

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**Informacijos sistemų katedra**

Vaida Labutytė

**Duomenų koncepcinių schemų vaizdavimas**  
**saugyklų schemomis**

Magistro darbas

Darbo vadovas

prof. dr. L. Nemuraitė

**KAUNAS, 2009**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**Informacijos sistemų katedra**

Vaida Labutytė

**Duomenų koncepcinių schemų vaizdavimas**  
**saugyklų schemomis**

Magistro darbas

Recenzentas

doc. dr. S. Maciulevičius

2009-01-

Vadovas

prof. dr. L. Nemuraitė

2009-01-

Atliko

IFM-3/4 gr. stud.

Vaida Labutytė

2009-01-12

**KAUNAS, 2009**

# Turinys

<b>1. ĮVADAS .....</b>	<b>5</b>
<b>2. EGZISTUOJANČIŲ SAUGYKLŲ SCHEMŲ PROJEKTAVIMO PROBLEMŲ IR GALIMYBIŲ ANALIZĖ .....</b>	<b>9</b>
2.1. TYRIMO METODIKA .....	9
2.2. PERĖJIMO NUO RELIACINIŲ PRIE DAUGIAMAČIŲ DUOMENŲ STRUKTŪRŲ METODŲ ANALIZĖ.....	10
2.2.1. <i>Verslo analitikos (angl. Business Intelligence) istorija.....</i>	<i>10</i>
2.2.2. <i>Pagrindiniai terminai ir sąvokos .....</i>	<i>11</i>
2.3. VARTOTOJŲ ANALIZĖ .....	12
2.3.1. <i>Vartotojų aibė, tipai ir savybės .....</i>	<i>12</i>
2.3.2. <i>Vartotojų tikslai ir problemos .....</i>	<i>12</i>
2.4. SAUGYKLOS KŪRIMO METODŲ ANALIZĖ.....	13
2.4.1. <i>Daugiamatės saugyklos reikalingumo vertinimas.....</i>	<i>13</i>
2.4.2. <i>Daugiamatės schemos pasirinkimas .....</i>	<i>14</i>
2.4.3. <i>Saugyklos kūrimo proceso analizė .....</i>	<i>15</i>
2.4.4. <i>Reikalavimų analizė .....</i>	<i>16</i>
2.4.5. <i>Saugyklų koncepcinių modelių analizė .....</i>	<i>17</i>
2.4.6. <i>Sudėtingų saugyklos schemos sudarymo atvejų apibrėžimas.....</i>	<i>20</i>
2.4.7. <i>Loginis modelis .....</i>	<i>21</i>
2.5. VERSLO ANALITIKOS ĮRANKIŲ LYGINAMOJI ANALIZĖ .....	22
2.6. ANALIZĖS IŠVADOS .....	26
<b>3. SUDĖTINGŲ SAUGYKLŲ STRUKTŪRŲ PROJEKTAVIMO METODIKA .....</b>	<b>27</b>
3.1. REIKALAVIMŲ SURINKIMAS.....	28
3.2. DAUGIAMATĖS SCHEMOS SUDĖTINGŲ ATVEJŲ ANALIZĖ .....	30
3.3. DUOMENŲ SAUGYKLOS SUKŪRIMAS .....	34
3.4. KUBO PROJEKTAVIMAS.....	34
3.5. DUOMENŲ SURINKIMO, IŠVALYMO IR ĮKROVIMO PROCESO REALIZAVIMAS .....	36
3.6. ATASKAITŲ SUKŪRIMAS.....	37
<b>4. EKSPERIMENTINIS TYRIMAS, TAIKANT METODĄ REALIAI SISTEMAI.....</b>	<b>38</b>
4.1. REIKALAVIMŲ SURINKIMAS.....	38
4.2. DAUGIAMATĖS SCHEMOS SUDĖTINGŲ ATVEJŲ ANALIZĖ .....	45
4.3. DUOMENŲ SAUGYKLOS SUKŪRIMAS .....	48
4.4. KUBO PROJEKTAVIMAS.....	48
4.5. DUOMENŲ SURINKIMO, IŠVALYMO IR ĮKROVIMO PROCESO REALIZAVIMAS .....	54
4.6. ATASKAITŲ SUKŪRIMAS.....	58
4.7. TESTAVIMAS.....	60
<b>5. METODO ĮVERTINIMAS .....</b>	<b>65</b>
<b>6. IŠVADOS .....</b>	<b>67</b>
<b>7. LITERATŪRA .....</b>	<b>69</b>
<b>8. TERMINŲ IR SANTRUMPŲ ŽODYNAS .....</b>	<b>72</b>
<b>9. PRIEDAI.....</b>	<b>73</b>
1 PRIEDAS. VAIZDŲ SUKŪRIMO UŽKLAUSOS .....	73
2 PRIEDAS. STRAIPSNIS LEIDINYJE „INFORMACINĖS TECHNOLOGIJOS'08“ .....	75

# **„Representation of Data Conceptual Schemas to Warehouse Schemas“**

## **SUMMARY**

The developers of real life Business Intelligence Systems often face the difficulties in transforming the relational database to multidimensional data model. Making mistakes in early phases can lead to incorrect aggregation results.

The number of various conceptual models and methods are presented in the field of building the OLAP system. But they involve only the subset of problems and often do not describe all parts of the whole process.

The main goal of this work is to establish a method which includes all steps (from gathering the customer requirements to developing the intelligent reports) of developing a Data Warehouse System and supports the solutions for difficult situations in creating a multidimensional model. The method was successfully adapted to the real Business Intelligence project using Microsoft tools (SQL Server 2005 Business Intelligence Development Studio, Integration Services, Analysis Services and Excel 2007) and could fit for any other enterprises which need to develop complex warehouse schemas.

Key words: multidimensional data model, OLAP, warehouse



# 1 Įvadas

Šiandien verslo analizės paslaugų paklausa auga labai sparčiai, kadangi įmonių verslo sėkmė dinamiškoje rinkoje labai priklauso nuo laiku gaunamos, patikimos ir lengvai suprantamos verslo informacijos, kuri gali būti panaudojama tiek įmonės strategijai planuoti, tiek kasdieniams uždaviniams spręsti, tiek veiklos rezultatams įvertinti. Verslo sąvoką galime suprasti plačiąją prasme, nes veiklos analizė svarbi bet kokio tipo organizacijai.

Patikima informacija vadovui ar atsakingam darbuotojui turi būti pateikta tinkamu metu, tinkamu formatu ir tinkama forma – tik tuomet ši informacija suteiks organizacijai galimybę siekti aukščiausių rezultatų. Būtent šiam tikslui ir naudojami saugyklų kūrimo, analizės įrankiai. Tačiau dar prieš naudojant šiuos įrankius, susiduriama su daug laiko kainuojančiu ir daugiausia problemų sukeliančiu saugyklos projektavimu, ypač nepatyrusiam projektuotojui. Todėl labai svarbu surasti būdą, kaip šį procesą palengvinti ir pagreitinti, o tuo pačiu užtikrinti saugyklos teisingumą.

Tam tikslui tyrimo sritimi buvo pasirinkta duomenų saugyklų projektavimas, o kaip objektas pasirinktas pats perėjimo iš reliacinės duomenų bazės į daugiamatę procesas. Darbo tikslas – nustatyti, kas yra netaisyklingos, sudėtingos duomenų saugyklų struktūros ir kaip jas realizuoti pereinant iš reliacinių duomenų bazių prie daugiamačių modelių. Darbe iškelti šie uždaviniai:

- ✓ apibrėžti sudėtingų duomenų saugyklų struktūrų sąvokas;
- ✓ parinkti tinkamiausią paprastų koncepcinių duomenų struktūrų vaizdavimo saugyklos duomenų struktūromis būdą;
- ✓ aprašyti duomenų saugyklos projektavimo procesą, apimančią sudėtingų atvejų realizavimą:
  - daug-su-daug ryšių;
  - sudėtinių (besikertančių) dimensijų hierarchijų;
  - teisingą agregavimo ir kitų atvejų vaizdavimą;
- ✓ realizuoti sistemą ar didesnę pavyzdį, kuris leistų patikrinti metodo tinkamumą;
- ✓ patikrinti, ar sukurti sudėtingų atvejų sprendimai duoda teisingus rezultatus.

Išanalizuoti šie literatūroje aprašyti metodai:

D.L. Moody ir M.A.R. Kortnik [1] rekomenduoja pirmiausia suskirstyti visus ER modelio objektus į transakcijų, komponentų ir klasifikacijos. Vėliau išskirti maksimalias hierarchijas, operatorius. Taip pat aptariamos įvairios schemų rūšys pagal objektų suskirstymą (plokščioji, snaigės, žvaigždės, žvaigždynų ir kt.). Tačiau užsimenama tik apie keletą sudėtingų atvejų.

J.N. Mazon, J. Trujillo, J. Lechtenborger [6] sukurtas metodas paremtas norminėmis formomis. Ypač akcentuojama vartotojų reikalavimų analizės svarba, aprašomi SD ir SR modeliai šiems reikalavimams vaizduoti. Taip pat pateikti specifiniai daugiamatžio modelio stereotipai, modelio teisingumas užtikrinamas tikrinant daugiamates normines formas (1NF, 2NF ir 3NF). Metodas gana sudėtingas ir kol kas nėra užbaigtas.

M. Golfarelli, S. Rizzi [10], [11] sukurtas modelis paremtas medžio struktūra. Medžio pagrindas – tai fakto lentelė su atributais, apsupta dimensijų ir jų atributų šakų. Hierarchijos vaizduojamos apibraukiant dimensijų atributus apskritimu. Metodas aprašo matavimų tipus (adityvius, pusiau adityvius, neadityvius), suderinamų faktų schemų persidengimą, pereinamosios schemos nuo ER prie DF modelio projektavimą. Taip pat pateikiamas ligoninės sistemos pavyzdys pritaikius šį metodą. Deja, metodas apima labai nedidelę aibę sudėtingų atvejų.

S. Lujin-Mora, P. Vassiliadis, J. Trujillo [12] sukurtas metodas paremtas UML įtraukiant papildomus stereotipus daugiamatei schemai vaizduoti. Pirmiausia pradedama nuo šaltinio sistemų koncepcinių schemų, aprašomi duomenų susiejimo lygiai (duomenų bazės ir duomenų srautų, lentelės, atributo). Transformacijos procesas suskaidomas į tris dalis: padalinimo, filtravimo ir agregavimo. Duomenų saugyklos kontekste naudojant UML atributai vaizduojami kaip FCME (angl. *First-class modeling elements*). Šis metodas daugiausia koncentruojasi ties ryšiais, jų transformacija ir susiejimu, bet visai neskiria dėmesio sudėtingiems daugiamatžio modelio atvejams.

N. Tryfona, F. Busborg, J. G. B. Christiansen [22] savo sukurtą modelį pavadino StarER. Šiame modelyje apjungiamas reliacinės duomenų bazės ER modelis ir daugiamatis žvaigždės modelis. Metodui iškeliami šie uždaviniai: vaizduoti faktus ir jų savybes, sujungti faktus su laiko dimensija, vaizduoti objektus (dimensijas), jų savybes ir ryšius, užfiksuoti ryšius tarp objektų ir faktų, atskirti dimensijas ir išskirti hierarchijas. Taip pat šiame modelyje nagrinėjami specializacijos, apibendrinimo, agregavimo ryšiai. Šis metodas nenagrinėja visų hierarchijos atvejų, pats modelis pernelyg perkrautas.

Dėl to buvo sudarytas metodas, kuris yra paprastas ir aiškus, apima visą veiklos analitikos projekto realizavimo procesą, kuriame išanalizuota dažnai sutinkamų sudėtingų atvejų aibė. Metodas susideda iš šių žingsnių:

- 1 **žingsnis.** Pirminių reikalavimų surinkimas
- 2 **žingsnis.** Reliacinės duomenų bazės išskyrimas analizės sričiai
- 3 **žingsnis.** Schemos skaidymas į duomenų centrus (angl. *Data Mart*)
- 4 **žingsnis.** Hierarchijų ir matavimų išskyrimas
- 5 **žingsnis.** Pirminis dimensijų ir faktų išskyrimas

- 6 **žingsnis.** Sudėtingų atvejų išskyrimas formuojant žvaigždės/snaigės schemą
- 7 **žingsnis.** Daugiamatės duomenų bazės sukūrimo skripto generavimas
- 8 **žingsnis.** Reikalingų vaizdų (angl. *views*) sukūrimas
- 9 **žingsnis.** Duomenų saugyklos lentelių užpildymas testiniais duomenimis
- 10 **žingsnis.** Pirminis kubo generavimas
- 11 **žingsnis.** Schemos teisingumo patikrinimas tiek kubo kūrimo aplinkoje, tiek kliento įrankio aplinkoje, stengiantis prisiderinti prie įrankio ypatumų
- 12 **žingsnis.** Galutinis kubo sutvarkymas
- 13 **žingsnis.** ETL proceso realizavimas
- 14 **žingsnis.** Ataskaitų kūrimas

Šio metodo pranašumas prieš kitus analizuotus metodus yra tas, kad jis gali būti pritaikytas tiek nesudėtingiems nedidelės apimties veiklos analitikos projektams, tiek visus svarbiausius sudėtingus atvejus (daug-su-daug ryšiai, įvairūs hierarchijų atvejai, lėtai kintančios dimensijos, specifiniai matavimai, fakto dimensija ir t.t.) apimančiam projektui. Metodą nesudėtinga įsisavinti ir pritaikyti realizuojant veiklos analizės sistemą pasirinkta duomenų saugyklų kūrimo technologija.

Remiantis sudarytu metodu, realizuota veiklos analitikos sistema, kurioje analizuojami pardavimų (užsakymų, sąskaitų faktūrų), mokymų išlaidų, išdirbtų/suplanuotų darbo valandų, priskyrimo veikloms, projektų įvertinimo duomenys pagal pardavimų tipus, išlaidų tipus, projektus, užduotis, darbuotojus, veiklas ir data. Pirmiausia surinkti ir struktūrizuoti vartotojų reikalavimai. Šios sistemos duomenys imami iš verslo valdymo sistemos Navision ir įmonės projektų registro duomenų bazių. Duomenys integruojami, transformuojami ir užkraunami į duomenų saugyklą. ETL procesas realizuotas naudojant Microsoft SQL Server Integration Services įrankį. Sukurti dimensijų ir faktų duomenų surinkimo, apdorojimo bei užkrovimo *.dtsx* paketai instaliuojami į Integration Services serverį, iš kurio paleidžiamas pagrindinis SQL Server Integration Services projekto paketas. Tokie duomenų įkrovimai vyksta automatiškai kiekvieno mėnesio pradžioje. Nurodžius duomenų saugyklą kaip duomenų šaltinį, suprojektuoti kubai. Projektuojant kubus išskirti sudėtingi atvejai, kurie realizuoti pasirenkant geriausią variantą. Viso suformuoti trys kubai: „Pinigai\_Valandos“, „Užduotys\_Valandos“, „Išlaidos\_Mokymams“. Apibrėžus hierarchijas, matavimus, ryšius, kubai apdorojami ir instaliuojami į Analysis Services serverį. Excel programoje sukuriama duomenų šaltinio *.odc* failai, kiekvienam kubui po vieną. Šiame faile nurodomi prisijungimo duomenys (angl. *Connection String*) prie kubo. Sėkmingai prisijungus prie kubo, kuriamos sukinio lentelės ir įvairiais pjūviais peržiūrimi duomenys.

Metodo taikymas realiai sistemai parodė, kad jis leidžia išspręsti perėjimo iš duomenų bazės ER (angl. *Entity Relationship*) diagramos prie daugiamačio modelio problemą. Taip pat leido supaprastinti modelį panaudojant šiukšlių dimensiją, užtikrino teisingus skaičiavimo rezultatus realizuojant daug-su-daug ryšį, nepilną hierarchiją. Buvo realizuotas optimalus skirtingų reikšmių skaičiavimo algoritmas, galimybė sekti atributų pasikeitimo istoriją panaudojant lėtai besikeičiančias dimensijas, kitos svarbios informacijos saugojimas fakto dimensijose pažymint ją vėliavėlėmis.

Darbo struktūra:

- ✓ antrame skyriuje atlikta egzistuojančių saugyklų schemų projektavimo metodų ir juos taikant kylančių problemų analizė, apibrėžti pagrindiniai daugiamačių modelių terminai, trumpai aprašyta verslo analitikos istorija, tarpusavyje palygintos dvi saugyklų kūrimo technologijos – Oracle ir MS SQL Server;
- ✓ trečiame skyriuje pateikta sudėtingų saugyklų projektavimo metodika, kuri apima visą duomenų saugyklos realizavimo procesą, pradedant vartotojų reikalavimais ir baigiant analitinių ataskaitų kūrimu;
- ✓ ketvirtame skyriuje metodas pritaikytas realiai sistemai naudojant MS SQL Server 2005 įrankius, aprašyti sprendimo testavimo žingsniai, metodas įvertintas lyginant jį su kitais mokslinėje literatūroje aprašytais metodais;
- ✓ penktame skyriuje apibendrinti darbo rezultatai.

Darbo tema atspausdintas straipsnis ir perskaitytas pranešimas konferencijoje „Informacinės technologijos '08“.

## 2 Egzistuojančių saugyklų schemų projektavimo problemų ir galimybių analizė

### 2.1 Tyrimo metodika

Projektuojant duomenų saugyklas susiduriama su daugybe neaiškumų. Nėra visų pripažinto ir naudojamo metodo, kuris padėtų išvengti su projektavimu susijusių klaidų. Jei projektavimo metu bus daromos klaidos, duomenų saugykla bus neefektyvi ir duomenys gali būti klaidingi. Reikia išanalizuoti problemas, su kuriomis susiduriama vaizduojant duomenų koncepcines schemas į saugyklų schemas ir sukurti geriausią sprendimą, kuris leistų kurti efektyvias duomenų saugyklas.

Šio tyrimo sritis - verslo duomenų saugyklos projektavimas. Tyrimo objektas - perėjimo iš reliacinės duomenų bazės į daugiamatę procesas.

Tyrimo tikslas – apibrėžti, kas yra netaisyklingos, sudėtingos duomenų saugyklų struktūros ir kaip jas realizuoti pereinant nuo reliacinių duomenų bazių prie daugiamatė modelių. Tyrimo metu bus sprendžiami šie uždaviniai:

1. apibrėžti sudėtingų duomenų saugyklų struktūrų sąvokas;
2. parinkti tinkamiausią paprastų koncepcinių duomenų struktūrų vaizdavimo saugyklos duomenų struktūromis būdą;
3. aprašyti duomenų saugyklos projektavimo procesą, apimančią sudėtingų atvejų realizavimą:
  - ✓ daug-su-daug ryšių;
  - ✓ sudėtinių (besikertančių) dimensijų hierarchijų;
  - ✓ teisingą agregavimo ir kitų atvejų vaizdavimą;
4. realizuoti sistemą ar didesnę pavyzdį, kuris leistų patikrinti metodo tinkamumą;
5. patikrinti, ar sukurti sudėtingų atvejų sprendimai duoda teisingus rezultatus.

Šiame tyrime buvo taikoma konstruktyviojo tyrimo arba projektavimo mokslo tyrimo metodologija. Pagal šią metodologiją turi būti atliekama visapusiška problemos analizė, sukuriama nauji artefaktai, atliekamas eksperimentinis tyrimas ir sukuriama naujos žinios. Tyrimo planas:

1. Analizės dalis
  - ✓ Atlikti literatūros šaltinių analizę pagal savo darbo tikslus
  - ✓ Atlikti panašių metodų analizę – t.y. panagrinėti kitų autorių sukurtus metodus, modelius
  - ✓ Atlikti galimų technologijų analizę (MS SQL ir Oracle)

## 2. Teorinė dalis

- ✓ Aprašyti sprendimus sudėtingiems ryšių atvaizdavimo į saugyklų schemas atvejams
- ✓ Aprašyti bendrą saugyklos schemas sudarymo procesą, apimančią ištirtus atvejus

## 3. Eksperimentinė dalis

- ✓ Išbandyti saugyklos schemas sudarymo procesą taikant jį didesniame pavyzdžiui
- ✓ Ištestuoti metodą – patikrinti, ar sprendimai yra teisingi

## 4. Įvertinimas

- ✓ Apibendrinti tyrimo rezultatus: įvertinti pritaikytų ir sukurtų sprendimų pranašumus ir trūkumus, jų naudą verslui ir analitikams.

### 2.2 Perėjimo nuo reliacinių prie daugiamačių duomenų struktūrų metodų analizė

Perėjimo iš reliacinės duomenų bazės į daugiamatę procesas susideda iš dviejų dalių: schemas projektavimo ir duomenų perkėlimo. Šiame tyrime pagrindinis dėmesys skiriamas sudėtingiems schemas projektavimo atvejams: įvairių hierarchijų tipams, daug-su-daug ryšių atvaizdavimui, netaisyklingoms dimensijoms, duomenų agregavimo ribojimams. Nagrinėjami daugiamačių schemų kokybės (normalizavimo) klausimai.

Daugiamatė normalizuota schema – tai tokia schema, kurioje užtikrintas teisingas agregavimas įvairiuose hierarchijų lygmenyse. Svarbu nustatyti teisingus ir neperteklinius hierarchijų lygius, išskirti visų dimensijų hierarchijų tipus ir pasirinkti teisingą jų realizavimo būdą. Taip pat svarbu užtikrinti, kad faktuose ir dimensijose nebūtų NULL reikšmių, būtų apibrėžti matavimai pagal nustatytas dimensijas norint išvengti nekorektiškų užklausų rezultatų.

#### 2.2.1 Verslo analitikos (angl. *Business Intelligence*) istorija

Pirmieji žingsniai verslo analitikos link žengti 1962 metais, kai buvo aprašyta pirmoji daugiamatė kalba ir tik visai neseniai verslo analitikos sprendimai pradėti diegti Lietuvos įmonėse. Svarbiausi verslo analitikos istorijos įvykiai pateikti 1 lentelėje.

#### 1 lentelė

[13], [14]

#### Verslo analitikos istorija

Metai	Įvykis	Paaiškinimas
1962	Pirmoji daugiamatė kalba	Ken Iverson savo publikacijoje „A Programming Language“ aprašo APL daugiamatę kalbą, operatoriams naudojančią graikiškus simbolius.
1989	Pirmą kartą pavartojamas terminas „Business	Howard Dresner laikomas „Business Intelligence“ termino tėvu. BI apibrėžia rinkinį konceptų ir metodų,

Metai	Įvykis	Paaiškinimas
	Intelligence“	leidžiančių pagerinti verslo sprendimų priėmimą naudojant faktais paremtas sistemas.
1993	Codd apibrėžia OLAP	E. F. Codd daugiamatei analizei sugalvoja OLAP terminą (On Line Analytical Processing). Taip pat Codd sukūrė 12 OLAP taisyklių.
1995	Verslo analitika perkeliama į internetą	Sprendimus palaikanti sistema tampa prieinama per internetą.
1996	Verslo analitika jungiama su OLAP	BusinessObjects 4.0 išleidžia pirmąjį įrankį leidžiantį kurti daugiamates ir paprastas ataskaitas iš kubų, dinamiškai sukurtų iš reliacinės duomenų bazės.
1997	Pirmasis interaktyvus internetinis įrankis, skirtas ataskaitoms formuoti	Business Objects išleidžia Web Intelligence, su kuriuo ataskaitas galima kurti naršyklėje.
1998	Išleidžiamas IBM DB2 OLAP Serveris	Iš visų duomenų formuojama reliacinė žvaigždės schema.
1999	Pristatomi Microsoft OLAP Servisai	Pasižymėjo patogumu, aukšto lygio saugyklos architektūra (ROLAP/MOLAP/Hybrid).
2000	Microsoft OLAP Services pervadina į Analysis Services	Pridedami Data Mining servisai, praplečiamas dimensijų apdorojimas, saugumas, daug dėmesio skirta skaičiavimų pagerinimui.

### 2.2.2 Pagrindiniai terminai ir sąvokos

Prieš pradėdant gilesnę duomenų saugyklų sudarymo analizę, reikia apibrėžti daugiamačiam projektavimui būdingas sąvokas.

R.Kimball pateikia tokį apibrėžimą: duomenų saugykla – tai OLTP duomenų kopija su tam tikra, patogia užklausoms ir analizei, struktūra. Taip pat svarbu pabrėžti, kad tai centrinė duomenų saugykla, į kurią surenkami ir integruojami duomenys iš skirtingų OLTP duomenų bazių [19]. Bill Inmon dar 1990 metais išskyrė šias pagrindines duomenų saugyklos savybes:

- ✓ orientuota į objektą (tiriamą sritį) – visi tiesiogiai susiję duomenys su nagrinėjama sritimi (pvz., pardavimais) yra surenkami ir patalpinami vienoje duomenų bazėje;
- ✓ integruota – visi duomenys, surinkti iš skirtingų duomenų šaltinių, turi būti suderinami;
- ✓ priklausoma nuo laiko – visi duomenys tiesiogiai ar netiesiogiai yra priklausomi nuo laiko ašies;
- ✓ nekintanti – duomenys, patekę į duomenų saugyklą nekinta. Tai leidžia atlikti dabarties ir praeities rezultatų palyginimo analizę.

Codd savo knygoje pateikia tokį kubo apibrėžimą: kubas – tai duomenų struktūra, leidžianti greitai analizuoti duomenis [2], nes kubas agreguoja faktus kiekviename dimensijos lygmenyje.

Faktas – tai lentelė, kurioje saugomi matavimai (faktai), t. y. skaičiuojamosios reikšmės ir išoriniai raktai, turintys ryšį su dimensijų pirminiais raktais.

Dimensija – tai tam tikros kategorijos lentelė, pagal kurią peržiūrimi susumuoti duomenys. Dimensijoms būdingos hierarchijos.

Hierarchija – tai tam tikra loginė struktūra, kurioje duomenys sudėlioti pagal lygius.

## **2.3 Vartotojų analizė**

### **2.3.1 Vartotojų aibė, tipai ir savybės**

Būtų galima išskirti šias vartotojų grupes:

- ✓ OLAP sistemos vartotojai – jie pateikia reikalavimus būsimai sistemai ir naudoja galutinai realizuotą sistemą;
- ✓ Reliacinės duomenų bazės administratorius – pateikia duomenis ir duomenų šaltinius saugyklos projektuotojui;
- ✓ Saugyklos projektuotojas –
  - išanalizuoja vartotojo reikalavimus;
  - sudaro pradinį koncepcinį daugiamatį modelį naudodamas UML ar kitoki diagramų braižymo įrankį;
  - sudaro duomenų modelį;
  - patikrina, ar reikalavimais paremtas koncepcinis modelis teisingas su turimais duomenų šaltiniais, kad užtikrintų, jog duomenų saugykla bus teisingai užpildyta duomenimis;
  - transformuoja koncepcinį daugiamatį modelį į loginį.

### **2.3.2 Vartotojų tikslai ir problemos**

Pirmoji vartotojų grupė (OLAP sistemos vartotojai) susiduria su sunkumais apibrėždami reikalavimus taip, kad jie būtų teisingai suprasti analitiko. Taip pat neretai analizuojant duomenis pagal dimensijas gaunami ne tie rezultatai, kurių buvo tikėtasi. Sistemos vartotojų tikslas – gauti sistemą, kuri atitiktų jų reikalavimus.

Antroji vartotojų grupė (reliacinių DBVS administratoriai) turi užtikrinti duomenų saugumą, teisingumą ir jų pateikimą laiku. Tačiau nusprendus reliacinę DB perorientuoti į daugiamatę, reikalinga papildoma administratoriaus kvalifikacija.

Tyrimo srities svarbiausia vartotojų grupė – trečioji (saugyklų projektuotojai). Projektuotojų tikslas – pateikti teisingai veikiančią, greitai užklaudas atliekančią, saugią sistemą. Kiekviename sistemos kūrimo etape ši vartotojų grupė susiduria su atitinkamomis problemomis, kurios siejasi su tyrimo uždaviniais.



## 2.4 Saugyklos kūrimo metodų analizė

### 2.4.1 Daugiamatės saugyklos reikalingumo vertinimas

Šiame etape reikalinga įvertinti, kokiais atvejais naudinga pereiti nuo reliacinių duomenų bazių prie daugiamačių duomenų saugyklų.

Jei sistema atlieka kasdienines verslo funkcijas, tokias kaip užsakymų įtraukimas į sąrašą, apskaita ir prekių atsargų kontrolė ir reikia atsakyti į tokius klausimus, kaip:

- ✓ Kodėl gegužę pardavimai buvo didesni nei birželį?
- ✓ Kurį produktą pardavimų skyrius parduoda labiausiai?
- ✓ Koks darbuotojų skaičius pagal skyrių?
- ✓ Kaip kompanijos pelningumas pasiskirsto pagal parduotuves?

šiuos tikslus padės pasiekti OLAP. OLAP ir reliacinės duomenų bazės palyginimas pateikiamas 2 lentelėje:

2 lentelė

OLTP ir OLAP duomenų bazės savybės

OLTP	OLAP
Pastoviai atnaujinami duomenys realiu laiku.	Istoriniai duomenys, kurie tik nuskaitomi porcijomis.
Naudojama normalizuota (3NF) duomenų bazė.	Duomenys denormalizuojami (reliaciniu požiūriu).
Sudėtingas duomenų modelis.	Duomenų modelis nesudėtingas.
Pabrėžiamas įrašų atnaujinimo greitis.	Pabrėžiamas įrašų paieškos greitis.
Sudėtinga replikacija.	Nesudėtinga replikacija.
Duomenų bazės dydis: 100 MB-GB	Duomenų bazės dydis: 100 GB-TB
OLTP yra pagrindinis duomenų šaltinis.	Duomenų šaltinį gali sudaryti kelios OLTP DB.
Užklausų greitis optimizuojamas mažinant indeksų skaičių.	Užklausų greitis optimizuojamas didinant indeksų skaičių.

