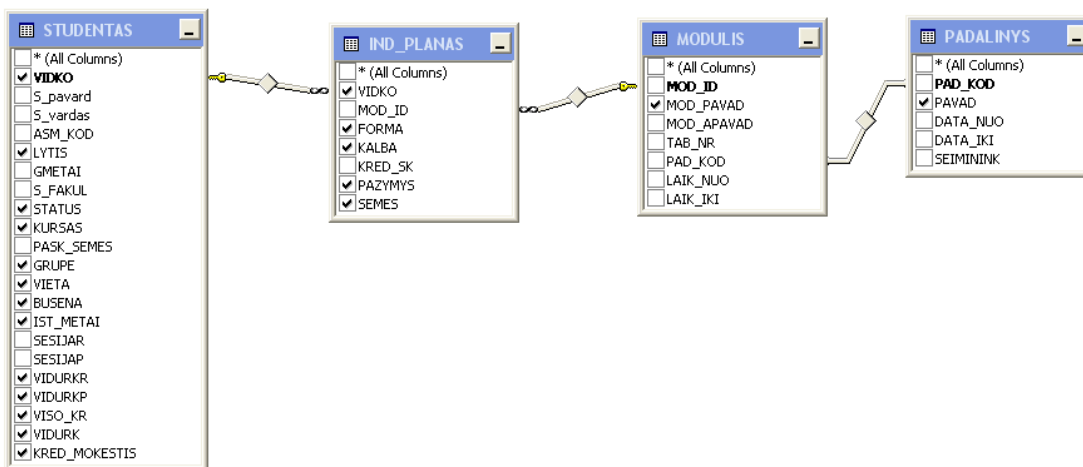


3.22 pav. Septintojo sukurto kubo struktūra

Šis kubas išskirtinis tuo, kad sukurtas pagal snaigės tipo schemą. Jis susideda iš trijų dimensinių lentelių grupių ir vienos centre esančios faktinės *2FaktIndPlanas* lentelės. Dvi dimensinės lentelės turi pogrupius: *2DimStudentas* dimensija turi pogrupį *2DimFakultetas*, o *2DimModulis* dimensija turi pogrupį *2DimDestytojas*, kuris savo ruožtu turi dar vieną pogrupį *2DimPareigos*. Snaigės tipo kubo privalumas tas, kad pavyzdžiui, kaip mano sukurtame kube nagrinėjant duomenis pagal dimensiją *2DimModulis* sužinome modulio pavadinimą ir modulio kodą, o pagal modulio kodą galime kreiptis į pogrupio dimensiją *2DimDestytojas*, kurioje galime sužinoti kokiam dėstytojui šis modulis priklauso ir tuo pačiu sužinoti šio dėstytojo asmeninę informaciją, tokią kaip dėstytojo lytis, vardas, pavardė bei pareigų kodas. Žinodami dėstytojo pareigų kodą, galime kreiptis į dar žemesnio lygio pogrupio dimensiją *2DimPareigos*, kurioje pagal dėstytojo pareigų kodą galime sužinoti kokios yra šio dėstytojo pareigos universitete. Atitinkamai nagrinėjant duomenis pagal dimensiją *2DimStudentas*, pagal padalinio kodą, galime kreiptis į žemesnio pogrupio dimensiją *2DimFakultetas*, iš kurios galime sužinoti kokiam fakultetui priklauso nagrinėjamas studentas.

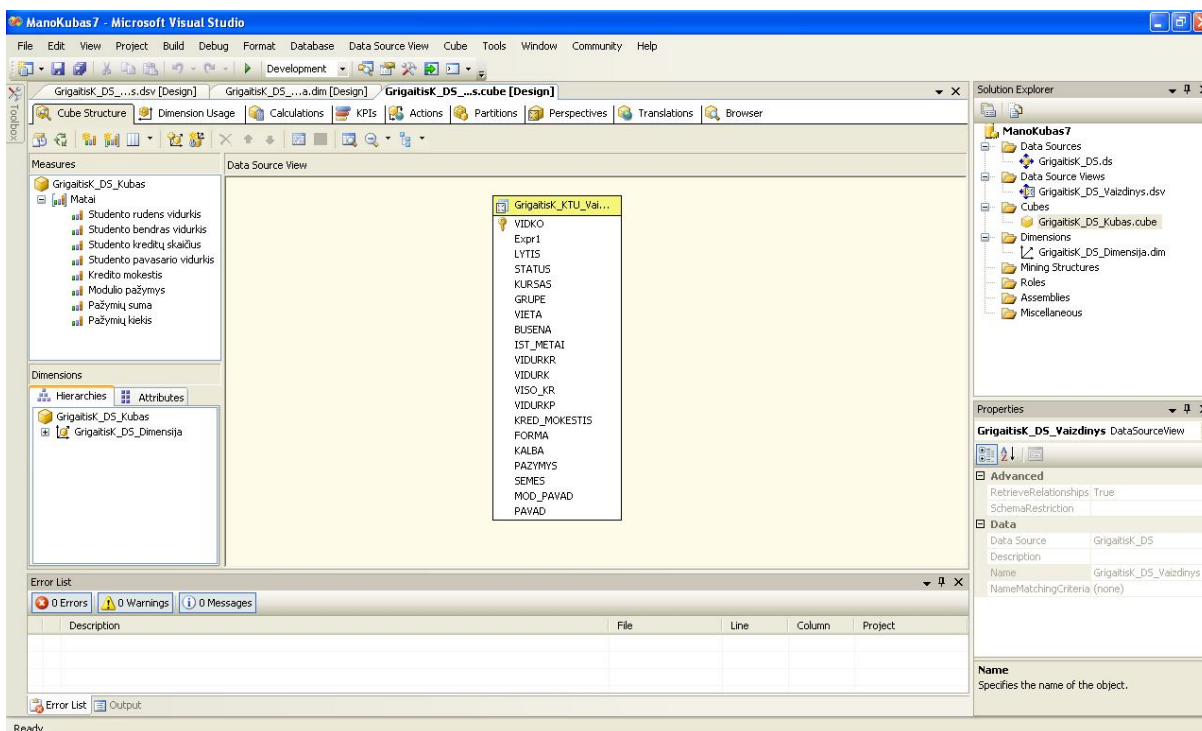
3.3.8. Aštuntasis sukurtas duomenų kubas

Aštuntasis kubas sukurtas pagal 3.25 paveiksle pateikto vaizdinio schemą:



3.23 pav. Aštuntojo sukurto kubo vaizdinys

Sukurto aštuntojo kubo struktūra pateikta 3.26 paveiksle.



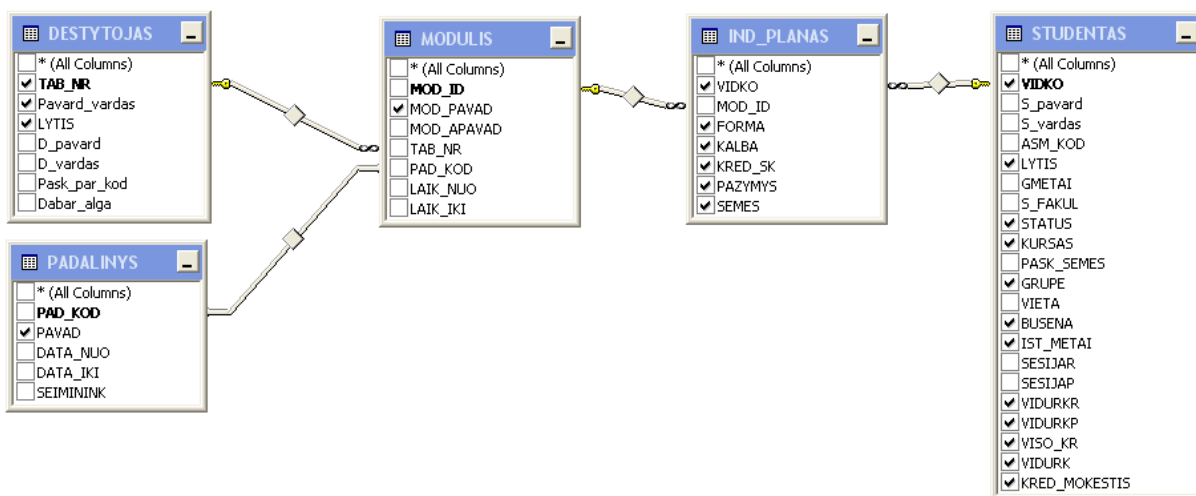
3.24 pav. Aštuntojo sukurto kubo struktūra

Aštuntojo kubo vaizdinio lentelių ryšiai skiriasi nuo sukurtos universiteto duomenų bazės diagramos ryšių. Šiame vaizdinyje lentelė padalinys ryšiu sujungta ne su lentele *studentas*, o su lentele *modulis*, sukuriant ryšį per *PAD_KOD* lauką. Ankstesniuose kubuose

studentų informaciją buvo galima lyginti fakultetų lygiu, o atlikus šiuos pakeitimus bei sukūrus šį kubą, galime informaciją lyginti katedrų lygyje. Taigi sukūrus šį kubą, buvo išplėstas analizės funkcionalumas, leidžiantis įvairiais būdais lyginti vienas katedras su kitomis.