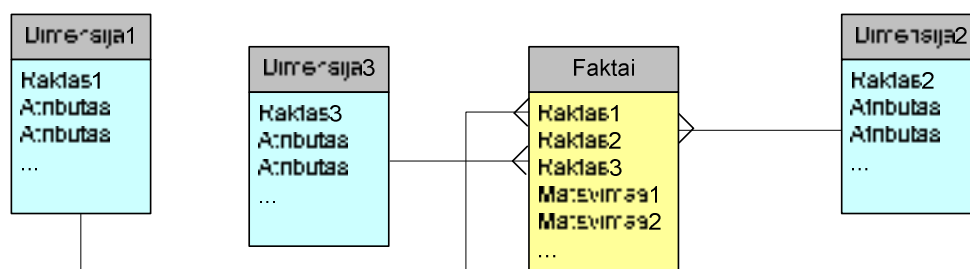
Tačiau tam, kad verslo analitikos (angl. *Business Intelligence*) sprendimai būtų efektyvūs, reikia, kad OLAP kubams naudojami duomenys būtų tvarkingi (angl. *clean*) ir integruoti. Šiam tikslui pasiekti reikalingi ETL (angl. *Extract Transform Load*) procesus palaikantys įrankiai, pvz. „MS SQL Server 2005“ įrankis „Integration Services“. Taigi dideliems duomenų srautams projektuojama duomenų saugykla.

Yra nemažai OLAP privalumų. Daugeliu atveju OLAP užklausos gali būti atliekamos daug greičiau negu reliacinės DB užklausos. Taip pat, OLAP suteikia užklausų lankstumą, kas leidžia peržiūrėti bet kurią dimensiją, lygį ar narį kaip eilutę ar stulpelį. Dėl greičio bei lankstumo kūrėjai ir renkasi OLAP.

### 2.4.2 Daugiamatės schemas pasirinkimas

Daugiamatės duomenų bazės modeliavimui naudojamos dviejų tipų schemas: žvaigždės arba snaigės. Taip pat gali būti modeliuojama sudėtingesnė žvaigždžių grupių schema, joje normalizavimo lygis mažesnis negu snaigės schemas atveju. Žvaigždės schema yra naudojama dažniausiai, nes jos privalumas – didelis užklausų greitis, kuris pasiekiamas visiškai denormalizuojant duomenų bazės lenteles. Snaigės schemas atveju gaunamas didesnis duomenų detalumas ir šiek tiek sutaupoma vietos atžvilgiu, tačiau sistema tampa labai sudėtinga dėl daugybės ryšių.

Žvaigždės schemas centre yra faktų lentelė. Tai yra daugiausiai naudojama dimensinių užklausų lentelė, joje saugomi realūs duomenys (faktai). Faktai yra skaitiniai atributai tokie kaip kiekis, suma, kurie gali būti sumuojami, skaičiuojamas jų vidurkis, maksimali, minimali reikšmės ir gali būti agreguojami įvairiomis statistinėmis operacijomis. Faktų atributus sudaro išmatuojamos skaitinės reikšmės apie dalyką, kurį aprašo faktų lentelė. Dimensijų atributai suteikia paaiškinamąją informaciją apie kiekvieną faktų lentelės įrašą. Šie atributai naudojami ryšiams tarp faktų lentelės ir susijusių dimensijų lentelės. Žvaigždės schemas pavyzdys pateiktas 1 paveiksle.



1 pav. Žvaigždės schemas pavyzdys

Schemas pasirinkimą lemia tik tai, kaip vienas ar kitas schemas pavidalas paveiks ETL procesą. Jei vartotojai naudoja informaciją iš OLAP, patogumas vartotojui ir užklausų greitis ateina iš OLAP sluoksnio, ne iš reliacinės duomenų bazės. Tai reiškia, kad nebūtina dramatiškai viską denormalizuoti ir jeigu reikia pasinaudoti snaigės schemas privalumais.

ETL požiūriu paprasčiausia realizuoti žvaigždės schema, ypač kuomet visa dimensijų informacija ateina iš to paties vieno šaltinio. Tokiu atveju daug paprasčiau atnaujinti pasikeitusią dimensijų lentelę naudojant pasikeitusių ar pridėtų eilučių fiksavimui skirtas ETL savybes.

Snaigės schema paprastai naudojama kuomet keletas dimensijų atributų yra iš skirtingų šaltinių. Tuomet atskyrus šiuos atributus į skirtingas lenteles paprasčiau atsekti pasikeitimus.

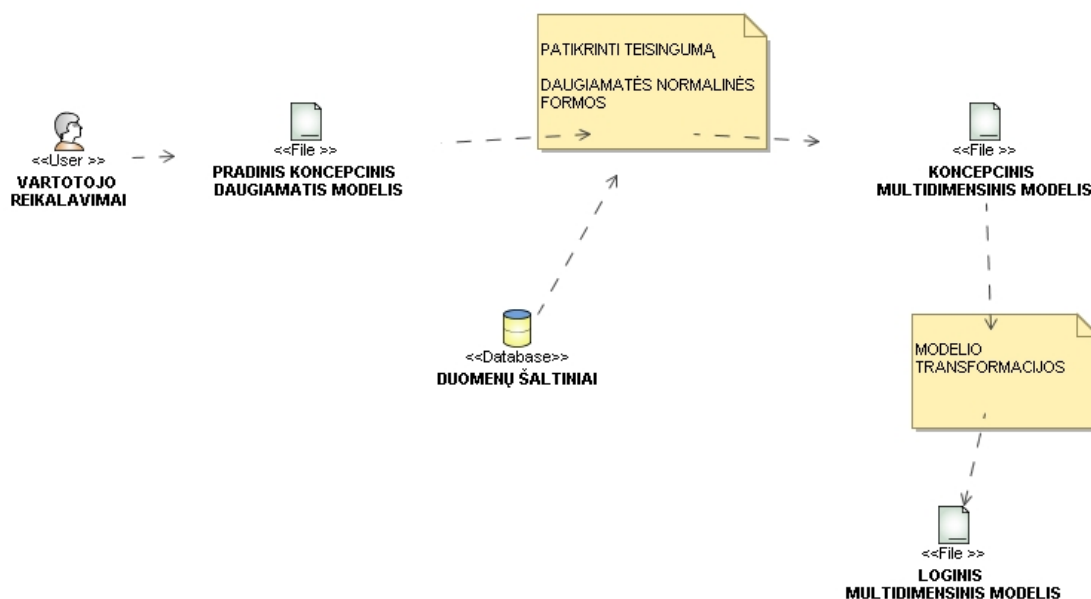
Projektuotojai kartais renkasi snaigės schema ir tada, kai dimensija turi stiprią natūralią hierarchiją (Pvz., Produkto dimensija, kuri skaidosi į Kategorijos, Subkategorijos ir Produkto).

Jei visi trys lygiai yra normalizuotoje lentelėje šaltinio sistemoje, tuomet gali būti lengviau valdyti ETL procesą turint dimensiją sudarytą ne iš vienos, o trijų lentelių.

### 2.4.3 Saugyklos kūrimo proceso analizė

Pasak D.L. Moody ir M.A.R. Kortnik [1], norint pereiti nuo reliacinio modelio prie daugiamačio, pirmiausia reikia duomenų modelyje esybės suskirstyti į transakcijų (pvz. pardavimai), komponentų (pvz. periodas, produktas, klientas) ir klasifikuojančias (pvz. kliento tipas, regionas, valstija). Antrasis žingsnis būtų identifikuoti hierarchijas (ryšius tarp dimensijų lygių). Trečiasis žingsnis – sudaryti dimensinius modelius panaudojant du operatorius: hierarchijos sutraukimas (denormalizacija) ir agregacija (naujos esybės sukūrimas sumuojamiems duomenims). Šie operatoriai naudojami skirtingai priklausomai nuo pasirinktos schemas. Ketvirtame žingsnyje atliekamos įvairios modifikacijos, aptariami tam tikri schemas projektavimo atvejai.

J.N. Mazo'n, J. Trujillo, J. Lechtenbo'rger [6] nuosekliai aprašo saugyklos projektavimo procesą (jos užpildymą duomenimis panaudojant daugiamačes normalines formas). Metodą galima būtų pavaizduoti tokia schema:



2 pav. Daugiamačės schemas sudarymo proceso apibendrintas modelis pagal J.N. Mazo'n, J. Trujillo, J. Lechtenbo'rger

Panašiai šį procesą apibrėžia M. Golfarelli, S. Rizzi [10]. Autoriai teigia, kad duomenų saugyklos sukūrimo procesą sudaro šeši žingsniai:

- ✓ Informacinės sistemos analizė (suprojektuojama duomenų bazės schema);
- ✓ Reikalavimų specifikavimas (gaunami faktai);
- ✓ Konceptinis modeliavimas (dimensinė schema);
- ✓ Dimensinės schemas teisingumo užtikrinimas;

- ✓ Loginis modeliavimas (loginė duomenų saugyklos schema);
- ✓ Fizinis modeliavimas (fizinė duomenų saugyklos schema).

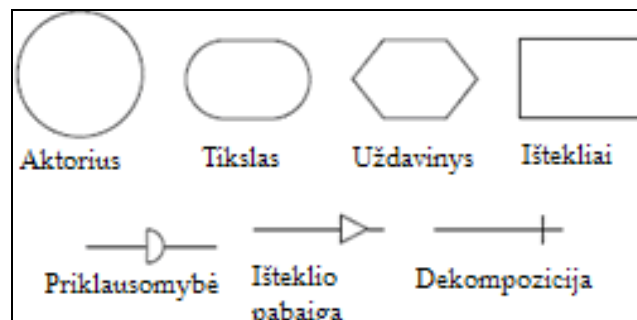
Šie procesai neapima konkrečių žingsnių ir jų detalizavimo, kaip pereiti nuo reliacinės duomenų bazės prie daugiamačio modelio. Taip pat autoriai nagrinėja tik paprastus daugiamatės schemas atvejus bei ryšius, arba nagrinėja tik konkrečią sudėtingų atvejų aibės sritį. Todėl šiame darbe bus sudarytas modelis, pilnai aprašantis visą verslo analitikos projekto kūrimo procesą ir apimantis didžiąją dalį sudėtingų atvejų, su kuriais dažniausiai susiduriama projektuojant daugiamatį koncepcinį modelį.

#### 2.4.4 Reikalavimų analizė

Paprastai duomenų saugyklos projektuojamos arba remiantis esamais duomenų šaltiniais ir kruopščia jų analize, arba vartotojo reikalavimais. Pirmuoju atveju supaprastinamas ETL proceso projektavimas, tačiau vartotojo reikalavimai nustumiami į antrą planą ir kūrėjas turi per mažai informacijos, kaip jam identifikuoti faktus, matavimus ir dimensijas. Antruoju atveju ETL proceso projektavimas ganėtinai sudėtingas, bet į pirmąjį planą iškeliami vartotojo reikalavimai [8]. Pasak J. Schiefer, B. List, R. M. Bruckner [9] reikalavimai duomenų saugyklai tiesiogiai įtakoja tiek techninį tiek duomenų saugyklos sistemos modeliavimo aspektą, bei projekto valdymą. Organizacinis modeliavimas susideda iš trijų etapų:

- ✓ Tikslų analizės,
- ✓ Faktų analizės,
- ✓ Atributų analizės [8].

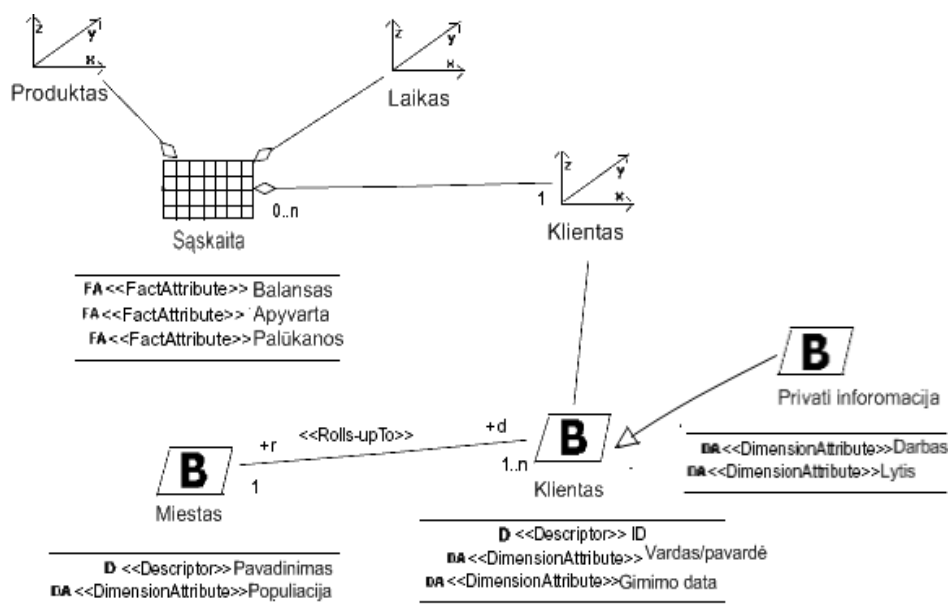
Tiksmai skirstomi į strateginius (aukščiausias abstrakcijos lygmuo), sprendimo (vidutinis abstrakcijos lygmuo) ir informacijos (žemiausias abstrakcijos lygmuo). Norint šiuos tikslus ir reikalavimus duomenims sumodeliuoti, naudojami SD (angl. *Strategic Dependency*) ir SR (angl. *Strategic Rationale*) modeliai. SD modelis parodo priklausomybės ryšius tarp aktorių organizacijos kontekste, o SR modelis skirtas apibrėžti aktorių interesus. Iš šių modelių gaunamas koncepcinis modelis [6]. Jiems nubraižyti naudojami šie pažymėjimai:



3 pav. Pažymėjimai SD ir SR modeliuose pagal J.N. Mazo'n, J. Trujillo, J. Lechtenbo'rger

## 2.4.5 Saugyklų koncepcinių modelių analizė

Minėjome, kad koncepcinis modelis gali būti gautas iš SD ir SR modelių. 4 paveiksle pavaizduotas modelis, kurį sudaro faktų lentelė (Sąskaita), sudaryta iš trijų matavimų (balansas, apyvarta, palūkanos). Ši faktų lentelė sujungta su dimensijų lentelėmis (klientas, produktas, laikas). Dėl aiškumo modeliuojama tik kliento dimensija.



4 pav. Pradinis koncepcinis modelis pagal J.N. Mazo'n, J. Trujillo, J. Lechtenbo'rger

Žymėjimai, naudojami modelyje, pateikti 3 lentelėje.

### 3 lentelė

Žymėjimai koncepciniame modelyje pagal J.N. Mazo'n, J. Trujillo, J. Lechtenbo'rger

Ikona	Stereotipas	Apibūdinimas
	Faktas	Šio stereotipo klasės koncepciniame modelyje vaizduoja faktus, susidedančius iš matavimų.
	Dimensija	Šio stereotipo klasės koncepciniame modelyje vaizduoja dimensijas, susidedančias iš hierarchijų.
	Bazė	Šio stereotipo klasės koncepciniame modelyje vaizduoja dimensijų hierarchijų lygius, susidedančius iš dimensijų atributų.
<b>FA</b>	Fakto atributas	Šio stereotipo atributai koncepciniame modelyje vaizduoja fakto atributus.
<b>DA</b>	Dimensijos atributas	Šio stereotipo atributai koncepciniame modelyje vaizduoja dimensijos hierarchijos lygio (bazės) atributus.
<b>D</b>	Deskriptorius	Šio stereotipo atributai koncepciniame modelyje vaizduoja dimensijos hierarchijos lygio (bazės) deskriptoriaus atributus.
<<Rolls-UpTo>>	Sutraukimas	Sutraukimas (vaizduoja ryšius tarp dviejų bazių). Vaidmuo R vaizduoja kryptį, kuria hierarchija sutraukiama (angl. <i>rolls up</i> ), vaidmuo D – kryptį, kuria hierarchija analizuojama gilyn (angl. <i>drills down</i> ).

Tačiau tai tik pradinis koncepcinis modelis, kurio pagerinimui naudojami QVT (angl. *Query/View/Transformation*) ryšiai, gaunami iš norminių formų. Metodo autoriai rekomenduoja šias priemones schemas teisingumui užtikrinti:

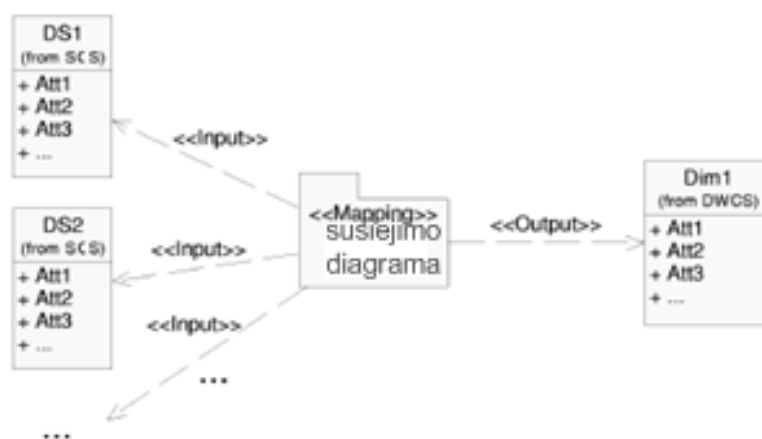
- ✓ turėti teisingą ir paruoštą duomenų modelį;
- ✓ sudaryti koncepcinį modelį UML klasių diagramos pavidalu;
- ✓ apibrėžti ryšius naudojant QVT kalbą ir panaudojant CASE įrankį verificuoti normines formas.

QVT ryšiai apibrėžti naudojami šie elementai:

- ✓ Vienas ar daugiau domenų,
- ✓ When sakinytis,
- ✓ Where sakinytis.

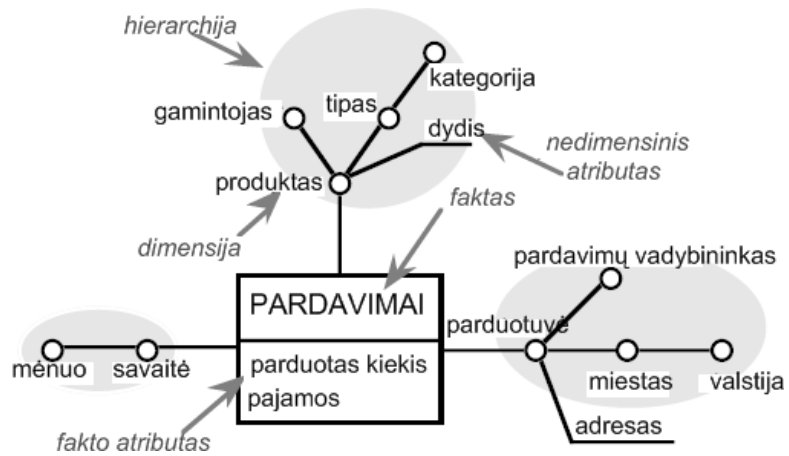
QVT ryšiams išreikšti yra naudojamos trys norminės formos. 1 NF reikalinga tam, kad užtikrinti, jog koncepcinė schema atitinka šaltinio duomenų bazių pateiktą informaciją. 2NF ir 3NF susijusi su NULL reikšmių išvengimu [6].

S. Lujin-Mora, P. Vassiliadis, J. Trujillo pataria projektuoti duomenų saugyklos koncepcinę schemą realizuojant ją UML (angl. *Unified Modeling Language*) paketais. Naudojant <<Input>>, <<Output>>, <<Mapping>> stereotipus modelyje galima pavaizduoti pradinio duomenų modelio ir daugiamatės schemas ryšius. [12]



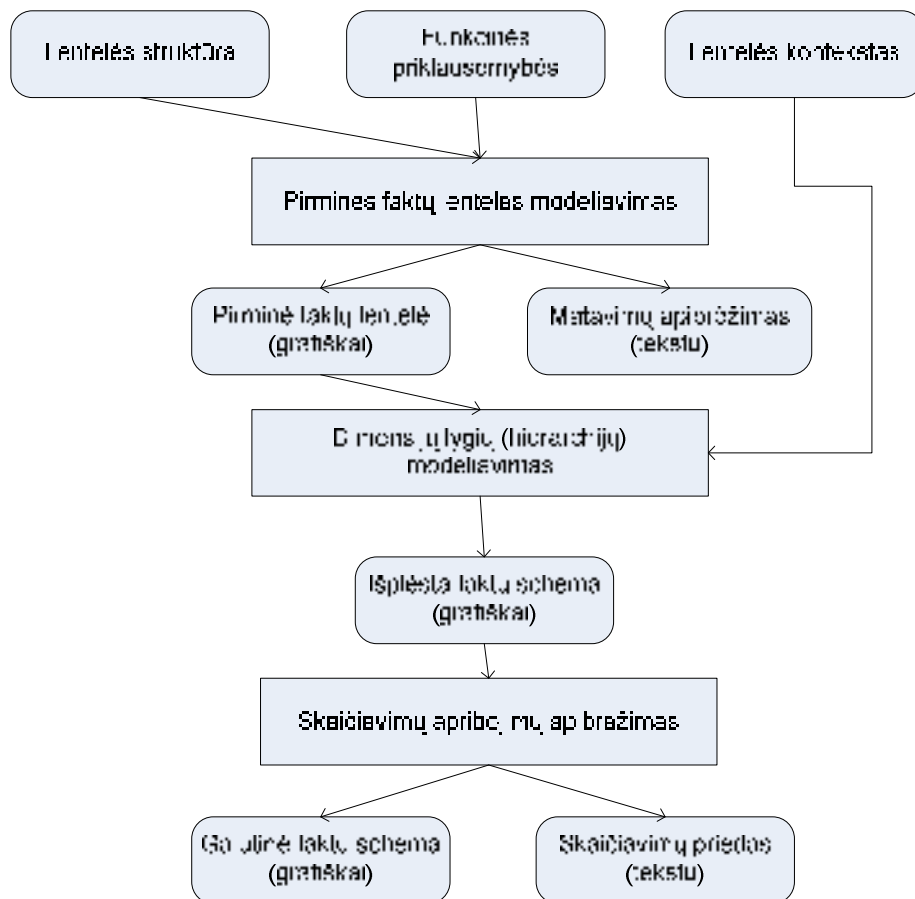
5 pav. Koncepcinis modelis pagal S. Lujin-Mora, P. Vassiliadis, J. Trujillo

M. Golfarelli, D. Maio, S. Rizzi [11] pataria koncepcinį modelį vaizduoti kaip medį:



6 pav. Konceptinis modelis pagal M. Golfarelli, D. Maio, S. Rizzi

J. Lechtenböcker aprašo koncepcinį modeliavimo procesą pateikdamas šio sudėtinio proceso schemą [15]:



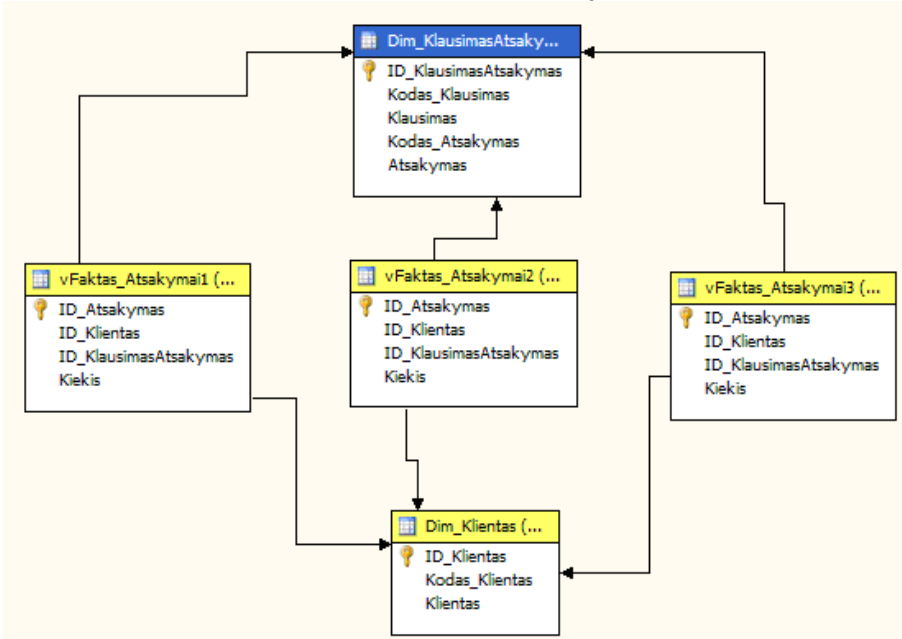
7 pav. Konceptinio modelio sudarymo proceso diagrama pagal J. Lechtenböcker

### 2.4.6 Sudėtingų saugyklos schemas sudarymo atvejų apibrėžimas

Neretai modeliuojant koncepcinę schemą susiduriama su sudėtingesniais atvejais, tokiais kaip: daug-su-daug ryšiai, sudėtinės dimensijos, įvairūs hierarchijų atvejai, apibendrinimo ir specializacijos ryšiai, agregavimo (sumavimo) ypatingi atvejai. 4 lentelėje trumpai apibūdinami autorių nagrinėjami atvejai, sprendimo būdai.

4 lentelė

#### Autorių nagrinėjamos problemos ir jų sprendimo būdai

Nagrinėjami atvejai	Autoriai
<p><b>Daug su daug ryšiai (angl. many-to-many relationships)</b></p> <p>Marco Ruso atliko ypač plačią many-to-many ryšių studiją, pažvelgdamas į duomenų vaizdavimą iš kitos perspektyvos, tačiau vis tikrai pripažindamas, kad toks ryšys prideda žvaigždės schemai sudėtingumo. Pirmiausia, jis aiškina paprastus (angl. <i>Classical</i>) daug-su-daug ryšius tarp dimensijų. Tokiam ryšiui vaizduoti įvedama be faktė faktų lentelė (angl. <i>Factless fact table</i>), kurios vaidmuo labai panašus į tarpinės lentelės vaidmenį reliacinėse duomenų bazėse. Vėliau aptariami sudėtingesni kaskadinio tipo daug-su-daug ryšiai (angl. <i>Cascading many-to-many relationships</i>), paprastai atsirandantys modeliuojant klausimynus, anketas su keletu pasirinkimų. Tai tokie atvejai, kuomet dimensija turi daug-su-daug ryšį iš abiejų pusių. Jeigu nagrinėjama matavimų grupė (angl. <i>Measure groups</i>) neturi reguliaraus tiesioginio ryšio su dimensija, tuomet ryšiui tarp tos dimensijos ir matavimų grupės apibrėžti naudojama tarpinė matavimų grupė.</p> <p>Taip pat aptariamas anketų sudarymo scenarijus, kuomet anketą sudaro klausimai su numatytais paprastais atsakymais ir pasirinkimais. Tipinė žvaigždės schema (viena faktų lentelė sujungta su klausimų/atsakymų dimensijomis ir apklaustųjų dimensija) tinkama, kuomet lyginame skirtingus atsakymus į tą patį klausimą. Tačiau ši schema tampa netinkama norint susieti atsakymus su daugiau nei vienu klausimu.</p> <p>Autorius aprašo tokį modelį, kuriame atsakymų faktų lentelė nenaudojama pačiame kube ir matoma tik duomenų šaltinio vaizduose. Ši lentelė naudojama formuojant atsakymų faktų vaizdus. Kliento lentelė atlieka tiek fakto tiek dimensijos vaidmenį.</p>  <p><b>8 pav. OLAP kubas anketos scenarijui</b></p> <p>Dar kiti autoriai rekomenduoja: panaikinti susikertančią esybę, pakeisti daug-su-daug ryšį į vienas-su-daug ryšiu apibrėžiant pirminį ryšį arba įtraukti šias esybes į saugyklą. Jos gali būti naudingos analitikams ekspertams, nors jų nebus galima analizuoti naudojant OLAP įrankius.</p>	<p>[1], [7], [21], [20]</p>



Nagrinėjami atvejai	Autoriai
<p><b><u>Ivairūs hierarchijų atvejai</u></b></p> <p>Sumavimui labai svarbu užtikrinti taisyklingas hierarchijų struktūras. Įvairiuose šaltiniuose netaisyklingos hierarchijos įvardinamos skirtingai.</p> <p><i>Daug su daug kardinalumas tarp hierarchijų dimensijose (angl. non-strict hierarchies).</i></p> <p>Tradiciškai OLAP palaiko tik vienas su daug ryšius tarp hierarchijų, kur kiekvienas žemesnio lygio narys priklauso tik vienam aukštesnio lygio nariui. Daug su daug ryšiai tarp hierarchijų sąlygoja dvigubas skaičiavimo reikšmes. Sukurtas modelis, kuris simetriškai palaiko dimensijas ir matavimus, leidžiant matavimus naudoti kaip dimensijas ir atvirkščiai. Kiti modeliai rekomenduoja naudoti tam tikras išorines sąlygas, skaičiavimams naudoti sudedamus matavimus ir pan.</p> <p><i>Nesubalansuotos (angl. unbalanced), nereguliaros (angl. irregular), asimetrinės (angl. asymmetric), rekursyvios (angl. recursive) hierarchijos</i> – tai tokios hierarchijos, kuriose hierarchijų keliai yra nevienodo ilgio. <i>Sudėtinės (angl. multiple)</i> – tai tokios hierarchijos, kuomet egzistuoja keli keliai į tą patį hierarchijos lygmenį. <i>Nepilnos (angl. ragged)</i> – tai tokios hierarchijos, kuomet tam tikri hierarchijos atributai neturi reikšmių. <i>Bendros (angl. shared), prisitaikančios (angl. conformed)</i> sutinkamos tuomet, kai keli kubai naudoja tą pačią dimensiją.</p> <p>Šioms problemoms spręsti patariama naudoti įvairias transformacijas, po kurių gaunamos simetrinės hierarchijos (simetrinė medžio struktūra). Patartina vengti netaisyklingų hierarchijų ir naudoti tik griežtas natūralias hierarchijas (angl. strict hierarchies).</p>	<p>[3], [4], [5], [7], [16], [17], [18]</p>
<p><b><u>Null reikšmių kontroliavimas</u></b></p> <p>Null reikšmės gali atsirasti tiek faktų tiek dimensijų lentelėse. Kad užtikrinti teisingus skaičiavimus, reikia dimensijų lentelėse sukurti atitinkamas „tuščias“eilutes su pirminio rakto reikšme pvz. -1, kurios atsirastų faktų lentelėje vietoj null reikšmių.</p> <p>Dimensijose atsirandančios null reikšmės sąlygoja negriežtų hierarchijų atsiradimą. Null reikšmės gali būti kontroliuojamos naudojant alternatyvius dimensijų lygius. Naudojant šį metodą išvengiama perteklinių reikšmių, užtikrinamas teisingas sumavimas.</p>	<p>[6], [18]</p>
<p><b><u>Distinct Count</u></b></p> <p>Skaičiuojant distinct count reikšmes tarpinė ir galutinė suma tai nėra sudėtos visos distinct reikšmės, kaip kad sumos atveju. Tokias situacijas galima spręsti užrašant skaičiavimus MDX kalba arba kuriant tam tikras matavimų grupes. Taip pat renkantis sprendimą labai svarbu atsižvelgti į laiko optimizavimą, nes SQL Server 2005 galima nustatyti distinct count matavimą, tačiau esant dideliems dimensijų duomenų kiekiams, jo apdorojimo procesas ilgai trunka.</p>	<p>[7]</p>

#### 2.4.7 Loginis modelis

Loginis modelis atspindi duomenų saugyklos loginius aspektus, tokius kaip ETL procesų apibrėžimą. Loginio modeliavimo metu koncepcinė schema transformuojama į atitinkamą loginę schemą pasirinktoje platformoje, atsižvelgiant į tam tikrus ribojimus (disko vietą, užklausų greitį ir pan.) [12]. Kadangi mūsų sukurtas metodas turi nepriklausyti nuo konkrečios platformos, šio modelio detaliau nenagrinėsime.

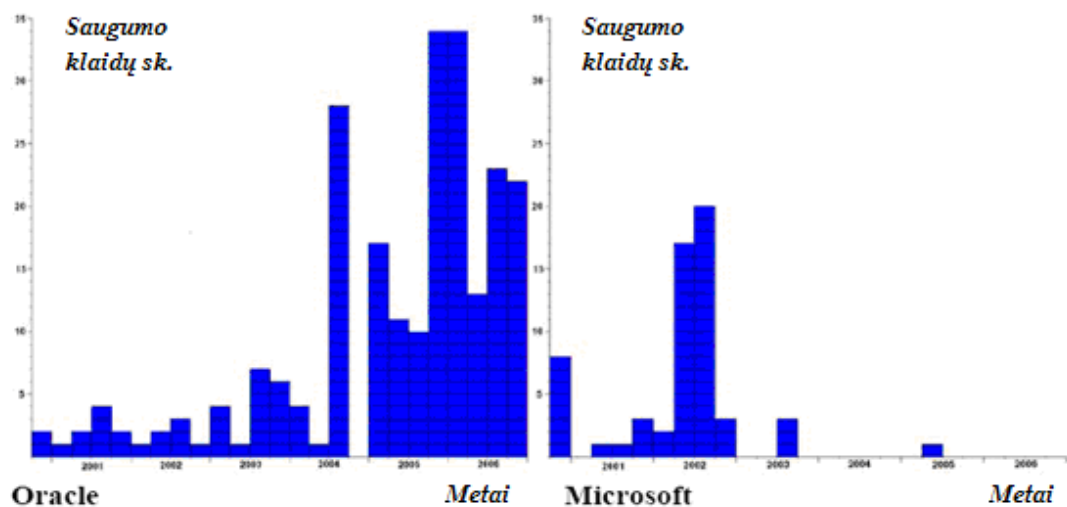
Esamų sprendimų yra labai daug, tačiau tik nedaugelis jų paliečia sudėtingesnius schemų projektavimo atvejus. Be to, nei vienas sprendimas neapima visų sudėtingų atvejų. Pavyzdžiui, [6] šaltinyje išsamiai nagrinėjami apibendrinimų atvaizdavimo atvejai, tačiau nenagrinėjamos alternatyvios dimensijos, daug-su-daug ryšių atvaizdavimas.

## 2.5 Verslo analitikos įrankių lyginamoji analizė

Kad būtų galima patikrinti gautų sprendimų teisingumą, reikia juos realizuoti. Tačiau reikia pabrėžti, kad sprendimai nėra priklausomi nuo tam tikros technologijos. Renkantis realizavimo technologiją, buvo analizuojamos dvi technologijos (serverio tipo duomenų bazės): „Microsoft SQL Server 2005“ ir „Oracle 10g RAC“ saugumo, eksploatacinės parengties, našumo, o labiausiai – tinkamumo verslo analitikai požiūriu.

### ▪ Saugumas

Oracle buvo ganėtinai saugus praeityje, bet nuo 2005 metų saugumo klaidų žymiai padaugėjo. 9 paveiksle pateiktuose grafikuose parodytos tik tos saugumo klaidos, kurios buvo aptiktos ir ištaisytos.



9 pav. Saugumo klaidų skaičius 2001 - 2006 metais

### ▪ Eksploatacinė parengtis (High Availability)

Tiek „SQL Server 2005“ tiek „Oracle 10g“ yra aukšto tinkamumo ir naudingumo lygio. Tačiau „SQL Server 2005“ suteikia galimybę organizacijai naudotis įmonėms skirtos versijos (angl. *Enterprise*) savybėmis su mažesne nuosavybės teisių įsigijimo kaina ir su didesniu lankstumu ir mažesniu sudėtingumu negu kad „Oracle 10g“.

### ▪ Eksploatacinės savybės ir išplečiamumas

Su panašia technine įranga – įmonės lygio serveris su 64 Intel Itanium dviem procesoriais – „SQL Server 2005“ pasirodo 7% geriau ir kainuoja 37% mažiau negu „Oracle 10g“.

### ▪ Kūrėjo produktyvumas

„SQL Server 2005“ integracija su „Microsoft Visual Studio“ ir su „Microsoft .NET“ technologijomis yra kur kas platesnė, vientisesnė, funkcionalesnė negu kad „Oracle 10 g“

integracija. 5 lentelėje matome, kad skirtingai, negu „Oracle“ BI savybės, „SQL Server“ BI savybės yra integruotos su „Visual Studio“.

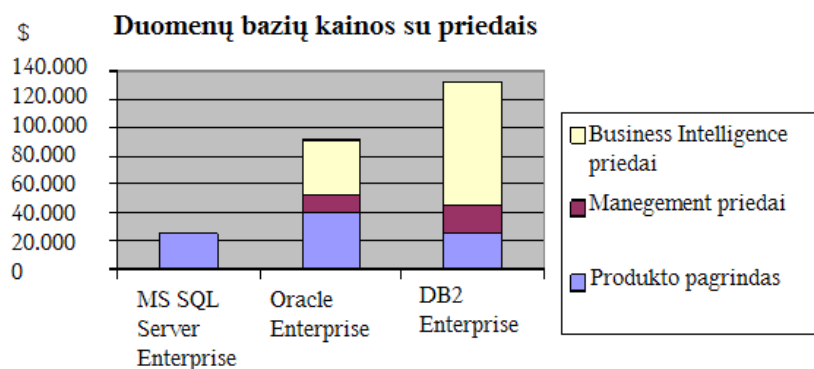
## 5 lentelė

### Visual Studio integracija su „SQL Server“ ir „Oracle“ programiniais paketais

Galimybė peržiūrėti DB objektus per Visual Studio	+ (per Server Explorer)	+ (per Oracle Explorer)
Galimybė ieškoti procedūrų klaidų per Visual Studio	+	-
Pagalba Visual Studio projektui	+	-
Automatinis DB objektų kūrimas per Visual Studio	+	-
Kontekstinė internetinė pagalba procedūrų redagavimui	-	+
BI technologijų integracija su Visual Basic	+	-

#### ▪ Nuosavybės teisių kaina

Kadangi pirkdamas „SQL Server 2005“ tu neturi dar papildomai pirkti įvairių priedų (saugumui užtikrinti, valdymo įrankių, ataskaitų ir analizės įrankių), pradinė kaina patrauklesnė vartotojui, negu kad „Oracle 10g“. Be to, „Oracle“ o“ politika didinti kainą už kiekvieną papildomą branduolį, jo DB licenzijos kainą padaro 4 kartus didesnę negu „SQL Server“. „Oracle“ o“ priežiūros ir palaikymo programos taip pat yra brangesnės. Duomenų bazių kainų palyginimas matomas 10 paveiksle.



10 pav. Duomenų bazių kainų palyginimas

#### ▪ BI (Business Intelligence) sistemų kūrimas

Nors BI programų kūrimui „Oracle“ pateikia daug ir įvairių įrankių, tačiau kiekvienas jų turi savitų ypatumų (anksčiau jie ir buvo perkami atskirai). Ir nors jie atlieka visas reikalingas funkcijas, tačiau kartu negali vientisai dirbti.

Tuo tarpu „Microsoft“ sukūrė integruotą BI serverio komponentų karkasą. „Integration Services“ naudojamos duomenų srautų valdymui iš ir į BI sistemą bei tarp jos komponentų. Duomenys lengvai keliauja tarp reliacinės aplinkos ir „Analysis Services“. „Reporting Services“ gali pasiekti reliacinius duomenis taip lengvai, kaip duomenis, esančius „Analysis

Services“. Stipri „SQL Server“ ir „Visual Studio“ integracija leidžia kurti BI programas su visu rinkiniu. Kadangi sukiniu lentelės ir yra kaip priedas „Analysis Services“, todėl Excel tampa neatskiriama BI programų dalimi.

Kuriamų programų tipų su „Oracle“ ir su „SQL Server“ populiarumas pavaizduotas 6 lentelėje :

**6 lentelė**

**„Oracle“ ir „SQL Server“ panaudojimas kuriant programas**

<b>Oracle</b>	<b>SQL Server</b>
1. Finansinės sistemos (angl. <i>Financial</i> )	1. <u>Verslo analitikos sistemos</u> (angl. <i>Business intelligence</i> )
2. Įmonės duomenų saugyklos (angl. <i>Enterprise data warehouse</i> )	2. Finansinės sistemos (angl. <i>Financial</i> )
3. <u>Verslo analitikos sistemos</u> (angl. <i>Business intelligence</i> )	3. Įmonės duomenų saugyklos (angl. <i>Enterprise data warehouse</i> )
4. Žmogiškųjų išteklių sistemos (angl. <i>Human resources</i> )	4. Ryšių su klientais valdymo sistemos (angl. <i>CRM</i> )
5. Ryšių su klientais valdymo sistemos (angl. <i>CRM</i> )	5. Žmogiškųjų išteklių sistemos (angl. <i>Human resources</i> )
6. Įmonės išteklių planavimo sistema (angl. <i>ERP</i> )	6. Žinių valdymo sistemos (angl. <i>Knowledge management</i> )

„Microsoft SQL“ kiekvieno produkto linijos leidimo savybės pateiktos 7 lentelėje [7]:

**7 lentelė**

**Microsoft SQL nauda ir pagrindinės savybės**

<b>Leidimas</b>	<b>Nauda</b>	<b>Dydis</b>	<b>Pagrindinės savybės</b>
Express	Greičiausias būdas sužinoti apie nesudėtingas duomenų valdymo programas, jas kurti ir naudoti. Šis leidimas nemokamas.	1 procesorius 1 GB RAM 4 GB DB	<b>4 GB duomenų bazė;</b> <b>Nesudėtingos ataskaitos;</b> Replikacija ir SSB klientas
Darbo grupė	Prieinamiausias ir paprasčiausias naudoti duomenų bazės sprendimas, skirtas mažesniems skyriams ir besiplečiančiam verslui.	1-2 procesoriai 3-GB RAM	Management Studio; Importavimas ir eksportavimas; Ribota replikacija; Klasteriai; <b>Atsarginių kopijų registravimo pristatymas</b>
Standartinis	Visa valdymo analizės platforma, skirta vidutiniam verslui ir didesniems skyriams.	1-4 procesoriai Neribota RAM	<b>Duomenų bazės atspindėjimas;</b> Pagrindinis ETL; Standartinis OLAP Server su Analysis Services; Standartinis ataskaitų kūrimas su Reporting Services; Duomenų gavyba; Pilnas replikavimas ir SSB publikavimas; Galimi 32 ir 64 bitų leidimai; Palaiko Itanium2 ir x64
Enterprise	Integruota duomenų	<b>Neribota skalė</b>	<b>Patobulintas duomenų bazės</b>

Leidimas	Nauda	Dydis	Pagrindinės savybės
	valdymo ir analizės platforma, skirta svarbioms įmonės verslui programoms.	ir skaidinių skaičius	<i>atspindėjimas, visos tinklo ir paralelinės operacijos bei momentinis duomenų bazės vaizdas;</i> Patobulinti analizės įrankiai, įskaitant OLAP ir duomenų gavybą; <i>Patobulintas ataskaitų kūrimas, įskaitant derinamas, plataus masto ir specialias ataskaitas;</i> Patobulintas ETL su sudėtingu duomenų maršrutu ir transformavimo galimybėmis; Galimi 32 ir 64 bitų leidimai; Palaiko Itanium2 ir x64

**Pastaba:** Pastorintu šriftu surašytos naujos „Microsoft SQL Server 2005“ savybės. Kiekviena toliau aprašyta versija turi anksčiau aprašyto leidimo savybes.

BI sistemos kūrimui buvo pasirinktas „SQL Server 2005“, nes visuose palyginimo punktuose jis lenkia „Oracle 10g“. Be to, viena iš svarbiausių „Microsoft SQL Server 2005“ savybių – aukšto laipsnio integracija tarpusavyje ir su kitais „Microsoft“ sukurtais produktais, pavyzdžiui, biuro programų paketu „Office“, „Visual Studio“. SQL Server 2005 platforma pateikta 11 paveiksle:



11 pav. SQL Server 2005 platforma

## 2.6 Analizės išvados

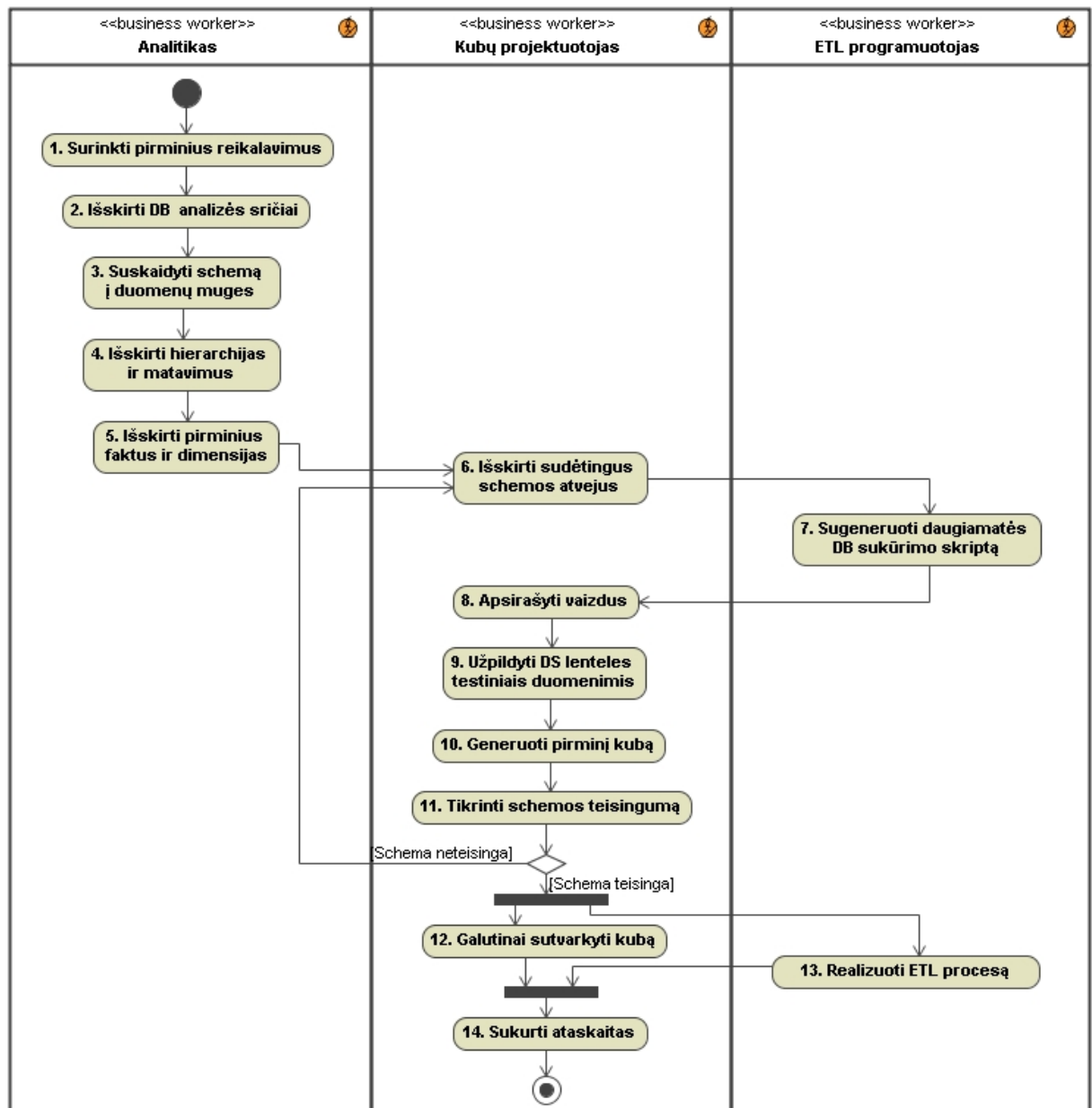
1. Saugyklų kūrimo metodų analizė parodė, kad kurti saugykla reikia pradėti nuo vartotojų tikslų ir poreikių modeliavimo ir jiems skirti daug dėmesio.
2. Žinomų šiuolaikinių autorių nuomone saugyklų schemoms modeliuoti tikslinga naudoti UML tada, kai kuriama nauja duomenų bazė, o kartu su ja kuriama ir saugykla. Tokiu atveju taisyklinga normalizuota duomenų bazė ir duomenų saugykla kuriamos iš vieno duomenų modelio, vėliau sugeneruojamas saugyklos modelis.
3. Kuriant saugykla iš esamų duomenų bazių dažnai tenka daryti labai individualius vienkartinis sprendimus, kurių automatizuoti (ir vaizduoti UML) neverta.
4. Paprastais atvejais normalizuota DB schema nesunkiai transformuojama į saugyklos schema, todėl tikslinga kurti normalizuotas saugyklų schemas, kadangi jos leidžia geriausiai pasinaudoti OLAP ir duoda teisingus analizės rezultatus.
5. Egzistuoja sudėtingos situacijos, kai nėra aišku, kaip saugykloje vaizduoti esamos reliacinės DB schema, taigi tolesnis darbo tikslas – rasti sprendimus sudėtingoms saugyklų schemų struktūroms vaizduoti ir išbandyti šiuos sprendimus, realizuojant juos pasirinktomis „MS SQL Server“ duomenų analizės priemonėmis.

### 3 Sudėtingų saugyklų struktūrų projektavimo metodika

Peržvelgus literatūroje sukurtus metodus, kaip pereiti nuo reliacinės duomenų bazės prie duomenų saugyklos, susidaro įspūdis, kad tam tikriems sudėtingiems atvejams reikia naudoti skirtingą metodiką. Reikia, kad metodas pasižymėtų tokiomis savybėmis:

- ✓ Turi būti paprastas ir suprantamas daugelio projektuotojų, kad nebūtų gaišamas bereikalingas laikas metodo įsisavinimui;
- ✓ Turi apimti didžiąją sudėtingų atvejų aibės dalį.

Metodo schema pateikta 12 paveiksle.



12 pav. Saugyklų projektavimo procesų diagrama

### 3.1 Reikalavimų surinkimas

#### 1 žingsnis. Pirminių reikalavimų surinkimas.

Šiame procese svarbu:

- ✓ Surinkti reikalavimus ir juos apibendrinti;
- ✓ Bendradarbiauti su IT skyriaus darbuotojais, susipažinti su naudojamais duomenų šaltiniais;
- ✓ Peržiūrėti naudojamus duomenis, inicijuoti reikalingus pakeitimus;
- ✓ Išanalizuoti naudojamą ataskaitas;
- ✓ Priskirti prioritetus susistemintiems reikalavimams.

#### 2 žingsnis. Reliacinės duomenų bazės išskyrimas analizės sričiai.

Šiame žingsnyje svarbu susikoncentruoti į analizės sritį, kuri atitinka klientų reikalavimus ir nagrinėti tik tuos duomenis, kurie reikalingi duomenų saugyklai. Patogu nusibraižyti ER diagramą arba duomenų bazės schemą, tada galima geriau stebėti, kaip vyksta tam tikrų ryšių ir objektų transformacija į daugiamatę schemą. Taip pat reiktų pateikti keletą testinių duomenų, kurie palengvintų snaičės schemas formavimą ir ETL proceso kūrimą.

#### 3 žingsnis. Schemas skaidymas į duomenų centrus (angl. *Data Mart*).

Šiame etape reikia susidaryti organizacijoje vykstančių procesų ir dimensijų matricą:

8 lentelė

Veiklos procesų ir dimensijų matrica

Veiklos procesai	Dimensijos		
	<u>Dimensija nr.1</u>	<u>Dimensija nr.2</u>	<u>Dimensija nr.3</u>
<u>Veiklos procesas nr. 1</u>	X	X	X
<u>Veiklos procesas nr. 2</u>	X	X	X
...	...	...	..

Duomenų centrai sudaromi pagal veiklos procesus. Dažna klaida, kuomet duomenų centrai sudaromi remiantis organizacijos skyriais, kadangi skirtingi skyriai neretai suinteresuoti tais pačiais veiklos procesais.

#### 4 žingsnis. Hierarchijų ir matavimų išskyrimas.

Remiantis 8 lentele ir surinktais reikalavimais, išskiriamos tam tikroje veiklos srityje nusistovėjusios hierarchijos, bei skaičiavimai (matavimai).



9 lentelė

Hierarchijų apibrėžimo lentelė

Hierarchijos pavadinimas	Lygmuo nr.1	Lygmuo nr.2	Lygmuo nr.3.	...
Hierarchija 1	aukščiausias lygmuo	žemesnis lygmuo	...	...
....	...	...	...	....

10 lentelė

Matavimų apibrėžimo lentelė

Matavimo pavadinimas	Formulė	Pastabos
Matavimas 1	COUNT (atributas 1)	...
....	....	....

5 žingsnis. Pirminis dimensijų ir faktų išskyrimas.

Šiame etape pavojinga per daug nenukrypti į duomenų bazės schemos detales ir suprojektuoti duomenų saugyklą, panašią į OLTP duomenų bazę.

Dimensijas patogiau formuoti iš sudaryto hierarchijų sąrašo, pridėdant kitus reikalingus atributus. Taip pat reikia apibrėžti veiklos (angl. *business*) ir surogatinius (angl. *surrogate*) raktus. Svarbu kiekvienai schemai (kubui) apibrėžti žemiausią dimensijų lygmenį.

Matavimai pagal atitinkamus detalumo lygius (angl. *granularity levels*) skirstomi į atskiras faktų lenteles, t.y. skirtingo detalumo matavimai negali būti toje pačioje faktų lentelėje. Teisingiausia būtų naudoti žemiausio lygmens duomenis modelyje, nes toks modelis yra lankstesnis tolimesniems pakeitimams.

Snaigės schema modeliuojama pasirinktame įrankyje, palaikančiame daugiamačius modelius. Tam pilnai pakanka ir „Visio Enterprise Architect Edition“ (Office 2003 arba Office 2007). Dimensijų atributų ir faktų matavimų pavadinimai turi atspindėti vartotojų veiklos reikalavimus.

Faktų sudėtingumą bei dimensijų detalumą faktuose atspindinti matrica:

11 lentelė

Dimensijų ir faktų matrica

Faktas	Dimensijos		
	<u>Dimensija nr.1</u>	<u>Dimensija nr.2</u>	<u>Dimensija nr.3</u>
<u>Faktas nr.1</u>	X (Lygmuo nr.1)	X (Lygmuo nr.1)	
<u>Faktas nr.2</u>	X (Lygmuo nr.1)	X(Lygmuo nr.2)	X(Lygmuo nr.3)
...	...	...	...

## 3.2 Daugiamatės schemos sudėtingų atvejų analizė

### 6 žingsnis. Sudėtingų atvejų išskyrimas formuojant žvaigždės/snaigės schema.

Dimensijos ir faktai skirstomi į kubus. Nerekomenduojama kiekvienam faktui kurti atskiro kubo, o verčiau sugrupuoti faktus, kurių matavimai susiję, į vieną kubą. Taip pat nereikia pernelyg perkrauti kubo, nes tai lems ilgiau trunkantį jo apdorojimą. Kubai gali naudoti tas pačias dimensijas, tokios dimensijos vadinamos bendromis (angl. *Shared Dimension*).

Projektuojant kubą reikia stengtis nenukrypti nuo žvaigždės schemas, jei yra keli faktai - formuoti taip vadinamus žvaigždynus. Prie snaigės schemas paprastai pereinama tada, kai:

- ✓ dimensijos duomenys yra iš skirtingų šaltinių;
- ✓ dimensija perkrauta tuščių arba besikartojančių reikšmių.

#### Atributo priskyrimas faktui arba dimensijai

Tam tikrais atvejais atributas gali būti tiek matavimas tiek dimensijos atributas. Tarkime, jei pagal tą atributą analizuojama (pvz. už kokią sumą pirkto klientas, turintis tam tikrą vaikų skaičių?), tuomet toks atributas (vaikų skaičius) yra dimensijos elementas. Jei su atributu atliekami tam tikri veiksmai (pvz. kiek vidutiniškai vaikų turi klientai?), tai toks atributas priskiriamas faktui kaip matavimas.

Taip pat yra atributų, kurių negalima priskirti išskirtoms dimensijoms: tai transakciją apibūdinančių vėliavėlių ir kitokių indikatorių. Yra keletas būdų, kaip vaizduoti šiuos žemo kardinalumo atributus:

- ✓ perkelti į faktų lentelę, tokiu būdu ją labai išplečiant;
- ✓ sukurti po atskirą dimensiją kiekvienam atributui ir taip bereikalingai apsunkinant modelį daugybe dimensijų;
- ✓ iš viso nenaudoti šių atributų, tikintis, kad niekas jų nepasiges;
- ✓ sugrupuoti juos vienoje taip vadinamoje šiukšlių dimensijoje (angl. *Junk Dimension*).

Priimtinausias yra paskutinis variantas. Tokio tipo dimensijoje atributai nėra artimai susiję (13 pav.). Jeigu visų galimų kombinacijų eilučių skaičius pakankamai mažas, šią dimensiją galima užpildyti iš anksto, kitu atveju – įterpti eilutes ETL'o metu.

Raktas	Transakcijos tipas	Apmokėjimo tipas	Apmokėjimo tipo grupė	Užsakymas internetu
1	Padengimas	Grynais	Grynieji	Taip
2	Pirkimas	Grynais	Grynieji	Taip
3	Pirkimas	Kortele	Kreditas	Ne
4	Pardavimas	Grynais	Grynieji	Taip
5	Pardavimas	Kortele	Kreditas	Ne

13 pav. Šiukšlių dimensija

Lėtai besikeičiančių dimensijų tipai

Kai kurie dimensijų atributai yra linkę kisti laike. Jei dimensija turi bent vieną tokį atributą, ją vadiname lėtai besikeičiančia dimensija (angl. *SCD – Slowly Changing Dimension*). Egzistuoja trys tokių dimensijų tipai:

- ✓ tipas 1 – nauja atributo užrašoma ant viršaus (UPDATE);
- ✓ tipas 2 – nauja reikšmė išsaugoma kitoje eilutėje (INSERT\UPDATE);
- ✓ tipas 3 – nauja reikšmė išsaugoma pridendant naują stulpelį dimensijoje (ADD COLUMN\UPDATE)

Raktas	Vardas	Pavardė	Skyrius	Pradžios data	Pabaigos data
1	Jurgis	Feironas	ADM	2008.01.01	2008.07.05
2	Jurgis	Feironas	PSS	2008.07.06	NULL

Raktas	Vardas	Pavardė	Skyrius	Dabartinė reikšmė
1	Jurgis	Feironas	ADM	FALSE
2	Jurgis	Feironas	PSS	TRUE

14 pav. SCD 2 tipo dimensijų realizacijos pavyzdžiai

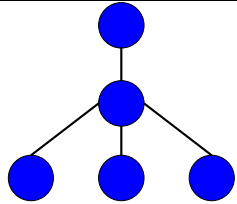
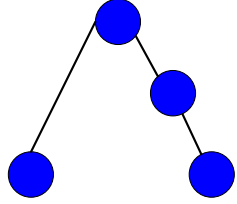
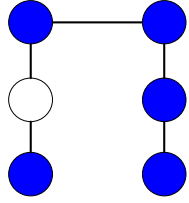
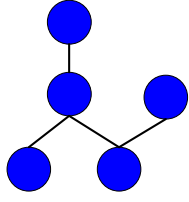
Hierarchijų tipai

Šiuolaikiniai OLAP įrankiai palaiko tiek griežtasias (angl. *strict*) arba natūralias hierarchijas, tiek leidžia apsibrėžti norimą kiekį kitokių hierarchijų. Hierarchijos reikalingos tam, kad duomenys būtų agreguojami kubo apdorojimo metu, o ne jo peržiūros metu. Apibrėžiant hierarchijas svarbu nurodyti ir ryšius tarp atributų hierarchijoje, kitaip duomenys bus agreguojami tik aukščiausiame ir žemiausiame lygiuose.