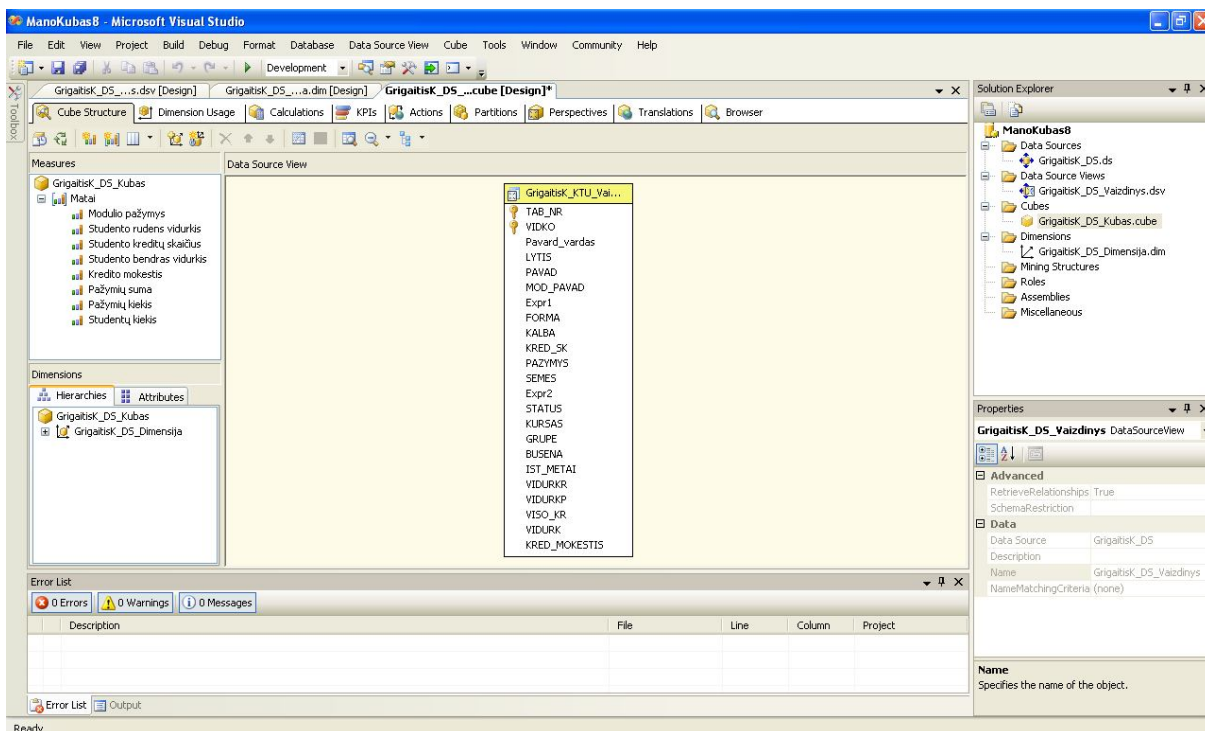
3.3.9. Devintasis sukurtas duomenų kubas

Devintasis kubas sukurtas pagal 3.27 paveiksle pateikto vaizdinio schemą:



3.25 pav. Devintojo sukurto kubo vaizdinys

Sukurto devintojo kubo struktūra pateikta 3.28 paveiksle.

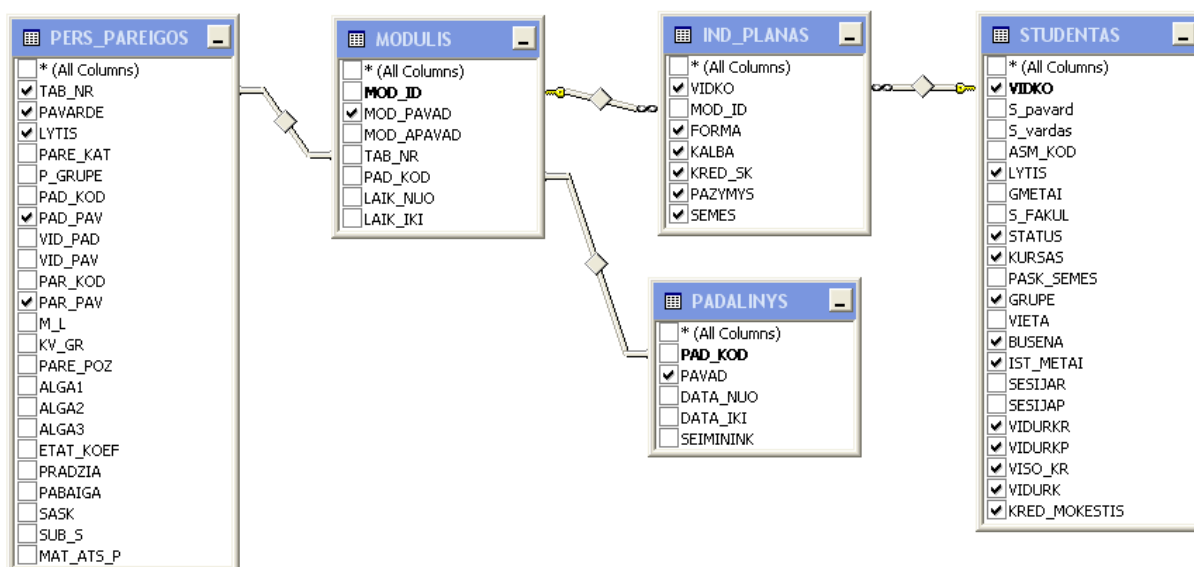


3.26 pav. Devintojo sukurto kubo struktūra

Šis kubas yra labai panašus į prieš tai sukurtą aštuntąjį kubą, tačiau jame papildomai prijungta lentelė *dėstytojas*. Tai išplėtė kubo funkcionalumą, įgalinant papildomai lyginti ne tik studentų, bet ir dėstytojų informacija katedrų lygyje.

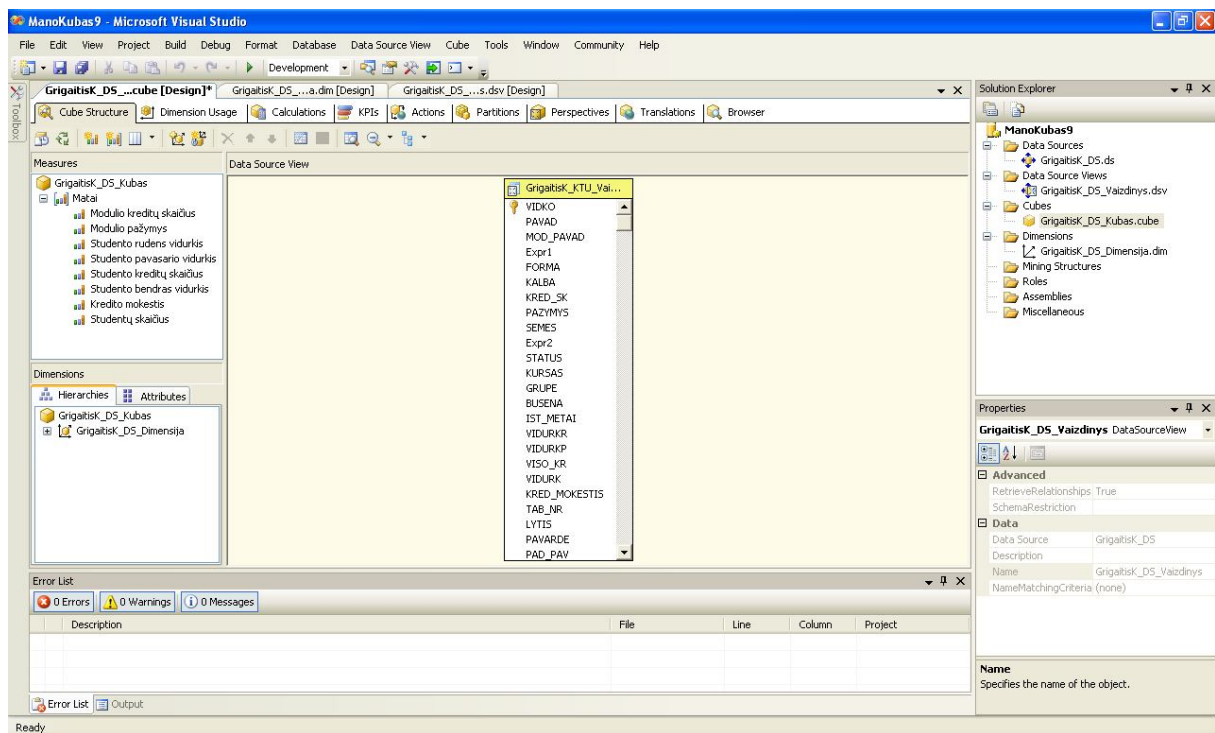
3.3.10. Dešimtas sukurtas duomenų kubas

Dešimtas kubas sukurtas pagal 3.29 paveiksle pateikto vaizdinio schema:



3.27 pav. Dešimtojo sukurto kubo vaizdinys

Sukurto dešimtojo kubo struktūra pateikta 3.30 paveiksle.



3.28 pav. Dešimtojo sukurto kubo struktūra

Dešimtojo kubo struktūra yra lygiai tokia pati kaip prieš tai sukurto devintojo kubo. Tačiau jam skiriamas žymiai didesnis dėmesys, nes jame panaudota lentelė, kurioje yra tikri universiteto dėstytojų duomenys. Remiantis šio kubo rezultatais, galima atlikti duomenų analizę, kuri atspindėtų realią situaciją universitete. Atliktos analizės tyrimo rezultatai bus pateikti kitame skyriuje

Kaip buvo minėta duomenų bazės kūrimo skyriuje, siekiant užtikrinti informacijos konfidencialumą, anksčiau visur buvo naudojami sumaišyti dėstytojų vardų ir pavardžių duomenys.

Norėdamas atlikti duomenų kubų kūrimo ir saugojimo modelių tyrimą, stengiausi sukurti kuo daugiau ir kuo įvairesnių kubų, kurie pasižymėtų skirtingomis savybėmis:

- apimtų skirtingus duomenis bei duomenų kiekius;
- būtų kuriami skirtingais keliais: vieni kuriami iš programa *SQL Server 2005* sukurtų vaizdinių, apimančių kelias lenteles bei sujungtų ryšiais, kiti iš vaizdinių, kurių kiekvienas atspindi konkrečią dimensiją, kuri bus kuriama kube ir kurių tarpusavio dimensijų ryšiai kuriami *SQL Server Business Intelligence Development Studio* programa;
- būtų skirtingų tipų: sukurti žvaigždės bei sniegės modeliais.

Iš viso buvo sukurta dešimt kubų. Visi jie naudojo tą pačią duomenų bazę, lenteles bei duomenis, todėl daugeliu atvejų vieni į kitus panašūs, tačiau skiriasi apimamų duomenų kiekiu, kūrimo ypatybėmis, struktūra ir t.t.

3.4. Testavimas

Testavimas buvo atliktas dviejuose skirtingų pajėgumų bei operacinių sistemų kompiuteriuose. Tiriamųjų kompiuterių charakteristikos:

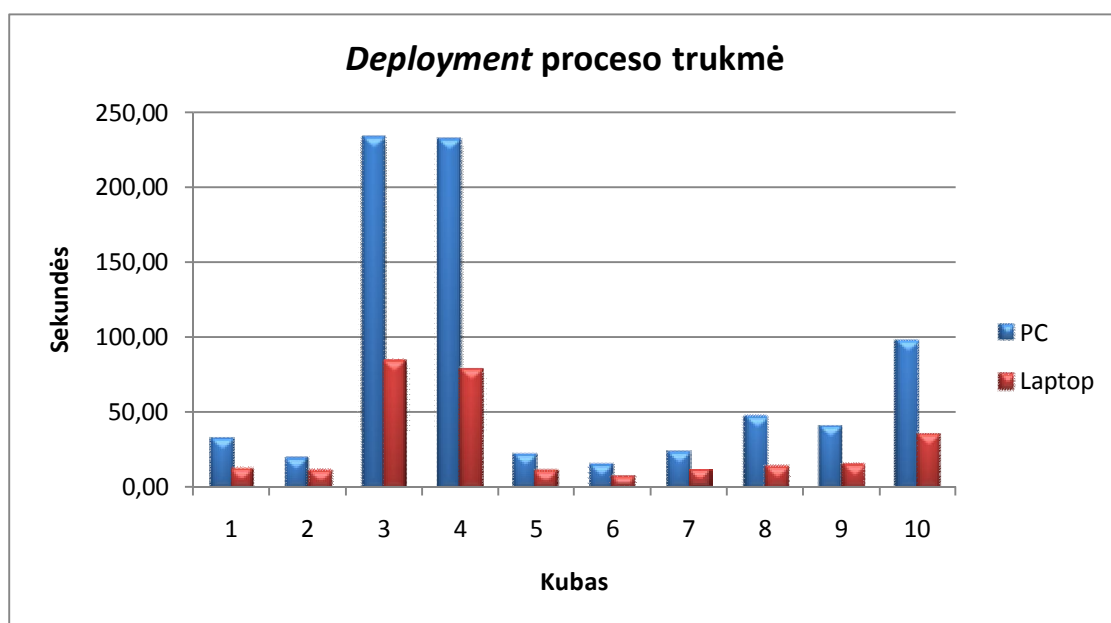
Asmeninis kompiuteris:

- Procesorius AMD Athlon 2700+ 2.16 Ghz
- 768 MB RAM atminties
- Windows XP Professional SP3

Nešiojamas kompiuteris:

- Procesorius Inter Core 2 Dou T8300 2.4 Ghz
- 3 GB RAM atminties
- Windows Vista Home Premium SP1

Daugiausia sistemos resursų reikalaujantis veiksmas yra *deployment* proceso vykdymas, kurio metu apdorojami duomenų kubo duomenys. Kubų *deployment* proceso vykdymo laiko priklausomybė nuo testuojamojo kompiuterio grafikas pateiktas 3.31 paveiksle.



3.29 pav. *Deployment* proceso vykdymo laiko priklausomybė nuo techninės įrangos pajėgumo

Kaip matome iš grafiko, kompiuterio techninės įrangos pajėgumas turi didelę įtaką sistemos greitaveikai ir su lėtesniu kompiuteriu *deployment* proceso vykdymo laikas vidutiniškai du kartus ilgesnis negu su greitu kompiuteriu. Taip pat paveiksle matome, kad esant sudėtingai kubo struktūrai, kubo duomenų apdorojimo laikas žymiai išauga (pvz., trečiasis ir ketvirtasis kubai).

3.5. Sistemos realizacijos išvados

- Sistemos realizacija buvo pradėta sukuriant duomenų saugyklą, kurios pagrindas yra KTU palikiminė duomenų bazė su universiteto dėstytojų, studentų, modulių bei kita informacija, kuri bus naudojama duomenų analizei.
- Duomenų saugykla sukurta panaudojant tarpinę programinę įrangą, kuri leido konvertuoti palikiminės duomenų bazės duomenis į *Microsoft SQL Server 2005* programinę įrangą.
- Siekiant atlikti kuo išsamesnę duomenų analizę duomenų kubų kūrimo metodu, buvo sukurta dešimt įvairių rūšių kubų, pradedant paprastais, žvaigždės tipo ir baigiant sudėtingu – snaigės tipo, kurie leido atskleisti bei išsamiau analizuoti duomenų kubų kūrimo proceso ypatybes.

4. KTU DS INTELEKTUALI ANALIZĖ

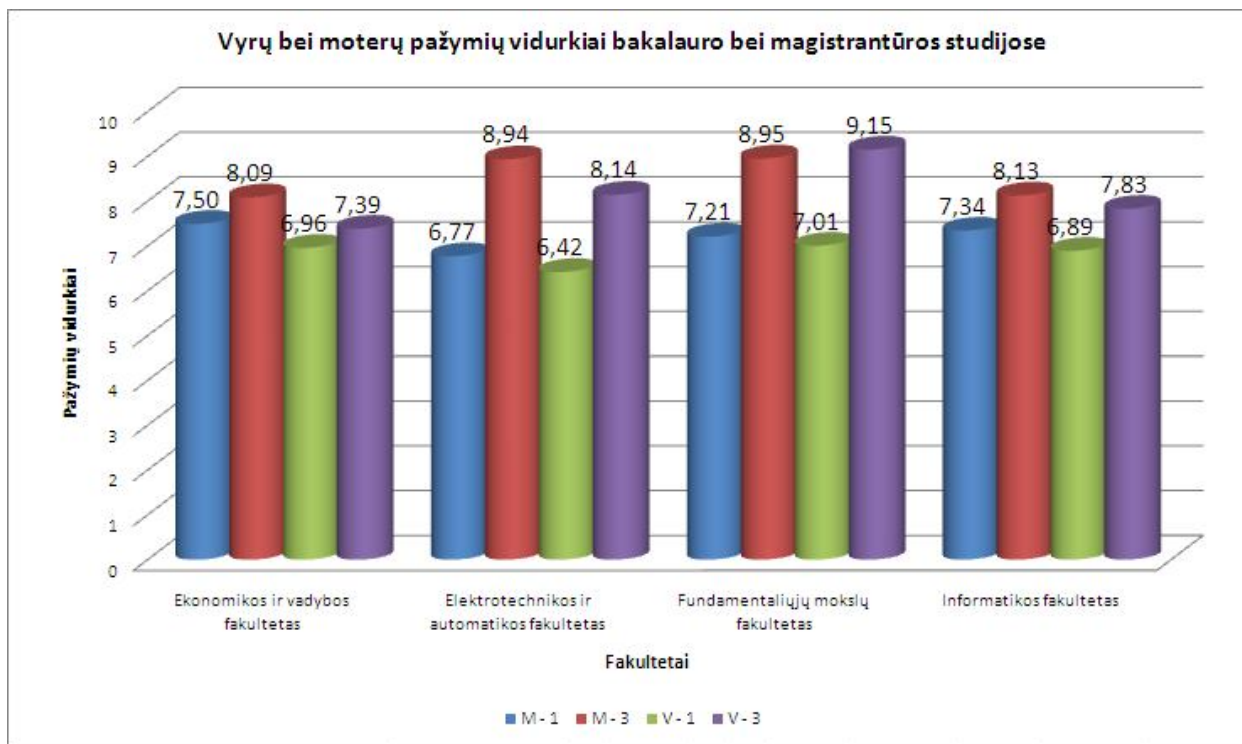
4.1. Intelektualios analizės tikslas

Šios dalies tikslas yra atlikti išsamią universiteto duomenų analizę ir atrasti sukinius bei indikatorines diagramas, kurios parodytų universiteto studijų trūkumus bei silpnąsias vietas ir padėtų spręsti esamas problemas. Tikslui pasiekti buvo naudojami ankstesniame skyriuje sukurti duomenų kubai, kurie leido analizuoti universiteto duomenis įvairiais pjūviais bei matais.

KTU duomenų saugykla apima universiteto veiklos duomenis nuo 1999 m. iki 2002 m. Šioje duomenų bazėje yra saugoma archyvinė studentų, dėstytojų, absolventų, modulių, individualių planų ir kita universiteto informacija.

4.2. Intelektualios analizės rezultatai

Norint atskleisti duomenų kubų analizės teikiamą naudą universitetui, buvo sukurta daug įvairių sukinių bei indikatorinių diagramų. Kaip jau minėjau anksčiau, sukiniai kuriami *Microsoft Excel 2007* programa, o indikatorinės diagramos *ProClarity Desktop Professional 6.3* programų paketu. Keli pavyzdžiai, kurių rezultatai galėtų būti naudingi universitetui, bus pateikti toliau.