Hierarchijas galima suskirstyti į šias grupes:

12 lentelė

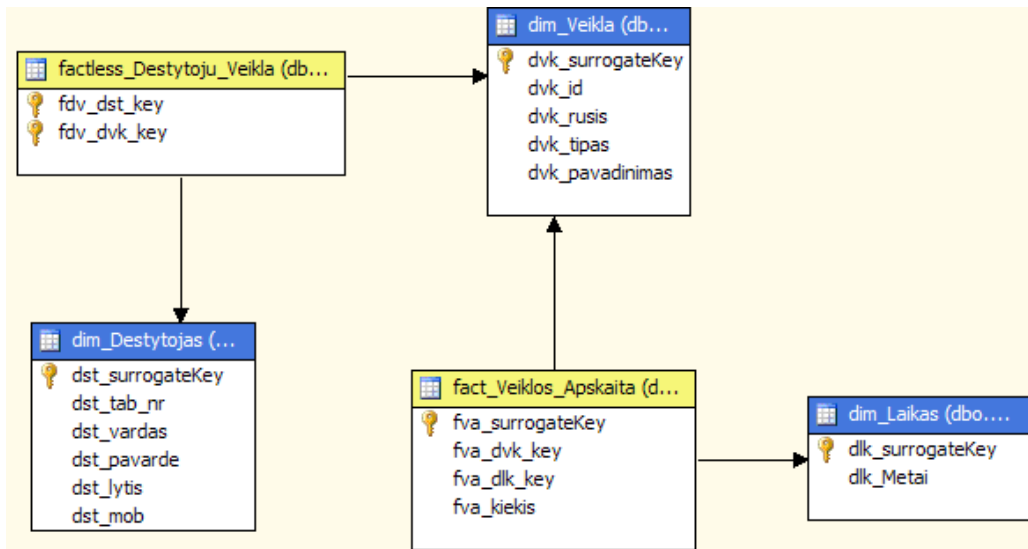
## Hierarchijų tipai

Hierarchijos tipas	Aprašymas	Atvaizdavimas
Subalansuota (angl. <i>balanced</i> )	Kiekvienas „vaikas“ priklauso tik vienam „tėvui“.	
Nesubalansuota (angl. <i>unbalanced</i> )	Lygiai tam tikrose hierarchijos šakose praleidžiami. Tokios hierarchijos būdingos rekursyviai OLTP DB ryšiui. OLAP srityje tokios hierarchijos vadinamos tėvo-vaiko (angl. <i>parent-child</i> ). Tokio tipo hierarchijas jau palaiko dauguma įrankių. Dimensijoje šiai hierarchijai realizuoti naudojami du laukeliai: viename rašoma „tėvo“ kodo reikšmė, antrame – „vaiko“. Dar kitais atvejais hierarchija realizuojama kaip subalansuota hierarchija (ypač jei lygių skaičius nėra labai kintantis). Tačiau jeigu reikalinga įrašų istorija, tokios hierarchijos realizuojamos kaip daug-su-daug ryšys.	 Pvz. organizacijos arba darbuotojo dimensija.
Nepilnos (angl. <i>ragged</i> )	Skirtingi atributai tam tikrame lygyje neturi reikšmių. Patariama nesančias reikšmes palikti tuščias arba įrašyti „tėvo“ reikšmę.	 Pvz. geografijos dimensija. (šalis → valstija → miestas)
Tinklo (angl. <i>network</i> )	Daugiau nei vienas tėvas. Tokiu atveju reikėtų hierarchiją skaidyti į keletą skirtingų hierarchijų.	

Tam tikri įrankiai (pvz. MS SQL Server Analysis Services) neleidžia kurti hierarchijų iš skirtingų dimensijų. Tokiu atveju reikia rinktis: jei hierarchija ypač reikalinga, sujungiamo dimensijas formuodami snaigę, tačiau modelis tampa sudėtingesnis ir užklausų greitis gali sumažėti.

### Daug – su –daug ryšiai

Šio tipo ryšys atsiranda tuomet, kai keletas dimensijos reikšmių gali būti priskirtos vienai fakto transakcijai. Paprastai daug-su-daug ryšiui modeliuoti naudojama befaktė (angl. *factless*) lentelė, kurioje nurodomos esamos daug-su-daug ryšio dimensijų reikšmės. OLTP duomenų bazėje tai atitinka tarpinę duomenų bazės lentelę.



15 pav. Daug-su-daug ryšio pavyzdys

Šiame pavyzdyje veiklos lentelėje yra publikacijų ir leidinių duomenys. Kiekvienas dėstytojas gali publikuoti kelias publikacijas ir kiekvieną publikaciją gali publikuoti vienas ir daugiau dėstytojų (analogiškai leidinių atžvilgiu). Todėl susiduriama su daug-su-daug ryšiu tarp dimensijų. Daug-su-daug ryšiui tarp veiklos ir dėstytojo dimensijų modeliuoti, sukuriama befaktė dėstytojų veiklos lentelė.

Daug-su-daug dimensijų užklausų greitis tiesiogiai priklauso nuo eilučių skaičiaus befaktėje lentelėje. Jei užklausos vykdomos lėtai, reiktų kurti particijas befaktei lentelei.

### Raktai

Kiekviena dimensija turi pirminį raktą „integer“ tipo (identity). Laiko dimensijoje geriausiai naudoti „integer“ tipo pirminį raktą datos formatu (pav. 20091020). Tuomet atliekant užklausas WHERE sąlygoje filtruojant pagal datą nereikės jungti laiko dimensijos.

Faktų lentelės raktą sudaro išorinių (angl. *foreign*) raktų rinkinys, tai vadinamasis sudėtinis (angl. *composite*) raktas. Tačiau, jei iš veiklos pusės tikėtina, kad fakte bus tokių pačių eilučių, patogu turėti atskirą „integer“ tipo pirminį raktą.

*Null* reikšmės gali atsirasti tiek faktų, tiek dimensijų lentelėse. Kad būtų galima užtikrinti teisingus skaičiavimus, dimensijų lentelėse reikia sukurti atitinkamas „tuščias“ eilutes su tam tikra rakto reikšme (pvz., „- 1“), kurios atsirastų faktų lentelėje vietoj tuščių reikšmių.

### 3.3 Duomenų saugyklos sukūrimas

#### 7 žingsnis. Daugiamatės duomenų bazės sukūrimo skripto generavimas.

Prieš generuodamas duomenų saugyklos lentelių sukūrimo skriptą, ETL programuotojas peržiūri modelį ir prideda papildomus atributus, reikalingus ETL procesui. Jei daugiamatės schemas projektavimui buvo naudojamas „Visio Enterprise Architect Edition“, duomenų bazės skriptas gali būti sugeneruotas pasirinkus: *Database -> Generate*

#### 8 žingsnis. Reikalingų vaizdų (angl. views) sukūrimas.

Vaizdai kuriami rašant SELECT užklausas ir išskiriant reikalingus laukus kubui, parenkant veiklos srities vartotojų naudojamus dimensijų pavadinimus, sutvarkant datas ir kitų laukų formatus, jungiant laukus, atliekant kitus paprastus pertvarkymus.

#### 9 žingsnis. Duomenų saugyklos lentelių užpildymas testiniais duomenimis.

Lentelės užpildomos testiniais duomenimis, stengiantis įterpti kuo daugiau ypatingų atvejų.

### 3.4 Kubo projektavimas

#### 10 žingsnis. Pirminis kubo generavimas.

Vedliu suformuojamas pirminis kubas atidžiai peržiūrint ryšius tarp faktų ir dimensijų.

#### Dimensijų tipai

Dimensijas pagal jų ryšio tipus galima skirstyti į penkias grupes (13 lentelė).

13 lentelė

Dimensijų tipai

Dimensijos tipas	Aprašymas
Reguliari dimensija (angl. <i>Regular Dimension</i> , <i>Star Dimension</i> )	Reguliaros dimensijos ryšys atspindi tiesioginį ryšį tarp dimensijos ir faktų lentelės, kurios siejasi per pirminį – išorinį (angl. <i>foreign</i> ) raktą tradicinėje žvaigždės scheme.
Nuorodos dimensija (angl. <i>Reference Dimension</i> , <i>Snowflake Dimension</i> )	Nuorodos dimensijos ryšys vaizduoja ryšį tarp dimensijų ir fakto snaigės scheme, t.y. kai norime sujungti vaiko dimensiją su faktu per tėvo dimensiją.
Vaidmens dimensija (angl. <i>Role playing Dimension</i> )	Ta pati dimensija gali būti sujungta su faktų rinkiniu per keletą vaidmenų. Pavyzdžiui, faktas Užsakymai gali turėti užsakymo datą, išsiuntimo datą ir gavimo datą. Tokiu atveju datos dimensija turės tris vaidmenis.

<b>Dimensijos tipas</b>	<b>Aprašymas</b>
Daug-su-daug dimensija (angl. <i>Many-to-many Dimension</i> )	Šis ryšys atsiranda tuomet, kai norime sujungti vieną faktą su keletu dimensijos narių.
Fakto dimensija (angl. <i>Fact based Dimension, Degenerate Dimension</i> )	Tai tokios dimensijos, kurių atributai yra fakte. Reikalinga dimensinė informacija yra laikoma fakte, kad išvengti kartojimosi.

### Matavimų tipai

Aprašant matavimus, reikia atkreipti dėmesį, kad tam tikriems matavimų tipams yra taikomi ribojimai (14 lentelė)

#### 14 lentelė

#### Matavimų tipai

<b>Matavimo tipas</b>	<b>Aprašymas</b>
Adityvus (angl. <i>Additive</i> )	Adityvus per visas dimensijas. (Pvz. SUM, COUNT, MIN, MAX).
Pusiau adityvus (angl. <i>Semi-additive</i> )	Adityvus per visas dimensijas, išskyrus vieną (laiko dimensiją). (Pvz. likutis, balansas)
Neadityvus (angl. <i>Non-additive</i> )	Neadityvus per visas dimensijas (Pav. procentai, koeficientai)

### Faktų tipai

#### 15 lentelė

#### Faktų tipai

<b>Fakto tipas</b>	<b>Aprašymas</b>
Transakcijos tipo (angl. <i>transaction</i> )	Saugo kiekvieną transakciją kai tik ji atsiranda tam tikru diskrečiu momentu – kai transakcija įvyksta.
Periodinė momentinė nuotrauka (angl. <i>periodic snapshot</i> )	Saugo didėjančius duomenis tam tikrais laiko periodais (pvz. visų duomenų nuotrauka daroma mėnesio pabaigoje). Vartotojai greitai gali gauti viso sumas, kadangi periodinė momentinė nuotrauka agreguoja daugelį faktų laike.
Sukaupta momentinė nuotrauka (angl. <i>accumulating snapshot</i> )	Sukaupta momentinė nuotrauka yra pastoviai atnaujinama laike. Tokiuose faktuose saugomos kelios datos, kad užfiksuoti, kada rūpimas objektas perėjo tam tikrą veiklos procesą.

### **11 žingsnis.** Schemas teisingumo patikrinimas tiek kubo kūrimo aplinkoje, tiek kliento įrankio aplinkoje, stengiantis prisiderinti prie įrankio ypatumų.

Tam tikrais atvejais reikia pritaikyti schemą arba ieškoti kito įrankio, kuris palaikytų tam tikrus schemas atvejus. Pastaba. Šis metodas išbandytas tik MS SQL Server įrankių aplinkoje.

### **12 žingsnis.** Galutinis kubo sutvarkymas

Sutvarkomos visos hierarchijos, ryšiai tarp atributų dimensijoje, ryšiai tarp matavimų grupių ir dimensijų, rašomos papildomas (MDX) užklauskos reikalingiems matavimams, matavimų grupėms arba matavimas iš skirtingų faktų, apibrėžiami veiklos rodikliai (angl. *KPI – Key Performance Indicators*), apibrėžiamos particijos, priskiriamos teisės vartotojams. Norint apibrėžti teises vartotojams, kuriami vaidmenys, kuriems priskiriami vartotojai. Priskyrus vartotoją vaidmeniui galima leisti naršyti tiek visą kubą, tiek riboti naršymą matavimams, dimensijai ar net tam tikriems atributams. Taip pat ribojimus galima užrašyti MDX kalba.

## **3.5 Duomenų surinkimo, išvalymo ir įkrovimo proceso realizavimas**

### **13 žingsnis.** ETL proceso realizavimas

Planuojant ETL procesą, reikėtų atlikti tokius projektavimo žingsnelius:

- ✓ Sukurti abstraktaus lygio žemėlapi;
- ✓ Atlikti detalią duomenų analizę ir šaltinių susiejimą. Patartina pasirinkti tam skirtą "Profiling Tool";
- ✓ Apsibrėžti, kas kiek laiko vyks duomenų įkrovimas;
- ✓ Apsibrėžti, koks kiekis istorinių duomenų bus įkrautas į kiekvieną lentelę (istorinis arba didėjimo įkrovimas).
- ✓ Sukurti particijų sukūrimo strategiją „Analysis Services“ faktų lentelėms;
- ✓ Sukurti duomenų ištraukimo iš kiekvieno šaltinio strategiją.

Apibendrintai, ETL projekto sukūrimą sudaro šie svarbūs punktai:

1. Apibrėžti pradinis duomenų paėmimo ir duomenų padėjimo šaltinius;
2. Apibrėžti šaltinių, kintamųjų konfigūracijas;
3. Realizuoti dimensijų apdorojimą;
4. Realizuoti faktų apdorojimą;
5. Apibrėžti informacijos apie paketų įvykdymą saugojimą (angl. *logging*);
6. Apibrėžti įvairius įvykius (jei pakete įvyko klaida);
7. Ištestuoti paketus po vieną, pamažu jungiant į visumą;
8. Instaliuoti paketus ir paleisti juos vykdyti kas tam tikrą periodą;



### 3.6 Ataskaitų sukūrimas

#### 14 žingsnis. Ataskaitų kūrimas

Ataskaitos kuriamos šablono pobūdžio, nes jei įrankis turi pakankamai funkcionalumo (kaip kad Excel ypač nuo 2007 versijos), vartotojai gali patys kurti dinamines ataskaitas, t.y. pjaustyti kubą jiems rūpimais pjūviais. Daugiau statines ataskaitas patogiau sukelti į Sharepoint įmonės portalą.

Microsoft ataskaitoms kurti rekomenduoja Reporting Services, Proclarity, Performance Point Server įrankius.

## 4 Eksperimentinis tyrimas, taikant metodą realiai sistemai

Eksperimento tikslas – patikrinti, ar metodą galima taikyti realiame gyvenime ir ar sukurti sprendimai teisingi. Metodas buvo taikomas kuriant aukšto lygio analitines ataskaitas, skirtas projektų, darbuotojų, veiklos grupių, pardavimų ir išlaidų analizei. Duomenys ataskaitoms surenkami iš dviejų šaltinių (Navision sistemos ir projektų registro), tada atrenkami, integruojami, agreguojami ir užkraunami į duomenų saugyklą. Didžiausiose dimensijose kaupiama iki 30 tūkst. įrašų, į faktus kas mėnesį įterpiama iki 50 tūkst. įrašų.

Ataskaitų vartotojai (užsakovai) – skyrių vadovai/-ės, jų asistentai/-ės ir direktoriai/-ės. Navision sistemoje pateikiamos ataskaitos yra statinės, vizualiai labai skurdžios, o analitiniai objektai negali būti hierarchiniai. Taip pat daugelio pjūvių negalima pamatyti dėl nesuintegruotų dviejų sistemų duomenų. Todėl ir buvo nuspręsta kurti verslo analitikos sistemą, kuri patenkintų nesančių ataskaitų ir galimybių poreikį.

### 4.1 Reikalavimų surinkimas

Reikalavimai surinkti susirinkimų metu juos užrašant neformalia kalba:

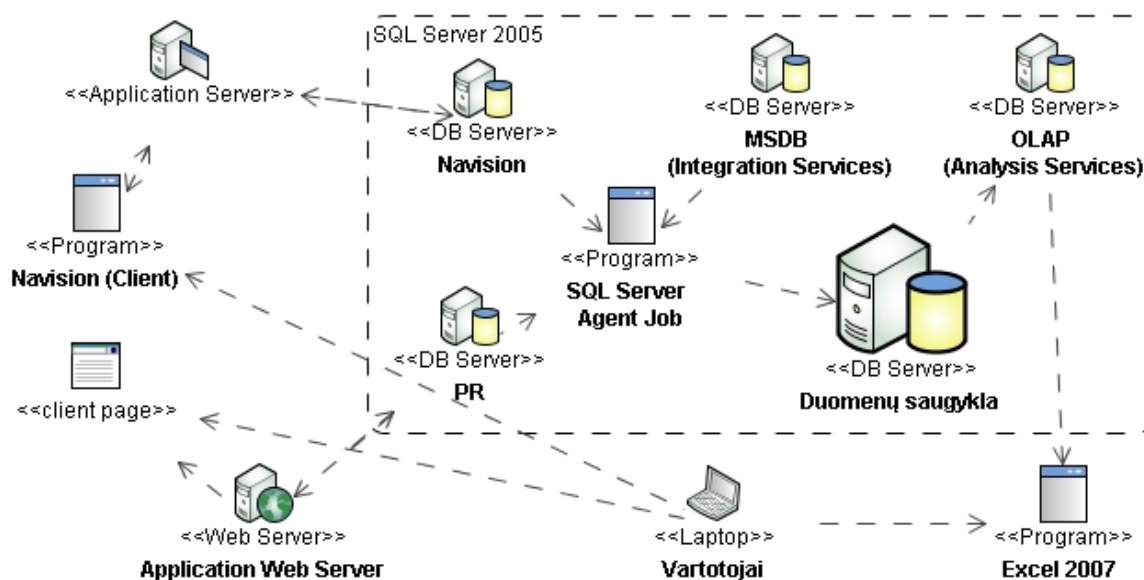
#### PLANUOJAMŲ BI ATASKAITŲ SĄRAŠAS IR STRUKTŪRA

1. Aukšto lygio ataskaita, kurioje turėtų būti vaizduojami KPI: faktinės ir planuotos valandos, KPI maksimali reikšmė procentinė iki 150 ar 200 maksimaliai galimų išdirbti valandų (kurios skaičiuojamos atsižvelgiant į nedarbo dienas), filtrai (projektų tipai ir periodas (paskutinis ketvirtis, paskutinis pusmetis, paskutiniai metai)), pardavimų suma (kiek faktiškai išrašyta sąskaitų faktūrų), projektų faktinė valandos kaina, darbuotojų skaičius (žmogaus mėnesių skaičius), vėluojančių projektų skaičius arba/ir kiek vidutiniškai dienų vėluojama (sumuojama pagal kiekvieno projekto kiekvienos užduoties vėlavimą), einamųjų projektų skaičius.
2. Ataskaita, kurią sudaro grafikas su faktinių ir planuotų val. skaičiumi pagal projekto tipą, veiklos grupę ir šią informaciją detalizuojančios ataskaitos. Iš šių ataskaitų yra galimybė peržiūrėti: projektą, skyrių, užduoties veiklas ir darbuotojus pasirinktu laiko momentu, komercinių projektų val. palyginimas su bendru val. skaičiumi. Pateikiamas detalesnis ataskaitų aprašymas:
  - 2.1. *Valandos pagal projektų tipus ir veiklos grupes.* Šioje ataskaitoje turėtų būti filtras laiko periodui pasirinkti (metai – mėnuo), du grafikai: vienas vaizduojantis faktiškai išdirbtas valandas pagal projekto tipus, kitas – veiklos grupes. Taip pat gali būti lentelė, kurioje bus pateikti duomenys (faktinis ir planuotas val. skaičius) pagal veiklos grupes.
  - 2.2. *Sugaištos ir planuotos valandos pagal projektą, taip pat projekto vadovo nuomonė procentais apie projekto įvykdymą.* Pirmiausia reiktų atkreipti dėmesį į projektui sugaištą faktinį laiką ir jam planuotą. Grafikas, apibūdinantis pasirinktą projektą, turėtų vaizduoti bendrą projekto val. pasiskirstymą pagal užduoties veiklas ir kiek kiekvienas skyrius ir/arba darbuotojas faktiškai praleido valandų pagal užduoties veiklą. Taip pat reikia matyti, kas yra projekto vadovas ir kokia, jo nuomone, projekto dalis procentais yra įvykdyta pasirinktu ataskaitos skaičiavimo periodu.
  - 2.3. *Vidutinis faktiškai sugaištų valandų skaičius.* Filtruojama turėtų būti pagal datą, projekto tipą, veiklos grupę ir projektą. Vaizduojamas grafikas pagal skyrius su detalizacija iki darbuotojų.

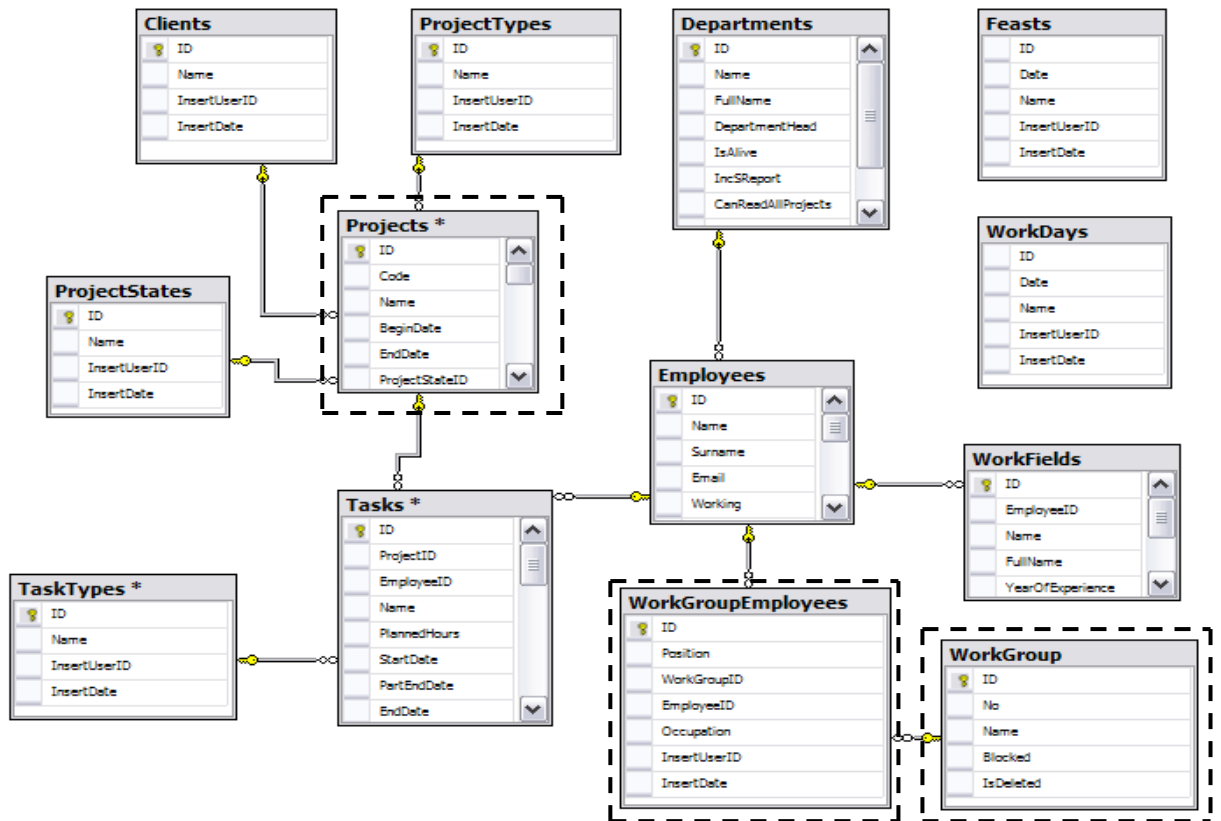
- 2.4. *Žmonių užimtumas pagal skyrius ir veiklos grupes, laisvos val. pagal skyrių.* Turėtų būti galima nurodyti laiko periodą, pasirinkti skyrių ir darbuotojus. Grafike matytume planuotas val. pagal skyrius su galimybe detalizuoti iki darbuotojų. Nesuplanuotas valandas turėtume matyti pagal skyrius.
3. *Faktinės pajamos.* Trys grafikai pagal klientą, veiklos sritį, veiklos grupę, projektą, taip pat skaičius užsakymų sumos. Ataskaitoje turėtų būti laiko periodo pasirinkimas. Filtras pagal klientus. Pasirinkus konkretų klientą, arba visus klientus, bus galima pamatyti faktinį pajamų pasiskirstymą pagal veiklos sritį, veiklos grupę, skyrių ir projektą.
  4. *Žmogaus mėnesių faktinis skaičius.* Tai ataskaita, kuri parodytų, kiek pasirinktu laiko momentu dirbo žmonių, t.y. jei pilną mėnesį dirbo 15 žm. pilnu etatu, 3 žm. po 0,5 etato ir 1 žmogus buvo priimtas į darbą nuo mėnesio 15 d., vadinasi, žmonių sk. dirbusių tą mėn. 17. Taip pat turi būti galimybė analizuoti pagal darbuotojų darbo sritis (specializacijas).
  5. *Faktinės sąnaudų mokymams.* Ataskaita turėtų atsakyti į klausimą, kiek kiekvienas skyrius išleido pinigų mokymams pasirinktu laiko momentu (pvz.: už metus). Vaizduoti galėtų duomenų lentelę, su filtru pagal skyrius su galima detalizacija iki darbuotojų.

Organizacijoje naudojami šie duomenų šaltiniai:

- ✓ projektų registro duomenų bazė;
- ✓ navision sistemos duomenų bazė.

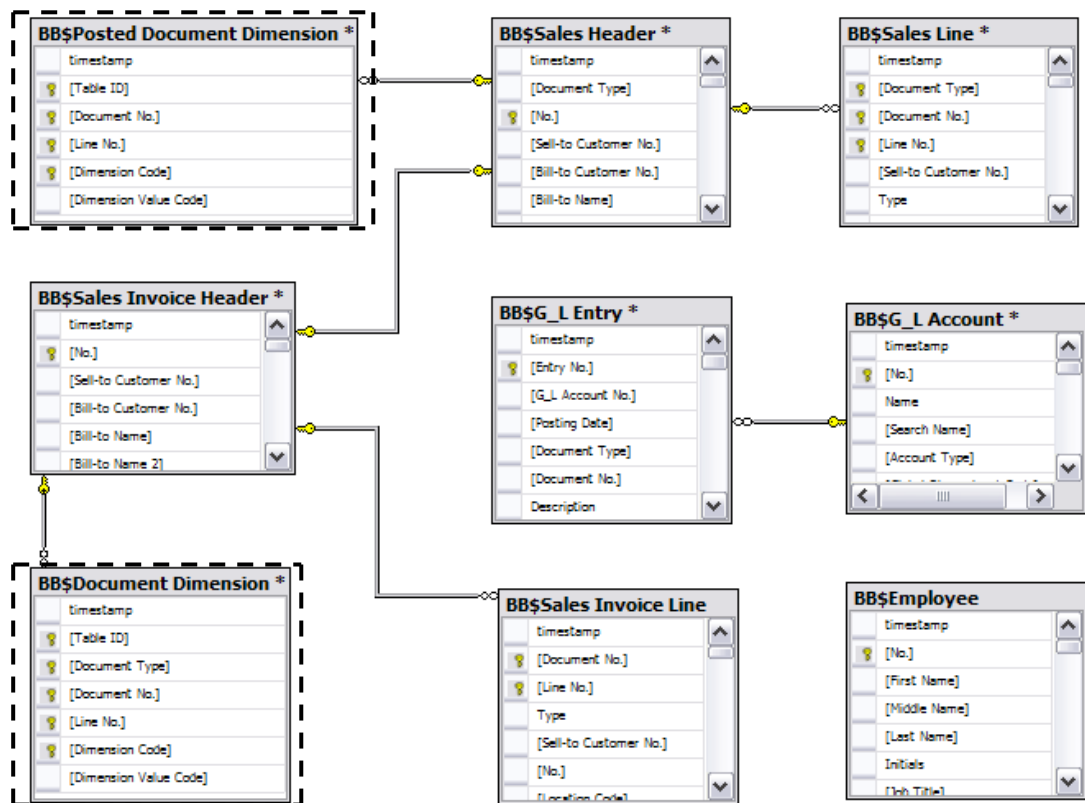


16 pav. Architektūros diagrama



17 pav. Projektų registro duomenų bazės išskirtų lentelių schema

Projektų registras – tai interneto sistema, kurioje fiksuojami projektai, darbuotojų atliktos užduotys ir joms sugaištų valandų skaičius. Taip pat yra galimybė suformuoti įvairias paruoštas ataskaitas, kurios padeda vadovams sudaryti planus stebint skyrių užimtumą.



18 pav. Navision sistemos duomenų bazės išskirtų lentelių schema

Navision sistemoje skaičiuojami darbuotojų atlyginimai, fiksuojami pirkimai ir pardavimai, pardavimų galimybės, planai ketvirčių tikslumu ir kita. Tai sistema su specifine lentelių struktūra, t.y. lentelės su daugybe laukų, kuriose nėra išorinių raktų.

Atlikus duomenų analizę, pastebėta, jog veiklos grupės yra tik preliminariuose užsakymuose ir išrašytose sąskaitose faktūrose (Navision sistema), tačiau jos niekaip nesisieja su darbuotojais ar projektais. Taip pat pastebėta, kad sistemose nėra ir kitų ataskaitose reikalingų duomenų. Todėl buvo inicijuoti šie pakeitimai sistemose (duomenų bazės diagramose pakeistos ar sukurtos lentelės apibrauktos punktyrine linija):

- ✓ projektų registro duomenų bazėje dbo.Projects lentelėje pridėtas papildomas stulpelis, skirtas projekto įvertinimui. Šiuos duomenis suves projektų vadovai.
- ✓ projektų registro duomenų bazėje papildomai sukurtos dbo.WorkGroupEmployees ir dbo.WorkGroup lentelės. dbo.WorkGroup lentelė užpildoma veiklų grupių duomenimis iš Navision sistemos. dbo.WorkGroupEmployees lentelėje kaupiami duomenys apie tai, koks darbuotojas kiek procentų dedikuotas tam tikrai veiklai.
- ✓ navision duomenų bazėje įvesta papildoma dimensijos reikšmė „PROJEKTAS“ ir dbo.[BB\$Posted Document Dimension] bei dbo.[BB\$Document Dimension] lentelėse pradėta vesti informacija apie užsakymų ir sąskaitų faktūrų eilučių sąsają su projektais.

Sistemų integravimas yra sudėtingas procesas ir jį reiktų pradėti planuoti kuo anksčiau, apsvarstyti kelis variantus. Iš BI pusės geriausiai turėti visus duomenis žemiausiame lygmenyje, todėl buvo svarstomas variantas, jog veiklos grupę reiktų priskirti įvedant užduotis darbuotojams, kurios priklauso tam tikram projektui. Tačiau klientai priešinosi šiam pakeitimui, kadangi užduotis laisvai įvedinėti gali bet kuris darbuotojas ir kiekvienai užduočiai nurodyti tam tikrą veiklą būtų pernelyg varginantis darbas. Tai lėmė veiklos grupių priskyrimą prie sąskaitų faktūrų ir užsakymų eilučių, tačiau taip pat - sudėtingus skaičiavimus faktų įkrovimo metu.

Duomenys bus kraunami kaip nuotraukos, įkrovimai vyks kas mėnesį. Kiekvieno įkrovimo datos saugomos lentelėje „Skaičiavimo data“. Kadangi ateityje gali iškilti poreikis dažnesniam duomenų įkrovimui, skaičiavimo datos dimensijoje yra stulpelis, nurodantis, įkrovimo tipą.

16 lentelė

Veiklos procesų ir dimensijų matrica

Procesai	Dimensijos						
	Data	Veiklos	Projektai	Užduotys	Darbuotojai	Klientai	Valiuta
Valandų apskaita pagal projektus	X	X	X	X	X	X	
Pardavimai	X	X	X			X	X
Projektų įvykdymas			X				
Darbuotojų apskaita					X		
Dedikacija		X			X		
Valandų apskaita pagal užduotis	X		X	X	X	X	
Išlaidų mokymams apskaita	X				X		

Lentelėje aprašomos viso įmanomos hierarchijos su skirtingais keliais, tačiau pasirinktas kliento pusės įrankis leidžia nurodyti hierarchijų laukus, dėl to kuriamos maksimaliai ilgiausio kelio hierarchijos.

17 lentelė

Hierarchijų apibrėžimo lentelė

Hierarchijos pavadinimas	Lygmuo nr.1	Lygmuo nr.2	Lygmuo nr.3.	Lygmuo nr.4.	Lygmuo nr.5.
Skaičiavimo Data	Laikotarpis	Metai	Ketvirtis	Mėnuo	Data
Periodas	Metai	Ketvirtis	Mėnuo	Data	
Veiklos hierarchija	Veiklos sritis	Veiklos grupė			
Darbuotojo hierarchija	Skyrius	Vardas ir Pavardė			
Projekto hierarchija	Klientas	Projekto tipas	Projektas		

18 lentelė

Matavimų apibrėžimo lentelė

Matavimo pavadinimas	Formulė	Pastabos
<u>Vidutinis darbuotojų skaičius</u>	$(VID_{sk_{praeitam\acute{e}n}} + SUM(VG_{darb}) / 100) / 2$	VID <sub>sk<sub>praeitamėn</sub></sub> - vidutinis darbuotojų skaičius per mėnesį praeito krovimo metu, VG <sub>darb</sub> – darbuotojui priskirtos veiklos grupės procentas krovimo momentu.
<u>Sąskaitų faktūrų suma</u>	SUM(SF)	SF – sąskaitų faktūrų suma
<u>Užsakymų suma</u>	SUM(UZSK)	UZSK – užsakymų suma
<u>Projektų įvykdymas</u>	(Procentai - %)	Projekto vadovo procentinis projekto įvykdymo įvertinimas.
<u>Išlaidų mokymams suma</u>	SUM(IŠL <sub>mokymai</sub> )	IŠL <sub>mokymai</sub> – išlaidos mokymams. Atrenkamos tik toks išlaidos, kurių dbo.[BB\$G_L Account].[No.] LIKE '6 13%'
<u>Dedikacija</u>	(Procentai - %)	Darbuotojas gali būti priskiriamas vienai ar kelioms veikloms, t.y. dedikuojamas tam tikra dalimi. Jei priskiriamas vienai veiklai, pvz.: v1, vadinasi 100 % savo darbo laiko dirba v1 veikloje, jei darbuotojas dirba keliose veiklose, tai ir procentai atitinkamai padalinami.

<b>Matavimo pavadinimas</b>	<b>Formulė</b>	<b>Pastabos</b>
<u>Vėlavimo dienos</u>	$(\text{SUM}(\text{VAL}_{\text{suplanuotos}}) - \text{SUM}(\text{VAL}_{\text{išdirbtos}})) * 8$	$\text{VAL}_{\text{suplanuotos}}$ – suplanuotos užduočiai valandos, $\text{VAL}_{\text{išdirbtos}}$ – išdirbtos valandos atliekant užduotį. Skaičiuojama tik toms užduotims, kurios nėra pasibaigusios.
<b>VALANDOS</b>		$\text{VAL}_{\text{išdirbtos}}$ – išdirbtos valandos žmogui sumuojant to paties mėnesio valandas, $\text{VAL}_{\text{suplanuotos}}$ – suplanuotos valandos žmogui sumuojant to paties mėnesio valandas, DD – darbo dienos tam tikrą mėnesį, $(\text{DD} * 8)$ – maksimaliai galimos išdirbti valandos per mėnesį, $\text{SUM}(\text{VG}_{\text{darb}})$ – veiklos grupių procentų suma tą mėnesį darbuotojui priskirtų veiklos grupių (Distinct). Jei yra bent viena nepriskirta veikla, tai daliname iš $(\text{SUM}(\text{VG}_{\text{darb}}) - 100\%)$ , jei visai nėra priskirtų veiklų – 100%.
<u>Išdirbtos valandos</u>	$(\text{SUM}(\text{VAL}_{\text{išdirbtos}}) * \text{VG}_{\text{darb}}) / \text{SUM}(\text{VG}_{\text{darb}})$	
<u>Suplanuotos valandos</u>	$(\text{SUM}(\text{VAL}_{\text{suplanuotos}}) * \text{VG}_{\text{darb}}) / \text{SUM}(\text{VG}_{\text{darb}})$	
<u>Dedikuotos valandos</u>	$(\text{DD} * 8 * \text{VG}_{\text{darb}} / 100 * \text{VG}_{\text{darb}}) / \text{SUM}(\text{VG}_{\text{darb}})$	

Remiantis išskirtomis hierarchijomis formuojamos dimensijos. Faktai formuojami remiantis išskirtais matavimais, stengiantis, kad dimensijos fakte būtų vienodo lygmens.

19 lentelė

## Dimensijų ir faktų matrica

Faktas	Dimensijos									
	Periodas	Skaiciavimo Data	Darbuotojai	Darbo Sritys	Projektai	Užduotys	Veiklos	Išlaidos	Tipas	Valiuta
<u>Projektų Ivertinimas</u>		X (Mėnuo)			X (Projektas)					
<u>Darbuotojų Skaičius</u>		X (Mėnuo)	X (Darbuotojas)							
<u>Priskyrimas</u>		X (Mėnuo)	X (Darbuotojas)				X (Veiklos gr.)			
<u>Valandos</u>	X (Mėnuo)	X (Mėnuo)	X (Darbuotojas)		X (Projektas)		X (Veiklos gr.)			
<u>Užsakymai ir SF</u>	X (Diena)	X (Mėnuo)			X (Projektas)		X (Veiklos gr.)		X (Tipas)	X (Valiuta)
<u>Užduočių Valandos</u>	X (Diena)	X (Mėnuo)	X (Darbuotojas)			X (Užduotis)				
<u>Befaktė Darbo Sritis</u>		X (Mėnuo)		X (Darbo Sritis)						
<u>Išlaidos Mokymams</u>		X (Mėnuo)	X (Darbuotojas /Skyrius)					X (Išlaidų rūšis)		

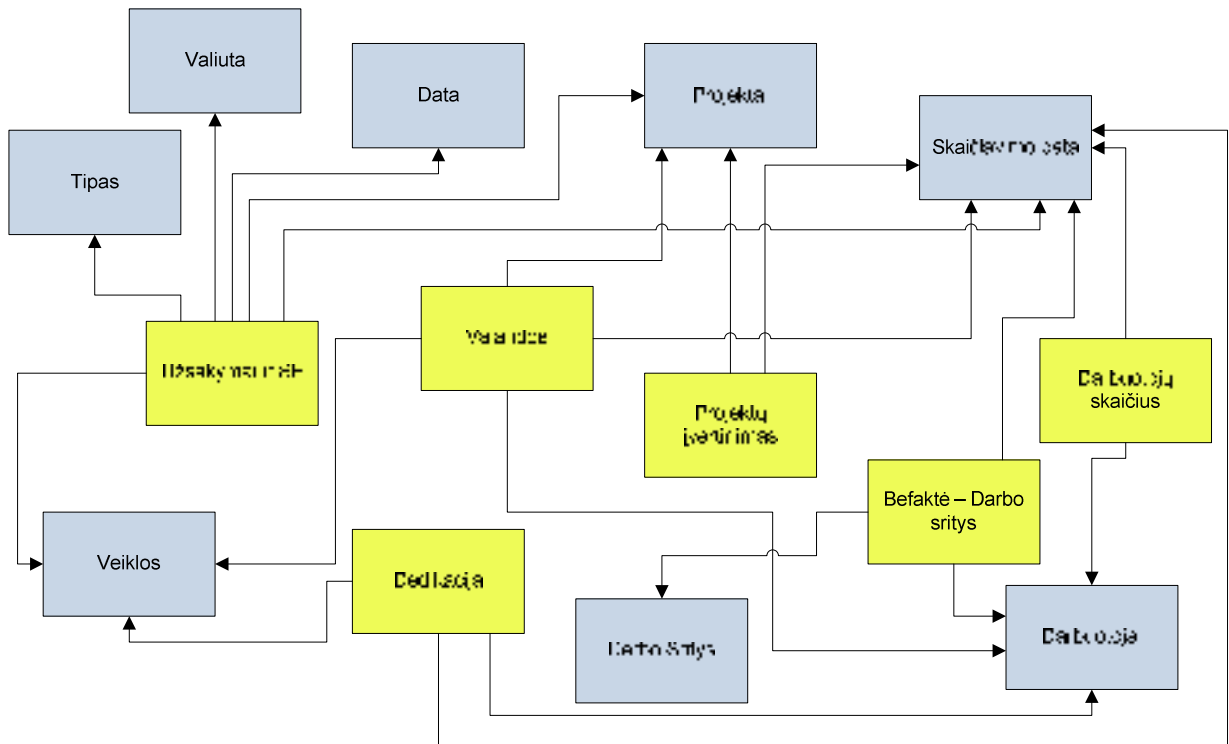


## 4.2 Daugiamatės schemos sudėtingų atvejų analizė

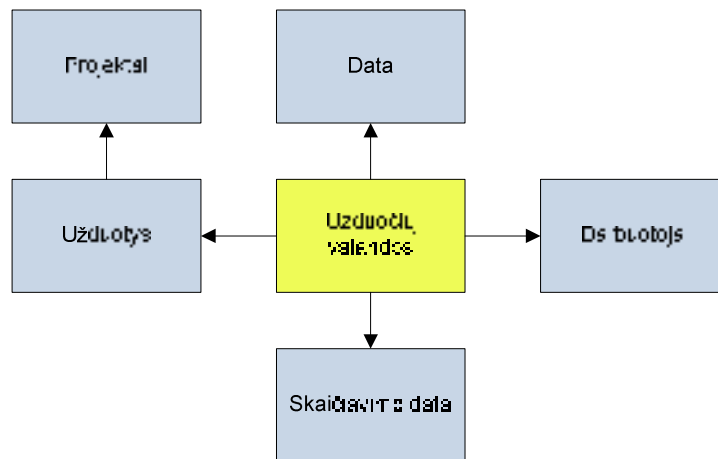
Remiantis 19 lentele suformuojami trys kubai:

- ✓ Pinigai\_Valandos
- ✓ Užduotys\_Valandos
- ✓ Išlaidos\_Mokymams

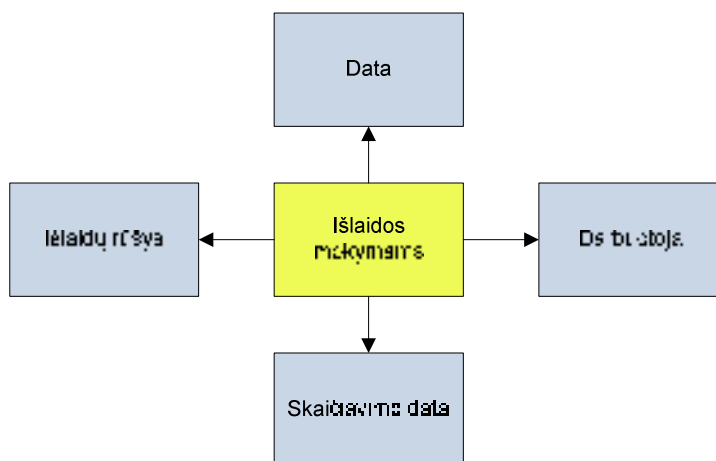
Schemose dimensijos pažymėtos mėlyna spalva, faktai – geltona.



19 pav. Kubo „Pinigai\_Valandos“ pirminė schema



20 pav. Kubo „Užduotys\_Valandos“ pirminė schema



21 pav. Kubo „Išlaidos\_Mokymams“ pirminė schema

Kubai gali naudoti tas pačias dimensijas (angl. *Shared Dimension*). Dimensijos, kurias naudoja keli kubai yra šios:

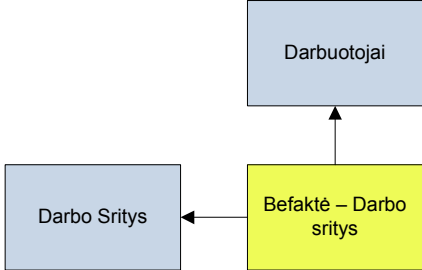
- ✓ Skaičiavimo data
- ✓ Data
- ✓ Darbuotojai
- ✓ Projektai

Sudarant pirmines kubų schemas, išskirti šie sudėtingi atvejai:

20 lentelė

Sudėtingų atvejų analizės lentelė

Nr.	Sudėtingas atvejis	Sudėtingo atvejo realizacija
1.	Šiukšlių dimensija (angl. <i>Junk Dimension</i> )	Dimensija „Tipas“. Ši dimensija naudojama SF arba užsakymo tipui filtruoti.
2.	Fakto dimensija (angl. <i>Degenerate, Fact Dimension</i> )	Dimensija „Užsak ir SF eilutės“. Ši dimensija, kuri suformuota iš fakto, leidžia filtruoti pagal užsakymų/SF numerius ir eilutes. Taip pat nuo vartotojo paslėptos ir naudojamos tik skaičiavimuose dimensijos: „Užduočių Valandos“ ir „Išlaidos mokymams“, kuriose yra vėlevėlės.
3.	Nepilna hierarchija (angl. <i>Ragged Hierarchy</i> )	Dimensija „Darbuotojai“. Fakto „Išlaidos Mokymams“ eilutėse darbuotojo dimensijos reikšmės yra tiek žemiausiame lygmenyje (darbuotojas) tiek aukštesniame lygmenyje (skyrius), dėl to dimensijos „Darbuotojai“ atributai tam tikruose lygiuose neturi reikšmių.
4.	Daug – su – daug ryšiai (angl. <i>Many-to-many Relationships</i> )	Tarp dimensijų „Darbuotojai“ ir „Darbo Sritys“. Darbuotojai gali turėti kelias specializacijas vienu metu ir tą pačią specializaciją gali turėti keli darbuotojai. Toks ryšys išskaidomas į vienas-su-daug ryšius papildomai sukuriant tarpinę be faktę lentelę (22 pav.).

Nr.	Sudėtingas atvejis	Sudėtingo atvejo realizacija																	
		 <p data-bbox="651 510 1481 548"><b>22 pav. Daug-su-daug ryšys tarp „Darbuotojai“ ir „Darbo Sritis“</b></p>																	
5.	<p data-bbox="296 566 579 745">Lėtai besikeičiančios dimensijos (angl. <i>Slowly Changing Dimensions - SCD</i>)</p>	<p data-bbox="651 566 1134 600">Dimensijos „Darbuotojai“ ir „Projektai“.</p> <p data-bbox="651 616 772 647"><b>21 lentelė</b></p> <p data-bbox="799 665 1334 698"><b>Lėtai besikeičiančių dimensijų detalizacija</b></p> <table border="1" data-bbox="718 714 1415 1077"> <thead> <tr> <th>Dimensija</th> <th>Stulpelis</th> <th>SCD tipas</th> </tr> </thead> <tbody> <tr> <td rowspan="5">Darbuotojai</td> <td>Skyrius</td> <td>2</td> </tr> <tr> <td>Skyrius*</td> <td>2</td> </tr> <tr> <td>Darbuotojo kodas</td> <td>1</td> </tr> <tr> <td>Pavardė</td> <td>1</td> </tr> <tr> <td>Pareigos</td> <td>2</td> </tr> <tr> <td>Projektai</td> <td>Projektų vadovo kodas</td> <td>2</td> </tr> </tbody> </table>	Dimensija	Stulpelis	SCD tipas	Darbuotojai	Skyrius	2	Skyrius*	2	Darbuotojo kodas	1	Pavardė	1	Pareigos	2	Projektai	Projektų vadovo kodas	2
Dimensija	Stulpelis	SCD tipas																	
Darbuotojai	Skyrius	2																	
	Skyrius*	2																	
	Darbuotojo kodas	1																	
	Pavardė	1																	
	Pareigos	2																	
Projektai	Projektų vadovo kodas	2																	
6.	<p data-bbox="296 1133 598 1216">Skirtingų reikšmių kiekis (angl. <i>Distinct Count</i>)</p>	<p data-bbox="651 1133 1485 1417">Šis matavimas architektūriškai išsiskiria iš kitų matavimų Analysis Services įrankyje. Dar AS 2000 versijoje buvo patariama Distinct Count matavimą atskirti į atskirą matavimų grupę (AS 2005 versijoje tai vyksta automatiškai), nes kubo apdorojimo metu atliekamas GROUP BY, kas ir lemia ilgesnį kubo apdorojimą. Taip pat ir kubo naršymo metu šis matavimas suskaičiuojamas realiu laiku.</p> <p data-bbox="651 1435 1485 1619">Jei dimensija SCD 2 tipo, būtina pakeisti Distinct Count matavimą panaudojant many-to-many ryšius, nes Distinct Count pateikia neteisingus rezultatus. Patariama tokį matavimą realizuoti ir naudojant MDX kalbą:</p> <pre data-bbox="683 1637 1305 1704">count (nonempty ([DimName] . [HierarchyName] . [LevelName] . members * [Measures] . [MyMeasure]))</pre> <p data-bbox="651 1722 1485 1906">Naudojant šį matavimą reikia turėti omenyje, kad AS NULL reikšmės paverčia į 0 ir skaičiuojant Distinct Count priskaičiuoja prie kitų skirtingų reikšmių. Jei eilučių skaičius particijoje viršija 20 mln., reikia kurti daugiau particijų.</p> <p data-bbox="651 1924 1485 2007">Buvo pasirinkta naudoti MDX kalbą pasitelkiant ir AS esančią Distinct Count realizaciją.</p>																	

### 4.3 Duomenų saugyklos sukūrimas

Su Visio nubraižyta daugiamatė schema pavaizduota 23 pav. Taip pat pateikiama ištrauka iš šios schemos sugeneruoto daugiamatės duomenų bazės sukūrimo skripto:

```
create database "MD_Datawarehouse"

go

use "MD_Datawarehouse"

/* Create new table "Fact_Dedication".
*/
/* "Fact_Dedication" : Table of Fact_Dedication
*/
create table "Fact_Dedication" (
    "fdd_ID" int identity not null,
    "fdd_dmp_ID" int null,
    "fdd_dct_ID" int null,
    "fdd_Calculation_dcl_ID" int null,
    "fdd_Dedication_Percent" float null)
go

alter table "Fact_Dedication"
    add constraint "PK_Fact_Occupation" primary key nonclustered ("fdd_ID")
go

/* Add foreign key constraints to table "Fact_Dedication".
*/
alter table "Fact_Dedication"
    add constraint "FK_Fact_Dedication_Dim_Activity" foreign key (
        "fdd_dct_ID")
        references "Dim_Activity" (
            "dct_ID")
go

alter table "Fact_Dedication"
    add constraint "FK_Fact_Dedication_Dim_Calculation" foreign key (
        "fdd_Calculation_dcl_ID")
        references "Dim_Calculation" (
            "dcl_ID") on update no action on delete no action
go

alter table "Fact_Dedication"
    add constraint "FK_Fact_Dedication_Dim_Employee" foreign key (
        "fdd_dmp_ID")
        references "Dim_Employee" (
            "dmp_ID")
go
```

Sukurtų vaizdų skriptas pateiktas antrame priede.

### 4.4 Kubo projektavimas

Vedliu suformavus kubus, peržiūrimi ryšiai tarp dimensijų ir matavimų grupių (24 – 26 pav.).

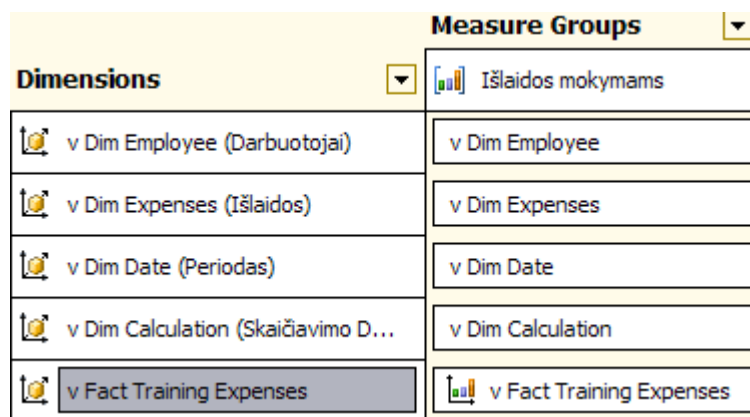


Measure Groups		▼	▼	▼	▼	▼	▼	▼	▼
Dimensions	▼	▼	▼	▼	▼	▼	▼	▼	▼
	▼	▼	▼	▼	▼	▼	▼	▼	▼
▼ Dim Employee (Darbuotojai)	▼ Dim Employee	▼ Factless Employee Fields	▼ Projektų įvertinimas	▼ Užsakymai ir SF	▼ Dedikacija	▼ Valandos			
▼ Dim Calculation (Skaičavimo Data)	▼ Dim Calculation	▼ Dim Employee	▼ Dim Calculation	▼ Dim Calculation	▼ Dim Employee	▼ Dim Employee			
▼ Dim Work Field (Darbo Sritis)	▼ Factless Employee Fields	▼ Dim Calculation	▼ Dim Project	▼ Dim Project	▼ Dim Calculation	▼ Dim Calculation			
▼ Dim Project (Projektai)									
▼ Dim Type (Tipas)									
▼ Dim Activity (Veiklos)									
▼ Dim Currency (Valiuta)									
▼ Dim Date (Periodas)									
▼ Fact Project Finance (Užsak ir SF ellu...)				▼ Fact Project Finance					

24 pav. Kubo „Pinigai\_Valandos“ dimensijų ir matavimų grupių ryšiai

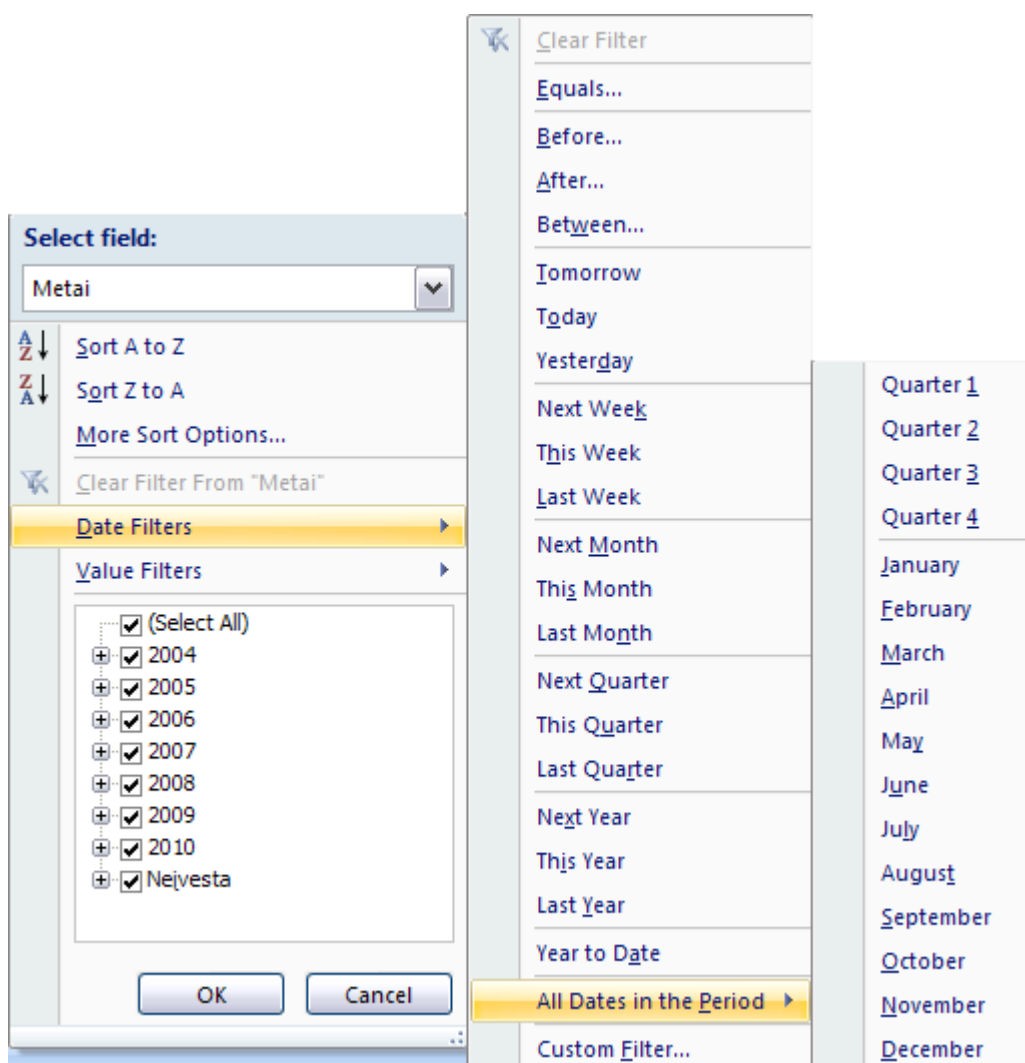
Measure Groups		▼	▼	▼	▼
Dimensions	▼	▼	▼	▼	▼
	▼	▼	▼	▼	▼
▼ Dim Project (Projektai)	▼ Uždavotys	▼ Uždavotys	▼ Uždavotys	▼ Uždavotys	▼ Uždavotys
▼ Dim Task (Uždavotys)	▼ Dim Task	▼ Dim Task	▼ Dim Task	▼ Dim Task	▼ Dim Task
▼ Dim Employee (Darbuotojai)	▼ Dim Employee	▼ Dim Employee	▼ Dim Employee	▼ Dim Employee	▼ Dim Employee
▼ Dim Date (Periodas)	▼ Dim Date	▼ Dim Date	▼ Dim Date	▼ Dim Date	▼ Dim Date
▼ Dim Calculation (Skaičavimo D...)	▼ Dim Calculation	▼ Dim Calculation	▼ Dim Calculation	▼ Dim Calculation	▼ Dim Calculation
▼ Fact Task Hours	▼ Fact Task Hours	▼ Fact Task Hours	▼ Fact Task Hours	▼ Fact Task Hours	▼ Fact Task Hours

25 pav. Kubo „Uždavotys\_Valandos“ dimensijų ir matavimų grupių ryšiai



26 pav. Kubo „Išlaidos\_Mokymams“ dimensijų ir matavimų grupių ryšiai

Taip pat svarbu datos dimensijai nurodyti „Time“ tipą. Tokio tipo dimensijoms Excel 2007 versijoje atsiranda papildomas datos filtras.






27 pav. Excel 2007 datos filtras „Time“ tipo dimensijai

Nustatyti šie dimensijų tipai:

22 lentelė

**Dimensijų tipai**

Dimensijos tipas	Dimensijos kube
Reguliari dimensija	Dim_Employee, Dim_Calculation, Dim_Date, Dim_Project (kubas „Pinigai_Valandos“), Dim_Type, Dim_Activity, Dim_Currency
Nuorodos dimensija 	Dim_Project (kubas „Užduotys_Valandos“)
Daug-su-daug dimensija 	Dim_Work_Field
Fakto dimensija 	Fact_Project_Finance, Fact_Task_Hours, Fact_Training_Expenses

Vieni matavimai aprašomi tiesiogiai naudojant tam tikrą matavimo funkciją, kiti – naudojant MDX (angl. *Multidimensional Expressions*) kalbą. Ši užklausų kalba skirta OLAP duomenų bazėms, panašiai kaip SQL – OLTP duomenų bazėms.