4.1 pav. Vyrų bei moterų pažymių vidurkiai bakalauro bei magistrantūros studijose pagal fakultetus

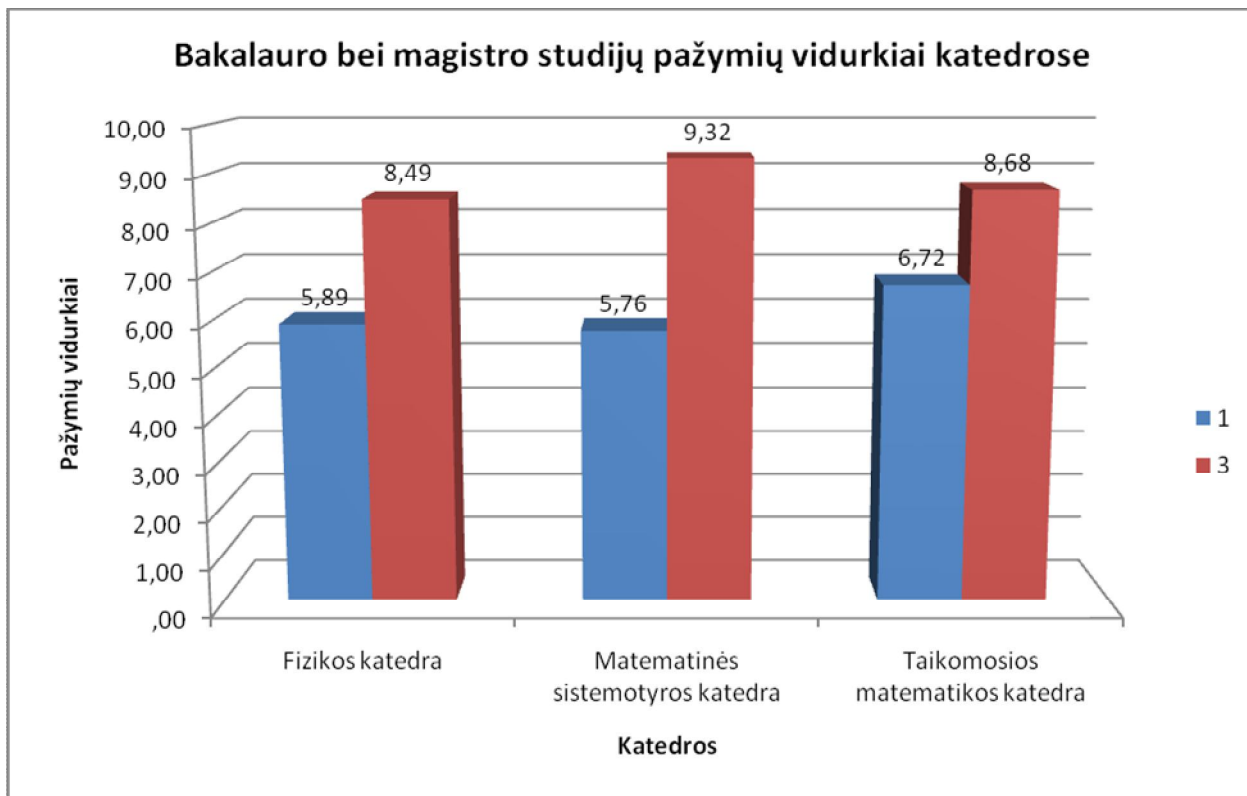
4.1 sukinys parodo studentų, studijuojančių bakalauro bei magistro studijose bendrą vidurkį pagal jų lytį įvairiuose fakultetuose. *M-1* stulpeliai rodo moterų vidurkius bakalauro studijose, *M-3* stulpeliai rodo moterų vidurkius magistrantūros studijose, *V-1* stulpeliai rodo vyrų vidurkius bakalauro studijose ir *V-3* stulpeliai – vyrų vidurkius magistrantūros studijose.

Šiame paveiksle galima matyti, kad bendra studijuojančiųjų pažymių vidurkių tendencija yra panaši. Tai yra, bakalauro studijų studentai mokosi mažesniais pažymiais nei magistrantūros. Tačiau negalima teigti, kad lytis yra parametras, pagal kurį būtų galima nuspręsti studento mokymosi lygį. Šioje diagramoje labiausiai išsiskiria Fundamentaliųjų mokslų fakulteto magistro studijų pažymių vidurkiai, kadangi jie yra aukščiausi. Taip pat šiame fakultete skirtumas, tarp bakalauro ir magistrantūros pažymių vidurkių, yra aukščiausias. Dėl šių priežasčių Fundamentaliųjų mokslų fakultetas pasirinktas tolimesnei analizei.

Fundamentaliųjų mokslų fakultetas susideda iš trijų katedrų: Fizikos, Matematinės sistematikos ir Taikomosios matematikos katedrų. Todėl jos bus detaliau panagrinėtos.

Kaip matyti 4.2 paveiksle, kuriame pateikta fundamentaliųjų mokslų fakulteto katedrų bakalauro bei magistro studijų vidurkiai, matematinės sistematikos katedroje studijuojančiųjų magistro studijas pažymiai yra aukščiausi, o tuo tarpu studijuojančiųjų bakalauro studijas pažymiai yra mažiausi. Remiantis universiteto istoriniais duomenimis, paprastai į magistrantūros studijas stoja tik dalis geriausiai bakalauro studijas baigusiu studentų. Todėl

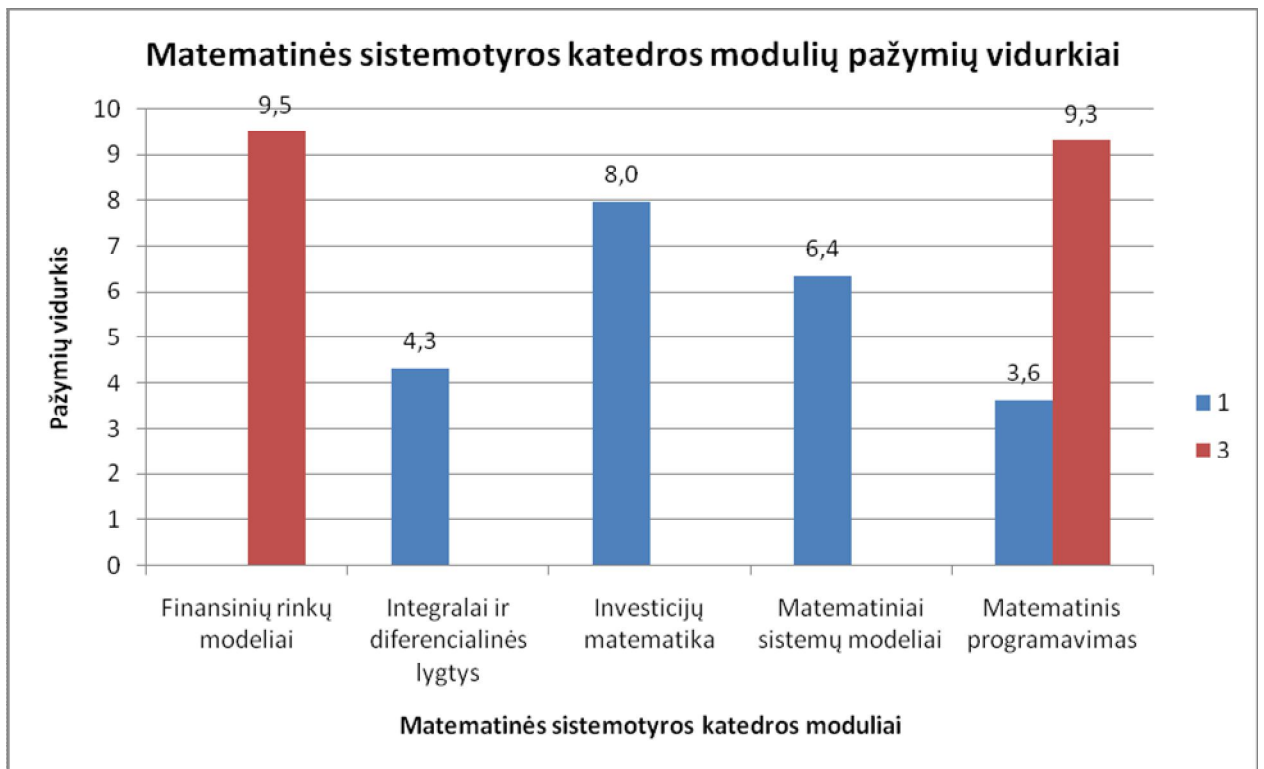
diagramoje matomi rezultatai yra paaiškinami logiškai. Tačiau iškyla klausimas dėl kokių priežasčių susidaro toks reikšmingas pažymių vidurkių skirtumas.



4.2 pav. Bakalauro bei magistro studijų vidurkiai fundamentaliųjų mokslų fakulteto katedrose

Norint išsiaiškinti tokį susidariusį skirtumą, bus panagrinėta Matematinės sistemos tyros katedra, jos moduliai, dėstytojai bei studentai.

Fundamentaliųjų mokslų fakulteto Matematinės sistemos tyros katedroje dėstomi tokie moduliai: finansinių rinkų modeliai, integralinės lygtys ir diferencialai, investicijų matematika, matematiniai sistemų moduliai bei matematinis programavimas. Pateiktame 4.3 paveiksle matyti, kad du iš šių modulių yra dėstomi magistro, o keturi – bakalauro studijose. Modulis matematinis programavimas dėstomas tiek bakalauro, tiek magistrantūros studijose, todėl kyla klausimas kodėl pažymių vidurkių skirtumas toks didelis. Analizės metu buvo nustatyta, kad šis modulis tiek bakalaurantams, tiek magistrantams dėstomas to pačio dėstytojo – Juliaus Mukulio (dėl dėstytojų konfidencialumo užtikrinimo, dėstytojų vardai bei pavardės yra sumaišytos).