23 lentelė

**Kubų matavimai**

Matavimas	Formulė
<b>Kubas „Pinigai_Valandos“</b>	
Vykstančių projektų skaičius	Count(vFact_Project_Fulfilment.fpf_Project_Flag)
Projektų skaičius	Count(vFact_Project_Fulfilment (Row))
Suma	Sum(vFact_Project_Finance.Suma)
Išdirbtos valandos	Sum(vFact_Hours.Išdirbtos valandos)
Suplanuotos Valandos	Sum(vFact_Hours.Suplanuotos Valandos)
Dedikuotos Valandos	Sum(vFact_Hours.Dedikuotos Valandos)
[Išdirbtos Valandos Viso]	([Periodas].[Periodas - MKMD].Visi, [Measures].[Išdirbtos Valandos])
[Vidutinė Valandos Kaina]	<code>IIF</code> ((([Measures].[Išdirbtos Valandos] is null) or [Measures].[Išdirbtos Valandos]=0, [Measures].[SF Suma], [Measures].[SF Suma]/[Measures].[Išdirbtos Valandos])
[Laisvos Valandos]	[Measures].[Dedikuotos Valandos]-[Measures].[Suplanuotos Valandos]
[Faktinis Darbuotojų Skaičius]	<code>avg</code> ( <code>DESCENDANTS</code> ([Skaičiavimo Data].[Skaičiavimo data - LMKMD].currentmember, [Skaičiavimo Data].[Skaičiavimo data - LMKMD].[Data])* [Measures].[Faktinis Darbuotojų Skaičius1])



<b>Matavimas</b>	<b>Formulė</b>
[Darbuotojų Dedikacija]	<code>avg(Descendants([Darbuotojai].[Darbuotojas Hierarchija].CurrentMember,[Darbuotojai].[Darbuotojas Hierarchija].[Vardas Pavardė]), IIF([Measures].[Fdd Dedication Percent] is NULL, 0, [Measures].[Fdd Dedication Percent]))/100</code>
[PV Projektų Įvertinimas]	<code>avg(Descendants([Projektai].[Projekto Hierarchija].currentMember,[Projektai].[Projekto Hierarchija].[Projektas]), ([Measures].[Fpf Project Fullfilment Percent]/100))</code>
[SF Suma]	<code>([Tipas].[v Dim Type].&amp;[2],[Measures].[Suma])</code>
[SF Suma Viso]	<code>([Periodas].[Periodas - MKMD].Visi, [Measures].[SF suma])</code>
[SF Suma %]	<code>IIF([Measures].[Suma]&lt;&gt;0,[Measures].[SF Suma]/[Measures].[Suma], NULL)</code>
[Užsakymų Suma]	<code>([Tipas].[v Dim Type].&amp;[1],[Measures].[Suma])</code>
[Užsakymų Suma Viso]	<code>([Periodas].[Periodas - MKMD].Visi, [Measures].[Užsakymų Suma])</code>
[Užsakymų Suma %]	<code>IIF([Measures].[Suma]&lt;&gt;0,[Measures].[Užsakymų Suma]/[Measures].[Suma],NULL)</code>
[Suma Viso]	<code>([Periodas].[Periodas - MKMD].Visi, [Measures].[Suma])</code>
<b>Kubas „Užduotys_Valandos“</b>	
Išdirbtos valandos	<code>Sum(vFact_Task_Hours.Išdirbtos valandos)</code>
Suplanuotos Valandos	<code>Sum(vFact_Task_Hours.Suplanuotos Valandos)</code>
Vėlavimo dienos	<code>Sum(vFact_Task_Hours.Vėlavimo dienos)</code>
Užduočių Skaičius	<code>DistinctCount(vFact_Task_Hours.fth_dts_ID)</code>
[Vykstančių Užduočių Skaičius]	<code>distinctcount(Descendants([Užduotys].[Projektas - Užduotis].CurrentMember, [Užduotys].[Projektas - Užduotis].[Užduotis])*[v Fact Task Hours].[Fth Task Flag].&amp;[-1])</code>
<b>Kubas „Išlaidos_Mokymams“</b>	
Išlaidų Suma Mokymams	<code>Sum(vFact_Training_Expenses.Išlaidų suma mokymams)</code>
[Vidutinė Išlaidų Suma Darbuotojui]	<code>IIF([Measures].[Darbuotojų skaičius] is null or [Measures].[Darbuotojų skaičius]=0,0,[Measures].[Išlaidų Suma Mokymams]/[Measures].[Darbuotojų skaičius])</code>
[Darbuotojų skaičius]	<code>distinctcount(DESCENDANTS([Darbuotojai].[Darbuotojas Hierarchija].currentmember,[Darbuotojai].[Darbuotojas Hierarchija].[Vardas Pavardė])*([Periodas].[Periodas - MKMD].Visi,[v Fact Training Expenses].[Fte Flag].&amp;[1]))</code>

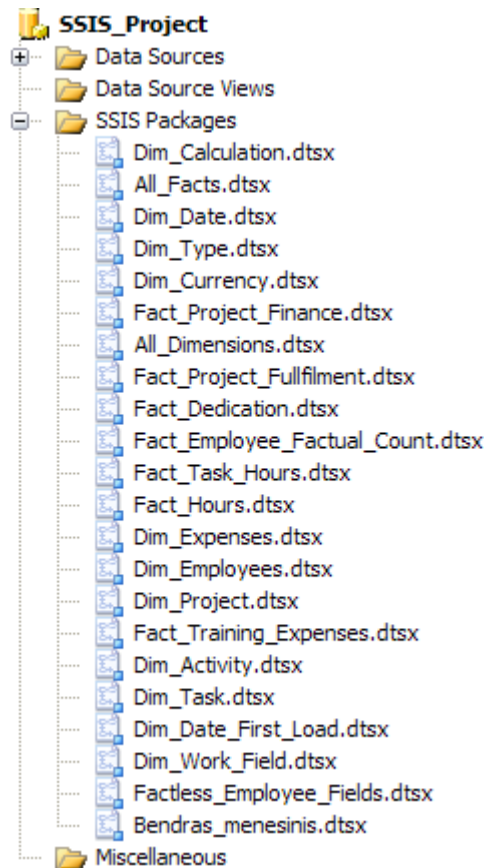
Sukurti šie veiklos rodikliai kube „Pinigai\_Valandos“ :

- ✓ Valandos kaina
- ✓ Užsakymų suma

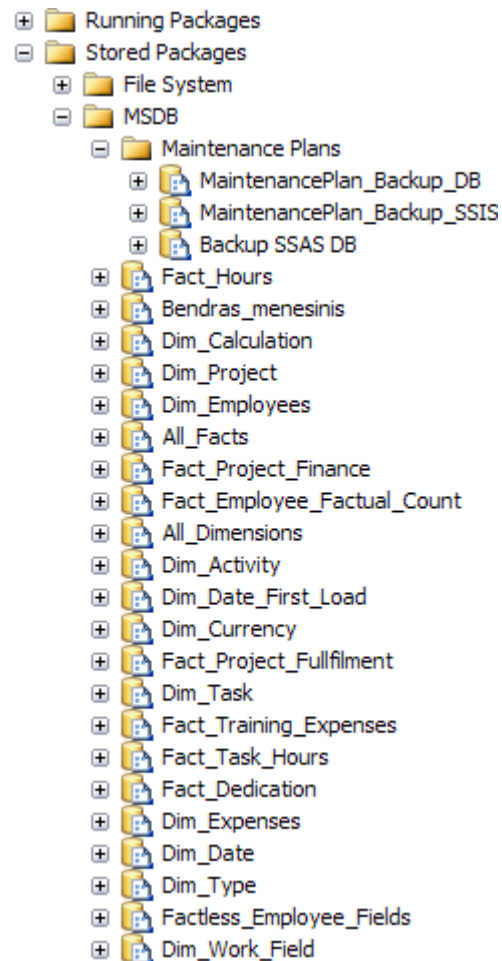
- ✓ SF suma
- ✓ SF ir užsakymų sumos

#### 4.5 Duomenų surinkimo, išvalymo ir įkrovimo proceso realizavimas

Duomenys paaimami, išvalomi, agreguojami ir užkraunami naudojant SSIS (angl. *SQL Server Integration Services*) įrankį. SSIS projektą BIDS aplinkoje (angl. *Business Intelligence Development Studio*) sudaro paketai (angl. *Packages*). Paketai instaliuojami į SQL Server Integration Serverį.

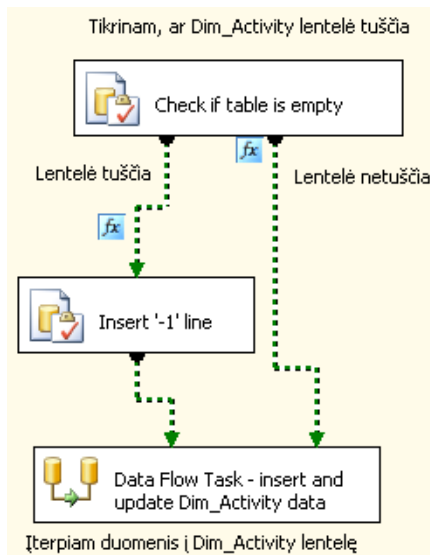


28 pav. SSIS projekto paketai BIDS aplinkoje

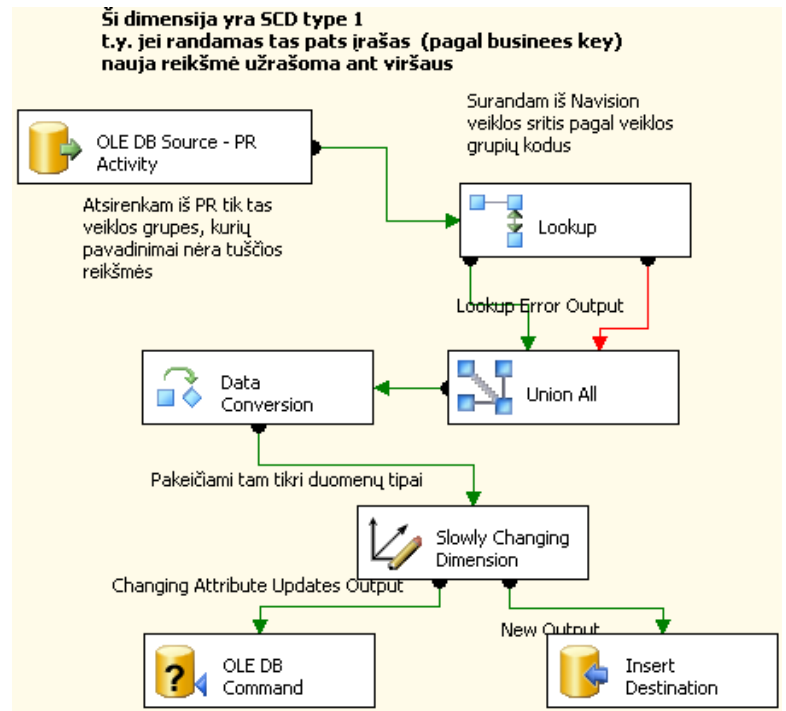


29 pav. SSIS projekto paketai instaliuoti IS serveryje

Pirmiausia kuriami dimensijų paketai – kiekvienai dimensijai po paketą. Pirmo krovimo metu užkraunama eilutė su identity „-1“ reikšme (toms faktų reikšmėms, kurių reikšmių nebus dimensijoje). „Data Flow“ lange atliekami detalesni duomenų apdorojimo veiksmai. Duomenys surenkami iš įvairių šaltinių, sutvarkomi duomenų tipai, pritaikomas SCD (angl. *Slowly Chaning Dimension*) elementas. Detalizuotas dimensijos Dim\_Activity paketas pavaizduotas 30-31 pav.

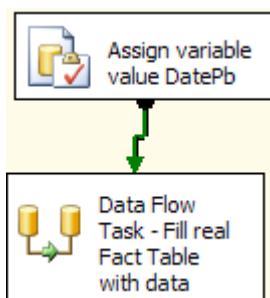


30 pav. Dim\_Activity paketo Control Flow langas

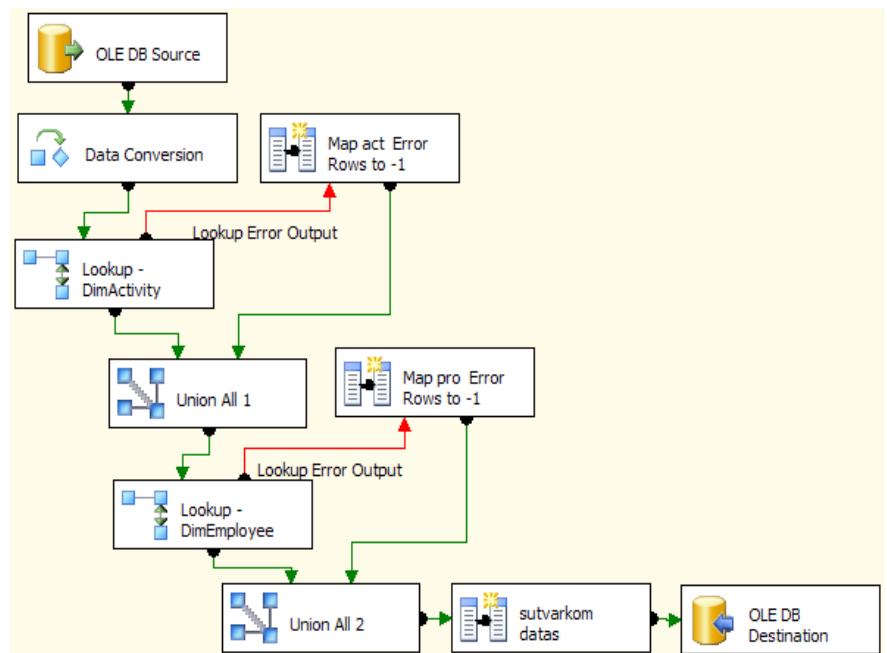


31 pav. Dim\_Activity paketo Data Flow langas

Suprojektavus dimensijų duomenų užkrovimo paketus, projektuojami faktų paketai. Rašoma pagrindinė užklausa visiems faktų duomenims surinkti, pakeičiami duomenų tipai. Ieškoma išorinio rakto reikšmė atitinkamoje dimensijoje. Reikšmėms, kurių rasti nepavyko, priskiriama „-1“ reikšmė. Detalizuotas fakto Fact\_Dedication paketas pavaizduotas 32-33 pav.



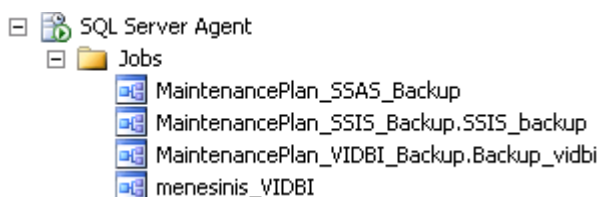
32 pav. Fact\_Dedication paketo Control Flow langas



33 pav. Fact\_Dedication paketo Data Flow langas



„SQL Server Agent Job“ yra naudojamas tam tikrų darbų automatiniam atlikimui. Šiame projekte sukurti keturi darbai :



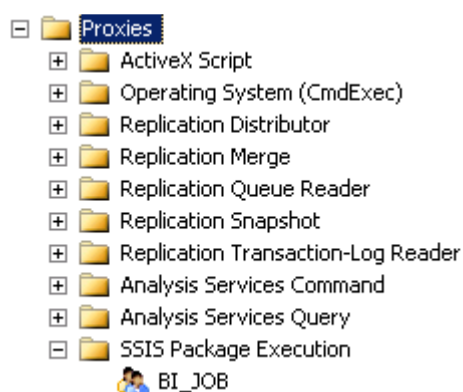
**37 pav. „SQL Server Agent“ darbai**

Pirmieji trys darbai kasdien sukuria po OLAP, MSDB ir duomenų saugyklų duomenų bazių kopiją (angl. *Backup*). Darbas „menesinis\_VIDBI“ atliekamas kiekvieno mėnesio pradžioje. Išskviečiamas pagrindinis paketas ir užkraunami duomenys į duomenų saugyklą.

Kadangi duomenims užkrauti paketai naudoja įvairius serverius, kad darbas įvyktų be klaidų, būtina sukurti vartotoją su teisėmis į PR, Navision ir duomenų saugyklos duomenų bases. Sukurtam vartotojui kuriamas „Credential“. Tada „SSIS Package Execution“ pakete sukuriamas „Proxy“ naudojant aprašytą kredencialą.



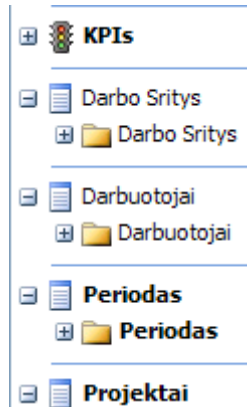
**38 pav. Sukurtas kredencialas**



**39 pav. Sukurtas pavaduotojas**

## 4.6 Ataskaitų sukūrimas

Excel – vienas paprasčiausių ir kartu daugiausiai galimybių turintis Microsoft BI įrankis. Nuo Excel 2007 versijos sukinio lentelė (angl. *PivotTable*) ypač atnaujinta. Vartotojas turi galimybę pats kurti ataskaitas pasirinkdamas dimensijų laukus, matavimus, KPI. Hierarchijos leidžia duomenis suskleisti/išskleisti. Visus atributus galima sudėlioti į paketus BIDS aplinkoje naudojant Analysis Services įrankį.

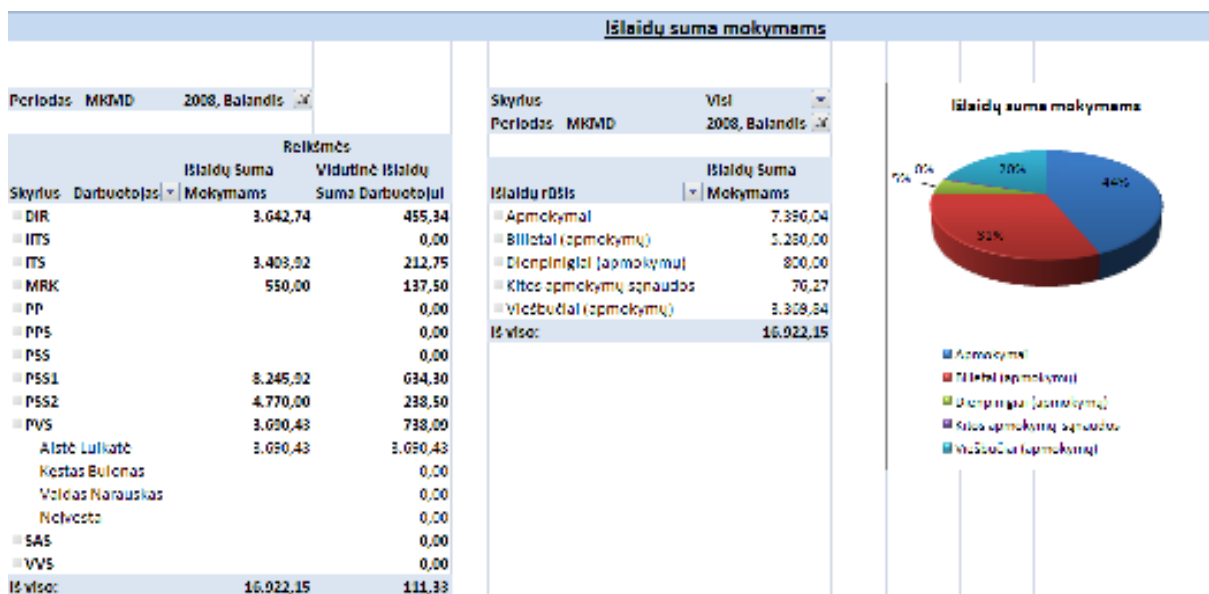


40 pav. Katalogai Pivot Table Field sąrašė

Taip pat veikia naršymas gilyn (angl. *drillthrough*). Pati aplinka tapo daug draugiškesnė vartotojui, yra galerija „Pivot Table“ spalvų šablonų. Filtruose galima rūšiuoti, nustatyti sąlygas (Contains, Greater Than, Between...). Atsiradus Excel Services, Excel Pivot Table ataskaitas galima publikuoti į Sharepoint kaip „Web parts“.

Buvo sukurta keturiolika Excel 2007 Pivot Table ataskaitų lapų, naudojančių kubą kaip duomenų šaltinį. Pateikiamos kelios tipinės ataskaitos.

„Išlaidų suma mokymams“ ataskaitoje pateikiamos šios Excel 2007 įrankio galimybės: hierarchijos suskleidimas-išskleidimas, dinamiškas „Pie“ tipo grafikas, filtrai.



41 pav. Excel 2007 Pivot Table „Išlaidų suma mokymams“ ataskaita

„Vidutinė valandos kaina“ ataskaitoje pateikiamos šios Excel 2007 įrankio galimybės: veiklos rodiklių atvaizdavimo galimybė, filtrai.

Vidutinė valandos kaina						
Periodas - MKMD	2008					
Veiklos Sritis	TSS					
Veiklos grupės	Dabartinė būseną	Tendencija	Vidutinė Valandos Kaina	Išdirbtos Valandos	SF Suma	
CRM diegimo darbai	●	→				
CRM palaikymo ir priežiūros darbai	●	↑	1600		1.600,00	
CRM sistemos paruošiamieji darbai, kuriuos atliks SAS	●	→				
Mobilumo sprendimai	●	↑	98011		98.011,00	
Mobilumo sprendimų priežiūra	●	↑	79280		79.280,00	
Navision diegimas	●	→				
Nenaudoti. Info paslauga	●	→				
Nenaudoti. M partner	●	→				0,00
Nenaudoti. Sistemos m-partner vartotojo licencijos	●	→				
Nenaudoti.CRM diegimo darbai	●	→				
Nenaudoti.CRM palaikymo ir priežiūros darbai	●	→				
Palaikymo ir techninės priežiūros darbai	●	→				
Programavimo darbai	●	↑	1111	2431	2.700.424,42	
Programinių sprend. palaikymo ir techninės priežiūros darbai	●	↑	1015081		1.015.081,20	
Projekto valdymas	●	→				
Projektų valdymo konsultavimas	●	→				
<b>Iš viso:</b>	●	↑	<b>1602</b>	<b>2431</b>	<b>3.894.396,62</b>	

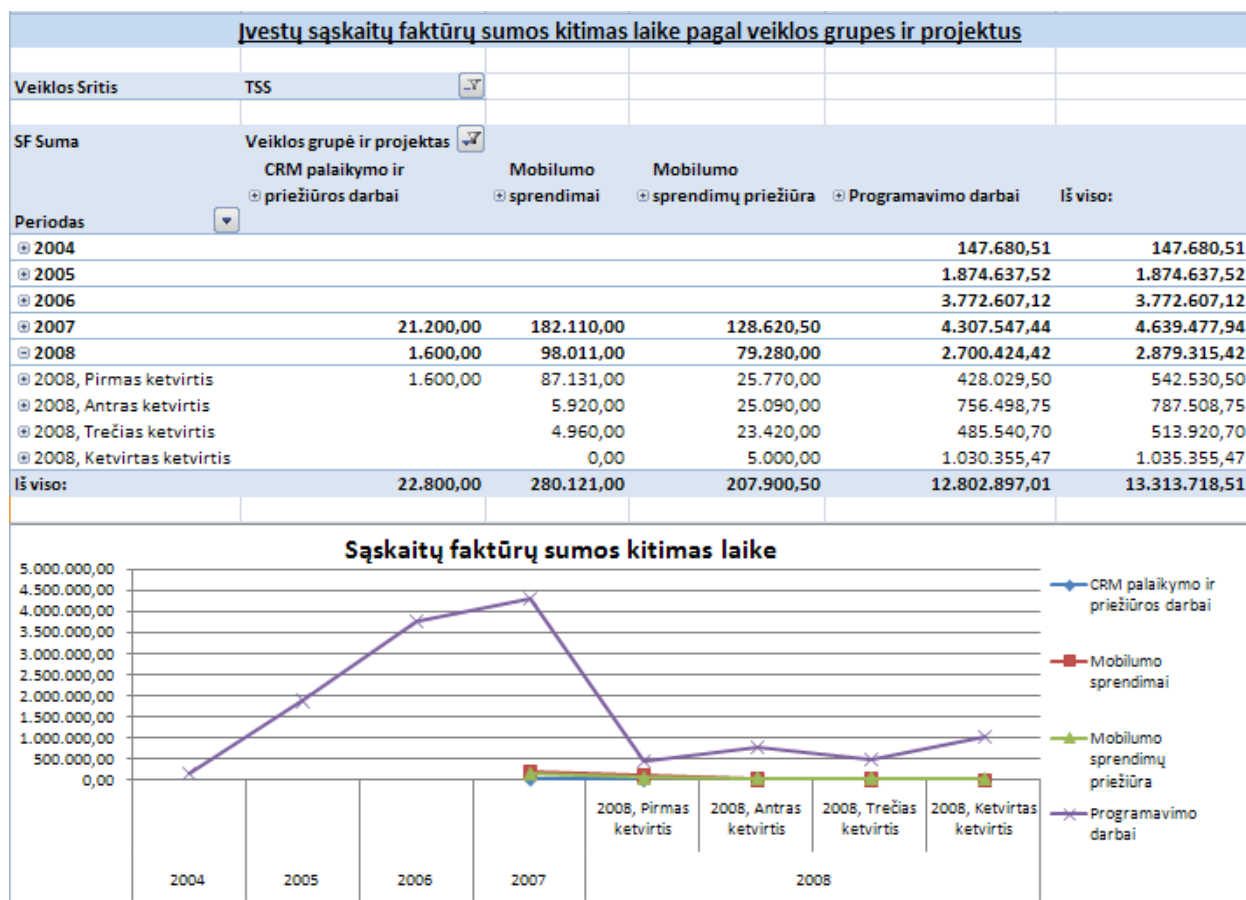
42 pav. Excel 2007 Pivot Table „Vidutinė valandos kaina“ ataskaita

„Projektų analizė“ ataskaitoje pateikiamos šios Excel 2007 įrankio galimybės: hierarchijos suskleidimas-išskleidimas, formatavimas pagal sąlygą (hierarchijos lygmenį), atributo ryšių vaizdavimas detaliaisiais įrašais (Projekto Būseną, Projekto Pavadinimas ir t.t.), hierarchijos laukų paslėpimas (Projekto Tipas), filtrai.

Projektų analizė						
Skaičiavimo data	2008, Spalis					
Reikšmės						
Projektų skaičius	Vykstančių projektų skaičius					
753	94					
Kliento Pavadinimas	(Multiple Items)					
Projekto Būseną	Vykdomas					
Periodas - MKMD	Visi					
Darbuotojas Hierarchija	Visi					
Veiklos Hierarchija	TSS					
Klientas - Projektas	Projekto Būseną	Projekto Pabaigos Data	Projekto Pavadinimas	Projekto Pradžios Data	PV Vardas Pavardė	Reikšmės
						SF Suma Užsakymų Suma Išdirbtos Valandos
☐ Klientas 24						67.896,00
Proj4	Vykdomas	Nejvesta	Proj4 pavadinimas	2001/11/15	Lilija Impalytė	67.896,00
☐ Klientas 35						19.680,00
☐ Klientas 58						112.240,00
Proj506	Vykdomas	2009/07/01	Proj506 pavadinimas	2006/11/01	Kornelija Butvilavičiūtė	112.240,00
☐ Klientas 63						57.602,15
Proj137	Vykdomas	Nejvesta	Proj137 pavadinimas	2003/01/27	Jeronimas Greičius	6.917,50
Proj254	Vykdomas	Nejvesta	Proj254 pavadinimas	2004/06/22	Vilius Malaiška	476,00
Proj434	Vykdomas	Nejvesta	Proj434 pavadinimas	2006/05/09	Vilius Malaiška	2.513,00
Proj606	Vykdomas	2008/10/31	Proj606 pavadinimas	2007/05/24	Jadvyga Mikliajeva	3.662,50
Proj752	Vykdomas	Nejvesta	Proj752 pavadinimas	2008/03/19	Motiejus Lesickas	12.092,50
Proj77	Vykdomas	Nejvesta	Proj77 pavadinimas	2002/09/05	Donata Mašalienė	29.760,40
Proj781	Vykdomas	2008/08/12	Proj781 pavadinimas	2008/06/12	Jadvyga Mikliajeva	1.675,75
Proj763	Vykdomas	Nejvesta	Proj763 pavadinimas	2008/04/01	Jadvyga Mikliajeva	504,50
☐ Klientas 86						24.500,00
<b>Iš viso:</b>						214.022,15 67.896,00 1720

43 pav. Excel 2007 Pivot Table „Projektų analizė“ ataskaita

„Įvestų sąskaitų faktūrų sumos kitimas laike pagal veiklos grupes ir projektus“ ataskaitoje pateikiamos šios Excel 2007 įrankio galimybės: hierarchijos suskleidimas-išskleidimas matricos tipo sukinio lentelėje, dinamiškas „Line“ tipo grafikas, filtrai.



44 pav. Excel 2007 Pivot Table „Įvestų sąskaitų faktūrų sumos kitimas laike pagal veiklos grupes ir projektus“ ataskaita

#### 4.7 Testavimas

Verslo analitikos projektų testavimas atliekamas keliais etapais:

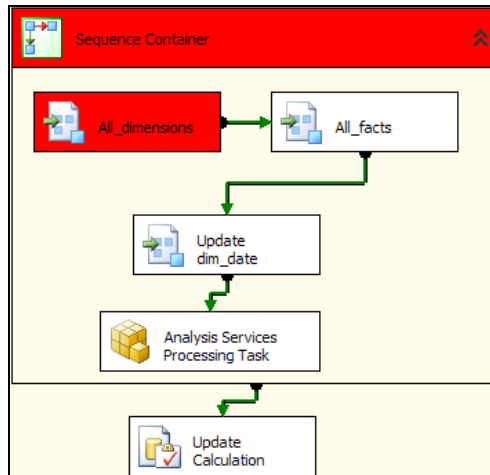
1. Integration Services paketų testavimas atliekamas tiek kiekvieno elemento lygmenyje, tiek kviečiant paketus, instaliuotus SQL Integration Services serveryje.

Taip pat testavimas gali būti atliekamas tiek iš viršaus į apačią (nuo stambiausių paketų link konkrečių elementų), tiek ir iš apačios į viršų (nuo smulkesnių elementų link stambių paketų).

SSIS projekto testavimo iš viršaus į apačią pavyzdys (toks testavimas būdingas tuomet, kai paleidus automatinį darbą jo metu įvyksta aiškiai nenurodyta klaida):

- ✓ Paleidžiame pagrindinį paketą „Bendras\_menesinis“

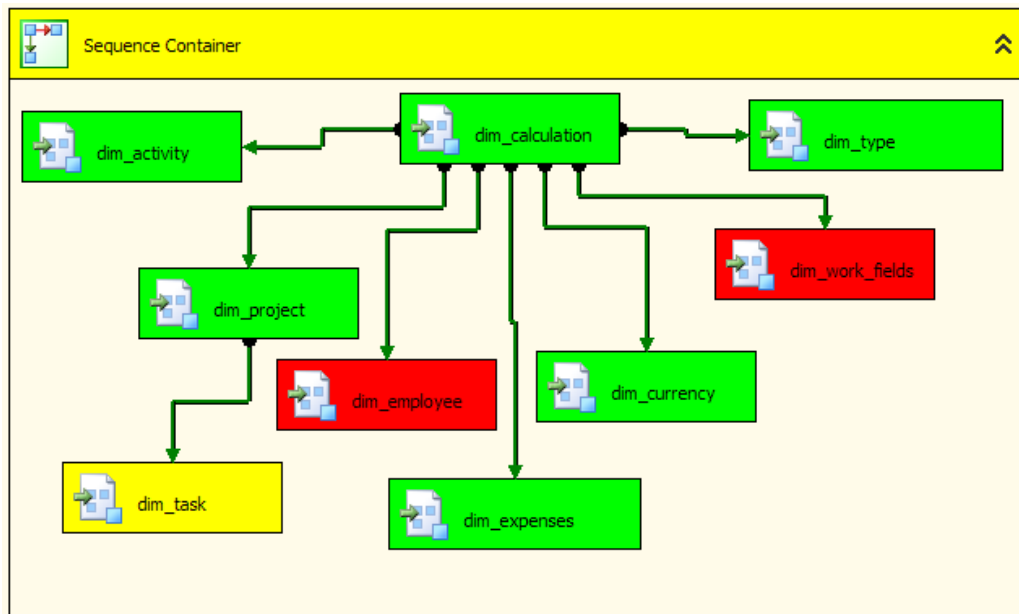




45 pav. Pagrindinio paketo paleidimas

(raudona spalva – klaida, geltona spalva – dar vykdomas, žalia spalva – sėkmingai įvykdytas)

✓ Paleidžiame paketą „All\_Dimensions“



46 pav. „All\_Dimensions“ paketo paleidimas

✓ Paleidžiame paketus: „Dim\_Employee“ bei „Dim\_Work\_Fields“ ir testuojame konkrečius paketus bei jų elementus.