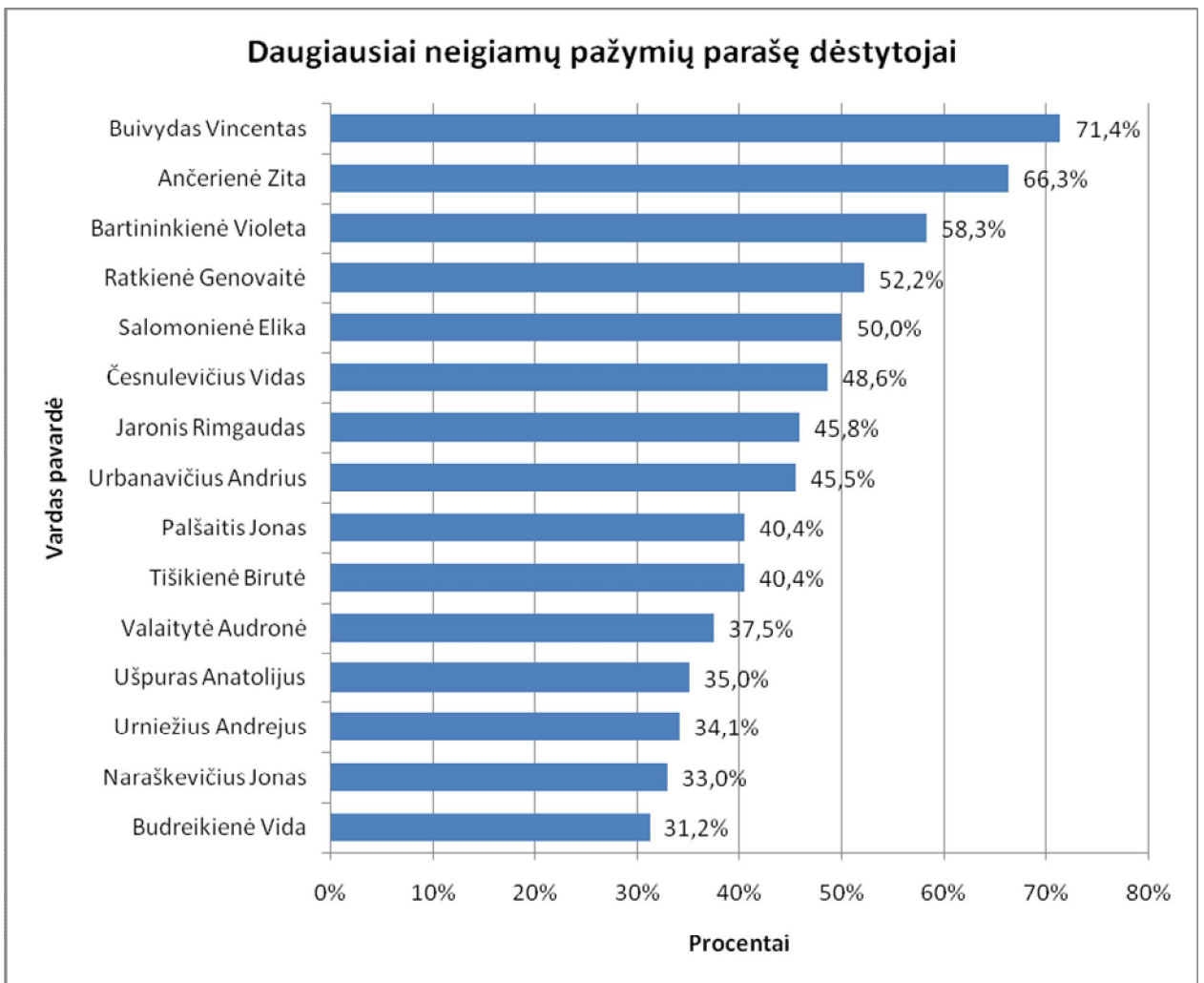


4.3 pav. Matematinės sistemotyros katedros modulių pažymių vidurkiai

Tolesnis šios analizės žingsnis galėtų būti su konkrečiu dėstytoju dėl jo modulio ir rašomų atitinkamų pažymių pagrįstumo. Naudojantis šiame magistriniame darbe sukurta KTU duomenų saugyklos analizės metodų sistema, būtų galima analizuoti papildomus rodiklius, susijusius su šiuo dėstytoju. Pavyzdžiui, neišlaikiusių bei išlaikiusių egzaminus studentų skaičiaus santykį, pareigas, gaunamą atlyginimą.

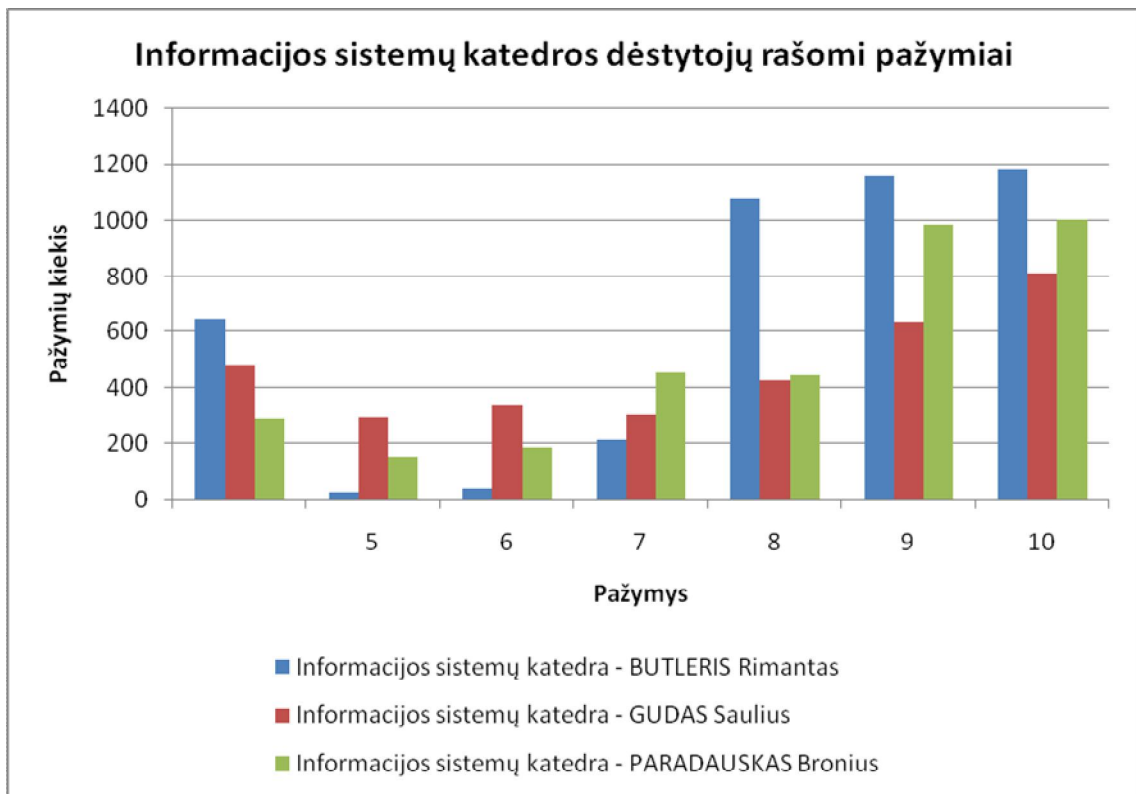
Šiuo pavyzdžiu buvo parodyta kaip galima atlikti konkretaus fakulteto, konkrečios katedros bei modulio analizę remiantis studijuojančiųjų studentų vidurkiu. Tokią analizę būtų galima pritaikyti kitiems pasirinktiems fakultetams, jų katedroms bei moduliams. Taip pat būtų galima pakeisti pagrindinį rodiklį, t.y. pažymių vidurkį, į studijuojančio studento būseną, kuri parodo ar studentas studijuoja, ar išbrauktas iš sąrašų ir t.t.

Taip pat reikšmingą dėstytojų analizę galima atlikti įvertinant dėstytojų rašomus studentams pažymius. Norint atlikti tokią analizę buvo surinkti visi universiteto dėstytojai ir jų rašomų pažymių kiekiai. Toliau buvo paskaičiuota procentinė dalis neigiamų pažymių kiekio nuo visų dėstytojų parašytų pažymių. Galiausiai buvo atrinkti tie dėstytojai, kurie yra parašę daugiau negu 30 proc. neigiamų pažymių, lyginant su visais pažymiais. 4.4 paveiksle pateikta diagrama, kurioje matome šiuos atrinktus dėstytojus.



4.4 pav. Daugiausiai procentaliai neigiamų pažymių parašę dėstytojai universitete

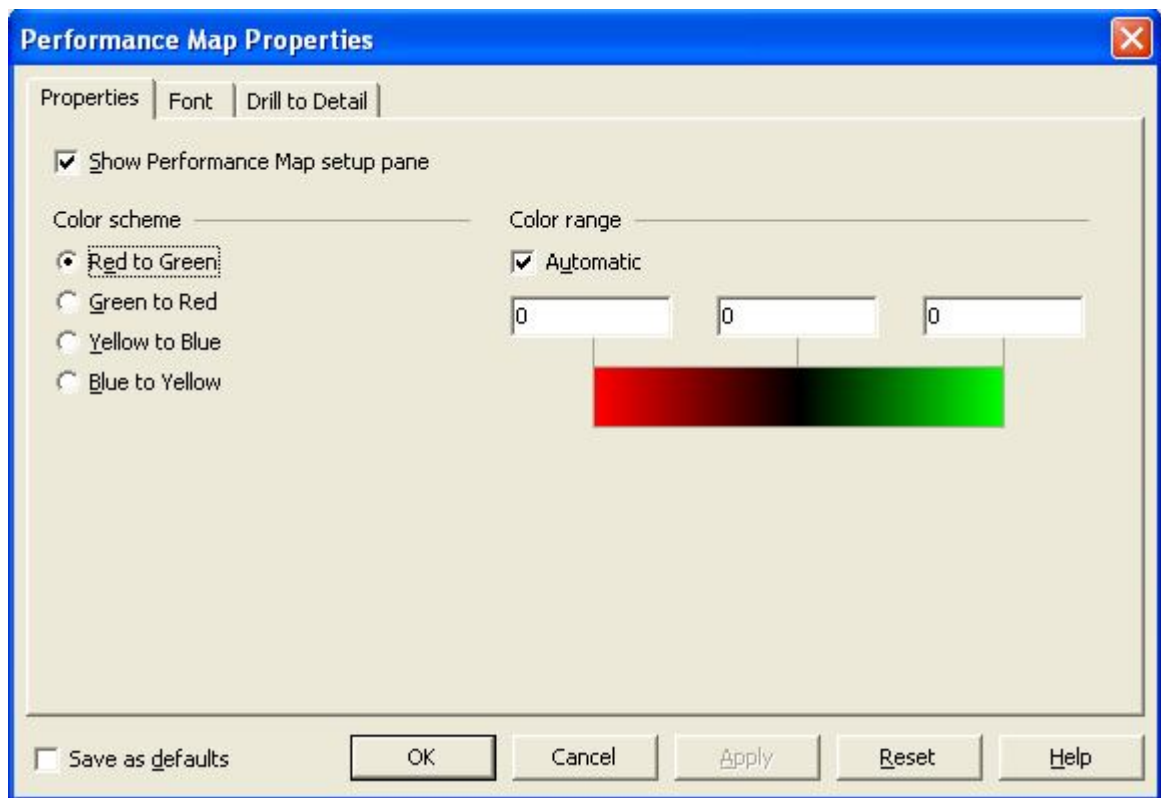
Kaip buvo aprašyta šio darbo realizacijos dalyje (žr. 3 skyrius), paskutinis duomenų kubas buvo sukurtas su tikrais universiteto dėstytojų duomenimis. Toliau bus pateikta keleto Informacijos sistemų katedros dėstytojų rašomų pažymių suvestinė.



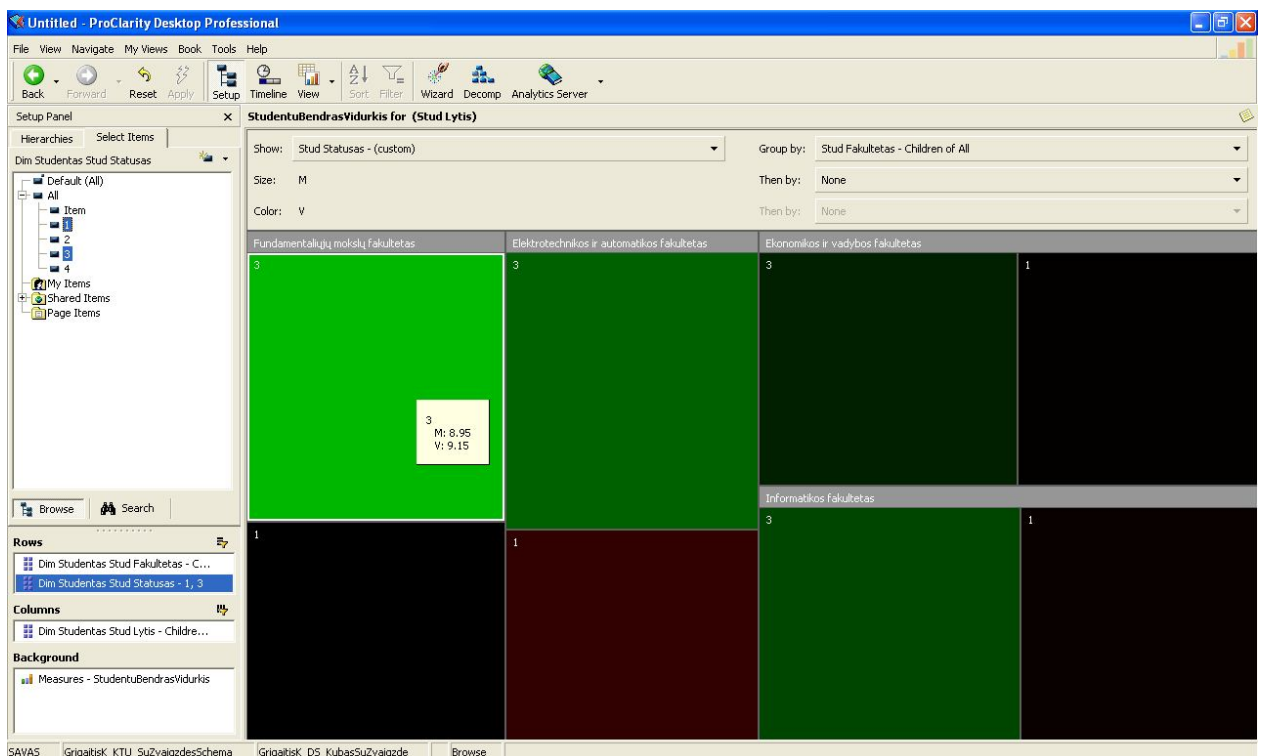
4.5 pav. Informacijos sistemų katedros dėstytojų rašomų pažymių kiekiai

Kaip matome, 4.5 paveikslė diagramoje yra pavaizduota trys Informacijos sistemų katedros dėstytojai. Diagramos horizontalioje ašyje pažymėti pažymiai dešimtbalėje sistemoje, o pirmasis stulpelis žymi neigiamus pažymius. Nesunku pastebėti, kad visi dėstytojai daugiausia parašo gerų pažymių – t.y. aštuonetų, devynetų ir dešimtų. Kalbant apie žemesnius pažymius, išsiskiria dėstytojo Rimanto Butlerio įvertinimai, kadangi jis vietoj pažymių penkių, šešių ir septynių, dažniau rašo neigiamus pažymius. Tuo tarpu kitų dėstytojų pažymių kiekio išsidėstymas yra tolygesnis.

Dabar bus pateiktos kelios indikatorinės *ProClarity Analytics* diagramos, kurios didelius duomenų kiekius atvaizduoja spalvų skalėje, taip leisdamos praplėsti duomenų kubų analizės metodus. Diagramos parametruose nurodoma kokios spalvos vaizduoja teigiamus, o kokios neigiamus rezultatus bet skaitine reikšme apibrėžiamos jų ribos. Mano atveju raudona spalva reiškia blogiausius rezultatus, o ryškiai žalia – geriausius. Galime pasirinkti iš keturių spalvinių vaizdavimo būdų. Parametrų nustatymo langas pavaizduotas 4.6 paveiksle.



4.6 pav. Performance Map vaizdavimo parametrų nustatymo langas

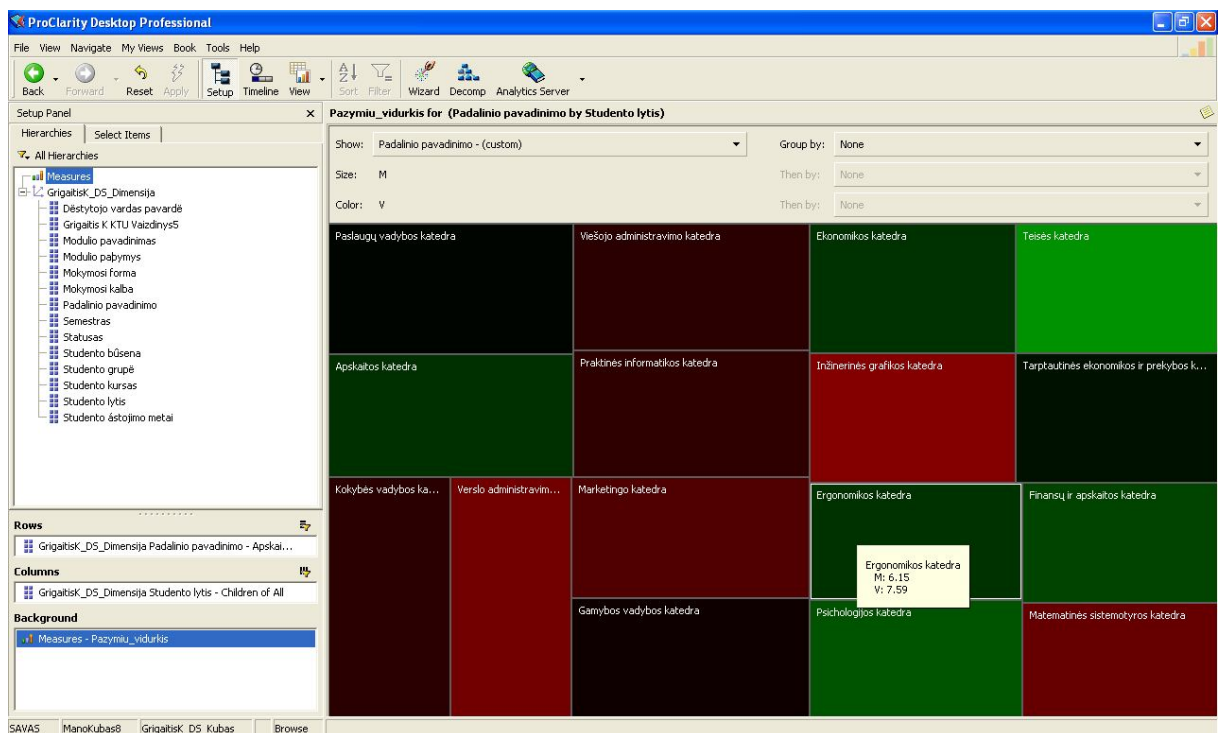


4.7 pav. Vyrų bei moterų pažymių vidurkiai bakalauro bei magistratūros studijose pagal fakultetus Performance Map ProClarity Analytics diagrama

Pirmoji indikatorinė diagrama pavaizduota 4.7 paveiksle bei yra sukurta panaudojant tokias pačias dimensijas ir matus kaip ir pirmojo sukinio kūrimo metu (žr. 4.1 pav.). Iš jos taip