2. Užkrautų duomenų saugykloje testavimas rašant SQL užklausas.

Kita vertus, tiesiogiai testuoti fakto duomenis nėra patogu, kadangi duomenis reikia sieti su dimensijų duomenimis, agreguoti žemiausio lygmens matavimus.

```

41 use MD_Datawarehouse
42 select * from Fact_Hours
43 where fhr_dmp_ID = 178
44 and fhr_Calculation_dcl_ID = 20081031
45 order by fhr_Calculation_dcl_ID, fhr_Task_ddt_ID

```

	fhr_ID	fhr_dct_ID	fhr_dmp_ID	fhr_dpr_ID	fhr_Task_ddt_ID	fhr_Calculation_dcl_ID	fhr_Used_Hours	fhr_Planned_Hours	fhr_Dedication_Hours
1	2520	-1	178	339	20081031	20081031	59	103,5	18,4
2	2521	-1	178	505	20081031	20081031	1	1	18,4
3	2522	-1	178	537	20081031	20081031	3,5	3,5	18,4
4	2523	-1	178	560	20081031	20081031	24	72	18,4
5	2524	-1	178	562	20081031	20081031	0,5	0,5	18,4
6	1210	-1	178	466	20081031	20081031	10	15	18,4
7	1211	-1	178	541	20081031	20081031	54	33	18,4
8	1212	-1	178	570	20081031	20081031	2	3	18,4
9	5218	-1	178	344	20081031	20081031	8	22	18,4
10	5219	-1	178	551	20081031	20081031	49,2	44,2	18,4
11	5220	-1	178	339	20081130	20081031	63	99	15,2
12	5221	-1	178	344	20081130	20081031	4,5	18	15,2
13	5222	-1	178	365	20081130	20081031	32	32	15,2
14	5223	-1	178	537	20081130	20081031	2,5	4	15,2

47 pav. SQL užklauso pavyzdys fakto Fact\_Hours testavimui

3. Kubo pjaustymas įvairiais pjūviais kubų sukūrimo įrankio aplinkoje.

Testavimas atliekamas tiek matavimams iš duomenų saugyklos, tiek naujai sukurtiems matavimams. Svarbu atkreipti dėmesį į sudėtingesnius matavimų realizavimo atvejus, nes sukurti elementai neretai turi specifinių savybių, kurios išryškėja tik detalaus duomenų testavimo metu (pvz. Distinct Count matavimas, Many-to-Many ryšys ir kita.) Taip pat labai svarbu apibrėžti, kuriuos matavimus kokiose dimensijose galima žiūrėti, kad apsaugoti klientus nuo nekorektiškų rezultatų.

Many-to-many ryšio tarp fakto Dedikacija ir dimensijos Darbo Sritys testavimo pavyzdys :

- ✓ Užkrauti duomenys peržiūrimi Analysis Services aplinkoje:

Skyrius	Vardas Pavardė	Darbo Sritis	Darbuotojų Dedikacija
[-] PSS1	[+] Donata Mašalienė		0.00%
	[+] Haroldas Sadauskas		0.00%
	[+] Haroldas Zulonas		100.00%
	[+] Henrikas Peldžius		0.00%
	[+] Ignotas Barinovas		100.00%
	[+] Jadvyga Mikliajeva		100.00%
	[+] Jeronimas Greičius		100.00%
	[+] Kostas Nedzvedskis		100.00%
	[+] Kotryna Galginaitė		50.00%
	[+] Kotryna Platovienė		100.00%
	[+] Laurynas Grigaliūnas	.NET Architektas	100.00%
		Architektas	100.00%
		BI Analitikas	100.00%
		C# Programuotojas	100.00%
		DB Administratorius	100.00%
		IT Inžinierius	100.00%
		IT Konsultantas	100.00%
		Java Programuotojas	100.00%
		Oracle Administratorius	100.00%
		PHP Programuotojas	100.00%
		Projekto Vadovas	100.00%
		Sharepoint Programuotojas	100.00%
		Sistemų Analitikas	100.00%
		SQL Server Administratorius	100.00%
		Testuotojas	100.00%
		Tinklų Administratorius	100.00%
		Web Programuotojas	100.00%
		Neįvesta	100.00%
		Total	100.00%
	[+] Laurynas Lukošius		100.00%

48 pav. Pradinio kubo duomenų peržiūrėjimo langas

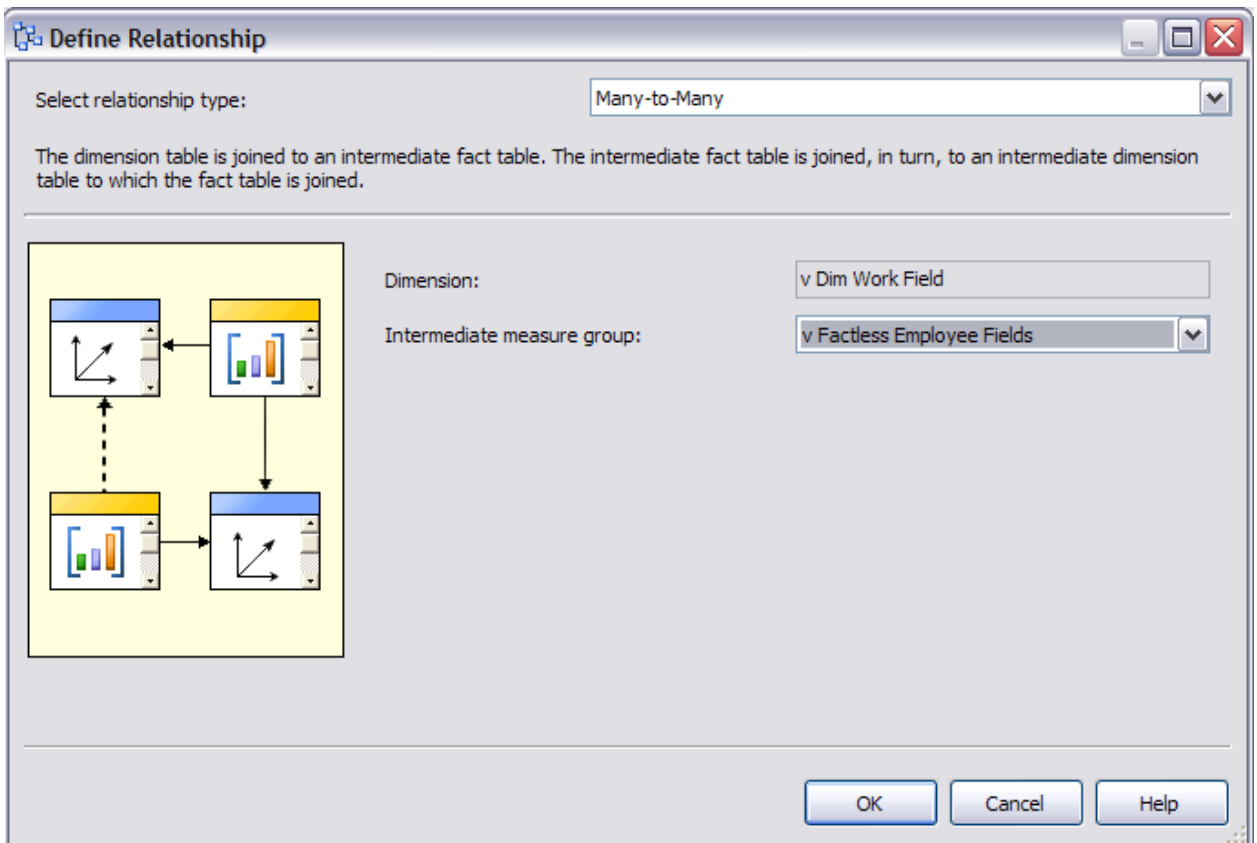
Nenurodžius many-to-many ryšio, kiekvienam darbuotojui priskiriamos visos dimensijoje Darbo Sritys esančios reikšmės.

- ✓ Nurodomas many-to-many tipo ryšys. Visi ryšiai tarp faktų ir dimensijų apibrėžiami „Dimension Usage“ lange.

Measure Groups		
Dimensions	Užsakymai ir SF	Dedikacija
v Dim Employee (Darbuotojai)		v Dim Employee
v Dim Calculation (Skaičiavimo Data)	v Dim Calculation	v Dim Calculation
v Dim Work Field (Darbo Sritys)		
v Dim Project (Projektai)	v Dim Project	

49 pav. Dimensijų ir matavimo grupių matrica

Iš pateikto ryšių sąrašo, parenkamas many-to-many ryšys, nurodant tarpinę befaktę lentelę kaip ryšį tarp fakto ir dimensijos.



50 pav. Many-to-many ryšio apibrėžimo langas

- ✓ Kubas apdorojamas, ir peržiūrėti duomenys:

Skryrius	Vardas Pavardė	Darbo Sritis	Darbuotojų Dedikacija
[-] PSS1	+ Donata Mašalienė		0.00%
	+ Haroldas Sadauskas		0.00%
	+ Haroldas Zulonas		100.00%
	+ Henrikas Peldžius		0.00%
	+ Ignotas Barinovas		100.00%
	+ Jadvyga Miklajeva		100.00%
	+ Jeronimas Greičius		100.00%
	+ Kostas Nedzvedskis		100.00%
	+ Kotryna Galginaitė		50.00%
	+ Kotryna Platovienė		100.00%
	+ Laurynas Grigaliūnas	IT Konsultantas	100.00%
		Projekto Vadovas	100.00%
		Total	100.00%
	+ Laurynas Lukošius		100.00%
	+ Matas Mameniškis		0.00%
	+ Morkus Bagdonas		50.00%
	+ Motiejus Lesickas		100.00%
	+ Petras Karaliūnas		100.00%
	+ Rudolfas Mažionis		100.00%
	+ Rudolfas Šimelis		100.00%
	+ Rudolfas Vyzas		100.00%
	+ Vilius Malaiška		100.00%

Pav. 51 Kubo peržiūros langas apibrėžus daug-su-daug ryšį

4. Duomenų tikrinimas kliento įrankyje.
5. Užtikrinti, kad užklausų teisingi rezultatai sutaptų duomenų saugykloje, kubo ir kliento įrankio aplinkose.
6. Užtikrinti, kad realizuota sistema atitinka vartotojo reikalavimus.

## 5 Metodo įvertinimas

Remiantis sudarytu metodu, buvo realizuota realiai veikianti sistema, kurioje gali būti atliekama valandų, užsakymų ir sąskaitų faktūrų sumų, dedikacijos procentų, projektų įvertinimo, išlaidų analizė atitinkamai pagal projektus, darbuotojus ir veiklas laiko ašyje. Vartotojai gali naršyti gilyn, filtruoti sudarytas šablonines ataskaitas bei kurti savo ataskaitas realiu laiku. Naujos ataskaitos formuojamos atidedant sukurtas hierarchijas ar atskirus dimensijų atributus į eilutes, stulpelius ar filtro zoną, veiklos stebėjimo rodiklius atidedant į eilutes ir matavimus atidedant į reikšmių zoną.

Metodo vertinimas atliktas lyginant savo metodą su kitų trijų autorių metodais šiame darbe siekiamų kriterijų atžvilgiu, t. y. išskiriant svarbiausias duomenų saugyklų projektavimo savybes ir sudėtingų atvejų sprendimus.

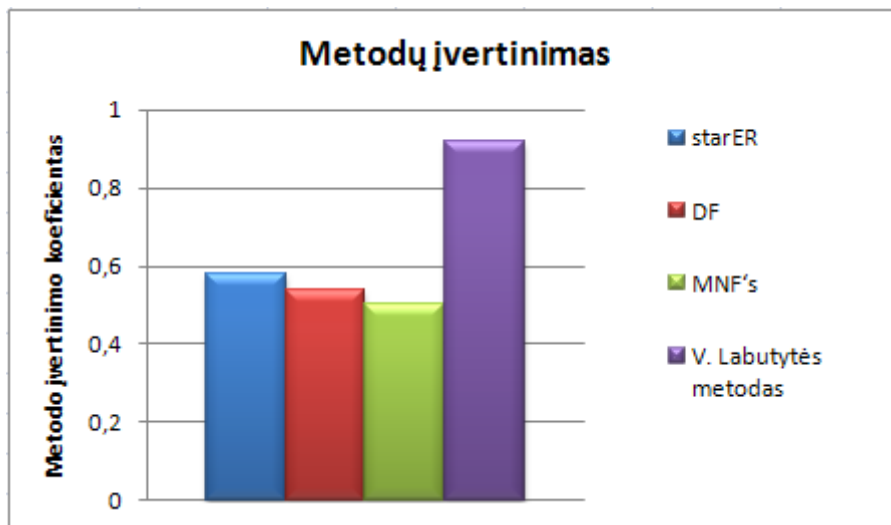
24 lentelė

Metodo įvertinimas

Savybės	Metodas				Max tšk.
	starER [22]	DF [11]	MNF's [6]	V. Labutytės metodas	
Vartotojo reikalavimų analizė	-	-	1	1	1
Galimybė atributus naudoti kaip faktus/dimensijų atributus	1	1	-	1	1
Specializacijos ir apibendrinimo ryšiai	1	-	1	-	1
Hierarchijų apibrėžimas, jų lygiai	1 (membership ryšys)	1	1	1	1
Sudėtinės hierarchijos (angl. <i>Multiple hierarchies</i> )	1	1	-	1	1
Negriežtos hierarchijos (angl. <i>non-strict, network hierarchies</i> )	1	-	-	1	1
Nesubalansuotos hierarchijos (angl. <i>unbalanced hierarchies</i> )	-	-		1	1
Nepilnos hierarchijos (angl. <i>ragged hierarchies</i> )	-	-		1	1
Dimensijų suskirstymas	1	-	-	1	1
Lėtai besikeičiančios dimensijos	-	-	-	1	
Many-to-many ryšys tarp faktų ir dimensijų	1	-	-	1	1
Faktų detalumo lygmens išskyrimas (angl. <i>granularity</i> )	-	1	-	1	1
Faktų skirstymas	-	-	-	1	1
Fakto dimensijos (angl. <i>degenerate/fact dimensions</i> )	-	1	-	1	1
Matavimų tipai	1	1	1	1	1
Matavimų išskyrimas, formulės	-	-	-	1	1

Savybės	Metodas				Max tšk.
	starER [22]	DF [11]	MNF's [6]	V. Labutytės metodas	
Distinct Count matavimas	-	-	-	1	1
Metodas nepriklauso nuo konkrečios technologijos	1	1	1	1	1
Metodo paprastumas, suprantamumas	4	5	2	5	5
Teisingumo užtikrinimas, testavimas	1	1	5	4	5
<b>Santykinis įvertinimas:</b>	<b>0,58 (14/24)</b>	<b>0,54(13/24)</b>	<b>0,50(12/24)</b>	<b>0,92(22/24)</b>	<b>24</b>

Metodų santykinio įvertinimo koeficientai palyginti stulpelinėje diagramoje 52 paveiksle.



**Pav. 52 Metodo įvertinimo stulpelinė diagrama**

StarER metodas apima ganėtinai nemažai atvejų, tačiau nėra detalaus perėjimo iki daugiamačio modelio. Taip pat neakcentuojami vartotojo reikalavimai, modelio teisingumas. Pats StarER modelis tampa pernelyg apkrautas, norint jame vaizduoti daugiau faktų, dimensijų, nes jame įtrauktas sudėtingų atvejų, hierarchijų, fakto detalumo vaizdavimas.

DF modelis pakankamai paprastas ir aiškus, tačiau šis metodas nenagrinėja daugybės sudėtingų atvejų, vienas jų – daug-su-daug ryšys.

Metodas, kuris remiasi vartotojo reikalavimais ir norminėmis formomis, yra pakankamai sudėtingas ir vien tokio metodo supratimas užima daug laiko. Be to, šis metodas nėra išbaigtas, neapima daug sudėtingų atvejų, įveda tik tam metodui būdingus stereotipus (pvz. hierarchijos lygmuo vadinamas baze (angl. *Base*)).

Šiame darbe sudarytas metodas neapima specializavimo/apibendrinimo ryšių, kadangi jie tiesiogiai neatsispindi duomenų bazių schemose. Būdingos faktų suskirstymo reikšmės paprastai nurodomos šiukšlių dimensijose (angl. *Junk Dimension*). Jei yra būtinybė, būdingas dimensijų atributų reikšmės galima pateikti lentelėje ar tiesiog išvardinti.

## 6 Išvados

1. Daugiamačių duomenų struktūrų projektavimo metodų analizė parodė, kad literatūroje aprašyti daugiamačių struktūrų modeliavimo metodai skirti nesudėtingiems atvejams ir jų nepakanka realioms sistemoms modeliuoti. Taip pat dauguma sudarytų metodų neaprašo perėjimo prie daugiamačio modelio proceso.
2. Šiame darbe identifikuoti sudėtingi saugyklų duomenų struktūrų projektavimo atvejai ir sudarytas metodas, kuris apima visą saugyklų kūrimo procesą:
  - ✓ įvertina vartotojo reikalavimus;
  - ✓ parodo, kaip pereiti nuo ER diagramos prie daugiamačio modelio;
  - ✓ pateikia sprendimus sudėtingų duomenų struktūrų atvejams;
  - ✓ apibrėžia žingsnius kubams sukurti;
  - ✓ nurodo, kaip parengti ETL procesą.
3. Taikant šį metodą praktiniam uždaviniui, atlikta dviejų technologijų (MS SQL Server ir Oracle) analizė ir pasirinkta MS SQL Server technologija dėl didesnio saugumo, mažesnės kainos, integracijos su kūrimo įrankiais (Visual Studio, Microsoft .NET) ir biuro įranga (Microsoft Office Excel), taip pat dėl išleistos naujosios Excel 2007 versijos, kuri dar labiau pritaikyta verslo analitikai.
4. Realios sistemos realizavimo metu paaiškėjo, kad pasirinkta technologija turi plačias funkcionalumo galimybes, tačiau turi nemažai trūkumų:
  - ✓ Analysis Services įrankyje nėra galimybės į hierarchiją įtraukti skirtingų dimensijų atributų;
  - ✓ Integration Services įrankyje realizuotas lėtai besikeičiančių dimensijų (SCD) elementas yra labai lėtas (pvz., 20 tūkst. eilučių apdorojama per ~18 min.), todėl esant daugiau duomenų reikia rinktis kitoki lėtai besikeičiančių dimensijų realizavimo variantą;
  - ✓ nėra aiškios dokumentacijos, ypač dėl integravimo paslaugų, SSIS paketų instaliavimo ir paleidimo;
  - ✓ SQL Server verslo analitikos (BIDS arba Visual Studio) aplinkos įrankiai naudoja nemažai resursų, todėl mažam projektui reikia minimaliai bent 2GB RAM atminties kompiuterio.
5. Remiantis sudarytu metodu, realizuota veiklos analitikos sistema. Sukurtos sudėtingos analitinės Excel 2007 ataskaitos, kuriomis sėkmingai naudojasi suinteresuoti vartotojai, o tai rodo, kad vartotojų reikalavimų surinkimo ir analizės metodika buvo parinkta teisingai.

6. Sistemos testavimas, kuris buvo atliekamas kiekviename metodo taikymo žingsnyje, patvirtino, kad taikant sukurtus sprendimus matavimų agregavimas vyksta teisingai.
7. Realios sistemos kūrimas taip pat parodė, kad kuriant veiklos analizės sistemas daugiausia sunkumų sukelia nestandartinės aplinkybės, kurioms reikia individualių sprendimų. Šiuo atveju tai buvo dviejų duomenų šaltinių integravimas ir dirbtinis žemesnio lygmens faktų skaičiavimas.
8. Realizuojant sistemą, pritaikyti ir išbandyti šie sudėtingų atvejų sprendimai, kurių visų neapima kitų autorių sukurti metodai:
  - ✓ daug-su-daug ryšiai (angl. *Many-to-many Relationships*);
  - ✓ skirtingų reikšmių skaičius (angl. *Distinct Count*);
  - ✓ šiukšlių dimensija (angl. *Junk Dimension*);
  - ✓ išsigimusi (fakto) dimensija (angl. *Degenerate Dimension*);
  - ✓ nepilna hierarchija (angl. *Ragged Hierarchy*);
  - ✓ lėtai besikeičiančios dimensijos (angl. *Slowly Changing Dimensions - SCD*).
9. Atlikta teorinė ir praktinė analizė taip pat leidžia teigti, kad dėl greitesnio sistemos veikimo ir dėl galimybės ateityje lengviau pakeisti modelį reikėtų stengtis, jei įmanoma, išvengti schemą apsunkinančių ryšių (kelių dimensijų jungimo, daug – su – daug ryšių) ir daugiausia naudoti žvaigždės schemą.
10. Darbo tema atspausdintas straipsnis ir perskaitytas pranešimas konferencijoje „Informacinės technologijos '08“.



## 7 Literatūra

- [1] **D.L. Moody, M.A.R. Kortnik.** From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. *Proceedings of the International Workshop on Design and Management of Data Warehouse (DMDW'2000)*, Stockholm, Sweden, June 5 – 6, 2000. p. 5-1 – 5-12.
- [2] **E.F. Codd, S.B. Codd, C.T. Salley.** Providing OLAP (On – line Analytical Processing) to User – Analysts: An IT Mandate. 1993
- [3] **A. Rozeva.** Dimensional Hierarchies – Implementation in Data Warehouse Logical Scheme Design. *International Conference on Computer Systems and Technologies - CompSysTech'07.*, 2007. p. 6.
- [4] **E. Pourabbas, M. Rafanelli.** Hierarchies and Relative Operators in the OLAP Environment, *Journal of ACM SIGMOD Record 29(1): 32-37*, 2000. p. 7.
- [5] **S. Mansmann, M.H. Scholl.** Extending Visual OLAP for handling irregular dimensional hierarchies. *DaWaK 2006, Springer – Verlag*, Heidelberg, Berlin, 2006, nr. 4081 in LNCS, p. 95 – 105.
- [6] **J.N. Mazo´n, J. Trujillo, J. Lechtenbo¨rger.** Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data & Knowledge Engineering 63 (2007) 725–751*. Received 13 April 2007; accepted 16 April 2007, p.725-751.
- [7] **M. Ruso.** The many-to-many revolution. *Advanced dimensional modeling with SQL Server Analysis Services 2005*, Version 1.0 Revision 93 – September 25, 2006. p. 88, prieiga per internetą:  
<http://www.sqlbi.eu/Portals/0/Downloads/M2M%20Revolution%201.0.93.pdf>
- [8] **P. Giorgini, S.Rizzi, M. Garzetti.** Goal-Oriented Requirement Analysis for Data Warehouse Design. *Proceedings 8th International Workshop on Data Warehousing and OLAP (DOLAP 2005)*, Bremen, Germany, 2005. p. 45-56.
- [9] **J. Schiefer, B. List, R. M. Bruckner.** A Holistic Approach for Managing Requirements of Data Warehouse Systems. *8th Americas Conf. on Information Systems*, 2002. p.11.

- [10] **M. Golfarelli, S. Rizzi.** Designing the data Warehouse: Key Steps and Crucial Issues. *Journal of Computer Science and Information Management*, 2(3), 1999. p.11.
- [11] **M. Golfarelli, D. Maio, S. Rizzi.** Conceptual Design of Data Warehouses from E/R Schemes, *International Journal of Cooperative Information Systems*, 1998. p. 10.
- [12] **S. Lujñn-Mora, P. Vassiliadis, J. Trujillo.** Data Mapping Diagrams for Data Warehouse Design with UML, *23rd International Conference on Conceptual Modeling*, 2004. p. 14
- [13] **N. Pendse.** The origins of today's OLAP products, *DSSResources.COM* on October 6, 2002, prieiga per internetą: <http://dssresources.com/papers/features/pendse10062002.html>
- [14] **D. J. Power.** A Brief History of Decision Support Systems. *DSSResources.COM version 2.8*, May 31, 2003Prieiga per internetą: <http://dssresources.com/history/dsshistoryv28.html>
- [15] **J. Lechtenbo"rger.** Data warehouse schema design, *In Proc. of the 10th BTW*, 2003. p. 10
- [16] **S. Rizzi.** Conceptual modeling solutions for the Data Warehouse, *Idea Group Publishing*, 2006. p. 26
- [17] **A. Abell'ó, J. Samos, F. Saltor.** Understanding Analysis Dimensions in a Multidimensional Object-Oriented Model, *3<sup>rd</sup> International Workshop on Design and Management on Data Warehouses (DWDM)*. *SwissLife*, 2001. p. 9
- [18] **W. Hümmer, W. Lehner, A. Bauer, L. Schlesinger.** A Decathlon in Multidimensional Modeling: Open Issues and Some Solutions, *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, September 04-06, 2002. p. 10
- [19] **R. Kimball, M.Ross.** Data Warehouse Toolkit (Second Edition). *The Complete Guide to Dimensional Modeling*, 2002. p. 421
- [20] **C. Jensen, A. Klygis, T. Pedersen, I. Timko.** Multidimensional data modeling for location-based services. *VLDB Journal*, 2004. p. 21
- [21] **I.-Y. Song, W. Rowen, C. Medsker, E. Ewen.** An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling.