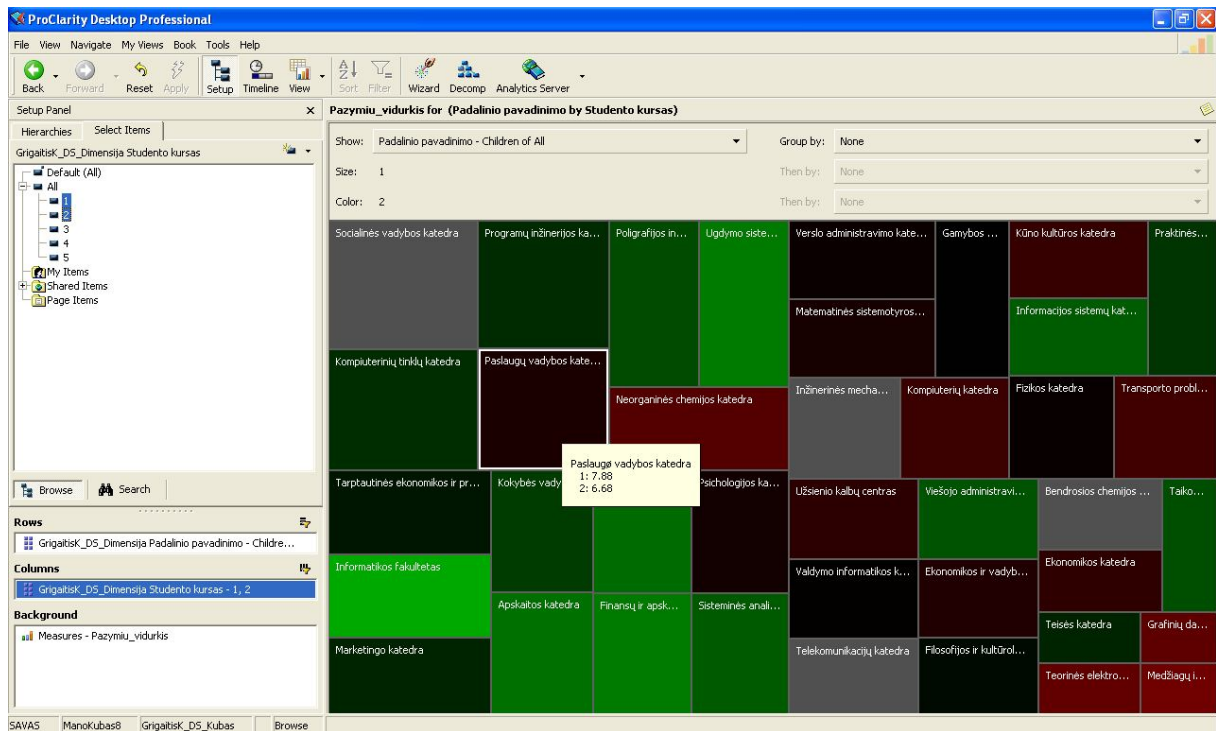
pat matome jog aukščiausi įvertinimai yra Fundamentaliųjų mokslų fakulteto magistro studijose, o žemiausi Elektrotechnikos ir automatikos fakulteto bakalauro studijose. Tačiau nagrinėjant indikatorinę diagramą, suku tiksliai įvertinti koks skirtumas yra tarp pažymių. Šioje diagramoje duomenų spalvų intervalas nurodytas nuo 4 iki 10. Tai reiškia, kad žaliausia spalva bus rodomi langeliai arčiausiai 10, juoda spalva žymi 7, o raudoniausia spalva ties 4.

Siekiant išanalizuoti kuo daugiau skirtingų duomenų, toliau bus nagrinėjamos indikatorinės diagramos, kurių parametrų deriniai dar nebuvo panaudoti kuriant įvairius sukinius.



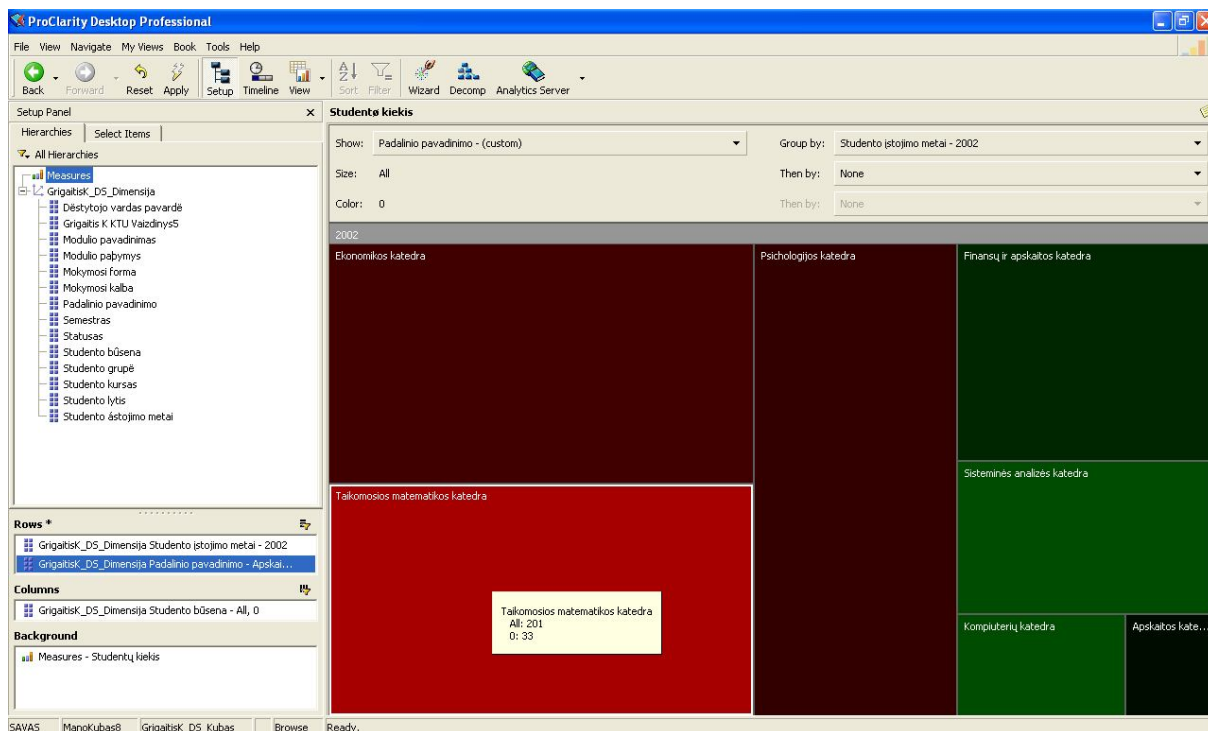
4.8 pav. Studentų vidurkiai pagal jų lytį įvairiose katedrose diagrama

4.8 paveiksle pavaizduota diagrama, kuri parodo studentų vidurkius pagal jų lytį įvairiose katedrose. Kaip matome iš paveiksluko, ergonomikos katedroje moterų vidurkis yra 6,15, o vyrų – 7,59. Pagal šią diagramą galime spręsti kurioje katedroje mūsų universiteto studentams sunkiausiai sekasi studijuoti.



4.9 pav. Studentų bendras vidurkis įvairiose katedrose pagal studijų pirmą bei antrą kursą
diagrama

4.9 paveiksle pavaizduota diagrama, kurioje matome studentų bendrą vidurkį įvairiose katedrose pagal pirmą bei antrą kursą. Kaip matome iš pavyzdžio, paslaugų vadybos katedroje pirmo kurso studentų bendras vidurkis yra 7,88, o antro kurso vidurkis yra jau tik 6,68. Tai parodo, kad pirmame kurse šioje katedroje dėstomi moduliai yra lengvesni nei dėstomi antrame kurse, todėl pvz. galima pasiūlyti universitetui sukeisti sunkesnius modulius su lengvesniais tarp kursų, kad studentams būtų lengviau studijuoti. Tai padėtų pagerinti studijų kokybę.



4.10 pav. Išbrauktų studentų kiekiai įvairiose katedrose

4.10 paveiksle pavaizduota diagrama, kurioje parodyta koks yra išbrauktų studentų, įstojusių 2002 metais, kiekis įvairiose pasirinktose katedrose. Langelių dydis atitinka bendrą įvairiose būsenose esantį studentų kiekį, o spalva – 0 būsenoje esančių studentų skaičių, kitaip tariant išbrauktų iš studentų sąrašų studentų skaičių. Pagal langelių spalvas aiškiai matyti kokiose katedrose išbrauktų studentų kiekis didžiausias. Kuo spalva ryškiau raudona, tuo šis skaičius didesnis bei atvirkščiai – kuo spalva ryškiau žalia, tuo šis skaičius mažesnis. Kaip matome Taikomosios matematikos katedroje išbrauktų studentų skaičius didžiausias – 33. Esant poreikiui, diagramą galima padaryti ne pagal katedras, o pagal fakultetus.

Išnagrinėjus kelis sukinių bei indikatorinių diagramų pavyzdžius, galima tarpusavyje palyginti jų pritaikomumo galimybes. Indikatorinės diagramos paprastai yra labiau tinkamos esant dideliame atributų kiekiui (pvz., kaip 4.9 pav.), kadangi spalvos leidžia greitai įvertinti pateiktus duomenis, išskirti blogiausius (raudoniausius) ir geriausius (žaliausius) rezultatus. Tačiau jei norima tiksliai matyti reikšmių skirtumus, geriau braižyti sukinius (pvz., kaip 4.1 pav.). Pritaikius sukinius ir indikatorines diagramas universiteto duomenų analizei, galima teigti, kad abi priemonės papildo vieną kitą, ir todėl atliekama išsamesnė analizė.

4.3. Intelektualios analizės išvados

- Panaudojant sudarytą kubų duomenis *Excel 2007* programa buvo kuriami įvairūs duomenų sukiniiai, kurių pagalba įvairiais pjūviais atlikta universiteto duomenų analizė.
- Ši analizė suteikia galimybę įvertinti universiteto studijas įvairiais aspektais bei priimti atitinkamus sprendimus. Pirmiausiai remiantis studentų pažymių vidurkiu buvo analizuojami pasirinkti universiteto fakultetai, bei ieškoma priežasčių, kodėl vienuose fakultetuose studijų pažymių vidurkiai yra labai aukšti arba žemi. Analizės metu buvo atrasta, kad konkretus dėstytojas naudoja neteisingą vertinimo sistemą.
- Buvo atliekamas dėstytojų vertinimas pagal jų rašomus neigiamus pažymius. Atrinkti daugiausiai neigiamų pažymių parašę dėstytojai. Pasinaudojus tikrais duomenimis sukurtu duomenų kubu, buvo atlikta pasirinktų Informacijos sistemų katedros dėstytojų rašomų pažymių analizė.
- Naudojant *ProClarity Analytics* programinę įrangą, buvo nubraižytos indikatorinės diagramos, leidžiančios vizualizuoti universiteto duomenų analizės rezultatus. Sukinių ir indikatorinių diagramų pritaikymas intelektualiai duomenų analizei suteikia galimybę gauti išsamesnius rezultatus.
- Šioje darbo dalyje buvo išnagrinėti tik keli universiteto duomenų analizės pavyzdžiai, kurie atskleidžia įvairių universiteto rodiklių ypatybes. Analizėje nagrinėjami pavyzdžiai buvo pasirinkti, atsižvelgiant į jų rezultatų reikšmę bei galimą pritaikomumą praktikoje. Tačiau, norint išplėsti atliekamą analizę galima įtraukti ir daugiau rodiklių.

IŠVADOS

1. Duomenų kubas apibrėžiamas savo matais ir dimensijomis. Yra dvi pagrindinės duomenų kubo kūrimo schemas – žvaigždės bei snaigės tipo. Egzistuoja įvairi programinė įranga, skirta duomenų verslo analizei atlikti, kuriomis kuriami duomenų kubai. *Microsoft SQL Server 2005* populiariausia duomenų verslo analizės programinė įranga. Papildomai galima naudoti *ProClarity Desktop Professional 6.3* programą, kuri išplečia duomenų kubų analizės galimybes.
2. Projektuojant sistemą, buvo naudojama unifikuota modeliavimo kalba (*UML*), kuria buvo nubraižytos panaudojimo atvejų, įrangos bei komponentų diagramos, aprašyti duomenų kubų, *Excel 2007* sukinių bei *ProClarity* grafikų kūrimo algoritmai veiklos ir sekų diagramos.
3. Palikiminės KTU duomenų bazės pagrindu realizuota duomenų saugykla bei, remiantis jos duomenimis, sukurta dešimt įvairių rūšių kubų, pradedant paprastais, žvaigždės tipo ir baigiant sudėtingu – snaigės tipo, kurie leido atskleisti bei išsamiau analizuoti duomenų kubų kūrimo proceso ypatybes.
4. Sukurti įvairūs duomenų sukiniai ir indikatorinės diagramos, kurių pagalba įvairiais pjūviais atlikta universiteto duomenų intelektualinė analizė. Atlikta studentų pažymių vidurkių analizė leido išsiaiškinti kas lemia aukštą skirtumą tarp studentų pažymių vidurkių. Taip pat atlikta didelę įtaką studentų studijų įvertinimui turinti analizė, kuri atskleidžia daugiausiai neigiamų pažymių parašiusius universiteto dėstytojus.
5. Atlikta *ProClarity Analytics* indikatorinės diagramos, rodančios studentų bendrą vidurkį įvairiose katedrose pagal studijų kursą, analizė atskleidė, kad modulių persikirstymas tarp kursų ar semestrų būtų reikšmingas studijų lygiui.
6. Ši intelektualinė analizė buvo atlikta tik keliais pjūviais, kuriais buvo atskleistos duomenų kubų analizės galimybės. Tačiau, norint padaryti daugiau reikšmingų universiteto studijų kokybei išvadų, reikalinga atlikti išsamesnę duomenų analizę, kuri apimtų daugiau rodiklių ir parametrų.
7. Atlikta analizė taip pat atskleidžia tam tikrus sistemos tobulinimo pasiūlymus. Pavyzdžiui į duomenų saugyklą įtraukti tokius duomenis kaip dėstytojų anoniminių įvertinimų, kuris gaunamas apklausiant universiteto studentus, rezultatus. Tokiu atveju būtų galima lyginti studentų nuomonę apie dėstytojus su dėstytojų rašomais pažymiais.

8. Tobulinant sistemą, būtų galima pasiūlyti studento individualaus plano lentelėje, kurioje saugomi studento mokomieji moduliai, rašyti ne tik atsiskaitymo pažymį, tačiau taip pat metus ir semestrą, kada tas pažymys parašytas. Tai leistų analizuoti konkretaus modulio pažymių vidurkio kitimą laike. Taip pat studento duomenų lentelėje būtų galima įtraukti papildomą paskaitų lankomumo atributą. Tai leistų vertinti studento pažymių vidurkio priklausomybę nuo paskaitų lankomumo.

LITERATŪRA

1. „OLAP duomenų bazės“, [žiūrėta 2007-10-08], prieiga per internetą:
http://itekspertas.projektas.lt/index.php?option=com_content&task=view&id=94&Itemid=53
2. „Introduction to Cubes“, [žiūrėta 2007-10-20], prieiga per internetą:
<http://msdn2.microsoft.com/en-us/library/ms174587.aspx>
3. Vigintas Šakys, „Duomenų saugyklos ir verslo duomenų analizė“ modulio medžiaga, [žiūrėta 2007-10-20], prieiga per internetą: <ftp://isd.ktu.lt/isd/Sakys/DSiVDA>
4. „SQL Server 2005 Editions“, [žiūrėta 2008-02-10], prieiga per internetą:
<http://www.microsoft.com/sql/editions/default.aspx>
5. „SQL Server 2005 funkcijų palyginimas“, [žiūrėta 2008-02-10], prieiga per internetą:
http://www.microsoft.com/lietuva/sql/prodinfo/features_comparison.aspx
6. „MySQL Database Software Products“, [žiūrėta 2008-02-03], prieiga per internetą:
<http://www.mysql.com/products/>
7. „MySQL“, [žiūrėta 2008-02-03], prieiga per internetą:
<http://en.wikipedia.org/wiki/MySQL>
8. „DB2 Product Family“, [žiūrėta 2008-02-03], prieiga per internetą: <http://www-306.ibm.com/software/data/db2/>
9. „IBM DB2“, [žiūrėta 2008-02-03], prieiga per internetą:
http://en.wikipedia.org/wiki/IBM_DB2
10. „Oracle Database“, [žiūrėta 2008-02-03], prieiga per internetą:
<http://www.oracle.com/database/index.html>
11. „Oracle Database“, [žiūrėta 2008-02-03], prieiga per internetą:
http://en.wikipedia.org/wiki/Oracle_Database
12. „MSDN Academic Alliance Software Center“, [žiūrėta 2007-11-11], prieiga per internetą: http://msdn62.e-academy.com/kaunu_comp
13. „Microsoft SQL Server“, [žiūrėta 2008-01-10], prieiga per internetą:
http://en.wikipedia.org/wiki/Microsoft_SQL_Server
14. „Microsoft SQL Server“, [žiūrėta 2008-01-10], prieiga per internetą:
http://en.wikipedia.org/wiki/Microsoft_SQL_Server
15. „SQL SERVER - Generic Architecture Image“, [žiūrėta 2008-01-10], prieiga per internetą: <http://blog.sqlauthority.com/2007/12/08/sql-server-generic-architecture-image>

16. „Microsoft SQL Server 2005 - SQL Server 2005 priedų ir funkcijų apžvalga“ [žiūrėta 2008-02-10], prieiga per internetą:
<http://www.microsoft.com/lietuva/sql/prodinfo/overview.mspx>
17. „ProClarity Analytics - Product data sheet“, [žiūrėta 2007-10-09], prieiga per internetą: <http://www.diraction.ch/shopdocs/files/ProClarityAnalytics.pdf>
18. „Welcome To dBASE“, [žiūrėta 2008-01-12], prieiga per internetą:
<http://www.dbase.com/>
19. „dBASE“, [žiūrėta 2008-01-12], prieiga per internetą:
<http://en.wikipedia.org/wiki/DBase>

PRIEDAI

TERMINŲ ŽODYNAS

- *UML* – (angl. *Unified Modelling Language*) unifikuota modeliavimo kalba;
- *DB* – (angl. *Database*) duomenų bazė;
- *OLE DB* – (angl. *Object Linking and Embedding, Database*) tai *Microsoft* kompanijos *COM* pagrindu sukurta sąsaja, leidžianti duomenų mainus tarp įvairių šaltinių;
- *TCP/IP* - (angl. *Transmission control protocol/Internet Protocol*) – standartinis duomenų perdavimo protokolų rinkinys, kurio pagrindu veikia internetas bei daugelis privačių komercinių tinklų;
- *API* - (angl. *Application programming interface*) aplikacijų programavimo sąsaja;
- *OLAP* - (angl. *Online Analytical Processing*) betarpiška duomenų analizė;
- *BI* – (angl. *Business Intelligence*) intelektualio verslo analizės sistema;
- *DS* – (angl. *Data Warehouse*) duomenų saugykla.

DUOMENŲ BAZĖS LENTELIŲ REIŠMIŲ PAAIŠKINIMAI

Atributo „BUSENA“ reikšmių lentelė

Kodas	Reikšmė
A	Atėjęs iš kito fakulteto
K	Kandidatas
L	Tyrimai Lietuvoje
M	Mainų programos studentas
N	Nėra dokumentų
P	Atlieka praktiką užsienyje
R	Kartos semestrą
S	„Socrates“ programos studentas
U	Tyrimai užsienyje
0	Išbrauktas
1	Studijuoja
2	Akademinėse atostogose
3	Kitoje aukštojoje mokykloje studijuoja
4	Stazuotė kitoje aukštojoje mokykloje
5	Laikiniai sustabdęs studijas
6	Perkeltas į kitą fakultetą
7	Atidėtas baigiamųjų darbų gynimas
8	Baigęs studijas
9	Dar nestudijuoja

Atributo „STATUS“ reikšmių lentelė

Kodas	Reikšmė
0	Profesinės (pirma pakopa)
1	Bakalauro
2	Specialiosios profesinės
3	Magistratūros
4	Doktorantūros
5	Tobulinimosi kursai

Atributo „PAZYMYS“ reikšmių lentelė

Kodas	Reikšmė
AT	Atidėta
BR	Išbraukta
IS	Įskaityta
NE	Neatvyko
NM	Nesumokėti mokesčiai
NS	Neįskaityta
0	Labai blogai
1	Labai blogai
2	Blogai
3	Nepatenkinamai
4	Nepatenkinamai
5	Silpnai
6	Patenkinamai
7	Vidutiniškai
8	Gerai
9	Labai gerai
10	Puikiai