*Proceedings of the Int. Workshop on Design and Management of Data Warehouses (DMDW 2001), Interlaken, 2001. p. 6-5 - 6-13.*

- [22] **N. Tryfona, F. Busborg, J. G. B. Christiansen.** starER: A Conceptual Model for Data Warehouse Design. *In Proc. of ACM 2nd Int. Workshop on Data Warehousing and OLAP (DOLAP), 1999. p.6*

## 8 Terminų ir santrumpų žodynas

- ✓ **SQL** – (**Structured Query Language**) – struktūrizuota užklausų kalba; populiariausia iš šiuo metu naudojamų kalbų, skirtų aprašyti duomenis ir manipuluoti jais reliacinių duomenų bazių valdymo sistemose.
- ✓ **OLAP** – (**Online Analytical Processing**) - šis terminas naudojamas norint apibūdinti programinius produktus, kurie leidžia visapusiškai analizuoti verslo informaciją realiuoju laiku.
- ✓ **UML** – (**Unified Modeling Language**) – vieninga modeliavimo kalba; modeliavimo ir specifikacijų kūrimo kalba, skirta specifiuoti, vaizduoti ir konstruoti programų dokumentus
- ✓ **ORACLE** – viena didžiausių ir dažniausiai naudojamų reliacinių duomenų bazių valdymo sistemų
- ✓ **MS** – (**Microsoft Corporation**) – Microsoft korporacija
- ✓ **BI** – (**Business Intelligence**) – verslo analitika
- ✓ **DB** – (**Data Base**) – duomenų bazė
- ✓ **ETL** – (**Extract, Transform, Load**) – duomenų ištraukimas iš įvairių šaltinių, jų transformavimas į reikiamus ir talpinimas į duomenų saugyklas
- ✓ **CRM** – (**Customer Relationship Management**) – ryšių su klientais valdymo sistema
- ✓ **ERP** – (**Enterprise Resource Planning**) – įmonės išteklių planavimo sistema
- ✓ **MDX** – (**Multidimensional Expressions**) – daugiamatės išraiškos
- ✓ **KPI** – (**Key Performance Indicators**) – esminiai veiklos indikatoriai
- ✓ **BIDS** – (**SQL Server Business Intelligence Development Studio**) – SQL Server verslo analitikos sistemų kūrimo aplinka (tai tas pats Visual Studio, tik pavadintas kitaip).
- ✓ **SCD** – (**Slowly Changing Dimension**) – lėtai kintanti dimensija. Tai tokia dimensija, kurios pasikeitusios atributų reikšmės neišsaugomos (užrašomos ant viršaus), išsaugomos naujoje eilutėje arba stulpelyje.
- ✓ **SSIS** – (**Microsoft SQL Server 2005 Integration Services**) – duomenų ištraukimo, transformavimo ir įkrovimo įrankis
- ✓ **SSAS** - (**Microsoft SQL Server 2005 Analysis Services**) – kubų projektavimo, apdorojimo įrankis

## 1 priedas. Vaizdų sukūrimo užklauso

```

/* Create view vDatos. */
CREATE VIEW dbo.vDatos
AS
SELECT      frf_Posting_ddt_ID AS ddt_ID_date
FROM        dbo.Fact_Project_Finance
UNION
SELECT      fhr_Task_ddt_ID
FROM        dbo.Fact_Hours
UNION
SELECT      fth_Task_ddt_ID
FROM        dbo.Fact_Task_Hours
UNION
SELECT      fte_Registration_ddt_ID
FROM        dbo.Fact_Training_Expenses

/* Create view vDim_Activity. */
/***** Object: View [dbo].[vDim_Activity] Script Date: 10/08/2008 15:12:59 *****/
CREATE VIEW dbo.vDim_Activity
AS
SELECT dct_ID, dct_Activity_Group_Code AS [Veiklos grupės kodas], dct_Activity_Group_Name AS
[Veiklos grupė], dct_Activity_Code AS [Veiklos sritis],
      dct_Activity_Name AS [Veiklos srities pavadinimas], dct_Start_Date, dct_End_Date
FROM    dbo.Dim_Activity

go

/* Create view vDim_Calculation. */
CREATE VIEW dbo.vDim_Calculation
AS
SELECT dcl_ID, dcl_Calculation_Year AS Metai, dcl_Calculation_Quarter_ID AS Ketvirtis,
      dcl_Calculation_Quarter, dcl_Calculation_Month_ID AS Mėnuo,
      dcl_Calculation_Month, dcl_Calculation_Period AS Laikotarpis,
      CONVERT(nvarchar(10), dcl_Calculation_Date, 111) AS Data
FROM    dbo.Dim_Calculation

go

/* Create view vDim_Currency. */
CREATE VIEW dbo.vDim_Currency
AS
SELECT dcr_ID, dcr_Currency_Code AS Valiuta, dcl_Currency_Name AS [Valiutos pavadinimas]
FROM    dbo.Dim_Currency

go

/* Create view vDim_Date. */
CREATE VIEW dbo.vDim_Date
AS
SELECT ddt_ID, CONVERT(nvarchar(10), ddt_Date, 111) AS Data, ddt_Year AS Metai, ddt_Quarter_ID AS
Ketvirtis, ddt_Quarter, ddt_Month_ID AS Mėnuo,
      ddt_Month, ddt_Is_Holiday, ddt_Is_Weekend
FROM    dbo.Dim_Date
WHERE   (ddt_Flag = 1)

go

/* Create view vDim_Employee. */
CREATE VIEW dbo.vDim_Employee
AS
SELECT dmp_ID, dmp_Employee_Code AS Inicialai, dmp_Name + ' ' + dmp_Surname AS [Vardas Pavardė],
      dmp_Department AS Skyrius,
      [dmp_Department*] AS [Skyrius*], dmp_Position AS Pareigos, CONVERT(nvarchar(10),
      dmp_Employment_Date, 111) AS [Įdarbinimo data],
      CONVERT(nvarchar(10), dmp_Dismissal_Date, 111) AS [Išėjimo iš darbo data]
FROM    dbo.Dim_Employee

go

/* Create view vDim_Expenses. */
CREATE VIEW dbo.vDim_Expenses
AS

```

```

SELECT dxp_Name AS [Išlaidų rūšis], dxp_Expenses_Code, dxp_ID, dxp_Description AS Aprašymas
FROM dbo.Dim_Expenses

go

/* Create view vDim_Invoice_Lines. */
CREATE VIEW dbo.vDim_Invoice_Lines
AS
SELECT dil_ID, dil_Invoice_Line_Code AS [SF eilutės], dil_Invoice_Code AS [SF numeris]
FROM dbo.Dim_Invoice_Lines

go

/* Create view vDim_Order_Lines. */
CREATE VIEW dbo.vDim_Order_Lines
AS
SELECT dol_ID, dol_Order_Line_Code AS [Užsakymo eilutės], dol_Order_Code AS [Užsakymo numeris]
FROM dbo.Dim_Order_Lines

go

/* Create view vDim_Project. */
CREATE VIEW dbo.vDim_Project
AS
SELECT      dpr_ID, dpr_Project_Code AS Projektas, CASE WHEN (dpr_Type IS NULL AND dpr_ID <> - 1)
              THEN 'Nenurodytas' ELSE dpr_Type END AS [Projekto tipas], dpr_Name AS
[Projekto pavadinimas], CONVERT(nvarchar(10), dpr_Project_Start_Date,
              111) AS [Projekto pradžios data], CONVERT(nvarchar(10),
dpr_Project_End_Date, 111) AS [Projekto pabaigos data],
              CASE WHEN (dpr_Client_Name IS NULL AND dpr_ID <> - 1) THEN 'Nenurodytas'
ELSE dpr_Client_Name END AS [Kliento pavadinimas],
              dpr_PM_Code AS [PV inicialiai], dpr_PM_Name + ' ' + dpr_PM_Surname AS [PV
Vardas Pavardė], dpr_PM_Department AS Skyrius,
              [dpr_PM_Department*] AS [Skyrius*], dpr_PM_Position AS [PV pareigos],
CONVERT(nvarchar(10), dpr_PM_Employment_Date, 111) AS [Įdarbinimo data],
              CONVERT(nvarchar(10), dpr_PM_Dismissal_Date, 111) AS [Išėjimo iš darbo
data], dpr_State AS [Projekto būseną]
FROM      dbo.Dim_Project

go

/* Create view vDim_Task. */
CREATE VIEW dbo.vDim_Task
AS
SELECT      dts_ID, dts_dpr_ID, dts_Task_ID AS Užduotis, CASE WHEN (dts_Task_Type IS NULL AND
dts_ID <> - 1)
              THEN 'Nenurodyta' ELSE dts_Task_Type END AS [Užduoties veikla], CASE WHEN
(dts_Task_Name IS NULL AND dts_ID <> - 1)
              THEN 'Nenurodyta' ELSE dts_Task_Name END AS dts_Task_Name, CASE WHEN
dts_Task_Plan_Start_Date = '9999.12.30 00:00:00' THEN NULL
              ELSE CONVERT(nvarchar(10), dts_Task_Plan_Start_Date, 111) END AS [Užduoties
pradžios data],
              CASE WHEN dts_Task_Plan_End_Date = '9999.12.30 00:00:00' THEN NULL ELSE
CONVERT(nvarchar(10), dts_Task_Plan_End_Date, 111)
              END AS [Užduoties planuojama pabaigos data], CASE WHEN dts_Task_Finish_Date
= '9999.12.30 00:00:00' THEN NULL ELSE CONVERT(nvarchar(10),
              dts_Task_Finish_Date, 111) END AS [Užduoties pabaigos data]
FROM      dbo.Dim_Task

go

/* Create view vFact_Dedication. */
CREATE VIEW dbo.vFact_Dedication
AS
SELECT fdd_ID, fdd_dmp_ID, fdd_dct_ID, fdd_Calculation_dcl_ID, fdd_Dedication_Percent
FROM      dbo.Fact_Dedication

go

/* Create view vFact_Employee_Factual_Count. */
CREATE VIEW dbo.vFact_Employee_Factual_Count
AS
SELECT fef_ID, fef_dmp_ID, fef_Calculation_dcl_ID, fef_Employee_Factual_Average AS [Vidutinis
darbuotojų skaičius Per mėnesį], fef_Flag
FROM      dbo.Fact_Employee_Factual_Count

go

/* Create view vFact_Hours. */

```

```

CREATE VIEW dbo.vFact_Hours
AS
SELECT fhr_ID, fhr_dct_ID, fhr_dmp_ID, fhr_dpr_ID, fhr_Used_Hours AS [Išdirbtos valandos],
fhr_Calculation_dcl_ID, fhr_Task_ddt_ID,
        fhr_Planned_Hours AS [Suplanuotos Valandos], fhr_Dedication_Hours AS [Dedikuotos
Valandos]
FROM dbo.Fact_Hours

go

/* Create view vFact_Invoices. */
CREATE VIEW dbo.vFact_Invoices
AS
SELECT fnv_ID, fnv_dil_ID, fnv_dcr_ID, fnv_dpr_ID, fnv_Invoice_ddt_ID, fnv_dct_ID,
fnv_Calculation_dcl_ID, fnv_Amount AS [SF suma]
FROM dbo.Fact_Invoices

go

/* Create view vFact_Orders. */
CREATE VIEW [dbo].[vFact_Orders]
AS
SELECT frd_ID, frd_dol_ID, frd_dct_ID, frd_dcr_ID, frd_dpr_ID, frd_Calculation_dcl_ID,
frd_Order_ddt_ID, frd_Amount AS [Užsakymų suma]
FROM dbo.Fact_Orders

go

/* Create view vFact_Project_Fulfilment. */
CREATE VIEW dbo.vFact_Project_Fulfilment
AS
SELECT fpf_ID, fpf_dpr_ID, fpf_Calculation_dcl_ID, fpf_Project_Fullfilment_Percent,
fpf_Project_Flag
FROM dbo.Fact_Project_Fulfilment

go

/* Create view vFact_Task_Hours. */
CREATE VIEW [dbo].[vFact_Task_Hours]
AS
SELECT fth_ID, fth_dts_ID, fth_dmp_ID, fth_Task_ddt_ID, fth_Used_Hours AS [Išdirbtos valandos],
fth_Planned_Hours AS [Suplanuotos valandos],
        fth_Lag_Days AS [Vėlavimo dienos], fth_Task_Flag, fth_Calculation_dcl_ID
FROM dbo.Fact_Task_Hours

go

/* Create view vFact_Training_Expenses. */
CREATE VIEW dbo.vFact_Training_Expenses
AS
SELECT fte_ID, fte_dxp_ID, fte_dmp_ID, fte_Registration_ddt_ID, fte_Calculation_dcl_ID,
fte_Amount AS [Išlaidų suma mokymams], fte_Flag
FROM dbo.Fact_Training_Expenses

```

## 2 priedas. Straipsnis leidinyje „Informacinės technologijos'08“

# DUOMENŲ KONCEPCINIŲ SCHEMŲ VAIZDAVIMAS SAUGYKLŲ SCHEMOMIS

Vaida Labutytė, Lina Nemuraitė

Informacijos sistemų katedra, Kauno technologijos universitetas

Studentų g. 50, LT-3031 Kaunas

Veiklos analitikos projektuose dažnai susiduriama su brangiai kainuojančiomis agregavimo klaidomis dėl neteisingai suprojektuoto daugiamačio koncepcinio modelio. Šiame straipsnyje bus apžvelgtas perėjimo iš reliacinės duomenų bazės į daugiamatę procesą, apibendrinti sudėtingiausi saugyklos projektavimo atvejai.

## 1 Įvadas

Šiandien verslo analizės paslaugų paklausa auga labai sparčiai, kadangi įmonių verslo sėkmė dinamiškoje rinkoje labai priklauso nuo laiku gaunamos, patikimos ir lengvai suprantamos verslo informacijos, kurią galima naudoti planuojant įmonės strategiją, sprendžiant kasdieninius uždavinius ir vertinant veiklos rezultatus. Verslo sąvoka suprantama plačiąja prasme, kadangi veiklos analizė svarbi bet kokio tipo organizacijai. Patikima informacija vadovui ar atsakingam darbuotojui turi būti pateikta tinkamu metu, tinkamu formatu ir tinkama forma – tik tuomet ši informacija suteiks organizacijai galimybę siekti aukščiausių rezultatų. Būtent šiam tikslui ir naudojami saugyklų kūrimo bei analizės įrankiai. Tačiau projektuojant duomenų saugyklą susiduriama su daugybę neišskumų. Nėra visų pripažinto ir naudojamo metodo, kuris padėtų išvengti su projektavimu susijusių klaidų. Jei projektavimo metu daromos klaidos, duomenų saugykla bus neefektyvi ir galima gauti klaidingus analizės rezultatus.

Straipsnyje analizuojama, kokie turėtų būti duomenų saugyklos projektavimo žingsniai, su kokiomis problemomis susiduriama vaizduojant duomenų koncepcines schemas saugyklų schemomis, nagrinėjami sprendimai, kurie leistų kurti efektyvias duomenų saugyklas.

Kuriant duomenų saugyklą svarbu suprasti, kokios savybės būdingos OLAP (angl. *Online Analytical processing*) ir OLTP (angl. *Online Transactional Processing*) sistemoms. OLAP užklauskos atliekamos daug greičiau, negu reliacinėse DB. OLAP suteikia interaktyvumą ir užklauskų lankstumą, kai bet kurią dimensiją, lygį ar narį galima peržiūrėti kaip eilutę ar stulpelį. OLAP skirta veiklos analizei, bet ne kasdieniniam transakcijų registravimui. Tačiau kad verslo intelekto (angl. *Business Intelligence*) sprendimai būtų efektyvūs, reikia, kad OLAP kubų duomenys būtų tvarkingi (angl. *clean*) ir integruoti. Šiam tikslui pasiekti reikalingi ETL (angl. *Extract Transform Load*) procesus palaikantys įrankiai, pvz., *MS SQL Server 2005 „Integration Services“*.

## 2 Daugiamatės schemos pasirinkimas

Daugiamačių duomenų modeliavimui naudojamos dviejų tipų schemos: žvaigždės arba snaigės, bet galimos ir sudėtingesnės žvaigždžių grupių schemos. Žvaigždės schema naudojama dažniausiai, jos privalumas – didelis užklauskų grietis, pasiekiamas visiškai denormalizuojant duomenų bazės lenteles. Snaigės schemose gaunamas didesnis duomenų detalumas ir šiek tiek sutaupoma vietos, tačiau sistema tampa labai sudėtinga dėl daugybės ryšių. Schemos pasirinkimą lemia tai, kaip vienas ar kitas schemos pavidalas paveiks ETL procesą. Jei vartotojai naudoja informaciją iš OLAP, patogumą ir užklauskų greitį lemia OLAP. Tai reiškia, kad nebūtina viską denormalizuoti, galima panaudoti snaigės schemos privalumus. ETL požiūriu paprasčiausia realizuoti žvaigždės schemą, ypač kai visa dimensijų informacija ateina iš vieno šaltinio. Tuomet daug paprasčiau atnaujinti pasikeitusią dimensijų lentelę, naudojant pokyčių fiksavimui skirtas ETL savybes.

Snaigės schema paprastai naudojama kai keli dimensijų atributai yra iš skirtingų šaltinių. Atskyrus šiuos atributus į skirtingas lenteles, paprasčiau sekti pasikeitimus. Projektuotojai kartais renkasi snaigės schemą ir tada, kai dimensija turi stiprią natūralią hierarchiją (pvz., kai produkto dimensija skaidosi į: kategorijas, subkategorijas ir produktus). Jei visi trys lygiai šaltinyje yra normalizuoti, tuomet gali būti lengviau valdyti ETL procesą turint dimensiją sudarytą ne iš vienos, o trijų lentelių.

## 3 Apibendrinta metodo schema

Pasak D. L. Moody ir M. A. R. Kortnik [1], norint pereiti nuo reliacinio modelio prie daugiamačio, pirmiausia reikia duomenų modelio esybes suskirstyti į transakcijų (pvz., pardavimus), komponentų (pvz., periodus, produktus, klientus) ir klasifikavimo (pvz., klientų tipai, regionai, valstijos) esybes. Antruoju žingsniu reikia identifikuoti hierarchijas (ryšius tarp dimensijų lygių). Trečiasis žingsnis – sudaryti dimensinius modelius naudojant du operatorius: hierarchijų sutraukimą (denormalizavimą) ir agregavimą (naujų esybių, vaizduojančių sumuojamus duomenis, sukūrimą). Ketvirtame žingsnyje atliekamos įvairios specifinės modifikacijos.

J. N. Mazo, J. Trujillo, J. Lechtenborger nuosekliai aprašo saugyklos projektavimo procesą ir jos užpildymą duomenimis užtikrinant daugiamatės normines formas [6]. Metodas prasideda nuo vartotojų reikalavimų identifikavimo, kur reikalavimai išreiškiami pradiniu koncepciniu daugiamačiu duomenų modeliu. Šio modelio teisingumas tikrinamas pradinių šaltinių atžvilgiu, naudojant daugiamatės normines formas. Iš daugiamačio koncepcinio modelio sudaromas loginis (ir fizinis) modeliai.



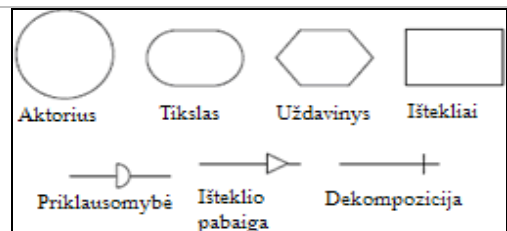
Panašiai šį procesą apibrėžia *M. Golfarelli* ir *S. Rizzi* [10]. Autoriai teigia, kad duomenų saugyklos sukūrimo procesą sudaro šeši žingsniai:

- ✓ Informacinės sistemos analizė (suprojektuojama duomenų bazės schema);
- ✓ Reikalavimų specifikuojimas (gaunami faktai);
- ✓ Konceptinis modeliavimas (sudaroma dimensinė schema);
- ✓ Dimensinės schemos teisingumo užtikrinimas;
- ✓ Loginis modeliavimas (sudaroma loginė duomenų saugyklos schema);
- ✓ Fizinis modeliavimas (sudaroma fizinė duomenų saugyklos schema).

### 3.1 Reikalavimų analizė

Duomenų saugyklos projektuojamos remiantis esamų duomenų šaltinių analize arba vartotojų reikalavimais. Pirmuoju atveju supaprastinamas ETL proceso projektavimas, tačiau vartotojo reikalavimai nustumiami į antrą planą ir kūrėjai turi per mažai informacijos, kaip identifikuoti faktus, matavimus ir dimensijas. Antruoju atveju ETL proceso projektavimas sudėtingas, bet į pirmąjį planą iškeliami vartotojų reikalavimai [8]. Pasak *J. Schiefer*, *B. List*, *R. M. Bruckner* reikalavimai duomenų saugyklai tiesiogiai veikia techninį, modeliavimo aspektą ir projekto valdymą [9]. Organizacinis modeliavimas susideda iš trijų etapų: tikslų, faktų ir atributų analizės [8].

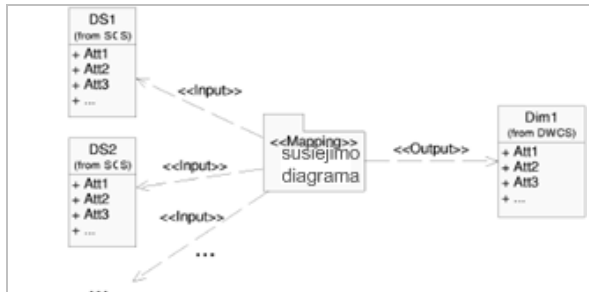
Tikslai skirstomi į strateginius, sprendimų ir informacijos lygius, kuriems vaizduoti naudojami SD (angl. *Strategic dependency*) ir SR (angl. *Strategic Rationale*) modeliai. SD modelis rodo priklausomybės ryšius tarp organizacijos aktorių, o SR – aktorių interesus. Iš šių modelių gaunamas konceptinis modelis [6]. Tikslams ir reikalavimams modeliuoti naudojami pažymėjimai, pavaizduoti 1 paveiksle.



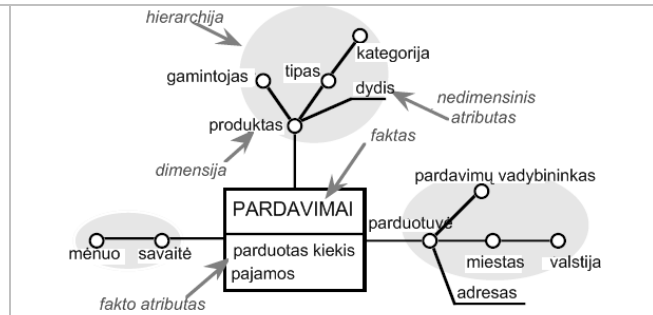
1 pav. Pažymėjimai SD ir SR modeliuose [6]

### 3.2 Konceptinis modelis

*S. Luján-Mora*, *P. Vassiliadis*, *J. Trujillo* siūlo vaizduoti duomenų saugyklos konceptinę schemą UML paketais (2, 4 pav.). Naudojant `<<Input>>`, `<<Output>>`, `<<Mapping>>` stereotipus, modelyje galima pavaizduoti ryšius tarp pradinio duomenų modelio ir daugiamatės schemos [12]. *M. Golfarelli*, *D. Maio*, *S. Rizzi* [11] siūlo vaizduoti konceptinį modelį kaip medį (3 pav.).

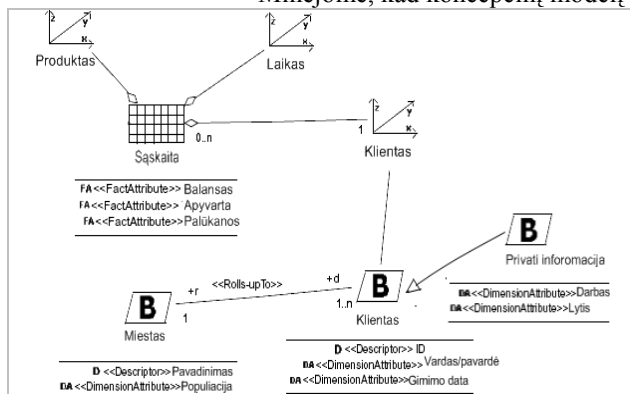


2 pav. Konceptinis modelis pagal *S. Luján-Mora*, *P. Vassiliadis*, *J. Trujillo*



3 pav. Konceptinis modelis pagal *M. Golfarelli*, *D. Maio*, *S. Rizzi*

Minėjome, kad konceptinį modelį galima gauti iš SD ir SR modelių (4 pav.).



4 pav. Pradinis konceptinis modelis pagal *J.N. Mazo'n*, *J. Trujillo*, *J. Lechtenborger*

Ikona	Stereotipas
	Faktas
	Dimensija
	Bazė (vaizduoja dimensijų hierarchijų lygius, susidedančius iš dimensijų atributų.)
<b>FA</b>	Fakto atributas
<b>DA</b>	Dimensijos atributas
<b>D</b>	Deskriptorius
	Sutraukimas (vaizduoja ryšius tarp dviejų bazių). Vaidmuo R vaizduoja kryptį, kuria hierarchija sutraukiama (angl. <i>rolls up</i> ), vaidmuo D – kryptį, kuria hierarchija analizuojama gilyn (angl. <i>drills down</i> ).

### 3.3 Sudėtingi saugyklos schemų sudarymo atvejai

Neretai modeliuojant koncepcinę schemą susiduriama su sudėtingais atvejais: *daug ir daug* ryšiais, sudėtinėmis dimensijomis, įvairiais hierarchijų atvejais, apibendrinimo ir specializavimo ryšiais, ypatingais agregavimo (sumavimo) atvejais. Toliau trumpai apibūdinami literatūroje nagrinėjami atvejai ir sprendimo būdai.

**Daug ir daug ryšiai (angl. many-to-many relationships)** [1], [2], [7], [13]. Marco Ruso atliko ypač plačią *daug ir daug* ryšių studiją. Klasikinius *daug ir daug* dimensijų ryšius jis vaizduoja be fakto faktų lentele (angl. *Factless fact table*). Kaskadiniai *daug ir daug* ryšiai atsiranda modeliuojant klausimynus ir anketas, kai dimensijos turi abipusius *daug ir daug* ryšius. Jeigu nagrinėjama matavimų grupė neturi tiesioginio ryšio su dimensija, tuomet ryšiui tarp tos dimensijos ir matavimų grupės apibrėžti naudojama tarpinė matavimų grupė. Dar kiti autoriai siūlo panaikinti susikertančią esybę, pakeisti *daug ir daug* ryšį į *vienas ir daug* ryšiu, apibrėžiant pirminį ryšį, arba įtraukti šias esybes į saugyklą. Jos gali būti naudingos analitikams ekspertams, nors jų negalima analizuoti OLAP įrankiais.

**Įvairūs hierarchijų atvejai** [3], [4], [5], [7], [14], [15], [16]. Sumavimui labai svarbu užtikrinti taisyklingas hierarchijų struktūras. Įvairiuose šaltiniuose netaisyklingos hierarchijos įvardinamos skirtingai. Tradiciškai OLAP palaiko tik *vienas ir daug* hierarchijų ryšius, kur kiekvienas žemesnio lygio narys priklauso tik vienam aukštesnio lygio nariui. Neklasikinės hierarchijos: *negriežtos* (angl. *non-strict hierarchies*); *nesubalansuotos* (angl. *unbalanced*), *nereguliaros* (angl. *irregular*), *asimetrinės* (angl. *asymmetric*) – tokios hierarchijos, kuriose hierarchijų keliai yra nevienodo ilgio. *Sudėtinės* (angl. *multiple*) – kai egzistuoja keli keliai į tą patį hierarchijos lygmenį. *Nepilnos* (angl. *ragged*) – kai tam tikri hierarchijos atributai neturi reikšmių. *Bendros* (angl. *shared*) ar *prisitaikančios* (angl. *conformed*) sutinkamos tuomet, kai keli kubai naudoja tą pačią dimensiją. Kadangi tikslinga naudoti tik griežtas hierarchijas, autoriai siūlo įvairias transformacijas, po kurių gaunamos simetrinės hierarchijos (simetriniai medžiai).

**Null reikšmių kontrolė** [6], [16]. *Null* reikšmės gali atsirasti tiek faktų, tiek dimensijų lentelėse. Kad būtų galima užtikrinti teisingus skaičiavimus, dimensijų lentelėse reikia sukurti atitinkamas „tuščias“ eilutes su tam tikra rakto reikšme (pvz., „-1“), kurios atsirastų faktų lentelėje vietoj nulinių reikšmių. Dimensijose dėl *null* reikšmių atsiranda negriežtos hierarchijos. *Null* reikšmės galima kontroliuoti naudojant alternatyvius dimensijų lygius.

**Distinct Count** [7]. Skaičiuojant *distinct count* reikšmes, tarpinė ir galutinė suma nėra sudėtos visos *distinct* reikšmės, kaip paprastos sumos atveju. Tokias situacijas siūloma spręsti MDX kalba arba kuriant tam tikras matavimų grupes. Renkantis sprendimą, svarbu atsižvelgti į laiko optimizavimą, nes SQL Server 2005 galima nustatyti *distinct count*, tačiau jo apdorojimo procesas ilgai trunka.

## 4 Išvados

Yra pasiūlyta nemažai daugiamačių schemos vaizdavimo būdų: medžio struktūra, UML, ER (ME/R), tačiau tik J.N. Mazo, J. Trujillo ir J. Lechtenborger metodas apima visą procesą nuo reikalavimų surinkimo iki galutinio daugiamačio modelio sukūrimo iš reliacinės duomenų bazės schemos. Šie autoriai pateikia visai naują norminių formų panaudojimo daugiamačiuose schemose idėją, nors ir neapibrėžia visų sudėtingų atvejų. Tuo tarpu kiti autoriai, kurie nagrinėja sudėtingus atvejus, neapibrėžia viso daugiamačių schemos sudarymo proceso.

Duomenų analizės įrankiuose atsirado pakeitimų, leidžiančių kurti neklasikines daugiamačių schemas, pavyzdžiui, *Microsoft Analysis Services 2005* (AS 2005) jau leidžia kurti tarpines, be fakto, *daug ir daug*, kelių hierarchijų dimensijas; dimensijos gali atlikti kelis vaidmenis. Tolimesnis šio tyrimo tikslas – išbandyti įvairius sudėtingus atvejus su MS SQL Server 2005, išsiaiškinti sprendimų trūkumus ir privalumus tiek sumavimo, tiek apdorojimo laiko atžvilgiu.

## Literatūra

- [1] D.L. Moody, M.A.R. Kortnik. From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. Proceedings of the International Workshop on Design and Management of Data Warehouse (DMDW'2000), Stockholm, Sweden, June 5 – 6, 2000. p. 5-1 – 5-12.
- [2] I.-Y. Song, W. Rowen, C. Medsker, E. Ewen. An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling. Proceedings of the Int. Workshop on Design and Management of Data Warehouses (DMDW 2001), Interlaken, 2001. p. 6-5 - 6-13.
- [3] A. Rozeva. Dimensional Hierarchies – Implementation in Data Warehouse Logical Scheme Design. International Conference on Computer Systems and Technologies - CompSysTech'07., 2007. p. 6.
- [4] E. Pourabbas, M. Rafanelli. Hierarchies and Relative Operators in the OLAP Environment, Journal of ACM SIGMOD Record 29(1): 32-37, 2000. p. 7.
- [5] S. Mansmann, M.H. Scholl. Extending Visual OLAP for handling irregular dimensional hierarchies. DaWaK 2006, Springer – Verlag, Heidelberg, Berlin, 2006, nr. 4081 in LNCS, p. 95 – 105.
- [6] J.N. Mazo, J. Trujillo, J. Lechtenborger. Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. Data & Knowledge Engineering 63 (2007) 725–751. Received 13 April 2007; accepted 16 April 2007, p.725-751.

- [7] M. Ruso. The many-to-many revolution. Advanced dimensional modeling with SQL Server Analysis Services 2005, Version 1.0 Revision 93 – September 25, 2006. p. 88, prieiga per internetą: <http://www.sqlbi.eu/Portals/0/Downloads/M2M%20Revolution%201.0.93.pdf>
- [8] P. Giorgini, S. Rizzi, M. Garzetti. Goal-Oriented Requirement Analysis for Data Warehouse Design. Proceedings 8th International Workshop on Data Warehousing and OLAP (DOLAP 2005), Bremen, Germany, 2005. p. 45-56.
- [9] J. Schiefer, B. List, R. M. Bruckner. A Holistic Approach for Managing Requirements of Data Warehouse Systems. 8th Americas Conf. on Information Systems, 2002. p.11.
- [10] M. Golfarelli, S. Rizzi. Designing the data Warehouse: Key Steps and Crucial Issues. Journal of Computer Science and Information Management, 2(3), 1999. p.11.
- [11] M. Golfarelli, D. Maio, S. Rizzi. Conceptual Design of Data Warehouses from E/R Schemes, International Journal of Cooperative Information Systems, 1998. p. 10.
- [12] S. Luján-Mora, P. Vassiliadis, J. Trujillo. Data Mapping Diagrams for Data Warehouse Design with UML, 23rd International Conference on Conceptual Modeling, 2004. p. 14
- [13] C. Jensen, A. Klygis, T. Pedersen, I. Timko. Multidimensional data modeling for location-based services. VLDB Journal, 2004. p. 21
- [14] S. Rizzi. Conceptual modeling solutions for the Data Warehouse, Idea Group Publishing, 2006. p. 26
- [15] A. Abelló, J. Samos, F. Saltor. Understanding Analysis Dimensions in a Multidimensional Object-Oriented Model, 3rd International Workshop on Design and Management on Data Warehouses (DWDWM). 2001. p. 9
- [16] W. Hümmel, W. Lehner, A. Bauer, L. Schlesinger. A Decathlon in Multidimensional Modeling: Open Issues and Some Solutions, [Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, September 04-06, 2002](#). p. 10

### **Transforming Conceptual Data Schemas to Data Warehouse**

Very often the expensive mistakes are made in Business Intelligence projects during the conceptual model design which leads to incorrect aggregations. The process of transforming relational database into multidimensional is analyzed in this paper, summarizing the most difficult designing cases.